Learning from Data - Week 4

Assignment 4

Mart Busger op Vollenbroek
S2174634

4.1.2

21,27%

4.1.3

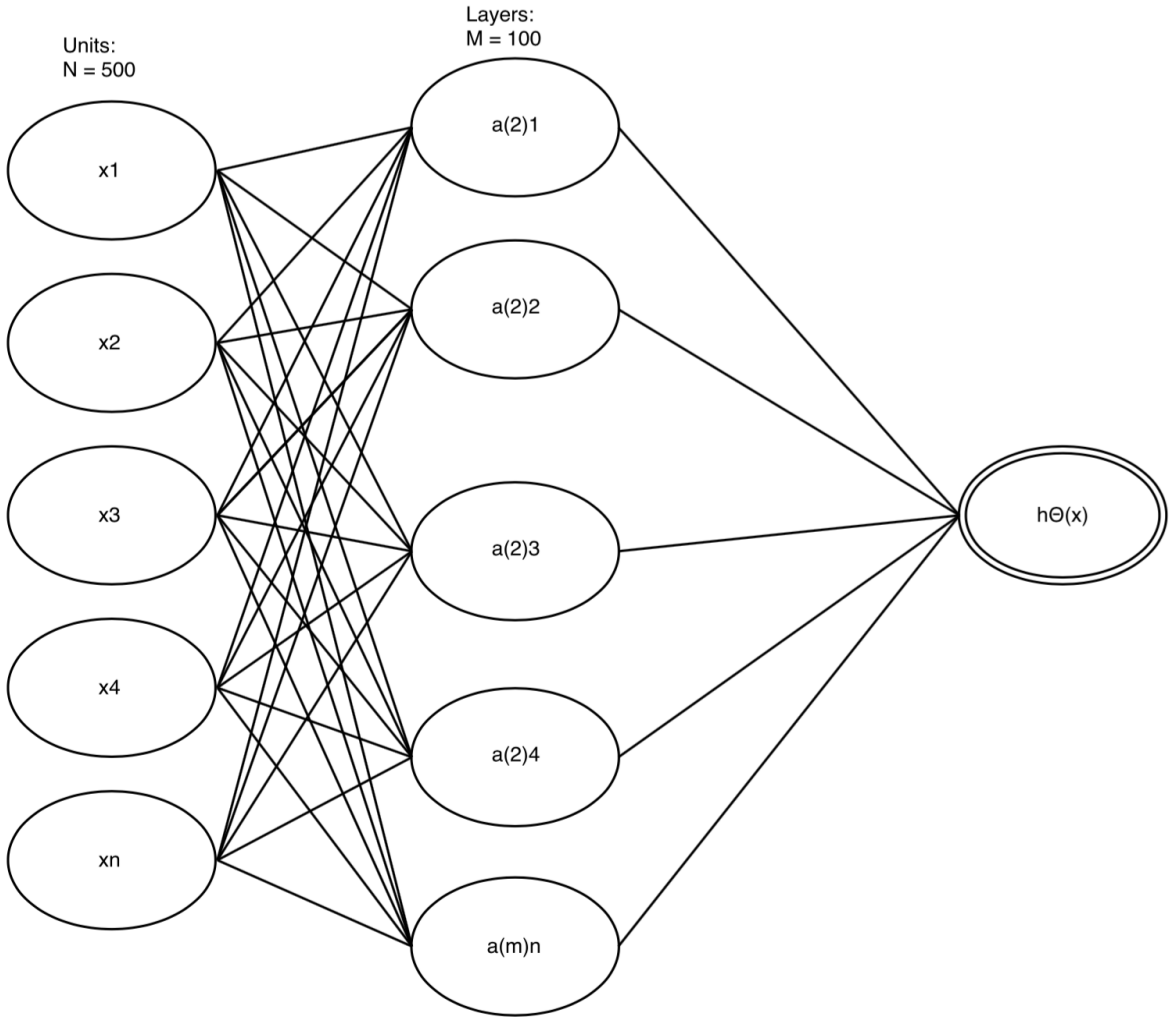
Settings —>	Units	Epochs	Bsize	Extra parameters	Train accuracy	Validation accuracy
Default	500	5	50		18,23	21,27
Setting 1	500	100	25		71,17	65,62
Setting 2	500	100	50		60,02	57,24
Setting 3	500	100	10		81,30	71,30
Setting 4	1000	50	10		73,61	68,23
Setting 5	500	100	5		87,37	70,68
Setting 6	500	100	10	lr=0,05 decay=0,1	94,76	70,03
Setting 7	500	100	10	Dropout(0.5)	81,30	71,30
Setting 8	500	100	10	Dropout(0.2)	99,62	71,36

4.1.4

The system I handed in which had the highest score on the test data was the system with setting 3. Because of the time it took to understand everything in the script, I haven't tested much with different settings in the lab. I tried altering the amount of units, epochs and the batch size used by the neural network. I found out that changing the amount of units per layer did not do much, close to nothing. Using a large amount of epochs (or layers), the accuracy went up for quite a bit but the time needed for the model te run also went up a lot. Lowering the batch size from 50 to 10 also helped with achieving higher scores. I used the standard SGD optimizer and tried out a few possibilities (only one is mentioned in the table above), but that did not seem to work for me. In the lab after the deadline had passed, I heard that using another optimizer would have had a larger impact on the accuracy score. I also tried using Dropout, which is used to reduce the difference between the training accuracy and the validation accuracy by randomly discarding some of weights. This however only seemed to boost my training accuracy, while the validation accuracy more or less stayed the same. After the lab I tested the script with MaxoutDense, which greatly improved the accuracy scores and I got even better scores using the 'Adam' optimizer.

4.1.5

Visual representation: Not sure how to do it, N is the amount of neurons per layer and M is the amount of layers.



4.2.1

1. - 4.

X1	X2	a(2)1	a(2)2	a(3)1	Result	a(2)1 Sigmoid	a(2)1 ReLU
0	**0**	10 (1)	-10 (0)	10 (1)	1	1	1
0	**1**	10 (1)	-30 (0)	10 (1)	1	1	1
1	**0**	-10 (0)	10 (1)	10 (1)	1	0	0
1	**1**	-10 (0)	-10 (0)	-10 (0)	0	0	0

2. Sigmoid: 1 / (1 + e^-x)

3. ReLU: 0 for x < 0, x for x = 0 ^ x > 0

4. Results from Sigmoid voor a2,1 a2,2 and a3,1 are in brackets.

5. The NAND logical function.