УДК 003.3; 004.85

РАЗРАБОТКА ПРЕДСКАЗАТЕЛЬНЫХ МОДЕЛЕЙ ДЛЯ РАСПОЗНАВАНИЯ РУКОПИСНЫХ БУКВ КЫРГЫЗСКОГО ЯЗЫКА

Орозобекова А. К., Турдубаева А. Б.
КГТУ им. И. Раззакова

В этой статье рассматривается модель распознавания рукописных букв, связанных с кыргызским языком, на основе искусственной нейронной сети. Эта модель была обучена с использованием набора данных, состоящего из 80213 рукописных букв, собранных Жумаевым И. (Дата Сайнтист). Результат точности составил 89%.

Ключевые слова: нейронная сеть, компьютерное распознавание, обработка естественного языка, глубокое обучение.

КЫРГЫЗ ТИЛИНИН КОЛ ЖАЗМА ТАМГАЛАРЫН БОЛЖОЛДОП ТААНУУ МОДЕЛИН ИШТЕП ЧЫГУУ

Орозобекова А. К., Турдубаева А. Б.
И. Раззаков атн. КМТУ

Бул макалада жасалма нейрон тармагынын негизинде Кыргыз тилине таандык кол менен жазылган тамгаларды таануу модели талкууланат. Бул модель Жумаев И (Дата Сайнтист) чогулткан 80213 кол жазма тамгалардан турган маалыматтар топтому (датасет) боюнча окутулган. Тактык натыйжасы 89% ды түздү.

Баштапкы сөздөр: нейрон тармагы, компьютердик таануу, табыгый тилди иштетүү, терең үйрөнүү.

DEVELOPMENT OF A MODEL FOR PREDICTIVE RECOGNITION OF HANDWRITTEN LETTERS OF THE KYRGYZ LANGUAGE

Orozobekova A. K., Turdubaeva A. B.
I. Razzakov named after KSTU

This article discusses a model for recognizing handwritten letters associated with the Kyrgyz language based on an artificial neural network.

This model was trained using a dataset consisting of 80213 handwritten letters collected by I. Zhumaev (Data Scientist). The accuracy result was 89%.

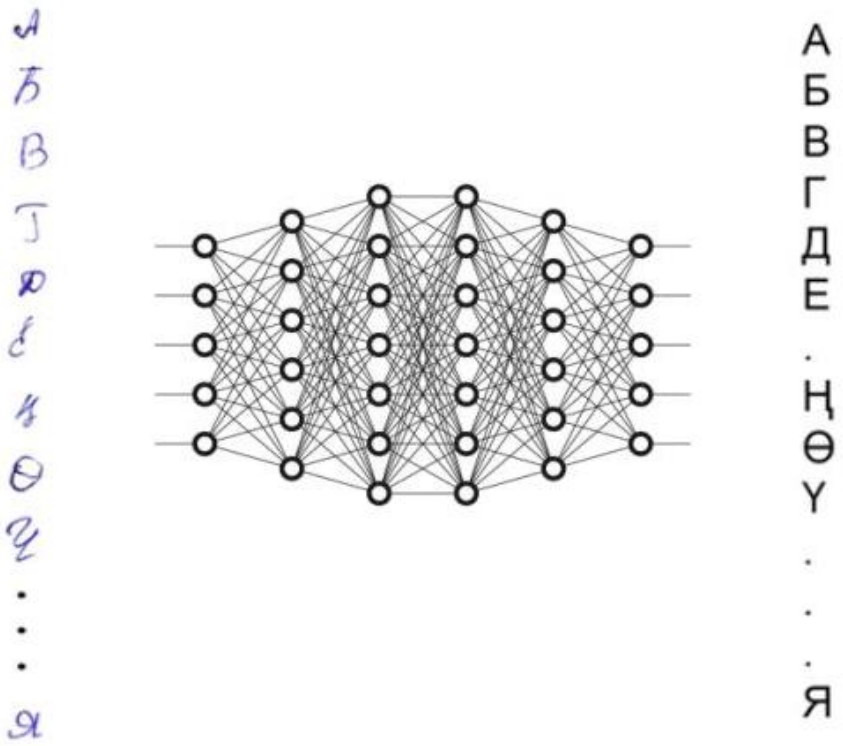
Keywords: neural network, computer science, natural language processing, deep learning.

Колжазманы таануу (HTR) – бул компьютердин жардамы менен колжазманы автоматтык түрдө транскрипциялоо ыкмасы. Колжазманы санариптештирүү көптөгөн компаниялардын бизнес процесстерин автоматташтыруу аркылуу адамдын ишин жеңилдетет. Бүгүнкү күндө текст таануу сыяктуу илим тармагы активдүү өнүгүп жатат. Изилдөөчүлөр текст таануу көйгөйүнө чоң көңүл бурушат. Адатта, компьютердик таануунун эки түрү бар. Эгерде символдор басма форматта болсо, анда таануу оптикалык белгилерди таануу (Optical Character Recognition), эгерде символдор кол менен жазылса – кол жазманы таануу процесси (Handwritten Text Recognition) деп аталат. Кол жазма текстин таануу кыйла татаал маселеси болгондуктан дагы толук чечиле элек. Ушуга байланыштуу бул теманы кароо актуалдуу деп эсептейбиз.

Ар бир табыгый тилдин өзгөчөлүгү болгондой эле Кыргызстанда кыргыз кирилл ариби 36 тамгадан туруп: 33 орус алфавитинен, жана кошумча кыргыз тилинин тыбыштык өзгөчөлүгүнө ылайык 3 тамгасы бар алар: Ң, Ү, Ө.

Моделди үйрөтүү үчүн көп сандагы аркандай адамдар жазган кол жазма тамгаларынан турган 80213 сүрөттөр колдонулду.

Нейрондук тармактын моделин түзүү. Бүгүнкү күндө текстти таануу үчүн китепканалардын кыйла саны түзүлгөн. Китепканаларды пайдалануу менен компьютердик таануу системасын иштеп чыгуу бир топ жөнөкөйлөтөт.

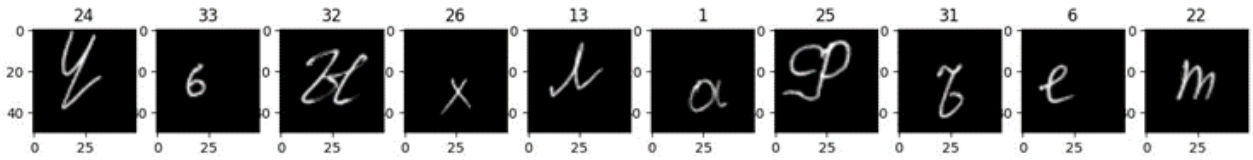


1 - сүрөт. Кол жазма тамгалардын үлгүсү.

Компьютердик таануу үчүн абдан популярдуу китепканалардын бири TensorFlow китепкансынын үстүндө иштеген Keras модулдук китепканасы колдонулду. Массивдер жана визуализациялар менен иштөө үчүн numpy жана matplotlib китепканаларын колдондук. Бул модель Google Colab булут чөйрөсүндө иштетилди, ал машина үйрөнүү долбоорлорун түздөн-түз браузерде сынап көрүүгө мүмкүндүк берет.

```
import numpy as np
import matplotlib.pyplot as plt
import gzip
from typing import List
from sklearn.preprocessing import OneHotEncoder
import tensorflow.keras as keras
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score
from keras.utils import to_categorical
import itertools
%matplotlib inline
```

2- сүрөт. Керектүү библиотекаларды жүктөө.



3 - сүрөт. Маалыматтар топтомун визуализациалоо.

Датасет окутуу train_test жана тестирлөө validation улгүсү болуп бөлүнөт.

```
training_dataset_x = input_train_x.reshape(-1, 50, 50, 1)
datagen = ImageDataGenerator(rescale=1.0/255.0,
        featurewise_center=True,
        samplewise_center=True,
        featurewise_std_normalization=True,
        samplewise_std_normalization=True,
        zca_whitening=False,
        rotation_range=20,
        zoom_range = 0.1,
        width_shift_range=0.1,
        height_shift_range=0.1,
        horizontal_flip=False,
        vertical_flip=False)
Y_train = to_categorical(input_train_y, num_classes=36)
train_iterator = datagen.flow(training_dataset_x, Y_train, batch_size=64 )
```

```
[ ] bx, by = train_iterator.next()
    bx.shape
```

4 - сүрөт. Маалыматтар топтому камтыган сүрөттөрдү иштетүүгө алдын ала окутуу үчүн маалымат топтомун кеңейтүү.

```
model2 = keras.Sequential([
    keras.layers.InputLayer(input_shape=(50, 50, 1)),
    keras.layers.Conv2D(32, kernel_size=(3, 3), activation='relu'),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Dropout(0.25),
    keras.layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Conv2D(128, kernel_size=(3, 3), activation="relu"),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Conv2D(256, kernel_size=(3, 3), activation="relu"),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),

    keras.layers.Dropout(0.25),
    keras.layers.Flatten(),

    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(36, activation='softmax')
])
model2.compile(optimizer='adam',
               loss='categorical_crossentropy',
               metrics=['accuracy'])
```

5 - сүрөт. Моделдин катмары.

Моделдин катмары (орус. сверточные слои) удаалаш түзүлгөндүктөн Sequential тибиндеги моделин алдык. keras.layers.Dense(128, - бул киргизүү катмары 128 нейрондон турат. Ар бир нейронго сүрөттөрдүн пикселдери жөнөтүлөт. activation='relu' – бул активация функциясы. Чыгаруу катмарынын класстары алфавиттин санына жараша 36 нейрондон турат. Активдештирүү функциясы 'softmax'. Окутуудан мурун моделди компиляциялоо керек. model3.compile(optimizer='adam', - окутуу параметрлерин көргөзүп, adam оптимизаторун колдондук. loss='categorical_crossentropy', - каталарды көргөзүүчү функцияссы ал эми metrics=['accuracy']) – сапат параметири.

Моделди үйрөтүү. Supervised learning (орус. обучение с учителем) болгондуктан биз үйрөтүүчү x_train жана туура жоопторду камтыган y_train функцияларын өткөрүп бердик. Ошондой эле epochs санынын параметри көргөзүлдү. Бир epochs: биздин бардык маалыматтар топтомубуз нейрондук тармак аркылуу бир жолу өтөт дегенди билдирет. epochs саны маалыматтардын көлөмүнүө жараша болот.

```
normalized_x = training_dataset_x / 255
history2 = model.fit (normalized_x, Y_train, epochs=15 )
```

```
Epoch 1/15
168/168 [==============================] - 27s 160ms/step - loss: 0.0091 - accuracy: 0.9974
Epoch 2/15
168/168 [==============================] - 27s 161ms/step - loss: 0.0089 - accuracy: 0.9981
Epoch 3/15
168/168 [==============================] - 28s 167ms/step - loss: 0.0121 - accuracy: 0.9970
Epoch 4/15
168/168 [==============================] - 27s 162ms/step - loss: 0.0113 - accuracy: 0.9961
Epoch 5/15
168/168 [==============================] - 27s 159ms/step - loss: 0.0092 - accuracy: 0.9970
Epoch 6/15
168/168 [==============================] - 27s 163ms/step - loss: 0.0029 - accuracy: 0.9991
Epoch 7/15
168/168 [==============================] - 28s 166ms/step - loss: 0.0010 - accuracy: 0.9998
Epoch 8/15
168/168 [==============================] - 27s 160ms/step - loss: 0.0186 - accuracy: 0.9946
Epoch 9/15
168/168 [==============================] - 29s 173ms/step - loss: 0.0076 - accuracy: 0.9976
Epoch 10/15
168/168 [==============================] - 26s 157ms/step - loss: 0.0084 - accuracy: 0.9976
Epoch 11/15
168/168 [==============================] - 27s 158ms/step - loss: 0.0035 - accuracy: 0.9985
Epoch 12/15
168/168 [==============================] - 26s 156ms/step - loss: 0.0039 - accuracy: 0.9993
Epoch 13/15
168/168 [==============================] - 26s 157ms/step - loss: 0.0271 - accuracy: 0.9931
Epoch 14/15
168/168 [==============================] - 26s 157ms/step - loss: 0.0054 - accuracy: 0.9980
Epoch 15/15
168/168 [==============================] - 26s 157ms/step - loss: 0.0074 - accuracy: 0.9980
```

6 - сүрөт. Н ормализация жана моделди үйрөтүү.

Epoch сапттын аягында loss параметири аркылуу каталар функциясын көрүүгө болот ал эми болжолдоо тактыгын accuracy параметри көргөзүп турат, жана ар бир Epoch саптары уланган сайын loss параметири азайып, accuracy параметри жогоруланганын байкайбыз. Демек моделибиз уламдам улам тагыраак иштөөдө деп айтсак болот. Бирок моделди үйрөтүүдө, терең нейрон тармактарынын көйгөйлөрүнүн бири - ашыкча үйрөтүү (орус.переобучение) коркунучун да көзөмөлдөп туруу зарыл. Ашыкча үйрөтүү – бул үйрөтүлгөн модель топтомунун мисалдарын гана жакшы таанып, окутуу процесине катышпаган башка мисалдарды тааныбаган же начар тааныган көрүнүш. Эгерде тармак өтө көп убакытта үйрөтүлсө же өтө көп параметр (салмактар) колдонулса, ашыкча үйрөнүү пайда болот. Бул белгилүү бир учурдан тартып нейрон тармагы жалпы маалыматтарга көз каранды болбой, аномалдуу маанилерди, каталарды камтыган айрым мисалдардын өзгөчөлүктөрүнө "үйрөнө" баштайт. Ашыкча үйрөнүү болбош үчүн Dropout колдондук бул терең нейрон тармактарында окуу көйгөйүн алдын алууга жардам берет.

```
predictions
y_pred = np.argmax(predictions, axis=1)
y_pred
u, c = np.unique(y_pred, return_counts=True)
print(dict(zip(u, c)))
y_pred += 1
u, c = np.unique(y_pred, return_counts=True)
print(dict(zip(u, c)))
len(raw_validation['id'].values)
len(y_pred)
```

```
{1: 4162, 2: 1553, 3: 667, 5: 2522}
{2: 4162, 3: 1553, 4: 667, 6: 2522}
8904
```

7 – сүрөт. Predictions. Тактыкты тесттик үлгү менен сындоодо 89% көргөздү.

Демек нейрондук моделибизди колдонуу менен кыргыз алфавитинин тамгаларын 89% так божомолдоп бере алабыз.

Корутунду :

- Маалыматтар топтомун окуу жана тесттик топтомдорго бөлүү.
- Маалыматтар топтому камтыган сүрөттөр оптималдаштырылды.
- Нейрон тармагынын архитектурасы түзүлдү.
- Окутуу параметрлери түзүлдү.
- Модель үйрөтүлдү жана predictions аркылуу текшерилди.

Тактыкты дагы жогорулатуу үчүн моделди өзгөртүп көрүүнү сунуштайбыз, мисалы башка оптимизатор колдонуп же Epoch санын көбөйтүп.

АДАБИЯТТАР

1. Львович И.Я, Мозговой А.А. “Моделирование распознования рукописного текста на основе скрытых марковских моделей” Воронеж 2016
2. Хобсон Лейн, Ханнес Хапке, Коул Ховард «Обработка естественного языка в действии» Питер 2020
3. «Нейросетевые методы в обработке естественного языка» / пер. с анг. А. А. Слинкина. – М.: ДМК Пресс, 2019. – 282 с.: ил.
4. https://habr.com/ru/post/533350/
5. https://habr.com/ru/post/505616/https://elibrary.ru/download/elibrary_36476386_55623147.pdf