Практическое задание 6
Вариант 2

Бабенко Никита, гр. 8383

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 вызвать функцию gen_data, которая возвращает два тензора:

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

Обратите внимание:

- Выборки не перемешаны, то есть наблюдения классов идут по порядку
- Классы характеризуются строковой меткой
- Выборка изначально не разбита на обучающую, контрольную и тестовую
- Скачивать необходимо оба файла. Подключать файл, который начинается с var (в нем и находится функция gen_data)

Ознакомиться с построением сверточных нейронных сетей на примере MNIST можно по ссылке

Классификация изображений с закрашенным или не закрашенным кругом
Файл 1
Файл 2

Решение

Разбиение данных на обучающие и тестовые, а также перемешивание – с помощью функции sklearn.model_selection.train_test_split.
Генерируется 1000 объектов.
Тензоры классов приводятся к категориальному виду.

Слои сверточной сети:

```
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu',
input_shape=(50, 50, 1)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu',
input_shape=(50, 50, 1)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
```

Функция активации - binary_crossentropy, оптимизатор – Adam
10 эпох, validation_split = 0.1, batch_size = 16

После обучения производится проверка на тестовых данных с помощью evaluate()

Листинг обучения:

```
Epoch 1/10
45/45 [==============================] - 1s 13ms/step - loss: 0.6867 -
accuracy: 0.5495 - val_loss: 0.6258 - val_accuracy: 0.6250
Epoch 2/10
45/45 [==============================] - 0s 9ms/step - loss: 0.6248 -
accuracy: 0.6195 - val_loss: 0.5180 - val_accuracy: 0.8250
Epoch 3/10
45/45 [==============================] - 0s 9ms/step - loss: 0.4626 -
accuracy: 0.8117 - val_loss: 0.3091 - val_accuracy: 0.9500
Epoch 4/10
45/45 [==============================] - 0s 9ms/step - loss: 0.2923 -
accuracy: 0.8971 - val_loss: 0.2137 - val_accuracy: 0.9500
Epoch 5/10
45/45 [==============================] - 0s 9ms/step - loss: 0.2656 -
accuracy: 0.8968 - val_loss: 0.1589 - val_accuracy: 0.9625
Epoch 6/10
45/45 [==============================] - 0s 9ms/step - loss: 0.1519 -
accuracy: 0.9628 - val_loss: 0.1274 - val_accuracy: 0.9625
Epoch 7/10
45/45 [==============================] - 0s 9ms/step - loss: 0.1108 -
accuracy: 0.9721 - val_loss: 0.1060 - val_accuracy: 0.9875
Epoch 8/10
45/45 [==============================] - 0s 9ms/step - loss: 0.0999 -
accuracy: 0.9733 - val_loss: 0.1030 - val_accuracy: 0.9625
Epoch 9/10
45/45 [==============================] - 0s 9ms/step - loss: 0.1046 -
accuracy: 0.9675 - val_loss: 0.0857 - val_accuracy: 0.9875
Epoch 10/10
45/45 [==============================] - 0s 9ms/step - loss: 0.0820 -
accuracy: 0.9803 - val_loss: 0.0809 - val_accuracy: 0.9875
```

Листинг evaluate

```
7/7 [==============================] - 0s 5ms/step - loss: 0.0766 -
accuracy: 0.9900
```

Точность на тестовой выборе – 99%, что является хорошим результатом. По точности на контрольных данных можно говорить об отсутствии переобучения, чему способствуют слои Dropout.