

實驗目的：

通道估測，我們有一個 Rayleigh fading channel，其 fading speed (f) = 0.01, $\sigma_b^2 = 0.00195557$ 的通道數值(可以用 matlab 的 import data，選擇.txt 檔案輸入資料)，兩行數值分別是實部 (Real part) 與虛部 (Imaginary part)。

通道觀測值的模型是：

$$y[T] = h[T] + n_0$$

$$\text{Where } h[T] = f \times h[T - 1] + \sigma_b \times n, n \sim N(0,1)$$

資料共有 10000 筆，拆分為 7000 訓練、3000 測試。拆完後以 sliding window 的方式切分輸入為過去五個時間點的資料、輸出標籤為下一個時間點的資料。

實驗介紹與實驗步驟：

1. 讀取資料
2. 資料前處理
3. 建立 Dataset 格式
4. 建立 LSTM Model 與 Loss
5. 進行神經網路模型的訓練 `model.fit()`
6. 畫出預測圖

(僅需要做步驟 4,5 建立模型並訓練的部分)

How to Build a LSTM Model

一個完整的網路包含輸入(Input Layer)、輸出(Output Layer)、隱藏層(Hidden Layer):

Input Layer:

```
tf.keras.layers.InputLayer(input_shape=(5, 2))
```

Output Layer:

```
tf.keras.layers.Dense(2),
```

Hidden Layer(LSTM):

```
tf.keras.layers.LSTM(units)
```

Compile Model

- 將以上你所建立的 hidden layer 串聯起來，並命名為 model

```
model = tf.keras.models.Sequential([  
    ## input layer,  
    ## hidden layer,  
    ## ...  
    ## output layer,  
])
```

- 設定 model 需用到的 Loss Function 與 optimizer

- `model.compile(loss= None, optimizer = None, metrics=['accuracy'])`
- loss: 選擇使用的 loss function，須為字串，如"MSE"

- **optimizer**: 選擇使用的 **optimizer**，須為字串，如"Adam"
- **metrics**: 在訓練與測試時的模型評估標準，若輸入 **metrics=['accuracy']**，會自動尋種 **binaryAccuracy** 或 **CategoricalAccuracy** 等等函式，也可使用 'mae' (mean absolute error) or 'mse' (mean square error) 等等
- **history = model.fit()**: 訓練模型，須給定訓練資料, **epoch**, **batch_size**, 驗證資料
 - **history** 會在完成的 **epoch** 記錄 **training loss & metrics value**，有需要可調用查看
- **model.summary()**: 顯示目前所建立之神經網路的層數與參數

[tf.keras.Model](#) | [TensorFlow v2.16.1](#)

使用之參數：

Train_data, test_data: 訓練與測試的輸入，**size=(6996,5,2), (2999,5,2)**

train_labels, test_labels: 訓練與測試的輸出標籤，**size=(6996,2), (2999,2)**

epoch: 需要訓練的次數(可自行調整大小)

batch: 每次訓練的 **sample** 數，暫定 32(可自行調整大小)

learning_rate: 學習率(根據 **optimizer** 決定，暫定預設值)

輸出結果範例：

