CS446: Machine Learning **Spring 2017**

Problem Set 5

Bangqi Wang *Handed In: April 12, 2017*

1. Answer to problem 1

   a. For this question, I will use i, j, k to represent input, hidden, and output layers:

$$\frac{\partial E_d}{\partial w_{jk}} = \frac{\partial E_d}{\partial o_k}\frac{\partial o_k}{\partial h_k}\frac{\partial h_k}{\partial w_{jk}} \tag{1}$$

$$\frac{\partial E_d}{\partial o_k} = -(t_k - o_k) \tag{2}$$

$$\frac{o_k}{h_k} = \begin{cases} 0, & h_k \leq 0 \\ 1, & h_k > 0 \end{cases} \tag{3}$$

$$\frac{h_k}{w_{kl}} = x_j \tag{4}$$

$$\Rightarrow w_{jk} \leftarrow w_{jk} + \gamma\sigma_k x_j \tag{5}$$

$$where, \quad \sigma_k = \frac{\partial E_d}{\partial h_k} = \begin{cases} 0, & h_k \leq 0 \\ -(t_k - o_k), & h_k > 0 \end{cases} \tag{6}$$

$$\gamma \text{ is the learning rate} \tag{7}$$

$$\frac{\partial E_d}{\partial w_{ij}} = \frac{\partial E_d}{\partial h_j}\frac{\partial h_j}{\partial w_{ij}} \tag{8}$$

$$= \sum_{k \in downstream(j)} \frac{\partial E_d}{\partial h_k}\frac{\partial h_k}{\partial h_j}\frac{\partial h_j}{\partial w_{ij}} \tag{9}$$

$$\frac{\partial E_d}{\partial h_k} = -\sigma_k \tag{10}$$

$$\frac{\partial h_k}{\partial h_j} = \begin{cases} 0, & h_j \leq 0 \\ w_{jk}, & h_j > 0 \end{cases} \tag{11}$$

$$\frac{\partial h_j}{\partial w_{ij}} = x_i \tag{12}$$

$$\Rightarrow w_{ij} \leftarrow w_{ij} + \gamma\sigma_j x_i \tag{13}$$

$$where, \quad \sigma_j = \frac{\partial E_d}{\partial w_{ij}} = \begin{cases} 0, & h_j \leq 0 \\ \sum_{k \in downstream(j)} -\sigma_k w_{jk} x_i, & h_j > 0 \end{cases} \tag{14}$$

$$\gamma \text{ is the learning rate} \tag{15}$$

b. ii. The tables below show the average accuracy for each parameter setting with ascending order for **CIRCLES** and **MNIST** datasets.

CIRCLES :

Accuracy%	batch_size	activation	learning_rate	layer_width
48.75	10	relu	0.1	50
49	100	tanh	0.01	10
49.75	50	tanh	0.01	10
50.875	50	tanh	0.01	50
52.125	100	tanh	0.01	50
60.375	100	relu	0.01	10
71.125	100	tanh	0.1	50
72.5	10	tanh	0.01	50
75.5	100	tanh	0.1	10
78.375	10	tanh	0.01	10
86.25	50	tanh	0.1	50
91.25	50	tanh	0.1	10
94.125	100	relu	0.01	50
96.625	50	relu	0.01	10
99.125	100	relu	0.1	10
99.75	10	relu	0.01	10
100	10	relu	0.01	50
100	10	relu	0.1	10
100	50	relu	0.1	10
100	10	tanh	0.1	10
100	10	tanh	0.1	50
100	50	relu	0.1	50
100	50	relu	0.01	50
100	100	relu	0.1	50

Table: Average Accuracy for **CIRCLES**

.

From the table above, we find that there are multiple parameter settings that can lead to 100% accuracy. The best parameter settings are shown in the table below.

Accuracy%	batch_size	activation	learning_rate	layer_width
100	10	relu	0.01	50
100	10	relu	0.1	10
100	50	relu	0.1	10
100	10	tanh	0.1	10
100	10	tanh	0.1	50
100	50	relu	0.1	50
100	50	relu	0.01	50
100	100	relu	0.1	50

Table: Best Parameter Settings for **CIRCLES**

MNIST :

Accuracy%	batch_size	activation	learning_rate	layer_width
51.16852385	10	relu	0.1	10
51.16852385	10	relu	0.1	50
95.70177429	50	relu	0.1	50
96.31951919	100	tanh	0.01	50
96.40294291	50	tanh	0.01	50
96.46133186	50	relu	0.1	10
96.54480431	10	relu	0.01	10
96.56152665	100	tanh	0.01	10
96.6032768	10	tanh	0.1	50
96.72844371	10	tanh	0.01	50
96.74506858	100	relu	0.1	50
96.76185358	50	tanh	0.01	10
96.84530514	50	relu	0.01	10
96.85365935	100	relu	0.1	10
96.86193002	10	relu	0.01	50
96.86197179	100	tanh	0.1	50
96.93709003	100	relu	0.01	50
96.95374972	50	tanh	0.1	50
96.96211089	50	relu	0.01	50
96.97885411	100	relu	0.01	10
97.1790975	100	tanh	0.1	10
97.1874517	10	tanh	0.01	10
97.24579192	10	tanh	0.1	10
97.30424352	50	tanh	0.1	10

Table: Average Accuracy for **MNIST**

.

From the table above, we find that there is a parameter setting that can lead to 97.3% accuracy. The best parameter setting is shown below.

Accuracy%	batch_size	activation	learning_rate	layer_width
97.30424352	50	tanh	0.1	10

Table: Best Parameter Setting for **MNIST**

iii. Learning curve:

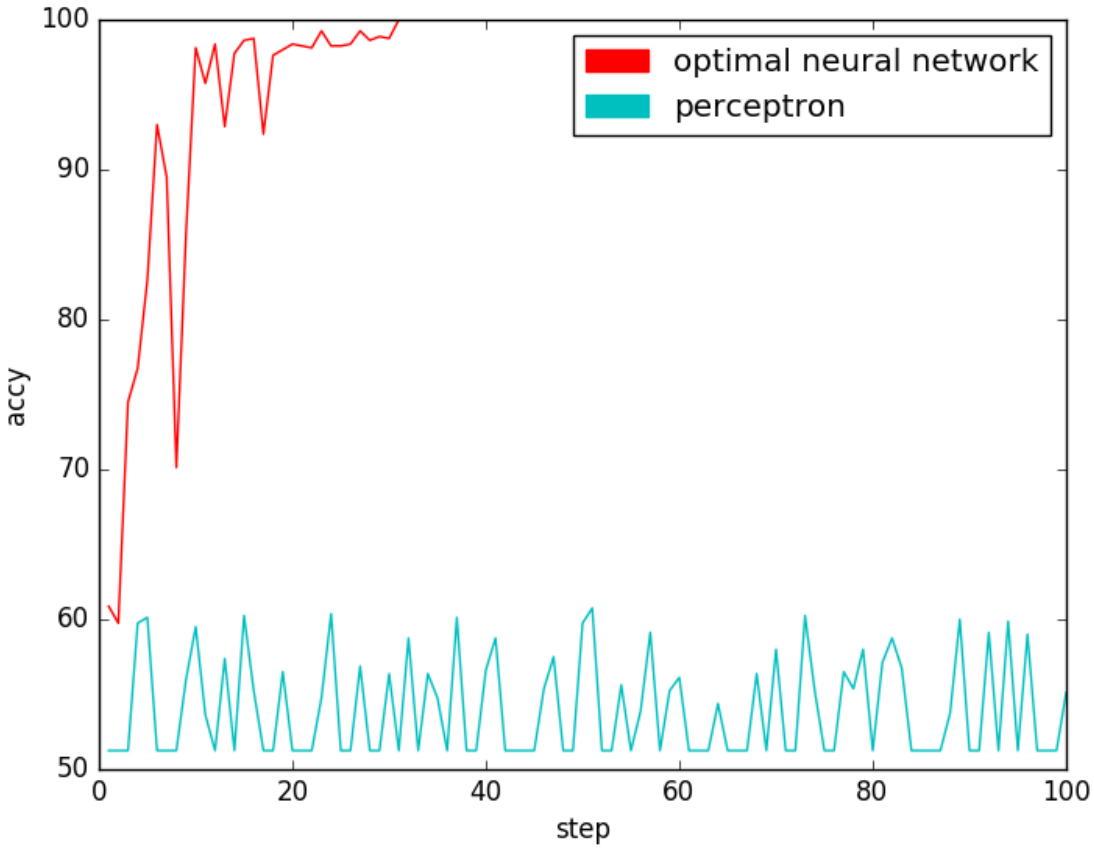

Figure 1: step vs accuracy for circle data

Accuracy of NN on the test data for CIRCLES: 100%
Accuracy of Perceptron on the test data for CIRCLES: 49.7%

For the dataset CIRCLES, NN has much better performance than Perceptron has. Since Perceptron is a linear seperator, it cannot seperate circle in two parts and the accuracy is around 50% which is no better than random guess. For learning curve, NN has much smoother curve than Perceptron, and the curve keeps increasing gradually.

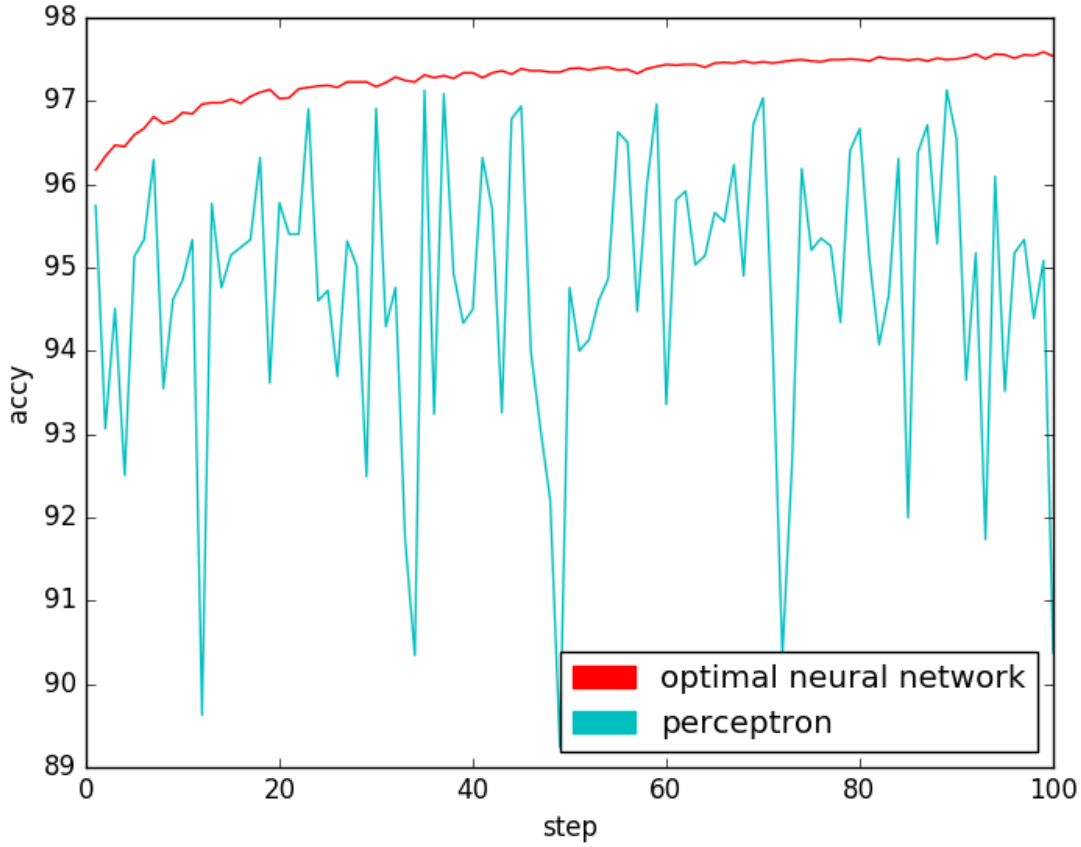

Figure 2: step vs accuracy for circle data

Accuracy of NN on the test data for MNIST: 96.8%
Accuracy of Perceptron on the test data for MNIST: 89.6%

For the dataset MNIST, NN and Perceptron have similar performance. The performance of NN is slightly better than the performance of Perceptron. NN has numerous hidden layers and it could seperate the data more accurately. While, Perceptron is linear seperate that can only seperate the data in two parts with a line. Therefore, it is reasonable that NN has better performance than Perceptron. For the learning curve, NN has much smoother curve than Perceptron has. The learning curve of Perceptron vacilates dramatically.

2. Answer to problem 2

   a. i. For **One-vs-All**: it learnt k classifiers.
      For **All-vs-All**: it learnt k^2 classifiers.
      ii. For **One-vs-All**: it used m samples.
      For **All-vs-All**: it used $\frac{2m}{k}$ samples.
      iii. For **One-vs-All**: it decided the label by the rule of winner takes all.
      For **All-vs-All**: it decided the label by the rule of majority vote.
      iv. For **One-vs-All**: it took $O(km)$ computational complexity.
      For **All-vs-All**: it took $O(k^2 \cdot \frac{2m}{k}) = O(km)$ computational complexity.

   b. I would prefer **One-vs-All**, because it is easier to implement, and has better performance efficiency-wise. **One-vs-All** evaluate k linear classifiers and do Winner Takes All. While, **All-vs-All** evaluate k^2 linear classifiers. It has more expressivity, but less example to learnt from.

   c. Yes, using a KERNEL PERCEPTRON changes the analysis. The computational complexity increases by a factor of m. Therefore, **One-vs-All** is $O(km^2)$, and **All-vs-All** is $O(km^2)$. I would prefer to use **All-vs-All** when using a KERNEL PERCEPTRON, because it can learn from relatively small training set and work in the dual space more effectively.

   d. For **One-vs-All**: the overall training time complexities is $O(kdm^2)$.
   For **All-vs-All**: the overall training time complexities is $O(k^2d \cdot (\frac{2m}{k})^2) = O(dm^2)$.
   In this situation, **All-vs-All** is the most efficient.

   e. For **One-vs-All**: the overall training time complexities is $O(kd^2m)$.
   For **All-vs-All**: the overall training time complexities is $O(k^2d^2 \cdot \frac{2m}{k}) = O(kd^2m)$.
   In this situation, **All-vs-All** and **One-vs-All** have same efficiency.

   f. For **Counting**: overall evaluation time complexities per example is $O(dm^2)$
   For **Knockout**: overall evaluation time complexities per example is $O(dm)$

3. Answer to problem 3

   a. i. $E[A] = 1$
   $E[B] = \frac{1}{P_{boy}} = \frac{1}{0.5} = 2$

   ii. The boy to girl ratio at the end of one generation in town is $1 : 1$.

   b. i.

$$P(A \cap B) = P(A|B) \cdot P(B)$$
$$P(A \cap B) = P(B|A) \cdot P(A)$$
$$P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$$
$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

   ii.

$$\begin{aligned} P(A \cap B \cap C) &= P(A \cap (B \cap C)) && (16)\\ &= P(A|(B \cap C)) \cdot P(B \cap C) && (17)\\ &= P(A|(B \cap C)) \cdot (P(B|C) \cdot P(C)) && (18)\\ &= P(A|B \cap C) \cdot P(B|C) \cdot P(C) && (19) \end{aligned}$$

   c. $E[X] = 1 \cdot P(A) + 0 \cdot (1 - P(A)) = P(A)$

   d. i. No, X is not independent of Y.

$$P(X = 0) = \frac{1}{15} + \frac{1}{10} + \frac{4}{15} + \frac{8}{45} = \frac{11}{18}$$
$$P(Y = 0) = \frac{1}{15} + \frac{1}{15} + \frac{4}{15} + \frac{2}{15} = \frac{8}{15}$$
$$P(X = 0) \cdot P(Y = 0) = \frac{11}{18} \times \frac{8}{15} = \frac{44}{135}$$
$$P(X = 0|Y = 0) = \frac{1}{15} + \frac{4}{15} = \frac{1}{3}$$
$$\Rightarrow P(X = 0|Y = 0) \neq P(X = 0) \cdot P(Y = 0)$$

   ii. Yes, X is conditionally independent of Y given Z.

$$P(X = 0 \cap Y = 0 \cap Z = 0) = \frac{1}{15}$$

$$P(X=0|Z=0)=\frac{1}{15}+\frac{1}{10}=\frac{1}{6}$$
$$P(Y=0|Z=0)=\frac{1}{15}+\frac{1}{15}=\frac{2}{15}$$
$$P(Z=0)=\frac{1}{15}+\frac{1}{10}+\frac{1}{15}+\frac{1}{10}=\frac{1}{3}$$
$$\Rightarrow P(X=0|Z=0)\cdot P(Y=0|Z=0)=\frac{1}{6}\times\frac{2}{15}=\frac{1}{45}$$
$$\Rightarrow P(X=0\cap Y=0\cap Z=0)\cdot P(Z=0)=\frac{1}{15}\times\frac{1}{3}=\frac{1}{45}$$
$$\Rightarrow P(X=0\cap Y=0\cap Z=0)=\frac{P(X=0|Z=0)\cdot P(Y=0|Z=0)}{P(Z=0)}$$

iii.

$$P(X=0|X+Y>0)=\frac{P(X=0\cap X+Y>0)}{P(X+Y>0)}$$
$$P(X=0\cap X+Y>0)=P(X=0\cap Y=1)=\frac{1}{10}+\frac{8}{45}=\frac{5}{18}$$
$$P(X+Y>0)=P(X=1\cap Y=0)+P(X=0\cap Y=1)+P(X=1\cap Y=1)$$
$$P(X+Y>0)=(\frac{1}{15}+\frac{2}{15})\times(\frac{1}{10}+\frac{8}{45})\times(\frac{1}{10}+\frac{4}{45})=\frac{1}{5}+\frac{5}{18}+\frac{17}{90}=\frac{2}{3}$$
$$\Rightarrow P(X=0|X+Y>0)=\frac{P(X=0\cap X+Y>0)}{P(X+Y>0)}=\frac{\frac{5}{18}}{\frac{2}{3}}=\frac{5}{12}$$