Mawaba Pascal Dao

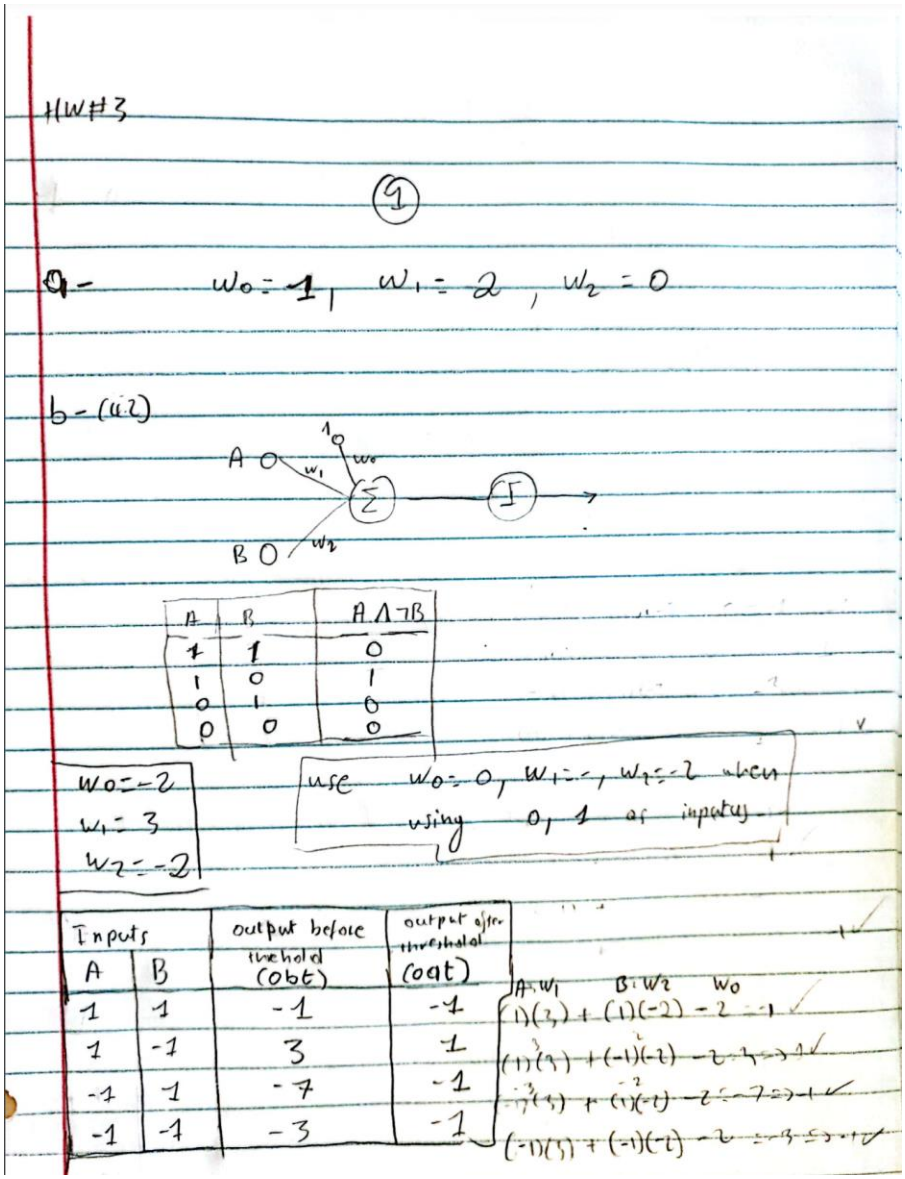
ML HW3

CSE5693

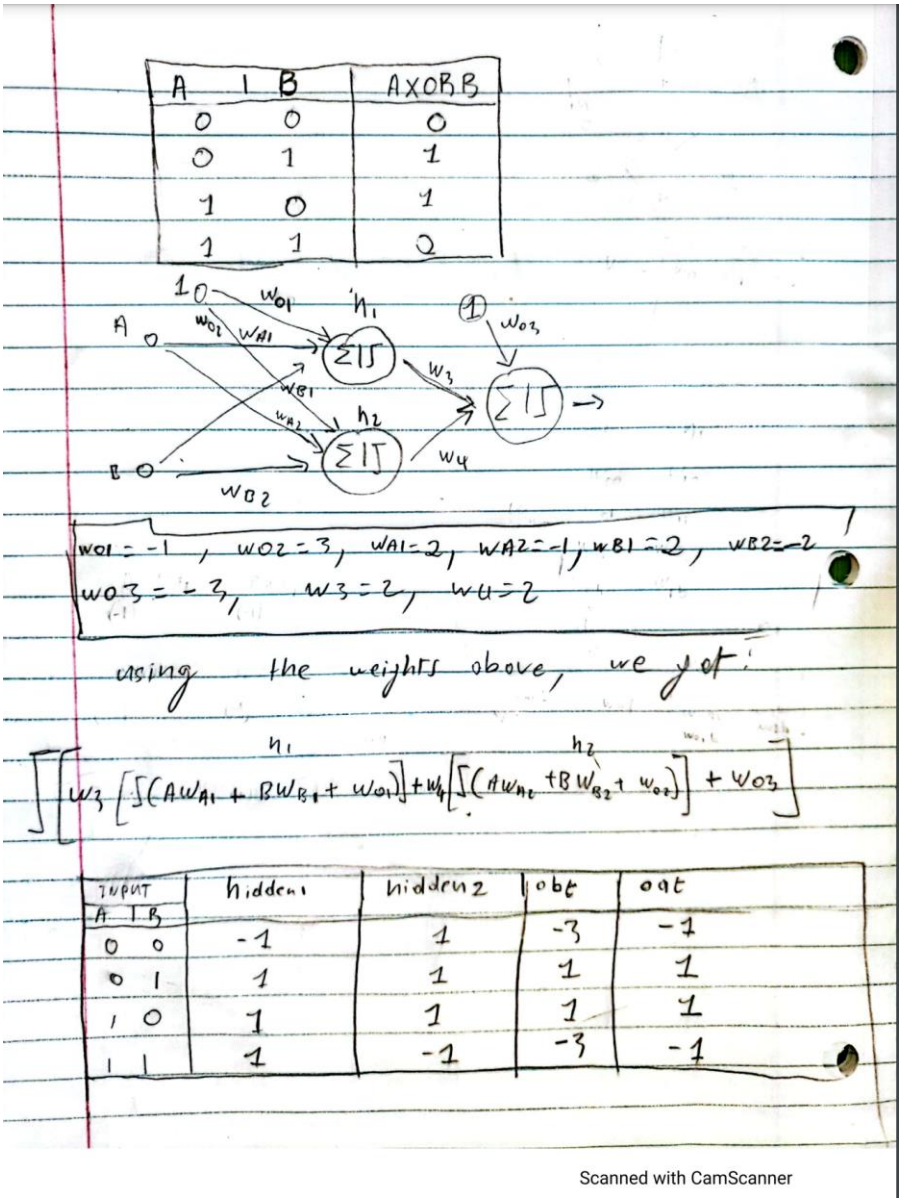
Dr. Chan

Mar 16, 2022

a.



b.



A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

$w_{01} = -1,\ w_{02} = 3,\ w_{A1} = -2,\ w_{A2} = -1,\ w_{B1} = 2,\ w_{B2} = -2$

$w_{03} = -3,\ w_3 = 2,\ w_4 = 2$

using the weights above, we get:

$$\int\left[w_3\left[\int(A w_{A1} + B w_{B1} + w_{01})\right] + w_4\left[\int(A w_{A2} + B w_{B2} + w_{02})\right] + w_{03}\right]$$

(h_1 = hidden1, h_2 = hidden2)

INPUT A	INPUT B	hidden1	hidden2	out	out
0	0	-1	1	-3	-1
0	1	1	1	1	1
1	0	1	1	1	1
1	1	1	-1	-3	-1

c.

There exist hidden unit weights that create the desired encoding. For example, w = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8] will successfully encode the input 8-unit representation into a single unit. However, a single unit encoding cannot be decoded by the output units' weights to 8 outputs. It is not possible to decode from 1 unit back to 8 units.

d. From programming assignment

i. 4 hidden values work because the network does not necessarily learn the standard binary encoding for 8 values. The network learns the best representation that minimizes the error. Although, 3 bits is the minimum number of hidden units required to decode 8 values, there is now upper bound.

   Using 4 hidden units converges significantly faster than using 3 hidden units. This was tested by setting a number max iterations to a small number (10) and observing that the 3-hidden units network achieves an accuracy of 25% while the 4-hidden unit's network achieves an accuracy of 100%. This is because the network has more representational “freedom” with 4 hidden units. With 3, it is forced to find the binary encoding necessarily, with 4 it is free to find any 4-bit representation that works.

ii. The performance with and without validation is comparable. Due to the early stop condition when using validation, we expect the network to be less prone to overfitting, however, we also risk interrupting training before a local minimum is found. The latter situation happens at x = 8 (i.e noise = 8*0.02 = 0.16) where the performance on the test set is significantly worse than without validation (See Fig 3). However, notice that at noise=0.2 the performance using validation is better (0.96) compared to no validation (0.94). This is because the early stop condition was met and the training was interrupted thus avoiding overfitting as shown in the image below

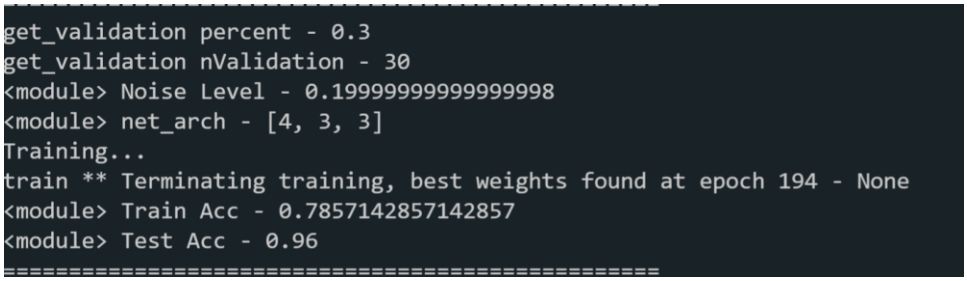


Fig1. With validation

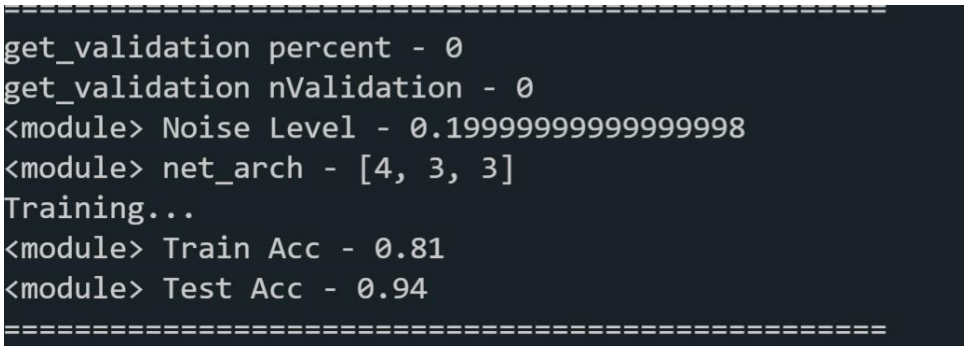


Fig2. Without validation

Notice that with validation (1st image) the accuracy is worse on the training set compared to without validation (2nd image)

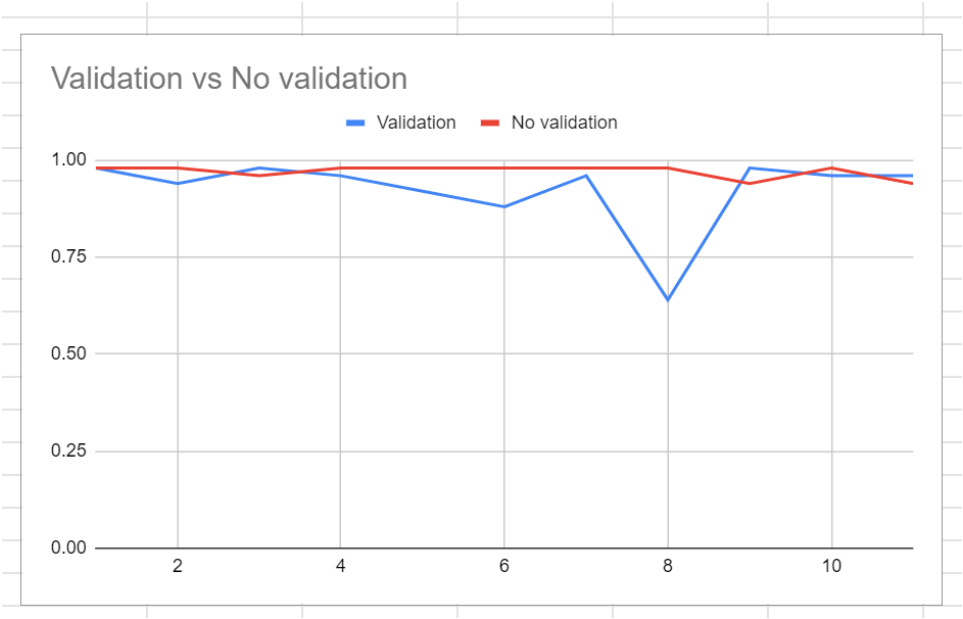


Fig 3. Comparison plot