

Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Сибирский государственный университет
телекоммуникаций и информатики»

(СибГУТИ)

Лабораторная работа №4

Выполнил: студент IV курса ИВТ,

гр. ИП-713

Михеев Н.А.

Проверила: ассистент кафедры ПМиК

Морозова К.И.

Новосибирск, 2020 г.

Цель

Целью данной лабораторной работы является разработка нейронной сети для решения задачи классификации или регрессии в зависимости от набора данных в рамках варианта. Лабораторная работа предполагает разработку на языке программирования Python с использованием библиотеки Keras.

Вариант задания:

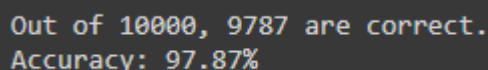
- Распознавание цифр на изображении (MNIST digits classification dataset)
- Нейросеть должна состоять из пяти полносвязных слоёв, обязательное использование ActivityRegularization, в качестве оптимизатора использовать RMSprop.

Результат работы

Для выполнения данного задания была разработана программа с использованием модуля keras из библиотеки tensorflow. Вся работа производилась на удаленном notebook в Google Collab.

Сначала был выгружен набор данных и произведена их нормализация, изменение формы обучающей и тестовой выборок по x для оптимизации с работой нейросети. Затем мы создали нашу последовательную (Sequential) модель, было задействовано 6 слоев, 4 из которых Dense (полносвязные) с различным количеством узлов, 2 из них ActivityRegularization для уменьшения ошибки обобщения в нейронных сетях с регуляризацией активности, в качестве метода регуляризации был выбран l1 - Активность рассчитывается как сумма абсолютных значений. Модель была скомпилирована с использованием оптимизатора, реализующего алгоритм RMSprop. Далее модель была обучена на обучающей выборке за 15 эпох, большее количество эпох не имеет смысла – идет переобучение модели.

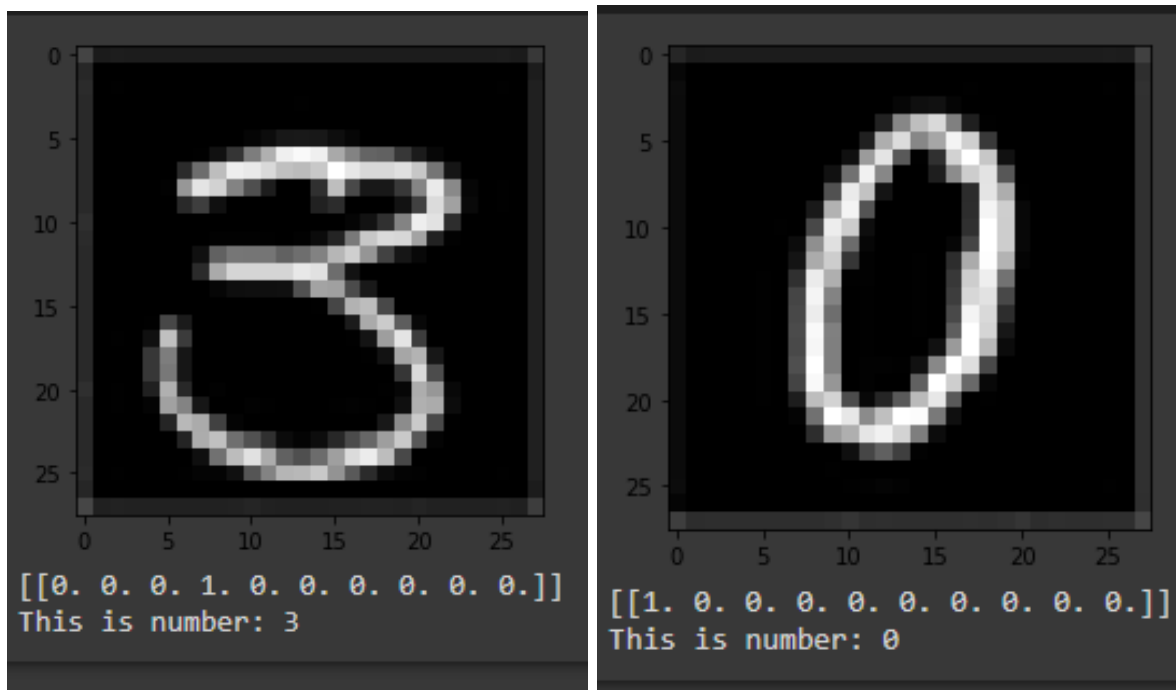
В результате модель имеет точность 99.8% на обучающей выборке, точность на тестовой выборке была подсчитана и равняется 97.87%.



```
Out of 10000, 9787 are correct.  
Accuracy: 97.87%
```

Рис.1 – Демонстрация точности работы программы.

Протестированные 2 изображения из интернета без проблем были определены нейросетью:



Листинг программы

```
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import *
from tensorflow.keras import utils
from tensorflow.keras.regularizers import *
from tensorflow.keras.preprocessing import image
import numpy as np

classes = [str(i) for i in range(10)]
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train.reshape(60000, 784)
x_test = x_test.reshape(10000, 784)
x_train = x_train / 255
x_test = x_test / 255
y_train = utils.to_categorical(y_train, 10)
y_test = utils.to_categorical(y_test, 10)

model = Sequential()
model.add(Dense(800, input_dim=784, activation='relu'))
model.add(Dense(400, activation='relu'))
model.add(ActivityRegularization(l1=0.001))
model.add(Dense(110, activation='relu'))
model.add(ActivityRegularization(l1=0.001))
model.add(Dense(len(classes), activation='softmax'))
model.compile(optimizer='RMSprop', loss='categorical_crossentropy', metrics=['accuracy'])
```

```

model.summary()

model.fit(x=x_train, y=y_train, batch_size=200, epochs=15, validation_split=0.2)

scores = model.evaluate(x_test, y_test, verbose=0)
print("Error: %.2f%%" % (100 - scores[1] * 100))

L=len(y_test)
correct = 0
fail = 0
YP=model.predict(x_test)
for i in range(L):
    if np.argmax(y_test[i]) == np.argmax(YP[i]):
        correct += 1
    else:
        fail += 1
print(f'Out of {L}, {correct} are correct.')
print(f'Accuracy: {correct / (correct + fail) * 100}%')

import matplotlib.pyplot as plt
from PIL import Image
from google.colab import files
files.upload()

img = Image.open('0.png').convert('L').resize((28, 28), Image.ANTIALIAS)
img = np.array(img)
plt.imshow(img, cmap = plt.cm.binary)
plt.show()
img = np.reshape(img,(1,784));
img = np.array([256-x for x in img])
y_pred = model.predict(img)
print(y_pred)
print(np.argmax(y_pred[0]))

```