

## Problem Set 5

Kaizhao Liang(kl2)

Handed In: April 4, 2017

## 1. Neural Network

a. The input, hidden and output layers are represented by i,j,k respectively.

$$\frac{\partial E_d}{\partial w_{jk}} = \frac{\partial E_d}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial w_{jk}}$$

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

$$\frac{\partial o_k}{\partial net_k} = \begin{cases} 0 & net_k \leq 0 \\ 1 & net_k > 0 \end{cases}$$

$$\frac{\partial net_k}{\partial w_{jk}} = x_j$$

$$w_{jk} \leftarrow w_{jk} + R \delta_k x_j$$

where  $R$  is learning rate,  $\delta_k = -\frac{\partial E_d}{\partial net_k} = \begin{cases} 0 & net_k \leq 0 \\ -(t_k - o_k) & net_k > 0 \end{cases}$

$$\frac{\partial E_d}{\partial w_{ij}} = \frac{\partial E_d}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}}$$

$$= \sum_{k \in \text{downstream}(j)} \frac{\partial E_d}{\partial net_k} \frac{\partial net_k}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}}$$

$$= \sum_{k \in \text{downstream}(j)} -\delta_k \frac{\partial net_k}{\partial net_j} x_i$$

where,  $\delta_k = -\frac{\partial E_d}{\partial net_j}, net_j = \sum w_{ij} x_i$

$$\frac{\partial net_k}{\partial net_j} = \frac{\partial net_k}{\partial o_j} \frac{\partial o_j}{\partial net_j}$$

$$\frac{\partial o_j}{\partial net_j} = \begin{cases} 0 & net_j \leq 0 \\ 1 & net_j > 0 \end{cases}$$

$$\frac{\partial net_k}{\partial o_j} = w_{jk}$$

So,

$$\frac{\partial E_d}{\partial w_{ij}} = \begin{cases} 0 & net_j \leq 0 \\ \sum_{k \in \text{downstream}(j)} -\delta_k w_{jk} x_i & net_j > 0 \end{cases}$$

So,

$$w_{ij} \leftarrow w_{ij} + R\delta_j x_i$$

$$\text{where, } \delta_j = \begin{cases} 0 & \text{net}_j \leq 0 \\ \sum_{k \in \text{downstream}(j)} -\delta_k w_{jk} & \text{net}_j > 0 \end{cases}$$

b. ii. **CIRCLES:**

[*domain, batch\_size, learning\_rate, activation\_function, hidden\_layer\_width*] :  
*accuracy*

[*'circles'*, 10, 0.1, *'relu'*, 10] : 100.0  
 [*'circles'*, 10, 0.1, *'relu'*, 50] : 100.0  
 [*'circles'*, 10, 0.1, *'tanh'*, 10] : 100.0  
 [*'circles'*, 10, 0.1, *'tanh'*, 50] : 100.0  
 [*'circles'*, 10, 0.01, *'relu'*, 10] : 95.375  
 [*'circles'*, 10, 0.01, *'relu'*, 50] : 100.0  
 [*'circles'*, 10, 0.01, *'tanh'*, 10] : 49.125  
 [*'circles'*, 10, 0.01, *'tanh'*, 50] : 51.0  
 [*'circles'*, 50, 0.1, *'relu'*, 10] : 100.0  
 [*'circles'*, 50, 0.1, *'relu'*, 50] : 100.0  
 [*'circles'*, 50, 0.1, *'tanh'*, 10] : 58.375  
 [*'circles'*, 50, 0.1, *'tanh'*, 50] : 54.75  
 [*'circles'*, 50, 0.01, *'relu'*, 10] : 74.875  
 [*'circles'*, 50, 0.01, *'relu'*, 50] : 94.25  
 [*'circles'*, 50, 0.01, *'tanh'*, 10] : 50.5  
 [*'circles'*, 50, 0.01, *'tanh'*, 50] : 49.75  
 [*'circles'*, 100, 0.1, *'relu'*, 10] : 99.75  
 [*'circles'*, 100, 0.1, *'relu'*, 50] : 100.0  
 [*'circles'*, 100, 0.1, *'tanh'*, 10] : 49.5  
 [*'circles'*, 100, 0.1, *'tanh'*, 50] : 51.875  
 [*'circles'*, 100, 0.01, *'relu'*, 10] : 57.875  
 [*'circles'*, 100, 0.01, *'relu'*, 50] : 78.875  
 [*'circles'*, 100, 0.01, *'tanh'*, 10] : 48.75  
 [*'circles'*, 100, 0.01, *'tanh'*, 50] : 48.875

**The best set for circle is:**

[*'circles'*, 10, 0.1, *'relu'*, 10] : 100.0

**MNIST:**

[*'mnist'*, 10, 0.1, *'relu'*, 10] : 95.6266212383  
 [*'mnist'*, 10, 0.1, *'relu'*, 50] : 96.2108866459  
 [*'mnist'*, 10, 0.1, *'tanh'*, 10] : 96.9119717182  
 [*'mnist'*, 10, 0.1, *'tanh'*, 50] : 96.8785966603  
 [*'mnist'*, 10, 0.01, *'relu'*, 10] : 96.5364152932  
 [*'mnist'*, 10, 0.01, *'relu'*, 50] : 96.5447625386  
 [*'mnist'*, 10, 0.01, *'tanh'*, 10] : 96.7700615844  
 [*'mnist'*, 10, 0.01, *'tanh'*, 50] : 96.2108727223  
 [*'mnist'*, 50, 0.1, *'relu'*, 10] : 96.5197138406

```

['mnist', 50, 0.1, 'relu', 50] : 96.4279080647
['mnist', 50, 0.1, 'tanh', 10] : 96.886929982
['mnist', 50, 0.1, 'tanh', 50] : 96.553068013
['mnist', 50, 0.01, 'relu', 10] : 96.7868048081
['mnist', 50, 0.01, 'relu', 50] : 96.7283671284
['mnist', 50, 0.01, 'tanh', 10] : 96.3444147251
['mnist', 50, 0.01, 'tanh', 50] : 96.1775185498
['mnist', 100, 0.1, 'relu', 10] : 96.6365613526
['mnist', 100, 0.1, 'relu', 50] : 96.4612831227
['mnist', 100, 0.1, 'tanh', 10] : 96.7784157917
['mnist', 100, 0.1, 'tanh', 50] : 96.2359005348
['mnist', 100, 0.01, 'relu', 10] : 96.6365613526
['mnist', 100, 0.01, 'relu', 50] : 96.6699642579
['mnist', 100, 0.01, 'tanh', 10] : 96.2108796841
['mnist', 100, 0.01, 'tanh', 50] : 96.1441295682

```

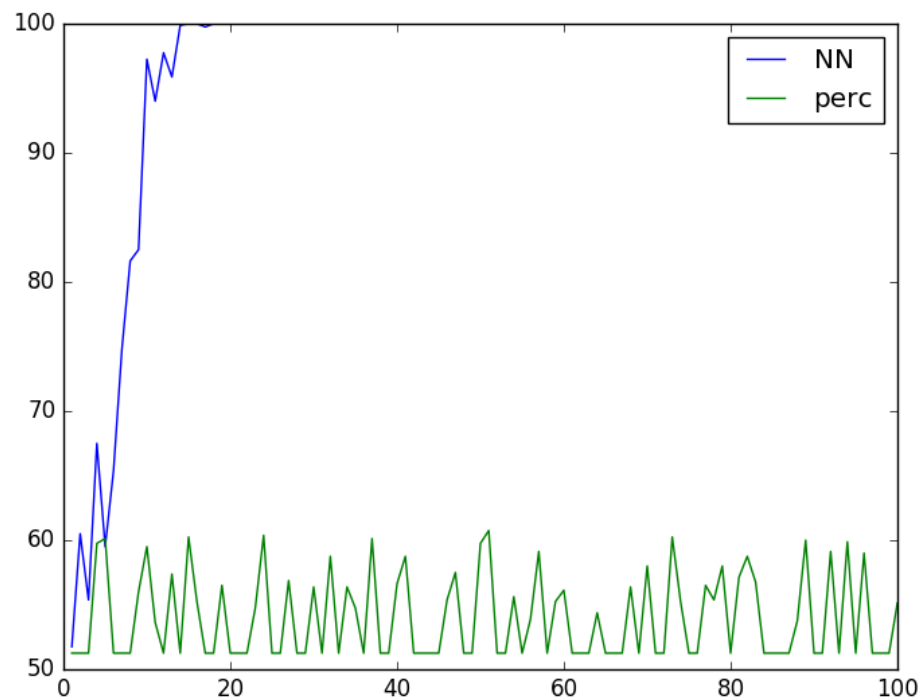
**The best set for mnist:**

```

['mnist', 10, 0.1, 'tanh', 10] : 96.9119717182

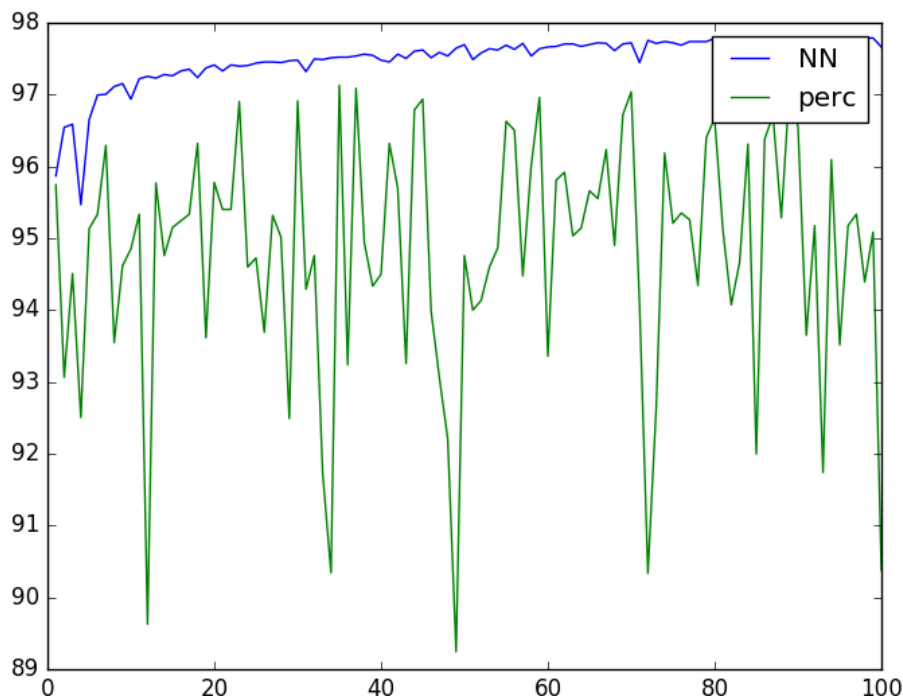
```

### iii. Circles:



test accuracy of NN:100%  
test accuracy of perceptron:49.5%

**Mnist:**



test accuracy of NN:96.673

test accuracy of perceptron:88.86%

Circle:

Perceptron basically fails to learn the Circle, because it's a linear separator and there is no way it could generate a good hypothesis separating the circle data in the primary feature space. Relatively, the Neural network learns perfectly on the circle, since theoretically neural network could learn any non-linear function. This is perfectly revealed in the learning curves. The curve\_NN increases singularly until it reaches the top, while the curve\_perceptron stays low below 50%.

Mnist:

The perceptron and NN's performances are close on this data set, with NN's slightly higher. Perhaps the mnist data is more linearly separable than the Circle. Interestingly, perceptron's learning curve fluctuates dramatically, although it gives a suprisingly decent result in the end, while the learning curve of NN grows almost singularly until it reaches the local maximum.

## 2. Multi-class classification

- a.
  - i. For One vs All,  $k$  separators are learnt.  
For All vs All,  $k^2$  separators are learnt.
  - ii. For One vs All, each of the separators uses  $m$  examples.  
For All vs All, each of the separators uses only  $\frac{2m}{k}$  samples.

- iii. For One vs All, I will decide the final label by the rule of Winner Takes All,  
 $f(x) = \operatorname{argmax}_i(w_i^T x)$   
 For All vs All, I will decide the final label by majority, outputting the label that has been predicted by most number of separators.
- iv. For One vs All, the complexity of training is  $O(km)$   
 For All vs All, the complexity of training is also  $O(km)$
- b. I would prefer One vs All. Although it is less expressive than All vs. All, it has more data to train on per classifier, while All vs. All could have very little number of samples to train on, depending on the size of m and k. Also One vs. all is simpler to implement given that it has less number of separators to learn.
- c. Yes, when using the kernel perceptron, the complexity for One vs. All is  $O(km^2)$  and the complexity for All vs. All is  $O(m^2)$ . Using All vs. All kernel perceptron allows us to work in the dual space more efficiently and learn with relatively small training set. So I would prefer to use All vs. All.
- d. One vs. All:  $O(kdm^2)$   
 All vs. All:  $O(dm^2)$   
 All vs. All is more efficient.
- e. One vs. All:  $O(kd^2m)$   
 All vs. All:  $O(kd^2m)$   
 They are equally efficient.
- f. Counting:  $O(k^2)$   
 KnockOut:  $O(k)$

### 3. Probability Review

- a. i.

$$E(A) = 1$$

$$E(B) = \lim_{n \rightarrow \infty} \sum_{i=1}^n i \cdot \left(\frac{1}{2}\right)^i = 2$$

- ii. Both of the towns boy-girl ratios maintain 1:1 at the end of a generation.
- b. i.

$$P(A|B) \cdot P(B) = P(A, B)$$

$$P(B|A) \cdot P(A) = P(A, B)$$

$$P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$$

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

- ii.

$$P(A, B, C) = P(A|B, C) \cdot P(B|C) \cdot P(C)$$

c.

$$E(X) = 1 \cdot P(A) + 0 \cdot (1 - P(A)) = P(A)$$

- d. i. No, because  $P(X = 0) = \frac{55}{90}$ ,  $P(Y = 0) = \frac{48}{90}$ ,  $P(X = 0) \cdot P(Y = 0) = \frac{88}{270}$ , but according to the table  $P(X = 0, Y = 0) = \frac{30}{90}$ , which means that  $P(X) \neq P(X|Y)$ , and the knowledge of Y will influence the probability of the X. By definition, they are not independent.
- ii. Yes, since in the table,  $P(X, Y|Z) = P(X|Z) \cdot P(Y|Z)$ , i.e.  $\frac{P(X, Y, Z)}{P(Z)} = \frac{P(X, Z)}{P(Z)} \cdot \frac{P(Y, Z)}{P(Z)} \Rightarrow P(X, Y, Z) = \frac{P(X, Z) \cdot P(Y, Z)}{P(Z)}$ . For example,  $P(X = 0, Y = 0, Z = 0) = \frac{1}{15}$ ,  $P(X = 0, Z = 0) \cdot P(Y = 0, Z = 0) = \frac{1}{6} \cdot \frac{2}{15} = \frac{1}{45}$ ,  $P(Z = 0) = \frac{1}{3}$  So  $P(X = 0, Y = 0, Z = 0) = \frac{P(X=0, Z=0) \cdot P(Y=0, Z=0)}{P(Z=0)}$ .
- iii.

$$P(X = 0|X + Y > 0) = \frac{P(X = 0, X + Y > 0)}{P(X + Y > 0)}$$

$$\begin{aligned} P(X = 0, X + Y > 0) &= P(X = 0, Y = 1) \\ &= P(X = 0, Y = 1, Z = 0) + P(X = 0, Y = 1, Z = 1) \\ &= \frac{1}{10} + \frac{8}{45} = \frac{5}{18} \end{aligned}$$

$$\begin{aligned} P(X + Y > 0) &= P(X = 0, Y = 1) + P(X = 1, Y = 0) + P(X = 1, Y = 1) \\ &= \frac{25}{90} + \frac{3}{15} + \frac{17}{90} = \frac{2}{3} \end{aligned}$$

So,

$$P(X = 0|X + Y > 0) = \frac{\frac{5}{18}}{\frac{2}{3}} = \frac{5}{12}$$