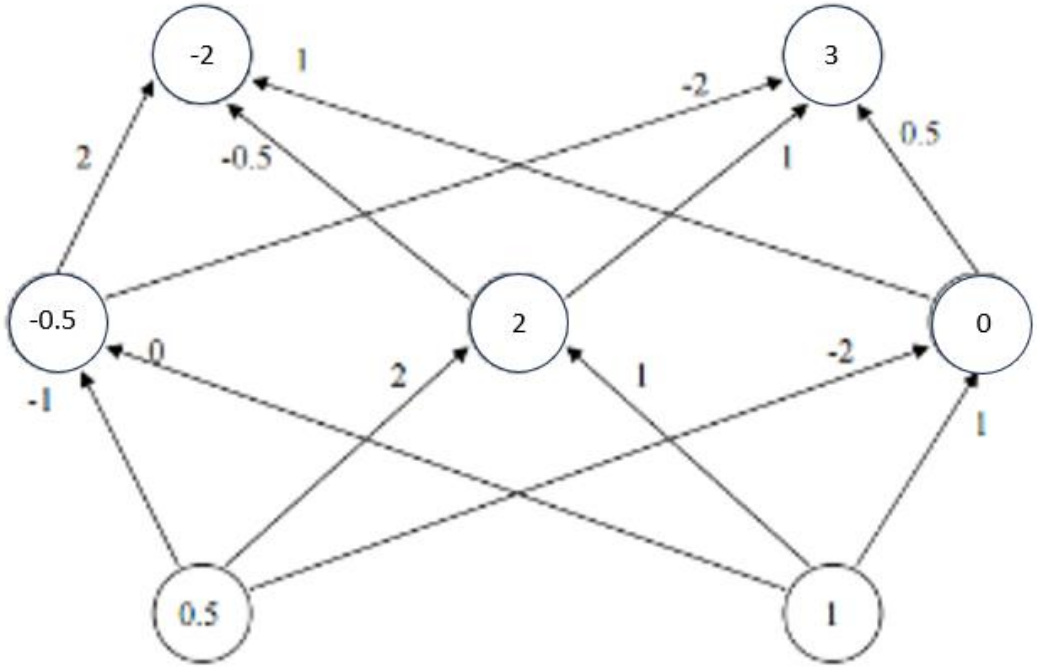
Bronco ID: 0 1 5 2 6 2 6 2 4

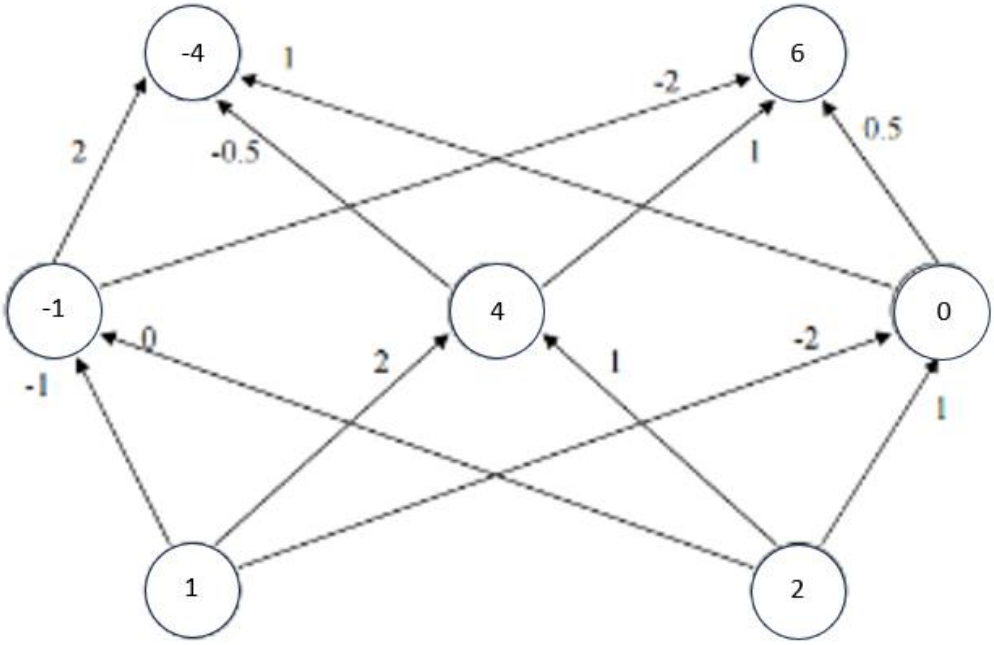
Last Name: Nguyen

First Name: Loc

1a. Result is as given for activation ➔ linear function where l(z) = a.z + b. Assuming b = 0, l(z) = a.z where given input (0.5, 1)

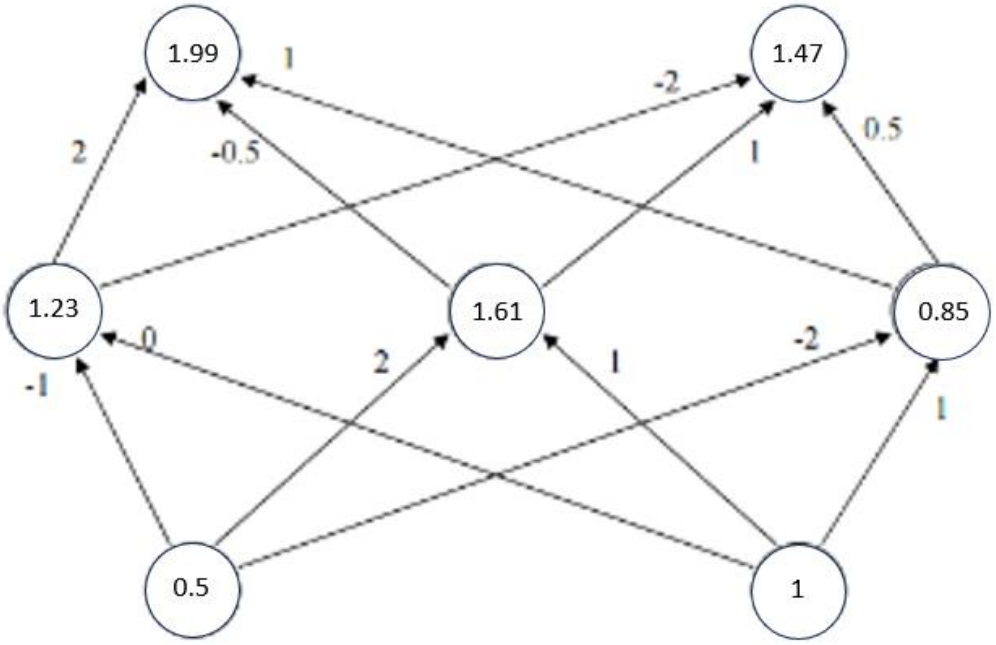


1b. Result is as given for activation ➔ linear function where l(z) = a.z + b. Assuming b = 0, l(z) = a.z where given input (1,2)

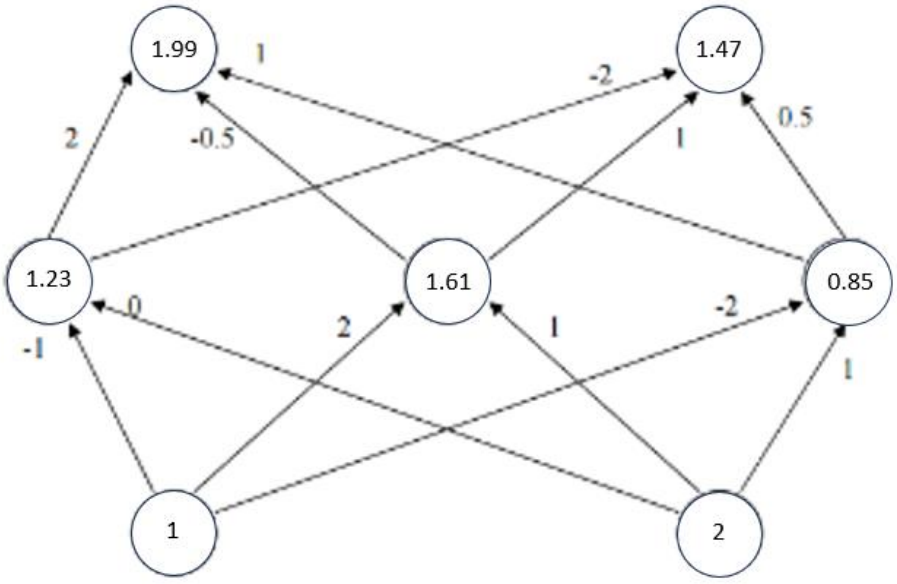


We need to repeat the computation because values in the “hidden” layer is calculated based on initial values and initial values has changed since the activation function is linear. Therefore, the “hidden” layer values will be different as well so we must recalculate.

1c. Result is as given for activation ➔Sigmoid function where $s(z) = 1 / (1 + e^{-z})$ where given input (0.5, 1)



1d. 1c. Result is as given for activation ➔Sigmoid function where $s(z) = 1 / (1 + e^{-z})$ where given input (1, 2)



We do not need to repeat the calculation because the activation function of Sigmoid function calculates values based on the weights between 2 nodes. However, the weights don't change despite the initial values changing so we don't need to recalculate.

2. https://github.com/Skyhorizon2021/CS_4210/blob/main/Assignment4/perceptron.py

3. https://github.com/Skyhorizon2021/CS_4210/blob/main/Assignment4/deep_learning.py

4. Dataset:

Outlook	Temperature	PlayTennis
Sunny	Hot	No
Overcast	Cool	Yes
Overcast	Hot	Yes
Rain	Cool	No
Overcast	Mild	Yes

Representation:

- Outlook <Sunny, Overcast, Rain>
- Temperature <Hot, Mild, Cool>

First generation:

		Prediction	Fitness	Place Ranking
C1	1001001	XXXOX	0.2	4th
C2	0100101	OXXOO	0.6	2nd
C3	1011000	OOOXO	0.8	1st
C4	1101100	OOXXX	0.4	3rd

Crossover C1 and C3 with mask 1110000 to produce two offspring of 1001000 and 1011001. These two offspring will replace C1 and C4 during the second generation since C1 and C4 has the lowest fitness.

C1 = 1001000; C2 = 0100101; C3 = 1011000; C4 = 1011001

Second generation:

		Prediction	Fitness	Place Ranking
C1	1001000	OOOXO	0.8	1st
C2	0100101	OXXOO	0.6	2nd
C3	1011000	OOOXO	0.8	1st
C4	1011001	XXXOX	0.2	3rd

Crossover C1 and C2 with mask 0001100 to produce two offspring of 1000100 and 0101001. These two offspring will replace C2 and C4 for worst performance.

C1 = 1001000; C2 = 1000100; C3 = 1011000; C4 = 0101001

Applying mutation on C3

		Mutation	Prediction	Fitness	Place Ranking
C1	1001000	X	OOOXO	0.8	2nd
C2	1000100	X	XOOXO	0.6	3rd
C3	1011000	1011010	OOOOO	1.0	1st
C4	0101001	X	OXOOX	0.6	3rd

Final Answer: C3 = 1011010 with accuracy = 1.0

5. Parameters given

Maximum weight capacity (C) = 15 kg.

Object	Tablet	Laptop	Projector
Weight (w)	5 kg	8 kg	10 kg
Value (v)	$ 570.00	$ 710.00	$ 640.00

Representation:

- Inclusion <1st, 2nd, 3rd>

First generation:

		Total weight (kg)	Fitness (values)
C1	000	0	0 (4th)
C2	001	10	640 (2nd)
C3	010	8	710 (1st)
C4	100	5	570 (3rd)

Crossover C2 and C4 with mask 110 to produce 101 and 000.

Crossover C2 and C3 with mask 110 to produce 011 and 000. However, 011 exceeds limit so we must use 010

C1 = 101; C2 = 000; C3 = 010; C4 = 000

Second generation:

		Total weight (kg)	Fitness (values)
C1	101	15	1210 (1st)
C2	000	0	0 (3rd)
C3	010	8	710 (2nd)
C4	000	0	0 (4th)

Crossover C1 and C1 with mask 100 to produce 101 and 101

Crossover C1 and C3 with mask 100 to produce 110 and 001.

C1 = 101; C2 = 101; C3 = 110; C4 = 001

Applying mutation on 101

		Total weight (kg)	Fitness (values)
C1	100	5	570 (3rd)
C2	100	5	570 (3rd)
C3	110	13	1280 (1st)
C4	001	10	640 (2nd)

Final Answer: C3 = 110 with max value of 1280 dollars and weight = 13 kg