File Edit View Insert Cell Kernel Navigate Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

1 Listing 5.1. Instantiating a small convnet

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

model = models.Sequential()#模型的建立
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))#3*3 * 32的局部特徵多募(
model.add(layers.MaxPooling2D((2, 2)))#降維 砍一半
model.add(layers.Conv2D(64, (3, 3), activation='relu')
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
#為四個Layers
```

2 Listing 5.2. Adding a classifier on top of the convnet

In [2]:
```
model.add(layers.Flatten())#壓平 將3D編為1D
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))#多個 機率 因為分析10個數字
```

In [3]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        320

 max_pooling2d (MaxPooling2D  (None, 13, 13, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 11, 11, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 5, 5, 64)         0
 2D)

 conv2d_2 (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
_________________________________________________________________
```

3 Listing 5.3. Training the convnet on MNIST images

In [5]:
```
from keras.datasets import mnist
from keras.utils import to_categorical

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
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#0-255 每一格的值 正規化使數值界在0~1之間

train_labels = to_categorical(train_labels)#將其類別化
test_labels = to_categorical(test_labels)

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 [==============================] - 51s 52ms/step - loss: 0.1859 - accuracy: 0.9415
Epoch 2/5
938/938 [==============================] - 48s 51ms/step - loss: 0.0470 - accuracy: 0.9850
Epoch 3/5
```

```
938/938 [==============================] - 45s 48ms/step - loss: 0.0324 - accuracy: 0.9899
Epoch 4/5
938/938 [==============================] - 48s 51ms/step - loss: 0.0242 - accuracy: 0.9925
Epoch 5/5
938/938 [==============================] - 41s 44ms/step - loss: 0.0189 - accuracy: 0.9940

<keras.callbacks.History at 0x243f8872b50>
```

In [6]:

```
test_loss, test_acc = model.evaluate(test_images, test_labels)
test_acc
```

```
313/313 [==============================] - 3s 8ms/step - loss: 0.0265 - accuracy: 0.9923

0.9922999739646912
```

In [8]:

```
from keras import layers
from keras import models

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'))
model.add(layers.Flatten())#3D->1D
model.add(layers.Dense(64, activation='relu'))#64dim
model.add(layers.Dense(10, activation='softmax'))#10值 機率
```

In [9]:

```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 26, 26, 32)        320

 max_pooling2d_4 (MaxPooling  (None, 13, 13, 32)       0
 2D)

 conv2d_7 (Conv2D)           (None, 11, 11, 64)        18496

 max_pooling2d_5 (MaxPooling  (None, 5, 5, 64)         0
 2D)

 conv2d_8 (Conv2D)           (None, 3, 3, 64)          36928

 flatten_2 (Flatten)         (None, 576)               0

 dense_4 (Dense)             (None, 64)                36928

 dense_5 (Dense)             (None, 10)                650

=================================================================
Total params: 93,322
Trainable params: 93,322
Non-trainable params: 0
_________________________________________________________________
```