ML-4

siddharth bhagvagar

October 2024

1 Task-1

the output of pendigits dataset, with 2 layers, 10 training rounds is

```
classification accuracy=0.8073
```

the output of pendigits dataset, with 4 layers, 40 units per hidden layer, 20 training rounds, sigmoid activation for the hidden layers is

```
classification accuracy=0.8814
```

2 Task-1b

The hyperparameter for pendigits dataset is 4 layers, 20 units per layer and 80 epochs which gives the accuracy of
1. 92.85
2. 92.77
3. 94.23
4. 93.68
5. 94.48
6. 93.48
7. 94.00
8. 92.20
9. 94.37
10. 94.14

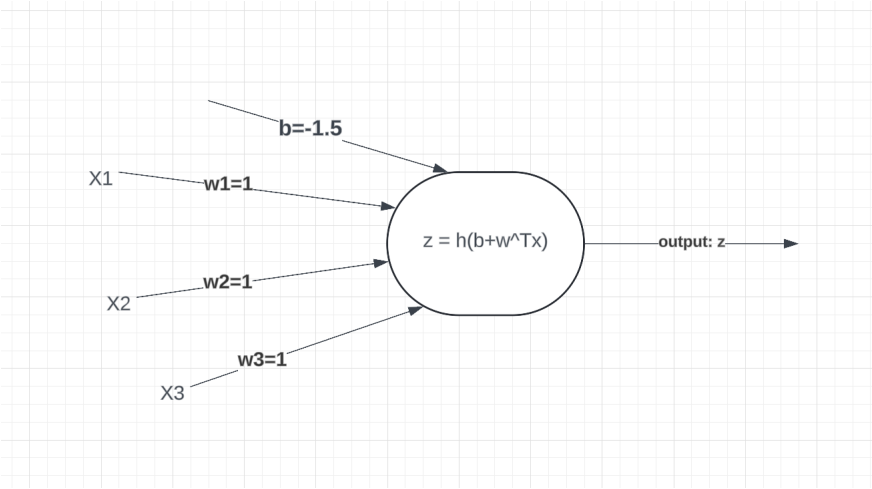
when the network is run 10 times which gives the best accuracy of approx 94%

3 Task-1c

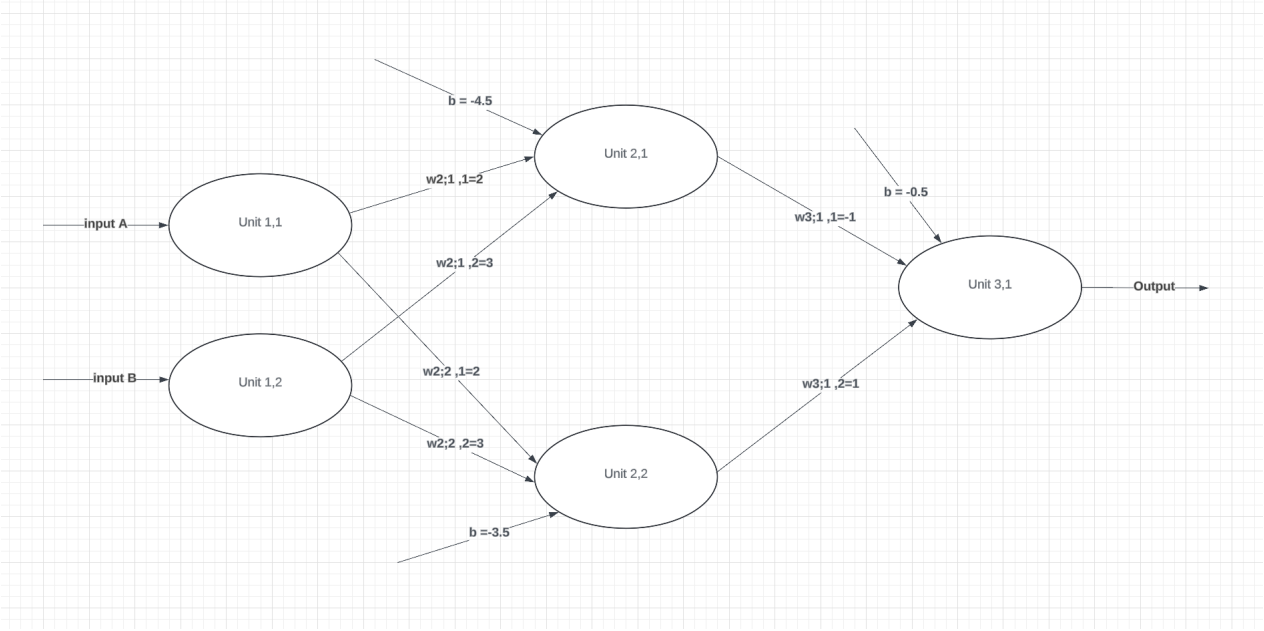
I changed the activation function of tanh in the hidden layers and the output layer as a sigmoid. which got me an accuracy of approx 97% with the hyperparameter (4 layers,128 units per layer, 40 epochs) with pendigits dataset.
nn_keras("../uci_data", "pendigits_string", 4, 100, 30)

4 Task-2

the diagram is



5 Task-3



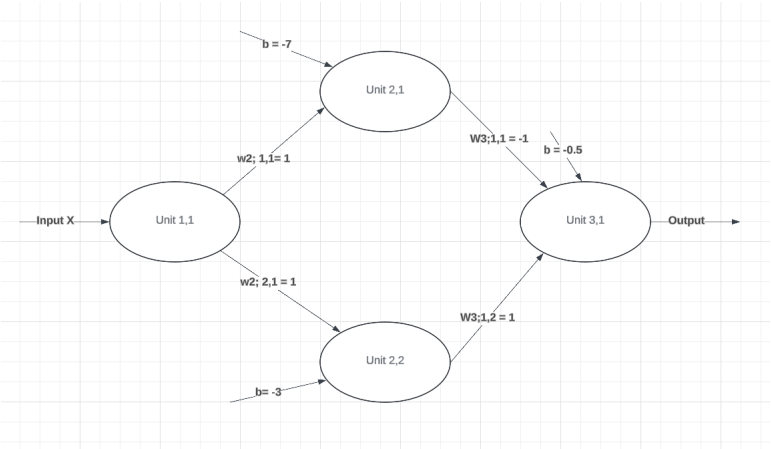
if 2A + 3B = 4,
Unit 2,1 = h(2A + 3B - 4.5) = h(4-4.5) = h(-0.5) = 0
Unit 2,2 = h(2A + 3B - 3.5) = h(4-3.5) = h(0.5) = 1

Unit 3,1 = h(1(1)-1(0)-0.5) = h(1-0.5) = h(0.5) = 1

if 2A + 3B = 5,
Unit 2,1 = h(2A + 3B - 4.5) = h(5-4.5) = h(0.5) = 1
Unit 2,2 = h(2A + 3B - 3.5) = h(5-3.5) = h(1.5) = 1
Unit 3,1 = h(1(1)-1(1)-0.5) = h(0-0.5) = h(-0.5) = 0

if 2A + 3B = 3,
Unit 2,1 = h(2A + 3B - 4.5) = h(3-4.5) = h(-1.5) = 0
Unit 2,2 = h(2A + 3B - 3.5) = h(3-3.5) = h(-0.5) = 0
Unit 3,1 = h(1(0)-1(0)-0.5) = h(0-0.5) = h(-0.5) = 0

6 Task-4



if X < 3, i.e X = 2
Unit 2,1 = h(2(1) - 7) = h(2-7) = h(-4) = 0
Unit 2,2 = h(2(1) - 3) = h(2-3) = h(-1) = 0
Unit 3,1 = h(0(1)-0(-1)-0.5) = h(0-0.5) = h(-0.5) = 0

if X > 7, i.e X = 8
Unit 2,1 = h(8(1) - 7) = h(8-7) = h(1) = 1
Unit 2,2 = h(8(1) - 3) = h(8-3) = h(5) = 1
Unit 3,1 = h(1(1)-1(-1)-0.5) = h(1-1.5) = h(-0.5) = 0

if 7 > X > 3, i.e X = 5

Unit 2,1 = h(5(1) - 7) = h(5-7) = h(-2) = 0
Unit 2,2 = h(5(1) - 3) = h(5-3) = h(2) = 1
Unit 3,1 = h(1(1)-0(-1)-0.5) = h(0.5) = 1

7 Task-5

initializing all weights to zero would likely lower classification accuracy than random initialization. The network would struggle to break symmetry, potentially converge more slowly, and learn like it has only one feature. Random initialization helps avoid these issues by providing a diverse starting point for the learning process.