assignment6_1

July 19, 2021

[1]:
```
from keras import layers
from keras import models
```

[3]:
```
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
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'))
```

[4]:
```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 3, 3, 64)          36928
=================================================================
Total params: 55,744
Trainable params: 55,744
Non-trainable params: 0
_________________________________________________________________
```

[5]:
```
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))
```

[6]:
```
model.summary()
```

```
Model: "sequential_1"
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 3, 3, 64)          36928
_________________________________________________________________
flatten (Flatten)            (None, 576)               0
_________________________________________________________________
dense (Dense)                (None, 64)                36928
_________________________________________________________________
dense_1 (Dense)              (None, 10)                650
=================================================================
Total params: 93,322
Trainable params: 93,322
Non-trainable params: 0
_________________________________________________________________
```

[7]:
```
from keras.datasets import mnist
from keras.utils import to_categorical
```

[8]:
```
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```

[9]:
```
train_images = train_images.reshape((60000, 28, 28, 1))
train_images = train_images.astype('float32')/255

test_images = test_images.reshape((10000, 28, 28, 1))
test_images = test_images.astype('float32')/255

train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)
```

[10]:
```
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.fit(train_images, train_labels, epochs=5, batch_size=64)
```

```
Epoch 1/5
938/938 [==============================] - 14s 14ms/step - loss: 0.3933 -
accuracy: 0.8767
Epoch 2/5
938/938 [==============================] - 12s 13ms/step - loss: 0.0513 -
```

```
accuracy: 0.9829
Epoch 3/5
938/938 [==============================] - 12s 13ms/step - loss: 0.0347 -
accuracy: 0.9889
Epoch 4/5
938/938 [==============================] - 12s 13ms/step - loss: 0.0247 -
accuracy: 0.9921
Epoch 5/5
938/938 [==============================] - 12s 12ms/step - loss: 0.0208 -
accuracy: 0.9936
```

[10]:
```
<tensorflow.python.keras.callbacks.History at 0x7fb9a0581280>
```

[11]:
```
test_loss, test_acc = model.evaluate(test_images, test_labels)

test_acc
```

```
313/313 [==============================] - 1s 4ms/step - loss: 0.0232 -
accuracy: 0.9929
```

[11]:
```
0.992900013923645
```

[12]:
```
test_loss
```

[12]:
```
0.02317359298467636
```

[ ]:
```

```