

Задание

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`)

Вариант 2

Генерация и подготовка данных:

```
data, labels = gen_data()
data, labels = sklearn.utils.shuffle(data, labels)
encoder = LabelEncoder()
encoder.fit(labels)
labels = encoder.transform(labels)
nums, width, height = data.shape
count = nums / 10
data_train = data[count:]
data_test = data[:count]
labels_train = labels[count:]
labels_test = labels[:count]
```

```
data_train = data_train.reshape(data_train.shape[0], width,
height, 1)
data_test = data_test.reshape(data_test.shape[0], width, height,
1)
```

Параметры сети:

```
batch_size = 10
num_epochs = 20
kernel_size = 3
pool_size = 2
conv_depth_1 = 32
conv_depth_2 = 64
drop_prob_1 = 0.25
drop_prob_2 = 0.5
hidden_size = 64
```

Архитектура сети:

```
model = Sequential()
model.add(Conv2D(conv_depth_1,      (kernel_size,      kernel_size),
padding='same',  activation='relu',  input_shape=(width,  height,
1)))
model.add(MaxPooling2D((pool_size, pool_size)))
model.add(Dropout(drop_prob_1))
model.add(Conv2D(conv_depth_2,      (kernel_size,      kernel_size),
padding='same',  activation='relu'))
model.add(MaxPooling2D((pool_size, pool_size)))
model.add(Conv2D(conv_depth_2,      (kernel_size,      kernel_size),
padding='same',  activation='relu'))
model.add(Dropout(drop_prob_2))
model.add(Flatten())
model.add(Dense(hidden_size, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

Результаты обучения приведены на рисунках 1 и 2.

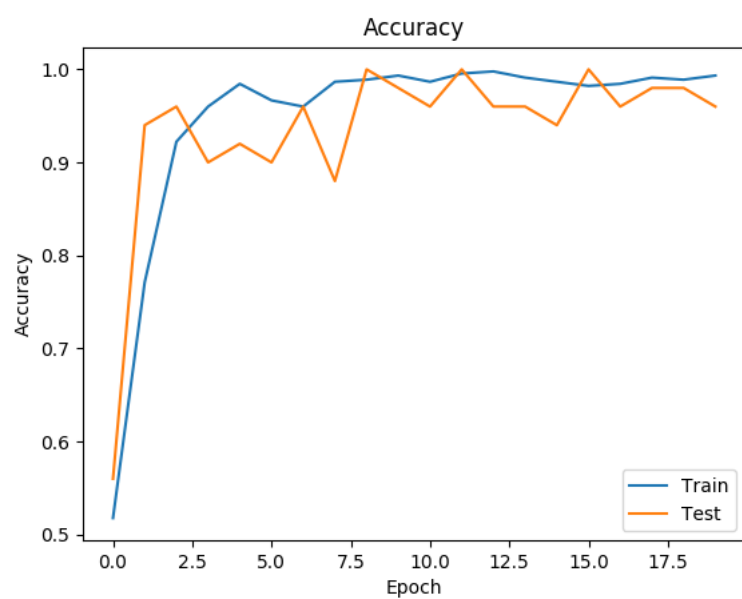


Рис. 1 – график точности

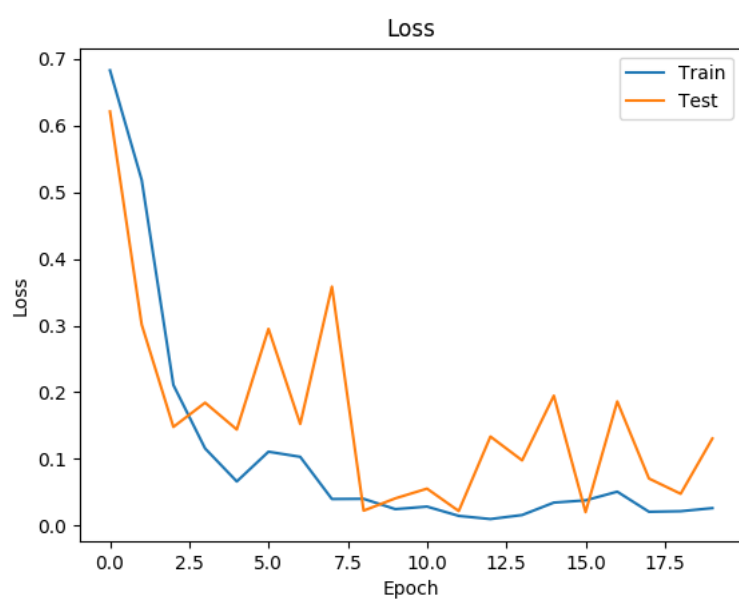


Рис. 2 – график ошибок

Точность на тестовых данных составила 0.95.