

## Week1

### 1. 模型架構

```
model = keras.Sequential([
    keras.layers.Dense(64, activation='relu', input_shape=(8,)),
    keras.layers.Dropout(0.3),
    keras.layers.Dense(32, activation='relu'),
    keras.layers.Dropout(0.3),
    keras.layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['mse', 'accuracy'])
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	576
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 1)	33

模型有兩個隱藏層並都加入 Dropout

第一個隱藏層是 64 個神經元，使用 ReLU 激活函數。

第二個隱藏層是 32 個神經元，一樣使用 ReLU 激活函數。

最後輸出層使用 Sigmoid 用於二元分類。

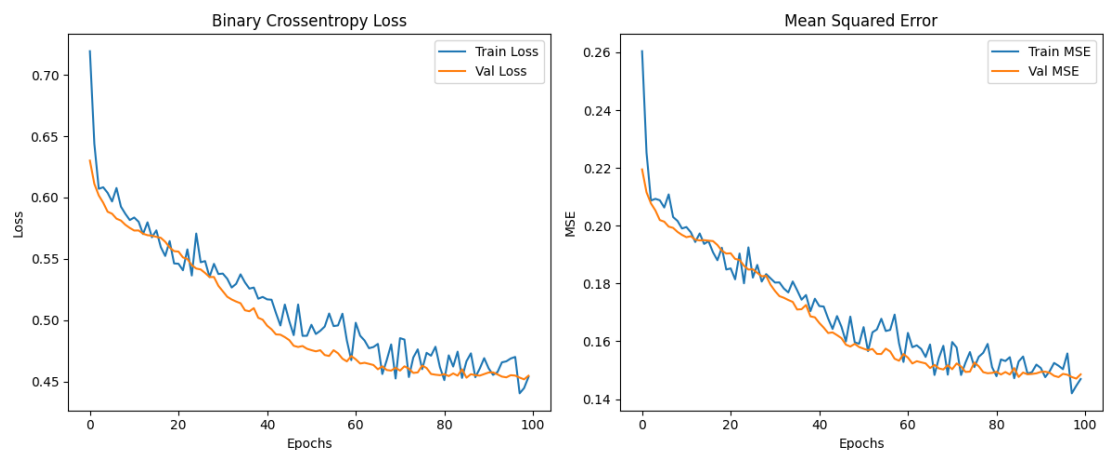
模型編譯使用 adam 優化器，損失函數使用 binary\_crossentropy

### 2. 訓練參數

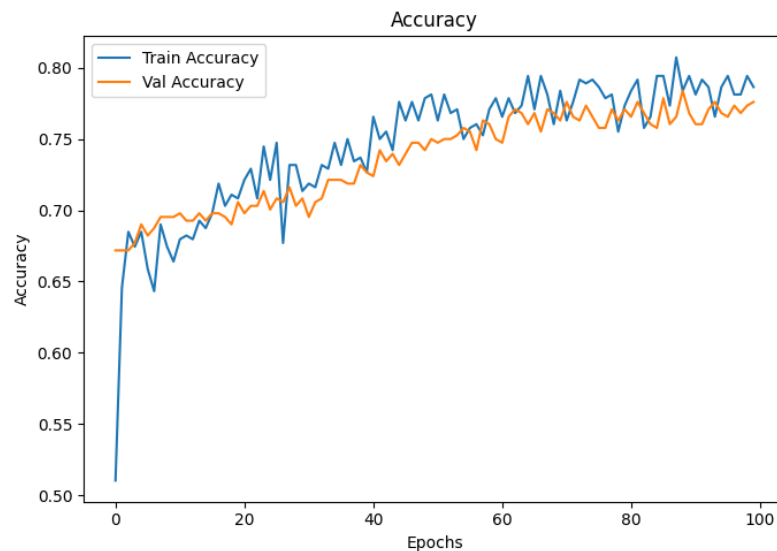
```
history = model.fit(trainin, traintgt, epochs=100, batch_size=10, verbose=1, validation_data=(testin, testtgt))
```

epochs=100, batch\_size=10

### 3. 訓練成果跟測試結果(loss, mse)



### 4. Accuracy and result



```
loss, mse, accuracy = model.evaluate(testin, testtgt, verbose=0)

print(f"Loss (Binary Crossentropy): {loss}")
print(f"Accuracy: {accuracy}")
print(f"MSE: {mse}")
```

71 ✓ 0.1s

• Loss (Binary Crossentropy): 0.4546477496623993  
Accuracy: 0.7760416865348816  
MSE: 0.14855331182479858

### 5. 心得：

這個作業我們在之前的課程有做過，但是當初是用老師的範例程式碼，這次是自己設計模型，在做的時候對於 loss function 要用 mse 還是 binary

**crossentropy** 考慮了很久，後來還是用 **binary crossentropy**。訓練曲線看起來波動不小，我們有嘗試使用 **L2** 正則化，雖然會變比較平滑但是準確率卻下降不少，並且目前的曲線我們認為還可以接受最後就沒有使用。雖然 **loss** 跟 **accuracy** 都還不是很理想，但是我們嘗試了許多方式都無法提高表現，或許是資料量太少的關係。

6. 組員名單：

11160801 陳禹豪 5 分，原因：進行實驗與模型設計

11360821 陳鈺安 5 分，原因：積極參與討論