DD2424 - Assignment 1
Bonus Point

Thi Thuy Nga Nguyen

Exercise 2

1 Optimize the performance of the network

I tried some suggestions to improve the training result:

(1) Train for a longer time and use your validation set to avoid overfiting. In particular, after every fifth epoch the accuracy on validation set is recorded and the training is stopped if the accuracy decreases.

(2) Decaying the learning rate by a factor 0.9 after each epoch.

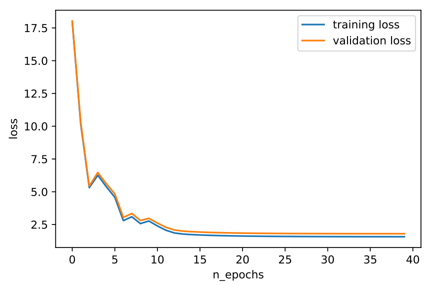
(3) Using Xavier initialization where the the weight W and threshold b are taken Gaussian random values with zero mean and standard deviation $\sigma = \frac{1}{\sqrt{d}} = \frac{1}{\sqrt{3072}}$.

(4) Shuffle the order of the training set before generating the batches at the beginning of every epoch.

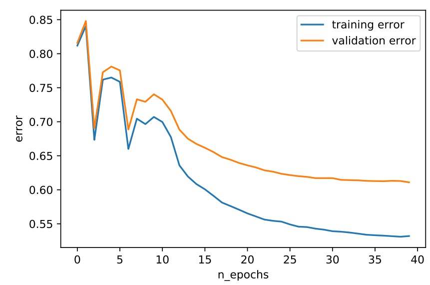
The best test accuracy obtained by using (2) is **38.69%** for the parameter ($\lambda = 0, n\_epochs = 40, n\_batch = 100, \eta = 0.1$). Some combinations gave the similar accuracy with the same parameter setting ($\lambda = 0, n\_epochs = 40, n\_batch = 100, \eta = 0.1$) for all cases:

- 38.5% in using (2) and (3);
- 38.58% in using (1), (2) and (3);
- 38.53% in using (2) and (4);
- 38.43% in using (2), (3) and (4)

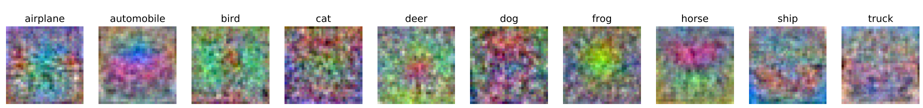
From the results, we can see that (2) brought the largest gain.



(a)



(b)



(c)

Figure 1: The loss (a), error (b) after each epoch and the image of the learnt weight (c) after completing training using improvement (3) with model parameters $\lambda = 0, n\_epochs = 40, n\_batch = 100, \eta = 0.1$.

2 Train network by minimizing the SVM multi-class loss

By using Hinge multi-class loss (SVM) we get the accuracy for 4 parameters settings:

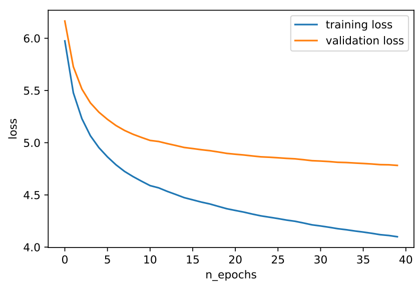
Mini-batch GD parameters	SoftMax + Cross-entropy	Hinge loss (SVM)
$(0, 40, 100, 0.1)$	18.99%	27.52%
$(0, 40, 100, 0.01)$	36.89%	28.24%
$(0.1, 40, 100, 0.01)$	33.38%	29.92%
$(1, 40, 100, 0.01)$	21.9%	21.31%

Table 1: A comparison of the accuracy between SoftMax+Cross-entropy loss and Hinge loss (SVM) in mini-batch gradient descent for multi-class training with some parameter settings $(\lambda, n\_epochs, n\_batch, \eta)$.

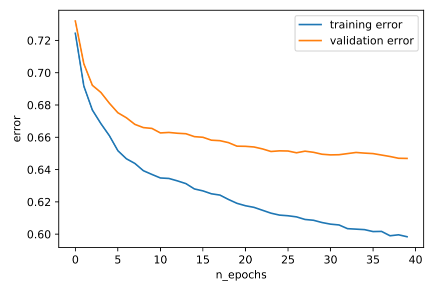
We can see that SVM shows poor results comparing to SoftMax+Cross-entropy (SMCE) loss. However, it gives better accuracy (similar SMCE) in the case of smaller learning rate. In the last case, when $\eta = 0.0001$, mini-batch GD using SMCE loss is underfitting. So SVM is better.

Mini-batch GD parameters	SoftMax + Cross-entropy	Hinge loss (SVM)
$(0, 40, 100, 0.001)$	35.88%	35.83%
$(0.01, 40, 100, 0.001)$	35.83%	35.71%
$(0.01, 40, 100, 0.0001)$	27.83%	34.81%

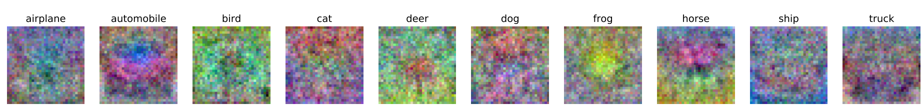
Table 2: A comparison of the accuracy between SoftMax+Cross-entropy loss and Hinge loss (SVM) in mini-batch gradient descent for multi-class training with some parameter settings $(\lambda, n\_epochs, n\_batch, \eta)$.



(a)



(b)



(c)

Figure 2: The loss (a), error (b) after each epoch and the image of the learnt weight (c) after completing training using Hinge loss (SVM) with model parameters $\lambda = 0, n\_epochs = 40, n\_batch = 100, \eta = 0.001$.