Assignment Two DD2424

For this assignment the goal is to build and train a two layer neural network which can handle multiple outputs. The dataset used for training and testing is the CIFAR-10 image dataset. The training is done using a mini-batch gradient descent applied to a cost function, computing the entropy loss.

Gradient Computations

In order to determine the state and performance of my gradient calculations, I have compared them with that of a more accurate, numerical method of calculating the gradients. The evaluation is done using a relative error which can be seen in equation 1.

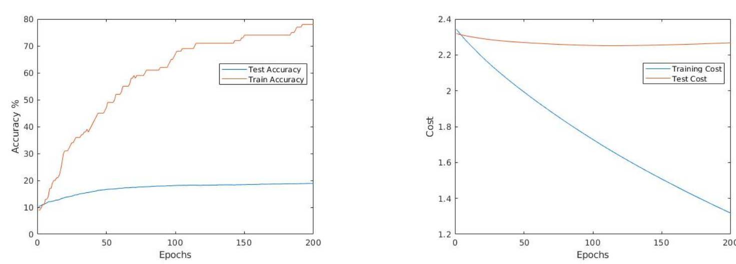
$$\frac{|g_a - g_n|}{max(eps, |g_a| + |g_n|)} \tag{1}$$

The results of the relative error between the two methods can be seen in table 1. The evaluation was done using two different batch sizes of the training examples. Since the results were deemed small enough, I deemed my method of calculating gradients sufficient.

Batches	2	50
W1	3.71e-10	3.56e-9
W2	8.05e-11	1.53e-9
b1	2.86e-10	8.30e-10
b2	5.72e-10	2.45e.8

Table 1: Relative error of numerical and analytical computation of gradients

For extra verification of my gradient calculations I also checked whether i could overfit the training data by using a small bath of training examples and iterating over them many times. In figure 1 we can see a good example of this being successful.

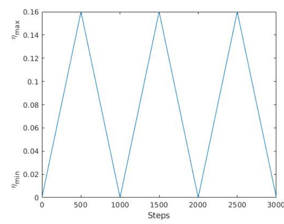


(a) Training and Test cost (b) Training and Test accuracy

Figure 1: Overfitting the data using 100 training samples, λ set to 0, 200 epochs, number of batches set to 100 and η at 0.01.

Cyclic Learning Rate

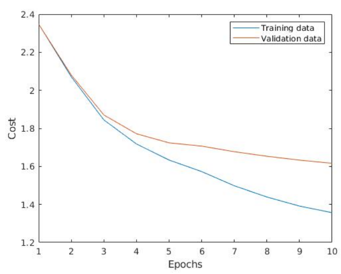
Cyclic learning rate is a method to help us deal with the problem of finding the optimal η. It enables us to deal with the trail and error approach of optimizing the value. In figure 2 we can see how the η parameter shifts during training up and down between a defined η_max and η_min with a set step size n_s between iterations.



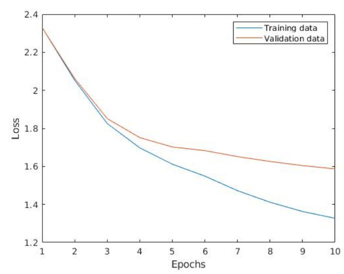
(a) Cyclic Learning Rate

Figure 2: η shifting between max and min values

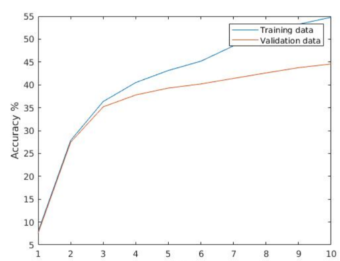
In figure 3 we can see an example of the cycle learning rate in progress. In just 1 cycle of training we managed to get a test accuracy of 44.6%.



(a) Cost



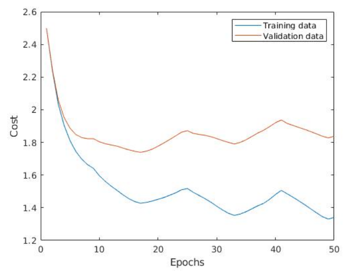
(b) Loss



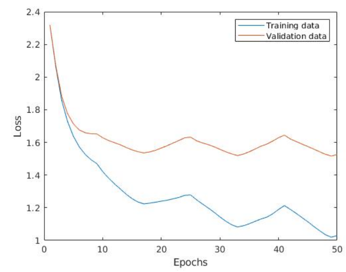
(c) Accuracy

Figure 3: Training done for 1 cycle with cyclic learning rate. Parameters set to $\lambda = 0.01$, number of batches = 100, η_{max} = 1e-1, η_{min} = 1e-5 and $n_s = 500$

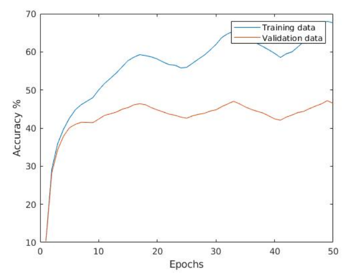
In figure 4 we can see more clearly the impact of the cyclic learning rate than in figure 3. In this example we train the model over three cycles which can be clearly seen looking at the waves of the cost, loss and accuracy graphs which is caused by the shifting η. The training resulted in a test accuracy of 46.1%.



(a) Cost



(b) Loss



(c) Accuracy

Figure 4: Training done for 3 cycles with cyclic learning rate. Parameters set to $\lambda = 0.01$, number of batches = 100, η_{max} = 1e-1, η_{min} = 1e-5 and $n_s = 800$

Lambda Search

Finding a good value for the regularization term λ can be very tricky. In this assignment we test an approach where we train the network with different λ values. Training is only done for two cycles and then we save training and test accuracy with the different λ values and evaluate its performance. For the tests, the λ values are generated by setting a λ_{min} and λ_{max} and using Equation 2 3.

$$L = \lambda_{min} + (\lambda_{max} - \lambda_{mim}) * random(0, 1) \tag{2}$$

$$\lambda = 10^L \tag{3}$$

For the tests I use a cyclic learning rate between η_{max} 1e-1 and η_{min} 1-e5 n_s set to 900. From 15 iterations with different lambda values the three best results can be seen in table 2. The training is done using 45 000 training examples and 5000 examples used for validation and performance evaluation.

	Test Accuracy	Train Accuracy	λ
1	52.40	58.46	1.458e-4
2	52.18	59.98	9.38e-4
3	51.98	62.21	2.65e-4

Table 2: Three best results with λ_{min} set to -1 λ_{max} set to -5, over 2 cycles

Looking at table 2 we can see that the best performing lambdas are small values between 1e-3 and 1e-4. I therefore tried another run where i set λ_{min} -3 and λ_{max} -7, to try and explore even smaller values and watch their performance.

	Test Accuracy	Train Accuracy	λ
1	51.42	58.15	2e-6
2	51.36	58.21	6.8e-5
3	51.14	58.04	5.5e-5

Table 3: Three best results with λ_{min} set to -3 λ_{max} set to -7, over 2 cycles

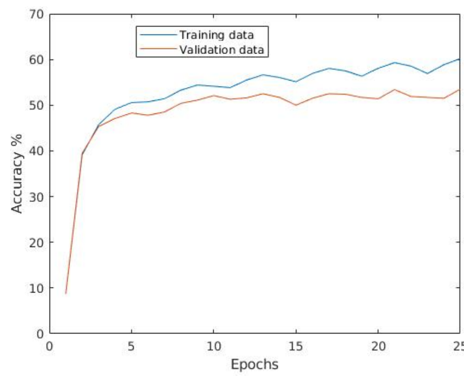
The results seen in table 3 shows that the best performances are very similar and none of them managed to outperform the previous trial run. Therefore as a third attempt i did another run with 15 lambda values with λ_{min} set to -2 λ_{max} set to -4.

	Test Accuracy	Train Accuracy	λ
1	52.72	57.41	5.19e-4
2	52.64	58.14	1.52e-4
3	52.38	57.70	4.25e-4

Table 4: Three best results with λ_{min} set to -2 λ_{max} set to -4, over 2 cycles

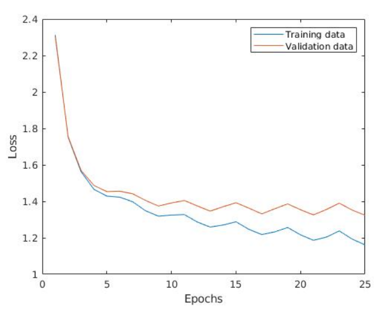
Final Training

Centering the search for lambda successfully enabled me to find an even better lambda value in terms of test performance which can be seen in table 4. Therefore as a result i used λ set to 5.19e-4 for a final training test. This time I trained a model over 5 cycles with all 50 000 training examples. For testing i used 1000 samples from the test batch. The cyclical learning rate was set η_{max} 1e-1 and η_{min} 1-e5, n_s was set to 1000. The accuracy of the model can be seen in figure 5. The test accuracy after 5 cycles of training was 53.5% and a training accuracy at 60.2%. The cost and loss calculated at each epoch of training for the test and training set can be found in figure 6.

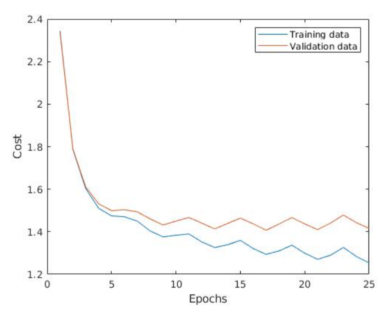


(a) Training and Test Loss

Figure 5: Final training with λ = 5.19e-4 over 5 cycles with 50 000 training examples. Test Accuracy 53.5% and training accuracy at 60.2%



(a) Training and Test Loss



(b) Training and Test Cost

Figure 6: Final training with λ = 5.19e-4 over 5 cycles with 50 000 training examples