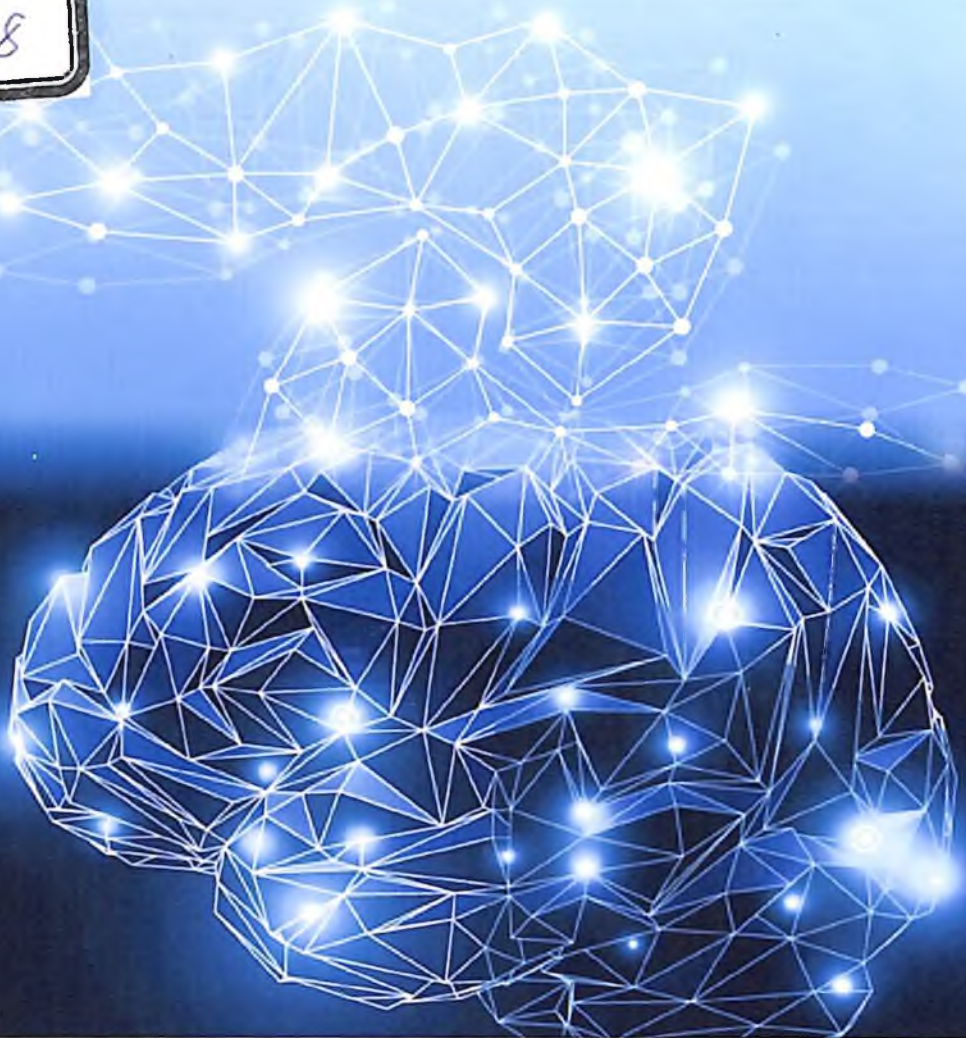


004
1498



Б.Б. МЎМИНОВ, К.Б.МУХАМАДИЕВА

СУНЪИЙ НЕЙРОН ТАРМОҚЛАРИ ТАСНИФИ

ЎЗБЕКИСТОН РЕСПУБЛИКАСИ
ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ

ЎЗБЕКИСТОН РЕСПУБЛИКАСИ
АХБОРОТ ТЕХНОЛОГИЯЛАРИ ВА КОММУНИКАЦИЯЛАРИНИ
РИВОЖЛАНТИРИШ ВАЗИРЛИГИ

МУҲАММАД АЛ-ХОРАЗМИЙ НОМИДАГИ
ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ

Б.Б. МЎМИНОВ, К.Б.МУХАМАДИЕВА

СУНЪИЙ НЕЙРОН ТАРМОҚЛАРИ ТАСНИФИ

(Монография)

ТОШКЕНТ – 2020

УЎК: 523.24
КБК: 22.652.8
М 267

Мўминов Б. Б., Мухамадиева К.Б. Сунъий нейрон тармоқлари таснифи. —энциклопедия - монография. —Т.: «Aloqachi», 2020. -228 б.

ISBN 978-9943-6396-6-9

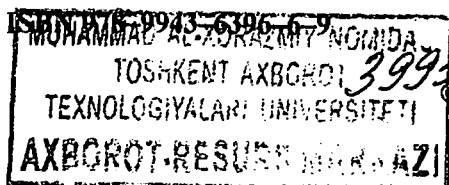
Мазкур монографияда сунъий нейрон тармоқлари асосида маълумотларга ишлов бериш йўналишида иш олиб бораётган изланувчилар учун сунъий нейрон тармоқлари борасида тушунча ҳосил қилиш, улар кўриниши билан танишиш, ишлаш тамойиллари, уларнинг имкониятлари, қўллаш соҳаси бўйича тушунарли ва ихчам тилда маълумот олишлари мумкин. Шунингдек, сунъий нейрон тармоқларининг моделлари, усуллари ва алгоритмлари, қўллаш услублари келтирилган.

УЎК: 523.24
КБК: 22.652.8

Такризчилар:

- Н.А.Маматов** – ТАТУ қошида АКТ Илмий-Инновацион Маркази, етакчи илмий ходим, т. ф. д.;
- Н.О.Рахимов** – Мухаммад ал-Хоразмий номидаги Тошкент ахборот технологиялари университети, Тизимли ва амалий дастурлаш кафедраси мудири, т. ф. д.

Монография Тошкент ахборот технологиялар университети Илмий техник кенгашининг қарори билан нашрга тавсия этилди.



© «Aloqachi» нашриёти, 2020.

Ї БОБ. СУЊИЙ НЕЙРОН ТАРМОҚЛАРИ НАЗАРИЙ АСОСЛАРИ

1.1. СуЊий интеллектнинг ривожланиш тарихи

Инсон онгига ўхшаш фикрлаш қобилиятига эга суЊий онгни яратиш борасидаги ғоя анча йиллар олдин пайдо бўлган эди. 1235-1315 йилларда Р.Луллий XIV асрларда турли масалаларни мулоҳазалар асосида ечадиган машинани яратишга ҳаракат қилди.

XVIII асрларда *Г.Лейбниц* (1646 - 1716) ва *Р.Декарт* (1596- 1650) бир – биридан холис ҳолда универсал тилни ишлаб чиқиш ғоясини олға сурди. Ушбу ғоялар суЊий интеллектнинг назарий асоси бўлиб қолди.

СуЊий интеллектни яратиш йўналиши биринчи ЭҲМ лар яратилгандан сўнг, XX асрнинг 40- йилларида кучайиб кетди. Шу вақтнинг ўзида *И.Винер* (1894- 1964) янги йўналиш фани – кибернетика бўйича ўзининг тадқиқотларини олиб борди.

СуЊий интеллект тушунчаси (artificial intelligence) 1956-йил Станфорд университети (АҚШ) семинарида биринчи марта эълон қилинди. Ушбу семинар ҳисоблаш ишларига эмас балки, мантиқий ишланмаларга йўналтирилган эди. СуЊий интеллект янги фан йўналиши сифатида тан олингандан кейин у асосий иккита йўналишга ажратилди: нейро кибернетика ва «қора яшиқ» кибернетикаси. Фақат ҳозирга келиб улар битта ягона йўналиш сифатида бирлаштирилди.

Нейрокибернетика асосий ғоясини қуйидагича ифодалаш мумкин. Фикрлаш қобилиятига ягона объект – бу инсон онгидир, шунинг учун барча «фикрлаш» имкониятига эга қурилма айнан онг фаолиятини такрорлаши лозимдир.

Демак, нейрокибернетика инсон миёси структурасига ўхшаш модел структурасининг аппарат кўринишга йўналтирилган. Физиологлар томонидан шу нарса аниқлаштирилганки, инсон миёси асосини ўзаро боғланган нерв ҳужайралари – нейронлар ташкил қилади. Шунинг учун нейрокибернетиклар иши нейронларга мос бўлган элементларни яратиш ва уларни ягона тизимга бирлаштиришга

йўналтирилган. Бундай тизимлар **нейрон тармоқлар** ёки **нейротармоқлар** деб ном олган.

Энг биринчи нейротармоқлар 50-йиллар охирида америкалик олимлар *Г.Розенблат* ва *П.Мак-Кигюк* томонидан ишлаб чиқилган. Бу инсон кўзи модели ва унинг инсон мияси билан ҳамкорликда ишлашини таъминловчи тизимни яратишга ҳаракат эди. Улар яратган қурилма **перцептрон** деган номни олди. У алифбе ҳарфларни фарқлаш қобилиятига эга бўлиб, бироқ улар ёзилишига таъсир қиларди. Масалан, *А*, *А* ва *А* ҳарфлар ёзилиши учта турли белги сифатида қабул қилинар эди. 70-80 йилларга келиб, сунъий интеллект йўналиши бўйича изланишлар сони камайиб борди. Энг биринчи натижалар самарасиз бўлиб чиқди, муаллифлар фикрича ўша давр учун компьютерлар хотира ҳажми ва иш кучи тезлиги етарлича бўлмаганлиги сабаб бўлди.

Бироқ, 80-йиллар ўрталарида Японияда янги авлод компьютерларини ишлаб чиқариш соҳасида билимлари асосида ишловчи нейрокомпьютерлари ишлаб чиқилди. Ўша даврда хотира ҳажми ва тезкорлик кўрсаткичларидаги чекланмалар йўқолиб, **транспьютерлар** -пайдо бўлди, катта ҳажмдаги процессорларга эга параллел компьютерлар ва транспьютерлардан **нейрокомпьютерлар** - инсон мияси стурктураси моделини яратишга бир қадам қолган эди. Нейрокомпьютерларни қўллашнинг асосий соҳаси - бу образларни аниқлашда қўллаш ҳисобланди.

Ҳозирги кунда нейротармоқларни яратишнинг учта йўналиши қўлланилмоқда:

аппарат – махсус компьютерларнинг, кенгайтма платалари, микросхемалар тўплами яратилиши бўлиб, улар қўйилган алгоритм бўйича ишлашга йўналтирилади;

дастурий - юқори тезкорликка эга компьютерлар учун дастурлар ва инструментларни яратиш. Нейротармоқлар компьютер хотирасида яратилади, барча амалларни эса уларнинг процессорлари амалга оширади;

гибрид – икки қисмни мувофиқлашуви. Ҳисоблашларнинг бир қисмини махсус кенгайтирилган платалар амалга оширса, (сопроцессорлар), қолган қисмини – дастурий воситалар.

«Қора ящик» кибернетикаси асосида нейрокибернетика асосига қарама-қарши бўлган принцип ётади. Бунда «фикрловчи» қурилма қандай жойлашганлигига аҳамиятсиз ҳолда, асосий инсон ақли каби берилганларни тўғри қайта ишлаши муҳим ҳисобланади.

Сунъий интеллектнинг ушбу йўналиши мавжуд компьютер моделларида интеллектуал масалаларни ечиш алгоритмларини излаб топишга йўналтирилган. 1956 -1963 йилларда инсон фикрлаш қобилиятининг моделлари ва алгоритмларини аниқлаш ва биринчи дастурларни ишлаб чиқиш ишлари олиб борилган. Бироқ, мавжуд фанлардан бирортаси- фалсафа, психология, лингвистика – айнан алгоритмни таклиф эта олмайди. Турли йўналишларда тадқиқотлар олиб борилган.

50-йиллар охирида лабиринтли излаш модели яратилди. Бу йўналишда масалани маълум бир график кўринишда келтирилиб, ҳолатлар муҳитини ифодалаб, ушбу графика асосида кирувчи берилганлар ва олинадиган натижаларгача бўлган йўлларнинг оптималини излаб топиш имконини беради. Ушбу моделни ишлаб чиқиш бўйича анча ишлар олиб борилган, бироқ амалий масалаларни ечишда бу усул ўз аксини топмади.

60 –йиллар бошида – **эвристик дастурлаш даври** бўлиб, **эвристика** – назарий жиҳатдан асосланмаган қонуният, бироқ излаш кенг муҳитида ҳолатлар сонини қисқартириш имконини беради. Эвристик дастурлаш – олдиндан берилган эвристика асосида стратегик амалларни ишлаб чиқиш деган маънони англатади.

1963- 1970 йилларда масалаларни ечишда математик мантик усулларини қўллаш ишлари бошланди. Қонуниятлардан фойдаланиш усули, яъни мавжуд аксиомалар негизида теоремаларнинг исботланиши асосида 1973 йил **Пролог** дастурлаш тили яратилди.

Сунъий интеллект амалий дастурлаш соҳасида 70 йиллар ўртасида аҳамиятли ўзгариш юз берди, бунда фикрлаш универсал алгоритминини

излаш ўрнига мутахассис-экспертлар аниқ билимларини моделлаштириш ғояси пайдо бўлди. АКШда билимларга таянган ҳолда ишловчи тижорат тизимлари ёки бошқача қилиб айтганда, **эксперт тизимлар** яратилди. Сунъий интеллект масалаларини ечишнинг янги ёндашуви – **билимларни ифодалаш** усули кириб келди. Тиббиёт ва кимё йўналишлари учун классик бўлиб қолган эксперт тизимлари - MYCIN ва DENDRAL яратилди. Интеллектуал технологиялар ривожлантиришнинг бир неча глобал лойиҳалари таклиф қилинган - ESPRIT (Европа Иттифоқи). DARPA (АКШ мудофаа вазирлиги), V авлод машиналари бўйича япон лойиҳаси ҳисобланади.

80-йиллар ўртасидан бошлаб сунъий интеллект тижорат лойиҳаларда қўлланила бошланди. Йиллар давомида эксперт тизимлари ишлаб чиқаришга катта ҳажмдаги маблағ ажратилиб, ўрганиш қобилиятига эга сунъий интеллект тизимлари изланишлар олиб борилмоқда ва ишлаб чиқилмоқда.

1.2.Сунъий интеллект ривожланиш босқичлари

1-босқич (50–йиллар) (Нейрон ва нейрон тармоқлар)

Бу давр кетма-кет амалларни бажарувчи ҳозирги вақт учун ўртача қувватга ва маълум бир ресурсли хотира, тезкорликка, масалалар ечимига эга бўлган машиналар пайдо бўлиши билан боғлиқ. Ушбу масалалар фақат ҳисоблаш амаллари билан боғлиқ бўлиб, бунинг учун масала ечими схемаси маълум бир расмий дастурлаш тилида изоҳланган. Бундай масалаларга адаптация масалалари киради.

2-босқич (60- йиллар)(Эвристик изланиш)

Машина «интеллект» қисмига маълумотларни излаш, саралаш, оддий усулларда умумлаштириш механизмлари қўшилиб, унда қайта ишланадиган маълумотлар моҳият жиҳатдан боғлиқсиздир. Айнан шу давр инсон фаолиятини автоматизациялаш масалаларини ҳал қилиш ва ривожлантириш учун туртки нуқтаси бўлиб хизмат қилди.

3-босқич (70- йиллар)(Билимларни ифодалаш)

Бу даврда олимлар учун айнан масалалар ечими янги алгоритмларини синтез қилиш учун билимлар (уларнинг ҳажми ва моҳияти) муҳимлиги аниқланди. Ушбу билимлар математик нуқтаи назардан изоҳга эга бўлмасдан, тўпланган тажриба асосида олинган билимнинг расмий хусусиятига эга бўлмасдан, фақат изоҳлар тўплами сифатида шаклланган билимлар инобатга олинмоқда. Бу билимлар турли соҳа мутахассислари, шифокор, химиклар, тадқиқотчилар ва бошқалар эришган ютуқлар, билимлар тўпламидир. Ушбу билимлар эксперт билимлар деб ном олган бўлиб, улар асосида ишловчи тизимлар консультант (маслаҳатчи) ёки эксперт тизимлар деб номланди.

4-босқич (80- йиллар)(билим оловчи машиналар)

СИ ривожланишининг тўртинчи босқичи илғор қадам билан олға сурилишга туртки бўлди. Эксперт тизимлар пайдо бўлганидан бошлаб, интеллектуал технологиялар ривожланишининг янги босқичи – интеллектуал тизимлар-консультантлар эраси бошланди, бунга қўйилган масала ечим йўллари кўрсатиб, уларни асослаб беради, ўрганиш ва билим соҳасини кенгайтириш имкониятига эга, инсон билан чекланган табиий тилда мулоқот қилиш имконига эга тизимлардир.

5-босқич (90- йиллар) (маълумотларни қайта ишлашнинг автоматизациялашган марказлари)

Алоқа тизимларининг ва улар ёрдамида бажариладиган масалаларнинг мураккаблашуви дастурий таъминотнинг «интеллектуаллигини» оширувчи даражага ўтишни тақозо этди, бунда тизимдан четдан туриб ҳуқуқсиз фойдаланишдан ҳимоялаш, ахборот ресурсларини ҳимоялаш, таҳдидларни олдини олиш, таҳлил ўтказиш ва керакли маълумотни излаш ва ҳ.к. ишларни таъминлаши керак.

Ҳимоя тизимларини яратишнинг янги кўриниши бу интеллектуал тизимларни қўллаш бўлиб қолди. Айнан улар асосида осон ўзгартириш мумкин бўлган муҳитни яратиш ва керакли масалалар ечимини таъминлаш мумкин.

6-босқич (2000- йиллар)(Робототехника)

Роботларни қўллаш соҳаси жуда кенг бўлиб, оддий атрофни тозаловчи роботлардан бошлаб, то замонавий ҳарбий ва космик техника кўринишларга эга бўлган. Моделлар навигация тизимлари ва периферияли датчиклар ўрнатилган.

7-босқич (2008-йиллар)(Сингулярлик)

Сунъий интеллект ва роботларни яратувчи сунъий интеллектларнинг яратилиши, инсонларнинг ҳисоблаш машиналари билан интеграцияланиши, инсон ақлий фаолиятининг биотехнология ҳисобидан юқори кўрсаткичга оширилиши шу босқич ютуқлари ҳисобланади.

Олимлар башорат қилишларича технологик сингулярлик 2030 йилларда етиши мумкин, технологик сингулярлик назариясини қўллаб қувватловчи мутахассисларнинг фикрича инсон онгидан фарқли ўлароқ онг яратилса, цивилизациянинг келажагини инсон ҳаракати белгиламай қолади.

1.3.Сунъий интеллект ривожланиш йўналишлари

Сунъий интеллект – бу информатиканинг бир қисми бўлиб, асосий моҳияти аппарат дастурий воситаларни ишлаб чиқишга йўналтирилган ва дастурчи бўлмаган фойдаланувчига масалани қўйиш ва уни ечиш имконини беради. Бунда интеллектуал масалаларни белгиланган тилда ЭҲМ ёрдамида ҳал қилиниши тушунилади.

1.3.1.Билимларга асосланган тизимларни ишлаб чиқиш ва уларни ифодалаш

Бу сунъий интеллектнинг асосий йўналиши бўлиб, билимларни ифодалаш моделини, маълумотлар базасини яратиш ва улар асосида эксперт тизим ишлаб чиқиш билан боғлиқ. Охирги йилларда билимларни структуралаш, ажратиб олиш модели ва усуллари ишлаб чиқилмоқда ва улар билимларни структурали қайта ишлашга олиб келмоқда.

1.3.2.Ўйин ва ижод

Сунъий интеллект одатий тусда интеллектуал ўйин масалаларни - шахмат, шашка ва бошқаларни ўз ичига олади. Унинг асосида бошланғич йўналишдаги ёндашув – яъни лабиринт модель ва эвристика ётади. Ҳозирда бу ёндашув тижорат йўналиш, чунки илмий жиҳатдан бу ёндашув самарасиз ҳисобланади.

1.3.3.Табиий тилда ва машинали таржима тизимни ишлаб чиқиш

50-йилларда сунъий интеллект тадқиқот йўналишларидан бири машинали таржима соҳаси бўлиб келди. Энг биринчи таржимон дастур – инглиз тилидан рус тилига таржима қилувчи бўлиб, ғоя бўйича сўзма-сўз таржима ва у самарасиз ҳисобланди. Ҳозирги замонда мураккаброқ модель қўлланилмоқда, бунда табиий тиллардаги маълумотлар анализ ва синтез асосида таржима қилинади. Таҳлил жараёнида қуйидагилар амалга оширилади:

Морфологик анализ – матнда сўзларнинг анализи;

Синтаксик анализ – гапларнинг грамматикаси ва сўзлар ўртасидаги боғланиш анализи;

Семантик анализ – ҳар бир гапни моҳиятга- йўналтирилган билимлар базасига асосланган ҳолда маъносини таҳлил қилиш;

Прагматик анализ – хусусий билимлар базасига асосланган ҳолда гаплар маъносини контекстли анализ қилиш ҳисобланса, синтез эса худди шу амалларнинг ўзгача кетма-кетликда бажарилиши ҳисобланади.

1.3.4.Тасвирларнинг аниқланиши

Сунъий интеллектнинг кенг тарқалган йўналиши бўлиб, тарихий асосни ҳосил қилади. Бунда ҳар бир объектга хоссалар матрицаси белгиланиб, у асосида образни аниқлаш ишлари олиб борилади. Ушбу йўналиш машинали ўрганишга жуда яқин бўлиб нейрокибернетика билан яқин боғланган.

1.3.5.Компьютернинг янги архитектураси

Айнан шу йўналиш янги аппарат архитектураларни ишлаб чиқиш ва ундан мантиқий ва символли маълумотларни қайта ишлашга йўналтирилган. Пролог- ва Лисп-машиналарни, V ва VI авлод компьютерлар яратилади. Сўнгги компьютерлар асосан маълумотлар базаси ва параллел компьютерларини ишлаб чиқишга йўналтирилган.

1.3.6.Интеллектуал роботлар

Роботлар – бу электромеханик машиналар бўлиб, инсон меҳнатини автоматизациялаш учун мўлжалланган.

Роботларни яратиш ғояси азалдан келиб чиққан бўлиб, иборанинг ўзи 20-йилларда пайдо бўлган, унинг муаллифи – чех ёзувчиси Карел Чапек. Ўша даврдан бошлаб бир неча робот авлодлари яратилди.

Қаттиқ схемали бошқарувга эга роботлар. Замоनावий ишлаб чиқариш соҳасида қўлланилиб келинаётган роботлар биринчи авлод роботлари ҳисобланади, улар дастурланадиган манипуляторлардир.

Сенсорли қурилмага эга адаптив роботлар. Уларнинг намунавий кўриниши мавжуд, бироқ ишлаб чиқаришда қўлланилмайди.

Ўз-ўзини бошқарадиган ёки интеллектли роботлар. Бу робототехниканинг сўнгги ҳолати бўлиб, унда асосий муаммо – бу машина ахборот қабул қилиш («кўриш») қобилиятини ҳосил қилиш ҳисобланади

1.3.7.Махсус дастурий таъминот

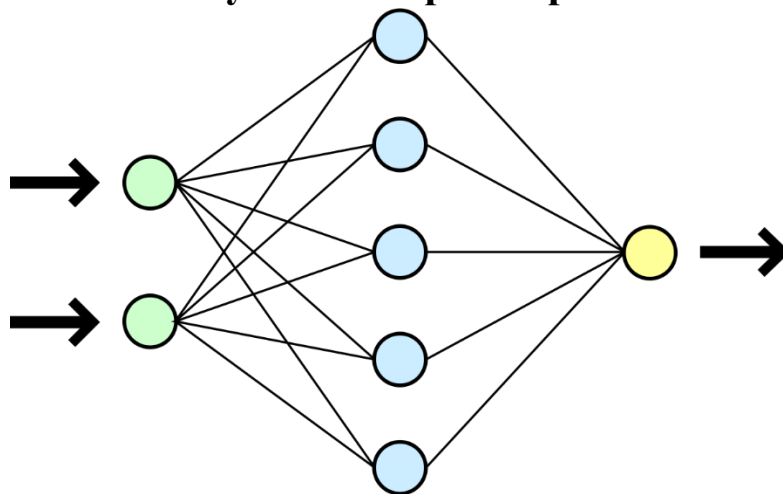
Ушбу йўналиш учун махсус дастурлаш тиллари ишлаб чиқилган бўлиб, айнан ҳисоблаш қонуниятларига бўйсунмайдиган масалаларни ечишга йўналтирилган. Ушбу тиллар маълумотларни символли қайта ишлашга йўналтирилган бўлиб, уларга - LISP, PROLOG, SMALLTALK, РЕФАЛ ва бошқалар киради. Шу билан бирга амалий

дастурлар пакети ишлаб чиқилмоқда, улар ишлаб чиқаришда қўлланиладиган интеллектуал тизимларни ёки сунъий интеллект дастурий инструменти учун амалий дастурлар пакети ишлаб чиқилмоқда, масалан, KEE, ARTS. Энг кенг тарқалган кўриниши бу бўш эксперт тизим ёки «қобик»лар бўлиб, масалан, VXSYS, MI ва бошқалар, улар таркибини билимлар базаси билан тўлдириб, турли кўринишдаги тизимларни яратиши мумкин.

1.3.8. Ўрганувчи ва мустақил ўрганувчи

Сунъий интеллект фаол ривожланаётган соҳа бўлиб, маълумотларни таҳлил қилган ҳолда билимларни автоматик тўплаб бориш ва умумлаштиришга йўналтирилган модуллар, усуллар ва алгоритмлар киради. Келтирилган мисоллар асосида ўрганиб бориш ва киритилаётган образларни таҳлил қилган ҳолда аниқлаш усуллари ўз ичига олади.

1.4. Сунъий нейрон тармоғи



1-расм. Содда нейротармоқ схемаси

Сунъий нейрон тармоқ (СНТ) – дастурий ва аппарат кўринишга келтириладиган математик модель бўлиб, биологик нейронлар тармоғи ишлаш принципи асосида яратилган. Энг биринчи нейрон тармоқлар У. Маккалок ва У. Питтс томонидан яратилган тармоқ биринчи қадамлар эди. Ўрганиш алгоритмлари ишлаб чиқилгандан сўнг яратилган моделлар амалиётда, яъни образларни аниқлашда

башорат қилиш, бошқарувда ва бошқа масалаларда қўлланила бошланди.

СНТ ўзаро боғланган ҳамкорликда ишлайдиган содда процессорлар (сунъий нейронлар) тизимини ҳосил қилади. Одатда бу процессорлар компьютер процессорларига нисбатан жуда содда ҳисобланади. Ҳар бир процессор фақат сигналлар билан ишлаб, бу сигналларни у ҳосил қилади ва у даврий равишда бошқа процессорларга узатиб туради. Маълум бир бошқарувга эга йирик тармоққа уланганлиги сабаб, ушбу содда процессорлар жуда мураккаб масалаларни ҳамжихатликда ечиш имконини беради

Машинали ўрганиш соҳаси бўйича қаралганда нейрон тармоқ образларни аниқлашда, дискриминант таҳлилда, Кластерлаш усулларида хусусий ҳол бўлиб ҳисобланади.

- Математика нуқтаи назаридан нейрон тармоқлар бу кўп параметрли чизиқсиз оптимизация масаласи ҳисобланади.

- Кибернетика нуқтаи назаридан нейрон тармоқлар адаптив бошқарув масалаларида ва робототехникада қўлланилади.

- Ҳисоблаш техникаси ва дастурлаш соҳаси ривожланиши нуқтаи назаридан қараганда нейрон тармоқ – самарали паралеллик масалаларини ҳал қилиш усули ҳисобланади.

- Сунъий интеллект нуқтаи назаридан СНТ коннективизм фалсафасининг асоси ҳисобланади ва табиий интеллектни моделлаштиришда компьютерли алгоритмларни қўллаш бўйича асосий йўналиш бўлиб келади.

Нейрон тармоқлар маълум маънода дастурланмайди, балки улар ўрганиш жараёнини ўтади. Ўрганиш фаолияти - нейрон тармоқларнинг алгоритмлашда асосий афзаллиги ҳисобланади. Ўрганиш жараёни - техник нуқтаи назаридан нейронлар ўртасида коэффициентларни аниқлаш ҳисобланади. Ўрганиш жараёнида нейрон тармоқга кирувчи ва чиқувчи маълумотлар ўртасида мураккаб боғланишларни аниқлаб топиш, ҳамда умумлаштириш имконияти мавжуд. Натижада, самарали ўрганиш асосида янги маълумотлар,

ноаниқ маълумотлар, бузилган маълумотлар қабул қилинганда тўғри жавоб қайтариш имконини беради.

1.4.1.Нейрон тармоқларни қўллаш соҳалари. Образларни аниқлаш. Тасниф масалалари

Образлар сифатида турли кўринишдаги объектлар: матн символлари, тасвирлар, товушли объектлар ва ҳ.к. Тармоқ ўрганиш жараёнида турли кўринишдаги образлар намунаси уларнинг қайси синфга мансублиги билан бирга берилади. Намуна, одатда хусусиятлар кўрсаткичлари вектори кўринишда ифодаланилади. Бунда барча хусусиятлар бирлашмаси келтирилган намуна синфини белгилаши лозим. Хусусиятлар етарлича бўлмаса, келтирилган намуна бир неча синфларга мансуб деб қабул қилинади. Тармоқ ўрганиш жараёни тугаганидан сўнг янги образлар ифодаланганда у мансуб бўлган синф аниқланиб берилади.

Бу кўринишдаги тармоқ топологияси чиқиш қатламида нейронлар сони аниқланган синфлар сонига тенг бўлади. Бунда нейрон тармоқ чиқиш қисми ва ифодаланилаётган синф ўртасида мослашув ўрнатилади. Тармоқга маълум бир образ ифодаланганда унинг чиқиш қисмига образ мансуб бўлган синфни белгиловчи маълумот пайдо бўлиши лозим, шу билан бирга қолган чиқиш қисмларида унга мос бўлган синфга мансуб эмаслиги тўғрисида маълумот пайдо бўлиши лозим. Агар бирдан чиқиш қисмларда айнан унга мос синфга мансублиги борасида маълумот пайдо бўлса у ҳолда тармоқ жавоби «ишончсиз» деб қабул қилинади.

Янги нейрон тармоқли архитектуралари яратилгандан бошлаб, улар кўплиги туфайли барчасини ўрганиб чиқиш, тартиблаш жуда мураккаб бўлиб келмоқда.

Нейрон тармоқларнинг график кўринишлари маълум бўлган архитектурани изоҳи келтирилган (2-расмга қаранг).

1.4.2. Нейрон тармоқ таснифи

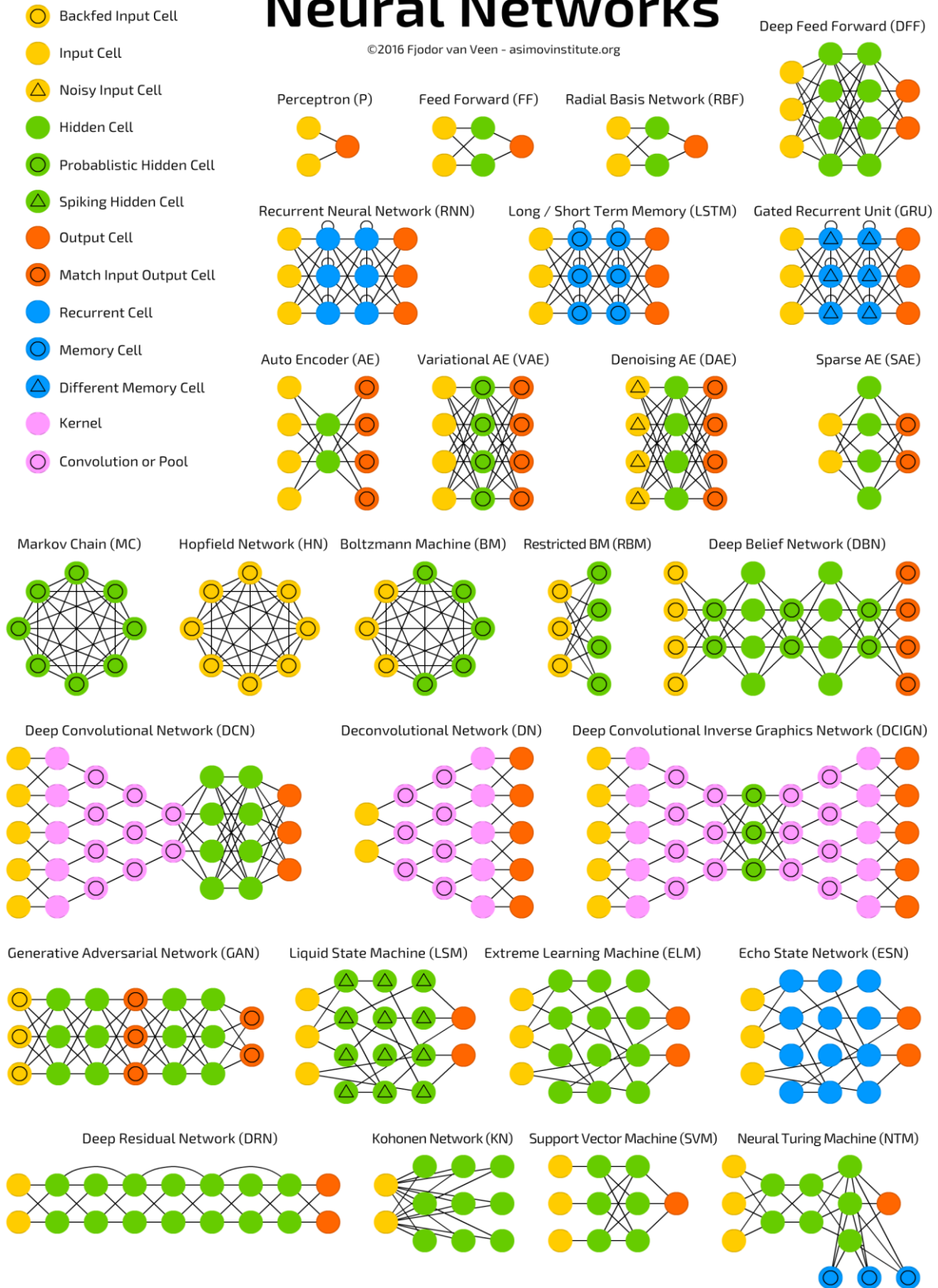
Нейрон тармоқларнинг харитасини тузишда асосий ечилмаган масала бу уларни қўллаш ҳолатини ифодаламаслигидир, масалан, (VAE) вариацион автошифраторлар, (AE) автошифраторлар каби кўриниши мумкин, бироқ ўрганиш жараёни мутлақо бошқа ҳисобланади. Ўргатилган тармоқлар кўринишларида қўллаш ҳолатлари жуда фарқланади, чунки бу ҳолатларда VAE генератор сифатида хизмат қилади, сигнал янги тармоқлар ҳосил қилинади. AE да эса кирувчи маълумотларни ўзи «хотирасида» мавжуд намуналар билан таққослаган ҳолда қабул қилади ва ўрганади.

Шуни таъкидлаш лозимки, барча қўлланиладиган аббревиатура умумий ҳолда қабул қилинган бўлсада, барчасини бирдек қабул қилиб бўлмайди. Масалан, RNN кўпинча рекурсив нейрон тармоқ деб қабул қилишади, бироқ кўпинча рекуррент нейрон тармоқларни белгилайди. Шу билан бирга RNN кўп такрорланадиган архитектура тармоқларда қўшимча тармоқ ҳолатда қўлланилишини учратиш мумкин, бунга LSTM, GRU ва икки томонлама йўналтирилган ҳолатлар киради. Худди шунга ўхшаш AE тармоқлар ҳам мавжуд, уларда ҳам VAE, DAE ва бошқаларни умумлашган ҳолда AE деб номлашади бироқ иш принципи фарқланади. Кўп аббревиатуралар «N» ҳарфи сони билан ҳам фарқланади, масалан, Convolutional Neural Network сўзида нейрон сўзини ишлатмасак CNN аббревиатура CN деб номланади, бироқ моҳият ўзгармайди.

Келтирилган архитектура рўйхатини сўнгги деб қабул қилишни тавсия этмаймиз. Чунки кундан кунга янги кўринишдаги архитектуралар яратилмоқда, улар ҳақида тўлиқ маълумот топиш мураккаб бўлиб қолади, шу сабабли ушбу рўйхатни СИ дунёси ҳақида маълум бир маълумот олиш кўрсатмаси деб қабул қилишингиз лозим бўлади.

A mostly complete chart of
Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org



2-расм. Асосий нейрон тармоқларнинг график кўринишлари

Жадвал 1. Ўрганиш хусусияти бўйича нейрон тармоқлар таснифи

УСУЛ	ИЗОҲ
Ўқитувчи ёрдамида ўрганиш	Компьютерга кирувчи маълумотлар ва тахминий чиқувчи натижалар «ўқитувчи» ёрдамида ифодаланилади. Асосий мақсад умумий кирувчи ва чиқувчи маълумотлар ўртасида умумий қонуниятни аниқлаш.
Ўқитувчисиз ўрганиш	Ўрганиш жараёнида алгоритмга кутилаётган натижа ифодаланилмайди, балки алгоритмнинг ўзи чиқувчи натижани белгилайди. Ўқитувчисиз ўрганиш асосий мақсад (маълумотлар ўртасида яширин қонуниятни аниқлаш) бўлиб қолади.
Мустахамлаш орқали ўрганиш	Компьютер дастур динамик муҳит билан ўзаро боғланган ҳолда маълум бир масалани бажариши лозим бўлади, бунда ўқитувчи дастур мақсадга яқинлашганлик даражасини белгиламайди.

Жадвал 2. Маълумотларни таҳлил қилиш бўйича нейрон тармоқлар таснифи

КАТЕГОРИЯ	ИЗОҲ
Тасниф	Ўқитувчи ёрдамида ўрганиш алгоритмлари кириш қисмига узатилаётган маълумотлар тегишли бўлган синфни аниқлайди.
Кластерлаш	Ўқитувчисиз ўрганиш алгоритмлари тури бўлиб, кирувчи маълумотлар бир ёки бир неча кластерларга тақсимланади.
Регрессия	Ўқитувчи ёрдамида ўрганиш алгоритмлари тури бўлиб, чиқувчи қийматлар узлуксиз ҳисобланади.

КАТЕГОРИЯ	ИЗОҲ
Ўлчовини камайтириш	Кирувчи маълумотларни қисқартирилган фазовий ўлчамда ифодалаш усулида соддалаштириш алгоритмлари.

Жадвал 3. Нейрон тармоқлар моделлари

АЛГОРИТМ	УСУЛ	КАТЕГОРИЯ	ИЗОҲ
Ordinary Least Squares Regression (OLSR)	Ўқитувчи ёрдамида ўрганиш	Регрессия	Чизиқли регрессия алгоритми.
Linear Regression	Ўқитувчи ёрдамида ўрганиш	Регрессия	Башорат қилишда чизиқли функцияни қўлловчи регрессия алгоритми синфи.
Logistic Regression	Ўқитувчи ёрдамида ўрганиш	Регрессия	Эҳтимоллик моделига асосланган регрессия алгоритми.
Stepwise Regression	Ўқитувчи ёрдамида ўрганиш	Регрессия	Регрессия алгоритми
Multivariate Adaptive Regression	Ўқитувчи ёрдамида ўрганиш	Регрессия	Чизиқли бўлмаган ҳолатларни ечишда

АЛГОРИТМ	УСУЛ	КАТЕГОРИЯ	ИЗОҲ
Splines (MARS)			қўлланиладиган регрессия алгоритми.
Locally Estimated Scatterplot Smoothing (LOESS)	Ўқитувчи ёрдамида ўрганиш	Регрессия	Башорат этиш учун қўлланиладиган силлиқ текисликни қурувчи регрессия алгоритми.
k-Nearest Neighbour (kNN)	Ўқитувчи ёрдамида ўрганиш	Регрессия, тасниф	Тасниф ва регрессия турига мансуб алгоритм бўлиб, натижани башорат этиши учун хусусиятлар текислигида k масофадаги қўшни маълумотларни қўллайди.
Learning Vector Quantization (LVQ)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Берилган намуналар асосида ўқитувчи ёрдамида ўрганиш усулини қўлловчи алгоритм таснифи.

АЛГОРИТМ	УСУЛ	КАТЕГОРИЯ	ИЗОҲ
Self-Organizing Map (SOM)	Ўқитувчи-сиз ўрганиш	Тасниф	Рақобатли ўрганиш усулини қўлловчи сунъий нейрон тармоқ тури.
Locally Weighted Learning (LWL)	Ўқитувчи ёрдамида ўрганиш	Регрессия, тасниф	Суст ўрганиш алгоритми
Ridge Regression	Ўқитувчи ёрдамида ўрганиш	Регрессия	Тихонов созланмаси деб номланади.
Least Absolute Shrinkage and Selection Operator (LASSO)	Ўқитувчи ёрдамида ўрганиш	Регрессия	Хусусиятларни ажратиб олиш ва назорат этиш регрессия алгоритми.
Elastic Net	Ўқитувчи ёрдамида ўрганиш	Регрессия	Турли кўринишдаги бошқарув усулларини мувофиқлаштирувчи регрессия алгоритми
Least-Angle Regression (LARS)	Ўқитувчи ёрдамида ўрганиш	Регрессия	Катта ҳажмдаги маълумотларга мўлжалланган регрессия алгоритми.

АЛГОРИТМ	УСУЛ	КАТЕГОРИЯ	ИЗОҲ
Classification and Regression Tree (CART)	Ўқитувчи ёрдамида ўрганиш	Регрессия, тасниф	Тасниф ва регрессия масалаларини ечишда дарахтсимон моделни қўлловчи рекурсив алгоритм.
Iterative Dichotomiser 3 (ID3)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Табиий тилда берилган маълумотларни қайта ишлаш масалаларида кенг қўлланувчи дарахтсимон алгоритм.
C4.5 and C5.0	Ўқитувчи ёрдамида ўрганиш	Тасниф	Маълумотларни энтропиялашга мўлжалланган дарахтсимон алгоритм.
Chi-squared Automatic Interaction Detection (CHAID)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Маркетинг соҳасида кенг қўлланиладиган дарахтсимон алгоритм кўриниши.

АЛГОРИТМ	УСУЛ	КАТЕГОРИЯ	ИЗОҲ
Decision Stump	Ўқитувчи ёрдамида ўрганиш	Тасниф	Дарахтсимон масалаларни ечиш кўриниши.
M5	Ўқитувчи ёрдамида ўрганиш	Регрессия, тасниф	Дарахтсимон масалаларни ечиш кўриниши.
Conditional Decision Trees	Ўқитувчи ёрдамида ўрганиш	Тасниф	Дарахтсимон масалаларни ечиш кўриниши.
Naive Bayes	Ўқитувчи ёрдамида ўрганиш	Тасниф	Содда тасодифий классификатор.
Gaussian Naive Bayes	Ўқитувчи ёрдамида ўрганиш	Тасниф	Содда тасодифий классификатор.
Multinomial Naive Bayes	Ўқитувчи ёрдамида ўрганиш	Тасниф	Матнли ҳужжатларга мўлжалланган Байес тармоқ кўриниши
Averaged One- Dependence Estimators (AODE)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Тасодифий классификатор.

АЛГОРИТМ	УСУЛ	КАТЕГОРИЯ	ИЗОҲ
Bayesian Belief Network (BBN)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Нейрон тармоқ кўриниши бўлиб, тасодифийликни ҳисоблаш учун қўлланиши мумкин.
Random Forest	Ўқитувчи ёрдамида ўрганиш	Регрессия, тасниф	Регрессия ва таснифлаш амалларини мувофиқлаштирган ҳолда ўрганиш усули.
k-Means	Ўқитувчисиз ўрганиш	Кластерлаш	Ўқитувчисиз ўрганишнинг содда алгоритмларидан бири.
k-Medians	Ўқитувчисиз ўрганиш	Кластерлаш	Кластерлаш алгоритми.
Expectation Maximisation (EM)	Ўқитувчисиз ўрганиш	Кластерлаш	Биологик маълумотларни қайта ишлашда кенг қўлланилади,

АЛГОРИТМ	УСУЛ	КАТЕГОРИЯ	ИЗОҲ
Hierarchical Clustering	Ўқитувчи-сиз ўрганиш	Кластерлаш	Кластерлар иерархиясини яратувчи кластерлаш алгоритми
Perceptron	Ўқитувчи ёрдамида ўрганиш	Регрессия, тасниф	Сунъий нейрон тармоқнинг содда тури
Back-Propagation	Ўқитувчи ёрдамида ўрганиш	Регрессия, тасниф	Сунъий нейрон тармоқнинг ўрганиш алгоритми.
Hopfield Network	Ўқитувчи ёрдамида ўрганиш	Тасниф	Сунъий нейрон тармоқнинг рекуррент кўриниши. Ассоциатив хотира.
Radial Basis Function Network (RBFN)	Ўқитувчи ёрдамида ўрганиш	Регрессия, тасниф	Сунъий нейрон тармоқнинг кўриниши бўлиб, радиал базавий функцияларни қўллайди ва вақт кўрсаткичли башоратларда қўлланилади.

АЛГОРИТМ	УСУЛ	КАТЕГОРИЯ	ИЗОҲ
Deep Boltzmann Machine (DBM)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Тасодифий Яширин ўзгарувчилардан иборат қатламларга эга бинар тасодифий Марков кетма-кетлигига ўхшаш Больцман машинаси.
Deep Belief Networks (DBN)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Чуқур нейрон тармоқнинг кўриниши.
Convolutional Neural Network (CNN)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Жонзотларнинг кўриш системасидан олинган структурага эга сунъий нейрон тармоқ кўриниши.
Stacked Auto-Шифраторс	Ўқитувчи-сиз ўрганиш	Тасниф	Ўқитувчисиз ўрганиш учун сунъий нейрон тармоқ кўриниши.
Principal Component Analysis (PCA)	Ўқитувчи-сиз ўрганиш	Ўлчовини камайтириш	Ўлчовни камайтириш алгоритми.

АЛГОРИТМ	УСУЛ	КАТЕГОРИЯ	ИЗОҲ
Principal Component Regression (PCR)	Ўқитувчи ёрдамида ўрганиш	Регрессия	Асосий компонентлар усулига асосланган Регрессия алгоритми.
Partial Least Squares Regression (PLSR)	Ўқитувчи ёрдамида ўрганиш	Регрессия	Асосий компонентлар регрессияси билан боғлиқ бўлган регрессия алгоритми.
Sammon Mapping	Ўқитувчи-сиз ўрганиш	Ўлчовини камайтириш	Ўлчовни камайтириш алгоритми.
Multidimensional Scaling (MDS)	Ўқитувчи-сиз ўрганиш	Ўлчовини камайтириш	Ўлчовни камайтириш алгоритми.
Projection Pursuit	Ўқитувчи-сиз ўрганиш	Ўлчовини камайтириш	Ўлчовни камайтириш алгоритми.
Linear Discriminant Analysis (LDA)	Ўқитувчи-сиз ўрганиш	Ўлчовини камайтириш	Ўлчовни камайтириш алгоритми.

АЛГОРИТМ	УСУЛ	КАТЕГОРИЯ	ИЗОҲ
Mixture Discriminant Analysis (MDA)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Аралаш кўринишдаги моделларга асосланган тасниф усули.
Quadratic Discriminant Analysis (QDA)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Аралаш кўринишдаги моделларга асосланган тасниф усули.
Flexible Discriminant Analysis (FDA)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Тасниф усули.
Boosting	Ўқитувчи ёрдамида ўрганиш	Тасниф	Машинали ўрганиш моделларига башорат аниқлигини ошириш мақсадида қўлланиладиган алгоритмлар.

АЛГОРИТМ	УСУЛ	КАТЕГОРИЯ	ИЗОҲ
Bootstrap aggregating (Bagging)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Машинали ўрганиш алгоритмлари аниқлик ва барқарорликни ошириш мақсадида яратилган алгоритм.
AdaBoost	Ўқитувчи ёрдамида ўрганиш	Тасниф	Бустинг алгоритми.
Stacked Generalization (blending)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Турли кўринишдаги машинали ўрганиш моделларини комбинациялаш учун қўлланилади.
Gradient Boosting Machines (GBM)	Ўқитувчи ёрдамида ўрганиш	Тасниф	Содда башорат қилувчи моделлар кўринишда таснифлаш ёндашувини таклиф этади.
Gradient Boosted Regression Trees (GBRT)	Ўқитувчи ёрдамида ўрганиш	Регрессия	Содда башорат қилувчи моделлар кўринишлар

АЛГОРИТМ	УСУЛ	КАТЕГОРИЯ	ИЗОҲ
			асосида башорат этувчи моделни таклиф этувчи регрессия модель.

Жадвал 4. “Компьютерли ўрганиш” масалаларини дастурлашда кенг қўлланиладиган дастурий кутубхоналар рўйхати

КУТУБХОНА	ДАСТУРЛАШТИЛИ	ИЗОҲ
Shogun	C++	Ушбу махсус пакет машинали ўрганиш учун мўлжалланган кенг камровли объектлар ва ўрганиш созланмаларига эга бўлган пакет бўлиб, маълумотларни таснифлаш, регрессия ёки чуқур ўрганишга мўлжалланган.
Weka	Java	Умумий қўлланиладиган пакет.
Kernlab	R	Ядро асосида таснифлаш ва ўлчовини камайтириш.
Dlib	C++	Портларга ажратиш, таҳрирлаш.
NLTK	Python	Чизикли регрессия, тартиблаш, таснифлаш.
Orange	Python	Янги фойдаланувчилар ва экспертлар учун маълумотларни визуаллаштириш ва таҳлил этиш учун очиқ кодли кутубхона. Катта ҳажмдаги ускуналар ёрдамида

КУТУБХОНА	ДАСТУРЛАШ ТИЛИ	ИЗОҲ
		интерактив жараён таъминланади.
Java-ml	Java	Машинали ўрганиш алгоритмининг коллекцияси.
pyML	C++; Python	Машинали ўрганиш учун Python тилида ёзилган интерактив объектга-мўлжалланган муҳит. Ядрога асосланган SVMга йўналтирилган PyML бошқа усуллар. Кутубхона Linux ва Mac OS X томонидан қўлланилади.
Mlpy	Python	Mlpy Python учун модул ҳисобланиб, NumPy/SciPy ва GNU Scientific кутубхонаси учун созланма ҳисобланади.
Pybrain	Python	pybrain - Python-Based Reinforcement Learning нинг қисқартмаси бўлиб, Artificial Intelligence and Neural Network Library (Python тилида ёзилган кутубхона, мустаҳкамлаш асосида ўрганиш, сунъий интеллект ва сунъий нейрон тармоқлар).
Torch	C++; Lua	Машинали ўрганиш ускуналари тўпламидан иборат ускуналарнинг илмий тўплами.

КУТУБХОНА	ДАСТУРЛАШ ТИЛИ	ИЗОҲ
scikit-learn	Python; Cython	Кенг қўлланиладиган кутубхона. Имконияти кенг ва қўлланишда содда ҳисобланади.
Theano	Python	GPU усули ёрдамида ишлайдиган самарали ҳисоблаш кутубхонаси. Чуқур ўрганишда қулайлик яратади.
Pylearn2	Python	Theano яратилган машинали ўрганиш ускуналарининг тўплами.
MDP	Python	Маълумотларни қайта ишлаш учун ускуналар модулли тўплами.
Spark	Java	Катта ҳажмдаги маълумотларни қайта ишлаш учун тезкор ва универсал макет.
Mahout	Java	Hadoop асосида яратилган машинали ўрганиш муҳити.
Mallet	Java	Табиий тил соҳасида статистик қайта ишлаш учун Java пакети.
JSAT	Java	Машинали ўрганиш ва статистик қайта ишлаш ускуналарнинг java пакети.
Accord.NET	.NET	.NET асосида илмий ҳисоблашларга мўлжалланган пакет.

КУТУБУХОНА	ДАСТУРЛАШ ТИЛИ	ИЗОҲ
Vowpal Wabbit	C++	Тез ўрганиш учун мўлжалланган, BSD лицензияси асосида ишлаб чиқилган.
MultiBoost	C++	Бустинг алгоритмларини қўлловчи C++ да яратилган пакет.
TensorFlow	Python, C++	Очиқ кодли Google кутубхонаси.

1.5.Тўғри чизикли нейрон тармоқлари

(Feed forward neural networks, FF or FFNN) ва перцептронлар (perceptrons, P) энг содда чизикли тақсимланган тармоқ бўлиб, улар кириш қисмидан чиқиш қисмига узатади. Нейрон тармоқлар одатда кириш қатлами нейронларидан, яширин қатлам нейронларидан ва чиқиш қатлами нейронларидан иборат. Бир қатлам нейронлари ўзаро боғланишга эга бўлмайди. Бироқ ҳар бир қатлам нейрони қўшни қатлам нейрони билан боғланган бўлади. Энг содда тармоқ иккита кириш ва битта чиқиш нейронидан иборат бўлади, у оддий мантиқий ифодани изоҳлаши мумкин. FFNN да одатда тескари боғланишда хатоликни аниқлаш усулида ўрганиш олиб борилади, яъни кириш қатлами жуфтлигига кутилаётган натижа узатилиб модель яратилади. Хатолик деганда, кутилаётган чиқувчи маълумотларнинг кирувчи маълумотларга нисбатан номувофиқлиги тушунилади. (масалан, ўртакватрат қийматнинг фарқланиши). Тармоқда етарлича яширин қатлам нейронлар мавжуд бўлса, у ҳолда кирувчи ва чиқувчи берилганлар ўртасида боғланишни ўрнатиши мумкин. Амалда эса бу кўринишдаги чизикли тақсимланган тармоқлардан фойдаланиш фақат бошқа кўринишдаги тармоқлар билан ҳамкорликда қўлланилади.

1943 йили «Нерв фаоллиги билан боғлиқ фикрларни мантиқий аниқлаш» мақоласида У. Мак-Каллок ва У. Питтс сунъий нейрон тармоқ тушунчасини биринчи марта олға сурди. Улар томонидан сунъий нейрон модели тақлиф этилди. 1949 йили Д. Хебб ўзининг «Ҳолатни ташкил этиш» номли ишида нейронларни ўқитишнинг асосий принципларини изоҳлади.

Бир неча йил ўтгач бу ғоялар асосида Америкалик нейрофизиолог Фрэнк Розенблатт инсон онги моделини ташкил қилувчи қурилма схемасини таклиф этди ва унга перцептрон деб ном берди. Перцептрон электромеханик хотира ячейкасида фотоэлементлардан сигналларни узатиб, сенсорли майдонни ҳосил қилади. Ячейкалар коннективизм асосида тасодифий ўзаро боғланади. Перцептрон образларни таснифлашини «ўрганиш» учун махсус итерацион усул ишлаб чиқилган бўлиб, инсон ўрганиш жараёни каби ўз хатоликларини тўғриланган ҳолда ўрганиш усули қўлланилган. Бундан ташқари у ёки бу ҳарфни аниқлаш учун перцептрон ҳарф ўзига хос хусусиятини статистик усулда ажратиб билган, бироқ индивидуал аҳамиятсиз қисмлари бундан мустасно. Демак шу усулда перцептрон турли кўринишда ёзилган ҳарфларни битта ягона образга умумлаштириш қобилиятига эга бўлган. Шундай бўлсада перцептрон имкониятлари чекланган, бир томони кўринмас бўлган ҳарфларни, ёки ҳажми жиҳатдан тўғри келмайдиган, бурилган, сурилган ҳарфларни ўрганган ҳарфлари билан солиштирган ҳолда ишончли аниқлаб билмаган.[7]

Перцептрон яратилишининг асосий мақсади образларни аниқлаш машинасини яратиш эмас, балки инсон онгини ишлаш принципининг моделини яратиш, яъни интеллект ишини чуқур ўрганиш ва тадбиқ этиш ҳисобланади.

Элементар перцептрон 3 турдаги элементлардан: S-элементлар, A-элементлар и битта R-элементдан иборат. S-элементлар бу – рецепторлар қатлами. Ушбу рецепторлар A-элементлар билан қўзғалиш ҳолатида боғланади. Ҳар бир рецептор икки ҳолатдан бирида бўлиши мумкин, бу тинч ва қўзғалишдир. A-элементлар чегараланган қийматда сумматор кўринишда ифодаланади. Яъни, рецепторлардан келувчи қўзғалишлар даражасининг йиғиндиси Қўзғалган A – элементлар сигналлари сумматорга R узатилади, бунда i-элементдан келувчи сигнал w_i коэффициент билан узатилади[10].

Кўриб чиқилган оддий элементларнинг барчаси содда ҳисобланади, чунки улар кескин ўзгариб турувчи функцияларни қўллайди. Мураккаб масалаларни ҳал қилиш учун бошқа кўринишдаги функциялардан фойдаланишни талаб этади, масалан чизиқли функциялар.

Натижада Розенблатт қуйидаги ғояларни олға сурди:

Перцептрон бу S-, A-, R-элементлардан иборат тармоқ бўлиб, ўзаро таъсирни таъминловчи ўзгарувчи матрицага W (элементлари W_{ij} – оғирлик коэффициентлари)га эга ва у тармоқнинг олдинги фаоллик ҳолати билан аниқланади;

Кетма-кетлик боғланишларга эга перцептрон бу S-элементга яқин бир элементдан d мантиқий масофада S-элементга яқин $d+1$ мантиқий масофада жойлашган элемент билан тугайдиган тизимга айтилади;

Кесишмали боғланишга эга перцептрон бир турдаги (S, A ёки R) элементлар ўртасида боғланиш мавжуд бўлган тизим бўлиб, ушбу элементлар S-элементлардан бир хил мантиқий масофада жойлашган, қолган боғланишлар эса кетма-кет кўринишда тескари боғланишли перцептрон мантиқий узоқ бўлган элементдан мантиқий яқин бўлган элементгача боғланиш бўлган тизимга айтилади;

Оддий перцептрон деб кўйидаги талабларга жавоб берувчи тизимга айтилади:

тизимда фақат битта R-элемент мавжуд (табиийки улар барча A-элемент оғирликлари билан боғланган);

кетма-кет боғланишга эга перцептрон бўлиб, у фақат S-элементдан A-элементга ва A-элементдан R-элементга ҳаракатланган;

S-элементлардан A-элементларгача бўлган барча боғланишларнинг (S–A боғланишлар) оғирликлари ўзгармасдир;

ҳар бир боғланишнинг узатиш вақти ёки нолга тенг ёки белгиланган ўзгармас τ тенг бўлади;

барча S-, A-, R-элементларни фаоллаштирувчи функциялар кўйидаги кўринишга тенг

$$U_i(t)=f(a_i(t)),$$

бунда $a_i(t)$ – u_i элементлар кириш қисмига бир вақтнинг ўзида келувчи барча сигналларнинг алгебраик йиғиндисидир.

Элементар перцептрон бу содда перцептрон бўлиб, унда барча элементлар – содда. Бу ҳолатда фаоллаштириш функцияси

$$C_{ij}(t)=U_i(t-\tau)W_{ij}(t)$$

кўринишга эга.

Олимлар перцептронларнинг хусусиятларини инобатга олган ҳолда 5та синфга ажратган:

Диаметри бўйича чекланган перцептрон – хусусий мантиққа эга аниқланган X ўлчам, маълум бир белгиланган қийматдан ошмайди.

Чекланган кетма-кетликка эга перцептрон– ҳар бир хусусий боғланиш чекланган сондаги X нуқтадан иборат.

Гамба перцептрони – ҳар бир хусусий боғланиш чегараланган чизикли функция кўринишида, яъни мини – перцептрон бўлиши лозим.

Тасодифий перцептронлар – чекланган перцептронлар бўлиб, хусусий боғланишлар тасодифий мантикий (bool) функциялар кўринишида бўлади.

Чекланган перцептрон – хусусий боғланишлар тўплами чексиз, бироқ аниқ қийматлар a_i кетма-кетлиги эса чекланган.

Ўрганиш алгоритми

Перцептронни ўрганишнинг классик усули – бу хатоликни тўғрилаш усули. Бу шундай усулки, жорий перцептрон таъсирланиши белгиланган ҳолатда бўлгунга қадар боғланиш оғирлик кўрсаткичи ўзгармас қолади. Перцептрон таъсирланишида ўзгариш сезилганда боғланиш оғирлик кўрсаткичи 1 га айланади, хатолик кўрсаткичига мос ҳолда унинг аксига ўзгаради (+/-).

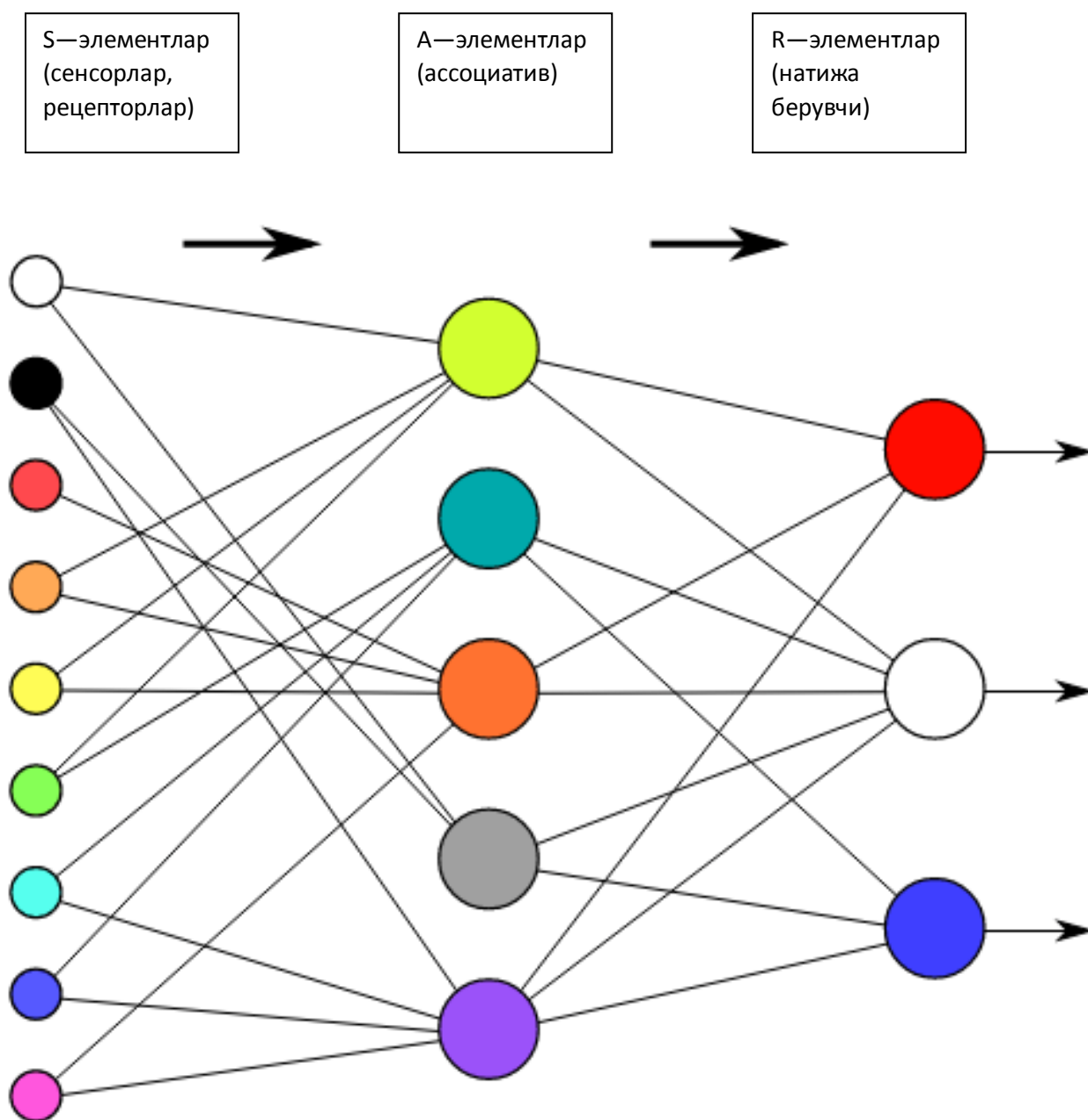
Кўйидаги чизмага эътибор киламиз:

Перцептрон маълумотнинг онг орқали қабул қилиш модели, S, A BA R элементлардан иборат		
БИТТА ЯШИРИН ҚАТЛАМГА ЭГА ПЕРЦЕПТРОН 1 қатлам S элементлар 2 қатлам A элементлар 3 қатлам R элементлар	БИР ҚАТЛАМЛИ ПЕРЦЕПТРОН ҳар бир S га битта A S-A боғланиш =+1 чегара A= +1	Розенблатт кўп қатламли перцептрон >1 қатлам A элемент Розенблатт кўп қатламли перцептрон махсус алгоритм бўйича S-A ўрганиш

3-расм. Перцептрон берилганларни қабул қилиш модели

Битта яширин қатламли перцептрон биттадан S, A, R қатламлардан иборат. Қатламлардаги ҳар бир нейрон кейинги қатлам ҳар бир нейрони билан боғлиқ.

Бир қатламли перцептрон биттадан S, A, R қатламлардан иборат, бироқ ҳар бир S нейронга бир A нейрон боғланган, S-A боғланиш оғирлик кўрсаткичи +1 га тенг, A чекланмаси (порог)+1 га тенг.



4- расм. Перцептрон тармоғи.

Кўп қатламли перцептрон икки кўринишга эга: Розенблатт кўп қатламли перцептрон ва Румельхарт кўп қатламли перцептрон.

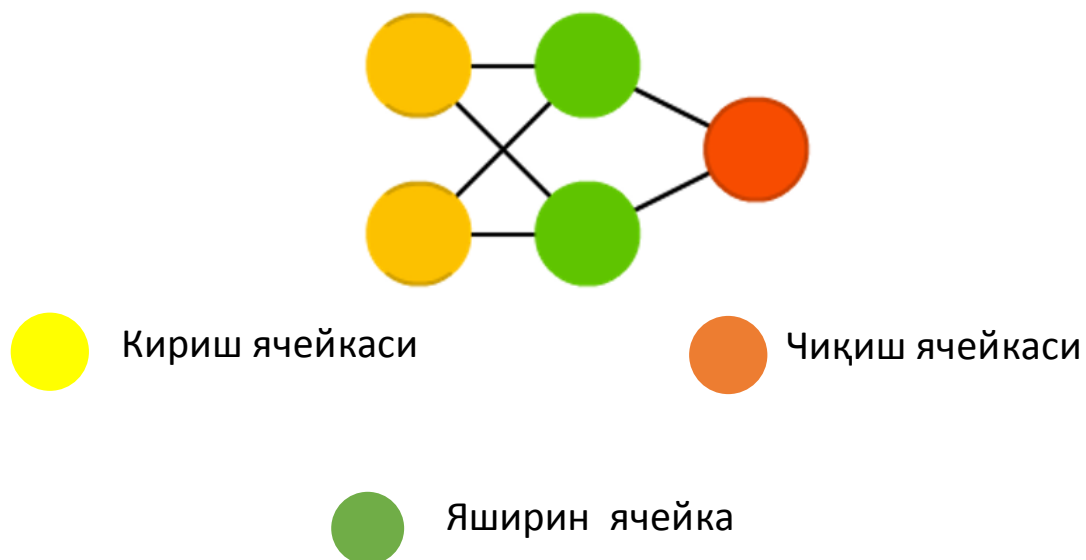
Розенблатт кўп қатламли перцептрон биттадан ортиқ А-элементлар қатламига эга.

Румельхарт кўп қатламли перцептрони бу Розенблатт кўп қатламли перцептроннинг хусусий ҳолати бўлиб, икки хусусиятга эга:

1. S-A боғланишлар ихтиёрий оғирликка эга бўлиши мумкин ва A-R боғланиш билан биргаликда ўрганиши мумкин.

2. Ўрганиш махсус алгоритм бўйича олиб борилиб, тескари боғланиш орқали хатоликни текшириш усули деб номланади.

1.6. Радиал-асосли функцияга эга тармоқлар (RBF)



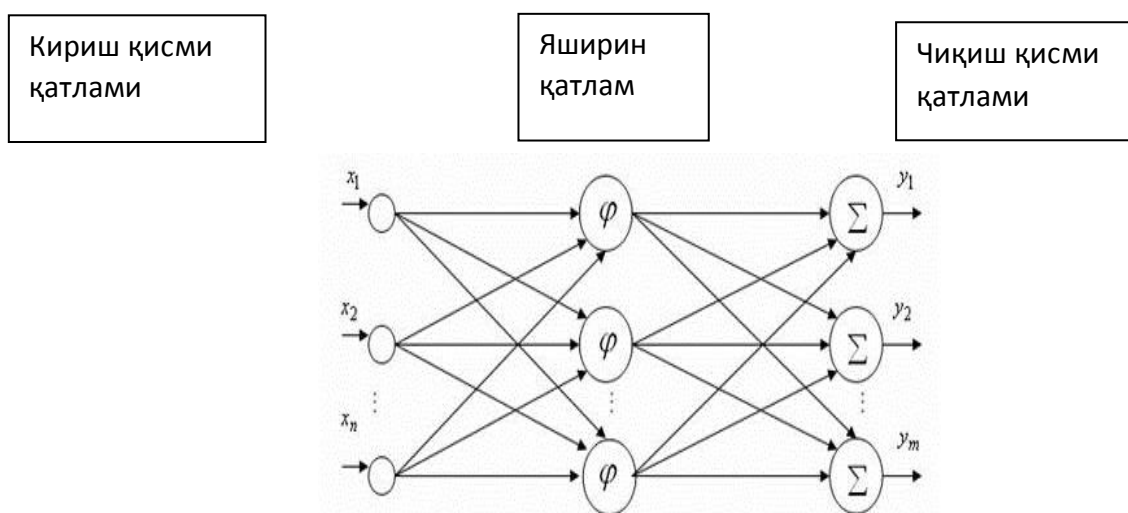
5-расм. Радиал-асосли функцияга эга тармоқ (RBF)

Радиал-асосли функцияга эга тармоқлар (RBF) фаоллаштириш функцияси нур кўринишда бўлган FFNN ҳисобланади. Айнан шу функцияга эга FFNN ўз номига эга бўлиб қолди.

RBF тармоқлар бир қатор афзалликларга эга бўлиб, тўғри тақсимланадиган кўп қатламли тармоқлар билан ўхшашликлар бўлсада битта оралиқдаги яширин қатлам ёрдамида чизиқли бўлмаган барча функцияларни ифодалаш мумкин. Бу эса қатламлар сонини қисқаришига олиб келади. Шу билан бир қаторда чиқиш қатламида қўлланиладиган чизиқли кетма-кетликларни

оптималлаштириш усулида оптималлаштириш имкони мавжуд. Локал маълумотларнинг минималлиги тескари боғланишда хатоликни аниқлаш усулини қўллашда вақт сарфланишини олдини олади. Шу сабабли **RBF тармоқлар** тескари боғланишда хатоликни аниқлаш усулини қўлловчи бошқа тармоқларга нисбатан **сезиларли даражада тезкор ишлайди**.

RBF тармоқларнинг камчиликлари: тармоқ маълумотлари экстраполяцияга берилмайди шу сабабли кириш векторлари катта ҳажмда бўлганда тармоқ йириклашади.



6 –расм. RBF тармоғи

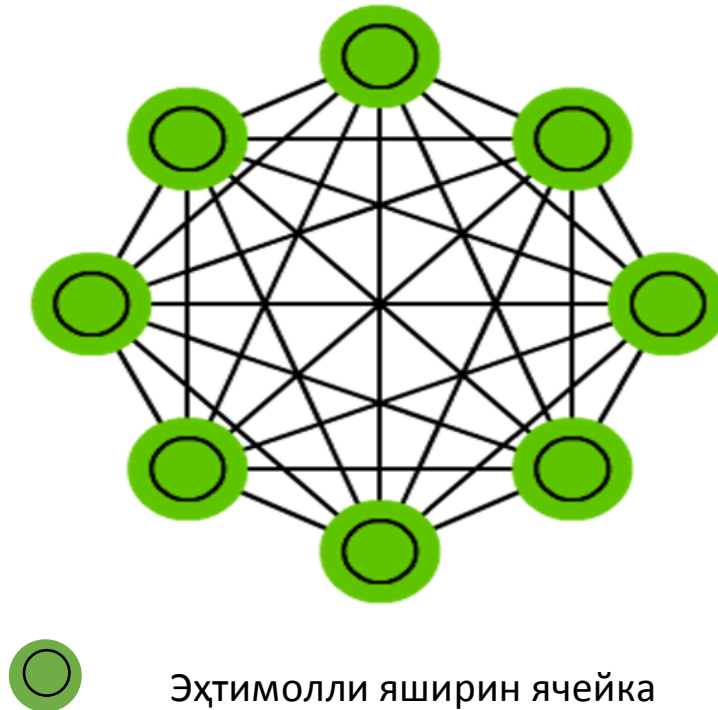
RBF тармоғининг содда кўриниши учта қатламдан иборат: оддий кириш қатлами бўлиб, биринчи қатлам оғирликларни белгилаш учун бошланғич маълумотларни тақсимлаш; радиал-симметрик фаоллаштириш функциясига эга яширин қатлами. Унда ҳар бир j - функция алоҳида эталон векторларни $\mathbf{w}_j^{(h)}$ оғирликлар вектори кўринишда сақлайди; чиқиш қатлами. **RBF** тармоғини яратиш учун қўйидаги шартларни бажариш лозим.

Биринчидан, эталоннинг мавжудлиги, яширин қатлам нейронлар вектори кўринишда ифодаланиш.

Иккинчидан эталон ва кириш вектор ўртасидаги масофани ўлчаш усулининг мавжудлиги. Одатда бу стандарт евклид масофа.

Учинчидан яширин қатлам нейронларни фаоллаштиришнинг махсус функциясининг мавжудлиги, у эса масофани ўлчаш усулини танлайди. Одатда Гаусс функцияси қўлланилиб, кучайтириш имконини беради.

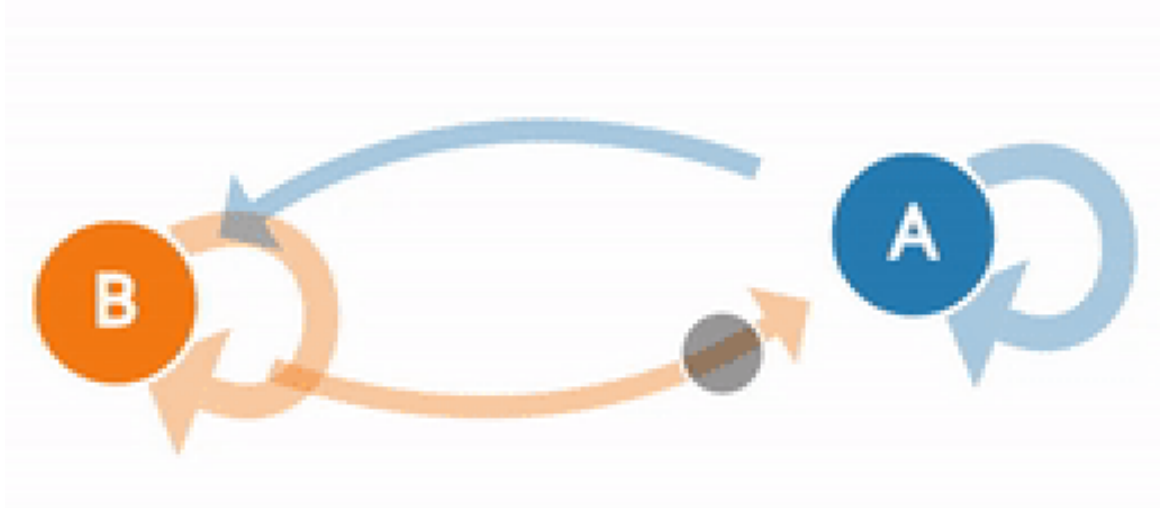
1.7.Марков занжири



7-расм. Марков занжири.

(Markov Chains, MC ёки discrete time Markov Chain, DTMC) – Больцман (BM) машинаси ва Хопфилда (HN) тармоқлари олдинги кўриниши бўлиб, жорий ҳолатдан қўшни ҳолатга ўтиш эҳтимоллик занжири кўрсатилади. Бундан ташқари занжир ўз хотирасига эга бўлмасдан, занжирдаги кейинги ҳолат жорий ҳолатдан келиб чиққан ҳолда эришилади, жорийдан олдинги бўлган ҳолатга ҳеч қандай боғланиш мавжуд бўлмайди. Марков занжирини нейрон тармоқ деб номлаб бўлмасда, уларга яқин бўлиб, Больцман (BM) машинаси ва Хопфилда (HN) тармоқларининг назарий асосини шакллантиради. Марков занжири ҳамма вақт ҳам тўлиқ ўзаро боғланган занжирни ҳосил қилмайди.

Марков занжири – амаллар кетма-кетлиги бўлиб, кейинги амал олдинги амал билан чамбарчас боғлиқ.



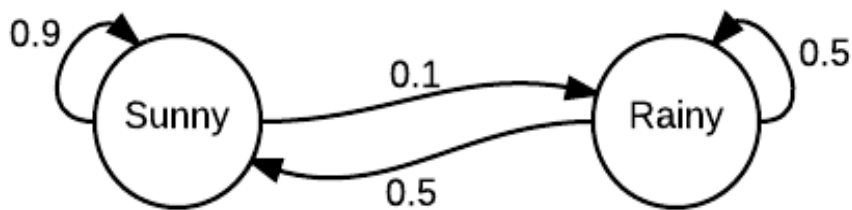
8-расм. Марков занжири ишлаш принципи.

Марков занжири – тасодифий амаллар кечишининг содда моделини ҳосил қилади. Турли соҳаларда, матнларни генерациялашдан бошлаб, молиявий моделлаштиришгача қўлланилиб келинмоқда, булардан энг таниқлиси бу Subreddit Simulator. Ушбу ҳолатда Марков занжирининг барча subreddit учун контент яратилишини автоматизациялашга йўналтирилган.

Марков занжири жуда ихчам ва содда бўлиб, бирон бир математик ёки статистик концепцияларни қўллашни талаб этмайди. Марков занжири эҳтимолли моделлаштиришни яратишда ва Data Science да қўллаш учун жуда қулай ҳисобланади.

Мисол

Фараз қилинг, об – ҳаво фақат икки ҳолатга эга: қуёшли ва булут. Жорий вақтда об-ҳавони аниқлаш муаммосиз, яъни ёки қуёшли ёки булут бўлади. Энди эртанги ҳавони башорат қилиш масаласи қўйилган бўлсин. Маълумки, об-ҳаво доимий ҳолатда ўзгариши мумкин эмас, ҳаво ўзгариши учун бир неча омиллар таъсири мавжуд бўлади. Бир неча йиллик кузатувчилар асосида шуни таъкидлаш мумкинки, булутли ҳаводан сўнг қуёшли ҳавонинг бўлиши эҳтимоллиги 0,25 га тенг. Икки кун кетма-кет булутли ҳаво бўлиш эҳтимоллиги 0,75 га тенг, чунки фақат икки ҳолат мавжуд.

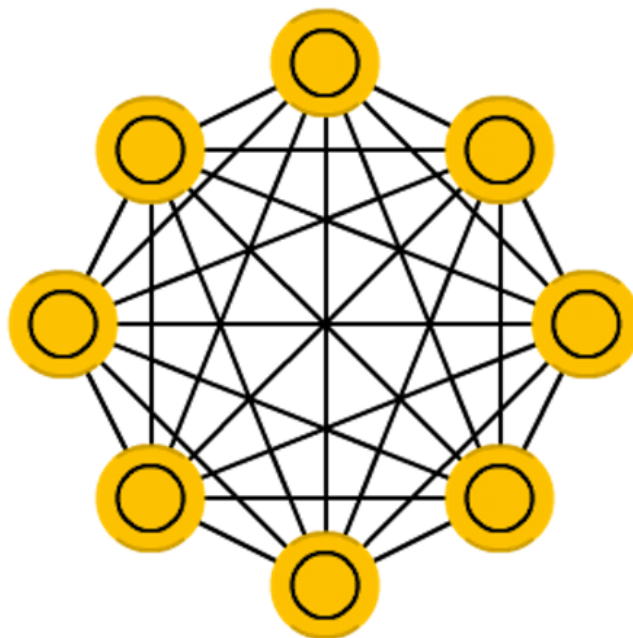


9-расм. Марков занжирига мисол.

Энди бир неча кунлик ҳавони жорий ҳаво кўрсаткичи бўйича башорат этиш мумкин. Ушбу мисол асосида Марков занжирининг асосий моҳиятини кўрсатади, яъни бир ҳолатдан иккинчи ҳолатга ўтишлар эҳтимолликлар тақсимоти асосида бажарилади ва Марков занжирини ҳосил қилади. Демак, Марков занжири мураккаб моделлаштириш усулларни ўрганишда асос бўлиб хизмат қилади.

1.8.Хопфилд Нейрон тармоқ

Тўлиқ боғланган тармоқ (ҳар бир нейрон барча билан боғланган) бўлиб, ҳар бир нейрон барча босқич ролини ўйнайди.



Кириш ячейкасига қайта мурожаат

10-расм. Хопфилд Нейрон тармоқ

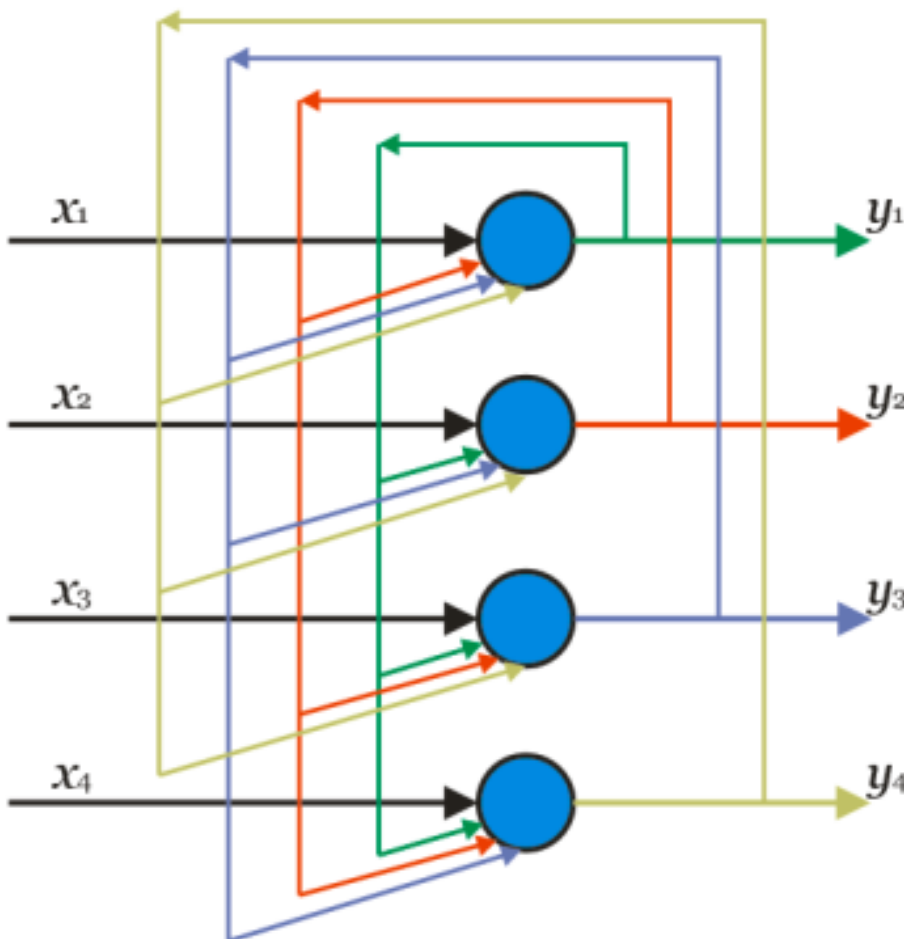
Ҳар бир нейрон ўргангунга кадар маълумотларни киритиш ролида, ўрганиш жараёнида яширин қатлам ролида, натижа учун чиқариш ролида бўлади. Оғирлик матрицаси барча «хотирага сақланган» векторларни хусусийлаштириш асосида яратилади. Шундай бир ҳолатга келтириладики, бир ёки бир неча образларга ўргатилган тизим ягоналаштирилган образга келиб қолади, чунки айнан шу образ стационар бўлиб қолади. Бу эса айнан кутилган ҳолатга келиб қолади деб бўлмайди. Ушбу тизим қисман стабиллашади, чунки умумий «энергия» ёки «ҳарорат» ўрганиш жараёнида сўниб боради. Ҳар бир нейрон фаоллашиш даражасига эга бўлиб, айнан шу ҳарорат билан ўлчанади, агар кирувчи берилганлар йиғиндиси ушбу даражадан ошиб борса, нейрон икки ҳолатдан бирига (одатда -1 ёки 1, баъзан 0 ёки 1) ўтиб қолади. Тармоқ тугунлари параллел равишда янгиланиши мумкин, бироқ улар кўпинча кетма-кет амалга оширилади. Бунда кетма-кетлик тартиби тасодифий ҳосил қилинади ва шу асосда нейронлар ҳолати янгиланади. Ҳар бир нейрон янгилангандан кейин, ҳолати ўзгармас бўлгандан кейин тизим стационар ҳолатга келиб қолади. Бундай тармоқлар кўпинча ассоциатив хотира деб номланади, чунки тармоқнинг ҳолати инсон кутган ҳолатга келиб қолади, буни ярми аниқ бўлган тасвирни охирига етказишга олиб келиниш билан ифодалаш мумкин. Демак, нейрон тармоқ бошланғич берилганларни шовқинли ҳолатда қабул қилиб образнинг қолган қисмини чизиш имконини беради [1].

Хэмминг нейрон тармоқлари бинар кириш векторларини таснифлаш масалаларини ечишда қўлланилади. Унинг асосий вазифаси кириш қисмига узатилган ноаниқ образни унга яқин бўлган образлар эталони бўйича таснифлаш ва мос бўлган синфга боғлаш ҳисобланади. Бунда асосий омил бу Хэмминг оралиғи бўлиб, у ноаниқ образ ва эталон образлар орасидаги фарқ билан белгиланади.

Хэмминг нейрон тармоқлари структураси икки қатламдан иборат бўлиб, нейронлар сони K синфлар сонига тенг ($K = M$). Кириш сони M образлар фарқини кўрсатувчи бинар ҳолатлари сонига. Киришда маълумотлар қиймати $\{-1; 1\}$ тўпламга мансуб. Чиқиш қийматлари

иккинчи қатлам кириш қисмига ва ўзининг кириш қисмига тескари боғланиш орқали узатилади.

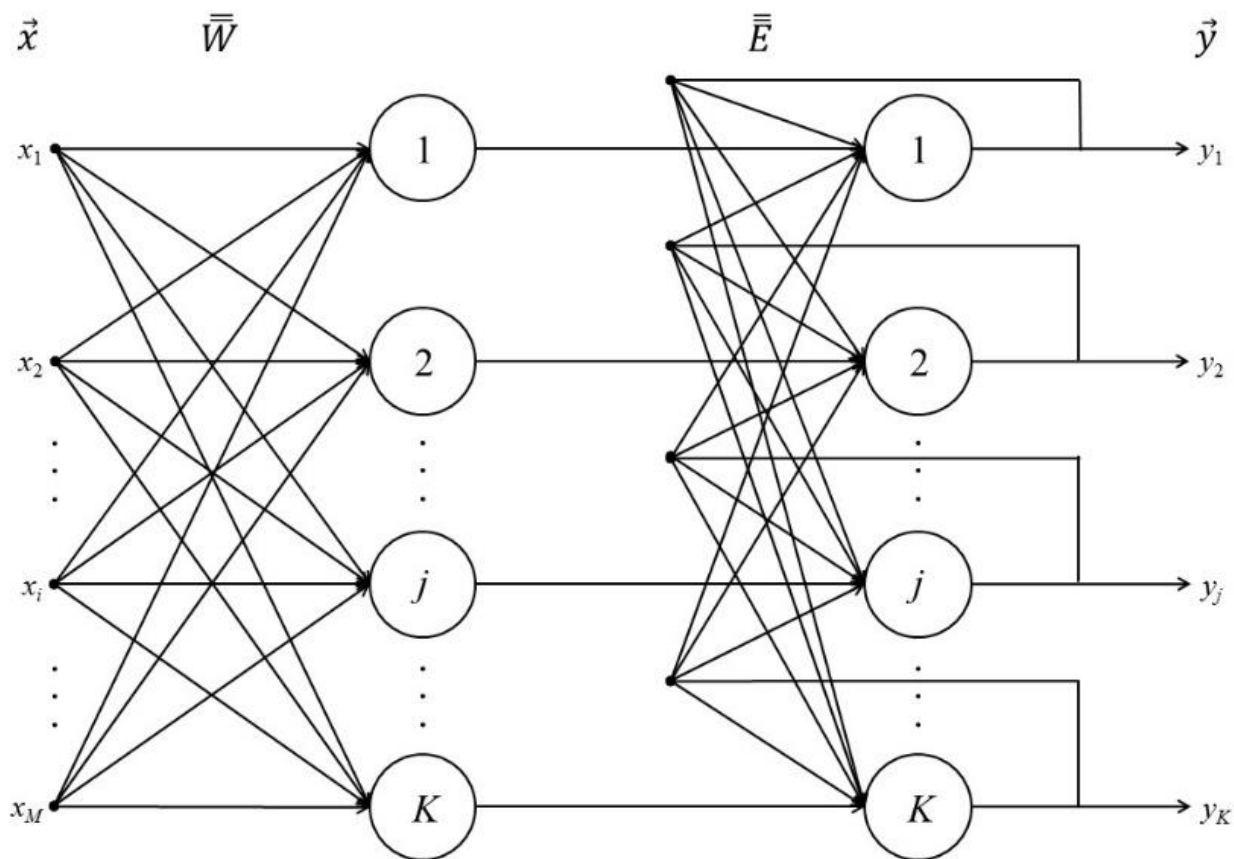
1.9.Хэмминг Нейрон тармоқлари



11-расм. Хэмминг нейрон тармоқлари

Хэмминг нейрон тармоғи орқали масалалар умумий ҳолда қуйидагича ечилади. Бинар векторлар кўринишда берилган бошланғич эталон образлар тўплами мавжуд. Улар ўзининг синфига мансуб. Тармоқда кириш қисмига узатилган номаълум образни барча маълум эталон образлар билан бирма-бир таққослаб чиқиш ва бирон бир синфга мансублиги ёки бирон бир синфга мансуб эмаслигини аниқлаш талаб этилади.

Хэмминг нейрон тармоғининг асосий моҳияти икки босқичдан иборат: ўрганиш ва амалда қўлланилиши.



12-расм. Хэмминг нейрон тармоғи структураси

Ўрганиш босқичида қуйидаги амаллар кетма-кетлиги бажарилади:

1.1. Эталон образларнинг \bar{X} матрицаси $K \times M$ катталиқда шакллантирилади (жадвал-1)

Жадвал 5. Хэмминг нейрон тармоғи эталон образларининг матрицаси

Кўриниш №	№ кировчи бинар ўзгарувчилар					
	1	2	...	i	...	M
1	x_{11}	x_{12}	...	x_{1i}	...	x_{1M}
2	x_{21}	x_{22}	...	x_{2i}	...	x_{2M}
...
j	x_{j1}	x_{j2}	...	x_{ji}	...	x_{jM}
...
K	x_{K1}	x_{K2}	...	x_{Ki}	...	x_{KM}

1.2. Биринчи қатлам нейронларнинг оғирлик коэффициентларини матрицаси ҳисобланади:

$$w_{ij} = \frac{1}{2} x_{ij} \quad (1)$$

Ёки матрица кўринишда ёзилади:

$$\bar{W} = \frac{1}{2} \bar{X} \quad (2)$$

1.3. Фаоллаштириш функцияси кўриниши, чегараси аниқланади

$$f(s) = \begin{cases} 0, & s \leq 0; \\ s, & 0 < s \leq T; \\ T, & s \geq T; \end{cases} \quad (3)$$

– параметри:

$$T = \frac{M}{2} \quad (4)$$

Демак, нейрон тармоқлар чиқиш қисмидаги қийматлар $[0, T]$ оралиғидаги барча қийматларни қабул қилиши мумкин.

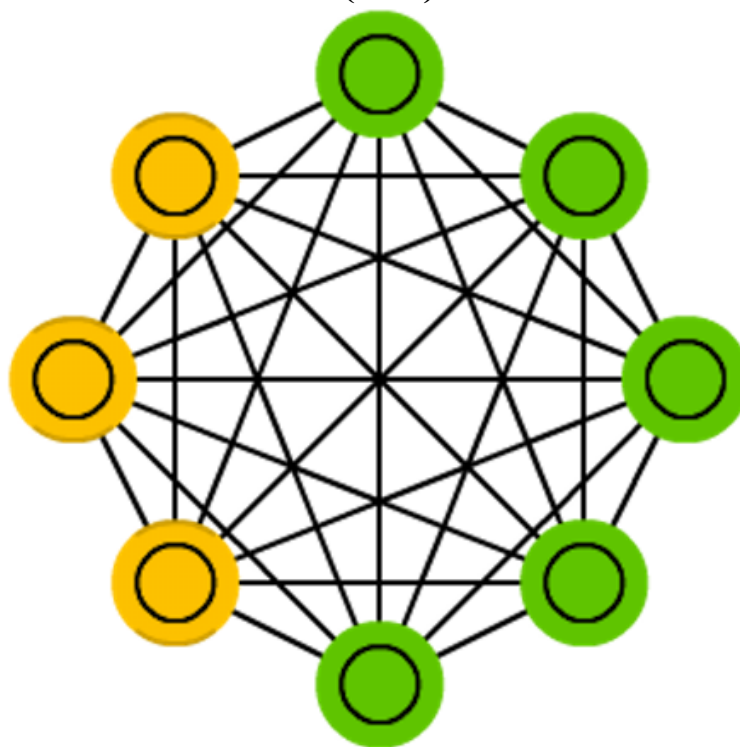
1.4. Иккинчи қатлам нейронларнинг тескари боғланиш синапсларининг қийматлари $K \times K$ ҳажмдаги квадрат матрица элементлари кўринишида берилади:

$$\varepsilon_{jp} = \begin{cases} 1, & j = p; \\ -\varepsilon, & j \neq p, \end{cases} \quad (5)$$

Бунда: $\varepsilon \in \left(0, \frac{1}{K}\right]$

Хэмминг нейрон тармоқнинг тескари боғланиш манфий оғирликка эга синапслари **ингибиторли**, ёки **тормозловчи** деб номланади.

1.10.Больцман (BM) машинаси



● Кириш ячейкасига қайта мурожаат

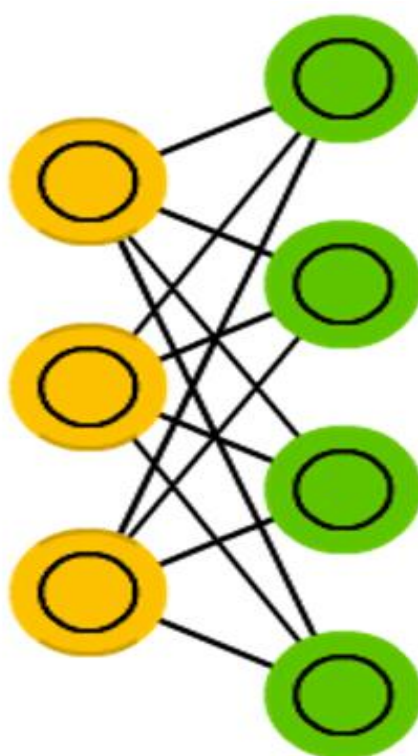
● Эҳтимолли яширин ячейка

13-расм Больцманн машинаси.

(Boltzmann machines, BM) кўп томонлама Хопфилда (HN) тармоғига ўхшаш бироқ, унда баъзи нейронлар кириш нейронлари сифатида, баъзилари эса яширин қатлам сифатида белгиланади. Кириш нейронлари ўз ҳолатларини янгилагандан сўнг чиқувчи нейронга айланиб қолади. Энг аввал оғирлик коэффициентлар тасодифий равишда ўзлаштирилади, сўнгра тескари тарқалиш усулда ёки *contrastive divergence* (Марков занжири ёрдамида градиент ҳисобланиши) алгоритми асосида ўрганиш амали бажарилади. BM – стохастик нейрон тармоғи, чунки ўрганиш жараёнида Марков занжири қўлланилади. Ўрганиш ва ишлаш жараёни Хопфилд тармоғи каби олиб борилади: нейронларга бошланғич аниқ ҳолатлар ўзлаштирилади, сўнгра занжир эркин ҳаракатлана бошлайди. Бу жараёнда нейронлар тармоқли ҳолатни қабул қилиши мумкин,

кирувчи ва яширин қатлам нейронлар ҳолатларига ўтишлар билан белгиланади. Фаоллашув умумий ҳарорат қиймати билан бошқарилади, ҳарорат пасайганда нейронлар энергияси ҳам қисқаради. Энергия қисқариши нейронлар стабиллигига олиб келади. Демак, агар ҳарорат тўғри белгиланган бўлса, тизим мувозанатга эришади[2].

Больцман чекланган машинаси (Restricted Boltzmann machine, RBM)



- Кириш ячейкасига қайта мурожаат
- Эҳтимолли яширин ячейка

14-расм. Больцман чекланган машинаси.

Больцманнинг оддий машинасига жуда ўхшаш бўлиб, RBMнинг фарқи унинг чекланганлигидадир, бу ўз навбатида қўллаш қулайлигини таъминлайди. Унда ҳар бир нейрон қолган барча нейронлар билан боғланмасдан, балки нейронлар гуруҳи бошқа нейронлар гуруҳи билан боғланган. Кириш нейронлари ўзаро

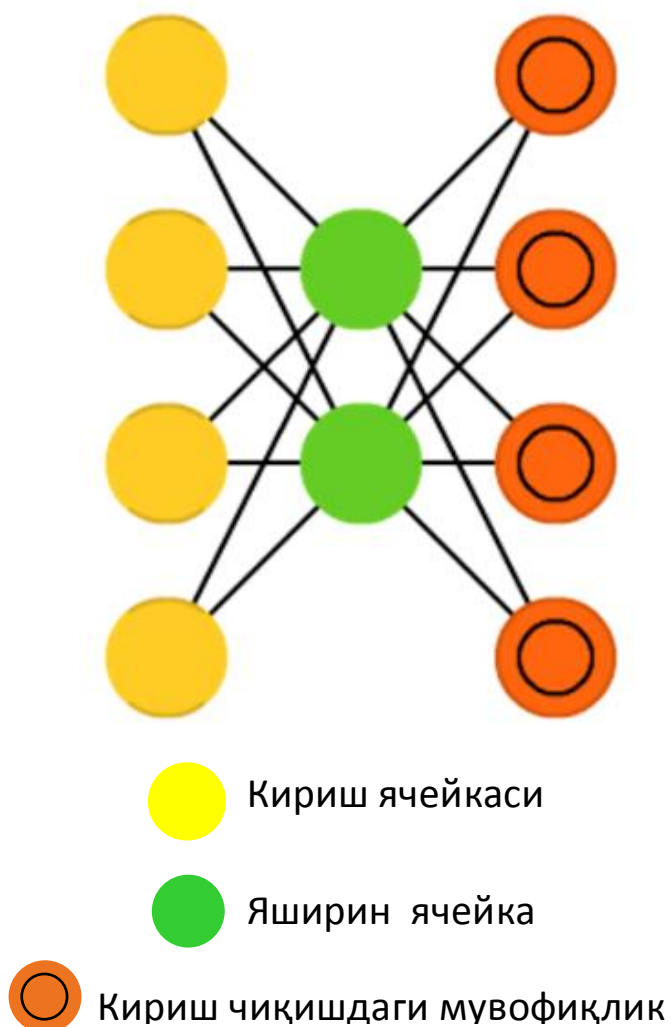
боғланмаган, яширин қатлам нейронлари ўртасида ҳам боғланиш йўқ. RBM ни FFPN каби ўрганиш мумкин, бироқ берилганлар бир қатламдан кейингига, сўнгра охиригача етиб, тескари боғланиш орқали хатоликни текшириш ўрнига, бир қатламдан ўтгандан кейин иккинчига узатилади ва шу ҳолатда тескари боғланган ҳолда хатоликни текшириш олиб борилади (forward-and-back propagation).

Больцман чекланган машинаси биринчи мартаба 1986 йили Пол Смоленски томонидан яратилган бўлиб, *Harmonium* деб номланган, бироқ Хинтон томонидан ўрганиш алгоритмлари ишлаб чиқилгандан кейин 2000-йиллар ўртасида кенг тарқала бошланди. Унинг номи оддий Больцман машинасининг модификацияси сифатида қабул қилинганидан келиб чиқган. Кейинчалик 2000 йиллардан сўнг Больцман машинаси сифатида эмас балки чуқур ўрганиш тармоқларининг алоҳида компоненти сифатида қабул қилинди. Больцман чекланган машинаси бир нечтасини каскадли бирлаштириш чуқур ишончли тармоқни шакллантиради, унда кўп қатлами нейрон тармоқлар тескари боғланишда хатоликни аниқлаш усулида ўқитувчисиз ўрганиш функциясига эга бўлади. Больцман чекланган машиналари хусусиятларидан бири асосан ўқитувчисиз ўрганиш жараёнига ўтиш бўлса, бироқ баъзи ҳолларда ўқитувчи иштирокида ўрганишга олиб боради. Машинанинг яширин қатламини ўрганиш жараёнида маълумотларнинг яширин хусусиятларини ифодалайди.

Больцманнинг чекланган машиналари кенг кўламда қўлланилади масалан, маълумотларни моҳиятини ўзгартирмасдан қисқартириш, тасниф масалалари, коллаборатив фильтрация, хусусиятларни аниқлаш (feature learning) ва тематик моделлаштиришда қўлланилади.

Больцман чекланган машинасидаги нейронлар икки томонлама ҳосил қилади, графнинг бир томонида кўринадиган нейронлар(кириш) мавжуд бўлса, иккинчи томонидан яширин, чунончи ҳар бир кўринадиган ва ҳар бир яширин нейронлар ўртасида боғланиш ҳосил қилинади. Бундай боғланишга эга тизим тармоғини ўрганиш жараёнида контрастли дивергенцияга эга градиент силжиш усулини қўллаб ўрганиш мумкин [3].

1.10.1Автоэнкодерлар

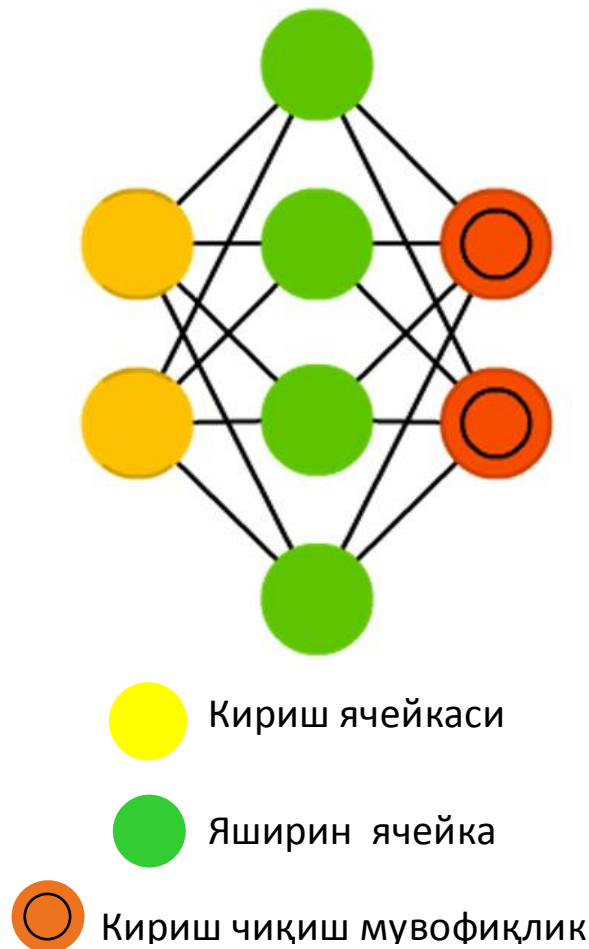


15-расм. Автоэнкодерлар

(**Autocoders, AE**) –FFNN га ўхшаш, яъни FFNN қўллашнинг бошқа усули бўлиб, янги архитектура деб бўлмайди. Автоэнкодерларнинг асосий моҳияти – маълумотларни автоматик кодлаш (шифрлашдагидек эмас, балки уларни архивлаш каби) ҳисобланади. Тармоқнинг ўзи шакли бўйича қум соатга ўхшайди, яъни яширин қатлам кириш ва чиқиш қатламга нисбатан юпқа ва шу билан бирга қўшни қатлам (умумий қатламлар жуфт ёки тоқ бўлганидан келиб чиққан ҳолда бир ёки икки қатлам)га нисбатан симметрикдир. Энг юпқа қатлам ҳар доим ўрта қатлам бўлиб унда маълумотлар максимал равишда сиқилган. Ўрта қатламгача бўлган қисм -кодлаштирувчи, ундан пастда жойлашганлар-декодлаштирувчи, ўртада эса– код.

АЕ хатоликни қайтариш усулида ўргатилади. Бунда кириш қисмга берилганлар ва олинган натижа ўртасидаги тафовут кўринишда хатолик қайтарилади. АЕ ни симметрик кўринишда тузиш учун кодлаштириш оғирликларни декодлаш оғирликларга баравар кўринишда ифодалаш лозим [4].

1.10.2.Сийрак автоэнкодер

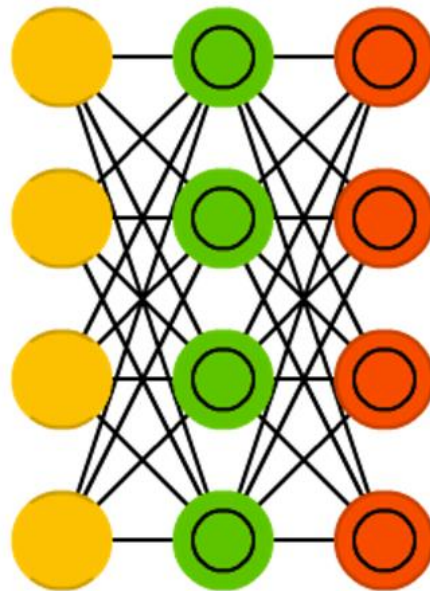





16-расм. Сийраклаштирувчи автоэнкодер.

(**Sparse autoencoder, AE**) –АЕ акси бўлиб, маълумотлар блокини қисқартирилган ҳолатда ифодалаш ўрнига, маълумотлар кенгайиш ҳолатга олиб келувчи кодлаштириштирилади. Бунда тизим энг аввал марказлаштирилиши сўнгра қайта ўз ҳажмига қайтиши ўрнига, ўрта қатлам кенгайтирилади. Ушбу кўринишдаги тармоқлар маълумотлар тўпламидан элементар деталларни ажратиб олишга кўмаклашади.

Агар SAE кўринишдаги тармоқларни АЕ тармоғи каби ўрганиш усулини қўлласак, аҳамиятсиз тармоқлар кетма-кетлигига эришардик, яъни чиқиш қисмида кириш қатлам билан бир хил ҳолат олинарди. Буни олдини олиш учун чиқиш қатламига кириш қатламидаги маълумотлар ва яширин қатламдаги нейронлар ҳар бир фаоллашув сонини жарима кўринишда кўшган ҳолда. Бу ҳолат қайсидир маънода биологик нейрон тармоқларни эслатади (spiking neural network), унда барча нейронлар туғён ҳолатда бўлмайди[5].

1.10.3. Вариацион автоэнкодер архитектураси



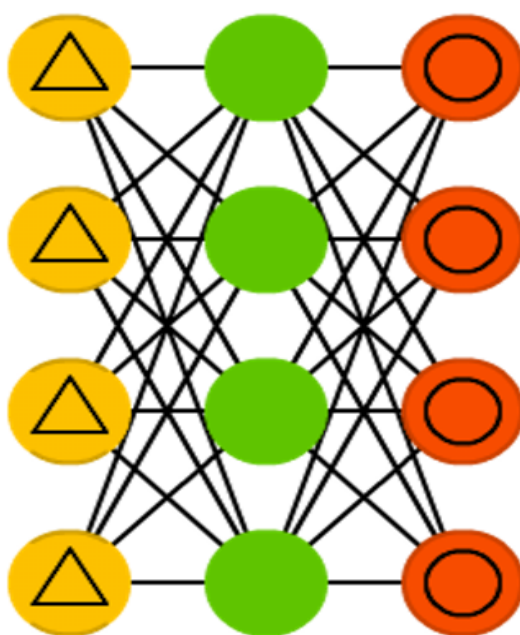
-  Кириш чиқиш ячейкадаги ўхшашлик
-  Эҳтимолли яширин ячейка
-  Кириш ячейкаси




17-расм. Вариацион автоэнкодер.

Вариацион автоэнкодер (VAE) одатдаги архитектурага ўхшаш бўлиб, бироқ уларни ўрганишда – кировчи маълумотларга яқин бўлган эҳтимолликка эга бўлган тақсимотни аниқлашга йўналтирилади. Маълум бир маънода асосий манбага қайтиш бўлиб, VAE ўзи Больцман машинасига ўхшаш. Шундай бўлсада, у асосан Байес математикасига таянади, яъни эҳтимоллик нуқтаи назаридан тушунарли бўлган мулоҳазалар, ифодалар бўлсада, улар мураккаб

ҳисоблашларни тақозо этади. Асосий иш принципини изоҳласак: бир ҳолатнинг иккинчи ҳолатга таъсир даражасини ҳисоблаш ва инобатга олиш ҳисобланади. Агар тармоқнинг маълум бир қисмида аниқ бир жараён бажарилса ва бошқа жараён бирон бошқа қисмида бўлса, улар боғланган бўлиши шарт эмас. Агар улар ўртасида боғланиш бўлмаса, бунда хатоликни аниқлаш бўйича силжиш инобатга олиши лозим. Ушбу ёндашув самарали ҳисобланади, чунки нейрон тармоқлар йирик занжир бўлиб, яширин қатлам бўйича боғланишларни аниқлашда айнан нейронлар ўртасидаги боғлиқсизликларни четлаштириш ишни осонлаштиради [6].

Тармоқ турғунлигини сақловчи (Маълумотлардаги шовқинликни пасайтирувчи) автоэнкодерлар

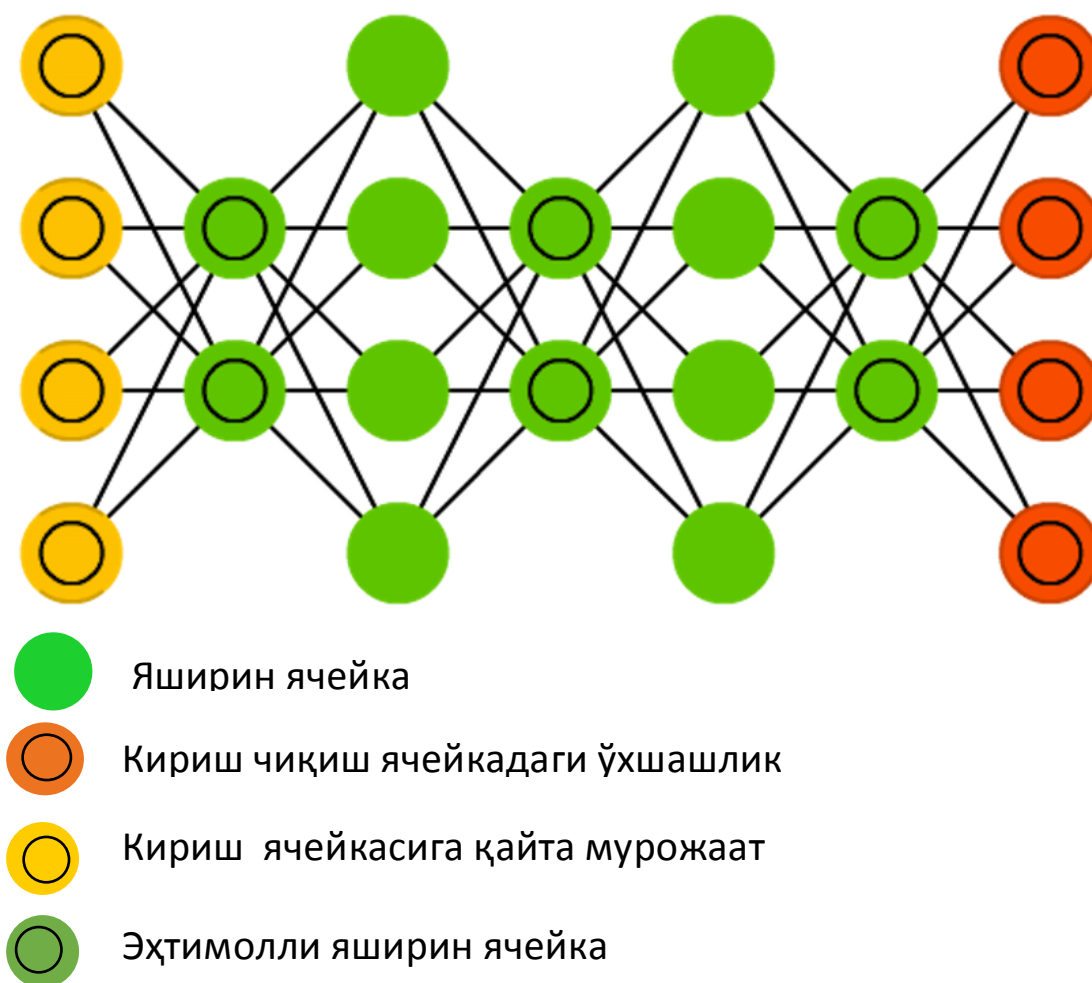


-  Кириш чиқиш ячейкадаги ўхшашлик
-  Орттирмали кириш ячейкаси
-  Яширин ячейка

18-расм. Тўсиқни пасайтирувчи автоэнкодерлар.

Тармоқ турғунлигини сакловчи (Denoising autoшифраторс, DAE) – шундай АЕ ларда кириш қатламига оддий «тоза» берилганлар эмас балки, қўшимча тўсиқлар билан бирга узатилади, масалан, тасвир аниқ кўринишда эмас, балки йирик нукталар кетма-кетлиги кўринишда узатиш бўлсин. Хатоликни аниқлаш кўрилган усул билан бажарилади, яъни чиқиш берилганларини кириш берилганларини тўсиқсиз ҳолати билан таққослаб аниқлаш ҳисобланади. Демак, ушбу усулда тармоқ кичик деталларни эмас балки йирик тузилмаларни эслаб қолади. Чунки кичик деталларни сақлашда тўсиқлар туфайли ўзгаришлар кўпаяди ва тармоқ ишлаши самарасиз бўлиб қолади [6].

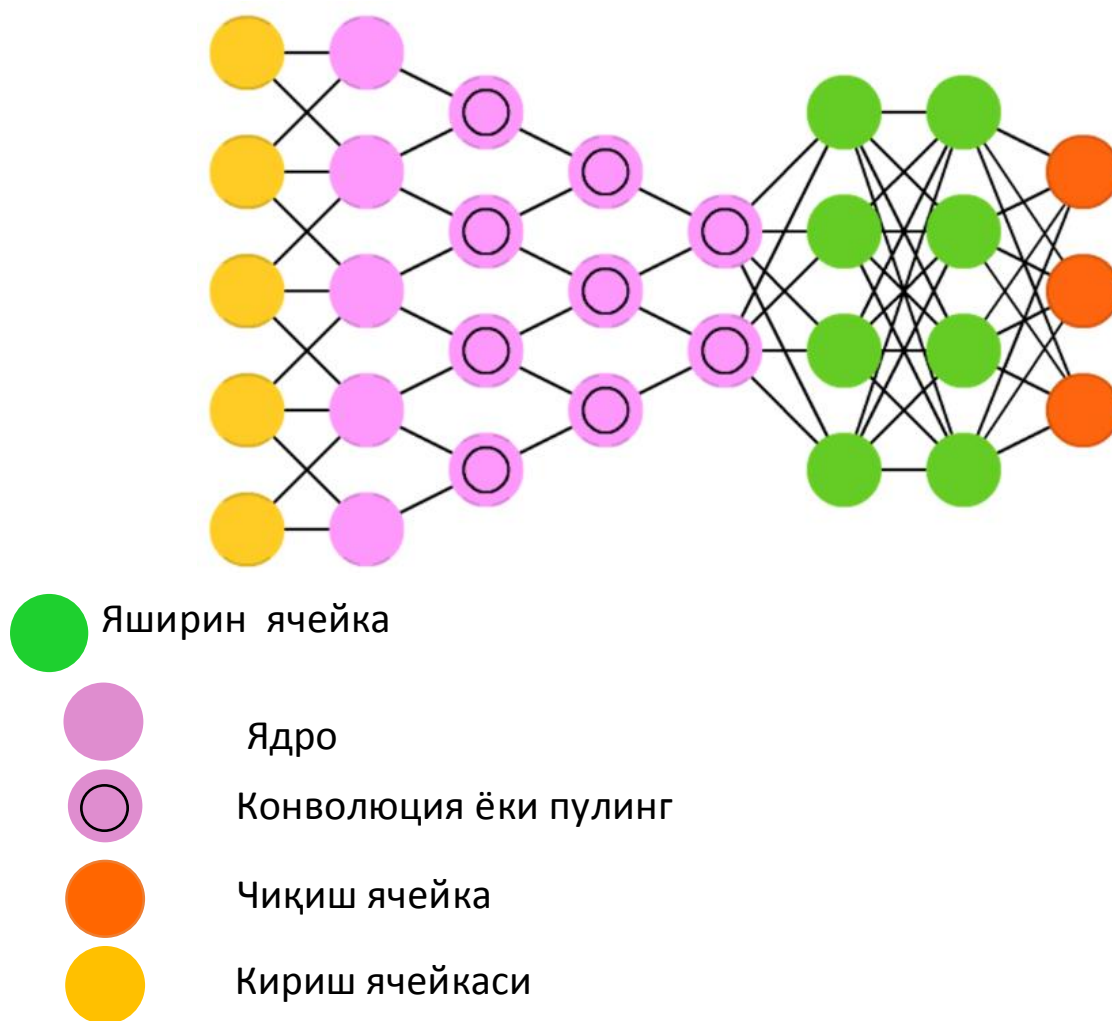
1.11. Чуқур ишончли тармоқлар



19-расм. Чуқур ишончли тармоқлар.

Чуқур ишончли тармоқлар (Deep belief networks, DBN) – ушбу тармоқлар, бир неча RBM ёки VAE композицияларидан иборат. Ушбу тармоқлар самарали эканлиги шундан иборатки, ҳар бир тармоқ бирма-бир ўрганиш жараёнидан ўтиб, кейинги тармоқ олдинги тармоқни кодлаштиришни ўрганиши лозим. Унда айнан жорий ҳолатда тўғри келадиган ечимлардан оптималини топади бироқ бу натижа оптималлигини кафолатламайди. DBN лар contrastive divergence усуллари асосида ёки хатоликни тескари йўналишда аниқлаш усулида ўрганиш жараёни олиб боради, RBM ёки VAE каби берилганларни эҳтимоллик модели кўринишда аниқ ифодалашни ўрганади. Кейинчалик ўргатилган ва стационар ҳолатга келтирилган моделни янги берилганларни яратишда қўллаш мумкин. [7].

1.11.1. Конволюцион нейрон тармоқлар

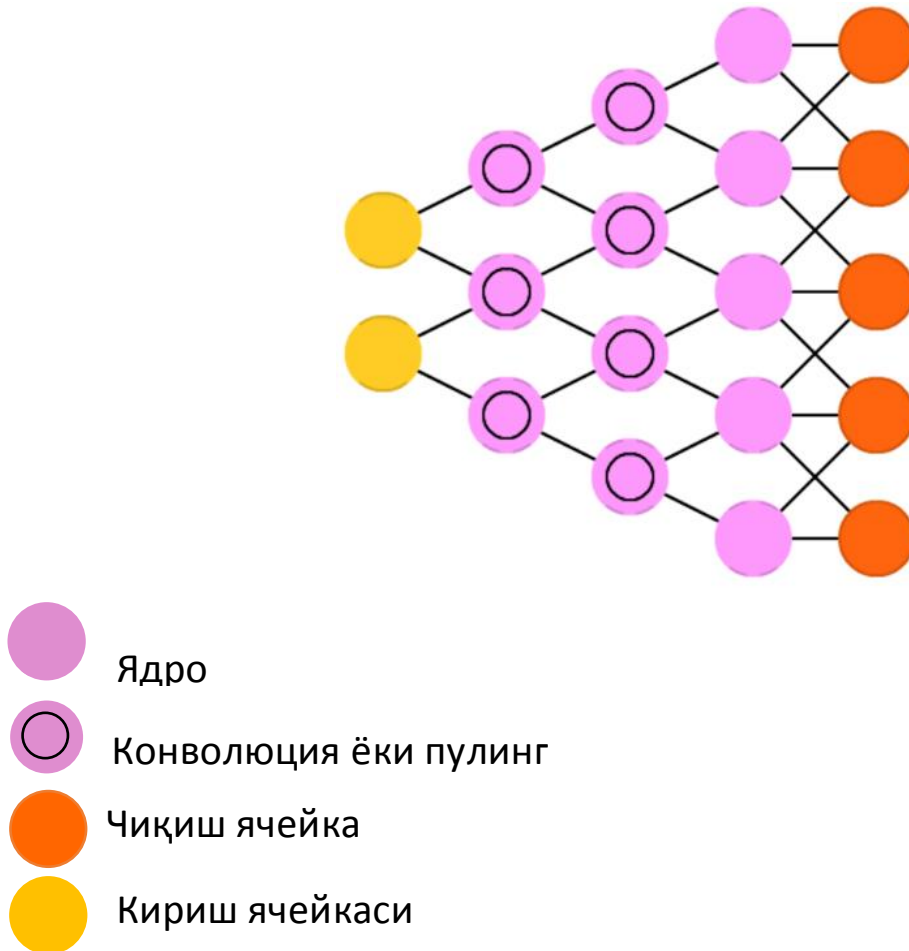


20-расм. Конволюцион нейрон.

Конволюцион нейрон тармоқлар (**convolutional neural networks, CNN**) ва чуқурлашган конволюцион нейрон тармоқлар, (**deep convolutional neural networks, DCNN**) бошқа тармоқлардан анча фарқланади, улар асосан тасвирларни қайта ишлашда, шу билан бирга бошқа кўринишдаги маълумотларни масалан аудио маълумотларни қайта ишлашда қўлланилиши мумкин. Одатий мисол сифатида CNN қўллаб бир неча кўринишдаги тасвирларни узатиш ва уларни таснифлаш, мушук расмини берганда «мушук» деб, итни расмини берганда «ит» деб жавоб чиқарилишини келтириш мумкин. CNN одатда қириш қисми «сканер» ҳолатда бўлиб, олинган маълумотларни барчасини қайта ишлашга мўлжалланмаган. Масалан, 200 x 200 Пиксел ҳажмдаги тасвирни чиқариш учун 40 000 тугундан иборат қатлам яратиш шарт эмас. Бунинг учун 20x20 ҳажмдаги сканерлаш қатлами яратилади, унга тасвирнинг 20 x 20 пиксели юкланади (одатда юқори чап бурчакдан бошлаб юкланади). Сўнгра, кириш қисмини ўрганиш учун ҳам қўллаш мумкин ва сканерни бир пиксел ўнгга суриб кейинги 20 x 20 пикселни ўрганади. Эътиборлиси шуки кириш 20 пикселларини кўчириш керак эмас, балки тасвирни 20 x 20 ҳажмдаги блокларга ажратиб у бўйича ҳаракат қилинади. Кириш қатламидаги маълумотлар конволюцион қатламдан ўтказилиб узатилиб, бунда ҳар бир тугун барча тугун билан боғланиш ҳолатида бўлмайди. Ҳар бир тугун фақат кўшни ячейкалар билан боғланади (боғланиш даражаси қўллаш давомида аниқланади, одатда бир нечтадан ортмайди). Конволюцион тармоқлар қисқариб бориб чуқурлашади ва кириш маълумотлар омилларини ажратиб олади (демак 20 тали қатлам, аввал 10 тали қатлам, сўнгра 5 қатламга ўтади). Бу ерда бараварга қисқариш қоидаси қўлланилиб, у аниқ белгиланган: 32, 16, 8, 4, 2, 1. Конволюцион урдан ташқари пул мавжуд бўлиб, бирлашма – бу деталларни филтрлаб олиш: қўлланиладиган бирлашма технологияси бу 2x2 пикселлар бўйича энг қизил бўлгани ўтказилади. Аудио учун CNN қўллашда асосан товуш тўлқинлари ва улар узунлиги бўйича сегментланган ҳолатда кириш қатламга узатилади. Амалда асосан CNN да кўпинча FFNN қўллаган ҳолда

чизиқсиз абстракцияланган функцияларни қўллаш имконини беради. Ушбу тармоқлар DCNN деб номланади, бироқ номланиши ва аббревиатураси кўпинча бир-бирини ўрнини босади [8].

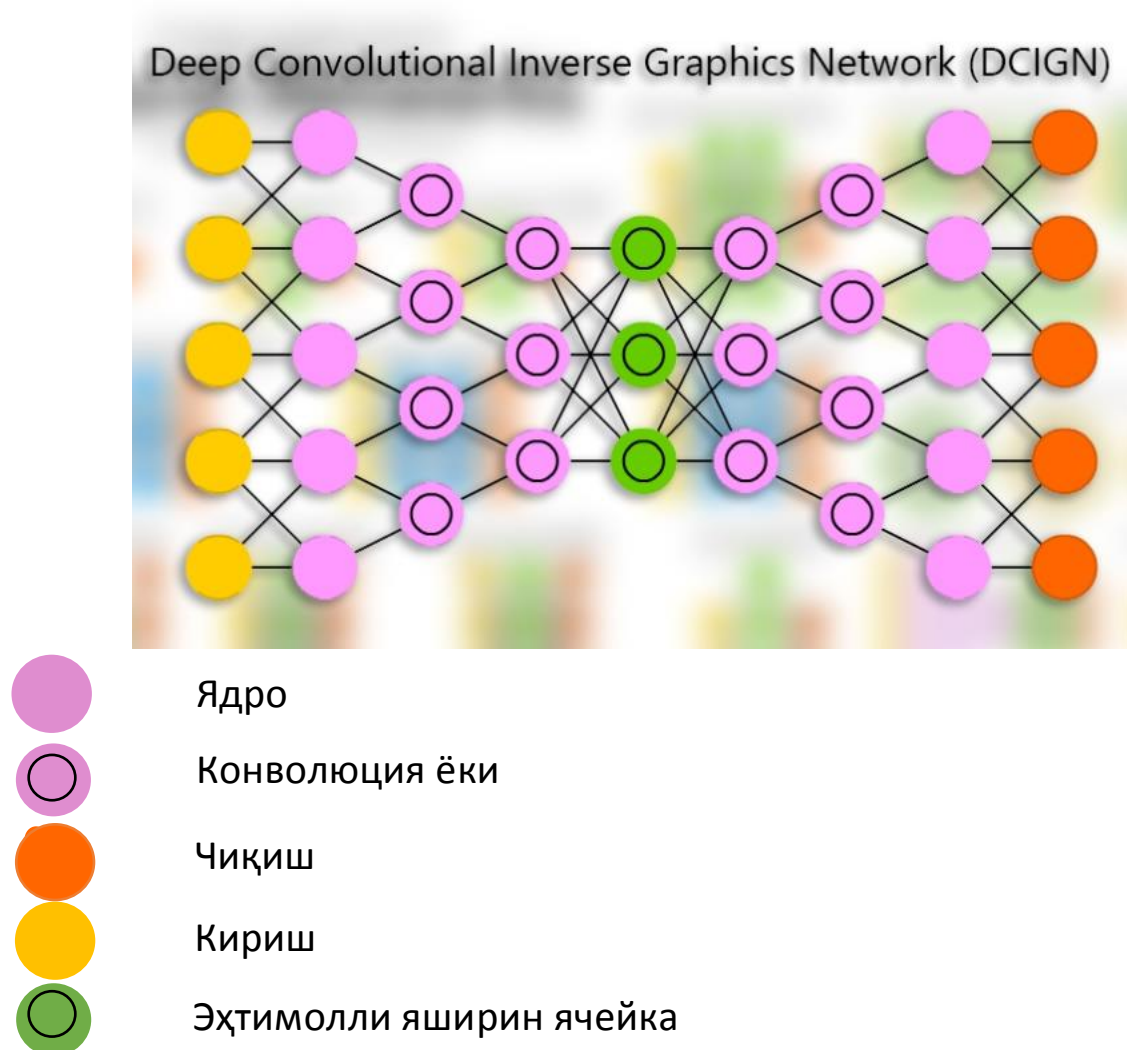
1.12. Деконволюцион нейрон тармоқлар



21-расм Деконволюцион нейрон.

Деконволюцион нейрон тармоқлар (**deconvolutional networks, DN**), тескари график тармоқлар деб номланиб, улар Конволюцион тармоқларнинг акси ҳисобланади. Фараз қилайлик, «мушук» сўзини ўрганиш учун мушуклар тасвирини қайта ишлаши асосида бериладиган тасвирларни таққослаш орқали ўрганиш олиб борилади. DNN тармоқини FFNN билан бирлаштириш мумкин. Кўп ҳолларда тармоқларга маълумотлар сатр курунишдамас, балки бинар таснифланган вектор кўрунишда: масалан, $\langle 0, 1 \rangle$ – бу мушук, $\langle 1, 0 \rangle$ – бу ит, $\langle 1, 1 \rangle$ эса – ҳам мушук, ҳам ит тушунилади. CNN да учрайдиган бирлаштириш қатлами ўрнига унинг акси бўлмиш интерполяция ёки экстраполяция амаллари мавжуд [9].

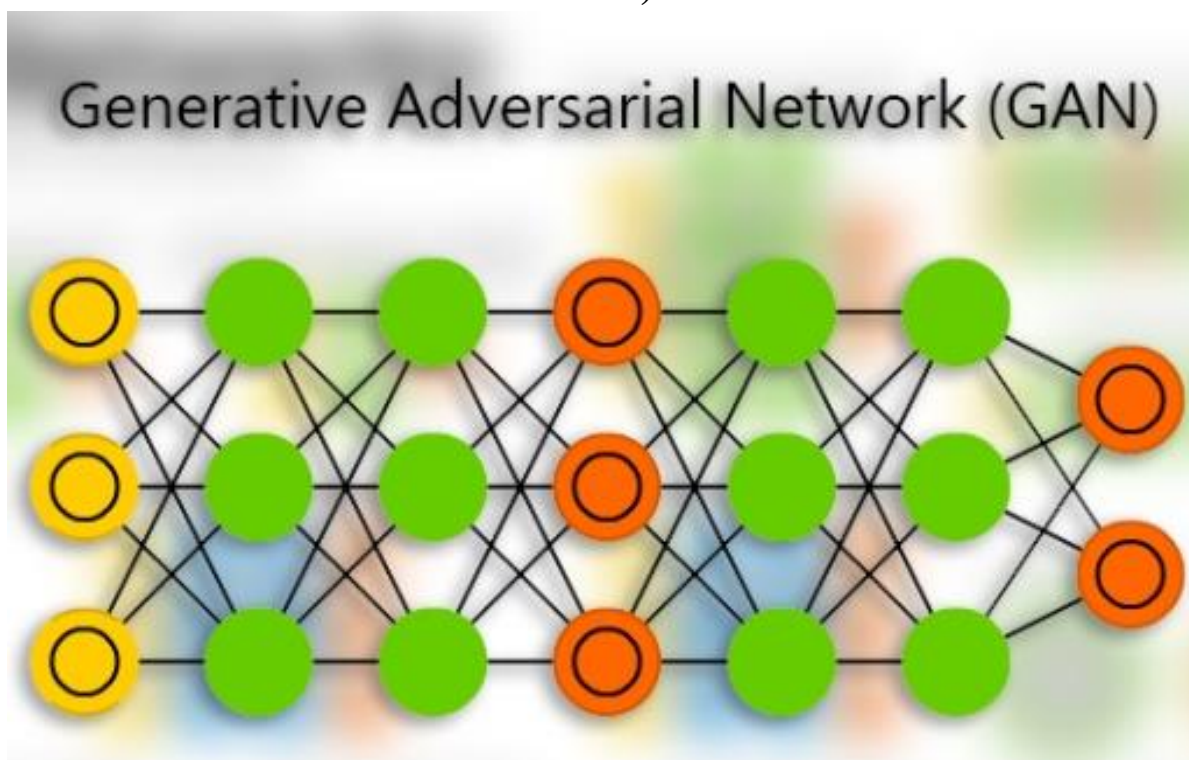
1.12.1. Чуқурлаштирилган тескари конволюцион график тармоғи






22-расм. Чуқурлаштирилган тескари конволюцион график тармоғи

Чуқурлаштирилган тескари конволюцион график тармоғи (deep convolutional inverse graphics network, DCIGN) вариацион автокодлаштирувчи тармоқ бўлиб, кодлаштирувчи ва диодлаштирувчи қисмлари мос равишда конволюцион ва деконволюцион бўлиб хизмат қилади. DCIGN эҳтимолликлар кўринишда объект белгиларини моделлаштиради. Масалан, мушук ва ит тасвирланган объектни яратишга ўрганиш мумкин, ҳолбуки тармоқ фақат мушук ва фақат ит кўрсатилган тасвирларни кўриб ўрганган. Шу билан бирга икки объектдан бирини олиб ташлаш имкони ҳам мавжуд. Чуқурлаштирилган тескари конволюцион график тармоғи хатоликни тескари ҳаракатда аниқлаш усулини қўллайди [9].

1.12.2. Генератив-рақобатли тармоқ (generative adversarial network, GAN)



-  Кириш ячейкасига қайта мурожаат
-  Кириш чиқиш ячейкадаги ўхшашлик
-  Яширин ячейка

23-расм. Генератив-рақобатли тармоқ.

Генератив-рақобатли тармоқ (generative adversarial network, GAN) икки нейротармоқлардан иборат: контент яратувчи генератор, уни баҳоловчи дискриминатор. Кўпинча бу мос равишда FFNN ва CNN тармоқларидир. Дискриминатор – тармоғи генератор томонидан яратилган ёки ўргатувчи маълумотларни қабул қилади. Дискриминаторга келаётган маълумот манбаига асосланиб хатолик шаклланади. Демак, генератор ва дискриминатор ўртасида «мусобақа» пайдо бўлади: биринчиси иккинчисини алдашни ўрганади, иккинчиси эса ёлғонни ечишни ўрганади. Ушбу кўринишдаги тармоқларни ўрганиш мушкул, чунки уларнинг ҳар бирини ўрганиш билан бир қаторда уларни ўзаро мувофиқлашган ҳолда ишлашини созлашни ҳам талаб этади [10].

1.13.Конволюцион нейрон тармоқлари (CNC, CNN)

Конволюцион нейрон тармоқлар (CNC, CNN) оддий нейрон тармоқларга ўхшайди: ўз оғирлик кўрсаткичини ва ўрнини ўзгартирувчи нейронлар асосида шакллантирилади. Ҳар бир нейрон маълум бир берилганларни қабул қилади, уларни скаляр кўпайтмасини олиб боради, баъзи ҳолларда чизикли бўлмаган тенгламаларни қўллайди. Оддий нейронлар каби CNN тўлиқлигича битта дифференцияланадиган тўлдириш функцияни ифодалайди (самарали тўлдириш): бир томондан пикселлари қайта ишланмаган тасвирлар, иккинчи томондан– тасвирни тавсифловчи синфларини ёки тасодифий синфлар гуруҳини чиқаради. Бунда тўлиқ боғланишга эга сўнгги қатламда йўқотиш функцияси мавжуд бўлиб, нейрон тармоқ билан боғланиш жараёнида ишлаб чиқилган барча қонуниятлар боғланиш функциялари CNN да пайдо бўлади. Ундаги фарқни кўриб чиқамиз.

Конволюцион нейрон тармоқлар архитектураси «кириш қисмидаги берилганлар тасвирни ҳосил қилади» принципини тахмин қилади, сўнгра маълум бир хусусиятларини архитектурасига мос қилиб аниқлайди. Ушбу хусусиятга асосланган ҳолда бошланғич маълумотларни тахмин қилган ҳолда тармоқ параметрларини қисқартирган ҳолда самарали фойдаланиш.

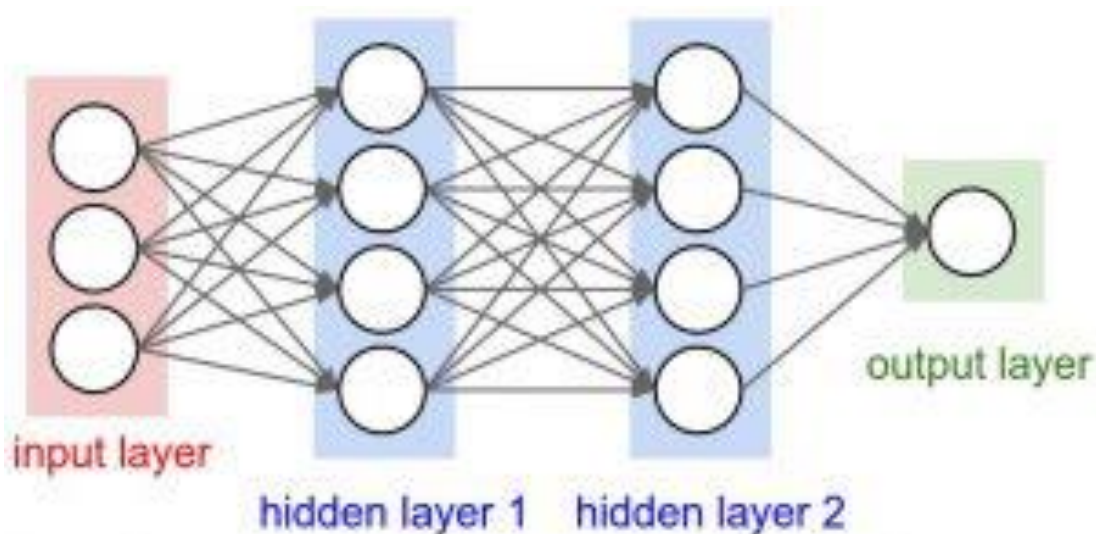
1.13.1.CNN архитектураси тўлиқ изоҳи

Маълумки, нейрон тармоқлар бошланғич маълумотларни қабул қилади (битта вектор) маълумотларни бир неча яширин қатламлардан ўтказиб маълумотлар силжишига эришади. Ҳар бир яширин қатлам бир неча нейронлардан иборат бўлиб, ҳар бир нейрон олдинги қатлам нейронлари билан боғланишга эга бўлади ва битта қатлам функциясида ўзаро мустақил ва ўзаро боғланиш эга бўлмайди. Сўнгги тўлиқ боғланган қатлам чиқиш қатлами деб номланиб, таснифни аниқлашда синфлар сонини ифодалайди.

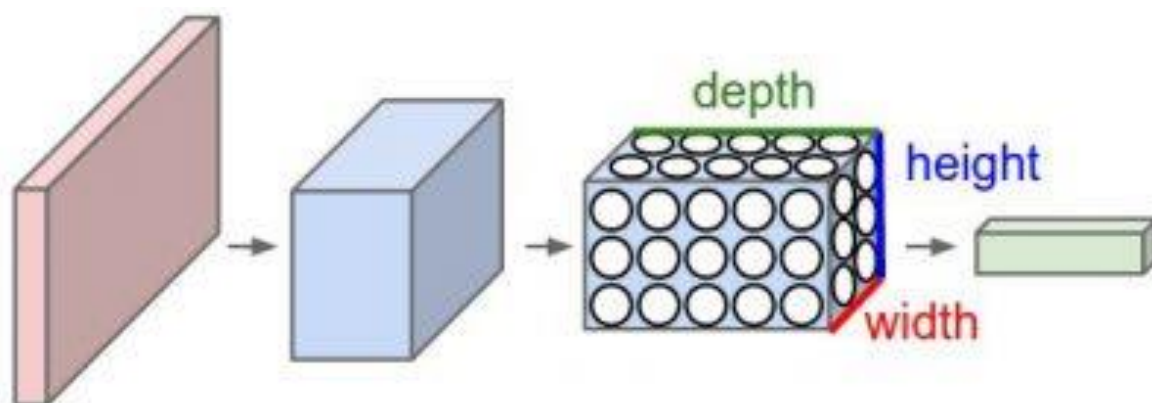
Оддий нейрон тармоқлар катта ҳажмдаги тасвирлар билан ишлашда масштаб танланмайди. CIFAR-10 компьютерли кўриш тизимида тасвир $[32 \times 32 \times 3]$ (32 – кенглиги, 32 – баландлиги, 3 –

ранглар канали) ҳажмга эга, шунинг учун оддий нейрон тармоғидаги биринчи яширин қатламга уланган нейрон 3 072 ($32 \times 32 \times 3$) оғирликка эга бўлади. Бир кўришда ўзгартириш мумкин деб бўлади, бироқ катта ҳажмдаги тасвирларда тўлиқ боғланган структура масштабланмайди. Катта рухсат кўрсаткичига эга тасвирлар, масалан, $[200 \times 200 \times 3]$, тўлиқ боғланган нейрон 120 000 оғирликка эга бўлади. Бундан ташқари, янги параметрларни қўшиш имконини берувчи бир неча нейронлар мавжуд бўлиши талаб этилсин. Тўлиқ боғланиш – жуда кўп ресурсни талаб этмокда ва катта ҳажмдаги параметрлар тез қайта ўрганишдан ўтади.

Конволюцион нейрон тармоқларга кирувчи маълумотлар тасвири бўлиб, бошқача тармоқни яратиш учун чекланишлар ҳосил қилади. Оддий нейрон тармоқлардан фарқли равишда CNN қатламлари нейронлар уч ўлчовли: энига, бўйига ва чуқурликка, яъни, ҳажмни шакллантирувчи ўлчамлар асосида шаклланади. Масалан, CIFAR-10 кириш қисмида фаоллашган кирувчи ҳажм сифатида қабул қилинади, ҳажми эса $32 \times 32 \times 3$ ўлчовда шакллантирилади. Нейронлар эса қатламнинг маълум бир қисмига фаоллашган ҳолда боғланади. Бундан ташқари, чиқувчи натижа қатлами айнан шу тизимда $1 \times 1 \times 10$ га тенг бўлади, CNN тўлиқ шакллантириш натижасида тўлиқ тасвири синфларни белгиловчи векторга айлантирилади, бу ўз навбатида чуқурлик ўлчови кўринишда изоҳланади. Қуйида ушбу жараён визуализацияси келтирилган



24-расм. Оддий нейрон тармоқ.



25-расм. Конволюцион нейрон тармоқ.

24- расмда оддий уч ўлчовли нейротармоқда жойлашган нейронлар жойлашуви.

25-расмда конволюцион уч ўлчовли нейротармоқда жойлашган битта қатламида нейронлар жойлашуви. CNN ҳар бир қатлами кирувчи 3D-ҳажмли маълумотни активлашган чиқувчи 3Dҳажмли нейронларга ўгиради. Ушбу мисолда қизил кирувчи қатлам тасвир кўринишда бўлганлиги сабабли эни ва бўйи тасвир ўлчами билан белгиланади, чуқурлиги эса 3 га тенг (қизил, яшил, кўк каналлар) бўлади.

Натижа: конволюцион нейротармоқларнинг асосини қатламлар ташкил қилади. Ҳар бир қатлам оддий API билан тавсифланади: у кирувчи 3D-ҳажмли маълумотларни чиқувчи 3D-ҳажмли маълумотларга ўгиради, бунда ўзгартирувчи функция параметрга эга бўлган ва эга бўлмаган кўринишда ўзгартиради.

1.13.2.CNN қўлланиладиган қатламлар

Юқорида айтилгандек схематик кўринишда CNN – бу қатламлар кетма-кетлиги ҳисобланади. Ҳар бир қатлам фаоллашган ҳажмни дифференциаллашган функция орқали бошқа ҳажм кўринишга ўгиради. Конволюцион нейрон тармоқларни шакллантириш учун 3 та асосий қатлам қўлланилади:

1. конволюцион,
2. пулинг (қуйи танланма ёки субдискретизация),

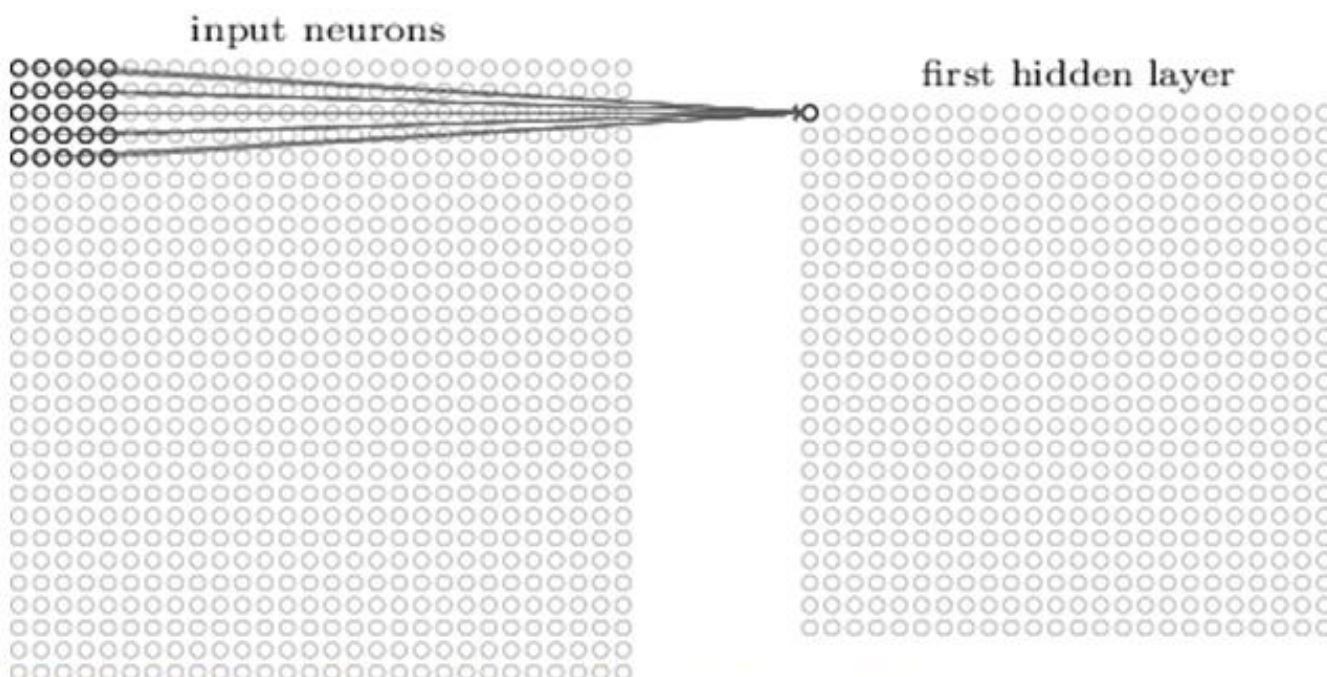
3. тўлиқ боғланган қатлам.

Ушбу қатламлар тўлиқ CNN архитектурасини шакллантириш учун қўлланилади.

CNN тармоғи шакллантиришни маълум бир мисолда тасвирни аниқлаш тизими таснифини CIFAR-10 кўринишда келтириб, [INPUT – CONV – RELU – POOL – FC] архитектурага эга бўлади.

- INPUT (кирувчи маълумотлар) $[32 \times 32 \times 3]$ да тасвир ҳақида бошланғич маълумотлар сақланади (бу мисолда 32 – эни, 32 – бўйи, 3 – ранглар канали R, G, B).

- CONV қатлами (конволюция қатлами) фильтр кўрсаткичи қийматларини тасвир пикселларининг бошланғич қийматларига кўпайтиради (элементлар бўйича кўпайтириш), сўнгра бу кўпайтмалар суммаланади. Киритилган тасвирнинг ҳар бир уникал позицияси қийматга эга бўлади. Масалан: 12 талик фильтр қўлланилса, ҳажми $32 * 32 * 12 [32 \times 32 \times 12]$ га тенг бўлади.



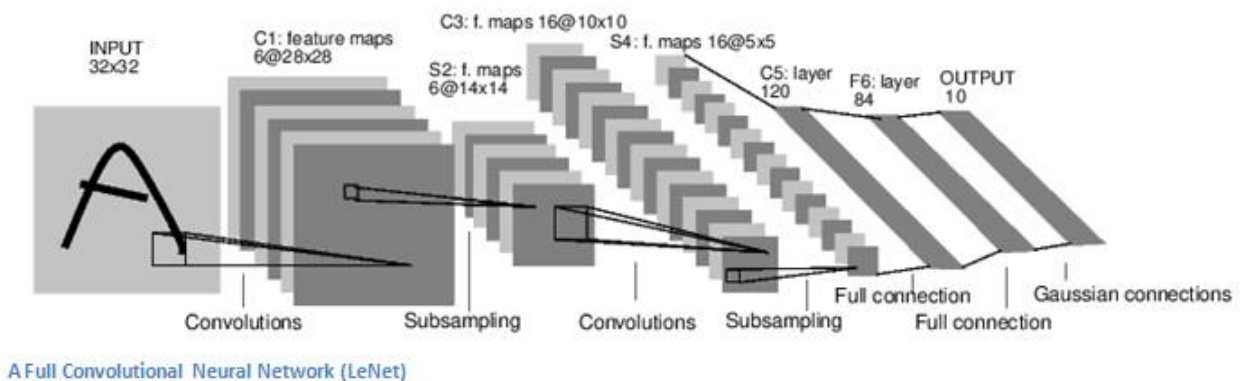
Visualization of 5 x 5 filter convolving around an input volume and producing an activation map

26-расм. Фаоллашган харитада 5x5 фильтр орқали киритилган тасвирни конволюция жараёни кўриниши.

- RELU Қатлам (чизикли ажратиш блоки) элементлар бўйича фаоллаштириш функциясини $f(x) = \max(0, x)$ қўллайди, бунда бошланғич кўрсаткич 0 деб ўрнатилади. Бошқача қилиб айтганда, RELU қуйидаги амалларни бажаради: агар $x > 0$, ҳажм ўзгармас қолади ($[32 \times 32 \times 12]$), агар $x < 0$, кераксиз қисмлар 0 га ўгириш орқали қирқилади.

- POOL Қатлами (пулинг қатлами) фазовий ўлчамларни (эни ва бўйини) қисқартириш амалларини бажаради, натижада ҳажми $[16 \times 16 \times 12]$ гача қисқартирилади. Яъни бу босқичда хусусиятлар харитасида чизикли бўлмаган зичланиш бажарилади. Ишнинг мантиқан бажарилиши қўйидагича: агар олдинги конволюция амалида маълум бир хусусиятлар аниқланган бўлса, кейинги қайта ишлашлар учун тўлиқ тасвир узатилиши шарт эмас, шунинг учун зичланган ҳолда тасвир узатилади.

- FC Қатлами (тўлиқ боғланган қатлам) керакли синфни аниқлаш учун N-ўлчамли векторни чиқаради (N – синфлар сони). Олдинги қатлам чиқиш қисмига (хусусиятлар харитаси) мурожаат орқали ва аниқ синфга мансуб бўлган хусусиятларни аниқлаш орқали амалга оширилади.



27-расм. Тўлиқ конволюцион нейрон тармоқ.

Айнан шу усулда CNN қатламлар бўйича ўтиш орқали бошланғич тасвирни кўчиради, бунда бошланғич пикселлар қийматларига асосланган ҳолда синфларни аниқлаш амали бажарилади. Аҳамиятли бўлмаган қатламлар ҳам параметрларга эга бўлади. Хусусан , конволюция қатлами ва тўлиқ боғланган қатлам ўзгартириш амалини

бажаради, бу эса бошланғич фаоллашган ҳажм кўрсаткич билан бир қаторда параметр (оғирлик ва силжиш кўрсаткичи)га асосланган ҳолда бажарилади. Иккинчи томондан чизиқли ажратиш блоки ва пулинг қатлами аниқ белгиланган функцияни қўллайди. CONV ва FC қатламдаги параметрлар градиентли ўтиш усули ёрдамида ўрганишни амалга оширилади, шунинг учун конволюцион нейротармоқ орқали синфни аниқлаш амали ҳар бир тасвирга мос равишда ўрганиш учун белгиланган кўрсаткичларга боғлиқ бўлиб қолади.

Натижада:

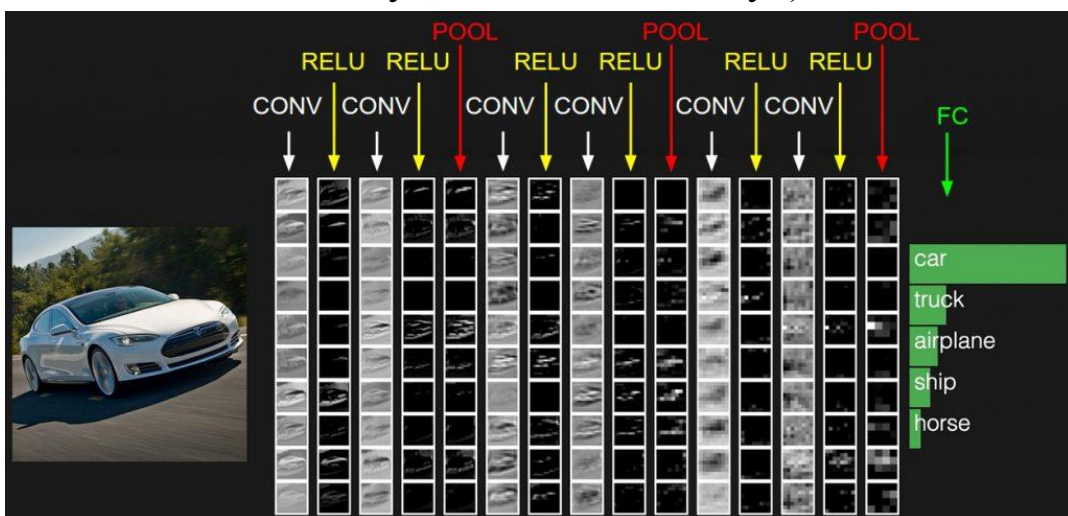
1. Энг содда ҳоллар учун CNN архитектураси – қатламлар жамланмаси бўлиб, берилган образни чиқувчи (масалан, синфни аниқлаш кўринишдаги) образга ўгириш.

2. Ҳар бир қатлам тасвирни қайта ишлаш жараёнининг маълум бир босқичи учун жавоб беради (конволюция қатлами, чизиқли тақсимлаш, пулинг ва тўлиқ боғланган қатлам, булар барчаси ҳозирда кенг тарқалган).

3. Ҳар бир қатлам кириш қисмида 3D ҳажмли маълумотни қабул қилади ва 3D-ҳажмни сақлаган ҳолда зичлаш функцияси орқали ўзгартиради.

4. Қатлам параметрларга эга бўлмаслиги мумкин (CONV ва FC да мавжуд, RELU ва POOL – эса йўқ).

5. Қатлам гиперпараметрларга эга бўлмаслиги мумкин (масалан, CONV, FC ва POOL да мавжуд, RELU – да эса йўқ).



28-расм. CNN қатламлари асосида тасвирни қайта ишлаш.

28-расмда кўрсатилгандек бошланғич ҳажм қайта ишланмаган пикселларга эга (чапда), сўнгги ҳажмда эса – аниқланган синф(ўнгда). Ҳар бир фаоллашган ҳажм тасвирни қайта ишлаш жараёнида устун ҳолатда кўрсатилган. 3D-ҳажмни визуаллаштириш мушкул бўлганда ҳар бир қатлам ҳажм кўриниши қатор кўринишга келтирилади. Сўнгги қатлам ҳажми ҳар бир синфнинг эҳтимоллик кўрсаткичини белгилайди, FC синфларни сараланган кетма-кетликда визуаллаштиради. Бу ерда изоҳланган архитектура қисқартирилган VGG (Visual Geometry Group) тармоқни ифодалайди.

Энди ҳар бир қатламни бирма-бир кўриб чиқамиз, улар ўртасидаги боғланишни, гиперпараметрлик хусусиятларини ўрганиб борамиз.

1.13.3.Конволюция қатлами

Конволюция Қатлами– CNN қатлам асосини яратувчи қатлам бўлиб, асосий вазифани амалга оширади.

«Ақлий» хусусиятларни инобатга олмаган ҳолда ишнинг мантиғи ва
ИЗОҲИ

Фараз қилинг, CONV қатлам «ақлий» ёки нейрон ёндашувсиз ишлайди. Конволюция қатламининг параметрлари ўрганиш филтрлари тўпламидан иборат. Ҳар бир филтр ўзига хос кичик ўлчамли (эни ва бўйига) фазога эга, бироқ, кирувчи тўлиқ ҳажм бўйича ўтади. Масалан, конволюцион нейрон тармоқнинг биринчи қатлам стандарт филтри $[5 \times 5 \times 3]$ ўлчамга тенг бўлиши мумкин. Олдинга қараб ҳаракатланганда кирувчи маълумотларни филтрнинг эни ва бўйига қараб ўтказамиз ва филтр ёзуви бўйича скаляр кўпайтмасини кирувчи маълумот бўйича амалга оширамиз. Тасвирнинг энига ва бўйига қараб филтр ўтиши давомида 2-ўлчовли фаоллашиш харитасини тузамиз ва унинг асосида фазонинг хоҳлаган қисмида аксланишни аниқлайди ва кейинчалик ифодалайди. Маълум бир визуал кўрсаткич аниқланганда фаоллашадиган филтрларни тармоқ ўрганиб чиқади. Бу биринчи қатламда маълум бир ўтиш чегараси, аниқ ранг тўплами бўлиши мумкин, тармоқнинг юқори босқичларида эса айланма нақшлар кўринишда бўлиши мумкин. Кейинги жараёнларни ҳар бир конволюция қатламида филтрлар

тўплами билан иш олиб борилади, улар эса ўз навбатида алоҳида 2-ўлчовли фаоллаштириш харитасини шакллантиради. Ушбу фаоллаштириш харитасини чуқурлик бўйича йиғиб бориб чиқувчи ҳажмни шакллантириш мумкин..

1.13.4.«Ақлий» ёндашув

Нейронлар билан аналогияларни ўтказиш ҳолати нуқтаи назардан қарасак, чиқувчи ҳажмдаги ҳар бир ёзувни чиқувчи нейрон сифатида қурилиб, улар кейинги қатламнинг кирувчи ҳажмнинг кичик бир қисмига мансуб сифатида боғланишга эга деб ҳисоблаш лозим. Улар ўз навбатида барча нейронларни фазовий параметрлар бўйича чапга ва ўнгга қараб тақсимлайди (улар бир хил филтър натижасидир). Нейронлар ўртасидаги боғланишни, фазовий жойлашувини ва параметрлар тақсимоли моделини ўрганиб чиқамиз.

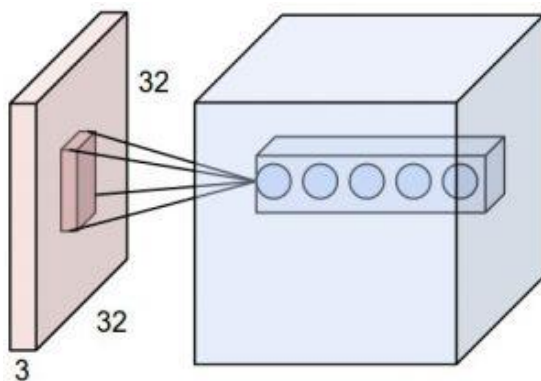
1.13.5.Локал боғланиш

Катта ҳажмдаги маълумотларнинг киритиш ҳолати борасида сўз борганда нейронлар ва бошланғич ҳажмга эга нейронлар ўртасида боғланишни ўрнатиш мақсадга мувофиқ бўлмайди. Унинг ўрнига ҳар бир нейронни кирувчи ҳажмнинг маълум бир локал соҳасига боғлаш қулай ҳисобланади. Ушбу фазовий боғланиш гиперпараметр ҳисобланиб, рецептив (сезувчанлик) майдони деб номланади. Сезувчанлик майдони филтърлаш майдонига тенг бўлиб қолади. Чуқурлик бўйича давомийлик кўрсаткичи бошланғич чуқурлик кўрсаткичига доим эквивалент ҳисобланади. Фазовий ўлчамларни (эни ва бўйини) кўриб чиқишда ассимметрия мавжудлиги, чуқурлик кўрсаткичи эса: эни ва бўйича локал бўлиб, бироқ тўлиқ чуқурлик кўрсаткичи бўйича ўтишини яна бир марта изоҳлаб ўтаемиз.

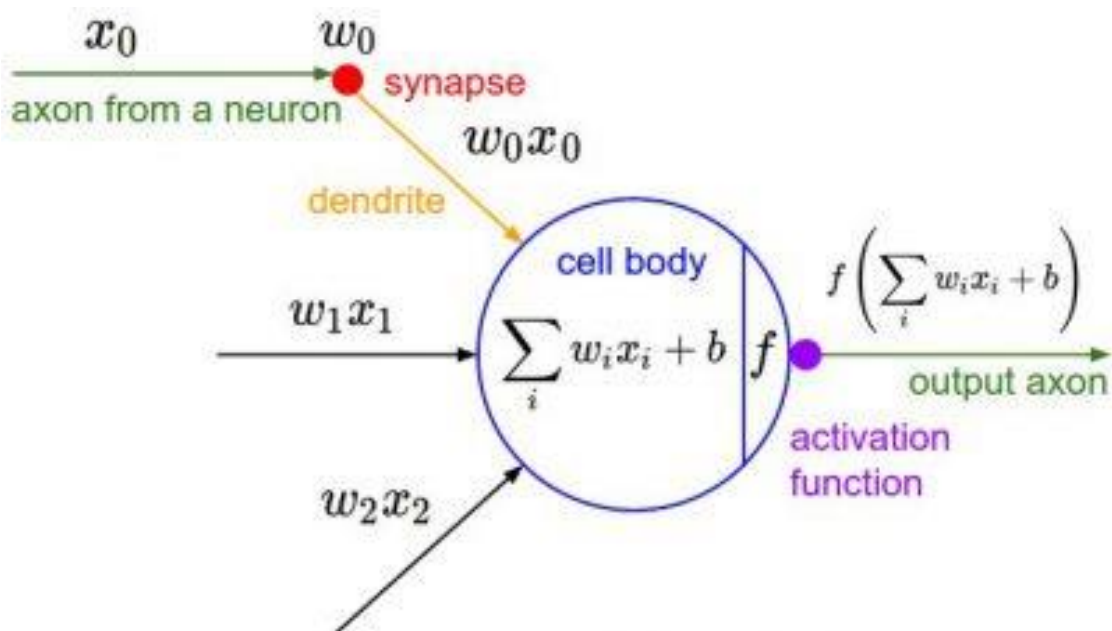
Мисол 1. Фараз қилайлик, кириш қисмига расм $[32 \times 32 \times 3]$ ўлчамга тенг бўлсин (масалан, RGB-тасвир CIFAR-10). Агар сезувчанлик майдони (филтър ўлчами) 5×5 га тенг бўлса, конволюция қатламидаги ҳар бир нейрон кирувчи ҳажмда $[5 \times 5 \times 3]$ ораликда оғирликка тенг бўлади бу натижада $5 * 5 * 3 = 75$ (+1 силжиш параметри) га тенг бўлади. Шунини таъкидлаш керакки, чуқурлик ўқи бўйича давомийлик 3 га тенг

бўлиши керак, шу ҳолдагина математик боғланиш тўғри белгиланган ҳисобланади.

Мисол 2. Фараз қилайлик кириш қисмига расм $[16 \times 16 \times 20]$ ўлчамга тенг бўлсин. У ҳолда сезувчанлик майдони 3×3 га тенг деб қабул қилиниб, конволюция қатламидаги ҳар бир нейрон умумий ҳолда кириш қисмида 180 ($3 \times 3 \times 20$) боғланишга тенг бўлади. Демак, бунда локал боғланиш эни ва бўйича (бу ерда— 3×3) кенгликда бўлиб, бироқ тўлиқ чуқурлик (20) бўйича ўтади.



29- расм. Конволюция қатлами нейронларининг локал сатҳли кириш қисми (кирувчи тасвир қисми) билан боғланиши.



30-расм. Нейронларнинг математик кўриниши.

29-расмда: Мисол кирувчи ҳажмли тасвир (қизил тўртбурчак, параметрлари $[32 \times 32 \times 3]$, CIFAR-10 тасвири) ва конволюцион нейрон тармоқнинг биринчи қатламидаги нейронларнинг тахминий ҳажми.

Конволюцион қатламдаги ҳар бир нейрон фақат кириш локал қисми билан боғланган бўлиб, фазовий давомийлик эни ва бўйи кўрсаткичи билан белгиланади, бироқ нейрон тўлиқ чуқурлик бўйича, яъни барча ранглар каналлини қамраб олади. Эътибор бериш лозимки, бунда бир нечта нейронлар, (айнан шу мисолда 5та), битта соҳани қамраб олувчи тўлиқ чуқурликдан ўтишади (устунлар чуқурлигини қуйида кўриб ўтилади).

30-расмда: Оддий нейротармоқлар синфига тааллуқли нейронлар ўзгармаслигича қолади: улар одатдагидай уларнинг оғирлик кўрсаткичи ва уларнинг чизиқли бўлмаган маълумотлари ўртасида скаляр кўпайтмасини ҳисоблайди, бироқ уларнинг боғланиши локал сатҳлар билан чеклаб қўйилади.

1.13.6.Фазовий жойлашув

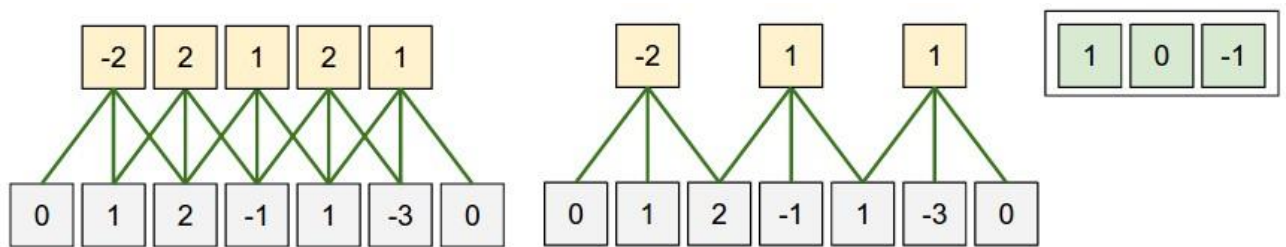
Конволюция қатламда ҳар бир нейроннинг кирувчи ҳажм билан боғланишини кўриб ўтдик, бироқ нейронлар сони борасида ёки чиқиш ҳажмида уларнинг жойлашуви борасида маълумот кўриб ўтилмади. Чиқувчи маълумотларнинг ўлчамини белгиловчи 3 та гиперпараметр мавжуд: чуқурлик, қадам ва нолларнинг сони.

1. Энг аввал чиқувчи ҳажмли маълумотларнинг гипер параметри—чуқурликдан бошлаймиз. Қўлланиладиган филтрлар сони билан эквивалентдир; ҳар бир филтр кириш қисмидаги маълумотларни кўриниши бўйича фарқларини ўрганади. Масалан, биринчи конволюция қатлами кириш қисмида қайта ишланмаган тасвирни маълумот сифатида қабул қилади, сўнгра чуқурлик ўқи бўйича нейронлар масалан маълум бирон ранглар тўпламини учратганида фаолланиши мумкин. Кирувчи тасвир ҳажмининг маълум бир соҳасига «Эътибор қаратган» нейронлар тўпламини чуқурлик устунни деб номлаймиз.

2. Иккинчиси филтр ўтадиган қадамни белгилаш лозим. Агар қадам 1га тенг бўлса, бир мартада филтр битта пиксельда силжиш амалга оширилади. Агар қадам 2 тенг бўлса (баъзан 3, бироқ амалда жуда ҳам камдан-кам) филтрлар ҳар доим 2 пиксель «сақраб» ўтади. Бу эса ўлчамли соҳа учун камрок натижа чиқариш имконини беради.

3. Маълумки кирувчи тасвир чегарасини ноллар билан қамраб олиш қулай ҳисобланади. Ноллар билан тўлдириш ўлчами учинчи гиперпараметр ҳисобланади. Ноллар билан тўлдиришнинг аҳамиятли томони бу чиқувчи натижавий тасвир ўлчамини бошқариш имконини беради (кўпинча кирувчи тасвир ўлчамини сақлаш учун ушбу хусусият қўлланилади, яъни кирувчи ва чиқувчи тасвир ўлчами бир хиллигини таъминлаш лозим бўлади).

Чиқувчи тасвир ўлчамини кирувчи тасвир ўлчами (W), конволюция қатлами нейронларининг сезувчанлик ўлчами (F), уларнинг силжиш қадами (S), кирувчи тасвирда тўлдирилган ноллар сони (P) қўллаган ҳолда функция орқали аниқлаш мумкин. Бунда $(W-F+2P)/S+1$ формула ёрдамида «тўғри келувчи» нейронлар сонини аниқлаш учун тўғри ҳисобланади. Масалан, кириш қисмида ўлчам 7×7 ва филтър 3×3 ўлчамда бўлиб, силжиш қадами 1 ва 0 тўлдиришга тенг бўлганда, чиқиш қисмида 5×5 ўлчамни оламиз. 2 қадам билан чиқишда 3×3 ўлчамни оламиз. Яна битта график мисолни кўрамиз.



31-расм. Нейронларнинг фазовий жойлашуви.

Нейронларнинг фазовий жойлашуви иллюстрацияси. Ушбу мисолда фақат битта текислик ўлчами (x ўқи) мавжуд, битта нейрон $F=3$ сезувчанлик майдонига эга, кириш ўлчами $W=5$ ва ноллар билан тўлдириш $P=1$ га тенг.

Чапда: кириш қисми бўйича $S=1$ қадам билан чўзилган нейрон, чиқувчи натижани $(5-3+1)/1+1 = 5$ ўлчамда чиқаради.

Ўнгда: $S=2$ қадамга эга нейрон бўлиб, чиқувчи ўлчами $(5-3+1)/2+1 = 3$ га тенг бўлади. Эътибор беринг, $S=3$ қадами тўлиқ ҳажмни қамраб олмайди. Арифметика нуқтаи назари бўйича қўйидагича аниқланади: $(5-3+2) = 4$ ва у 3 га қолдиқсиз бўлинмайди. Ушбу мисолдаги

нейронлар оғирлик кўрсаткичи – $[1,0,-1]$ га тенг бўлади (расмда ўнгда кўрсатилган), шу сабабли уларнинг силжиши нолга тенг бўлади. Бу оғирлик барча нейронлар учун умумийдир.

Ноллар билан тўлдириш усулини қўллаш. Юқорида келтирилган мисолда кириш ўлчами 5 га тенг чиқиш қисмидаги ўлчам – 5 тенглигича қолди. Бу имконият сезувчанлик қатламнинг ўлчами 3 га тенг бўлганда пайдо бўлади, ҳамда 1 га тенг бўлган ноллар билан тўлдириш усулини қўладик. Агар ноллар билан тўлдириш мавжуд бўлмаганда, чиқувчи ҳажмли маълумот ўлчами 3 га тенг бўлар эди, чунки айнан 3 та нейрон тўлиқ ҳажм бўйича идеал силжиши мумкин эди. Умумий ҳолда ноллар билан тўлдиришларни $P = (F-1)/2$ деб белгилаб, қадамни $S=1$ деб ҳисобланса, кирувчи ва чиқувчи ҳажм бир хил фазовий ўлчамга эга бўлади. Ноллар билан тўлдириш усули айнан бошланғич катталиқни сақлаш, яъни эни ва бўйи ўлчамини ўзгармас сақлаш учун қўлланилади.

Қадамлардаги чекланишлар. Гиперпараметрларнинг фазовий жойлашуви ўзаро симметрик чекланишларга эга. Масалан, кирувчи ҳажмли маълумотнинг ўлчами $W=10$ га тенг бўлганда, ноллар билан тўлдириш мавжуд бўлмайди ($P=0$), фильтр ўлчами $F=3$ га тенг бўлганда, $S=2$ га тенг қадамни қўллаш имкони бўлмайди, чунки $(W-F+2P)/S+1 = (10-3+0)/2+1 = 4.5$ га тенг бўлади, яъни каср сон нейронлар симметрик жойлашмаганлигини билдиради. Демак, ушбу гиперпараметрларнинг жойлашуви нотўғри ҳисобланади ва **ConvNet** кутубхонаси қайта ўрнатиб, ноллар билан тўлдириш, кириш қисмидаги маълумотларни қирқиш, коррективровка бажариш имконини бермайди. «Конволюция нейрон тармоқлари архитектураси» да кўрсатиладиганидек, CNN ўлчамини ва барча ўлчамларни аниқлаш катта муаммо бўлиб, бу масалаларни ноллар билан тўлдириш ва лойиҳалашнинг баъзи принциплари билан ҳал этиш мумкин.

Амалий мисол. Крижевский Архитектураси ва бошқалар, ImageNet мусобақасида 2012 йил ғолиб чиққан бўлиб, тасвир ўлчами

[227×227×3]га тенг. Биринчи конволюция қатламида сезувчанлик майдони **F=11 га тенг бўлган**, қадами **S=4**, ҳамда ноллар билан тўлдириш мавжуд бўлмаган (**P=0**) нейронларни қўллаган. Бунда $(227-11)/4+1 = 55$ га тенг бўлиб, конволюция қатлами чуқурлиги **K=96 га чиқувчи ҳажмли маълумот ўлчами** [55×55×96] га тенг. Ҳар бир 55*55*96 нейронлар кейинги кириш қисмининг [11×11×3] ўлчамли сатҳи билан боғланган. Бундан ташқари барча 96 нейронлар тўплам бўйича турли оғирлик кўрсаткичига эга [11×11×3] ўлчамли сатҳи билан боғланган. Расмий равишда тасвир ўлчами 224×224 га тенг, аниқки бу нотўғри ҳисобланади, чунки барча амаллар бажарилганда $(224-11)/4+1$ ифода бизга бутун сонни натижа сифатида бермайди. Бу натижани тўғрироғи ноллар билан тўлдириш усулида олинган, бироқ бу ҳақида айтиб ўтилмаган.

1.13.7. Параметрларни ҳамкорликда қўллаш

Конволюция қатламида параметрларни ҳамкорликда қўллаш принципи асосан параметрлар сонини назорат этиш учун қўлланилади. Юқорида келтирилган амалий мисол биринчи конволюция қатламида $55*55*96 = 290\,400$ нейронларга эга ва уларнинг ҳар бири $11*11*3 = 363$ оғирликка ва 1 силжишига эга. Умумий ҳисоблаганда $290400 * 364 = 105\,705\,600$ биринчи конволюция нейрон тармоқлари қатламида параметрлар сонини ҳосил қилади. Чунончи бу жуда катта сон ҳисобланади.

Демак, қўйидаги усул ёрдамида параметрлар сонини қисқартиришимиз мумкин: агар битта хосса маълум бир фазовий кенглик (x, y) да ҳисоблашларда тўғри келса, айнан шу хоссани (x2,y2) юзада ҳам қўллаш самарали бўлиши. Бошқача қилиб айтганда, битта 2-ўлчовли чуқурлик қирқими – **чуқурликлар қирқими** (масалан, [55×55×96] ўлчамли ҳажмли маълумотлар тўпламида [55×55] ўлчамга эга 96та чуқурлик кўрсаткичига эга), бир хил оғирлик ва силжишларни қўллаш учун ҳар бир қирқим нейронларини сақлаймиз. Параметрларни тақсимлаш схемаси туфайли конволюциянинг биринчи қатлами (ҳар бир чуқурлик қирқими бўйича) тахминан 96 та уникал оғирлик тўпламига эга бўлади ва

умумий сони $96 \cdot 11 \cdot 11 \cdot 3 = 34\,848$ уникал оғирликка ёки 24 944 параметрга (+96 силжишлар) эга. Барча $55 \cdot 55$ нейронлар ҳар бир чуқурлик қирқимда бир хил параметрларни қўллашни бошлайди. Амалиётда тескари боғланиш орқали хатоликни текшириш усулида ҳар бир нейрон хажмли маълумотлар тўпламида оғирлик градиентини аниқлайди, бироқ чуқурлик қирқимлари бўйича тўлдирилиб борилади ва ҳар бир қирқим учун битта оғирликлар тўпламини янгилаб боради. Эътибор беринг: битта чуқурлик қирқимидаги барча нейронлар битта оғирлик векторини қўлласса, ҳар бир чуқурлик қирқимидаги конволюция қатламидаги тўғри силжишларни нейрон оғирликлари конволюцион сифатида қабул қилиш мумкин (шу ҳолатдан келиб чиқган ибора– конволюцион қатлам). Айнан шу сабабли оғирликлар тўплами учун у умумий ҳисобланиб, кирувчи маълумотларни йиғилган ҳолатда филътр кўринишда қўллаш ҳисобланади.



32- расм. Крижевский ва бошқалар ўрганаётган филътрлар кўриниши.

Крижевский ва бошқалар ўрганаётган филътрларга мисоллар. Ҳар бир келтирилган 96 филътрлар $[11 \times 11 \times 3]$ ўлчамга эга, ва улар ҳамкорликда $55 \cdot 55$ нейронларни битта чуқурлик қирқимида қўллайди. Параметрларни ҳамкорликда қўллаш мантиқий боғланишга асосланади: агар тасвирнинг аниқ бир қисмида горизонтал боғланишлар аниқланса тасвирнинг бошқа қисмлари учун самарали ва тушунарли бўлиши лозим. Конволюция қатламининг $55 \cdot 55$ ҳажмдаги

турли қисмлари учун горизонтал боғланишни аниқлаш учун қайта ўрганиш мавжуд эмас.

Баъзан параметрларни ҳамкорликда қўллаш мақсадга мувофиқ эмас. Одатда CNN га узатилаётган тасвир марказлаштирилган структурага эга бўлган ҳоллар билан боғлиқ бўлиб, тасвир аксини қайта ишлашда айнан марказ учун алоҳида қолган қисмлари учун алоҳида хусусиятлар ўрганилиши лозим. Масалан, оддий мисол қилиб марказлаштирилган ҳолатда инсоннинг суратини кўриш мумкин. Табиийки, кўзлар билан боғлиқ хусусиятлар ва соч билан боғлиқ хусусиятлар турли текислик нуқталари асосида ўрганилиши лозим. Демак, бу ҳоллар учун параметрларни ҳамкорликда қўллаш схемасидан воз кечиб, унинг ўрнига **локал боғланган қатлам** схемасидан фойдаланиш мақсадга мувофиқ..

1.13.7.NumPy мисоллар

Юқорида келтирилган маълумотларни аниқлаштириш учун аниқ мисолни код кўринишда кўриб чиқамиз. Фараз қилайлик, барча кирувчи маълумотлар– бу NumPy-массив X. Бунда:

- Чуқурлик устун кўрсаткичи (x, y) нуқтасида фаоллашуви эса X[x, y:].
- D чуқурликдаги фаоллашув картасига эквивалент бўлган чуқурлик қирқими фаоллашуви X[d] бўлади.

Фараз қилсак, кирувчи ҳажм X (бу NumPy-массив) шакли X.shape: (11,11,4) кўринишда бўлсин. Кейинги қадам, фараз қилсак, ноллар билан тўлдириш қўлланилмасин, (**P=0**), филтер катталиги **F=5** га, қадам эса **S=2** га тенг бўлсин. Демак, чиқувчи ҳажмнинг фазовий ўлчами $(11-5)/2+1 = 4$ га тенг бўлади, яъни эни ва бўйини тўртга барабар деб ўрнатилади. Чиқувчи ҳажмда фаоллашув харитаси (V деб номласак) қуйидаги кўринишда бўлади (ушбу мисолда бир неча элементлар ҳисобланади):

- $V[0,0,0] = np. sum(X[:5:5:] * W0) + b0$
- $V[1,0,0] = np. sum(X[2:7:5:] * W0) + b0$
- $V[2,0,0] = np. sum(X[4:9:5:] * W0) + b0$
- $V[3,0,0] = np. sum(X[6:11:5:] * W0) + b0$

Эслатиб ўтамизки, NumPy да * амали массивларнинг элементлар бўйича кўпайтмасини билдиради. Бунда, оғирлик вектори W0 ушбу нейрон оғирлик вектори ҳисобланади, b0 эса – бу силжиш кўрсаткичи бўлиб, W0 қиймати W0.shape: (5,5,4) шакл кўрсаткичидан келиб чиқади, бунда филтър ўлчами 5 га тенг, кирувчи ҳажм чуқурлик кўрсаткичи– 4 га тенг. Ҳар бир нуктада оддий нейрон турлардагидек скаляр кўпайтмалар олиб борилади. Шу билан бирга (параметрларнинг ҳамкорликда қўллаш ҳисобидан) битта оғирлик ва силжиш қўлланилади, эни бўйича ўлчаш қадами 2 га тенг бўлади. Иккинчи фаоллаштирувчи харитани чиқиш ҳажмида кўриш учун қўйидаги амал бажарилади:

```

▪ V[0,0,1] = np. sum (X[:5:5:] * W1) + b1
▪ V[1,0,1] = np. sum (X[2:7:5:] * W1) + b1
▪ V[2,0,1] = np. sum (X[4:9:5:] * W1) + b1
▪ V[3,0,1] = np. sum (X[6:11:5:] * W1) + b1
▪ V[0,1,1] = np. sum (X[:5,2:7:] * W1) + b1 (у ўқи бўйича ўтиш)
▪ V[2,3,1] = np. sum (X[4:9,6:11:] * W1) + b1 (иккала ўқлар бўйича ўтиш)

```

Иккинчи фаоллашув харитасини ҳисоблашда бошқа параметрлар (W1) қўлланилишини инобатга олган ҳолда бу ерда иккинчи ўлчамлар V чуқурлик асосида индексация олиб борилади. Юқорида келтирилган мисолда қисқартирилган ҳолда V массивни қолган қисмларини тўлдириш учун конволюция қатлами бажарувчи баъзи амалларига тўхтаб ўтмадик. Эслатиб ўтамизки, фаоллашув харитаси фаоллашиши функцияси ёрдамида элементлар бўйича бирма-бир ҳаракатланишади, бироқ бу функция бу ерда келтирилмаган.

1.13.8. Конволюция қатлами бўйича амаллар кетма-кетлиги

Конволюция қатлами бўйича хулоса келтирамиз:

- **W1xH1xD1** ўлчамдаги маълумотларни қабул қилади
- 4 гиперпараметрни талаб этади:
 1. **K** филтърлар сони,

2. F фазовий тарқалиш,
3. қадами S ,
4. ноллар билан тўлдириш сонли ифодаси P .

• Ўлчам $W2 \times H2 \times D2$, ҳажмини кўради, бунда:

1. $W2 = (W1 - F + 2P) / S + 1$

2. $H2 = (H1 - F + 2P) / S + 1$ (эни ва бўйи бир хил ҳисобланади)

3. $D2 = K$

• Параметрларни ҳамкорликда қўллаш усули ёрдамида $F \cdot F \cdot D1$ оғирликларни филтрларга узатиш бажарилиб ва умумий ҳолда оғирликларни $(F \cdot F \cdot D1) \cdot K$ сонда ва K силжишда беради.

• Чиқиш ҳажмида d - чуқурлик кесими ($W2 \times H2$ ўлчамда) бу натижа бўлиб, бу кириш ҳажмида S қадамга эга d - силжиш асосида конволюция d - филтрланиш натижасидир.

Гиперпараметрларнинг кенг тарқалган қийматлари: $F=3$, $S=1$, $P=1$ га тенг. Бироқ, умумий тартиб ва эмпирик қоидалар мавжуд бўлиб, улар асосида гиперпараметрлар аниқ қийматлари белгиланади.

1.14. Демо-конволюция тармоғи

Қўйида ишлайдиган конволюцион - қатлам демо-версияси келтирилади. Уч ўлчовли маълумотларни визуаллаштириш мушкул бўлганлиги сабабли, барча ҳажмлар (кирувчи ҳажм (кўк), ҳажмлар оғирлик кўрсаткичи (қизил), чиқувчи ҳажм (яшил)) ҳар бири сатрда жойлашган чуқурлик қирқими асосида визуаллаштирилади. Кирувчи ҳажм ўлчами $W1=5$, $H1=5$, $D1=3$ тенг бўлсин, конволюция қатламининг параметрлари– $K=2$, $F=3$, $S=2$, $P=1$ га тенг. Яъни 3×3 ўлчамга тенг бўлган 2 филтрга эга бўлиб, у 2 қийматга тенг қадам билан ишлайди. Чунончи чиқувчи ҳажм фазовий ўлчами $(5-3+2)/2+1 = 3$ га тенг бўлади. Бундан ташқари, ноллар билан тўлдириш кўрсаткичи $P=1$ га тенг бўлган ҳолда кирувчи ҳажмга қўллаб ташқи чегараларини нолга ўзгартиради. Қўйида келтирилган визуализация чиқувчи фаоллашув қийматларини бирма-бир текшириб чиқади (яшил) ва ҳар бир элемент айнан чиқувчи маълумотларни (кук) элементларини ўзаро кўпайтириш ва филтрлаш(қизил) орқали

ҳисобланишини, ҳамда қийматларни кетма-кет суммалаш ва натижаларни силжиш бўйича коррективка қилишини ифодалайди.

1.14.1. Матрицаларни кўпайтириш кўринишда қўллаш.

Конволюция амали асосан кирувчи маълумотлар локал соҳалари билан филтър кўрсаткичлари ўртасида скаляр кўпайтмани бажаради. Конволюция қатламини қўллашнинг умумий принципи юқорида келтирилган хусусиятни бажарган ҳолда конволюция қатламининг ўтиши натижасида ягона катта матрица кўринишга келтириш ҳисобланади:

1. Кирувчи тасвирнинг локал қисмлари жараён давомида устунлар кўринишда чўзилади ва улар одатда **im2col** деб номланади. Масалан, кириш қисмига $[227 \times 227 \times 3]$ ўлчамдаги тасвир қабул қилинса, 4 кадам асосида филтърлар ёрдамида $[11 \times 11 \times 3]$ ўлчамгача қисқартирилиши мумкин, сўнгра $[11 \times 11 \times 3]$ ўлчамдаги пикселлар блокини вектор кўринишда алоҳида вектор-устун кўринишда $11 \times 11 \times 3 = 363$ ўлчамда чўзиш мумкин бўлади. Ушбу жараённинг такрорланиши эни ва бўйига нисбатан $(227 - 11) / 4 + 1 = 55$ та ўрин бериб, X_col матрицани $[363 \times 3025]$ ўлчамда беради, бунда ҳар бир устун чўзилган майдонга эга бўлади (умумий ҳолда $55 \times 55 = 3025$ майдонга эга бўлади).

2. Конволюция қатламининг оғирлик кўрсаткичи мос равишда сатр бўйлаб чўзилади. Масалан, агар $[11 \times 11 \times 3]$ ўлчамга эга 96 филтър мавжуд бўлса, $[96 \times 363]$ ўлчамли W_row матрицасини беради.

3. Конволюция жараёнининг натижаси битта катта кўпайтириш матрицасининг $pr.dot (W_row, X_col)$ амалга оширилиши билан эквивалент бўлиб келмоқда, бунда ҳар бир филтър ва майдон ўртасида скаляр кўпайтма ҳисобланади. Кўрилаётган мисолда ушбу амал натижаси $[96 \times 3025]$ кўринишда бўлади, чиқиш қисмида ҳар бир локал қисм учун филтър кўпайтмаси ҳосил қилинади.

4. Натижа эса $[55 \times 55 \times 96]$ ўлчамга нисбатан бошланғич кўринишга келтирилиши лозим:

Бундай ёндашув, сезиларли камчиликка эга бўлиб, X_col матрицаси баъзи қийматларининг кўп маротаба такрорланиши сабабли хотирадан самарали фойдаланиш имконини бермайди. Бошқа

томондан, асосий устунлиги матрицаларнинг самарали кўпайтмалари тўпламидан мақсадга мувофиқ равишда қўллаш мумкин. $im2col$ иш принципи ғоясини пулингни ишга туширишда қўллаш мумкин.

1.14.2. Тескари боғланиш орқали хатоликни текшириш усули.

Конволюция амали учун тескари кайтиш жараёни (берилганлар учун ҳам, оғирликлар учун ҳам) – бу ўша оддий конволюция амали бўлиб, фазовий тўнтарилган филтрларга эга бўлади.

1x1 ўлчамда конволюция

Энг аввал шуни айтиш лозимки кўпгина тажрибаларда (масалан, **Network in Network**) 1x1 форматдаги конволюция усули қўлланилган. Бу кўринишдаги конволюция усулини хусусан иккиламчи сигналларда қўллаш умуман ноқулай ҳисобланади, чунки одатда сигналлар 2 ўлчамли бўлиб, 1x1 ўлчамдаги конволюция аҳамиятсиз бўлиб қолади (бу оддий нуқтали масштаблаш бўлиб қолади). Бироқ CNN да бу ўзгача бажарилади, чунки 3-ўлчовли маълумотлар билан ишлаб, филтр тўлиқ фазовий чуқурлик бўйича ўтиб филтрланади. Масалан, кириш қисмида $[32 \times 32 \times 3]$ ўлчамдаги тасвир қабул қилинса, 1x1 ўлчамдаги конволюция амали бажарилса, у ҳолда самарали равишда 3-ўлчовли скаляр кўпайтма бажарилади (чунки кириш қисми 3 каналдан иборат).

1.14.3. Кенгайтирилган конволюция

Янги тадқиқотлар (**Фишер Ю. ва Владлен Колтун мақоласида**) натижасида конволюция қатламининг яна битта гиперпараметри – дилатация (кенгайтириш) аниқланди. Юқорида конволюция қатламининг кетма-кет жойлашган филтрлари ишини кўриб чикдик, бироқ шундай филтрлар мавжудки, ҳар бир ячейка оралаб (сакраб) ишлайди. Бу ораликлар кенгайтмалар деб номланади. Мисол: 3 ўлчамга тенг w филтрнинг битта ўлчамига нисбатан x кириш қисмида $w[0]*x[0] + w[1]*x[1] + w[2]*x[2]$ амали бажарилсин. Бу 0 дан дилатация эканлигини билдиради. 1 дилатацияси учун куйидаги ҳисоблашлар олиб борилиши лозим: $w[0]*x[0] + w[1]*x[2] + w[2]*x[4]$. Бошқача қилиб айтганда 1 га тенг оралик мавжуд бўлиб қолади. 0-кенгайтмали филтрлар билан мувофиқлашган

қўлланилиши баъзи ҳоллар учун самарали ҳисобланади, чунки кирувчи фазовий берилганларни кам қатламлар асосида бирлаштириш имконини беради. Масалан, ҳар бир филтър устки қисмида 2та 3x3 ўлчамли конволюция қатлами ўрнатилса, 2-қатлам нейронлари 5x5 ўлчамдаги кириш қисми функцияси (яъни 5x5 нейронлар самаралилик майдони) бўлиб қолади. Агар кенгайтирилган конволюция қатлами қўлланилса, самаралик майдони тезроқ кенгайиб боради.

1.15. Пулинг қатлам

CNN архитектурасида одатда конволюция қатламли кетма-кетлигида ораларига пулинг (қуйи танлаш) қатламини ўрнатиш оддий ҳолат бўлиб ҳисобланади. Унинг вазифаси тасвир фазовий ўлчамини босқичма-босқич кичрайтириш бўлиб, тармоқда параметрлар ва ҳисоблашлар сонини камайтириши, ҳамда қайта ўрганиш жараёнини назорат қилиш имконини бериш ҳисобланади. Пулинг қатламга кирувчи берилганларнинг фазовий чуқурлиги кўрсаткичидан боғлиқсиз ҳолда максимум функциясини қўллаган ҳолда тасвирни фазовий масштаблаш амалини бажаради. Кўпинча 2x2 ўлчамга эга ва қадами 2 га тенг филтърли қатлам қўлланилади; бундай қатламга кирувчи берилганларни фазовий чуқурлигини ҳар бир қисм учун ҳам энига ҳам бўйига дискретизация бўйича 2 баробар қискартиради, бунда 75% га халос этилади. Ҳар бир МАХ амали 4 та қийматдан энг максималини танлайди. Бунда фазовий чуқурлик ўлчами ўзгармас қолади. Умуман олганда, пулинг қатламини қўйидагича ишлайди:

- **W1xH1xD1** ўлчамга эга берилганларни кириш қисмида қабул қилади

- 2та гиперпараметрни талаб этади:

1. **F** унинг давомийлигини,

2. **S** қадамини.

- Бошланғич параметрларни киритади, чунки аниқ белгиланган бошланғич функцияни ҳисоблайди.

Қуйи танлашларда ноллар билан тўлдириш қўлланилмайди. Амалиётда пулинг қатламини фақат 2та вариациясини **F=3**, **S=2** гиперпараметрлари билан (умумлаштирувчи қуйи танлаш деб

номланади) ва энг кенг тарқалган қатлам $F=2$, $S=2$ гиперпараметрларга эга кўринишлари. Катта ҳажмдаги майдонга эга пулинглар деструктив ҳисобланади.

1.15.1. Умумий пулинг

Максимал қуйи танлаш ҳолати учун қўшимча сифатида шуни айтиш лозимки, пулинг қатламлар бошқа функцияларни ҳам бажариши мумкин. Масалан, қуйи танлашларнинг ўртача кўрсаткиччилиги ёки L2-нормаллашган танловни ҳосил қилиш. Эҳтимоллик жиҳатдан ўрталашган танлаш кўпинча қўлланилиб келинган бироқ, охириги вақтларда қўлланилиши қулай бўлган максимизацияланган танловга нисбатан иккиламчи бўлиб колди.

Тескари боғланишда хатоликни аниқлаш усулида

Конволюция қатламини ўрганиш давомида $\max(x, y)$ амали учун тескари боғланишда хатоликни аниқлаш жараёни кириш қисми учун градиент сифатида қабул қилинади, қайсики, тўғри боғланишда кенгрок қийматларга эга бўлган. Бинобарин, тўғри боғланишда пулинг қатламида максимал фаоллашиш индексини аниқлаш жараёни бажарилиши лозим, чунки тескари боғланишда градиентлаш йули самарали белгиланиши керак бўлади.

1.15.2. Пулингдан холис бўлиш

Баъзи мутахассислар пулинг амалини ортиқча ҳисоблаб, қўлламайди. Масалан, Д. Т. Спрингенберг, А. Досовицкий, Т. Брокс ва М. Ридмиллер ўзининг ишида пулинг қатлаמידан холис бўлишни таклиф этиб, унинг ўрнига архитектурада конволюция қатламларни такроран қўллаш усулини таклиф этган. Тасвир ифодаланиши ўлчамини қисқартириш учун катта ҳажмдаги конволюция қатламини қўллашни тавсия қилди. Субдискретизацияни рад этиш генератив моделларда ўрганиш жараёни учун вариацион автосоциатор (VAE) ёки рақобатдошли генератив тармоқ (GAN) кўринишда асосий ўринни ўйнайди. Яқин орада конволюцион нейрон тармоқ архитектураси пулинг қатлаמידан ҳосил бўлади ёки жуда кам сонда қўлланила бошланади.

1.16. Тўлиқ боғланган қатлам

Оддий нейро тармоқлар каби тўлиқ боғланган қатламда нейронлар олдинги қатлам барча фаол нукталари билан тўлиқ боғланишга эга. Уларнинг фаоллашув кўрсаткичлари матрицалараро силжиш жараёнида матрицаларни кўпайтириш орқали аниқланади.

1.16.1. Тўлиқ боғланган қатламларни конволюцион қатламга ўзгартириш

Тўлиқ боғланган қатламлар ва конволюцион қатлам ўртасидаги фарқ шундан иборатки, конволюция қатлами нейронлари кириш қисми локал соҳалари билан боғланган ва улар ҳамкорликда параметрлардан фойдаланиши мумкин. Бироқ икки қатламда ҳам нейронлар ўз хусусиятлари билан фарқлансада скаляр кўпайтмани ҳисоблайди ва шунинг учун улар функционал кўриниши айнан бир хил. Бундан ташқари, бу икки кўринишдаги қатламлар негизида конвертациялаш мумкин:

- Барча кўринишда конволюцион қатлам учун унинг вазифасини бажариши мумкин бўлган тўлиқ боғланган қатлам мавжуд. Оғирликлар матрицаси катта ҳажмдаги матрица кўринишда бўлиб, баъзи блоклардан ташқари (локал боғланганлик сабабли) у асосан 0 лар билан тўлдирилган, кўпгина оғирлик кўрсаткичлари (параметрларни ҳамкорликда қўлланилиши учун) тенг бўлади.

- Ва акси, барча кўринишда тўлиқ боғланган қатлам конволюцион қатламга ўзгартирилиши мумкин. Масалан, тўлиқ боғланган қатлам чуқурлиги $K=4096$ тенг бўлиб, $[7 \times 7 \times 512]$ ўлчамга тенг бўлган кириш берилганларига йўналтирилган бўлсин, уни қуйидаги параметрларга эга конволюцион қатламга ўгириш мумкин: филтър ўлчами $F=7$, ноллар билан тўлдириш мавжуд эмас ($P=0$), қадами $S=1$ ва филтър чуқурлиги $K=4096$ га тенг. Бошқача қилиб айтганда, филтър ўлчамини кирувчи берилганлар ўлчами билан бир хил ўрнатилади. Чиқиш қисмида $[1 \times 1 \times 4096]$ ўлчамни оламиз, чунки чуқурликнинг бинар устунни барча кириш берилганлари бўйича «тўғри» тартибда ўтади, шунинг учун бошланғич тўлиқ боғланган қатлам берадиган натижани беради.

1.16.2.Қайта ўзгартириш роли

Тўлиқ боғланган қатламларни конволюцион қатламга ўзгартириш имкони амалда кенг қўлланилмоқда. Кириш қисмида $[224 \times 224 \times 3]$ ўлчамга эга тасвирни қабул қилиб, унинг ҳажмини фаоллашиш ҳажмига $[7 \times 7 \times 512]$ гача қисқартириш учун бир қатор конволюция ва пулинг қатламларни қўлловчи конволюцион нейрон тармоқни кўриб ўтамиз. AlexNet архитектурасида бу амал 5 та пулинг қатламини қўллаш орқали амалга оширилиб, унда кирувчи фазовий тасвир дискретизацияси қисқартирилади ва натижада тасвир ҳажми $224/2/2/2/2/2 = 7$ га тенг бўлиб қолади. AlexNet архитектураси 2 та 4096 ҳажмга эга тўлиқ боғланган қатламини ва классларни ҳисоблаш учун 1000 нейронга эга 1 та тўлиқ боғланган қатламини қўллайди. Изоҳланган алгоритм бўйича 3 та қатламдан барчасини йиғувчи қатламга ўзгартиришимиз мумкин:

1. $[7 \times 7 \times 512]$ ўлчамдаги кириш қисмига йўналтирилган тўлиқ боғланган биринчи қатламини йиғувчи қатлам билан ўзгартиришимиз мумкин, бунда унинг фильтри ўлчами $F=7$ га тенг бўлиб, чиқувчи берилганлар ҳажми $[1 \times 1 \times 4096]$ кўринишда бўлади.

2. Иккинчи тўлиқ боғланган қатламини филтър ўлчами $F=1$ га тенг бўлган йиғувчи қатлам билан ўзгартириш мумкин, бунда чиқувчи берилганлар ҳажми $[1 \times 1 \times 4096]$ кўринишда бўлади.

3. Учинчи тўлиқ боғланган қатламини филтър ўлчами $F=1$ га тенг бўлган йиғувчи қатлам билан ўзгартириш мумкин, бунда чиқувчи берилганлар ҳажми $[1 \times 1 \times 1000]$ кўринишда бўлади.

Ушбу ҳар бир қайта ўзгартиришлар натижасида конволюция қатламини филтрида оғирлик кўрсаткичлари матрицасини W ўзгартириши (масалан, кўринишини ўзгартириш) мумкин. Демак, бу ўзгартиришлар асосида бир мартага “ўтиш” да CNNни катта ҳажмдаги тасвир кўпгина соҳалари бўйича ўтказиш мумкин.

Масалан, тасвир ўлчами 224×224 бўлиб, унинг ҳажми $[7 \times 7 \times 512]$, яъни 32 га қисқартирилди. У ҳолда 384×384 ўлчамдаги тасвирни ушбу ўзгартирилган архитектура бўйича ўтказилса унга эквивалент бўлган $[12 \times 12 \times 512]$ ҳажмдаги тасвирни беради, чунки $384/32 = 12$. Натижада

битта [1x1x1000] ҳажмга эга баҳолаш вектори ўрнига, тўлиқ 6×6 баҳолаш массиви синфига эга бўлиб, 384×384 тўлиқ ҳажми бўйича ўтказилади.

Конволюция нейротармоғи 224x224 ўлчамда 384×384 тасвир бўйича 32 пиксел қадам билан ўтишининг натижаси ва бирламчи конвертацияланган CNN ўтиш билан бир хил натижа беради.

Табиийки, қайта ишланган нейрон тармоқнинг бир марталик ўтиши, оригинал CNN нинг 36 позиция бўйича бир неча мартаба итерацияланишига нисбатан самарали ҳисобланади. Бу усул одатда унумдорликни кўтариш учун амалда қўлланилади. Масалан, тасвир ўлчамини катталаштириш лозим бўлганда конвертацияланган конволюцион нейротармоқ қўлланилади ва барча тасвир фазовий қисмлари баҳоланади ва у ўртача кўрсаткичга келтирилади.

CNN қадамини 32 пикселдан паст ҳолатда бошланғич кўринишда тасвирга жорий этиш учун тўғри ўтиш қийматларини қайта кўпайтириш орқали амалга ошириш мумкин. Масалан, агар 16 пикселли қадамни қўллаш лозим бўлса, ўзгартирилган конволюция нейротармоғи бўйича олинган натижаларни икки мартаба бирлаштириш лозим бўлади: энг аввал бошланғич тасвир ўтказилади, сўнгра - тасвир эни ва бўйи бўйича 16 пикселга сурилган ҳолатда иккинчи мартаба ўтказилади.

1.17.CNN архитектураси

Кўриб ўтганимиздек, конволюцион нейротармоқлар одатда 3 хил асосий қатламдан иборат, буларга: конволюция, пулинг ва тўлиқ боғланган қатламлар киради. Фаоллаштириш функцияси ҳам қатлам кўринишда жорий этилиб, улар чизиксиз функция кўринишда ҳар бир элемент асосида олиб борилади. Қуйида ушбу қатламлар ўзаро ҳамкорликда тўлиқ CNN ҳосил қилишини кўриб ўтаемиз.

1.17.1.Қатламлар кўриниши

Кўпинча CNN архитектурасини куришда бир неча қатламлар кетма-кетлиги қўлланилиб, конволюция, ректификация қатламли бир неча марта кўйилиб, улардан сўнг пулинг қатлами кўйилади; ушбу шаблон бошланғич тасвир жуда кичик ўлчамга етказилгунча

қўлланилади. Маълум бир вақтдан сўнг тўлиқ боғланган қатламларга ўтиш амалга оширилиб, сўнгги тўлиқ боғланган қатлам чиқувчи маълумотга, масалан синфини аниқлаш маълумотига эга бўлади. Бошқача қилиб айтганда, энг кенг тарқалган CNN архитектураси қуйидаги схема кўринишда бўлади:

INPUT -> [[CONV -> RELU]*N -> POOL?]*M -> [FC -> RELU]*K -> FC

Бунда «*» такрорланишини билдиради, POOL? Эса қўшимча пулинг қатламини кўрсатади. Бундан ташқари, $N \geq 0$, (одатда $N \leq 3$), $M \geq 0$, $K \geq 0$ (одатда $K < 3$) тенг бўлади. Қуйида келтирилган намуна бўйича CNN архитектураси келтирилган:

- INPUT -> FC, чизикли классификаторни жорий этади. Бунда $N = M = K = 0$.
- INPUT -> CONV -> RELU -> FC
- INPUT -> [CONV -> RELU -> POOL]*2 -> FC -> RELU -> FC. Бунда ҳар бир пулинг қатлами орасида конволюция қатлами жойлашган.
- INPUT -> [CONV -> RELU -> CONV -> RELU -> POOL]*3 -> [FC -> RELU]*2 -> FC. Бунда 2 конволюция қатлами ҳар бир пулинг қатлами орасида ётади.

Йирик ва чуқур тармоқларда бу архитектуруни қўллаш самарали натижа беради, чунки бир нечта конволюцион қатламларнинг бир неча марта қўлланилиши кириш қисмидаги маълумотларнинг тўлиқ хусусиятларини очиқ беришга ва сўнгра уларни пулинг қатламида қўллаш имконини беради.

Одатда, йирик конволюция майдонини қўллашдан кўра, бир нечта катта бўлмаган филтрлаш қатламларидан фойдаланиш самарали ҳисобланади. Фараз қилинг, қатламларда чизиксиз ўтишни инобатга олиб, 3×3 ўлчамдаги конволюция қатламини бирма-бир қўямиз. Бунда

хар бир нейрон биринчи конволюция қатлами кириш қисмида 3×3 ўлчамга эга бўлади. Иккинчи қатламда нейронлар биринчи қатлам бўйича 3×3 ўлчамга ва кириш қисмининг 5×5 ўлчамга эга бўлади. Худди шундай учинчи қатламда нейронлар 3×3 ўлчам иккинчи қатлам бўйича ва кириш қисмининг 7×7 ўлчамга эга бўлади. Фараз қилинг, ушбу учта конволюция қатлами ўрнига битта 7×7 ўлчамли рецептив майдонга эга битта CONV-қатлам қўлланилсин. Бу қатлам нейронлари кириш қисми (7×7) кенгликда рецептив майдонига эга бўлади, бироқ бир неча камчиликлар мавжуд.

Биринчидан, 3та конволюция қатлампдан иборат стек қатлам хусусиятларини ифодаловчи чизиксиз функцияга эга бўла туриб, кириш қисмининг чизикли функциясини ҳисоблайди.

Иккинчидан, барча кириш қисми берилганлари тўлалигича C та каналларга эга. Унда битта конволюция қатлами ўлчами 7×7 бўлиб, S_x эса $(7 \times 7 \times C) = 49C^2$ параметрларига эга бўлиб, 3 қатламдан иборат стек 3х фақат $(S_x (3 \times 3 \times C)) = 27C^2$ параметрга эга бўлади. Маълумки, кичик филътрага эга CONV –қатламларнинг бир нечтасини «қўллаш» усули кам параметрлар ёрдамида кирувчи маълумотлар хусусиятларининг кучли томонларини ифодалашга имкон беради. Бу ерда битта камчилик бу оралиқ даврда олинадиган натижаларнинг барчасини сақлаш учун катта ҳажмдаги хотирани талаб этилиши ҳисобланади.

Амалиётда эса ImageNet маълумотлар базаси билан ишловчи усул қўлланилади. Мутахассислар керакли архитектурани танлашда 90% сараланиб четлаштирилади, чунки айнан ImageNet билан ишловчи архитектурани танлаш, ўрганиш жараёнидан ўтган модельни юклаш ва қўйилган масала мослаштириб сошлашнинг ўзи кифоя. Конволюция нейротармоғини нольдан бошлаб ўрганиш мутахассислардан талаб этилмайди.

1.17.2. Аниқ мисоллар

Конволюция нейротармоқларнинг энг кенг тарқалган архитектурасини кўриб чиқамиз:

- **LeNet.** энг биринчи бўлиб Ян Лекун 1990-йилларда конволюция нейро тармоғини самарали кўллаб архитектура ишлаб чиқишга эришди. **LeNet архитектураси** почта индекслари, рақамлар ва ҳ.к. аниқлаш ва ўқиб олишда кўлланилган.

- **AlexNet.** Бу архитектура Алекс Крижевский, Илья Суцкевер ва Джефф Хинтон ишлари натижасида эришилган бўлиб, CNN нинг компьютер видеокузатув соҳасида машҳур бўлишига олиб келди. **AlexNet архитектураси** 2012 йил **ImageNet ILSVRC Challenge** да тақдим этилиб, барча рақобатларда ютиб чиқди (иккинчи ўринни эгаллаган архитектурада 26% хатоликка нисбатан 16% хатолик билан ютди).

- **ZF Net.** 2013 йил ILSVRC ютган Мэтью Зеллер ва Роб Фергюснинг конволюция нейрон тармоғи бўлиб, **ZF Net** (аббревиатураси Zeiler ва Fergus) ном билан машҳур. Ушбу архитектура AlexNet версиясининг такомиллаштирилган ҳолати бўлиб, конволюция тармоғи ўртача ўлчами оширилиб, бирини қатлам филътрининг қадами камайтирилди.

- **GoogLeNet.** 2014 йили юқорида кўрсатилган танловда ютиб чиқган **Шегеди** Google корпорациясининг бошқа ходимлари томонидан ишлаб чиқилган CNN. Бу архитектуранинг ютуғи шундан иборатки, кириш қисми модулини (Inception Module) ишлаб чиқиш ва жорий этиш бўлиб, параметрлар сонини 60 млн дан 4 млнгача қисқартириш имконини берди. Параметрларнинг қисқартирилиши тармоқ юқори қисмидаги тўлиқ боғланган қатламларни ўрта пулинг қатламлари билан алмаштириш ҳисобидан амалга оширилди.

- **VGGNet.** GoogLeNet дан кейин тез орада ILSVRC 2014 танловида **Карен Симонян ва Эндрю Циссерман** тармоғи эълон қилинди. Мутахассислар унинг имкониятларини намоиш этиб, асосий унумдорлик омили бу тармоқнинг чуқурлик кўрсаткичидир. Бу тармоқ 16 та конволюция ва тўлиқ боғланган қатламга эга бўлиб, бир жинсли архитектура 3×3 ўлчамда конволюция ва 2×2 пулинг амалини бошидан охиригача бажаради. Бошланғич модель Caffe фреймворкда чуқур ўрганиш учун Plug and Play режимида ишлайди.

• **ResNet. Қолдиқли тармоқ (Residual Network)**, Кайминг Хе томонидан ишлаб чиқилган бўлиб, ILSVRC танловида ғолиб чиқди. Асосий хусусиятлари – пакетли нормаллаштиришни интенсив ва махсус скип-боғланишлар қўлланилади. Архитектура сўнгида тўлиқ боғланган қатламлар мавжуд эмас. Ҳозирги кунда ResNet конволюцион нейротармоқлар соҳасида кенг қўлланиладиган ва энг юқори ютуқ ҳисобланади.

1.17.3.VGGNet изоҳи

VGGNet тўлиқ изоҳини кўриб ўтамиз. Тармоқ тўлалигича конволюция қатламларидан иборат бўлиб, 3×3 ўлчамда 1-қадам билан конволюция амалини бажаради, 1 билан тўлдирди. Шу билан бирга пулинг қатлами 2×2 ўлчамда максимал танловни 2-қадам билан амалга оширади ва ноллар билан тўлдирмайди. Ҳар бир қадамда ўлчам ўзгаришини ифодалаш ва оғирлик кўрсаткичи сони ўзгаришини кузатиш мумкин:

```

INPUT: [224×224×3] хотира: 224*224*3=150К оғирлик кўрсаткичи:
0
CONV3-64: [224×224×64] хотира: 224*224*64=3.2М оғирлик
кўрсаткичи: (3*3*3)*64 = 1,728
CONV3-64: [224×224×64] хотира: 224*224*64=3.2М оғирлик
кўрсаткичи: (3*3*64)*64 = 36,864
POOL2: [112×112×64] хотира: 112*112*64=800К оғирлик
кўрсаткичи: 0
CONV3-128: [112×112×128] хотира: 112*112*128=1.6М оғирлик
кўрсаткичи: (3*3*64)*128 = 73,728
CONV3-128: [112×112×128] хотира: 112*112*128=1.6М оғирлик
кўрсаткичи: (3*3*128)*128 = 147,456
POOL2: [56×56×128] хотира: 56*56*128=400К оғирлик кўрсаткичи:
0
CONV3-256: [56×56×256] хотира: 56*56*256=800К оғирлик
кўрсаткичи: (3*3*128)*256 = 294,912
CONV3-256: [56×56×256] хотира: 56*56*256=800К оғирлик
кўрсаткичи: (3*3*256)*256 = 589,824

```

CONV3-256: [56×56×256] хотира: $56*56*256=800\text{К}$ оғирлик кўрсаткичи: $(3*3*256)*256 = 589,824$

POOL2: [28×28×256] хотира: $28*28*256=200\text{К}$ оғирлик кўрсаткичи: 0

CONV3-512: [28×28×512] хотира: $28*28*512=400\text{К}$ оғирлик кўрсаткичи: $(3*3*256)*512 = 1,179,648$

CONV3-512: [28×28×512] хотира: $28*28*512=400\text{К}$ оғирлик кўрсаткичи: $(3*3*512)*512 = 2,359,296$

CONV3-512: [28×28×512] хотира: $28*28*512=400\text{К}$ оғирлик кўрсаткичи: $(3*3*512)*512 = 2,359,296$

POOL2: [14×14×512] хотира: $14*14*512=100\text{К}$ оғирлик кўрсаткичи: 0

CONV3-512: [14×14×512] хотира: $14*14*512=100\text{К}$ оғирлик кўрсаткичи: $(3*3*512)*512 = 2,359,296$

CONV3-512: [14×14×512] хотира: $14*14*512=100\text{К}$ оғирлик кўрсаткичи: $(3*3*512)*512 = 2,359,296$

CONV3-512: [14×14×512] хотира: $14*14*512=100\text{К}$ оғирлик кўрсаткичи: $(3*3*512)*512 = 2,359,296$

POOL2: [7×7×512] хотира: $7*7*512=25\text{К}$ оғирлик кўрсаткичи: 0

FC: [1×1×4096] хотира: 4096 оғирлик кўрсаткичи: $7*7*512*4096 = 102,760,448$

FC: [1×1×4096] хотира: 4096 оғирлик кўрсаткичи: $4096*4096 = 16,777,216$

FC: [1×1×1000] хотира: 1000 оғирлик кўрсаткичи: $4096*1000 = 4,096,000$

TOTAL хотира: $24\text{М} * 4 \text{ bytes} \cong 93\text{МВ}$ / image (only forward! ~*2 for bwd)

TOTAL параметрлар: 138М параметрлар

Шунга эътибор бериш лозимки, хотиранинг кўп қисми энг биринчи конволюция қатламларида қўлланилади, параметрларнинг кўпчилиги эса тўлиқ боғланган қатламларда охиргиларида жойлашган. Ушбу

аниқ мисолда тўлиқ боғланган қатлам да 100 та оғирлик кўрсаткичи мавжуд, буларнинг умумий сони 140 млн.

1.17.4. Ҳисоблашлар бўйича тавсиялар

CNN қуришда асосий тўсиқ бу хотира. Замоनावий график процессорлар хотира лимити 3, 4 ёки 6 Гб, бироқ энг замонавий GPU 12 Гб га эга. Учта асосий хотира талаб этилади:

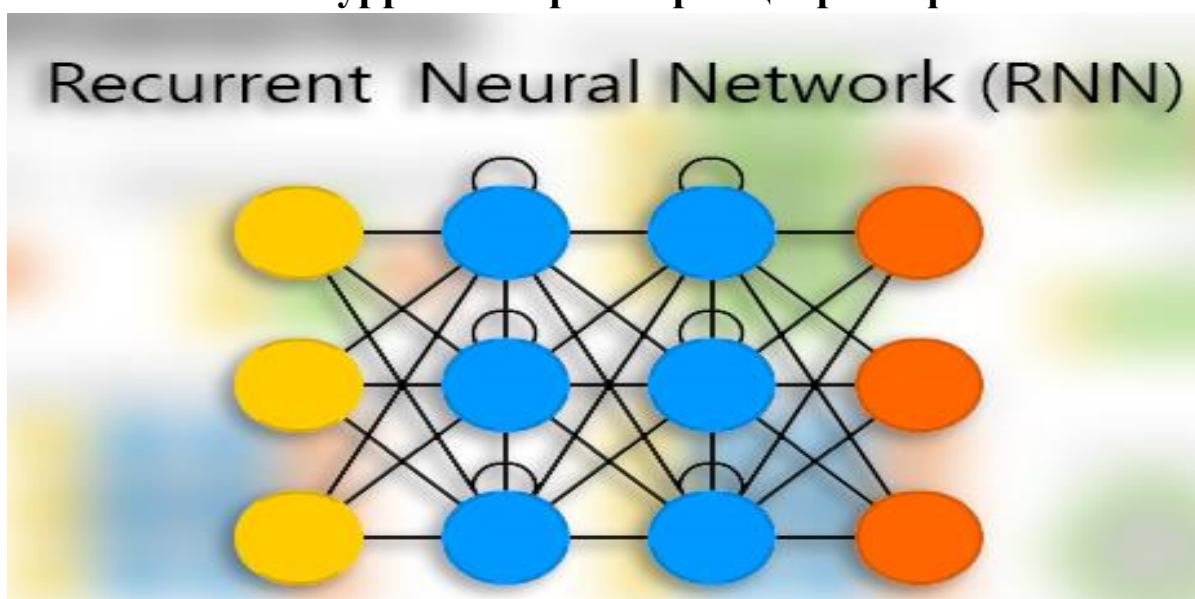
- Маълум бир ораликда берилганлар ҳажми. Бу ҳар бир CNN қатламида қайта ишланмаган фаоллашиш сигналлари ва уларнинг градиентлари. Одатда фаоллашиш сигналлари конволюция нейротармоқнинг биринчи қатламларида жойлашган бўлади. Шунини айтиш жоизки, фаоллашиш сигналлари тескари боғланишда хатоликни аниқлашда қўлланилади. Бироқ унумли фойдаланишда CNN фақатгина тадқиқот жараёнида қўлланилади ва натижада фаоллашувлар сони фақат жорий ҳолатни сақлаш ҳисобидан қисқариши мумкин. Бунда олдинги ҳолатлар пастки қатламларга бирин – кетин сурилади.




- Параметрлар ўлчами. Бу тармоқ параметрлари сони, уларнинг тескари боғланишда хатоликни аниқлашдаги градиенти ва кэш кадамидан иборат. Демак, параметрлар векторини сақлаш учун талаб этиладиган хотира тахминан 3 га тенг бўлган коэффициентга кўпайтирилиши лозим.

- Барча конволюция нейротармоғининг қўлланилишида **кўп жинсли** хотирадан фойдаланиш лозим, яъни тасвир берилганлари пакети кўринишда бўлиб, янада такомиллашиш имкони билан қўллаш.

Қийматларнинг (фаоллашув, градиент ва ҳ.к.) умумий сони аниқ бўлгандан сўнг уни Гб ларга ўтказдириш керак бўлади. Қийматлар сони 4 га кўпайтирилса байтлар сони келиб чиқади, сўнгра 1 024 га бир неча марта бўлиш орқали хотирани килобайт, сўнг мегабайт ва гигабайтда оласиз. Агар қурилган тармоқ “мос келмаса”, уни пакет ўлчамини қисқартириш орқали «мослаштириш» мумкин, чунки хотиранинг кўп қисми айнан фаоллашув орқали эгалланади.

1.18.Рекуррент нейрон тармоқлар синфи

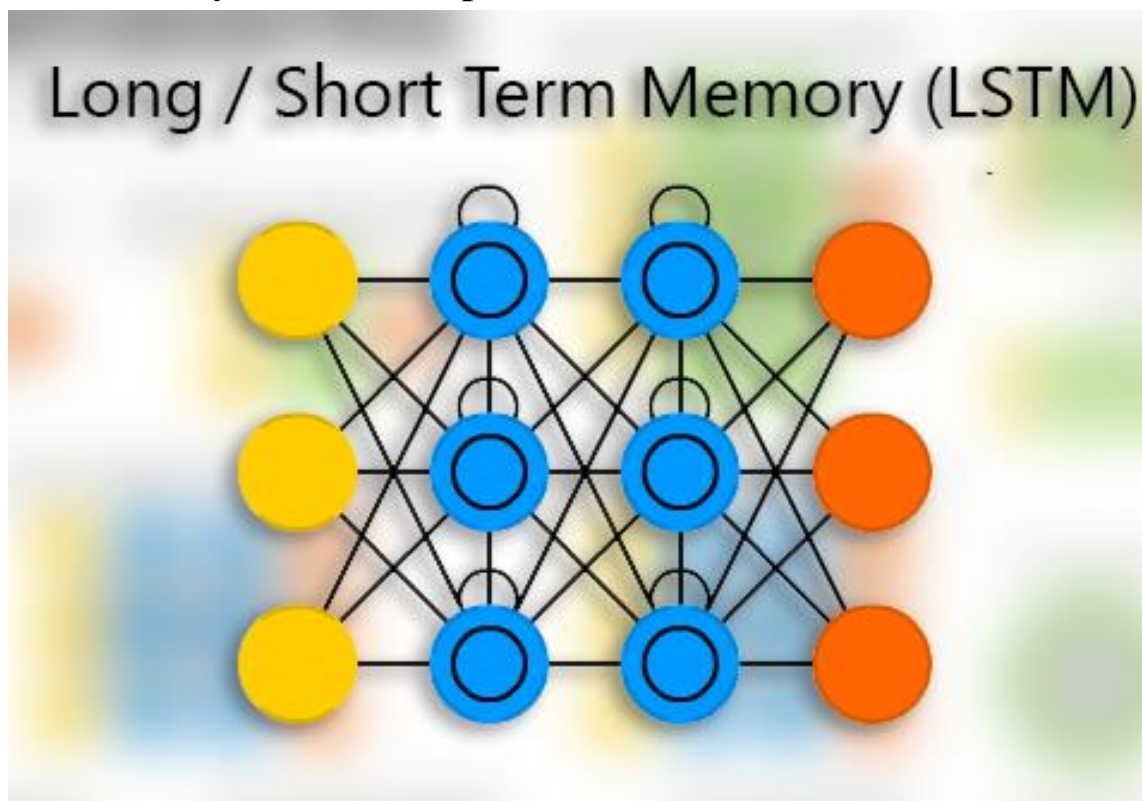





-  Кириш ячейкаси
-  Чиқиш ячейкаси
-  Такрорланган ячейка

33-расм. Рекуррент нейрон.

Рекуррент нейрон тармоқ (recurrent neural network, RNN) FFNN тармоқ туркумидаги тармоқ бўлиб, бироқ RNN нейронлар маълумотларни фақат олдинги қатламдан олиш билан чекланмайди, балки тўлиқ рекуррент тармоқдан қабул қилиб олади. Яъни келаётган барча маълумотлар кетма-кетлигини хотирада сақлашни ўрганади. RNN мураккаблиги – градиентнинг пасайиб бориб йўқолиши бўлиб, вақт ўтиши билан тармоқ сақлаётган маълумотларини йўқота бошлайди. Гарчи нейрон ҳолатига таъсир қилмасдан, фақат оғирлик кўрсаткичини аниқлашда таъсир қилсада, кетма-кетликлар ҳақидаги маълумотлар айнан уларда сақланади. Рекуррент нейрон тармоқлар асосан маълумотларни автоматик тўлдиришда қўлланилади.

Узоқ ва қисқа муддатли хотира



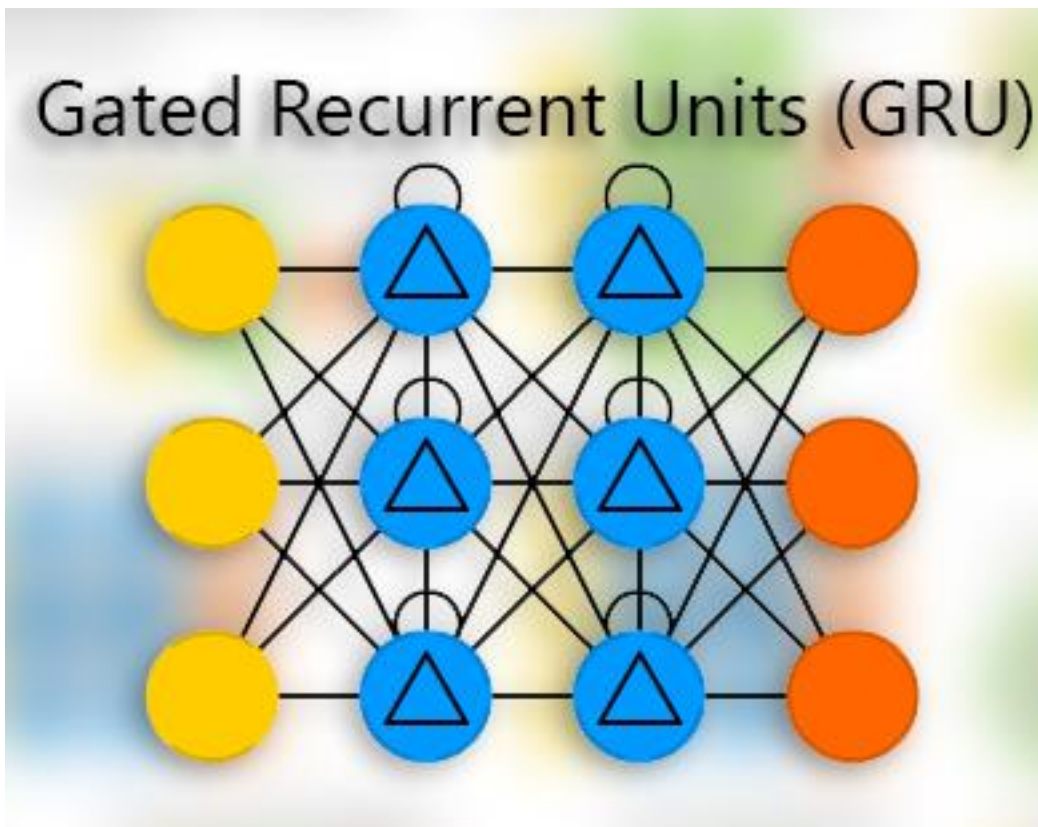
-  Кириш
-  Чиқиш ячейкаси
-  Турли хотирали ячейка




34-расм. Узоқ ва қисқа муддатли хотира.

Узоқ ва қисқа муддатли хотира (long short term memory, LSTM) тармоғи рекуррент нейрон тармоқлардаги маълумот йўқотиш муаммосини ҳал этади, бунда филтрларни ва мавжуд хотира ячейкасини қўллайди. Ҳар бир нейрон хотира ячейкасига ва учта филтрга эга: кирувчи, чиқувчи ва унутувчи. Филтр мақсади—маълумотларни химоялаш. Кирувчи филтр олдинги қатламдан сақлаш лозим бўлган маълумотлар ҳажмини белгилайди. Чиқувчи филтр эса, кейинги қатлам қабул қилувчи маълумотлар ҳажмини белгилайди. Унутувчи филтр хотирада қийматларни сақлаш даражасини назорат этади, масалан, агар тармоқ бирон бир китобни ўрганаётган жараёнида янги бўлимга ўтса, олдинги бўлимдаги баъзи

бир матнларни унутади. LSTM тармоғи мураккаб структураларни яратишга қодир, бироқ катта ҳажмдаги ресурсларни талаб этади.

Бошқариладиган рекуррент блоки



-  Кириш
-  Чиқиш ячейкаси
-  Хотира

35-расм. Бошқариладиган рекуррент блок.

Бошқариладиган рекуррент блоки (gated recurrent unit, GRU)– бу рекуррент нейрон тармоқларнинг мантиқий механизми бўлиб, узоқ ва қисқа муддатли хотира тармоғининг бир кўриниши ҳисобланади. GRU бир филътрага кам бўлиб, боғланиш кўриниши ҳам ўзгача: кириш, чиқиш ва унутиш филътри ўрнига битта янгилаш филътри қўлланилади. Бу филътр орқали олдинги ҳолатдан олинадиган маълумотни ва олдинги қатламдан олинадиган маълумотларни таҳлил қилади. Ҳолатни ташлаш филътри унутиш филътри каби ишлайди, бироқ жойлашуви бошқача. Кейинги қатламга ҳолат ҳақида тўлиқ

маълумот узатилади –яъни чиқиш фильтри бу ерда мавжуд эмас. GRU ва LSTM ўртасидаги фарқ шундаки, GRU тезроқ ва содда ишлайди, бироқ унинг имкониятлари чекланган.

1.19.Тьюринг нейрон машинаси



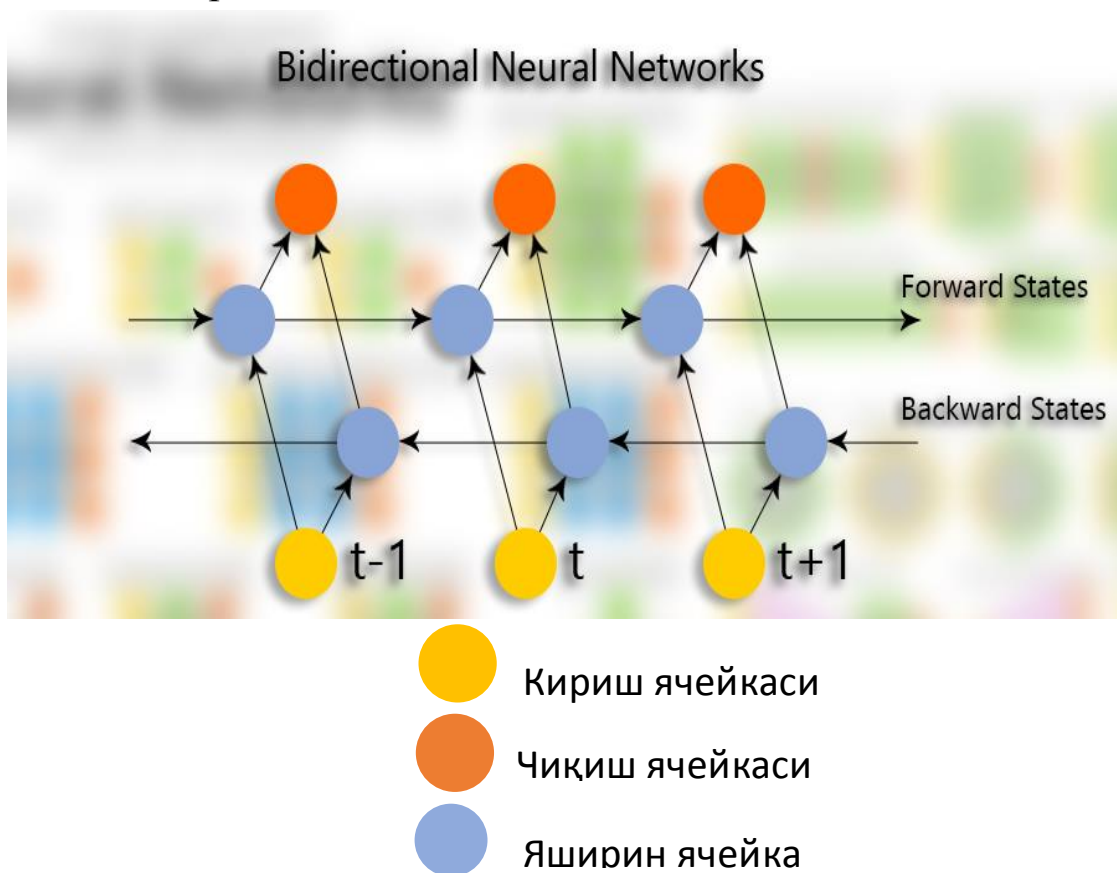
36-расм. Тьюринг нейрон.

Тьюринг нейрон машинаси (neural Turing machine, NTM) узок ва қисқа муддатли хотирага эга абстракт тармоқ сифатида қаралади, шу билан бирга нейрон тармоқ ичида кечадиган жараённи изоҳлайдиган тармоқ деб қабул қилинади. Хотира ячейкаси нейрондан алоҳида жойлашган. Яъни Тьюринг машинасидагидек ҳисоблаш ячейкаси хотира ячейкасидан ажратилган (Тьюринг машинаси), бироқ тизимда ҳисоблашлар белгиланган тартибда бажарилади. Демак, NTM да тармоғини аниқ дастурлаш орқали эмас, балки тесқари боғланишда хатоликни аниқлаш усулида ўрганиш мумкин. Бу ўз навбатида

маълумотларни сақлаш тизимини ва нейрон тармоғи имкониятларини бирлаштиришга олиб келади, айнан шу сабабли бу тармоқ Тьюринг машинаси деб номланади: яъни маълумотларни ўқиш, ёзиб олиш ва ўқиб олган маълумот турига қараб ўз ҳолатини ўзгартириши мумкин ва уни тўлиқ Тьюринг деб номланади

1.20. Икки йўналишли рекуррент нейротармоқ

Икки йўналишли рекуррент нейротармоқ, узоқ ва қисқа муддатли хотирага ва икки йўналишли бошқарувчи рекуррент блокга эга икки йўналишли тармоқ

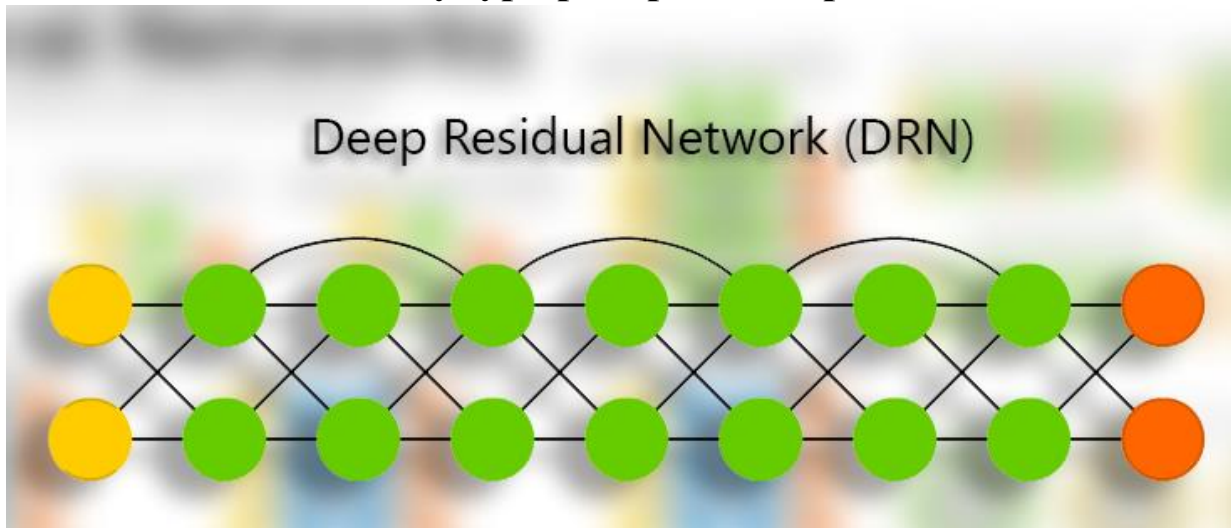


37-расм. Икки йўналишли рекуррент нейротармоқ (BRNN).

Икки йўналишли рекуррент нейротармоқ (BRNN), узоқ ва қисқа муддатли хотирага (BLSTM) ва икки йўналишли бошқарувчи рекуррент блокга (BGRU) эга икки йўналишли тармоқ схемада кўрсатилмаган, улар ўзининг бир йўналишли тармоқда нусхасини такрорлайди, бироқ улар қўллайдиган маълумотлар фақат олдинги қатламдан эмас балки кейинги қатламдан ҳам олинади. Масалан,

LSTM каби бир йўналишли тармоқ «мушук» сўзини аниқлашда ҳарфлар бир йўналишда бериб бажарилса, икки йўналишда кейинги ҳарфни ҳам бериб аниқлашади. Шу каби тармоқлар тасвирларни кенгайтириш ва бўшлиқларни тўлдириш ҳолатларида қўллаш учун мўлжалланган.

1.21. Чуқур орттирмали тармоқ

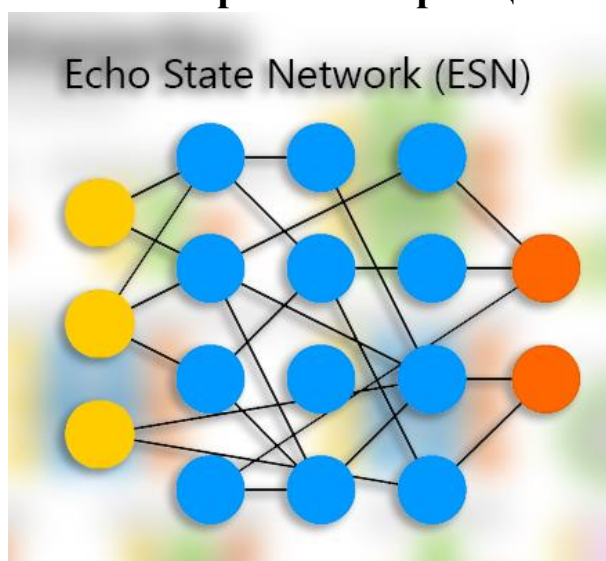





- Кириш ячейкаси
- Чиқиш ячейкаси
- Яширин ячейка

38-расм. Чуқур қолдиқли тармоқ.

Чуқур қолдиқли тармоқ (deep residual network, DRN) чизиқли йўналтирилган чуқур нейротармоқ бўлиб, алоҳида қатламлар орасида қўшимча боғланишлар мавжуд. Шу билан бирга чуқур қолдиқли тармоқ рекуррент тармоқларга жуда ўхшаб кетади, шу билан бирга узоқ ва қисқа муддатли тармоқ билан кўпинча таққосланади. DRN тармоқларни шаблонлар асосида ўрганиш мумкин, шаблон сатҳи 150 қатламгача бўлиши мумкин [11].

1.22. Нейрон эхо-тармоқ

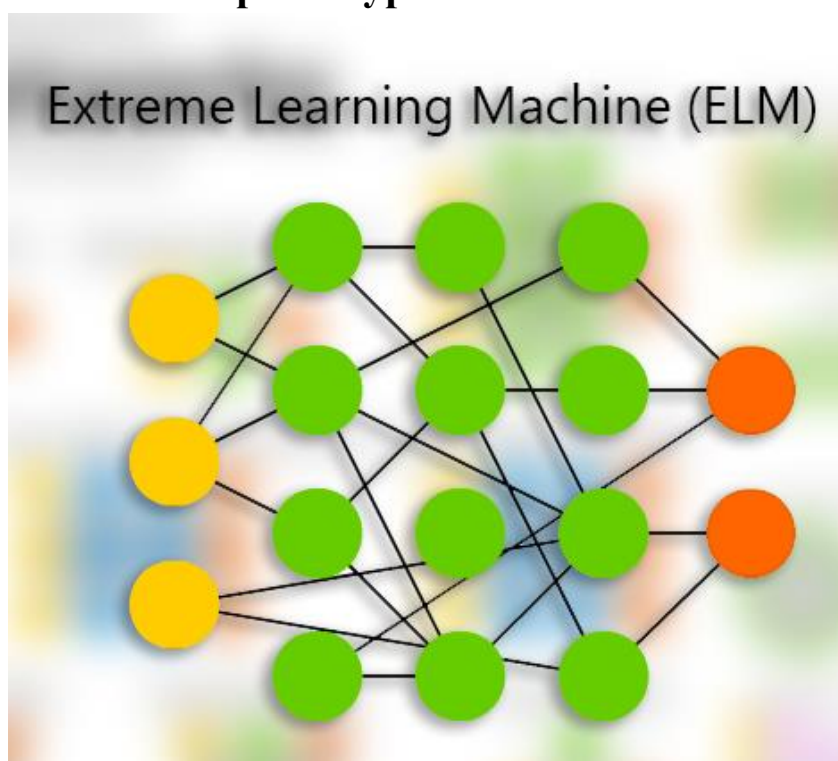





-  Кириш ячейкаси
-  Чиқиш ячейкаси
-  Рекуррент ячейка

39-расм. Нейрон эхо-тармоқ.

Нейрон эхо-тармоқ (echo state network, ESN) рекуррент нейрон тармоқнинг бир кўриниши ҳисобланиб, битта яширин қатламга эга. Ушбу қатлам резервуар сифатида хизмат қилиб, нейронлар ўртасидаги сийрак тасодифий боғланишларга эга. Резервуар ичидаги боғланиш аниқ фиксирланган, бироқ чиқиш қатлами билан боғланишни ўқитиш мумкин. Яширин қатлам ҳолати резервуарнинг олдинги ҳолатлари ҳамда кириш ва чиқиш сигналларнинг олдинги ҳолатлари бўйича ҳисобланади. ESN ўрганиш учун тескари боғланишда хатоликни аниқлаш усулини қўллаб бўлмайди, бироқ кириш қатламига маълумотларни узатган ҳолда янгилаб туриб, чиқувчи маълумотлар ҳолати ўзгаришини кузатиш лозим. Эхо-тармоқ вақт кўрсаткичи асосида самарали ишлайди.

1.23.Экстремал ўрганиш машинаси

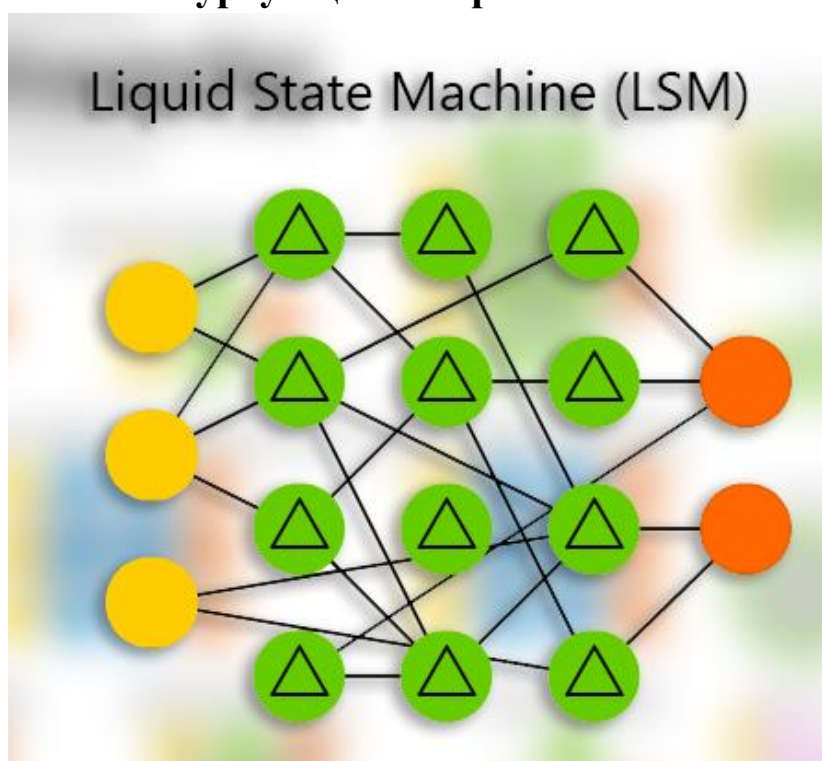





-  Кириш ячейкаси
-  Чиқиш ячейкаси
-  Яширин ячейка

40-расм. Экстремал ўрганиш машинаси.

Экстремал ўрганиш машинаси (**extreme learning machine, ELM**)—бу ўрганиш алгоритми бўлиб, кўриниши бўйича битта яширин қатламга эга чизиқли тақсимланган тармоқ ва унда нейронлар тасодифий боғланишга эга. Оғирлик кўрсаткичи яширин қатлам учун тасодифий белгиланса, очик қатлам учун мавҳум тескари матрица орқали олинади. Яширин ва очик қатламлар учун оғирлик кўрсаткичлар бир қадам мобайнида ўрганилади. ELM кўпинча турғун ҳолатлар усулига ва ESN га ўхшаб кетади, бироқ FFNN каби қўлланилади.

1.24. Турғун ҳолатлар машинаси



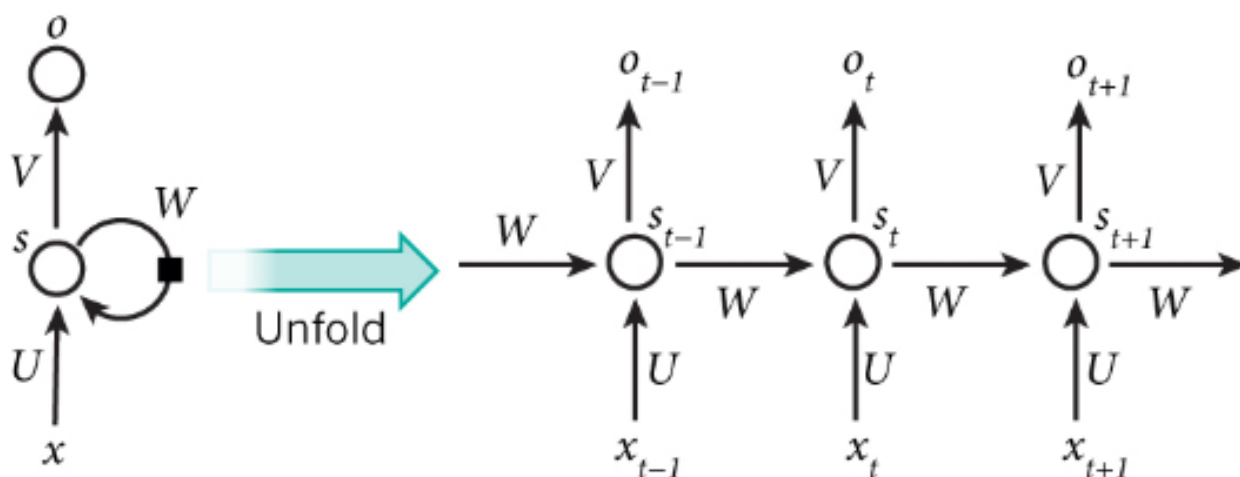
-  Кириш ячейкаси
-  Чиқиш ячейкаси
-  Боғланган яширин ячейка

41-расм. Турғун ҳолатлар машинаси.

Турғун ҳолатлар машинаси (**liquid state machine, LSM**) нейрон эхо-тармоқ билан ўхшашлиги мавжуд, уларнинг фарқи LSM да фаоллашувнинг сигмоидиал функция ўрнига чегарани белгилаб оладиган функция қўлланилади, бунда ҳар бир нейрон жамлаб борувчи хотира ячейкаси бўлиб хизмат қилади. Нейрон янгиланган қўшни нейронлар қийматлари йиғиндисига тенг бўлмайди, балки ўзининг қийматига қўшиб олади, белгиланган чегарага етганда бошқа нейронларга эълон қилинади.

1.25.RNN LSTM ёрдамида таржима жараёни

Нейрон машинали таржима (NMT neural machine translation) – бу машинали таржима тизими бўлиб, машинали таржима жараёнида аниқлик ва тезкорлик кўрсаткичини ошириш учун қўлланилади. NMT тармоқ иши оддий шифратор-дешифраторга асосланган. NMT да қўлланиладиган нейрон тармоқлар бу рекуррент нейрон тармоқлар (RNN) ҳисобланади. Айнан RNN нинг танланганлигига сабаб, циклик структуранинг мавжудлиги, янги такрорланаётган кетма-кетликни ўрганиш имконини беради. Бундан ташқари, икки тил асосидаги гапларни кетма-кетлик кўринишда сақлаш имконини беради. RNN структураси қуйидаги расмда келтирилган.



42-расм. RNN структураси.

Юқорида келтирилган кўриниш бўйича шуни англаш мумкин-ки, битта қатламни бир неча қатламларга бўлиниши мумкин, натижада олдинги босқичдаги маълумотларни битта ячейкада сақлаш имконини беради. бошланғич ва натижа маълумотлар ўртасидаги берилганлар ўртасидаги муносабат олдиндан маълум бўлса, RNN ёрдамида кетма-кетликлар харитасини ифодалаш мумкин.

X ва Y- мос равишда бошланғич ва натижавий тилларда берилган гаплар берилган бўлсин. RNN **кодлашга** бошланғич гап X ни бир нечта белгиланган ўлчамдаги векторларга ўзгартиради. Ушбу дешифратор битта сўзни битта чиқиш кўринишда эҳтимоллик асосида узатади.

$$P(Y|X) = P(Y|X_1, X_2, X_3, \dots X_m) \quad (6)$$

Бунда x_1, x_2, \dots, x_m бу шифратор ёрдамида кодланган аниқ векторлар ҳисобланади. Юқорида келтирилган қонуният асосида тенглама кўриниши қуйи кўринишга ўзгаради. Кейинги сўзни декодлашда қўлланиладиган символи гаплар векторлари орқали башорат қилинади. Юқорида келтирилган ифода қуйидаги кўринишда бўлади.

$$P(Y|X) = P(y_i | y_0 y_1, y_2, \dots, y_{i-1}; X_1, X_2, X_3, \dots X_m) \quad (7)$$

Таксимотдаги ҳар бир таркиб softmax (**Softmax** – бу кўп ўлчовли ҳолатлар учун логистик функция) луғатдаги ҳар бир сўз асосида функция кўринишда келтирилади.

Бироқ RNN қўллаш самарали бўлсада, узун гаплар билан ўрганиш ҳолатлар учун муаммолар пайдо бўлади, чунки “узоқ муддатли боғланиш муаммоси” мавжуд. Бунга сабаб, RNN да t вақт оралиғи учун натижа қуйидаги кўринишда аниқланади

$$\frac{\partial h_t}{\partial h_0} = \frac{\partial h_t}{\partial h_{t-1}} * \frac{\partial h_{t-1}}{\partial h_{t-2}} * \dots * \frac{\partial h_2}{\partial h_1} * \frac{\partial h_1}{\partial h_0} \quad (8)$$

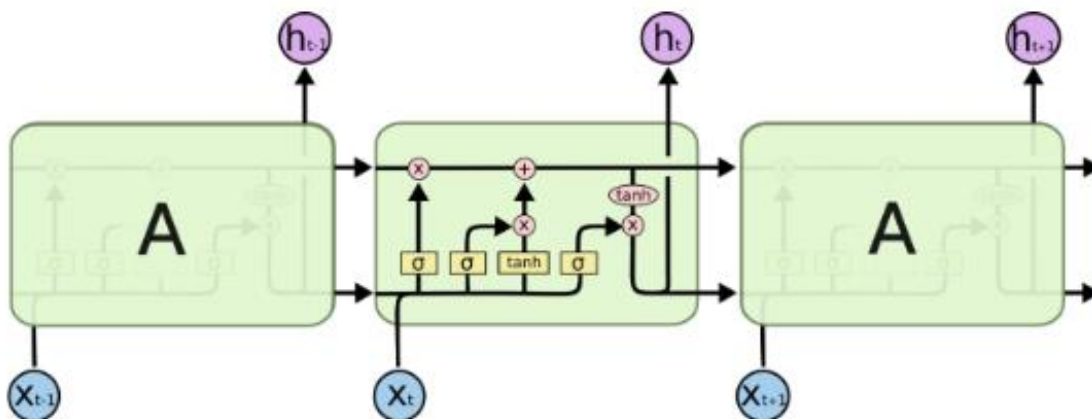
Демак, мультипликативлик эффекти сабаб, катта ҳажмдаги гапларни таржима қилишда самарадорлик ва аниқлик кўрсаткичлари камаяди. Қуйидаги масалада кўриб чиқамиз. Башорат этиш масаласида гапдаги кейинги сўзни аниқлаш учун ўгириш тили модели қўлланилса, катта ҳажмдаги гапларда маълум бир қисмларда бўшлиқлар пайдо бўлади. Яъни аниқланган контекстлар ва аниқланмаган контекстлар ўртасида қолиб кетишлар пайдо бўлади ва усул самарасиз бўлиб қолади. Бироқ, «Мен зўр футбол ўйинчисиман ва мен дарвозага коптокни тўғри тепаман» каби гапларда, таржимасида гапнинг иккинчи қисми айнан биринчи қисм асоси сифатида хизмат қилади. Гапларда бир нечта вариантлар мавжуд бўлиши мумкин (масалан, бошланғич кетма-кетлик, ўрта кетма-кетлик ва қуйи кетма-кетлик). Бу муаммо маълумотлар жойлашуви, яъни сўзлар кетма-кетлиги ва катта ҳажмдаги гаплар билан ишлаганда яққол кўриниб туради. Амалда ўрганиш жараёнида RNN қўллаш мушкул бўлади, чунки барча тилларда мураккаб контекстли боғланиш

мавжуд бўлиб, RNN кодлаш ва декодлашда қўллаш мақсадга мувофиқ эмас. RNN камчилигини олдини олиш мақсадида узоқ ва қисқа муддатли хотира (LSTM) моделини кодлаш ва декодлашда қўллаш лозим бўлади.

1.26. Узоқ ва қисқа муддатли хотира (LSTM) модели.

Узоқ муддатли ва қисқа муддатли хотира (LSTM) RNNнинг бир кўриниши бўлиб, машинали таржима тизимларида баъзан RNN LSTM билан ўзгартирилади. LSTM ҳам занжир структурасига эга бўлиб, RNN такрорлаш модулидан фарқланади.

Нейрон тармоқнинг битта қатлами ўрнига 4 қатламли модуль мавжуд. Ўрганиш жараёнида бу қатламлар ҳам шу модуллар оралиғида ҳамда, бошқа модуллар оралиғида ўзаро боғланишади. LSTM модулининг типик структураси расмда келтирилган.



43-расм. LSTM модулининг типик структураси.

LSTMда такрорланадиган модули тўртта ўзаро боғланадиган қатламга эга. Бу модулда ўрганиш жараёни 4 та амали учун 4 кириш қисмига эга. Биринчи қисм, агар "қайта тиклаш - калит" (f_t , Бу ўрганиш жараёнида қатламнинг қайси қисмидан бошлаб олдинги маълумотларни унутиш кераклигини белгилайди). Кейинги сигмоид қатлами, бу кириш-калит (i_t) қатлами бўлиб, тизимнинг янгиланадиган қийматларини белгилайди. Учинчи калит - \tanh функцияли қатлам бўлиб, янги вектор қиймат-номзодларини, \tilde{C}_t яратади, уни модулда ҳолат кўринишда қўшади. Сўнгида, чиқиш қисми тўртинчи қатламни белгилайди. Бу ҳам \tanh функцияси бўлиб, кейинги модуллар

ҳолатларини яратади. Ушбу барча функциялар ифодалари қуйидаги кўринишда келтирилади:

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c * [h_{t-1}, x_t] + b_c)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o)$$

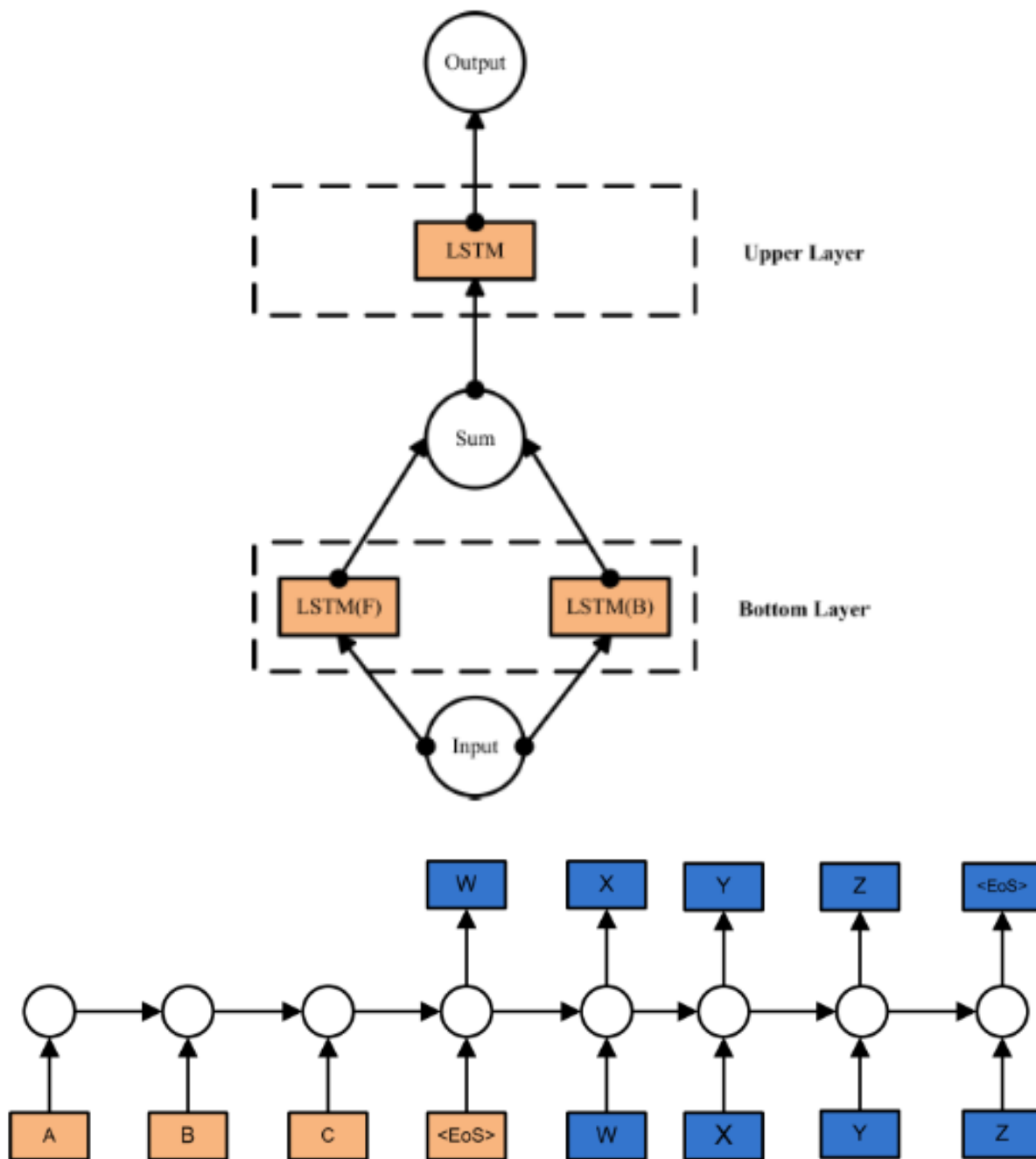
$$h_t = o_t * \tanh(C_t) \quad (9)$$

Расмда кириш қисмидан АБВ қабул қилади ва WXYZ ни натижа гап кўринишда чиқаради. Гап охирига етгандан (EOS-end of sentences) сўнг модель башорат қилишни тўхтатилади - чиқариш сигнали узатилади.

1.27.Нейрон машинали таржима тизимни созлаш

А. Шифратор ва дешифратор Нейрон машинали таржимада LSTM да икки йўналишли кодлаш қўлланилади. Бу кодировшик маълум бир вақт ичида чиқиш қисмидаги маълумот фақат олдинги маълумот билан бир қаторда кейинги маълумотга ҳам боғлиқлик концепциясига асосланган. Бу ғояни қўллаб, битта чиқиш қисмига, икки томонга йўналтирилган яширин қатламларни қўллашга LSTM йўналтирилади. Бу ўзгаришга эга LSTM версияси икки йўналишли LSTM (Bi-LSTM) деб номланади. Bi-LSTM жорий тармоқ кириш қисми маълумотларини ошириш учун қўлланилган. LSTM дан фарқли равишда, Bi-LSTM да жорий ҳолатдан бошлаб кейинги кирувчи сигналларга мурожаат этиш имкони мавжуд. Расмда икки йўналишли шифратор архитектураси келтирилган. Расмда кўрсатилган кодировшик бир қатламли шифратор ҳисобланади. Google нейрон машинали таржимонда кодировшик 8 қатламли бўлиб, дешифратор

вақти маълумотни қайта ишлаш учун қўлланилади. Ўрганиш жараёни самарали бўлиши учун бир нечта LSTM қатламларни ҳам шифратор да ҳам дешифраторда қўллаш тавсия этилади.

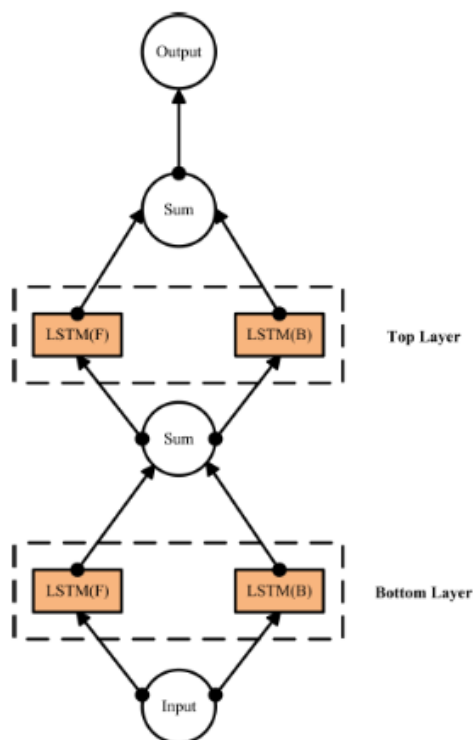


44-расм. LSTM тармоғида гапларни моделлаштириш

Bi-LSTMли шифраторнинг икки йўналишли конструкцияси

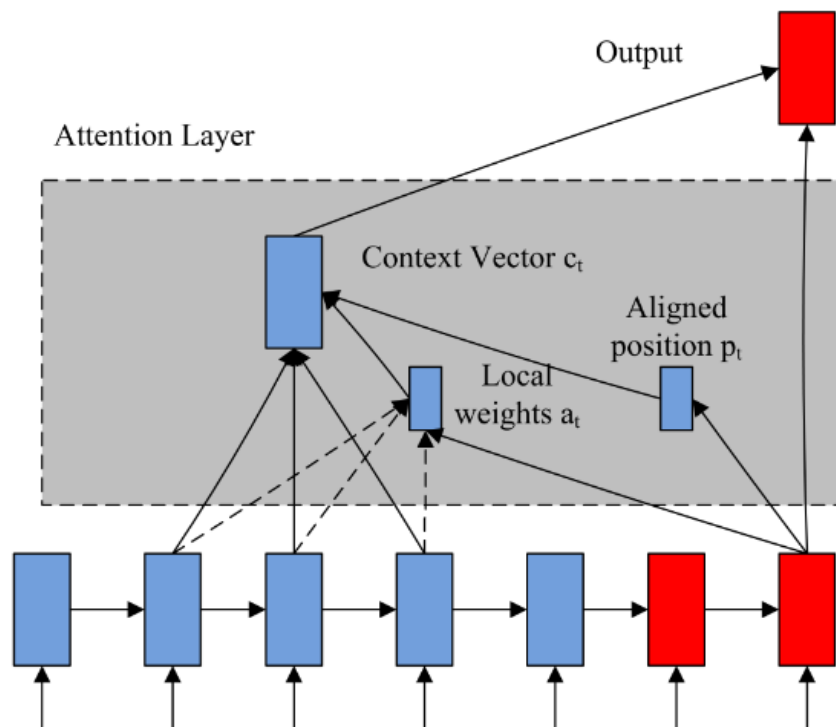
Дешифраторнинг вазифаси сўзлар векторини натижавий тилга декодлаш ҳисобланади.

Икки қатламли дешифраторнинг кўриниши расмда келтирилган.



45-расм. Дешифратор қатламнинг архитектураси

В. Attention (эътиборга олиш) қатлами. Моделда attention қатлами бу шифратор ва дешифратор ўртасидаги кўприк бўлиб, attention моделининг икки кўриниш мавжуд; глобал ва локал. Глобал attention модели ғояси бу контекст c_t векторларини чиқаришда кодировшик барча яширин қатламлари ҳолатларини инобатга олиши лозим. Глобал attention модели, a_t , қайсики, у берилган ўлчамдаги ўзгарувчиларни тўғрилаш вектори бўлиб, у маълум бир узунликда бўлади ва бошланғич қийматлардан бошлаб вақт қадамларига тенг. У жорий яширин h_s ҳолатлар билан бошланғич яширин h_t ҳолатларни таққослаш орқали ҳосил қилинади. Ўгиришда тилнинг моделини яратиш концепцияси локал attention моделдан фарқланади. Attention локал моделда, модель энг аввал битта тўғриланган катори кетма-кетлик r_k натижавий тил учун башорат этади. Марказ деб ҳисобланган жорий бошланғич r_t позиция ёрдамида, атроф позициядаги контекст векторлари c_t аниқланади. Бу тизимда локал attention модел қўлланилди. Attention қатлами ишини кўрсатувчи чизма расмда келтирилган.

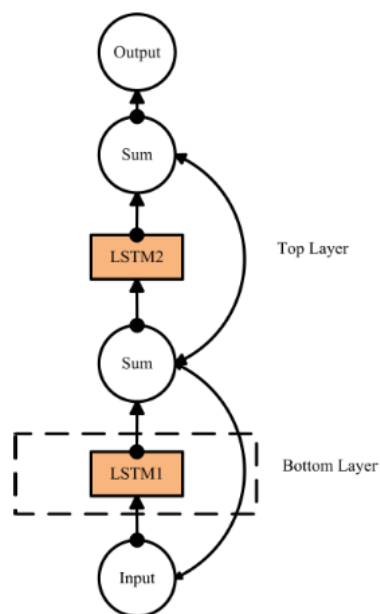


46-расм. Локал attention модели.

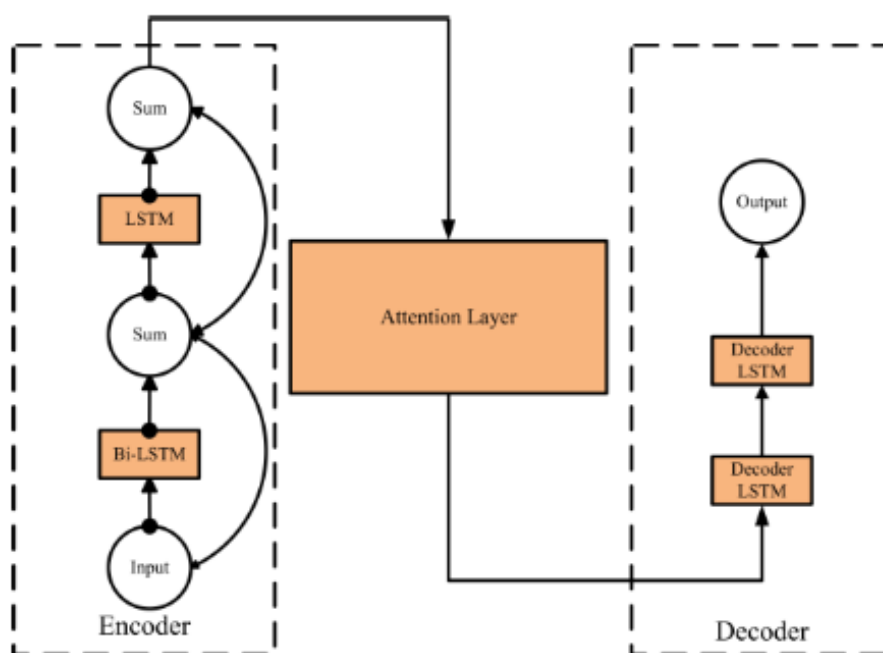
С. Қолдиқ боғланишлар ва кўприк

Нейрон тармоқ самарадорлиги унинг чуқурлик кўрсаткичига боғлиқ. Бироқ, градиент йуқолиши ва ёйилиши ҳисобидан тармоқ ҳажмининг ошиши, ўрганиш жараёни мушкул бўлишига олиб келади. Бу муаммо ечилган бўлиб, кириш қисми, оралиқ қисм, ва натижавий қисмларни алоҳида моделлаштириш ғояси қўлланилган. Бу қолдиқ боғланишлар деб номланади. Қолдиқ боғланишларда қатламнинг кириш қисмини кейинги қатламга узатишдан олдин чиқиш қисмига қўшилади. Расмда LSTM1 чиқиш қисми кириш қисмига қўшилиб LSTM2 кириш қисм сифатида узатилади. Маълумки, қолдиқ боғланишлар қайтиб тескари ўтиш жараёнида градиентли оқимни яхшилади бу эса ўз навбатида чуқур тармоқларни ўрганишини таъминлайди.

Қўшимча қатлам шифратор ва дешифратор қатламлари оралиғига керак бўлиб, расмда шифратор, дешифратор, қолдиқ боғланишлар ва кўприкдан иборат тизим келтирилган. Расмда гап векторларга ажратилиши ва натижавий тил attention векторлари кўринишига келтирилиши график ифодаланган.



Encoder

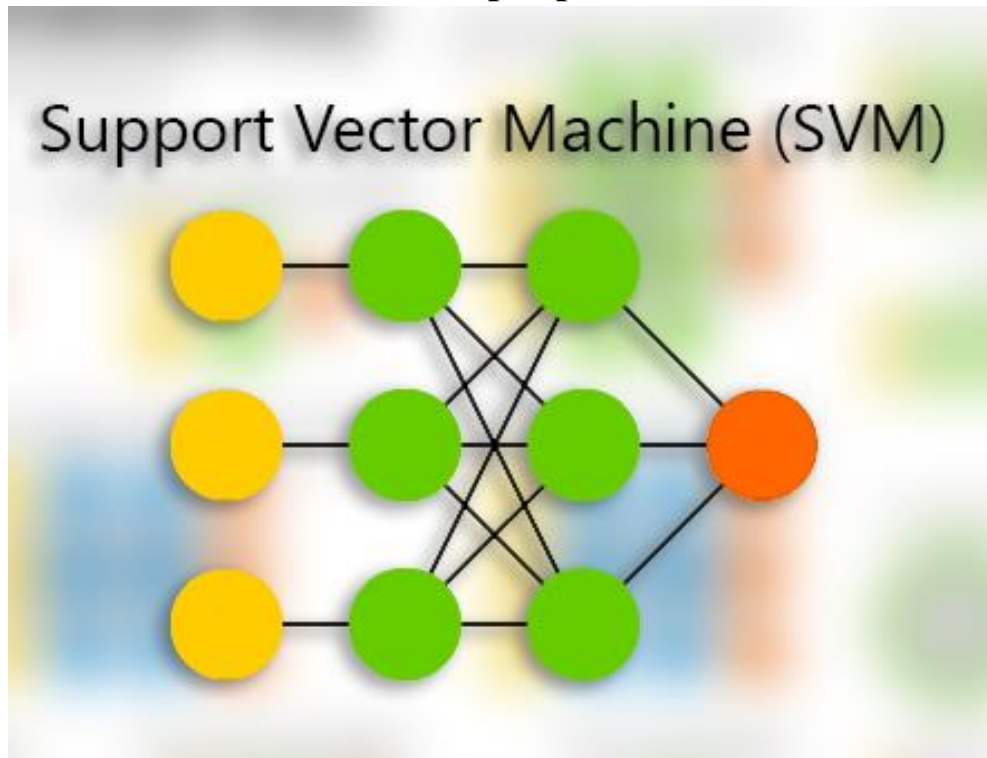


47- расм. Гап векторларга ажратилиши ва натижавий тил attention векторлари кўринишига келтирилиши

Статистик иборалар асосидаги машинали таржима тизимида аниқлик кўрсаткичи ва катта ҳажмдаги маълумотларни қайта ишлаш учун узоқ вақт талаб этилиши муаммолари мавжуд. Юқорида RNN ва LSTM қўллаб ушбу муаммоларни ечиш усулларини кўриб чиқдик.

Натижада нейрон машинали таржима тизимлари катта ҳажмдаги қатламга эга шифратор ва дешифраторлар ёрдамида юқори самарадорлик кўрсаткичини кўрсатиши мумкин.

1.28.Таянч векторлар машинаси



Кириш ячейкаси



Чиқиш ячейкаси



Яширин ячейка

48-расм. Таянч векторлар машинаси.

Таянч векторлар машинаси (support vector machine, SVM) ўқитувчи билан ўрганиш асосида ишловчи алгоритмлари таснифи оиласи ҳисобланади. SVM оптималлаштириш билан боғлиқ масалаларнинг оптимал ечимини топади. Таянч векторлар машинасининг классик версияси чизиқли тақсимланадиган маълумотларни категория бўйича ажратиш имкони мавжуд. Масалан, Леопольд ва Матроскин образларида мушуклар ўртасидаги фарқни аниқлаб бериши мумкин. Ўрганиш жараёнида тармоқ маълумотларни икки ўлчовли муҳитга ўтказди ва уларни максимал аниқликда тўғри

чизик билан ажратиб синфларга бўлинади бунда чизикнинг ҳар бир томонига бир синфга мансуб маълумотлар жойлашиши ва шу икки томонда жойлашган энг яқин нуқталар ўртасида масофа максимал бўлишига эришилади. Ушбу масофа бўшлиқ деб, нуқталар эса таянч векторлар деб номланади. Тўғри чизикни белгилашда бўшлиқ максималлаштирилади бу эса синфларга оптимал бўлишни таъминлайди. Шу билан бирга SVM n -ўлчовли маълумотларни ажратиш имконини беради. Шунини таъкидлаш лозимки, таянч векторлар машинасини ҳар доим ҳам нейрон тармоқ сифатида қаралмайди.

1.29. Таянч векторлар машинаси асосида маълумотларни таснифлаш

(Support Vector Machine, SVM). Бундай тасниф кенг кўламда қўлланилади. Таснифнинг энг бошланғич масаласи камида иккита синфдан биттасига мансублигини аниқлаш ҳисобланади. Одатда бу объект R бўшлиқда n ўлчовли вектор ҳисобланади. Вектор координаталари объектнинг алоҳида атрибутларини изоҳлайди. Масалан, берилган RGB моделида, ***c ранги***, уч ўлчовли муҳитда вектор ҳисобланади: $c=(red, green, blue)$.

Агар синфлар фақат иккита бўлса бинар тасниф деб айтилади. Агар синфлар бир нечта бўлса— кўпсинфлилик (мультисинфли) тасниф деб номланади. Шу билан бирга ҳар бир синф образи – объектлар мавжуд бўлиши мумкин бўлиб, уларнинг қайси синфга мансублиги олдиндан маълум бўлади. Бундай кўринишдаги масалаларни ўқитувчи ёрдамида ўрганиш деб, олдиндан аниқ бўлган маълумотлар эса ўрганиладиган тўплам деб юритилади. (Эслатма: агар синфлар аввалидан берилмаган бўлса, кластерлаш масаласи ҳисобланади.)

Демак, ўқитувчи ёрдамида ўрганишга мисол кўрамиз, тасниф масаласининг математик изоҳи қўйидагича: масалан X –объектлар фазоси бўлсин (масалан, \mathbb{R}^n), Y – эса синфлар (масалан, $Y = \{-1,1\}$). Ўрганиладиган тўплам эса: $(x_1, y_1), \dots, (x_m, y_m)$. $F : X \rightarrow Y$ (классификатор) функциясини тузиш масаласи қўйилган бўлиб, x тасодифий объектни y синф билан таққослаш керак бўлсин.

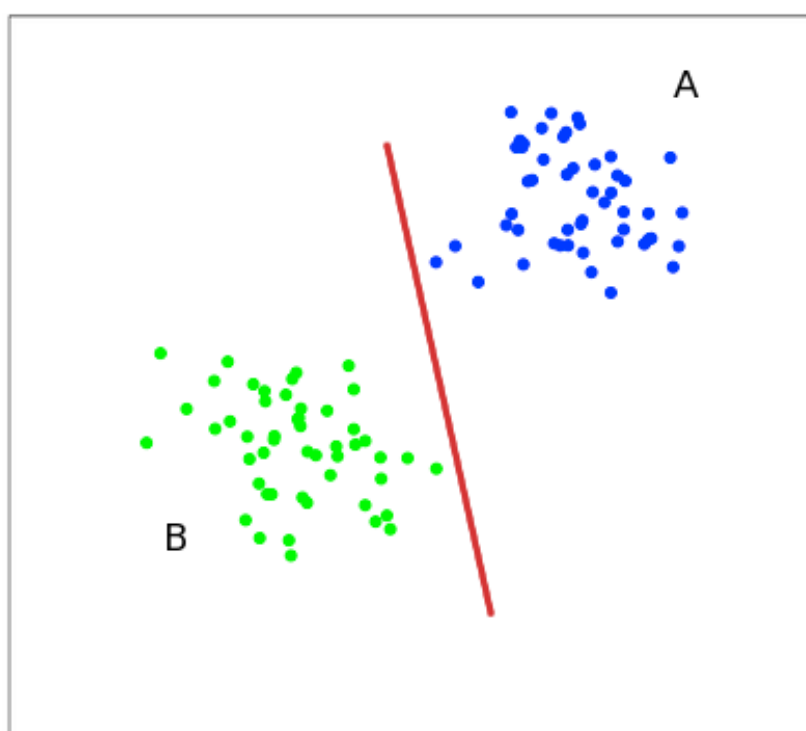
1.29.1.Таянч векторлар машинаси

Мультиклассификация масаласи сифатида ишлаши мумкин бўлсада, ушбу усул бинар классификатор усулига мансуб.

Усул ғоясини қўйидаги мисол сифатида кўрсатиш мумкин: текисликда нуқталар берилган бўлиб, улар икки синфга бўлинган, расмда ушбу икки синфни ажратадиган чизик ўтказилган (қизил чизик). Кейинчалик барча янги нуқталар автоматик равишда қўйидаги усулда таснифланади:

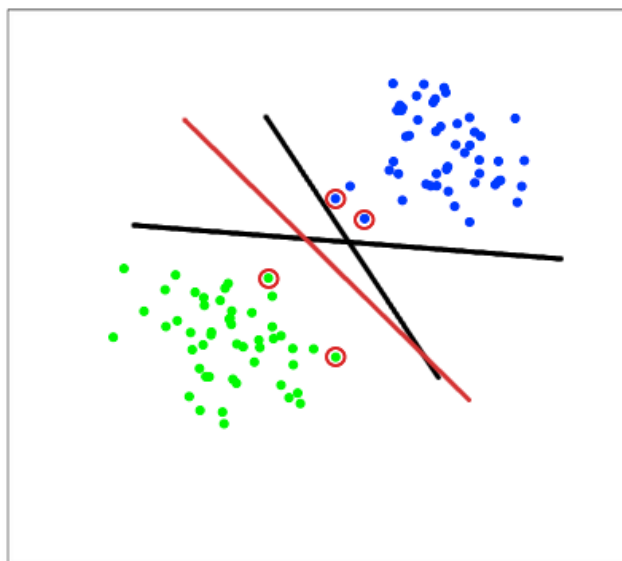
чизикдан юқори нуқта А синфга мансуб,

чизикдан паст нуқта В синфга мансуб.

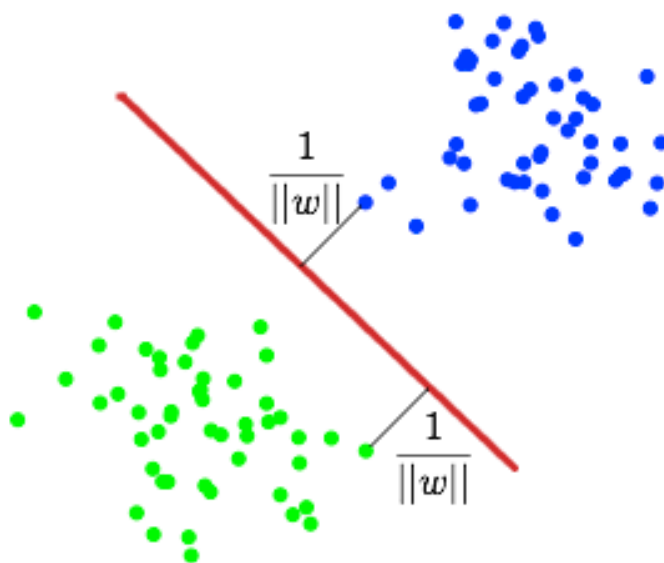


49-расм. Таянч векторлар усулига мисол.

Ушбу тўғри чизик ажратувчи чизик деб номланади. Бироқ, катта ҳажмдаги текисликда тўғри чизик асосда синфларга ажратиш бўлмайди, чунки «чизикдан паст» тушунчаси ёки «чизикдан юқори» тушунчалари ўз моҳиятини йўқотади. Шунинг учун тўғри ўрнига фазовий - гипертекислик қўлланилиб, унинг ўлчов кўрсаткичи бошланғич фазога нисбатан бир кам ўлчамда бўлади. Масалан, R^3 да гипертекислик сифатида оддий икки ўлчовли текислик ҳисобланади. Кўриб ўтилаётган мисолда синфга ажратиш учун бир неча чизиклар мавжуд:



50-расм. Таснифлаш учун синф нуқталарининг жойлашуви

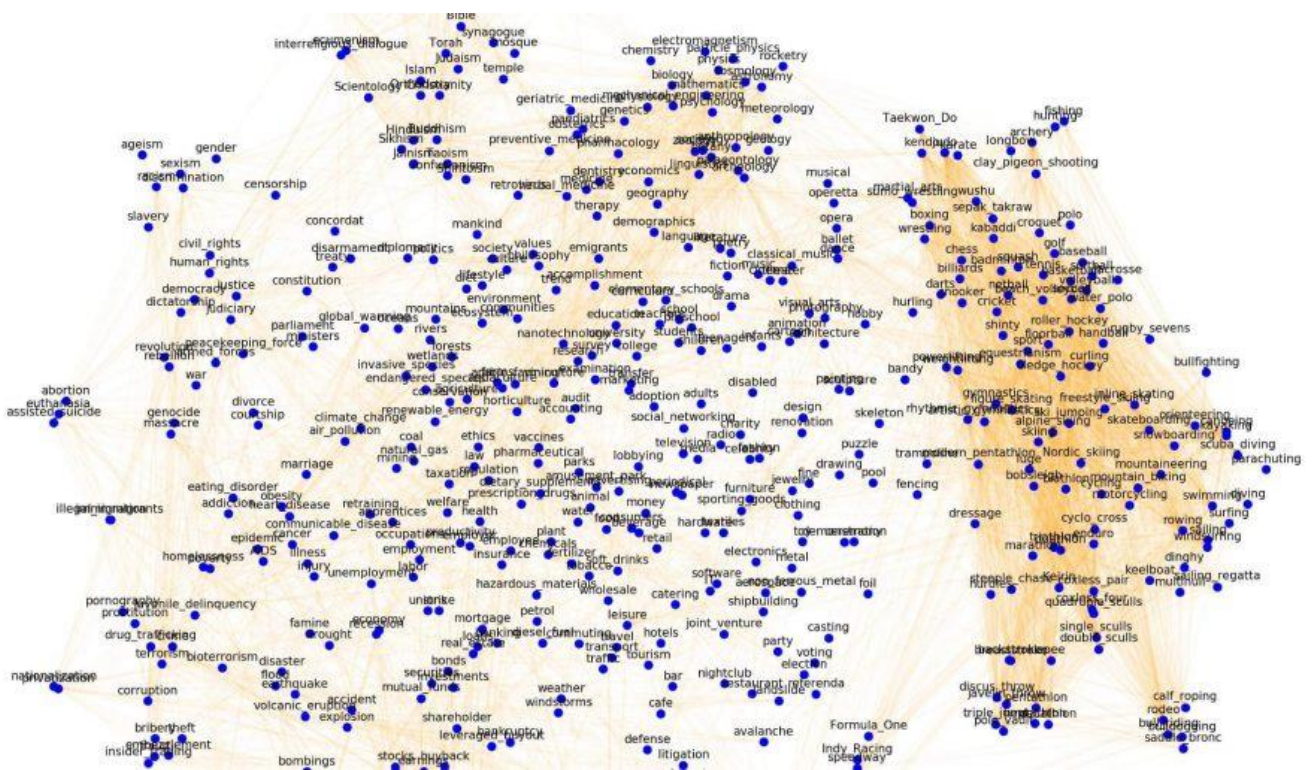


51-расм. Векторлар жойлашуви.

Тасниф нуқтаи назаридан қаралганда тўғри чизик шундай танланиши керакки чизикдан ҳар бир синф нуқталаригача бўлган масофа максимал қийматга эга бўлиши лозим. Бошқача қилиб айтганда, синфларга аниқ ажратувчи чизикни белгилаш лозим. Бундай чизик умуман олганда – оптимал ажратувчи гипертекислик деб айтилади.

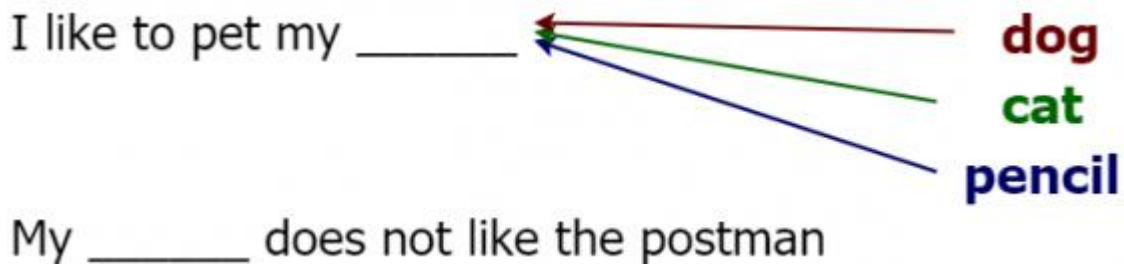
Ажратувчи гипертекисликка яқин жойлашган векторлар эса таянч векторлар деб номланади.

1.30. Word2Vec



52-расм. Сўзларнинг векторли кўриниши: барча ечим контекстда

Сўзлар вектори (word vectors) – сўзларнинг сонли ифодаси бўлиб, улар ўртасида мантик боғланишни сақлайди. Масалан, cat (мушук) вектори учун энг яқин сўз dog (ит). Бироқ pencil (қалам) сўзининг векторли кўриниши cat векторидан анча фарқланади. Бу асосан иккита сўзнинг контекстда биргаликда учрашиш эҳтимоллиги билан аниқланади (яъни, [cat, dog] ёки [cat, pencil]). Қуйидаги гап таркибини кўриб ўтамыз:



53-расм. Word2Sec га мисол.

Бунда қайси сўзлар тўғри келиши аниқ (pencil тўғри келмайди). Айнан нима учун тўғри келмаслигини аниқлаш лозим бўлади. Грамматика, талаффуз тўғри келса (тил миқёсида), нима учун тўғри келмайди? Барчаси айнан контекстга бориб тақалади, pencil маъно жиҳатдан тўғри келмайди. Бу мисол бўйича контекст аҳамиятли эканлигини билдиради. Word2vec алгоритми контекстни қўллайди ва сўзларнинг сонли ифодасини шакллантиради. Шунинг учун битта контекстда биргаликда келган сўзлар ўхшаш векторларга эга бўлади.

1.30.1. Word2vec ни қўллаш

Реал лойиҳаларда Word2vec қўлланилишини тушуниш учун google scholar тизимига кириб, NLP билан боғлиқ маълумотларни излаймиз (масалан, савол-жавоб тизимлари, чат-ботлар, машинали таржима ва ҳ.к.). 2013 йилдан кейинги йиллар, яъни word2vec пайдо бўлган муддатдан бошлаб қўшилган хужжатлар бўйича филтрлаймиз ва сўзларнинг векторли ифодаланиши бўйича жуда кўп мақолаларни кўриш мумкин.

Сўзларнинг векторли ифодаланиши кўпгина соҳаларда қўлланилади:

Маълум бир тилни моделлаштиришда;

Чат-ботлар;

Машинали таржима;

Савол-жавоб тизимлари;

...ва бошқалар.

Барча замонавий NLP иловалари асосан word2vec алгоритмларига асосланган. Сўзлар ифодаланишининг мавжуд моделларини такомиллаштиришни кўриб ўтамиз. Улар ёрдамида семантик ўхшаш сўзларни бир-бирига яқин векторларда ифодалаш имконини бериб, маъно жиҳатдан узок бўлган сўзларни бошқача ифодалайди. Бу модел хусусияти самарали натижага олиб келади.

Сўзлар векторини яратиш кетма-кетлиги

Бошланғич кўрсаткичга эга бўлмаган сўзлар кетма-кетлигини ўрганиш учун бир неча масалаларни ечиш лозим:

Берилганларни бўлақларга ажратилган шаклларни ҳосил қилиш [кириш сўзи, чиқиш сўзи], бунда ҳар бир сўз n узунликдаги иккиламчи вектор кўринишга эга бўлади, бунда i - қиймат i - ўринда 1 билан белгиланса, қолган ҳоллар учун 0 билан белгиланади (one-hot кодлаш);

Кириш ва чиқиш қисмларига one-hot векторларни қабул қилувчи моделни яратиш;

Тўғри сўзни башорат қилувчи йўқотишлар функциясини аниқлаш, ва моделни оптималлаштириш;

Ўхшаш сўзлар ўхшаш векторларга эга эканлигига амин бўлиб, модель сифатини аниқлаш.

Бошланғич матндан тартибланган берилганларни яратиш

Қуйидаги мисолни кўриб ўтамыз

The cat pushed the glass off the table. (мушук стаканни столдан туширди)

Керакли берилганлар қуйидагича шакллантирилади расмдаги ҳар бир кавс бирламчи контекст ойнаси. Кўк рангли майдон кирувчи one-hot вектор (мақсадли сўз) ни ифодалайди, қизил майдон-чиқувчи one-hot векторни (мақсадли сўздан ташқари контекст ойнасидаги барча сўзлар яъни контекстли сўз). Битта контекст ойнасидан иккита берилганлар элементи ҳосил қилинади (яъни битта мақсадли сўзга иккита кўшни сўз мо келади). Ойна ўлчами одатда фойдаланувчи томонидан аниқланади. Контекст ойнаси қанчалик катта бўлса, модель иши шунча самаралидир, бироқ алгоритм бажарилиш вақтига таъсир қилади. Мақсадли сўз билан мақсадли берилганларни адаштирмаслик лозим, улар умуман бошқа бошқа.

Бизнинг нейротармоқ юқорида келтирилган кирувчи берилганлар асосида ўрганиш жараёнини ўтади. Бизга қуйидагилар лозим:

Кирувчи one-hot векторлар тўплами;

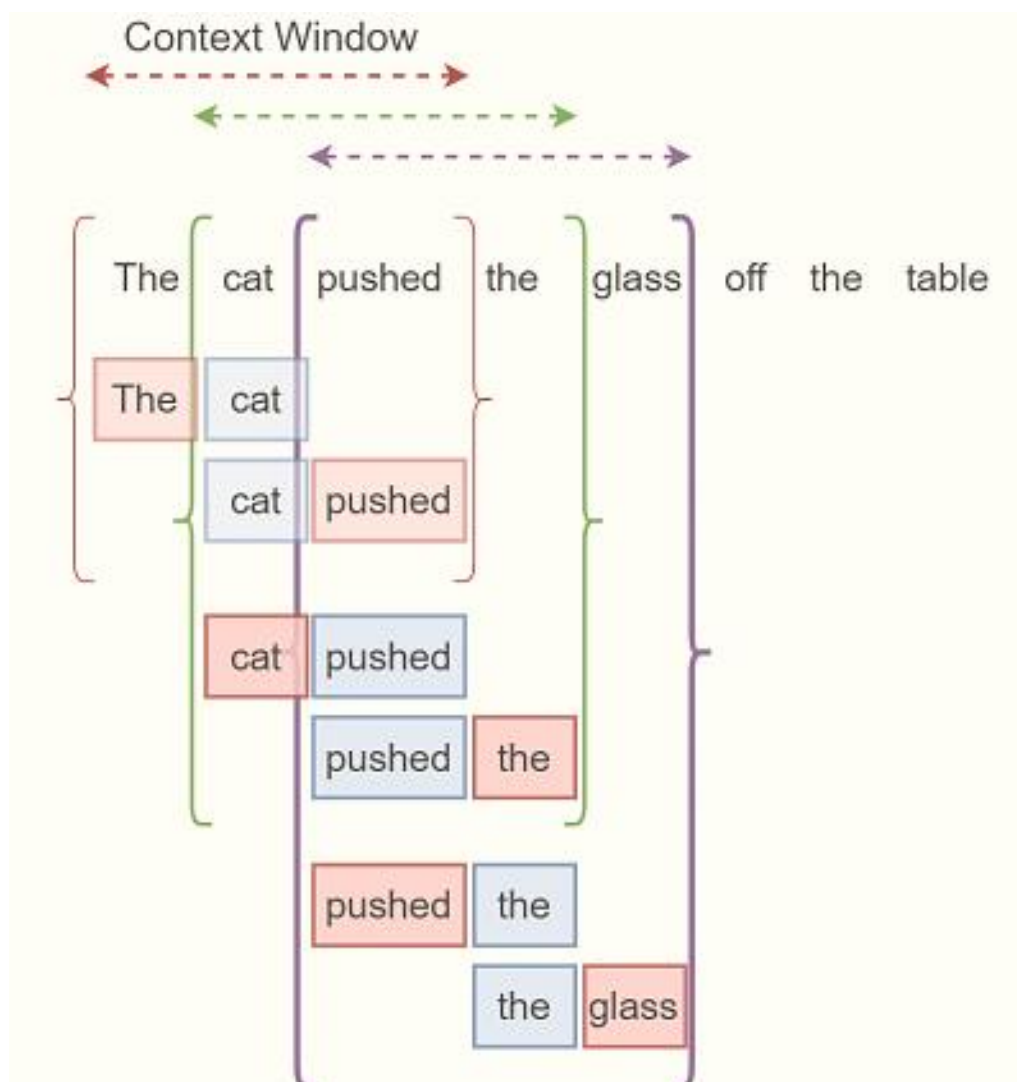
Чиқувчи one-hot векторлар тўплами (ўрганиш жараёнидан сўнг);
embedding layer;

нейротармоқ.

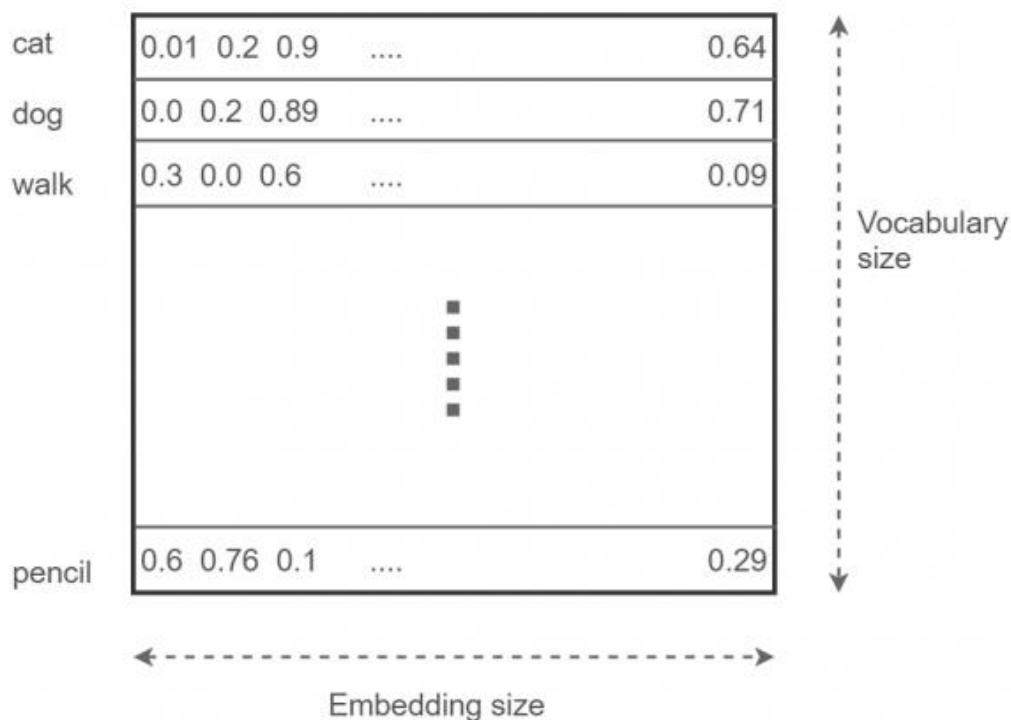
Сўнгги иккита қисмни батафсил кўриб ўтамыз.

Embedding layer

embedding layer дан бошлаймиз. Луғатда мавжуд барча сўзлар векторларини сақлайди фараз қилинг катта ҳажмдаги матрица [луғатдаги сўзлар сони x сўзларнинг ихчамлашган векторли ифодасининг ўлчамини ифодалайди]). Ушбу ўлчам (embedding size) соланадиган параметр ҳисобланади. У қанчалик катта бўлса, модель шунчалик самарали бўлади (бирок embedding size ҳажми маълум бир чегарага етганда унумдорлик ошиши тўхтайдди). Ушбу йирик матрица (нейротармоқ каби) тасодифий равишда инициаллаштирилади ва оптималлаштириш жараёнида битлар бўйича созланиб борилади. Қуйидаги кўринишга эга



54-расм. embedding layer и нейротармоқни аниқлаш



55-расм. embedding layer да яратилган сўзлар матрицаси

Нейрон тармоқ

Моделнинг сўнгги элементи бу- нейрон тармоқ. Ўрганиш жараёнида нейротармоқ кириш векторини қабул қилиб, барча сўзлар тўплами бўйича контекстга мос тушадиган сўзлар кетма-кетлигини башорат қилишга уринади (шу билан бирга, буни қуйидагича ҳам ифодалаш мумкин, яъни сўзлар one-hot кодланишининг чизиқли комбинацияси деб ҳам тушуниш мумкин). Сўнгга йўқотиш функцияси ёрдамида нотўғри тасниф учун моделни “жазолаш” тўғриси учун “тақдирлаш” мумкин. Ҳозирги мисолда битта кириш ва битта чиқиш берилганларни бир мартага қайта ишлаш билан чекланиб қоламиз. Ҳақиқий лойиҳаларда берилганлар батч (яъни, масалан 64 та элементдан иборат гуруҳлар) кўринишда қайта ишланади. Ўрганиш жараёнини умумий кўринишда изоҳлаймиз.

Кирилган сўз (мақсадли сўз) га мос келувчи векторни embedding layer дан аниқлаймиз;

Ушбу векторни нейротармоқга “истеъмолга” берамиз, сўнгга тўғри келувчи (контекстга мос) сўзни таклиф этишга ҳаракат қиламиз;

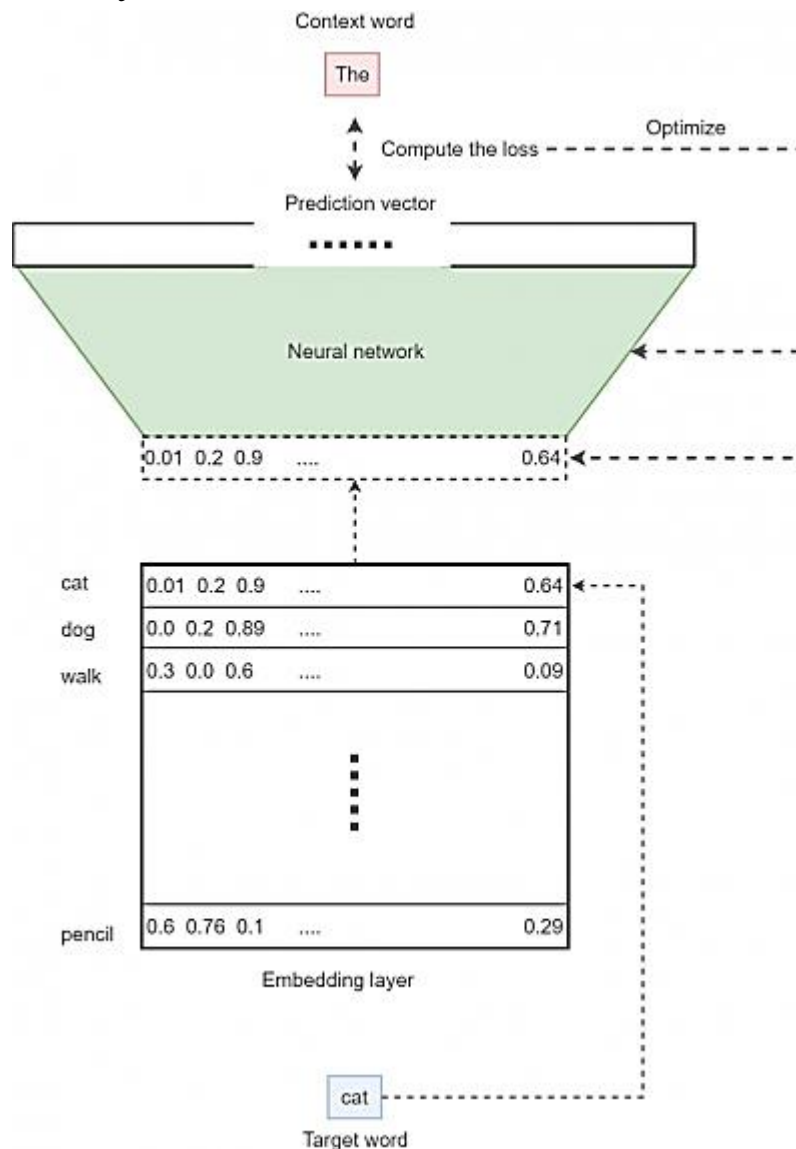
Таклиф қилинган сўз ва контекст ойнасида мавжуд сўзни таққослаб, йўқотиш функциясини аниқлаймиз;

Йўқотиш функциясини градиентли стохастик пасайиш билан бирга қўллаб, нейротармоқни ва embedding layer ни оптималлаштирамиз.

Башорат қилишда softmax функциясини қўллаб, эҳтимолликлар тақсимоти бўйича башоратни нормаллаштиришга олиб келинади.

Барчасини бирлаштирамиз

Word2vec алгоритмнинг барча ташкил этувчиларни била туриб, уларни бирлаштириш имкони пайдо бўлади. Модель ўрганиш жараёнидан ўтгандан сўнг embedding layer ни дискка сақлаш ва семантикали векторлардан тўлақонли фойдаланиш мумкин бўлади. Умумий кўриниши қуйидагича:



56-расм. Skip-gram алгоритми

Бу модель skip-gram алгоритми деб номланиб, word2vec нинг битта алгоритми ҳисобланади ва айнан унга эътибор ажратамиз. Бошқа бир алгоритм “сўзларнинг чексиз тўплами ”(continuous bag-of-words model, CBOW).

Йўқотишлар функцияси: моделини оптималлаштириш

Калит қисмдан бири бу– йўқотишлар функцияси бўлиб, стандарт кесишмали энтропия функцияси (softmax cross entropy loss) тасниф масалаларининг самарали ечимидир. Бироқ word2vec модели учун бу функция етарлича қулай эмас, бу асосан содда таҳлил масалаларида, масалан икки кўринишда чиқиш ҳолатлари: мусбат ва манфий бўлган масалаларида самарали ҳисобланади. Сўзлар билан ишлаш масалаларида қайсики миллиардгача сўзлар билан ишлашга тўғри келса, луғат матрица ҳажми 100000 гача ёки ундан ҳам ошиб бориши мумкин, натижада softmax-нормаллаштириш мураккаблашади. Бунга сабаб, softmax тўлиқ ҳисоблаши учун барча чиқиш тугунлари бўйича йўқотиш функциясини ҳисоблашига тўғри келади.

Шунинг учун қўллашга ихчам бўлган алтернатив, яъни sampled softmax loss функциясини қўллаймиз. Стандарт кесишмали энтропиядан анча фарқланади.

Энг аввал кесишмали энтропия функциясини ҳисоблаймиз, яъни мақсадли сўзнинг мавжуд контекстли сўзи билан башорат қилинган контекстли сўз ва ҳақиқий сўзга мослиги билан аниқланади. Сўнгра кросс-энтропия йўқотиш k кўрсаткичини, яъни мос келмайдиган намуналар (мақсадли сўз + контекст ойнадан ташқари бўлган сўз) ни аниқлаш лозим бўлади, бунда маълумотлардаги шовқинлик тақсимоти қўлланилади.

Йўқотиш функцияси қуйидагича аниқланади:

$Loss = SigmoidCrossEntropy(Prediction, Correct Word)$

$$+ \sum_{1}^K E_{Noise ID} SigmoidCrossEntropy(Prediction, Noise ID)$$

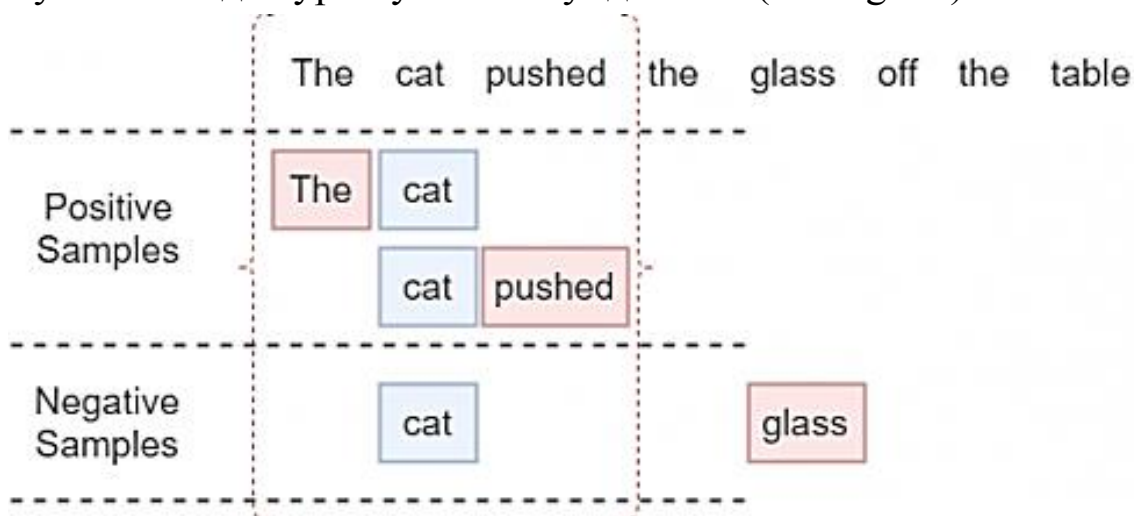
SigmoidCrossEntropy бу хатолик кўрсаткичи бўлиб, бошқаларга боғлиқсиз ҳолда битта чиқиш тугуни бўйича аниқлаш мумкин. Бу

ечим луғат ҳажми ошган ҳоллар учун жуда самарали ҳисобланади. Бу функция қандай амалга оширилишини билиш шарт эмас, чунки TensorFlow да бу функция созланган, бироқ k қандай параметр эканлигини англаш лозим. Энг муҳими –sampled softmax loss икки хил объект билан ишлаганда хатолик сифатида нимани аниқлашини англаш ҳисобланади:

Башорат қилинган векторда тўғри танланган контекстли сўз индекси (контекст ойнасидаги сўз индекси);

k шовқинли сўзлар индекси.

Буни мисолда кўриб ўтамиз. Бунда $k = 1$ (cat + glass):



57-сўзлар контекстини башорат қилиш схемаси.

TensorFlow да: skip-gram алгоритмининг амалга оширилиши

Бу қисмда барча қисмларни бирлаштириб, алгоритмни амалга оширилишини кўриб ўтамиз. Қуйидагиларни кўриб ўтамиз:

берилганлар генератори;

(TensorFlow)да skip-gram модели;

skip-gram алгоритмни ишга тушириш.

Берилганларни генерациялаш

Кодларни чуқур ўрганиб чиқмаймиз, чунки берилганларни генерациялаш ички механизмини кўриб чиқдик. Амалларни Python да кўриб чиқамиз:

```
def generate_batch (batch_size, window_size):
    global data_index
```

```

# two numpy arrays to hold target words (batch)
# and context words (labels)
batch = np.ndarray (shape=(batch_size), dtype=np.int32)
labels = np.ndarray (shape=(batch_size, 1), dtype=np.int32)

# span defines the total window size
span = 2 * window_size + 1

# The buffer holds the data contained within the span
queue = collections.deque(maxlen=span)

# Fill the buffer and update the data_index
for _ in range(span):
    queue.append(data[data_index])
    data_index = (data_index + 1) % len(data)

for i in range(batch_size // (2*window_size)):
    k=0
    # Avoid the target word itself as a prediction
    for j in list(range(window_size))+list(range(window_size+1,2*window_size+1)):
        batch[i * (2*window_size) + k] = queue[window_size]
        labels[i * (2*window_size) + k, 0] = queue[j]
        k += 1

# Everytime we read num_samples data points, update the queue
queue.append(data[data_index])

# If end is reached, circle back to the beginning
data_index = (data_index + np.random.randint(window_size)) %
len(data)

return batch, labels

```

skip-gram моделни аниқлаймиз

энг аввал баъзи бир гиперматнларни аниқлаймиз:

```
batch_size = 128
```

```
embedding_size = 64
```

```
window_size = 4
```

```
num_sampled = 32 # Number of negative examples to sample.
```

batch_size жорий вақтда биз қайта ишлайдиган берилганлар элементлар сонини белгилайди. embedding_size бу вектор узунлиги. Window_size гиперпараметри контекст ойна ўлчамини белгилайди. Ва охиргиси, num_sampled – йўқотиш функциясидаги мос келмайдиган намуналар сони (k). Сўнгра кирувчи ва чиқувчи берилганларни аниқлаймиз:

```
tf.reset_default_graph()
```

```
# Training input data (target word IDs).
```

```
train_dataset = tf.placeholder(tf.int32, shape=[batch_size])
```

```
# Training input label data (context word IDs)
```

```
train_labels = tf.placeholder(tf.int32, shape=[batch_size, 1])
```

train_dataset кириш қисмига сўзлар идентификаторлари рўйхатини batch_size қабул қилиб, танланган мақсадли сўзлар тўпламини ифодалайди. Танланган мақсадли сўзларга мос келувчи контекстли сўзлар batch_size рўйхатини Train_labels ифодалайди.

Сўнгра нейрон тармоқ параметрларини аниқлаймиз:

```
#####
```

```
#      Model variables      #
```

```
#####
```

```
# Embedding layer
```

```
embeddings = tf.Variable(tf.random_uniform([vocabulary_size, embedding_size], -1.0, 1.0))
```

```
# Neural network weights and biases
```

```
softmax_weights = tf.Variable(
```

```
    tf.truncated_normal([vocabulary_size, embedding_size],
```

```
stddev=0.1 / math.sqrt(embedding_size))
```

```
)
```

```
softmax_biases = tf.Variable(tf.random_uniform([vocabulary_size],-0.01,0.01))
```

Embedding layer -----TensorFlow да embeddings ўзгарувчи орқали, softmax_weights - ўзгарувчи оғирлик кўрсаткичи, силжиш параметри-softmax_biases орқали белгиланади.

Embedding layer ва нейротармоқни бирлаштириб, натижани оптималлаштирамиз:

```
# Look up embeddings for a batch of inputs.
```

```
embed = tf.nn.embedding_lookup(embeddings, train_dataset)
```

tf.nn.embedding_lookup функцияси кириш қисмига embedding layer ва сўзлар идентификаторлари тўпламини қабул қилади (train_dataset), чиқиш қисмига мос келувчи векторни узатади.

Sampled softmax loss функцияси :

```
#####
```

```
# Computes loss #
```

```
#####
```

```
loss = tf.reduce_mean(tf.nn.sampled_softmax_loss(weights=softmax_weights, biases=softmax_biases, inputs=embed, labels=train_labels, num_sampled=num_sampled, num_classes=vocabulary_size) )
```

Бунда tf.nn.sampled_softmax_loss кириш қисмига оғирлик кўрсаткичи (softmax_weights), олдинги функциялар асосида олинган силжишлар (softmax_biases), embed тўпламини, тўғри келувчи контекстли сўзлар идентификаторларини (train_labels), шовқинли намуналар сонини (num_sampled), луғат ҳажмини (vocabulary_size) қабул қилади.

embedding layer параметрлари ва нейротармоқ бўйича йўқотиш функциясини оптималлаштирамиз:

```
#####
```

```
# Optimization #
```



```
#####
optimizer = tf.train.AdamOptimizer(0.001).minimize(loss)
Нормируем embedding layer:
#####
#           For evaluation           #
#####
norm        =        tf.sqrt(tf.reduce_sum(tf.square(embeddings),      1,
keepdims=True))
normalized_embeddings = embeddings / norm
Кодни ишга туширамиз.
TensorFlow моделни қандай ишга тушириш лозим? Энг аввал
session ни аниқлаймиз ва ўзгарувчиларни тасодифий равишда
инициаллаштирамиз.
num_steps = 250001
session = tf.InteractiveSession()
# Initialize the variables in the graph
tf.global_variables_initializer().run()
print('Initialized')
average_loss = 0
Сўнгра олдиндан белгиланган қадамлар асосида берилганлар
гуруҳини шакллантирамиз: мақсадли сўзлар (batch_data) ва контекст
сўзлар (batch_labels):
for step in range(num_steps):
    # Generate a single batch of data
    batch_data, batch_labels = generate_batch( batch_size, window_size)
Сўнгра шакллантирилган ҳар бир гуруҳ учун embedding layer ни ва
нейрон тармоқни session.run([optimize, loss],...) ёрдамида
оптималлаштирамиз. Шу билан бирга хатоликни аниқлаймиз, бу
унинг камайишига ишонч ҳосил қилиш учун лозим бўлади.
# Optimize the embedding layer and neural network
# compute loss
feed_dict = {train_dataset : batch_data, train_labels : batch_labels}
_, l = session.run([optimizer, loss], feed_dict=feed_dict)
```

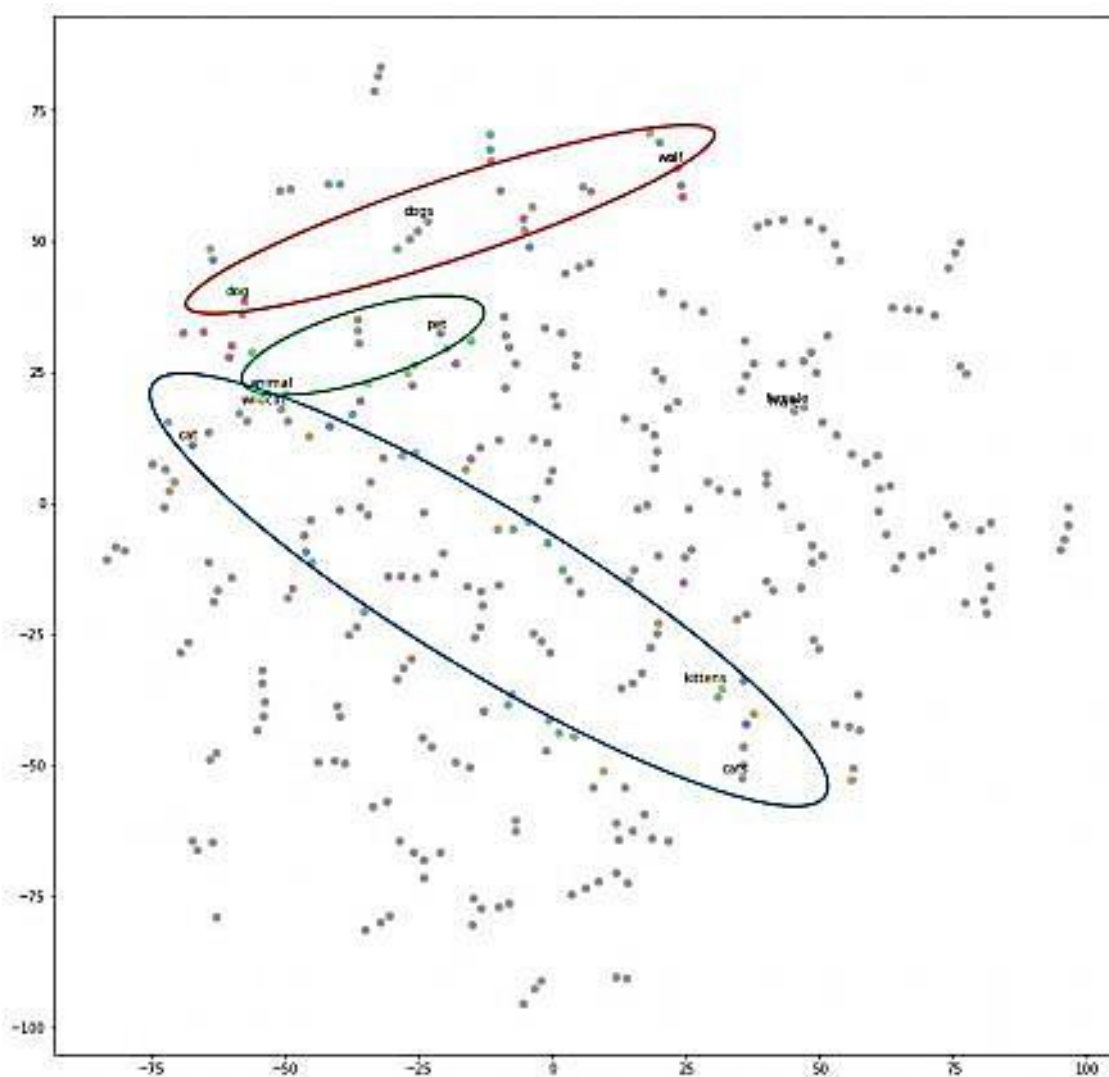
Ҳар бир беш минг кадамда экранга ўртача хатоликни чиқарамиз:

```
if (step+1) % 5000 == 0:  
    if step > 0:  
        average_loss = average_loss / 5000  
        print('Average loss at step %d: %f' % (step+1, average_loss))  
        average_loss = 0
```

натижада векторлар олиниб, у кейинчалик маълум бир сўзларни визуаллаштириш учун қўлланилиши мумкин:

```
sg_embeddings = normalized_embeddings.eval()  
session.close()
```

Агар натижани t-SNE га ўхшаш алгоритм ёрдамида визуаллаштирак қуйидаги кўринишни оламиз:

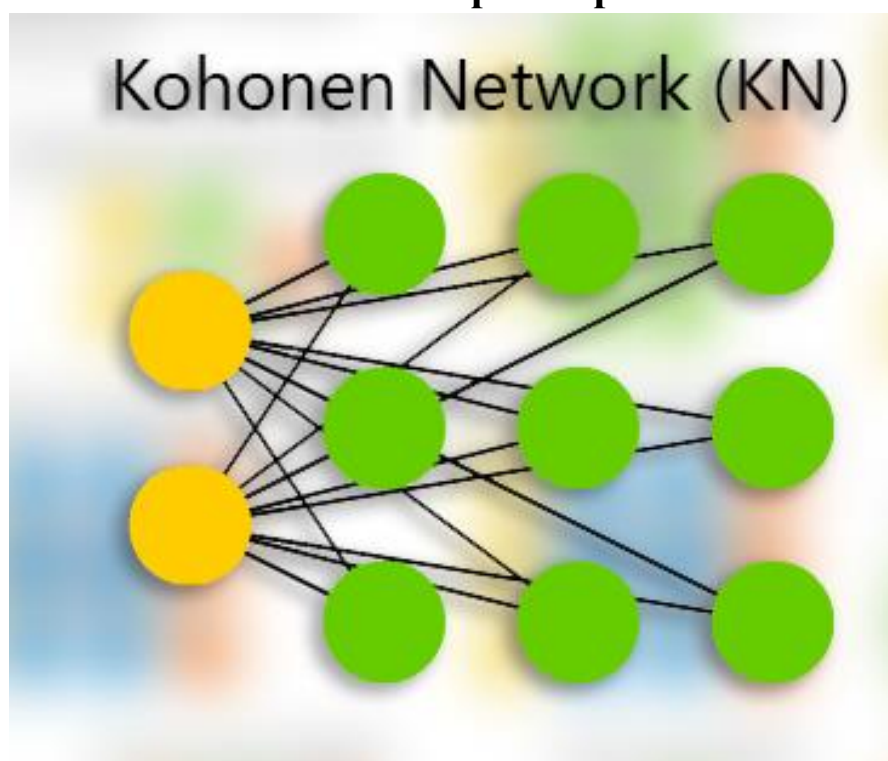


58-расм. Сўзларнинг семантик нуқтаи назардан жойлашуви.

Мушук сўзига яқин сўзларни кўриш мумкин, улар маълум бир майдонда жойлашган (cat, kitten, cats, wildcat), ит сўзига яқин бўлган сўзлар (dog, dogs, wolf) бошқа майдонда жойлашган. Бу сўзлар ўртасида жойлашган сўзлар (масалан, animal ёки pet) маъно жиҳатдан ҳам мушук ҳам ит сўзларига яқин эканлигини кўриш мумкин.

Хулоса қилиб айтганда, сўзларнинг векторли ифодаси– жуда аниқ ишловчи инструмент бўлиб, замонавий машина асосида ўрганишни яхшилашга улкан ҳисса қўшади. word2vec нинг асосий ишлаш принципини кўриб ўтдик. Сўнгра skip-gram алгоритми изоҳи унинг TensorFlow да амалга оширилишини кўриб ўтдик. Натижа кўринишда сўзларнинг векторли кўринишини визуаллаштириб, унинг семантикаси сақланганлигига ишонч ҳосил қилдик.

1.31.Кохонен нейрон тармоғи



59-расм. Кохонен тармоғи.

Кохонен нейрон тармоғи (Kohonen network, KN), ёки ўзи ташкилланувчи харита (self organising (feature) map, SOM, SOFM) рақобатли ўрганиш усули қўлланилиб, маълумотларни ўқитувчисиз таснифлаш учун қўлланилади. Тармоқ структураси битта ягона

нейронлар қатламига эга бўлиб, Кохонен қатлами деб номланади ва у мослашган чизиқли сумматорлардан иборат. KN кириш қисмига маълумотлар кирганда, тармоқ уларга максимал мос тушувчи нейронни аниқлайди. Сўнгра, нейронлар юқори аниқликда мослашиш учун кўшни нейронларни силжитиб ўзгаришади. Кохонен тармоғининг энг содда кўринишига кирувчи маълумотларни «ғолиб барчасини эгаллайди» қабалида қайта ишлайди. Чиқиш қатламининг ҳар бир нейрони икки қийматдан бирини: нол ёки бирни қабул қилади. Юқори сигнал 1 ни ўзлаштиради, қолган сигналлар 0га айланади[12].

Кохонен Нейрон тармоқлари нейротармоқ архитектурасининг ўқитувчисиз ўрганувчи кўринишига яққол мисол ҳисобланади. Ечиладиган масалаларда: маълумотларни кластерлаш ёки хусусиятларнинг башорати киради. Шу билан бирга Кохонен тармоғи маълумотларни максимал йўқотмаган ҳолда ҳажман қисқартириш имконини беради.

Олдин кўриб чиқилган нейрон тармоқлар танланмалар асосида ўқитувчи кўмагида ўрганиш имконига эга бўлиб, ўзига хос кирувчи ва чиқувчи векторлар жуфтлигидан иборат кўпгина мисолларни ўз ичига олади. Бунда чиқувчи қийматлар оғирлик коэффицентларнинг созланмаларида бевосита қатнашади. Кохонен нейрон тармоқларида чиқувчи векторлар ўрганувчи танланмаларда мавжуд бўлиши мумкин ёки мавжуд бўлмаслиги ҳам мумкин, ҳар ҳолда ўрганиш жараёнида қатнашмайди. Яъни синапсларни тўғрилашда чиқиш қисмлари мўлжал сифатида қўлланилмайди. Нейрон тармоқнинг айнан шу созланмаси принципи ўзи ўрганувчи деб номланади.

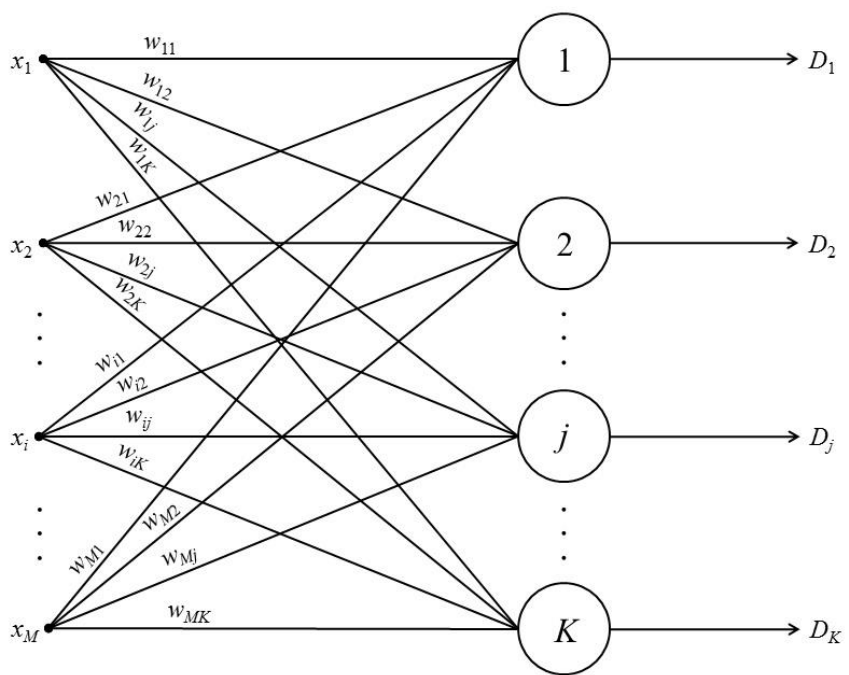
Кўриб ўтилаётган архитектурада сигналлар кириш қисмидан чиқиш қисмига тўғри йўналишда узатилади. Нейрон тармоқ структураси силжиш коэффиценти бўлмаган ягона нейрон (Кохонен) қатламига эга. Оғирлик коэффицентлар умумий сони кўпайтма кўринишда аниқланади:

$$N_W = MK \quad (10)$$

Нейронлар сони кластерлар сонига тенг бўлиб, ўрганилаётган мисолларни бошланғич тақсимлаш ва қайта тақсимлаш шу

кластерларда амалга оширилади. Нейрон тармоқнинг кириш қисмидаги ўзгарувчилар ўрганилаётган объект хусусиятини изоҳловчи омиллар сонига тенг ва у асосда қайси кластерга мансублиги аниқланади.

Кохонен нейрон тармоғи ўз-ўзини структуралаш ва ўз-ўзини ўрганиши жараёнини фарқлаш лозим бўлади. Оддий ўз-ўзини ўрганишда тармоқ аниқ белгиланган структурага эга, яъни нейронлар сони умуман ўзгармас қолади. Ўз-ўзини структуралашда аксинча, тармоқ доимий белгиланган структурага эга бўлмайди. Ҳолиб нейронгача бўлган масофага боғлиқ ҳолда ушбу нейрон берилган мисолни кластерлашда ёки кириш қисмига узатилган мисол учун маълум бир оғирлик кўрсаткичига эга янги кластер яратилади. Кохонен тармоғи ўз-ўзини структуралаш жараёнида алоҳида нейронлар четлаштирилиши мумкин.



60-расм. Кохонен нейрон тармоқ умумий структураси.

Кирувчи ўзгарувчиларни нормаллаштириш $[-1, 1]$ ёки $[0, 1]$ оралиғида амалга оширилади.

Берилган архитектурали нейрон тармоқлар фаолият жараёнида учта асосий босқичлар мавжуд: ўрганиш, кластерли таҳлил ва амалий қўлланилиши.

Кохонен ўрганиш алгоритми босқичлардан иборат бўлиб, унинг таркиби структура типига боғлиқ: доимий (ўз-ўзини ўргатувчи тармоқ) ёки ўзгарувчан (ўз-ўзини структуралайдиган тармоқ). Ўз-ўзини ўргатувчи тармоқда қўйидаги кетма-кетлик бажарилади:

1. Тармоқ структурасини ифодалаш (Кохонен қатлами нейронлар сони) (К).

2. Оғирлик коэффициентининг қийматлар асосида тасодифий инициаллаштириш, қўйидаги шартларни қаноатлаштириши лозим:

–бошланғич танлов $[-1, 1]$ ораликда нормаллаштириш:

$$|w_{ij}| \leq \frac{1}{\sqrt{M_t}} \quad (11)$$

– бошланғич танлов $[0,1]$ ораликда нормаллаштириш:

$$0.5 - \frac{1}{\sqrt{M}} \leq w_{ij} \leq 0.5 + \frac{1}{\sqrt{M}} \quad , \quad (12)$$

Бунда M – тармоқнинг кириш қисмидаги берилганлар сони–изланиш олиб борилаётган объект хусусиятлари кўрсаткичлари.

3. Жорий ўрганиш даври бўйича кириш қисмига тасодифий мисол узатилиши ва кириш қисми векторидан барча кластерлар марказигача бўлган Евклид масофа аниқланиши:

$$R_i = \sqrt{\sum_{i=1}^M (\tilde{x}_i - w_{ij})^2} \quad (13)$$

4. R_j Энг кичик қиймат бўйича ғолиб нейрон танланади j , кириш қисмидаги вектор қийматига энг яқин бўлади. Фақат танланган нейрон учун оғирлик коэффициентлари коррекцияси амалга оширилади:

$$w_{ij}^{(q+1)} = w_{ij}^{(q)} + v (\tilde{x}_i - w_{ij}^{(q+1)}), \quad (14)$$

Бунда v –ўрганиш тезлиги коэффициенти.

5. Такрорланиш 3 қадамдан бошланиб жараёни бир ёки бир неча тугаллаш шартлари бажарилгунча такрорланади:

–ўрганиш даврининг тугалланиши;

–ўрганиш даврининг сўнгида оғирлик кўрсаткичи коэффицентининг берилган ораликда сезиларли даражада ўзгармаганлиги;

–ўрганиш учун белгиланган вақт чегарасининг тугаганлиги.

Ўрганиш тезлиги коэффиценти $[0, 1]$ ораликдаги доимий қиймат орқали ёки бир даврдан бошқа даврга ўтиш жараёнида камайиб боровчи ўзгарувчи қиймат билан берилиши мумкин[12].

Кохонен тармоғининг ўз-ўзини структуралаш алгоритми маълум бир ўзгаришларга дуч келади:

1. Критик масофа $R_{кр}$ берилиб, кириш қисмига кирувчи мисол ва ғолиб нейрон оғирлик кўрсаткичи орасидаги энг максимал Евклид масофага тўғри келади. Бошланғич структура нейронларга эга бўлмайди. Тармоқ кириш қисмига ўрганувчи танлов учун энг биринчи мисол узатилганда биринчи нейрон яратилиб, унинг оғирлик коэффиценти кириш қисмида берилган қийматга тенг қилиб олинади.

2. Тармоқ кириш қисмига жорий ўрганиш даврига мос келувчи янги тасодифий танланган мисол узатилади. Берилган мисол ва ҳар бир кластер марказигача бўлган Евклид масофа (3) ифода орқали ҳисобланади ва ғолиб-нейрон энг кичик масофа R_{min} асосида аниқланади.

3. Агар $R_{min} \leq R_{кр}$, шарт бажарилса, мос ҳолда ғолиб нейрон оғирлик коэффиценти коррекцияси (4) ифода бўйича амалга оширилади. Акс ҳолда тармоқ структурасига янги нейрон қўшилиб, оғирлик коэффиценти кириш қисмига берилган мисол қийматлари билан бир хил бўлган оғирлик коэффицент қийматига тенг қилиб олинади.

4. Ушбу процедура 2 бўлимдан бошлаб такрорланади, агар ўрганишнинг сўнги даврида бирон-бир кластер фойдаланилмай қолса, ушбу нейронлар Кохонен тармоғи структурасидан четлаштирилади.

5. Белгиланган структурали ўз-ўзини ўрганиш алгоритмига эга тармоқда кўрсатилган шартнинг биронтаси бажарилса ҳисоблаш тўхтатилади.

Ўз-ўзини ўрганиш ва ўз-ўзини структуралаш алгоритмининг яна бир модификациясида ғолиб нейрон билан бир қаторда бошқа нейронларнинг ҳам оғирлик коэффицентлари коррекцияланади. Бунинг учун ўрганиш тезлиги коэффицентлари қўлланилиб, кластер марказигача масофа R_j ошиши натижасида мос равишда камайиб боради:

$$v_j = v_0 \left(1 - \frac{1}{1 + e^{-\beta(R_j - R_{xp})}} \right), \quad (15)$$

Бунда $R_{кр}$ –масофанинг критик қиймати: қанчалик қисқа бўлса, шу даражада ўрганилаётган мисолда кластерга яқин оғирлик кўрсаткичлари сезиларли даражада тахрирланади. Узок жойлашганлари эса аҳамиятсиз ҳолда ўзгаради; β –параметр бу масофанинг тезлик коэффицентига чизиқсиз боғлиқлик даражасини белгилайди; v_0 – бу жорий ўрганиш даврига таъсир ўтказувчи тезлик коэффицентлари (максимал мавжуд бўлган) базали.

Берилган мисолни ўрганиш жараёнида ҳар бир кластер учун ўртача масофани R_k қиймати сифатида қабул қилиш мумкин. β параметрни $3,0 \pm 0,5$ га тенг деб қабул қилиш тавсия этилади.

Одатда, Кохонен нейрон тармоғида ўз-ўзини структуралашда яна бир муаммога дуч келиш мумкин. Бир томондан, баъзи бир кластерлар маълумотлари ҳажми жуда ҳам кам бўлиб, кейинчалик уларни умумлаштиришда қийинчилик туғдириш мумкин. Иккинчи томондан, баъзи кластерлар аксинча жуда катта ҳажмда мисолларга эга бўлганда, кластер ҳажмини тахрирлаш ва ўта юкланганлик муаммосини ҳал қилиш учун қўшимча параметр белгилаб, у асосда ҳар бир кластер ($N_{пр}$) яратилиши учун чекланмаларни белгилайди.

Агар янги мисол бирон бир кластерга тааллуқли бўлиб, бироқ бу кластер максимал ҳажмга эга бўлса, янги кластер яратилиб, унинг марказида кластер мисоллардан бирининг ($N_{пр}+1$) ўзгарувчилари векторини ташкил этади, бироқ улар жорий кластер марказидан узок масофада жойлашган бўлади.

Ўрганиш жараёнидан ўтган нейрон тармоқ устида кластерли таҳлил процедураси ўтказилади яъни, бу – кластерни шакллантирган мисоллар таркиби сон ва сифат жиҳатдан таҳлил этиб, улар асосида кластер хусусиятларини изоҳлаш процедураларидир. Шунини таъкидлаш лозимки, кластерларни изоҳлашда фақатгина кириш қисмига қабул қилинаётган ўрганилаётган танловнинг берилганлари қийматлари эмас, балки кластерларни шакллантиришда иштирок этмайдиган ўзгарувчилар қийматлари ҳам инобатга олинади. Хусусан, бу изоҳда барча кластерни шакллантирган барча мисоллар қийматларининг ўртачалари борасидаги маълумотлар ҳам кириши мумкин. Бундан ташқари, ҳар бир кластер учун унда қатнашувчи мисоллар бўйича ўрта квадратик силжишлар ёки дисперсия борасида ҳам маълумотларга эга бўлиш мақсадга мувофиқ.

Кохонен нейрон тармоғини амалий қўллашда янги мисол кириш қисмига узатилади ва у мавжуд бўлган бирон бир кластерга боғлиқ ёки (яқин атрофдаги кластер марказигача масофа узун бўлган ҳоллар учун) бирорта кластерга ҳам боғлиқ эмас деб қабул қилинади. Агар кластерга боғлиқлик аниқланса, кластерли таҳлил асосида олинмайдиган изоҳи ва кластер бўйича ечимлар қабул қилинган мисолга жорий этилиши лозим.

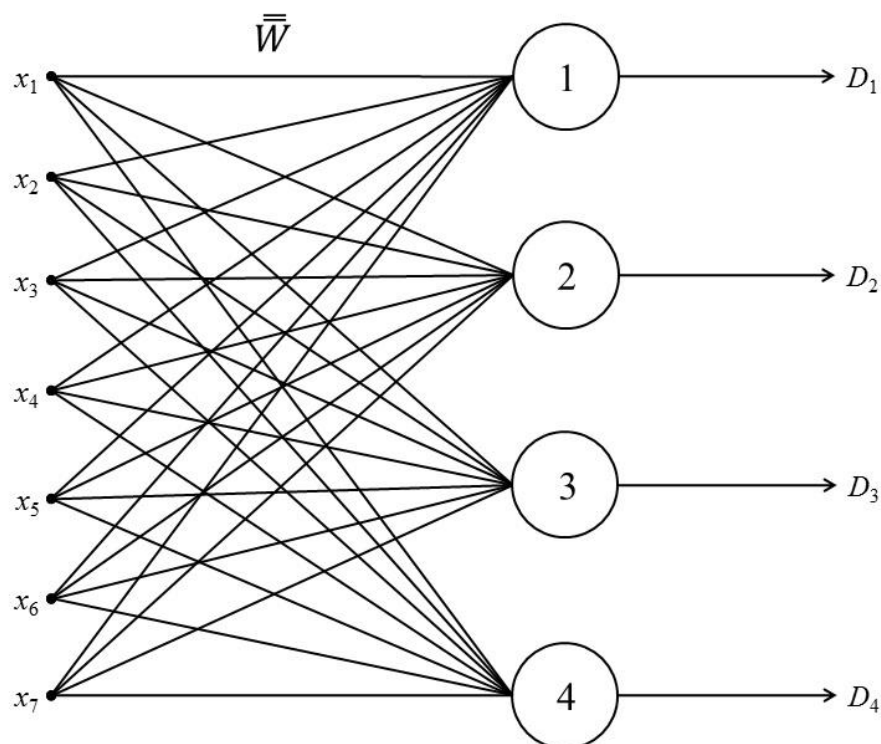
Кластерлаш натижаларини визуаллаштириш ҳисобидан Кохонен тармоғини амалий қўллаш енгиллашади. Натижада тармоқда ўз-ўзини ўрганиш (ўз-ўзини структуралаш) жараёни кластерлар тўплами ҳосил бўлиб, уларнинг ҳар бири ўзининг маркази (мас равишда нейрон оғирлик коэффицентлари қийматлари) ва уни шакллантирган мисоллар сони асосида олиб борилади. Мавжуд кластерлар марказлари орасидаги Евклид масофани аниқлаш ва уларни Кохонен харитасида график изоҳлаш– икки ўлчовли структура кўринишга келтириш осон бажарилиб, бу графика асосида ҳар бир кластер ҳажми ва кўринишини аниқлаш билан бирга унинг қўшни кластерлари борасидаги маълумотларни ҳам кўриш мумкин.

Кохонен нейрон тармоғида ўз-ўзини ўрганиш жараёнига мисол кўриб ўтамиз, бунда талабалар гуруҳининг ўзлаштириш кўрсаткичини ўрганиш мисолида бўлсин. Бошланғич танлов б-жадвалда келтирилган.

Жадвал 6. Талабалар ўзлаштириш тўғрисидаги бошланғич маълумотлар тўплами

Ми- сол №	Исми, Шарифи, Фамилия	Жин- си x_1	Барча ўтиш балли ни олиш x_2	Фанлар бўйича рейтинги:					стипен дия коэфф. x_8
				тарих x_3	мух. Графи- ка x_4	Ма- тем. x_5	орг. кимё x_6	физи ка x_7	
1	Варданян	Э	Ҳа	60	79	60	72	63	1,00
2	Горбунов	Э	Йўқ	60	61	30	5	17	0,00
3	Гуменюк	А	Йўқ	60	61	30	66	58	0,00
4	Егоров	Э	Ҳа	85	78	72	70	85	1,25
5	Захарова	А	Ҳа	65	78	60	67	65	1,00
6	Иванова	А	Ҳа	60	78	77	81	60	1,25
7	Ишонина	А	Ҳа	55	79	56	69	72	0,00
8	Климчук	Э	Йўқ	55	56	50	56	60	0,00
9	Лисовский	Э	Йўқ	55	60	21	64	50	0,00
10	Нетреба	Э	Йўқ	60	56	30	16	17	0,00
11	Остапова	А	Ҳа	85	89	85	92	85	1,75
12	Пашкова	А	Ҳа	60	88	76	66	60	1,25
13	Попов	Э	Йўқ	55	64	0	9	50	0,00
14	Сазон	А	Ҳа	80	83	62	72	72	1,25
15	Степоненко	Э	Йўқ	55	10	3	8	50	0,00
16	Терентьева	А	Ҳа	60	67	57	64	50	0,00
17	Титов	Э	Ҳа	75	98	86	82	85	1,50
18	Чернова	А	Ҳа	85	85	81	85	72	1,25
19	Четкин	Э	Ҳа	80	56	50	69	50	0,00
20	Шевченко	Э	Йўқ	55	60	30	8	60	0,00

Берилган тақсимоти аниқ 4 та кластер бўйича амалга оширилиши лозим. Кириш қисмига қабул қилинувчи берилганлар сифатида $x_1 - x_7$ ўзгарувчилар хизмат қилса, x_8 ўзгарувчи эса ўрганишда қўлланилмайди, бироқ унинг қиймати кластерли таҳлилда қўлланилади. Шу кўринишда тармоқ битта нейронлар қатлаидан иборат бўлиб, 7 та кириш ва 4 та чиқиш қисмига эга бўлади (Расм. 61).



61-расм. Талабалар маълумотларини кластерлашнинг Кохонен нейрон тармоқ структураси.

Кириш қисмига қабул қилинувчи ўзгарувчилар аналогли қийматларини чизиқли нормаллаштириш амалини $[0, 1]$ оралиқ учун (3.1)-тенглама бўйича бажарамиз. Қуйидаги усулда дискрет қийматларни изоҳлаймиз:

- талаба жинси: 0 – аёл, 1 – эркак;
- барча фанлардан ўтиш балини олиш: 0 – йўқ, 1 – ҳа.

Нормаллаштириш натижаси қуйидаги жадвалда (7) келтирилган.

Нейрон тармоқнинг 28 та оғирлик коэффициентларини қийматлар билан инициаллаштириб чиқамиз (8- жадвалда), бунда (7) келтирилган чекланмалар инобатга олинган.

Жадвал 7. Талабалар ўзлаштириш маълумотларининг нормаллаштирилган танлови

Мисол №							
1	1,00	1,00	0,17	0,78	0,70	0,77	0,68
2	1,00	0,00	0,17	0,58	0,35	0,00	0,00
3	0,00	0,00	0,17	0,58	0,35	0,70	0,60
4	1,00	1,00	1,00	0,77	0,84	0,75	1,00
5	0,00	1,00	0,33	0,77	0,70	0,71	0,71
6	0,00	1,00	0,17	0,77	0,90	0,87	0,63
7	0,00	1,00	0,00	0,78	0,65	0,74	0,81
8	1,00	0,00	0,00	0,52	0,58	0,59	0,63
9	1,00	0,00	0,00	0,57	0,24	0,68	0,49
10	1,00	0,00	0,17	0,52	0,35	0,13	0,00
11	0,00	1,00	1,00	0,90	0,99	1,00	1,00
12	0,00	1,00	0,17	0,89	0,88	0,70	0,63
13	1,00	0,00	0,00	0,61	0,00	0,05	0,49
14	0,00	1,00	0,83	0,83	0,72	0,77	0,81
15	1,00	0,00	0,00	0,00	0,03	0,03	0,49
16	0,00	1,00	0,17	0,65	0,66	0,68	0,49
17	1,00	1,00	0,67	1,00	1,00	0,89	1,00
18	0,00	1,00	1,00	0,85	0,94	0,92	0,81
19	1,00	1,00	0,83	0,52	0,58	0,74	0,49
20	1,00	0,00	0,00	0,57	0,35	0,03	0,63

Жадвал 8. Кохонен тармоқнинг оғирлик коэффицентлари бошланғич қийматлари

<i>j</i> кластер №	Оғирлик коэффицентлари w_{ij}						
	w_{1j}	w_{2j}	w_{3j}	w_{4j}	w_{5j}	w_{6j}	w_{7j}
1	0,20	0,20	0,30	0,40	0,40	0,20	0,50
2	0,20	0,80	0,70	0,80	0,70	0,70	0,80
3	0,80	0,20	0,50	0,50	0,40	0,40	0,40
4	0,80	0,80	0,60	0,70	0,70	0,60	0,70

Ўз-ўзини ўрганиш алгоритмини қуйидаги кўринишда созлаймиз. Энг аввал ўрганиш тезлигининг бошланғич коэффицентини 0,30-га тенг деб оламиз, у 0,05 қадам билан ўзгарсин. Натижада, 6 та ўрганиш даври турли тезлик коэффицентини билан бажарилади, уларнинг ҳар бирида битта нейрон учун 20 та оғирлик коэффицентини таҳрири бажарилади.

Нейрон тармоқ кириш қисмига тасодифий танланган мисол № 10 узатилсин. (3) ифодага асосланиб, тўртта кластернинг жорий марказигача масофани ҳисоблаймиз. Улар мос равишда 0,98, 1,65, 0,65 ва 1,32 га тенг бўлади. Энг қисқа масофа учинчи кластерга мос келади ва хулоса қилсак, учинчи нейрон-ғолиб нейрон ҳисобланади, шунинг учун айнан шу нейрон оғирлик кўрсаткичи (11.4) ифода бўйича таҳрирланиши лозим бўлади.

Оғирлик коэффицентларнинг янги қийматлари: $w_{13} = 0,86$, $w_{23} = 0,14$, $w_{33} = 0,40$, $w_{43} = 0,51$, $w_{53} = 0,39$, $w_{63} = 0,32$, $w_{73} = 0,28$ га тенг бўлади. Қолган нейронлар оғирлик кўрсаткичлари ўзгармас қолади.

Сўнгра шу усулда тармоқ кириш қисмига танлов кетма-кетлиги тасодифий қабул қилинади. Барча 20 та мисол қабул қилингандан сўнг, ўрганишнинг кейинги даври бошланади, бунда ўрганиш тезлиги коэффицентини 0,05 га қисқартирамиз. Кохонен тармоғи тўлиқ ўрганиш цикли натижасида оғирлик кўрсаткичларининг якуний қиймати олинади, у жадвал 4-да келтирилган.

Жадвал 9. Кохонен тармоғи тўлиқ ўрганиш цикли натижасида оғирлик кўрсаткичларининг якуний қиймати

<i>j</i> кластер №	w_{ij} оғирлик коэффицентлари						
	w_{1j}	w_{2j}	w_{3j}	w_{4j}	w_{5j}	w_{6j}	w_{7j}
1	0,06	0,06	0,21	0,52	0,36	0,55	0,57
2	0,00	1,00	0,50	0,80	0,80	0,80	0,73
3	1,00	0,00	0,04	0,48	0,26	0,22	0,42
4	1,00	0,99	0,69	0,77	0,79	0,78	0,81

Ўргатилган нейрон тармоқ учун кластер таҳлил процедурасини бажарамиз. Барча тўртта кластер тўлдирилган бўлиб, бироқ уларнинг сонли қиймати фарқланади. 1- кластер биттагина – № 3 га мос келди.

2- энг катта ҳажмдаги кластерга № 5–7, 11, 12, 14, 16, 18 мисоллар мос келди. 3- кластерга № 2, 8–10, 13, 15, 20 мисоллар мос келди. 4-сига – № 1, 4, 17, 19. Таҳлил натижасида олинган сифат кўрсаткичлари ва кластерлар изоҳи жадвал 5 да келтирилган.

Кластерлар изоҳида стипендияларни олиш борасидаги хулоса x_8 ўзгарувчи қийматлари таҳлили асосида олинди, бу ўзгарувчи ўрганиш жараёнида қатнашмади.

Жадвал 10. Кластерли таҳлил натижаси

Кластер №	Кластер ҳажми	жинси	Барча ўтиш баллини олиш	Ўртача рейтинг	Стипендия Коэфф.	Изоҳ
1	1	А	Йўқ	55	0,00	Қониқарли ўзлаштирган, битта ёки бир нечта фандан ўтиш балини олмаган ва стипендия олмайдиган талаба қизлар
2	8	А	Ҳа	72	0,97	Яхши ўзлаштирган, барча фанлардан ўтиш балини олган ва стипендия оладиган талаба қизлар.

3	7	Э	Йўқ	40	0,00	Қониқарсиз ўзлаштирган, битта ёки бир нечта фандан ўтиш балини олмаган ва стипендия олмайдиган талаба йигитлар
4	4	Э	Ҳа	73	0,94	Яхши ўзлаштирган, барча фанлардан ўтиш балини олган ва стипендия оладиган талаба йигитлар.

Олинган изоҳлар ва ўрганган нейрон тармоқга асосланган ҳолда бошқа бирон бир талаба борасида айнан шу кўринишдаги берилганларни киритиб маълумот олиш мумкин бўлади. Шунитак таъкидлаш лозимки, амалий кўллаш босқичида нормаллаштириш чегарасидан чиқувчи берилганларни учратиш мумкин, бироқ бу «Ғолиб»-нейронни танлаш ва бу янги мисолни мавжуд кластерга боғлаш муаммосиз бажарилади.

1.31.1.Кохонен тармоқда ўрганиш

Кохонен тармоқларида ўқитувчисиз ўрганиш усули қўлланилади. Тармоқни ўрганиш учун рақобатлик механизми қўлланилади. Тармоқнинг кириш қисмига x вектори узатилганда кирувчи вектор оғирлик кўрсаткичидан кам фарқланадиган нейрон «Ғолиб» чиқади. «Ғолиб» нейрон учун -кўйидаги ифода амалга оширилади

$$d(x, w_j) = \max_{1 \leq i \leq n} d(x, w_i) \quad (16)$$

бунда n –нейронлар сони, j – «Ғолиб» нейрон рақами, $d(x, w_j)$ – x ва w векторлар ўртасидаги масофа.

Объектлар ўртасида масофа кўрсаткичи сифатида кўпинча Евклид ўлчами қўлланилади

$$d(x, w_j) = \|x - w_j\| = \sqrt{\sum_{j=1}^m (x_j - w_{ij})^2} \quad (17)$$

бошқа кўринишдаги ўлчамлар қўлланилади.

Фаолликнинг рақобатлик функцияси сумматорлар қийматларини таҳлил этади ва нейрондан чиқувчи сигналларни шакллантиради, битта энг максимал қийматга эга «Ғолиб - нейрон»дан ташқари барча нейронлар учун 0 ни шакллантиради. Демак, чиқиш вектори битта элементи 1 га қолганлари 0 га тенг бўлади. *Фаол нейрон рақами* кирувчи вектор мансуб бўлган гуруҳ(кластер)ни аниқлайди.

Кохонен тармоғида кирувчи қийматларни нормаллаштириш мақсадга мувофиқ. Бунинг учун қўйидаги формулалардан фойдаланиш мумкин:

$$x_{Ni} = \frac{x_i}{\sqrt{\sum_{i=1}^m x_i^2}}, \quad x_{Ni} = \frac{x_i}{|x_i|} \quad (18)$$

бунда x_{ni} – кирувчи векторнинг нормаллаштирилган компоненти.

Кирувчи маълумотларни нормаллаштириш тармоқ ўрганиш жараёнини тезлаштиради.

Ўрганиш жараёнидан олдин тармоқнинг инициализацияси олиб борилади, яъни оғирликлар векторини қийматлари аниқланади. Содда ҳолларда оғирлик кўрсаткичлари тасодифий берилади. *Кохонен тармоғи ўрганиш жараёни* бир қатор қадамларнинг циклик такрорланишидан иборат:

1. кириш қисмига бошланғич маълумотларнинг узатилиши. Одатда бу кирувчи векторлар ичидан тасодифий танланган ҳисобланади.

2. Ҳар бир нейрон чиқишдаги натижани аниқлаш.

3. «Ғолиб» нейронни аниқлаш (Улар кирувчи вектор компонентлардан оғирлик кўрсаткичи билан унча кўп фарқ қилмайди).

4. «Ғолиб» чиққан нейрон оғирлик кўрсаткичини *Кохонен қоидаси* бўйича созлаш.

$$w_i^{k+1} = w_i^{(k)} + \eta_i^{(k)} [x - w_i^{(k)}] \quad (19)$$

бунда \mathbf{x} – кирувчи вектор, k – ўрганиш цикли рақами, $\eta_i^{(k)}$ – i - нейроннинг k - ўрганиш циклидаги ўрганиш тезлиги коэффициентлари.

5. Агар ўрганиш тугамаган бўлса, биринчи қадамга ўтиш.

Демак, оғирликлар векторига кирувчи векторлар яқин бўлган нейрон, янада яқин бўлиши учун янгиланади. Натижада ушбу нейрон кирувчи қисмга берилган векторлар ўзига яқин бўлса ютади, ўзидан анча фарқланадиган вектор узатилса, ютқазади. Ўрганиладиган векторларнинг кўп узатилиши натижасида шундай нейронга эришиладики, қайсики вектор кластерга мансуб бўлса 1 ни, вектор кластерга мансуб бўлмаса 0 ни беради. Демак, шу усулда тармоқ кирувчи векторларни таснифлашни ўрганади.

Кохонен тармоғи ўрганишида «суст» нейронлар муаммоси пайдо бўлади. Кўшни қатламларнинг чекланган фарқи баъзи нейронларнинг таъсирланмаслиги. Бошланғич оғирлик векторларга эга ва кирувчи векторлардан узоқликда жойлашган нейронлар рақобатлашиш имконияти бўлмайди, бунда фарқи йўқ канча даражада ўрганиш жараёни олиб борилсин. Натижада бундай нейронлар ўрганиш жараёнида қатнашмайди ва ҳеч қандай ҳолда ҳам «ютган» ҳолатда бўлмайди. Бундай нейронлар «суст ёки ўлик» нейронлар деб юритилади, чунки улар ҳеч қандай аҳамиятли функцияни бажармайди. Демак, кирувчи маълумотлар кам сонли нейронлар томонидан ифодаланилади. Шу сабабли барча нейронлар «ғолиб» бўлишига имкон бериш лозим. Бунинг учун ўрганиш алгоритмини ўзгартириб, «ўлик» нейронлар фаол қатнашишини таъминлаш лозим бўлади.

Масалан, ўрганиш алгоритмини ўзгартириб, ютган нейрон фаоллиги йўқотилади. Нейронлар фаоллигини руйхатга олиш усулларида бири p_i потенциалларни ўрганиш жараёнида ҳисоблаш. Энг аввали нейронларга p потенциал ўзлаштирилади, n –нейронлар

(кластерлар) сони. k - ўрганиш циклида потенциал қўйидагича аниқланади:

$$p_i(k) = \begin{cases} p_i(k-1) + \frac{1}{n}, i \neq j, \\ p_i(k-1) - p_{\min}, i = j, \end{cases} \quad (20)$$

бунда j –нейрон-ғолиб рақами.

Агар потенциал қиймати $p_i(k)$ p_{\min} маълум бир даражадан пасайса, нейрон– "дам олиш" ҳолатидан чиқади. $P_{\min} = 0$ бўлган ҳолат учун нейронлар курашишдан четлаштирилмайди. $P_{\min} = 1$ ҳолат учун нейронлар навбат билан ғолиб чиқишади, чунки ҳар бир ўрганиш цикли учун фақат биттаси курашга тайёр бўлиб қолади. Амалиётда эса жуда яхши натижа $p_{\min} \gg 0.75$ бўлган ҳолатда олинади.

Neural Network Toolbox алгоритмида “ўлик” нейронлар олдини олиш учун нейронлар силжишини ўзгартириш усули қўлланилади [9]. Ўлик нейронлар таъсирсизлигини инобатга олувчи созланмалар *learncon* (силжишни ўрганиш) функцияси кўринишда қўйидагича амалга оширилади. Созлашнинг биринчи амалида рақобатлашаётган қатлам нейронларига бир хил кўринишдаги фаоллик параметри ўзлаштирилади

$$c_0 = \frac{1}{N}, \quad (21)$$

бунда $N - N$ рақобатлашаётган қатлам нейронлар сони кластерлар сонига тенг. Созлаш жараёнида *learncon* (силжишни ўрганиш) функцияси ушбу параметр қийматларини фаол нейронлар учун ошириб, фаол бўлмаган нейронлар учун камайтириб ўзгартиради. Фаоллик параметрлари вектори формуласи қўйидаги кўринишда бўлади:

$$c^{k+1} = (1 - r_i)c^{(k)} + r_i s^{(k)}, \quad (22)$$

бунда r_i – созлаш тезлиги параметри; k – ўрганиш цикли рақами; $s^{(k)}$ – ўрганиш k – циклидаги адаптив сумматорлар чиқиш векторлари. Силжиш вектори компонентлари қўйидаги формула бўйича аниқланади

$$b_i^{(k+1)} = e^{(1 - \ln c_i^{k+1})} - b_i^{(k)}, \quad (23)$$

Ғолиб – нейрон учун силжиш кўрсаткичи камаяди, бошқа нейронлар учун ошади. Созлаш тезлиги параметри r_i жимлик ҳолати

бўйича 0.001 га тенг. Фаол нейронлар учун силжиш кўрсаткичининг ошиши кириш қийматлари диапазонини кенгайтириш имконини беради, нофаол нейрон эса кластерни шакллантиришни бошлайди. Натижада янги кириш векторларини жамлашни бошлайди. Бу иккита афзалликни беради. Нейрон оғирлик кўрсаткичи кириш қисми векторидагидан фарқланганлиги учун рақобатда ғолиб чиқмайди ва ўрганиш даврида силжиш ҳажми етарлича катта бўлиб, рақобатбардош бўлиб қолади. Натижада оғирлик векторлари маълум бир векторлар гуруҳига яқинлашиб қолади. Нейрон ғолиб чиқганда унинг силжиш кўрсаткичи қисқаради. Чунончи, “ўлик” нейронларнинг фаоллашуви масаласи ҳал килинди.

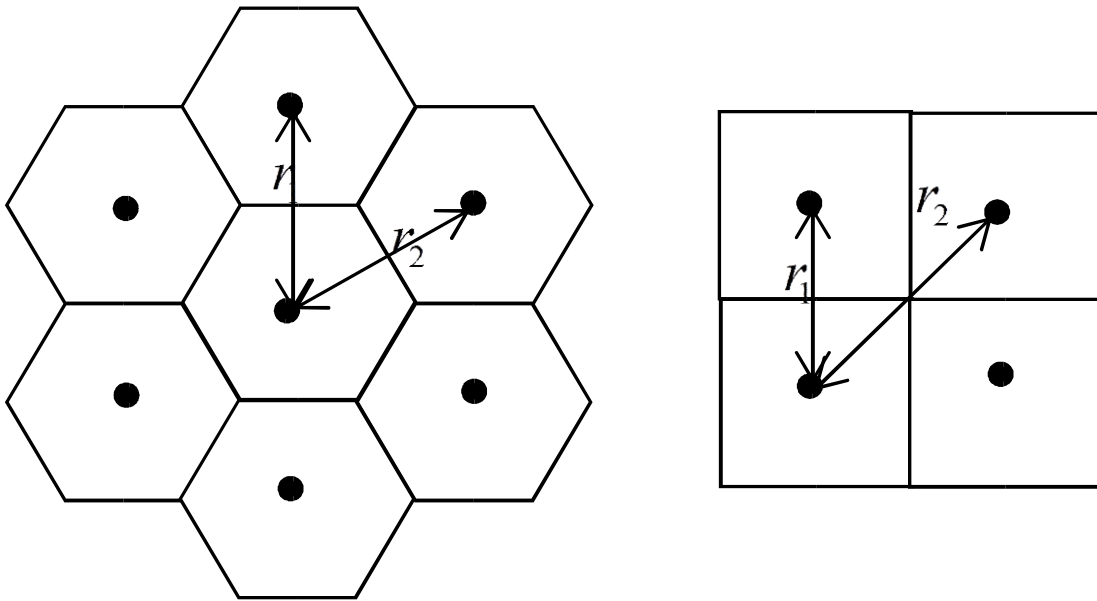
Иккинчи афзаллик бу силжиш билан боғлиқ бўлиб, фаоллик параметри қийматларини тўғрилаб чиқиш ва кириш қисми векторлари сонини тахминан бир хил бўлишини таъминлаш имконини беради. Демак, агар кластерлардан бири бошқасига нисбатан кириш векторлари сони кўп бўлса, тўлиб кетган соҳаси қўшимча нейронларни қабул қилиб ҳажми жиҳатдан кичик кластерларга бўлинади.

1.31.2.Кохонен харитаси

Кохонен харитасини тузиш принципи

Кохонен харитаси (ўз-ўзини структураловчи харита, ёки SOM – self-organizing map) [1–4, 10] икки ўлчовли харитада объектнинг кўп ўлчовли хусусиятларини визуаллаштириш учун қўлланилади. Кохонен харитаси катта ҳажмдаги кириш берилганларини кичик ўлчамдаги массив элементлари (одатда, икки ўлчовли) кўринишда ифодалаш амалини бажаради. Кохонен харитаси Кохонен тармоғига ўхшаб кетади. Ундаги фарқ, кластерлар маркази бўлмиш нейронлар харитада маълум бир структурада (одатда, икки ўлчовли графикада) тартибланган. Ўрганиш жараёнида харитада ғолиб –нейрон билан бир қаторда қўшни нейронлар оғирлик кўрсаткичи ҳам соланади. Натижада, Кохонен тармоғида маълум бир метрикадаги кириш векторлари битта нейрон (кластер маркази)га боғлиқ, Кохонен харитасида эса турли бир-бирига яқин нейронларга боғлиқ бўлиши

мумкин. Одатда нейронлар икки ўлчовли графикада тўртбурчак ёки олтибурчакли катаклар тугунларида жойлашган бўлади. Қўшни - нейронлар харитада жойлашганлик масофаси бўйича аниқланади. Расм 62 да олтибурчак ва тўртбурчакли катаклар марказида жойлашган нейронлар кўриниши келтирилган. Олтибурчакли катаклар харитадаги объектлар ўртасидаги Декарт масофани аниқроқ ифодалайди, чунки қўшни катаклар учун марказгача масофа бир хил.



62-расм. Олтибурчак ва тўртбурчак катаклар.

Ҳар бир катакка Кохонен тармоғининг бир нейрони мос келади. Яъни Кохонен харитасида катаклар сони нейронлар сонига тенг, Кохонен тармоғидаги нейронлар кўп сони кластерлар сонига тенг. Харитада катаклар сони тасвири аниқ ифодалаш даражасига боғлиқ ва у экспериментал танланади.

Катакка тушган кириш вектори танланган компоненти хусусиятлари ҳар бир катак учун ҳисобланади. Ушбу характеристика кўрсаткичига мос равишда маълум бир рангга бўялади.

Кохонен харитасида катак рангига қараб мавжуд кластер структураси, кластерлар сони ва алоҳида ўзгарувчилар ўртасида боғланишлар борасида қандайдир гипотеза юритиш мумкин. Бу гипотезалар эса бошқа усуллар ёрдамида текширилиши ва

тасдиқланиши мумкин. Хариталар одатда берилганларни бошланғич таҳлил босқичида қўлланилади, яъни аниқ натижа олиш учун эмас, балки масаланинг умумий кўринишини тушуниш учун қўлланилади.

1.31.3. Кохонен харитасида ўрганиш

Кетма-кет ўрганиш алгоритми

*Кетма-кет (Iterative) ўрганиш алгоритми*да оғирлик кўрсаткичи янгиланиши ўқув танланма ўрганилгандан сўнг нейронларда ўрганиш жараёни Кохонен тармоғи нейронлари асосида олиб борилади. Уларнинг фарқи шундаки, ғолиб-нейрон билан бир қаторда *атрофдаги (neighborhood)* ёки ғолиб – нейрон ўрганиш радиусига (*radius of learning*) тааллуқли нейронлар ҳам ўрганишади. Ғолиб-нейрон атрофида жойлашган нейрон деб қабул қилиниши учун харитада жойлашуви бўйича ғолиб нейрон ва айнан шу нейрон ўртасидаги масофа маълум бир қийматдан кам бўлиши лозим (ўрганиш жараёнида нейронлар оғирлик кўрсаткичлари ўзгаради, бироқ харитадаги ўрни ўзгармас қолади). Бундай алгоритм *WTM (Winner Takes Most* – барчасини ғолиб олади) кўринишдаги алгоритм ҳисобланади. Оддий алгоритмда ғолиб-нейрон ва унинг оғирлик кўрсаткичи атрофида мавжуд барча нейронлар ўрганиш жараёнидан *Кохонен қоидаси бўйича* (адаптация) ўтади (2). Оғирлик кўрсаткичи ғолиб-нейрон оғирлиги чегарасидан ташқарида жойлашган нейронлар ўзгармас қолади. Оғирлик кўрсаткичидаги чекланмалар ва ўрганиш тезлиги коэффициенти функция билан белгиланиб, унинг қийматлари такрорланиш ошган сари уларнинг қиймати камайиб боради.

Кохонен қоидасидаги ўзгаришлар ўрганиш тезлиги коэффициенти $h_i^{(k)}$ икки қисмга бўлинади: яқин жойлашганлик функцияси $h_i(d, k)$ ва ўрганиш тезлиги функцияси $a(k)$

$$\eta_i^{(k)} = \eta_i(d, k) \cdot a(k), \quad (24)$$

яқин жойлашганлик функцияси сифатида ёки константа

$$\eta_i(d, k) = \begin{cases} \text{const}, & d_i \leq \sigma(k), \\ 0, & d_i > \sigma(k) \end{cases} \quad (25)$$

ёки Гаусс функцияси қўлланилади

$$\eta_i(d, k) = e^{\frac{d_i}{2\sigma(k)}} \quad (26)$$

бунда Гаусс функциясини қўллаш самарали натижа беради.

$$d_i = \left\| r_i - r_{c_j} \right\| \quad (27)$$

Ифодаларда нейрон ва ғолиб-нейрон ўртасидаги масофа, c_j , r_i ва r_{c_j} – i -ва ғолиб c_j -нейрон ўртасидаги харитадаги координаталар, $\left\| r_i - r_{c_j} \right\|$ харитада i - ва c_j катаклар ўртасидаги масофа. $\sigma(k)$ функцияси ўрганиш цикли бўйича камайиб кетувчи функция ҳисобланиб, кўпинча чизикли камаювчи функция қўлланилади.

Ўрганиш тезлиги функциясини $a(k)$ кўриб чиқамиз. Ушбу такрорлашлар сонига нисбатан камайиб бориш функциясини ҳам ифодалайди. Ушбу функциянинг 2 та варианты қўлланилади: чизикли ва ўрганиш жараёни такрорлашидан тескари пропорционалли кўриниши

$$a(k) = \frac{A}{k+B} \quad (28)$$

бунда A ва B – константалар. Бу функцияни қўллаш натижасида ўрганиш танловидаги барча векторлар ўрганиш жараёни самарадорлигига бир хил улуш қўшади.

Ўрганиш жараёни иккита босқичдан иборат: биринчи босқичда–хусусиятлар тўпламида оғирлик кўрсаткичлари коэффициентлари вектори кетма-кетлиги тўғриланади, яъни ўрганиш тезлиги ва радиуси қиймати юқори танланади ва натижада нейронлар вектори мисоллар кетма-кетлигига мос равишда қўйиб чиқади (нейронлар вектори ўрни харитада эмас, балки хусусиятлар тўплами соҳасида ўзгаради). Сўнгги босқич – *созлаш босқичида* ўрганиш тезлиги параметрлари ошиш ҳисобидан оғирлик кўрсаткичлари аниқ кетма-кетликда қўйилади. Ўрганиш жараёни тармоқ хатолик чегараси энг кичик қийматга етганча такрорланади.

Пакетли ўрганиш алгоритми

Ҳозирги кунга келиб, Кохонен харитасида ўрганиш жараёни учун Кохонен харитасининг пакетли ўрганиш алгоритми (*Batch-Learning*

SelfOrganizing Map) [3, 11] кенг қўлланилмоқда. Ушбу алгоритмда энг аввал барча мисоллар ифодаланади, сўнгра оғирлик кўрсаткичлари янгиланади. Алгоритмда нормаллаштирилган кириш векторлари қўлланилади. Алгоритм B та ўтиш сонига эга, ва k -ўтишда қуйидаги амаллар бажарилади.

1. Барча N та кириш вектори узатилади ва ҳар бир x_j нчи кириш вектори билан барча нейронлар оғирлик кўрсаткичлари векторлари w_i ўртасидаги Евклид масофа ҳисобланилади.

$$c_j = \operatorname{argmin} \left\{ \left\| r_i - r_{c_j} \right\| \right\} \quad (29)$$

2. Ғолиб-нейрон рақами аниқланади

Кириш векторининг барча компонентлари каби оғирлик кўрсаткичларининг барча векторлари янгиланади.

$$w_i^{\text{new}} = \frac{\sum_{j=1}^N h_{c_j,i} x_j}{\sum_{j=1}^N h_{c_j,i}} \quad (30)$$

бунда $h_{c_j,i}$, – кўрсаткич атрофи функцияси

$$h_{c_j,i} = \exp \left(\frac{\left\| r_i - r_{c_j} \right\|^2}{2\sigma^2(k)} \right) \quad (31)$$

r_i ва r_{c_j} – i - ва c_j -ғолиб-нейроннинг харитадаги координаталари, $\left\| r_i - r_{c_j} \right\|$ – харитада i -ва c_j -катаклар ўртасидаги масофа, $\sigma(k)$ параметр ўтишлар сони ошган сари камайиб кетади

$$\sigma(k) = \sigma_0 \left(1 - \frac{k}{k_{\max}} \right), \quad (32)$$

σ_0 – танланадиган бошланғич қиймат, k_{\max} – ўтишларнинг максимал сони.

Ўтиш жараёни такрорланиши, (3) функция натижаси етарли даражада камайгунча давом этади.

Нейрон газ алгоритми

Нейрон газ алгоритми [1, 12] бу - газ ҳаракати динамикасига ўхшашлиги сабабли шундай номланган бўлиб, бошқа алгоритмлардан фарқли равишда яқинлашиш тезлигини таъминлайди. Ушбу алгоритмда ўрганиш жараёнининг ҳар бир k -циклида барча нейронлар

Ҳолиб нейрондан узоқлик даражаси кетма-кетлигида саралаб чиқилади. Ҳар бир нейрон учун яқинлик функцияси қиймати аниқланади. i -нейрон учун ушбу функция қуйидаги кўринишга эга

$$\eta_i^{(k)} = e^{\frac{m_i}{\lambda^{(k)}}}, \quad (33)$$

бунда m_i –нейрон рақами бўлиб, саралаш натижаси белгиланади (Ҳолиб –нейрон учун бу рақам нолга тенг); l – параметр (кенглик параметри), кетма-кет ўрганиш алгоритмидаги параметрига ўхшаш бўлиб, итерация ошиши билан камайиб боради. i -нейрон оғирлик кўрсаткичи вектори тахрири қуйидаги формула бўйича амалга оширилади

$$w_i^{(k+1)} = w_i^{(k)} + \eta_i^{(k)} a_i^{(k)} [x^{(k)} - w_i^{(k)}], \quad (34)$$

бунда $\eta_i^{(k)}$ – яқинлик функцияси бўлиб, (6) формула бўйича аниқланади; $a_i^{(k)}$ – ўрганиш тезлиги функцияси.

Ҳолиб –нейрон оғирлик кўрсаткичи вектори WTA алгоритми бўйича аниқланади. Бироқ алгоритмдан фаркли равишда WTA барча нейронлар оғирлиги аниқланади.

$\lambda^{(k)}$ ва $\lambda_i^{(k)}$ параметрлари итерация сони ошган сари камайиши лозим

$$\lambda^{(k)} = \lambda_{max} \left(\frac{\lambda_{min}}{\lambda_{max}} \right)^{k/k_{max}}, \quad (35)$$

$$a_i^{(k)} = a_i^{(0)} \left(\frac{a_{min}}{a_i^{(0)}} \right)^{k/k_{max}} \quad (36)$$

Бунда k – жорий итерация рақами, k_{max} – итерациянинг максимал қиймати, λ_{min} ва λ_{max} – λ параметрнинг минимал ва максимал қиймати, $a_i^{(0)}$ – ўрганиш тезлигининг бошланғич қиймати, a_{min} – k_{max} га мос бўлган ўрганиш тезлигининг минимал қиймати бўлиб,

$m_i > K$ ҳолат учун ҳисоблаш ҳажмини камайтириш учун $h_i^{(k)} = 0$ деб олинади. Бунда K – берилади. Яъни нейронларнинг K кўрсаткичи соланади.

Барча кўриб ўтилган алгоритмлар нейронлар сонини талаб этади.

Нейрон газ кенгайиш алгоритми (Growing Neural Gas) [13] кириш қисмидаги берилганлар кластерлашлаш имконини беради ва тармоқ ўрганиш жараёнида нейронлар сонини ҳисоблайди.

Кохонен харитаси инициаллаштириш

Кохонен харитасини тузишдан олдин, нейронлар тармоғи конфигурацияси (одатда, олтибурчак) ифодаланилиши ва харита учун нейронлар сони берилиши лозим. Нейронлар сони харита аниқлик даражасини белгилайди. Бунда кўп сонли нейронлардан иборат харита ўрганиш жараёни учун кўп вақтни талаб этади.

Бундан кейин *инициализация амалга оширилади* – яъни нейронлар оғирлик кўрсаткичининг бошланғич қийматлари ўзлаштирилади. Содда ҳоллар учун оғирлик кўрсаткичини тасодифий кичик сонлар билан инициаллаштириш мумкин. Маълумки [3], ўрганиш жараёнига сарфланувчи вақтни тежаш учун хаританинг бошланғич ҳолатини тартиблаб чиқиш жуда катта самара беради. Бошланғич оғирлик кўрсаткичи сифатида ўрганиш мисоллари кетма-кетлигидан тасодифий танланган қийматларни қўллаш яхши натижа беради. Бу ерда *чизиқли инициаллаштириш* [3] самарали бўлиб, харитада оғирлик кўрсаткичи бошланғич ҳолатларини тартибланиш имконини беради. Бунинг учун кириш векторларини марказлаштирилган ҳолатга ўзгартириб, нолли математик кутилиш векторлар кўринишда матрица $\dot{X} = [\dot{x}_1 \dot{x}_2 \dots \dot{x}_n]$ яратилади. Бунда марказлаштирилган кириш векторларидан тузилган ҳолда коварицион матрица ҳисобланади (37).

$$K = \frac{1}{n-1} \dot{X} \dot{X}^T \quad (37)$$

К матрицанинг иккита энг катта қиймати мавжуд (бу қийматлар мусбат, яъни матрица мусбат аниқланган). Тўрт ва олти бурчакли графика тўғри бурчак асосда қурилади, уларнинг ўлчами К матрицанинг икки энг катта қийматига тенг. Нейронлар оғирлик кўрсаткичининг бошланғич қиймати К матрица энг катта икки қиймати векторларининг чизиқли комбинацияси кўринишда яратилади. Оғирлик кўрсаткичининг чизиқли комбинацияси қурилган тўртбурчакда жойлашган координатасига тенг бўлади. Математик

нуқтаи назардан оғирлик кўрсаткичининг бошланғич қийматлари К матрицанинг иккита вектори асосида муҳитдан олинади.

Оғирлик кўрсаткичи векторлари бошланғич қийматлари тартибланади ва оғирлик кўрсаткичи харитаси бўйича яқинлаштиришни енгиллаштиради. Шу сабабли, созлаш босқичидан бошлаб, ўрганиш жараёнини бошлаш мумкин. Бироқ, чизиқли белгилаш жараёни “бўш”нейронларни, яъни кириш векторига умуман яқин бўлмаган нейронларни яратиши мумкин. Оғирлик кўрсаткичининг энг яхши бошланғич кўрсаткичлари асосий компонентлар чизиқсиз усулини қўллаш имконини беради.

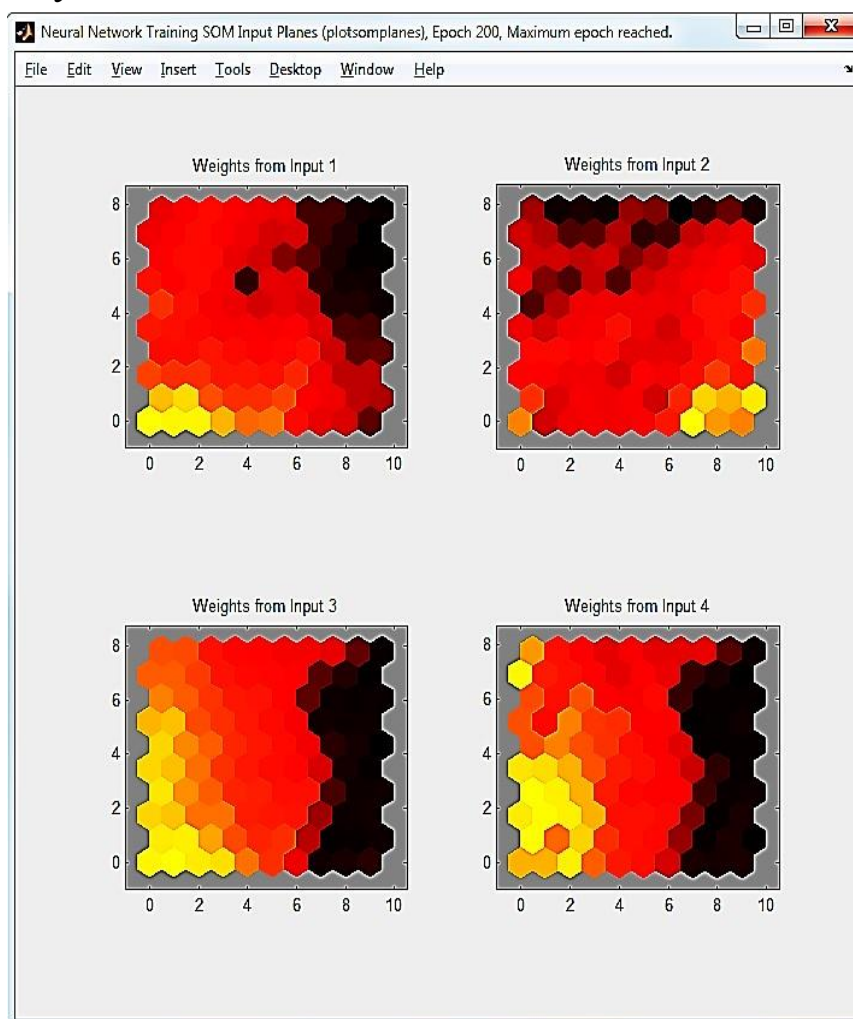
Кохонен харитаси инициаллаштиришда кластерлаш алгоритмларидан бири қўлланилиши мумкин, масалан, *k-means*: кластерлаш алгоритми ёрдамида харитада мавжуд нейронлар сони тенг равишда кластерлар шакллантирилади. Сўнгра хаританинг аниқ созлаш ишлари амалга оширилади.

Инициализациянинг мураккаб усули тармоқ ўрганиш вақтини қисқартиради, бироқ инициализациялаштириш вақтини оширади. Шунинг учун харитани яратишда сарфланадиган тўлиқ вақтини инобатга олиш лозим.

1.31.4. Кохонен харитасининг график кўриниши

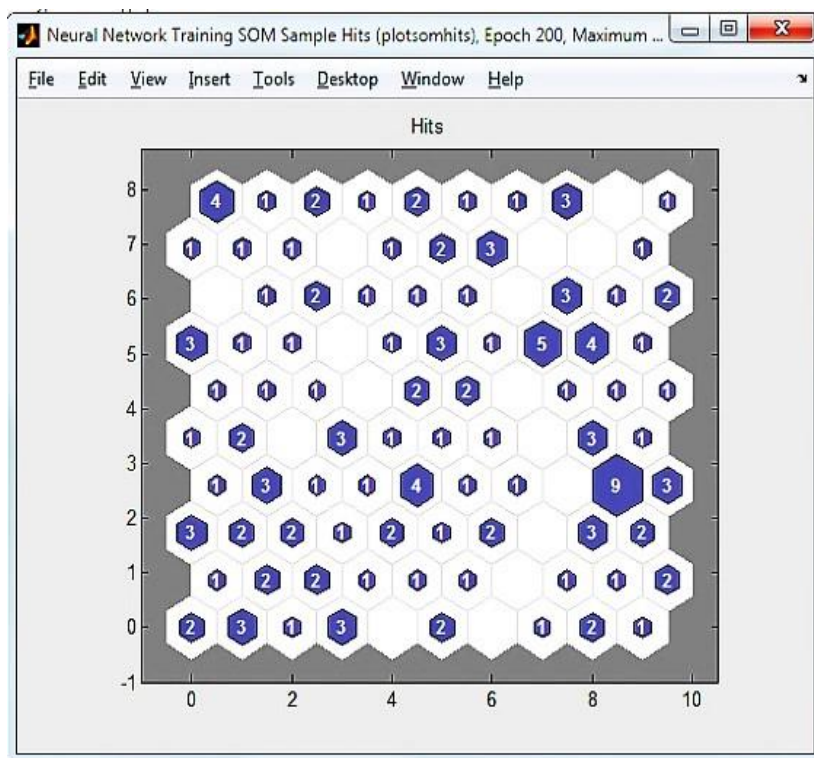
Хусусиятлари вектори яқин бўлган объектлар битта катакка ёки харитада жуда яқин жойлашган катакларга тушишади. Шунинг учун Кохонен икки ўлчовли харитасида кўп ўлчовли хусусиятлар вектори жойлашувини ифодалайди. Одатда, объектлар аниқ қайси бир параметрлар бўйича ўхшашликка эга эканлигини таҳлил этиш талаб этилса, Кохонен харитаси рангланиши қўлланилади. Бунинг учун таҳлил қилинаётган параметрлар (кириш векторлари компонентлари) сонига қараб хариталар яратилади. Ҳар бир харита объектнинг битта параметрига мос келади. Харита катаклари катакдаги нейрон оғирлик кўрсаткичлари қийматларига мос равишда турли рангларга (ёки кулранг кўринишларига) бўянади. Оғирлик кўрсаткичлари қийматлари диапазони ажратилади. Ҳар бир диапазонга маълум бир ранг (кулранг кўриниши) мос келади ва харита катаклари шунга мос

бўлган ранг билан “бўянади”. Оғирлик кўрсаткичи графикасига кўриниши компонентли текислик– *Component Planes* деб номланиб, Расм. 63 (MATLAB Neural Network Toolbox да графика ҳосил қилинган) да келтирилган. Графикада тўқ рангдагилар бўянганлар бу оғирлик кўрсаткичи юқори бўлганлар ҳисобланади. Оғирлик кўрсаткичи яқин бўлганлар бир хил рангда бўянган. Расмдан кўриниб турибдики, ўрганилаётган объектлар биринчи ва иккинчи кириш қисми бўйича кескин фарқланади, бироқ учинчи ва тўртинчи кириш қисми бўйича жуда яқин.



63-расм. Текисликдаги компонентлар.

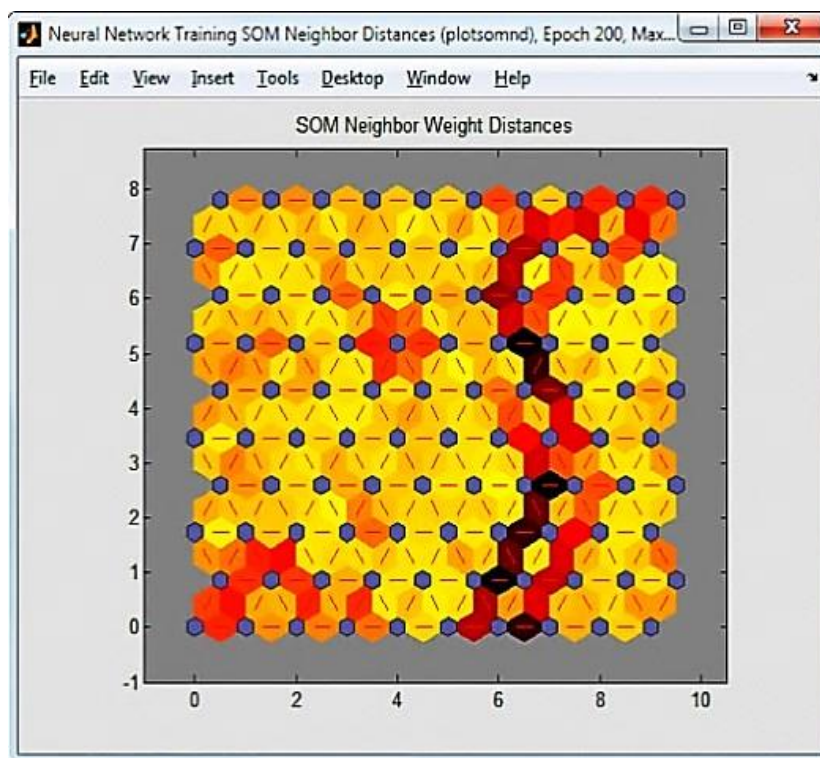
Таҳлил қилишда, хаританинг ҳар бир катаги (нейрони) билан кириш қисмининг нечта векторлари боғланганлигига эътибор бериш лозим. Бунинг учун MATLAB Neural Network Toolbox дастурида *кластерларда мисолларнинг жойлашув графиги* қурилади (Расм.64).



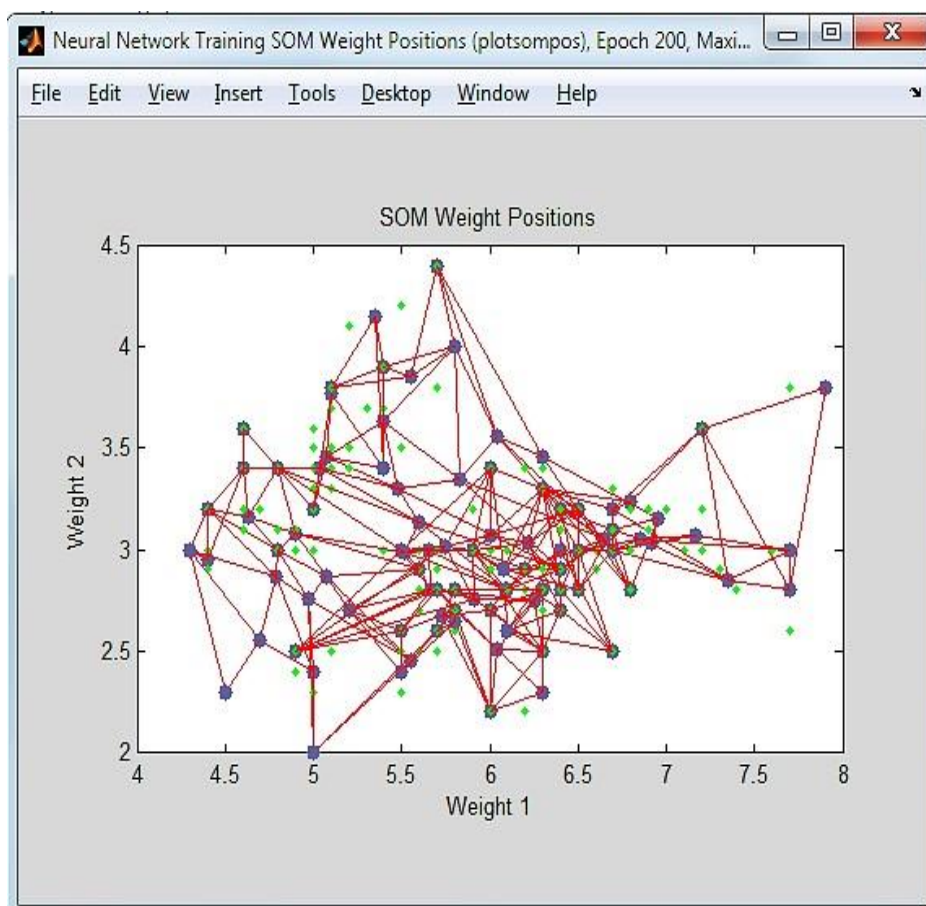
64-расм. Кластерларда мисолларнинг жойлашув графиги.

Шу билан бирга яқин жойлашган нейронлар оғирлик кўрсаткичи векторлари орасидаги масофани аниқлаш ҳам тавсия этилади. Бунинг учун *ягоналаштирилган масофа матрицаси U-matrix* ва Семмона (*J. W. Sammon*) ифодаси қўлланилади. [17]. *ягоналаштирилган масофа матрицаси – U-matrix* [18] элементлари нейрон оғирлик кўрсаткичи ва яқин жойлашган нейронлар оғирлик кўрсаткичлари ўртасидаги фаркни ифодалайди. Бу қийматнинг катталиги нейрон атрофдагилардан кескин фарқланишини ва бошқа кластерга боғлиқлигини билдиради (Расм 65). MATLAB Neural Network Toolbox дастурида қурилган масофа матрицаси кўрсатилган. Олтибурчак нейронларни ифодалайди. Чизиклар эса атрофдаги нейронлар билан боғланишни ифодалайди. Боғланиш қисмлардаги ранглар нейронлар ўртасидаги масофани кўрсатади. Тўқ ранглар масофа катталигини билдиради. Тўқ рангли сегментлар хаританинг пастки қисмдан юқори ўнг қисмга қараб тақсимланган. Харитада объектлар иккита кластерга гуруҳланган.

Семмон проекцияси [19] векторлар ўртасидаги масофани сақлаган ҳолда кириш қисмидаги векторларнинг текисликдаги чизиксиз ифодаси. Одатда ўзаро яқин нейронлар чизиклар орқали боғланган.



65-расм. Масофанинг бирлашган матрицаси.



66-расм. Оғирлик кўрсаткичларининг жойлашуви.

MATLAB Neural Network Toolbox дастурида *оғирлик кўрсаткичининг жойлашув графиги* курилади (Расм. 66), унда кириш қисми векторлари координаталари биринчи икки компонент орқали аниқланиб нукталар орқали ифодаланган. Нейронлар оғирлик кўрсаткичлари векторлари биринчи икки компонент координаталари нукталари бошқа рангда ифодаланган бўлади. Қўшни нейронлар чизиклар орқали боғланган. Шунини таъкидлаш лозимки, яқин жойлашган нейрон деб катакларда яқин нейронлар эмас балки хусусиятлар соҳасида яқин жойлашган нейронлар ҳисобланади.

1.32. Ҳўқитувчи иштирокида ўрганувчи вектор квантлаш тармоқлари (LVQ-сети)

LVQ-тармоқлари вектор квантлашни ўрганиш жараёнига йўналтирилган (*LVQ – Learning Vector Quantization*) [3] бўлиб, ўқитувчи иштирокида ўрганувчи Кохонен қатламини ташкил этади. LVQ-тармоқни куриш учун n кластерлар (нейронлар) сони, синфланишлар сони m ($n \neq m$) ва ҳар бир кластер маълум бир синфга мансублиги белгиланиши лозим. Масалан, беморлар анализлари 5 кластерга бўлинган бўлиб, улардан 2 таси соғлом инсонларга, қолган 3 таси касал инсонларга тааллуқли. Кластерларни синфлар бўлиш пропорцияси, ўрганилаётган ҳолатларни мос равишда синфларга бўлиш пропорцияси бўйича амалга ошириш мумкин. Кластерлар ва синфлар рақамларини тасодифий сонлар билан ифодалаш мумкин. Бироқ тармоқ иши натижасини қулай ифодалаш учун кластерларни кетма-кет тартибда тавсия этилади. Кўрилаётган мисолда биринчи икки кластер соғлом инсонлар синфига тўғри келади, кейинги 3 таси эса - касал беморлар синфига. Ўрганиш жараёнида LVQ-тармоқ нейронлар оғирлик кўрсаткичини белгилашда ўрганилаётган ҳолатлар ва кластерлар битта синфга мансуб деб олинади. Ўргатилган LVQ-тармоқ кириш қисмидаги векторларни синфларни инобатга олган ҳолда кластерлайди. Масалан, бир беморнинг аниқ бир анализи касал инсонлар синфига тааллуқли кластерларнинг бирига боғлиқ. Ушбу

кластер таркибига кирган анализлар хусусиятларини фақат шу анализларни ўрганиш натижасида аниқлаб бўлади.

LVQ-тармоқнинг бир нечта ўрганиш алгоритмлари маълум бўлиб[3], энг содда *LVQ1* ўрганиш алгоритми, k -ўрганиш циклида қуйидаги кўринишда бўлади.

1. Ўрганиш танловининг \mathbf{x} вектор учун c рақамли нейрон мавжуд бўлиб, унинг оғирлик кўрсаткичи векторлари ва \mathbf{x} ўртасидаги евклид масофа $w_c^{(k)}$ энг минимал.

2. Голиб-нейрон оғирлик кўрсаткичи вектори қуйидагича таҳрирланади:

– агар $w_c^{(k)}$ ва \mathbf{x} битта синфга мансуб бўлса:

$$m_{k+1}^c = m_k^c + a_{(k)}[x - m_k^c] \quad (38)$$

бунда $a^{(k)}$ – ўрганиш тезлиги коэффициенти.

Бошқа нейронлар оғирлик кўрсаткичи ўзгармас қолади.

3. Кейинги ўрганилаётган вектор танланади ва 1-қадамдан бошлаб такрорланади.

4. таснифланган векторлар сони ошиши тугагунча 1–3 қадамлар такрорланади.

Агар кириш қисмидаги вектор тармоқда тўғри таснифланса, унда мос равишда оғирлик векторлари кириш қисмидаги векторлар томон силжийди. Агар кириш қисмидаги вектор тармоқда нотўғри таснифланса, унда мос равишда оғирлик векторлари кириш қисмидаги векторларга қарама-қарши томонга силжийди.

Ўрганиш тезлиги коэффициенти $0 < a^{(k)} < 1$ ўрганиш цикли ошган сари камайиб бориши лозим. Аммо, $a^{(k)}$ нинг бошланғич қиймати етарлича кичик бўлиши, масалан, 0,1 лозим.

LVQ2.1 алгоритмида *LVQ1* алгоритми қондаси бўйича бир вақтнинг ўзида кириш қисми вектори \mathbf{x} га яқин иккита оғирлик кўрсаткичи векторлари w_i ва w таҳрирланади.

Бунда векторлардан бири тўғри синфга , иккинчиси -нотўғри синфга мансуб. w_i ва w векторларнинг кириш қисми вектори \mathbf{x} га яқинлик омили s нинг s кенглик соҳасига тушиши ҳисобланади. Агар

d_i ва d_j – x дан w_i ва w гача бўлган Евклид масофа бўлиб, x соҳага тушиши учун,

$$\min \left(\frac{d_i}{d_j}, \frac{d_j}{d_i} \right) > \frac{1-s}{1+s} \quad (39)$$

Кенглик диапазонини 0,2 дан 0,3 гача белгилаш тавсия этилади.

LVQ-тармоқларнинг [3] ва бошқа ўрганиш алгоритмлари маълум бўлиб, оғирлик кўрсаткичи бошланғич қийматлари сифатида синфларга мос равишда ўрганилаётган ҳолатлар тасодифий векторларини қўллаш мумкин. *LVQ*-тармоқларда ўрганиш давомида қайта ўрганиш имкони мавжуд. Кўп қатламли перцептрон каби қайта ўрганиш кўпгина ҳолларда натижани текшириш ва қайта ўрганиш амаллари кетма-кетлиги тартибида олиб борилади.

Катта ҳажмдаги кластерлар ҳолати учун *LVQ*-тармоқ иши натижасини ифодалаш мураккаб ҳисобланади. Чунки энг аввал кириш қисмидаги вектор мансуб бўлган синф аҳамиятли ҳисобланади. Икки қатламдан иборат *LVQ*-тармоқ қўлланилиши [20] да келтирилган: биринчи қатлам Кохонен қатлами ҳисобланиб, кириш қисми векторларини берилган n та кластерга кластерлайди. Иккинчи чизиқли қатлам ушбу кластерларни битта m синфга бирлаштиради ($n \nrightarrow m$). Натижада тармоқ кириш қисми векторларини таснифлайди. Бу ёндашув MATLAB тизимининг Neural Networks Toolbox [9] воситасида жорий этилган. Кириш қисми қатлами *LVQ*-тармоқ ўрганиш алгоритми бўйича ўрганади (Neural Networks Toolbox тизимида *LVQ1* ва *LVQ2.1* алгоритмлари жорий этилган). Чизиқли қатлам ўрганиш жараёнини ўтмайди, балки маълум бўлган кластерлар ва синфлар орасидаги боғланишлар асосида шаклланади. Биринчи қатлам n та элементдан иборат векторни узатади, бунда фақат битта элемент бирга тенг, қолганлари нолга тенг. Бирга тенг бўлган элемент вектордаги рақами кириш вектори мансуб бўлган кластер рақамига тенг бўлади. Чиқиш қисмидаги чизиқли қатлам оғирлик кўрсаткичлари матрицаси m та қатори мавжуд.

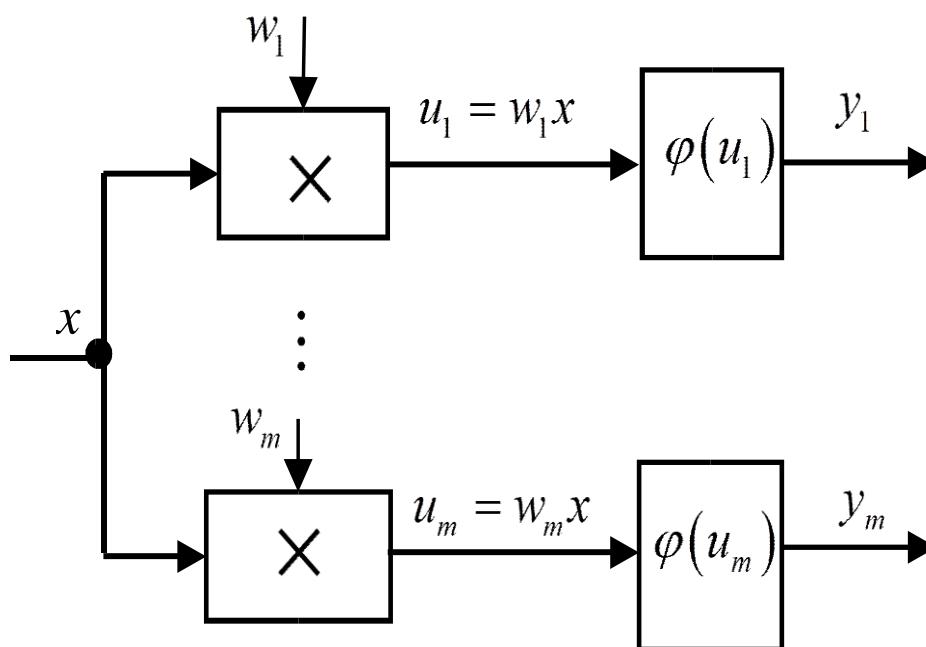
Матрица қатори синфларга тўғри келади. Устунлар сони эса n га тенг. Устунлар кластерларга тўғри келади. Ҳар бир устунда фақат

битга элемент бирга тенг бўлади. Бу элемент кластер мансуб бўлган синфни кўрсатади. Чизиқли қатлам оғирлик кўрсаткичлари матричасининг биринчи қатлами чиқиш векторига кўпайтмаси n элементдан иборат векторни шакллантиради. Бирга тенг элемент рақами идентификацияланган синф рақамига тенг.

1.33. Қарши тақсимланган тармоқ

Қарши тақсимланган тармоқ (*Counterpropagation Network*) [21–22] Р. Хехт-Нильсеном (R. Hecht-Nielson) [23] томонидан таклиф этилган бўлиб, икки қатламли тармоқни ташкил этади, биринчи қатлам бу Кохонен қатлами, иккинчиси –С. Гроссберг (S. Grossberg) қатлам [24]. Қарши тақсимланган тармоқ аниқлик кўрсаткичи бўйича кўп қатламли персептронга етмасада, бироқ ўрганиш жараёни жуда тез амалга оширилиб, бир қатор самарали хусусиятга эга. Қарши тақсимланган тармоқлар тартибга солинган векторларни қўллайди.

Гроссберг қатлам нейронлари *Гроссберг чиқиш юлдузлари (Outstar)* деб номланади. Гроссберг чиқиш юлдузлари (Расм.67) скаляр кириш қисмига ва векторли чиқиш қисмига эга.



67-расм. Гроссберг чиқиш юлдузи.

Гроссберг чиқиш юлдузи фаоллашув функцияси ёрдамида қайта ўзгартиришни [22] амалга оширади.

$$y_i = \varphi(w_i x), \quad i = 1, 2, \dots, m$$

$$\varphi(u_i) = \begin{cases} u_i, & \text{агар } -1 \leq u_i \leq 1, \\ 1, & \text{агар } 1 < u_i, \\ -1, & \text{агар } u_i < -1. \end{cases} \quad (40)$$

Қарши тақсимланган тармоқ ўрганиш жараёни иккита қадамдан иборат.

Биринчи қадамда юқорида кўриб ўтилган алгоритмлар бўйича Кохонен қатлами ўрганишни бошлайди. Ўрганилган Кохонен қатлами нолдан фарқланган битта компонентга эга векторни узатади.

Иккинчи қадамда Гроссберг қатламининг ўқитувчи ёрдамида ўрганиш амалга оширилади. Гроссберг қатламида оғирлик кўрсаткичини тахрирлаш қуйидаги формула бўйича амалга оширилади

$$w_{ij}^{(k+1)} = w_{ij}^{(k)} + \beta (y_j - w_{ij}^{(k)}) u_{ki} \quad (41)$$

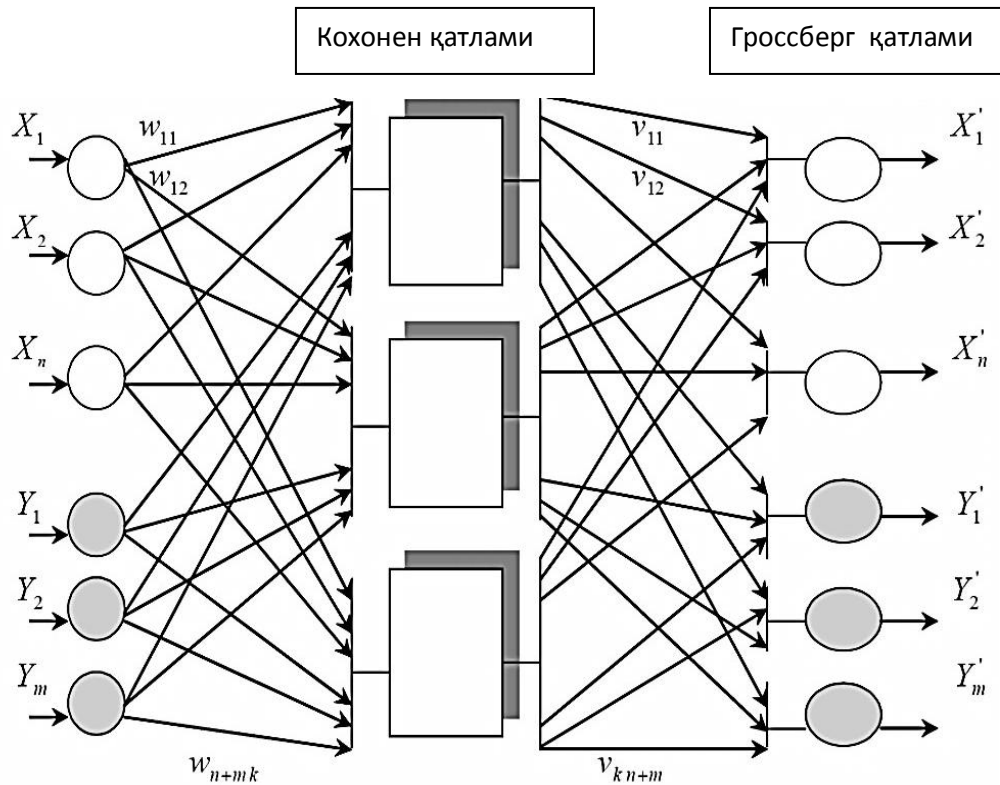
бунда k –ўрганиш цикли рақами, w_{ij} – j нейроннинг i -оғирлик кўрсаткичи, u_{ki} – Кохонен қатламининг i -нейрон чиқиши (фақат битта Кохонен нейрони нолдан фарқли бўлади, y_j – керакли чиқиш векторларининг j -компоненти, β – ўрганиш тезлиги коэффициенти (энг аввал [21] β қийматини $\sim 0,1$ га тенг деб олиш сўнгра ўрганиш давомида камайтириш тавсия этилади,).

Кохонен қатлами чиқиш қисмидаги вектор ғолиб –нейроннинг битта ягона компонентига эга бўлганлиги сабабли, Гроссберг қатлами нейронлари ва Кохонен қатлами ғолиб-нейрон боғловчи чиқиш қатлам оғирлик кўрсаткичлари мувофиқлаштирилади.

Агар ўрганиш жараёнидан ўтган тармоқга ўрганилаётган танловга мансуб бўлмаган вектор узатилса, энг аввал Кохонен қатлами мансуб кластер аниқланади, сўнгра Кохонен қатлам ғолиб-нейрон чиқиш қисмига сигнал узатилади. Унинг чиқиш қисмида чиқиш вектори шаклланиб, унинг элементлари қиймати аниқланган кластер марказ координатасига тенг бўлади.

Кўриб ўтилган тармоқ бир йўналишли текис тақсимланган тармоқ ҳисобланади. Икки йўналишли тескари тақсимланган тармоқларда кириш қисмидаги векторлар бўйича мос тушувчи кириш векторини аниқлаш имконини беради. Айнан шу хусусият сабабли “тескари

тақсимланган” ибораси қўлланилади. Икки йўналишли тескари тақсимланган тармоқ структураси 68-расмда келтирилган.



68-расм. Икки йўналишли тескари тақсимланган тармоқ

Ўрганиш жараёнида x ва y векторлар кириш қисми векторлари каби ва мақсадли чиқиш векторлари каби қўлланилади. x векторлари чиқиш қатламининг чиқиш векторлар x' ни ўрганиш учун қўлланилса, y векторлари эса чиқиш векторлари y' ни ўрганишда қўлланилади. Тармоқнинг ўрганиш жараёни бир йўналишли тескари тақсимланган тармоқ ўрганиш жараёни каби амалга оширилади. Фарқи фақат векторлар ўлчамининг ошганлигидир. Агар ўрганиш даврини ўтаган тармоқ кириш қисмига ўрганиш жараёнида қатнашмаган x вектор ва унга мос бўлган чиқиш вектори y узатилса, чиқиш қисмида x' ва y' векторлари аппроксимацияси олинади. Аммо бундай ечим аҳамиятли маънога эга эмас: чунки маълум бир ифоданинг кириш ва чиқиш векторлари аниқ берилган. Агар кириш қисмига фақат x узатилса

чиқиш қисмида $x\epsilon$ ва $u\epsilon$ аппроксимациялари олинади. Яъни x , u да ифодаланилади. Агар u вектор аниқ бўлса, фақат u кириш қисмига узатилса, чиқиш қисмида $x\epsilon$ ва $u\epsilon$ ни оламиз. $x\epsilon$ га эга бўлиш бу тескари бу u ни x да ифодалаш эканлигини билдиради.

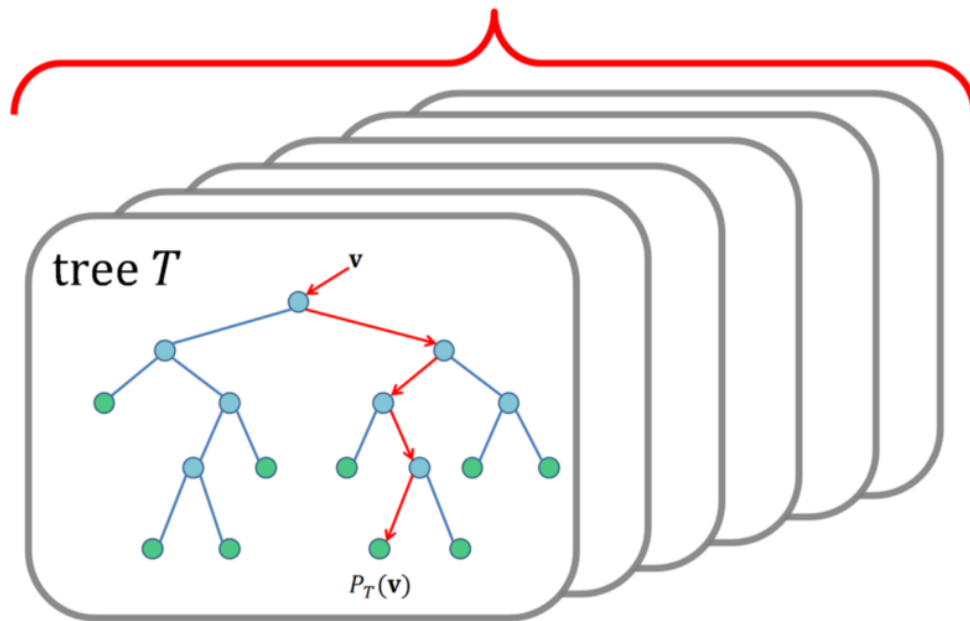
Биринчи боб бўйича хулоса

Боб сунъий нейрон тармоқлари назарий асосларига бағишланган бўлиб, унда сунъий интеллектнинг ривожланиш тарихи, босқичлари ва йўналишлари келтирилган. Билимларга асосланган тизимларни ишлаб чиқиш ва уларни ифодалаш, табиий тилда ва машинали таржима тизимни ишлаб чиқиш, тасвирларнинг аниқлаш модель ва усулларининг таҳлил ва қўлланилиши орқали қўллаш масалаларини ҳал қилиш амалга оширилган. Компьютернинг янги архитектураси, интеллектуал роботлар, махсус дастурий таъминот, ўрганувчи ва мустақил ўрганувчи сунъий нейрон тармоғи, нейрон тармоқларни қўллаш соҳалари, образларни аниқлаш. таснифлаш масалаларни амалиётга тадбиғига бағишланган. Нейрон тармоқ таснифи, тўғри чизиқли нейрон тармоқлари, марков занжири, Хопфилд Нейрон тармоқ, Хэмминг Нейрон тармоқлари, Больцман (ВМ) машинаси, Автоэнкодерлар, Сийрак автоэнкодер, Вариацион автоэнкодер архитектураси, CNN архитектураси тўлиқ изоҳи, CNN қўлланиладиган катламлар каби масалаларни амалиётга тадбиғи ва улардан фойдаланиш усуллари илмий ва амалий жиҳатдан ёритиб, шахсий фикр мулоҳазалар келтирилган.

II БОБ. СУНЪИЙ НЕЙРОН ТАРМОҚЛАРИ АЛГОРИТМЛАРИ ВА ФОЙДАЛАНИШ УСЛУБЛАРИ

2.1.Ечимлар дарахти алгоритмлари

Decision Forest



69-расм. Дарахтсимон ечимлар алгоритми.

Итератив дихотомизатор 3 (ID3) –бу юқоридан пастга қараб дарахтсимон структурани ҳосил қилади. Бунда илдиздан бошланиб, ҳар бир тугунда текширилиши лозим бўлган атрибут танланади. Ҳар бир атрибут маълум бир статистик восита ёрдамида баҳоланади, бу атрибут асосида берилганларни самарали тақсимлаш амалга оширилади. Танланган атрибут илдиз ҳисобланиб, унинг қийматлари тармоқланади, сўнгра жараён бошқа атрибутлар асосида давом этади. Атрибут танлангандан сўнг орқага қайтиш имкони бўлмайди.

С4.5 ва С5.0 (ёндашувнинг икки кўриниши) - С4.5, кейинги итерация Quinlan – бу ID3нинг энг янги версияси. Янги функциялар (ID3га нисбатан): (i) ҳам узлуксиз ҳам дискрет функцияларни қабул қилади; (ii) тўлиқ бўлмаган берилганлар тугунини қайта ишлайди; (iii) яқинлаштириш аниқлиги тушиши билан боғлиқ муаммоларни юқоридан паст усули асосида, одатда “қирқим” номига эга усул ҳал этади; ва (iv) турли оғирлик кўрсаткичлари хусусияти кўринишда

ўрганиш жараёни қийматлари қўлланилиши мумкин. C5.0, Quinlan энг сўнгги итерацияси ҳисобланади.

Тасниф ва регрессия дарахти (CART) –CART одатда “ечимлар дарахти” иборасининг аббревиатураси сифатида қўлланилади. Умумий ҳолда CART нинг реализацияси юқорида келтирилган C4.5 реализациясига ўхшаш. Битта фарқли томони бу CART қийматли тақсимлашга асосланган дарахтни яратади, бунда берилганларга рекурсив ҳолатда қўлланилади, ўз навбатида C4.5 қоидалар тўпламини яратиш босқичига эга.

Chi-квадрат ўзаро боғланишни автоматик аниқлаш (CHAID) - натижаларни ва башорат қилинган қийматларни категория бўйича ўзаро боғланишини аниқлаш алгоритми. Оптимал ечимга етмагунга қадар, барча башорат қилинувчи қийматлар ўртасида категориялар бўйича кесишмали жадвалларни яратади. Оптимал ечимга етгандан сўнг қайта тақсимлаш амали тўхтатилади. CHAID башорат моделини ёки дарахтини яратади, унинг ёрдамида берилганларни умумлаштириш, улар ўртасидаги боғланишларни изоҳлаш имконини беради. CHAID ни таҳлил қилганда номинал, тартибланган ва узлуксиз берилганларни қўллаш мумкин, бунда чексиз башорат қилинувчи қийматлар кузатувчилар сонига мос равишда категорияларга бўлинади. Категорияга эга катта ҳажмдаги берилганлар тўпламида шаблонларни аниқлашда самарали бўлиб, берилганларни умумлаштиришнинг қулай усули ҳисобланади, чунки берилганлар ўртасидаги муносабатлар осон визуаллаштирилади.

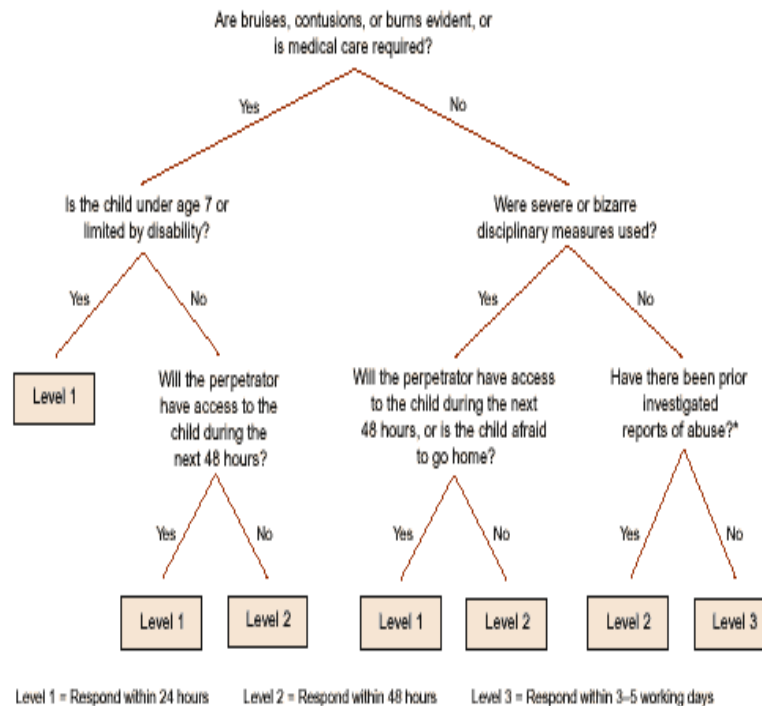
Decision Stump - ML модели бир даражали ечим дарахтидан иборат бўлиб, битта ички тугун (илдиз)га эга ва у сўнгги тугунлар (япроқлар) билан боғланган. Бу модел битта кириш функцияси қийматлари асосида башорат ўтказади.

M5- M5 ечимлар дарахти бўлиб, тугунларда чизиқли регрессия функцияси имконияти мавжуд. Аниқлик кўрсаткичи билан бирга катта ҳажмдаги юзга яқин атрибутлар асосида масалаларни ечиш имконига эга. M5 дарахт модели –регрессия масалаларни ўрганиш учун ечимлар дарахти бўлиб, Y ўзгарувчи қийматларини башорат этиш учун

қўлланилади. М5 дарахти CART дарахтида қўлланиладиган ёндашувни қўлласада, ўрта квадратик хатоликни функция кўринишда танлаганда, сўнгги тугун учун константани ўзлаштирмайди, балки чизикли регрессия кўп ўлчовли моделига тўғри келади.

Демак, ечимлар дарахти – оддий детерминистик структураси бўлиб, аниқ таснифлаш масала бўйича қарор қабул қилиш қоидасини моделлаштиришда қўлланилади. Ҳар бир тугунда битта хусусият танланиб, айнан ўша бўйича қарор қабул қилинади. Тақсимланиш тугаши учун энг сўнгги тугунда нуқталар сийракланиши лозим. Бундай япроқли тугун натижани ифодалайди (турли синфлар бўйича тасниф эҳтимоллигидир).

Қуйидаги чизма тўлиқ изоҳ беради:

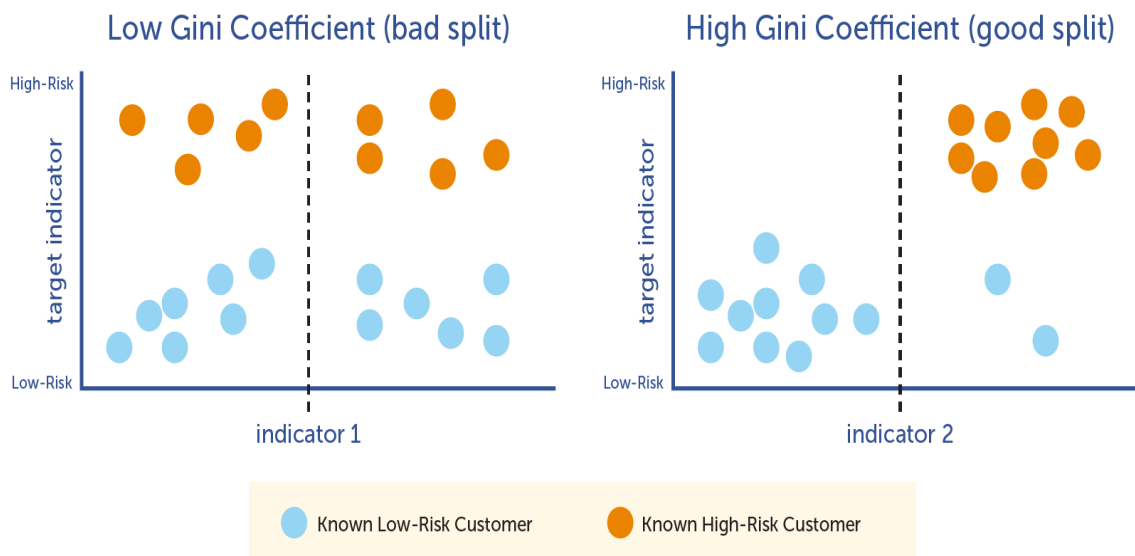


70-расм. Турли синфлар бўйича тасниф эҳтимоллиги

Қуйидагича амалга оширилади

Ечимлар дарахти самарадорлигида ҳал қилувчи омил бу самарали тармоқлаш жараёнидир. Биз ҳар бир тугунни тақсимлаш максимал аниқ бўлиши лозим. Аниқлик деганда синфларга тўғри бўлиш ва у асосда билимларни кенгайтириш унумдорлигини билдиради.

Тасвирда 1000 та сўз тақсимот ифодаси келтирилган ва унда маълум бир интуиция келтирилган:



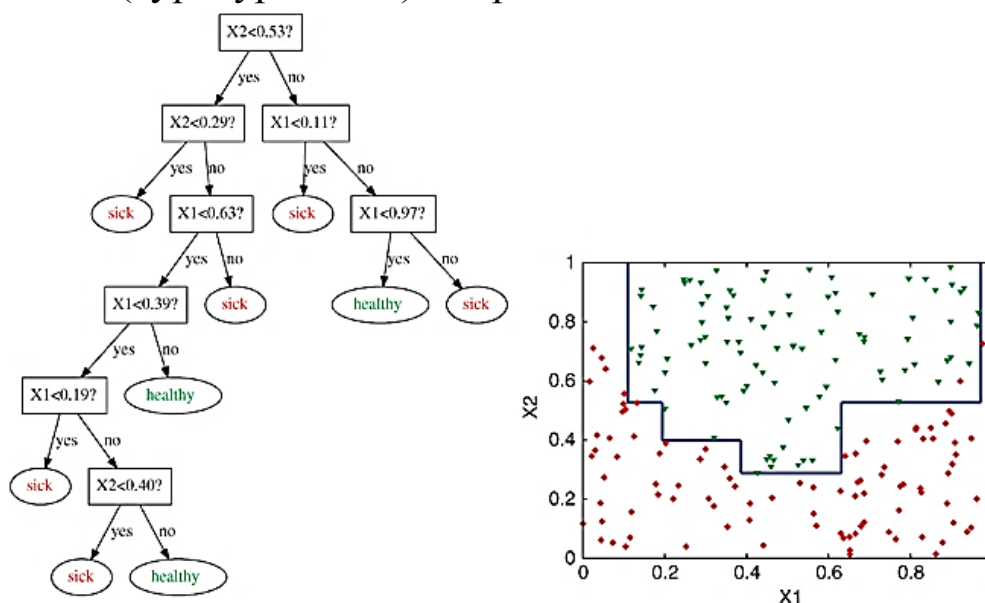
71-расм. 1000 та сўз тақсимот ифодаси.

Иккита кенг тарқалган тақсимот усулида келтирилган:

1. Джин аралашмаси
2. Маълумотнинг олиниши

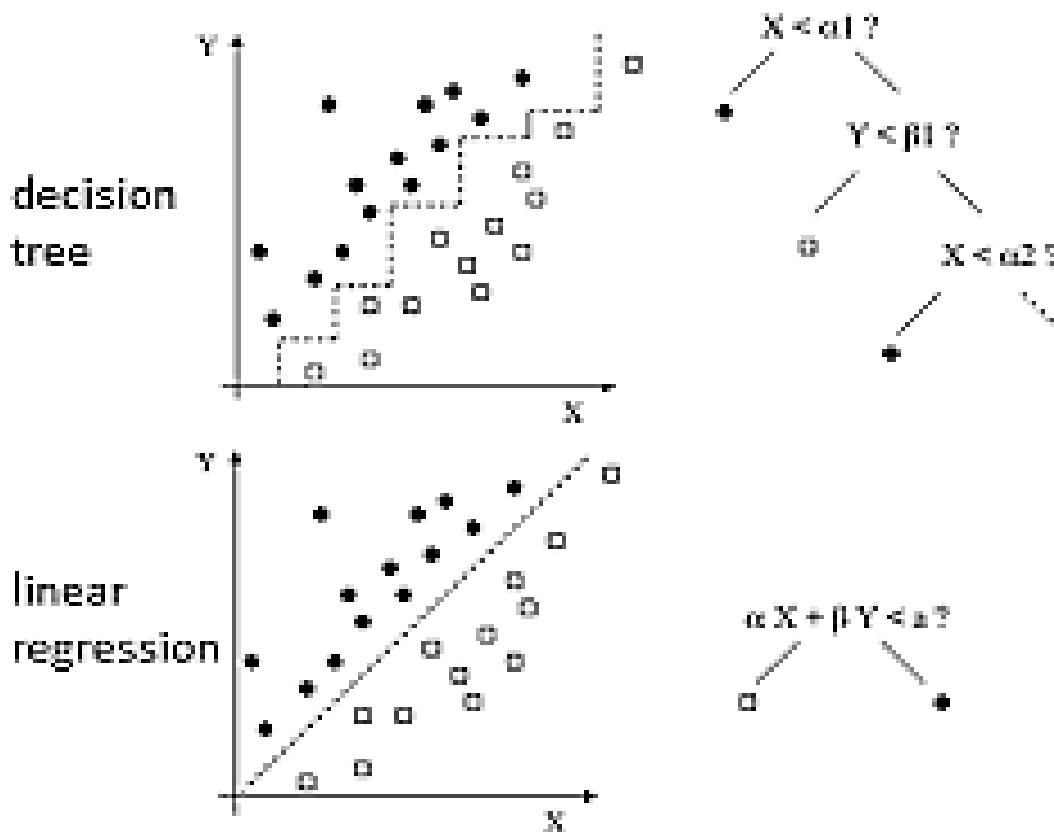
Визуаллаштириш:

Ҳар бир тақсимот тўғри чизикга олиб келади ва берилганлар тўпламини икки қисмга таснифлайди. Демак, якуний қарор чегараси тўғри чизикдан (тўртбурчакдан) иборат.



72-расм. Ечимлар дарахти асосида берилганлар тўпламини икки қисмга таснифлаш

- Регрессияга нисбатан ечимлар дарахти дарахтлар таснифи чегараларини бирлаштирганда зинама-зина кўринишга олиб келди.



73-расм. Дарахтлар таснифи чегараларини бирлаштирганда зинама-зина кўриниш

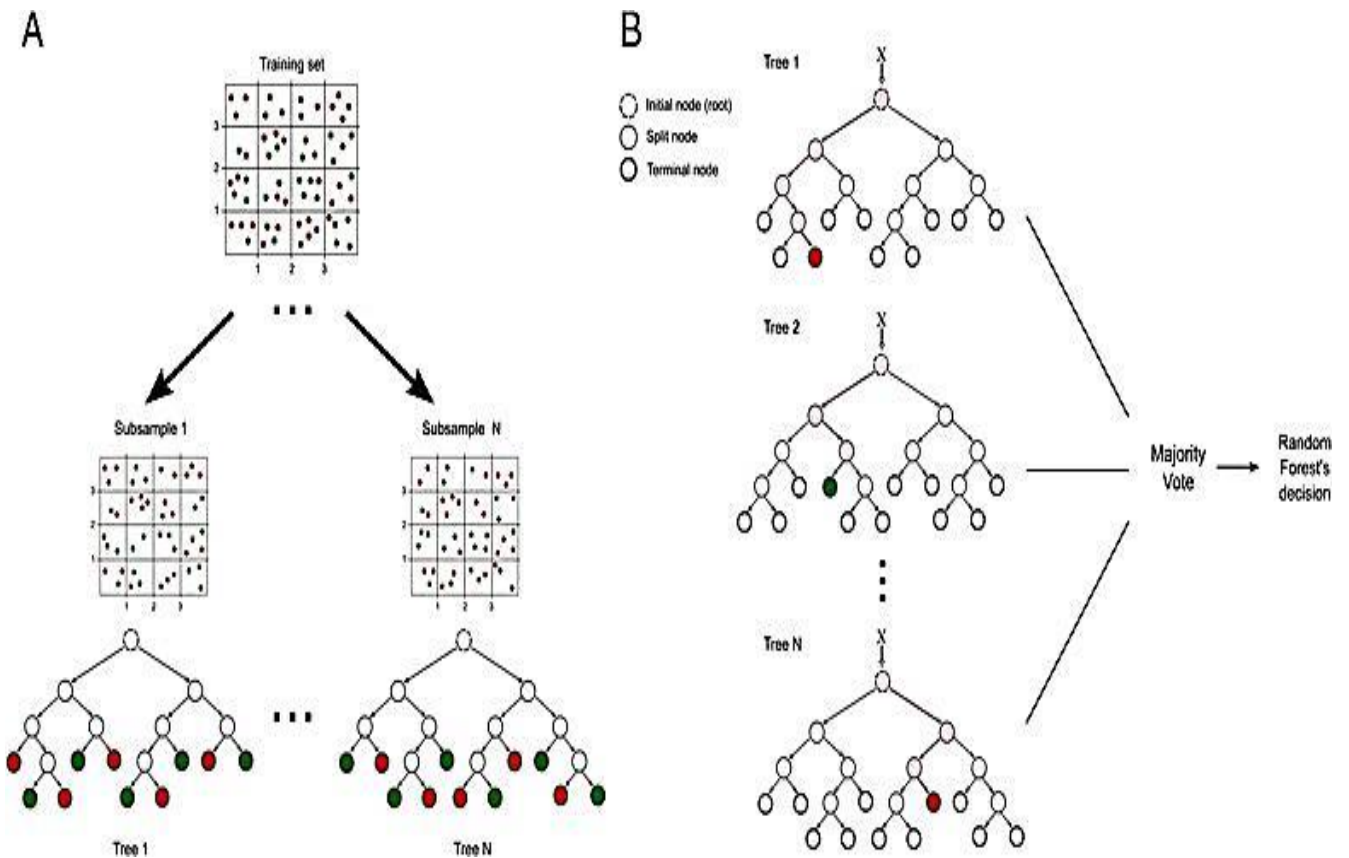
2.2.Тасодифий ўрмон

Тасодифий ўрмон-бу ечимлар дарахти алгоритмининг такомиллаштирилган кўриниши. Тасодифий ўрмоннинг асосий ғояси, бир нечта ечимлар дарахти тўпламидан иборат (“тасодифий ўрмон” тушунчаси шу ердан келиб чиққан). Ҳар бир ечимлар дарахти аралаш классификатор (берилганларнинг қуйи тўплами) ни беради. Уларнинг ҳар бири берилганларнинг турли хусусиятларини фиксирлайди. Бу дарахтлар ансамбли экспертлар гуруҳи каби ишлаб, бу ерда ҳар бир эксперт фақат ўзининг фан соҳаси бўйича текширади. сўнгра тасниф жараёнида “энг кўп овоз олган” синфни

шакллантиради. Экспертлар мисолида изоҳлайдиган бўлсак, бу битта саволни барча экспертларга бериб, жавоблар варианти ҳам бериледи ва кўпчилик танлаган жавоб вариантига қараб овозлар ҳисобланади. Регрессия кўринишида эса барча дарахтлар бўйича ўртачасини олиб натижани башорат қилиш мумкин. Шу билан бирга бошқа дарахтларга нисбатан қарор қабул қилувчи дарахтларни валидация жараёни орқали аниқлаштиришимиз мумкин.

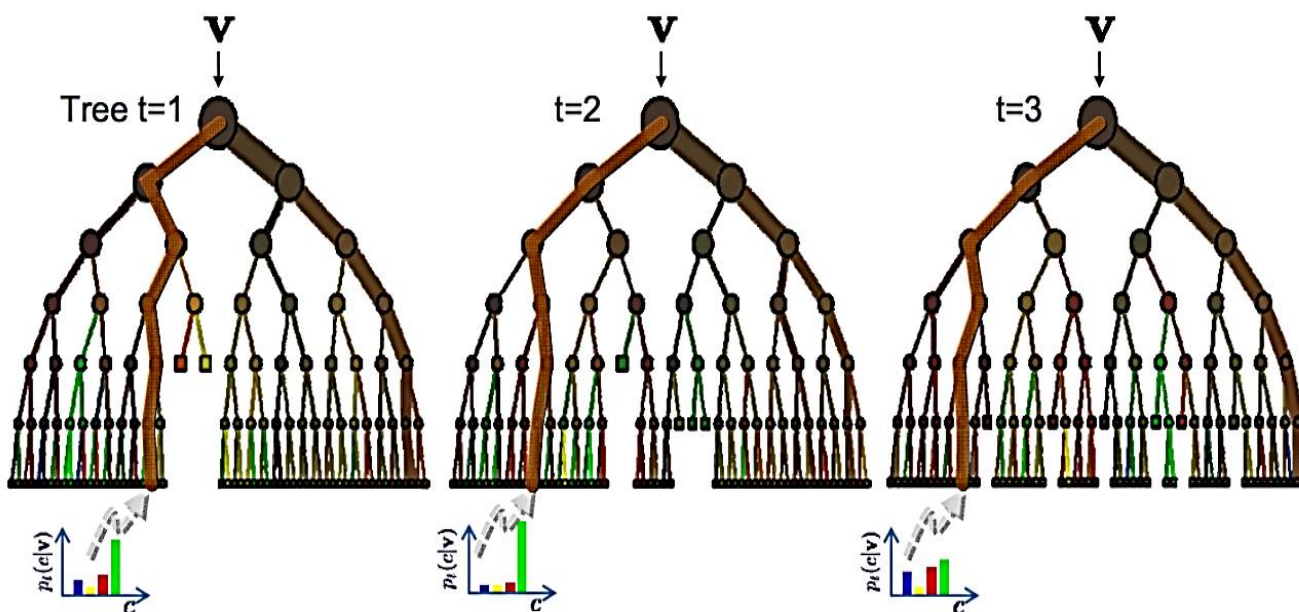
Визуаллаштириш:

- Тасниф учун кўпгина “овозлар” эксперт (дарахт)лар томонидан бериледи.



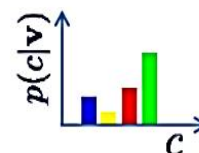
74-расм. Тасодифий ўрмон- ечимлар дарахти алгоритми

- Шу билан бирга эҳтимолликлардан фойдаланиб, таснифлаш учун чекланмани ўрнатишимиз мумкин.



The ensemble model

Forest output probability
$$p(c|\mathbf{v}) = \frac{1}{T} \sum_t p_t(c|\mathbf{v})$$



75-расм. Тасодифий ўрмонда асосий гиперпараметрлар

Тасодифий ўрмонда асосий гиперпараметрлар

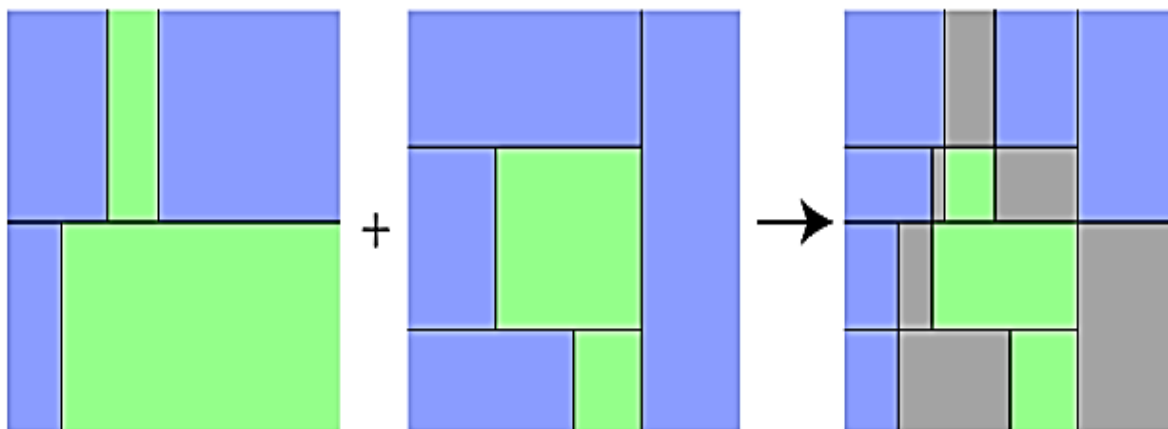
1. **N-tree**: ўрмон дарахтлари сони. Одатда 100тагача. Дарахтларнинг кўплиги баъзан ортиқчалик қилади.

2. **mtry**: аниқ дарахт бўйича тақсимотда қўлланилиши мумкин бўлган тасодифий танланадиган ўзгарувчилар сони.

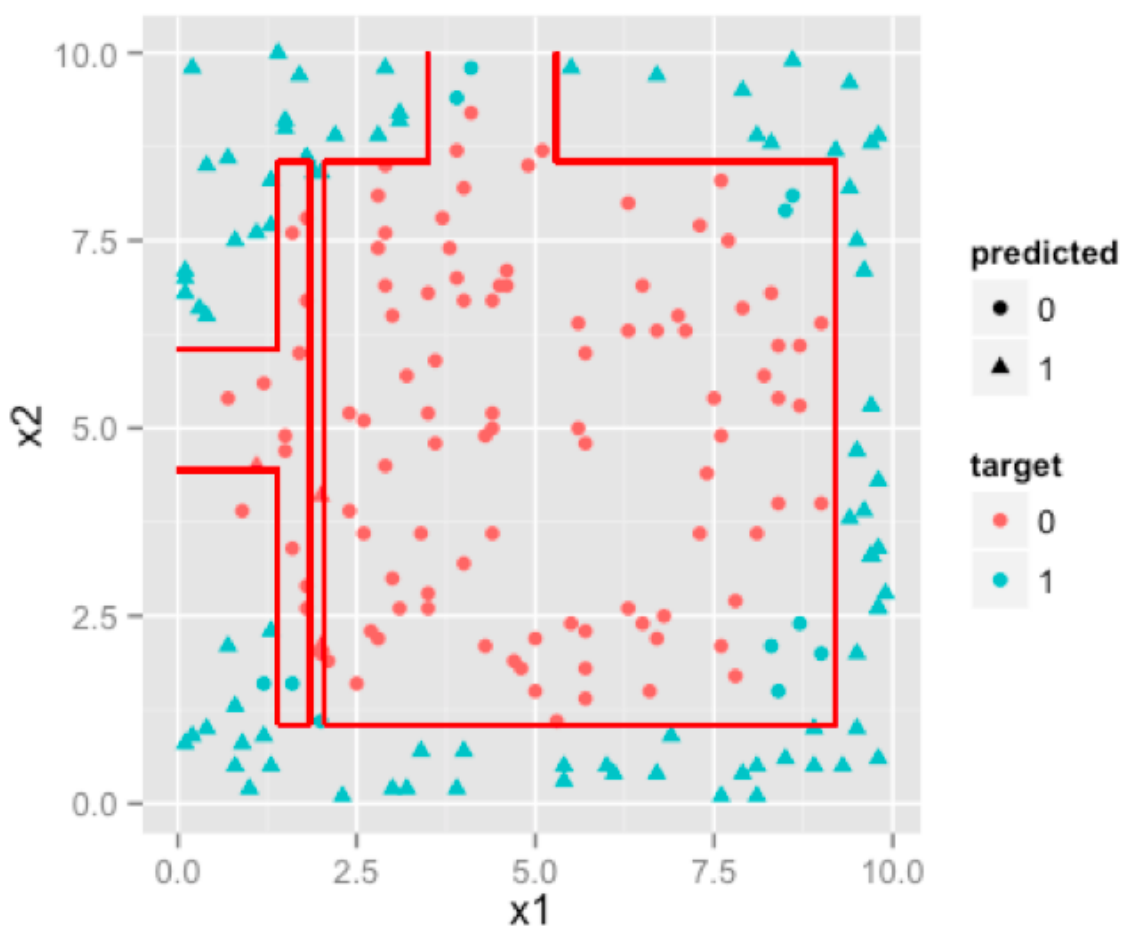
3. **ўзгартирмоқ**: намуна танлови ўзгартириш ёки ўзгармас ҳолда амалга оширилиши лозим.

тасодифий ўрмон ечимлар чегараси

Random Forest кўпгина дарахтлар тўпламини қўллаши сабабли, қарор қабул қилиш чегараларини мураккаб даражада белгилаши мумкин. Қуйида тасодифий ўрмон яратиши мумкин бўлган қарор чегаралари кўриниши келтирилган:



76-расм. тасодифий ўрмон яратиши мумкин бўлган қарор чегаралари кўриниши



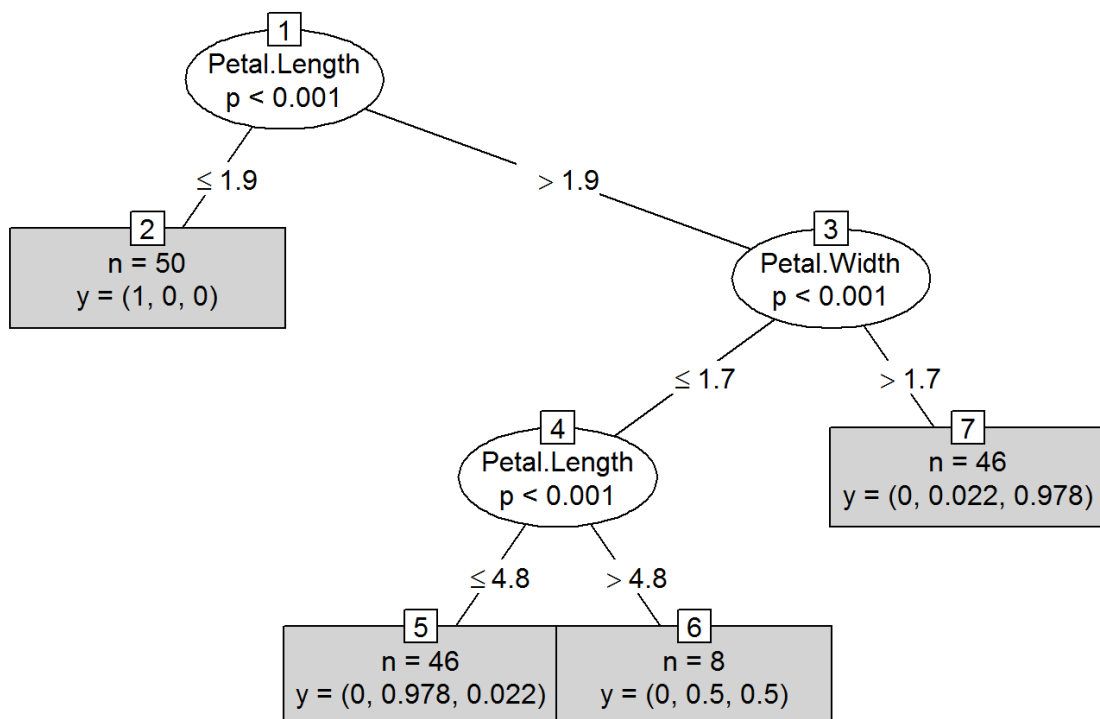
77-расм. Random Forest кўпгина дарaxтлар тўпламини

2.2.1.R да тасодифий ўрмон.

```
#Random Forest in R using IRIS data
#Split iris data to Training data and testing data
ind <- sample(2,nrow(iris),replace=TRUE)
train <- iris[ind==1,]
test <- iris[ind==2,]
head(train, 3)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 3      4.7      3.2      1.3      0.2 setosa
## 5      5.0      3.6      1.4      0.2 setosa
## 7      4.6      3.4      1.4      0.3 setosa

#Try plotting how a decision tree for IRIS will look like
x <- ctree(Species ~ ., data = iris)
plot(x, type = "simple")
```



78-расм. Дастурнинг график кўриниши.

```

#Train a RF model
model <- randomForest(Species~., data=train , ntree=50, mtry =
2,proximity=TRUE)
#Print RF model details
print(model)

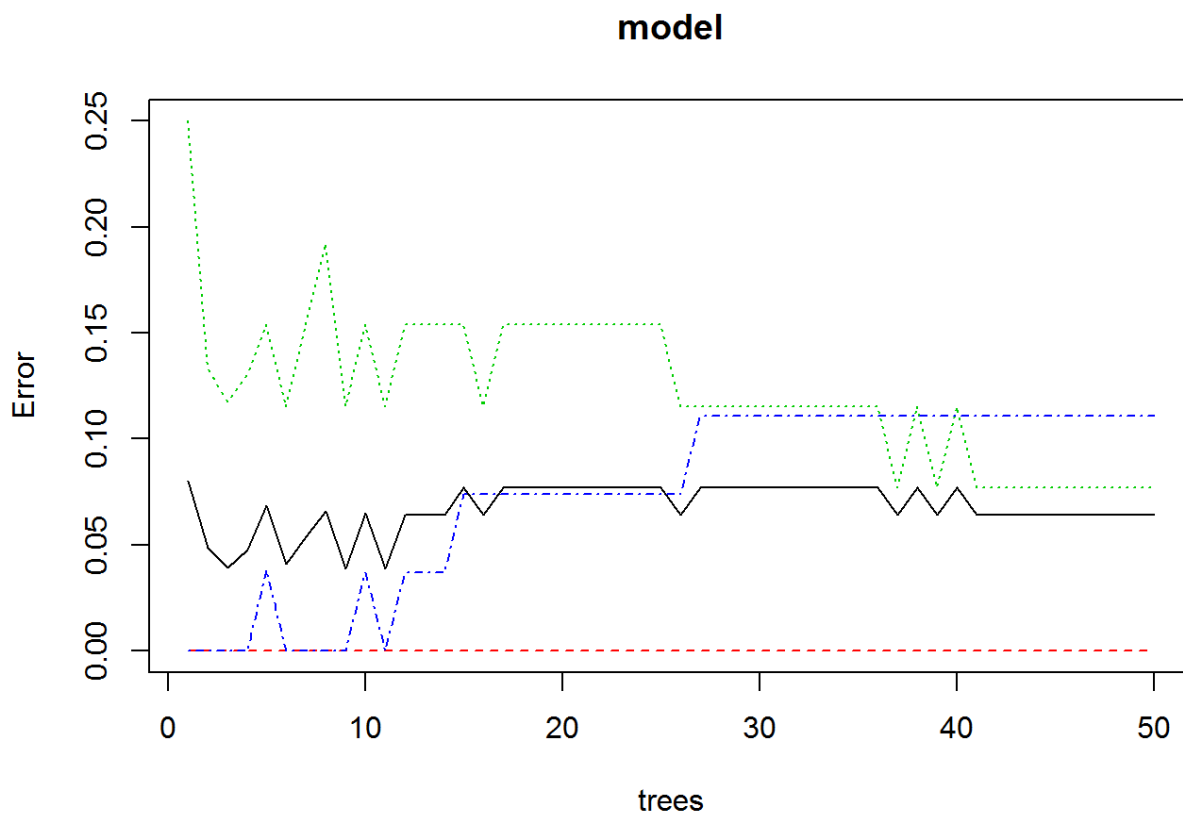
##
## Call:
## randomForest(formula = Species ~ ., data = train, ntree =
50, mtry = 2, proximity = TRUE)
##           Type of random forest: classification
##           Number of trees: 50
## No. of variables tried at each split: 2
##
## OOB estimate of error rate: 6.41%
## Confusion matrix:
##      setosa versicolor virginica class.error
## setosa      25      0      0 0.00000000
## versicolor  0      24      2 0.07692308
## virginica   0      3      24 0.11111111

#Plot error vs ntree for classes
plot(model)

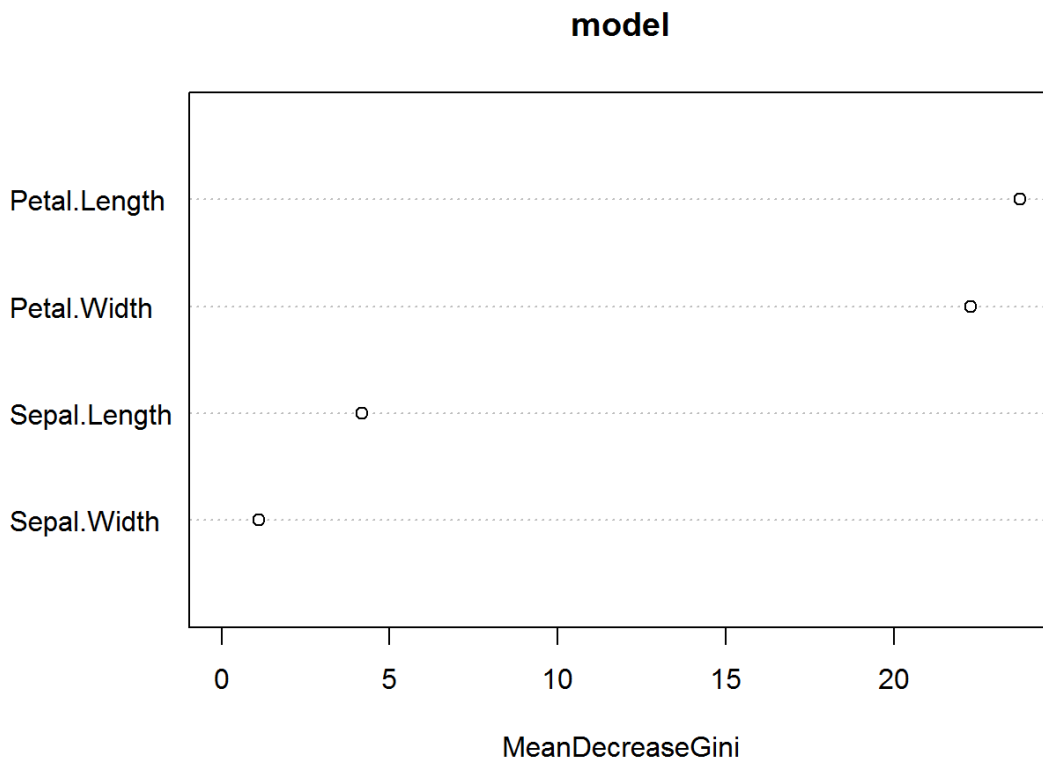
#Use the value of ntree where error becomes constant

#Plot variable importance matrix
varImpPlot(model)

```



79-расм. Дастур моделининг график кўриниши.



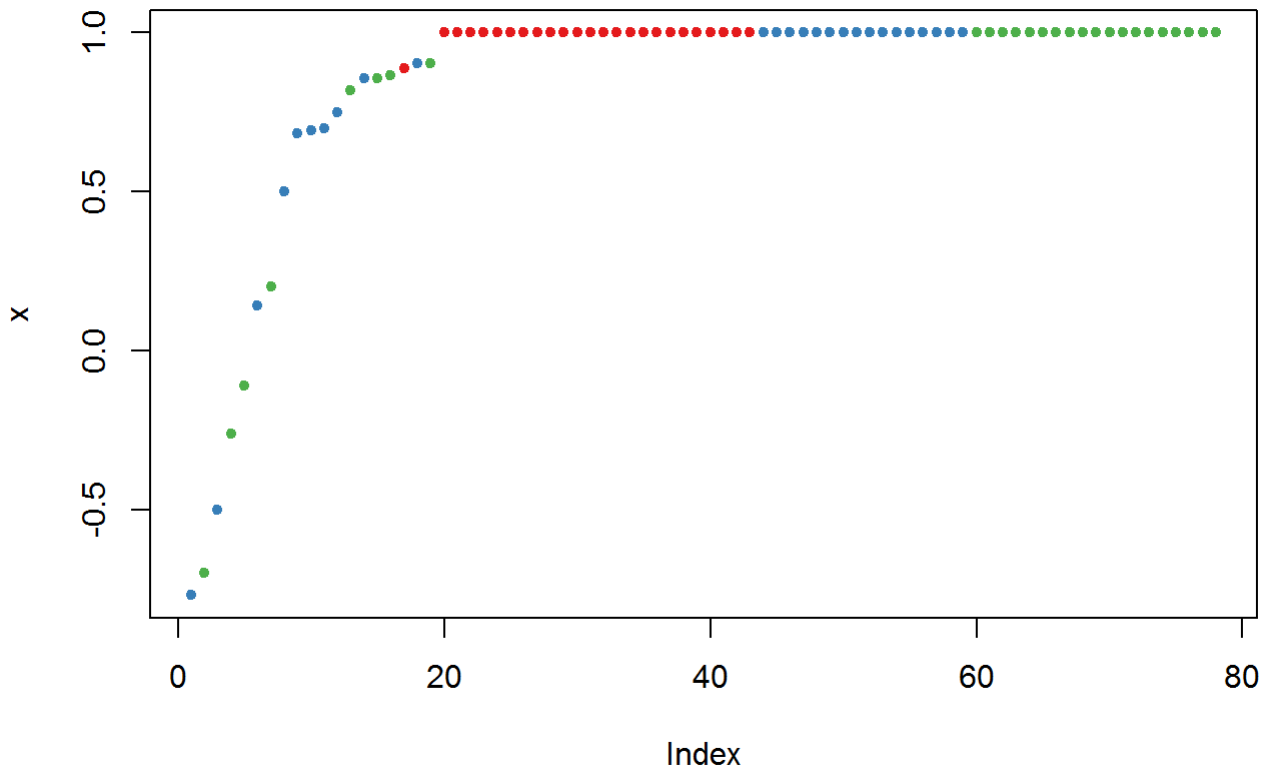
80-расм. Дастур моделининг график кўриниши.

```
#Test the model on testdata
pred <- predict(model , newdata=test)
table(pred, test$Species)
```

```
##
## pred      setosa versicolor virginica
## setosa    25      0      0
## versicolor 0      23      1
## virginica  0      1      22
```

```
#Plot the margin (positive -> correct classification)
```

```
plot(margin(model,test$Species))
```



81-расм. Дастур моделининг график кўриниши

```
#Search for optimal value of mtry for random forest
```



```
tune &lt;- tuneRF(iris[,-5],iris[,5])
```

```
## mtry = 2 OOB error = 5.33%
```

```
## Searching left ...
```

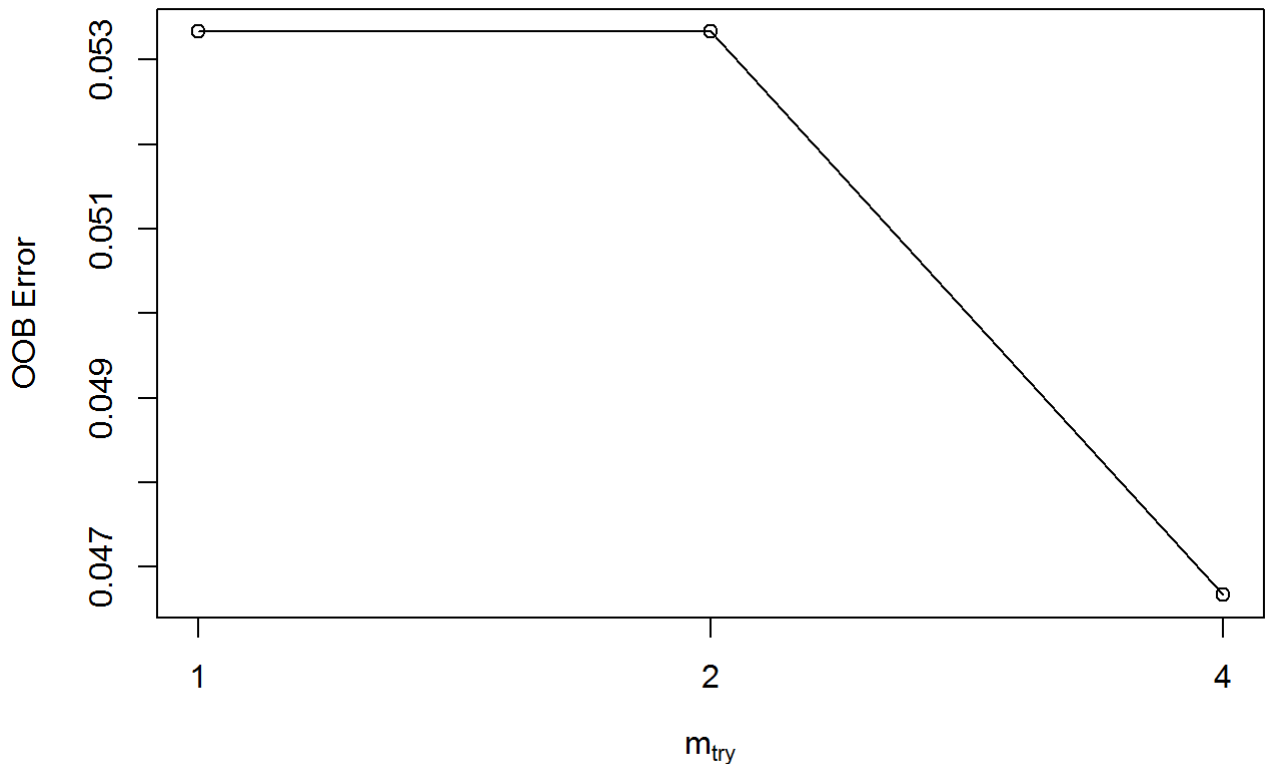
```
## mtry = 1 OOB error = 5.33%
```

```
## 0 0.05
```

```
## Searching right ...
```

```
## mtry = 4 OOB error = 4.67%
```

```
## 0.125 0.05
```



82-расм. Дастур моделининг график кўриниши

Ютуқ ва камчиликлари.

Ютуқлари:

- Қарор қабул қилишнинг аниқ моделларидан бири.
- Катта ҳажмдаги берилганлар билан ишлашда самаралидир.

- Ўзгарувчилар аҳамиятлиларини ажратиш олиш учун қўлланилиши мумкин.
- Функцияни ишлаб чиқишни талаб этмайди (масштабланиш ва нормаллаштириш)

Камчиликлари:

- Шовқинли берилганлар билан ишлашда орттирмаллилик пайдо бўлиши.
- Ечимлар дарахтларидан фарқли равишда натижалар мураккаб талқин қилинади.
- Гиперпараметрлар юқори аниқликда созланишини талаб этади.

Иловалар

Тасодифий ўрмон турли соҳаларга тадбиқ этилган бўлиб, баъзи иловалар қуйидаги имкониятларни ўз ичига олади:

- Объектларни аниқлаш.
- Молекуляр биология (аминокислота кетма-кетлигини таҳлил этиш)
- Массофали зондирование (образларни аниқлаш)
- Астрономия (юлдузлар галлактикаси таснифи ва ҳ.к.)

2.3. Байес алгоритми

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$$P(c|X) = P(x_1|c) \times \dots \times P(x_n|c) \times P(c) \quad (42)$$

Содда Байес усули– бу синфда муайян функциянинг мавжудлиги бошқа бир функциянинг мавжудлигини белгиламаслигини, яъни боғлиқлик мавжуд эмаслигини билдиради. Амалда қўллаш асосида олинган қийматлар эҳтимолликни аниқлаш усулини белгилайди.

Масалан, $P(c|x)$ қиймати $P(c)$ дан $P(x)$ эса $P(x|c)$ аниқлаш, бу усул катта ҳажмдаги берилганлар тўплами билан ишлашда самарали ҳисобланади.

Содда Байес Гауссланган усули – эҳтимолликлар тақсимотини гауссланган (нормал) деб қабул қилади. Узлуксиз тақсимот учун содда Байес Гаусс алгоритми танлаш алгоритми ҳисобланади.

Мультиномиал Байес - бу содда Байес усулининг аниқ амалий кўриниши бўлиб, бунда $P(\text{Feature}_i | \text{Class})$ мультиномиал тақсимлаш (катламлар сони, эҳтимоллиги ва ҳ.к.). Бу асосан ҳужжатларни таснифлаш масалаларини ечишда қўлланилади (ҳужжат қайси категорияга спорт, сиёсат, технология ва ҳ.к. киришини аниқлаш). Классификатор қўллайдиган функциялар/ башорат қилинувчи қийматлар, ҳужжатларда такрорланаётган сўзлар мавжудлиги.

Битта боғланиш бўйича ўртача баҳолаш-содда байес классификаторларда атрибутлар ўртасида боғланишни аниқлаш масалаларини ечиш учун ишлаб чиқилган. Кўпинча AODE, ҳисоблаш ҳажми оширилган содда байес тармоғига нисбатан аниқроқ классификаторларни ишлаб чиқади.

Байес ишонч тармоғи (BBN) - эҳтимоллик график модели (статистик модел кўриниши) бўлиб, ўзгарувчилар ва улар шартли боғланишларни циклик йўналтирилган графиги орқали ифодалайди (DAG). Масалан, байес тармоғи касаллик ва симптомлар ўртасиди эҳтимолли боғланишларни ифодалаш мумкин. Симптомларни инобатга олган ҳолда, мавжуд касаллик эҳтимоллигини ҳисоблаш учун тармоқдан фойдаланиш мумкин. BBN – бу диаграмманинг махсус кўриниши (йўналтирилган граф деб номланган) бўлиб, ўзаро боғланган эҳтимолликлар жадвалига эга.

Байес тармоғи (BN) - байес тармоғининг мақсади шартли боғланишларни моделлаштириш бўлиб, яъни мўлжаллар графига берилганлар ўртасида шартли боғланиш мавжудлиги сабабини ифодалашни англатади. Улардан фойдаланган ҳолда графикда ихтиёрий ўлчам борасидаги хулосани айнан омиллар асосида чиқариш имконини беради.

Яширин Марков моделлари (НММ) –график моделлар эҳтимолликлари синфи бўлиб, кузатилаётган ўзгарувчилар тўпламида мавжуд номаълум (яширин) ўзгарувчилар кетма-кетликларини башорат қилиш имконини беради. Масалан, унинг ёрдамида об-ҳавони (яширин ўзгарувчи) башорат қилишимиз мумкин, бунда бирон киши кийган кийими борасидаги маълумот асос (кузатув) бўлиб хизмат қилади. НММни Байес тармоғи сифатида қараш мумкин, яъни вақт тақсимоти бўйича кузатувлар ёрдамида маълум вақт оралиқларида олинган натижалар асосида, башорат қилишда аниқлик кўрсаткичи юқори бўлган яширин ҳолатлар кетма-кетлигини олишга йўналтирилган.

Шартли тасодифий майдонлар (CRF)- машинали ўрганишнинг классик модели бўлиб, кетма-кет моделлар ўрганиши учун қўлланилади. Бу кўринишдаги классификатор турли синфлар бўйича ечим чегарасини моделлаштиради. Бу классификатор ва моделларни яратувчилар ўртасидаги фарқ шундаки, биринчи кўриниш, бу эҳтимолликлар шартли тақсимоти бўйича моделлаштиради, яъни $P(y | x)$, моделларни яратиш усулида эҳтимолликлар ҳамкорлик тақсимотларни моделлаштиришга, яъни $P(x, y)$ ҳаракат қилади. Уларнинг асосий принципи бу кириш кетма-кетлигига логистик регрессияни қўллашдир. Яширин Марков моделлар CRF билан баъзи умумий хусусиятга эга бўлиб, улардан бири бу кириш кетма-кетлигида қўлланилишидир. Кўпинча CRFs НЛП (нейро-лингвистическое программирование) да қўлланилади.

Мисол кўриб ўтамиз:

CRFs – дарахтсимон ёки ноаниқ графлар кўринишдаги кетма-кетликларни, яъни структураланмаган кетма-кетликни умумлаштириш учун қўлланилади. Кўрилаётган мисолимизда чизиқли кетма-кетликка эга CRF структурасига асосланамиз. Мисол сўнгида эса умумий CRF ва чизиқли кетма-кетликка эга CRF билан умумлаштирган ҳолда изоҳ берилади.

2.3.1. Асосий назария

Машинали ўрганиш тизимларининг кўпчилиги CRF қўллайди, бунда биологик кетма-кетликдан бошлаб, компьютерли видео кузатув, табиий тил маълумотларини қайта ишлаш масалаларида қўлланилиб келинган, CRF бўйича 20000 дан ортиқ мақолалар чиқарилган.

Сўнгги йилларда CRF моделлари LSTM билан бирлаштирилган ва энг самарали натижалар олинган. НЛП соҳасида CRF ни кетма-кетликлар қондаси деб ҳисобланади: агар аниқлик даражасини оширмоқчи бўлсангиз маълумотлар кетма-кетлигига LSTM қатламига ўрнатишнинг ўзи етарли бўлиб ҳисобланади.

Кетма-кетликлар таснифи масаласида асосий мақсад кетма-кетликлар эҳимоллиги (y) ни киритилаётган кетма-кетликлар вектори (X) га нисбатан аниқланади. Унинг кўриниши $P(y|X)$.

Энг аввал, ифодалар изоҳини кўриб чиқамиз:

- Ўрганиш тўплами: кириш ва мақсадли натижа кетма-кетликлар жуфтлиги $\{(X_i, y_i)\}$

- I - кирувчи кетма-кетликлар векторлари: $X_i = [x_1, \dots, x_l]$

- I - нчи мақсадли кетма-кетликлар тўплами: $Y_i = [y_1, \dots, y_l]$

- l -кетма-кетликлар узунлиги.

(X, y) танлов учун доимий тасниф масаласида $P(y|X)$ ни ҳисобланади, бунда кетма-кетликдаги k - позициядаги ҳар бир элемент учун эҳтимолликлар кўпайтмаси бажарилади, бунда $1 \leq k \leq l$:

$$P(y|X) = \prod_{k=1}^l P(y_k|x_k) = \prod_{k=1}^l \frac{\exp(U(x_k, y_k))}{Z(x_k)} = \frac{\exp(\sum_{k=1}^l U(x_k, y_k))}{\prod_{k=1}^l Z(x_k)} \quad (43)$$

$P(y_k|x_k)$ ифодадан нормаллаштирилган экспонентани қўллаб моделлаштирамиз. Бу *softmax* амали билан бир хил бўлиб, нейрон тармоқларда кенг қўлланилади. *Exp* қўллаш лозим бўлган ҳолларни кўриб ўтамиз:

1. **Миқдорнинг етарлича бўлмаганлиги:** жуда кичик сонларни кўпайтиришда кичик сонлар олинади бу ўз навбатида *миқдор етмаслигига* олиб келади.

2. **Манфий бўлмаган натижалар:** барча қийматлар 0 ва мусбат сонлар оралиғида ифодаланади.

3. Монотон ҳолда ўсиб борувчи: бунда юқори даражали қийматларни оширади, паст даражали қийматларни камайтиради. Бу жараён *argmax* амали билан бир хилдир.

Бир тўпланда икки ўзгарувчи: иккита символни тўпландан танлаймиз: U ва Z . Улар изоҳини кўриб ўтамиз.

$U(x, y)$ танловлар ёки бирламчи кўрсаткичлар каби бўлиши мумкин. Яъни y кўрсаткичга нисбатан x вектор k - вақт кадам оралиғида ифодаланилади. Буни LSTM нинг k - чиқиш қисми сифатида қабул қилишингиз мумкин. Аслида назарий жиҳатдан x вектор турли кўринишда бўлиши мумкин. Амалда эса x вектор атрофдаги элементларнинг кетма-кетликлар занжири ҳисобланади, яъни киритиш ойнасидаги сўзлардан иборат. Ҳар бир *унар* омил моделда оғирлик кўрсаткичи асосида белгиланган. LSTM натижаси деб қабул қилинса, англаш анча осон бўлади.

$Z(x)$ ни одатда **тақсимлаш функцияси** деб номланади. Буларнинг барчасини оддий ҳол омили деб қабул қиламиз, чунки эҳтимоллик даражалари: ҳар бир кўрсаткич учун қиймати 1 га тенг деб баҳоланиши лозим. *Softmax* функциясининг махражи каби қабул қилиш лозим.

Юқорида, softmax функциясини қўллаган ҳолда эҳтимолликни аниқлайдиган доимий таснифловчи модельни кўриб ўтдик. Кейинги кадамда янги оғирлик кўрсаткичини қўшган ҳолда y_k сўнг y_{k+1} келиш эҳтимолликни моделлаштириш лозим бўлади. Бу модель асосида кетма-кетликлар ўртасида боғланишни аниқлаймиз, демак *чизиқли-занжирли CRF* тушунчаси пайдо бўлади. Бунинг учун, олдинги эҳтимолликни $P(y_k + 1|y_k)$ га кўпайтириб, экспоненциал хусусият сифатида фойдаланишимиз мумкин, $U(x, y)$ унар кўриниши ва ўрганилаётган янги $T(y, y)$ баҳолаш йиғиндиси кўринишига олиб келади:

$$P(y|X) = \frac{\exp(\sum_{k=1}^l U(x_k, y_k) + \sum_{k=1}^{l-1} T(y_k, y_{k+1}))}{Z(X)} \quad (44)$$

$T(y, y)$ – бу матрица (nb_labels, nb_labels) катталиқда бўлиб, ҳар бир қиймат ўрганиладиган параметр деб қабул қилинади ва i -меткадан j -й меткага ўтишни билдиради. Янги ўзгарувчиларни кўриб ўтамиз:

- **Чиқариб ташлаш ёки унар баллар (U):** бу баллар x_k кириш берилганларини инобатга олган ҳолда y_k бўлиш эҳтимоллигини белгилайди.

- **Ўтиш кўрсаткичи (T):** бу кўрсаткич y_k дан кейин y_{k+1} кетма-кетликда келиш эҳтимоллигини белгилайди.

- **Тақсимот функцияси (Z):** нормаллаштириш коэффициенти бўлиб, охири босқичда эҳтимолликни аниқлаш учун қўлланилади.

Демак, тақсимот функцияси Z ни аниқлаш қолди:

$$Z(X) = \sum_{y'_1} \sum_{y'_2} \dots \sum_{y'_k} \dots \sum_{y'_l} \exp(\sum_{k=1}^l U(x_k, y'_k) + \sum_{k=1}^{l-1} T(y'_k, y'_{k+1})) \quad (45)$$

$Z(X)$ ҳисоблаш мураккаблиги кўриниб турибди, унда жуда кўп цикллар ичма-ич жойлашган. Бу сумма ҳар бир вақт кесимида бўлиши мумкин бўлган барча комбинациялар йиғиндиси. Аниқроғи, меткалар тўплами устида $l!$ (факториал) ҳисоблаш лозим бўлади. Бу вақт борасида мураккабликка $O(l! |Y|^2)$ олиб келади.

Бу борада самарали натижа олиниши учун рекуррент боғланишни ва динамик дастурлашни қўллаш мумкин. Ушбу алгоритм бажарилиши кетма-кетлигига қараб тўғри алгоритм ёки тескари алгоритм деб номланади.

Код

Кодни яратишда энг аввал `nn.Module` в *pytorch* дан олинган CRF синфни яратиш билан бошлаймиз, бунда градиентни автоматик кузатиш имкони берилади.

Махсус чекланишлар қўйиб тўлиш чегараси бўйича ҳаракатни аниқлаш имконини беради.

```
import torch
```

```

from torch import nn
class CRF(nn.Module):
    """
    Linear-chain Conditional Random Field (CRF).
    Args:
        nb_labels (int): number of labels in your tagset, including special
symbols.
        bos_tag_id (int): integer representing the beginning of sentence
symbol in
        your tagset.
        eos_tag_id (int): integer representing the end of sentence symbol in
your tagset.
        batch_first (bool): Whether the first dimension represents the batch
dimension.
    """
    def __init__(
        self, nb_labels, bos_tag_id, eos_tag_id, batch_first=True
    ):
        super().__init__()
        self.nb_labels = nb_labels
        self.BOS_TAG_ID = bos_tag_id
        self.EOS_TAG_ID = eos_tag_id
        self.batch_first = batch_first
        self.transitions = nn.Parameter(torch.empty(self.nb_labels,
self.nb_labels))
        self.init_weights()
    def init_weights(self):
        # initialize transitions from a random uniform distribution between -
0.1 and 0.1
        nn.init.uniform_(self.transitions, -0.1, 0.1)
        # enforce constraints (rows=from, columns=to) with a big negative
number
        # so exp(-10000) will tend to zero

```



```
# no transitions allowed to the beginning of sentence
self.transitions.data[:, self.BOS_TAG_ID] = -10000.0
# no transition allowed from the end of sentence
self.transitions.data[self.EOS_TAG_ID, :] = -10000.0
```

2.3.2. Йўқотиш функцияси моҳияти

Тасниф масаласининг асосий моҳияти ўрганиш давомидаги хатоликни минималлаштиришдир. L йўқотиш функциясини аниқлаб амалга оширишимиз мумкин, бунда кириш маълумотлари сифатида башорат натижалари ва аниқ кўрсатмаларни қабул қилади, сўнгра улар тенг бўлса нолни, агар тенг бўлмаса хатоликни билдиради.

Эътибор беринг, $P(\mathbf{y} | \mathbf{X})$ ни ҳисоблаш, яъни айнан уни максималлаштириш керак бўлади. Бу масалани минималлашув кўринишга келтириш учун эҳтимолликнинг тескари логарифмини оламиз.

Эҳтимолликнинг логарифмик тескари йўқотиш функцияси каби маълум (NLL -Loss). Бизнинг мисолда бу қуйидаги кўринишда бўлади: $L = -\log(P(\mathbf{y} | \mathbf{X}))$. \log -функциясини қўллаган ҳолда $\log(a/b) = \log(a) - \log(b)$, қуйидаги ифодани оламиз:

$$\begin{aligned}
 -\log(P(\mathbf{y} | \mathbf{X})) &= -\log\left(\frac{\exp(\sum_{k=1}^l U(x_k, y_k) + \sum_{k=1}^{l-1} T(y_k, y_{k+1}))}{Z(\mathbf{X})}\right) = \\
 &\quad \log(Z(\mathbf{X})) - \\
 &\quad \log(\exp(\sum_{k=1}^l U(x_k, y_k) + \sum_{k=1}^{l-1} T(y_k, y_{k+1}))) = \log(Z(\mathbf{X})) - \\
 &\quad (\sum_{k=1}^l U(x_k, y_k) + \sum_{k=1}^{l-1} T(y_k, y_{k+1})) = Z_{\log}(\mathbf{X}) - (\sum_{k=1}^l U(x_k, y_k) + \\
 &\quad \sum_{k=1}^{l-1} T(y_k, y_{k+1})) \quad (46)
 \end{aligned}$$

Бунда Z_{\log} ҳисоблашлар \log функциясидан келиб чиққан ҳолда аниқланади. Бу алгоритмни кейинчалик қўллашда бизнинг масалани ечимини осонлаштиради. Бунинг учун код кўриниши қандай бўлишини кўриб ўтамиз:

```

def forward(self, emissions, tags, mask=None):
    nll = -self.log_likelihood(emissions, tags, mask=mask)
    return nll
def log_likelihood(self, emissions, tags, mask=None):
    """Compute the probability of a sequence of tags given a sequence of
    emissions scores.
    Args:
        emissions (torch.Tensor): Sequence of emissions for each label.
            Shape of (batch_size, seq_len, nb_labels) if batch_first is True,
            (seq_len, batch_size, nb_labels) otherwise.
        tags (torch.LongTensor): Sequence of labels.
            Shape of (batch_size, seq_len) if batch_first is True,
            (seq_len, batch_size) otherwise.
        mask (torch.FloatTensor, optional): Tensor representing valid
positions.
            If None, all positions are considered valid.
            Shape of (batch_size, seq_len) if batch_first is True,
            (seq_len, batch_size) otherwise.
    Returns:
        torch.Tensor: the log-likelihoods for each sequence in the batch.
            Shape of (batch_size,)
    """
    # fix tensors order by setting batch as the first dimension
    if not self.batch_first:
        emissions = emissions.transpose(0, 1)
        tags = tags.transpose(0, 1)
    if mask is None:
        mask = torch.ones(emissions.shape[:2], dtype=torch.float)
    scores = self._compute_scores(emissions, tags, mask=mask)
    partition = self._compute_log_partition(emissions, mask=mask)
    return torch.sum(scores - partition)

```

Тўғри ечим – бу *NLL* *йўқотиш* функцияси бўлиб, оддий *log_эхтимоллик* функциясидан олдин минус белгисини кўямиз. *Log_эхтимоллик* функциясининг ўзи бошланғич натижаларни баҳолаш, *log* кетма-кетлиги ҳисоблаш усулини аниқлайди.

```
input = [['lorem', 'ipsum 'dolor', 'sit', 'amet'],
         ['another', 'sentence', 'here', '<pad>', '<pad>']]
mask = [[1, 1, 1, 1, 1],
        [1, 1, 1, 0, 0]]
```

Ифода сурат қисмини ҳисоблаш: натижалар

Биз *log* ни *exp* га нисбатан қўллаганимиз сабабли, бу сурат кўриниш ҳисобланади- яъни ҳар бир вақт оралиғида кўрсаткичлар ва ўзгаришлар йиғиндиси ҳисобланади.

```
def _compute_scores(self, emissions, tags, mask):
```

Args:

emissions (torch.Tensor): (batch_size, seq_len, nb_labels)

tags (Torch.LongTensor): (batch_size, seq_len)

mask (Torch.FloatTensor): (batch_size, seq_len)

Returns:

torch.Tensor: Scores for each batch.

Shape of (batch_size,)

"""

```
batch_size, seq_length = tags.shape
```

```
scores = torch.zeros(batch_size)
```

```
# save first and last tags to be used later
```

```
first_tags = tags[:, 0]
```

```
last_valid_idx = mask.int().sum(1) - 1
```

```
last_tags = tags.gather(1, last_valid_idx.unsqueeze(1)).squeeze()
```

```
# add the transition from BOS to the first tags for each batch
```

```
t_scores = self.transitions[self.BOS_TAG_ID, first_tags]
```

```
# add the [unary] emission scores for the first tags for each batch
```

```
# for all batches, the first word, see the correspondent emissions
```

```
# for the first tags (which is a list of ids):
```

```

# emissions[:, 0, [tag_1, tag_2, ..., tag_nlabels]]
e_scores = emissions[:, 0].gather(1,
first_tags.unsqueeze(1)).squeeze()
# the scores for a word is just the sum of both scores
scores += e_scores + t_scores
# now lets do this for each remaining word
for i in range(1, seq_length):
    # we could: iterate over batches, check if we reached a mask
symbol
    # and stop the iteration, but vecotrizing is faster due to gpu,
    # so instead we perform an element-wise multiplication
is_valid = mask[:, i]
previous_tags = tags[:, i - 1]
current_tags = tags[:, i]
# calculate emission and transition scores as we did before
e_scores = emissions[:, i].gather(1,
current_tags.unsqueeze(1)).squeeze()
t_scores = self.transitions[previous_tags, current_tags]
# apply the mask
e_scores = e_scores * is_valid
t_scores = t_scores * is_valid
scores += e_scores + t_scores
# add the transition from the end tag to the EOS tag for each batch
scores += self.transitions[last_tags, self.EOS_TAG_ID]
return scores

```

Ушбу кодни тушуниш учун тўплам ичидаги барча ҳолатлар учун амаллар бир хил деб қабул қилиш керак. Демак, ҳар бир тўплам учун биринчи сўз теги `tags[:, 0]` юклатилади. Худди шундай вақт қадамлари бўйича йиғиндини ҳисоблаб узунликни аниқлаймиз, бу эса олдинги мисол бўйича `[5, 3]` қийматда бўлиши мумкин. Амалда факториал кўрсаткичли кўринишда формой (`batch_size,`) бўлиши мумкин. Масалан, 28 сатрга эътибор қаратсак:

`emissions[:, 0].gather(1, first_tags.unsqueeze(1)).squeeze()`

1. энг аввал бошланғич вақт оралиғидан барча кетма-кетликларни `emissions[:, 0]` оламиз, бунда тензор (`batch_size, nb_labels`) кўринишда қайтарилади.

2. Сўнгра `LongTensor first_tags` да жойлашган барча устун (`dim=1`) қийматларини оламиз, улар (`batch_size,`) кўринишда бўлади. `Emissions` кўриниши 2D-матрица бўлганлиги сабабли, биз сўнгги ўлчамни `first_tags` шундай олишимиз керакки, натижа(`batch_size, 1`):
`first_tags.unsqueeze (1)` кўринишда бўлсин.

3. Кейинги қадам, икки кўриниш ҳам бир хиллигини инобатга олиб, `gather` функциясини қўллаган ҳолда `first_tags` таркибидан белгиланган ўлчамда танловларни олиш мумкин: `emissions[:, 0].tanлаш(1, first_tags.unsqueeze(1))`

4. Натижада (`batch_size, 1`) кўринишдаги матрицага эга бўламиз ва уни қисқартирган ҳолда 1D `LongTensor`га эришамиз.

Бу содда процедура кодларда берилган ўлчамда кўрсаткичларни гуруҳлаш учун қўлланилади.

Бу код борасида сўнгги изоҳ сифатида шуни айтиш лозимки, тўлдириш симболи билан боғлиқ кўрсаткичларни инобатга олмаган ҳолда қуйидаги амални бажариш керак бўлади: икки вектор элементарларини ўзаро кўпайтирган ҳолда 0 гача етказамиз ва вақт оралиғи ўтиш вазифасини бажаради.

Тақсимлаш функциясини ҳисоблаш: тўғри алгоритм

Демак, етарлича баллар ҳисоблангандан сўнг, махраж қисмига эътиборни қаратамиз. Тақсимлаш функциясини самарали ҳисоблаш учун тўғри алгоритмни қўллаймиз. Қуйида уни *log* соҳада ҳисоблашни қисқача кўриб ўтамыз.

Тўғри алгоритмнинг кодли кўриниши кўриб ўтамыз:

1) ҳар бир y'_2 қийматларини инициаллаштириш:

$$\alpha_1 (y'_2) = \sum_{y'_1} \exp(U(x_i, y'_1) + T(y'_1, y'_2)) \quad (47)$$

2) барча y'_{k+1} қийматлар учун $k=2$ дан $\ell-1$ гача ҳисоблаш, (`log-space`):

$$\log(\alpha_k (y'_{k+1})) = \log \sum_{y'_k} \exp(U(x_k, y'_k) + T(y_k, y_{k+1}) + \log(\alpha_{k-1} (y'_k))) \quad (48)$$

3) энг охири,:

$$Z(X) \log \sum_{y'_l} \exp(U(x_l, y'_l) + \log(\alpha_{k-1} (y'_k))) \quad (49)$$

Шуни таъкидлаш лозимки, 2-қадамда \exp кетма-кетликлар йиғиндисини оламиз. Жорий ҳисоблашда y'_k қиймати жуда катта бўлса, экспонента жуда йирик сонгача ошиши мумкин. Энг аввал кетма-кетлик охиридан олиб борамиз ва тўлиб кетиш муаммосини олдини оламиз бу амал барқарорлигини қуйидагича амалга оширамиз:

$$\log \sum_k \exp((z_k) = \max(z) + \log \sum_k \exp (z_k - \max(z)) \quad (50)$$

Чап томондаги ифода ўнг томондагига тенглигини қуйидаги исботлайди:

$$\begin{aligned} _ &= \log \text{sum}(\exp(z_k)) \\ &= \log \text{sum}(\exp(z_k - c) * \exp(c)) \\ &= \log \exp(c) + \log \text{sum}(\exp(z_k - c)) \\ &= c + \log \text{sum}(\exp(z_k - c)) \end{aligned}$$

PyTorch алгоритми ёрдамидаги код келтирилади:

```
def _compute_log_partition(self, emissions, mask):
```

```
    """Compute the partition function in log-space using the forward-
algorithm.
```

```
    Args:
```

```
        emissions (torch.Tensor): (batch_size, seq_len, nb_labels)
```

```
        mask (Torch.FloatTensor): (batch_size, seq_len)
```

```
    Returns:
```

```
        torch.Tensor: the partition scores for each batch.
```

```

        Shape of (batch_size,)
        """
        batch_size, seq_length, nb_labels = emissions.shape
        # in the first iteration, BOS will have all the scores
        alphas = self.transitions[self.BOS_TAG_ID, :].unsqueeze(0) +
emissions[:, 0]
        for i in range(1, seq_length):
            alpha_t = []
            for tag in range(nb_labels):
                # get the emission for the current tag
                e_scores = emissions[:, i, tag]
                # broadcast emission to all labels
                # since it will be the same for all previous tags
                # (bs, nb_labels)
                e_scores = e_scores.unsqueeze(1)
                # transitions from something to our tag
                t_scores = self.transitions[:, tag]
                # broadcast the transition scores to all batches
                # (bs, nb_labels)
                t_scores = t_scores.unsqueeze(0)
                # combine current scores with previous alphas
                # since alphas are in log space (see logsumexp below),
                # we add them instead of multiplying
                scores = e_scores + t_scores + alphas
                # add the new alphas for the current tag
                alpha_t.append(torch.logsumexp(scores, dim=1))
            # create a torch matrix from alpha_t
            # (bs, nb_labels)
            new_alphas = torch.stack(alpha_t).t()
            # set alphas if the mask is valid, otherwise keep the current values
            is_valid = mask[:, i].unsqueeze(-1)
            alphas = is_valid * new_alphas + (1 - is_valid) * alphas
        # add the scores for the final transition

```

```

last_transition = self.transitions[:, self.EOS_TAG_ID]
end_scores = alphas + last_transition.unsqueeze(0)
# return a *log* of sums of exps
return torch.logsumexp(end_scores, dim=1)

```

Юқорида келтирилган код баҳолаш кўрсаткичини аниқлаш амалига жуда ўхшаш бўлиб, аслида эса, биз уларни олдинги терациясига қараб йиғиб борамиз. Фақат битта сатр биз учун янгилик:

* $alphas = is_valid * new_alphas + (1 - is_valid) * alphas$: бу сатрдаги кодда биз $alpha$ жорий қийматини то ўтиш қисмига етмагунча янгисига ўзгартирамиз. Ўтиш позициясига етганда ўз ҳолича сақлаймиз. Буни қуйидаги мисолда $i=1$ вақт оралиғида ишлаш кўринишида келтирилган:

```

>>> mask
tensor([[1., 0., 0.],
        [1., 1., 0.],
        [1., 1., 1.]])

>>> alphas
tensor([[[-0.7389, -0.6433, -0.0571, -0.3587, -2.1117],
         [ 1.0372,  1.8366, -0.9350, -1.2656, -0.5815],
         [ 0.1011,  0.7373,  0.0929, -0.8695,  0.7016]])

>>> new_alphas
tensor([[11.1889, 10.6471, 11.0028, 11.0248, 11.0909],
        [10.3975, 11.0104,  8.5674, 10.2359, 13.9150],
        [10.1440,  9.9298, 11.3141, 10.1534, 10.3397]])

>>> is_valid = mask[:, 1].unsqueeze(-1)

>>> is_valid
tensor([[0.],
        [1.],
        [1.]])

>>> is_valid * new_alphas + (1 - is_valid) * alphas
tensor([[[-0.7389, -0.6433, -0.0571, -0.3587, -2.1117],

```


[10.3975, 11.0104, 8.5674, 10.2359, 13.9150],
[10.1440, 9.9298, 11.3141, 10.1534, 10.3397]]

Биз 2- ва 3- кетма-кетликни янгиладик, бироқ 1-ни ўзгармас қолдирдик, сабаби timestep $i=1$ да биз ўтиш позициясига эришдик.

Шуни таъкидлаш лозимки, `logsumexp` қўлланилганда лог-муҳитга ўтилади, шунинг учун баҳолар тўпламига α ни қўшиб қўйишнинг ўзи етарли. Ва сўнгида яна бир `logsumexp` амалини олиб сўнгида охириги қийматларни қайтариш имконини олишимиз мумкин.

Энг самарали кўрсаткич кетма-кетлигини аниқлаш:

Тақсимлаш функцияси аниқлагандан сўнг, тескари алгоритмни ҳисобласак, яъни кетма-кетликни тескари кесишиши аниқланганлик натижасида $P(y_k | X)$ ни ҳар k кадамда максималлаштириш кўрсаткичини аниқлаш мумкин. Агар CRF тақсимот тўғри деб қабул қилинган ҳолат учун оптимал ечим ифодасини қуйида келтириш мумкин:

$$P(y_k | X) = \frac{\exp(U(x_k, y_k) + \log(\alpha_{k-1}(y_k)) + \log(y_k(\beta_{k+1})))}{\sum_{y'_k} \exp(U(x_k, y'_k) + \log(\alpha_{k-1}(y'_k)) + \log(\beta_{k-1}(y'_k)))}$$

(51)

Бунда α баҳолаш тўғри алгоритмда аниқланади β -баҳолаш эса тескари алгоритмдан. Шунинг учун энг самарали y^* кетма-кетликларини аниқлаш учун ҳар бир кадамда argmax ҳисобланади:

$$y^* = \operatorname{argmax}_{y_{k1}, y_2, \dots, y_l} P(y_k | X) \quad (52)$$

```
def decode(self, emissions, mask=None):
```

```
    """Find the most probable sequence of labels given the emissions  
using  
the Viterbi algorithm.
```

```
    Args:
```

```
        emissions (torch.Tensor): Sequence of emissions for each label.
```

```
        Shape (batch_size, seq_len, nb_labels) if batch_first is True,
```

```

        (seq_len, batch_size, nb_labels) otherwise.
    mask (torch.FloatTensor, optional): Tensor representing valid
positions.
    If None, all positions are considered valid.
    Shape (batch_size, seq_len) if batch_first is True,
    (seq_len, batch_size) otherwise.
Returns:
    torch.Tensor: the viterbi score for the for each batch.
        Shape of (batch_size,)
    list of lists: the best viterbi sequence of labels for each batch.
"""
if mask is None:
    mask = torch.ones(emissions.shape[:2], dtype=torch.float)
scores, sequences = self._viterbi_decode(emissions, mask)
return scores, sequences
def _viterbi_decode(self, emissions, mask):
    """Compute the viterbi algorithm to find the most probable sequence
of labels
    given a sequence of emissions.
Args:
    emissions (torch.Tensor): (batch_size, seq_len, nb_labels)
    mask (Torch.FloatTensor): (batch_size, seq_len)
Returns:
    torch.Tensor: the viterbi score for the for each batch.
        Shape of (batch_size,)
    list of lists of ints: the best viterbi sequence of labels for each batch
"""
    batch_size, seq_length, nb_labels = emissions.shape
    # in the first iteration, BOS will have all the scores and then, the max
    alphas = self.transitions[self.BOS_TAG_ID, :].unsqueeze(0) +
emissions[:, 0]
    backpointers = []
    for i in range(1, seq_length):

```

```

alpha_t = []
backpointers_t = []
for tag in range(nb_labels):
    # get the emission for the current tag and broadcast to all labels
    e_scores = emissions[:, i, tag]
    e_scores = e_scores.unsqueeze(1)
    # transitions from something to our tag and broadcast to all
batches
    t_scores = self.transitions[:, tag]
    t_scores = t_scores.unsqueeze(0)
    # combine current scores with previous alphas
    scores = e_scores + t_scores + alphas
    # so far is exactly like the forward algorithm,
    # but now, instead of calculating the logsumexp,
    # we will find the highest score and the tag associated with it
    max_score, max_score_tag = torch.max(scores, dim=-1)
    # add the max score for the current tag
    alpha_t.append(max_score)
    # add the max_score_tag for our list of backpointers
    backpointers_t.append(max_score_tag)
# create a torch matrix from alpha_t
# (bs, nb_labels)
new_alphas = torch.stack(alpha_t).t()
# set alphas if the mask is valid, otherwise keep the current values
is_valid = mask[:, i].unsqueeze(-1)
alphas = is_valid * new_alphas + (1 - is_valid) * alphas
# append the new backpointers
backpointers.append(backpointers_t)
# add the scores for the final transition
last_transition = self.transitions[:, self.EOS_TAG_ID]
end_scores = alphas + last_transition.unsqueeze(0)
# get the final most probable score and the final most probable tag
max_final_scores, max_final_tags = torch.max(end_scores, dim=1)

```

Витерби алгоритми

Демак, энг эҳтимолли кўрсаткичлар кетма-кетлигини аниқлаш учун тескари алгоритмни ҳисоблаш лозим бўлмайди, чунки тўғри алгоритмда ҳар бир вақт қадами давомида максимал баллни кузатишнинг ўзи етарлидир. Сўнгра тугаганидан кейин max (argmax) амаллари буйича тескари йўналишда балларни максималлаштирувчи кетма-кетликни дешифрлаш мумкин бўлади. Қуйидаги код шуни амалга оширади :

```
def decode(self, emissions, mask=None):
    """Find the most probable sequence of labels given the emissions
using
the Viterbi algorithm.
Args:
    emissions (torch.Tensor): Sequence of emissions for each label.
        Shape (batch_size, seq_len, nb_labels) if batch_first is True,
        (seq_len, batch_size, nb_labels) otherwise.
    mask (torch.FloatTensor, optional): Tensor representing valid
positions.
    If None, all positions are considered valid.
    Shape (batch_size, seq_len) if batch_first is True,
    (seq_len, batch_size) otherwise.
Returns:
    torch.Tensor: the viterbi score for the for each batch.
        Shape of (batch_size,)
    list of lists: the best viterbi sequence of labels for each batch.
    """
    if mask is None:
        mask = torch.ones(emissions.shape[:2], dtype=torch.float)
    scores, sequences = self._viterbi_decode(emissions, mask)
    return scores, sequences
def _viterbi_decode(self, emissions, mask):
    """Compute the viterbi algorithm to find the most probable sequence
of labels
```

given a sequence of emissions.

Args:

emissions (torch.Tensor): (batch_size, seq_len, nb_labels)

mask (Torch.FloatTensor): (batch_size, seq_len)

Returns:

torch.Tensor: the viterbi score for the for each batch.

Shape of (batch_size,)

list of lists of ints: the best viterbi sequence of labels for each batch

"""

```
batch_size, seq_length, nb_labels = emissions.shape
```

```
# in the first iteration, BOS will have all the scores and then, the max
```

```
alphas = self.transitions[self.BOS_TAG_ID, :].unsqueeze(0) +
```

```
emissions[:, 0]
```

```
backpointers = []
```

```
for i in range(1, seq_length):
```

```
    alpha_t = []
```

```
    backpointers_t = []
```

```
    for tag in range(nb_labels):
```

```
        # get the emission for the current tag and broadcast to all labels
```

```
        e_scores = emissions[:, i, tag]
```

```
        e_scores = e_scores.unsqueeze(1)
```

```
        # transitions from something to our tag and broadcast to all
```

```
batches
```

```
        t_scores = self.transitions[:, tag]
```

```
        t_scores = t_scores.unsqueeze(0)
```

```
        # combine current scores with previous alphas
```

```
        scores = e_scores + t_scores + alphas
```

```
        # so far is exactly like the forward algorithm,
```

```
        # but now, instead of calculating the logsumexp,
```

```
        # we will find the highest score and the tag associated with it
```

```
        max_score, max_score_tag = torch.max(scores, dim=-1)
```

```
        # add the max score for the current tag
```

```
        alpha_t.append(max_score)
```

```

    # add the max_score_tag for our list of backpointers
    backpointers_t.append(max_score_tag)
# create a torch matrix from alpha_t
# (bs, nb_labels)
new_alphas = torch.stack(alpha_t).t()
# set alphas if the mask is valid, otherwise keep the current values
is_valid = mask[:, i].unsqueeze(-1)
alphas = is_valid * new_alphas + (1 - is_valid) * alphas
# append the new backpointers
backpointers.append(backpointers_t)
# add the scores for the final transition
last_transition = self.transitions[:, self.EOS_TAG_ID]
end_scores = alphas + last_transition.unsqueeze(0)
# get the final most probable score and the final most probable tag
max_final_scores, max_final_tags = torch.max(end_scores, dim=1)

```

Бу алгоритм Витерби алгоритми номи билан маълум. Бу `log_partition` функциясида қўллаган тўғри алгоритмга жуда ўхшаш, бироқ барча кетма-кетликлар учун доимий баллар ўрнига, максимал балл ва уларни максималлаштирувчи кўрсаткичлар мавжуд. Бошқача қилиб айтганда, факториал `logsumexp` ўрнига факториал `.Max` ни, яъни максимал ва `argmax` ни қайтарувчи амални қўлладик.

Буларнинг барчаси сўнгги кўрсаткичларни олиб, тескари кетма-кетликка ўтиш орқали и “`argmax`” ни аниқлаш ҳисобланади. Айтиб ўтилганлар қуйидаги кўриниш кодига эга бўлади:

```

# find the best sequence of labels for each sample in the batch
best_sequences = []
    emission_lengths = mask.int().sum(dim=1)
    for i in range(batch_size):
        # recover the original sentence length for the i-th sample in the
batch
        sample_length = emission_lengths[i].item()

```

```

# recover the max tag for the last timestep
sample_final_tag = max_final_tags[i].item()
# limit the backpointers until the last but one
# since the last corresponds to the sample_final_tag
sample_backpointers = backpointers[: sample_length - 1]
# follow the backpointers to build the sequence of labels
sample_path = self._find_best_path(i, sample_final_tag,
sample_backpointers)
# add this path to the list of best sequences
best_sequences.append(sample_path)
return max_final_scores, best_sequences
def _find_best_path(self, sample_id, best_tag, backpointers):
    """Auxiliary function to find the best path sequence for a specific
sample.
    Args:
        sample_id (int): sample index in the range [0, batch_size)
        best_tag (int): tag which maximizes the final score
        backpointers (list of lists of tensors): list of pointers with
shape (seq_len_i-1, nb_labels, batch_size) where seq_len_i
represents the length of the ith sample in the batch
    Returns:
        list of ints: a list of tag indexes representing the bast path
    """
# add the final best_tag to our best path
best_path = [best_tag]
# traverse the backpointers in backwards
for backpointers_t in reversed(backpointers):
    # recover the best_tag at this timestep
    best_tag = backpointers_t[best_tag][sample_id].item()
    # append to the beginning of the list so we don't need to reverse it
later
    best_path.insert(0, best_tag)
return best_path

```

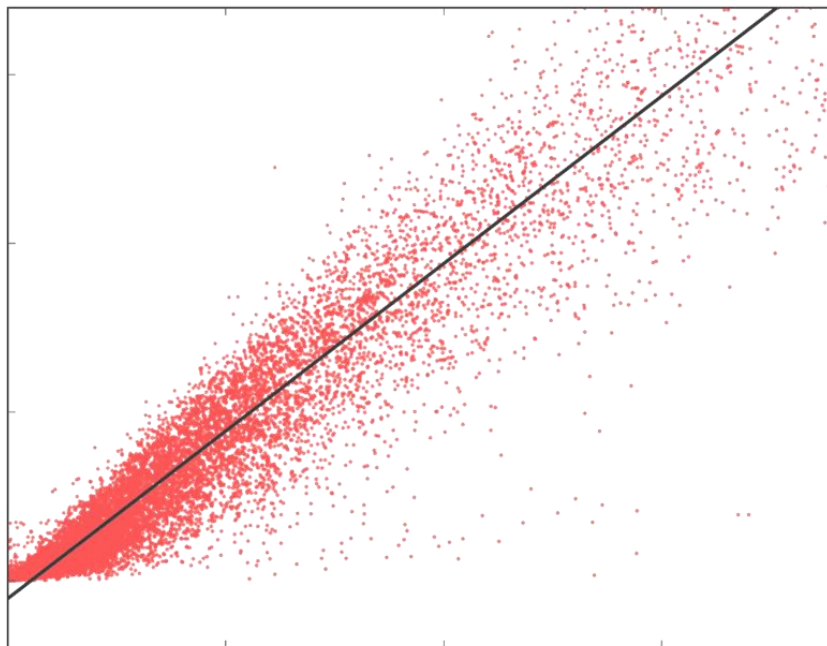
Эътибор қилинг, ҳар бир намуна учун `backtrace` такрорлаб, ҳар бир вақт қадами учун кўрсаткич қўйилган ҳолда `best_path` ни бошидан максималлаштиради. Натижада биринчи элемент биринчи кўрсаткичга мос келувчи рўйхат ҳосил қилинади.

Барча намуналар учун `find_best_path` амалини бажариш билан ишни тугатамиз.

Хулоса

Агар CRF моделини ишлаб чиқаришда қўлламоқчи бўлсангиз, уни текширишдан ўтказилган ва самарали амалга оширилган кўринишини масалан, `pytorch` пакетини ёки `allennlp` кутубхонаси таклиф қилган моделини қўллашни тавсия қилинади.

2.4.Регрессион алгоритмлар



83-расм. Регрессион алгоритмлар.

Энг кичик квадратларнинг доимий регрессияси (OLSR) – чизиқли регрессия усули бўлиб, модел яратиш асосида номаълум параметрларни баҳоланишидир, бунда кузатилаётган ва башорат қилинган берилганлар ўртасида хатолик квадратлари суммасини минималлаштиради (кузатилаётган қийматлар ва баҳолаётган қийматлар).

Чизиқли регрессия- реал қийматларни баҳолашда қўлланилади. (уйлар нархи, кўнғироқлар сони, савдо умумий ҳажми ва ҳ.к.), Узлуксиз ўзгарувчи асосида амалга оширилади.

Логистик регрессия- дискрет қийматларни баҳолаш учун қўлланилади (иккилик қийматлар, масалан, 0/1, ҳа /йўқ, рост/ ёлғон), эркин ўзгарувчилар асосида амалга оширилади.

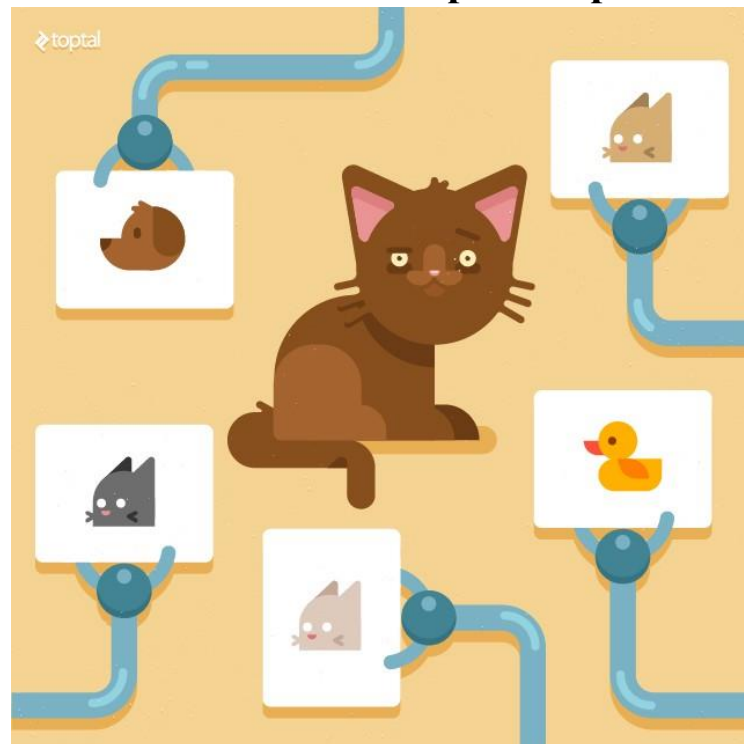
Қадамли регрессия- токи функциялар бўйича оптимал баллга эришмагунча моделга бирма-бир функцияларни қўшиб боради. Қадамли танлаш тўғри ва тескари йўналишда бирма-бир бажарилиб, ўзгарувчиларни киритиш ва ўчириш амаллари ўзгарувчиларнинг барқарор тўпламига эришилгунча кетма-кет бажарилади.

Адаптив регрессиянинг кўп омилли сплайни (MARS) – регрессиянинг мослашувчан усули бўлиб, нозизиқ муносабатларни ва ўзаро боғланишларни излайди, бу эса ўз навбатида башорат аниқлик кўрсаткичини максималлаштиради. Ушбу алгоритмлар нозизиқ ҳисобланиб, моделни нозизиқ шаблонларга мослаштиришни талаб этмайди.

Тарқоқ диаграммани локал баҳолаш орқали текислаш (LOESS)- икки ўзгарувчи орасида текис чизиқни танлаш ёки башорат қилинган тўрттагача ўзгарувчилар бўйича текис сиртни танлаш усули. Бунда агар берилганлар чизиқли тақсимотга эга бўлмаса, регрессия усулини қўллашингиз мумкинлигини билдиради. Қўллайдиган регрессия локал-белгиланган регрессия деб номланади. Боғланган ва боғлиқсиз ўзгарувчилар ўртасида нозизиқ муносабат бўлган ҳолатларда ҳам LOESS ни қўллаш мумкин. Ҳозирги кунга келиб кўпгина алгоритмлар (масалан, тўғри боғланишга эга классик нейрон тармоқ, таянч векторлар машиналари, яқин қўшни берилганлар алгоритмлари ва ҳ.к.) глобал ўрганиш тизимини ҳосил қилади. Бунда улар глобал йўқотиш функциялари (масалан, сумма квадрати хатолиги)ни қисқартиришда қўлланилади. Локал ўрганиш тизимлари ўрганиш глобал муаммосини бир нечта соддароқ муаммоларга бўлиб чиқади. Глобал усулнинг камчиликларидан бири бу параметрнинг ҳеч қандай қиймати етарлича яқинлашиш кўрсаткичини таъминламайди.

Бироқ LOESS ни қўллаш -глобал функцияга яқинлашиш кўрсаткичи альтернативини ҳосил қилади.

2.5. Ансамбль алгоритмлар



84-расм. Ансамбль алгоритми.

Ансамбль усули- ўрганиш алгоритмлари бўлиб, классификаторлар тўпламини яратади, сўнгра янги берилганларни башорат эҳтимоллиги бўйича таснифлайди. Ансамблнинг бошланғич усули байес ўртача усули бўлиб, охиридаги алгоритмлари эса чиқувчи маълумотлар хатоликларини тўғрилаш, суммалаш ва оптималлаштириш ишларини ўз ичига олади.

Boosting- алгоритмлар синфи бўлиб, заиф ўқувчиларни кучли ўқувчилар учун стимуллаштиради (яъни аниқ тасниф асосида тўғриловчи классификатор, янги берилганлар учун тахминлар ўрнига аниқ кўрсатмаларни белгилайди). Ансамблнинг бу усули, жорий ўрганиш алгоритмининг барча моделини башорат қилишда аниқликни ошириш имконини беради. Бу усул дарахт кетма-кетлиги (тасодифий танлов) учун мос келиб, ҳар бир босқичдаги мақсад бу олдинги дарахтдаги аниқ хатоликни топиш ҳисобланади. Энг аввал олдиндан тахмин қилиш ва ўрганиш жараёнидаги дисперсияни баргараф этиш

учун қўлланилади. Одатда бир нечта баҳолаш кўрсаткичи асосида умумий башорат қилиш имконини бериб, битта баҳолаш кўрсаткичига нисбатан ишончилиликни оширади (бир нечта заиф ёки ўрта ҳол предикторларни бирлаштириб битта кучли предиктор яратиш имконини беради).

Bootstrapped Aggregation (Bagging) –одатда ечим дарахтида дисперсия кўрсаткичини камайтириш ҳолларда қўлланилади. Асосий ғоя бу ўрганиш танловидан тасодиқий танланган берилганлар асосида тўпламларни яратиш бўлиб, ҳар бир тўплам ечимлар дарахтини ўрганиш учун қўлланилади. Натижада турли моделлар ансамблига эга бўламиз. Турли ечим дарахтлари бўйича башоратлар ўртача қиймати қўлланилиб, битта ечим дарахтига нисбатан ишончлироқ ҳисобланади.

AdaBoost - қисқа ечим дарахти билан бирга қўлланилиб, биринчи дарахт яратилгандан сўнг, ҳар бир ўрганиш экземплярда ечим дарахти унумдорлиги кўрсаткичи аниқлаб олинади ва у асосида кейинги яратиладиган ечим дарахти ҳар бир ўрганиш экземплярига таъсир даражаси аниқланади. Башорат қилиниши мушкул бўлган берилганлар оғирлик кўрсаткичи катта, осон башорат қилинувчи ҳоллар эса кичик оғирлик кўрсаткичига эга бўлади. Моделлар бирин-кетин яратилиб, уларнинг ҳар бири оғирлик кўрсаткичлари коэффициентларни ўрганиш экземпляр учун янгилайди, бу кўрсаткичлар кетма-кетликдаги кейинги дарахт ўрганиш жараёнига таъсир ўтказади. Барча ечим дарахтлари яратилгандан сўнг, янги берилганлар учун башоратлар қилиниб, ҳар бир дарахнинг унумдорлигини ўрганиш жараёнидаги кўрсаткич бўйича баҳоланади.

Йиғилган, умумлаштирилган (аралаштириш) -конволюция, аралаштириш ва умумлаштириш турли номли бир хил жараён ҳисобланади. Бу процедуралар башорат аниқлигини ошириш учун мўлжалланган бўлиб, бир нечта машинали ўрганиш моделини бирлаштириш ёки комбинациялаш орқали амалга оширилади. Мантиқан бу янги модель ўрганиш жараёнидан ўтувчи ансамблли алгоритмлар бўлиб, икки ва ундан ортиқ моделлар ёки берилганлар тўплами башоратини умумлаштиради.

Gradient Boosting Machines (GBM)- стимуллаштириш асосида кенгайтириш усули. Градиентни ошириш = градиент кўрсаткичи + кучайтириш. Кучайтирилган алгоритм бўлиб, катта ҳажмдаги берилганлар билан ишлашда қўлланилиб, аниқ башорат қилиш учун қўлланилади.

Стимуллаштирилган градиентли регрессияга эга дарахтлар (GBRT) - тасниф ва регрессиянинг мослашувчан параметрсиз усули. Башоратнинг суст моделлари ансамбли кўринишдаги моделни, одатда ечим дарахтларини яратади. GBRT одатда моделни босқичма-босқич яратиб, уларни умумлаштиради ва тасодифий дифференциаллашган йўқотиш функциясини оптималлаштириш имконини беради.

Random Forest- ечим дарахтлари ва пакетларда кенгайтмалар алгоритми ансамбли. Эътибор беринг, ечимлар дарахти тўплами «ўрмон» деб номланади. Яна бир қадамни бажаришни талаб этиб, бунда берилганларнинг тасодифий тўплами билан бирга объектларни ҳам тасодифий танлайди, яъни дарахт яратилишида барча объектлар қўлланилмайди. Янги объектни атрибутлар асосида таснифлаш учун ҳар бир дарахт таснифни аниқлайди ва бу дарахт шу синфга “овоз беради” деб айтилади. “Ўрмон” энг кўп “овоз” олган таснифни танлайди.

Яндекс CatBoost технологияси

Бугунги кунда Яндекс Open Source-да ўзининг CatBoost кутубхонасини яратди, бу компаниянинг машинали ўрганиш соҳасидаги кўп йиллик тажрибасини ҳисобга олган ҳолда ишлаб чиқилган. Унинг ёрдами билан сиз моделларни турли-туман маълумотларга, шу жумладан рақамлар шаклида (масалан, булутлар турлари ёки товарлар тоифаси) тасаввур қилиш қийин бўлган моделларга самарали ўргатишингиз мумкин. Манба коди, ҳужжатлар, бенчмарклар ва керакли воситалар аллақачон GitHub да Apache 2.0.лицензияси остида чоп этилган.

CatBoost-бу градиент Boostingга асосланган янги машинали ўрганиш усули. Ранжирлаш, башорат ва тавсиялар куриш муаммоларни ҳал қилиш учун Яндексга жорий этилади. Бундан

ташқари, у аллақачон Европа ядровий тадқиқотлар ташкилоти ва Yandex Data Factory саноати мижозлари билан ҳамкорлик доирасида қўлланилмоқда. Хўш, CatBoost бошқа очиқ аналоглардан қандай фарқ қилади?

"Машинали ўрганиш" атамаси 50-ларда пайдо бўлди. Бу атама компьютерга инсонга осонлик билан берилган муаммоларни ҳал қилишни ўргатишга уринишни англатади, аммо уларни ҳал қилиш йўлини расмийлаштириш қийин. Машинали ўрганиш натижасида компьютер аниқ кўрсатилмайдиган ҳатти-ҳаракатларни намойиш қилиши мумкин. Ҳозирги кунда кўпчилигимиз буни билмаган ҳолда кўп мартаба машинали ўрганиш асосидаги самарали ютуқларга дуч келамиз, Ижтимоий тармоқларда кузатувли боғланган маълумотлар тўпламини яратиш, онлайн-дўконларда "ўхшаш товарлар" рўйхатлари, банкларда кредитлар бериш ва пул айланмаси ҳисоботини, инсон мойиллик даражасини аниқлашда ишлатилади. Машинали ўрганиш технологиялари фотосуратларда ёки кўплаб фотосуратларда шахсларни қидиришни амалга оширади. Иккинчидан, нейрон тармоқлари ҳаётда ишлатилади ва изланишлар олиб борилмоқда, бу эса ҳар қандай мураккабликдаги муаммоларни ҳал қилиш учун нотўғри фикр бўлиши мумкин.

Нейрон тармоқлари ёки градиент Boosting

Аслида, машинали ўрганиш кўринишлари жуда кўп бўлиб, турли хил усуллар қўлланилади ва нейро тармоқлари улардан фақат биттаси. Бунинг тасдиғи Kaggle платформасида турли танловларда турли усуллар рақобатлашган, бироқ жуда кўп ҳолларда градиент boosting ғалаба қозонган.

Нейрон тармоқлари муайян вазифаларни мукамал ҳал қилади - масалан, бир хил маълумотлар билан ишлаш керак бўлганда, ёки тасвир, овоз, матндан иборат маълумотлар билан ишлаганда самарали ҳисобланади. Яндексда улар бизга қидирув сўровларини яхшироқ тушунишга, интернетда шунга ўхшаш расмларни қидиришга, навигаторда овозингизни тан олишга ва яна кўп нарсаларга ёрдам беради. Лекин бу машинали ўрганиш учун барча вазифалар эмас.

Фақат нейрон тармоқлар томонидан ҳал етилмайдиган жиддий муаммолар мавжуд – улар градиент boosting га муҳтож. Бу усул жуда кўп маълумотлар мавжуд бўлган ва уларнинг тузилиши турли бўлган ҳолларда самарали ҳисобланади.

Мисол учун, кўплаб омиллар (ҳарорат, намлик, радар маълумотлари, фойдаланувчи кузатувлари ва бошқалар) ҳисобга олинмайдиган аниқ об-ҳаво башоратига муҳтож бўлсангиз. Ёки қидирув натижаларини сифат жиҳатидан тартибга солиш керак бўлганлиги, бу ўз вақтида Яндексни ўз машинали ўрганиш усулини ишлаб чиқишга ундаган.

Матрикснет

Биринчи қидирув тизимларидан бири бўлиб, ҳозирги тизимлар каби эмас, балки соддароқ эди. Аслида, биринчи кўринишдаги излаш тизимида сўзлар асосида сайтлардан излаш тизими жуда оз бўлиб, улар орасида ҳеч қандай рақобат мавжуд бўлмаган. Кейин саҳифалар кўпайиб кетиб, улар рейтингга муҳтож бўлиб қолди. Турли хил асоратлар -сўзларнинг частотаси, tf-idf ҳисобга олинди. Кейин ҳар қандай мавзуда жуда кўп саҳифалар бор эди, биринчи муҳим ютуқ – улар ҳаволаларни ҳисобга олишни бошладилар.

Кўп ўтмай, Интернет тижорат жиҳатдан муҳим бўлиб қолди ва ўша пайтда мавжуд бўлган оддий алгоритмларни алдашга уринаётган кўплаб ёлғончилар пайдо бўлди. Ва иккинчи муҳим ютуқ бор эди - қидирув тизимлари қайси саҳифалар яхши эканини тушуниш учун фойдаланувчиларнинг ҳатти -ҳаракатлари ҳақида ўз билимларидан фойдалана бошладилар.

Ўн йил муқаддам, инсон ҳужжатларни тартибга солиш масаласини ҳал этди. Ҳар қандай сўров жуда кўп: юз минглаб, кўпинча миллионлаб натижалар бериши мумкин. Уларнинг аксарияти қизиқ эмас, фойдасиз, фақат сўров сўзларини тасодифан эслатиб туради ёки одатда спам ҳисобланади. Сизнинг сўровингизга жавоб бериш учун сиз топилган барча натижалардан ўнга энг яхшисини дарҳол танлашингиз керак бўлади. Буни эса мақбул сифат билан амалга оширадиган дастурни ёзиш дастурчи учун мураккаб вазифа бўлиб

қолди. Кейинги ўтиш жараёни содир бўлди – қидирув тизимларида машинали ўрганишдан фаол фойдалана бошланди.

Яндекс 2009-да градиент Boostingга асосланган Матрикснет усулини жорий қилди. Фойдаланувчиларнинг умумий фикри ва "инсонларнинг донолиги" бу тизимда рейтинг ёрдам бериши мумкин. Веб-сайтлар ва одамларнинг ҳатти-ҳаракати ҳақидаги маълумотлар турли хил омилларга айланади, уларнинг ҳар бири рейтинг формуласини яратиш учун матрицадан фойдаланилади. Аслида, тартибланиш формуласи энди машинали кўринишда ёзилади. Айтганча, айрим омиллар сифатида биз нейрон тармоқларнинг натижаларини (масалан, ўтган йили айтилган Палек алгоритми) ишлатамиз.

Матрицанинг муҳим хусусияти шундаки, у қайта ўқишга чидамли. Бу сизга кўплаб рейтинг кўрсаткичларини ҳисобга олиш имконини беради ва айна пайтда машинали мавжуд бўлмаган нақшларни топишида кам вақт сарфлаб маълумот олиш имконини беради. Бошқа машинали ўрганиш усуллари камроқ омилларга боғлиқ ҳолда боғланишларни аниқлашга ёки кўпроқ намуна – яъни ўргатишга асос бўлган маълумотларни талаб қилиши мумкин бўлади.

Матрицанинг яна бир муҳим хусусияти шундаки, рейтинг формуласи жуда тор сўровлар синфлари учун алоҳида созланиши мумкин. Мисол учун, фақат мусиқа талаблари бўйича қидирув сифатини самаралаштириш. Шу билан бирга, сўровларнинг қолган синфлари рейтингини туширмайди.

Градиент Boostingга асосланган деярли ҳар қандай замонавий усул рақамлар билан ишлайди. Киришда мусиқа жанрлари ёки ранглар мажмуи кўринишда мавжуд бўлса ҳам, бу маълумотлар ҳали ҳам рақамлар тилида тасвирланган бўлиши керак. Бу уларнинг моҳиятини нотўғри талқин қилишга ва моделнинг аниқлигини пасайишига олиб келади.

Буни дўконда маҳсулот каталоги билан оддий мисолда намоиш этамиз. Маҳсулотлар бир-биридан анча кўп фарқланиб ва улар орасида тартибланган боғланиш мавжуд эмас, бу уларни тартибга солиш ва ҳар бир маҳсулотга мазмунли рақамни белгилаш имконини

беради. Шунинг учун, бу ҳолатда, ҳар бир маҳсулотга оддий ID берилади. Бу рақамларнинг тартиби ҳеч нарса демайди, лекин алгоритм бу тартибни ишлатади ва ундан нотўғри хулосалар чиқаради.

Машинали ўрганиш билан шуғулланадиган тажрибали мутахассис аниқ хусусиятларни рақамга айлантиришнинг янада интеллектуал усулини таклиф қилиши мумкин, аммо дастлабки ишлов бериш ахборотнинг бир қисмини йўқотишга ва якуний ечим сифатининг ёмонлашишига олиб келади.

Шунинг учун машинали ўрганиш фақат рақамлар билан эмас, балки тўғридан-тўғри тоифалар билан ишлашни ўрганиши муҳим эди, улар томонидан мустақил равишда аниқланади. CatBoost ҳам рақамли белгилар ҳам категориялар билан бир хил даражада яхши ишлаши учун мўлжалланган. Шу билан бирга, муқобил ечимларга нисбатан, альтернатив маълумотлар билан ишлаш юқори сифат натижа беради. У банк соҳасидан бошлаб, ишлаб чиқаришга қадар турли соҳаларда қўлланилиши мумкин. Бу технология номи Categorical Boosting сўзидан келиб чиқган.

Бенчмарклар

Кутубхонанинг назарий фарқлари ҳақида узоқ вақт гапириш мумкин, аммо амалда бир марта кўрсатиш яхшироқдир. Аниқлик учун CatBoost кутубхонасининг ишини XGBoost, LightGBM и H2O очиқ аналоглари билан очиқ ахборот марказларида солиштирилди.

CatBoost амалда

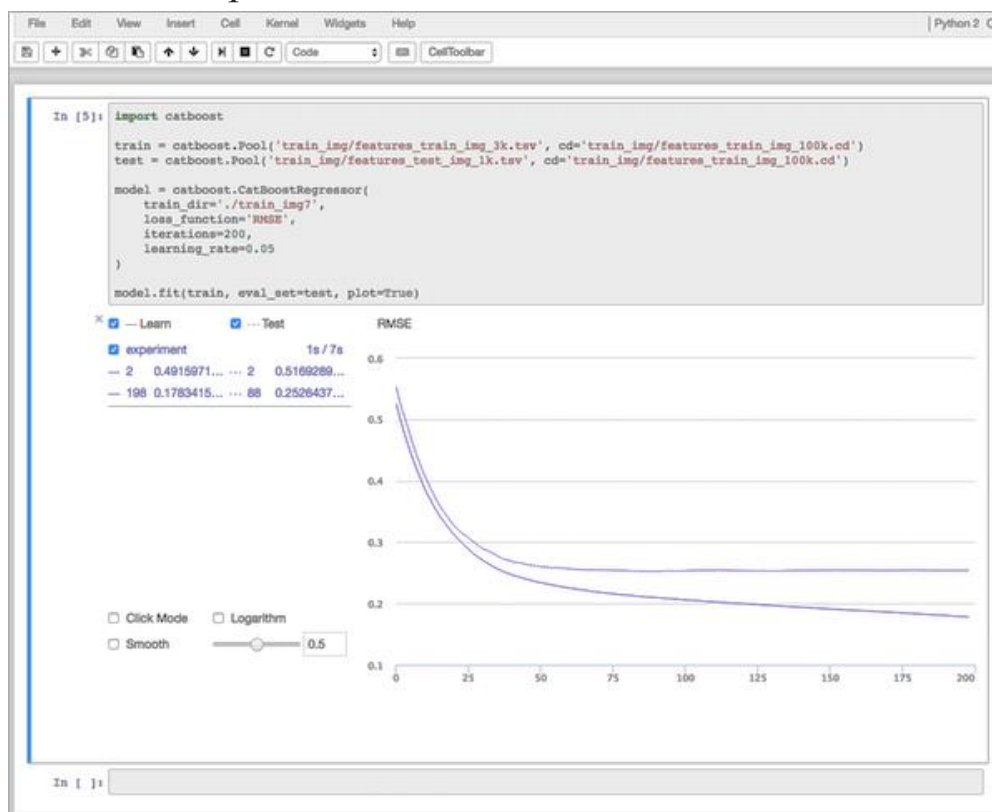
Янги усул аллақачон Яндекс хизматларида синовдан ўтган. У кидирув натижаларини яхшилаш, Яндекс Дзен тавсияларини кўрсатиш учун ишлатилган ва Метиум технологиясидаги об - ҳаво башоратини ҳисоблаш учун қўлланилиб, барча ҳолатларда Матрикснетдан яхшироқ натижа кўрсатди. Келажакда CatBoost бошқа соҳаларда ҳам тадбиқ этилади.

CatBoost Европа ядровий тадқиқотлар ташкилоти билан ҳамкорлик доирасида ҳам фойдаланишга муваффақ бўлди. Катта адрон коллайдерда LHCb детектори мавжуд бўлиб, у оғир кварцларнинг

ўзаро таъсирларида материя ва анти-материя ассиметриясини ўрганиш учун ишлатилади. Экспериментда қайд этилган турли зарраларни аниқ кузатиб бориш учун детекторда бир нечта махсус қисмлар мавжуд бўлиб, уларнинг ҳар бир заррачаларнинг махсус хусусиятларини аниқлайди. Бу ҳолда энг қийин вазифа детекторнинг турли қисмларидан заррача борасида маълумотларни энг аниқ, умумий маълумот кўринишда бирлаштиришдир. Бу ерда машинали ўрганиш айна муддао бўлиб қолади. CatBoost маълумотларини бирлаштириш учун олимлар якуний ечимнинг сифат кўрсаткичларини яхшилашга муваффақ бўлишди. CatBoost натижалари бошқа усуллар ёрдамида олинган натижалардан яхшироқ эканлиги исботланган.

CatBoost-дан қандай фойдаланиш керак?

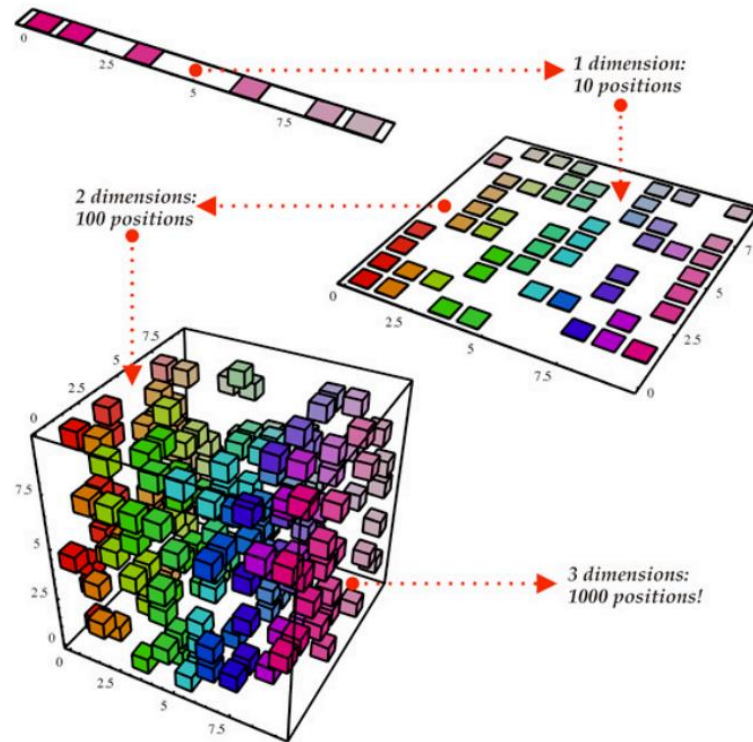
CatBoost билан ишлаш учун уни компьютерингизга ўрнатиш кифоя. Кутубхона Linux, Windows ва macOS операцион тизимларини қўллаб-қувватлайди ва Python R. дастурлаш тилларида мавжуд. Шу билан бирга Яндекс визуаллаштириш дастури CatBoost Viewerни ишлаб чиқди, у ўз навбатида графиклар асосида ўрганиш жараёнини кузатиш имконини беради.



85-расм. CatBoost-ўрганиш жараёнини графика кузатиш.

CatBoost- OPEN SOURCE -кўринишда мавжуд бўлган биринчи рус тилидаги машинали ўрганиш технологиясидир. Кутубхонани очик кодли кўринишда жойлаштиришга сабаб машинали ўрганиш соҳаси ривожига йўл очиб бериш ҳисобланади.

2.6. Ўлчамни кичрайтириш алгоритмлари



86-расм. Ўлчамни кичрайтириш алгоритми:

Ўлчамларни камайтириш – ўлчамларни камайтириш алгоритми бу бошқа алгоритмлар ечимлар дарахти, тасодифий “Ўрмон”, РСА ва факторли таҳлил асосидаги ўрганилаётган тасодифий ўлчамларни камайтириш имконини беради

Асосий компонентлар таҳлили (РСА) - статистик процедура бўлиб, корреляцияланган кузатувлар тўпламини чизикли корреляция қилинмаган асосий компонентлар деб номланган ўзгарувчилар тўпламига ўгириш. Биринчиси энг муҳим компонент, сўнг иккинчи, учинчи ва ҳ.к. давом этади.

Эркин компонентли таҳлил (ІСА)- тасодифий ўлчамлар, катталиклар ёки сигналлар тўплами негизда ётган яширин омилларни аниқлашнинг статистик усули.

Асосий компонентлар регрессияси (PCR) – мультиколлинеарлик мавжуд берилганлар тўпламининг регрессиясини таҳлил қилиш усули. Асосий ғоя, бу асосий компонентларни аниқлаш, сўнгра бу компонентларнинг баъзиларини чизиқли регрессия моделида предиктор кўринишда қўллаш, энг кичик квадратлар усулини қўллаш асосида амалга оширилади.

Энг кичик квадратларнинг қисман регрессияси (PLSR) -PCR одатда предиктор ўзгарувчилар ўзгаришини изоҳловчи компонентни яратади, бунда натижа қийматлари инобатга олинмайди. PLSR эса натижа қийматларини инобатга олган ҳолда кам сонли компонентлар асосида модель яратишга олиб келади.

Sammon Mapping - алгоритм, кўп ўлчовли муҳитни кам ўлчовли муҳит кўринишда ифодалаб, кўп ўлчовли фазо нуқталари ўртасидаги масофалар структурасини сақлаган ҳолда кам ўлчовли фазода ифодалайди. Бу эса маълум бир берилганлар тўпламини нозик ифодалашнинг оптимал кўринишини танлаш имконини беради. Гарчи PCA дисперсияни оддийгина максималлаштирсада, баъзан бошқа бир ўлчамни максималлаштириш талаб этилиши мумкин, бунда ўзгартиришдан кейин ҳам мураккаб структура сақланиб қолади. Шундай ўлчамлар турли бўлиб, улардан бири Саммон харитаси. Бу асосан берилганларни бошланғич таҳлилида қўллашга тўғри келади.

Кўп ўлчовли масштаблаш (MDS) - берилганлар тўпламининг алоҳида ҳоллар учун ўхшашлик даражасини визуаллаштириш воситаси.

Projection Pursuit – статистик усул кўриниши бўлиб, кўп ўлчовли берилганлар ичида энг “аҳамиятли” ларни излаш амалини ўз ичига олади. Кўпинча тўғри тақсимотдан четлашган ҳолатлар энг аҳамиятли ҳисобланади.

Чизиқли дискриминант таҳлил (LDA) –агар тасниф алгоритми керак бўлса, энг аввал логистик регрессиядан бошлаш лозим. Бироқ, LR одатда синфлар таснифининг иккита муаммоси билан чекланиб қолади. Агар масала иккитадан ортиқ синфларга эга бўлса, LDA қўллаш лозим. LDA ўлчамни кичрайтириш алгоритми каби ҳам

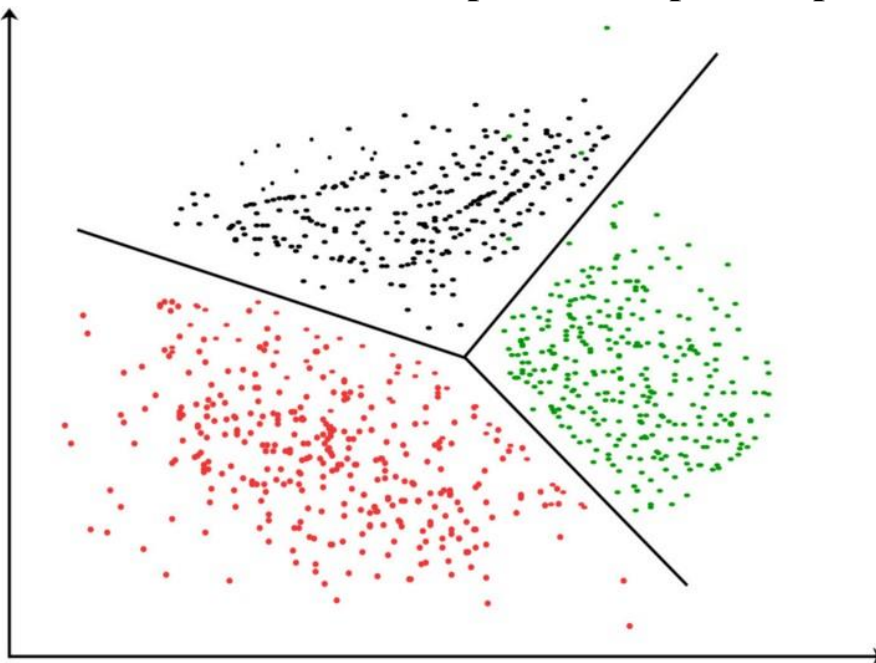
ишлайди, бунда ўлчамлар сонини оригинал ҳолатдан $G-1$ гача кичрайтиради, бунда G –синфлар сони

Аралашма дискриминант таҳлили (MDA) –чизиқли дискриминант таҳлилнинг кенгайтирилган ҳолати. Таснифнинг бошқариладиган усули бўлиб, аралаш моделларга асосланган.

Квадрат дискриминант таҳлил (QDA) -Чизиқли дискриминант таҳлил фақат чизиқли чегараларни ўрганиши мумкин. Квадрат дискриминант таҳлил эса квадратик чегараларни ўрганади (шунинг учун мослашувчан). Бироқ, LDA дан фарқли равишда, QDA да ҳар бир синф ковариацияси бир хил деган тахмин мавжуд эмас.

Мослашувчан дискриминант таҳлил (FDA) – классификацион модель бўлиб, чизиқли регрессия моделлари мувофиқлашувига асосланган. Бунда, чизиқли тақсимот учун берилганлар аниқ кўрсатишда ва дискриминант сатҳ учун бир нечта регрессия сплайнлари мавжудлигини таъминлашда оптимал баҳолаш қўлланилади.

2.7.Кластерлаш алгоритмлари



87-расм. Кластерлаш алгоритми:

К-воситалар- KNN дан фарқланадиган алгоритм бўлиб (уларни адаштириш керак эмас), K бу X берилганларни K кластерга тақсимлаш

лозимлигини билдиради. Бунда ҳар бир берилган нуқтаси унга яқин бўлган кластерга боғланади. Асосий ғоя бу барча кластерлар учун кластерларда мавжуд масофалар квадратининг йиғиндисини қисқартириш ҳисобланади.

Бир каналли кластерлаш – иерархик кластерлаш усуллариининг бири ҳисобланиб, у кластерларни пастдан юқорига қараб гуруҳлашга асосланган. Битта боғланишли кластерлашда иккита кластерларнинг ўхшашлиги – бу кластерларда ўхшаш элементларининг кўплигини билдиради.

К-медианлар – К алгоритмнинг бир варианты ҳисобланади. Асосий ғоя ҳар бир кластер учун ўртача қийматни ҳисоблаш (унинг центроидини аниқлаш) ўрнига, медианани ҳисоблаймиз.

Максимал кутилиш (EM) – К ўртача қийматига аналог ҳисобланиб, кластерлар эҳтимолликни ифодалайдиган оғирлик кўрсаткичига эга кластерлар берилганларни ўзлаштиради. Устунлик томони шундаки, модел генерацияланиб боради, чунки ҳар бир модел учун эҳтимолликлар тақсимооти аниқланади.

Иерархик кластерлаш – берилганларни кластерлар тақсимлаш бир босқичда амалга оширилмайди. Унинг ўрнига бир нечта босқичлар қўлланилиб, барча берилганларга эга битта кластердан бошлаб, битта берилганга эга N кластер билан тугайди.

Норавшан кластерлаш – кластерлаш кўринишида ҳар бир берилган биттадан ортиқ кластерга боғлиқ.

DBSCAN (қўшимчали зичликликка асосланган фазовий кластерлаш) – юқори кўрсаткичли зичликка эга кластерларни паст кўрсаткичли зичликка эга кластерлардан ажратиш. DBSCAN учун иккита параметр талаб этилади: икки нуқта орасидаги энг минимал масофа ва текисликни шакллантиришда энг кичик зичлик кўрсаткичи. Бу бир-бирига яқин (одатда евклид масофа) минимал сонли нуқталарни гуруҳлаш деган маънони англатади .

ОПТИКА (Кластер структурасини идентификациялаш учун нуқталарни тартиблаш)- Асосий ғоя DBSCAN га ўхшаш бўлиб,

бирок унда мавжуд камчиликни бартараф этади: бу турли зичликка эга берилганлар ичида муҳим кластерларни аниқлаш масаласидир.

Манфий бўлмаган матрицали факторизация (NMF) - чизиқли - алгебраик модель бўлиб, катта ўлчамли векторларни кичик ўлчамли векторлар орқали ифодалаш кўринишига олиб келади. Асосий компонентлар таҳлили (PCA) га ўхшаш, аммо NMF векторлар манфий бўлмаслигини талаб этади. Уларни кичик ўлчамда жойлаштириганда коэффицентлар ҳам манфий бўлмаслигини талаб этади.

Дирихле латентли тақсимоти (LDA) – эҳтимоллик модели кўриниши бўлиб, корпусда мавжуд тематикани аниқлаш. Масалан, ҳужжатда тўпланган сўзлар кузатув объекти бўлса, кластерларни белгилаш учун иккита эҳтимоллик кўрсаткичи лозим бўлади: $P(\text{сўз} | \text{мавзу})$, берилган мавзуларда сўзнинг мавжудлик эҳтимоллиги. $P(\text{мавзулар} | \text{ҳужжатлар})$, берилган ҳужжатда мавзуларнинг мавжудлик эҳтимоллиги. Бу қийматлар бошланғич белгиланган тасодифий қийматлар асосида аниқланади. Сўнгра улар ҳар бир ҳужжатдаги ҳар бир сўз учун такрорланиб борилади, натижада улар мавзуси аниқланади

Гаусс аралаш модели (GMM). Асосий ғоя – бу кўп ўлчовли Гаусс эҳтимолликлар тақсимоти аралашмасини аниқлаш ҳисобланиб, улар ёрдамида барча кўринишдаги бошланғич берилганларни моделлаштириб бўлсин. Шу билан бирга k -means каби кластерларни аниқлашда ҳам қўллаш мумкин. Мақсад содда бўлиб, берилганларни аниқ ифодаловчи гауссли параметрларни топиш ҳисобланади.

2.8.К-оралиқда қўшни хусусиятларни қўллаш усулининг таснифлаш масалаларни ечишда қўлланилиши

k -оралиқда қўшни хусусиятларни қўллаш (англ.: *k-nearest neighbors method, k-NN*) усули - таснифлаш масалаларни ечиш усулларида бири ҳисобланади.

Фараз қилинсинки, аниқ таснифланган объектлар сони берилган (яъни ҳар бир объект мансуб бўлган синф маълум). Мақсад, янги киритилган объект мавжуд бўлган синфлар ичидан қайси бирига

мансуб эканлигини аниқловчи қоидани ишлаб чиқиши керак (бунда синфлар олдиндан аниқ ҳисобланади).

k-NN негизида қўйидаги тартиб амалга оширилади: *янги объект ўзининг қўшни объектларнинг кўпчилиги мансуб бўлган синфга мансуб деб ҳисобланади.* «Қўшни» объект деб ўрганилаётган объектга қайсидир маънода яқин бўлган объектлар тушунилади.

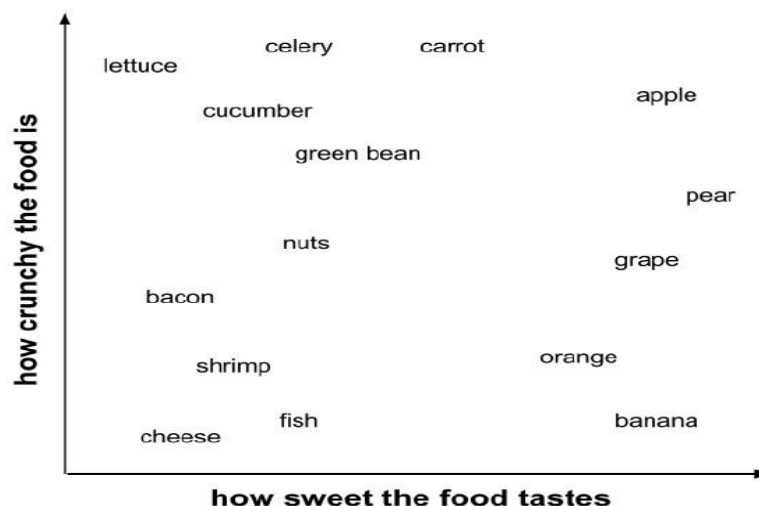
Бунда, объектлар ўртасидаги яқинлик даражасини, яъни улар ўртасидаги «масофани» аниқлаб билиш лозим бўлади. Бироқ бу евклид масофа бўлиши шарт эмас. Бунда объектлар ўлчов кўрсаткичи, масалан, ранги, шакли, таъми, ҳиди, (инсонлар ўртасида аниқлаш бўлган ҳолат учун) қизиқишлари, хулқ атвори ва бошқалар бўлиши мумкин. Табиийки, kNN усулини қўллаш учун маълум бир бирлик (яъни яқинлик функцияси) берилган бўлиши лозим.

Маълум бир хусусиятлари билан яқин бўлган объектлар бошқа хусусиятлари билан ҳам яқин бўлиши мумкин (яъни битта синфга мансуб бўлиши мумкин).

k-NN усулини содда мисолда кўриб чиқамиз.

Маҳсулот	Ширинлик	Зичлик	Синфи
яблоко	9		мева
бекон	1		протеин
банан	10		мева
...

Маҳсулот хусусиятлари 10 баллик шкалада баҳоланмоқда, бу кийматлар 2 ўлчовли текисликда координата нуқталари кўринишида жойлашади. Икки хусусияти иккита координата ўқлари бўйича белгиланиб, маҳсулотлар жойлашуви аниқланади, натижада қўйидаги расм ҳосил бўлади.



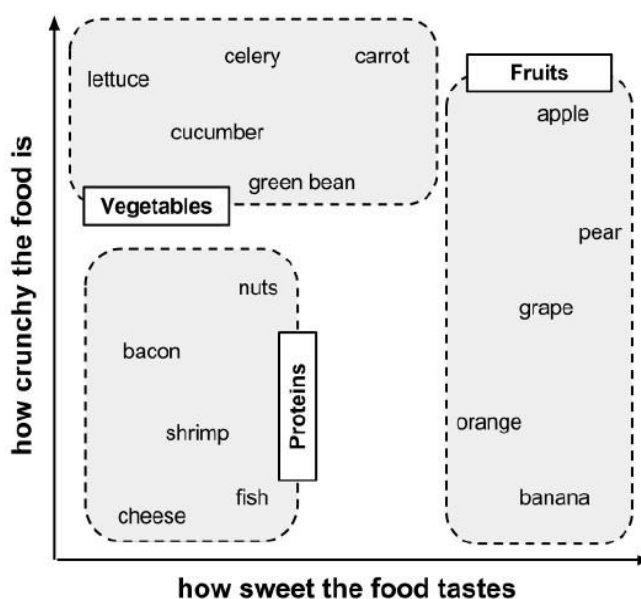
88-расм. Сўзларнинг 2 ўлчовли текисликда жойлашуви.

Кўриб ўтиладиган маҳсулотлар тури(синфланиши) устунлар кўрсаткичи бўйича кўрсатилган, графикда жойлашуви бўйича улар визуал синфлари яққол кўриниб туради:

– Чап томон юқорида жойлашган маҳсулотлар сабзавотлар «гуруҳлашган» (бодринг, сабзи, кўкат) – ширин бўлмаган, бироқ карсиллаш хусусиятига эга.

– Чап томон пастда –протеинга бой маҳсулотлар (бекон, қисқичбақа, пишлок, балиқ, ёнғоқ) –ширин бўлмаган, карсиллаш хусусиятига эга эмас.

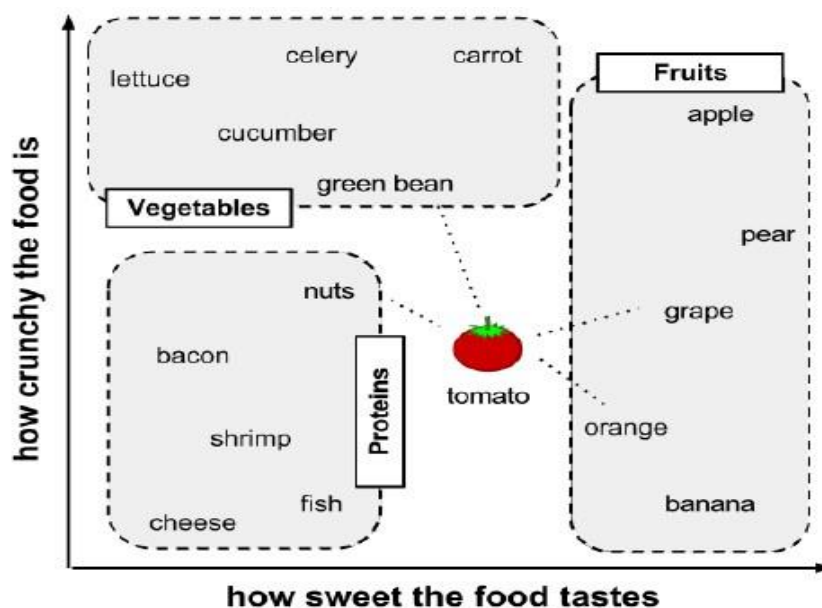
– Ўнг томондан мевалар «жойлашиб олган» (олма, нок, узум, апельсин, банан) – бошқа синфланишга нисбатан улар ширин, бироқ карсиллаш кўрсаткичи бўйича турли кўринишга эга.



89-расм. Маҳсулотлар гуруҳларга тақсимот схемаси.

№	Маҳсулот	Синфи	Помидорга яқинлик даражаси
1	апельсин	мева	1,4
2	узум	мева	2,2
3	қисқичбақа	протеин	3,5
4	бекон	протеин	3,6
5	ёнғоқ	протеин	3,6
6	Пишлоқ	протеин	4,0
7	Дуккаклилар	протеин	4,2
8	бодринг	сабзавот	5,9
9	олма	мева	6,1
10	сабзи	сабзавот	6,8
11	Кўкат	сабзавот	7,0
12	Кўкат - салат	сабзавот	7,2
13	банан	мева	9,2

Маҳсулот учта синфни ташкил қилмоқда, энди янги берилган маҳсулот қайси синфга мансублигини аниқлаш масаласи турсин.



90-расм. Берилган сўзнинг гуруҳга мансублигини аниқлаш.

Помидор – бу сабзавот ёки мева?

k-NN усули бўйича уни k нчи қўшни объектлари синфига мансуб деб қабул қилинади. Объектлар ўртасидаги масофани *Евклид кўрсаткичи*, яъни объектлар ўртасидаги масофани (x_1, y_1) ва (x_2, y_2) координаталарни $\sqrt{((x_1 - y_1)^2 + (x_2 - y_2)^2)}$ орқали ифодалаймиз помидор ширинлик кўрсаткичи 3, карсиллаш кўрсаткичи 7 бўлса, олмага нисбатан яқинлик даражаси тахминан 6,1; 3,6, 9,2, га тенг бўлади. Қўйида помидорга нисбатан маҳсулотларнинг жойлашувини ошиш тартибида келтирамиз.

Сўнгра k қийматни танлаб, қўшни объектлар мансуб бўлган синфни аниқлаш лозим бўлади. Чунки k=1 бўлса, яқин қўшни объект – битта ва у апельсин– мева. k=2 бўлганда апельсин ва узум, ҳар иккиси мева. k=3 бўлганда 2 мевага (апельсин ва узум) ва қисқичбақа (протеин) га эга бўламиз. Демак, k-NN усули яна: «мева» деган жавобни беради. k=4 бўлганда жавоб «бир хил эҳтимоллик билан мева ёки протеин». k=5, k=6, k=7, k=8 бўлганда «ғолиб» протеин ҳисобланади. Ушбу жараёни k ошган сайин давом эттириш мумкин. Демак, k-NN усулида олинadиган натижа k параметр танланмасига жуда ҳам боғлиқ бўлиб қолади.

Шундай савол туғилади: k -NN усулида олинадиган хато натижаларни қисқаришига олиб келадиган k параметр қиймати қандай танланилиши лозим?

k қиймати жуда кичик деб танланилса, синфи нотўғри танланилган объект «воз кечиш» га олиб келади, яъни, нотўғри қарорга олиб келади. Агар k параметр қиймати оширилса, бундай объектлар йўқотилиши олдини олиш мумкин. Бироқ, бунда бошқача хатолик туғилади. Буни англаш учун k қийматини объектлар сонига N тенг деб оламиз. Бунда, табиийки оммабоп синф «ғолиб» чиқади, ва бунда объектгача масофа ҳеч қандай аҳамиятга эга бўлмайди. k параметрнинг оптимал қиймати танланганидан сўнг «*bias-variance tradeoff*» деб номланади, яъни «воз кечиш» ва «дисперсия» ўртасида келишувга олиб келади. Амалиётда кўпинча $k = \lfloor \sqrt{N} \rfloor$ қўлланилади, яъни бизни мисолда $k=3$ ва натижада тасниф натижасида, *помидор – мева* бўлиб қолади.

Танлаш «тўғри» лигига ишончингиз комил бўлса k ни кичик қийматда танлаш мумкин. Шу билан бирга «*weighted voting*» номли усул ҳам мавжуд (яъни «ўлчамга эга танланма»), ўрганилаётган объект кўшни объектлари узоқда жойлашган объектларга нисбатан катта оғирлик кўрсаткичига эга. k -NN усулни қўллашнинг яна бир омили–бу берилганларни бошланғич тайёрлаш.

2.9.k-NN усулни қўллашдан олдин маълумотларни бошланғич тайёрлаш

Эслатиб ўтамизки, кўриб ўтилаётган мисолда иккита хусусият (ширинлик даражаси ва “қарсиллаш” хусусияти) инобатга олинапти, бир хил шкалада ўлчаланилиб, қийматлар 0 дан 10 кўрсаткичда белгиланган. Амалда турли хусусиятлар турли ўлчов бирликка ва кўрсаткичда ўлчаланиши мумкин. Реал объектлар ўртасида фарқни аниқлашда хатоликларга ва муаммоларга олиб келиши мумкин, уларни олдини олиш учун k -NN усулни қўллашдан олдин *нормаллаштириш (масштаблаштириш)* олиб борилади (англ.: *scaling*).

Нормаллаштиришнинг турли хил усуллари мавжуд. Кўпинча қўлланиладиган усулларни кўриб ўтамиз:

$$x_i \equiv \frac{x_i - x_{min}}{x_{max} - x_{min}}. \quad (53)$$

(1) – формула хусусиятларнинг абсолют қийматидан нисбий қийматларга ўтишини кўрсатади. Янги ўзгарувчилар учун афзаллик томони бу уларнинг 0 дан 1 гача қийматларни (ёки фоиз олинса 0 дан 100) гача қабул қилади.

Масштабластиришнинг иккинчи усули:

$$x_i \equiv \frac{x_i - \bar{x}}{s}, \quad (54)$$

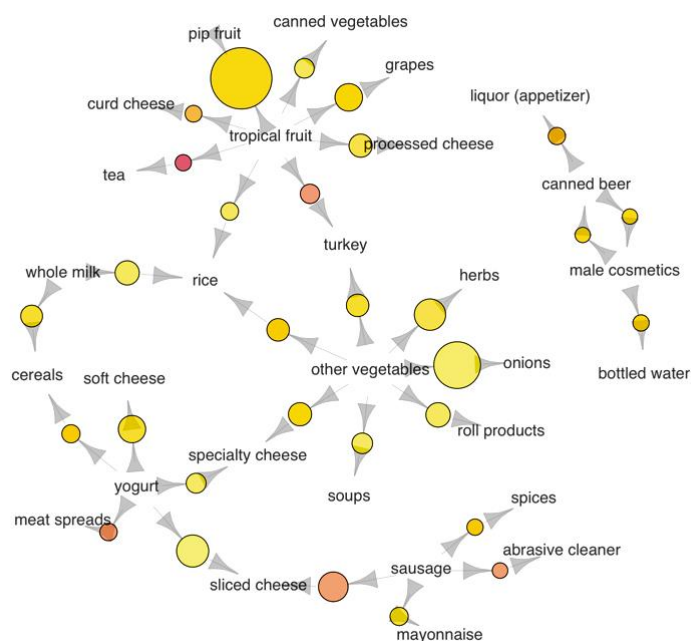
бунда \bar{x} – ўртача танланма (яъни $\bar{x} \equiv \frac{1}{n} \sum_{i=1}^n x_i$), s – танланган ўртаквадратик силжиш (яъни,

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}. \quad (55)$$

маълумки, агар ξ тўғри тақсимотга μ ва σ параметрга эга бўлса, у ҳолда $\eta \equiv \frac{\xi - \mu}{\sigma}$ тўғри тақсимотга эга, бироқ унинг параметрлари тақсимоти 0 ва 1 га мос равишда тенг ҳисобланади (бундай ифодалар *стандарт Гаусс* деб номланади).

Барча хусусиятларни ҳам унинг сони билан ифодалаб бўлмайди. Бунинг учун *dummy coding* га олиб келинади. Масалан, «жинси» номли хусусиятни эркаклар учун 1, аёллар учун 0 билан ифодалаш мумкин. [13]

2.10. Ассоциация қондаси бўйича ўрганиш алгоритмлари



91-расм. Ассоциация қондаси бўйича ўрганиш алгоритми.

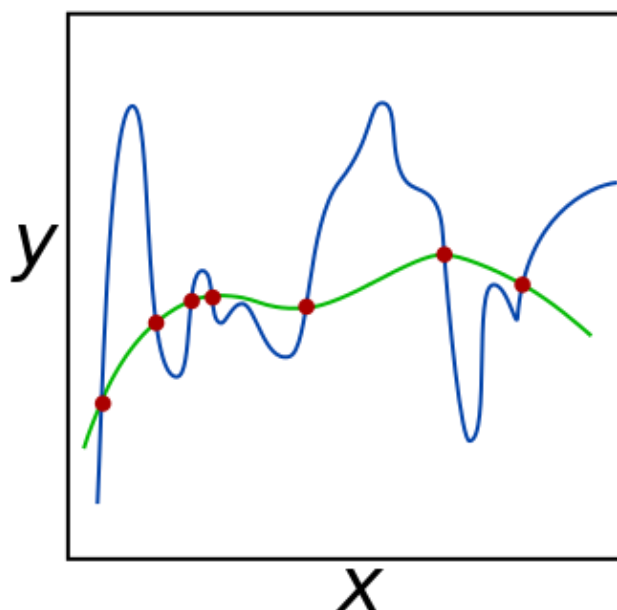
Ассоциация қоидаларини ўрганиш – бу транзакция тўпламини инобатга олиб, асосий мақсад транзакциядаги бошқа элементлар асосида кейинги кирувчи элементларни башорат қилиш қоидасини топиш

Apriori - берилганларни интеллектуал таҳлил этишда муҳим ўрин эгаллайди. Маълум бир жисмларни (бир ёки бир нечта жисмлар тўплами) ва улар хос ассоциация қоидаларини мэйнинг жараёнида қўллаш самарали ҳисобланади. Одатда бу алгоритм кўп сонли транзакцияга эга маълумотлар базасида қўлланилади. Масалан, маҳсулотлар мижозлар томонидан супермаркетларда харид қилинади. Apriori алгоритми тизим аъзолари сонини қуйидаги принципда қисқартиради: агар элементлар тўплами тез-тез учрашса, демак, унинг қуйи тўплами ҳам тез-тез учрайди.

Eclat (эквивалентлик синфини ўзгартириш) - Apriori алгоритмидан фарқи шундаки, излаш амали кенглик бўйича эмас, балки чуқурлик бўйича бажарилади. Apriori алгоритмида маҳсулотга асосланган элемент (1, 2, 3, 3 ва ҳ.к. позициялар) қўлланилади. Eclat алгоритмида эса транзакция элементлар асосида узатилади (сават 100, 200 ва ҳ.к.).

FP (тез-тез такрорланадиган ҳолат) – жорий транзакциялар бўйича бозордаги харидни таҳлил қилишга ёрдам беради. Маъно жиҳатдан биргаликда харид қилиниши мумкин бўлган маҳсулотлар тўпламини аниқлашга уринади FP-Growth Apriori га нисбатан афзалдир, чунки Apriori тез-тез қўлланиладиган элементларни аниқлаш учун транзакция маълумотларини қайта-қайта сканерлаш учун кўп вақт талаб этади.

2.11.Тартибга солиш алгоритмлари



92-расм. Тартибга солиш алгоритмлари:

Тизмалар регрессияси (L2 тартибга солиш) – асосий ғоя –бу берилганларнинг қайта ўрганиш муаммоларни ечиш. Стандарт чизиқли полиномиал регрессион модели айнан ўзгарувчилар хусусиятларида юқори даражадаги коллениарлик (озод ҳадлар ўртасида чизиқли боғланишнинг мавжудлиги) бўлган ҳолларда инкирозга учрайди. Тизмалар регрессияси ўзгарувчиларга силжиш квадрат коэффицентини кўшади. Бундай силжиш квадрат коэффиценти ўзгарувчилар ёйилишини таъминлаб модель дисперсиясини камайтиради. Тизмалар регрессиясида битта камчилик мавжуд, у якуний моделнинг барча n функцияларини ўз ичига олади.

Абсолют кичрайтириш ва танлаш оператори (LASSO, L1 Regularization) – тизмалар регрессиясидан фарқли равишда фақат катта коэффицентларни қисқартиради. Гиперпараметр θ жуда катта бўлган ҳоллар учун Лассо натижасида баъзи коэффицентлар кўрсаткичи нольга тенг бўлиб қолади. Демак, Лассо натижасида олинган моделлар танланган ўзгарувчилар асосида тузилиб, регрессион таҳлил ёрдамида олинган моделга нисбатан уларни англаб олиш жуда осон.

Эгилувчан тўр- лассо ва тизмалар регрессиясини баъзи хусусиятларини мужассамлаштирган. Лассо кўпгина функцияларни четлаштиради, тизмалар регрессияси функциялар таъсирини қисқартиради, бу одатда башорат қилишда аҳамиятга эга бўлмаган у қийматлар. Бу алгоритм эса бир неча функциялар таъсирини қисқартирсада, бироқ барча функцияларни четлаштирмайди.

Кичик бурчак остида регрессия(LARS) - тўғри қадамли регрессия билан бир хил бўлиб, ҳар бир босқичда олдинги натижа асосида предикторни аниқлайди. Бир хил корреляцияга эга бир неча предикторлар мавжуд бўлганда, битта предиктор бўйлаб ҳаракатланиш ўрнига предикторлар ўртасидаги бурчак бўйича ҳаракатланади.

Иккинчи боб бўйича хулоса.

Бобда ечимлар дарахти алгоритмлари, тасодифий ўрмон, г да тасодифий ўрмон, байес алгоритмининг асосий назария, йўқотиш функцияси моҳияти, регрессион ва ансамбль алгоритмлари, ўлчамни кичрайтириш алгоритмлари, кластерлаш алгоритмлари, k-оралиқда кўшни хусусиятларни қўллаш усулининг таснифлаш масалаларни ечишда қўлланилиши моҳияти ва мазмуни, knn усулни қўллашдан олдин маълумотларни бошланғич тайёрлаш, ассоциация қоидаси бўйича ўрганиш алгоритмлари ва тартибга солиш алгоритмларининг умумий таснифи ва амалиётда фойдаланиш бўйича тавсиялар ишлаб чиқилган.

Умумий хулоса

Бугунги кунда сунъий интеллект ҳаётимизга мустаҳкам кириб, кўплаб муаммоларни ҳал қилишда ёрдам беради. Илмий-фантастик киноларнинг келажагини яқинлаштирадиган сунъий интеллектнинг энг истиқболли йўналишларидан бири нейрон тармоқлардир. Зотан, улар тижоратда, айниқса, маркетинг ишларида фаол фойдаланилмоқда, хавфсизлик ва бошқа соҳаларда қўлланилади. Ушбу соҳада олиб борилган тадқиқотлар, масалан, Microsoft ва Google каби энг илғор компаниялар билан шуғулланади, бу деярли ҳар куни бу соҳада янги кашфиётлар пайдо бўлишига ёрдам беради.

Сунъий нейрон тармоқлари биологик принцип асосида қурилган бўлиб, бир қатор тахминлар билан ишлаб, кўплаб боғланишлар асосида амалларни бажаради. Инсон мияси сингари, бу тармоқлар ҳам ўрганиш қобилиятига эга. Сунъий нейрон тармоқлари учун таълим-бу вазифани самарали ҳал қилиш учун тармоқ архитектураси (нейронлар ўртасидаги алоқалар тузилиши)ни ва синаптик боғланишлар оғирлик кўрсаткичлари(сигналларга таъсир қилувчи коэффициентлари) ни созлаш жараёни тушунилади. Одатда нейрон тармоқни ўқитиш баъзи намуналар асосида амалга оширилади [1]. Тренинг давомида тармоқ белгиланган вазифаларни яхшироқ бажаришга, белгиланган буйруқларга жавоб беришга киришади.

Қуйида энг кенг тарқалган соҳаларни кўриб ўтамыз:

Маълумотни топиш, тасвирни аниқлаш. 2016 йилнинг кузида Яндекс нейрон тармоқларга асосланган янги Палех кидирув алгоритмини ишга туширди, Google аналоглари «Колибри» ва RankBrain. Ушбу алгоритмлар аниқроқ кидиришга ёрдам беради. Палех саҳифа сарлавҳаларини таҳлил қилади ва уларнинг маъносини англаб олади, тез орада барча матн шу жараённи ўтади [4].

Тасвирни аниқлаш - бу турли нейрон тармоқлари томонидан аниқланган тасвир асосида кидириш тизимида берилган сўров бўйича ўхшаш тасвирларни излаб топишни амалга оширади Яндекс ва Google, деб камида энг машҳур кидирув тизими деб олиш [5] мумкин. Юклаб олиш ёки шунга ўхшаш тасвирлар учун кидирув вазифасини танлаб,

расмда сичқонча тугмасини босиш, фойдаланувчи у муваффақиятли кўраётганлиги ва аналогларини аниқлашга нейрон тармоғи буйруқ беради, кейинги янги юкланган фото тасвирланган қайси кўриниш эканлигини таснифлайди ва тэглар яратади. Аммо технология яна бир кадам олға ташлади: миллионлаб одамларнинг фотосуратлари бирма-бир кўриб ўтилиб, аниқланган ҳолда қонуният ҳосил қилди ва шовшув бўлган FindFaceга ўхшаш инсонларни аниқлаб бериши мумкин. 2015-йилда юзни таниб олиш бўйича халқаро танловда ушбу лойиҳа энг яхши деб топилди, ҳатто Google-дан таниб олиш технологиясини ҳам четлаб ўтди. Ва 2016 йилда нейрон тармоқлар видеотасвирларда ўз юзини яширган ҳолда тасвирга олинган кўришларни ўргандилар ва уларни аниқлашга эришилди. Биргина YouTube-ёзувидан қайта ишланган тасвирлар 80-90 фоизни, фото муҳаррирлари ёрдамида яхшилаб ўзгартирилган тасвирларни танишиш аниқлиги 50-75 фоизгача бўлган. Энди одамни яширин ҳолатда қолиб кетганда юз кўринишни аниқлаш жараёнига муҳтожлик қолмади.

Нутқни аниқлаш, таржима қилиш, такрорлаш. Инсон овозли маълумотини Okey Google қабул қилиш ва англаш имкониятига эга, аммо Google томонидан сотиб олинган DeepMind нейрон тармоғи инсон нутқини янада аниқроқ имитация қилишни ўрганди [6]. Шунга айтиш керакки, ҳозирги вақтда хорижий сўзларни таржима қилиш технологияси нейрон тармоқлари туфайли доимий равишда такомиллаштирилмоқда. Яқинда сиз чет эллик киши билан гаплашадиган тилни билишингиз шарт эмас бўлиб қолади. Чунки, яқинда иккита технологияни бирлаштирган ҳолда, бемалол тўғридан тўғри таржима қилиб юкланадиган технологиядан фойланишингиз мумкин бўлади. Яқинда Google ўзининг СИ ҳар қандай профессионалдан кўра лаблар ҳаракати бўйича ўқишни ўрганганини эълон қилди. Нейрон тармоқ орқали олинган фотосуратларда бўлгани каби, 5 минг соатлик турли теледастурлар ёзувлари ҳам ўтказиб юборилган, натижада DeepMind сўзларнинг бир қисмини ютган ҳолда ҳам лаблар ҳаракати бўйича ўқишни ўрганди. Ҳозирги вақтда DeepMind 30% одамлар орасида профессионаллардан кўра лаблар

ҳаракати бўйича ўқиш билан яхши ишлайди [3]. Буларнинг барчаси субтитр яратиш ва callcentr ларда ёрдамчиларни ишлатиш учун катта салоҳиятга эга.

Санъат. Нейрон тармоқлари, масалан, белгиланган параметрларда бўлгани каби, фотосуратни қайта ишлаши мумкин. Масалан, муаллиф томонидан кўрсатилган репродукцияга ўхшаш услубда оддий расмга айлантириш ёки эскизни барча элементларини чизиш орқали санъат маҳсулига айлантириш имкони мавжуд. Шунингдек, нейрон тармоқ ўз хоҳишига кўра яқуний тасвир услубини мустақил равишда танлаган ҳолда яратиши мумкин. Нейрон тармоқлари мусиқа ёзиш, баъзи оддий куйлар яратиш, ёки улар мусиқа билан биргаликда унга мос тушувчи сўзлар қўйиб тўлиқ албом ёзиш каби имкониятлар мавжуд. Бироқ мусиқа ўзи одамлар томонидан яратилган, ва кейин қўшма натижада сўзлар қўйилиб асарлар яратилган, лекин инсон томонидан яратилган асардан бутунлай фарқ қилиниши аниқланган. Фильм учун биринчи трейлерни яратиш, бадиий филмни суратга олган сценарийни ёзиш - санъат соҳаси энди буларнинг барчасини инсон эмас, нейрон тармоқлар амалга ошириши мумкин.

Илмий. Нейрон тармоқлари веб-сайтлар учун ноёб матнларни ёзадилар, ҳали профессионал эмас, бироқ жуда маъноли, баъзи СИ ахборот агентликлари учун янгиликлар ёзадилар. Бундан ташқари, улар илмий мақолалар яратадилар. Тажриба шуни кўрсатдики, нейрон тармоқ базасига бир неча ўнлаб ёзилган мақолалар жойлаштирилганда, тармоқ уларни таҳлил қилиб, янги мақолалар ёзди ва уларни бир қатор илмий журналларга ҳам юборди, бу ерда баъзилари ҳатто омма эътиборига ҳавола этилди. Бу ҳақиқат ушбу журналларни таҳрирловчиларининг бепарволиги ва тармоқ томонидан ёзилган мақолаларнинг юқори сифати ҳақида гапириш мумкин. Агар илм-фан ҳақида гапирадиган бўлсак, унда сунъий интеллект жуда тез ривожланмоқда.

Тиббиётда деярли ҳар куни нейрон тармоқлардан фойдаланишнинг янги кашфиётлари пайдо бўлмоқда, бу касалликнинг кўз қобиғи кўриниши асосида касалликни аниқлаш кашфиётининг ўзи

бу бир янгилик. Акушер роботлар, беморларни парвариш қилишни ўрганиш учун ўзларига ўхшаш бўлган ички маълумот алмашиниш имконияти бўлган роботлар, инсон танасида яшовчи ва касалликнинг ҳар қандай бошланишини бартараф этадиган нанороботлар ҳақида деярли ҳақиқат деб ҳисобланган. Автомобилсозликда ҳайдовчининг функцияси кузатувчининг функциясига ўзгартирилган ҳолда ўз-ўзини ўргатадиган машиналар. Дронлар ва роботлар ер бўйича бошқаришни ўрганиш, минимал тўқнашувлар ва ҳар қандай сирт билан ҳаракат қилишни ўрганишади. Илм-фан тараққиёти юқори хавфли ҳудудларда инсонни даволаш ва инсон ўрнига ёрдам бериб, минглаб ҳаётни сақлаб қолишга ёрдам беради.

Хизматлар соҳаси. Юқорида айтилганларни таҳлил қилиб, инсон ўрнини босадиган соҳаларни тушуниш учун етарли бўлади. Зотан, хатларни ўқиш ва тўғри жавобни таклиф қилиш орқали электрон почталарга жавоб беришга ёрдам берадиган роботлар ҳам мавжуд. Мижозларнинг саволларига жавоб беришни ўрганадиган онлайн маслаҳатчилар, биринчи навбатда ҳақиқий мененжерларни кузатиб, жавоб беришга ҳаракат қиладилар. Агар жавоблари нотўғри бўлса, мененжерлар жавобларни тўғрилаб, уни ҳисобга олган ҳолда ўзгаришларни амалга оширадилар. Лука компанияси яна илғор натижаларга эришди, у одамнинг хатти-ҳаракатларини кузатиб борадиган ва унинг электрон нусхасини яратиш, унинг хабарларини ўрганиш, бошқа одамлар билан тўлиқ мулоқот қилиш, маълумот топиш, ўқитиш учун нейрон тармоқ яратди. Нейро тармоқлари ҳозирда фойдаланувчиларни ўрганмоқда ва маълум бир истеъмолчининг эҳтиёжига мувофиқ реклама таклиф қилмоқда. Келажакда биз мижозларни тўлиқ автоматлаштирилган кўллаб-қувватлаш тизимини йўлга қўйишни кутмоқдамиз, ижтимоий тармоқлар, гуруҳлар, онлайн-дўконларнинг барча мененжерлари сунъий интеллект билан бошқарилади, саволларга жавоб беради, муаммоларни почта, телефон орқали ҳал қилади, буларнинг барчаси тезкор ва сифатли амалга оширилади.

Юқорида айтиб ўтилганлар, соҳаларда нейрон тармоқларни қўлланилиши ёки аллақачон ишлатилишининг кичик бир қисми бўлиб, ишлаб чиқиладиган ёки режада бўлган нейрон тармоқлар анча кўп. 2011 йилдан бошлаб нейрон тармоқлар асосида СИ соҳасига йиллик инвестициялар ҳажми 15 баробар ошди, бироқ бу соҳада ривожланаётган Start-Урлар сонига назар ташласангиз, бу ўн минглаб ва таҳлилчиларнинг тадқиқотларига кўра, юзлаб одамлар бир неча йил ичида юз миллиардлаб доллар турадиган инвестицияни кўзламоқда. Бугунги кунда фақат битта юзни аниқлаш бозори 3 миллиард долларга тенг ва бу нейрон тармоқларнинг фақат битта йўналиши. Бундай тез ривожланиш инсон ҳаётининг кўплаб соҳаларида такомиллашиб, мунтазам ишларни осонлаштиради, бироқ айтилган пайтда кўплаб ишларни қисқартириш хавфи пайдо бўлади, баъзан эса бутун иш ўрнини тўлиқ йўқ қилишга олиб келади. Чунки нейрон тармоқ уни тезроқ, яхшироқ ва арзонроқ қилади. Замонавий ёндашувлар вазифаларни бажариш учун янги йўналишлар, янги иш воситаларига эга бўлади. Унда бутун дунё ва ҳаёт ўзгаради.

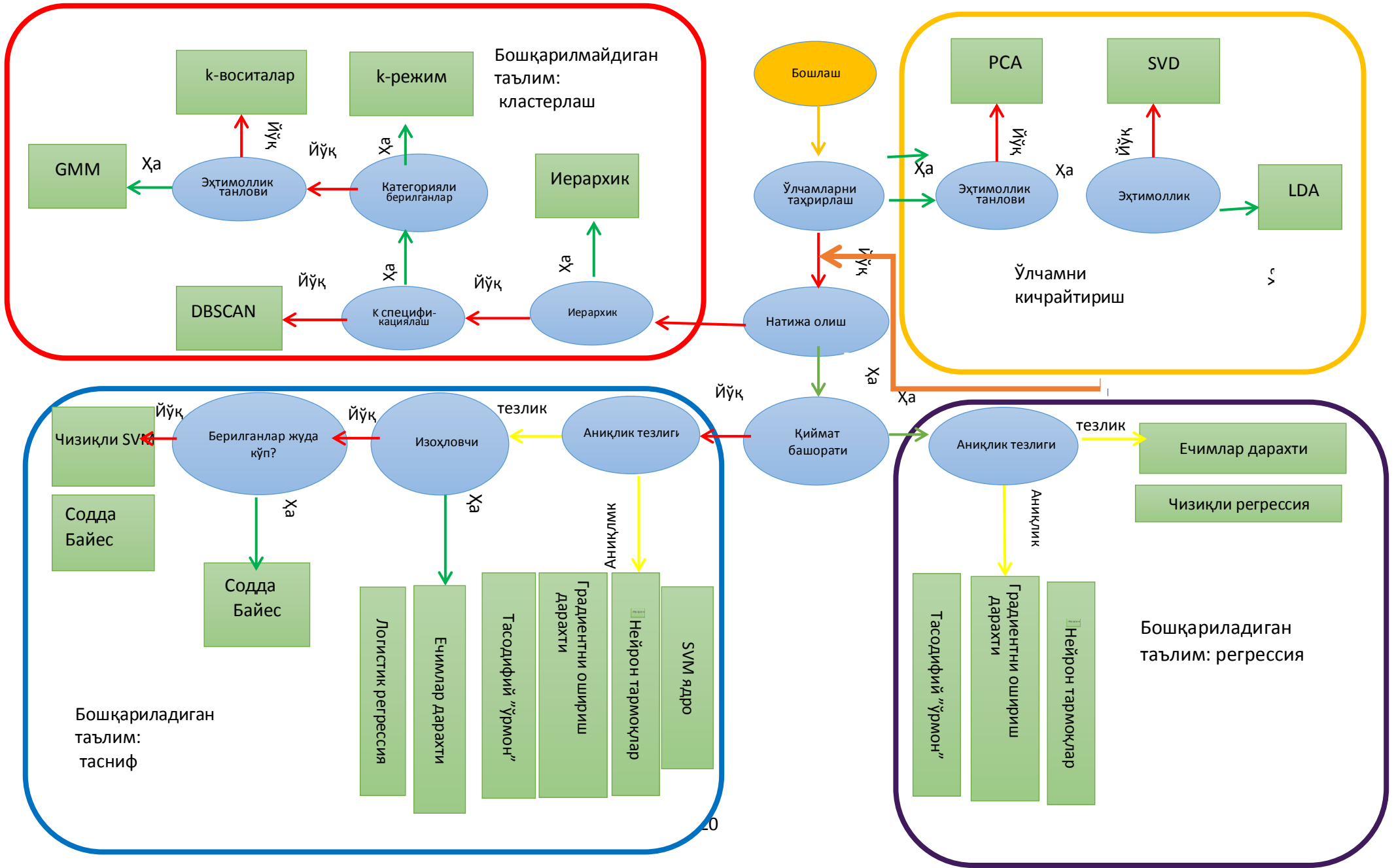
Энди, ҳар қандай машинали ўрганиш алгоритми билан ишлашни истовчи мутахассислар олдига “Қайси алгоритмни танлаш?” деган савол кўяди. Саволга жавоб бир неча омилларга асосланиб олинади.

Ҳаттоки, яхши мутахассис ҳам бир нечта алгоритмни қўллаб сўнгра улардан бирини танлайди. “Битта алгоритмни танлаш керак” деган ғояни олға сурмаймиз, балки энг аввал бир нечта омилга таянадиган алгоритмларни қўллаб кўришни тавсия берамиз.

Келтирилган вариантлар бошланғич тадқиқотчилар ва аналитиклар учун мўлжалланган соддалаштирилган тавсиялар келтирилди.

Келтирилган алгоритмлар, машинали ўрганиш бўйича тадқиқотчилар, экспертлар тавсиялари ва маслаҳатлари негизда тузилган. Турли фикрларни умумлаштириб, фарқларни ажратиб тузилди. Баъзан битта йўналиш алгоритм кўринишлари қўйилган масала ечимига жуда ҳам мос тушса, баъзан ҳеч бири ҳам тўғри келмаслиги мумкин. Энг тўғриси энг кенг тарқалган алгоритмларни

Аниқ масалаларни ечишда машинали ўрганиш алгоритмини танлаш вариантлари



Адабиётлар:

1. Bayes classifier [Электронный ресурс]. Режим доступа: <http://dataaspirant.com/naive-bayes-classifier-machine-learning/> - Дата доступа: 12.05.2017.
2. Resources for Text, Speech and Language Processing [Электронный ресурс]. Режим доступа: <http://www.cs.technion.ac.il/~gabr/resources/pointers.html> - Дата доступа: 20.01.2017.
3. Investigations on dynamic neural networks [Электронный ресурс]. Режим доступа: <http://people.idsia.ch/~juergen/SeppHochreiter1991Thesis>
4. AdvisorSchmidhuber. pdf - Дата доступа: 03.02.2017 Khachaturova K.R. Information technology as a means of development of creative abilities of primary school pupils in natural science lessons // Глобальный научный потенциал. 2015. № 9 (54). С. 111-113.
5. Brett Lantz. Machine Learning with R. Pack Publishing. Birmongham-Mumbai, 2013.
6. Brown E.W. Applying Neural Networks to Character Recognition. Northeastern University internal paper.
7. Bishop C. M. Neural Networks for Pattern Recognition. Oxford University Press Inc., 2003.
8. Pinkus A. Approximation theory of the MLP model in neural networks. Acta Numerica, 1999.
9. Haykin S. Neural Networks: A Comprehensive Foundation. NY: Macmillan, 1994.
10. Nigrin A. Neural Networks for Pattern Recognition. Cambridge, MA: The MIT Press, 1993.
11. Zurada J.M. Introduction To Artificial Neural Systems. Boston: PWS Publishing Company, 1992.
12. DARPA Neural Network Study, AFCEA International Press, 1988.
13. http://www.osp.ru/text/302/179978/_p1.html.
14. <https://bi.snu.ac.kr/Courses/g-ai09-2/hopfield82.pdf>

15. https://www.researchgate.net/profile/Terrence_Sejnowski/publication/242509302_Learning_and_relearning_in_Boltzmann_machines/links/54a4b00f0cf256bf8bb327cc.pdf
16. <https://apps.dtic.mil/dtic/tr/fulltext/u2/a620727.pdf>
17. <https://pdfs.semanticscholar.org/f582/1548720901c89b3b7481f7500d7cd64e99bd.pdf>
18. <https://papers.nips.cc/paper/3112-efficient-learning-of-sparse-representations-with-an-energy-based-model.pdf>
19. <https://arxiv.org/pdf/1312.6114v10.pdf>
20. <https://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks.pdf>
21. <http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf>
22. <https://arxiv.org/pdf/1503.03167v4.pdf>
23. <https://arxiv.org/pdf/1406.2661v1.pdf>
24. <https://arxiv.org/pdf/1512.03385v1.pdf>
25. <https://neuronus.com/theory/nn/955-nejronnye-seti-kokhonena.html>
26. <https://feisky.xyz/machine-learning/appendix/algorithms.html>
27. <https://towardsdatascience.com/ml-algorithms-one-sd-%CF%83-74bcb28fafb6>

МУНДАРИЖА

I БОБ. СУНЪИЙ НЕЙРОН ТАРМОҚЛАРИ НАЗАРИЙ АСОСЛАРИ ..3	
1.1.Сунъий интеллектнинг ривожланиш тарихи	3
1.2.Сунъий интеллект ривожланиш босқичлари	6
1.3.Сунъий интеллект ривожланиш йўналишлари	8
1.3.1.Билимларга асосланган тизимларни ишлаб чиқиш ва уларни ифодалаш	9
1.3.2.Ўйин ва ижод.....	9
1.3.3.Табиий тилда ва машинали таржима тизими ишлаб чиқиш ..	9
1.3.4.Тасвирларнинг аниқланиши	10
1.3.5.Компьютернинг янги архитектураси.....	10
1.3.6.Интеллектуал роботлар.....	10
1.3.7.Махсус дастурий таъминот.....	10
1.3.8.Ўрганувчи ва мустақил ўрганувчи	11
1.4.Сунъий нейрон тармоғи	11
1.4.1.Нейрон тармоқларни қўллаш соҳалари. Образларни аниқлаш. Тасниф масалалари	13
1.4.2.Нейрон тармоқ таснифи	14
1.5.Тўғри чизиqli нейрон тармоқлари.....	31
1.6.Радиал-асосли функцияга эга тармоқлар (RBF).....	36
1.7.Марков занжири	38
1.8.Хопфилд Нейрон тармоқ	40
1.9.Хэмминг Нейрон тармоқлари	42
1.10.Больцман (BM) машинаси	45
1.10.1Автоэнкодерлар	48
1.10.2.Сийрак автоэнкодер.....	49
1.10.3.Вариацион автоэнкодер архитектураси	50
1.11.Чуқур ишончли тармоқлар	52
1.11.1.Конволюцион нейрон тармоқлар	53
1.12.Деконволюцион нейрон тармоқлар	55
1.12.1.Чуқурлаштирилган тескари конволюцион график тармоғи .	56
1.12.2.Генератив-рақобатли тармоқ (generative adversarial network, GAN).....	57
1.13.Конволюцион нейрон тармоқлари (CNC, CNN)	58
1.13.1.CNN архитектураси тўлиқ изоҳи	58
1.13.2.CNN қўлланиладиган қатламлар.....	60
1.13.3.Конволюция қатлами.....	64
1.13.4.«Ақлий» ёндашув.....	65

1.13.5.Локал боғланиш	65
1.13.6.Фазовий жойлашув	67
1.13.7.Параметрларни ҳамкорликда қўллаш.....	70
1.13.7.NumPy мисоллар	72
1.13.8.Конволюция қатлами бўйича амаллар кетма-кетлиги.....	73
1.14.Демо-конволюция тармоғи.....	74
1.14.1.Матрицаларни кўпайтириш кўринишда қўллаш.	75
1.14.2.Тескари боғланиш орқали хатоликни текшириш усули.	76
1.14.3.Кенгайтирилган конволюция	76
1.15.Пулинг қатлам.....	77
1.15.1.Умумий пулинг	78
1.15.2.Пулингдан холис бўлиш	78
1.16.Тўлиқ боғланган қатлам.....	79
1.16.1.Тўлиқ боғланган қатламларни конволюцион қатламга ўзгартириш	79
1.16.2.Қайта ўзгартириш роли	80
1.17.CNN архитектураси.....	81
1.17.1.Қатламлар кўриниши	81
1.17.2.Аниқ мисоллар	83
1.17.3.VGGNet изоҳи	85
1.17.4.Ҳисоблашлар бўйича тавсиялар.....	87
1.18.Рекуррент нейрон тармоқлар синфи	88
1.19.Тьюринг нейрон машинаси	91
1.20.Икки йўналишли рекуррент нейротармоқ.....	92
1.21.Чуқур орттирмали тармоқ	93
1.22.Нейрон эхо-тармоқ	94
1.23.Экстремал ўрганиш машинаси.....	95
1.24.Турғун ҳолатлар машинаси	96
1.25.RNN LSTM ёрдамида таржима жараёни	97
1.26.Узоқ ва қисқа муддатли хотира (LSTM) модели.	99
1.27.Нейрон машинали таржима тизими солаш.....	100
1.28.Таянч векторлар машинаси	105
1.29.Таянч векторлар машинаси асосида маълумотларни таснифлаш	106
1.29.1.Таянч векторлар машинаси	107
1.30.Word2Vec.....	109
1.30.1.Word2vec ни қўллаш.....	110
1.31.Кохонен нейрон тармоғи	122
1.31.1.Кохонена тармоғда ўрганиш	134

1.31.2.Кохонен харитаси	138
1.31.3.Кохонен харитасида ўрганиш	140
1.31.4.Кохонен харитасининг график кўриниши	145
1.32.Ўқитувчи иштирокида ўрганувчи вектор квантлаш тармоқлари (LVQ-сети).....	149
1.33.Қарши тақсимланган тармоқ.....	152
Биринчи боб бўйича хулоса	155
II БОБ. СУНЪИЙ НЕЙРОН ТАРМОҚЛАРИ АЛГОРИТМЛАРИ ВА ФОЙДАЛАНИШ УСЛУБЛАРИ.....	156
2.1.Ечимлар дарахти алгоритмлари	156
2.2.Тасодифий ўрмон	160
2.2.1.R да тасодифий ўрмон.	164
2.3.Байес алгоритми	169
2.3.1.Асосий назария	172
2.3.2.Йўқотиш функцияси моҳияти	176
2.4.Регрессион алгоритмлар	191
2.5.Ансамбль алгоритмлар	193
2.6.Ўлчамни кичрайтириш алгоритмлари	201
2.7.Кластерлаш алгоритмлари.....	203
2.8.К-оралиқда қўшни хусусиятларни қўллаш усулининг таснифлаш масалаларни ечишда қўлланилиши	205
2.9.k-NN усулни қўллашдан олдин маълумотларни бошланғич тайёрлаш	210
2.10.Ассоциация қоидаси бўйича ўрганиш алгоритмлари	211
2.11.Тартибга солиш алгоритмлари.....	213
Иккинчи боб бўйича хулоса	214
Умумий хулоса	215
Аниқ масалаларни ечишда машинали ўрганиш алгоритмини танлаш вариантлари	220
Адабиётлар	221

Б.Б. МҰМИНОВ, К.Б.МУХАМАДИЕВА

**СУНЬИЙ НЕЙРОН
ТАРМОҚЛАРИ
ТАСНИФИ**

(Монография)

Тошкент – «Aloqachi» – 2020

Мухаррир: Қ. Матқурбонов
Тех. муҳаррир: А. Тоғаев
Мусаввир: Б. Эсанов
Мусахҳиҳа: Ф. Тагаева
Компьютерда
саҳифаловчи: Б. Бердимуродов

Нашр. лиц. ii № 176, 11.06. 2010.
Босишга рухсат этилди 14.11.2019.
Бичими 60x84 ¹/₁₆. «Times Uz» гарнитураси.
Шартли босма табағи 14,75. Нашр босма табағи 14,25.
Адади 60. Буюртма № 47

«Nihol print» ОК да чоп этилди.
Тошкент шаҳри, Мухтор Ашрафий кўчаси, 99./101.