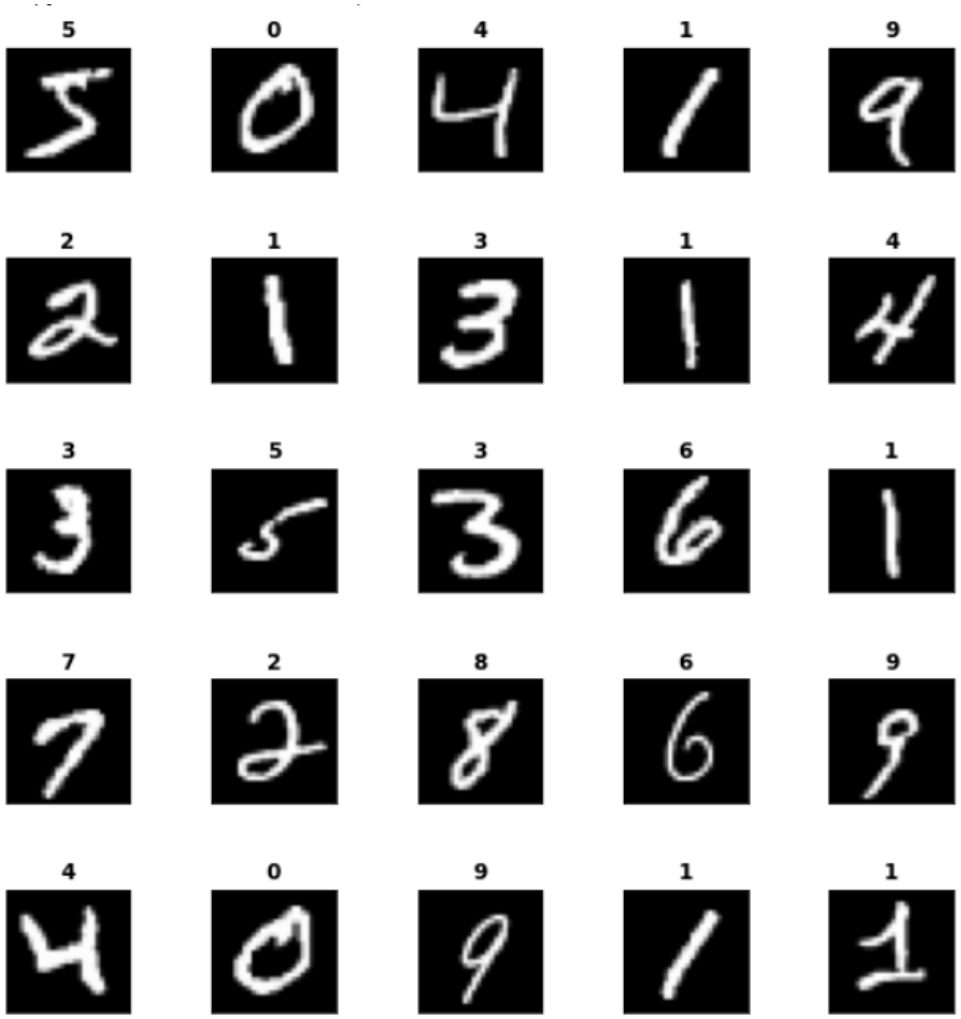
Задание.
Настроить предобученную нейронную сеть для классификации рукописных цифр. Набор данных - MNIST.В качестве исходной берется любая модель из табл. 1 в ЛК6_2021.

Описание выполненных действий
Для применения предобученной нейронной сети, нужно изменить размер массивов x_train, x_test. Поэтому x_train приведем к размеру (60000, 48, 48, 3), x_test (10000, 48, 48, 3). Фактически, таким образом мы увеличили каждой “изображение цифры” до размера 48*48 пикселей и добавили 3 цветовых канала.

```
#загрузка MNIST
print("number of classes: ", num_classes)
imagesTrain, labelsTrain, imagesTest, labelsTest = ReadBIN(n)
x_train, y_train, x_test, y_test, y_train_cat, y_test_cat = buf_x_y(num_classes)

x_train = x_train.reshape(-1, 28, 28, 1)
x_test = x_test.reshape(-1, 28, 28, 1)
OutputData25(x_train, y_train)
```

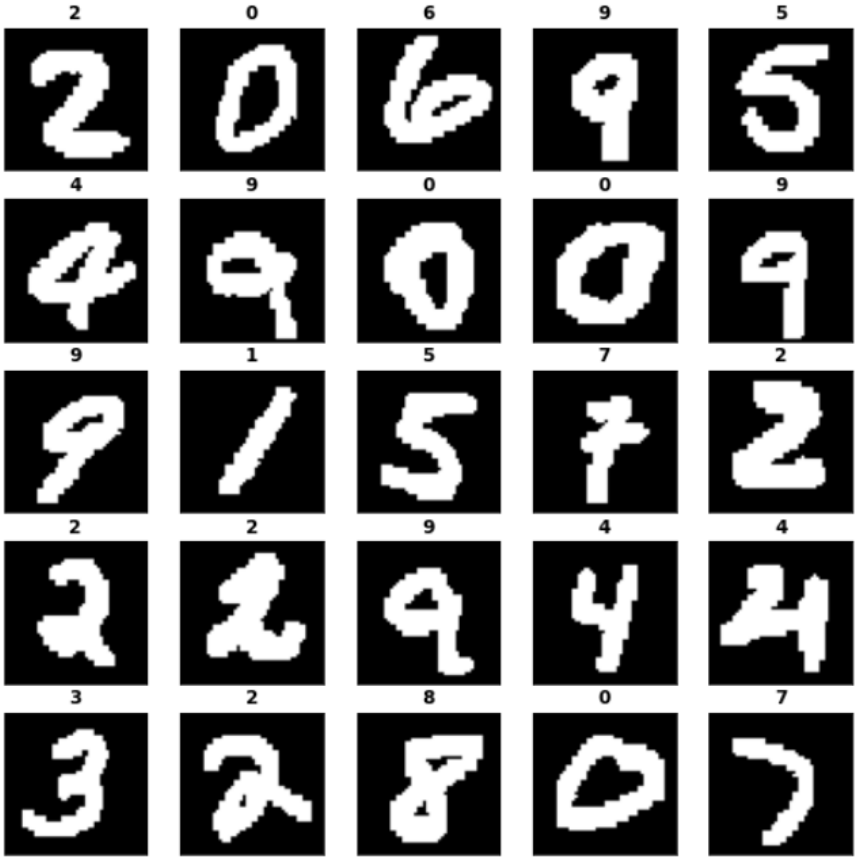



```
#3 канал
x_train = np.concatenate([x_train] * 3, axis=3)
x_test = np.concatenate([x_test] * 3, axis=3)

x_train /= 255
x_test /= 255
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
#новый размер изображения
IMG_SIZE = 48
x_train = np.array([img_to_array(array_to_img(im).resize((IMG_SIZE,IMG_SIZE),
resample=Image.BICUBIC)) for im in x_train])
x_test = np.array([img_to_array(array_to_img(im).resize((IMG_SIZE,IMG_SIZE),
resample=Image.BICUBIC)) for im in x_test])

print(x_train.shape)
print(x_test.shape)
```

```
OutputData48(x_train, y_train )
```



Загружаем предобученную модель без последних полносвязных слоев и слоя Flatten, расположенного перед ними с параметром include_top = False.
Включить в новую модель все слои, кроме двух последних полносвязных – это слой fc2 с размером выхода 4096 и классифицирующий слой predictions, сделаем включенные в модель слои необучаемыми, а затем добавить полносязный слой с размером выхода, равным, например, 512 и классифицирующий слой с размером выхода, равным числу классов
Добавляем новые слои, последний слой - классифицирующий слой с размером выхода = 10.
Обучаем модель, выводим графики потери и точности, выводим неверно классифицированные изображения, точность по классам.

Model.summary() полученной модели

Model: "model"

_____

Layer (type) Output Shape Param #

input_1 (InputLayer)	[(None, 48, 48, 3)]	0
block1_conv1 (Conv2D)	(None, 48, 48, 64)	1792
block1_conv2 (Conv2D)	(None, 48, 48, 64)	36928
block1_pool (MaxPooling2D)	(None, 24, 24, 64)	0
block2_conv1 (Conv2D)	(None, 24, 24, 128)	73856
block2_conv2 (Conv2D)	(None, 24, 24, 128)	147584
block2_pool (MaxPooling2D)	(None, 12, 12, 128)	0
block3_conv1 (Conv2D)	(None, 12, 12, 256)	295168
block3_conv2 (Conv2D)	(None, 12, 12, 256)	590080
block3_conv3 (Conv2D)	(None, 12, 12, 256)	590080
block3_pool (MaxPooling2D)	(None, 6, 6, 256)	0
block4_conv1 (Conv2D)	(None, 6, 6, 512)	1180160
block4_conv2 (Conv2D)	(None, 6, 6, 512)	2359808
block4_conv3 (Conv2D)	(None, 6, 6, 512)	2359808
block4_pool (MaxPooling2D)	(None, 3, 3, 512)	0
block5_conv1 (Conv2D)	(None, 3, 3, 512)	2359808
block5_conv2 (Conv2D)	(None, 3, 3, 512)	2359808
block5_conv3 (Conv2D)	(None, 3, 3, 512)	2359808
block5_pool (MaxPooling2D)	(None, 1, 1, 512)	0
flatten (Flatten)	(None, 512)	0
batch_normalization (BatchNo	(None, 512)	2048
dropout (Dropout)	(None, 512)	0

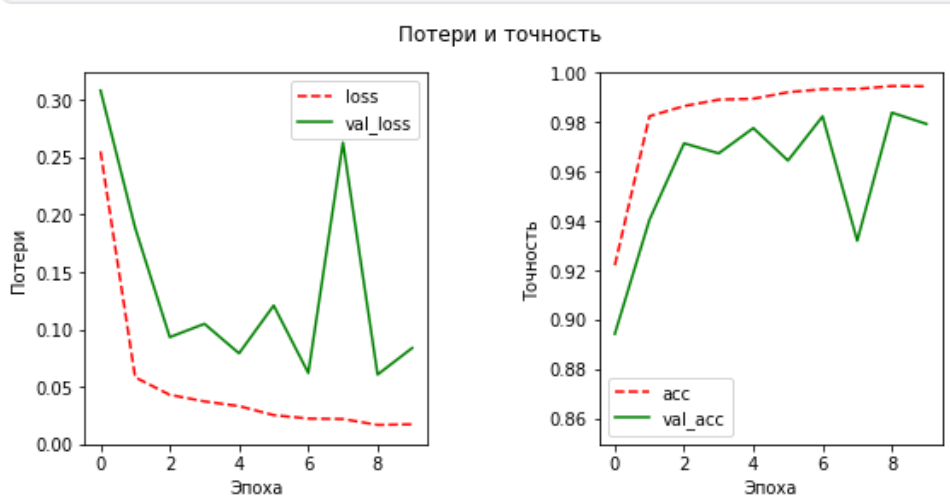
dense (Dense)	(None, 256)	131328
batch_normalization_1 (Batch	(None, 256)	1024
dropout_1 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 10)	1290

Total params: 14,883,274
Trainable params: 4,886,666
Non-trainable params: 9,996,608

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 48, 48, 3)]	0
block1_conv1 (Conv2D)	(None, 48, 48, 64)	1792
block1_conv2 (Conv2D)	(None, 48, 48, 64)	36928
block1_pool (MaxPooling2D)	(None, 24, 24, 64)	0
block2_conv1 (Conv2D)	(None, 24, 24, 128)	73856
block2_conv2 (Conv2D)	(None, 24, 24, 128)	147584
block2_pool (MaxPooling2D)	(None, 12, 12, 128)	0
block3_conv1 (Conv2D)	(None, 12, 12, 256)	295168
block3_conv2 (Conv2D)	(None, 12, 12, 256)	590080
block3_conv3 (Conv2D)	(None, 12, 12, 256)	590080
block3_pool (MaxPooling2D)	(None, 6, 6, 256)	0
block4_conv1 (Conv2D)	(None, 6, 6, 512)	1180160
block4_conv2 (Conv2D)	(None, 6, 6, 512)	2359808
block4_conv3 (Conv2D)	(None, 6, 6, 512)	2359808
block4_pool (MaxPooling2D)	(None, 3, 3, 512)	0
block5_conv1 (Conv2D)	(None, 3, 3, 512)	2359808
block5_conv2 (Conv2D)	(None, 3, 3, 512)	2359808
block5_conv3 (Conv2D)	(None, 3, 3, 512)	2359808
block5_pool (MaxPooling2D)	(None, 1, 1, 512)	0
flatten (Flatten)	(None, 512)	0
batch_normalization (BatchNo	(None, 512)	2048

```
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 256)               131328
_________________________________________________________________
batch_normalization_1 (Batch (None, 256)               1024
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                1290
=================================================================
Total params: 14,883,274
Trainable params: 4,886,666
Non-trainable params: 9,996,608
_________________________________________________________________
```

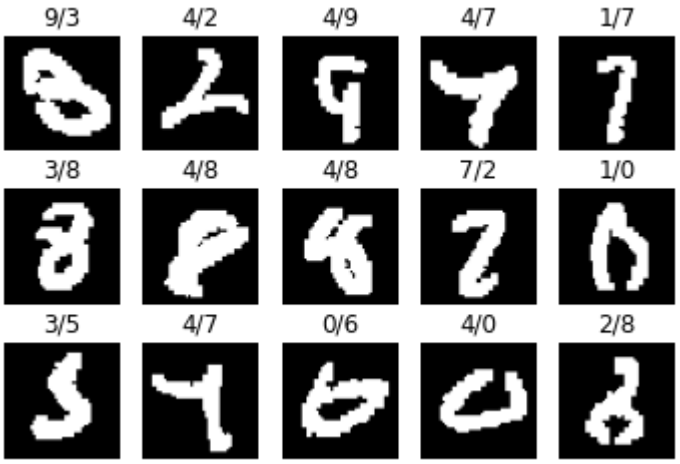
Графики обучения



Полученная точность классификации изображений обучающего и проверочного множеств MNIST.

Прогноз
Потери при тестировании: 0.0786
Точность при тестировании: 97.78000116348267%
Всего изображений в тестовой выборке: 10000
Число верно классифицированных изображений: 9778
Точность: 97.78%
Неверно классифицированно: 222
Индекс | Прогноз | Правильный класс
18 9 3
43 4 2

62	4	9
124	4	7
175	1	7
184	3	8
242	4	8
290	4	8
321	7	2
324	1	0
340	3	5
358	4	7
445	0	6
490	4	0



Точность по классам

0 : 0.9683673469387755
1 : 1.0
2 : 0.9864341085271318
3 : 0.995049504950495
4 : 0.9959266802443992
5 : 0.9708520179372198
6 : 0.9665970772442589
7 : 0.9795719844357976
8 : 0.9188911704312115
9 : 0.977205153617443