

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по практической работе №6
по дисциплине «Искусственные нейронные сети»
Тема: Процесс решения задач с применением нейронных сетей в
библиотеке Keras

Студентка гр. 8383

Максимова А.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы

Необходимо построить сверточную нейронную сеть, которая будет классифицировать черно-белые изображения с простыми геометрическими фигурами на них.

К каждому варианту прилагается код, который генерирует изображения.

Для генерации данных необходимо вызывать функцию `gen_data`, которая возвращает два тензора:

1. Тензор с изображениями ранга 3
2. Тензор с метками классов

Уточнение:

- Выборки не перемешаны, то есть наблюдения классов идут по порядку
- Классы характеризуются строковой меткой
- Выборка изначально не разбита на обучающую, контрольную и тестовую

Задание

6 вариант

Классификация изображений по количеству крестов на них. Может быть 1, 2, 3.

Выполнение работы

1. Были импортированы все необходимые для работы классы и функции.

```
import var6
import matplotlib.pyplot as plt
from sklearn.utils import shuffle
from keras.utils import np_utils
from keras.models import Model
from keras.layers import Input, Convolution2D, MaxPooling2D, Dense, Dropout, Flatten
from sklearn.preprocessing import LabelEncoder
```

2. Загрузка данных была выполнена с помощью функции `gen_data()`, которая возвращает кортеж из 2 массивов NumPy.

```
dataset_image, dataset_labels = var6.gen_data(size=train_size+test_size)
```

3. Была выполнена обработка входных данных: данные выборки были перемешаны с помощью функции `shuffle()`, строковые метки классов преобразованы в целочисленные, выборка была разбита на обучающую, контрольную и тестовую.

```
def edit_data(image, labels):  
    image_shuffle, labels_shuffle = shuffle(image, labels)  
    encoder = LabelEncoder()  
    labels_shuffle_enc = encoder.fit_transform(labels_shuffle)  
    labels_shuffle_enc_cat = np_utils.to_categorical(labels_shuffle_enc)  
    return image_shuffle, labels_shuffle_enc_cat
```

```
# разделение данных  
train_image = dataset_image[0:train_size]  
test_image = dataset_image[train_size:]  
  
train_labels = dataset_labels[0:train_size]  
test_labels = dataset_labels[train_size:]
```

4. В результате многократного изменения модели, были выбраны следующие гиперпараметры:

```
# гиперпараметры  
batch_size = 16  
num_epochs = 20  
  
kernel_size = 3 # размер ядра 3x3  
pool_size = 2  
  
conv_depth_1 = 32 # кол-во ядер  
conv_depth_2 = 64  
  
drop_prob_1 = 0.25  
drop_prob_2 = 0.5  
  
dense_size_1 = 256 # кол-во нейронов в полносвязном слое  
dense_size_2 = 3
```

5. Была построена и обучена следующая модель сверточной нейронной сети:

```
# сеть: conv1[32] -> pool1 -> drop1 -> conv2[32] -> pool2 -> drop2 -> flatten -> dense1[256] -> drop3 -> dense2[3]
inp = Input(shape=(height, width, depth))
conv_1 = Convolution2D(filters=conv_depth_1, kernel_size=(kernel_size, kernel_size),
                      padding='same', activation='relu')(inp)
pool_1 = MaxPooling2D(pool_size=(pool_size, pool_size))(conv_1)
drop_1 = Dropout(rate=drop_prob_1)(pool_1)

conv_2 = Convolution2D(filters=conv_depth_2, kernel_size=(kernel_size, kernel_size),
                      padding='same', activation='relu')(drop_1)
pool_2 = MaxPooling2D(pool_size=(pool_size, pool_size))(conv_2)
drop_2 = Dropout(rate=drop_prob_1)(pool_2)

flat = Flatten()(drop_2)
dense_1 = Dense(dense_size_1, activation='relu')(flat)
drop_3 = Dropout(rate=drop_prob_2)(dense_1)
out = Dense(dense_size_2, activation='softmax')(drop_3)

model = Model(inputs=inp, outputs=out)
model.compile(loss='categorical_crossentropy', optimizer='Adam', metrics=['accuracy'])
hist = model.fit(train_image, train_labels, batch_size=batch_size, epochs=num_epochs,
                verbose=1, validation_split=0.25)
model.evaluate(test_image, test_labels, verbose=1)
```

6. В качестве протокола оценки было выбрано выделение из общей выборки отдельного проверочного набора данных.

Результаты обучения сверточной нейронной сети

График потерь нейронной сети на обучающих и тестовых данных

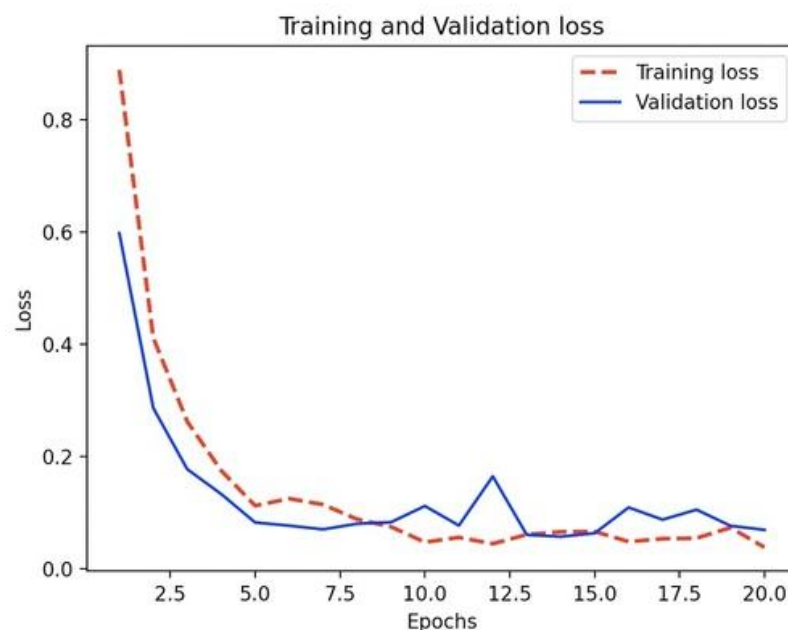
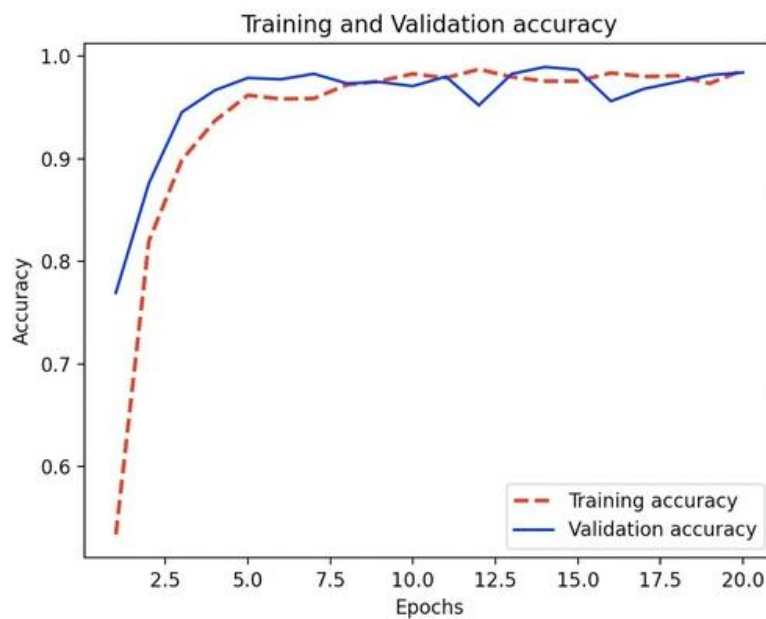


График точности нейронной сети на обучающих и тестовых данных



Значения работы нейронной сети на контрольных данных:

```
loss: 0.0475 - accuracy: 0.9910
```

Выводы

Была построена сверточная нейронная сеть, решающая задачу классификации из трех классов. Наибольшая достигнутая точность работы нейронной сети имела значение 99%.