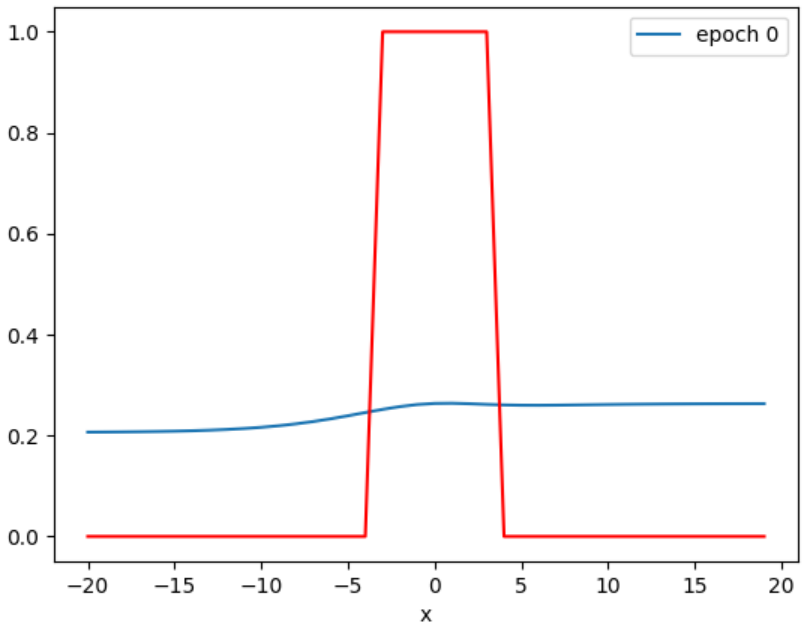
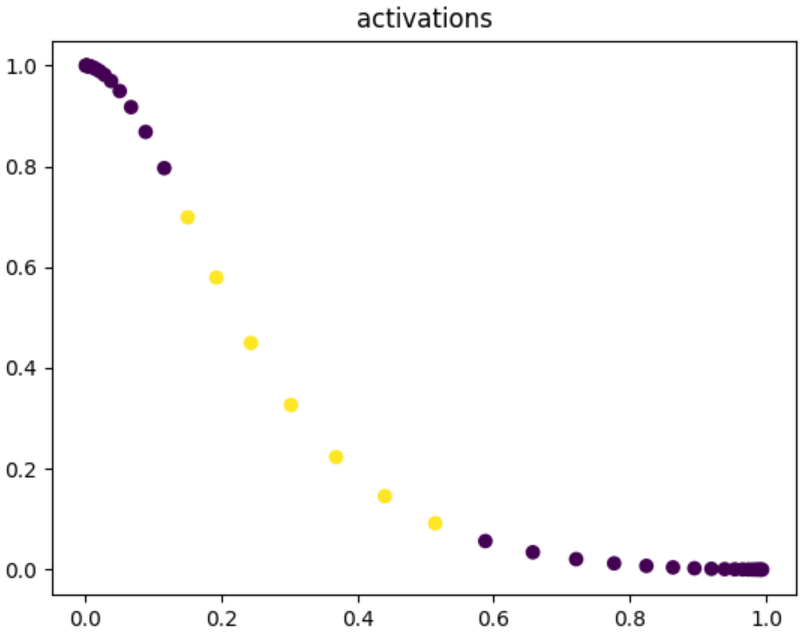
03a_simple1hiddenlayer.py (Original)

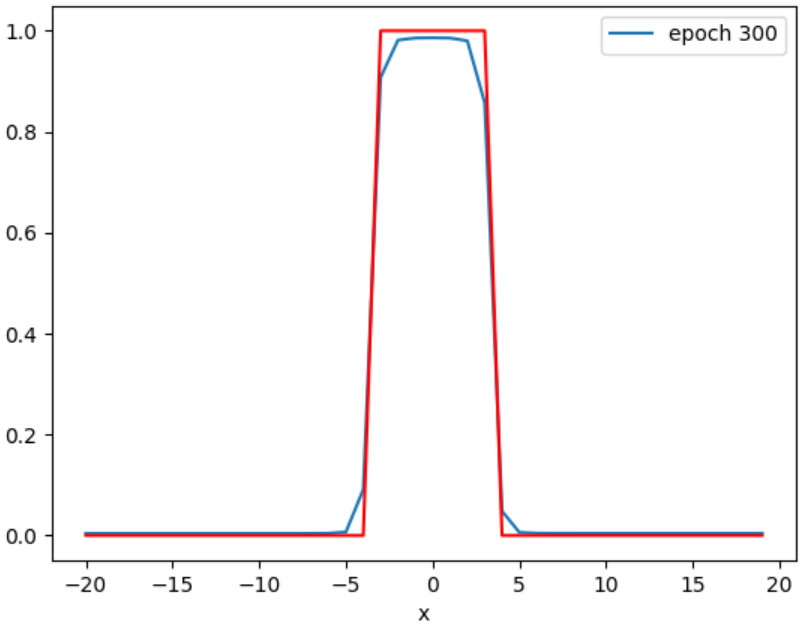
Simple 1-hidden-layer NN
Data: X in [-20, 20], H=2, SGD, 1000 epochs, Cross-Entropy
PyTorch, manual_seed=0

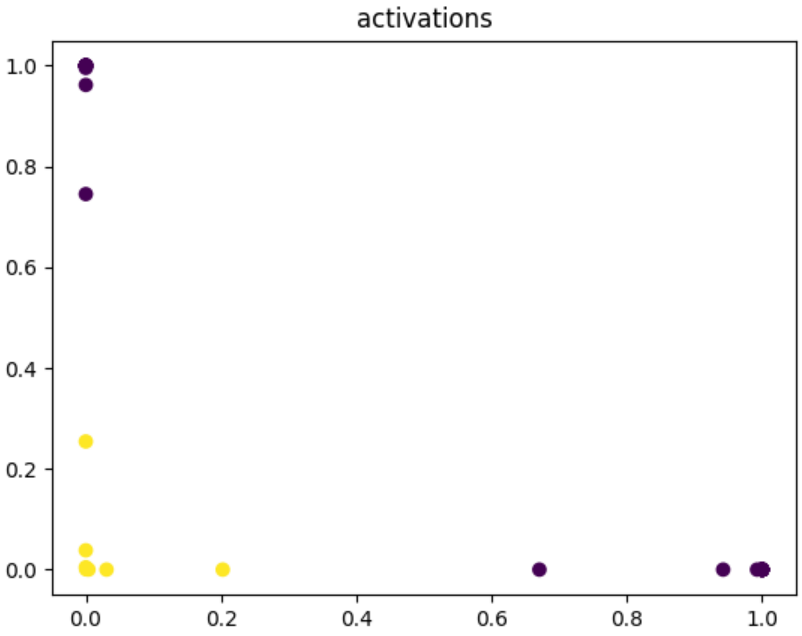
Script Output

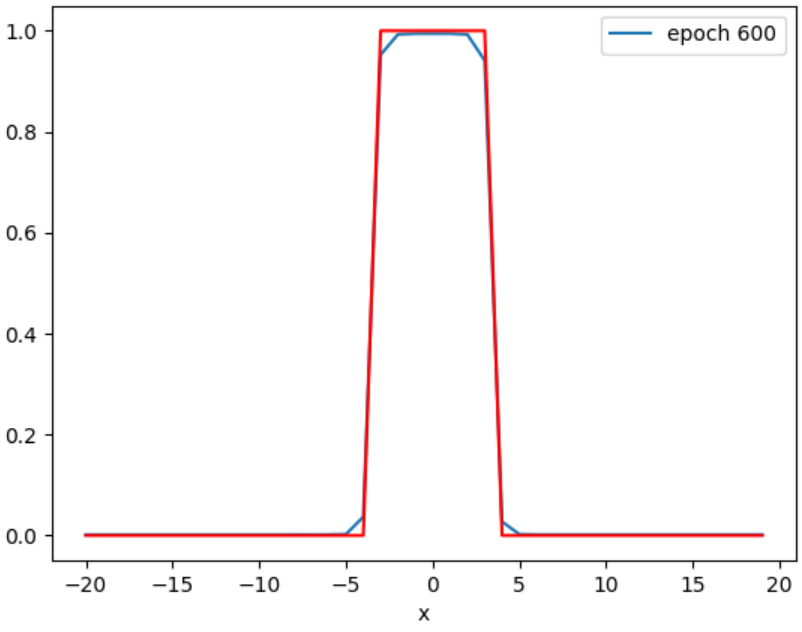
Prediction for x=0.0: tensor([0.9969], grad_fn=<SigmoidBackward0>)
Predictions for X_: tensor([[0.9969],
    [0.9963],
    [0.9680]], grad_fn=<SigmoidBackward0>)
Thresholded predictions: tensor([[True],
    [True],
    [True]])

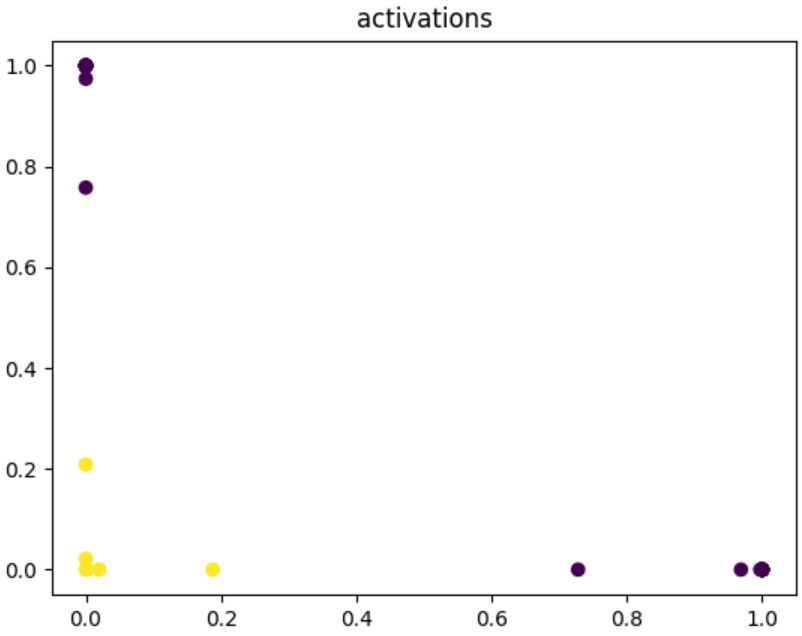


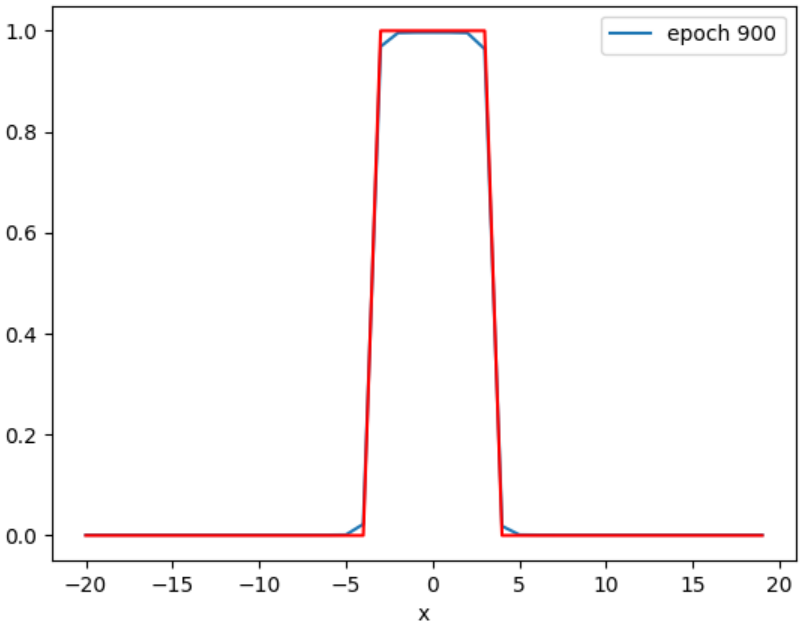

activations
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

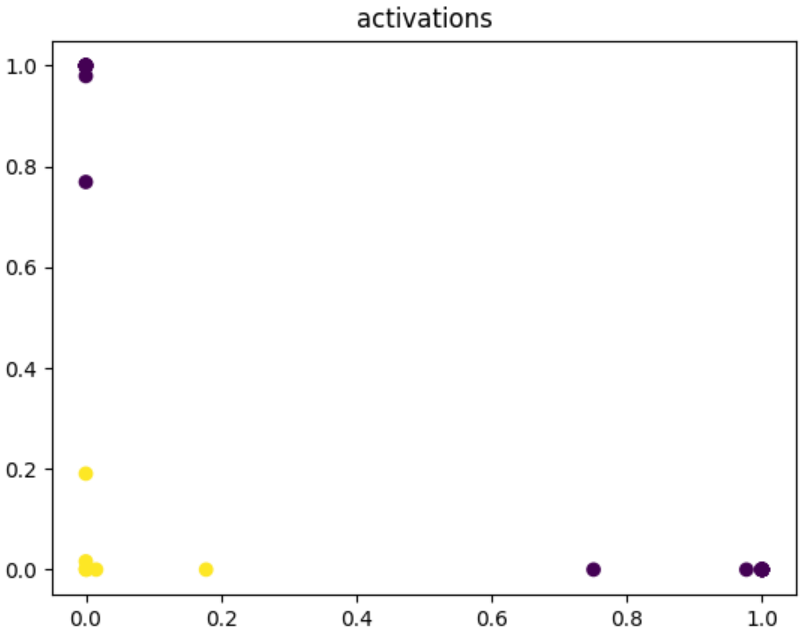

epoch 300
1.0
0.8
0.6
0.4
0.2
0.0
−20
−15
−10
−5
0
5
10
15
20
x

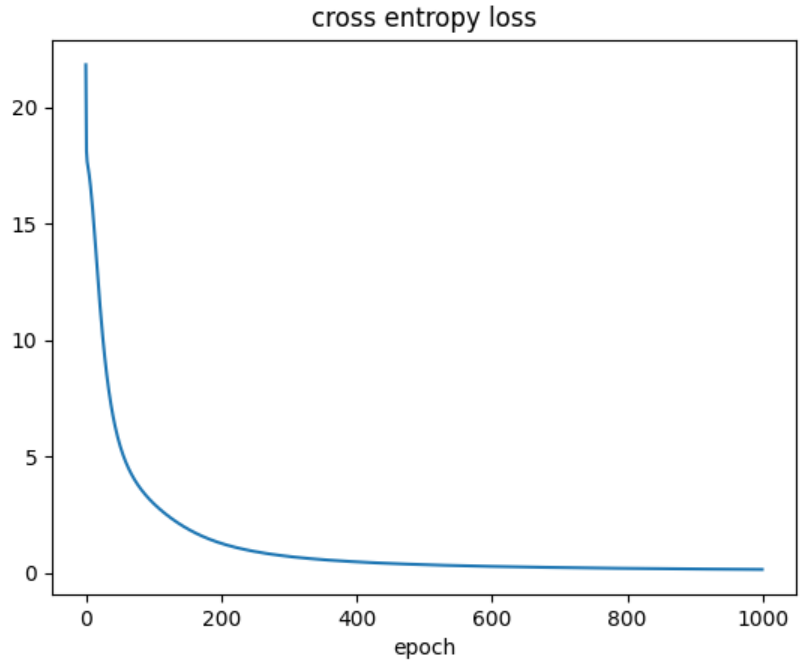

activations
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


epoch 600
1.0
0.8
0.6
0.4
0.2
0.0
−20
−15
−10
−5
0
5
10
15
20
x


activations
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


epoch 900
1.0
0.8
0.6
0.4
0.2
0.0
−20
−15
−10
−5
0
5
10
15
20
x


activations
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

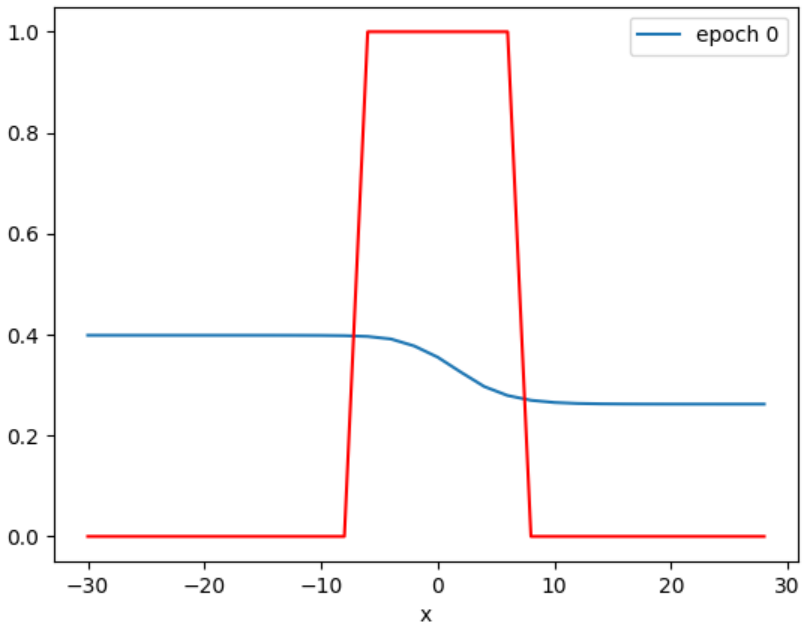

cross entropy loss
20
15
10
5
0
0
200
400
600
800
1000
epoch

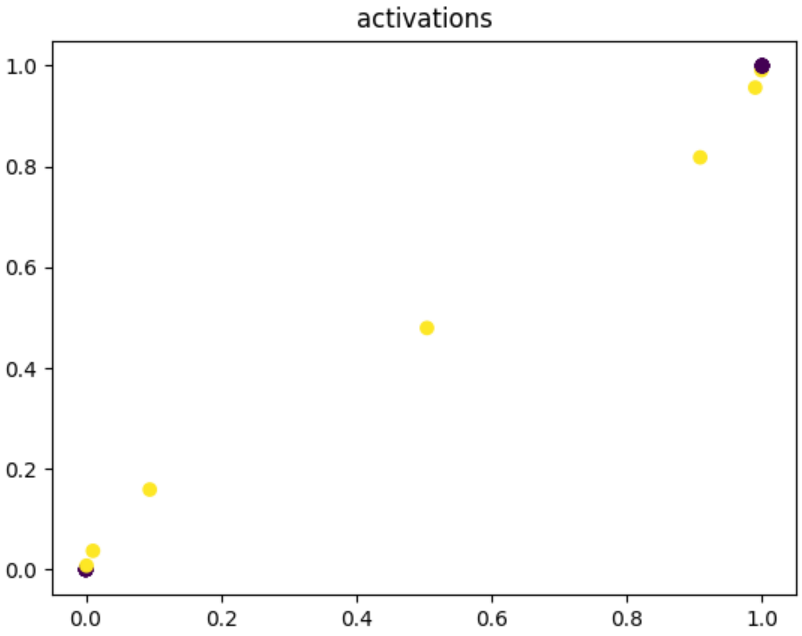
03a_simple1hiddenlayer_rev02.py (rev.02)

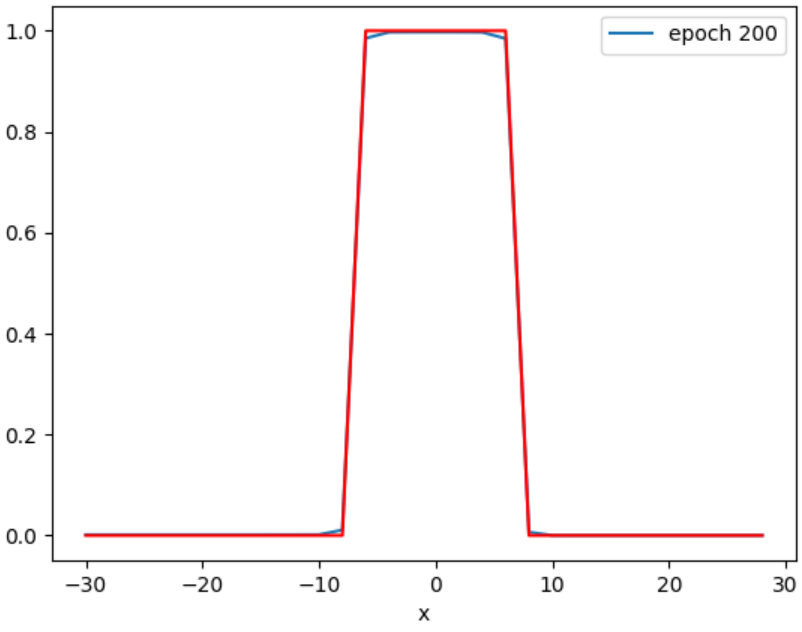
Simple 1-hidden-layer NN
Data: X in [-30, 30], H=4, Adam, 800 epochs, Cross-Entropy
PyTorch, manual_seed=42

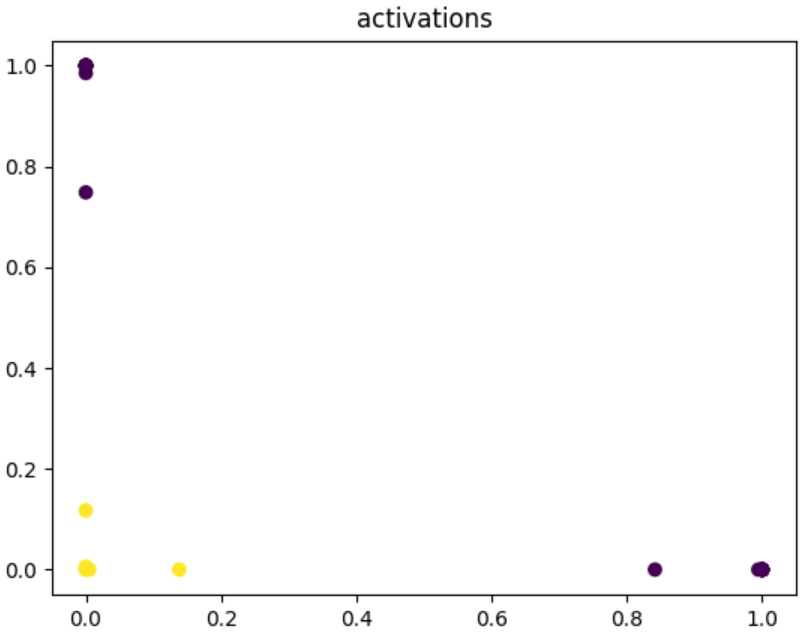
Script Output

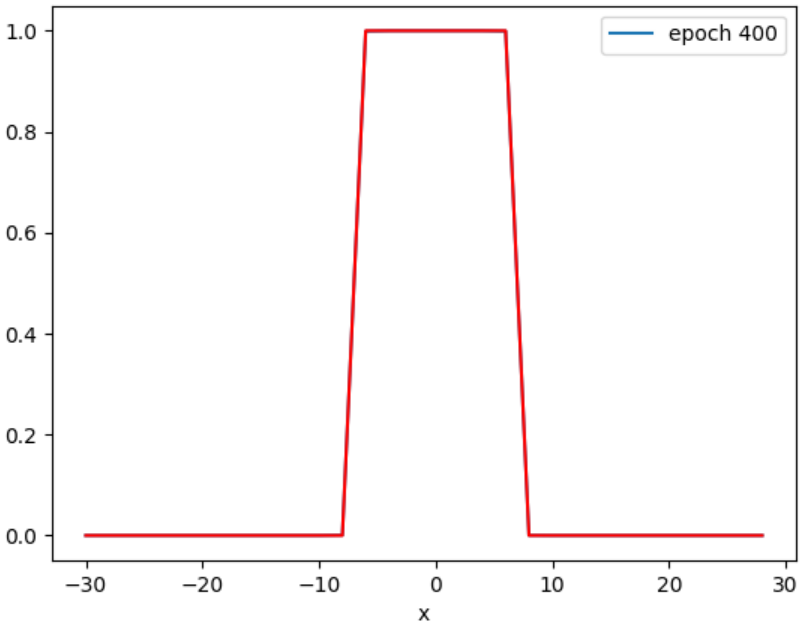
Prediction for x=5.0: tensor([1.0000], grad_fn=<SigmoidBackward0>)
Predictions for X_: tensor([[1.0000e+00],
[1.3452e-09],
[8.7578e-08]], grad_fn=<SigmoidBackward0>)
Thresholded predictions: tensor([[True],
[False],
[False]])

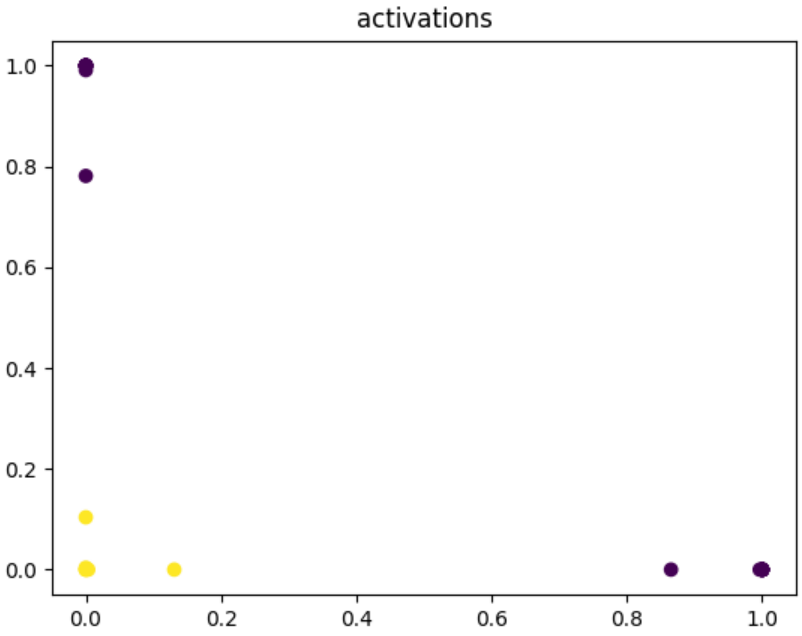


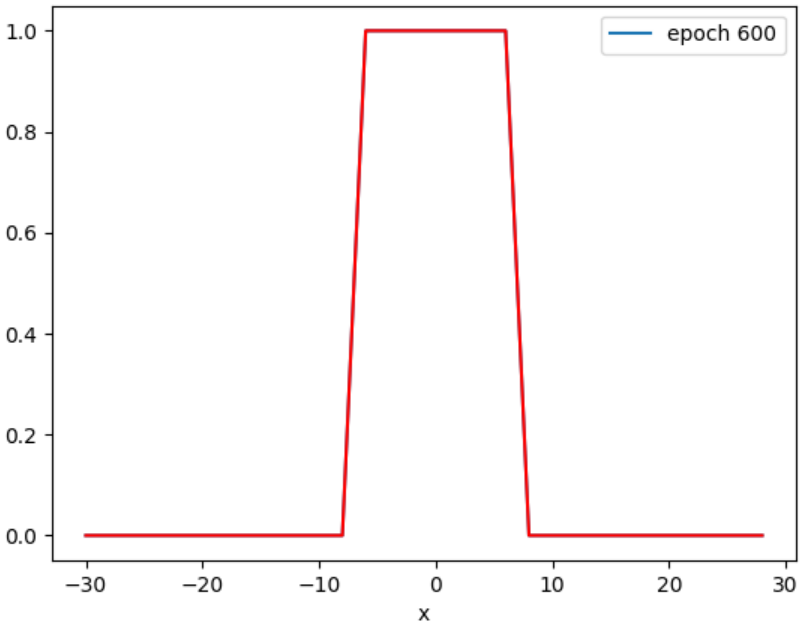

activations
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

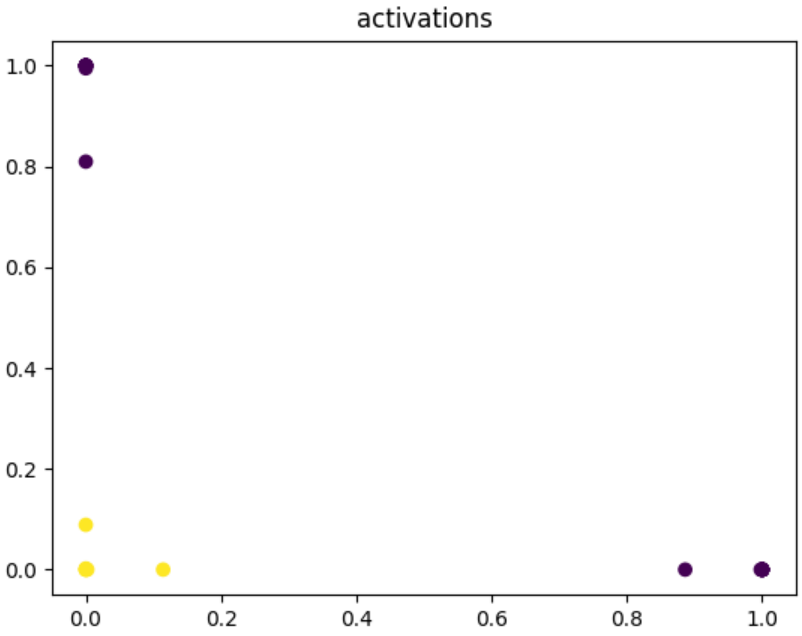

epoch 200
1.0
0.8
0.6
0.4
0.2
0.0
−30
−20
−10
0
10
20
30
x

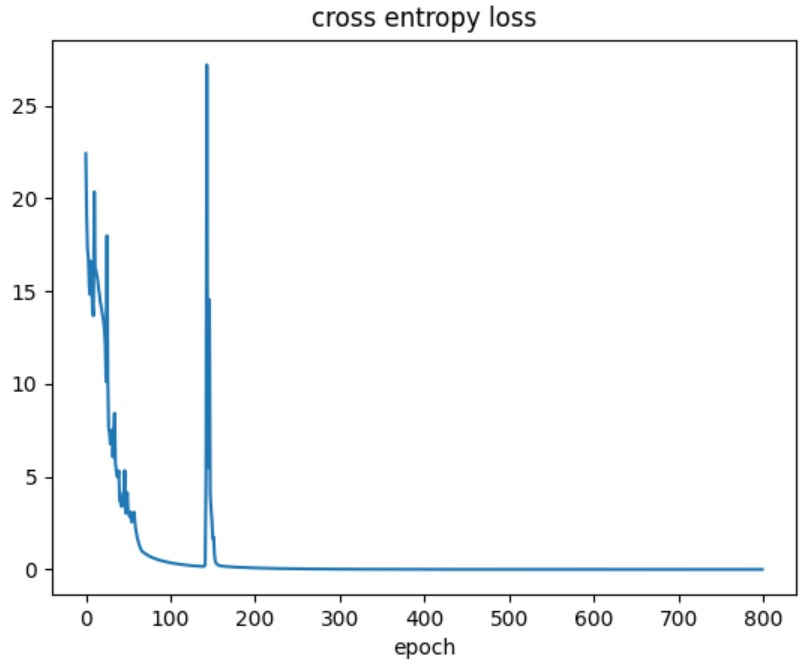

activations
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


epoch 400
1.0
0.8
0.6
0.4
0.2
0.0
−30
−20
−10
0
10
20
30
x


activations
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


epoch 600
1.0
0.8
0.6
0.4
0.2
0.0
−30
−20
−10
0
10
20
30
x

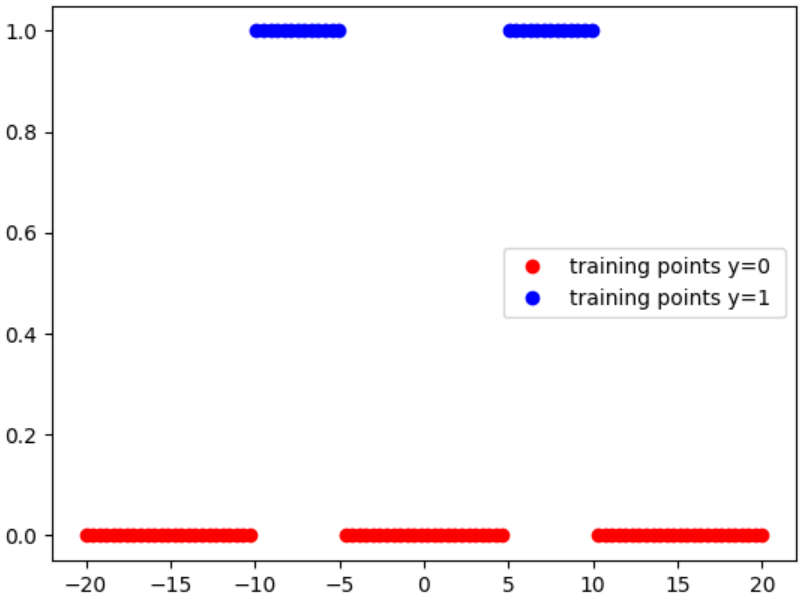

activations
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

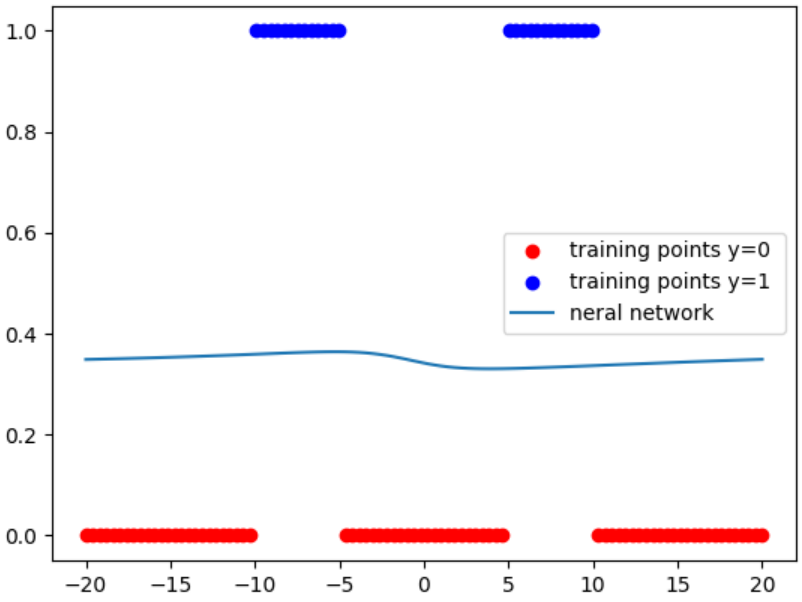

cross entropy loss
25
20
15
10
5
0
0
100
200
300
400
500
600
700
800
epoch

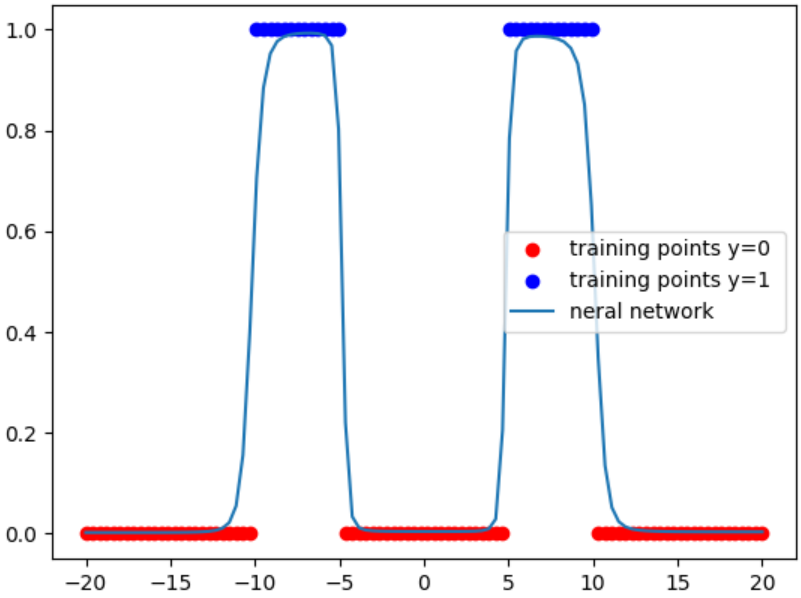
03b_multiple_neurons.py (Original)

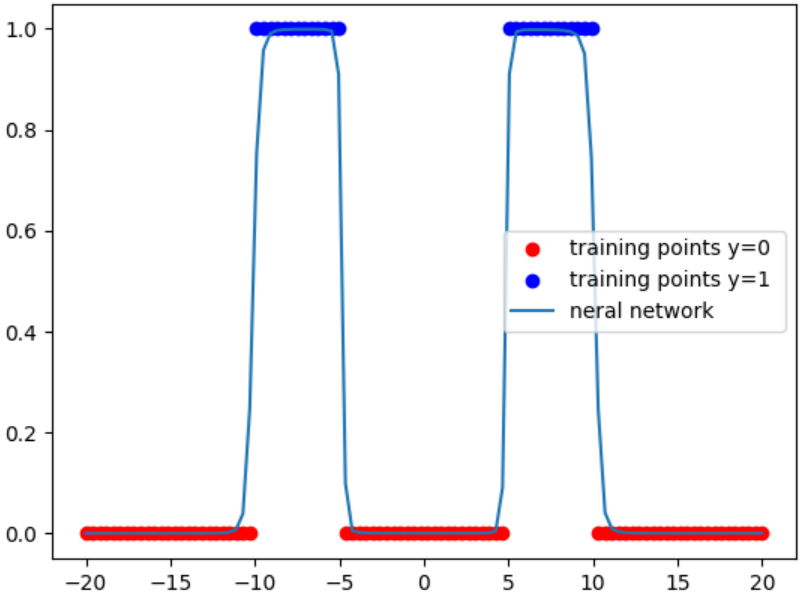
More hidden neurons
Data: X in [-20, 20], H=2, SGD, 1000 epochs, Cross-Entropy
PyTorch, manual_seed=0

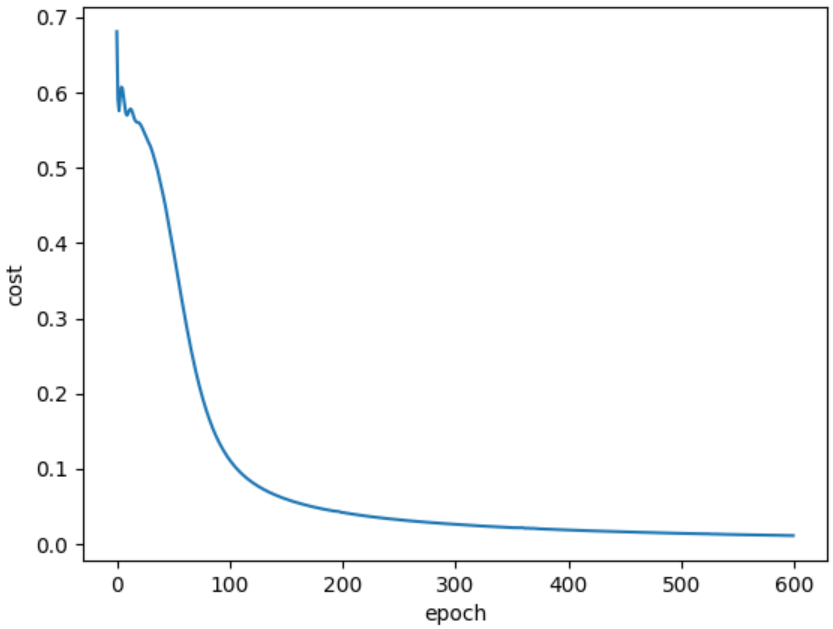
Script Output






1.0
0.8
0.6
0.4
0.2
0.0
−20
−15
−10
−5
0
5
10
15
20
training points y=0
training points y=1
neral network


1.0
0.8
0.6
0.4
0.2
0.0
−20
−15
−10
−5
0
5
10
15
20
training points y=0
training points y=1
neral network

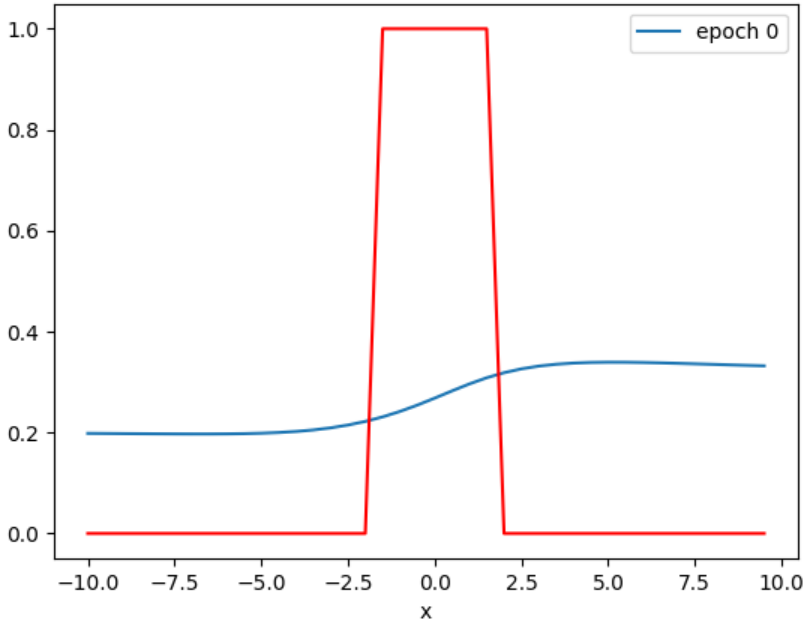

0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
cost
0
100
200
300
400
500
600
epoch

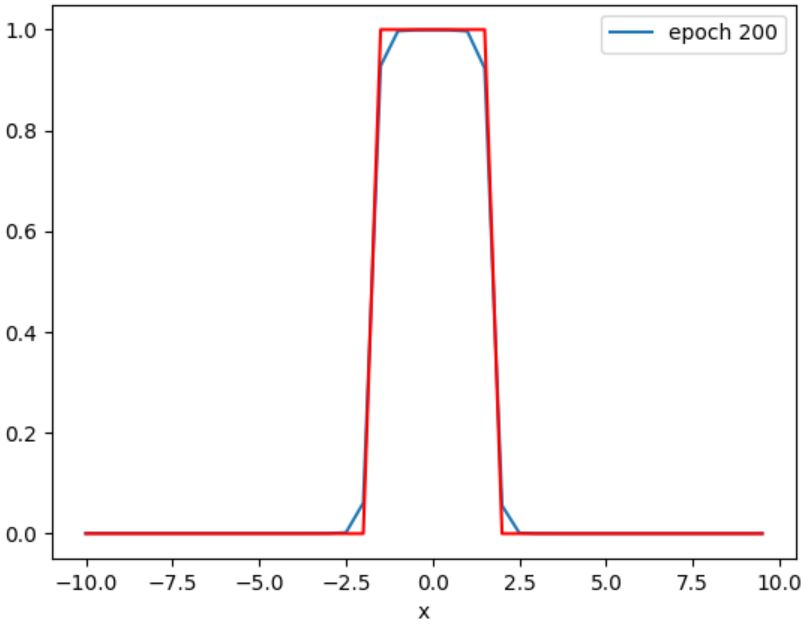
03b_multiple_neurons_rev02.py (rev.02)

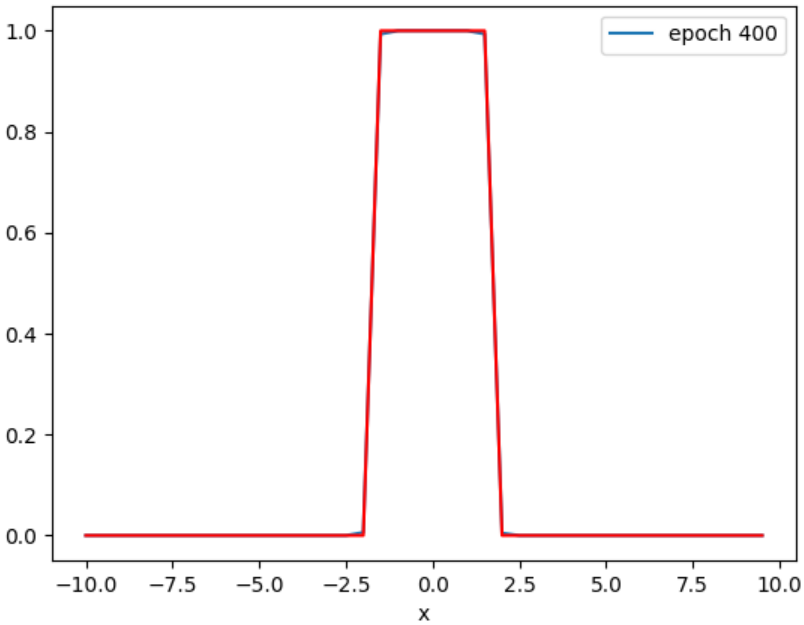
More hidden neurons
Data: X in [-10, 10], H=5, Adam, 600 epochs, Cross-Entropy
PyTorch, manual_seed=123

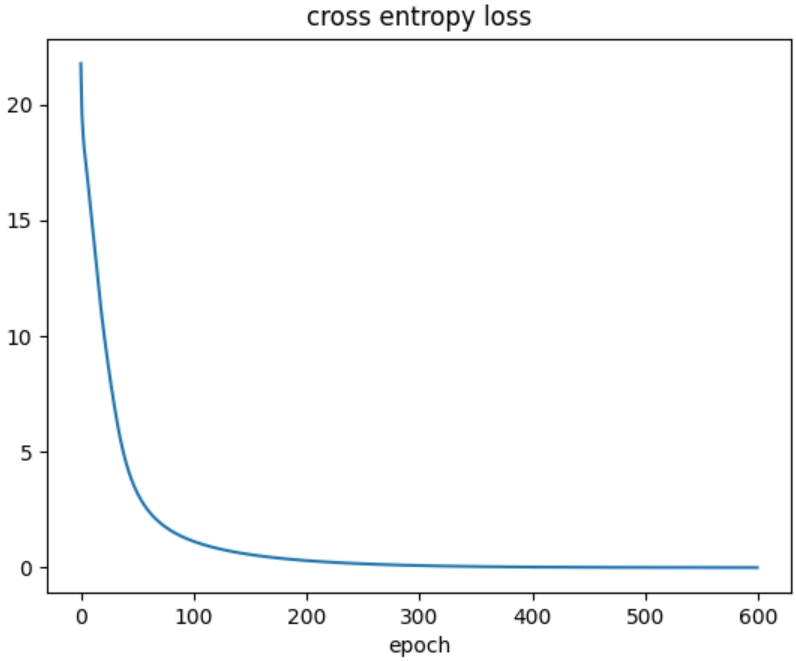
Script Output

```
Prediction for x=1.0: tensor([1.0000], grad_fn=<SigmoidBackward0>)
Predictions for X_: tensor([[1.0000e+00],
        [8.5903e-09],
        [1.9491e-07]], grad_fn=<SigmoidBackward0>)
Thresholded predictions: tensor([[ True],
        [False],
        [False]])
```




epoch 200
1.0
0.8
0.6
0.4
0.2
0.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
x


epoch 400
1.0
0.8
0.6
0.4
0.2
0.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
x


cross entropy loss
20
15
10
5
0
0
100
200
300
400
500
600
epoch

03c_xor_v2.py (Original)

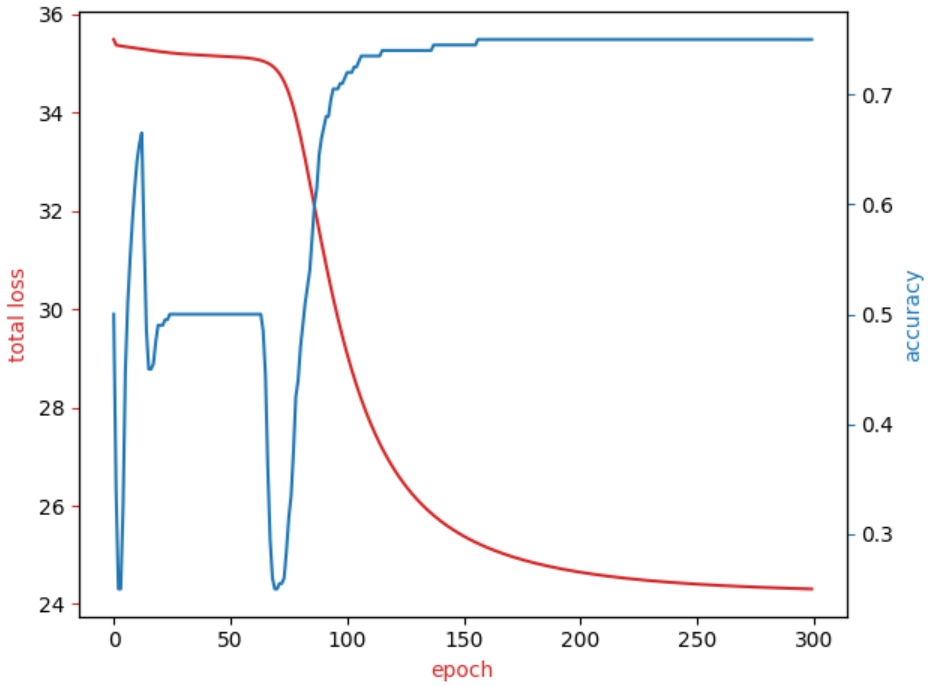
Noisy XOR, 1/2/3 neurons
Data: N=100, SGD, 500 epochs, BCELoss
PyTorch, manual_seed=0

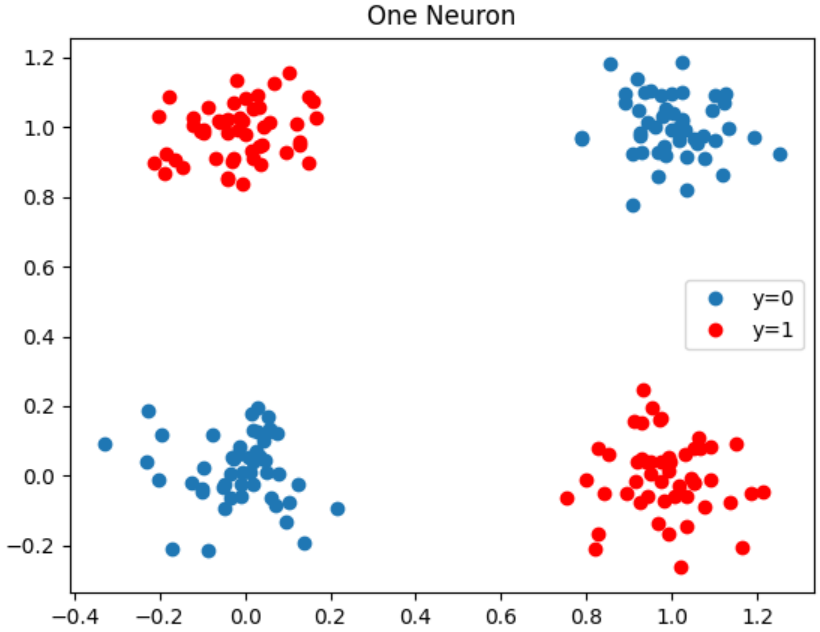
Script Output

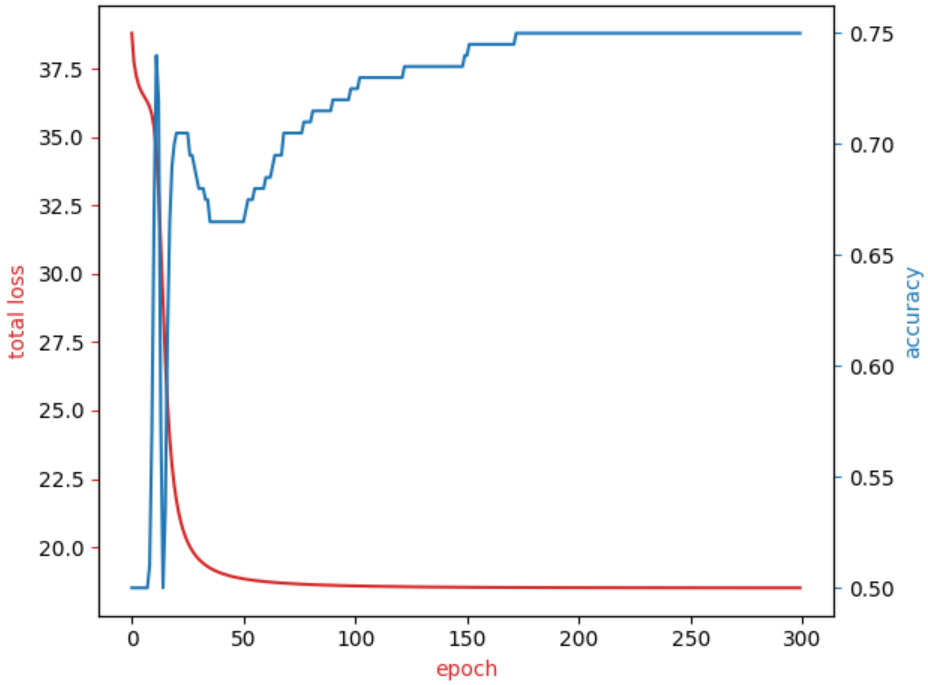
03c_xor_v2_rev02.py (rev.02)

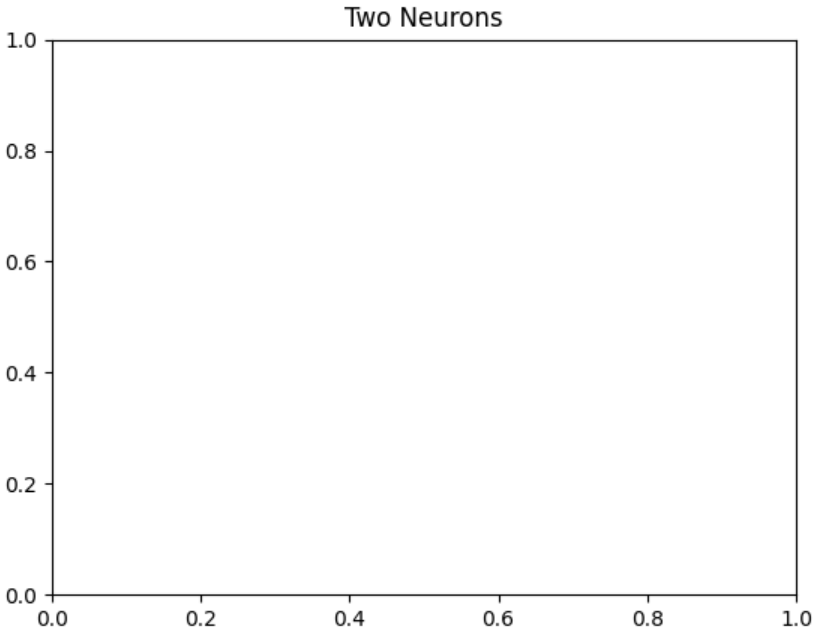
Noisy XOR, 1/2/3 neurons
Data: N=200, Adam, 300 epochs, BCELoss, batch=4
PyTorch, manual_seed=99

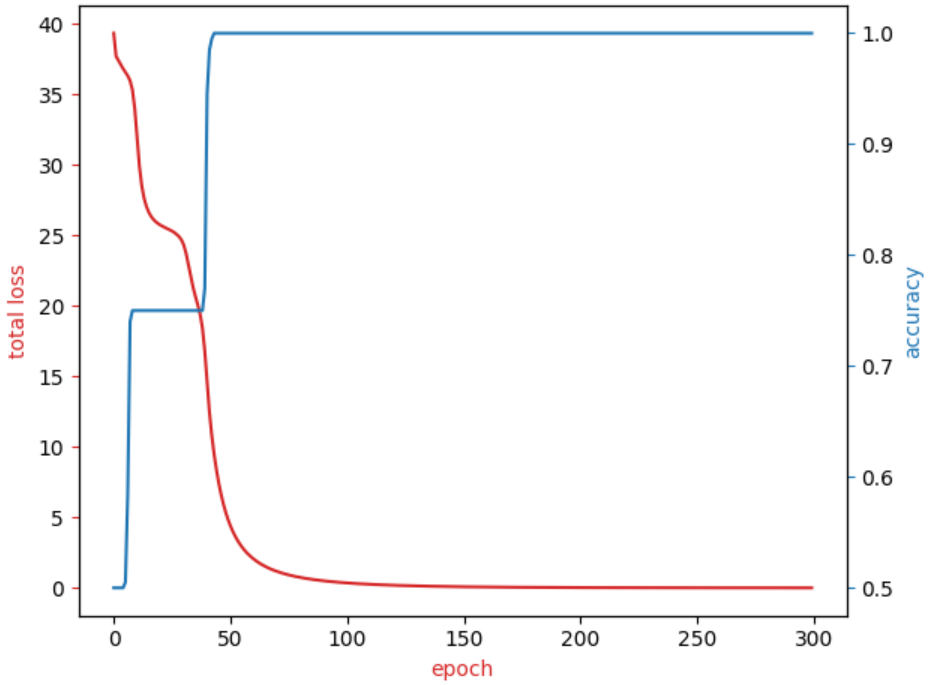
Script Output

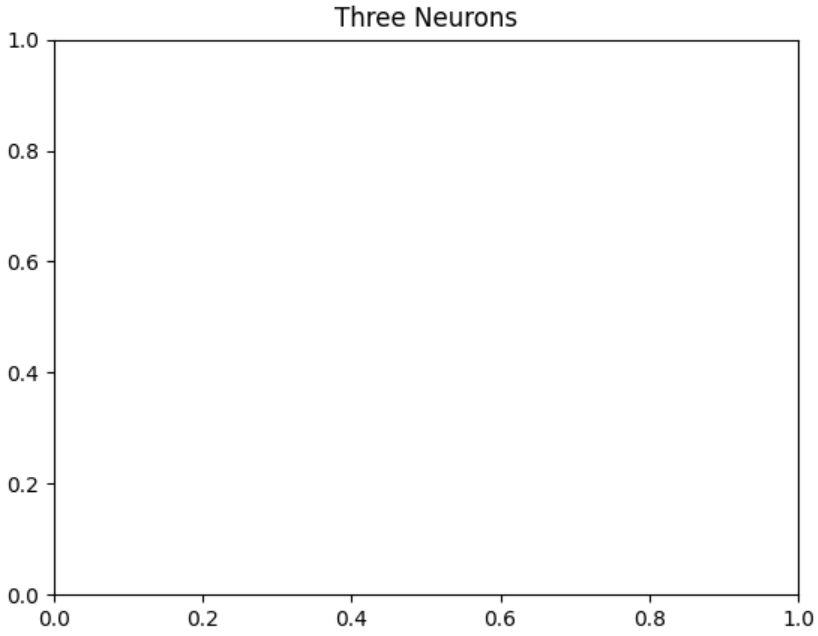





total loss
accuracy
epoch
37.5
35.0
32.5
30.0
27.5
25.0
22.5
20.0
0.75
0.70
0.65
0.60
0.55
0.50
0
50
100
150
200
250
300


Two Neurons
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


total loss
accuracy
epoch
0
5
10
15
20
25
30
35
40
0.5
0.6
0.7
0.8
0.9
1.0
0
50
100
150
200
250
300

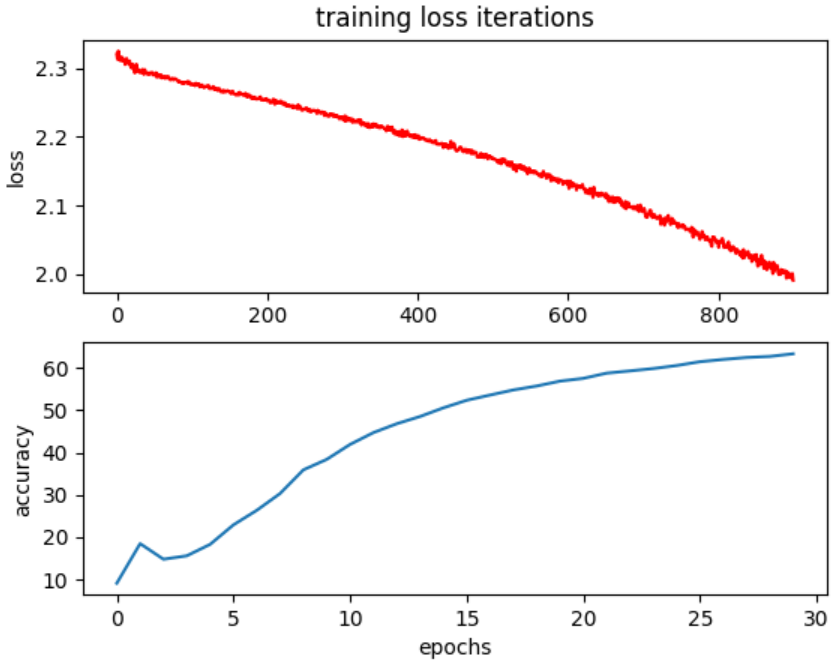

Three Neurons
0.0
0.2
0.4
0.6
0.8
1.0

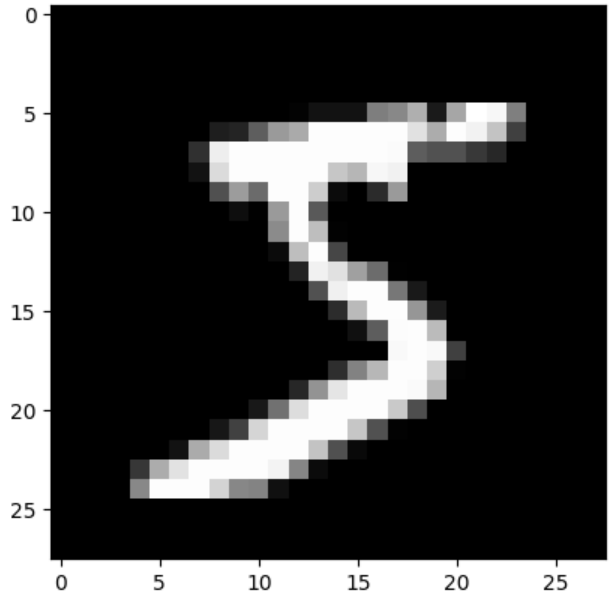
03d_one_layer_neural_network_MNIST.py (Original)

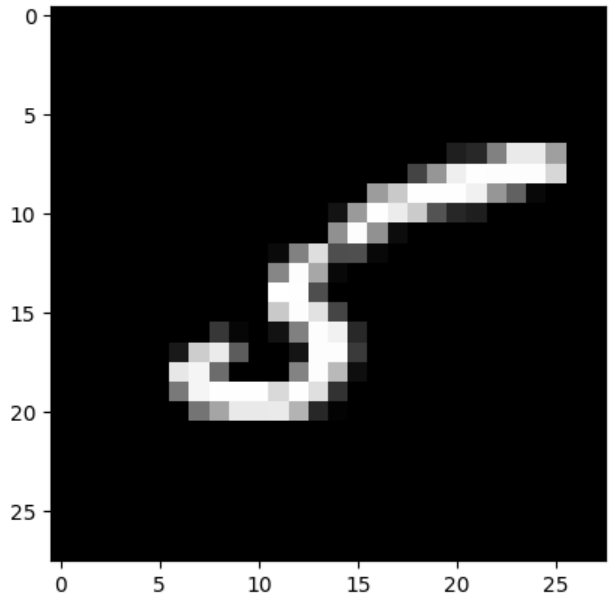
MNIST, 1 hidden layer
H=50, SGD, 2 epochs, CrossEntropyLoss
PyTorch, manual_seed=0

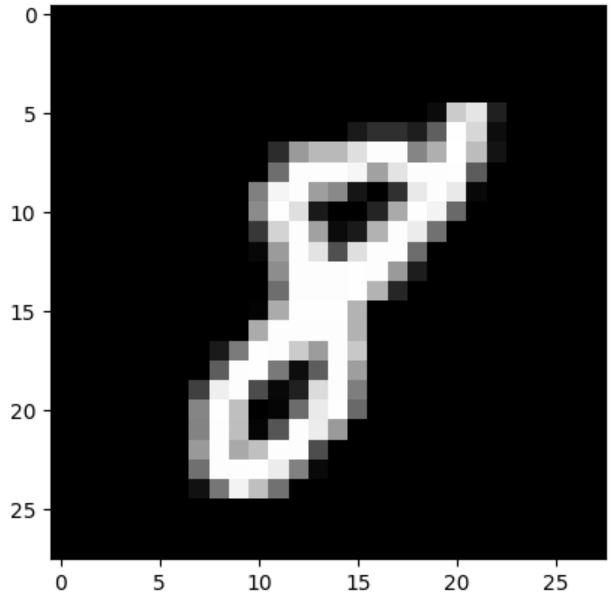
Script Output

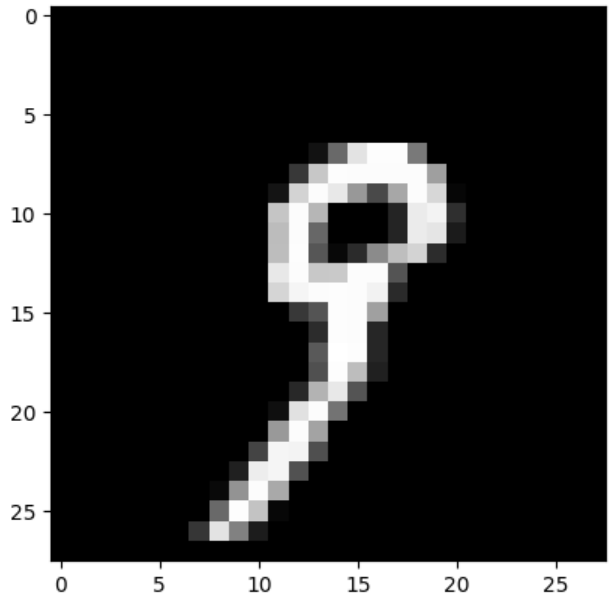
The following are the parameters for the layer 1
The size of weights: torch.Size([100, 784])
The size of bias: torch.Size([100])
The following are the parameters for the layer 2
The size of weights: torch.Size([10, 100])
The size of bias: torch.Size([10])

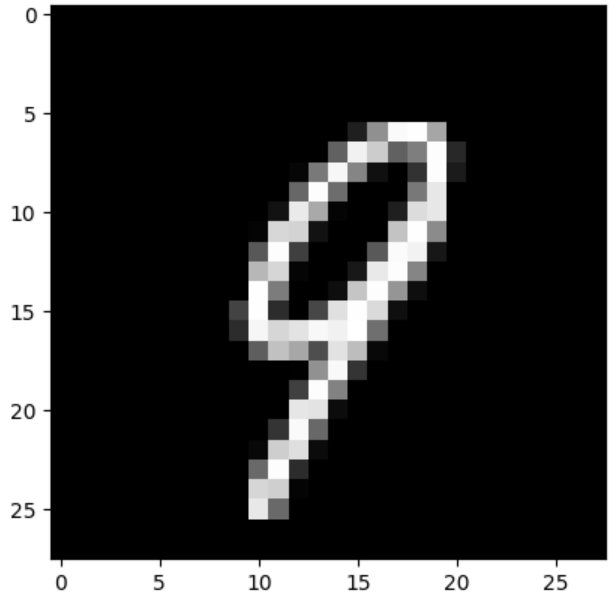



0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25

d_one_layer_neural_network_MNIST_rev02.py (rev.0

MNIST, 1 hidden layer
H=128, Adam, 3 epochs, CrossEntropyLoss
PyTorch, manual_seed=2024

Script Output

```
Epoch [1/3], Loss: 120.6589
Epoch [2/3], Loss: 51.2925
Epoch [3/3], Loss: 35.5328
Accuracy of the network on the 10000 test images: 97.40%
```

03e_activationfuction_v2.py (Original)

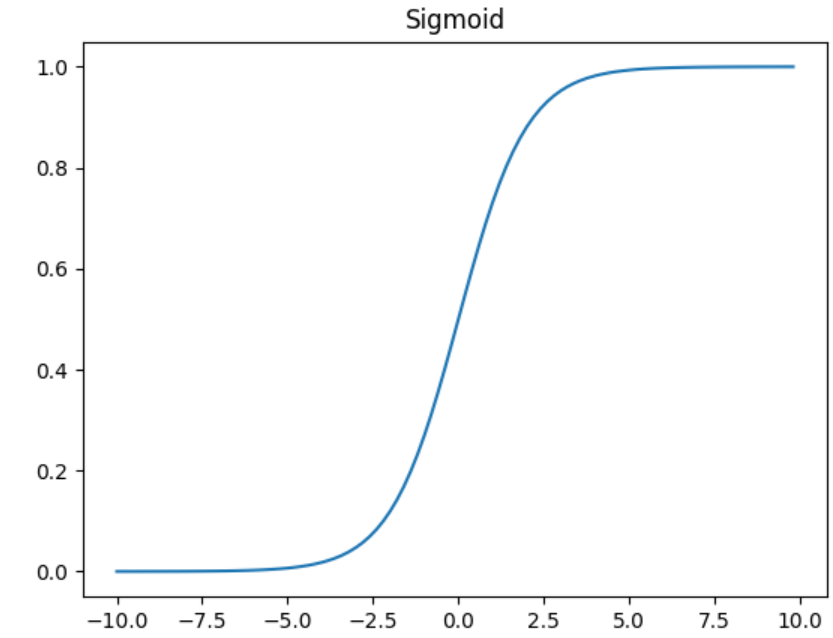
Activation functions: Sigmoid, Tanh, ReLU
Data: z in [-5, 5], manual_seed=2

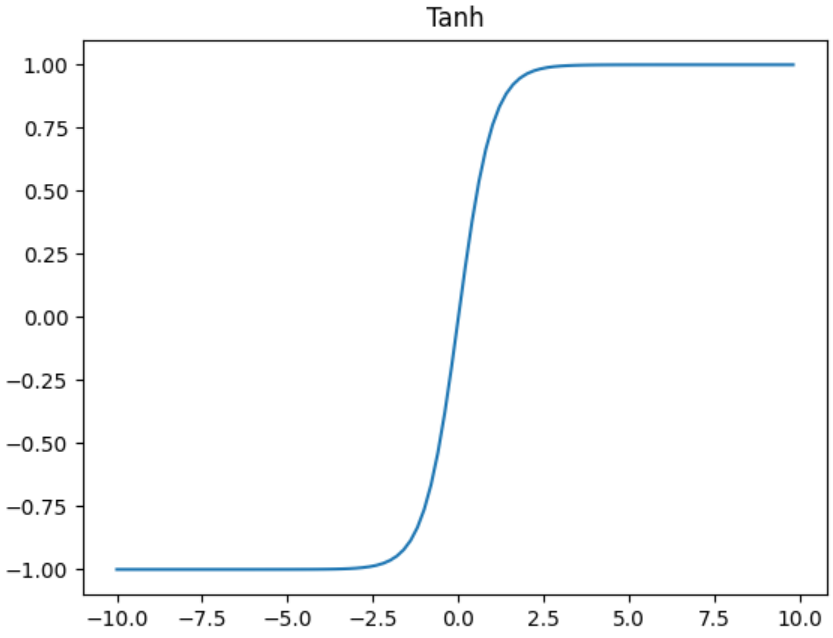
Script Output

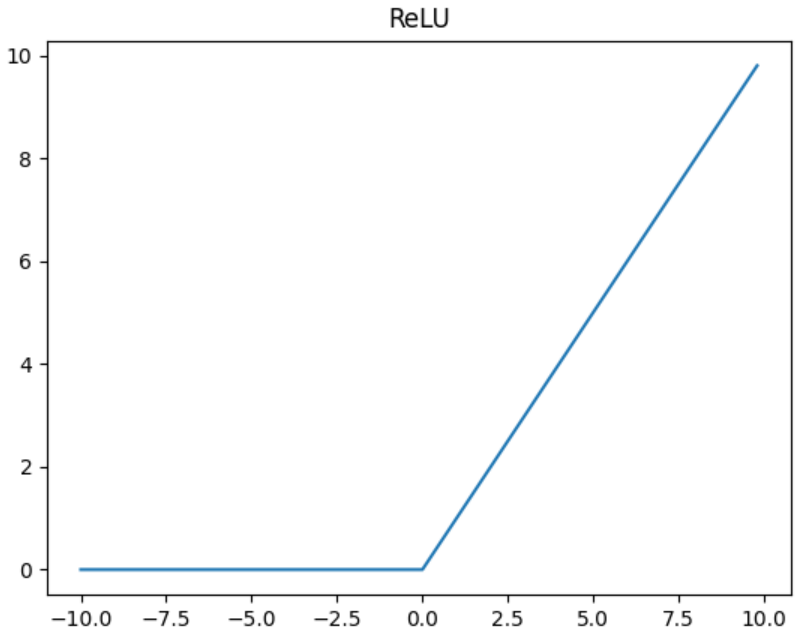
03e_activationfuction_v2_rev02.py (rev.02)

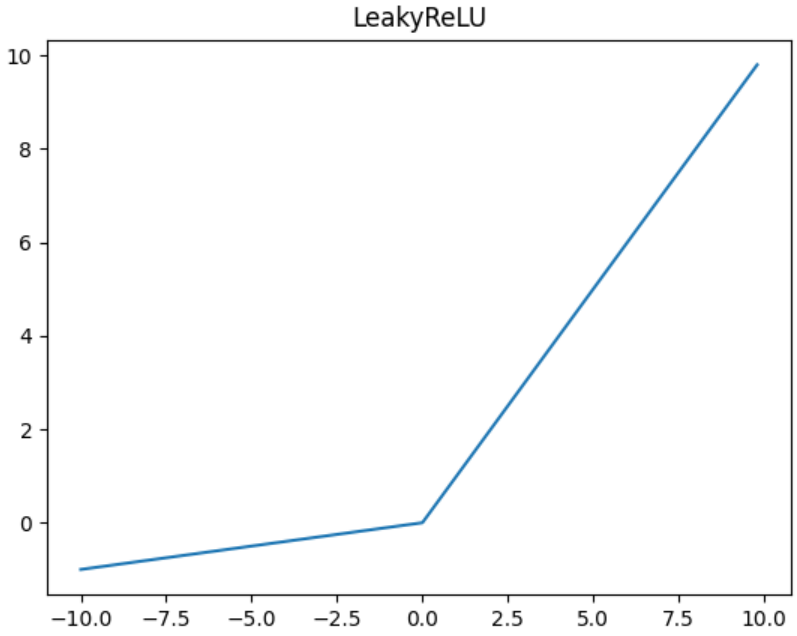
Activation functions: Sigmoid, Tanh, ReLU, LeakyReLU
Data: z in [-10, 10], manual_seed=2025

Script Output





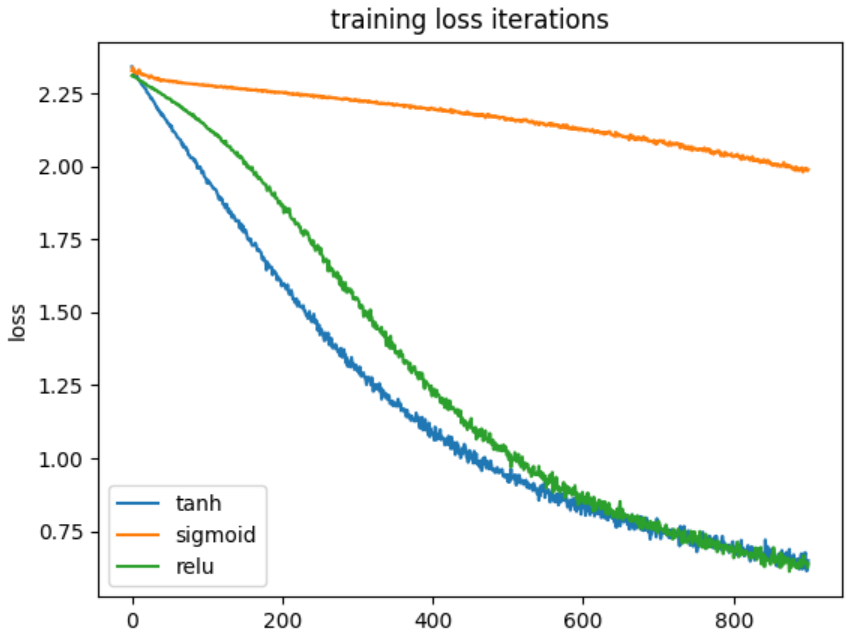

ReLU
10
8
6
4
2
0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

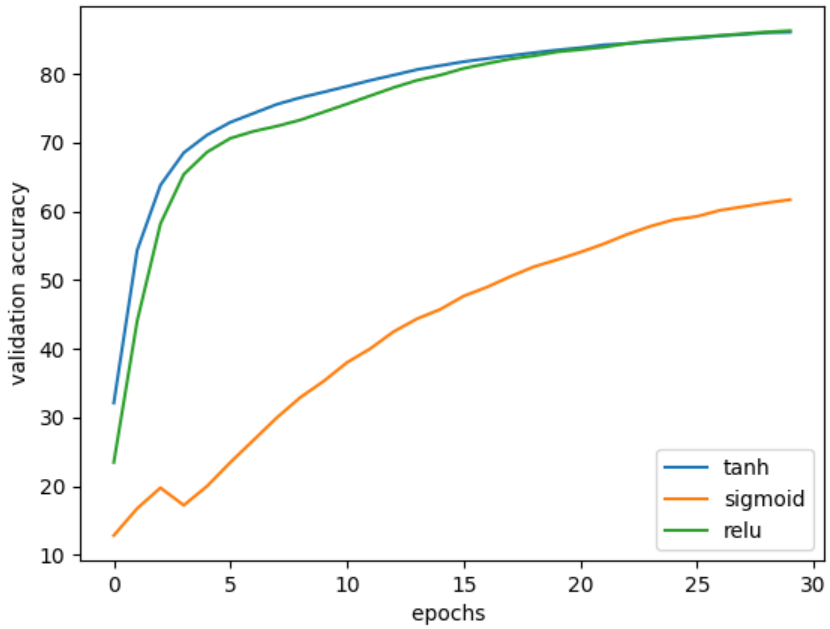

LeakyReLU
10
8
6
4
2
0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

03f_mist1layer_v2.py (Original)

MNIST, test activations
H=50, SGD, 2 epochs, CrossEntropyLoss
Activations: Sigmoid, Tanh, ReLU
PyTorch, manual_seed=0

Script Output





03f_mist1layer_v2_rev02.py (rev.02)

MNIST, test activations
H=64, Adam, 2 epochs, CrossEntropyLoss
Activations: ReLU, Sigmoid, Tanh
PyTorch, manual_seed=2026

Script Output

Activation: relu, Epoch [1/2], Loss: 172.5290
Activation: relu, Epoch [2/2], Loss: 83.0604
Activation: relu, Accuracy on 10000 test images: 95.70%
Activation: sigmoid, Epoch [1/2], Loss: 272.2486
Activation: sigmoid, Epoch [2/2], Loss: 112.4545
Activation: sigmoid, Accuracy on 10000 test images: 93.95%
Activation: tanh, Epoch [1/2], Loss: 170.0293
Activation: tanh, Epoch [2/2], Loss: 80.9195
Activation: tanh, Accuracy on 10000 test images: 95.62%