

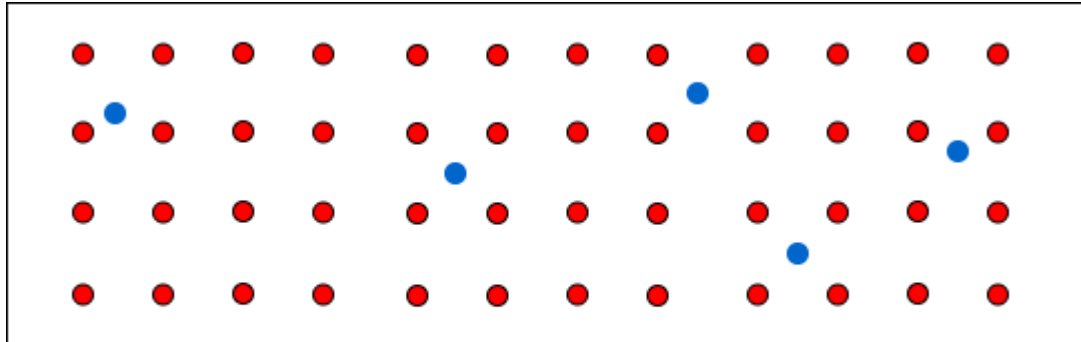
# AI 無線通訊系統實驗

## Lab 2 Localization with AI method

Author: 蕭安紘 助教

### 實驗目的

建立一個室內定位之神經網路，藉此判斷使用者位於哪個位置



● Reference Point      ● Test Point

### 實驗介紹

#### Fingerprinting Localization Method

##### Off-line Phase

在預先設定已知位置之參考點(Reference Point)上蒐集無線訊號之特徵，建立指紋資料庫 (Fingerprinting Database)

##### On-line Phase

將預定位之使用者的無線訊號特徵與資料庫進行比對以求出使用者的位置

#### Hidden Layer(1D-CNN):

```
Object2 = tf.keras.layers.Conv1D(filter, stride, padding='same',  
activation='activation_function')(Object1)
```

Object2: 物件名稱

**filter**: 輸出維度

**stride**: 定義卷積計算時每次所要跳躍的長度

**activation**: 挑選想使用的 **activation function**，這邊也可以不設定激活函數

Object1: 輸入之物件

範例:

```
c1 = tf.keras.layers.Conv1D(32, 5, padding='same')(inputs)
```

#### Maxpooling

保留主要特徵同時減少參數，防止過度擬合(Over-fitting)，提高模型的通用性

範例:

```
c1 = tf.keras.layers.MaxPool1D(padding='same')(c1)
```

#### BatchNormalization

透過 normalization，將偏離的分布拉回到標準化的分布，使激活函數的輸入值落在激活函數比較敏感的區域(e.g. [0,1])，進而使梯度變大，加速學習的收斂速度，避免梯度消失的問題 (Vanishing gradient problem)

範例:

```
c1 = tf.keras.layers.BatchNormalization()(c1)
```

### Flatten

將資料由高維度攤平成一維，常用於將 CNN 所提取出之特徵轉換維度，以利於特徵作為全連接層的輸入

範例:

```
f1 = tf.keras.layers.Flatten()(c4)
```

### Dropout

如果模型參數過多，而訓練資料量過少，可透過將一些隱藏節點設為 0，防止過度擬合的狀況發生

範例:

```
l1 = tf.keras.layers.SpatialDropout1D(rate)(c1)
```

**rate**: 選擇要忽略(隱藏節點設為 0)的比例

### 評估模型

利用套件對量化模型對測試資料的表現

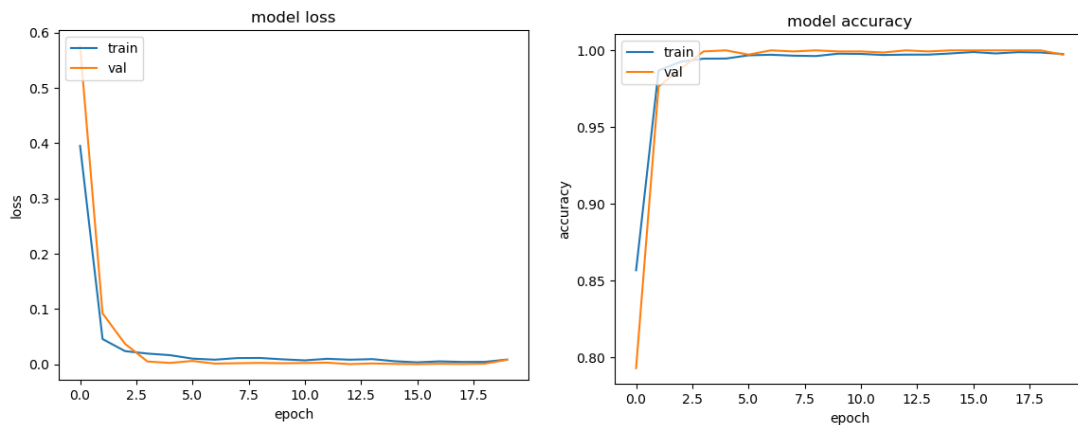
範例:

```
model.evaluate(test_dataset, test_label, batch_size = 256)
```

## 實驗步驟

1. 讀取資料，其中包含訓練資料與測試資料，請將資料路徑設為您所使用的電腦中的路徑  
    **訓練資料**: 資料包含 4 位受測者在 7 個位置時收集之 CSI 資料，每種資料個 600 筆，總計 14400 筆資料  
    **測試資料**: 同上，但於不同時間收集
2. 資料前處理(Reshape, One-hot Encoder)
3. 建立包含 Input Layer、4 層 1D-CNN 網路(filter: 32, 64, 128, 128)並選用 `tf.nn.leaky_relu` 作為激活函數
4. 在利用 1D-CNN 提取特徵後，利用 `tf.keras.layers.Flatten()` 將 tensor 的維度轉換成一維
5. 建立兩層 DNN 網路(Node: 256, 64)與 Output Layer(Node: 10)進行特徵映射 (Feature Mapping)
6. 編譯(compile)網路架構，loss 選擇 "categorical\_crossentropy"，optimizer 選擇 "Adam"，並利用 `model.summary()` 在 console 裡印出如 Lab1 的圖
7. 進行神經網路模型的訓練 `model.fit()`

8. 畫出訓練過程中的精準度(Accuracy)與損失函數(Loss Function)



9. 利用 `model.evaluate()` 測試資料之印出準確度

```
14400/14400 [=====] - 0s 14us/sample - loss: 1.3142 - accuracy: 0.8132
```

## 基礎題

完成以上實驗步驟並找助教 Demo

## 加分題(先完成基礎題才可以 demo)

1. 將訓練資料進行 Shuffle(此動作為將資料順序打亂，避免相同分類的資料連續訓練，導致在一個 batch 裡過度擬合)

Hint:

```
random.shuffle()
```

2. 在每層 CNN 後面加上 Maxpooling、BatchNormalization 和 Dropout

Hint:

講義實驗介紹處有提示