Deep Learning in Data Science (DD2424)
Report to Assignment 2

Cong Gao

April 5, 2020

1 Introduction

In this assignment, I mainly trained and tested a two layer network with multiple outputs to classify images from the CIFAR-10 dataset. I trained the network using mini-batch gradient descent applied to a cost function that computes the cross-entropy loss and an L_2 regularization term on the weight matrix. I also used cyclic learning rate algorithm to eliminate much of the trial-and-error associated with finding a good learning rate and some of the costly hyper-parameter optimization over multiple parameters. And finally I did coarse and fine search for λ, i.e., the amount of regularization term.

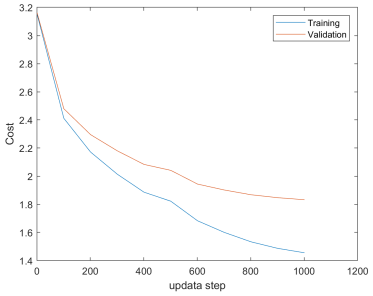
2 Results and conclusions of assignment 2

2.1 Analytic gradient computations check

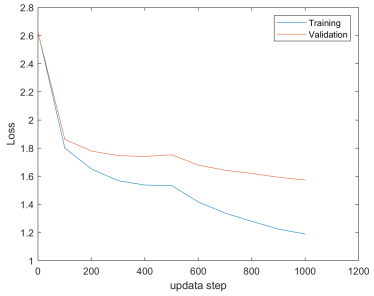
The numerical gradient should be compared to the analytical gradient to ensure that the analytical gradient is computed correctly. My way to check gradient computation was to compare the numerically and analytically computed gradient vectors (matrices) on reduced version of the input data with reduced dimensionality, by examining their absolute differences and declaring if all these absolute differences are small ($\leq 1e-6$).
All the absolute differences of W_2 are smaller than $1e-7$.
All the absolute differences of b_2 are smaller than $1e-8$.
All the absolute differences of W_1 are smaller than $1e-7$.
All the absolute differences of b_1 are smaller than $1e-8$.
Then I can draw the conclusion that my analytical gradient computation was correct.

2.2 Results of cyclic learning rate

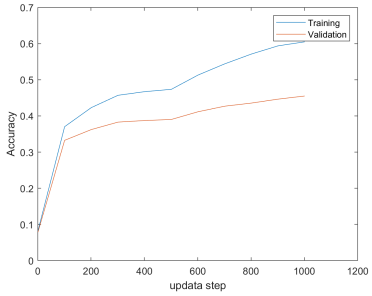
The curves for training and validation loss/cost when using the cyclic learning rates (for just one cycle) with $eta\_min = 1e-5$, $eta\_max = 1e-1$, $\lambda = 0.01$, $n_s = 500$, $batch\_size = 100$ are as follows.
After only one cycle of training is performed, the classification accuracy on test set is 46.19%



(a) Cost plot

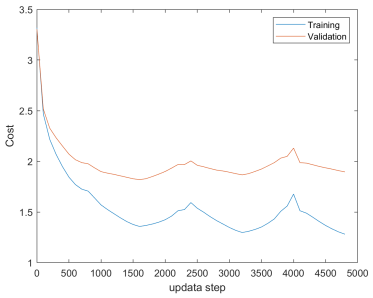


(b) Loss plot

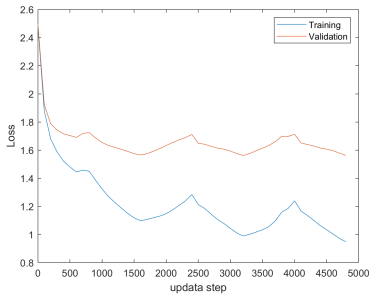


(c) Accuracy plot

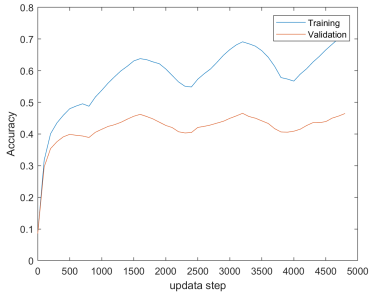
As we can see from the graphs above, during the only one cylce training process, the cost/loss keeps decreasing and the accuracy keeps increasing.
The curves for training and validation loss/cost when using the cyclic learning rates (for three cycle) with $eta\_min = 1e-5$, $eta\_max = 1e-1$, $\lambda = 0.01$, $n_s = 800$, $batch\_size = 100$ are as follows.
After three cycles of training, the classification accuracy on test set is 47.79%.



(a) Cost plot



(b) Loss plot



(c) Accuracy plot

From the graphs above, we can clearly see that there are three cycles in the training process and at the end of each cycle, the cost/loss is minimal of the whole cycle, that is, at the end of each cycle of training the network should be in the vicinity of a local minimum. Therefore, if we use cyclic learning rate algorithm, we should let the training process terminate at the end of a cycle to improve the performance.

2.3 Coarse search for λ

To perform the search, I used all the five batches minus a subset for validation (5000 images).
The range of the values I searched for λ is $[10^{-5}, 0.1]$, I used two cycles to training the network during the coarse search for λ. And other parameters are $eta\_min = 1e-1$, eta_max, $batch\_size = 100$, $n\_s = 2floor(n/n_batch) = 900$. The validation performance of these parameter settings are as follows.

λ	10^{-5}	$10^{-4.5}$	10^{-4}	$10^{-3.5}$
Validation accuracy	51.38%	51.22%	51.48%	52.20%
λ	10^{-3}	$10^{-2.5}$	10^{-2}	$10^{-1.5}$
Validation accuracy	51.32%	51.70%	50.90%	46.22%

Table 1: Validation Accuracy as lambda varies

The hyper-parameter settings for the 3 best performing networks are:

1. $\lambda = 10^{-4}$

	Training data	Validation data	Test data
Accuracy	58.93%	51.76%	51.11%

Table 2: Classification Accuracy after two cycles training

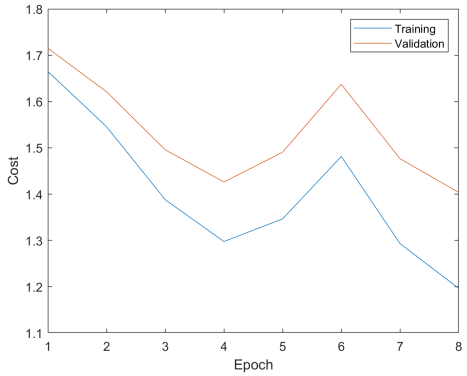


Figure 3: Cost plot

2. $\lambda = 10^{-3.5}$

	Training data	Validation data	Test data
Accuracy	58.72%	51.84%	50.59%

Table 3: Classification Accuracy after two cycles training

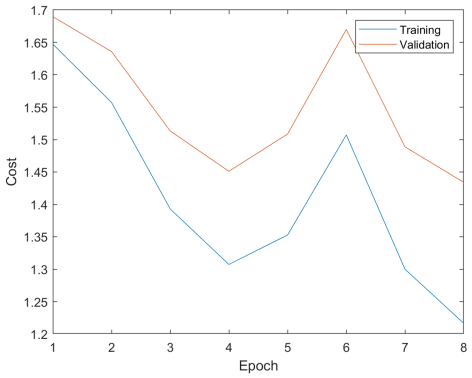


Figure 4: Cost plot

3. $\lambda = 10^{-2.5}$

	Training data	Validation data	Test data
Accuracy	57.34%	51.90%	51.64%

Table 4: Classification Accuracy after two cycles training

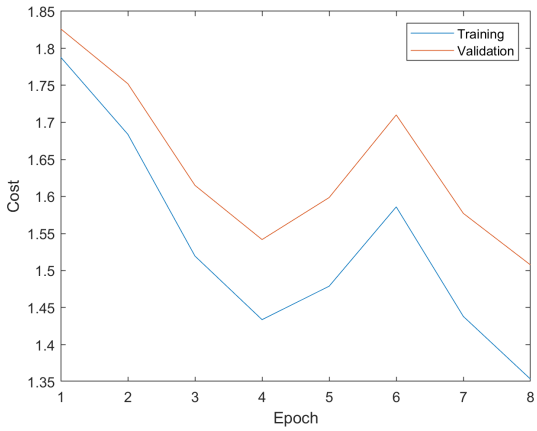


Figure 5: Cost plot

2.4 Fine search for λ

According to the result of coarse search for λ, the range of the values of fine search for λ is $[10^{-4}, 10^{-2}]$. I used three cycles to trained the network, and other parameters are $eta\_min = 1e-1$, eta_max, $batch\_size = 100$, $n\_s = 2floor(n/n_batch) = 900$. The validation performance of these parameter settings are as follows. The hyper-parameter settings for the 3 best performing networks are:
1. $\lambda = 7e-4$

	Training data	Validation data	Test data
Accuracy	61.04%	51.52%	50.76%

Table 5: Classification Accuracy after three cycles training

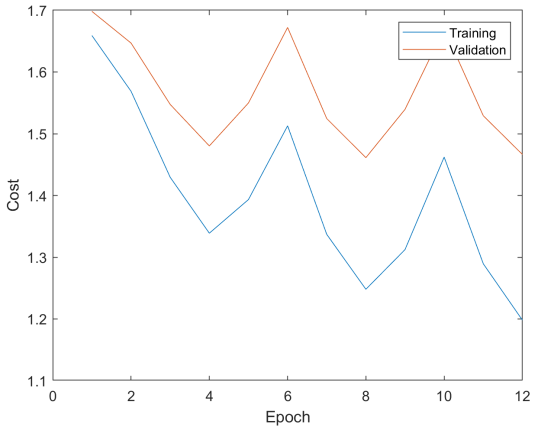


Figure 6: Cost plot

2. $\lambda = 2.9e - 3$

	Training data	Validation data	Test data
Accuracy	58.83%	52.08%	51.86%

Table 6: Classification Accuracy after three cycles training

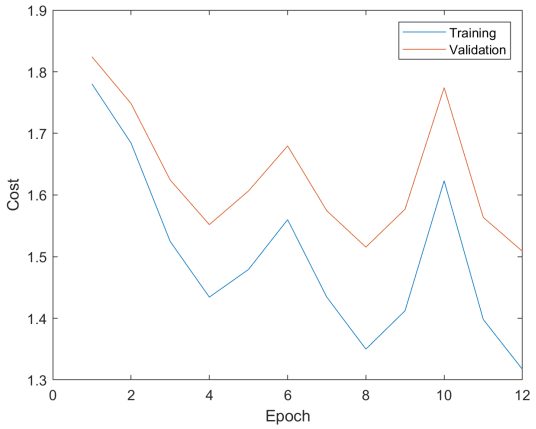


Figure 7: Cost plot

3. $\lambda = 4.3e - 3$

	Training data	Validation data	Test data
Accuracy	57.29%	52.28%	52.13%

Table 7: Classification Accuracy after three cycles training

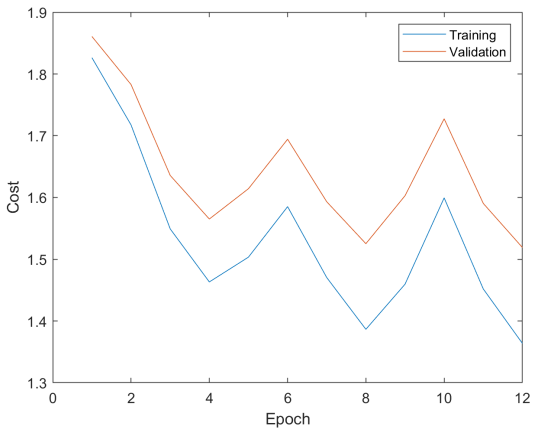


Figure 8: Cost plot

2.5 Best learnt network

According to performance on the validation set, the best found λ setting is $\lambda = 4.3e-3$. I trained the network on all the training data except for 1000 examples in a validation set, for 3 cycles. Other parameters are: $eta\_min = 1e-1$, eta_max, $batch\_size = 100$, $n\_s = 2floor(n/n_batch) = 980$. The graph of the training and validation loss is shown as below.

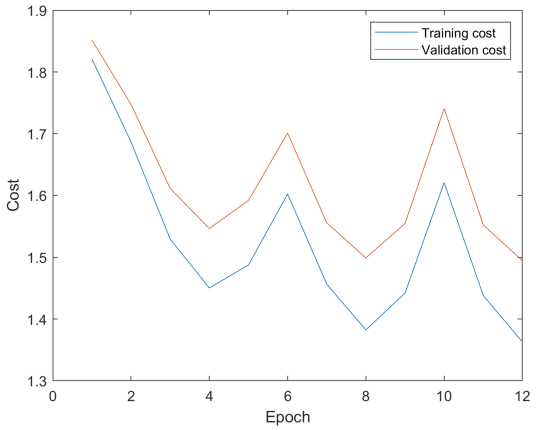


Figure 9: Cost plot for three cycles training

The learnt network's performance (classification accuracy) is shown in the following table.

	Training data	Validation data	Test data
Accuracy	57.51%	52.30%	52.52%

Table 8: Classification Accuracy after three cycles training