

Lab 2 結報

姓名：郭朝恩 學號：109611035

1. 請敘述在課堂上實作之神經網路的架構

```
input_layer = tf.keras.layers.Input(shape=(Ns, Nt*Nr))

c1 = tf.keras.layers.Conv1D(32, 5, padding='same', activation=tf.nn.leaky_relu)(input_layer)
c2 = tf.keras.layers.Conv1D(64, 5, padding='same', activation=tf.nn.leaky_relu)(c1)
c3 = tf.keras.layers.Conv1D(128, 5, padding='same', activation=tf.nn.leaky_relu)(c2)
c4 = tf.keras.layers.Conv1D(128, 5, padding='same', activation=tf.nn.leaky_relu)(c3)

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

d1 = tf.keras.layers.Dense(256, activation='relu')(f1)
d2 = tf.keras.layers.Dense(64, activation='relu')(d1)

output = tf.keras.layers.Dense(6, activation='softmax')(d2)
```

網路架構包含 input layer、4 層 1D-CNN、1 層 flatten、2 層 DNN、以及 output layer。

Input layer 以三維 tensor 作為輸入，其中第一維是 batch size（可以變化），第二維 Ns 是 sensors 的數量，第三維 Nt*Nr 是時間樣本數 Nt 和 range bins 的數量 Nr。

該網路由 4 層 1D-CNN、2 層 fully connected DNN 和 softmax 輸出層組成。4 個 CNN 分別有 32、64、128、128 個 filter，每個 size 為 5，padding 設置為 'same' 以確保輸出大小與輸入大小相同。使用的 activation function 是 leaky ReLU。

然後將最後一層 CNN 的輸出 flatten 為一維。接著通過兩層 DNN，分別有 256、64 個 neuron，activation function 使用 ReLU。

最後，輸出層由 6 個 neuron 組成，代表 6 個 class 的機率分布，並使用 softmax 作為 activation function。

2. 請敘述 Maxpooling、Dropout 與 Batch Normalization 的作用

在加分題中，題目指定在每層 CNN 後面加上 Maxpooling、Dropout 和 Batch Normalization，可以提高性能並減少過度擬合。

MaxPooling 會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。

透過 normalization，將偏離的分布拉回到標準化的分布，使 activation function 的輸入值落在 activation function 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)

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

3. 心得

這次實驗也蠻簡單的，加入 CNN，還有防止 **over-fitting** 的方法，也讓我們比較 **data** 經過 **shuffle** 後，順序才不會影響訓練。不過介紹室內定位的部分被快速帶過，希望之後增加講解通訊的內容。