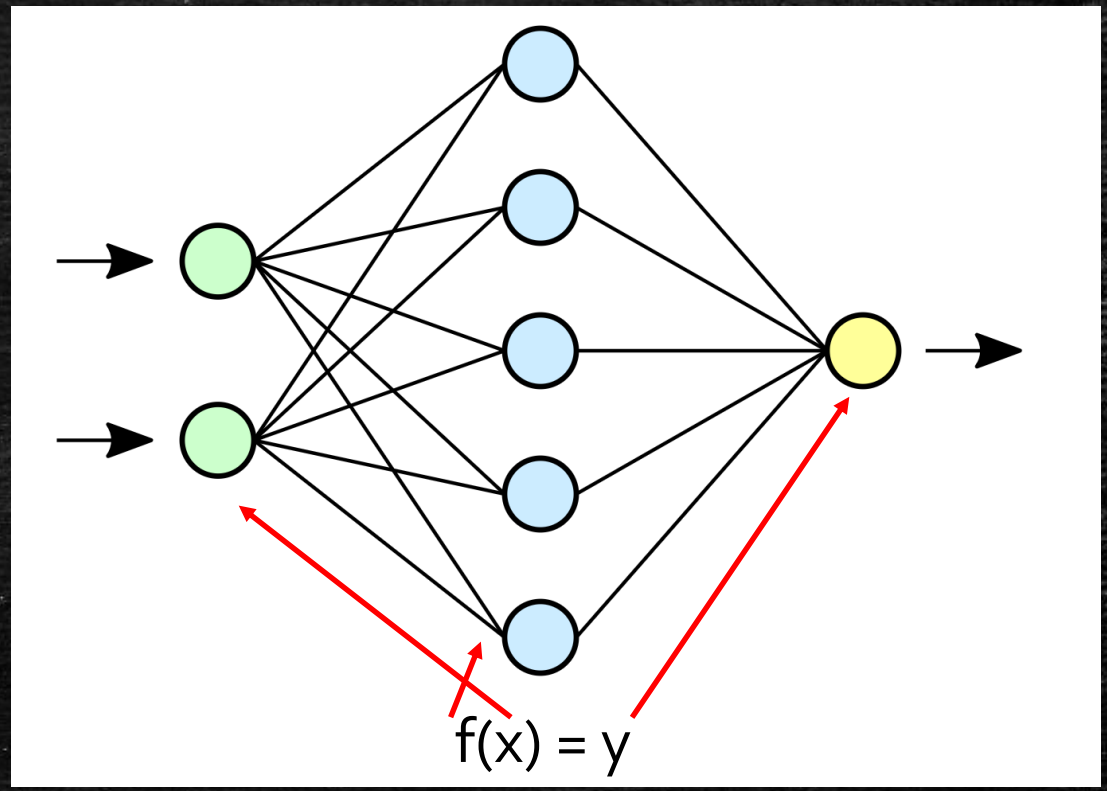
2024 鐵人賽 – 我數學就爛要怎麼來學 DNN 模型安全
Day 04 – DNN 模型基本概念

大綱

- 類神經網路
  - 結構
  - 神經元
  - 計算方式
- 結論

類神經網路結構

- 機器學習就是為了求一個 f(x) = y , x 是輸入，y 是預測結果
- 而類神經網路的結構就可以做這樣子的對應，分為
  - 輸入層
  - 輸出層
  - 隱藏層：輸入層和輸出層之間眾多神經元和鏈結組成的各個層面



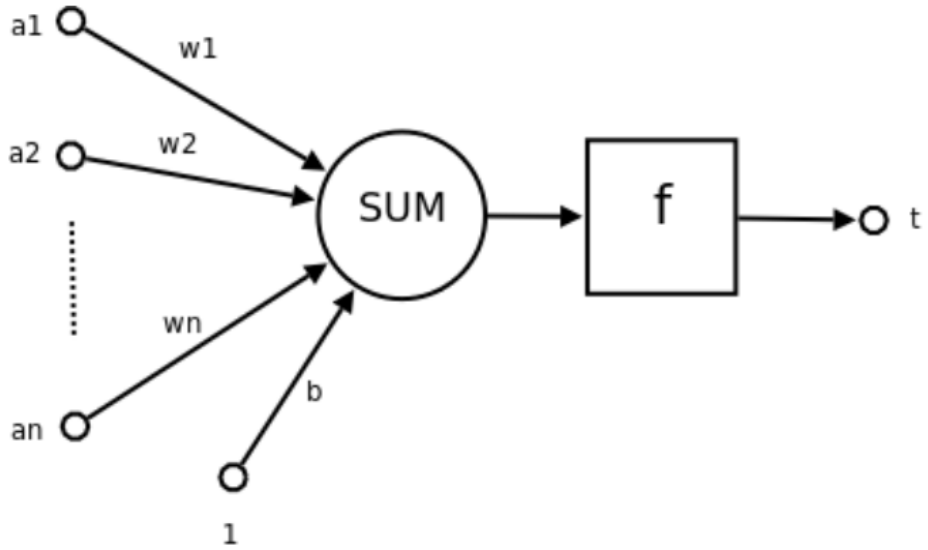
神經元 -重點中的重點

數學表示 $t = f(\overrightarrow{W'}\vec{A} + b)$

- $\vec{W}$為權向量，$\overrightarrow{W'}$為$\vec{W}$的轉置
- $\vec{A}$為輸入向量
- b為偏置
- f為傳遞函式

可見，一個神經元的功能是求得輸入向量與權向量的內積後，經一個非線性傳遞函式得到一個純量結果。

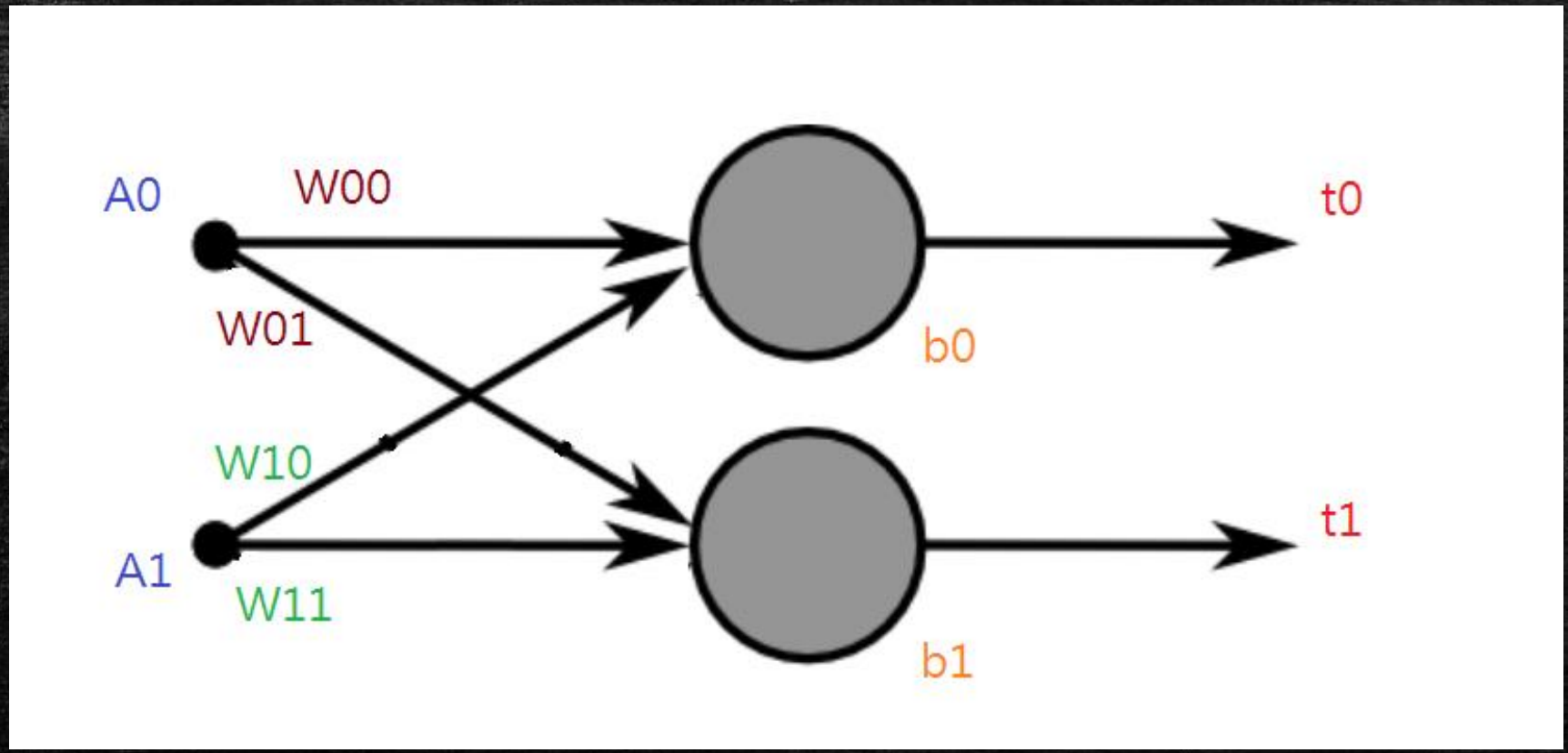
神經元示意圖：



- a1~an為輸入向量的各個分量
- w1~wn為神經元各個突觸的權重值(weight)
- b為偏置(bias)
- f為傳遞函式，通常為非線性函式。一般有traingd(),tansig(),hardlim()。以下預設為hardlim()
- t為神經元輸出

計算方式

$$\begin{bmatrix} W_{00} & W_{10} \\ W_{01} & W_{11} \end{bmatrix} \times \begin{bmatrix} A_0 \\ A_1 \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} t_0 \\ t_1 \end{bmatrix}$$



怎麼證明?

- 來寫個程式吧

In [ ]:
```
# 建立屬於自己的 model
model = Sequential()
model.add(Dense(2, input_dim=2))
```

In [ ]:
```
# 先隨便設定一些 function 後進行 compile
model.compile(optimizer = tf.optimizers.Adam(),
              loss = 'sparse_categorical_crossentropy',
              metrics=['accuracy'])
model.summary()
```

In [ ]:
```
# 因為只有一個 layer, 所以直接顯示第 0 個 layer 的參數來看看
print(model.layers[0].get_weights())
print(type(model.layers[0].get_weights()[0]))
print(type(model.layers[0].get_weights()[1]))
```

In [ ]:
```
# 設定單位矩陣(對角線為1), 偏移參數為 0
weight = [ np.array([ [1,0], [0,1] ]), np.array([0,0]) ]
model.layers[0].set_weights(weight)
```

In [ ]:
```
print(model.layers[0].get_weights())
```

In [ ]:
```
# 這個單純的模型就只是輸入甚麼，輸出就是甚麼
for i in range(-2,2) :
    for j in range(-2,2) :
        print("({},{}) = {}".format(i,j,model.predict([[i,j]]) ))
```

結論

- 神經元是類神經網路 Deep Neural Network (深度神經網路)的主要精神
- 但其實它的計算方式沒想像中的繁雜，透過一些簡單的操作就可以理解它背後的運算方式