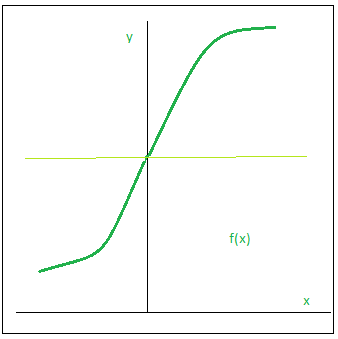
Experiment – 2

Aim:- Performing Activation Function by using Sigmoid and Tanh functions.

Activation Function:- Activation Function is a mathematical function which is applied at the output node to introduce non-linearity in a Artificial Neuron. It is used to determine weather neuron is active or not based on input values.

Activation Function used in this program are:-

Sigmoid:- It is most used activation function in Neural Network and Logistic Regression. It’s Output range between 0 and 1. It’s look like a S-curve because its value increases from 0 to 1

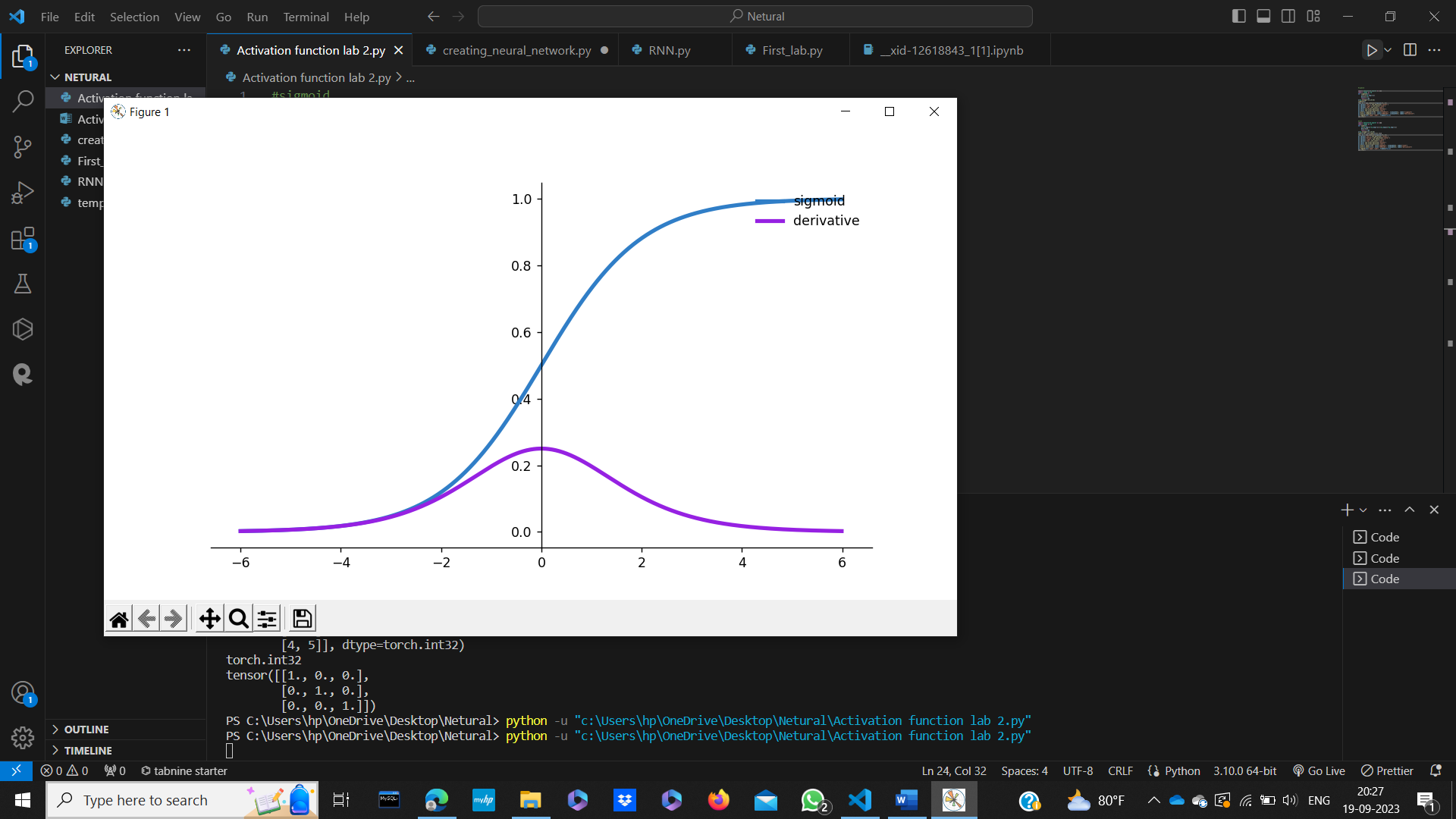


Source:- Geeks For Geeks.

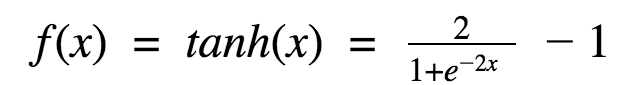
**Code:-**

|  |
| --- |
| #sigmoid  import matplotlib.pyplot as plt  import numpy as np  def sigmoid(x):  s=1/(1+np.exp(-x))  ds=s\*(1-s)  return s,ds  x=np.arange(-6,6,0.01)  sigmoid(x)  fig, ax = plt.subplots(figsize=(9, 5))  ax.spines['left'].set\_position('center')  ax.spines['right'].set\_color('none')  ax.spines['top'].set\_color('none')  ax.xaxis.set\_ticks\_position('bottom')  ax.yaxis.set\_ticks\_position('left')  ax.plot(x,sigmoid(x)[0], color="#307EC7", linewidth=3, label="sigmoid")  ax.plot(x,sigmoid(x)[1], color="#9621E2", linewidth=3, label="derivative")  ax.legend(loc="upper right", frameon=False)  plt.show() |

**Output:-**

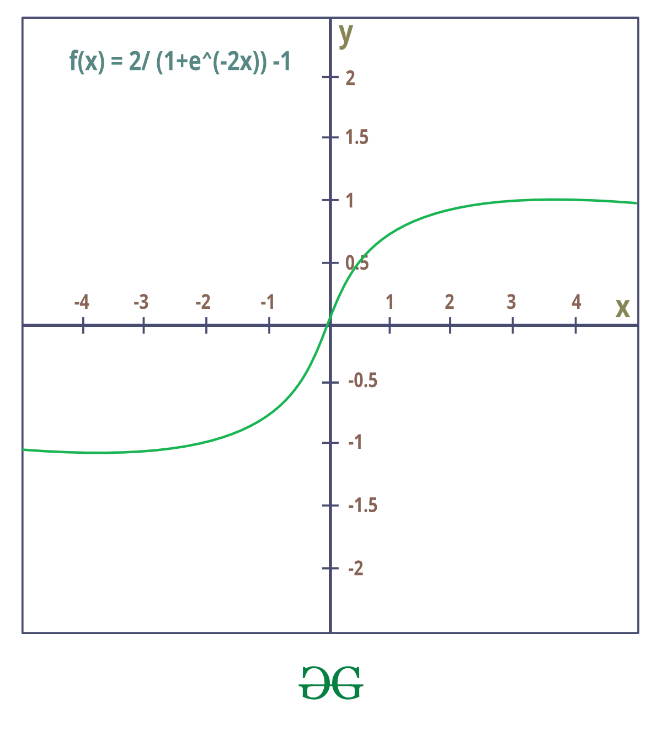


Tanh:- It is another activation function in neural network, whose value lies between -1 and 1. It is also S-shape curve.



Or

tanh(x) = (e^(x) - e^(-x)) / (e^(x) + e^(-x))

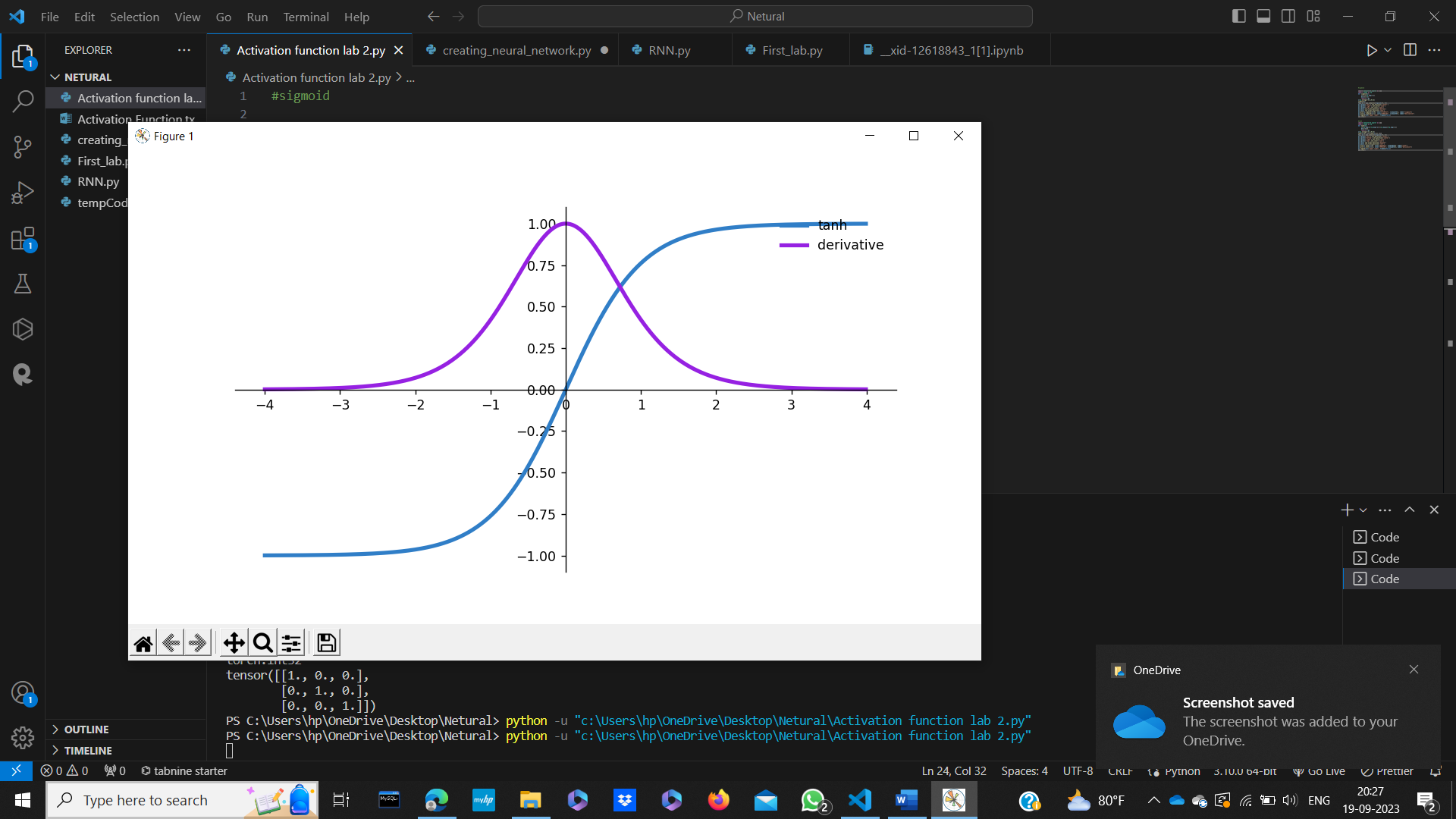


Source:-Geeks For Geeks

**Code:-**

|  |
| --- |
| #tanh  import matplotlib.pyplot as plt  import numpy as np  def tanh(x):  t=(np.exp(x)-np.exp(-x))/(np.exp(x)+np.exp(-x))  dt=1-t\*\*2  return t,dt  z=np.arange(-4,4,0.01)  tanh(z)[0].size,tanh(z)[1].size  fig, ax = plt.subplots(figsize=(9, 5))  ax.spines['left'].set\_position('center')  ax.spines['bottom'].set\_position('center')  ax.spines['right'].set\_color('none')  ax.spines['top'].set\_color('none')  ax.xaxis.set\_ticks\_position('bottom')  ax.yaxis.set\_ticks\_position('left')  ax.plot(z,tanh(z)[0], color="#307EC7", linewidth=3, label="tanh")  ax.plot(z,tanh(z)[1], color="#9621E2", linewidth=3, label="derivative")  ax.legend(loc="upper right", frameon=False)  plt.show() |

**Output:-**



**Learning:-** So, in this experiment we have learn how to implement Sigmoid and Tanh function numpy and math plot library.