Machine Learning CSE 5693

Assignment 3 – Artificial Neural Network

Written Assignment:

Book assignment:

4.1 Assume the function is:

$$y = ax + b$$

This line cross the point (-1, 0), (0, 2). So the function will be:

$$y = 2x + 2$$

$$0 = 2 + 2x - y$$

So, w0 = 2, w1 = 2, w2 = -1.

4.9 I think it could be learned, but when I test it the result is not good enough. But I also test this problem with only one input (1.0, 2.0 … 8.0). The network can do this task very well (but must with more than 8 hidden units, one hidden layer). So I think the problem is In the 1 x N x 8 network, I artificially separate the input value to 1.0 ~ 8.0. So the network can learning it. But in this case, which is 8 x 1 x 8 network, the one hidden layer cannot separate those number well by self-learning. So the output layer cannot know what different or the boundary of each input (the 1 hidden output). And in the 8 x 1 x 8 network, for the only one hidden node, the 8 weight will update individually. But actually, the input and delta for each node is same. (This network input and output has some kind of symmetrical if we look at whole data set.). So after training, the weight for the network didn't learning enough information. I think that is why the identity problem can't be learned by 8 x 1 x 8 network.

With the programming assignment:

I.

The output value for each node is a sigmoid function, in this case we can regard the value near 1.0 is 1 and near 0.0 is 0. So each hidden node represent 1 bit in this case. And 8 number can be represent by minimum 3 bit. So 4 bit will also work (any size greater than 3 will work in this situation).
The hidden layer represent the binary value of each input.
The value of each hidden layer node represent 1 bit.

II.

In my ANN algorithm, I don't see any performance changes between the using validation set

and not using the validation set. Because when I test the program, the curve of error rate always decrease both training and validation set. So the algorithm think the best approach is the last iteration. So the networks which algorithm used to predict the test accuracy is the same network. In this situation, validation didn't help much with predict a more accurate value.

Programming Result:

Identity Set:

Learning Rate:	0.9
Momentum:	0.3
Random Value Range:	-0.1 ~ 0.1
Hidden Layer:	1
Hidden Unit Per Layer:	3
Input Unit:	8
Output Unit:	8
Training Iterations:	10000

Input	Hidden Values			Output
1000 0000	0.00,	0.99,	0.91 (011)	0.98, 0.00, 0.00, 0.02, 0.01, 0.00, 0.00, 0.01
0100 0000	0.99,	1.00,	0.04 (110)	0.00, 0.98, 0.00, 0.02, 0.00, 0.00, 0.02, 0.01
0010 0000	0.99,	0.00,	0.76 (101)	0.00, 0.00, 0.97, 0.02, 0.01, 0.00, 0.01, 0.00
0001 0000	0.99,	0.98,	1.00 (111)	0.00, 0.01, 0.00, 0.97, 0.00, 0.00, 0.00, 0.00
0000 1000	0.13,	0.10,	0.99 (001)	0.02, 0.00, 0.02, 0.00, 0.98, 0.01, 0.00, 0.00
0000 0100	0.02,	0.01,	0.07 (000)	0.00, 0.00, 0.00, 0.00, 0.01, 0.98, 0.01, 0.02
0000 0010	0.96,	0.20,	0.00 (100)	0.00, 0.02, 0.02, 0.00, 0.00, 0.01, 0.98, 0.00
0000 0001	0.06,	0.90,	0.01 (010)	0.02, 0.02, 0.00, 0.00, 0.00, 0.01, 0.00, 0.98

Learning Rate:	0.9
Momentum:	0.3
Random Value Range:	-0.1 ~ 0.1
Hidden Layer:	1
Hidden Unit Per Layer:	4
Input Unit:	8
Output Unit:	8
Training Iterations:	10000

Input	Hidden Values				Output
1000 0000	0.64,	0.01,	0.02,	0.80	0.99, 0.01, 0.00, 0.01, 0.01, 0.00, 0.01, 0.00
0100 0000	0.99,	0.99,	0.44,	0.98	0.01, 0.98, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00
0010 0000	0.03,	0.98,	0.99,	0.99	0.00, 0.01, 0.98, 0.01, 0.01, 0.01, 0.00, 0.00
0001 0000	0.01,	0.95,	0.01,	0.75	0.00, 0.00, 0.01, 0.98, 0.00, 0.00, 0.01, 0.00
0000 1000	0.01,	0.01,	0.78,	0.52	0.01, 0.00, 0.01, 0.00, 0.98, 0.01, 0.00, 0.01
0000 0100	0.02,	0.95,	0.87,	0.01	0.00, 0.00, 0.01, 0.00, 0.00, 0.99, 0.01, 0.00
0000 0010	0.71,	0.75,	0.02,	0.02	0.01, 0.01, 0.00, 0.01, 0.00, 0.01, 0.98, 0.01
0000 0001	0.97,	0.04,	0.97,	0.02	0.00, 0.0, 0.00, 0.00, 0.01, 0.00, 0.01, 0.99

Tennis Set:

Learning Rate:	0.6	
Momentum:	0.15	
Random Value Range:	-0.1 ~ 0.1	
Hidden Layer:	1	
Hidden Unit Per Layer:	3	
Input Unit:	4	(Outlook, Temperature, Humidity, Wind)
Output Unit:	2	(TRUE, FALSE)
Training Iterations:	20000	

Training Set Accuracy:	0.9	(Sometimes 1.0)
Test Set Accuracy:	0.25	(Sometimes 0.5)

Learning Rate:	0.6	
Momentum:	0.15	
Random Value Range:	-0.1 ~ 0.1	
Hidden Layer:	1	
Hidden Unit Per Layer:	3	
Input Unit:	11	(Each value for each attribute is an input unit)
Output Unit:	2	(TRUE, FALSE)
Training Iterations:	20000	

Training Set Accuracy:	1.0
Test Set Accuracy:	0.5

Iris Set:

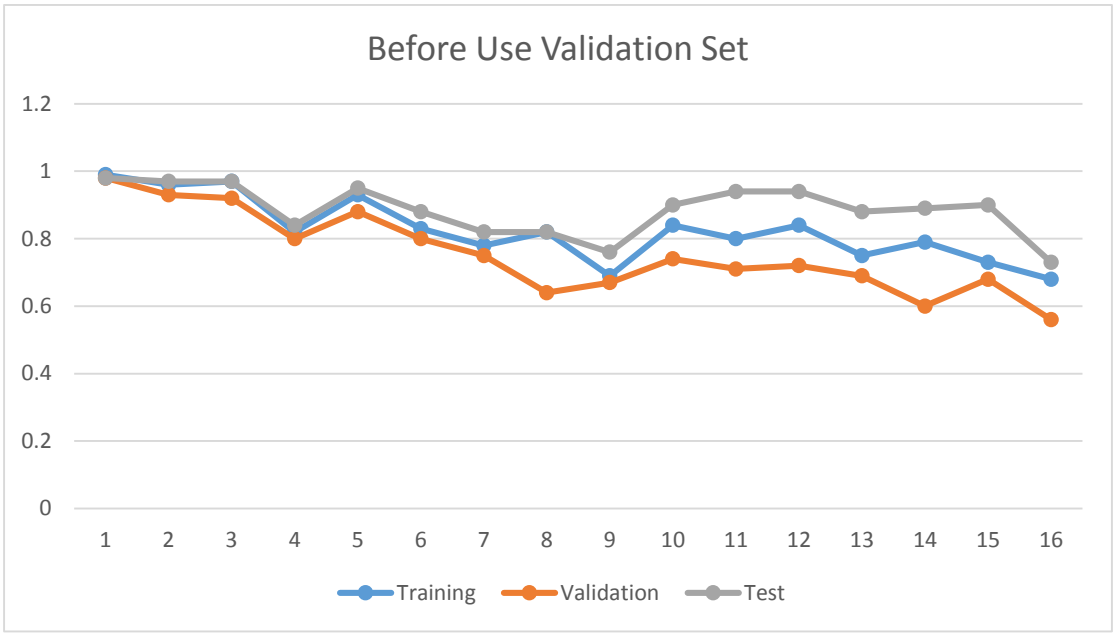
Learning Rate:	0.9
Momentum:	0.3
Random Value Range:	-0.1 ~ 0.1
Hidden Layer:	1
Hidden Unit Per Layer:	5
Input Unit:	4 (sepal-length, sepal-width, petal-length, petal-width)
Output Unit:	3 (Iris-setosa, Iris-versicolor, Iris-virginica)
Training Iterations:	20000

Training Set Accuracy:	0.99
Test Set Accuracy:	0.96

Iris Set With Noise:

Learning Rate:	0.9
Momentum:	0.3
Random Value Range:	-0.1 ~ 0.1
Hidden Layer:	1
Hidden Unit Per Layer:	5
Input Unit:	4 (sepal-length, sepal-width, petal-length, petal-width)
Output Unit:	3 (Iris-setosa, Iris-versicolor, Iris-virginica)
Training Iterations:	20000

Noise:	0.00	0.02	0.04	0.06	0.08	0.10	0.12	0.14	0.16	0.18	0.20	0.22	0.24	0.26	0.28	0.30
Before Pruning																
Training:	0.99	0.96	0.97	0.82	0.93	0.83	0.78	0.82	0.69	0.84	0.80	0.84	0.75	0.79	0.73	0.68
Validation:	0.98	0.93	0.92	0.80	0.88	0.80	0.75	0.64	0.67	0.74	0.71	0.72	0.69	0.60	0.68	0.56
Test:	0.98	0.97	0.97	0.84	0.95	0.88	0.82	0.82	0.76	0.90	0.94	0.94	0.88	0.89	0.90	0.73
After Pruning																
Training:	0.98	0.96	0.97	0.82	0.93	0.83	0.77	0.82	0.69	0.84	0.80	0.84	0.75	0.78	0.72	0.68
Validation:	0.98	0.93	0.92	0.80	0.88	0.80	0.75	0.64	0.67	0.75	0.72	0.72	0.69	0.60	0.69	0.55
Test:	0.97	0.97	0.97	0.84	0.95	0.88	0.82	0.82	0.76	0.90	0.93	0.94	0.88	0.92	0.90	0.73


Before Use Validation Set
1.2
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Training
Validation
Test