

22/8

## 5. Study Of Activation Functions and its roles

Aim: To study activation used in neural network and analyse their role

Algorithm:

\* Define common AF

\* Sigmoid

\* Hyperbolic Tangent

\* Rectified Linear Unit

\* Leaky RELU

\* Softmax

$$\frac{1}{1+e^{-x}} \quad (0,1)$$

\* When a range of input values

\* Apply this with all of them

\* Plot the outputs

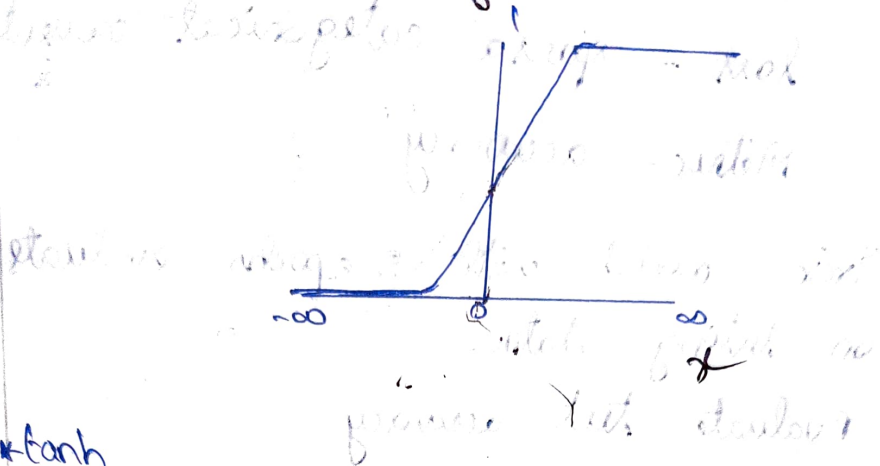
\* Observe how each func handles linearity.



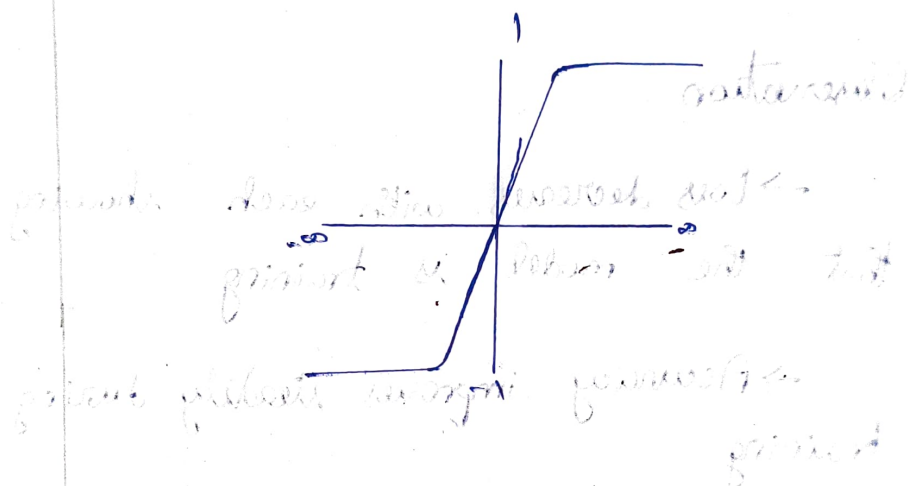
# PLOTS

other inputs

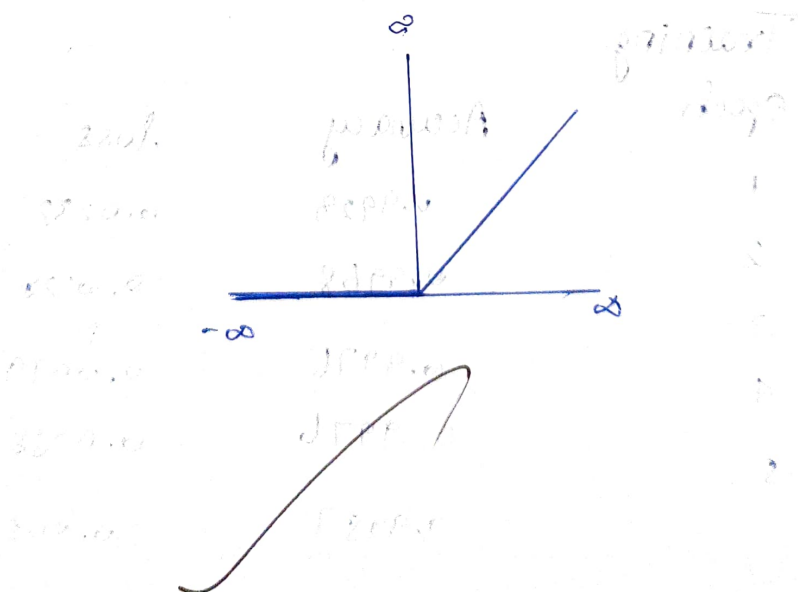
\*Sigmoid: sharp at center  
 gradient decreases rapidly  
 saturates above & below linear



\*tanh



\*RELU:



ReLU is the most common  
 because it is simple and  
 does not saturate for positive  
 values, allowing for a constant  
 gradient flow.

Observation:

\* Sigmoid maps input to range (0,1)  
useful for probabilities

\* Tanh maps to range (-1, 1)

\* RELU outputs +ve value dir.

\* L-Relu outputs +ve values.

\* L-Relu allows small -ve values  
preventing the dying RELU prob

\* Softmax converts output to  
prob distribution

Result:

\* Therefore study of activation  
and its role are completed.

⑧

Testing:

For  $x=5$ :

Sigmoid = 0.993

Tanh: 0.999

Relu = 5

L.R = 5

Range:

Sigmoid - (0,1)

Tanh - (-1,1)

Relu = [0, ∞)

Leaky Relu: (0, ∞)

Softmax: (0,1)

Results:

$x$	sigmoid	tanh	Relu	L.R
5	0.9933	0.999	5	5
7	0.999	0.999	7	7
-1	0.268	-0.761	0	0

15.ipynb (7) - JupyterLab

10.1.38.19/user/ra2311047010019/lab/tree/DEEP%20LEARNING/I5.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name	Last Modified
data	25 days ago
EX1.ipynb	last month
L_3.ipynb	25 days ago
L2.ipynb	25 days ago
L3.ipynb	25 days ago
L4.ipynb	18 days ago
I5.ipynb	28 seconds ago
L6.ipynb	28 seconds ago
Logistic reg...	last month

Logistic reg X L\_3.ipynb X L4.ipynb X jupyter-ra2 X L2.ipynb X EX1.ipynb X I5.ipynb X L6.ipynb X jupyter-ra2 X

Code

Notebook Python 3 (ipykernel)

```
[2]: import torch
import torch.nn.functional as F

Input: tensor([-2., -1., 0., 1., 2.])
Sigmoid: tensor([0.1192, 0.2689, 0.5000, 0.7311, 0.8808])
Tanh: tensor([-0.9640, -0.7616, 0.0000, 0.7616, 0.9640])
ReLU: tensor([0., 0., 0., 1., 2.])
Leaky ReLU: tensor([-0.2000, -0.1000, 0.0000, 1.0000, 2.0000])
Softmax: tensor([0.0117, 0.0317, 0.0861, 0.2341, 0.6364])

[5]: x = torch.tensor([-2.0, -1.0, 0.0, 1.0, 2.0])

[6]: sigmoid = torch.sigmoid(x)
tanh = torch.tanh(x)
relu = F.relu(x)
leaky_relu = F.leaky_relu(x, negative_slope=0.1)
softmax = F.softmax(x, dim=0) # Softmax over the tensor elements

[7]: print("Input:", x)
print("Sigmoid:", sigmoid)
print("Tanh:", tanh)
print("ReLU:", relu)
print("Leaky ReLU:", leaky_relu)
print("Softmax:", softmax)

Input: tensor([-2., -1., 0., 1., 2.])
Sigmoid: tensor([0.1192, 0.2689, 0.5000, 0.7311, 0.8808])
```

Simple 2 7 Python 3 (ipykernel) | Idle Mem: 420.48 MB

Mode: Command Ln 1, Col 1 I5.ipynb 0

Type here to search

9:06 AM 9/9/2025

15.ipynb (7) - JupyterLab

+

← → ↺

Not secure 10.1.38.19/user/ra2311047010019/lab/tree/DEEP%20LEARNING/I5.ipynb

☆ ⓘ ⋮

File Edit View Run Kernel Tabs Settings Help

+

📁

⬆

🔄

Filter files by name 🔍

/ DEEP LEARNING /

Name	Last Modified
data	25 days ago
EX1.ipynb	last month
L_3.ipynb	25 days ago
L2.ipynb	25 days ago
L3.ipynb	25 days ago
L4.ipynb	18 days ago
<b>I5.ipynb</b>	38 seconds ago
L6.ipynb	38 seconds ago
Logistic reg...	last month

Logistic reg

L\_3.ipynb

L4.ipynb

jupyter-ra2

L2.ipynb

EX1.ipynb

**I5.ipynb**

L6.ipynb

jupyter-ra2

+

📄 + ✂ 📄 📄 ▶ ⏮ ⏭ Code ⌵

Notebook ⓘ Python 3 (ipykernel) ⓪

```
Input: tensor([-2., -1., 0., 1., 2.])
Sigmoid: tensor([0.1192, 0.2689, 0.5000, 0.7311, 0.8808])
Tanh: tensor([-0.9640, -0.7616, 0.0000, 0.7616, 0.9640])
ReLU: tensor([0., 0., 0., 1., 2.])
Leaky ReLU: tensor([-0.2000, -0.1000, 0.0000, 1.0000, 2.0000])
Softmax: tensor([0.0117, 0.0317, 0.0861, 0.2341, 0.6364])

[ ]:
```

Simple 2 7 ⓘ Python 3 (ipykernel) | Idle Mem: 422.71 MB

Mode: Command ⚠ Ln 1, Col 1 I5.ipynb 0

Type here to search

📁

📄

🔍

📧

🔥

9:12 AM 9/9/2025