9-дәріс. Нейрондық желілер және терең оқыту

Пәні: Интеллектуалды ақпараттық жүйелер

Дәрістің мақсаты

Студенттерге нейрондық желілердің құрылымын, оқыту алгоритмін, негізгі терең оқыту архитектураларын, бағалау әдістерін және жауапты қолдану қағидаларын түсіндіру; нақты міндет үшін модельді саналы таңдауға негіз қалыптастыру.

Күтілетін оқу нәтижелері

- жасанды нейрон, салмақ, ығысу (Смещение) және активация функциясын түсіндіреді;
- алға және кері тарату кезеңдерін байланыстырады;
- CNN, RNN және transformer архитектураларын дерек түрімен сәйкестендіреді;
- overfitting пен underfitting белгілерін анықтайды;
- классификация метрикаларын қате салдарына қарай таңдайды;
- әділдік, құпиялық және адам бақылауының маңызын негіздейді.

Нейрондық желілер және терең оқыту

Бүгін біз машиналық оқытудың ең қуатты бағыттарының бірі — нейрондық желілер мен терең оқытуды қарастырамыз. Өткен дәрісте модельдің деректерден заңдылық үйренетінін талқыладық. Енді сол идеяны жалғастырып, күрделі заңдылықтарды көпқабатты модель қалай үйренетінін түсінеміз. Біздің мақсатымыз тек атауларды жаттау емес. Біз нейронның қандай есеп орындайтынын, қате қалай өлшенетінін, салмақтар қалай жаңартылатынын және модельдің қай жағдайда сенімді болатынын біртұтас үдеріс ретінде көреміз.

Қарапайым мысал. Мысалы, электрондық пошта қызметі хатты «қалыпты» немесе «спам» деп бөледі. Қарапайым жағдайда бірнеше ереже жеткілікті болуы мүмкін. Бірақ хаттардың тілі мен құрылымы жиі өзгеретіндіктен, нейрондық желі көптеген мысалдан жасырын заңдылықтарды үйренеді.

Менің түсіндіруімдегі негізгі ой: Интеллектуалды ақпараттық жүйелер. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Дәрістің мақсаты және оқу нәтижелері

Енді мен алдыңғы ойды келесі ұғыммен байланыстырып түсіндіремін.

Бұл бөлімде мен «дәрістің мақсаты және оқу нәтижелері» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Дәріс соңында сіздер жасанды нейрон, салмақ, ығысу (Смещение), активация функциясы, қабат, жоғалту функциясы және градиент ұғымдарын түсіндіре аласыздар. Сонымен қатар толық байланысқан желі, convolutional neural network, recurrent neural network және transformer архитектураларының қандай деректерге сәйкес келетінін ажыратасыздар. Біз академиялық тәуекелді болжау мысалын қолданамыз. Бұл мысал арқылы кіріс белгілерден ықтималдыққа дейінгі жолды, оқыту және тексеру деректерінің қызметін, артық үйрену мәселесін және адам бақылауының қажеттілігін талдаймыз.

Қарапайым мысал. Мысалы, дәріс соңында мен сіздерге суретті тану есебін берсем, сіздер CNN архитектурасын ұсынып қана қоймай, неліктен оны таңдағандарыңызды, сапаны қандай метрикамен өлшейтіндеріңізді де түсіндіре алуларыңыз керек.

Менің түсіндіруімдегі негізгі ой: Нейрондық желінің негізгі ұғымдарын түсіндіру. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Машиналық оқытудан нейрондық желіге көшу

Осыны түсінгеннен кейін мен сіздерді келесі қадамға алып өтемін.

Бұл бөлімде мен «машиналық оқытудан нейрондық желіге көшу» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Өткен дәрісте біз классификация мен регрессияны қарастырдық. Көп жағдайда адам деректен маңызды белгілерді алдын ала таңдайды. Мысалы, студенттің GPA көрсеткіші,

сабаққа қатысуы және кеш тапсырған жұмыстары беріледі. Бірақ сурет, дыбыс немесе еркін мәтін сияқты деректерде пайдалы белгілерді қолмен анықтау өте қиын. Нейрондық желі бастапқы деректерден аралық белгілерді өзі құра алады. Бір қабат қарапайым өрнектерді, келесі қабат олардың комбинациясын, ал жоғарғы қабат шешімге тікелей қажет күрделі ұғымды үйренеді. Терең оқытудың негізгі артықшылығы осында.

Қарапайым мысал. Мысалы, кәдімгі кестелік модельге біз «қатысу пайызы» сияқты дайын белгіні береміз. Ал суретпен жұмыс істегенде нейрондық желі жиек, пішін және текстура сияқты аралық белгілерді деректің өзінен үйренеді.

Менің түсіндіруімдегі негізгі ой: Классикалық модель белгілерді дайын күйінде қабылдайды. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Биологиялық нейрон және жасанды модель

Енді сұрақты сәл тереңдетіп, оның жүйе ішіндегі қызметін қарастырайық.

Бұл бөлімде мен «биологиялық нейрон және жасанды модель» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Нейрондық желі атауы адамның жүйке жүйесінен алынғанымен, жасанды нейрон биологиялық нейронның өте қарапайым математикалық моделі екенін ұмытпау керек. Биологиялық нейрон көптеген сигналды қабылдап, оларды біріктіреді және белгілі жағдай орындалса, келесі жасушаларға импульс жібереді. Жасанды нейрон да бірнеше санды қабылдайды, әр санның маңызын салмақпен белгілейді, оларды қосады және активация функциясы арқылы нәтиже шығарады. Ұқсастық идея деңгейінде ғана. Жасанды желінің есептеу тәртібі мен биологиялық мидың жұмысы бірдей деп айту ғылыми тұрғыдан дұрыс емес.

Қарапайым мысал. Мысалы, адам ыстық затқа қолын тигізгенде жүйке жасушалары сигнал жібереді. Жасанды нейронда температура мәні сан ретінде беріледі, салмақпен көбейтіледі және функция арқылы шешімге айналады. Бұл тек ұқсастық, толық биологиялық көшірме емес.

Менің түсіндіруімдегі негізгі ой: Дендриттер сигнал қабылдайды. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Жасанды нейронның математикалық моделі

Бұл бөлім бізді келесі маңызды түсінікке табиғи түрде әкеледі.

Бұл бөлімде мен «жасанды нейронның математикалық моделі» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Жасанды нейронның жұмысын төрт қадаммен түсіндіруге болады. Алдымен кіріс мәндері беріледі. Әр кіріс өз салмағына көбейтіледі. Содан кейін көбейтінділер қосылып, оларға ығысу (Смещение) деп аталатын b мәні қосылады. Осыдан z саны шығады. Соңғы қадамда z активация функциясына беріледі. Мысалы, академиялық тәуекел моделінде сабаққа қатысуға теріс салмақ, тапсырмаларды уақытында орындауға оң салмақ берілуі мүмкін. Салмақтың таңбасы әсердің бағытын, ал абсолют шамасы әсердің күшін көрсетеді. Бұл салмақтарды адам қолмен бекітпейді, модель оларды деректерден үйренеді.

Қарапайым мысал. Мысалы, x_1 — қатысу, x_2 — тапсырмаларды уақытында тапсыру, x_3 — аралық бақылау нәтижесі болсын. Желі әр көрсеткішті өз салмағына көбейтіп, олардың жиынтығынан тәуекел бағасын шығарады.

Менің түсіндіруімдегі негізгі ой: Кірістер: x_1, x_2, …, x_n. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

$$\mathbf{z = w_1x_1 + w_2x_2 + ... + w_nx_n + b}$$

$$\mathbf{y = f(z)}$$

Белгі	Мағынасы	Қарапайым мысал
x_i	Кіріс белгісі	сабаққа қатысу пайызы

Белгі	Мағынасы	Қарапайым мысал
w_i	Белгінің салмағы	қатысудың шешімге ықпалы
b	Ығысу (Смещение)	шешім шегін жылжыту
f(z)	Активация функциясы	нәтижені ықтималдыққа айналдыру

Аудиторияға сұрақ: салмақ пен кіріс мәнінің айырмашылығы неде?

Активация функциясының қызметі

Енді мен алдыңғы ойды келесі ұғыммен байланыстырып түсіндіремін.

Бұл бөлімде мен «активация функциясының қызметі» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Егер барлық қабат тек сызықтық амал орындаса, қанша қабат қоссақ та, бүкіл желі бір ғана сызықтық түрлендіруге тең болар еді. Сондықтан активация функциясы қажет. Ол модельге сызықтық емес байланыстарды үйренуге мүмкіндік береді. Мысалы, студент тәуекелі тек GPA төмендеген сайын бірқалыпты өспеуі мүмкін. Қатысу, тапсырма мерзімі және пән күрделілігі бірге әсер етуі ықтимал. Активация функциясы осындай күрделі шекараларды құруға жағдай жасайды. Функцияны таңдау қабаттың міндетіне байланысты: жасырын қабаттарда бір түрі, шығыс қабатында басқа түрі қолданылады.

Қарапайым мысал. Мысалы, екі студенттің GPA көрсеткіші бірдей болуы мүмкін, бірақ біреуінің қатысуы тұрақты, екіншісінікі күрт төмендеген. Активация функциясы белгілердің осындай бірлескен, сызықтық емес әсерін модельдеуге көмектеседі.

негізгі ой: Сызықтық емес байланыс қосады. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Негізгі активация функциялары

Осыны түсінгеннен кейін мен сіздерді келесі қадамға алып өтемін.

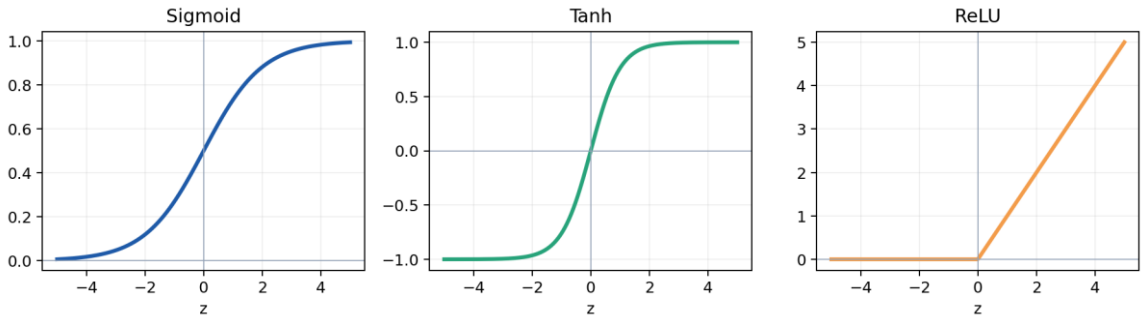
Бұл бөлімде мен «негізгі активация функциялары» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Sigmoid көбіне екі классты міндеттің шығыс қабатында қолданылады, себебі нәтижені ықтималдық ретінде түсіндіруге ыңғайлы. Tanh нөлдің айналасында симметриялы, бірақ терең желілерде градиенттің кішірею мәселесіне ұшырауы мүмкін. ReLU теріс мәндерді нөлге айналдырып, оң мәндерді сақтайды. Есептеуі жеңіл болғандықтан, ол жасырын қабаттарда кең таралған. Softmax бірнеше класстың бағаларын қосындысы бірге тең ықтималдықтарға айналдырады. Мысалы, өтінішті «төмен», «орта» және «жоғары тәуекел» кластарына бөлуге болады.

Қарапайым мысал. Мысалы, «тәуекел бар ма?» деген екі классты есептің соңында sigmoid қолданамын. Ал өтінішті бес бөлімнің біріне жіберсем, softmax арқылы бес ықтималдық аламын.

Менің түсіндіруімдегі негізгі ой: Sigmoid: 0 мен 1 аралығындағы мән. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

1-сурет. Негізгі активация функциялары



Функция	Мәндер аралығы	Қолданылатын орны	Ерекшелігі
Sigmoid	0 мен 1	Екі классты шығыс	Ықтималдықты түсіндіру ыңғайлы
Tanh	−1 мен 1	Кейбір жасырын қабаттар	Нөлдің айналасында симметриялы
ReLU	0 және оң мәндер	Жасырын қабаттар	Есептеуі жеңіл және жылдам
Softmax	Қосындысы 1	Көп классты шығыс	Кластар ықтималдығын береді

Перцептрон

Енді сұрақты сәл тереңдетіп, оның жүйе ішіндегі қызметін қарастырайық.

Бұл бөлімде мен «перцептрон» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын. Перцептрон — нейрондық желілер тарихындағы негізгі модельдердің бірі. Ол кірістерді салмақтайды және нәтиже белгілі шектен асса, бір классқа, аспаса, екінші классқа жатқызады. Геометриялық тұрғыдан перцептрон деректерді түзу сызықпен немесе жоғары өлшемде гипержазықтықпен бөледі. Бұл қарапайым міндеттерге жеткілікті. Бірақ XOR сияқты сызықтық бөлінбейтін есепті бір перцептрон шеше алмайды. Осы шектеу көпқабатты желілерге қажеттілікті көрсетті. Бірнеше нейрон мен жасырын қабаттар бірігіп, күрделі шешім шекарасын құра алады.

Қарапайым мысал. Мысалы, екі белгі бойынша студенттерді түзу сызықпен бөлуге болатын болса, перцептрон жеткілікті. Егер шешім «бір белгі жоғары, екіншісі төмен болғанда ғана оң» сияқты күрделі шартқа тәуелді болса, бір қабат жеткіліксіз болады.

Менің түсіндіруімдегі негізгі ой: Бір қабатты қарапайым классификатор. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Аудиторияға сұрақ: неге sigmoid функциясын барлық жасырын қабатта қолданбаймыз?

Көпқабатты перцептрон

Бұл бөлім бізді келесі маңызды түсінікке табиғи түрде әкеледі.

Бұл бөлімде мен «көпқабатты перцептрон» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Көпқабатты перцептронда нейрондар қабаттарға біріктіріледі. Кіріс қабаты есептеу жасамайды, ол белгілерді келесі қабатқа өткізеді. Жасырын қабаттар салмақталған қосынды мен активацияны қайталайды. Шығыс қабаты міндетке сәйкес нәтиже береді. Егер тәуекел бар немесе жоқ деп анықтасақ, бір sigmoid нейроны жеткілікті болуы мүмкін. Үш деңгейлі тәуекел үшін softmax қолданамыз. Әр қабат алдыңғы қабаттың нәтижесін қабылдайтындықтан, модель қарапайым белгілерден күрделі ішкі көрініс құрады. Осы ішкі көрініс representation деп аталады.

Қарапайым мысал. Мысалы, бірінші жасырын қабат қатысу мен тапсырма тәртібін бөлек өңдейді, екінші қабат оларды «оқу белсенділігі» деген аралық көрініске біріктіреді, ал шығыс қабаты тәуекел ықтималдығын береді.

Менің түсіндіруімдегі негізгі ой: Кіріс қабаты деректі қабылдайды. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Терең оқыту ұғымы

Енді мен алдыңғы ойды келесі ұғыммен байланыстырып түсіндіремін.

Бұл бөлімде мен «терең оқыту ұғымы» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Терең оқыту — көптеген өңдеу қабаттары бар нейрондық желілер арқылы деректердің иерархиялық көрінісін үйренетін машиналық оқыту бағыты. «Терең» сөзі қабат санына қатысты, бірақ нақты бір міндетті сан жоқ. Терең модель суретте алдымен жиектерді, кейін пішіндерді, одан соң нысандарды тануы мүмкін. Мәтінде әріптер мен сөздерден бастап контекст пен мағынаға дейінгі көрініс қалыптасады. Терең модель автоматты белгі үйренуде күшті, алайда ол сапалы дерек, жеткілікті есептеу қуаты және мұқият бағалау қажет етеді. Күрделілік әрқашан жақсы сапаға кепіл болмайды.

Қарапайым мысал. Мысалы, бет суретін танитын желінің алғашқы қабаты сызықтарды байқайды, келесі қабаты көз бен мұрынға ұқсас пішіндерді құрайды, ал жоғарғы қабаты тұтас бет бейнесін ажыратады.

Менің түсіндіруімдегі негізгі ой: Бірнеше жасырын қабат. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Алға тарату

Осыны түсінгеннен кейін мен сіздерді келесі қадамға алып өтемін.

Бұл бөлімде мен «алға тарату» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын. Алға тарату кезінде ақпарат кірістен шығысқа қарай қозғалады. Әр нейрон алдыңғы қабат мәндерін қабылдап, салмақталған қосындыны және активацияны есептейді. Мысалы, студент деректері қалыпқа келтірілген соң бірінші жасырын қабатқа беріледі. Ол бірнеше аралық белгі жасайды. Келесі қабат осы белгілерді біріктіреді. Соңғы sigmoid нейроны 0,78 мәнін шығарса, модель тәуекел ықтималдығын 78 пайыз деп бағалайды. Бұл әлі дұрыс немесе қате екенін айтпайды. Болжамды нақты белгімен салыстырып, жоғалту функциясын есептеу қажет.

Қарапайым мысал. Мысалы, бір студенттің қалыпқа келтірілген белгілерін желіге беремін. Қабаттар есептеуді ретімен орындап, соңында 0,78 мәнін шығарады. Бұл мән болжам, оны бірден ақиқат деп қабылдамаймын.

Менің түсіндіруімдегі негізгі ой: Кіріс бірінші қабатқа беріледі. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Жоғалту функциясы

Енді сұрақты сәл тереңдетіп, оның жүйе ішіндегі қызметін қарастырайық.

Бұл бөлімде мен «жоғалту функциясы» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Жоғалту функциясы модель қатесін бір санмен өлшейді. Регрессияда орташа квадраттық қате жиі қолданылады. Екі классты классификацияда binary cross entropy, бірнеше класста categorical cross entropy қолайлы. Егер модель нақты тәуекел бар студентке төмен ықтималдық берсе, шығын жоғары болады. Дұрыс әрі сенімді болжамда шығын төмендейді. Оқыту алгоритмі жоғалтуды азайтуға тырысады. Бірақ бір ғана оқыту жиынындағы төмен жоғалту жеткіліксіз. Модель бұрын көрмеген деректерде де дұрыс жұмыс істеуі керек. Сондықтан валидация және тест нәтижелерін бөлек бағалаймыз.

Қарапайым мысал. Мысалы, нақты белгі 1, ал модель 0,9 берсе, қате аз. Егер модель 0,1 берсе, қате әлдеқайда үлкен болады. Жоғалту функциясы осы айырманы оқыту алгоритмі түсінетін санға айналдырады.

Менің түсіндіруімдегі негізгі ой: Болжам мен нақты жауап айырмасын өлшейді. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Аудиторияға сұрақ: 0,78 ықтималдық міндетті түрде дұрыс шешім дегенді білдіре ме?

Градиент және градиенттік түсу

Бұл бөлім бізді келесі маңызды түсінікке табиғи түрде әкеледі.

Бұл бөлімде мен «градиент және градиенттік түсу» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Градиент әр параметр сәл өзгергенде жоғалту қалай өзгеретінін көрсетеді. Егер белгілі салмақты көбейту қатені өсірсе, оны азайту қажет. Градиенттік түсу параметрлерді градиентке қарсы бағытта жаңартады. Жаңарту формуласының қарапайым түрі: жаңа салмақ ескі салмақтан оқу жылдамдығы мен градиент көбейтіндісін азайту арқылы табылады. Оқу жылдамдығы тым үлкен болса, алгоритм ең жақсы нүктеден секіріп кетуі мүмкін. Тым кіші болса, оқу өте баяу жүреді. Практикада Adam сияқты оптимизаторлар әр параметр үшін қадамды икемдеп, оқытуды тұрақтандырады.

Қарапайым мысал. Таудың тұманды беткейінен төмен түсіп келе жатқанымызды елестетейік. Біз бүкіл жолды көрмейміз, тек тұрған жеріміздегі ең төмен бағытты бағалаймыз. Градиенттік түсу де параметрлерді осындай шағын қадамдармен өзгертеді.

Менің түсіндіруімдегі негізгі ой: Градиент қатенің өзгеру бағытын көрсетеді. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Кері тарату

Енді мен алдыңғы ойды келесі ұғыммен байланыстырып түсіндіремін.

Бұл бөлімде мен «кері тарату» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын. Кері тарату немесе backpropagation желінің барлық салмағына арналған градиентті тиімді есептейді. Алдымен алға тарату арқылы болжам мен жоғалту табылады. Кейін есептеу жолы кері бағытта өтіп, әр операцияның қатеге қосқан үлесі анықталады. Мұнда математикалық талдаудағы тізбекті туынды ережесі қолданылады. Соңында оптимизатор салмақтарды жаңартады. Кері тарату модельдің өзі жауапты кері айтып үйренуі емес. Ол қателіктің параметрлерге тәуелділігін есептеу әдісі. Бір алға және бір кері өту бір оқыту қадамының негізін құрайды.

Қарапайым мысал. Мысалы, соңғы болжам қате шықса, мен қатені тек соңғы қабатқа тіркемеймін. Кері тарату оның алдыңғы қабаттардағы әр салмаққа қаншалықты байланысты екенін есептеп береді.

Менің түсіндіруімдегі негізгі ой: Шығыс қатесі есептеледі. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Эпоха, batch және итерация

Осыны түсінгеннен кейін мен сіздерді келесі қадамға алып өтемін.

Бұл бөлімде мен «эпоха, batch және итерация» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Үлкен деректер жиынын бір қадамда өңдеу жадқа сыймауы мүмкін. Сондықтан деректер шағын batch топтарына бөлінеді. Әр batch үшін алға тарату, жоғалту, кері тарату және параметр жаңарту орындалады. Оқыту жиынының барлық batch топтары өңделгенде бір эпоха аяқталады. Мысалы, 3200 жазба және batch өлшемі 32 болса, бір эпохада 100 итерация болады. Эпоха саны аз болса, модель толық үйренбеуі мүмкін. Тым көп болса, оқыту дерегін жаттап алып, жаңа мысалдарға нашар бейімделуі ықтимал. Валидация графигі осы тепе-теңдікті бақылауға көмектеседі.

Қарапайым мысал. Мысалы, 3200 жазбаны 32 мысалдан бөлсем, 100 batch шығады. Әр batch бір итерация береді, ал барлық 100 итерация аяқталғанда бір эпоха өтеді.

Менің түсіндіруімдегі негізгі ой: Эпоха: бүкіл оқыту жиынын бір рет көру. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Аудиторияға сұрақ: кері тарату мен параметрді жаңарту бір ұғым ба?

Деректерді дайындау

Енді сұрақты сәл тереңдетіп, оның жүйе ішіндегі қызметін қарастырайық.

Бұл бөлімде мен «деректерді дайындау» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Нейрондық желінің сапасы деректер дайындығына тікелей байланысты. Әртүрлі масштабтағы белгілерді стандарттау немесе нормалау оқытуды жеңілдетеді.

Категориялық мәндер one hot encoding немесе embedding арқылы беріледі. Бос мәндерді жою әрқашан дұрыс емес, себебі дерек азаюы немесе жүйелі ығысу (Смещение) пайда болуы мүмкін. Ең маңызды қауіптердің бірі — data leakage. Мысалы, семестр соңындағы қорытынды бағаны семестр ортасындағы тәуекелді болжауға қолдансақ, модель болашақ ақпаратты алдын ала көріп қояды. Бұл тестте керемет нәтиже беріп, нақты қолдануда сәтсіздікке әкеледі.

Қарапайым мысал. Мысалы, GPA 0 мен 4 аралығында, ал қатысу 0 мен 100 аралығында берілсе, үлкен сан модельге шамадан тыс ықпал етуі мүмкін. Сондықтан мен белгілерді салыстырмалы масштабқа келтіремін.

Менің түсіндіруімдегі негізгі ой: Сандық белгілерді масштабтау. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Оқыту, валидация және тест

Бұл бөлім бізді келесі маңызды түсінікке табиғи түрде әкеледі.

Бұл бөлімде мен «оқыту, валидация және тест» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Деректерді үш бөлікке ажыратудың мақсаты әртүрлі. Оқыту жиыны желінің салмақтарын жаңарту үшін қолданылады. Валидация жиыны архитектура, оқу жылдамдығы, эпоха саны сияқты гиперпараметрлерді таңдауға көмектеседі. Тест жиыны барлық таңдау аяқталғаннан кейін бір рет соңғы бағалау үшін сақталады. Егер тест нәтижесіне қарап қайта-қайта модельді өзгертсек, тест те жанама түрде оқытуға қатысып кетеді. Уақыттық деректерде кездейсоқ бөлу дұрыс болмауы мүмкін. Онда бұрынғы кезеңді оқытуға, кейінгі кезеңді тексеруге бөлу нақты қолдану жағдайын жақсы көрсетеді.

Қарапайым мысал. Мысалы, кітаптағы тапсырмамен үйренуді оқыту жиынына, дайындық сынағын валидацияға, ал қорытынды емтиханды тестке ұқсатуға болады. Қорытынды емтихан сұрақтарын алдын ала көре берсек, баға әділ болмайды.

Менің түсіндіруімдегі негізгі ой: Оқыту жиыны параметрлерді үйретеді. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Гиперпараметрлер

Бұл бөлімде мен «гиперпараметрлер» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Параметрлерді модель деректерден үйренеді, ал гиперпараметрлерді әзірлеуші оқыту алдында таңдайды. Қабат саны, әр қабаттағы нейрон саны, оқу жылдамдығы, batch өлшемі, dropout үлесі және оптимизатор түрі осы топқа жатады. Үлкен желі күрделі заңдылықты үйрене алады, бірақ есептеу шығыны және артық үйрену қаупі өседі. Сондықтан модельді міндетке сай қарапайым нұсқадан бастаған дұрыс. Гиперпараметрлерді валидация нәтижесі бойынша салыстырамыз. Бір ғана метрикаға қарап таңдау орнына, тұрақтылықты, есептеу уақытын және түсіндірмелілікті де ескереміз.

Қарапайым мысал. Мысалы, мен бір модельде екі жасырын қабат, екіншісінде төрт қабат қолданамын. Екеуін валидация дерегінде салыстырып, тек дәлдігіне емес, тұрақтылығы мен есептеу уақытына да қараймын.

Менің түсіндіруімдегі негізгі ой: Қабат және нейрон саны. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Аудиторияға сұрақ: тест жиынын неге модель баптауға қолдануға болмайды?

Артық үйрену және жеткіліксіз үйрену

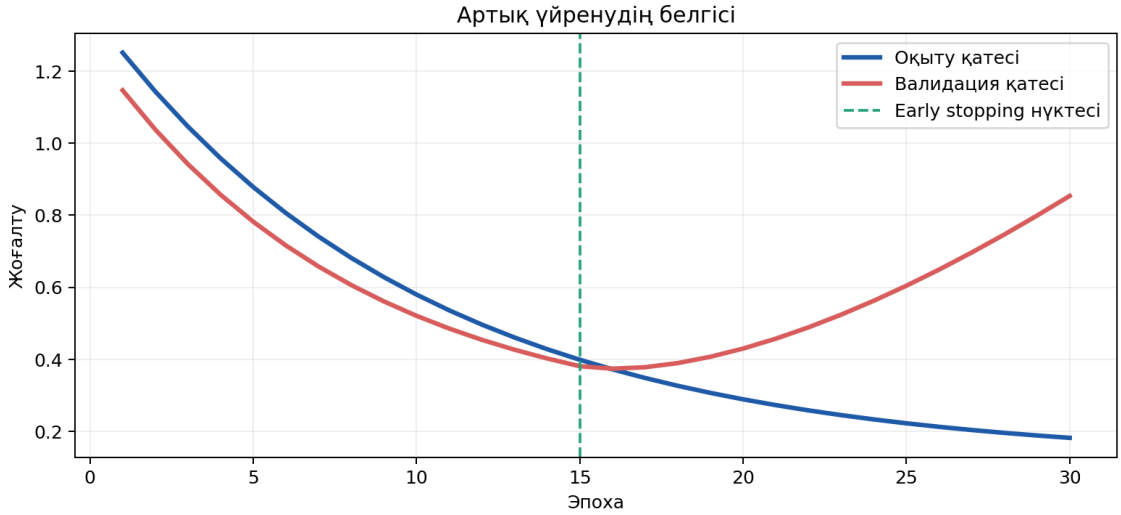
Бұл бөлімде мен «артық үйрену және жеткіліксіз үйрену» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Жеткіліксіз үйренуде оқыту мен валидация нәтижелері екеуі де нашар болады. Бұған модельдің тым қарапайым болуы, белгілердің әлсіздігі немесе оқытудың жеткіліксіздігі себеп болуы мүмкін. Артық үйренуде оқыту нәтижесі өте жақсы, ал валидация сапасы төмендейді. Бұл модельдің жалпы заңдылықтың орнына кездейсоқ шу мен жеке мысалдарды жаттағанын көрсетеді. Екі жағдайды loss және metric графиктері арқылы байқауға болады. Негізгі мақсат оқыту дерегін мінсіз қайталау емес, бұрын көрмеген студенттер үшін сенімді болжам жасау екенін үнемі есте ұстаймыз.

Қарапайым мысал. Мысалы, студент есептің шешу әдісін түсінбей, дайын жауаптарды жаттап алса, таныс есепте жақсы нәтиже көрсетеді, ал жаңа есепте қиналады. Overfitting дәл осы жағдайға ұқсайды.

негізгі ой: Underfitting: модель заңдылықты толық игермейді. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

2-сурет. Оқыту және валидация қатесінің өзгеруі



Графикте оқыту қатесі төмендей береді, ал валидация қатесі белгілі бір сәттен кейін өсе бастайды. Мен осы айырмашылықты артық үйренудің негізгі белгісі ретінде түсіндіремін. Жасыл үзік сызық early stopping үшін орынды нүктені көрсетеді.

Регуляризация тәсілдері

Енді сұрақты сәл тереңдетіп, оның жүйе ішіндегі қызметін қарастырайық.

Бұл бөлімде мен «регуляризация тәсілдері» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Регуляризация модельдің күрделілігін бақылап, жалпылауды жақсартады. L2 үлкен салмақтарға айып қосып, шешімнің тұрақты болуына ықпал етеді. L1 кейбір салмақтарды нөлге жақындатып, сирек модель қалыптастыра алады. Dropout оқыту кезінде нейрондардың бір бөлігін кездейсоқ өшіреді. Осылайша желі бір ғана байланысқа шамадан тыс тәуелді болмайды. Early stopping валидация қатесі нашарлай бастағанда оқытуды тоқтатады. Суреттерде бұру, кесу және жарықты өзгерту арқылы data augmentation қолдануға болады. Әдісті дерек мағынасын бұзбайтындай таңдау қажет.

Қарапайым мысал. Мысалы, dropout кезінде оқыту барысында кейбір нейрондарды уақытша өшіремін. Бұл топтық жұмыста әр қатысушыға кезекпен демалыс беріп, қалғандарын да мәселені өздері шешуге үйретуге ұқсайды.

негізгі ой: L1 және L2 салмақтарды шектейді. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Аудиторияға сұрақ: оқыту қатесі азайып, валидация қатесі өссе, қандай мәселе бар?

Слайд 21. Convolutional Neural Network

Бұл бөлім бізді келесі маңызды түсінікке табиғи түрде әкеледі.

Бұл бөлімде мен «convolutional neural network» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Convolutional neural network немесе CNN көбіне сурет пен кеңістіктік құрылымы бар деректерге қолданылады. Кішкентай сүзгі сурет бойымен қозғалып, жиек, бұрыш немесе текстура сияқты жергілікті белгілерді анықтайды. Бір сүзгінің салмақтары барлық аймақта ортақ қолданылатындықтан, толық байланысқан желіге қарағанда параметр саны азаяды. Терең қабаттар қарапайым белгілерді біріктіріп, күрделі нысандарды таниды. CNN медициналық бейнелерді талдау, өндірістегі ақауды анықтау және құжат суреттерін тану міндеттерінде пайдалы. Бірақ оның нәтижесі оқыту деректерінің сапасы мен әртүрлілігіне тәуелді.

Қарапайым мысал. Мысалы, құжат сканеріндегі мөрді іздегенде бір сүзгі суреттің әр бөлігін қарап шығады. Мөр жоғарыда немесе төменде орналасса да, бірдей белгі анықталуы мүмкін.

негізгі ой: Кеңістіктік жергілікті белгілерді үйренеді. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Recurrent Neural Network

Енді мен алдыңғы ойды келесі ұғыммен байланыстырып түсіндіремін.

Бұл бөлімде мен «recurrent neural network» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Recurrent neural network немесе RNN тізбектегі элементтерді ретімен өңдеп, алдыңғы жасырын күйді келесі қадамға береді. Осы арқылы сөздердің ретін немесе уақыттық сигналдың тарихын ескереді. Қарапайым RNN ұзақ тізбекте маңызды ақпаратты жоғалтуы мүмкін. Бұған градиенттің жоғалуы немесе жарылуы әсер етеді. LSTM және GRU қақпа механизмдері арқылы ақпаратты сақтау мен ұмытуды басқарады. Олар мәтін, сөйлеу және уақыт қатарларында кең қолданылды. Қазіргі көптеген тілдік міндеттерде трансформерлер басым болғанымен, RNN идеясы тізбектік модельдеудің дамуын түсіну үшін маңызды.

Қарапайым мысал. Мысалы, «ол банкке барды» және «ол өзеннің банкіне барды» деген сөйлемдерде сөз мағынасы алдыңғы контекстке тәуелді. RNN тізбектегі бұрынғы ақпаратты жасырын күйде сақтауға тырысады.

негізгі ой: Тізбектегі алдыңғы күйді ескереді. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Transformer және attention

Бұл бөлімде мен «transformer және attention» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Transformer архитектурасының негізгі идеясы — attention механизмі. Ол әр элементтің басқа элементтермен байланысын есептеп, ағымдағы міндет үшін маңызды ақпаратқа көбірек салмақ береді. Мысалы, сөйлемдегі есімдік қай сөзге қатысты екенін анықтауда алыста тұрған сөз маңызды болуы мүмкін. RNN тізбекті кезекпен өңдесе, transformer элементтерді параллель өңдей алады. Бұл үлкен деректерде оқытуды жылдамдатады. Transformer модельдері мәтін генерациясы, аударма, сұраққа жауап беру, сурет пен дыбыс өңдеу салаларында қолданылады. Алайда олар көп есептеу ресурсымен бірге дерек сапасын, қауіпсіздікті және жауапты пайдалануды талап етеді.

Қарапайым мысал. Мысалы, ұзақ сөйлемдегі «ол» есімдігінің кімге қатысты екенін анықтағанда attention басқа сөздердің бәрімен байланысын салыстырып, маңызды сөзге көбірек назар аударады.

негізгі ой: Attention маңызды элементтер байланысын өлшейді. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Архитектура	Негізгі дерек түрі	Басты идея	Қарапайым қолдану
MLP	Кестелік дерек	Толық байланысқан қабаттар	Академиялық тәуекелді болжау
CNN	Сурет, кеңістіктік дерек	Жергілікті сүзгілер	Құжаттағы мөрді тану
RNN немесе LSTM	Тізбек, уақыт қатары	Алдыңғы күйді сақтау	Уақыт бойынша сұраныста болжау
Transformer	Мәтін және ұзақ контекст	Attention байланыстары	Өтініш мәтінін түсіну

Аудиторияға сұрақ: RNN ұзақ мәтінде неге маңызды ақпаратты жоғалтуы мүмкін?

Transfer learning

Енді сұрақты сәл тереңдетіп, оның жүйе ішіндегі қызметін қарастырайық.

Бұл бөлімде мен «transfer learning» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Transfer learning бір міндетте үйренген көріністерді басқа міндетке пайдалануға мүмкіндік береді. Үлкен сурет немесе тіл моделін нөлден оқыту қымбат. Сондықтан жалпы деректе алдын ала оқытылған модель алынып, жаңа шағын дерекке бейімделеді. Кейде бастапқы қабаттар қатырылады, тек соңғы қабаттар оқытылады. Кейде барлық параметр аз оқу жылдамдығымен fine tuning арқылы жаңартылады. Бұл тәсіл дерек аз болғанда пайдалы, бірақ бастапқы модель мен жаңа міндеттің сәйкестігін тексеру керек. Алдын ала модельдегі ығысу (Смещение)лар мен лицензиялық шектеулер де жаңа жүйеге өтуі мүмкін.

Қарапайым мысал. Мысалы, миллиондаған жалпы суретте үйретілген модельді университет зертханасындағы құрал ақауларын тануға бейімдей аламын. Маған нөлден оқытуға қарағанда аз дерек пен уақыт қажет болады.

негізгі ой: Алдын ала үйретілген модель пайдаланылады. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Модель сапасының метрикалары

Бұл бөлім бізді келесі маңызды түсінікке табиғи түрде әкеледі.

Бұл бөлімде мен «модель сапасының метрикалары» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Класстар тең болмағанда accuracy жаңылыстыруы мүмкін. Егер тәуекелді студенттер тек бес пайыз болса, барлығын «тәуекел жоқ» деп айтқан модель 95 пайыз accuracy алады, бірақ пайдалы емес. Recall нақты тәуекел жағдайларының қаншасын тапқанымызды көрсетеді. Precision модель тәуекел деп белгілегендердің қаншасы шынымен тәуекел екенін өлшейді. F1 осы екі метриканың гармониялық ортасы. Метрика таңдау қате түрінің салдарына байланысты. Академиялық қолдауда қауіпті студентті өткізіп алу маңызды болса, recall жоғары болуы керек. Бірақ жалған дабыл көбейсе, кеңесші ресурсы тиімсіз жұмсалады.

Қарапайым мысал. Мысалы, 100 студенттің бесеуі ғана тәуекел тобында болсын. Барлығына «тәуекел жоқ» деген модель 95 пайыз accuracy көрсетеді, бірақ бес студенттің ешқайсысын таппайды. Сондықтан recall мәнін міндетті түрде қараймын.

Менің түсіндіруімдегі негізгі ой: Accuracy жалпы дұрыс жауап үлесін береді. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Метрика	Қарапайым сұрақ	Қай кезде маңызды
Accuracy	Барлық жауаптың қаншасы дұрыс?	Кластар шамамен тең болғанда

Метрика	Қарапайым сұрақ	Қай кезде маңызды
Precision	Тәуекел дегендердің қаншасы шынымен тәуекел?	Жалған дабыл қымбат болғанда
Recall	Нақты тәуекелдің қаншасын таптық?	Қауіпті жағдайды өткізіп алмау керек болғанда
F1	Precision мен recall теңгерімі қандай?	Екі қате түрі де маңызды болғанда

Түсіндірмелілік

Енді мен алдыңғы ойды келесі ұғыммен байланыстырып түсіндіремін.

Бұл бөлімде мен «түсіндірмелілік» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Терең модельдің ішкі есебі күрделі болғандықтан, пайдаланушы шешімнің негізін түсінгісі келеді. Global explanation модельдің жалпы мінезін сипаттайды. Local explanation бір студент бойынша қандай белгілер ықпал еткенін көрсетеді. Feature importance, SHAP және LIME сияқты әдістер осы мақсатта қолданылады. Бірақ түсіндірме модельдің міндетті түрде әділ немесе дұрыс екенін дәлелдемейді. Ол тек шешімге әсер еткен үлгіні көрсетеді. Егер дерек дұрыс емес болса, түсіндірме де сол мәселені бейнелейді. Сондықтан түсіндірмелілікті дерек сапасы, метрика, әділдік және адам тексеруімен бірге қолданамыз.

Қарапайым мысал. Мысалы, модель студентті жоғары тәуекелге жатқызса, эдвайзер «қатысу төмендеді және үш тапсырма кеш тапсырылды» деген түсіндірмені көре алады. Бірақ ол түсіндірме болжамның міндетті түрде дұрыс екенін дәлелдемейді.

негізгі ой: Қай белгілер шешімге әсер етті. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Аудиторияға сұрақ: accuracy 95 пайыз болған модель неге пайдасыз болуы мүмкін?

Слайд 27. Әділдік, құпиялық және қауіпсіздік

Бұл бөлімде мен «әділдік, құпиялық және қауіпсіздік» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Нейрондық желі өткен деректерден үйренеді. Егер деректе белгілі топқа қатысты теңсіздік болса, модель оны қайталауы мүмкін. Сондықтан нәтижені жалпы метрикамен ғана емес, тиісті топтар бойынша да тексереміз. Жеке деректі жинау мақсаты нақты және көлемі қажет шамамен шектелуі тиіс. Қолжетімділік пен сақтау мерзімі басқарылуы керек. Сонымен қатар adversarial мысалдар, деректі улау және модельді рұқсатсыз көшіру сияқты қауіптер бар. Білім беру саласында модель қолдау ұсынуға көмектесуі мүмкін, бірақ студент тағдырын автоматты түрде шешетін жалғыз құралға айналмауы тиіс.

Қарапайым мысал. Мысалы, тарихи деректе белгілі бір топқа қолдау аз көрсетілсе, модель осы теңсіздікті қалыпты заңдылық ретінде үйренуі мүмкін. Сондықтан мен нәтижені тек жалпы емес, маңызды топтар бойынша да тексеремін.

негізгі ой: Деректегі ығысу (Смещение) шешімге өтеді. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Сквозной мысал: академиялық тәуекел

Енді сұрақты сәл тереңдетіп, оның жүйе ішіндегі қызметін қарастырайық.

Бұл бөлімде мен «сквозной мысал: академиялық тәуекел» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Енді барлық ұғымды бір мысалға біріктірейік. Университет студенттің келесі төрт аптада оқу қиындықтарына тап болу ықтималдығын бағалағысы келеді. Кіріске бағалау сәтіне дейінгі қатысу, тапсырма мерзімі және аралық бақылау нәтижесі беріледі. Деректер

тазаланып, масштабталады. Көпқабатты желі sigmoid шығысы арқылы ықтималдық береді. Модель тарихи деректе оқытылады, кейінгі кезеңде тексеріледі. Шешім шегі кеңесші ресурсы мен қате салдарын ескеріп таңдалады. Жоғары тәуекел автоматты жаза емес. Ол эдвайзерге студентпен сөйлесіп, қолдау қажеттігін анықтауға арналған белгі болуы керек.

Қарапайым мысал. Мысалы, 0,82 тәуекел бағасын алған студентке жүйе автоматты түрде жаза қолданбайды. Эдвайзер деректерді тексеріп, студентпен сөйлеседі және техникалық мәселе, денсаулық немесе оқу жүктемесі сияқты нақты себепті анықтайды.

негізгі ой: Мақсат: келесі төрт аптадағы тәуекелді бағалау. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Нейрондық желіні жобалау кезеңдері

Бұл бөлім бізді келесі маңызды түсінікке табиғи түрде әкеледі.

Бұл бөлімде мен «нейрондық желіні жобалау кезеңдері» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Жобаны архитектура таңдаудан емес, мәселені анықтаудан бастаймыз. Қандай шешімді қолдаймыз, қате кімге қалай әсер етеді және табысты немен өлшейміз деген сұрақтарға жауап керек. Кейін деректің заңдылығы, сапасы, өкілдігі және құпиялық талаптары тексеріледі. Нейрондық желімен қатар қарапайым baseline модель құрылады. Егер қарапайым модель жеткілікті болса, күрделі жүйенің шығыны ақталмауы мүмкін. Оқыту мен тәуелсіз бағалаудан кейін ғана жүйе шектеулі ортада енгізіледі. Нақты қолдануда дерек өзгеретіндіктен, сапа, ығысу (Смещение) және істен шығу белгілері тұрақты бақыланады.

Қарапайым мысал. Мысалы, мәтіндік өтініштерді бөлімдерге бөлу жобасында алдымен бөлімдер тізімін және қате бағыттаудың салдарын анықтаймын. Содан кейін ғана дерек жинап, қарапайым baseline мен күрделі модельді салыстырамын.

негізгі ой: Мәселе мен табыс өлшемін анықтау. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Аудиторияға сұрақ: жоғары тәуекел болжамы қандай әрекетке әкелуі тиіс?

Практикалық мини тапсырма

Бұл бөлімде мен «практикалық мини тапсырма» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Шағын топтарда мына жағдайды талдаңыздар: университет студенттің сұрағын тиісті қызмет бөліміне автоматты бағыттағысы келеді. Алдымен бұл классификация ма, әлде регрессия ма екенін анықтаңыз. Кіріс ретінде қандай дерек беріледі, шығыс не болады? Қысқа мәтінді өңдеу үшін қандай архитектура орынды және неге? Accuracy жеткілікті ме, әлде әр бөлім бойынша precision мен recall қарау керек пе? Құпиялық және қате бағыттау қандай салдарға әкеледі? Топ өз шешімін екі минутта түсіндірсін. Мұнда жалғыз дұрыс архитектурадан гөрі негізделген таңдау маңызды.

Қарапайым мысал. Мысалы, «жатақханадағы интернет жұмыс істемейді» деген өтініш ақпараттық технологиялар бөліміне түсуі керек. Ал «жатақханаға орналасу тәртібі қандай?» деген сұрақ басқа бөлімге жіберіледі. Бір сөздің өзі жеткіліксіз, сөйлем мағынасын ескеру қажет.

негізгі ой: Міндет түрін анықтаңыз. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Қорытынды

Бұл бөлімде мен «қорытынды» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Бүгін біз нейрондық желіні жұмбақ технология емес, нақты математикалық және инженерлік жүйе ретінде қарастырдық. Жасанды нейрон кірістерді салмақтап, активация арқылы шығыс береді. Қабаттар осы қарапайым операцияларды біріктіріп, күрделі

көріністерді үйренеді. Алға тарату болжам жасайды, жоғалту функциясы қатені өлшейді, кері тарату градиентті есептейді, ал оптимизатор параметрлерді жаңартады. CNN, RNN және transformer әртүрлі дерек құрылымына бейімделген. Соңғы нәтиженің құндылығы тек жоғары метрикамен өлшенбейді. Дерек сапасы, әділдік, құпиялық, түсіндірмелілік және адам бақылауы да жүйе сапасының бөлігі.

Қарапайым мысал. Мысалы, студенттік тәуекелді болжау жүйесінде математикалық модель, сапалы дерек, дұрыс метрика және эдвайзердің кәсіби шешімі бірге жұмыс істегенде ғана нәтиже пайдалы болады.

негізгі ой: Нейрон салмақталған қосынды мен активацияны есептейді. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Бақылау сұрақтары және әдебиеттер

Енді сұрақты сәл тереңдетіп, оның жүйе ішіндегі қызметін қарастырайық.

Бұл бөлімде мен «бақылау сұрақтары және әдебиеттер» тақырыбын қарапайымнан күрделіге қарай түсіндіремін. Алдымен негізгі мағынасын ашып, содан кейін оны нақты жағдаймен байланыстырамын.

Бақылау сұрақтарына жауап бергенде терминді ғана емес, оның қызметін түсіндіріңіздер. Негізгі әдебиеттер: Ian Goodfellow, Yoshua Bengio және Aaron Courville, Deep Learning; Christopher Bishop және Hugh Bishop, Deep Learning: Foundations and Concepts; Aston Zhang және әріптестері, Dive into Deep Learning; Simon Haykin, Neural Networks and Learning Machines. Келесі сабаққа дайындық ретінде бір қолдану саласын таңдаңыздар. Сол салада нейрондық желінің кірісі, шығысы, архитектурасы, бағалау метрикасы және негізгі этикалық тәуекелі қандай болатынын қысқаша жазыңыздар.

Қарапайым мысал. Мысалы, бақылау сұрағына жауап бергенде «CNN суретке арналған» деп қана айтпай, жергілікті кеңістіктік белгілер мен ортақ сүзгілердің неліктен пайдалы екенін түсіндіріңіздер.

негізгі ой: Неліктен активация функциясы қажет?. Осы ойды есте сақтасақ, келесі бөлімдегі жаңа ұғымды түсіну жеңіл болады.

Негізгі терминдер

Термин	Қысқаша анықтама
Жасанды нейрон	Кірістерді салмақтап, ығысу (Смещение) қосып, активация арқылы шығыс есептейтін элемент.
Салмақ	Кіріс белгінің модель шешіміне сандық ықпалын реттейтін параметр.
Активация функциясы	Желіге сызықтық емес байланыстарды үйренуге мүмкіндік беретін функция.
Жоғалту функциясы	Болжам мен нақты жауап арасындағы қатені өлшейтін функция.
Градиент	Параметр өзгергенде жоғалтудың қалай өзгеретінін көрсететін туындылар жиыны.
Кері тарату	Тізбекті туынды арқылы желі параметрлерінің градиентін есептеу алгоритмі.
Эпоха	Оқыту жиынын толық бір рет өңдеу.
Regularization	Артық үйренуді азайтатын әдістер жиыны.
Embedding	Дискретті нысанды үйренілетін тығыз вектор ретінде көрсету.
Transfer learning	Алдын ала үйретілген модель білімін жаңа міндетке бейімдеу.

Пайдаланылған және ұсынылатын әдебиеттер

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016.

2. Bishop C. M., Bishop H. Deep Learning: Foundations and Concepts. Springer, 2024.
3. Zhang A., Lipton Z. C., Li M., Smola A. J. Dive into Deep Learning. Cambridge University Press, 2023.
4. Haykin S. Neural Networks and Learning Machines. Pearson, 2009.
5. Vaswani A. et al. Attention Is All You Need. Advances in Neural Information Processing Systems, 2017.