CS3600 P5 Writeup

Paras Jain

April 2016

1 Q5: Learning with restarts

Refer to Table 1 for the results of this phase of analysis.

Table 1: Results from Q5 - statistics and test performance over randomized 5-trial with default settings for hidden layers.

	μ	σ	max	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
Car Dataset	87.3%	0.8%	88.1%	87.8%	87.5%	88.1%	86.7%	86.0%
Pen Dataset	90.6%	0.4%	91.2%	90.1%	90.3%	91.1%	90.8%	90.6%

2 Q6: Varying the hidden layer

Refer to Figure 2 for a visualization of the learning curves over varying model complexities.

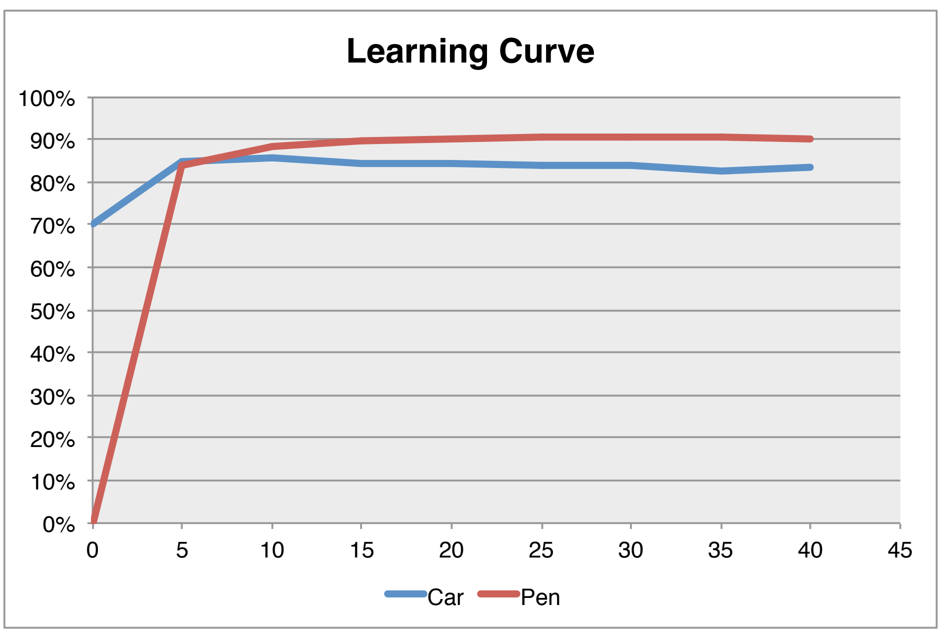

Figure 1: Learning curves over varying model complexities for the car and pen datasets. Refer to Table 2 for summary statistics of the data.

For the car dataset, it seems that beyond 5 perceptrons in the hidden layer, performance gains are minimal. In fact, for the car dataset, performance drops slightly past 10 perceptions. This is a sign of slight overfitting where adding more complexity fails to extract any more signal from the training dataset.

	Car			Pen		
# of Perceptrons	μ	σ	max	μ	σ	max
0	70.0%	0.0%	70.0%	0.0%	0.0%	0.0%
5	84.9%	2.2%	88.2%	83.8%	1.0%	85.3%
10	85.7%	1.0%	86.9%	88.5%	1.0%	89.7%
15	84.2%	0.9%	85.6%	89.5%	0.6%	90.5%
20	84.5%	1.1%	85.8%	90.2%	0.6%	90.9%
25	84.0%	0.9%	85.5%	90.5%	0.3%	91.0%
30	83.7%	1.6%	85.8%	90.6%	0.1%	90.7%
35	82.7%	1.4%	84.7%	90.4%	0.2%	90.7%
40	83.3%	1.1%	84.6%	90.1%	0.3%	90.5%

Table 2: Summary statistics for randomized 5-trial experiment of test performance over varying hidden layer sizes

This makes sense as there are few attributes in the car dataset so with 5 perceptrons, we have effectively constructed a hyperplane for each attribute to split the classification class. As the target hypothesis is not too complex (as seen in the last project with decision trees), 5 hidden perceptrons is sufficient. Even with just the output layer, performance is still quite good at 70%.

For the pen dataset, it seems that beyond 15 perceptrons in the hidden layer, marginal performance is minimal. Interestingly, there doesn't seem to be signs of overfitting in the test range. This result makes sense as the target hypothesis is much more complex than the car dataset. However, with just 16 input perceptrons, it makes sense that an elbow was observed at 15 hidden perceptrons occurs.

3 Q7: XOR

See Figure 3 for the results over varying sizes of the hidden layer. Note that 0 corresponds to no hidden layer.

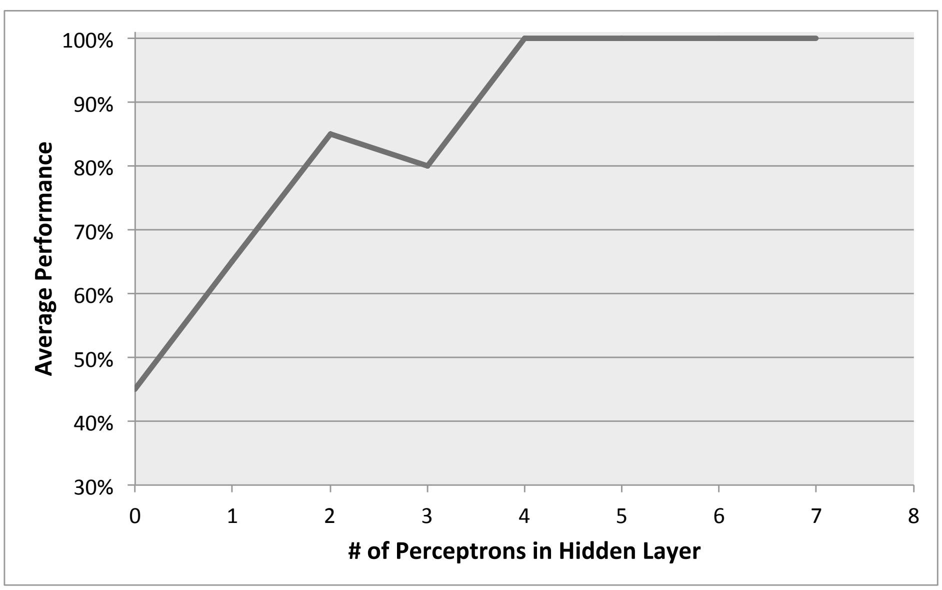

Figure 2: XOR performance over 5000 training iterations over varying hidden layer sizes. Randomized 5-trial average performance

Note that with no hidden layer, average performance is below 50%. With 4 perceptrons in the hidden layer, average performance is at 100%. This makes sense to me as the training dataset contains 4 instances

(2^2 length 2 input bitstrings) so with 4 perceptrons, each perceptron could learn a weighting for each example to produce an activation that would output the correct output bit.

4 Q8: Extra credit dataset (Iris flower dataset)

I chose the Iris flower classification dataset[1] as it is a very common dataset in the machine learning community and is relatively well understood.

Note: Before passing in each feature into the neural network, each attribute was normalized using zero-mean-unit-variance normalization ($x' = \frac{x-\mu_x}{\sigma_x}$). This just makes the values a bit easier for the neural net to process and also reduces training time by requiring less absolute scale for the NN weights.

Table 3: Results from Q8 - statistics and test performance over randomized 5-trial with 4 perceptrons in the hidden layer for the Iris flower classifiation dataset. 200 iterations max.

	μ	σ	max	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
Iris Dataset	92.8%	1.0%	94%	94%	92%	92%	94%	92%

These results (refer to Table 3) are quite good as the NN was able to classify the Iris flower dataset with a relatively high accuracy.

[1] https://archive.ics.uci.edu/ml/datasets/Iris