

國立雲林科技大學機械工程系

機器學習課堂報告

Activation Function



學生：機械四B B10711150 葉書廷

指導教授：吳英正 教授

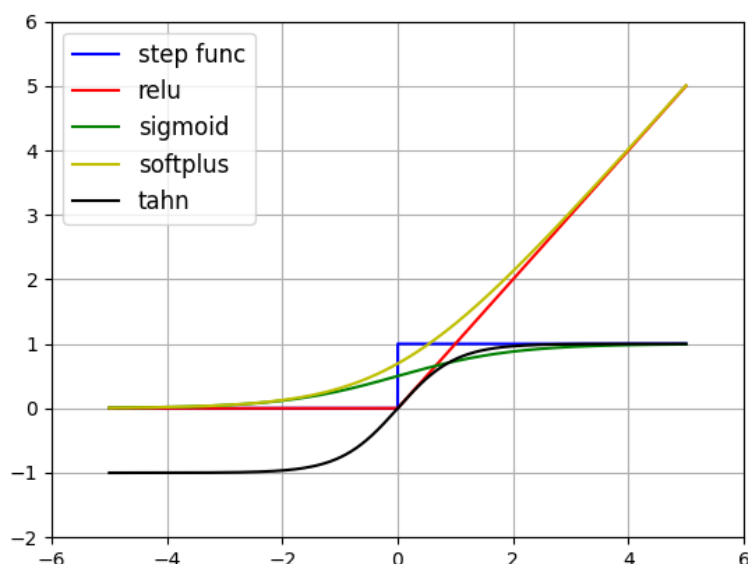
日期：民國 111 年 3 月 20 日

## 目的：

活化函數(Activation Function)在神經網路中扮演著將訊號整併並傳遞的角色，一般來說，每層神經網路的最後，都有一個活化函數，將前一層的訊號處理完後輸出給下一層或者輸出成結果，所以在神經網路中，活化函數絕對是不可或缺的，本報告將各活化函數繪製於圖表上呈現，並將通常擔任分類終點的活化函數 Softmax 以 print 的方式印出使用前後的差異。

## 結果：

程式輸出結果如下(圖一)，此圖將 5 種不同的活化函數(Step Function、Relu、Sigmoid、Softplus、Tahn)繪製於同一張圖上，可以藉此看出他們之間的差異



(圖一)、活化函數圖形

Softmax 函數之輸出前後如(圖二)，可以看出其具有表現機率的特性，在使用各種神經網路分類的時候經常使用，將 x 各值的差距與 softmax(x)各值的差距比較後也能發現其放大最大值機率的特性。

```
x = [0.3, 2.9, 4.0]
softmax(x) = [0.01821127 0.24519181 0.73659691]
```

(圖二)、Softmax 輸出前後情形

程式撰寫：

主程式：

```
import matplotlib.pyplot as plt
import numpy as np
from Basic import Act_Func

x = np.linspace(-5.0, 5.0, 10000)
x1 = [0.3, 2.9, 4.]
y = [Act_Func.step_func(x), Act_Func.relu(x), Act_Func.sigmoid(x),
      Act_Func.softplus(x), Act_Func.tahn(x)]
print('x =', x1)
print('softmax(x)=', Act_Func.softmax(x1))
plt.plot(x, y[0], 'b', label='step func')
plt.plot(x, y[1], 'r', label='relu')
plt.plot(x, y[2], 'g', label='sigmoid')
plt.plot(x, y[3], 'y', label='softplus')
plt.plot(x, y[4], 'k', label='tahn')
plt.grid(True, which='both')
plt.ylim(-2.0, 6.0), plt.xlim(-6.0, 6.0)
plt.legend(loc='upper left', prop={'size': 12})
plt.show()
```

**Basic 中的 Act\_Func：**

```
import numpy as np
class Act_Func:
    def step_func(x):
        return np.array(x > 0, dtype=np.int)
    def sigmoid(x):
        return 1/(1+np.exp(-x))
    def relu(x):
        return np.maximum(0, x)
    def softmax(x):
        x = x - np.max(x, axis=-1, keepdims=True)
        return np.exp(x) / np.sum(np.exp(x), axis=-1, keepdims=True)
    def softplus(x):
        return np.log(1+np.exp(x))
    def tahn(x):
        return np.tanh(x)
```