

10901-05339機器學習導論

開始於	2020年 11月 24日(二) 09:10
作答狀態	已經完成
完成於	2020年 11月 24日(二) 09:24
所用時間	13 分鐘 38 秒
分數	得分 8.00 /配分10.00(80%)

試題1
答對
得分 1.00/配
分1.00

在CNN中的池化層(Pooling) 可以進行降階處理，若選擇Max Pooling，則下面特徵圖中的2x2灰色區域經Max Pooling後的輸出為何？

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1

→

?	

單選：

- ☒ a. 2
- ☐ b. 3
- ☐ c. 1
- ☐ d. 4

你答對了!
正確答案：2

試題2

答對

得分 1.00/配
分1.00

關於下面的圖片內容，其中指出CNN能夠辨別大小不同的相同物件的原因是基於其架構中的什麼機制？

Why CNN for Image

- Subsampling the pixels will not change the object



We can subsample the pixels to make image smaller

➡ Less parameters for the network to process the image

單選：

- ☐ a. convolution
- ☒ b. Max Pooling
- ☐ c. Softmax
- ☐ d. Flatten

你答對了!

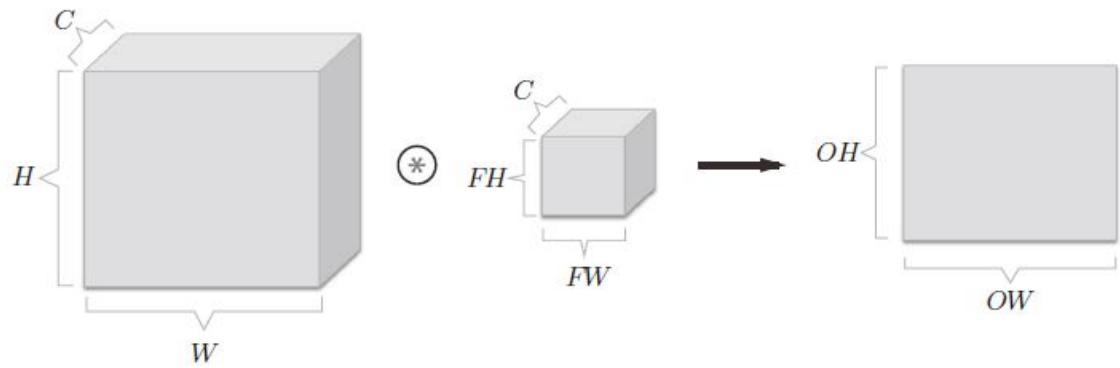
正確答案：Max Pooling

試題3

答錯

得分 0.00/配
分1.00

下圖說明一個(通道深度,長,寬)分別為(C, H, W)的資料與(通道深度,長,寬)為(C, FH, FW)的卷積核進行卷積運算的結果。試問卷積運算得到的特徵圖之深度應為何?



單選：

- ☐ a. 特徵圖之深度應為 $2C$
- ☒ b. 特徵圖之深度應為 C
- ☐ c. 特徵圖之深度應為 1
- ☐ d. 特徵圖之深度應為 3

你答錯了!

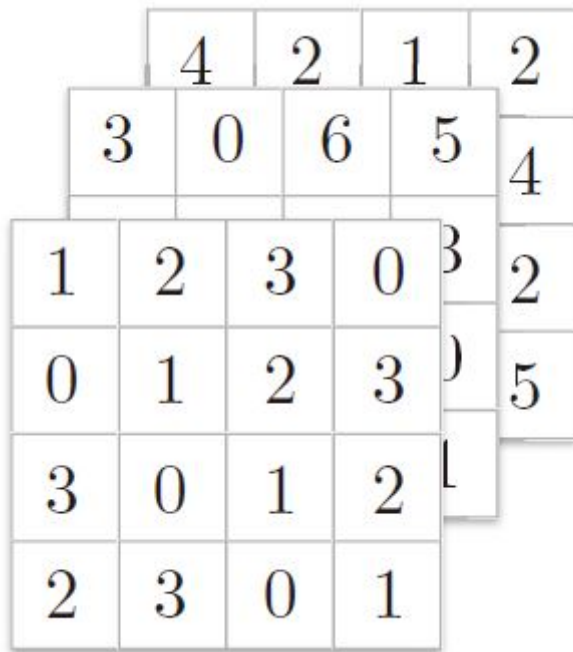
正確答案：特徵圖之深度應為 1

試題4

答對

得分 1.00/配
分1.00

下面的特徵圖為 $4 \times 4 \times 3$ 的彩色影像資料，也就是深度(depth)為3；若我們在CNN中選擇使用 3×3 的卷積核來與這個特徵圖進行卷積運算，那麼一個卷積核的參數量是多少(不考慮偏移量的情況)?



單選：

- ☐ a. $3 \times 3 \times 1$ ，也就是有9個參數
- ☒ b. $3 \times 3 \times 3$ ，也就是有27個參數
- ☐ c. $3 \times 3 \times 2$ ，也就是有18個參數
- ☐ d. $3 \times 3 + 1$ ，也就是有10個參數

你答對了!

正確答案： $3 \times 3 \times 3$ ，也就是有27個參數**試題5**

答錯

得分 0.00/配
分1.00

關於下列敘述何者為誤?

單選：

- ☐ a. $y = f(t)$ 是一般常見的函式，代表給定一個 t 值，丟到 f 函式中會回傳一個值給 y 。
- ☐ b. 假設有一個函式 $f(t)$ ， t 的可能範圍是 $\{0, 1, 2\}$ ，且 $f(t=0) = 10$ ； $f(t=1) = 20$ ； $f(t=2) = 7$ ；則 $y^* = \arg \max f(t) = 20$
- ☒ c. $y = \max f(t)$ 代表： y 是 $f(t)$ 函式所有的值中最大的output。
- ☐ d. $y = \arg \max f(t)$ 代表： y 是 $f(t)$ 函式中，會產生最大output的那個參數 t 。

你答錯了!

正確答案：假設有一個函式 $f(t)$ ， t 的可能範圍是 $\{0, 1, 2\}$ ，且 $f(t=0) = 10$ ； $f(t=1) = 20$ ； $f(t=2) = 7$ ；則 $y^* = \arg \max f(t) = 20$

試題6

答對

得分 1.00/配
分1.00

關於下面程式的內容，哪一個答案不正確？

```
from keras.datasets import mnist
(X_train, y_train), (X_test, y_test) = mnist.load_data()
#reshape data to fit model
X_train = X_train.reshape(60000,28,28,1)
X_test = X_test.reshape(10000,28,28,1)
from keras.utils import to_categorical
#one-hot encode target column
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
# Building the model
from keras.models import Sequential
from keras.layers import Dense, Conv2D, Flatten
#create model
model = Sequential()
#add model layers
model.add(Conv2D(64, kernel_size=3, activation='relu', input_shape=(28,28,1)))
model.add(Conv2D(32, kernel_size=3, activation='relu'))
model.add(Flatten())
model.add(Dense(10, activation='softmax'))
#compile model using accuracy to measure model performance
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
#train the model
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=3)
```

單選：

- ☒ a. 數字的資料庫共70000筆，每筆影像資料為28x28x1，選擇其中10000筆為訓練資料
- ☐ b. 這是一個訓練0-9等十個數字辨識的程式
- ☐ c. activation='relu' 代表選擇relu為活化函數
- ☐ d. 迭代次數只有三次

你答對了！

正確答案：數字的資料庫共70000筆，每筆影像資料為28x28x1，選擇其中10000筆為訓練資料

試題7

答對

得分 1.00/配
分1.00

下列敘述何者為誤？

單選：

- ☐ a. CNN的convolution layer中，要訓練的網路參數即是filter matrix的係數
- ☒ b. CNN 的convolution layer是屬於fully-connected 的神經網路架構
- ☐ c. CNN的convolution layer中，經過filter卷積運算得到的輸出影像也稱為feature map
- ☐ d. CNN的Max pooling主要目的是為了將影像降階為更低解析度的小影像

你答對了！

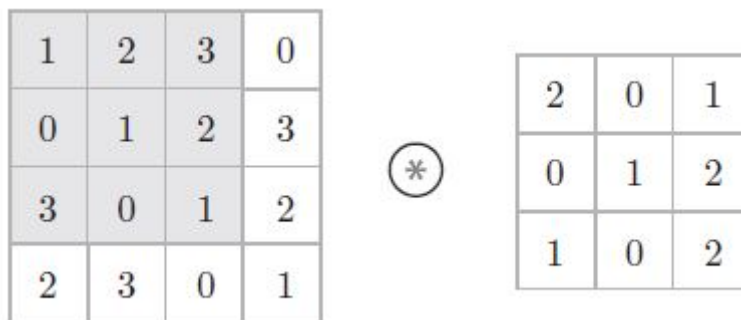
正確答案：CNN 的convolution layer是屬於fully-connected 的神經網路架構

試題8

答對

得分 1.00/配
分1.00

若有一張8x8特徵圖與3x3卷積核進行卷積運算，如下圖；則灰色區域與卷積核運算的結果為何？



單選：

- ☐ a. 16
- ☐ b. 6
- ☐ c. 7
- ☒ d. 15
- ☐ e. 14

你答對了！

正確答案：15

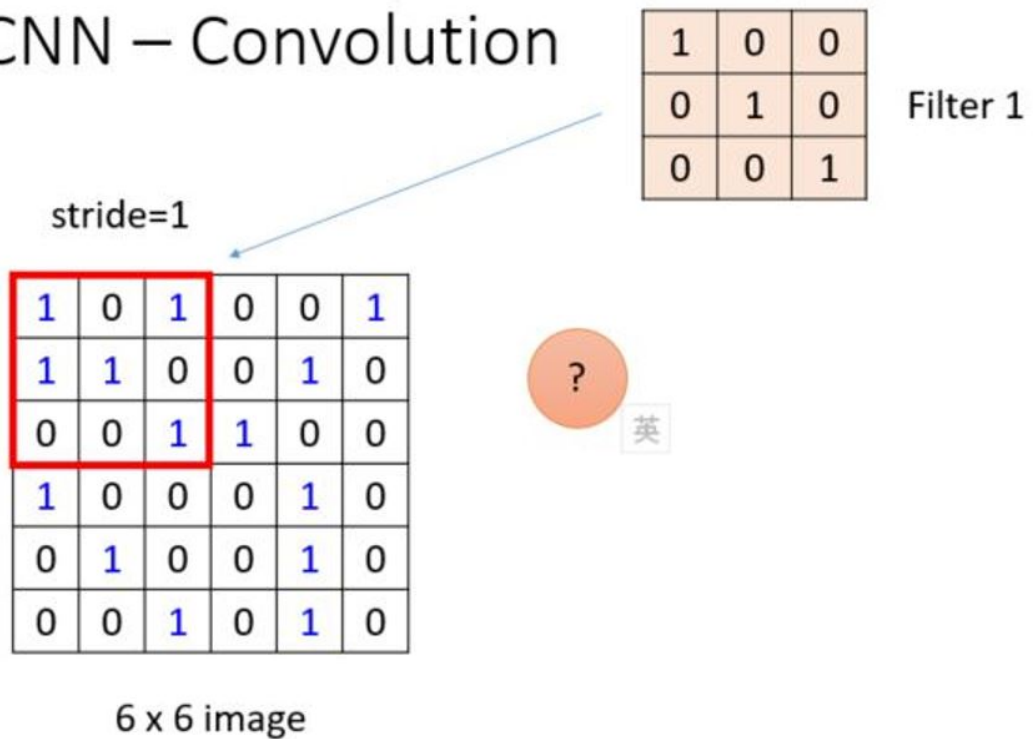
試題9

答對

得分 1.00/配
分1.00

下面圖片中的CNN convolution之運算結果為何？

CNN – Convolution



單選：

- ☐ a. 1
- ☒ b. 3
- ☐ c. 7
- ☐ d. 5

你答對了!

正確答案：3

試題10

答對

得分 1.00/配
分1.00

處理mnist資料時，每一張影像都是28x28的灰階影像(深度為1)，若我們在進行第一層的卷積運算時，選擇卷積核的size 為3x3，且卷積核的數目為64，如下面程式碼所示。則經過第一層卷積層運算後，得到的特徵圖資料應為何？

```
#create model
model = Sequential()
#add model layers
model.add(Conv2D(64, kernel_size=3, activation='relu', input_shape=(28,28,1)))
```

單選：

- ☒ a. 產生特徵圖之長,寬,深度應為(26, 26, 64)
- ☐ b. 產生特徵圖之長,寬,深度應為(13, 13, 25)
- ☐ c. 產生特徵圖之長,寬,深度應為(26, 26, 1)
- ☐ d. 產生特徵圖之長,寬,深度應為(26, 26, 25)

你答對了!

正確答案：產生特徵圖之長,寬,深度應為(26, 26, 64)

Return to: 11月 23日 - 11月 2... ➡