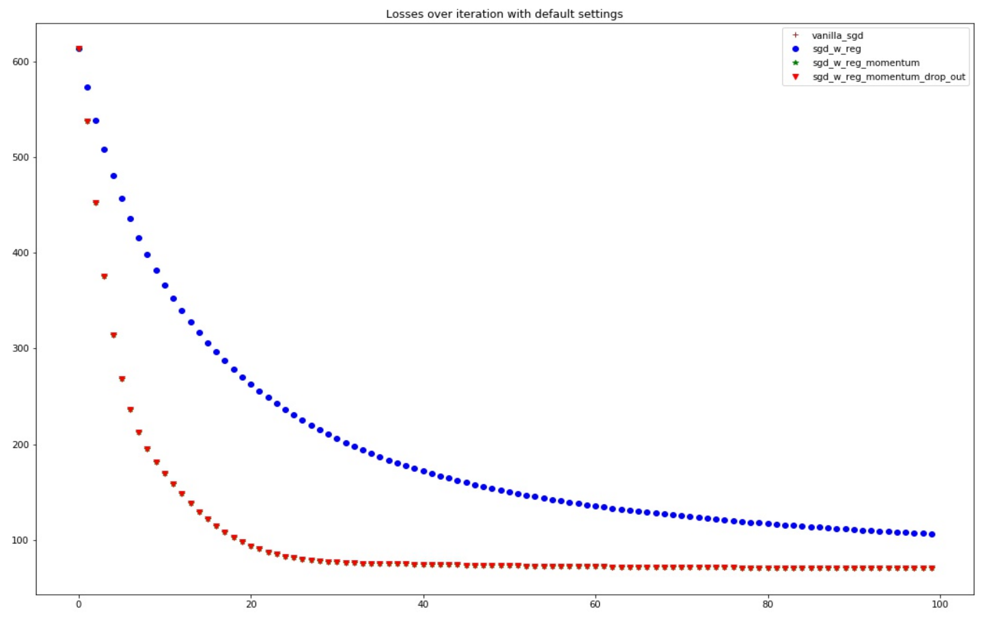
Question 5: Building a Neural Networks

Part 1: For a ReLu network:

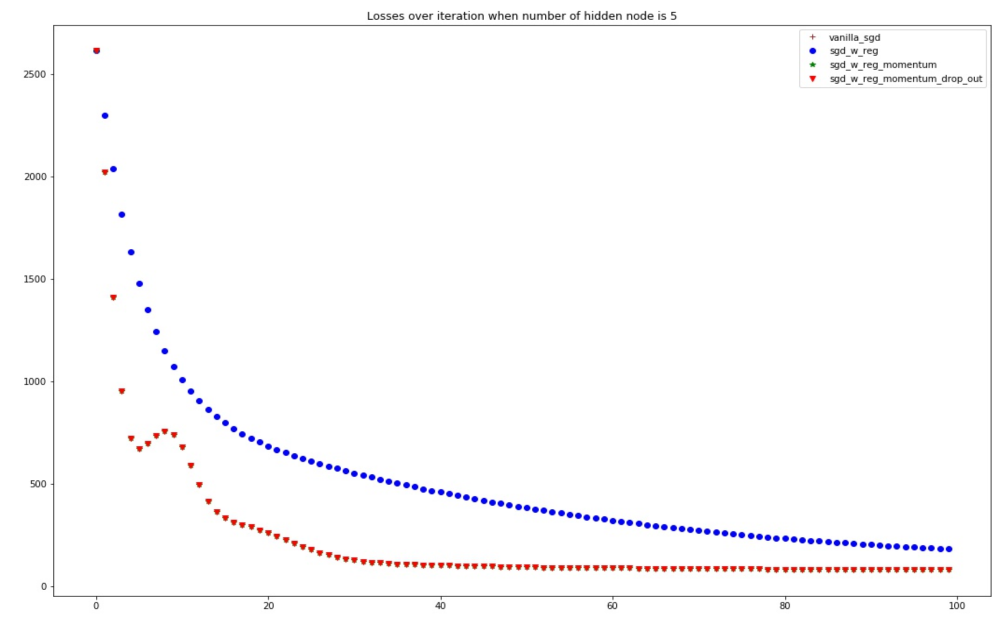
A

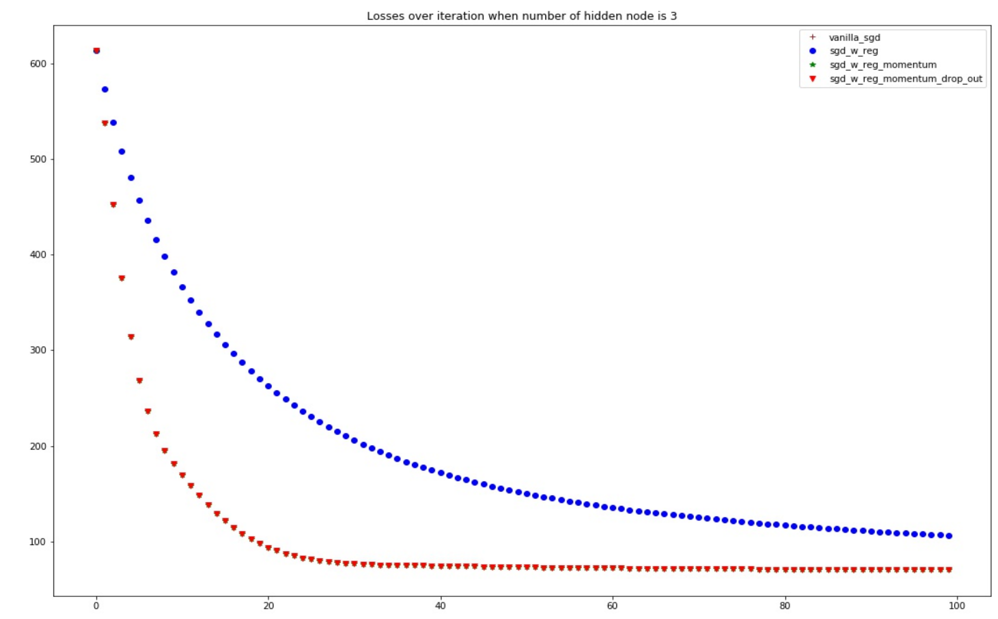


(a) From the figure above, we can see vanilla-sgd and sgd-w-reg have relative similar performance, while sgd-w-momentum and sgd-w-reg-momentum-drop-out have similar performance. The later two perform better than the previous two methods. We can also see that after 100 iterations, sgd-w-momentum and sgd-w-reg-momentum-drop-out achieved same loss.

(b) The momentum methods works obviously better since momentum helps accelerate gradients in the right direction during training process. Therefore, we can see loss drops faster in momentum methods with same number of iterations. However, both regularization and dropout does little effect on the performance, and which suggest that the model may be underfit under 100 iterations.

B


Losses over iteration when number of hidden node is 5
vanilla_sgd
sgd_w_reg
sgd_w_reg_momentum
sgd_w_reg_momentum_drop_out
0
500
1000
1500
2000
2500
0
20
40
60
80
100


Losses over iteration when number of hidden node is 3
vanilla_sgd
sgd_w_reg
sgd_w_reg_momentum
sgd_w_reg_momentum_drop_out
100
200
300
400
500
600
0
20
40
60
80
100

(a) Both sgd-w-momentum and sgd-w-reg-momentum-drop-out achieved same loss after 100 iterations. They perform better than the first two methods.

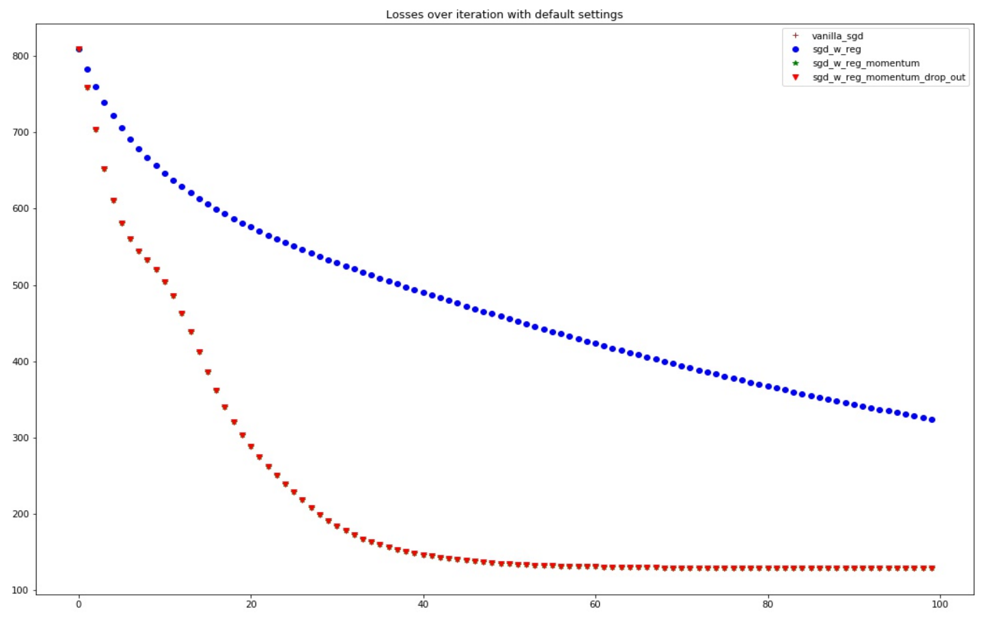
(b) The performance is not better than when we use 3 hidden nodes. The results suggest that the model starts to overfit when number of hidden node is 5.

C

Model	**Time**
vanilla-sgd	1.4158
sgd-w-reg	1.1950
sgd-w-momentum	1.0863
sgd-w-reg-momentum-drop-out	0.9850

Part 2: For a Sigmoid network

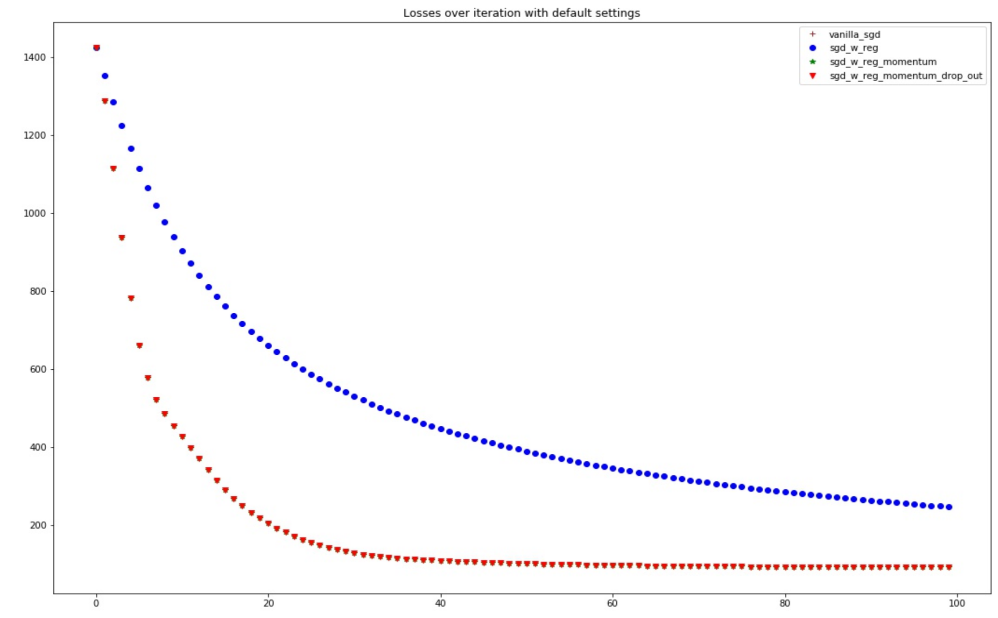
A

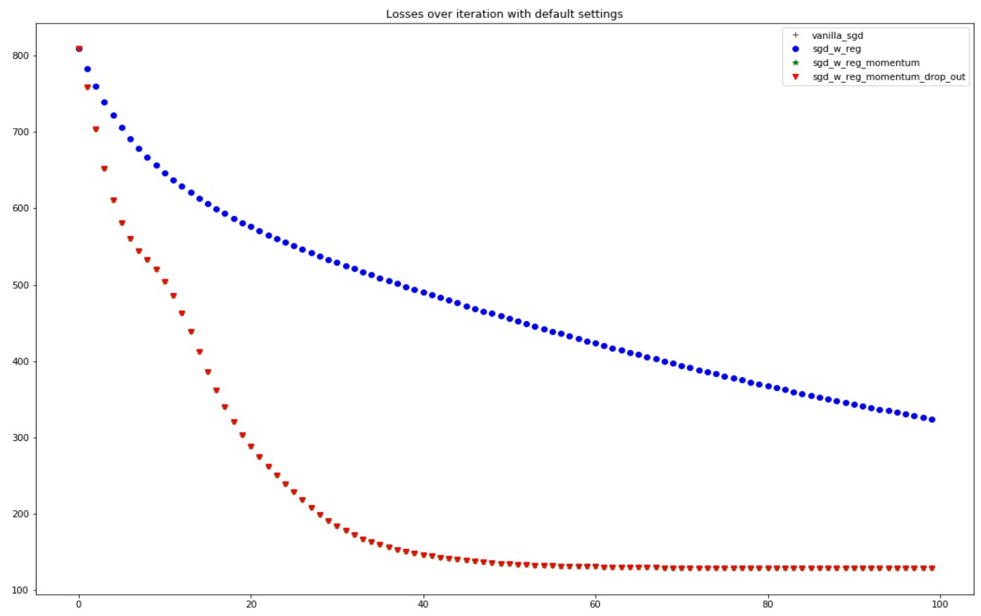


From the figure above, we can see vanilla-sgd and sgd-w-reg have relative similar performance, while sgd-w-momentum and sgd-w-reg-momentum-drop-out have similar performance. The later two perform better than the previous two methods. We can also see that after 100 iterations, sgd-w-momentum and sgd-w-reg-momentum-drop-out achieved same loss.

The momentum methods works obviously better since momentum helps accelerate gradients in the right direction during training process. Therefore, we can see loss drops faster in momentum methods with same number of iterations. However, both regularization and dropout does little effect on the performance, and which suggest that the model may be underfit under 100 iterations.

B





Both sgd-w-momentum and sgd-w-reg-momentum-drop-out achieved same loss after 100 iterations. They perform better than the first two methods. This is better than when number of hidden nodes is 3, which means that we need more complex models to learn better from the data.

C

Model	**Time**
vanilla-sgd	1.1793
sgd-w-reg	1.1569
sgd-w-momentum	1.0662
sgd-w-reg-momentum-drop-out	1.2817

D

(1) Relu is better.

(2) RuLu is better.

Model	**3-node-Relu Loss**	**3-node-Sigmoid Loss**
vanilla-sgd	106.0580	324.2820
sgd-w-reg	106.0685	324.3093
sgd-w-momentum	69.9962	128.6403
sgd-w-reg-momentum-drop-out	69.9962	128.6404

Model	**5-node-Relu Loss**	**5-node-Sigmoid Loss**
vanilla-sgd	182.3663	247.1844
sgd-w-reg	182.3803	247.2096
sgd-w-momentum	79.4184	91.3771
sgd-w-reg-momentum-drop-out	79.4184	91.3771

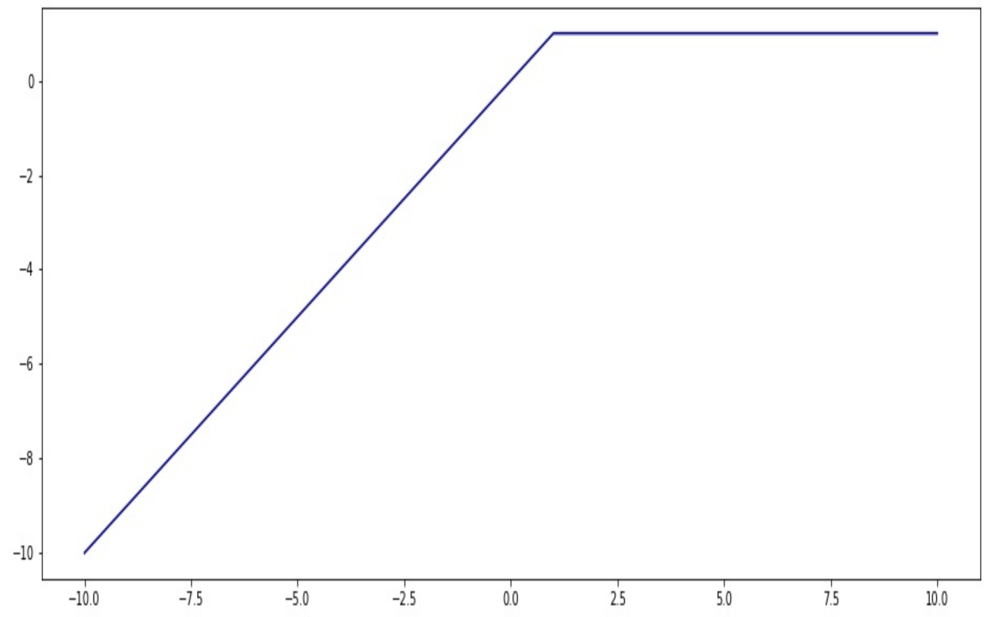
Question 6: Activation Functions

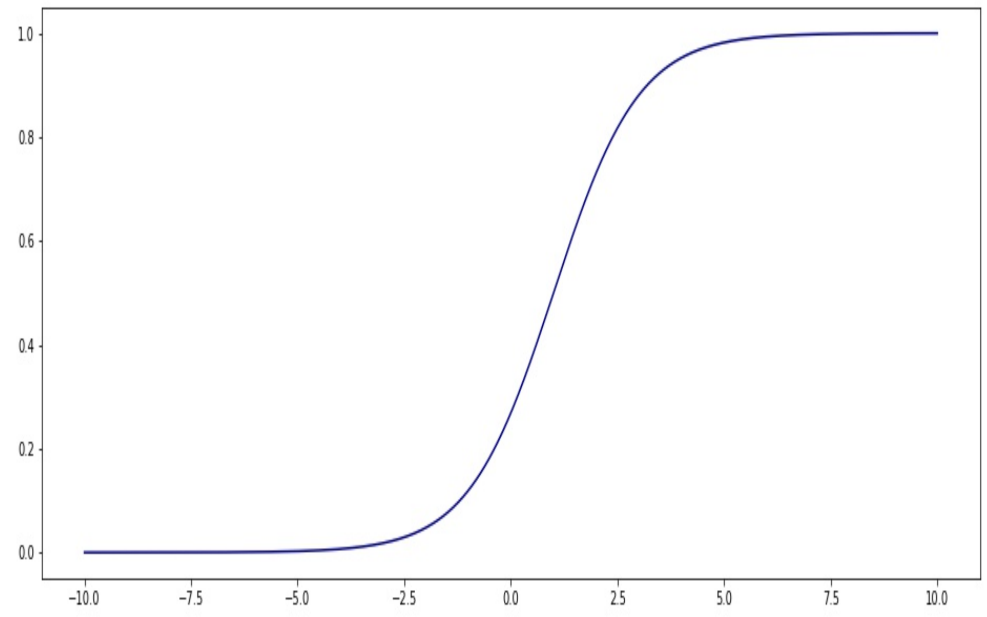
We create a python script to simulate different model structures and activation functions, with random data. Since there are numerous plots for each of structure and method (more than a thousand each), therefore I only plot the first 50 different shape and placed them in folder so that you can check. Here I only randomly choose some plots to illustrate.

Function:

$$y = V \cdot ReLu(wx + b) + v_b$$
$$y = V \cdot Sigmoid(wx + b) + v_b$$

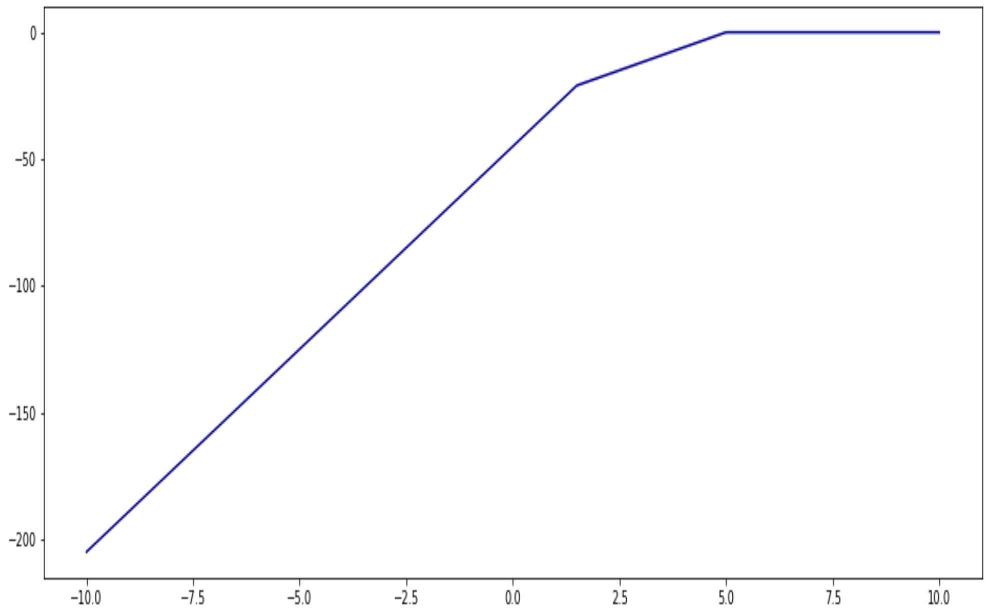
One Layer One Node

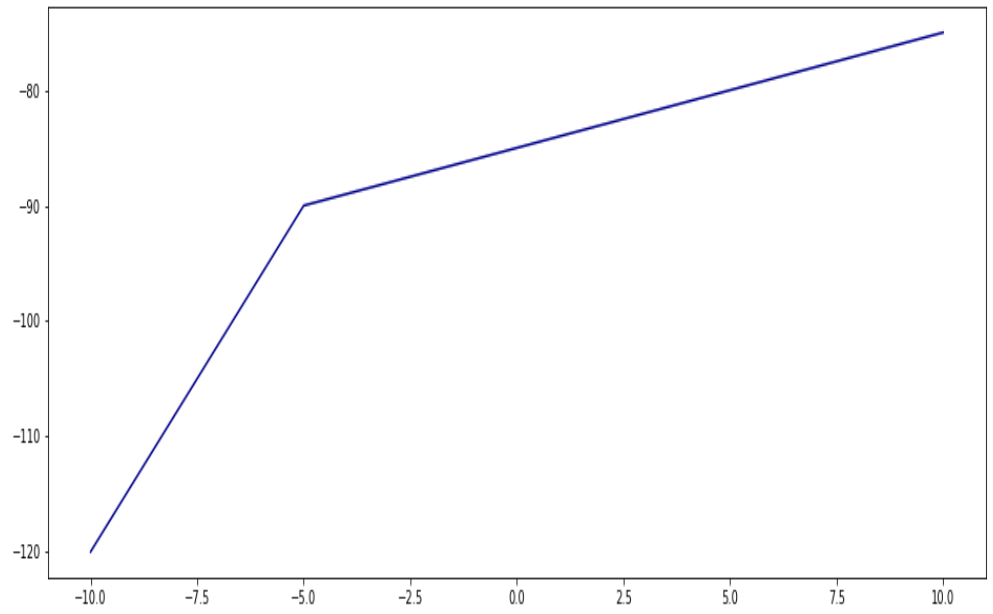


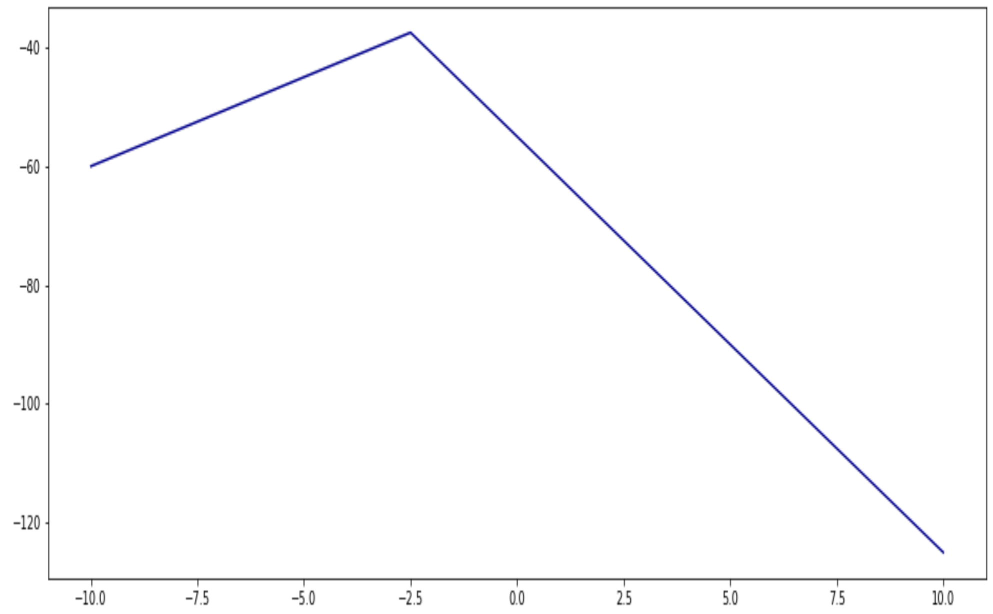


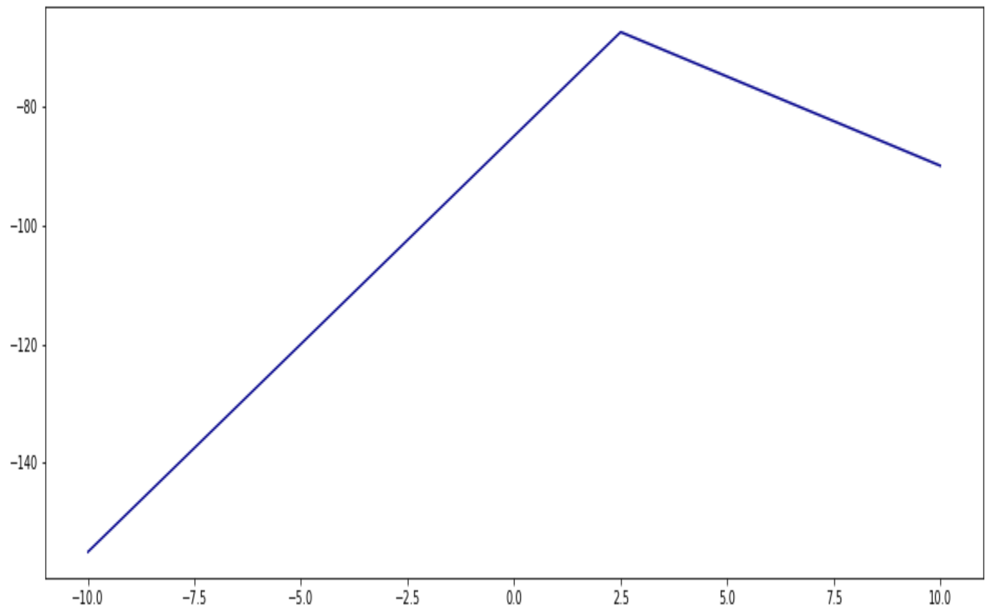
One Layer Two Nodes

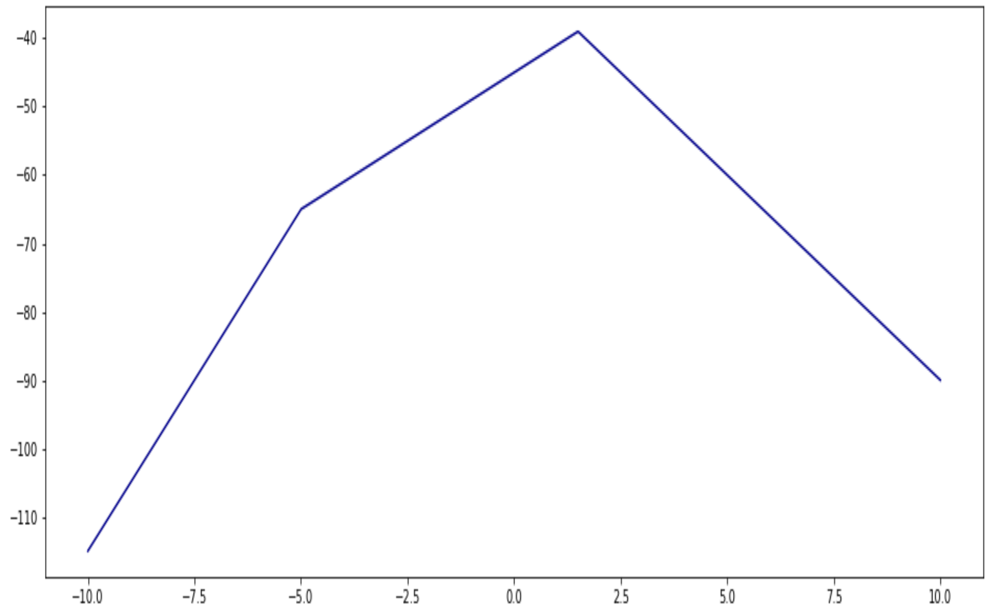
ReLu



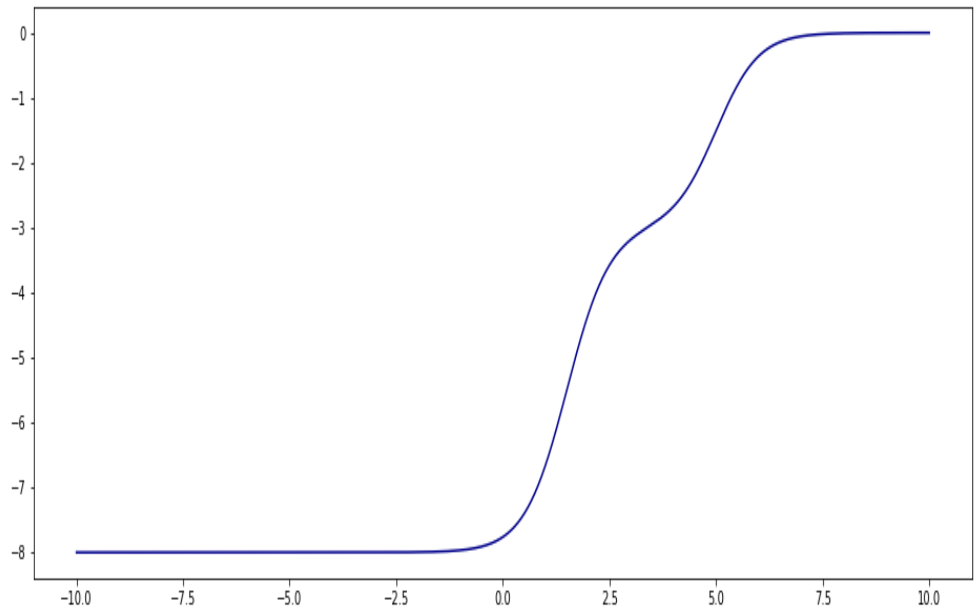

−80
−90
−100
−110
−120
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

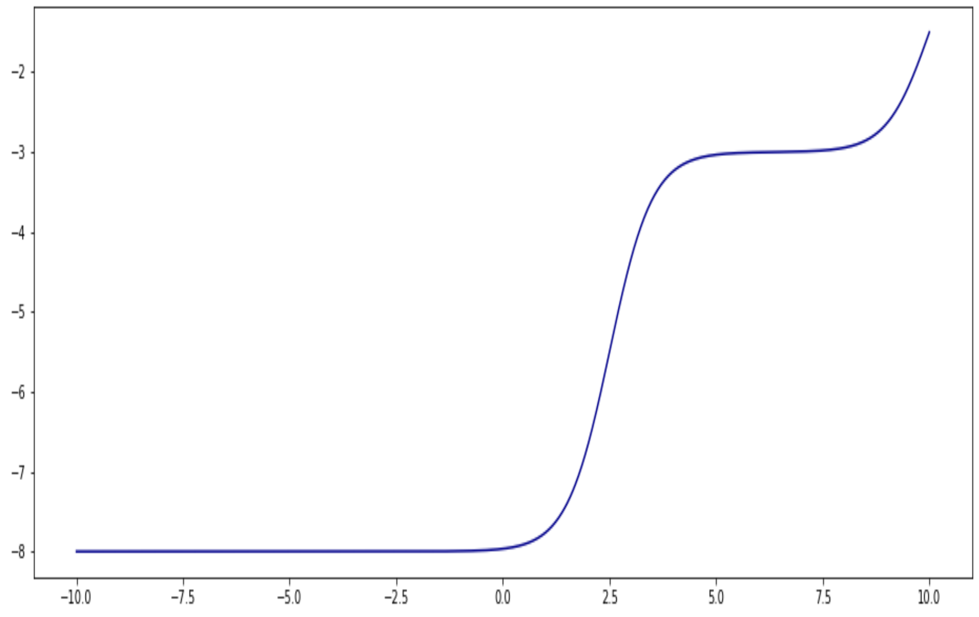

−40
−60
−80
−100
−120
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

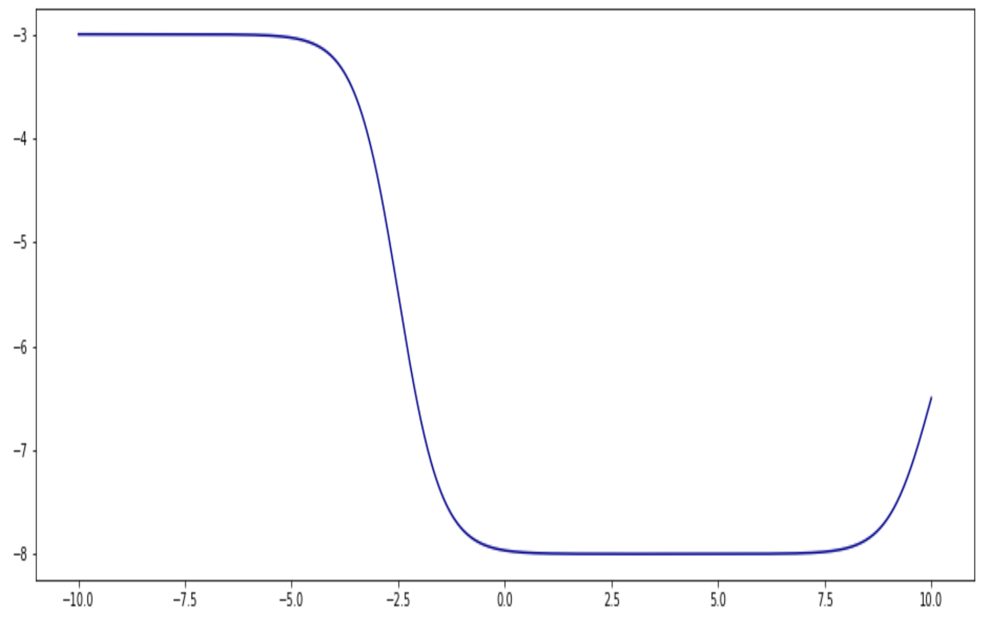

−80
−100
−120
−140
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

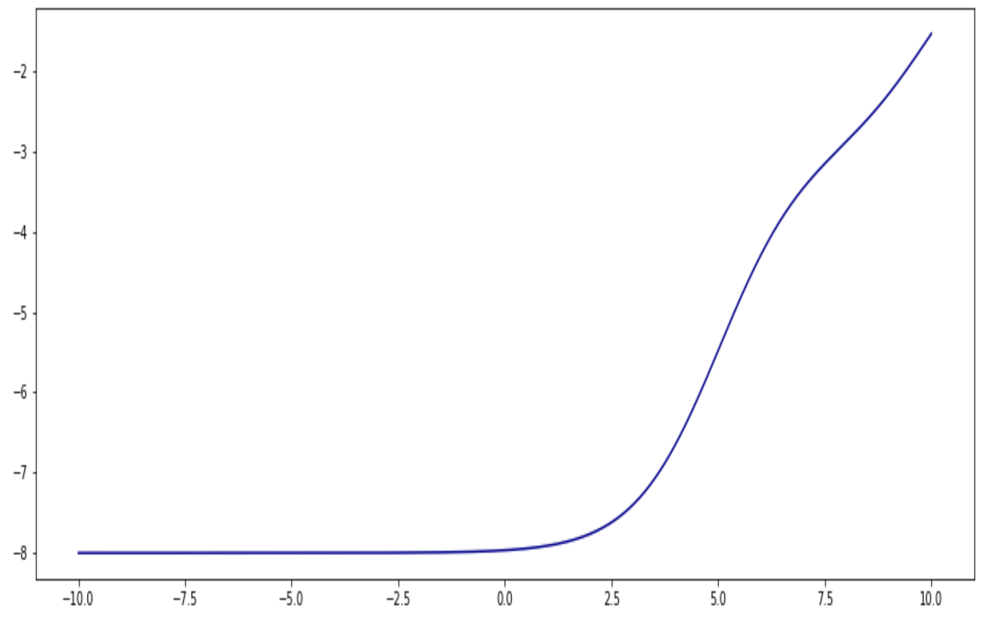

−40
−50
−60
−70
−80
−90
−100
−110
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

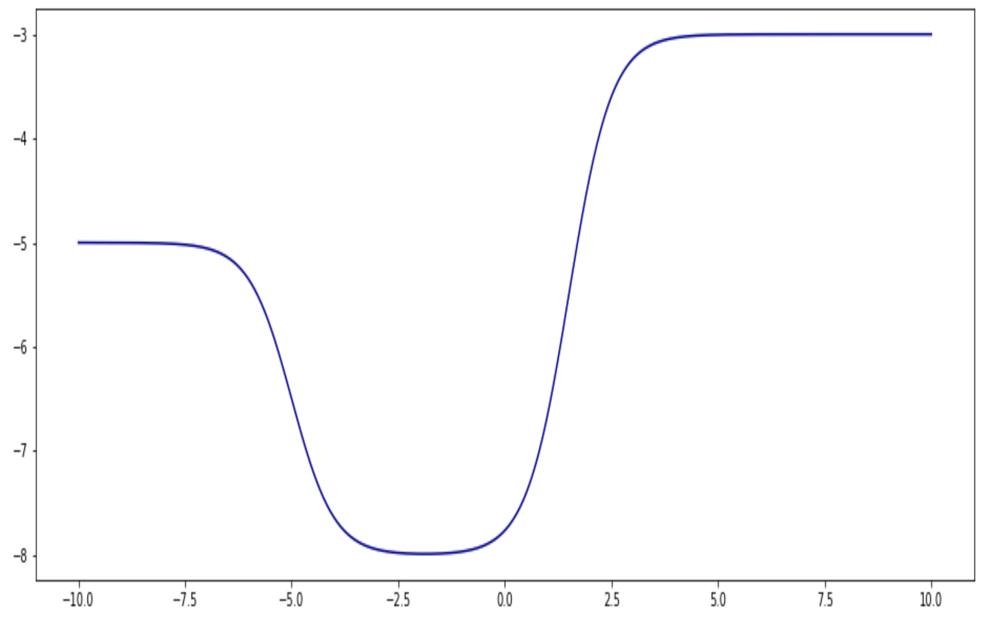
Sigmoid


0
−1
−2
−3
−4
−5
−6
−7
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


−2
−3
−4
−5
−6
−7
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

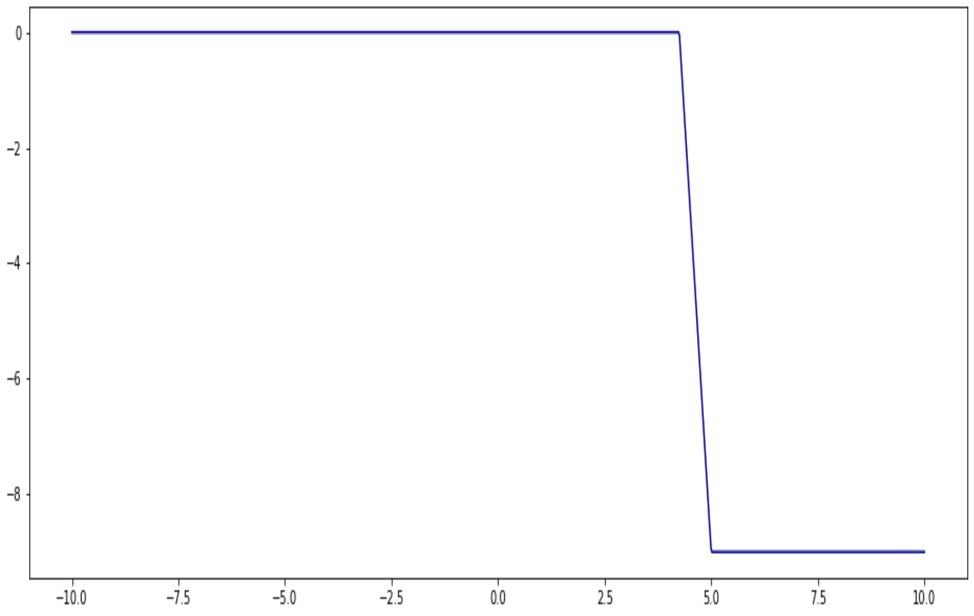

−3
−4
−5
−6
−7
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

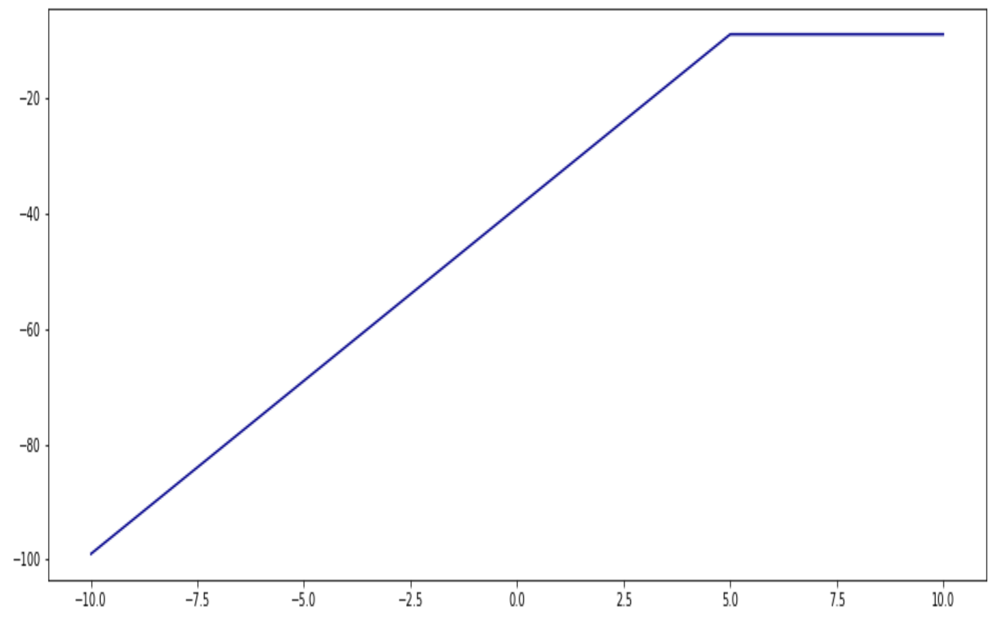

−2
−3
−4
−5
−6
−7
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

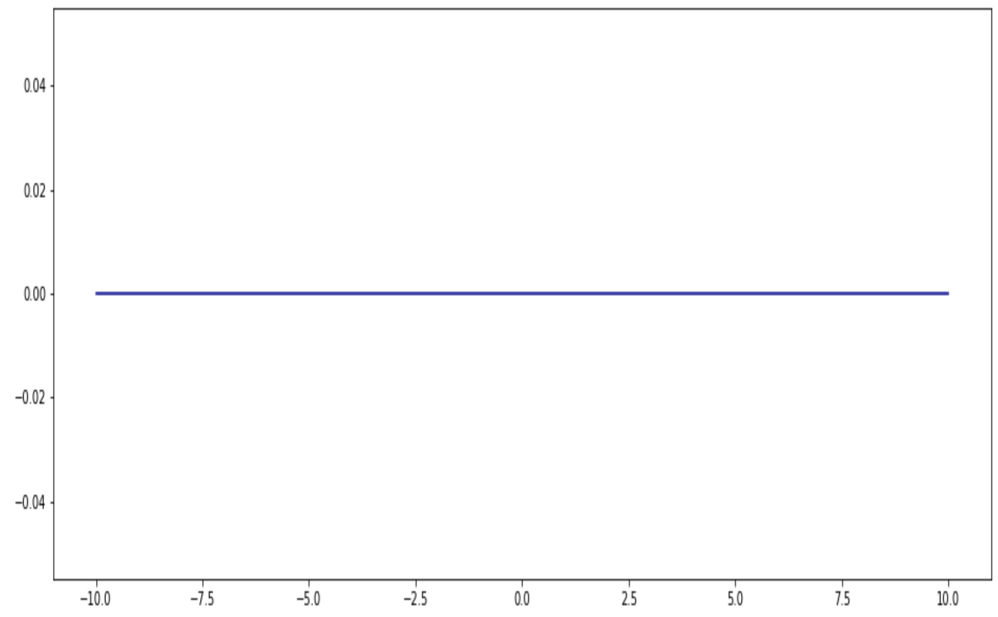

−3
−4
−5
−6
−7
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

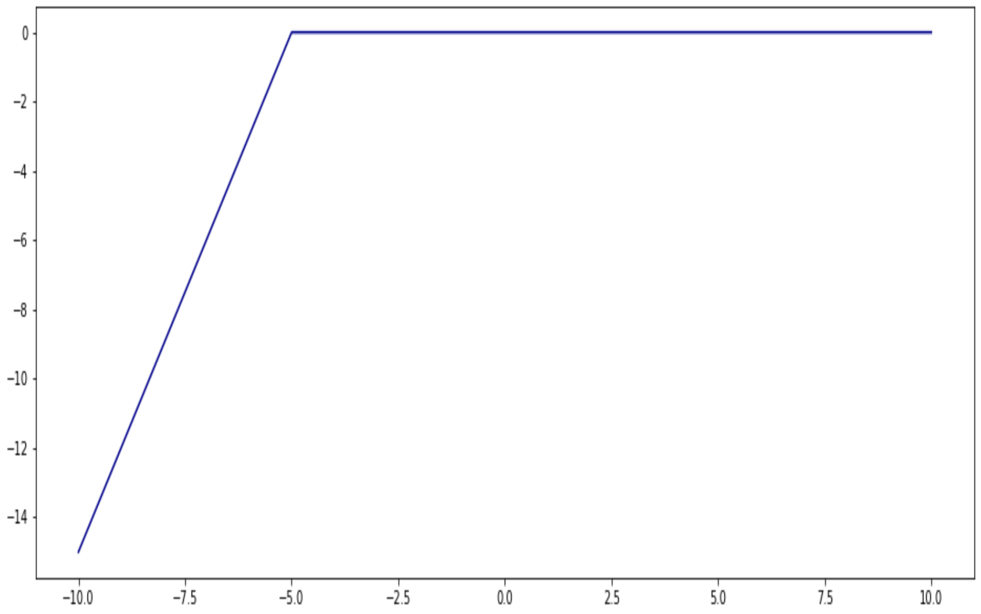
Two Layers One Node

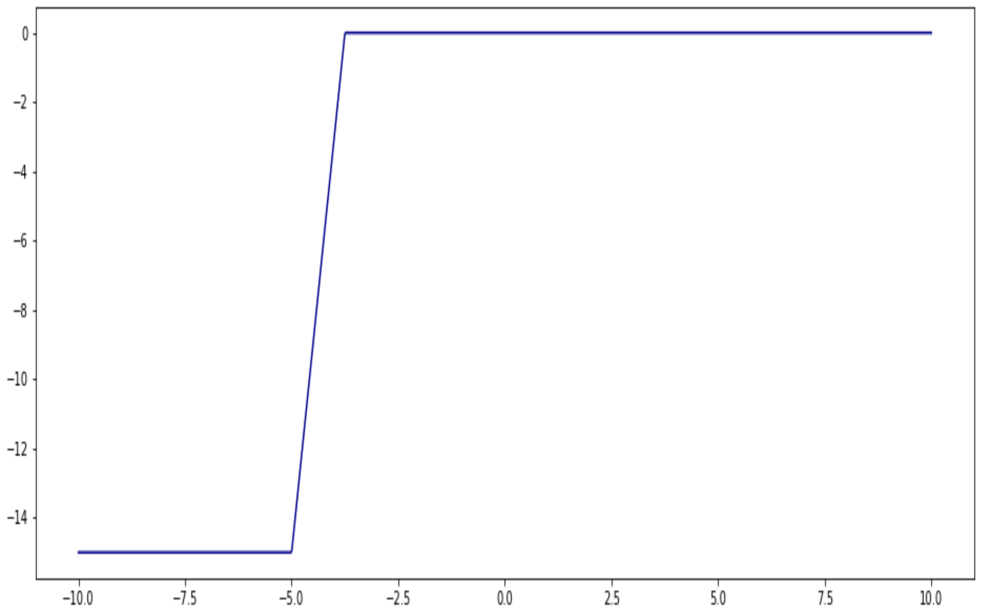
ReLu



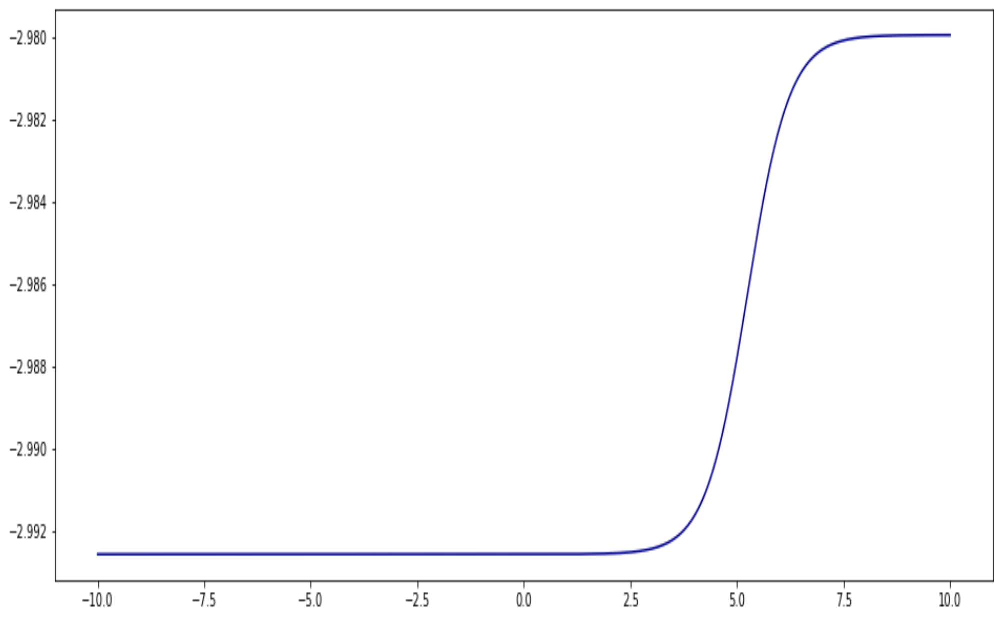

−20
−40
−60
−80
−100
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

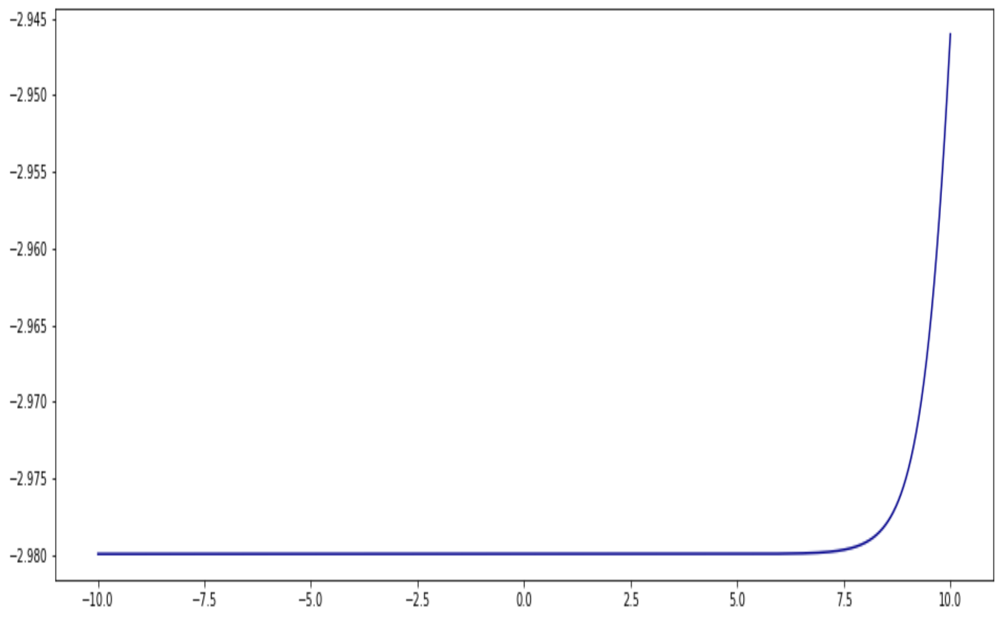

0.04
0.02
0.00
−0.02
−0.04
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

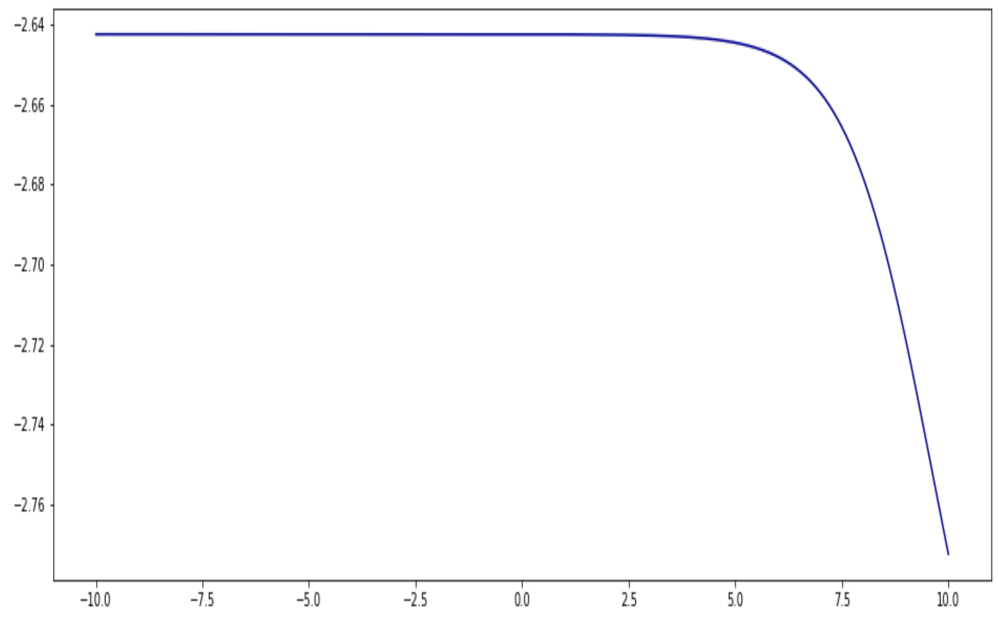

0
−2
−4
−6
−8
−10
−12
−14
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

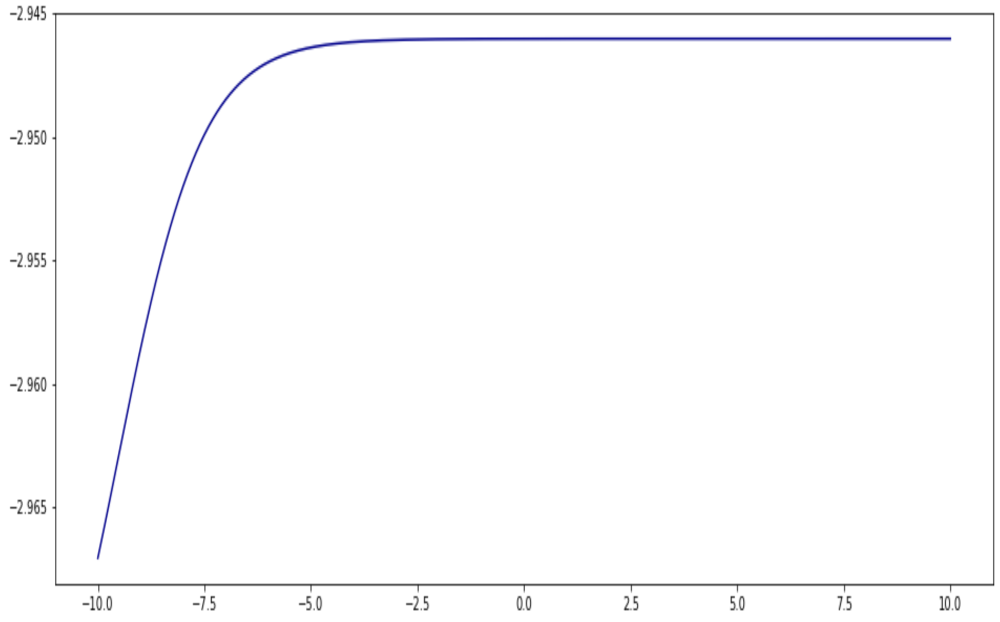

0
−2
−4
−6
−8
−10
−12
−14
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

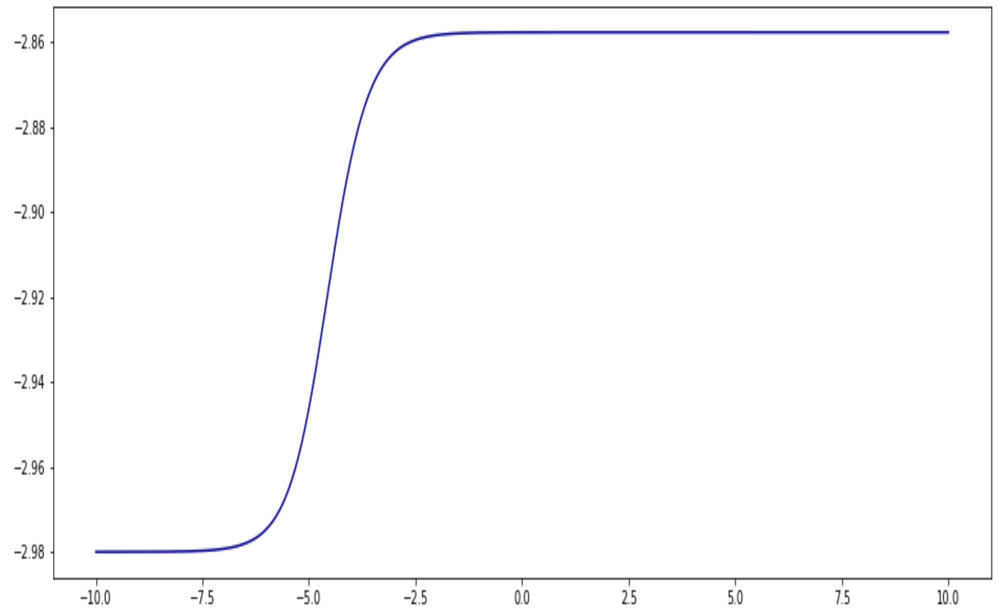
Sigmoid


−2.980
−2.982
−2.984
−2.986
−2.988
−2.990
−2.992
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


−2.945
−2.950
−2.955
−2.960
−2.965
−2.970
−2.975
−2.980
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

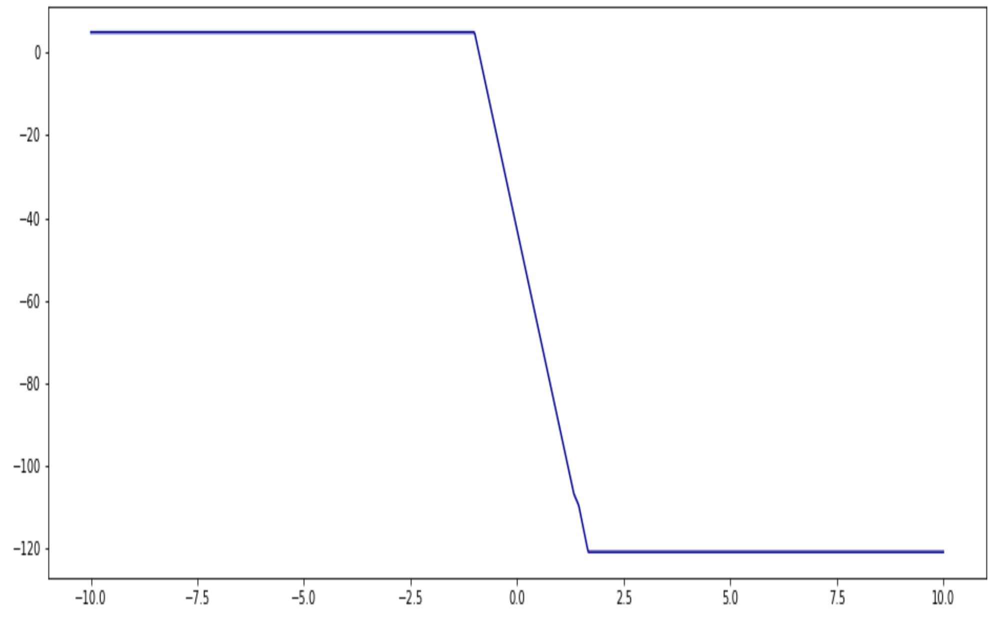

−2.64
−2.66
−2.68
−2.70
−2.72
−2.74
−2.76
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

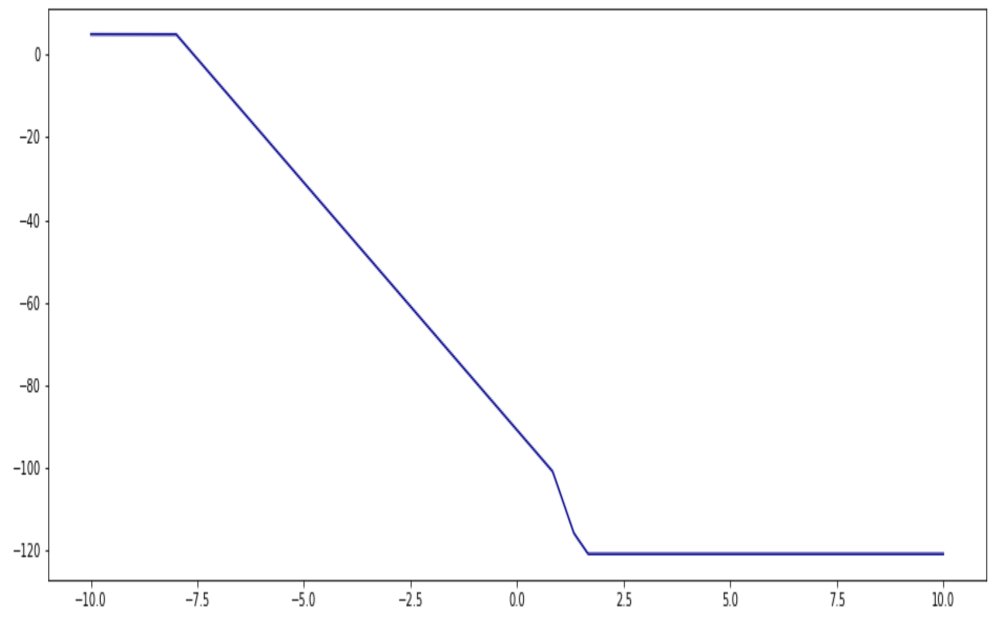

−2.945
−2.950
−2.955
−2.960
−2.965
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

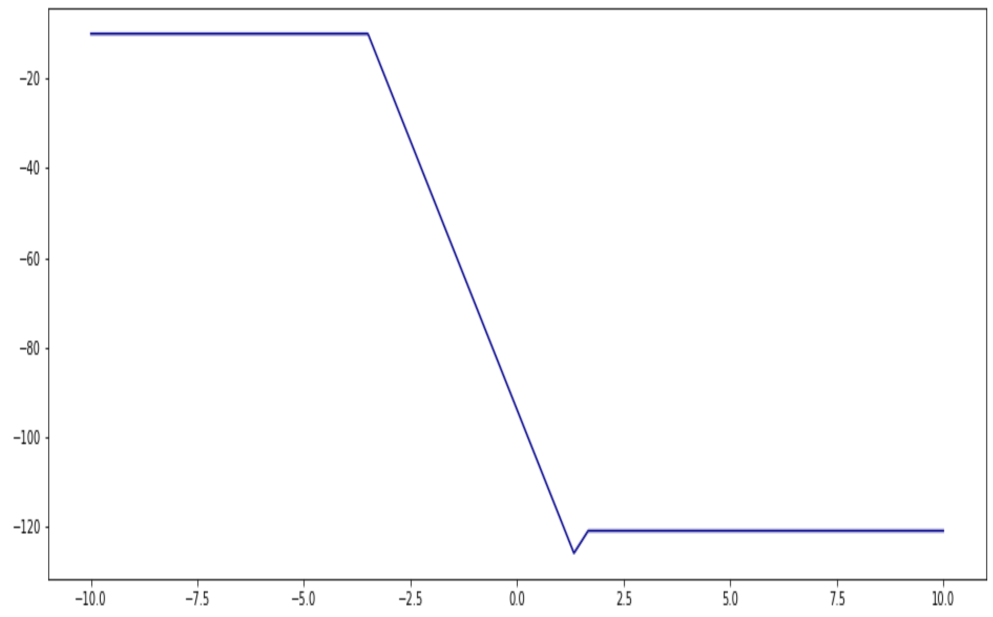

−2.86
−2.88
−2.90
−2.92
−2.94
−2.96
−2.98
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

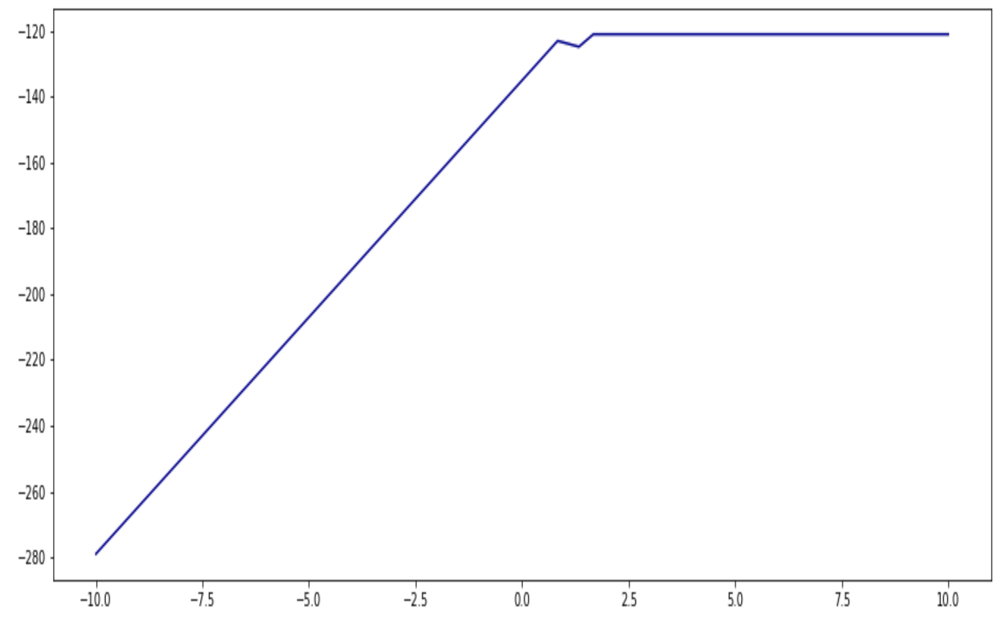
Two Layers Two Nodes

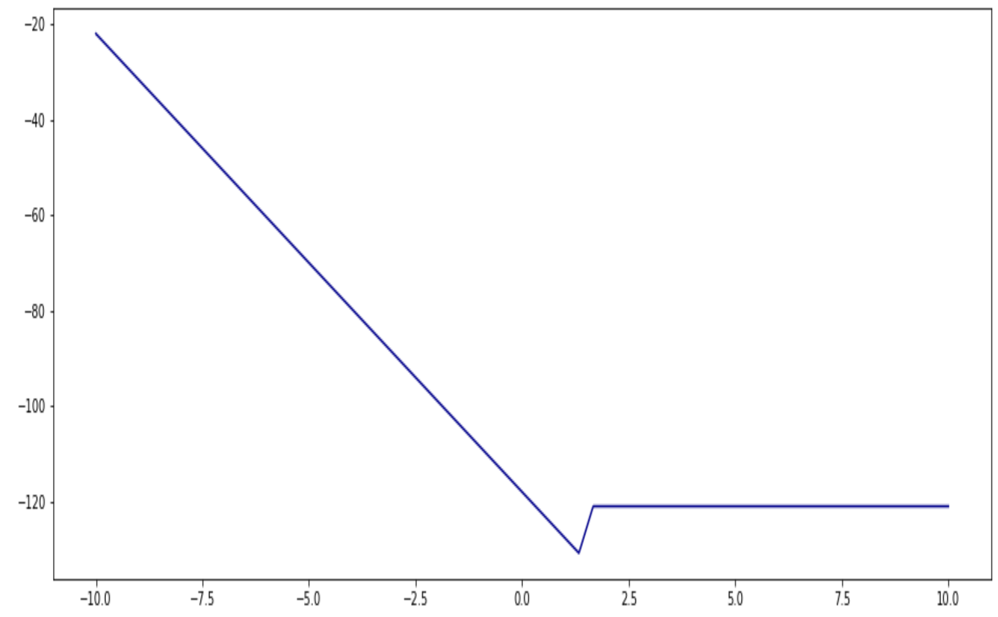
ReLu



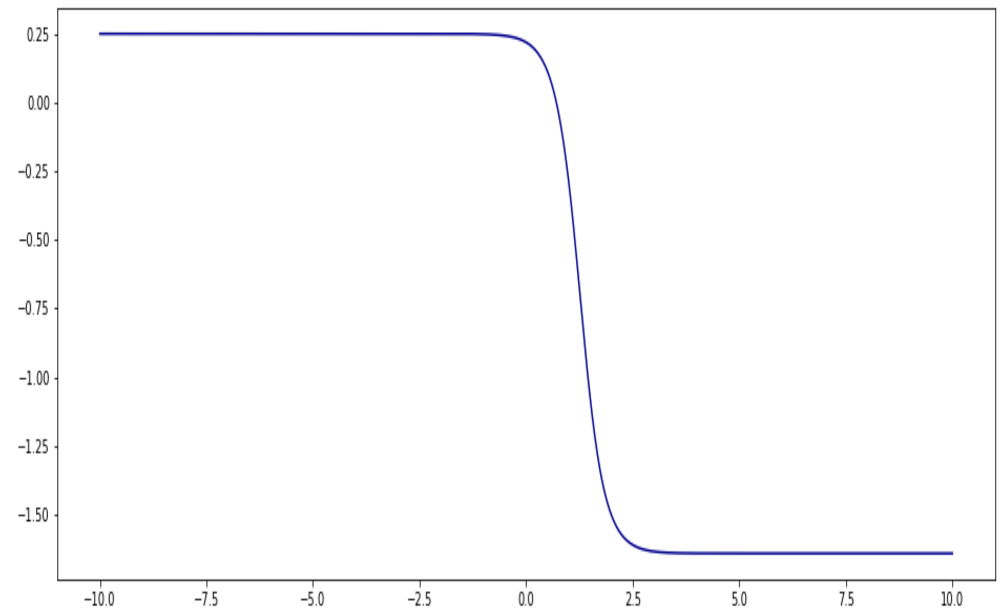

0
−20
−40
−60
−80
−100
−120
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

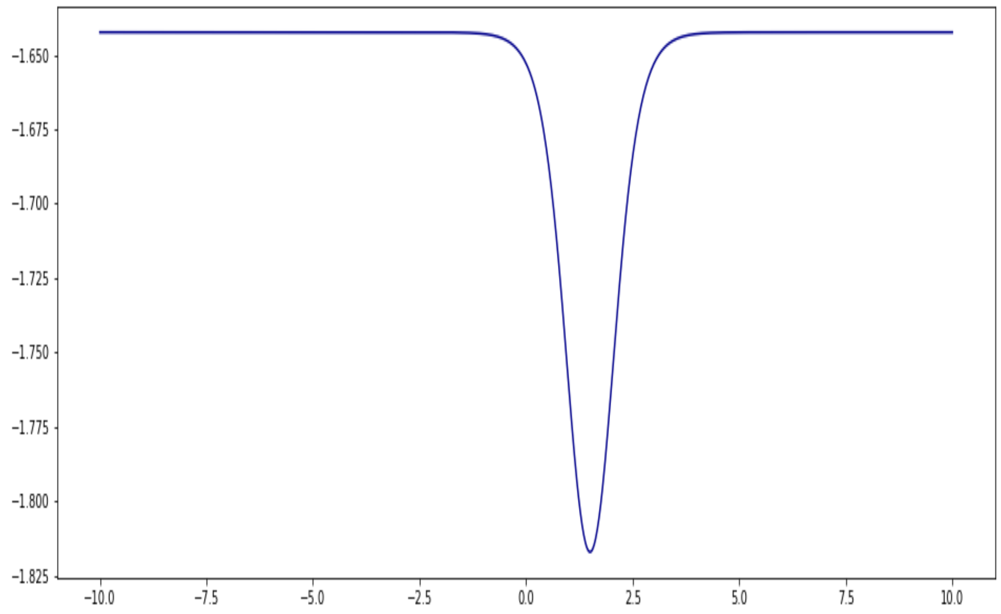

−20
−40
−60
−80
−100
−120
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

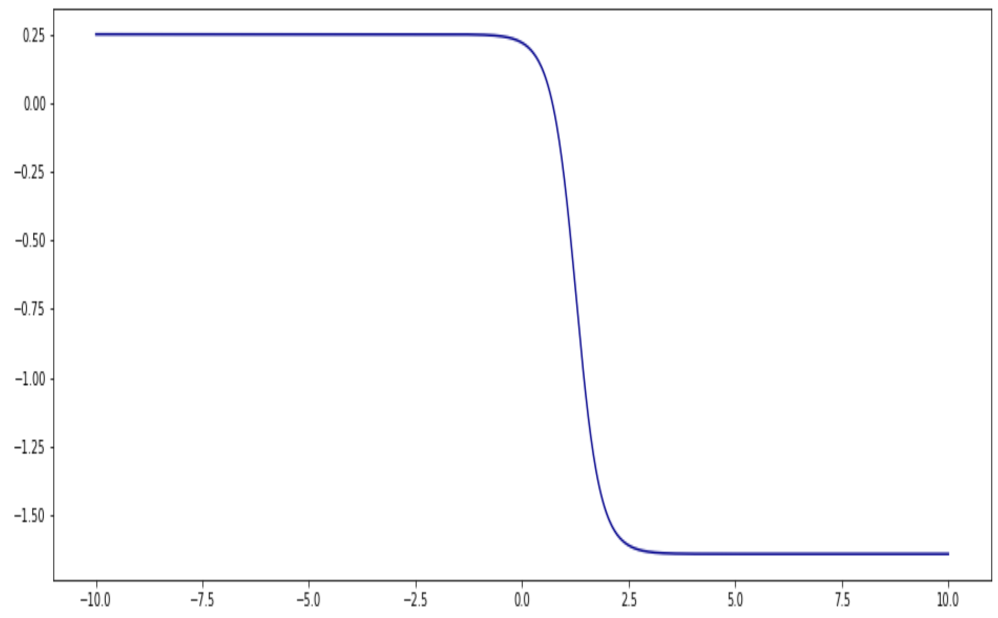

−120
−140
−160
−180
−200
−220
−240
−260
−280
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

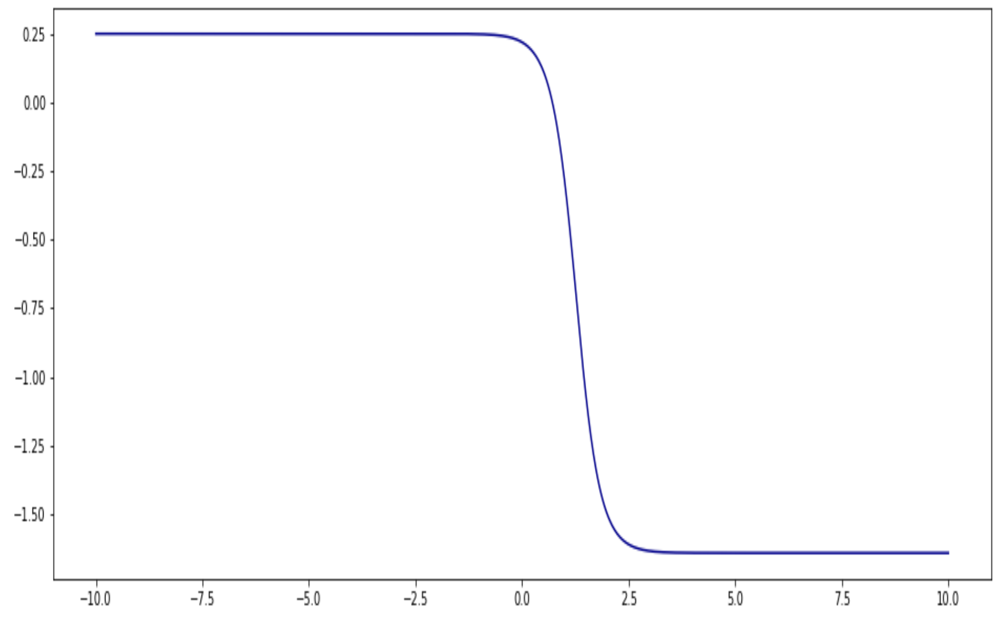

−20
−40
−60
−80
−100
−120
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

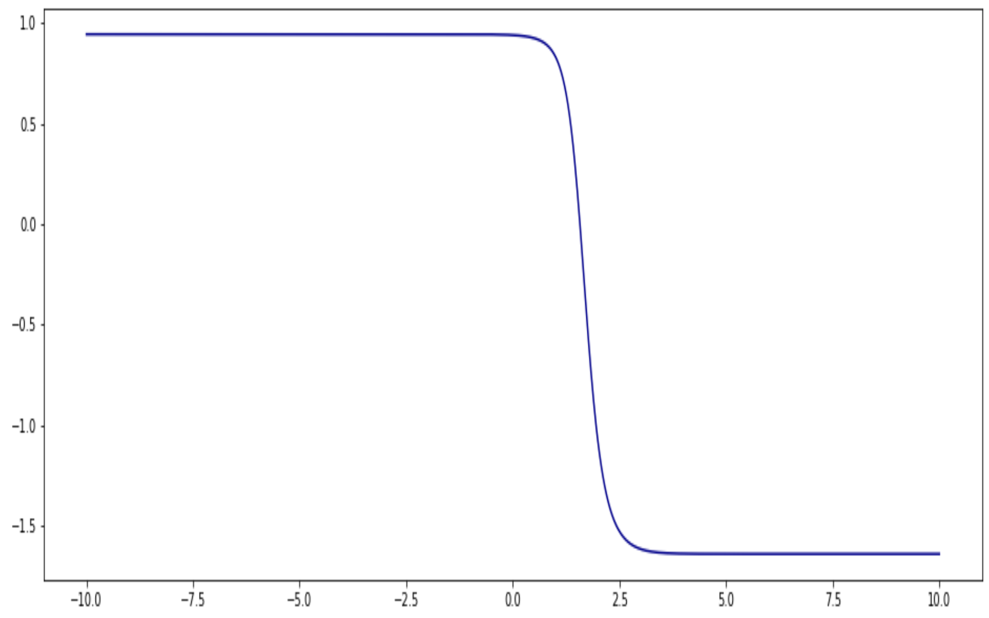
Sigmoid


0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


−1.650
−1.675
−1.700
−1.725
−1.750
−1.775
−1.800
−1.825
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


1.0
0.5
0.0
−0.5
−1.0
−1.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0