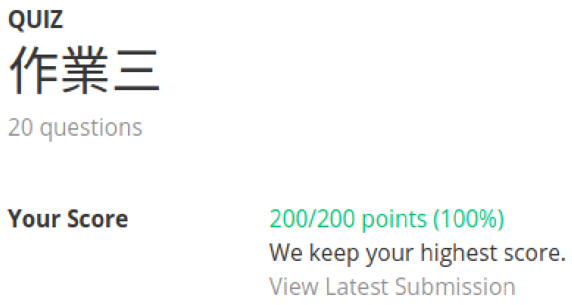
Machine Learning Foundation HW3

b04902060

December 2017

1



QUIZ

作業三

20 questions

Your Score 200/200 points (100%)

We keep your highest score.

View Latest Submission

2

On the left hand side, $(I-H)^2 = I^2 - IH - HI + H^2$.
Since I is an identity matrix, we can know that $I^2 = I$, $IH = H$ and $HI = H$.
And, $H^2 = (X(X^TX)^{-1}X^T) * (X(X^TX)^{-1}X^T) = X(X^TX)^{-1}X^T = H$.
Therefore, $(I-H)^2 = I^2 - IH - HI + H^2 = I - H - H + H = I - H$.

3

Using SGD to find minimum, we update w by $\mathbf{w}_{t+1} = \mathbf{w}_t - err'(\mathbf{w})$, where $err'(\mathbf{w})$ is the gradient. When $err(\mathbf{w}) = max(0, -y\mathbf{w}^T\mathbf{x})$, we can discuss it in 2 case:
1. $y\mathbf{w}^T\mathbf{x} > 0$. In this case, $err'(\mathbf{w}) = 0$, which means that $\mathbf{w}_{t+1} = \mathbf{w}_t$.
2. $y\mathbf{w}^T\mathbf{x} < 0$. In this case, $err'(\mathbf{w}) = -y\mathbf{x}$, which means that $\mathbf{w}_{t+1} = \mathbf{w}_t + y\mathbf{x}$.
This is exactly PLA. PLA updates $\mathbf{w}$ by $\mathbf{w}_{t+1} = \mathbf{w}_t + y\mathbf{x}$ only when the prediction is wrong, which means that $y\mathbf{w}^T\mathbf{x} < 0$. Therefore, when $err(\mathbf{w}) = max(0, -y\mathbf{w}^T\mathbf{x})$, SGD is PLA.

4

We know that second-order Taylor's expansion can be written as

$$E_2(\Delta u, \Delta v) = E(u,v) + \Delta u \frac{\partial E(u,v)}{\partial u} + \Delta v \frac{\partial E(u,v)}{\partial v} +$$

$$\frac{1}{2!}\left[(\Delta u)^2 \frac{\partial^2 E(u,v)}{\partial u^2} + 2\Delta u \Delta v \frac{\partial^2 E(u,v)}{\partial u \partial v}(\Delta v)^2 \frac{\partial^2 E(u,v)}{\partial v^2}\right]$$

To find the optimal $(\Delta u, \Delta v)$ that minimize $E_2(\Delta u, \Delta v)$, we have to compute the gradient:

$$\frac{\partial E_2(\Delta u, \Delta v)}{\partial \Delta u} = \frac{\partial E(u,v)}{\partial u} + \Delta u \frac{\partial^2 E(u,v)}{\partial u^2} + \Delta v \frac{\partial^2 E(u,v)}{\partial u \partial v} = 0$$

$$\frac{\partial E_2(\Delta u, \Delta v)}{\partial \Delta v} = \frac{\partial E(u,v)}{\partial v} + \Delta v \frac{\partial^2 E(u,v)}{\partial v^2} + \Delta u \frac{\partial^2 E(u,v)}{\partial u \partial v} = 0$$

If we use matrix to represent it, it becomes:

$$\begin{bmatrix} \frac{\partial^2 E(u,v)}{\partial u^2} & \frac{\partial^2 E(u,v)}{\partial u \partial v} \\ \frac{\partial^2 E(u,v)}{\partial u \partial v} & \frac{\partial^2 E(u,v)}{\partial v^2} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} \frac{\partial E(u,v)}{\partial u} \\ \frac{\partial E(u,v)}{\partial v} \end{bmatrix}$$

The matrix above is Hessian matrix. We can multiple the inverse of the Hessian matrix and $\nabla E(u,v)$ to get the direction. And since we're computing the minimum, the direction should be negative. Therefore, the answer is $-(\nabla^2 E(u,v))^{-1} \nabla E(u,v)$.

5

The negative log likelihood is

$$\frac{-1}{N} \sum_{n=1}^{N} \ln\left(\frac{exp(\mathbf{w}_{y_n}^T \mathbf{x}_n)}{\sum_{i=1}^{k} exp(\mathbf{w}_i^T \mathbf{x}_n)}\right) = \frac{-1}{N} \sum_{i=1}^{N} \left(\mathbf{w}_{y_n}^T \mathbf{x}_n - \ln\left(\sum_{i=1}^{k} exp(\mathbf{w}_i^T \mathbf{x}_n)\right)\right)$$

Therefore, it quite obvious that E_{in} is

$$\frac{1}{N} \sum_{i=1}^{N} \left(\ln\left(\sum_{i=1}^{k} exp(\mathbf{w}_i^T \mathbf{x}_n)\right) - \mathbf{w}_{y_n}^T \mathbf{x}_n\right)$$

6

To calculate $\frac{\partial E_{in}}{\partial \mathbf{w}_j}$, we deal with two parts in the summation.
First, we can compute the log part like below.

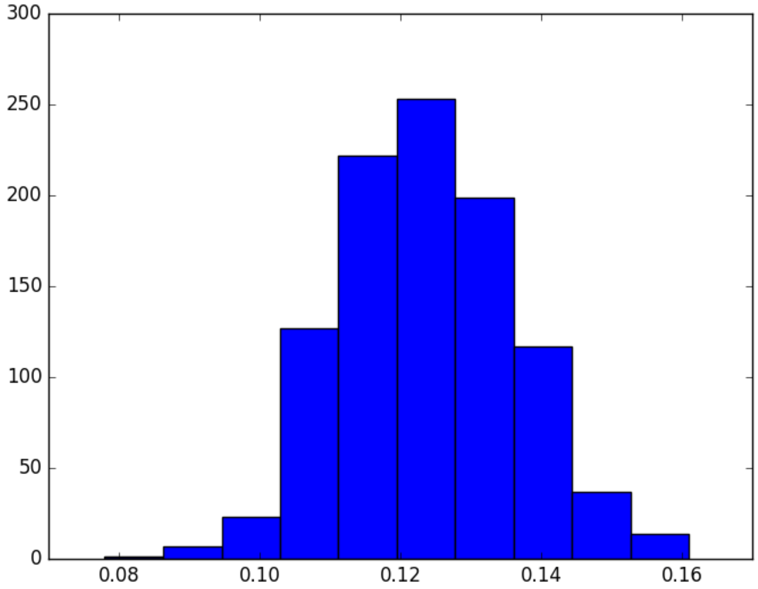
$$\frac{\partial \ln\left(\sum_{i=1}^{k} exp(\mathbf{w}_i^T \mathbf{x})\right)}{\partial \mathbf{w}_j} = \frac{1}{\sum_{i=1}^{k} exp(\mathbf{w}_i^T \mathbf{x})} * exp(\mathbf{w}_j^T \mathbf{x}) * \mathbf{x} = h_j(\mathbf{x})\mathbf{x}$$

The other one is $\mathbf{w}_{y_n}^T \mathbf{x}_n$. It will only matters when j is the correct answer. If it is correct, then $\frac{\partial \mathbf{w}_{y_n}^T \mathbf{x}_n}{\partial \mathbf{w}_j}$ is $\mathbf{x}_n$.
Add two parts together and we get the answer.

$$\frac{\partial E_{in}}{\partial \mathbf{w}_j} = \frac{1}{N} \sum_{n=1}^{N} \Bigg((h_j(\mathbf{x}) + [\![y_n = j]\!])\, \mathbf{x}_n \Bigg)$$

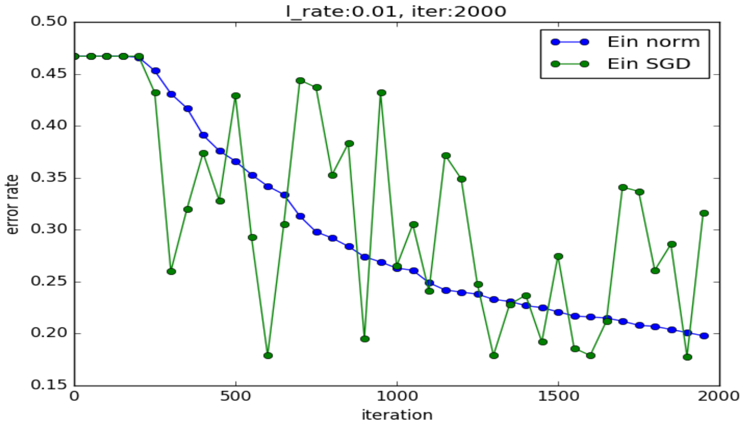
7

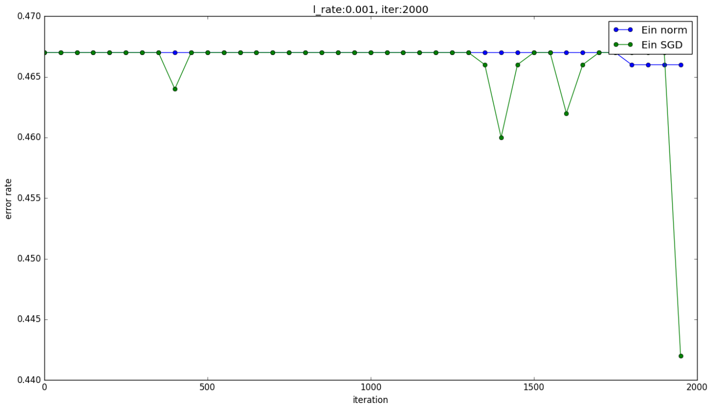
The number of different Eout in each round.



8

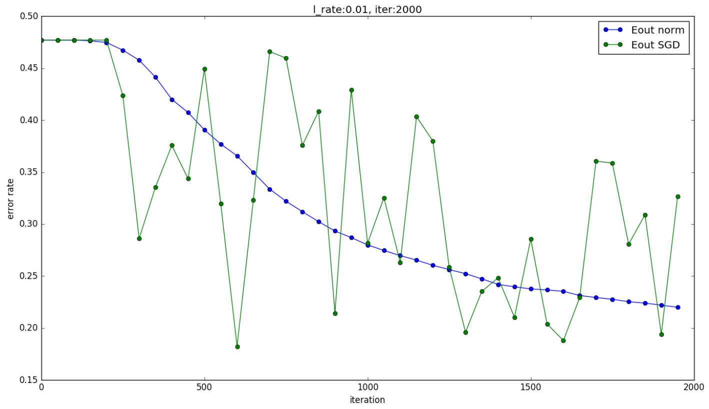
When learning rate is 0.001, error rates is almost unchanged. I think it is probably due to the computing issue in implementation. The scale is too small that computer might ignore or miss it. As to gradient descent and stochastic gradient descent, we can see big differences between them. In normal case, error rate decreases smoothly. In stochastic case, error rate changes dramatically, but the trend is also decreasing.

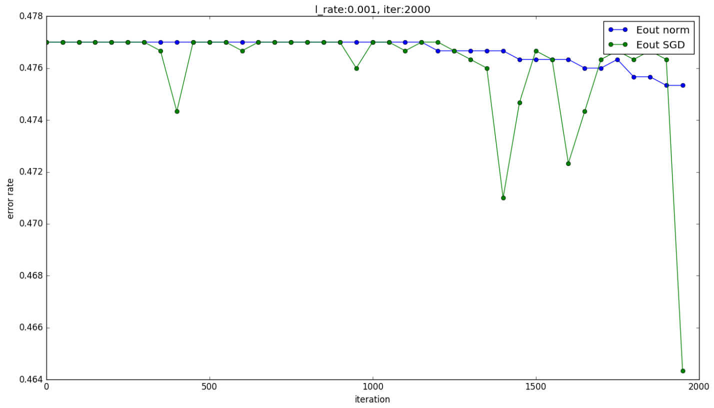




9

The phenomena of leaning rate and gradient descent are very similar to Question 8. The difference is that the average Eout is higher than Ein. It quite makes sense, since the model is trained and tested by different sets of data.


l_rate:0.01, iter:2000
Eout norm
Eout SGD
error rate
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0
500
1000
1500
2000
iteration


l_rate:0.001, iter:2000
Eout norm
Eout SGD
error rate
0.478
0.476
0.474
0.472
0.470
0.468
0.466
0.464
0
500
1000
1500
2000
iteration