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

  * Sigmoid

  * Hyperbolic Tangent

  * Rectified Linear Unit

  * Leaky RELU

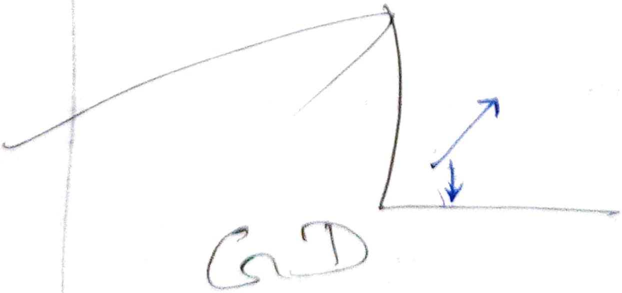
  * Softmax

* then a range of input values

* Apply this with all of them

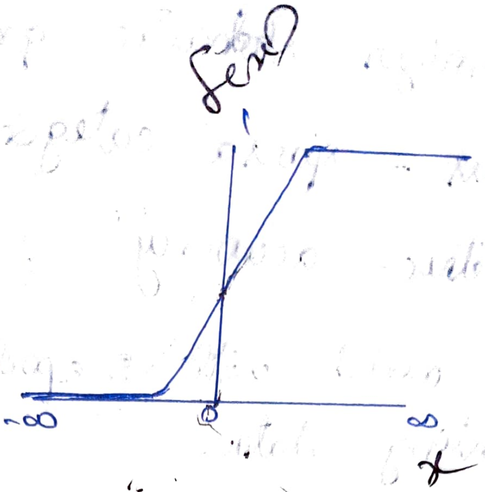
* plot the outputs

* Observe how each func handles linearity.

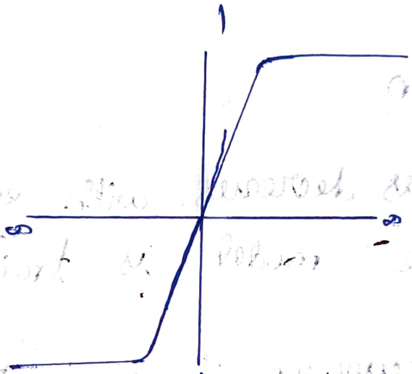


PLOTS

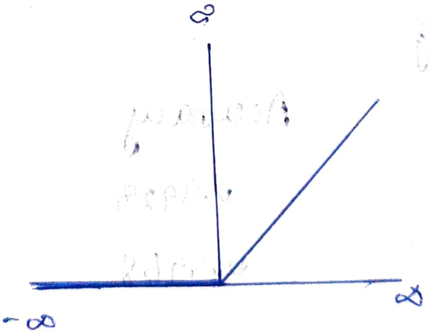
*Sigmoid:

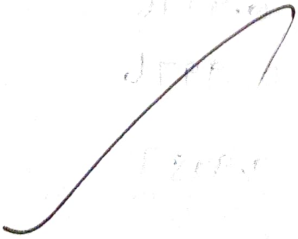


*tanh



*RELU:





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.

Testing:

For $x=5$

Sigmoid = 0.993

Tanh: 0.999

Relue : 5

L.R = 5

Range:

Sigmoid - (0,1)

Tanh - (-1,1)

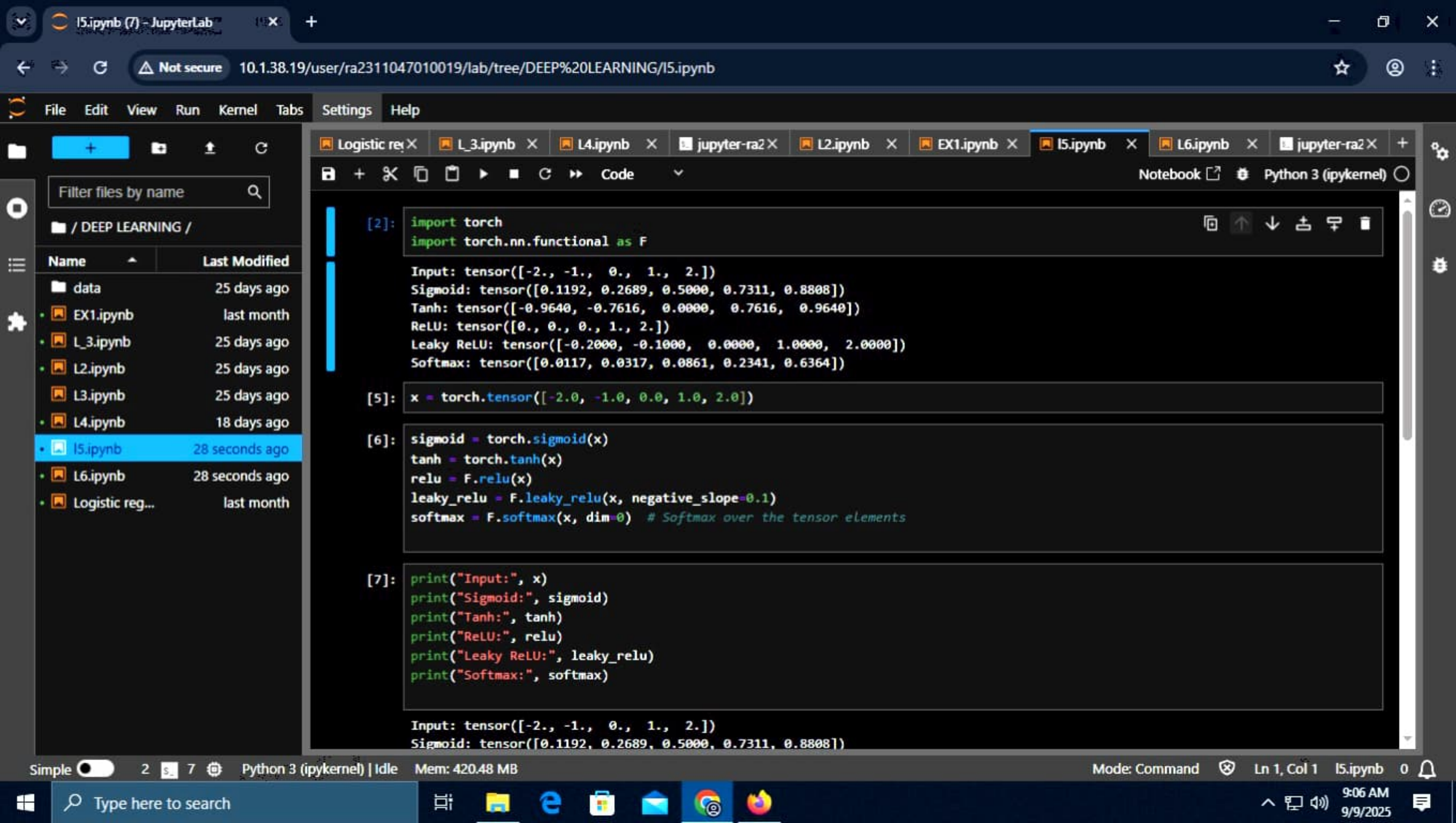
Relu = [0,∞)

Leaky Relu : [0,∞)

Softmax : (0,1)

Results:

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



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
l5.ipynb	28 seconds ago
L6.ipynb	28 seconds ago
Logistic reg...	last month

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

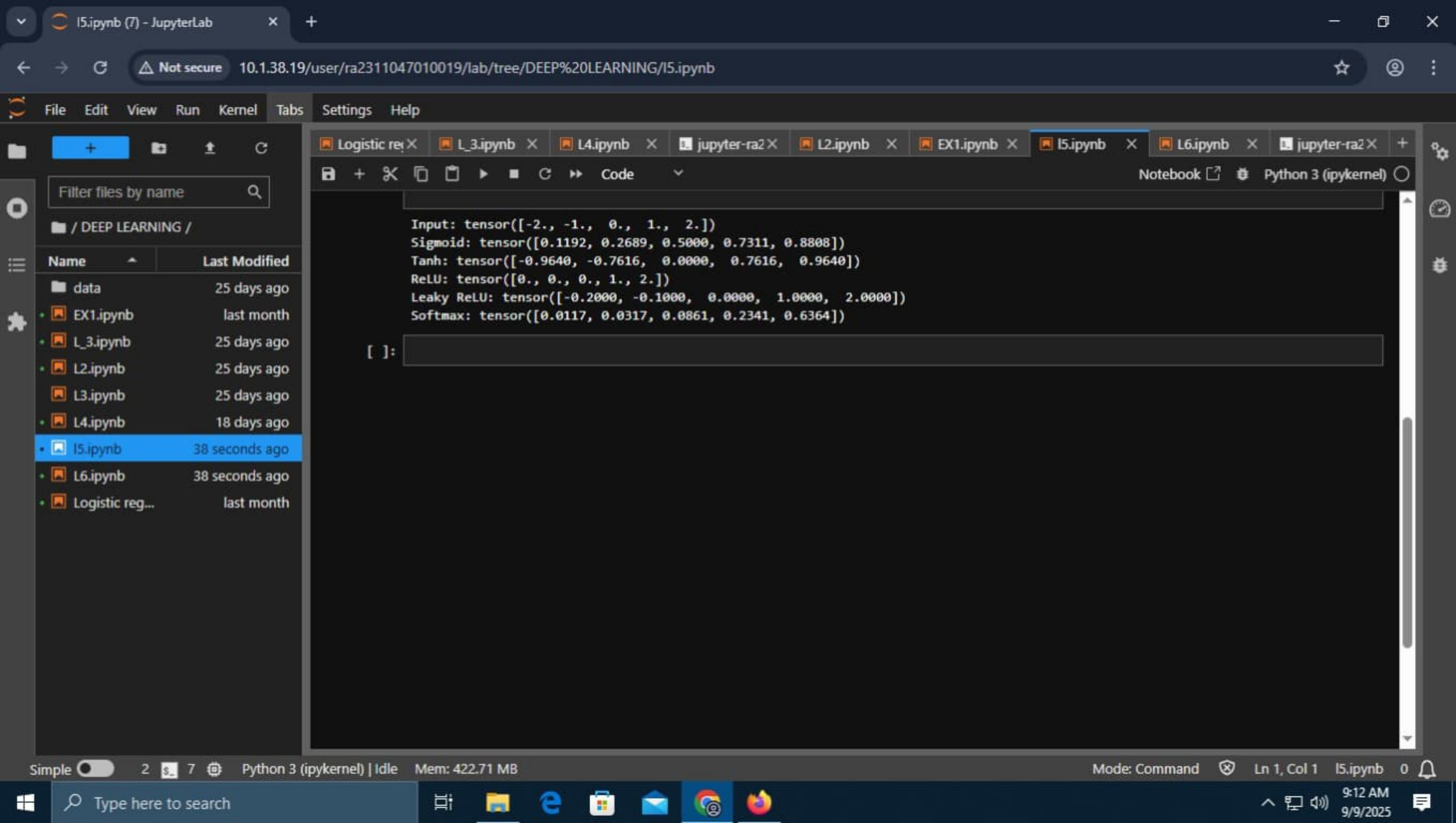
```
Input: tensor([-2., -1.,  0.,  1.,  2.])
Sigmoid: tensor([0.1192, 0.2689, 0.5000, 0.7311, 0.8808])
Tanh: tensor([-0.9640, -0.7616,  0.0000,  0.7616,  0.9640])
ReLU: tensor([0., 0., 0., 1., 2.])
Leaky ReLU: tensor([-0.2000, -0.1000,  0.0000,  1.0000,  2.0000])
Softmax: tensor([0.0117, 0.0317, 0.0861,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



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	38 seconds ago
L6.ipynb	38 seconds ago
Logistic reg...	last month

```
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])
```

```
[ ]:
```