4.1.2:
Installation finally works after a lot of trying (windows..)

```
Train on 12260 samples, validate on 3066 samples
Epoch 1/5
12260/12260 [==============================] - 2s - loss: 3.1072 - acc: 0.1684 - val_loss: 3.0460 -
val_acc: 0.1774
Epoch 2/5
12260/12260 [==============================] - 1s - loss: 3.0506 - acc: 0.1732 - val_loss: 3.0290 -
val_acc: 0.1774
Epoch 3/5
12260/12260 [==============================] - 1s - loss: 3.0347 - acc: 0.1790 - val_loss: 3.0126 -
val_acc: 0.1778
Epoch 4/5
12260/12260 [==============================] - 1s - loss: 3.0200 - acc: 0.1786 - val_loss: 3.0020 -
val_acc: 0.1778
Epoch 5/5
12260/12260 [==============================] - 1s - loss: 3.0051 - acc: 0.1823 - val_loss: 2.9854 -
val_acc: 0.2127
```

4.1.3:
see results.pdf

4.1.4:
The choices for the architecture have been purely based on achieving a high as possible score, losing sight of most other measures,
such as the difference between the performance on the training data and the development set. Unfortunately, although performance is great,
my system is very overfit to the training data, making it to specific to do that well on the test data.
The used method has been very empirically: try changing one parameter > test it > higher score? leave it there /lower score? undo
By working this way, only one parameter was changed per try, and so several parameters at once was not tested, although they might perform better together than alone.

4.1.5:
see NN_visualisation.pdf
- Not sure about the number of layers, not specified in the script (nr of epochs is the number of loops the system makes over the same network i thought.)
- Not sure about the size of the hidden layer. 2x layer1? layer1!? layer1Â²?

4.2:

See TT.pdf

1:

x1	x2	(3)a1
0	0	True(30)
0	1	True(10)
1	0	True(10)
1	1	False(-10

2:

x1	x2	(3)a1	Sigmoid (2)a1

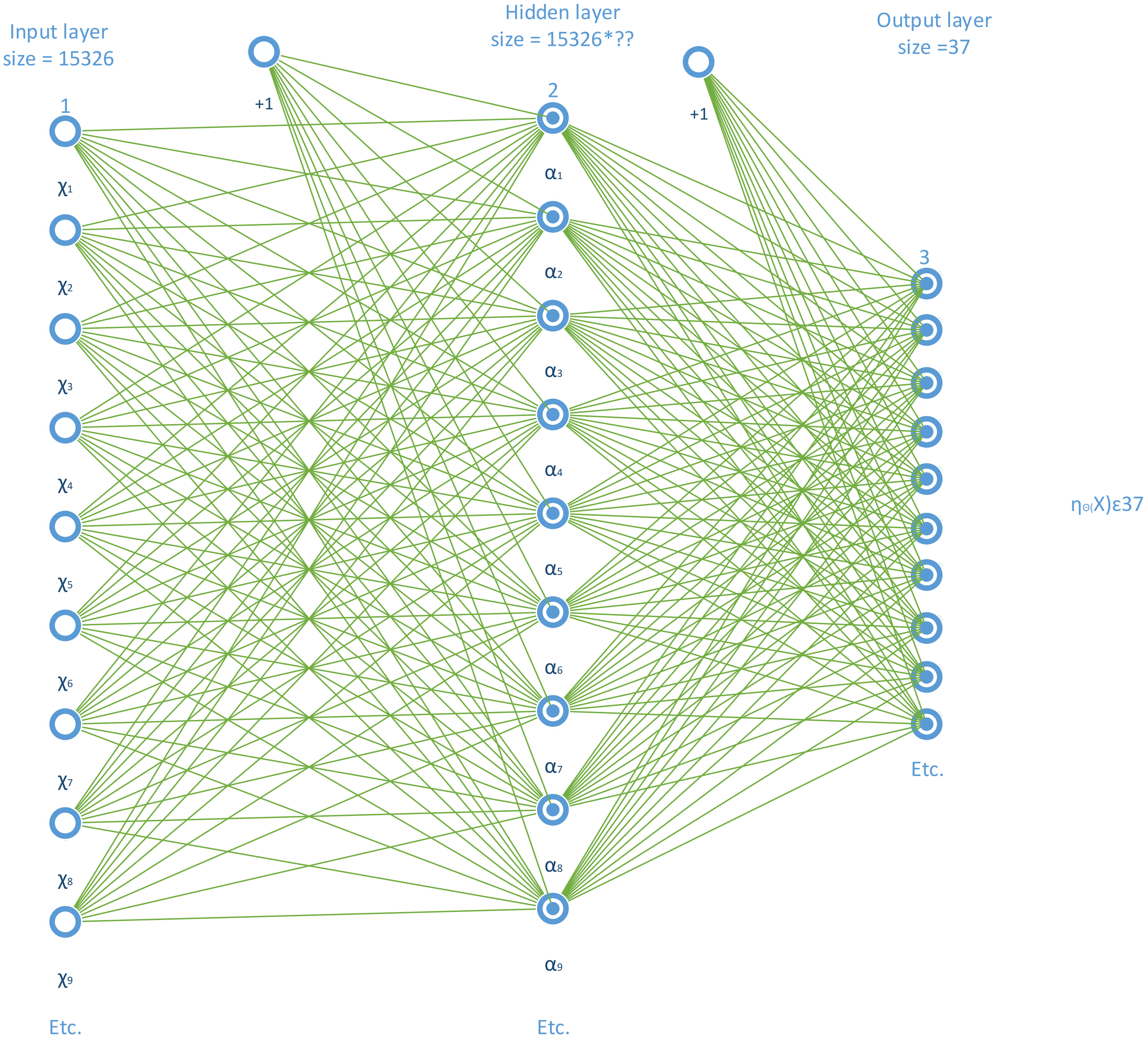
0	0	True(30) = 1	10
0	1	True(10) = 1	10
1	0	True(10) = 1	-10
1	1	False(-10) = 0	-10

3: Rectified linear starts at zero, and a sigmoid starts at -1.

x1	x2	(3)a1	Result (3)a1	Sigmoid (2)a1	Result Sigmoid (2)a1	Sigmoid (2)a2	Result3 Sigmoid (2)a2	Relu (2)a1	Result Relu (2)a1
0	0	30	1	10	1	-10	0	10	1
0	1	10	1	10	1	-30	0	10	1
1	0	10	1	-10	0	10	1	-10	0
1	1	-10	0	-10	0	-10	0	-10	0
			NAND		?		?		?

4: I do not recognize the outputs as being functions. 1100 and 0010

5: NAND


Input layer
size = 15326
+1
Hidden layer
size = 15326*??
+1
Output layer
size =37
1
χ_1
χ_2
χ_3
χ_4
χ_5
χ_6
χ_7
χ_8
χ_9
Etc.
2
α_1
α_2
α_3
α_4
α_5
α_6
α_7
α_8
α_9
Etc.
3
Etc.
$\eta_{\Theta(}X) \varepsilon 37$

Units	Epochs	bsize	optimizer	core layers	ACC	ACC_VAL
500	default	default	default	default	0.1823	0.2127
1500	default	default	default	default	0.1889	0.1791
2500	default	default	default	default	0.1866	0.2629
3500	default	default	default	default	0.1576	0.1676
2500	100	default	default	default	0.6116	0.5903
500	100	default	default	default	0.6002	0.5724
500	100	100	default	default	0.4754	0.4628
500	100	10	default	default	0.8131	0.7120
500	100	10	Adam	default	0.9980	0.7312
500	50	50	Adam	default	0.9921	0.7175
500	50	50	RMSprop	default	0.9248	0.7081
100	50	10	Adadelta	default	0.8330	0.6986
500	5	50	Adam	drop+maxd	0.8907	0.7436
500	10	50	Adam	drop+maxd	0.9243	0.7547
500	20	50	Adam	drop+maxd	0.9982	0.7736
500	20	50		maxoutden.	0.9982	0.7736

x1	x2	(3)a1	Result (3)a1	Sigmoid (2)a1	Result Sigmoid (2)a1
0	0	30	**1**	10	**1**
0	1	10	**1**	10	**1**
1	0	10	**1**	-10	**0**
1	1	-10	**0**	-10	**0**
			NAND		**?**

Sigmoid (2)a2	Result3 Sigmoid (2)a2	Relu (2)a1	Result Relu (2)a1
-10	**0**	10	**1**
-30	**0**	10	**1**
10	**1**	-10	**0**
-10	**0**	-10	**0**
	?		**?**