DD2424 - Assignment 1

Thi Thuy Nga Nguyen

1 Exercise 1

1.1 Gradient Computation

The numerically computed gradients, g_n, and the analytically computed gradients, g_a, were compared by using relative error

$$error = \frac{|g_a - g_n|}{max(\epsilon, |g_a| + |g_n|)}.$$

With setting $\epsilon = 10^{-15}$ and initializing the weights W and the threshold b to have Gaussian random values with zero mean and standard deviation 0.01, we get the result for first 100 images in data_batch_1:

```
The number of errors (relative error > 1e-6) of W: 83
The maximum of relative error of W: 2.3180494388577465e-05
The number of errors (relative error > 1e-6) of b: 0
The maximum of relative error of b: 9.296465219806587e-09
```

The number of errors (the relative error is larger than 10^{-6}) for the weights W of size 10×3072 is small and the maximum relative error ($\approx 2.10^{-5}$) is not to large.

1.2 Learning Results

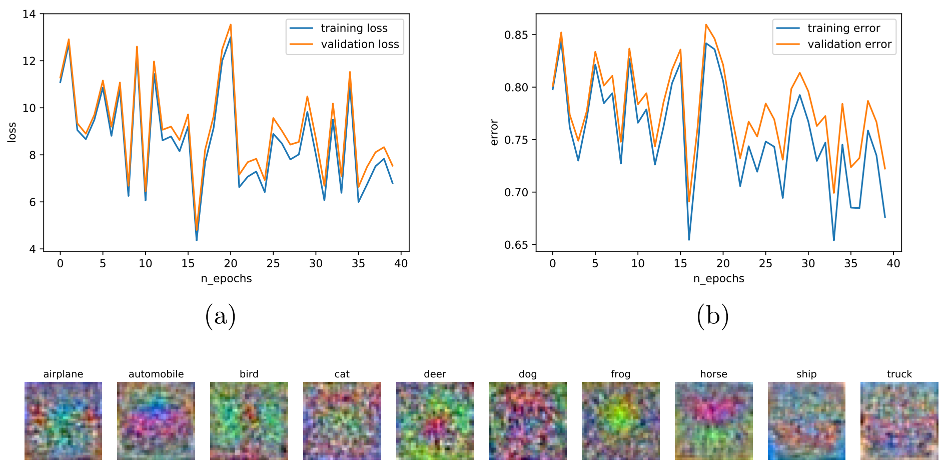
Consider 4 parameters settings for mini-batch gradient descent we get the results of the loss/cost and the error after each epoch and the image representing learnt weight matrix W after completing of training:

- Figure 1 shows results for $\lambda = 0, n\_epochs = 40, n\_batch = 100, \eta = 0.1$. The final test accuracy is 28.28%.
- Figure 2 shows results for $\lambda = 0, n\_epochs = 40, n\_batch = 100, \eta = 0.01$. The final test accuracy is 36.89%.
- Figure 3 shows results for $\lambda = 0.1, n\_epochs = 40, n\_batch = 100, \eta = 0.01$. The final test accuracy is 33.37%.

- Figure 4 shows results for $\lambda = 1, n\_epochs = 40, n\_batch = 100, \eta = 0.01$. The final test accuracy is 21.9%.

From the result we can see that:

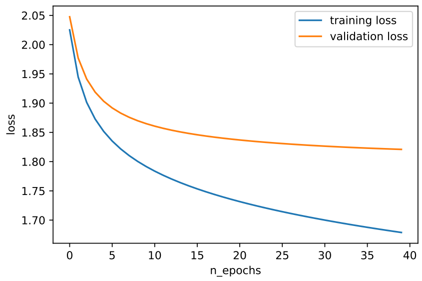
- Increasing the regularization decreases the difference between training and validation data in both cost and error. So it decreases overfitting of the model.
- A large correct learning rate makes the cost and error curve with many oscillations and they are not really convergent. On the other hand, with a small learning rate, the convergence is slow.

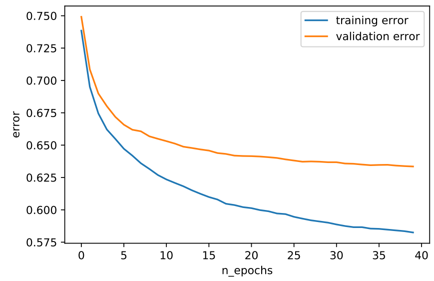


(a) (b)

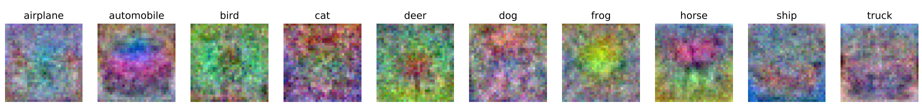
(c)

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



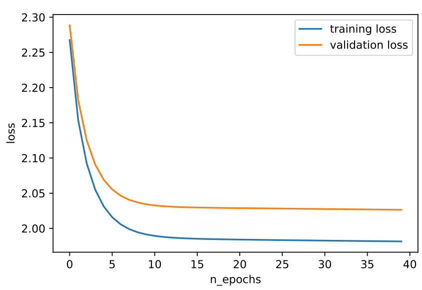


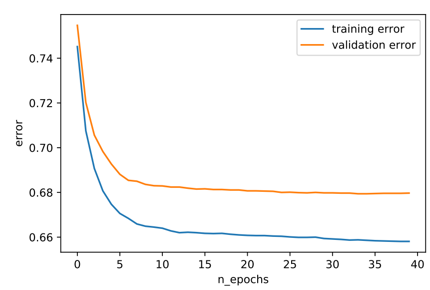
(a) (b)



(c)

Figure 2: The loss (a), error (b) after each epoch and the image of the learnt weight (c) after completing training with model parameters $\lambda = 0, n\_epochs = 40, n\_batch = 100, \eta = 0.01$.

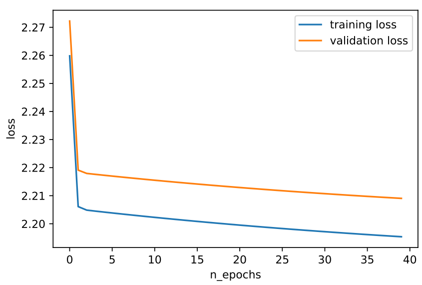




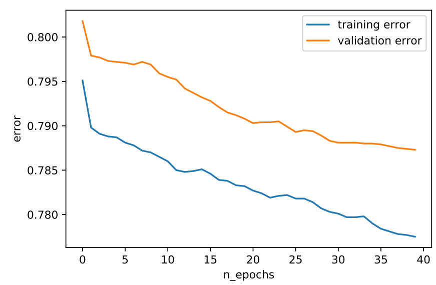
(a) (b)

(c)

Figure 3: The loss (a), error (b) after each epoch and the image of the learnt weight (c) after completing training with model parameters $\lambda = 0.1, n\_epochs = 40, n\_batch = 100, \eta = 0.01$.



(a)



(b)

(c)

Figure 4: The loss (a), error (b) after each epoch and the image of the learnt weight (c) after completing training with model parameters $\lambda = 1, n\_epochs = 40, n\_batch = 100, \eta = 0.01$.