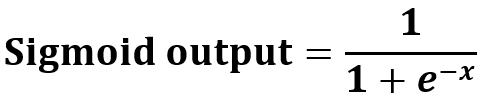
Sigmoid() 函數簡單來講就是個映射函數，將任何變量（這些先寫成 **x**）映射到 [0, 1] 之間。通常被用來當作機器學習領域 (Machine Learning) 神經網路的激活函數 (Activation Function)。

最常看到的應用情境，就是當我們在訓練模型做二分類的時候。我們將模型的最後一層神經網路設定為只有一個神經元，而我們再將最後神經元所輸出的值輸入 Sigmoid() 函數。這樣一來，我們就會得到一個介於 [0, 1] 之間的數值了。

這樣有什麼好處呢？剛剛不是提到我們要做的是二分類嗎？我們只需要設定閥值，將小於 0.5 的值通通判定為 0、大於 0.5 的值通通判斷為 1 (其實你高興的話，反過來也無所謂)，就可以做出二分類的預測了。

至於模型要怎麼知道自己輸出的值應該會分到哪個類別呢？那就是 **Backward propagation** (**反向傳播**) 去更新神經網路權重的問題了，本篇文章只關注在 Sigmoid() 函數上。

Sigmoid() 函數的公式則如下：

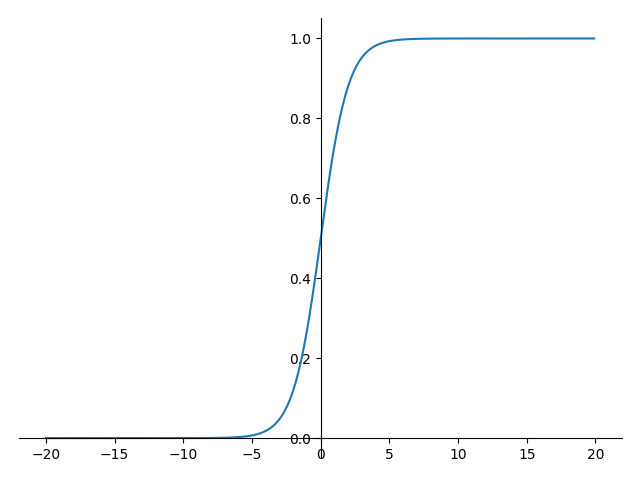
Sigmoid function

**用程式實作 Sigmoid() 函數**

為了驗證這個公式，我寫了個 Python 的小程式來畫圖。

*# -\*- coding: utf-8 -\*-*  
**import** **math**  
**import** **matplotlib.pyplot** **as** **plt**  
  
x = []  
dx = -20  
**while** dx <= 20:  
    x.append(dx)  
    dx += 0.1  
  
  
**def** sigmoid(x):  
    **return** 1/(1+math.exp(-x))  
  
  
px = [xv **for** xv **in** x]  
py = [sigmoid(xv) **for** xv **in** x]  
  
  
plt.plot(px, py)  
ax = plt.gca()  
ax.spines['right'].set\_color('none')  
ax.spines['top'].set\_color('none')  
ax.xaxis.set\_ticks\_position('bottom')  
ax.spines['bottom'].set\_position(('data',0))  
ax.yaxis.set\_ticks\_position('left')  
ax.spines['left'].set\_position(('data',0))  
plt.show()

​Output:

Sigmoid function

可以任意調整 x 輸入的範圍，基本上圖形的 output y 值都會被壓在 [0-1] 之間。

**應用**

* Keras 和 PyTorch 裡頭都有這個函數可以直接調用。
* Sigmoid 在輸出接近 0 或 1 的時候可能產生梯度消失 (vanishing gradient)。