

+ Code + Text

RAM 0 / 16GB
Disk 0 / 100GB

Editing

```
import matplotlib.pyplot as plt # For plotting graphs
import numpy as np             # For exp, log etc.,
```

Step Activation

[] 3 cells hidden

Sigmoid Function

[] 3 cells hidden

Tanh Activation

[] 3 cells hidden

Activation Functions.ipynb · Colab

colab.research.google.com/drive/1CjH3JuCYCgHrI6lp50MglZgwwfemphk#scrollTo=6DeBDdKdOBjd

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM 100% Disk 100% Editing

import matplotlib.pyplot as plt # For plotting graphs
import numpy as np # For exp, log etc.,

▶ Step Activation
[] 3 cells hidden

▶ Sigmoid Function
[] 3 cells hidden

▶ Tanh Activation
[] 3 cells hidden

▶ ReLU Activation

6:56 PM 27-Feb-22

Activation Functions.ipynb · Colab

colab.research.google.com/drive/1CjH3JuCYGgHrI6lp50MglZgwwfemphk#scrollTo=6DeBDdKdOBjd

colab

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM 100% Disk 100%

Editing

[3] import matplotlib.pyplot as plt # For plotting graphs
import numpy as np # For exp, log etc.,

Step Activation

3 cells hidden

Sigmoid Function

3 cells hidden

Tanh Activation

3 cells hidden

ReLU Activation

6:57 PM 27-Feb-22



Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved



Comment



Share



+ Code + Text

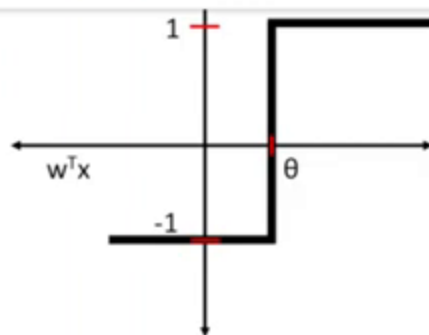


RAM

Disk



Editing



Binary step

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

Unit step function.

```
[1] def step(x):
```



Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved



Comment



Share



+ Code + Text



RAM

Disk



Editing



Binary step

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

Unit step function.

```
def step(x):
    if x >= 0:
        return 1
    else:
        return 0
```

```
[5] x=np.arange(-6,6,0.01)
```



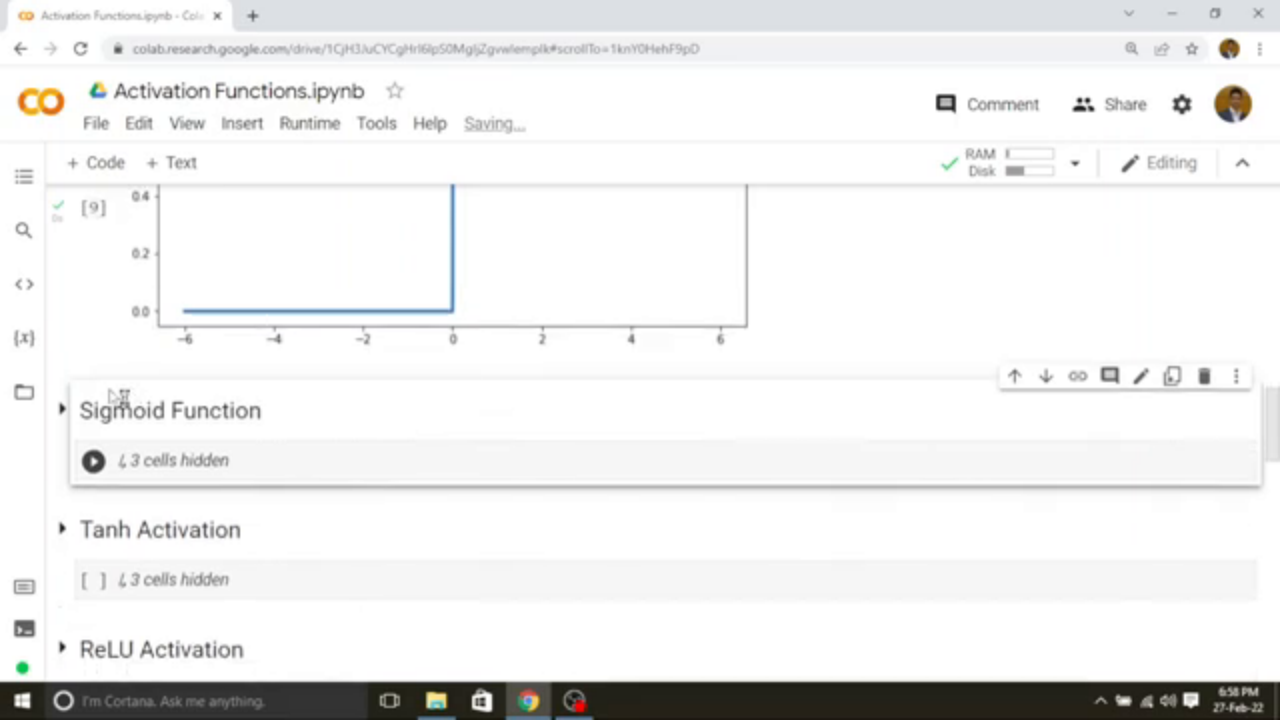
$1 \text{ for } x \geq 0$ Unit step function.

```
[8] 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]

# Plotting
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 Function



+ Code + Text

RAM Disk Editing

Sigmoid Function

$$S(x) = \frac{1}{1 + e^{-x}}$$



+ Code + Text

```
[6] def sigmoid(x):
    s = (1/(1+np.exp(-x))) # sigmoid function
    return s
```


+ Code + Text

RAM Disk Editing



```
[10] def sigmoid(x):
      s = (1/(1+np.exp(-x))) # sigmoid function
      return s
```

```
x=np.arange(-6,6,0.01)
# Plotting
fig, ax = plt.subplots(figsize=(9, 5))
ax.plot(x,sigmoid(x), color="#307EC7", linewidth=3, label="sigmoid")
ax.legend(loc="upper left", frameon=False)
fig.show()
```

Tanh Activation

[] 3 cells hidden



Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved



Comment



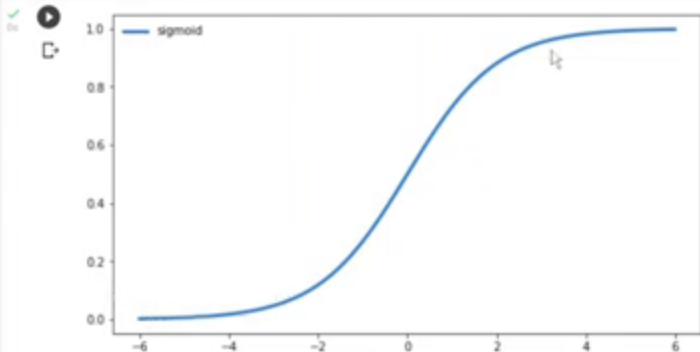
Share



+ Code + Text

RAM
Disk

Editing



Tanh Activation

3 cells hidden



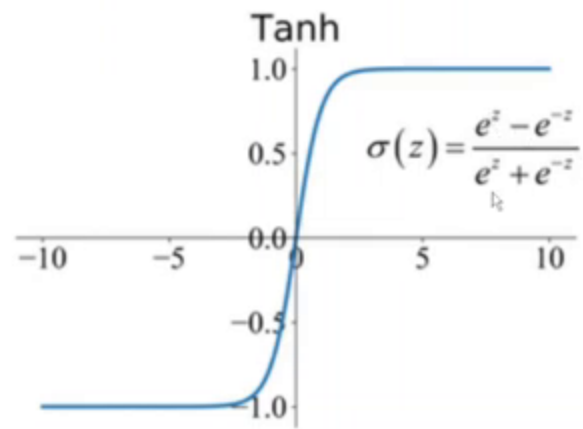
Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings Profile

+ Code + Text

RAM Disk Editing





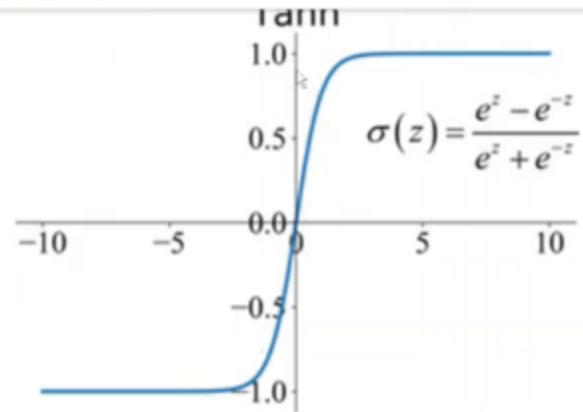
Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

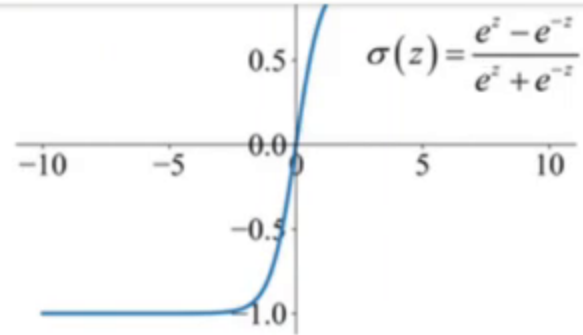
Comment Share Settings Profile

+ Code + Text

RAM 100% Disk 100% Editing



```
[ ] def tanh(x):
    f=(np.exp(x)-np.exp(-x))/(np.exp(x)+np.exp(-x))
```



```
def tanh(x):
    t=(np.exp(x)-np.exp(-x))/(np.exp(x)+np.exp(-x))
    # t = np.tanh(x)
    return t
```



Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved



Comment



Share



+ Code + Text

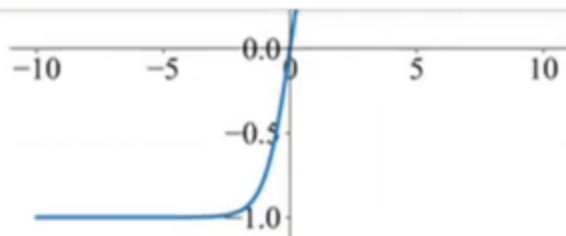


RAM

Disk



Editing

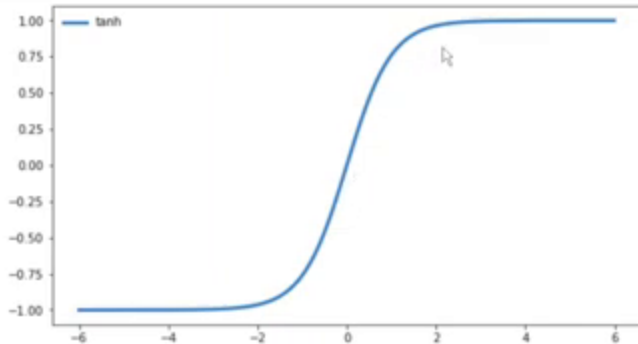


```
def tanh(x):
    t=(np.exp(x)-np.exp(-x))/(np.exp(x)+np.exp(-x))
    # t = np.tanh(x)
    return t
```

```
[ ] x=np.arange(-6,6,0.01)
# Plotting
fig, ax = plt.subplots(figsize=(9, 5))
ax.plot(x,tanh(x), color="#307EC7", linewidth=3, label="tanh")
```

+ Code + Text

RAM Disk Editing



↑ ↓ 🔗 💬 ✎ 📄 🗑️ ⋮

ReLU Activation

3 cells hidden



Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved



Comment



Share



+ Code + Text



RAM

Disk



Editing

ReLU Activation

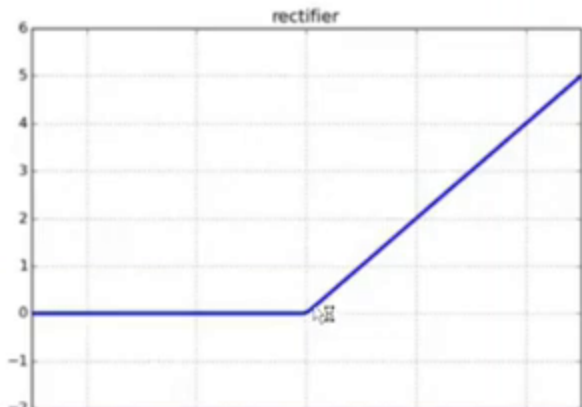


$f(x) =$



0 for $x < 0$

x for $x \geq 0$





Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved



Comment



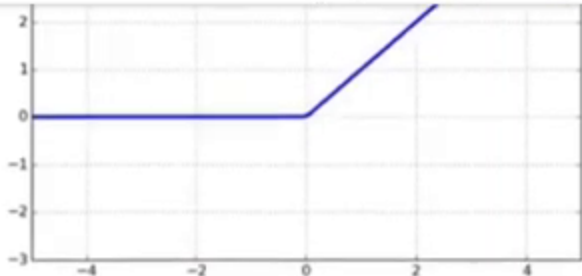
Share



+ Code + Text

RAM 100%
Disk 100%

Editing

 $f(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x \geq 0 \end{cases}$ 

```
[ ] def relu(x):  
    y = max(0,x)  
    return y
```

```
[ ] x=np.arange(-6,6,0.01)  
    relu_output = [relu(i) for i in x]  
    # plotting
```



Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved



Comment



Share



+ Code + Text



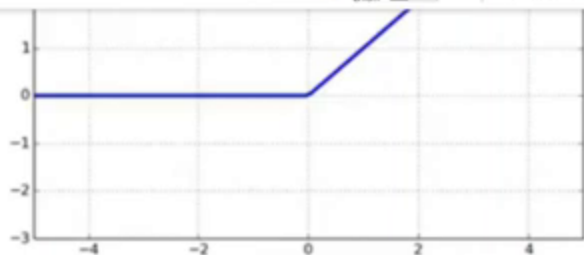
RAM

Disk



Editing

x for x => 0

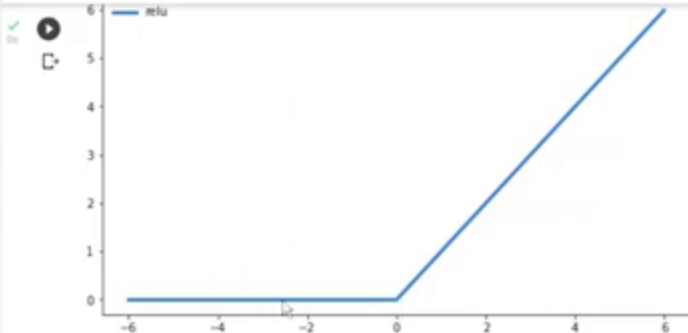


```
def relu(x):
    y = max(0, x)
    return y

[ ] x=np.arange(-6,6,0.01)
    relu_output = [relu(i) for i in x]
    # Plotting
    fig, ax = plt.subplots(figsize=(9, 5))
```

+ Code + Text

RAM Disk Editing



Leaky ReLU

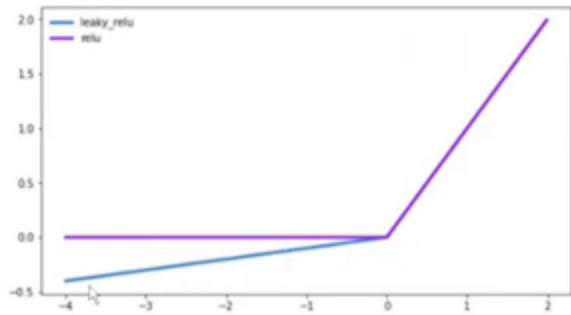
3 cells hidden



+ Code + Text

RAM 8 Disk 100% Editing

Leaky ReLU



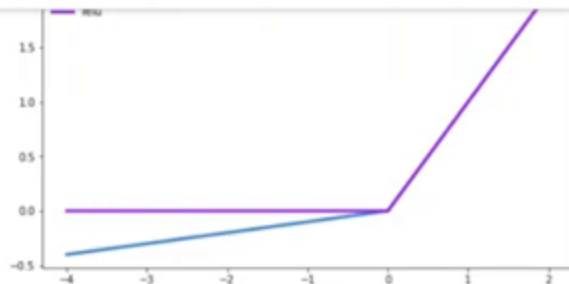
Leaky ReLU

$$f(x) = \max(0.1x, x)$$

```
[ ] def leaky_relu(x):
    y = max(0.1*x, x)
```

+ Code + Text

RAM Disk Editing



Leaky ReLU

$$f(x) = \max(0.1x, x)$$

```
def leaky_relu(x):
    y = max(0.1*x, x)
    return y
```

```
[ ] x=np.arange(-6,6,0.01)
    leaky_relu_output = [leaky_relu(i) for i in x]
```



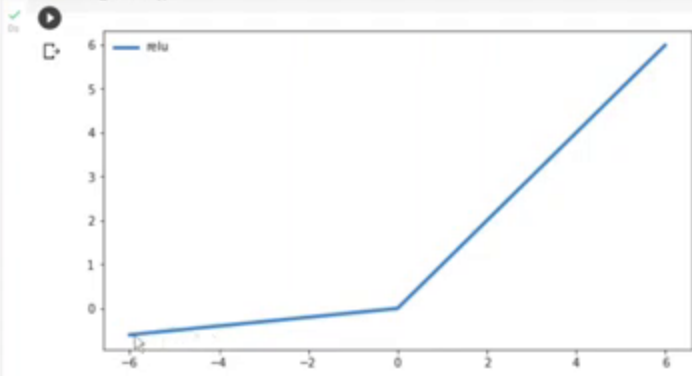
Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings Profile

+ Code + Text

RAM Disk Editing



Navigation icons: up, down, search, comment, edit, copy, delete, etc.

Softplus Activation

3 cells hidden



Comment



Share



+ Code + Text



RAM



Disk

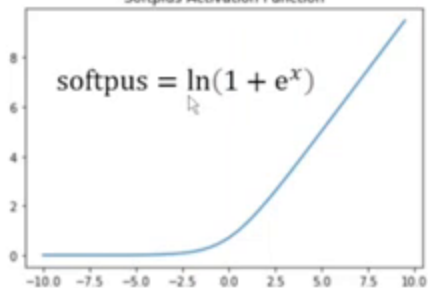


Editing

Softplus Activation



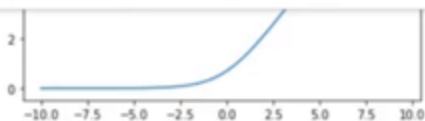
Softplus Activation Function



```
[ ] def softplus(x):
    s = np.log(1+np.exp(x))
```

+ Code + Text

RAM Disk Editing



```
[18] def softplus(x):
      s = np.log(1+np.exp(x))
      return s
```

```
x=np.arange(-6,6,0.01)
# Plotting
fig, ax = plt.subplots(figsize=(9, 5))
ax.plot(x,softplus(x), color="#307EC7", linewidth=3, label="softplus")
ax.legend(loc="upper left", frameon=False)
fig.show()
```

GeLU Activation



Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved



Comment



Share



+ Code + Text



RAM

Disk

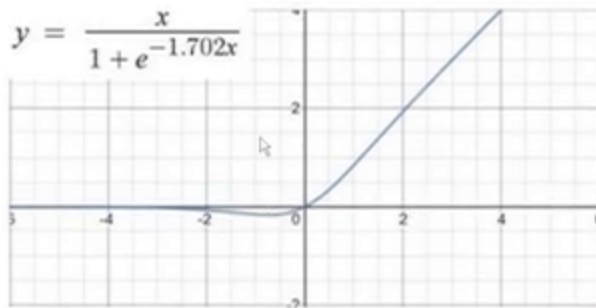


Editing



GeLU Activation

$$y = \frac{x}{1 + e^{-1.702x}}$$





Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved



Comment



Share



+ Code + Text

RAM
Disk

Editing

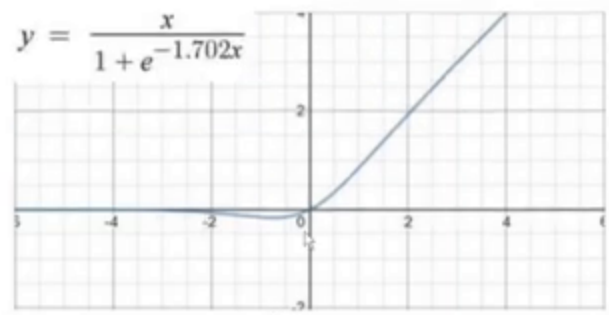


```
def sigmoid(x):
    s = (1/(1+np.exp(-x))) # sigmoid function
    return s

def gelu(x):
    g = x * sigmoid(1.702*x)
    return g
```

```
[ ] z=np.arange(-6,6,0.01)
```

+ Code + Text



```
[20] def sigmoid(x):
      s = (1/(1+np.exp(-x))) # sigmoid function
      return s
```

Activation Functions.ipynb - Colab

colab.research.google.com/drive/1CjH3JuCYCgHrI6lp50MgljZgwwfempfk#scrollTo=UUn2VxRyahjZ

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM 100% Disk 100%

Editing

Swish

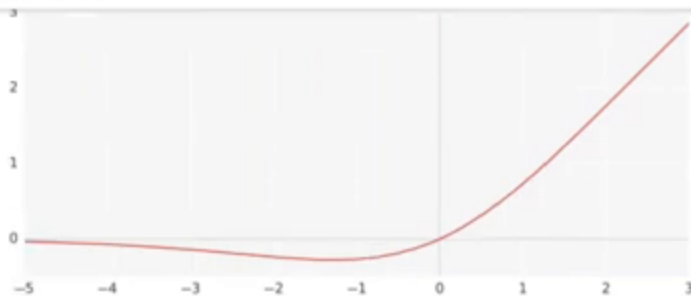
$$f(x) = x * \text{sigmoid}(x)$$

Im Cortana. Ask me anything.

7:05 PM 27-Feb-22

+ Code + Text

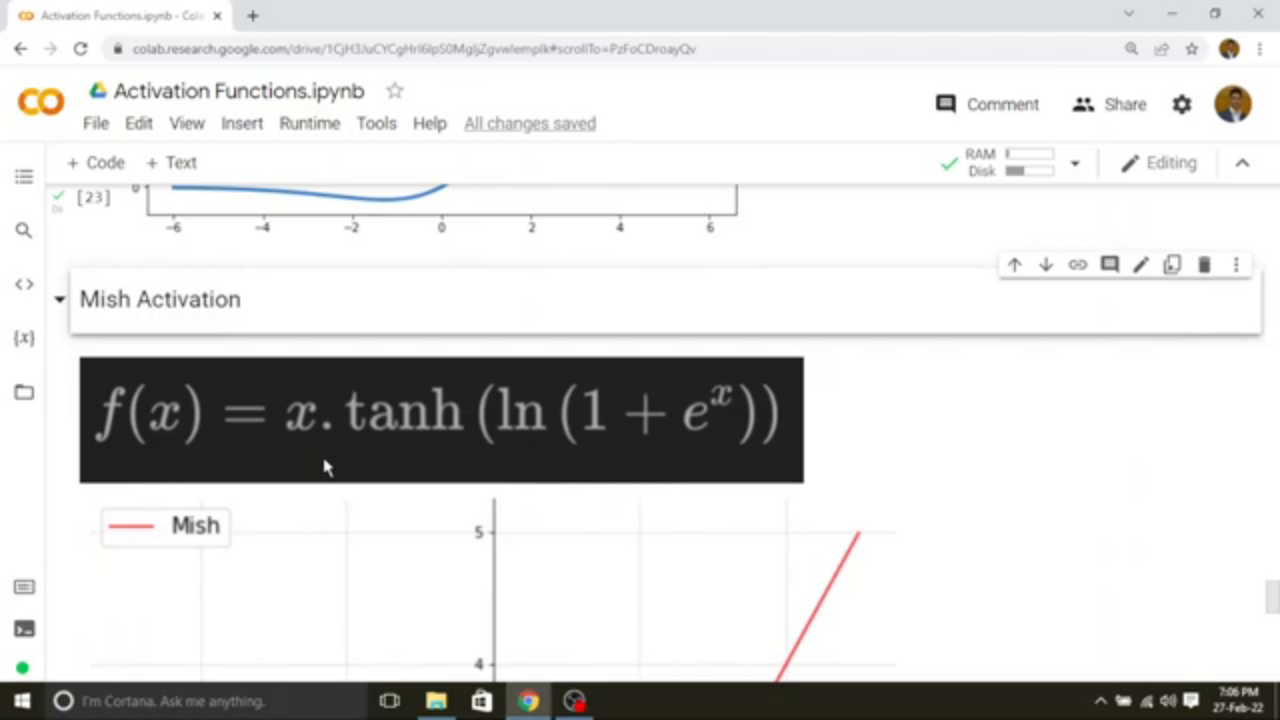
RAM Disk Editing

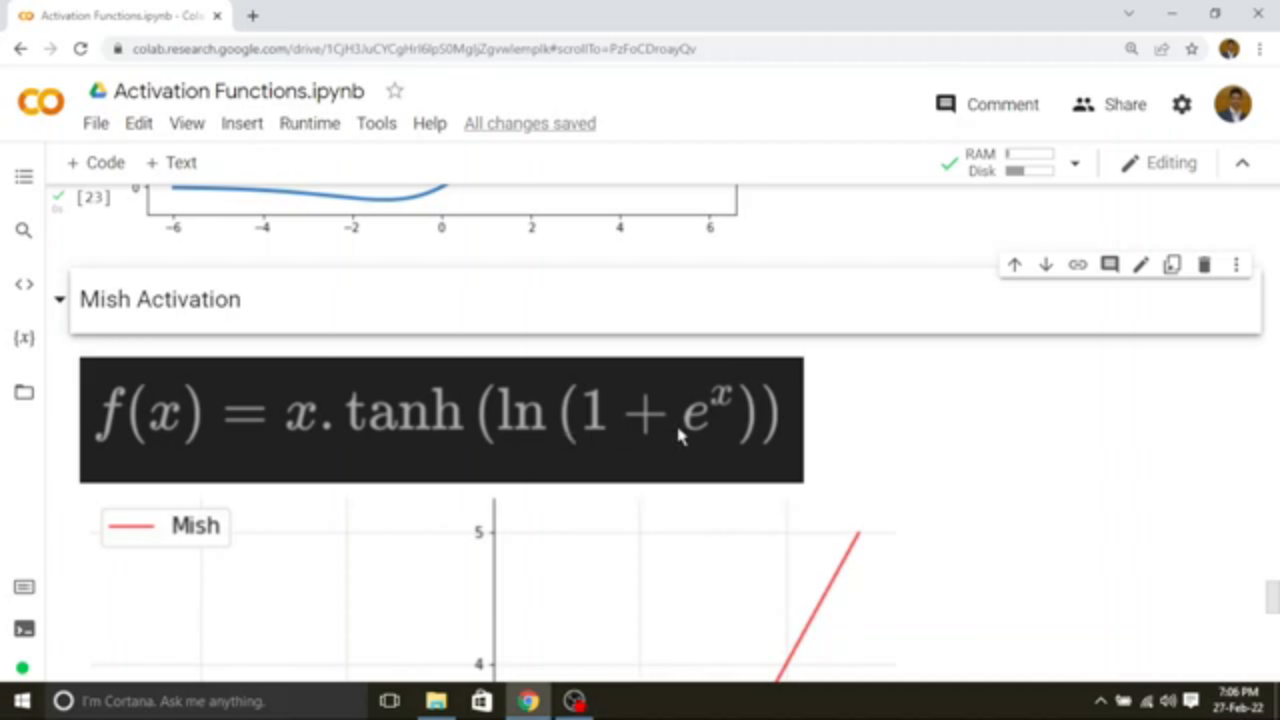


```
def swish(x):
    s = x * (1 / (1 + np.exp(-x)))
    return s
```

+ Code + Text

```
z = np.arange(-6, 6, 0.01)
```





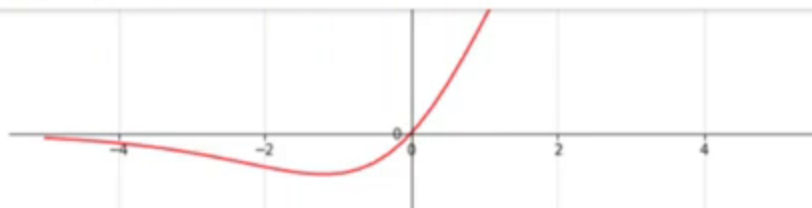
Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing



```
def mish(x):
    m = x * np.tanh(np.log(1+np.exp(x)))
    return m
```

```
[ ] z=np.arange(-6,6,0.01)

fig, ax = plt.subplots(figsize=(9, 5))
ax.plot(z,mish(z), color="#9621E2", linewidth=3, label="mish")
ax.legend(loc="upper left", frameon=False)
fig.show()
```



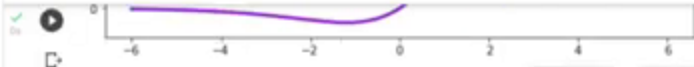

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings Profile

+ Code + Text

RAM 8 GB Disk 100% Editing



+ Code + Text

Maxout Activation

$$h(x) = \max(Z_1, Z_2, \dots, Z_n)$$

$$h(x) = \max(W_1 \cdot x + b_1, W_2 \cdot x + b_2, \dots, W_n \cdot x + b_n)$$





Maxout Activation

$$h(x) = \max(Z_1, Z_2, \dots, Z_n)$$

$$h(x) = \max(W_1 \cdot x + b_1, W_2 \cdot x + b_2, \dots, W_n \cdot x + b_n)$$



Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings Profile

+ Code + Text

RAM Disk Editing



```
import numpy as np

x = np.random.random((2,5))
print(x.shape)

# 12 hidden neurons
W = np.random.random((12,5))
b = np.random.random((12))
print(W.shape,b.shape)
z = np.dot(x, W.T) + b
print(z.shape)
z_ = z.reshape((2,4,3))
o = np.max(z_, axis=2)
print(o.shape)
```



Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved



Comment



Share



+ Code + Text



RAM

Disk



Editing



```
import numpy as np
```

```
x = np.random.random((2,5))
print(x.shape)
```

```
# 12 hidden neurons
W = np.random.random((12,5))
b = np.random.random((12))
print(W.shape,b.shape)
z = np.dot(x, W.T) + b
print(z.shape)
z_ = z.reshape((2,4,3))
o = np.max(z_, axis=2)
print(o.shape)
```

colab.research.google.com/drive/1CjH3JuCYCgHrI6lp50MgljZgwwfemphk#scrollTo=Tr2nyPNgbgTc

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM 100% Disk 100% Editing

⏮

⏪

⏩

⏭

⚙

📄

🗑

⋮

```
import numpy as np

x = np.random.random((2,5))
print(x.shape)

# 12 hidden neurons
W = np.random.random((2,5))
b = np.random.random((12))
print(W.shape,b.shape)
z = np.dot(x, W.T) + b
print(z.shape)
z_ = z.reshape((2,4,3))
o = np.max(z_, axis=2)
print(o.shape)
```

▶ Softmax Activation

[] 4 3 cells hidden

7:08 PM 27-Feb-22

colab.research.google.com/drive/1CjH3JuCYCgHrI6lpS0MgljZgwwfemphk#scrollTo=Tr2nyPNgbgTc

Activation Functions.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM 100% Disk 100% Editing

⏮

⏪

⏩

⏭

⚙

📄

🗑

⋮

▶

```
import numpy as np

x = np.random.random((2,5))
print(x.shape)

# 12 hidden neurons
W = np.random.random((12,5))
b = np.random.random((12))
print(W.shape,b.shape)
z = np.dot(x, W.T) + b
print(z.shape)
z_ = z.reshape((2,4,3))
o = np.max(z_, axis=2)
print(o.shape)
```

▶ Softmax Activation

[] 4 3 cells hidden

Windows Taskbar

7:09 PM 27-Feb-22

colab.research.google.com/drive/1CjH3JuCYCgHr16lp50MgljZgwwfempfk#scrollTo=Tr2nyPNgbgTc

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM 8 Disk 100 Editing

import numpy as np

x = np.random.random((2,5))

print(x.shape)

12 hidden neurons

W = np.random.random((12,5))

b = np.random.random((12))

print(W.shape,b.shape)

z = np.dot(x, W.T) + b

print(z.shape)

z_ = z.reshape((2,4,3))

o = np.max(z_, axis=2)

print(o.shape)

Softmax Activation

[] 4 3 cells hidden

7:09 PM 27-Feb-22

Activation Functions.ipynb · Colab

colab.research.google.com/drive/1CjH3JuCYCgHrI6lpS0MgIjZgwwfempIk#scrollTo=Tr2nyPNgbgTc

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

RAM 8 GB Disk 1.5 GB

Editing

```
import numpy as np

x = np.random.random((2,5))
print(x.shape)

# 12 hidden neurons
W = np.random.random((12,5))
b = np.random.random((12))
print(W.shape,b.shape)
z = np.dot(x, W.T) + b
print(z.shape)
z_ = z.reshape((2,4,3))
o = np.max(z_, axis=2)
print(o.shape)
```

(2, 5)

(12, 5) (12,)

(2, 12)

(2, 4)

Softmax Activation

7:09 PM

27-Feb-22

Activation Functions.ipynb - Colab

colab.research.google.com/drive/1CjH3JuCYCgHrI6lpS0MgljZgwwfempfk#scrollTo=Tr2nyPNgbgTc

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM 100% Disk 100% Editing

import numpy as np

x = np.random.random((2,5))

print(x.shape)

12 hidden neurons

W = np.random.random((12,5))

b = np.random.random((12))

print(W.shape,b.shape)

z = np.dot(x, W.T) + b

print(z.shape)

z_ = z.reshape((2,4,3))

o = np.max(z_, axis=2)

print(o.shape)

(2, 5)

(12, 5) (12,)

(2, 12)

(2, 4)

Softmax Activation

Activation Functions.ipynb · Colab

colab.research.google.com/drive/1CjH3JuCYCgHrI6lpS0MgljZgwwfempfk#scrollTo=Tr2nyPNgbgTc

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM 100% Disk 100%

Editing

```
x = np.random.random((2,5))
print(x.shape)

# 12 hidden neurons
W = np.random.random((12,5))
b = np.random.random((12))
print(W.shape,b.shape)
z = np.dot(x, W.T) + b
print(z.shape)
z_ = z.reshape((2,4,3))
o = np.max(z_, axis=2)
print(o.shape)
```

(2, 5)

(12, 5)

(2, 12)

(2, 4)

Softmax Activation

[] 4 cells hidden

I'm Cortana. Ask me anything.

7:10 PM 27-Feb-22

Activation Functions.ipynb - Colab

colab.research.google.com/drive/1CjH3JuCYGgHrI6lpS0MgljZgwwfemphk#scrollTo=Tr2nyPNgbgTc

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM 100% Disk 100%

Editing

```
x = np.random.random((2,5))
print(x.shape)

# 12 hidden neurons
W = np.random.random((12,5))
b = np.random.random((12))
print(W.shape,b.shape)
z = np.dot(x, W.T) + b
print(z.shape)
z_ = z.reshape((2,4,3))
o = np.max(z_, axis=2)
print(o.shape)
```

(2, 5)

(12, 5) (12,)

(2, 12)

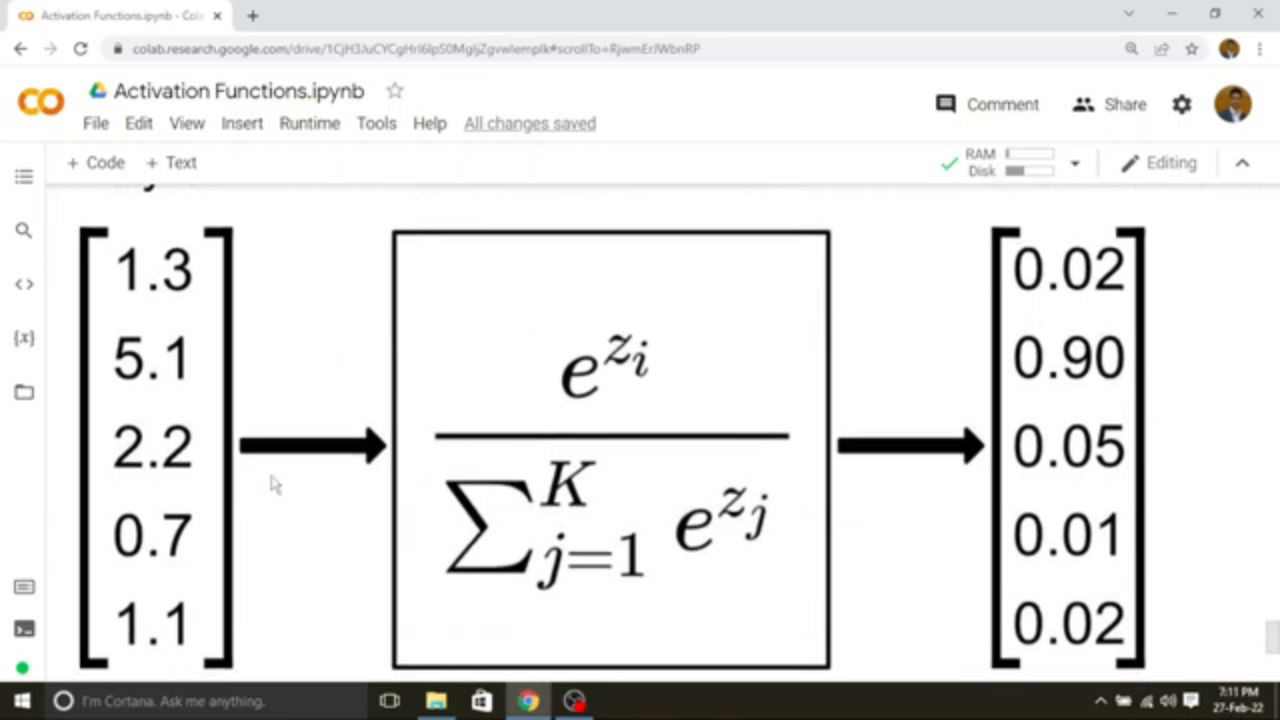
(2, 4)

Softmax Activation

[] 4 3 cells hidden

Windows Taskbar

7:10 PM 27-Feb-22



Activation Functions.ipynb

colab.research.google.com/drive/1CjH3JuCYGgHr16lp50MglZgwwfemphk#scrollTo=RjwmEr/WbnRP

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM 100% Disk 100% Editing

1.3

5.1

2.2

0.7

1.1

→

$$\frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

→

0.02

0.90

0.05

0.01

0.02

7:11 PM 27-Feb-22



Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings Profile

+ Code + Text

RAM Disk Editing

$$\begin{bmatrix} 2.2 \\ 0.7 \\ 1.1 \end{bmatrix}$$

$$\sum_{j=1}^K e^{z_j}$$

$$\begin{bmatrix} 0.01 \\ 0.02 \end{bmatrix}$$

```
def softmax(x):
    """Compute softmax values for each sets of scores in x."""
    e_x = np.exp(x)
    return e_x / e_x.sum(axis=0)
```

```
x = [1.3, 5.1, 2.2, 0.7, 1.1]
print(softmax(x))
print(softmax(x).sum())
```

Activation Functions.ipynb · Colab

colab.research.google.com/drive/1CjH3JuCYCgHrI6pS0MgjZgwwlempIk#scrollTo=jsByS24Gbu7l

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM 100% Disk 100% Editing

0.7

1.1

$$\sum_{j=1}^K e^{z_j}$$

0.01

0.02

def softmax(x):
 """Compute softmax values for each sets of scores in x."""
 e_x = np.exp(x)
 return e_x / e_x.sum(axis=0)

[] x = [1.3,5.1,2.2,0.7,1.1]
print(softmax(x))
print(softmax(x).sum())

All at one place

I'm Cortana. Ask me anything.

7:11 PM 27-Feb-22

+ Code + Text

RAM Disk Editing

0.7
1.1

$$\sum_{j=1}^K e^{z_j}$$

0.01
0.02

```
def softmax(x):
    """Compute softmax values for each sets of scores in x."""
    e_x = np.exp(x)
    return e_x / e_x.sum(axis=0)
```

```
[ ] x = [1.3, 5.1, 2.2, 0.7, 1.1]
print(softmax(x))
print(softmax(x).sum())
```




Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment

Share



+ Code + Text

RAM
Disk

Editing

$$\begin{bmatrix} 0.7 \\ 1.1 \end{bmatrix}$$
$$\sum_{j=1}^K e^{z_j}$$
$$\begin{bmatrix} 0.01 \\ 0.02 \end{bmatrix}$$

```
def softmax(x):  
    """Compute softmax values for each sets of scores in x."""  
    e_x = np.exp(x)  
    return e_x / e_x.sum(axis=0)
```

```
[ ] x = [1.3, 5.1, 2.2, 0.7, 1.1]  
print(softmax(x))  
print(softmax(x).sum())
```

Activation Functions.ipynb - Colab

colab.research.google.com/drive/1GJH3JuCYCgHrI6lpS0MgljZgwwfemphk#scrollTo=1yCCd/D/8HG...

colab

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

RAM Disk

Editing

+ Code + Text

1.1

0.02

[27] def softmax(x):
 """Compute softmax values for each sets of scores in x."""
 e_x = np.exp(x)
 return e_x / e_x.sum(axis=0)

▶

x = [1.3,5.1,2.2,0.7,1.1]
print(softmax(x))
print(softmax(x).sum())

⌵

[0.02019046 0.90253769 0.02966053 0.01108076 0.01653055]
1.0

All at one place

7:12 PM
27-Feb-22

Activation Functions.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings Profile

+ Code + Text

RAM Disk Editing

```
[28] x = [1.3, 5.1, 2.2, 0.7, 1.1]
print(softmax(x))
print(softmax(x).sum())

[0.02019046 0.90253769 0.04966053 0.01168076 0.01653055]
1.0
```

All at one place

2 cells hidden

```
z=np.arange(-6,6,0.01)

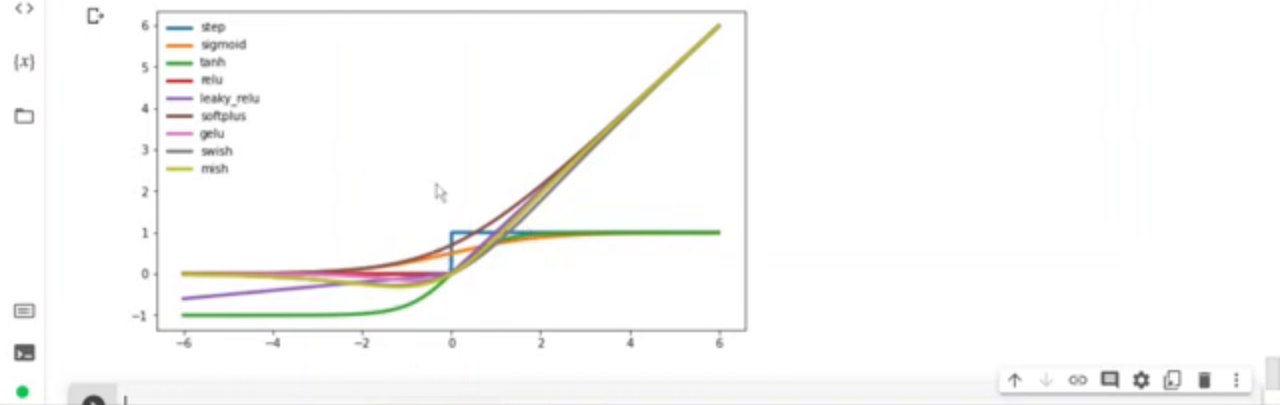
fig, ax = plt.subplots(figsize=(9, 5))

ax.plot(z, step_output, linewidth=3, label="step")
ax.plot(z, sigmoid(z), linewidth=3, label="sigmoid")
ax.plot(z, tanh(z), linewidth=3, label="tanh")
ax.plot(z, relu_output, linewidth=3, label="relu")
ax.plot(z, leaky_relu_output, linewidth=3, label="leaky_relu")
ax.plot(z, softplus(z), linewidth=3, label="softplus")
ax.plot(z, gelu(z), linewidth=3, label="gelu")
ax.plot(z, swish(z), linewidth=3, label="swish")
ax.plot(z, mish(z), linewidth=3, label="mish")

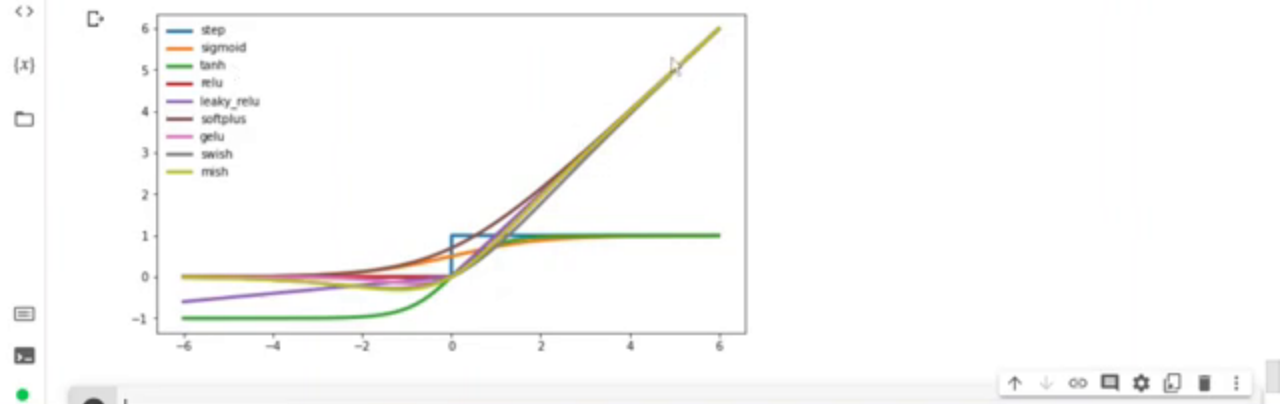
ax.legend(loc="upper left", frameon=False)
fig.show()
```

[]

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



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



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

ax.plot(z, step_output, linewidth=3, label="step")
ax.plot(z, sigmoid(z), linewidth=3, label="sigmoid")
ax.plot(z, tanh(z), linewidth=3, label="tanh")
ax.plot(z, relu_output, linewidth=3, label="relu")
ax.plot(z, leaky_relu_output, linewidth=3, label="leaky_relu")
ax.plot(z, softplus(z), linewidth=3, label="softplus")
ax.plot(z, gelu(z), linewidth=3, label="gelu")
ax.plot(z, swish(z), linewidth=3, label="swish")
ax.plot(z, mish(z), linewidth=3, label="mish")

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



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

ax.plot(z, step_output, linewidth=3, label="step")
ax.plot(z, sigmoid(z), linewidth=3, label="sigmoid")
ax.plot(z, tanh(z), linewidth=3, label="tanh")
ax.plot(z, relu_output, linewidth=3, label="relu")
ax.plot(z, leaky_relu_output, linewidth=3, label="leaky_relu")
ax.plot(z, softplus(z), linewidth=3, label="softplus")
ax.plot(z, gelu(z), linewidth=3, label="gelu")
ax.plot(z, swish(z), linewidth=3, label="swish")
ax.plot(z, mish(z), linewidth=3, label="mish")

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

