Қ. Жұбанов атындағы Ақтөбе өңірлік университеті

Физика және математика факультеті

Информатика және АТ кафедрасы

6B06103-Есептеу техникасы және бағдарламалық қамтамасыз ету мамандығы



КУРСТЫҚ ЖҰМЫС

ТАҚЫРЫБЫ: ЖАСАНДЫ НЕЙРОНДЫҚ ЖЕЛІ АЛГОРИТМІ

Орындаған: Жамбулов С. Ж.

Тексерген : Ермағанбетов Т. К.

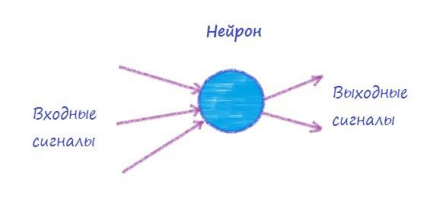
Ақтөбе 2023

|  |  |
| --- | --- |
| МАЗМҰНЫ |  |
| Кіріспе …………………………………………………………………........ | 3 |
| 1. Кіріс сигналдарын біріктіру …………………………............................. | 3 |
| 1.1 Элементті белсендіру функциясы ……………….................................. | 4 |
| 2. НЕЙРОНДЫҚ ЖЕЛІЛЕРДІҢ АРХИТЕКТУРАСЫ …………............... | 4 |
| 3. НЕЙРОНДЫҚ ЖЕЛІНІ ОҚЫТУ ………………………………............. | 6 |
| 4. T-Rex ойынын талдау ............................................................................... | 7 |
| 5. Мысал ......................................................................................................... | 8 |
| Қорытынды……………………………....…………………………............. | 11 |
| Қолданылған әдебиеттер.............................……………………………….. | 11 |
|  |  |

**Кіріспе**

Жасанды нейрондық желі машиналық оқыту алгоритмінің ішкі жиыны болып табылады. Ол биологиялық нейрондық желілердің құрылымы мен функцияларына негізделген. Бұл желілер бір-біріне сигнал жіберетін көптеген нейрондардан тұрады. Яғни, жасанды миды жасау үшін нейрондарды имитациялап, оларды нейрондық желі құрайтындай етіп қосу керек. Нейрондық желі тек математикалық модель емес. Биологиялық нейрондық желі жүйке жасушаларынан тұратыны сияқты, жасанды нейрондық желі нейрондар немесе процессорлар деп аталатын көптеген элементтерден тұрады. Адам миының жұмысын көшіре отырып, ол қатаң алгоритм мен формулалар бойынша әрекет етіп қана қоймайды, сонымен қатар бұрынғы тәжірибені жинақтайды және пайдаланады. Анау. нейрондар үйренуге қабілетті.

Стандартты жасанды нейрондық желі кіріс деректер деңгейінен, бір немесе бірнеше жасырын қабаттардан және шығыс деректер деңгейінен тұрады. Әрбір қабатта бірнеше нейрондар болады. Кіріс және шығыс нейрондары сыртқы ортамен тікелей байланысты. Олардың арасында жасырын нейрондар байланысады. Нейрондық желі - нейрондардың жиынтығы бір-бірімен белгілі бір жолмен байланысты. Бір нейронды қарастырайық (Сурет-1):



Сурет-1(Бір нейронды желі)

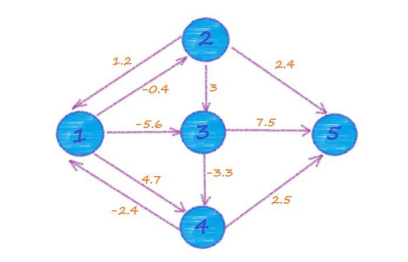
Нейрон – кіріс сигналдарының жиынынан шығыс сигналын (белгілі бір ереже бойынша) есептейтін элемент. Яғни, бір нейронның негізгі әрекеттер тізбегі келесідей:

* Алдыңғы желі элементтерінен сигналдарды қабылдау;
* Кіріс сигналдарын біріктіру;
* Шығыс сигналын есептеу;
* Шығу сигналының келесі нейрондық элементтерге берілуі желілер;

1. **Кіріс сигналдарын біріктіру**

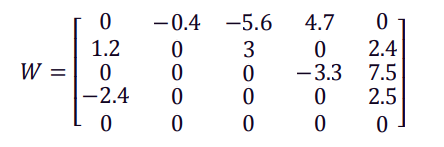
Әрбір нейрон бірнеше кіріс сигналдарын қабылдай алатындықтан, нейрондық желіні модельдеу кезінде осы сигналдардың барлығын біріктірудің белгілі бір ережесін орнату қажет. Ал қосылыстардың өлшенген мәндерін жинақтау ережесі жиі қолданылады.

Көбінесе нейрондар арасындағы байланыс құрылымы W матрицасы түрінде бейнеленеді, ол салмақ матрицасы деп аталады. 𝑤𝑖𝑗 матрицалық элементі формуладағыдай i элементінен j элементіне өтетін қосылымның салмағын анықтайды. Салмақ матрицаларының қалай құрастырылғанын түсіну үшін қарапайым нейрондық желіні қарастырайық:



Сурет-2(Салмағы бар матрица)

Мұндай нейрондық желінің салмақ матрицасы келесі пішінге ие болады:



**1.1. Элементті белсендіру функциясы.**

Шығу сигналдарын қарастырайық. Әрбір желі элементі үшін белгілі бір ереже бар, оған сәйкес оның шығыс мәні элементтің біріктірілген кірісінің мәнінен есептеледі. Бұл ереже белсендіру функциясы деп аталады. Ал шығыс шамасының өзі нейронның белсенділігі деп аталады. Кез келген математикалық функциялар белсендіру функциясы ретінде әрекет ете алады, мұнда мысал ретінде ең жиі қолданылатын кейбіреулер берілген:

* шекті функция - егер біріктірілген кірістің мәні белгілі бір мәннен төмен болса, онда белсенділік нөлге тең, егер одан жоғары болса бір.
* логистикалық функция.

Стандартты жасанды нейрондық желі кіріс қабатынан, бір немесе бірнеше жасырын қабаттардан және шығыс қабатынан тұрады. Әр қабатта бірнеше Нейрон бар. Кіріс және шығыс нейрондары тікелей сыртқы ортаға қосылады. Жасырын нейрондар олардың арасында байланысады.

1. **НЕЙРОНДЫҚ ЖЕЛІЛЕРДІҢ АРХИТЕКТУРАСЫ**

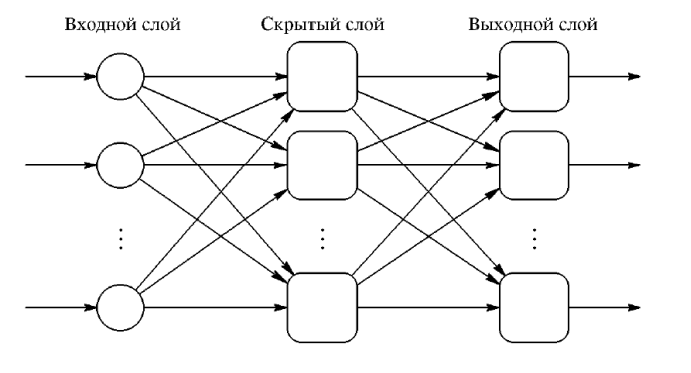
Толық байланысқан нейрондық желілерде әрбір нейрон өзінікін береді қалған нейрондарға, соның ішінде өзіне шығу сигналы. Барлығы кіріс сигналдары барлық нейрондарға беріледі. Желінің шығыс сигналдары болу барлық немесе кейбір нейрондық шығыс сигналдары бірнеше жолақтан кейін желінің жұмыс істеуі.

Негізгі нейрондық желі өзара байланысты жасанды нейрондардың үш қабатын қамтиды:

* Енгізу қабаты. Жасанды нейрондық желіге сыртқы әлемнен ақпарат енгізу деңгейінен түседі. Енгізу түйіндері деректерді өңдейді, талдайды немесе жіктейді және келесі деңгейге береді.
* Жасырын қабат. Жасырын қабаттар кіріс деңгейінен немесе басқа жасырын қабаттардан кіріс алады. Жасанды нейрондық желілерде жасырын қабаттар көп болуы мүмкін. Әрбір жасырын қабат алдыңғы қабаттың шығысын талдайды, оны өңдейді және келесі қабатқа береді.
* Шығару қабаты. Шығару деңгейі жасанды нейрондық желі арқылы барлық деректерді өңдеудің соңғы нәтижесін береді. Оның бір немесе бірнеше түйіндері болуы мүмкін. Мысалы, екілік жіктеу есебін шешу кезінде (иә/жоқ) шығыс қабатында «1» немесе «0» нәтижесін беретін бір шығыс түйіні болады. Дегенмен, бірнеше жіктеу жағдайында шығыс қабаты бірнеше шығыс түйінінен тұруы мүмкін.

Көп қабатты (қабатты) нейрондық желілерде нейрондар біріктіріледі қабаттар. Қабатта бірыңғай кіріс сигналдары бар нейрондар жиынтығы бар.

Қабаттағы нейрондардың Саны кез келген болуы мүмкін және басқа қабаттардағы нейрондардың санына тәуелді емес. Жалпы жағдайда желі солдан оңға қарай нөмірленген қабаттардан тұрады. Сыртқы кіріс сигналдары кіріс қабатының нейрондық кірістеріне беріледі (ол көбінесе нөлдік нөмірленеді), ал желінің шығысы соңғы қабаттың шығысы болып табылады. Кіріс және шығыс қабаттарынан басқа, көп қабатты нейрондық желіде бір немесе бірнеше жасырын қабаттар бар. Кейбір q қабатының нейрондық шығыстарынан келесі қабаттың нейрондық кірістеріне (q+1) байланыстар сериялық деп аталады (Сурет-3).



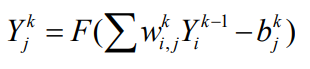
Сурет-3(Жасанды нейрондардың қабаттары)

**3. НЕЙРОНДЫҚ ЖЕЛІНІ ОҚЫТУ**

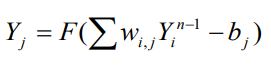
Нейрондық желіні оқыту-бұл нейрондық желінің параметрлері желілер осы желінің ортасын модельдеу арқылы конфигурацияланады кіріктірілген. Оқыту түрі параметрлерді реттеу әдісімен анықталады. Мұғаліммен және мұғалімсіз оқыту алгоритмдерін ажыратыңыз.

Көп қабатты нейронды оқыту алгоритмі:

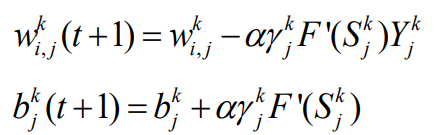
1. к қабатының j-ші нейронының шығыс мәні формула бойынша есептеледі:



1. Шығу қабатының j-ші нейронының шығыс мәні формула бойынша есептеледі:



1. Желі қатесінің функционалдығы , мұнда шығу қабатының j-ші нейрондық қатесі.
2. Нейрондардың салмақ коэффициенттері мен орын ауыстырулары формулалар бойынша есептеледі:



1. **T-Rex ойынын талдау**

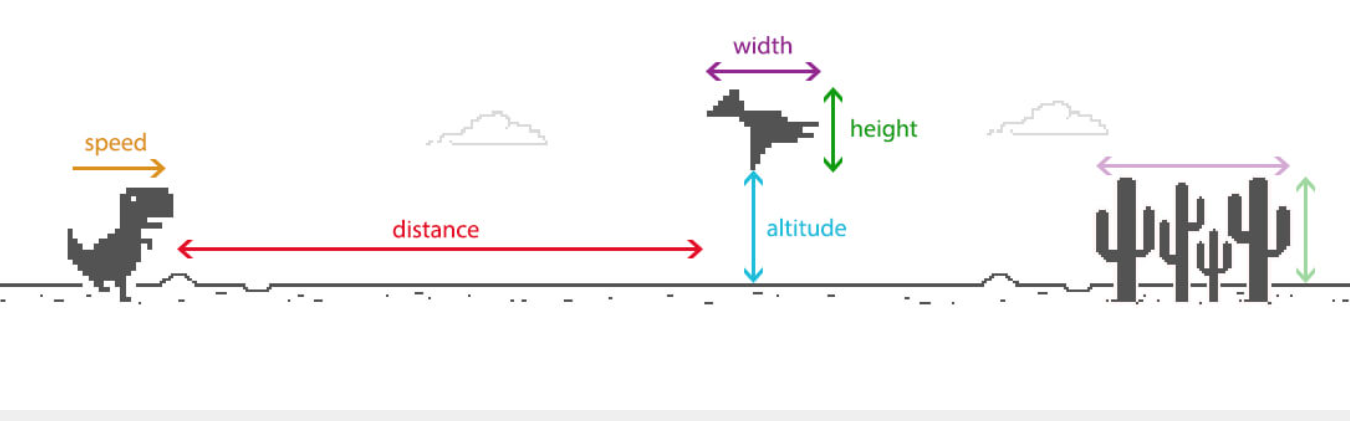
Ойлап қарасақ, ойын туралы шешімдеріміз өте қарапайым: не секіреміз, не секірмейміз (күтеміз).Енді бұл шешімдер неге байланысты?

Бұл келесі кедергіге байланысты. Сіз кедергі болған кезде секіруіңіз керек, бірақ кедергі аяқталғаннан кейін міндетті түрде кері құлап кету үшін секіру алдында күту керек. Және олардың бірнеше түрі бар: кейбіреулері биік, кейбіреулері ұзын, ал кейбіреулері құстар (сондықтан биіктікте). Нақты сөзбен айтқанда, біздің шешімдеріміз мыналарға байланысты болады деп қорытынды жасауға болады:

* Келесі кедергінің ені
* Келесі кедергінің ұзындығы
* Келесі кедергінің биіктігі
* T-Rex пен келесі кедергі арасындағы қашықтық

Ескеретін тағы бір параметр бар: ойында алға жылжып келе жатқанда, біздің T-Rex жылдам және жылдам қозғалады. Жылдам қозғалғанда, ертерек секіру керек. Сондықтан шешімдеріміз жылдамдығымызға да байланысты болады.

Бізге керегі осы ғана. Схема арқылы қорытындылау үшін (Сурет-4):



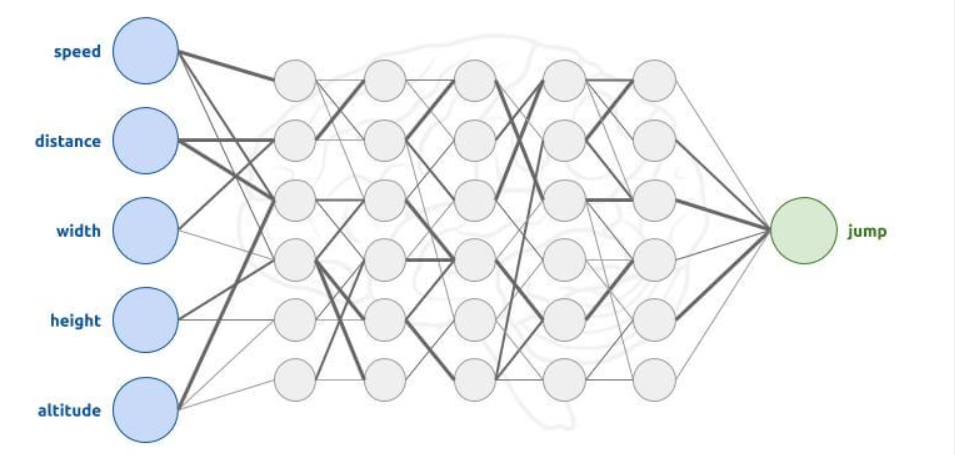
Сурет-4 (T-Rex ойынының параметрлері)

Кез келген сәтте біз оған 5 параметрімізді береміз және ол бізге секіру керек пе, жоқ па дейді. Және біз бірнеше миллисекундтан кейін қайтадан бастаймыз, және т.б., ойынның соңына дейін.

Кішкене бақылау, мен ұсынған нәрседе бізде екі мүмкін нәтиже бар: біз секіреміз немесе секірмейміз. Бірақ екеуін бір уақытта (немесе олардың кез келгенін мүлде) орындау мүмкін емес. Сондықтан біз бір ғана нәтижеге жеңілдетеміз: егер біз секірмесек 0 болады немесе секірсек 1 болады.

Қарапайым тілмен айтсақ, бұл сіздің миыңыз сияқты. Біздің миымызда көптеген мүмкін кірістер (көру, естеліктер, сезімдер,...), көптеген есептеу нейрондары бар, олар шығару әрекеттерін (сөйлеу, қозғалу,...) жасауы мүмкін. Бұл алгоритмнің мақсаты ұқсас мінез-құлыққа ие болуға тырысу: кірістерге (T уақытындағы T-Rex параметрлері), есептеулердің аралық қадамдарына, содан кейін шығыстарға ие болу (біздің жағдайда секіру керек пе, жоқ па білу үшін) .

Орталық бөлік желілердің бейберекеттігі. Сізге миды бірнеше тілімдерге кесу жеткілікті. Оның әрбір тілімдері («жасырын қабат» деп те аталады) іргелес бөліктермен (оның ішінде кіріс параметрлерімен және шығыс шешімдерімен) байланысты. Кішкене қиындықты қосу үшін, бұл сілтемелердің бәрі бірдей емес, әрқайсысы азды-көпті маңызды (ол неғұрлым маңызды болса, шешім қабылдау жолында соғұрлым ескеріледі):



Сурет-5 (Жасанды нейрондық желінің жасырын қабаты)

Мұнда, қабаттардың саны, түйіндердің саны, салмақтарды анықтау үшін біз оларды өзіміз жасай аламыз (тіпті осы мысал үшін де солай жасадым). Бірақ біз күрделірек қосымшалармен келген кезде (T-Rex сияқты 5 параметр, есептеу жылдамдығы бар,...), біз енді желіні өзіміз анықтай алмаймыз. Біздің T-Rex корпусында тек 5 параметр және бір ғана шығыс бар, кейбір қолданбалар үшін жүздеген кіріс және сонша шығыс болуы мүмкін. Бізде неғұрлым көп кіріс болса, соғұрлым жасырын қабаттар мен түйіндер қажет болады.Жасырын қабаттар неғұрлым көп болса, AI шешімдерді дәлірек қабылдай алады, бірақ есептеуге соғұрлым ұзақ уақыт кетеді.

1. **Мысал**

Python кітапханасы мен TensorFlow шеңберін пайдаланып жасанды нейрондық желіні оқытуға арналған алгоритмнің қарапайым мысалын келтірейік. Бұл мысал екілік жіктеу мәселесін шешу үшін қарапайым нейрондық желіні құруды қамтиды.

# Қажетті кітапханаларды импорттау

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.optimizers import SGD

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.datasets import make\_classification

# Мысал үшін синтетикалық деректерді жасаңыз

X, y = make\_classification(n\_samples=1000, n\_features=20, n\_classes=2, random\_state=42)

# Деректерді жаттығу және тест жиынтықтарына бөлу

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Қарапайым нейрондық желіні құру

model = Sequential()

model.add(Dense(16, input\_dim=20, activation='relu'))

model.add(Dense(1, activation='sigmoid'))

# Модельді құрастыру

model.compile(loss='binary\_crossentropy', optimizer=SGD(lr=0.01), metrics=['accuracy'])

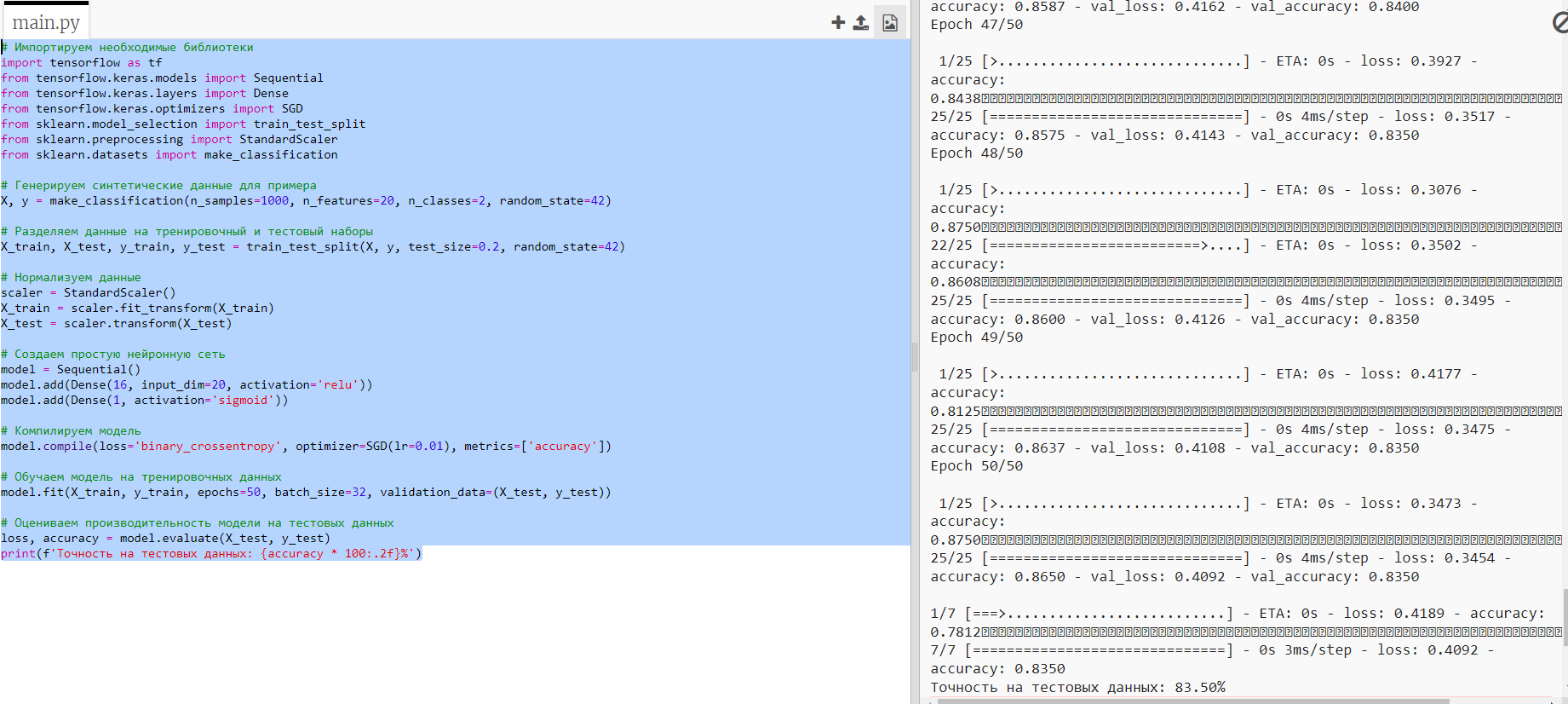
# Модельді жаттығу деректеріне үйрету

model.fit(X\_train, y\_train, epochs=50, batch\_size=32, validation\_data=(X\_test, y\_test))

# Сынақ деректері бойынша модельдің өнімділігін бағалау

loss, accuracy = model.evaluate(X\_test, y\_test)

print(f'Точность на тестовых данных: {accuracy \* 100:.2f}%')



Сурет-6 (Мысалдың компиляциядағы жауабы)

Мысалы сізде пациенттер туралы деректер бар деп елестетіп көріңіз және бірнеше медициналық параметрлерге сүйене отырып, науқастың қант диабетімен ауыратынын немесе ауырмайтынын болжайтын модель жасағыңыз келеді. Сіздің деректеріңізде қандағы қант, қысым, дене салмағының индексі және т.б. сияқты ақпарат болуы мүмкін.

* Деректерді дайындау:

Пациенттер туралы ақпарат пен сынып белгілерін қамтитын деректер жинағын жүктеп алыңыз немесе жасаңыз. Деректерді жаттығу және сынақ жиынтықтарына бөліңіз.

* Модель құру:

TensorFlow және Keras көмегімен жасанды нейрондық желі жасаңыз.

Желі архитектурасын, соның ішінде қабаттар санын, әр қабаттағы нейрондар санын және белсендіру функцияларын анықтаңыз. Сәйкес шығын функциясы мен оңтайландырғышты таңдау арқылы модельді құрастырыңыз.

* Модельді оқыту:

Модельді жаттығу деректеріне үйретіңіз. Әр дәуірдегі шығындардың дәлдігі мен функциясын бағалау үшін оқу процесін қадағалаңыз.

* Модельді бағалау:

Сынақ деректерінде модельдің өнімділігін бағалаңыз. Дәлдік, толықтық, дәлдік және F1 өлшемі сияқты көрсеткіштерді есептеңіз.

* Модельді тестілеу:

Пациенттердің жаңа деректері үшін сыныптарды болжау үшін оқытылған үлгіні пайдаланыңыз. Бұл тапсырма медициналық саладағы практикалық жіктеу мәселелерін шешу үшін жасанды нейрондық желілерді қалай құру және оқыту керектігін түсінуге көмектеседі.

**Қорытынды**

Дәстүрлі машиналық оқыту әдістері бағдарламалық жасақтаманың дұрыс жұмыс істеуі үшін адамның өзара әрекеттесуін талап етеді. Деректер зерттеушісі бағдарламалық жасақтама талдауы қажет тиісті функциялар жинағын қолмен анықтайды. Бұл шектеу бағдарламалық жасақтаманы жасау мен басқаруды жалықтыратын және уақытты қажет ететін процесс етеді.

Екінші жағынан, терең оқытуда деректер зерттеушісі бағдарламалық жасақтамаға тек бастапқы деректерді береді. Терең оқыту желісі мүмкіндіктерді өздігінен шығарады және өз бетінше көбірек үйренеді. Ол құрылымдалмаған деректер жиынын (мәтіндік құжаттар сияқты) талдай алады, деректер атрибуттарына басымдық бере алады және күрделірек мәселелерді шеше алады.

**Қолданылған әдебиеттер**

1. Барский, А.Б. Логические нейронные сети: Учебное пособие / А.Б. Барский. - М.: Бином, 2013.
2. ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ И ИХ ПРИЛОЖЕНИЯ, Ф.М. ГАФАРОВ, А.Ф. ГАЛИМЯНОВ, – 2018
3. AЛГОРИТМ ОБУЧЕНИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ НА ОСНОВЕ БОЛЬШИХ ВЫБОРОК С ПОВТОРЯЮЩИМИСЯ ПРИМЕРАМИ, П. Л. Папаев, С. П. Дударов
4. <https://aws.amazon.com/ru/what-is/neural-network/>
5. <https://www.dewep.net/realisations/en/t-rex-machine-learning-neural-network-genetic-algorithm>
6. <https://www.youtube.com/watch?v=cWVyHuYYIzA>