# DD 2424 - Assignment 2

Jia Fu (jiafu@kth.se)

April 23, 2021

## Gradient Check

Here I computed the relative errors between numerically computed gradients $g_n$ and analytically computed gradients $g_a$:

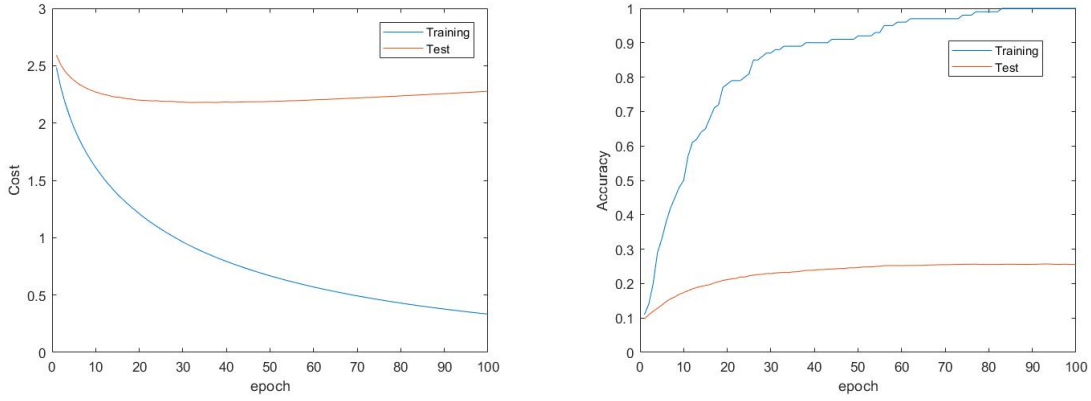
$$\frac{|g_a - g_n|}{\max(\text{eps}, |g_a| + |g_n|)}$$

I set `eps = 1e-5`. The numerically computed gradients were obtained by the given MATLAB function `ComputeGradsNumSlow` which is more accurate than `ComputeGradsNum`. The comparison was based on the max and mean values in the relative error matrices of `W1`, `W2`, `b1`, and `b2`. I used two training samples and 20 dimensions for speed and numerical precision issues, $\lambda = 0$ was applied.

| | `W1` | `W2` | `b1` | `b2` |
|---|---|---|---|---|
| Max | 1.3285e-07 | 2.6845e-07 | 1.7232e-09 | 1.0283e-10 |
| Mean | 6.8163e-10 | 5.1486e-09 | 1.3700e-10 | 4.3088e-11 |

Table 1: Relative Errors of `W1`, `W2`, `b1`, `b2`

Here it can be seen that the max relative errors of `W1`, `W2`, `b1`, and `b2` are all below 1e-6, indicating that the analytically computed gradients are fine for the following sanity check. Then I checked whether the network could overfit 100 training samples and get a very low loss after training for 100 epochs. A good example of this can be seen in figure 1, which indicates that my gradient computations and mini-batch gradient descent algorithm are implemented correctly.



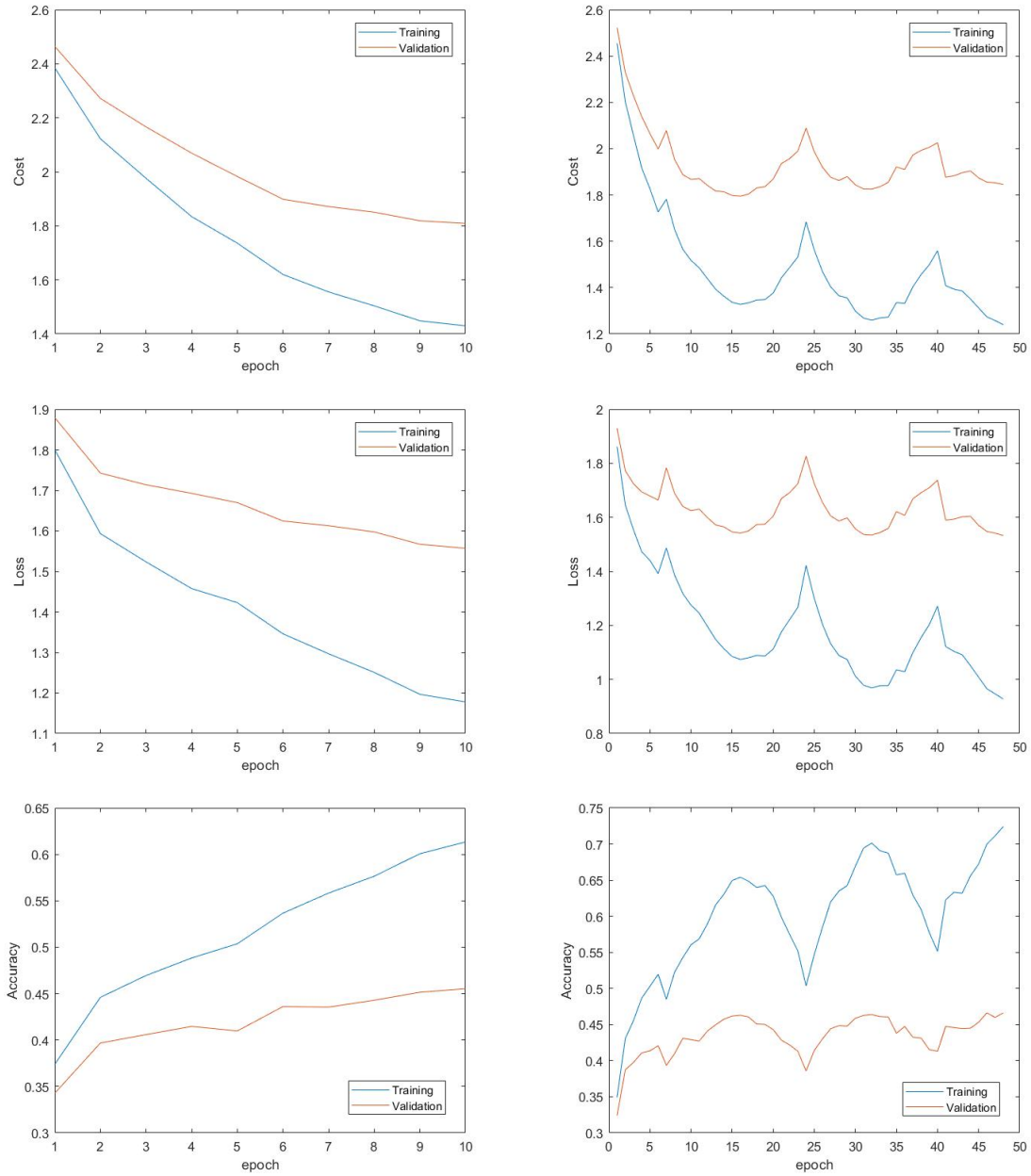

(a) Training and Test Cost

(b) Training and Test Accuracy

Figure 1: `eta = 0.001`, `lambda = 0`, `n_batch = 10`

## Cyclical Learning Rates

In this part, only one batch of the training data was used. The results of one and three cycles' training are shown in figure 2 (a) and (b) respectively. Except for `n_s`, all other parameters were set the same: `eta_min = 1e-5`, `eta_max = 1e-1`, `lambda = 0.01`, and `n_batch = 100`.




(a) **One cycle's training** (`n_s = 500`) (b) **Three cycles' training** (`n_s = 800`)

Figure 2: The impact of cyclic learning rates

The test accuracy 46.43% is achieved in one cycle's training, and 47.29% for three cycles. It's worth noting that when we do three cycle's training, accuracy curves of training and validation data will decrease with the increase of $\eta$ and vice versa in the second and third cycles in general, these two curves will keep growing when we do one cycle's training. Thu, we need to try more (at least not one) cycles to find the best results.

## Lambda Search

In this section, 45000 data points were used for training and 5000 data points for validation. `n_s = 2 * floor(n / n_batch)` and 2 cycles' training are implemented. Other hyper-parameters settings were: `n_batch = 100`, `eta_min = 1e-5`, `eta_max = 1e-1`. Coarse search is done on log scale, let $l = log_{10}\lambda$, then

$$l = l_{min} + (l_{max} - l_{min}) * random(0, 1)$$

In my experiments for the coarse search I set $l_{min} = -5$ and $l_{max} = -1$, a uniform grid with 20 different values were tested. Three best results are:

| | $\lambda$ | Validation Accuracy |
|---|---|---|
| 1 | 0.001390 | 52.46% |
| 2 | 0.000049 | 52.36% |
| 3 | 0.000962 | 52.34% |

Table 2: $\lambda$ settings for the 3 best performing networks from the coarse search

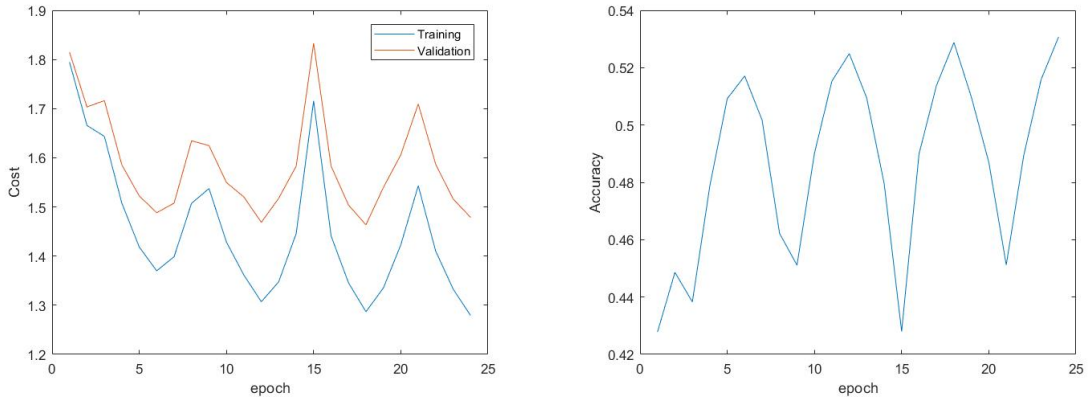
Based on the best $\lambda$ obtained in the coarse search, the range of $\lambda$ was set as 0.001 – 0.005 in the fine search. Other parameters were identical to the coarse search, a uniform grid with 10 different values were tested. Three best results are:

| | $\lambda$ | Validation Accuracy |
|---|---|---|
| 1 | 0.003143 | 53.08% |
| 2 | 0.001689 | 52.28% |
| 3 | 0.001619 | 52.26% |

Table 3: $\lambda$ settings for the 3 best performing networks from the fine search

## Final Results

Finally, the network was trained on all the training data, except for 1000 examples in a validation set, for 4 cycles. `lambda = 0.003143` obtained in the fine search and larger `n_s = 1470` were employed. Other hyper-parameters settings: `n_batch = 100`, `eta_min = 1e-5`, `eta_max = 1e-1`. Figure 3 shows the cost curves of training and validation data. The test accuracy can reach **53.07%** after training.




(a) Training and Validation Cost

(b) Test Accuracy

Figure 3: Network's performance based on the optimal parameters settings