Московский государственный технический университет имени Н.Э.Баумана

Кафедра «Системы обработки информации и управления»

ОТЧЕТ

Лабораторная работа №4
по дисциплине
«Проектирование интеллектуальных систем»
на тему
«Сохранение модели и TensorBoard»

Выполнил:

Студент ИУ5-24М

Гаврилюк А.Г.

Москва, 2020

In [1]:

```
import tensorflow as tf
import keras
from tensorflow.keras import datasets, models, layers
from keras.preprocessing.image import ImageDataGenerator
import os
from keras.constraints import maxnorm
from keras.optimizers import SGD
from keras.callbacks import ModelCheckpoint, TensorBoard
```

```
Using TensorFlow backend.
```

Параметры для обучения модели

In [2]:

```
BATCH_SIZE = 32
CLASSES_COUNT = 10
EPOCHS_COUNT = 5
PREDICTIONS_COUNT = 20
```

Загружаем и нормализуем датасет

In [3]:

```
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()
train_images, test_images = train_images / 255.0, test_images / 255.0
```

Обучающая и тестовая выборка

In [4]:

```
train_images.shape, test_images.shape
```

Out[4]:

```
((50000, 32, 32, 3), (10000, 32, 32, 3))
```

Функция создания модели сверточной сети

In [5]:

```
def create_model():
    model = models.Sequential()

    model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
    model.add(layers.MaxPooling2D((2, 2)))

    model.add(layers.Conv2D(64, (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))

    model.add(layers.Conv2D(64, (3, 3), activation='relu'))

    model.add(layers.Flatten())
    model.add(layers.Dense(64, activation='relu'))
    model.add(layers.Dense(10))

    model.compile(
        optimizer='adam',
        loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
        metrics=['accuracy']
    )

    return model
```

In [6]:

In [6]:

```
model = create_model()
model.summary()
```

```
WARNING:tensorflow:From /Users/alexandr/Учеба/Мага/giis/env/lib/python3.7/site-
packages/tensorflow_core/python/ops/resource_variable_ops.py:1630: calling
BaseResourceVariable.__init__ (from tensorflow.python.ops.resource_variable_ops) with constraint i
s deprecated and will be removed in a future version.
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d (Conv2D)              (None, 30, 30, 32)        896       
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 15, 15, 32)        0         
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 13, 13, 64)        18496     
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 6, 6, 64)          0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 4, 4, 64)          36928     
_________________________________________________________________
flatten (Flatten)            (None, 1024)              0         
_________________________________________________________________
dense (Dense)                (None, 64)                65600     
_________________________________________________________________
dense_1 (Dense)              (None, 10)                650       
=================================================================
Total params: 122,570
Trainable params: 122,570
Non-trainable params: 0
_________________________________________________________________
```

In [17]:

```
os.mkdir('model_weights')
```

Сохраняем контрольные точки и данные для построения графа

In [40]:

```
import datetime

checkpoint_path = 'model_weights/my_ckpt.ckpt'
tensoboard_logs_dir = 'logs/{}'.format(datetime.datetime.now().strftime('%d.%m.%Y_%H,%M,%S'))

my_callbacks = [
    ModelCheckpoint(filepath=checkpoint_path, save_weights_only=True,),
    TensorBoard(log_dir=tensoboard_logs_dir),
]
model.fit(
    train_images,
    train_labels,
    validation_data=(test_images, test_labels),
    epochs=EPOCHS_COUNT,
    batch_size=BATCH_SIZE,
    callbacks=my_callbacks
)
```

```
Train on 50000 samples, validate on 10000 samples
Epoch 1/5
50000/50000 [==============================] - 28s 551us/sample - loss: 0.6392 - acc: 0.7745 - val
_loss: 0.8641 - val_acc: 0.7101
Epoch 2/5
50000/50000 [==============================] - 26s 513us/sample - loss: 0.6042 - acc: 0.7882 - val
_loss: 0.8786 - val_acc: 0.7078
Epoch 3/5
50000/50000 [==============================] - 26s 520us/sample - loss: 0.5690 - acc: 0.7965 - val
_loss: 0.8657 - val_acc: 0.7171
Epoch 4/5
```

```
50000/50000 [==============================] - 26s 526us/sample - loss: 0.5320 - acc: 0.8113 - val
_loss: 0.8837 - val_acc: 0.7208
Epoch 5/5
50000/50000 [==============================] - 26s 520us/sample - loss: 0.5001 - acc: 0.8228 - val
_loss: 0.9553 - val_acc: 0.7044
```

Out[40]:

```
<tensorflow.python.keras.callbacks.History at 0x13b818110>
```

Создаем модель и производим расчеты на необученной модели

Видим, что точность равна 10%

In [20]:

```
model = create_model()
loss, accuracy = model.evaluate(test_images, test_labels)

loss, accuracy
```

```
10000/10000 [==============================] - 1s 114us/sample - loss: 2.3054 - acc: 0.1012
```

Out[20]:

```
(2.305408290863037, 0.1012)
```

Восстанавливаем веса модели

Видно, что теперь точность достигла 70%

In [21]:

```
model.load_weights(checkpoint_path)
loss, accuracy = model.evaluate(test_images, test_labels)

loss, accuracy
```

```
10000/10000 [==============================] - 1s 118us/sample - loss: 1.1974 - acc: 0.7030
```

Out[21]:

```
(1.19739929356575, 0.703)
```

Сохраняем всю модель

In [27]:

```
os.mkdir('model')
```

In [28]:

```
model = create_model()
model.fit(
    train_images,
    train_labels,
    validation_data=(test_images, test_labels),
    epochs=EPOCHS_COUNT,
    batch_size=BATCH_SIZE,
)
model.save('model/my_model.h5')
```

```
Train on 50000 samples, validate on 10000 samples
Epoch 1/5
50000/50000 [==============================] - 28s 567us/sample - loss: 1.5199 - acc: 0.4444 - val
_loss: 1.2350 - val_acc: 0.5575
Epoch 2/5
```

```
50000/50000 [==============================] - 27s 531us/sample - loss: 1.1402 - acc: 0.5980 - val
_loss: 1.0752 - val_acc: 0.6240
Epoch 3/5
50000/50000 [==============================] - 27s 534us/sample - loss: 0.9923 - acc: 0.6486 - val
_loss: 1.0815 - val_acc: 0.6195
Epoch 4/5
50000/50000 [==============================] - 27s 533us/sample - loss: 0.9032 - acc: 0.6810 - val
_loss: 0.9968 - val_acc: 0.6562
Epoch 5/5
50000/50000 [==============================] - 28s 551us/sample - loss: 0.8332 - acc: 0.7076 - val
_loss: 0.8873 - val_acc: 0.7002
```

Восстанавливаем модель

In [30]:

```
test_model = models.load_model('model/my_model.h5')
test_model.summary()
```

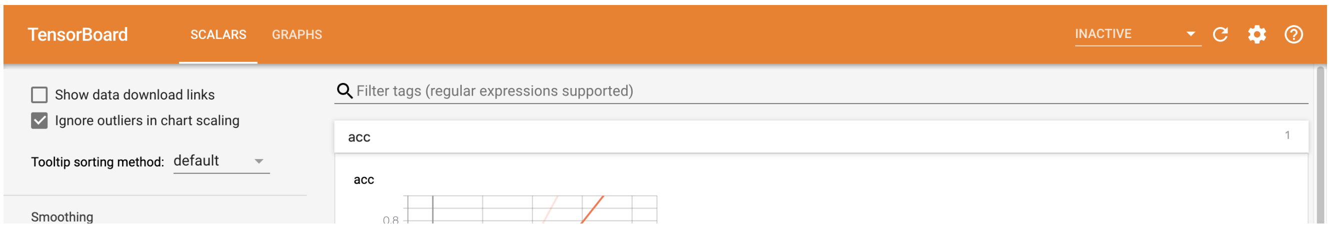
```
WARNING:tensorflow:From /Users/alexandr/Учеба/Мага/giis/env/lib/python3.7/site-
packages/tensorflow_core/python/ops/init_ops.py:97: calling GlorotUniform.__init__ (from
tensorflow.python.ops.init_ops) with dtype is deprecated and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
WARNING:tensorflow:From /Users/alexandr/Учеба/Мага/giis/env/lib/python3.7/site-
packages/tensorflow_core/python/ops/init_ops.py:97: calling Zeros.__init__ (from
tensorflow.python.ops.init_ops) with dtype is deprecated and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_18 (Conv2D)           (None, 30, 30, 32)        896       
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 15, 15, 32)        0         
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 13, 13, 64)        18496     
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 6, 6, 64)          0         
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 4, 4, 64)          36928     
_________________________________________________________________
flatten_6 (Flatten)          (None, 1024)              0         
_________________________________________________________________
dense_12 (Dense)             (None, 64)                65600     
_________________________________________________________________
dense_13 (Dense)             (None, 10)                650       
=================================================================
Total params: 122,570
Trainable params: 122,570
Non-trainable params: 0
_________________________________________________________________
```

In [31]:

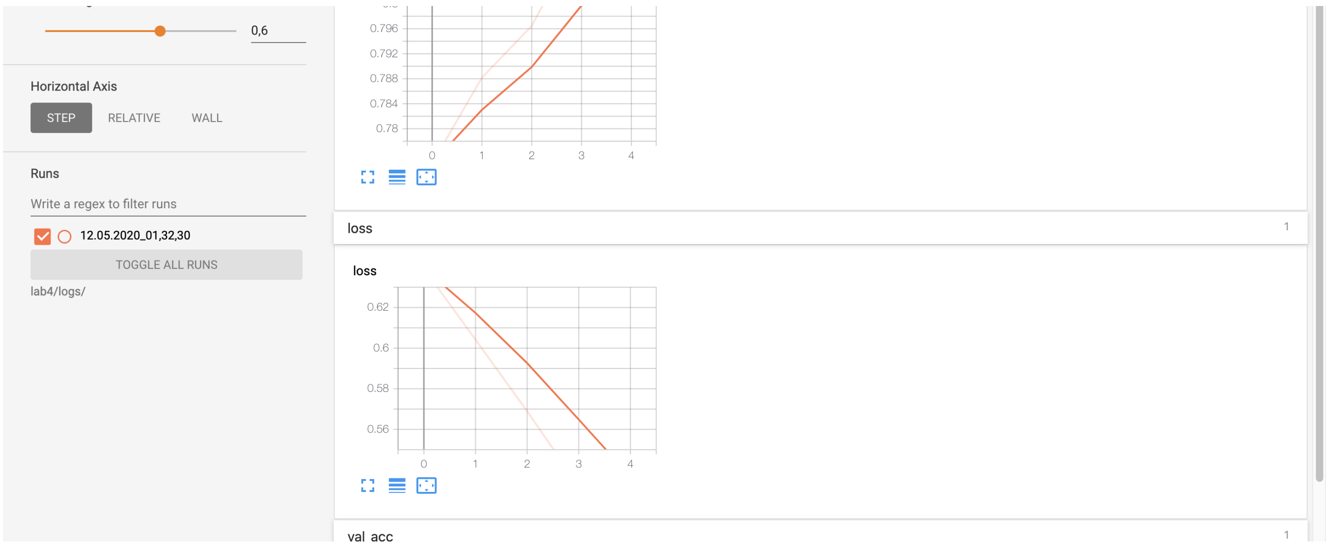
```
loss, accuracy = model.evaluate(test_images, test_labels)
```

```
10000/10000 [==============================] - 1s 146us/sample - loss: 0.8873 - acc: 0.7002
```

Метрики в Tensorboard



Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default



Контрольные вопросы

1. Как включить TensorBoard?

tensorboard --logdir [путь до папки с метриками]

2. Как сбросить граф?

tf.reset_default_graph() для 1 версии tf

tf.keras.backend.clear_session() для 2 версии tf

3. Зачем нужны коллекции?

Коллекция - это объект похожий на словарь, в котором мы храним элементы узлов графа.

4. Перечислите команды для добавления переменных в сводную статистику.

tensoboard_logs*dir = 'logs/{}'.format(datetime.datetime.now().strftime('%d.%m.%Y*%H,%M,%S'))

my_callbacks = [TensorBoard(log_dir=tensoboard_logs_dir),]

При обучении модели использовать my_callbakcs

model.fit(..., callbacks=my_callbacks)

Список литературы

[1] Google. Tensorflow. 2018. Apr. url - https://www.tensorflow.org/api_docs/python/tf/train/Saver.

[2] Google. TensorBoard. 2018. Apr. url - https://www.tensorflow.org/programmers_guide/summaries_and_- tensorboard.