

УАҚЫТ ҚАТАРЛАРЫН ТАЛДАУҒА АРНАЛҒАН ТЕРЕҢ ОҚЫТУДЫҢ ЗАМАНАУИ ӘДІСТЕРІ

Жақсылық Әділет Қасымжанұлы

6B06120 - «Бағдарламалық инженерия» БББ 3 курс студенті

М.Х.Дулати атындағы Тараз университеті, Тараз қ., Қазақстан Республикасы

Өзгелді Нұрболат Ғаниұлы

Т.Рысқұлов атындағы №48 орта мектебінің 9 сынып оқушысы

Тараз қ., Қазақстан Республикасы

Ғылыми жетекші Адилова А.К.

ak.adilova@dulaty.kz

Уақыт қатарын талдау - бұл уақыт қатарының деректерімен жұмыс істейтін немесе трендтерді талдайтын статистикалық әдіс. Ол маңызды статистиканы және басқа деректер сипаттамаларын алу үшін уақыт сериясының деректерін талдау әдістерін қамтиды. Бұл серияда біз уақытқа тәуелді деректерді болжау және түсіну үшін қолданылатын негізгі құралдарды зерттей отырып, классикалық уақыт қатарының модельдерін қарастырамыз.

Технологияның дамуымен және деректер көлемінің ұлғаюымен уақыт серияларын талдау әдістері де айтарлықтай өзгерістерге ұшырады. ARIMA және экспоненциалды тегістеу сияқты дәстүрлі статистикалық тәсілдер көптеген жылдар бойы талдаудың негізі болды және стационарлық және маусымдық уақыт қатарларымен жұмыс істеуде тиімді болды.

Машиналық оқыту және терең оқыту әдістерінің пайда болуымен уақыт серияларын талдаудың жаңа мүмкіндіктері пайда болды. Машиналық оқыту кешігу белгілері және маусымдық көрсеткіштер сияқты қосымша белгілерді қолдануға мүмкіндік береді, бұл болжамдардың дәлдігін көбейтеді. Сонымен қатар, LSTM, GRU, TCN және Transformer сияқты заманауи терең оқыту архитектуралары күрделі заңдылықтар мен ұзақ мерзімді тәуелділіктерді анықтай алады.

Уақыт қатарының деректері белгілі бір уақыт аралығында қайталанатын өлшеулер нәтижесінде жиналған бақылаулардан тұрады. Бұл бақылаулар график түрінде ұсынылуы мүмкін, мұнда уақыт әдетте осьтердің бірі болып табылады. Уақыт қатарлары ұйымдарға уақыт өте келе тенденциялар мен қайталанатын заңдылықтарды анықтайтын факторлар туралы түсінік алуға көмектеседі. Деректерді визуализациялауды зерттей отырып, біз маусымдық тенденцияларды анықтай аламыз және осы заңдылықтардың себептерін анықтай аламыз. Заманауи аналитикалық платформалар интерактивті және егжей-тегжейлі визуалды көріністерді ұсына отырып, дәстүрлі сызықтық графиктерге қарағанда көбірек ұсынады.

Қайталанатын нейрондық желілер (Recurrent neural network, RNN) - элементтер арасындағы байланыстар бағытталған тізбекті құрайтын нейрондық желілердің бір түрі. Осының арқасында уақыт бойынша оқиғалар сериясын немесе дәйекті кеңістіктік тізбектерді өңдеу мүмкіндігі пайда болады. Көп қабатты перцептрондардан айырмашылығы, қайталанатын желілер өздерінің ішкі жадын еркін ұзындықтағы тізбектерді өңдеу үшін пайдалана алады. Сондықтан RNN желілері біртұтас нәрсе бөлінген тапсырмаларда қолданылады, мысалы: қолжазбаны, сөйлеуді тану үшін. Қарапайымнан күрделіге дейінгі қайталанатын желілер үшін көптеген әртүрлі архитектуралық шешімдер ұсынылды. Соңғы уақытта ұзақ мерзімді және қысқа мерзімді жады бар желілер (long short-term memory) және басқарылатын қайталанатын блок (Gated Recurrent Unit) ең көп таралды.

RNN нейрондардан тұрады, яғни бірлескен жұмыс барысында күрделі тапсырмаларды орындайтын деректерді өңдеу түйіндері. Нейрондар кіріс, шығыс және бірнеше жасырын қабаттарға бөлінеді. Кіріс қабаты өңдеу үшін ақпарат алады, ал шығыс қабаты нәтижені қайтарады.

Жойылаттын градиент - бұл модельдің оқу градиенті нөлге жақындайтын жағдайы. Градиенттің жоғалуы RNN ұсынылған деректер бойынша тиімді оқу мүмкіндігін жоғалтқанын білдіреді, бұл оқудың болмауына әкеледі. Оқытылмаған модель нақты өмірде жақсы жұмыс істей алмайды, өйткені оның салмағы дұрыс емес. RNN ұзақ деректер тізбегін өңдеу кезінде жоғалып кету немесе жарылыс градиентінің жоғары қаупімен сипатталады.

LSTM (long short-term memory) және GRU (Gated Recurrent Unit)-дағы жад механизмдері, оларды уақыт қатарларын болжауда қолдану.

Ұзақ, қысқа мерзімді жады бар желілер қайталанатын нейрондық желілердің бір түрі болып табылады, алдыңғы кірулер туралы ақпаратты ұзақ уақыт сақтауға қабілетті. Олар байланысты ақпаратты есте сақтауға және ұмытуға мүмкіндік беретін жад блоктарын пайдаланатын ағымдағы кірістерден тұрады. Олар ақпаратты ұзақ ретпен сақтауға және жаңартуға мүмкіндік беретін ерекше құрылымға ие.

LSTM мынандай жұмыстарды атқарады:

- Ұяшық күйі: ұяшық күйі бүкіл тізбек тізбегі бойынша ақпаратты жіберетін конвейер таспасы ретінде әрекет етеді. Ол ақпараттың сақталуына мүмкіндік бере отырып, өзінің күйін сақтай алады.

- Қақпақтар: LSTM ақпарат ағынын басқару үшін үш қақпақты пайдаланады;

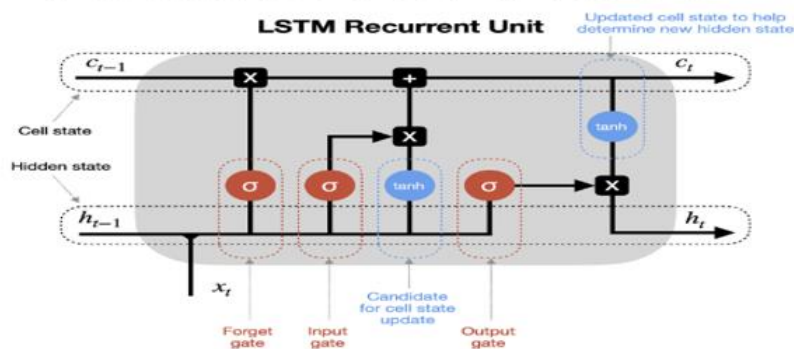
- Ұмытылған қақпа: ұяшық күйінен қандай ақпаратты алып тастау керектігін анықтайды;

- Кіріс клапаны: ұяшық күйінде қандай жаңа ақпаратты сақтау керектігін анықтайды;

- Шығу клапаны: ұяшық күйіне және кіріс деректеріне байланысты ақпарат шығысын басқарады.

Бұл элементтер LSTM-ге ақпаратты іріктеп есте сақтауға және ұмытуға көмектеседі, бұл оларды дәйекті деректердегі ұзақ мерзімді тәуелділіктерді өңдеуде тиімді етеді (1-сурет).

LONG SHORT-TERM MEMORY NEURAL NETWORKS



1-сурет. LSTM схемасы.

GRU-бұл LSTM-дің жеңілдетілген нұсқасы. Олар ұмыту және енгізу қақпаларын бірін жаңарту қақпасына біріктіреді және ұяшық күйі мен жасырын күйді біріктіреді. GRU жұмысына қысқаша сипаттама:

- Қалпына келтіру элементі: өткен ақпараттың қанша мөлшерін ұмыту керектігін шешеді;
- Деректерді жаңарту: ескі ақпараттың қай бөлігін сақтау керектігін және қайсысын қосу керектігін анықтайды.

GRU желілерінің қарапайымдылығы есептеулерді жылдамдатып және іске асыруды жеңілдетуге мүмкіндік береді, сонымен бірге жойылып бара жатқан градиент мәселесін тиімді шешеді.

Ұзақ мерзімді тәуелділіктер - бұл қайталанатын нейрондық желінің нәтижесі көптеген қадамдар бұрын алынған кірістерге байланысты болатын жағдайлар. Бұл ұзақ мерзімді тәуелділік және мұндай ұсыныстарды жасауға немесе талдауға тырысатын қайталанатын нейрондық желінің жұмысына әсер етуі мүмкін.

LSTM және GRU әдістерін салыстыру.

1. Қақпалар саны:

- LSTM: Қақпаның үш түрі бар - ұмыту қақпасы (forget gates), кіру қақпасы (input gates) және шығу қақпасы (output gates);
- GRU: қақпаның екі түрі бар - жаңарту қақпасы және қалпына келтіру қақпасы.

2. Ұяшық күйі:

- LSTM: ақпарат ағынын қосымша басқаруға мүмкіндік беретін бөлек ұяшық күйі мен жасырын күйді қамтиды;
- GRU: тек жасырын күйді пайдаланады, бұл оның құрылымын жеңілдетеді.

3. Ақпаратты басқару:

- LSTM: ұмыту қақпасы мен кіріс қақпасы ақпарат ағынын дербес басқарады, бұл қандай ақпараттың сақталуын немесе жойылуын дәлірек реттеуге мүмкіндік береді;
- GRU: жаңарту қақпасы бір уақытта өткен күйден сақталған ақпаратты және қосылған жаңа ақпаратты бақылайды, бұл әлемдік реттеуге әкеледі.

Егер оқу жылдамдығы маңызды болса және қол жетімді есептеу ресурстары шектеулі болса, GRU қарапайым құрылымына және LSTM-ге қарағанда аз параметрлерге байланысты қолайлы болады. Қарапайым немесе кішірек деректер жиыны үшін GRU жеткілікті болса, өнімділікті жоғалтпай, деректерде лайықты түрде тиімді оқи алады.

```
from keras.models import Sequential
from keras.layers import LSTM, Dense

model = Sequential()
model.add(LSTM(50, input_shape=(100, 1))) # 100 time steps, 1 feature
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')

from keras.layers import GRU

model = Sequential()
model.add(GRU(50, input_shape=(100, 1)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
```

2-сурет. Негізгі LSTM мен GRU моделі.

Екі модельді де уақыт тізбегін талдау үшін сәтті пайдалануға болады, бірақ олардың арасындағы таңдау тапсырмаға байланысты. Егер есептеу қуаты шектеулі болса немесе деректердің тәуелділігі қысқа болса, онда GRU неғұрлым қолайлы таңдау болуы мүмкін. Егер деректердегі күрделі және ұзақ мерзімді тәуелділіктерді ескеру қажет болса, онда LSTM жақсырақ болады (2-сурет).

```

model = Sequential()
model.add(LSTM(100, input_shape=(100, 1)))
model.add(Dense(10, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

```

```

model = Sequential()
model.add(GRU(50, return_sequences=True, input_shape=(100, 1)))
model.add(GRU(50))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')

```

3-сурет. Бірнеше қабатты LSTM мен GRU.

LSTM және Gru көп деңгейлі модельдері модельдің күрделі және көп қабатты деректерден үйрену қабілетін жақсарту үшін қайталанатын нейрондық желілердің бірнеше қабаттарын пайдаланатын негізгі модельдердің кеңеюі болып табылады (3-сурет).

```

model = Sequential()
model.add(LSTM(100, input_shape=(100, 1)))
model.add(Dense(10, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

```

```

model = Sequential()
model.add(GRU(100, input_shape=(100, 1)))
model.add(Dense(10, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

```

4-сурет. Жіктеуге арналған LSTM мен GRU.

Қайталанатын нейрондық желілер мәтін, уақыт қатарлары немесе алдыңғы мәндерге тәуелділік маңызды басқа деректер түрлері сияқты дәйекті деректерді өңдеу үшін пайдаланылады (4-сурет).

```

model = Sequential()
model.add(LSTM(100, return_sequences=True, input_shape=(100, 1)))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

```

```

model = Sequential()
model.add(GRU(100, return_sequences=True, input_shape=(100, 1)))
model.add(GRU(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

```

5-сурет. Стекті LSTM мен GRU.

Стек LSTM және GRU (Stacked LSTM and GRU)-LSTM (Long Short-Term Memory) және Gru (Gated Recurrent Unit) сияқты стандартты қайталанатын нейрондық желілердің (RNN) кеңейтілген нұсқалары, олар стек (немесе көп қабатты) ішінде бір-бірінің үстінде орналасқан қайталанатын нейрондық желілердің бірнеше қабаттарынан тұрады түрінде. Бұл модельдерге деректердегі неғұрлым күрделі және дерексіз үлгілерді тиімдірек түсіруге және үйретуге мүмкіндік береді (5-сурет).

```
from keras.layers import Dropout

model = Sequential()
model.add(LSTM(100, input_shape=(100, 1)))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
model = Sequential()
model.add(GRU(100, input_shape=(100, 1)))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

6-сурет. Dropout реттеуі бар LSTM GRU.

Dropout-қайта оқытуды болдырмау үшін нейрондық желілерде қолданылатын реттеу әдісі. Бұл модель белгілі бір нейрондарға тым көп тәуелді болмас үшін және деректерді жалпылауды үйрену үшін оқытудың әр қадамында нейрондардың бір бөлігін кездейсоқ "өшіруден" тұрады. Бұл модельдің бұрын-соңды болмаған жаңа деректермен жұмыс істеу қабілетін жақсартады (6-сурет).

Dropout көмегімен LSTM және GRU туралы айтатын болсақ, бұл әдіс мәтінді жіктеу, уақыт қатарлары немесе басқа дәйекті деректер сияқты күрделі тапсырмаларда модельдердің жалпы өнімділігін жақсарту үшін қолданылады.

LSTM мен GRU арасындағы таңдау көптеген факторларға, соның ішінде деректердің мөлшері мен сипатына ие, қажетті өңдеу қуатына және жобаның нақты мақсаттарына байланысты. LSTM неғұрлым күрделі құрылымға ие және ақпараттың ұзақ реттілікте сақталуын қамтамасыз етсе де, Gru өзінің тиімділігі мен шағын және орташа деректердегі оқу жылдамдығына байланысты жеңеді.

Қорыта келгенде, уақыт қатарларын талдау әдістері нақты өмірде негізделген шешімдер қабылдау үшін өте маңызды болып қала береді және әзірлеушілер осы әдістерді одан әрі жақсарту үшін жұмыс істеуді жалғастыруда.

Болашақта нейрондық желілер сияқты тереңірек машиналық оқытуды және SARIMA көмегімен терең оқыту және болжау үлгілері сияқты дәлірек статистикалық үлгілерді пайдалану арқылы уақыт серияларын талдау және болжау әдістерін жақсартуға болады. Алайда, болжамдардың дәлдігі аймаққа байланысты әр түрлі болуы мүмкін және нәтижелерді қосымша талдауды қажет ететіндігін ескеру қажет.

Машиналық оқыту мен тереңдетілген оқытудың пайда болуы уақыт қатарларын болжау мүмкіндіктерін едәуір кеңейтті. Сызықтық регрессияны, кездейсоқ ормандарды және градиентті күшейтуді қоса алғанда, машиналық оқыту әдістері кідіріс көрсеткіштері мен маусымдық әсерлер сияқты қосымша мүмкіндіктерді қосуға мүмкіндік береді, бұл оларды

болашақ тенденцияларды болжауда икемді және тиімдірек етеді. Бұл әдістер сызықтық емес қатынастарды модельдеуде жақсы жұмыс істейді және уақыт қатарларының әртүрлі типтеріне бейімделе алады.

Lstm, GRU, TCN және Transformer сияқты терең оқыту үлгілері ұзақ мерзімді тәуелділіктерді анықтау және деректердегі күрделі үлгілерді ашу арқылы уақыт қатарын талдауды келесі деңгейге көтерді. Олардың үлкен деректер жиынтығын өңдеу және күрделі қатынастарды үйрену қабілеті оларды дәстүрлі әдістер жетіспейтін сценарийлерде ерекше құнды етеді. Бұл модельдер жоғары өзгергіштікті немесе ұзақ мерзімді тенденцияларды қамтитын тапсырмаларды болжауда тиімділігін дәлелдеді, бұл жақсартылған дәлдік пен түсінікті қамтамасыз етеді.

Уақыттық қатарларды талдаудың сәйкес әдісін таңдау деректердің сипатына және талдаудың нақты мақсаттарына байланысты. Классикалық статистикалық тәсілдер қарапайым стационарлық уақыт қатарлары үшін қолайлы, ал машиналық оқыту және тереңдетіп оқыту әдістері күрделі, сызықтық емес және динамикалық деректер үшін жақсартылған өнімділікті қамтамасыз етеді. Дәстүрлі модельдерді машиналық оқытудың озық әдістерімен және тереңдетіп оқыту әдістерімен үйлестіре отырып, дәлірек болжамдарға қол жеткізуге және тереңірек түсінік алуға болады, бұл сайып келгенде әртүрлі салаларда шешім қабылдаудың жақсаруына әкеледі.

Қолданылған әдебиеттер тізімі

1. М.Ф.Баймухамедов, А.М.Баймухамедова, С.Н.Боранбаев. Жасанды интеллект: қазіргі заманғы теория және тәжірибе. Оқу құралы, 1-бөлім/ - Алматы: Бастау, 2020. - 248 б.
2. П.А.Пылов. Основы работы с моделями машинного и глубокого обучения: учебное пособие/ - Москва, Вологда: Инфра-Инженерия, 2023. - 256 с.
3. В.И. Горбаченко. Машинное обучение: учебное пособие / - М.: Ай Пи Ар Медиа, 2023. - 218 с.
4. С.В.Запечников. Основы интеллектуального анализа данных и машинного обучения. конспект лекций. учебное пособие / - Москва: Национальный исследовательский ядерный университет «МИФИ», 2022. - 136 с.
5. А.Ж.Асамбаев. Жасанды интеллект негіздері. - Алматы: Эверо, 2017. - 168 б.
6. Джон Д.Келлехер Глубокое обучение. Самый краткий и понятный курс глава 1, стр 8.