Минобрнауки России

Федеральное государственное бюджетное образовательное

учреждение высшего образования

**НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ИМ. Р.Е. АЛЕКСЕЕВА**

ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

|  |  |  |
| --- | --- | --- |
|  |  |  |

Курс “Аппаратное и программное обеспечение роботизированных систем”

**Отчет по лабораторной работе №3**

Выполнил:

Балашов М.А.

Проверил: Гай В.Е.

Нижний Новгород 2021

Цель работы: Классификация изображений с использованием свёрточных нейронных сетей. Выполнить анализ статьи, разобрать структуру сети, реализовать сеть в Keras, оценить точность работы сети.

Вариант: данные- 4, модель сети – 3.

import keras

import tensorflow as tf

from tensorflow.python.keras.preprocessing.image import ImageDataGenerator

from keras.applications.resnet50 import ResNet50 , preprocess\_input

from keras.models import Sequential

from keras.layers.core import Flatten, Dense, Dropout

# Пути к наборам данных

train\_dir = '/content/drive/MyDrive/keras/train'

test\_dir = '/content/drive/MyDrive/keras/test'

# Размер изображений

img\_width, img\_height = 300, 300

# Размер мини-выборки

batch\_size = 30

# Кол-во изображений для обучения

nb\_train\_samples = 1034

# Кол-во изображений для теста

nb\_test\_samples = 133

# Создание генератора изображений

datagen = ImageDataGenerator(rescale=1. / 255)

train\_generator = datagen.flow\_from\_directory(

    train\_dir,

    target\_size=(img\_width, img\_height),

    batch\_size=batch\_size,

    class\_mode='categorical')

test\_generator = datagen.flow\_from\_directory(

    test\_dir,

    target\_size=(img\_width, img\_height),

    batch\_size=batch\_size,

    class\_mode='categorical')

# Создание экземпляра модели сети ResNet50

resnet50 = ResNet50(

    input\_shape=(img\_width, img\_height, 3),

    include\_top=False,

    weights="imagenet"

)

resnet50.trainable = False

resnet50.summary()

# Создание модели составной сети (Составная сеть)

model = Sequential()

# Добавляем сверточные слои

model.add(resnet50)

# Преобразуем двумерный массив MobileNet в одномерный

model.add(Flatten())

# Полносвязный слой

model.add(Dense(256, activation='relu'))

# Слой регуляризации (для предотвращения переобучения)

model.add(Dropout(0.4))

# Кол-во классов

model.add(Dense(3, activation='sigmoid'))

model.build(input\_shape = (None, 500, 500, 3))

model.summary()

model.add(Dense(3, activation='sigmoid'))

model.build(input\_shape = (None, 300, 300, 3))

model.summary()

Model: "sequential\_36"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

module\_wrapper\_36 (ModuleWra (None, 10, 10, 2048) 23587712

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_36 (Flatten) (None, 204800) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_72 (Dense) (None, 200) 40960200

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_36 (Dropout) (None, 200) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_73 (Dense) (None, 3) 603

=================================================================

Total params: 64,548,515

Trainable params: 40,960,803

Non-trainable params: 23,587,712

model.compile(loss='categorical\_crossentropy',

              optimizer='SGD',

              metrics=['accuracy'])

model.fit\_generator(

    train\_generator,

    steps\_per\_epoch=nb\_train\_samples // batch\_size,

    epochs=12,

    validation\_data=test\_generator,

    validation\_steps=nb\_test\_samples // batch\_size)

/usr/local/lib/python3.7/dist-packages/keras/engine/training.py:1915: UserWarning: `Model.fit\_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.

warnings.warn('`Model.fit\_generator` is deprecated and '

Epoch 1/12

34/34 [==============================] - 14s 314ms/step - loss: 1.6031 - accuracy: 0.8210 - val\_loss: 1.0989 - val\_accuracy: 0.8250

Epoch 2/12

34/34 [==============================] - 10s 294ms/step - loss: 1.3445 - accuracy: 0.8301 - val\_loss: 1.0988 - val\_accuracy: 0.8417

Epoch 3/12

34/34 [==============================] - 10s 292ms/step - loss: 1.1791 - accuracy: 0.8192 - val\_loss: 1.0985 - val\_accuracy: 0.8500

Epoch 4/12

34/34 [==============================] - 10s 293ms/step - loss: 1.0988 - accuracy: 0.8069 - val\_loss: 1.0987 - val\_accuracy: 0.8250

Epoch 5/12

34/34 [==============================] - 10s 291ms/step - loss: 1.0989 - accuracy: 0.8972 - val\_loss: 1.0986 - val\_accuracy: 0.8417

Epoch 6/12

34/34 [==============================] - 10s 293ms/step - loss: 1.0986 - accuracy: 0.8475 - val\_loss: 1.0986 - val\_accuracy: 0.8250

Epoch 7/12

34/34 [==============================] - 10s 290ms/step - loss: 1.0988 - accuracy: 0.8253 - val\_loss: 1.0986 - val\_accuracy: 0.8417

Epoch 8/12

34/34 [==============================] - 10s 289ms/step - loss: 1.0987 - accuracy: 0.8265 - val\_loss: 1.0983 - val\_accuracy: 0.8500

Epoch 9/12

34/34 [==============================] - 10s 289ms/step - loss: 1.0989 - accuracy: 0.8182 - val\_loss: 1.0988 - val\_accuracy: 0.8250

Epoch 10/12

34/34 [==============================] - 10s 294ms/step - loss: 1.0989 - accuracy: 0.8217 - val\_loss: 1.0984 - val\_accuracy: 0.8417

Epoch 11/12

34/34 [==============================] - 10s 291ms/step - loss: 1.0987 - accuracy: 0.8296 - val\_loss: 1.0986 - val\_accuracy: 0.8250

Epoch 12/12

34/34 [==============================] - 10s 290ms/step - loss: 1.0986 - accuracy: 0.8478 - val\_loss: 1.0984 - val\_accuracy: 0.8500

scores = model.evaluate\_generator(test\_generator, nb\_test\_samples // batch\_size)

print('Точность работы на тестовых данных: %.2f%%' % (scores[1]\*100))

/usr/local/lib/python3.7/dist-packages/keras/engine/training.py:1948: UserWarning: `Model.evaluate\_generator` is deprecated and will be removed in a future version. Please use `Model.evaluate`, which supports generators.

warnings.warn('`Model.evaluate\_generator` is deprecated and '

Точность работы на тестовых данных: 85.00%