

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

Сламбек Б.А.

Нейрондық желілер негізінде адамның бет-әлпет эмоциясын тану

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»
мамандығы

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ
МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

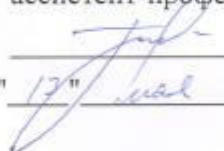

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

ҚОРҒАУҒА ЖІБЕРІЛДІ

ПИ кафедра меңгерушісі

техника ғылымдарының
кандидаты, доцент,
ассистент-профессор

 Р. Юнусов
" 17 "  2019ж.

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

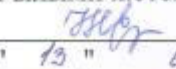

Тақырыбы: “Нейрондық желілер негізінде адамның бет-әлпет эмоциясын тану”

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»
мамандығы

Орындаған

Б.А. Сламбек

Ғылыми жетекші, лектор

 І.Н. Ерсари
" 13 "  05 2019 ж.

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӨТБАЕВ УНИВЕРСИТЕТІ

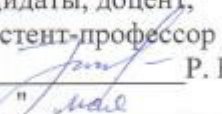

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

БЕКІТЕМІН

ПИ кафедра меңгерушісі,
техника ғылымдарының
кандидаты, доцент,
ассистент-профессор

 Р. Юнусов
"13"  2019ж.

**Дипломдық жобаны орындауға
ТАПСЫРМА**

Білім алушыға Сламбек Бахытжан Арапбекұлы
Тақырыбы: Нейрондық желілер негізінде адамның бет-әлпет эмоциясын тану
Университет ректоры бұйрығының № 1828-6 "13" наурыз 2019 ж. шешімімен
бекітілген.

Орындалған жобаның өткізу мерзімі "14" мамыр 2019 ж.

Дипломдық жобаның бастапқы мәліметтері: Жобаның төлқұжаты, технология бойынша техникалық құжаттама, техникалық тапсырма.

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

a) Tensorflow технологиясын қолдана отырып қимылдарды анықтау;
b) тұжырымдамаға сәйкес толыққанды оқытуға арналған ақпараттар қорын жинау;

в) бағдарлама анықтылығын дамыту, жақсарту;

г) бағдарламаны құру, қателіктерді анықтау және түзету, тестілеу;

Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен):

14 графикалық материалдар бар.

Ұсынылған негізгі әдебиеттер: 7 әдебиеттер тізімінен

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Кіріспе	12.12.2018	ескерту жасау
2. Тапсырма қойылымы және бағдарламалау ортасын таңдау	15.01.2019	ескерту жасау
3. Бағдарламалық камтаманы жобалау.	08.02.2019	ескерту жасау
4. Бағдарламаны әзірлеу	14.02.2019	ескерту жасау
5. Бағдарламаның мүмкіндіктерін дамыту	12.03.2019	ескерту жасау
6. Дипломдық жобаға түсіндірме жазба жазу	10.04.2019	ескерту жасау

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған
қолтаңбалары

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	Рамазан А.Б Тьютор	13.05.19	
Бағдарламалық бөлім	Байгаринов А.М. Техн.ғыл.магистр, лектор	13.05.19	

Ғылыми жетекші _____ I.N. Ерсари

Тапсырманы орындауға қабылдап алған студент _____ Б.А. Сламбек

Күні _____ «1» маусым 2018 ж.

МАЗМҰНЫ

	Кіріспе	8
1	Нейрондық желілердің негізгі түсінігі	9
1.1	Нейрондық желілердің тарихы	9
1.2	Жасанды нейрондық желі түсінігі	10
1.3	Биологиялық нейрон және жасанды нейрон	12
1.4	Жасанды зияткерлік жүйелердің жіктелуі	14
1.5	Нейрондық желілер теориясының элементтері	17
1.6	Машиналық оқытуды қолдану ауқымы	20
2	Нейрондық желілердің ақпараттық моделі және эмоцияны тану	23
2.1	Нейрондық желілердің жұмыс жасау принципі	23
2.2	Бір бағытты көп қабатты желілер	24
2.3	Көп қабатты персептронды оқыту	25
2.4	Кохонен желілері	26
2.5	Радиалды нейрондық желілер	27
2.6	Функционалды диаграмма	29
2.6.1	USE CASE диаграммасы бойынша	29
2.6.2	Класс(Сынып) диаграммасы бойынша	30
3	Нейрондық желіні қолданып бет - әлпеттегі эмоцияны тану	31
3.1	Әдістерді пайдаланып эмоцияны тануды белсендіру	31
3.2	Бағдарламаны жасау барысында пайдаланылған технологиялар	31
3.3	Жасанды интеллектуалдық эмоцияны тану жүйесі	32
	Қорытынды	37
	Пайдаланылған әдебиеттер тізімі	38
	А Қосымшасы	39
	Б Қосымшасы	41
	Спецификация	46

АҢДАТПА

Дипломдық жұмыстың негізгі өзектілігі, адамның бет-әлпет эмоциясын шынайы уақытта жылдам әрі нақты тану. Ақпараттық жүйе толықтай Python, OpenCV, Tensorflow, Jupyter, Keras технологияларын қолдану арқылы жасалды.

Дипломдық жоба 14-сурет, 3-кесте және 4-формуладан, сондай-ақ кіріспеден, үш бөлімнен, қорытындыдан, пайдаланылған әдебиеттер тізімінен және қосымшадан тұрады. Кіріспеде жобаның мақсаты, өзектілігі мен міндеттері көрсетіледі.

Бірінші бөлімде нейрондық желілер тарихы және түсінігі туралы сипаттама берілген.

Екінші бөлімде адам эмоциясын тануға арналған әдістер, нейрондық желілерді оқыту жайында, жобаның логикалық және бағдарламалық құрылымы туралы сипаттамалар берілген.

Үшінші бөлімде адам эмоциясын тану үшін белсендірілген және оңтайлы әдістер туралы сипаттама берілген.

Құрастырылған ақпараттық жүйені жасау мен ендіру жолдары сәйкестендіріліп, жобада анық айтылған. Жобаны іске асыру барысында қойылған мақсатқа қол жеткізіліп, сәйкес мәселелері шешілді – «Нейрондық желілер негізінде адамның бет-әлпет эмоциясын тану» жүйесі құрастырылды.

Бұл мақалада Tensorflow технологиясы арқылы модель негізінде болжау жасап эмоцияны тану мүмкіндіктері жасалған. Jupyter notebook бағдарламалау ортасында Python бағдарламалау тілінде және көптеген қосымшалардың көмегімен әзірленген бағдарлама.

АННОТАЦИЯ

Основная актуальность дипломной работы заключается в том, чтобы быстро и четко распознать эмоции лица в реальном времени. Информационная система полностью разработана с использованием технологий Python, OpenCV, Tensorflow, Jupyter, Keras. Дипломный проект состоит из рис.14, таблицы 3 и формулы 4, а также введения, трех разделов, заключения, списка использованной литературы и приложения. Во введении указывается цель, актуальность и задачи проекта.

В первой части дается описание об истории и понятии нейронных сетей.

Во второй части представлены характеристики о методах распознавания человеческих эмоций, об обучении нейронных сетей, логической и программной структуре проекта.

В третьей части дается описание активизированных и оптимальных методов для распознавания человеческих эмоций.

В проекте четко изложены пути разработки и внедрения разработанной информационной системы. В ходе реализации проекта была достигнута поставленная цель, решены соответствующие проблемы – создана система "распознавание выражение лица на основе нейронных сетей".

В данной статье представлены возможности распознавания эмоций, которые делают прогнозирование на основе модели с помощью технологии Tensorflow. Jupyter notebook программа, разработанная в среде программирования на языке программирования Python и с помощью многих приложений.

ANNOTATION

The main relevance of the thesis is to quickly and clearly recognize the emotions of the face in real time. The information system is fully developed using Python, Opencv, Tensorflow, Jupyter, Keras Technologies.

The diploma project consists of pictures 14, table 3 and formula 4, as well as an introduction, three sections, conclusion, list of references and Appendix. The preamble indicates the purpose, relevance and objectives of the project.

The first part describes the history and concept of neural networks.

The second part presents the characteristics of the methods of recognition of human emotions, learning neural networks, logical and software structure of the project.

The third part describes the activated and optimal methods for recognizing human emotions.

The draft clearly spells out the way in the development and implementation of the developed information system. During the implementation of the project, the goal was achieved, the relevant problems were solved – the system "facial expression recognition based on neural networks" was created.

This article presents the possibilities of recognition of emotions that make predictions based on models using technology Tensorflow. Jupyter notebook is a program developed in a programming environment in the Python programming language and using many applications.

КІРІСПЕ

Біздің өміріміздің әр түрлі салаларын автоматтандыру ескеріле бермейтін процесс. Көптеген процестер компьютерлік жүйелерде сипатталады және сақталады. Бұл деректерді тиімді және табысты өңдеуге, оларды сақтауға және тез іздеп табуға мүмкіндік береді. Менің дипломдық жұмысымда 11 892 сурет, 48x48 пикселде, *.jpg, *.png пішінінде эмоцияның 7 сыныбы бар, 1 – ашулану, 2 – жиіркену, 3 – қорқу, 4 – қуану, 5 – мұңаю, 6 – таңырқау, 7 – бейтарап.

Қазіргі уақытта жасанды нейрондық желілер жасанды зияткерлік жүйелерінде белсенді қолданылады. “Эмоцияларды оқу алгоритмі оларды 100%-ға сәйкестендіреді деп айтуға болмайды”, – деп есептейді Кембридж университетінің технологиялық ғылымдар кандидаты Тадас Балтрушайтис (Tadas Baltrusaitis), ол жаңа бағдарламаның рецензенті болып табылады.

Жасанды нейрондық желілер дамудың түрлі жолдарын береді, ал бағдарламалық қамтамасыз ету оларды пайдалануда үлкен көмегі бар. Таңдалған ақпараттық модель практикалық міндеттерді шешу үшін пайдаланылатын болады. Дипломдық жұмыстың негізгі міндеттері:

- Жасанды интеллектуалды жүйелердің қолданыстағы түрлерін, сондай-ақ олардың жұмыс істеу әдістерін зерттеу.
- Адамның бет-әлпет эмоциясын тану үшін оңтайлы нейрондық желілердің ақпараттық модельін таңдау.
- Бет-әлпетті танудың қазіргі әдістерін зерттеу және олардың арасында әмбебап әдістерді таңдау.
- Зияткерлік жүйе үшін тұлғаның қозғалыс белсенділігін тану алгоритмін жүзеге асыру және сипаттау және практикалық есепті шешу.

Осы зерттеудің нысаны мимикалық өрнектерді тану әдістері мен тәсілдері болып табылады. Зерттеу пәні жасанды нейрондық желілердің ақпараттық модельдері, сондай-ақ адамның қозғалыс белсенділігін тану алгоритмін іске асыру және сипаттау болып табылады.

Компьютерлік оқыту қысқа уақыттың ішінде көптеген салаларда қолданыста, атап айтар болсақ:

1. IT-сала: қосымшаларды әзірлеу (сөйлеуді тану, спамның анықталуы және т.б.)
2. Жарнамалық компаниялар (клиенттерге қызмет көрсетуді болжау)
3. Маркетингтік зерттеулер.
4. Медициналық диагностика (пациенттің медициналық тарихын талдай отырып, айқын емес адамға қауіпті аурулардың бұрын белгісіз белгілерін қосу және белгілеу үшін).
5. Техникалық диагностика.
6. Биоинформатика және басқа да көптеген салаларда.

1 Нейрондық желінің негізгі түсінігі

Нейрондық желілердің тарихы

«Нейрондық желі» ұғымы пайда болғаннан бері 150 жыл өткен екен. Бұл ұғым алғаш пайда болған уақытта, ми және жұлындағы нейрондарды зерттеуге қатысты қолданылған. Жасанды нейронды желілер аумағындағы зерттеулер үшін жандандыру түрлі кезеңнен өтті.

Негізі жасанды нейронды желілер туралы 1940 жылдардан бастап сөздер айтыла бастаған-ды.

Алғашқы зерттеулер 1940 жылдардағы Мак Каллок пен Питстің жұмыстарынан көріне бастаған, олар кез келген логикалық және арифметикалық алгоритмдерді нейрондық желілер көмегімен жүзеге асыруға болатынын көрсетті. 1949 жылы нейрондық желілерді үйренудің бастапқы нүкесі болатын заңды қалаған Дональд Олдинг Хэбб модельін атап өту керек. Дональд Олдинг Хэбб теориясы өзін-өзі үйренудің өзіндік үлгісі болып табылады, онда тестілеу жүйесі экспериментатордың араласуынсыз қажетті тапсырманы орындау үшін үйретіледі. Екінші кезең Ф.Розенблактың 1958 жылы объектілерді жіктеуге арналған персепрон деп аталатын нейрондық желіні ұсынды, персептронның үйлесу теоремасына және қарапайым персептронның мүмкіндіктерінің шектеулілігін көрсеткен. Сонымен қатар, дайындалған персепрон бұрын қолданылмаған нысандарды өз бетінше жіктеуге қабілетті болды, алайда қателіктерде жоқ емес еді. Мински мен Пеперттің жұмыстарының нәтижелері көпшілік зерттеушілердің, әсіресе есептеуіш техника ғылымдары аясында жұмыс жасаған зерттеушілердің қызығушылығын арттырды. Осыдан кейін нейронды желілерге деген зерттеулер шамамен 20 жылға дейін дамусыз тыныштықта болды. 1980 жылдардың басынан жасанды нейрондық желілерге деген қызығушылық қайта жандандырыла түсті. 1982 жылы Американдық биофизик Дж.Хопфилд нейрондық желінің дербес түрін ұсынды, ол өз атымен аталды, және осы Дж.Хопфилдтің туындысы пайда болғаннан соң нейрондық желілерге деген қызығушылық күрт өсті.

Жоғары өнімді дербес компьютерлер пайда болған кезде нейрондық желілерді модельдеу мүмкіндігі пайда болды. Келесі жылдары көптеген тиімді алгоритмдер табылды: қарама-қарсы ағын желісі, екі жақты ассоциативті жады, т.б. Дж.Хопфилдтің энергиялық тәсілі мен ең алғаш Вербастың ұсынған көп қабатты персептрон үшін кері таралу алгоритмінің дүниеге келуімен байланысты болды. Бұл алгоритм Румельхарттың өз жұмысында жария ету негізінде баршаға белгілі болды. Содан кейін 1986 жылы Уильямс, Румельхарт және Хинтон бірлескен жұмысы жасалды, ол қабатты нейрондық желілердің қуы мінез-құлықтарды анықтау, тану туралы маңызды сұраққа жауап берді. Осыдан кейін Хинтон ұсынған алгоритм көптеген өзгерістерге ұшырады.

1980 жылдарда біртұтас теориялық негіз бірте-бірте қалыптасты, оның негізінде көптеген желілер бүгінгі күні жасалынады. Пайда болған

проблемаларды шешу үшін соңғы екі онжылдықта дамыған теория кеңінен қолданылды.

90-шы жылдары нейрондық желілер бизнесте қолданыла бастады, онда олар үлкен тиімділік көрсетті, көптеген проблемаларды шешуде өнімге деген сұранысты банктің клиенттерінің төлем қабілеттілігін талдау үшін болжау.

2007 жылы Торонто университетінде Джеффри Хинтон нейрондық желілерді терең зерттеуге арналған алгоритмдер құрды. Желінің төменгі қабаттарын оқытуда Хинтон стохастикалық қайталама нейрондық желіні ұсынатын шектеулі Больцман машинасын қолданды. Желіні оқып болғаннан кейін алынған бағдарлама мәселені жылдам шеше алады (мысалы, фотосуреттегі тұлғаларды іздеу, тұлғаның эмоциясын тану). Бұл мүмкіндік қазір барлық сандық камераларға орнатылған.

Мамандардың бағалауы бойынша, нейрондық желілер мен нейрокомпьютерлерді жобалау саласында технологиялық даму күтілуде. Соңғы жылдары көптеген жаңа мүмкіндіктер ашылып, осы салада жұмыс ғылымға, технологияға және экономикаға маңызды үлес қосып келеді. Нейрондық модельдеуді зерттеу алпыс жылдан астам жүргізілгеніне қарамастан, ақпаратты өңдеу процесі соңына дейін анық болатын мидың бірде-бір аймағы жоқ. Сондай-ақ, импульстер тізбегі түрінде ақпаратты беру кодын анықтауға болатын бірде-бір нейрон жоқ. Қазіргі уақытта жұмыс принципі бойынша ерекшеленетін нейрондық желілердің көптеген конфигурациясы бар, олар әртүрлі тапсырмаларға бағытталған.

1.2 Жасанды нейрондық желі түсінігі

Адам миы шамамен 86 миллиард нейроннан тұрады. Олардың жиынтығы нейрондық желіні құрайды. Әр нейрон көптеген дендрит пен бір аксоннан тұратын жеке жасуша. Ондағы аксон тармақталып, ақпаратты бірден бірнеше жасушаға жеткізе алады. Нейрондар электр импульстерін жіберіп қана қоймай, химиялық сигналдар жіберетін де қабілетке ие.

Нейрондық желілер – бұл, мәліметтер қорынан көзге көрінбейтін заңдылықтарды ала отырып, мысалдар негізінде оқып үйрене алатын алгоритмдер тобының жалпыланған атауы.

Нейрожелілік деген атауға ие болған компьютерлік технологиялар адамның бас миындағы нейрондардың құрылым принципі мен қызмет етуіне ұқсас жұмыс жасап өте кең шеңбердегі мәселелерді шешуге мүмкіндік береді. Нейрожелілік технологиялар маманға анықталмағандық уақыттың жетіспеушілігі және ақпараттық қорлардың шектеулілігі жағдайында маңызды және шынайы емес шешімдерді жеңіл қабылдауды қамтамасыз етеді.

Нейрондық желілер – бұл, белгілі бір жолмен бір-бірімен байланысқан нейрондар жиынтығы. Биологиялық нейрондардың негізгі құрылымы мен жұмыс жасау принципіне мыналар жатады: Жүйке жүйесінің қызметінде

бастапқы рөлді өзара жүйке талшықтары арқылы байланысқан арнайы (торшалар) клеткалар нейрондар атқарады.

Нейрондар бір-біріне әр түрлі қуатты және жиіліктік электр импульстарын жібере алады.

Нейрон дейндригтерден олар арқылы басқа нейрондардан сигналдар қабылданады. Нейрондар және аксондар барлық клетканың өмір сүруін қамтамасыз етеді, Аксон – бұл ұзын талшық, аксон арқылы нейрон басқа нейрондарға сигналдарды жібере алады. Аксон басқа нейрондардың дейндригтерімен сигналдың қуатына әсер ететін арнайы түзілістер синапстар арқылы байланысады. Қандай да бір нейронға басқа бірнеше нейрондардан бір уақытта алынған сигналдар қосылады. Егер қосынды сигналдың қуаты қайсы бір шекаралық мәннен артатын болса онда сигналдың ұзақтығы да маңызды болады. Онда нейрон қозып өзінің меншікті импульсін генерациялап аксон арқылы береді. Осы үрдістің математикалық үлгісін құру үшін мынандай болжамдарды ескеру қажет:

Әрбір нейрон алынған сигналдың қуатына тәуелді оның қозу шарттарын анықтайтын берілу функциясына ие болады. Берілу функциялары уақытқа тәуелсіз синапс арқылы өткенде сигнал сызықты өзгереді, яғни сигналдың қуаты синапастың салмағы немесе нейронның сәйкестік кіріс салмағы деп аталатын қандай да бір санға көбейтіледі. Нейронның жұмысы синхрондалған, яғни сигналдың нейроннан нейронға өту уақыты тұрақталған және барлық байланыстар үшін бірдей болып табылады.

Нейрондық желілер – қарапайым процессорларлық (жасанды нейрондар) жүйелердің бір-бірімен байланысы. Мұндай процессорлар әдетте өте қарапайым (әсіресе дербес компьютерлерде қолданылатын процессорлармен салыстырғанда). Осындай желінің әрбір процессоры мезгілінде қабылдайтын сигналдармен ғана жұмыс істейді және ол басқа өңдеушілерге мерзімді түрде жіберетін сигналдармен жұмыс істейді. Дегенмен, бақыланатын өзара әрекеттестігі бар өте үлкен желіге қосылғандықтан, бұл процессорлар өте күрделі тапсырмаларды орындауға қабілетті, өйткені жүйеде нейрондық желілер дайындалады.

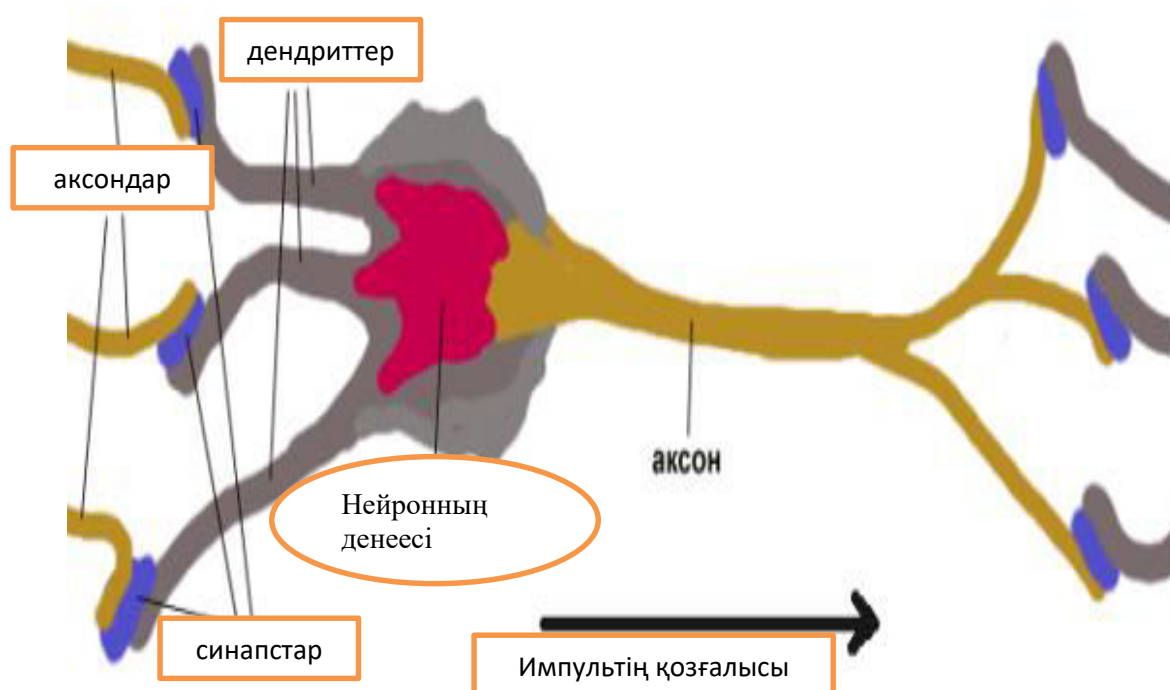
Нейрондық желілер кең ауқымды деректерді өңдеу және талдау міндеттерін шеше алады - үлгіні тану және жіктеу, болжау, бақылау және т.б.

Ең жиі пайдаланылатын бағдарламалар нейроимитаторлар болып табылады – олар қарапайым компьютерлерде жұмыс істейді, және қазіргі заманғы нейроалгоритмдер үлкен көлемдегі ақпаратты жылдам өңдеуге мүмкіндік береді. Күрделі нақты проблемаларды шешу үшін нейрондық желілерді пайдаланылады. Бағдарламалар нейрондық желілерді құру, үйрету және манипуляциялау, бастапқы деректер, нейрондық желілердің қасиеттері және нейрондық шешімдер, сондай-ақ ең қарапайым әрекеттер тізбектерін орындаудың автоматтандырылған процедуралары, мысалы, оңтайлы желі параметрлерін және оқу алгоритмін анықтауға арналған негізгі әрекеттер жиынтығын қамтиды. Нейрондық желілерді бағдарламалау бағдарламалық кодты жазу емес, желіні оқытуды білдіреді. Дәл осы оқыту арқасында желі

деректер (кіріс және шығыс) арасындағы тәуелділікті анықтауға, қорытындылауға, нәтижелерді жеңілдетуге, күрделі есептерді анағұрлым оңай бөлуге арналған білімді пайдалануға мүмкіндік береді.

Биологиялық нейрон және жасанды нейрон

Адам миы және оның жүйке жүйесі жүйке талшықтарымен байланыстырылған нейрондардан тұрады. Нейрондар арасында жүйке талшықтарының көмегімен электр импульстері беріледі. Тірі ағзамен кездесетін барлық әрекеттер, терінің, көздің, аурудың, ойлау процестерінің барлық тітіркенуі нейрондардың өзара әрекетінен туындайды. Биологиялық нейронның құрылысы 1-суретте көрсетілген.



1-сурет – Биологиялық нейрон

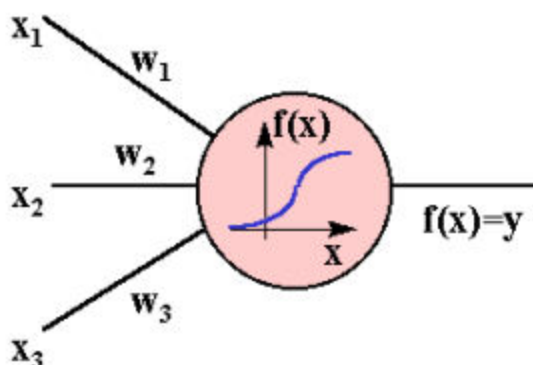
Дендрит – нейронның импульстарын қабылдайды;

Аксон – нейрон импульсін береді;

Синапстар – аксон мен дендрит байланысы үшін импульс күшіне әсер ететін түзілімдер.

Синапстан өткен кезде импульс күші белгілі бір санға өзгереді (синапс салмағы). Нейронға бірнеше дендриттер бойынша импульстер түскенде, олар қосылады. Егер жиынтық импульс табалдырықтан асып кетсе, онда нейрон қозу жағдайына ауысады, өз импульсін қалыптастырады және оны одан әрі аксон бойымен жібереді. Тиісті нейронның мінез-құлқы өзгеруі мүмкін, себебі

синапстардың салмағы уақытпен өзгереді. Сипатталған процестің математикалық модельі 2-суретте көрсетілген.

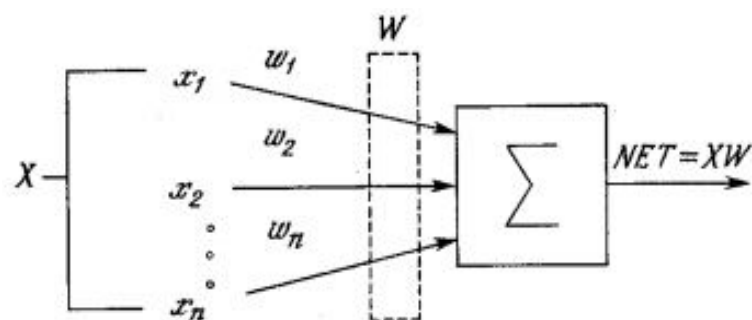


2-сурет – Нейронның математикалық модельі

Бұл модель үш кіріс (дендерриттер) бар нейронды сипаттайды, онда синапстардың w_1, w_2, w_3 салмақтары бар, оларға сәйкес x_1, x_2, x_3 күші тиісінше сәйкес келеді.

Алынған жиынтық импульс $x = x_1w_1 + x_2w_2 + x_3w_3$ нейрон $F(x)$ өткізу функциясына сәйкес түрлендіреді.

$y = f(x) = f(x_1w_1 + x_2w_2 + x_3w_3)$ – шығу импульсінің қуаты. Қорытындылай келе, кіріс түрінде x_k (вектор) сандар жиынтығын аламыз. Содан кейін нейрон шығу кезінде кейбір санды береді. Жасанды нейрон келесідей көрінеді: оның кіруіне көптеген сигналдар түседі, олардың әрқайсысы бір мезгілде басқа нейронның шығуы болып табылады. Мұндай кіріс тиісті салмаққа көбейтіледі, содан кейін туындылар нейронды белсендіру деңгейін анықтайды. Бұл идеяны іске асыратын модель 3-суретте көрсетілген:



3-сурет – Нейронды белсендіру модельі

Мұнда x_1, x_2, \dots, x_n сигналдар жиынтығы жасанды нейронның кірісіне өтеді, олардың жиынтығында олар X векторымен белгіленеді. Бұл сигналдар кіретін биологиялық нейронға ұқсас. Сонымен қатар, сигнал да тиісті салмақ w_1, w_2, \dots, w_n көбейтіледі, содан кейін Σ жинақтау бірлігіне қосылады. Әрбір салмақ бір синаптикалық қосылымның күшіне тең биологиялық нейрон.

Әдетте NET деп аталатын шығу өлшенген элементтер алгебралық түрде қалыптасатын жиынтықтаушы блоктан құрылады. NET сигналы активтендіру деп аталатын қарапайым сызықты функциямен түрлендіріледі. Ол F белгіленеді және OUT шығыс сигналын береді.

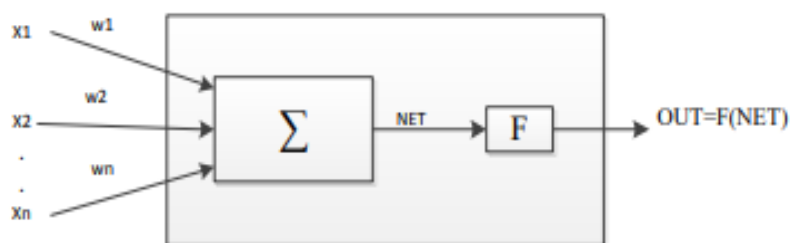
$$OUT = K(NET),$$

K - тұрақты, шекті функциялар,

Егер $NET > T$, онда $OUT = 1$

Қалған жағдайда $OUT = 0$

T - нейрондық желінің нақты модельдейтін тұрақты шекті шама. 4-суретте белсендіру функциясы бар жасанды нейрон көрсетілген:



4-сурет – Белсендірілген нейрон функциясы

F деп белгіленген блок, NET сигналын қабылдайды және OUT сигналын шығарады. F – кез келген NET мәндері үшін OUT мәндері кейбір соңғы аралыққа тиесілі болса, қысу функциясы деп аталады.

Жасанды нейрон моделі биологиялық нейронның көптеген қасиеттерін елемейді. Мысалы, жүйенің динамикасына әсер ететін уақыт кідірісі.

Кіріс сигналдары бірден шығу сигналын тудырады. Сонымен қатар, жасанды нейрон биологиялық нейронды синхрондау функциясының әсерін ескермейді. Дегенмен, тірі нейронның және жасанды нейронның ерекше ұқсастығын атап өту керек. Ақпараттық технологиялар саласындағы нейрондық желілердің орнын анықтау үшін жасанды зияткерлік жүйелердің классификациясына жүгіну қажет.

Жасанды зияткерлік жүйелердің жіктелуі

Жасанды зияткерлік жүйелерге тән ерекшеліктер:

Біріншіден, коммуникативтік дағдылар дамытады пайдаланушының компьютері жүйемен өзара әрекеттесу жолын сипаттайды. Зияткерлік жүйемен диалогта еркін сұраныспен жүйеге жүгіну мүмкіндігі жоқ. Сонымен бірге зияткерлік(интеллектуалдық) жүйенің тілі табиғи тілге барынша жақын болуы керек.

Екіншіден, нашар рәсімделген мәселелерді шешу, яғни, белгілі бір шешім жоқ, бірақ жағдайға, қолданыстағы деректерге және түпкілікті нәтижеге

байланысты стандартты емес тәсілдерді талап етеді. Жасанды нейрондық желілер көмегімен нашар формализацияланатын (қалыптасатын) тапсырмалар тиімді шешіледі.

Үшіншіден, өзін-өзі зерттеу мүмкіндігі – яғни, белгілі бір жағдайдағы жинақталған тәжірибеден интеллектуалды жүйені білу немесе түсіну қабілеті. Жүйені алдын ала оқыту үшін өңделген бастапқы деректер қажет.

Ұсынылған ерекшеліктерге сәйкес, интеллектуалды жүйелерді 1-кестедегідей бөлуге болады.

Кесте-1 – Жасанды зияткерлік (интеллектуалды) жүйелердің түрлері:

Жасанды зияткерлік жүйе түрі	Жасанды зияткерлік жүйенің типі
Коммутативті қабілеті бар жүйелер	- зияткерлік деректер қорлары; табиғи тілдік интерфейстер; - гипермәтіндік жүйелер; мәтіндік анықтамалық жүйелер; - танымдық графика.
Сараптамалық жүйелер	-жіктеуші жүйелер; - алдын ала анықтау жүйелері; - трансформациялаушы жүйелер; - көп антенналық жүйелер.
Өзін-өзі оқыту жүйесі	- индуктивті жүйелер; - нейрондық жүйелер; - прецеденттердегі жүйелер; - ақпараттық қоймалар;
Адаптивті жүйелер	– CASE - технологиялар; - компоненттік технология.

Зияткерлік дерекқорлар үнемі дерекқордан сұраныс бойынша, анық сақталмайтын, бірақ дерекқордан алынуы мүмкін қажетті ақпаратты алу мүмкіндігімен ерекшеленеді.

Тілдік интерфейс табиғи тілдің құрылымын білім берудің ішкі деңгейіне түрлендіреді. Ол зияткерлік дерекқорларға, құжаттық мәтіндік ақпаратты контекстік іздеуге, басқару жүйелеріндегі дауысты енгізу командаларын, шет тілдерінен машиналық аударма үшін пайдаланылады.

Гипермәтіндік жүйелер мәтіндік ақпараттың дерекқорларында пайдаланылады, онда кілт сөзді іздеу қажет, кілт сөздерді күрделі семантикалық ұйымдастыру қажет.

Мәтіндік анықтамалық жүйелерде пайдаланушы проблеманы (жағдайды) сипаттайды, ал жүйе қосымша диалогтың көмегімен оны нақтылайды және осы үшін қолайлы ұсыныстарды іздестірулерді орындайды. Мұндай жүйелер

құжаттама жүйелеріне қосымша ретінде құрылады және білімді тарату жүйелерінің класына жатады.

Көрнекі және интеграцияланған түрдегі графикалық бейнелер оқылатын жағдайдың көптеген параметрлерін сипаттайды.

Эксперттік жүйелер жинақталған білім базасы негізінде мәселелерді шешуге арналған, ол қарастырылып жатқан проблемалық салада сарапшылардың тәжірибесін көрсетеді.

Көптеген агент жүйелер-динамикалық негізде алынған нәтижелерді алмасу арқылы білімнің бірнеше түрлі көздеріне негізделген интеграциямен сипатталатын динамикалық жүйелер.

Өзін-өзі оқыту жүйесі нақты өмір жағдайларын автоматты түрде жіктеу әдістеріне негізделген.

Өзін-өзі тану жүйелерінің сипаттамалары:

- өзін-өзі оқыту жүйесі «мұғаліммен», әрбір мысал үшін белгілі бір жағдайға байланысты оның атрибуты құндылығы анық белгіленеді.
- «мұғалімсіз» өзін-өзі оқытатын жүйелер классификация белгілері мәндерінің жақындығы бойынша жүйе жағдайлардың сыныптарын өзі бөледі.

Индуктивтік жүйелер жеке бөліктен жалпы мысалдарды қорытады, ал жинақтау процесі былайша жүзеге асырылады:

- 1) Берілген жиындардан жіктеу белгісі таңдалады (дәйекті немесе ереже бойынша).
- 2) Таңдалған функцияның мәніне сәйкес, көптеген мысалдар жиындарға бөлінеді.
- 3) Мысалдың бір класқа жататындығы туралы тексеру жүргізіледі.
- 4) Егер мысалдардың қандай да бір ішкі жиыны бір ішкі сыныпқа тиесілі болса, яғни ішкі жиынның барлық мысалдары класс құраушы белгінің мәні сәйкес келсе, онда жіктеу процесі аяқталады. (жіктеудің қалған белгілері қарастырылмайды).
- 5) Сәйкес келмейтін мәні бар мысалдар жиыны үшін сыныптау процесі 1-тармақтан бастап жалғасады.

Бұдан әрі шешілетін міндеттер тұрғысынан зияткерлік ақпараттық жүйелерге жүгіну керек. Мұнда келесі жүйелерді ерекшелеуге болады:

- басқару жүйелері және анықтамалық жүйелер;
- компьютерлік лингвистика жүйесі;
- тану жүйелері;
- ойын жүйелері;
- зияткерлік ақпараттық жүйелерді құру жүйелері.

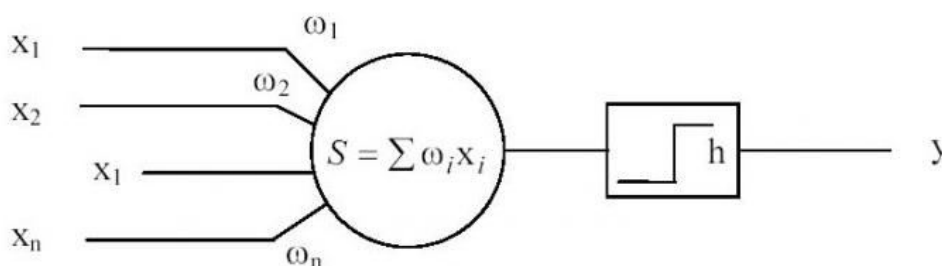
1.5 Нейрондық желілер теориясының элементтері

Нейрондық желі – бұл бір түрдегі көптеген элементтерден тұратын есептеу құрылымы. Бұл элементтер қарапайым функцияларды орындайды, ал жасанды нейрондық желіде болып жатқан барлық процестерді тірі организмдердің жүйке жүйесінде болып жатқан процестермен байланыстыруға болады. Себебі нейрондық желі жұмысы адамның бас миының жұмыс жасау принципіне ұқсас жұмыс жасайды.

Нейрондық желілер өз табиғаты бойынша сызықсыз болып келеді, оларда әзірленген технологияны (нейрожелінің ақпараттық модельін) бірден пайдалануға мүмкіндік жоқ. Көптеген жылдар бойы сызықтық модельдеу модельдеудің негізгі әдісі болды, себебі ол үшін оңтайландыру процедуралары жақсы әзірленген.

Нейрондық желілерде білімдер нейротәрізді элементтердің (немесе жай ғана нейрондар) жиынының күйінде және олардың арасындағы байланыстарда сақталады.

Мак-Каллок-Питтс нейронының қазіргі кезде де жеке нейронды сипаттау үшін қолданылып жүрген формалды моделі 5-суретте көрсетілген.



5-сурет – Мак – Каллок - Питтс нейронның формальді моделі

Мұндағы x_i – нейронның i кірісіндегі (синапс) сигнал;

w_i – нейронның i кірісіндегі (синапс) салмақ;

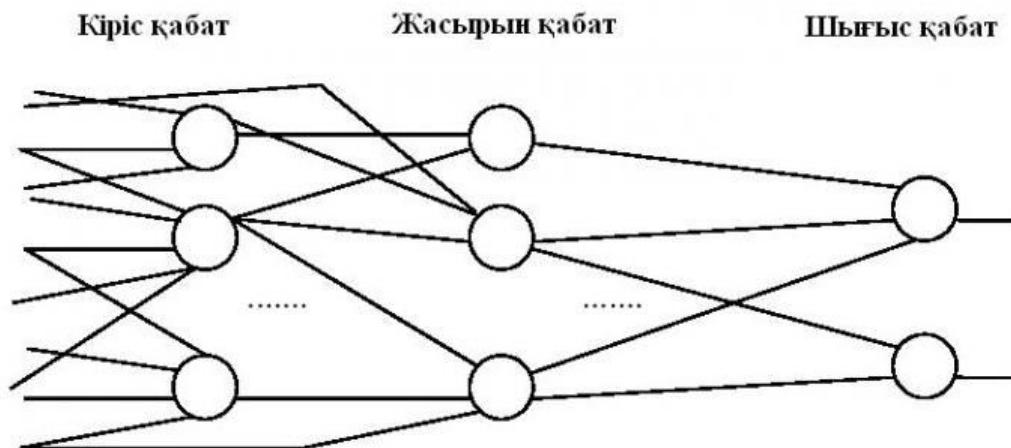
y – нейронның шығысы;

h – нейронның жұмыс жасау табалдырығы.

Модельде нейронның кірісіндегі сигналдар сомасы h табалдырықты мәнімен салыстырып, егер ол сол мәннен артық болса, шығыста сигнал қалыптасады. Нейрондардың қазіргі модельдерінде табалдырықты функция жағдайда беру функциясы немесе нейронды белсенді ету функциясы деп аталатын сызықты емес $y = "f(S)"$ функциясымен алмастырылады.

Белгілі бір түрде байланысқан нейрондардан қандай да бір кірістер және шығыстар саны бар нейрондық желі құрылды. Әдетте түйіндердің (нейрондардың) үш типін ажыратады – кіріс (нейрондардың кіріс қабаты немесе Input Layer), шығыс (шығыс қабат немесе Output Layer), және жасырын қабаттар (Hidden Layers).

Тікелей байланысқан нейрондық желіні 6-суреттен көруге болады.



6-сурет – Тікелей байланысты нейрондық желі

Нейрондық желі жұмысы екі кезеңнен тұрады: желіні кіріс ақпаратқа(кіріс вектор) «дұрыс» немесе адекватты жауап беруге үйрету және оқытылған желіні кіріс векторларын тану үшін қолдану. Соңғы кезеңді жиі жағдайда тестілеу деп атайды. Басқаша айтқанда, желі кіріс векторларды тануға, яғни кіріс векторлардың танылған класына сай келетін шығыс векторларды жасауға үйретіледі. Бұл кезде кіріс векторларының шығыс векторларына сәйкестігіне қатысты білімдер синапстардың салмағы мен нейрондардың табалдырығында сақталады. Коннективистік тәсіл жағдайында (коннекцианизм) – тек қана синапстардың салмағында сақталады. Кейде кіріс векторы ретінде кіріс және шығыс векторларының конкатенациясы түсінеміз және бұл вектордың барлық разряды желіні оқыту және тестілеуді пайдаланылмауы мүмкін. Нейрондық желілердің кейбір модельдерінде (мысалы, Хопфилд моделі) кіріс және шығыс сигналдар ажыратылмайды және оларға сәйкес кірістер (шығыстар) желінің жұмысы барысында рольдерімен алмаса алады.

1957 жылы Джон фон Нейман нейрондық желінің перцепрондық нейрондық желіден мүлдем бөлек архитектурасын жасады. Оның берілген жұмысы үшін маңызды болып табылатын қасиеттері келесі болып табылады.

- Нейрон кіріс сигналдарын өңдейтін қандай да бір қарапайым құрылғы.
- Түпнұсқада конъюнкция, дизъюнкция және инверсиялар қолданылған.
- Нейрон орындалатын функцияны динамикалық түрде өзгерте алады. Бір нейрон әртүрлі уақытта әртүрлі функцияларды орындай алады. Уақыттың бір мезгілінде мүмкін болатын функциялардың біреуін ғана орындай алады.
- Нейрон басқа нейрондармен байланысуын динамикалық түрде өзгерте алады.
- Нейрондық желінің бір бөлігі желінің басқа бөлігінің күйіне талдау жасай алады.

- Нейрондық желінің бір бөлігі желінің басқа бөлігінің топологиясын өзгерте алады.

Функциялардың математикалық толық жиынында олардың күрделілігі тек есептеу тиімділігінде көрініс табады. Фон Нейман өз жұмысында көрсеткендей, мұндай нейрондық желі Тьюринг машинасына эквивалентті болып табылады. Бұл кезде тек қана бірденгейлі метаөтілім керек, оның нәтижесінде мұндай нейрондық желі Тьюринг машинасына эквивалентті ақырлы автоматты және программалы таспаны тікелей жүзеге асырады.

Фон Нейман желісінде бастысы – нейрон орындайтын функция емес.

Бастысы – өздік рефлекс пен өздік модификацияға қабілеттілік болып табылады.

Мысалы, желіде қандай да бір құрылым қалыптасты делік. Сипатталып отырған желіде нейрондық желінің бір бөлігі басқа бөліктің құрылымына талдау жасай алады. Сосын осы талдаудың негізінде қандай да бір шешімдерді қабылдап нейрондардың типтерін немесе байланыстарға өзгеріс енгізе алады.

Нейрондық желінің бір бөлігі нейрондық желінің басқа бөлігін жеке нейрон – ұяшықтарға динамикалық түрде қатынас құра отырып, олардың күйін оқып, немесе өзгерту арқылы жады банкісі ретінде пайдалана алады.

ДНҚ молекуласы мен мұндай автоматтағы таспа аналогиясын жүргізу қызық. Оның үстіне, нейрондық желінің бір бөлігі «құрастырушы жең» шығарып жеке нейрондардан кейбір функцияны орындайтын құрылғыны жинай алады.

Бұл кезде «таспада» орналасқан жады негізінде желінің қандай да бір бөлігі құралатын ДНҚ ретінде қолданылады.

Семантикалық нейрондық желілер мен Фон Нейманның идеяларын осы желілерге қолдану барысында пайда болатын салдарларды қарастырайық.

Фон Нейман желісінде байланыстар топологиясына шектеу қойылады және тек логикалық мәндер ғана өңделеді. Фон Нейман желісінің негізінде құрылған семантикалық нейрондық желіде байланыстар топологиясына шектеу жоқ және нақты емес мәндерді өңдеу орындалады. Фон Нейман желісіндегі барлық нейрондар тактілермен синхрондалады. Семантикалық нейрондық желілерде синхрондалған және синхрондалмаған нейрондар бар.

Фон Нейман желісінен айырмашылығы, семантикалық нейрондық желілерде нейрондардың топологиясына шек қойылмайды, бұл Фон Нейман жасағандай нейрондардың қатыстық адрестелуін мүмкін емес етеді. Бұл жағдайда абсолютті адрестеуді енгізу керек.

Семантикалық нейрондық желі қазіргі есептеу техникасының жабдықтарымен тиімді жүзеге асырыла алады. Семантикалық нейрондық желі жасушасының постулатталған еркіндігінің арқасында Тьюринг машинасына эквивалентті. Бұл оның негізінде Тьюринг машиналарында есептелінетін кез-келген функцияны есептейтін жүйені жасауға болады. Мысалы, мұндай желі жеке жағдай қатені кері тарататын қабатты персептронды модельдей алады.

Осылайша персептронның нейрондарын мұндай желінің қосу, көбейту және белсендіру функциясын жеке орындайтын дербес нейрондардан

құрастыруға болады. Персептронды оқыту алгоритмін мұндай желінің персептронға сай келетін үзіндісін талдайтын және өзгертетін жеке үзінді ретінде жүзеге асыруға болады.

1.6 Машиналық оқытудың қолдану ауқымы

Нейрондық желілерді машинаға көшіру немесе нейрондық желілер негізінде машина жасау жөніндегі ой алғаш рет өткен ғасырдың 40-жылдарында туындаған. Ол кездері ми туралы таным-білім бүгінгідей кең ауқымда болмаған. Және есептеуіш техникалардың қуаты төмен еді. Бүгінде ақпаратты компьютерге енгізу қиын дүние емес. Оны арнайы бағдарлама, кодтар арқылы жасауға болады, Адам миының толықтай бағдарламалық көшірмесін жасау әліге дейін мүмкін болмағанымен, нейрондық желілерді белгілі бір мақсатта қолданатын бағдарламалар бар. Оған мысал – VI Group компаниясының офисында орнатылған камералар. Ол камералар ғимаратқа кіретін жерде орнатылған. Олар адамдарға және көліктерге арналып жасалған. Егер офис ішіне қандай да бір адам келетін болса, ол адам камераға қарап өтуі керек, егер ол адам VI Group компаниясында жұмыс жасайтын болса, онда есік ашылады, егер жұмыс жасамайтын болса, есік сол жердегі қызметкердің рұқсатымен ғана ашылады. Машиналарға арналған жүйеде осылай жұмыс жасайды.

Машиналық оқыту – бұл ғылымдағы үрдіс, ал келешекте компьютерлік оқыту мәселесін шешетін технологиялар. Бірден айтқым келеді, бұл толыққанды оқытуға жол берілмейді, бұл адамның оқытуымен салыстыруға болады. Оқу процесінде қалыптасқан машина білімі адам қабілетіне қарай нақты ақылды шешімдерді жасай алмайды. Осыған қарамастан, машина жасау қажеттілігі күрт өсуде. Машиналық оқытудың тек математиктер мен алгоритмдер саласынан кетіп, IT - бизнесмендер әлеміне, содан кейін қарапайым адамдардың әлеміне тереңірек еніп жатыр. Бұл адамға зор пайда әкеледі.

Машиналық оқытудың мақсаты адам қызметінің түрлі салаларында күрделі кәсіби мәселелерді шешудің жартылай немесе толық автоматтандыруы болып табылады. Машиналық оқытудың кең ауқымды қосымшалары бар:

- Санат: Биоинформатика қосымшалары;
- Санат: Медициналық қосымшалар;
- Медициналық диагностика;
- Санат: Геология және геофизикадағы қолданбалар;
- Санат: Әлеуметтану қосымшалары;
- Категория: Экономикадағы өтініштер;
- Несие ұпайлары;
- Тұтынушыларға күтім көрсетуді болжау (болжалды болжау) ;
- Алаяқтықты анықтау;

Қордың техникалық сараптамасы;
Нарықтың қадағалауы;
Санат: Технологиялық қосымшалар;
Техникалық диагностика;
Робототехника;
Компьютер көрінісі;
Сөйлеуді тану;
Категория: Office автоматтандыру қосымшалары;
Мәтінді тану;
Спамды анықтау;
Құжаттарды санаттау;
Қолжазбаны тану;

Машиналық қосымшаларды қолдану ауқымы үнемі кеңейуде. Кең таралған ақпараттандыру ғылым, өндіріс, бизнес, көлік, денсаулық сақтау саласындағы көптеген деректер жинақталуына әкеледі. Бұл жағдайда туындайтын болжау, басқару және шешім қабылдау проблемалары көбінесе прецеденттерден үйренуге азаяды. Бұрын мұндай деректер болмаған кезде, бұл тапсырмалар мүлдем басқа әдістермен шешілмеген немесе мүлде басқа әдістермен шешілген.

Іс-тәжірибелік тапсырмалардың жалпы тұжырымы. Көптеген нысандар (жағдайлар) және көптеген мүмкін жауаптар (жауаптар, реакциялар) бар. Жауаптар мен нысандар арасында қандай да бір корреляция бар, бірақ ол белгісіз. Тек прецеденттердің түпкілікті жиынтығы - оқыту нысаны деп аталатын «нысан, жауап» жұптары белгілі. Осы деректер негізінде, кез-келген ықтимал кіріс нысаны үшін жеткілікті нақты жіктеуіш жауапты құруға қабілетті алгоритм салу керек, яғни жабық тәуелділікті қалпына келтіру керек. Бұл тәуелділік міндетті түрде аналитикалық түрде көрсетілмейді және мұнда нейрондық желілер эмпирикалық қалыптасқан шешімнің принципін іске асырады. Мұның маңызды ерекшелігі - дайындалған жүйені жалпылама білдіру қабілеттілігі, яғни қолданыстағы жаттығу жиынынан тыс деректерге барабар жауап беру. Жауаптардың дұрыстығын өлшеу үшін бағалаудың сапа функциясы енгізіледі.

Бұл тұжырым функциялардың жуықтауының классикалық мәселелерін қорыту болып табылады. Аппроксимацияның классикалық мәселелерінде объектілер нақты сандар немесе векторлар болып табылады. Нақты қосымшаларда нысандар туралы деректер енгізу толық емес, дәл емес, сандық емес, гетерогенді болуы мүмкін. Бұл мүмкіндіктер машинада оқыту әдістерінің алуан түріне әкеледі.

Терең оқыту (deep learning) – деректерді тереңірек түсінуді үйренетін машиналық оқыту алгоритмдерінің классы.

Болжаудың мәселесі нысандар болашақ үшін болжам жасау қажет болған сәтте аяқталатын уақыттық сериялардың сегменттері болып табылады. Болжаудың проблемаларын шешу үшін регрессиялық немесе жіктеу әдістерін

бейімдеу жиі мүмкін болады, ал екінші жағдайда шешім қабылдау міндеттері туралы көп айтылады.

Бақыланбаған оқыту – бұл жағдайда жауаптар орнатылмаған және сіз нысандар арасындағы тәуелділіктерді іздегіңіз келеді. Кластерлеу міндеті – объектілерді буындардың ұқсастығы туралы деректерді пайдаланып кластерлерге топтау. Сапа функционалдығын әртүрлі түрде анықтауға болады, мысалы, орташа қапсырмалардың және іштей қашықтықтардың арақатынасы.

Қауымдастық ережесін табу міндеті (қауымдастықтың ережелерін үйрену). Бастапқы деректер сипаттамалар түрінде берілген. Нысандар сипаттамасында жиі кездесетін белгілердің осындай белгілерін және осындай белгілерді табу қажет.

Сенімділік доменін құру міндеті (кванттық бағалау) – берілген үлгілік фракциясы бар өте тегіс шекарасы бар ең аз көлемді аймақ.

Өлшемділікті төмендету мақсаты – кейбір конверсиялау функцияларының көмегімен, үлгі нысандар туралы маңызды ақпаратты жоғалтпастан бастапқы ерекшеліктерге негізделген жаңа мүмкіндіктердің ең аз санына өту. Сызықтық қайта құрулар классында ең танымал мысал негізгі компоненттік әдіс болып табылады.

Жетіспейтін мәндерді толтыру міндеті матрицасында жоғалған мәндерді ауыстыру болып табылады - олардың болжамды мәндері бойынша белгілер.

Жартылай оқыту (жартылай бақыланатын оқыту) мұғаліммен немесе мұғалімсіз оқытудың аралық орнын алады. Әрбір пайдалану жағдайы – «жауап» жұбы, бірақ жауаптар пайдалану жағдайларының бөліктері ғана белгілі. Қолданбалы міндеттің мысалы олардың кейбіреуі қазірдің өзінде кейбір рубрикаларға тағайындалған жағдайда, көптеген мәтіндерді автоматты түрде айқындау.

Динамикалық оқыту (онлайн-оқыту) мұғаліммен бірге және мұғалімсіз оқыту сияқты болуы мүмкін. Ерекшелену – жағдайлардың ағынын пайдалану. Әрбір прецедент туралы дереу шешім қабылдау және сол уақытта жаңа прецеденттерді ескере отырып, тәуелділікті модельді аяқтау қажет. Міндеттерді болжау кезінде, уақыт факторы маңызды рөл атқарады.

Белсенді оқыту оқушының өзін-өзі тану мүмкіндігіне ие болуымен сипатталады.

2 Нейрондық желілердің ақпараттық моделі және адамның бет-әлпет эмоциясын тану жүйесі

2.1 Нейрондық желілердің жұмыс жасау принципі

Нейрондық желілер дейндрит арқылы ақпаратты қабылдап, аксон арқылы желідегі келесі нейронға жіберетін нейрон жұмысына қарапайым мысал келтіріп көрелік. Сіз досыңызбен бір дәмхананың жазғы алаңқайында орырсыз делік. Сыртта отырғандықтан, құлағыңыздың дабыл жарғағындағы тербелістер арқылы сезілетін түрлі дыбыстарды естисіз. Ол ішкі құлаққа тербеліс жібереді. Өз кезегінде тербелісті қабылдайтын түкшелі жасушалар арнайы сұйықтықта тербеледі. Осы жасушалар нейрондарға сигналдар жібереді. Тізбектегі бірінші нейрондар тербеліс жиілігін анықтап, мағынасын келесілеріне жодайды. Ал келесі нейрондары шуды түрлеріне қарай ажыратып, ұқсастығы бойынша бірнеше каналға бөледі. Мұнан әрі ми шу туралы қабылданған ақпараттың арасынан көліктің дыбысын ажыратады. Артынша оның жеңіл немес жүк көлігінен шыққан дыбыс екендігін анықтайды. Жинақталған тәжірибеге сүйеніп, миымыз дыбысы арқылы көліктерді қозғалтқышының көлеміне қарай топтастырады. Бірақ досыңызбен әңгімелесіп отырғандықтан, ми досыңыздың сөзін ең басты дыбыс ағымы ретінде ажыратады. Осы кезде көзіңізді жұмсаңызда, досыңыздың сөйлеп жатқандығын түсініп, дауысынан тани аласыз. Оның сөзін жекелей түсініп қана қоймай, интонациясын, көңіл күйін(эмоциясын), әр сөзінің мағынасын түсінесіз. Мұның барлығы ішкі құлағыңыз қабылда, мыңдаған нейрондар өңдеген ауадағы қарапайым тербелістер арқылы жүзеге асады.

Жасанды нейрондық желілердің мақсаты – дәл осы жоғарыдағы процесті қайталау. Бірақ ол арнайы оқусыз мүмкін емес. Біз дүниеге келген бойда сөздер мен өзге заттардың дыбыстарын бірден ажырата алмаймыз. Уақыт өте келе миымызда жинақталған ақпарат, тәжірибелердің көмегімен дыбыстарды ажыратуды үйрене бастаймыз. Өмірге келген сәттен бастап өзін “ана” ретінде таныстыратын әйелді үнемі көреміз. Сонда да “ана” сөзін айту үшін, оны белгілі бір адаммен байланыстыру үшін 1 жылдан астам уақыт қажет. Айтпақшы, адам сөйлеген кезде ауыз қуысында тіл мен тістің арасында қандай байланыс әрекеттестік болып жатқандығын көре алмағандықтан, дыбыстауға байланысты қиындықтар туындайды. Сондықтан оны тәжірибе жүзінде байқауға немесе білім алуға тырысамыз. Дәл осы сияқты, жасанды нейрондық желілерді ұғыну үшін де білім қажет.

2.2 Бір бағытты көп қабатты желілер

Көп қабатты нейрондық желі формальды түрде нейрондар деп аталатын қарапайым өңдеу элементтерінің жиынтығы ретінде анықталуы мүмкін. Нейрондар қабаттар бойынша ұйымдастырылған және синапстар деп аталатын бір бағытты байланыстармен біріктірілген.

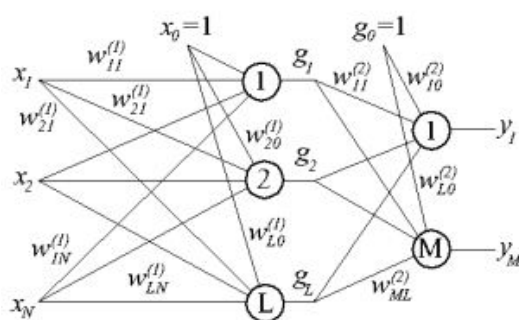
Әдетте желі көптеген сенсорлық элементтерден тұрады (кіріс түйіндер) құрайтын кіріс қабаты; бір немесе бірнеше жасырын және нейрондардың бір шығу қабаты.

Бір бағыттағы көп қабатты нейрондық желілер кең таралған, себебі олардың қарапайым математикалық сипаттамасы бар. 70-ші жылдардың соңынан бастап көп қабатты желілер оқытудың ыңғайлы алгоритмінің арқасында дамыды.

Нейрондық желілеріді бір қабаттан екінші қабатқа тікелей тарату немесе (бір нейрондан екіншісіне) жіберу үшін кіріс сигналдары пайдаланыды. Дәл осындай желілер көп қабатты персептрон деп аталады, ол кіріс қабатынан, жүйенің ішіндегі жасырын есептеуіш қабаттардан және нейрондардың шығу қабатынан тұрады.

Көп қабатты персептрон – бұл бір бағытты желінің сигмоидальды түрі немесе типі.

Сигмоидальды функция – барлық жерде біркелкі өсуші дифференциалды s-бейнелі сызықты емес функция, оны формальды нейронда активтендіру функциясы ретінде қолдануға өте ыңғайлы. Сигмоид әлсіз сигналдарды күшейтіп, күшті сигналдардан қанықпауға мүмкіндік береді. Көп қабатты персептронның құрылымдық сұлбасын 7-суреттен көре аламыз:



7-сурет – Көп қабатты персептронның құрылымдық сұлбасы

Мұндағы:

- $g_l, L=1,2,\dots,l$ – нейрондардың бірінші қабатынан шығатын сигналдар;
- $(m), m=1,2,\dots$ – жақшадағы индекстер жоғарыдағы нейрон қабатының нөмірін білдіреді;
- x_1, x_2, \dots, x_n – кіріс қабатын қалыптастыратын кіріс сигналдары;
- g_1, g_2, \dots, g_n – бірінші жасырын қабат;

– y_1, y_2, \dots, y_m – шығу қабатын қалыптастыратын шығу сигналдары;
Желі құрылымын белгілеуге ыңғайлы болу үшін "N-L-M" кодтамасы.
Нейрон қабаттарының шығыс сигналдарын есептеуге арналған формулалар:

$$g_l = f(\sum_{j=0}^N w_{lj}^{(1)} \cdot x_j), l = 1, 2, \dots, L \quad (1)$$

$$y_i = f(\sum_{l=0}^L w_{il}^{(2)} \cdot g_l) = f(\sum_{l=0}^L w_{il}^{(2)} \cdot f(\sum_{j=0}^N w_{lj}^{(1)} \cdot x_j)), i = 1, 2, \dots, M \quad (2)$$

Жеке оқыту үлгісінің мақсатты функциясы $\langle X, D \rangle$ төмендегідей көрінеді:

$$E(W) = \frac{1}{2} \sum_{i=1}^M (y_i - d_i)^2 \quad (3)$$

2.3 Көп қабатты персептронды оқыту

Көп қабатты персептронды оқыту мақсаты: X кіріс векторы кезінде Y Шығыс векторы мен күтілетін d мәндерінің векторы барынша сәйкес келуі үшін $W(1)$ және $W(2)$ желісі таразыларының мәндерін таңдау қажет.

Персептронды оқыту әдістерінің бірі қателікті кері тарату алгоритмі. Векторда желі қабаттары бойынша өту жолының екі түрі болады – біреуі түзу, екіншісі кері. Бірінші жағдайда вектор кіріс қабатынан қозғала бастайды, одан әрі желі бойынша қабаттан қабатқа қарай қозғалады. Бұл жағдайда желі ішінде шығыс сигналдар жиынтығы жасалынады. Кері өту кезінде синаптикалық салмақ теңдеседі, шығушы желі нақты түрде шегеріледі, осыдан кейін қате туралы сигналы келіп түседі. Одан әрі сигнал қарама-қарсы синаптикалық байланыстар бағытында желі бойынша таратылады. Сондықтан бұл әдіс қатенің кері таралуы деп аталады.

Сигмоидалді типтегі көп қабатты желінің моделі және оның оқыту әдісі салыстырмалы қарапайымдылығы мен қолайлылығына қарамастан, оқу үрдісін баяулататын қиындықтарға ие немесе желінің мүлдем үйренуіне жол бермейді. Мысалы, түзету нәтижесінде таразылардың мәндері өте үлкен шамаға айналу.

Бұл жағдай өндірістегі көптеген нейрондардың үлкен құндылықтар жасайтынына әкеледі, бірақ осы мәндердегі туындылар өте аз болады. Оқу процесінде жіберілген қате осы туындыға пропорционалды болғандықтан, процесс тоқтатылуы мүмкін.

Сонымен қатар, қадамның соңғы өлшемін таңдау қажет. Бұл мәселеде тек тәжірибеге сүйенуге болады. Бұл алгоритм ұқсастықтың (жинақтық) дәлелі бар. Қадам өлшемі тікелей ұқсастыққа әсер етеді: немесе қадам өлшемі өте аз болса,

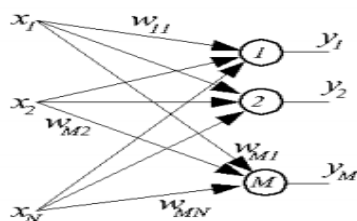
ол тым баяу болады, немесе қадамның тым үлкен мөлшеріне байланысты тұрақты тұрақсыздыққа ұшырайды.

Тағы бір қиындық – уақытша тұрақсыздық. Желі әріптерді тани алады, бірақ егер біреуін үйренсе, екіншісін ұмытып кетсе жаттығу мағынасы болмайды.

2.4 Кохонен желілері

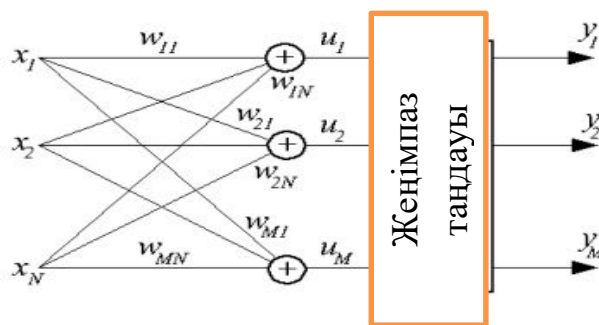
Жіктеудің әр түрлі міндеттерін шешу үшін Кохонен желілері деп аталатын нейрондық желілердің жеке сыныбы қолданылады.

Жіктеудің негізгі міндеті қандай да бір белгілер бойынша нысандарды сыныпқа бөлуден тұрады. Мұндай бөлуді негізі нысанның векторлы параметрлері болып табылады. Мұндай желілер бәсекелестік негізінде өзін-өзі ұйымдастыру желілері деп те аталады. Шығуда ең үлкен сигнал жеке болады, қалған сигналдары нөлге барады. Кохонен желісін 8-суретте көрсетілген.



8-сурет – Кохонен желісі

Желі ЖБА (Жеңімпаз Барлығын Алады) түріндегі нейрондар негізінде салынған, олар 9-суреттегідей көрінеді.



9-сурет – Жеңімпаз барлығын алады

Барлық бәсекелес нейрондар үшін кіріс сигналдары бірдей, шығыс сигналдары мынадай формула бойынша есептеледі:

$$u_i = \sum_{j=0}^n (w_{ij}) * x_j \quad (4)$$

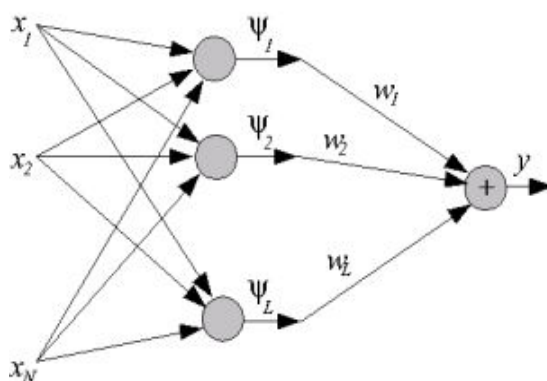
Нәтижелерді салыстырғаннан кейін, жеңіске жеткен нейрон таңдалады (ең үлкен мән u_i). Шығу сигнал u_i алады мәні 1, жеңіске жеткен жағдайда, қалған барлық нейрондық сигналдар 0 мәнін алады.

2.5 Радиалды нейрондық желілер

Радиалды базистік функциялар желісі – симметриялық нейрондардың аралық (жасырын) қабаты бар сигналдың тікелей таралуының нейрондық желісі болып табылады. Мұндай нейрон осы кіріс векторынан оған сәйкес келетін "орталыққа" дейінгі қашықтықты кейбір сызықсыз заң бойынша түрлендіреді. Радиалды нейрондық желілер радиалды нейрондардың көмегімен жасалады. Өз орталығының маңында осы нейрондарды белсендіру функциясы нөлдік емес. Бұл жергілікті аппроксимация деп аталады.

Аппроксимация – объектілерді бастапқы түрге ауыстыратын, кейбір мағынада жеңілдетілген түрде ауыстыратын ғылыми әдіс депте айтылады. Математикалық тұрғыдан қарағанда, көптеген сигналдық типті сигмоидальды желілер бірнеше X R M айнымалыларының функциясын Y R M. айнымалы мәндер жиынтығына айналдырады.

Сигмоидаль функциясы нейрондарды белсендіру функциясының рөлін атқарады. Кіріс деректерінің бүкіл диапазонынан нөлдік мәнге ие болғандықтан, оның барлық нейрондары кіріс деректерін желі ішіндегі шығыс деректеріне түрлендіруге қатысады. Радиалды желі екі қабатқа ие: бірінші қабат радиалды нейрондардан тұрады, ал екінші (шығу) бір нейроннан немесе бірнеше сызықты тұрады. 10-суретте Радиалды желі сұлбасы.



10-сурет – Радиалды желі сұлбасы

$\Psi_i(\|X - C_i\|)$ – i -ші радиалды нейронның базистік функциясы немесе белсендіру функциясы.

Радиалды нейрондық желілер оқу процесі үшін бастапқы шарттарды таңдауды жеңілдетеді, оңтайлы шешімге жылдам қол жеткізуді қамтамасыз етеді.

Радиалды желілердің алынған архитектурасы құрылымы бар, бір жасырын қабаты бар, сигмоидальдық желілердің көп қабатты құрылымына ұқсас. Аталған ұқсастыққа қарамастан, бұл типтердің желілері бір-бірінен мүлдем ерекшеленеді. Мұндай желілер сигмоидальді желілерімен салыстырғанда қорыту қабілетін әлсіз. Бұл жетіспеушілік оқыту сыныптарын ішкі сыныпқа бөлу есебінен өтеледі. Бұл ретте әрбір ішкі сыныптың айналасында радиалды базистік функция іске асырылатын өзінің "орталығы" болады. Сонымен қатар, радиалды желілерді оқыту алгоритмі бірегей. Тек бір жасырын қабат және нейронның оқыту деректері саласымен тығыз байланысы болған жағдайда, оқудың бастапқы кезеңіндегі нүкте оңтайлы шешімге жақын болады.

2-ші кестеде нейрондық желілердің бөлінген ақпараттық модельдері бойынша екінші тараудың салыстырмалы талдауы көрсетілген. Кестеде нейрондық желілердің негізгі ақпараттық модельдері (көлденеңінен) көрсетілген, жасанды нейрондық желілерді пайдалана отырып шешуге болатын негізгі міндеттер тігінен көрсетілген.

Кесте-2 – НЖА(Нейрондық желілерге ақпарат)

Міндеттер \ НЖА	Бір бағытты көп қабатты желілер	Рекуррентті нейрондық желілер	Кохонен Желісі	Радиалды нейрондық желілер
Ассоциативті жады	+	+	-	+
Ақпаратты қысу	+	+	-	-
Болжау	+	+	+	+
Оңтайландыру	-	+	+	
Сыныптама (Класификация)	+	-	+	+
Кластерлеу (Кластеризация)	-	-	+	+

Ассоциативті жады – бұл машина жадының ерекше түрі, өте жылдам іздеу қосымшаларында қолданылады.

Ақпаратты қысу – алынған көлемді азайту үшін ақпаратты алгоритмдік қысу. Деректерді сақтау және беру құрылғыларын тиімді пайдалану үшін қолданылады.

Болжамдау – жинақталған тәжірибе мен ағымдағы болжамдар негізінде дамудың ықтимал жолдарын бағалау.

Оңтайландыру – берілген шарттар мен шектеулер кезінде есептің ең жақсы шешімін табу.

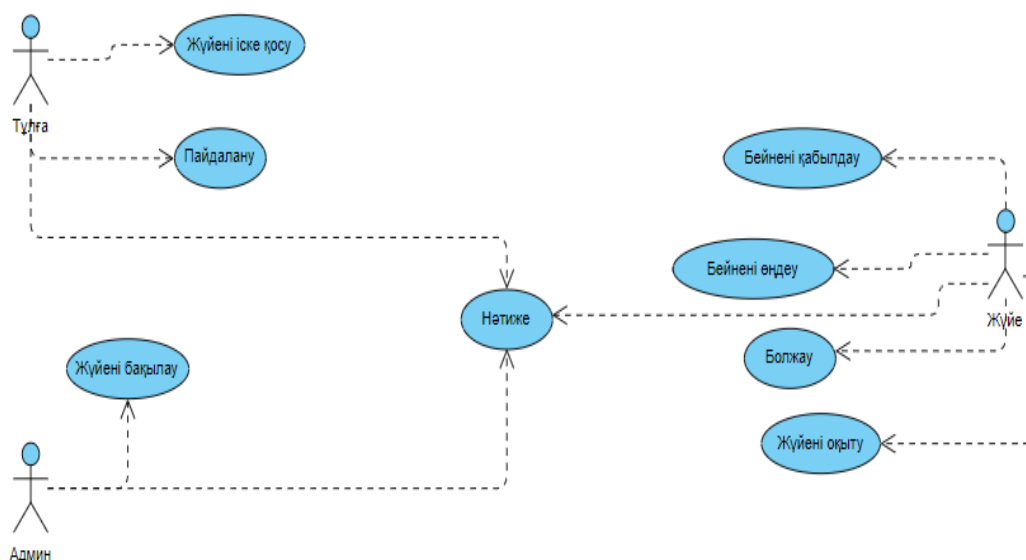
Сыныптама(Классификация) – қандай да бір белгілері бойынша нысандарды (объектілерді) кластарға бөлу жолымен міндеттерді шешу.

Кластерлеу – әрбір кластер ұқсас нысандардан тұратын кластерлер деп аталатын бөлінбейтін ішкі жиындарға бөлінген нысандарды бөлуді талап ететін міндет.

2.6 Функционалды диаграммалар

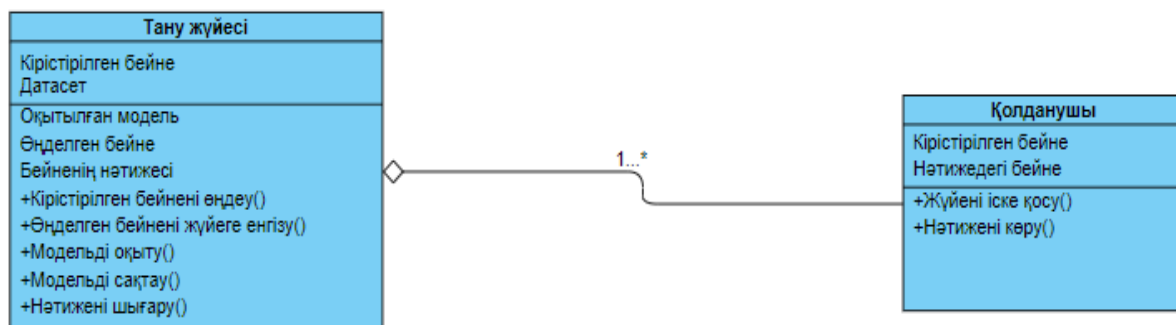
Өңдеу функционалдығын сипаттау – қолдану варианттар диаграммасы, функция орындалу реттері – әрекеттердің тізбектер диаграммасы (талаптар және шектеулер), өңдеу элементтерінің бір-бірімен байланыс және бар болу сипаттамасы – кооперативті диаграммалар (талаптар және шектеулер). Интерфейсті жобалау. Объектілік жобаның өңделуі, ОМ 6 қосымшасына сәйкес, негізінен «прецеденттер диаграммасы» немесе екінші аталуы – «Қолдану варианттар диаграммасы» (ҚВД) басталады. Прецеденттер диаграммасы барлық қалған жоба диаграммалары үшін арналып жасалады. Ал оның екінші аталуы, негізінен, оның атқаратын ролін толық көрсетеді деп айтуға болады – пайдаланушы жүйесінің қолдану варианттар диаграммасы (актерлар, әрекет жасаушы объект).

2.6.1 Use Case диаграммасы бойынша



11-сурет – Use Case диаграммасы

2.6.2 Класс(Сынып) диаграммасы бойынша



12-сурет – Класс(Сынып) диаграммасы

3 Нейрондық желі қолданып бет-әлпеттегі эмоцияны тану

3.1 Әдістерді пайдаланып эмоцияны тануды белсендіру

Нейрондық желілерді қолданып көптеген қызметтік құрылғылар немесе қоғамға пайдасы бар бағдарламалар жасауға болады және қазіргі уақытта жасалыпта жатыр. Сол бағдарламалардың бірі нейрондық желілер көмегімен адамның бет-әлпет эмоциясын тану бағдарламасы.

Эмоция деген барлық тірі жан-жануар, жәндіктер, жорғалаушы, бүкіл түрі жан иесінде болатын белгілі.

Бұл бағдарламаны бастамас бұрын, нейрондық желілерді осы бағдарламаға оқыту қажет, сол оқытылған жүйе бойынша тесттілеу керек, сондай-ақ тесттіленген жүйені модельдеп оны қолданысқа шығаруға әбден болады. Эмоцияны тану бірнеше этаптардан тұрады:

- 1) Бағдарламаға оңтайлы келетін және қажет болған кітапханаларды және тілдерді орнату.
- 2) Бағдарламаны тесттен және оқытудан өткізу.
- 3) Көптеген суреттерді жинақтап сыныптарға бөлу. Атап айтқанда: , 1 – ашулану, 2 – жиіркену, 3 – қорқу, 4 – қуану, 5 – мұңаю, 6 – таңырқау, 7 – бейтарап.
- 4) OpenCV технологиясын қосу және пайдалану.

Адамның бет-әлпет эмоциясын тану үшін түрлі әдістер бар, және барлығы түрліше қолданылады, менің дипломдық жұмыста қолданатын әдісім CNN әдісі. Бұл дегеніміз, деректер қорында жинақталған көптеген суреттермен адамның бет-әлпет эмоциясын тану. Яғни, адамның кел бетін деректер қорындағы суреттермен салыстырып, қандай да бір элементке (суретке) сәйкес келсе, сол элементтің атын шығарып береді.

3.2 Бағдарламаны жасау барысында пайдаланылған технологиялар

Айтып өткеніміздей нейрондық желілер көмегімен бет-әлпет эмоциясын тану үшін бірнеше бағдарламалық тілдер және технологиялар қажет. Ол бағдарламалар мыналар:

TensorFlow – адамның қабылдау сапасына қол жеткізу үшін бейнелерді автоматты түрде табу және топтастыру үшін нейрондық желіні құру және оқыту мәселелерін шешу үшін Google әзірлеген компьютерлік оқытуға арналған ашық бағдарлама кітапханасы. Ол зерттеуге және Google өнімдерін дамытуға қолданылады. Кітапханамен жұмыс істеудің негізгі API-і Python үшін орындалады, C ++, Haskell, Java, Go және Swift бағдарламалары бар. Бұл DistBelief жабық жобасының жалғасы. TensorFlow бастапқыда Google-дің Brain

командасымен Google-дің ішкі қолданысы үшін әзірленді, 2015 жылы жүйе ашық Apache 2.0 лицензиясы.

Keras – Python-да жазылған ашық нейрондық желілік кітапхана. Бұл DeepLearning4j, TensorFlow және Theano жоғары үстем құрылысы. Ол тереңдетілген оқыту желілерімен тиімді жұмыс істеуге бағытталған және ықшам, модульдік және кеңеюге арналған. Ол ONEIROS зерттеу жұмыстарының бөлігі ретінде құрылды (Open Nairo - Electronic Intelligent Robot Operating System) , оның негізгі авторы және қолдаушысы - Google инженері Франсуа Шолле (Google Francois).

Google Keras-ті негізгі TensorFlow кітапханасында қолдайтындығы жоспарланды, алайда Scholle Keras-ды бөлек қосуға айналдырды, өйткені тұжырымдамаға сай, Keras – бұл компьютерлік оқытудың түпкілікті жүйесі емес, интерфейс. Keras компьютерлік есептеуіш кітапхана ретінде пайдаланылатын ғылыми есептеу кітапханасына қарамастан, нейрондық желілерді қалыптастыруды жеңілдететін жоғары деңгейлі, интуитивті абстракциялар жиынтығын ұсынады. Microsoft корпорациясы Keras және CNTK төмен деңгейлі кітапханаларды қосу бойынша жұмыс жүргізуде

NumPy – Python бағдарламалау тілінің ашық бастапқы коды. Келтірілген тілдерде іске асырылған математикалық алгоритмдер (мысалы, Python) жиі құрастырылған тілдерде (мысалы, Fortran, C, Java) іске асырылған сол алгоритмдерге қарағанда әлдеқайда баяу жұмыс істейді. NumPy кітапханасы көп өлшемді массивтермен жұмыс істеу үшін оңтайландырылған есептік алгоритмдерді (функциялар мен операторлар түрінде) іске асыруды қамтамасыз етеді. Нәтижесінде, массивтерде (матрицалар) операциялардың реті ретінде көрінетін кез келген алгоритм және NumPy арқылы іске асырылған MATLAB-та орындалатын баламалы код жұмыс істейді.

Жасанды интеллектуалдық эмоцияны тану жүйесі

Профильді теория және негізгі эмоцияларды пайдаланып, сондай-ақ эмоциялардың өзгеруі, нақты эмоциялардың математикалық модельін сипаттауға болады. Кез келген мимикалық өрнек талданады және түсінікті зияткерлік жүйе тіліне аударылады. Математикалық модель құру бойынша негізгі және өзара байланысты кезеңдер мыналар болып табылады:

- 1) Кадрдағы бет-әлпетті анықтау;
- 2) Жасанды жүйені түсіну, адамның қатты қозғалуы немесе жарық жеткіліксіз болған жағдайда кадрдағы адам бейнесі табылмайды.
- 3) Бірегей бет сипаттамаларын ерекшелеуге мүмкіндік беретін адамның эмоцияларын ажырата білу (мысалы, көздің кескіні мен пішіні, тұлғаның ұзарту, бетіндегі ауыздың орналасуы немесе деректер қорындағы кескіндермен салыстырып сәйкесін шығару).

Адамның миы мұның бәрін автоматты түрде және бірден жасайды. Компьютерлер адам сияқты божаудың осындай жоғары деңгейіне қабілетсіз, сондықтан оларды осы процестегі әрбір қадамға жеке үйретуге тура келеді.

Интеллектуалды жүйелерді әзірлеу және жобалау кезеңдерінің санын анықтауға әртүрлі көзқарастар бар. Бұл көптеген факторларға, атап айтқанда, болашақ зияткерлік жүйе функцияларының сипатына, оларды пайдалану саласына, дамыған аспаптық құралдардың болуына және т.б. байланысты.

Эмоцияларды автоматты тану бағдарламалық жүйелерінің кейбірін қарастырайық.

Бағдарламаны талдау үшін негізгі кестеде үш өлшем таңдалды: жүйенің мүмкіндіктері, артықшылықтары мен кемшіліктері 3-кестеде көрсетілген.

«Жүйенің мүмкіндіктері» – бағаны жүйені іске асыру кезінде қолданылған әдісті қарастырады.

Кесте-3 – Эмоцияларды тануды бағдарламалық қамтамасыз ету

Эмоцияны тану жүйесі	Жүйе туралы ақпарат	Жүйе мүмкіндіктері	Артықшылықтары	Кемшіліктері
Face Reader	Әзірлеуші Компания: бетке бейне түсірілім пайдаланылады	Эмоцияларды тану, этникалық тиістілігін анықтау, жасанды бет модельін жасау.	89% дәлдікпен тану; эмоцияларды толық немесе толық анықтауға болады (бейне), толық визуализация (гистограммалар, эмоциялар диаграммалары).	жасқа дейінгі балаларды танымайды; көзілдірікте адамның эмоцияларын дәлме-дәл анықтау;
EMotion Software және GladOrSad	Әзірлеуші Компания: Visual Recognition (Нидерланды)	Жүйе көз бұрышы және ауыз бұрышы сияқты негізгі аймақты анықтаумен бет жасайды.	Ашуланған, қайғы, қорқыныш, таңқаларлық, ашуланған және бақыт. Есептеу машинасына талап етпейді.	Іске асыру алгоритмінің белгісіз мәліметтері (кемшіліктер анықталмаған)
	Әзірлеуші Компания: MMER-Systems (Германия)	Белгілі бір Деформацияланатын бетперде салу.	Алты негізгі эмоцияны таниды, жынысын, жасын, этникалық егер бұрын фотосурет базаға жүктелген болса.	Жүктелетін деректерді толық қамту, өйткені веб камерамен жұмыс істеуге болады;
Face Security	Әзірлеуші Компания: Cognite (Германия)	Жоба бірегей базалар мен қандай да бір санаттағы адамдардың деректер банкіне өңдеуге арналған.	Кластерлік конфигурация; адамдардың суреттер тізімін сұрыптау; іздеу үлгілері қолжетімді.	Нашар жарықтандырылған адамдарды танымайды; тек алдында орналасқан адамдарды тану.

Жасанды зияткерлік жүйелерді (ЖЗЖ) құру процесі шартты түрде бес кезеңге бөлінгені 14-сурет көрсетілген:



-сурет – Жобалау кезеңдері (ЖЗЖ)

Есептердің анықталуын сәйкестендіру және олардың сипаттамаларын сәйкестендіру. Жобаланатын жүйеге техникалық тапсырма әзірленеді, жүйені пайдаланушылар шеңбері шектеледі.

Сарапшылар шеңбері туралы білімді көрсететін пәндік аумақтың негізгі ұғымдарын таңдау. Білім беру инженері білімді ұсыну және шешім қабылдау процедураларын ресми түрде анықтайды. Пән саласы туралы сарапшының білімін ұсынудың сипатты сұлбасын таңдауды анықтайтын ұғымдар анықталады және тұжырымдалады.

Білім беру формализмін таңдау және шешімдерді шығару механизмін анықтау. Білімді ұсыну үшін әзірленген құрылым келесі кезеңді-жүйенің білім базасын тікелей құру үшін негіз болып табылады.

Білім беру тілін таңдау немесе жасау. Ереже тұжырымдалған және таңдалған тілде ұсынылған соң, оларды білім инженері деректер қорына енгізеді.

Нақты тексеру есептерін шешу жолымен жүйені тестілеу. Зияткерлік жүйелерді құру кезеңдері қатаң реттелмеген болып табылады. Олардың кейбіреулері арасында уақытша және мазмұнды шекара өткізу қиын. Олар зияткерлік жүйелерді жобалау процесін шамамен бір дәрежеде сипаттайды.

Нейрондық желіні қолдану арқылы эмоцияларды танудың жасанды зияткерлік жүйесін әзірлеу міндетті түрде нейрожеліні оқытуды талап етеді. Нақты тапсырмаға нейрондық желіні құру үшін, нейронаралық қосылыстар параметрлерінің мәнін таңдау керек.

Бұл процесс бет сипаттамаларының жиынтығын шығару үшін үлкен көлемді деректерді және компьютердің үлкен өнімділігін талап етеді.

Бірінші тарау бойынша қорытындылар:

Жасанды зияткерлік жүйелер түрлері, шешілетін міндеттер мен пайдалану әдістері бойынша жіктеледі. Жасанды Зияткерлік Жүйелер(ЖЗЖ) кең спектрлі

есептерді шешу үшін ең тиімді жүйені және дәл әдісті таңдауға мүмкіндік береді.

ЖЗЖ әзірлеу процесі қатаң реттелмеген болып табылады. Шартты түрде жобалауды бес кезеңге бөлуге болады:

есеп сипаттамаларын сәйкестендіру;

білімді ұсыну үшін ұғымдарды табу;

білім беру құрылымын жобалау;

білім беру ережелерін қалыптастыру;

білім беру ережелерін бағалау.

Жасанды нейрондық желі деп – математикалық модель, сондай-ақ биологиялық нейрондық желілер принципі бойынша құрылған оның бағдарламалық және аппараттық іске асырылуы. Бұл ұғым адам миында өтетін процестерді модельдеу кезінде пайда болды.

Жасанды нейрондық желіні оқыту күрделі процесс, мұнда бос параметрлер желі орнатылған ортаны модельдеу нәтижесінде жүзеге асырылып құрылады.

Нейрондық желілер бірқатар проблемаларға ие, олардың бірі нейрондық желілерді әрекетте тиімді пайдалануға мүмкіндік беретін қандай да бір жүйені құру үшін оңтайлы модельді таңдау болып табылады.

Профильдеуді пайдалану қауіпсіздіктің, бизнес, банктік және басқа да салаларда суреттерді немесе видео деректерді тану проблемаларын шешу кезінде пайдаланылады.

ҚОРЫТЫНДЫ

Адам эмоциясын тануға қабілетті зияткерлік жүйені жүзеге асыру – бұл пән саласын терең зерттеуді, өзіндік талдауды, эксперименталды мәліметтерді өңдеу және таңдау кезіндегі іскерліктер мен дағдыларды, дискретті математика, геометрия, бағдарламалау, сонымен қатар психология және ғылымның басқа да көптеген салаларында терең білімді талап ететін техникалық және математикалық күрделі міндетті шешу.

Тану жүйелерін әзірлеу кезінде нейрондық желілерді пайдалану нашар қалыптасқан деректерді құрылымдауға, деректерді өңдеу процесін жеделдетуге, нейрондық желінің ақпараттық моделі орналастырылған ортаның барабарлығын бағалауға мүмкіндік береді.

Метрикалық жүйе эмоциялардың прототиптерін жасауға мүмкіндік береді, соның арқасында әзірлеуші нейрондық желімен «өзара әрекеттесу» мүмкіндігіне ие, яғни, эмоциялар анықталған прототипі негізінде оқыту.

Менің дипломдық жұмысымның өзектілігі – адамның бет-әлпет эмоциясын шынайы уақытта жылдам әрі нақты тану. Мен көздеген мақсатыма жету үшін жетекшімен бірге осы бағдарламаны жасау барысында қандай кедергілер болатынын алдын - ала болжап, ол кедергілердің шешу жолдарын тауып, нақтылай айтқанда, дипломдық жұмысты аяқтадым.

Қазіргі технологияның дамыған уақытында кейбір адамдардан роботтар озып жұмыс жасап жатқаны бәрімізге мәлім.

Деректер жиынтығы көптеген адамдардың эмоционалды суреттерінен тұрады, нақтылай айтқанда адамның бет-әлпет кескіндерінен тұрады. Бүгінгі таңда нейрондық желілерді әрі қарай зерттеу ғылым мен техниканың көптеген салаларында, сондай-ақ адам қызметінде табысты қолданылатын ғылыми бағыт болып табылады. Заманауи тану жүйелерін дамытуда басты назар қазіргі кезде медицинада, прототиптерде және адам қызметінің басқа да салаларында 3D кескіндерді семантикалық сегменттеу саласына ауысады - бұл өте күрделі алгоритмдер.

Барлық қойылған міндеттер орындалды, яғни басты мақсатқа қол жеткізілді деп айтуға болады.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1 Маркова С.В., Жиглалов К.Ю. применение нейронной сети для создания системы распознавания изображений // Фундаментальные исследования. – 2017. – № 8-1. – 60-64.б

2 Айвазян С.А., Бухстабер В.М., Энюков И.С., Мешалкин Л.Д. Қолданбалы статистика: Өлшемді жіктеу және азайту. - М.: Қаржы және статистика, 1989.

3 Urban Sound Classification // Сайттың электронды нұсқасы <https://www.kaggle.com/pavansanagapati/urban-sound-classification/kernels>

4 Фундаментальные исследования // Сайттың электрондық нұсқасы <http://www.fundamental-research.ru/ru/article/view?id=41621>

5 Андрей Созыкин Сохранение Нейросети в процессе обучения // Сайттың электронды нұсқасы <http://www.itproger.com>. 2015. – № 4-1.– 3.б

6 Маргарита Акулич Распознавание лиц в маркетинге и ритейле // Сайттың электрондық нұсқасы <https://www.litres.ru/margarita-akulich/raspoznavanie-lic-v-marketinge-i-riteyle/chitat-onlayn/>

7 Д.В.Брилюк, В.В.Старовойтов Распознавание человека по изображению лица нейросетевыми методами // Сайттың электрондық нұсқасы <https://www.researchgate.net/publication/>

А қосымшасы

(міндетті)

Техникалық тапсырма

А.1.5 Адам эмоциясын тануға арналған жүйені жасақтауға техникалық тапсырма

Техникалық тапсырма адам эмоциясын тануға арналған жүйені жасақтауға бағытталған. Жүйені кез келген адам пайдалана алады және психофизиология мамандары үшін қолдануға арналған.

А.1.5.1 Мақсаты

Жүйе адам бейнесі негізінде жасалған суреттердегі адам эмоциясын анықтап, болжау жасау мақсатына арналған.

А.1.5.2 Жүйенің функционалды талаптары

Жүйе келесі функцияларды орындай алу керек:

- жүйенің инициализациясы;
- кіріске берілген суреттің дұрыс өңделуі және берілген сурет негізінде жасалған қорытынды ақпарат;
- кіріске берілген бейне негізінде қажетті ақпаратты көрсету;

Бастапқы деректер:

- адамның эмоциялы суреттері;
- модельге арналған оқыту ақпараттары;

Нәтижесі:

- класс атауын экранға шығару;

А.1.5.3 Техникалық талаптар

Жүйе IBM үйлесімді дербес компьютерлерде жұмыс істеуі керек. Ең аз конфигурация: процессор түрі - Pentium және одан жоғары; Жад сыйымдылығы - 64 МБ және одан жоғары.

А қосымшасының жалғасы

А.1.5.4 Бағдарламалық үйлесімділік талаптары

Жүйе Win32 API-ді іске асыратын Windows операциялық жүйелерінің бақылауында жұмыс істеуі керек.

Б қосымшасы

(міндетті)

Бағдарлама коды

```
from keras.callbacks import CSVLogger, ModelCheckpoint, EarlyStopping
from keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from keras.layers import Activation, Convolution2D, Dropout, Conv2D
from keras.layers import AveragePooling2D, BatchNormalization
from keras.layers import GlobalAveragePooling2D
from keras.models import Sequential
from keras.layers import Flatten
from keras.models import Model
from keras.layers import Input
from keras.layers import MaxPooling2D
from keras.layers import SeparableConv2D
from keras import layers
from keras.regularizers import l2
import pandas as pd
import cv2
import numpy as np

dataset_path = 'test/*.jpg'
image_size=(48,48)
batch_size = 32
num_epochs = 110
input_shape = (48, 48, 1)
validation_split = .2
verbose = 1
num_classes = 7
patience = 50
base_path = 'models/'
l2_regularization=0.01
data = pd.read(dataset_path)
pixels = data['pixels'].tolist()
width, height = 48, 48
faces = []
for pixel_sequence in pixels:
    face = [int(pixel) for pixel in pixel_sequence.split(' ')]
    face = np.asarray(face).reshape(width, height)
    face = cv2.resize(face.astype('uint8'),image_size)
    faces.append(face.astype('float32'))
```

Б қосымшасының жалғасы

```
faces = np.asarray(faces)
faces = np.expand_dims(faces)
emotions = pd.get_dummies(data['emotion']).as_matrix()
return faces, emotions
def preprocess_input(x, v2=True):
    x = x.astype('float32')
    x = x / 255
    if v2:
        x = x - 0.5
        x = x * 2
    return x
data_generator = ImageDataGenerator(
    featurewise_center=False,
    featurewise_std_normalization=False,
    rotation_range=10,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=.1,
    horizontal_flip=True)

model = mini_XCEPTION(input_shape, num_classes)
regularization = l2(l2_regularization)

img_input = Input(input_shape)
x = Conv2D(8, (3, 3), strides=(1, 1), kernel_regularizer=regularization,
use_bias=False)(img_input)
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = Conv2D(8, (3, 3), strides=(1, 1), kernel_regularizer=regularization,
use_bias=False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
residual = Conv2D(16, (1, 1), strides = (2, 2), padding = 'same', use_bias =
False)(x)
residual = BatchNormalization()(residual)
x = SeparableConv2D(16, (3, 3), padding='same', kernel_regularizer =
regularization, use_bias = False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = SeparableConv2D(16, (3, 3), padding='same', kernel_regularizer =
regularization, use_bias = False)(x)
x = BatchNormalization()(x)
```

Б қосымшасының жалғасы

```
x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
x = layers.add([x, residual])
residual = Conv2D(32, (1, 1), strides=(2, 2), padding='same',
use_bias=False)(x)
residual = BatchNormalization()(residual)
x = SeparableConv2D(32, (3, 3), padding='same', kernel_regularizer =
regularization, use_bias = False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = SeparableConv2D(32, (3, 3), padding='same', kernel_regularizer =
regularization, use_bias = False)(x)
x = BatchNormalization()(x)
x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
x = layers.add([x, residual])
residual = Conv2D(64, (1, 1), strides=(2, 2),padding='same',
use_bias=False)(x)
residual = BatchNormalization()(residual)
x = SeparableConv2D(64, (3, 3), padding = 'same', kernel_regularizer =
regularization,use_bias = False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = SeparableConv2D(64, (3, 3), padding='same', kernel_regularizer =
regularization,use_bias = False)(x)
x = BatchNormalization()(x)
x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
x = layers.add([x, residual])
residual = Conv2D(128, (1, 1), strides=(2, 2), padding='same', use_bias =
False)(x)
residual = BatchNormalization()(residual)
x = SeparableConv2D(128, (3, 3), padding='same', kernel_regularizer =
regularization, use_bias = False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = SeparableConv2D(128, (3, 3), padding='same', kernel_regularizer =
regularization, use_bias = False)(x)
x = BatchNormalization()(x)
x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
x = layers.add([x, residual])
x = Conv2D(num_classes, (3, 3), padding='same')(x)
x = GlobalAveragePooling2D()(x)
output = Activation('softmax',name='predictions')(x)
model = Model(img_input, output)
```

Б қосымшасының жалғасы

```
model.summary()
log_file_path = base_path + '_emotion_training.log'
csv_logger = CSVLogger(log_file_path, append=False)
early_stop = EarlyStopping('val_loss', patience=patience)
reduce_lr = ReduceLROnPlateau('val_loss', factor = 0.1, patience =
int(patience/4), verbose = 1)
trained_models_path = base_path + '_mini_XCEPTION'
model_names = trained_models_path + '.{epoch:02d}-{val_acc:.2f}.hdf5'
model_checkpoint = ModelCheckpoint(model_names, 'val_loss', verbose = 1,
save_best_only = True)
callbacks = [model_checkpoint, csv_logger, early_stop, reduce_lr]
faces, emotions = 'test'
faces = preprocess_input(faces)
num_samples, num_classes = emotions.shape
xtrain, xtest, ytrain, ytest = train_test_split(faces, emotions, test_size=0.2,
shuffle = True)
model.fit_generator(data_generator.flow(xtrain, ytrain, batch_size),
                    steps_per_epoch=len(xtrain) / batch_size, epochs=num_epochs,
verbose=1, callbacks=callbacks, validation_data=(xtest,ytest))
from keras.preprocessing.image import img_to_array
import imutils
import cv2
from keras.models import load_model

import numpy as np
detection_model_path =
'haarcascade_files/haarcascade_frontalface_default.xml'
emotion_model_path = 'models/ABS_XCEPTION.hdf5'
face_detection = cv2.CascadeClassifier(detection_model_path)
emotion_classifier = load_model(emotion_model_path, compile=False)
EMOTIONS = ["angry" ,"disgust","scared", "happy", "sad", "surprised",
"neutral"]
cv2.namedWindow('Emotion!')
camera = cv2.VideoCapture(0)
while True:
    frame = camera.read()[1]
    frame = imutils.resize(frame,width=400)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces =
face_detection.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,minSize=30,
30),flags=cv2.CASCADE_SCALE_IMAGE)
    canvas = np.zeros((250, 300, 3), dtype="uint8")
```

Б қосымшасының жалғасы

```
frameClone = frame.copy()
if len(faces) > 0:
    faces = sorted(faces, reverse=True,
    key=lambda x: (x[2] - x[0]) * (x[3] - x[1]))[0]
    (fX, fY, fW, fH) = faces
    roi = gray[fY:fY + fH, fX:fX + fW]
    roi = cv2.resize(roi, (48, 48))
    roi = roi.astype("float") / 255.0
    roi = img_to_array(roi)
    roi = np.expand_dims(roi, axis=0)

preds = emotion_classifier.predict(roi)[0]
emotion_probability = np.max(preds)
label = EMOTIONS[preds.argmax()]
for (i, (emotion, prob)) in enumerate(zip(EMOTIONS, preds)):

    #Тексттік белгі құру
    text = "{:}: {:.2f}%".format ( emotion, prob * 100)
    w = int ( prob * 300)
    cv2.rectangle(canvas, (7, (i * 35) + 5),
    (w, (i * 35) + 35), (0, 0, 255), -1)
    cv2.putText(canvas, text, (10, (i * 35) + 23),
    cv2.FONT_HERSHEY_SIMPLEX, 0.45,
    (255, 255, 255), 2)
    cv2.putText(frameClone, label, (fX, fY - 10),
    cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)
    cv2.rectangle(frameClone, (fX, fY), (fX + fW, fY + fH), (0, 0, 255), 2)
    cv2.imshow('Emotion', frameClone)
    cv2.imshow("Text Emotion", canvas)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
camera.release()
cv2.destroyAllWindows()
```