ann1

import matplotlib.pyplot as plt

import numpy as np

#1. Step activation function

def step(x):

if x>=0:

return 1

else:

return 0

x=np.arange(-6,6,0.01)

step\_output=[step(i)for i in x]

fig,ax=plt.subplots(figsize=(9,5))

ax.plot(x,step\_output, color="#307Ec7", linewidth=3, label="step")

ax.legend(loc="upper left", frameon=False)

fig.show()

#Sigmoid activation function

def sigmoid(x):

y=(1/(1+np.exp(-x)))

return y

x=np.arange(-6,6,0.01)

output = [sigmoid(i)for i in x]

fig,ax=plt.subplots(figsize=(9,5))

ax.plot(x,output, color="#307Ec7", linewidth=3, label="sigmoid")

ax.legend(loc="upper left", frameon=False)

fig.show()

# Tanh activation function

def tanh(x):

y=(np.exp(x)-np.exp(-x))/(np.exp(x)+np.exp(-x))

return y

x=np.arange(-6,6,0.01)

output=[tanh(i)for i in x]

fig,ax=plt.subplots(figsize=(9,5))

ax.plot(x,output, color="#307Ec7", linewidth=3, label="tanh")

ax.legend(loc="upper left", frameon=False)

fig.show()

#Relu activation function

def relu(x):

return max(0,x)

x=np.arange(-6,6,0.01)

output=[relu(i)for i in x]

fig,ax=plt.subplots(figsize=(9,5))

ax.plot(x,output, color="#307Ec7", linewidth=3, label="relu")

ax.legend(loc="upper left", frameon=False)

fig.show()

#Leaky relu activation function

def leaky\_relu(x):

return max(0.1\*x,x)

x=np.arange(-6,6,0.01)

output=[leaky\_relu(i)for i in x]

fig,ax=plt.subplots(figsize=(9,5))

ax.plot(x,output, color="#307Ec7", linewidth=3, label="leaky\_relu")

ax.legend(loc="upper left", frameon=False)

fig.show()

#Softplus activation function

def softplus(x):

return np.log(1+np.exp(x))

x=np.arange(-6,6,0.01)

output=[softplus(i)for i in x]

fig,ax=plt.subplots(figsize=(9,5))

ax.plot(x,output, color="#307Ec7", linewidth=3, label="softplus")

ax.legend(loc="upper left", frameon=False)

fig.show()

#Glu activation function

def glu(x):

return (x\* sigmoid(1.702\*x))

x=np.arange(-6,6,0.01)

output=[glu(i)for i in x]

fig,ax=plt.subplots(figsize=(9,5))

ax.plot(x,output, color="#307Ec7", linewidth=3, label="glu")

ax.legend(loc="upper left", frameon=False)

fig.show()

#Swish activation function

def swish(x):

return (x\* sigmoid(x))

x=np.arange(-6,6,0.01)

output=[swish(i)for i in x]

fig,ax=plt.subplots(figsize=(9,5))

ax.plot(x,output, color="#307Ec7", linewidth=3, label="swish")

ax.legend(loc="upper left", frameon=False)

fig.show()