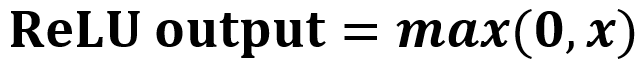
ReLU function (Rectified Linear Unit)，常被翻譯為『修正線性單元』，是一種神經網路當中常用的 Activation function。被認為具有一定程度的生物原理。（雖然我並不清楚是什麼原理。）

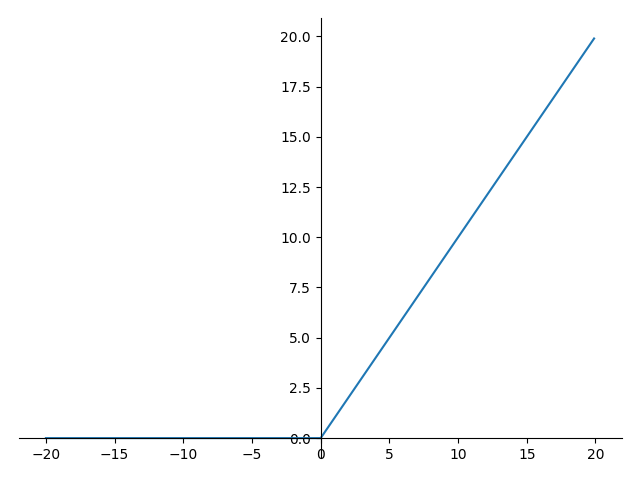
以下是 ReLU 的公式:



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** ReLU(x):  
    **if** x < 0: **return** 0  
    **else**: **return** x  
  
  
px = [xv **for** xv **in** x]  
py = [ReLU(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:



可以任意調整 x 輸入的範圍。  
我們可以看到，在 x < 0 的部份，輸出一直都是 0。

**Leaky ReLU**

Leaky ReLU function 是一種 ReLU 的變種。如果說 ReLU function 是將所有的負值設為 0，那麼 Leaky ReLU 便是將負值乘上一個大於 0 的斜率。（其實也有小於 0 的情況？雖然我沒看過就是了。）

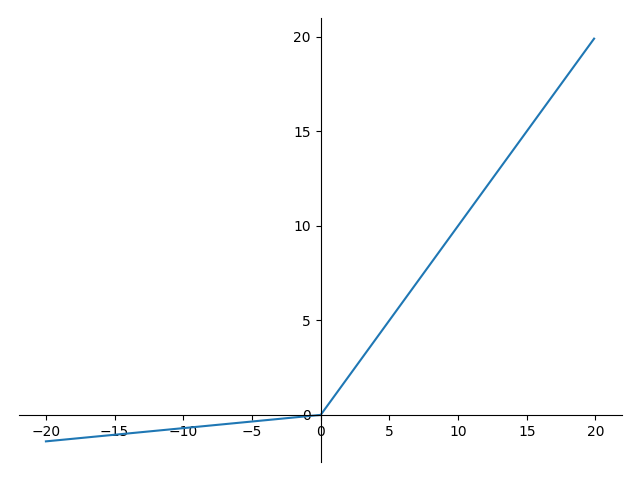
公式：

https://i0.wp.com/clay-atlas.com/wp-content/uploads/2019/10/image-36.png?resize=640%2C59&ssl=1

以下我再次寫了個小程式， a 值我固定為 0.07 討個幸運：

*# -\*- coding: utf-8 -\*-*  
**import** **matplotlib.pyplot** **as** **plt**  
  
a = 0.07  
x = []  
dx = -20  
  
  
**while** dx <= 20:  
    x.append(dx)  
    dx += 0.1  
  
  
**def** LeakyReLU(x):  
    **if** x < 0: **return** a\*x  
    **else**: **return** x  
  
  
px = [xv **for** xv **in** x]  
py = [LeakyReLU(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:



可以看到與原本的 ReLU 不同的左側。

**應用**

* Keras 和 PyTorch 裡頭都有這個函數可以直接調用。
* 由於線性導致計算速度很快
* 收斂速度很快
* 輸入是負數時，若是 Learning Rate 過大，可能會有問題