

# CS534 — Homework Assignment 3 — Solutions

## 1. Table of results:

hw2-1:

N	J48	NeuralNet	1NN
10	0.500	0.470	0.495
20	0.313	0.318	0.368
50	0.313	0.288	0.365
100	0.310	0.265	0.358
200	0.305	0.270	0.335
400	0.280	0.248	0.358

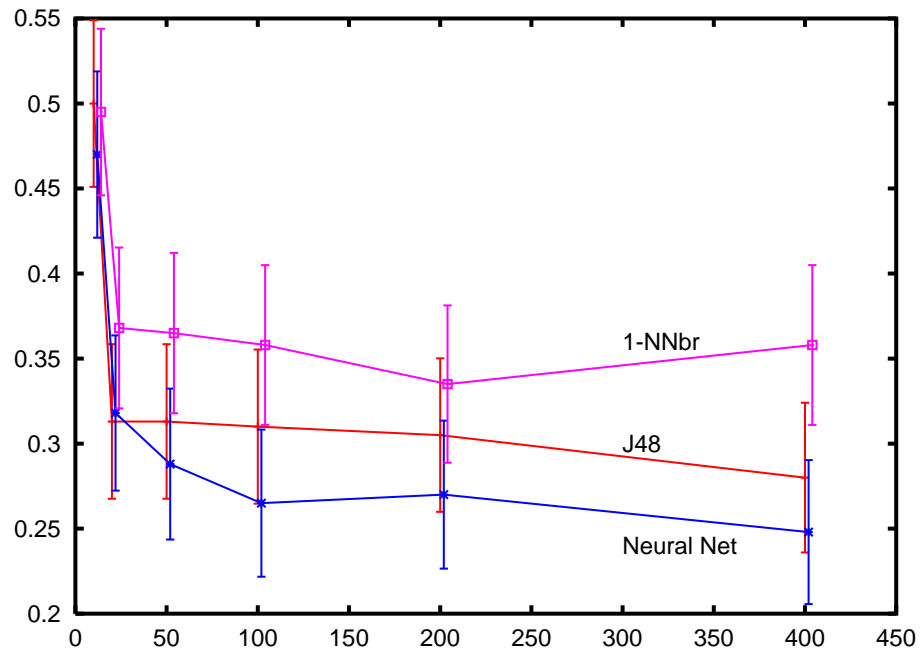
hw2-2:

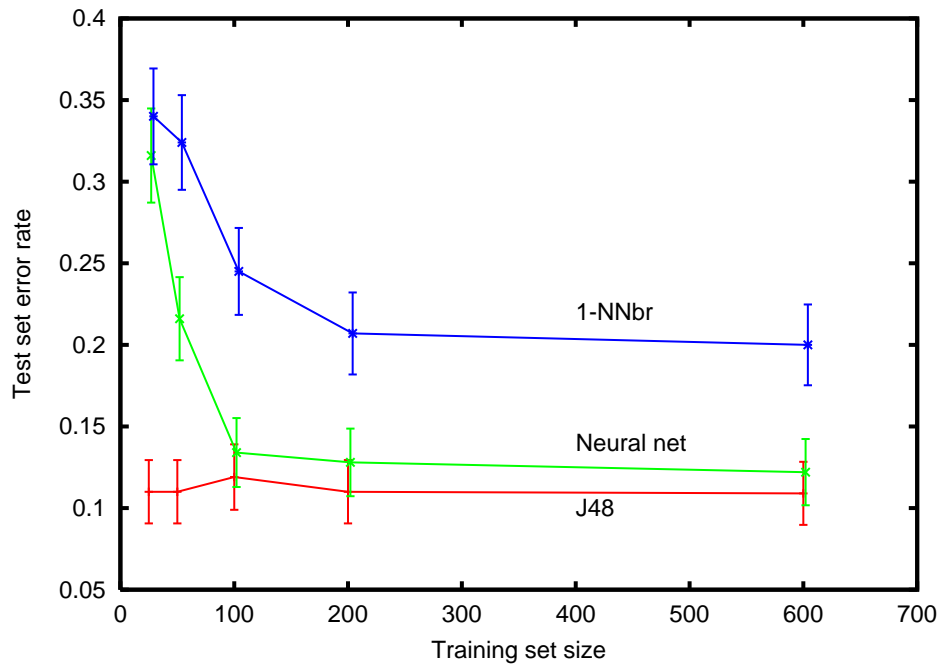
N	J48	NeuralNet	1NN
25	0.110	0.316	0.340
50	0.110	0.216	0.324
100	0.119	0.134	0.245
200	0.110	0.128	0.207
600	0.109	0.122	0.200

br:

N	J48	NeuralNet	1NN
614	0.1028	0.0457	0.0261

## 2. Graphs of results:





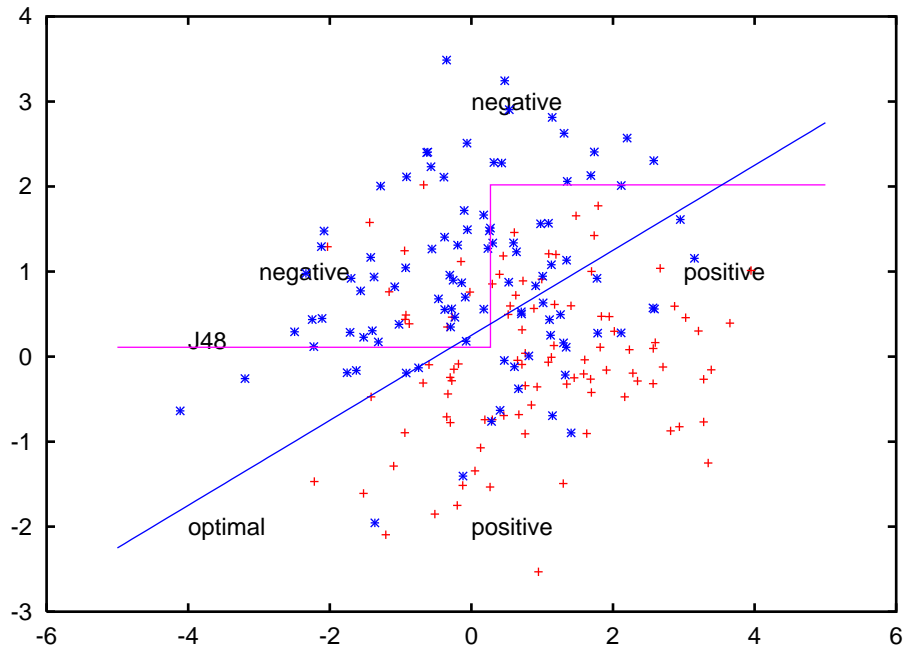
3. Plot of decision boundaries for decision trees.

Here is the J48 decision boundary for hw2-1-200

```

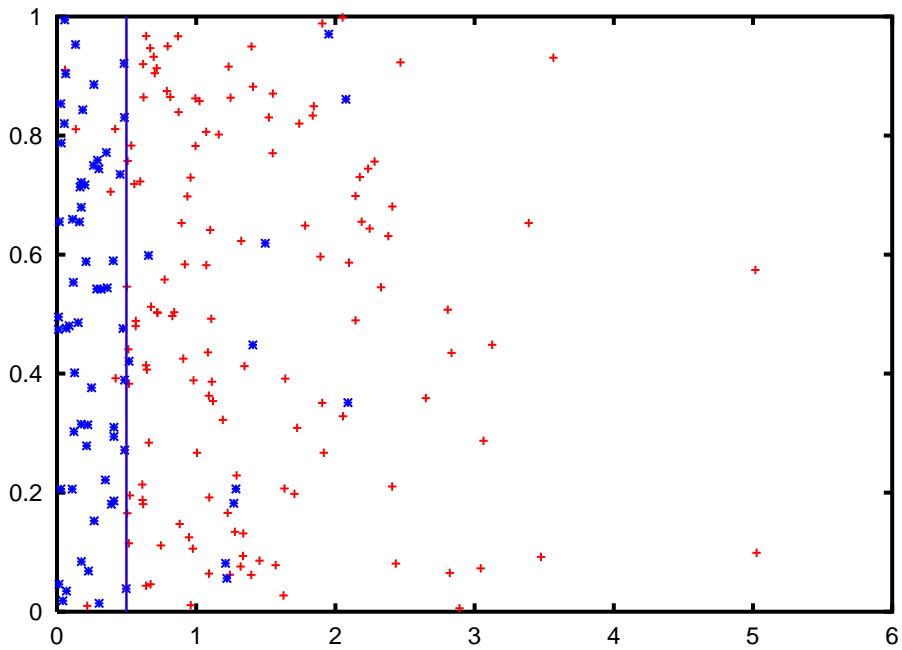
y <= 0.110082: positive (75.0/17.0)
y > 0.110082
|   y <= 2.019612
|   |   x <= 0.269892: negative (51.0/12.0)
|   |   x > 0.269892: positive (56.0/26.0)
|   y > 2.019612: negative (18.0)

```



Here is the J48 decision boundary for hw2-2-200

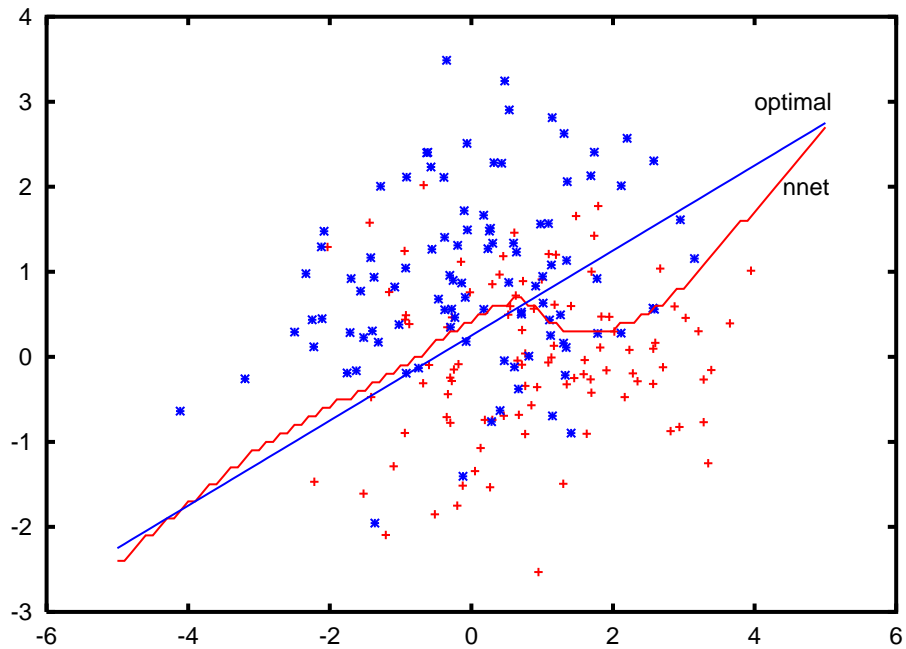
```
x <= 0.496702: negative (63.0/6.0)
x > 0.496702: positive (137.0/11.0)
```



Here, J48 has essentially found the optimal decision boundary (0.500 versus 0.497). Both lines are plotted in the figure, but they overlap completely.

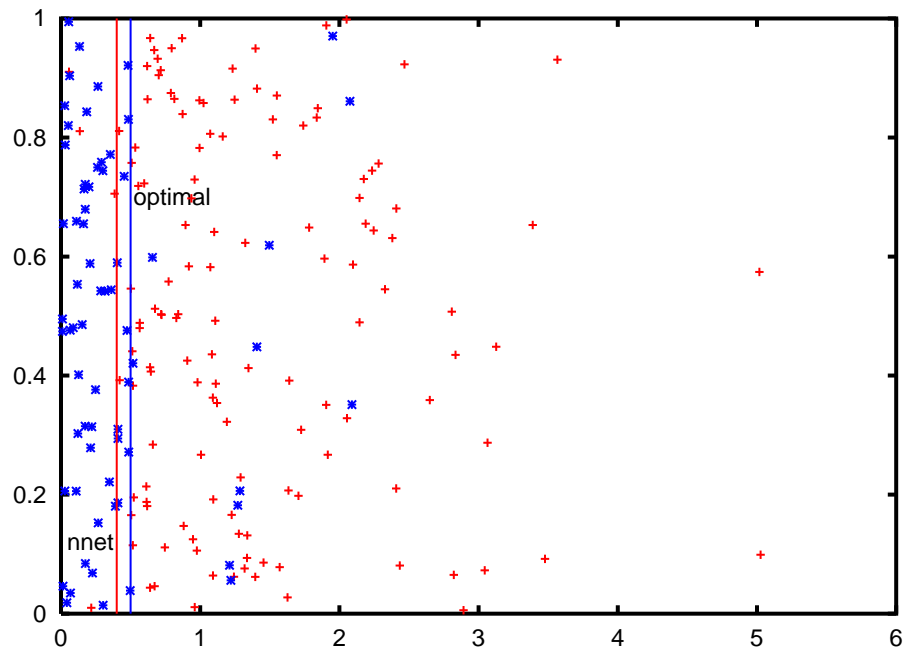
4. Plot of decision boundaries for neural networks.

Boundary for hw2-1:



The neural network did quite well here. It almost found the optimal boundary for  $x < 0$ , but a got a little confused for  $x > 0$ .

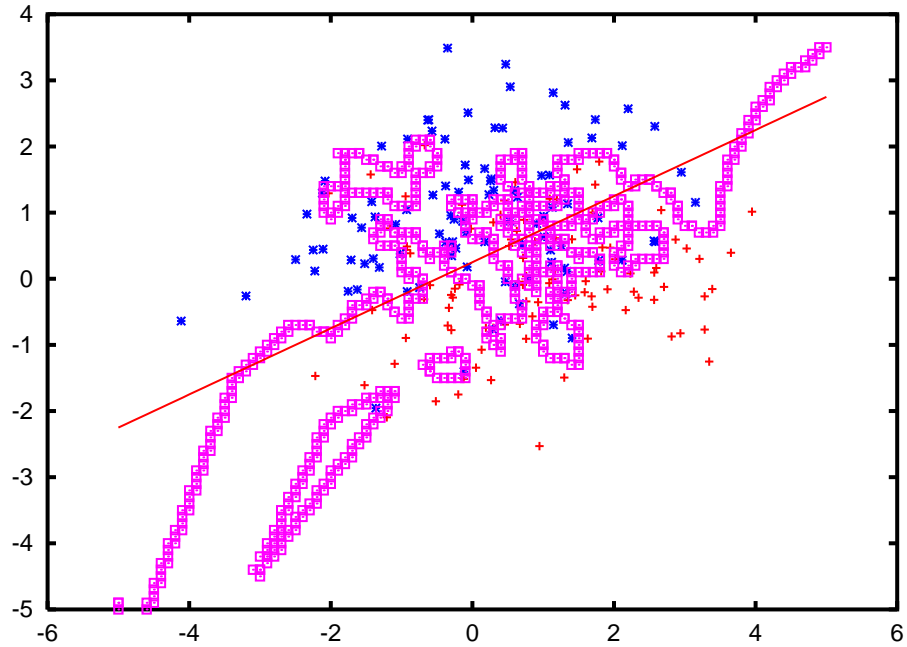
Boundary for hw2-2:



This graph is misleading, because the grid.arff file only has a resolution of 0.1, so we know that the neural network placed its boundary somewhere between  $x = 0.4$  and  $x = 0.5$ . So it has done nearly perfectly.

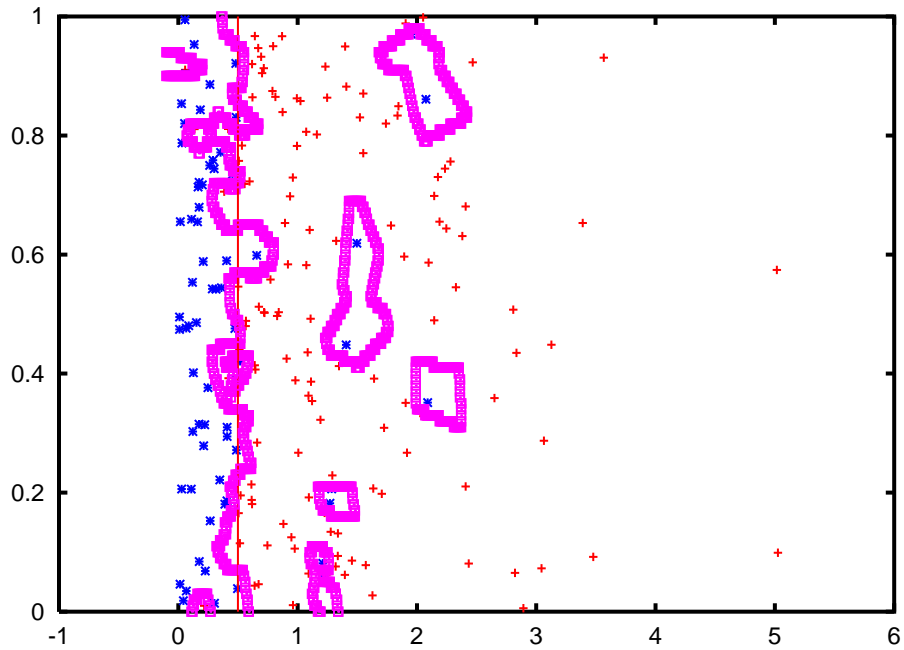
### 5. Decision boundary of IBk

Boundary for hw2-1:



To construct the 1-NNbr decision boundary, I scanned the grid in both the X and Y directions to find points where the predicted class changed. I then plotted all of these points. We can see that the nearest neighbor algorithm does very poorly on this data.

Boundary for hw2-2:



To construct this boundary, I generated a new “grid” data set with much finer sampling in the range of the hw2-2 data. I again scanned in both X and Y directions and plotted all boundary crossings. We can see that 1-NN does a very poor job here as well.

6. Three additional random seeds of the neural network code

seed	5 hidden	40 hidden
0	0.2875	0.2800
1234	0.2525	0.2625
4321	0.2750	0.2800
41234444	0.2825	0.2900

I notice two things here. First, there is a lot of random fluctuation depending on the random seed. Second, the larger network is maybe performing slightly worse (3 out of 4 seeds).

7. Increasing  $k$  in the k-nearest neighbors

k	error rate
1	0.365
3	0.350
5	0.323
7	0.280
9	0.263
11	0.245
13	0.250
15	0.253

It looks like 11 neighbors gives the best performance.