

004.89

S.N. DEVITSYNA, AS. KARGIN, T.N. BALABANOVA

THE DEVELOPMENT OF A NEURAL NETWORK MODEL FOR VOICE AUTHENTICATION OF A USER

The paper uses the possibilities of applying the neural network technologies for biometric authentication systems. The voice is selected as the biometrics. A simple multi-layer perceptron model is used for voice authentication.

Keywords: information technologies, artificial intelligence, neural network technologies, biometric authentication, information security.

В настоящее время в мире наблюдается стремительное развитие информационных технологий, что приводит к появлению новых возможностей для решения различных задач. В частности, в области информационной безопасности и защиты данных, одним из наиболее перспективных направлений является применение нейронных сетей. Нейронные сети позволяют решать задачи распознавания образов, классификации данных и т.д. В данной статье рассматривается применение нейронных сетей для задачи аутентификации пользователя по его голосу. Для решения этой задачи используется простая модель многоуровневой перцепции (МЛП). В качестве биометрических данных используется голос пользователя. В работе описаны алгоритмы обучения и тестирования модели. Результаты экспериментов показывают, что предложенная модель способна эффективно решать задачу аутентификации пользователя по его голосу. В работе также приведены ссылки на литературу, посвященную данной теме.

Ключевые слова: информационные технологии, искусственный интеллект, технологии нейронных сетей, биометрическая аутентификация, информационная безопасность.

I. Введение

В настоящее время в мире наблюдается стремительное развитие информационных технологий, что приводит к появлению новых возможностей для решения различных задач. В частности, в области информационной безопасности и защиты данных, одним из наиболее перспективных направлений является применение нейронных сетей. Нейронные сети позволяют решать задачи распознавания образов, классификации данных и т.д. В данной статье рассматривается применение нейронных сетей для задачи аутентификации пользователя по его голосу. Для решения этой задачи используется простая модель многоуровневой перцепции (МЛП). В качестве биометрических данных используется голос пользователя. В работе описаны алгоритмы обучения и тестирования модели. Результаты экспериментов показывают, что предложенная модель способна эффективно решать задачу аутентификации пользователя по его голосу. В работе также приведены ссылки на литературу, посвященную данной теме.

II. Описание модели

Для решения задачи аутентификации пользователя по его голосу используется простая модель многоуровневой перцепции (МЛП). В качестве биометрических данных используется голос пользователя. В работе описаны алгоритмы обучения и тестирования модели. Результаты экспериментов показывают, что предложенная модель способна эффективно решать задачу аутентификации пользователя по его голосу. В работе также приведены ссылки на литературу, посвященную данной теме.

- info() -
.

```
class Database:

    def __init__(self, path):

        self.df = self.open_db(path)

    def open_db(self, path):
        return pd.readcsv(path)

    def convert_labels(self):

        self.df.loc[self.df['label']=='male', 'label'] = 0
        self.df.loc[self.df['label']=='female', 'label'] = 1

    def split_data(self, df):
        train_input = df.sample(frac=0.8, random_state=200)
        test_input = df.drop(train_input.index)
        return train_input, test_input

    def get_data(self):
        self.convert_labels()
        train_input, test_input = self.split_data(self.df)

        np.random.seed(1)

        self.testX = np.array(test_input.iloc[:, 0:20])
        self.test_Y = np.array(test_input.iloc[:, 20])
        self.trainX = np.array(train_input.iloc[:, 0:20])
        self.train_Y = np.array(train_input.iloc[:, 20])
```

3 -

```
class Model():

    def __init__(self, path):
        self.db = Database(path)
        self.db.get_data()

    def call(self):
        self.model = Sequential()
        self.model.add(Dense(4, input_dim=20, activation='relu'))
        self.model.add(Dense(2, activation='relu'))
        self.model.add(Dense(1, activation='sigmoid'))
        self.model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

    def train(self, epochs=100, batch_size=16):
        start_time = time.time()
        self.model.fit(self.db.train_X, self.db.train_Y, epochs=epochs, batch_size=batch_size, validation_split=0.2)
        print("FIT: {} seconds".format(time.time() - start_time))

    def test(self):
        results = self.model.evaluate(self.db.test_X, self.db.test_Y, batch_size=128)
        print("TEST RESULT: test loss {}, test accuracy {}".format(results[0], results[1]))

    def info(self):
        self.model.summary()
        plot_model(self.model, 'keras_model.png', show_shapes=True)
```

4 -

```
65 def main():
66
67     model = Model('db_voices.csv')
68
69     model.call()
70
71     model.train(epochs=200, batch_size=16)
72
73     model.test()
74
75     model.info()
76
77 if __name__ == "__main__":
78     try:
79         main()
80     except:
81         raise
```

5 -

```
root@kali:~/projects/voice_verification$ cd Vprojects/voice_verification
root@kali:~/projects/voice_verification$ ls
class.py db_vocals.csv keras_1tddel.png main.py test.py
root@kali:~/projects/voice_verification$ python3 main.py
```

6 -

.7.

```
root@kali:~/projects/voice_verification$ python3 main.py
root@kali:~/projects/voice_verification$ python3 main.py
Epoch 191/288 - 0s 922us/step - loss: 8.1762 - accuracy: 8.9373 - val_loss: 8.1668 - val_accuracy: 8.9467
Epoch 192/288 - 0s 921us/step - loss: 8.1795 - accuracy: 8.9334 - val_loss: 8.1658 - val_accuracy: 8.9448
Epoch 193/288 - 8s 947us/step - loss: 8.1733 - accuracy: 8.9383 - val_loss: 8.1712 - val_accuracy: 8.9448
Epoch 194/288 - 8s 928us/step - loss: 8.1882 - accuracy: 8.9359 - val_loss: 8.1443 - val_accuracy: 8.9566
Epoch 195/288 - 8s 918us/step - loss: 8.1881 - accuracy: 8.9319 - val_loss: 8.1549 - val_accuracy: 8.9467
Epoch 196/288 - 8s 987us/step - loss: 8.1795 - accuracy: 8.9393 - val_loss: 8.1432 - val_accuracy: 8.9527
Epoch 197/288 - 8s 898us/step - loss: 8.1687 - accuracy: 8.9398 - val_loss: 8.1441 - val_accuracy: 8.9527
Epoch 198/288 - 0s 948us/step - loss: 8.1738 - accuracy: 8.9359 - val_loss: 8.1521 - val_accuracy: 8.9428
Epoch 199/288 - 8s 951us/step - loss: 8.1614 - accuracy: 8.9388 - val_loss: 8.1619 - val_accuracy: 8.9428
Epoch 200/288 - 0s 987us/step - loss: 8.1648 - accuracy: 8.9423 - val_loss: 8.1524 - val_accuracy: 8.9428
FIT: 24.471717834472656 seconds
5/5
TEST RESULT: test_loss 8.28388242738246918, test_accuracy 8.9385993914684187
Model: "sequential"

Layer (type) Output Shape
dense (Dense) (None, 4)
dense_1 (Dense) (None, 2)
dense_2 (Dense) (None, 1)

Total params: 97
Trainable params: 97
Non-trainable params: 8
root@kali:~/projects/voice_verification$
```

7 -

.7,
accuracy (0.9423 => 94 %) —
val_accuracy (0.9428 => 94 %) —
test_accuracy (0.9305 => 93 %) —

93 %.

1. [2013, 4 (28)].
: <https://elibrary.ru/item.asp?id=21375345> (20.06.2020).
2. [19.06.2020].
Habr. 2017. : <https://habr.com/ru/post/336198/> (19.06.2020).
3. [19.06.2020].
Information Security. 2009. 4. : <http://lib.itsec.ru/artides2/Oborandteh/neyrosetevye-tehnologii-v-biznese> (19.06.2020).
4. [10.06.2020].
2019. .46. 1. .148-160. URL:
<https://elibrary.ru/item.asp?id=38468584> (10.06.2020).
5. Devitsyna S, Eletskaia T, Meshkov A. Developing facial recognition software to control access to campus facilities / InnoCSE 2019 - Innovative Approaches in Computer Science within Higher Education. Proceedings of the 2nd Workshop on Innovative Approaches in Computer Science within Higher Education, 2019, p. 68-76. Available at: <http://ceur-ws.org/Vol-2562>. (02.06.2020).
6. Quesada A. 5 algorithms to train a neural network [19.06.2020].
https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network (19.06.2020).
7. Chris Nicholson A Beginner's Guide to Neural Networks and Deep Learning [19.06.2020].
: [https://wiki.pathmind.com/neural-network#:~:text=Networks %20 %26 %20 Artificial %20Intelligence-,Neural %20Network %20 Definition,labeling %20or %20clustering %20raw %20input](https://wiki.pathmind.com/neural-network#:~:text=Networks%20%26%20Artificial%20Intelligence-,Neural%20Network%20Definition,labeling%20or%20clustering%20raw%20input). (19.06.2020).
8. Everything you need to know about Neural Networks and Backpropagation - Machine Learning Easy and Fun [19.06.2020].
: <https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a> (19.06.2020).
9. Mean, median, mode and range from Table [21.06.2020].
<http://www.alamandamaths.com/domains/data/calculate-mean-median-mode-and-range-from-a-frequency-table-7> (21.06.2020).

« »,
« »
. +79785038856
E-mail: sndevitsyna@sevsu.ru

« »,
« »
. +79821478530
E-mail: kyvees13@gmail.com

« »,
« »
. +79194323779
E-mail: sozonova@bsu.edu.ru