An Introduction to Deep Learning With Python

[2.1] A first look at a Neural Network

Prof. Yuzo Iano

pgs: 27 - 30

In [1]:
```
import keras
from keras.datasets import mnist
```

```
Using TensorFlow backend.
```

In [2]:
```
(train_images, train_labels),(test_images, test_labels) = mnist.load_data()
```

In [3]:
```
# Let's look  at the Training Data
print('Train Shape', train_images.shape)
print('Train Label Length', len(train_labels))
train_labels
```

```
Train Shape (60000, 28, 28)
Train Label Length 60000
```

Out[3]:
```
array([5, 0, 4, ..., 5, 6, 8], dtype=uint8)
```

In [4]:
```
# The Test Data
print('Test Shape', test_images.shape)
print('Test Label Length', len(test_labels))
test_labels
```

```
Test Shape (10000, 28, 28)
Test Label Length 10000
```

Out[4]:
```
array([7, 2, 1, ..., 4, 5, 6], dtype=uint8)
```

The Network Architecture

In [5]:
```
from keras import models
from keras import layers

network = models.Sequential()
network.add(layers.Dense(512, activation = 'relu', input_shape = (28*28, )))
network.add(layers.Dense(10, activation = 'softmax'))

network.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 512)               401920
_________________________________________________________________
dense_2 (Dense)              (None, 10)                5130
=================================================================
Total params: 407,050
Trainable params: 407,050
Non-trainable params: 0
_________________________________________________________________
```

The Compilation Step

In [6]:
```
network.compile(optimizer = 'rmsprop',
                loss = 'categorical_crossentropy',
                metrics = ['accuracy'])
```

Preparing the image data

In [7]:
```
train_images = train_images.reshape((60000, 28 * 28))
train_images = train_images.astype('float32')/ 255

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

Preparing the labels

In [8]:
```
from keras.utils import to_categorical

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

Train the model

In [9]:
```
network.fit(train_images, train_labels, epochs = 5, batch_size = 128)
```

```
Epoch 1/5
60000/60000 [==============================] - 9s 150us/step - loss: 0.2560 - acc: 0.
9260
Epoch 2/5
60000/60000 [==============================] - 5s 90us/step - loss: 0.1037 - acc: 0.9
692
Epoch 3/5
60000/60000 [==============================] - 5s 86us/step - loss: 0.0680 - acc: 0.9
797
Epoch 4/5
60000/60000 [==============================] - 5s 88us/step - loss: 0.0498 - acc: 0.9
848
Epoch 5/5
60000/60000 [==============================] - 5s 86us/step - loss: 0.0372 - acc: 0.9
891
```

Out[9]:
```
<keras.callbacks.History at 0x289ca49fa20>
```

In [10]:
```
test_loss, test_acc = network.evaluate(test_images, test_labels)
print('TEST ACC: ', test_acc)
```

```
10000/10000 [==============================] - 1s 56us/step
TEST ACC:  0.98
```


Pablo Minango

- pablodavid218@gmail.com