Compare Sigmoid Function with ReLU

Ref: https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function

What is Sigmoid Function?

A Sigmoid Function 是一個具有S形狀曲線的數學函數，常見的 sigmoid function 像是 logistic function, hyperbolic tangent, arctangent.
而在機器學習的領域裡， sigmoid function 通常使用 logistic function, 所以也稱為 logistic sigmoid function.
所有 sigmoid function 都具有映射全部數線上的數值到一個很小的範圍，例如 (0,1)、(-1,1)，因此 sigmoid function 的一個功能是將實數轉換為一個可以解釋機率的數值
最常廣泛使用的 sigmoid function 是 logistic function, 其映射任意實數到(0,1)， 因為這個特性在深度學習領域被用來當作人工神經網路的激活函數(activation function)
而當 sigmoid function 放在機器學習模型的最後一層時可以把模型的輸出轉換成機率分數，如此一來可以更好的解釋以及操作模型的輸出
Sigmoid function 是邏輯回歸模型中重要的一部份。邏輯回歸是對二類分類的線性回歸的修改版，將一個或多個實際數值的輸入轉換成機率，像是消費者購買產品的機率。

Sigmoid Function Formula

所有 sigmoid 函數都是 monotonic(單調)且有鐘形的第一導數，下圖為 sigmoid function，可以看到具有基本的 S 形狀

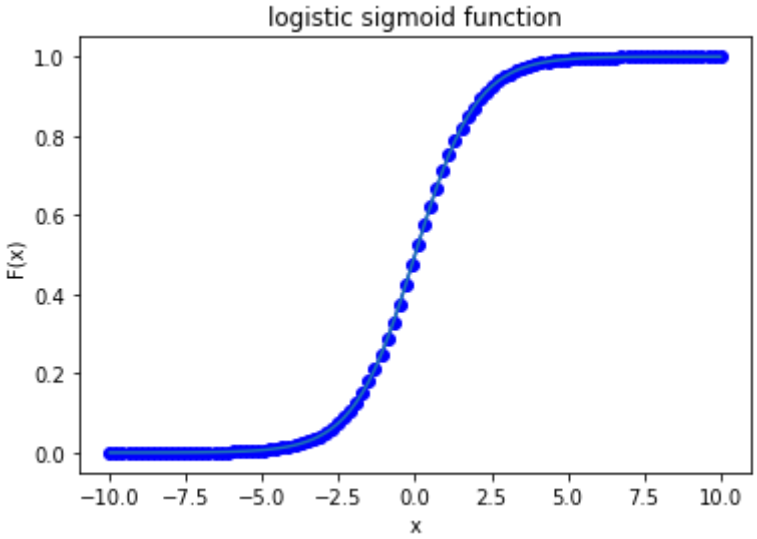
In [ ]:
```
from scipy.special import expit
import matplotlib.pyplot as plt
import numpy as np

# def range of x-values
x = np.linspace(-10, 10 ,100)

# calculate sigmoid function for each x-value
y = expit(x)

# create plot
plt.title('logistic sigmoid function')
plt.plot(x, y, color = 'blue', marker = "o")
plt.plot(x,y)
plt.xlabel('x')
plt.ylabel('F(x)')
plt.show
```

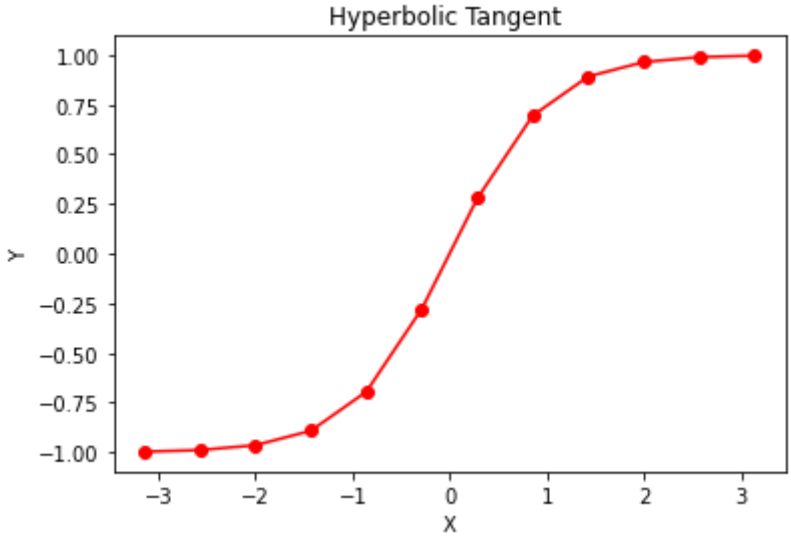
Out[ ]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Hyperbolic Tangent Function Formula

In [ ]:
```
in_array = np.linspace(-np.pi, np.pi, 12)
out_array = np.tanh(in_array)

plt.plot(in_array, out_array, color = 'red', marker = "o")
plt.title("Hyperbolic Tangent ")
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```



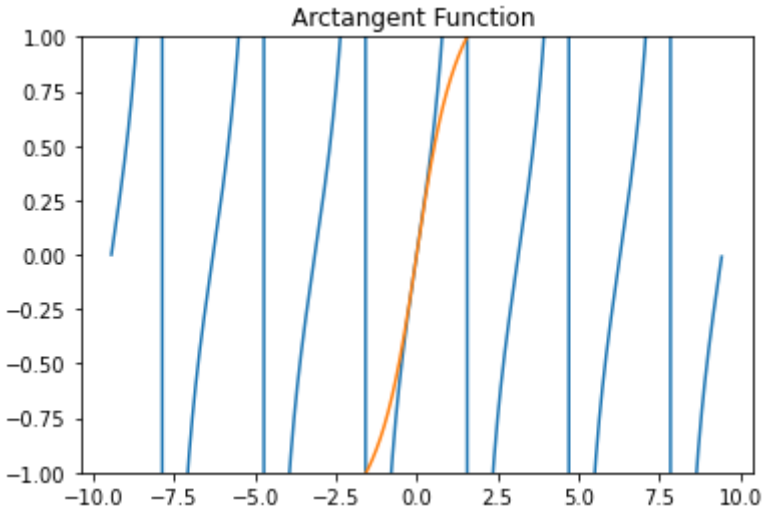
Arctangent Function Formula

In [ ]:
```
x = np.arange(-3*np.pi, 3*np.pi, step = 0.02)

plt.ylim(-1,1)
tan = np.tan(x)
arctan = np.arctan(x)

plt.title('Arctangent Function ')
plt.plot(x,tan)
plt.plot(x,arctan)
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x192fcecb970>]
```



Summary of three sigmoid functions

Sigmoid function	Logistic function	Hyperbolic Tangent function	Arctangent Function
Value in the limit x →-∞	0	-1	-π/2
Value at x = 0	0.5	0	0
Value in the limit x →∞	1	1	π/2
Converges	Fast	Very fast	Very slow

What is Relu(rectified linear unit)?

Relu 是一個非線性的激活函數，用於多層神經網路或是深度神經網路，函數可以用下面方程式表示

$$f(x) = max(0, x) \tag{1}$$

According to eq(1), the output of Relu is the max value between 0 amd the input value. An output is equal to zero when the input value is negative and the input value when the input is positive. Thus, we can rewrite equation 1 as follows:

where x = an input value

$$f(x) = \begin{cases} 0, . & \text{if } x<0 \\ 1, . & \text{if } x \geqq 0 \end{cases} \tag{2}$$

The purpose of Relu AKA Compare Sigmoid Function with Relu

一般來說，一些盛行的非線性激活函數，像是 sigmoid function(logistic) 或是 hyperbolic tangent ，都是用在神經網路去得到激活函數的值去對應到每一個神經元。
而最近 Relu function 被用來計算傳統神經網路或深度神經網路 paradigms(範例)中的激活值 Relu 取代 Sigmoid 和 Hyperbolic tangent 的理由如下：

1. Computation saving
   與傳統的激活函數相比，ReLu函數能夠加快深度神經網絡的訓練速度，因為ReLu的導數

在正的輸入下是1。由於是常數，深度神經網絡在訓練階段不需要花費額外的時間來計算誤差項。

2. Sloving the vanishing gradient problem

   當層數增加時，ReLu函數不會引發梯度消失的問題。這是因為這個函數沒有一個漸進的上下限。因此，最早的一層（第一個隱藏層）能夠接收來自最後幾層的錯誤，以調整層間的所有權重。

   相比之下，像sigmoid這樣的傳統激活函數被限制在0和1之間，所以對於第一個隱藏層來說，誤差變得很小。這種情況會導致神經網絡的訓練效果不佳。

3. non-linear shape

   Relu 的非線性結構可以在神經網路的輸入和輸出之間複雜的關係建立模型。

   這點和 sigmoid 或是 hyperbolic tangent 來說的線性結構來說很重要，因為現實世界中的問題都是非線性關係，無法用線性函數去準確地建立模型