hw4

1.

$$\begin{tikzcd} x \arrow[r, "V"] & z_1 \arrow[r, "\overset{\text{(activate).}}{\text{ReLU}}"] & a_1 \arrow[r, "W"] & z_2 \arrow[r, "\overset{\text{(activate)}}{\text{SoftMax}}"] & a_2 \\ \text{input} & \text{hidden layer} & & & \\ (6000\times 784)+1 & \text{~~785×200~~} & +\text{bias} & & \\ \text{Stochastic: } 785\times 1 & 200\times 1 & 201\times 1 & 10\times 1 & 10\times 1 \end{tikzcd}$$

Loss : $J = -\sum_{k=1}^{10} y_k \ln z_k(x)$.

Softmax : $a_{2j} = g(z_{2j}) = \dfrac{e^{z_{2j}}}{\sum_{k=1}^{10} e^{z_{2k}}}$ $\qquad \dfrac{\partial a_{2j}}{\partial z_{2j}} = \begin{cases} a_{2j}(1-a_{2j}), & i=j \\ -a_{2j}a_{2i}, & i\neq j \end{cases}$

use this result :

$$\Rightarrow \frac{\partial L}{\partial z_{2i}} = -\Big(y_i(1-a_{2i}) + \sum_{k\neq i} -a_{2k}y_k\Big).$$

$$= a_{2i}y_i - y_i + \sum_{k\neq i} a_{2k}y_k$$

$$= a_{2i}\Big(\sum_i y_i\Big) - y_i = a_{2i} - y_i$$

$$\Rightarrow \delta_i^{L} = \sum_i a_{2i} - y_i \quad \Rightarrow \underline{\delta^L = a_2 - y} \quad (10 \text{ by } 1).$$

$$\delta_i^{L-1} = \delta_i^{L}\cdot\frac{\partial z_{2i}}{\partial a_{1j}}\cdot\frac{\partial a_{1j}}{\partial z_{1k}} = (a_{2i}-y_i)\cdot W_{ij}\cdot\{0,1\} \searrow \begin{cases} \text{else} \rightarrow 1 \\ \text{if } a_{1j}\leq 0 \rightarrow 0 \end{cases}$$

if ~~$(a_{2i}-y_i)W_{ij}\leq 0 \rightarrow 0$~~

$$\Rightarrow \underset{201\times 10}{\delta^{L-1}} = \underset{201\times 10}{W^T}\cdot\underset{10\times 1}{\delta^L}\cdot\underset{201\times 200}{\big(\text{ReLU}'(a_1)\big)^T} \Rightarrow \underset{1\times 10}{(\delta^L)^T}\cdot\underset{10\times 201}{W}\cdot\underset{201\times 200.}{\text{ReLU}'(a_1)}$$

$$\Rightarrow 1\times 200$$

$$\Rightarrow \frac{\partial L}{\partial W} = \underset{10\times 1}{\delta^L}\cdot\underset{1\times 201}{a_1^T}$$

$$\Rightarrow \frac{\partial L}{\partial V} = \underset{200\times 1}{(\delta^{L-1})^T}\cdot\underset{1\times 785}{X_i^T} \quad \Rightarrow 200\times 785$$

3. ① learning rate → 0.0001 minibatch size = 50 at beginning.

decay rate → 0.95 every 1000 iteration.

initialize W, V with. multivariate_normal. (zero mean. $\sigma = 0.05$).

use Nesterov accelerated gradient to increase speed of convergence.
(apply the momentum before calculating gradient)

②. final : 99.7% training accuracy. 99.6% validation accuracy.

③ Runing time : first, I use batch=50, iter=10,000 → 20 mins.

Then increase batch size. 50 → 100 → 200 → 500 → 1000.
→ 2000 → 5000 → 10000 → 20000.

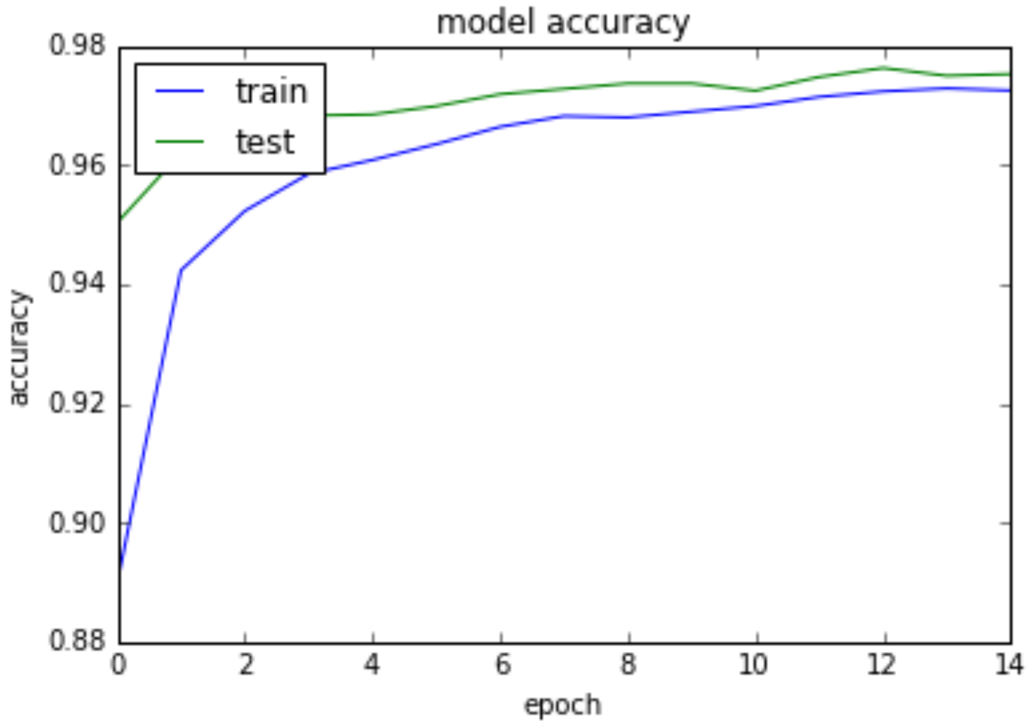
for each batch size. I iterate 5 epoches.

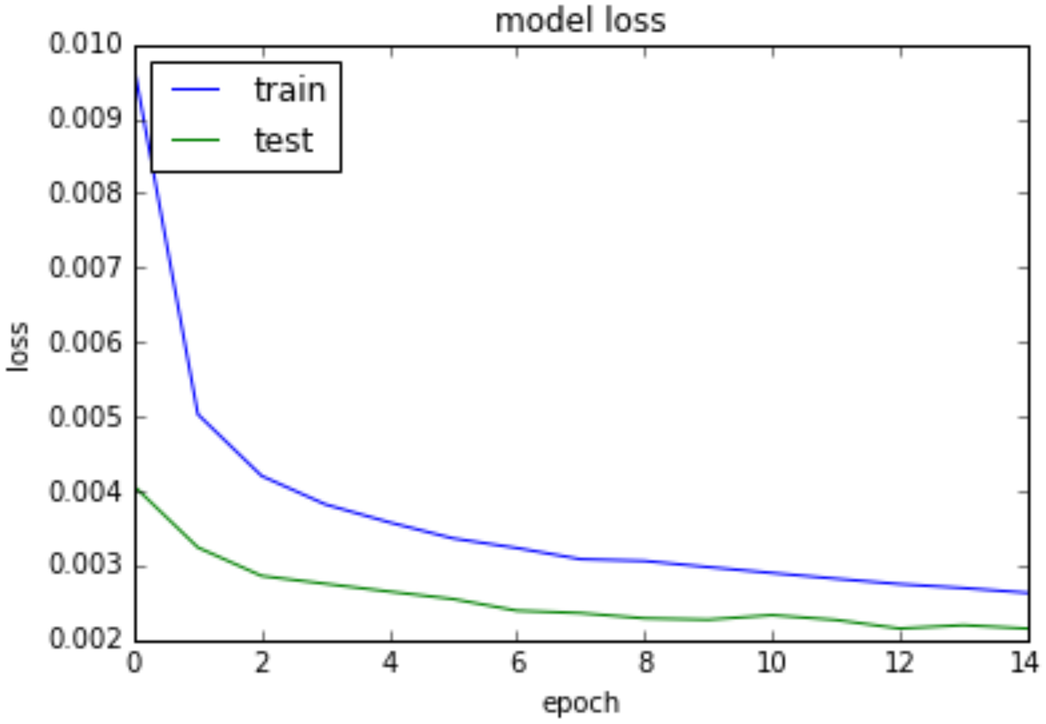
Although, I don't know why, the accuracy ~~was~~ increased after I added batch size. (for batch =50, 97% is the highest, can't be improved any more).

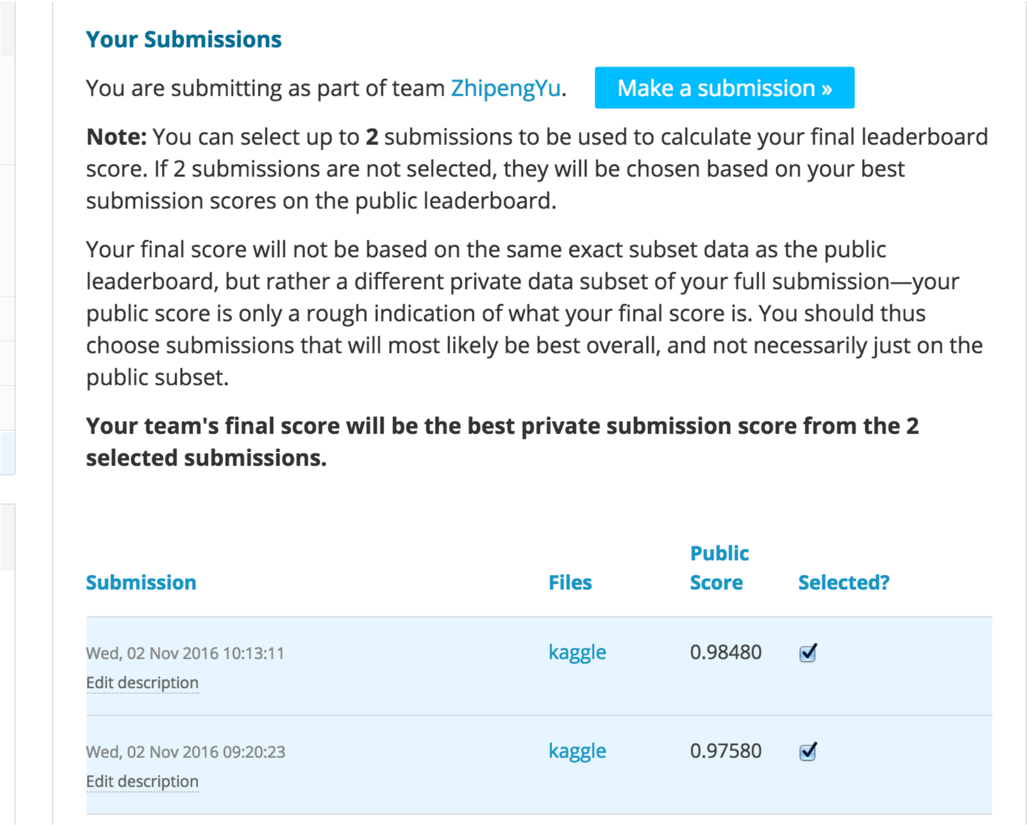
④ plot and save.

4. I used (1) L2 regularization. (2) use log (x+0.1) transformation to preprocess. (3) Nesterov momentum. (4) to keep numerical stability. I used trick mentioned. in .pdf.

$$\frac{e^{z_i} - \max\{e^{z}\}}{\sum_{k}^{n} e^{z_k} - \max\{e^{z}\}}.$$


model accuracy
accuracy
epoch
0.98
0.96
0.94
0.92
0.90
0.88
0
2
4
6
8
10
12
14
train
test


model loss
loss
epoch
0.010
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0
2
4
6
8
10
12
14
train
test



Your Submissions

You are submitting as part of team ZhipengYu.

Note: You can select up to **2** submissions to be used to calculate your final leaderboard score. If 2 submissions are not selected, they will be chosen based on your best submission scores on the public leaderboard.

Your final score will not be based on the same exact subset data as the public leaderboard, but rather a different private data subset of your full submission—your public score is only a rough indication of what your final score is. You should thus choose submissions that will most likely be best overall, and not necessarily just on the public subset.

Your team's final score will be the best private submission score from the 2 selected submissions.

Submission	Files	Public Score	Selected?
Wed, 02 Nov 2016 10:13:11 Edit description	kaggle	0.98480	☑
Wed, 02 Nov 2016 09:20:23 Edit description	kaggle	0.97580	☑