Assignment One DD2424

For this assignment the task was to build, train and test a one layer network with multiple outputs to classify images using the CIFAR-10 dataset. By implementing a mini-batch gradient descent method I succefully managed to reach an accuracy of around 36%. The functions were written using matlab through help of the lab assignment information and lecture notes.

To test and evaluate the accuracy of my implementation I cross checked the gradient vector results generated by my own gradient computations versus a slower more accurate implementation and compared their relative errors which can be seen in equation 1. I calculated the average relative error, average absolute error as well as the max error between the two gradient calculations of both the absolute and the relative calculations, results can be seen in table 1 and 2.

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

Once I was assured to have a sufficiently accurate function for calculating the gradients, I carried on with the training process. To verify that my training was working properly, I checked the cost function after 1 epoch versus that of results provided by the professor responsible for the assignment. By initializing the weights and bias vector to the same specified values, I managed to verify that the training was working as intended.

	N = 20 $\lambda = 0$	N = 100 $\lambda = 0$	N = 20 $\lambda = 0.1$	N = 100 $\lambda = 0.1$
Max W	5.64e-10	9.73e-10	4.62e-10	9.78e-10
Max b	4.01e-10	7.56e-10	4.85e-10	4.26e-10
Mean W	5.57e-13	1.59e-12	5.22e-13	1.32e-12
Mean b	1.90e-10	4.83e-10	2.30e-10	2.15e-10

Table 1: Absolute Error

	N = 20 $\lambda = 0$	N = 100 $\lambda = 0$	N = 20 $\lambda = 0.1$	N = 100 $\lambda = 0.1$
Max W	8.24e-7	2.34e-7	1.51e-8	4.26e-7
Max b	3.93e-8	1.50e-7	1.39e-8	1.35e-7
Mean W	2.80e-10	1.89e-10	1.16e-11	2.76e-10
Mean b	5.50e.9	2.28e-8	4.44e-09	1.97e-8

Table 2: Relative Error

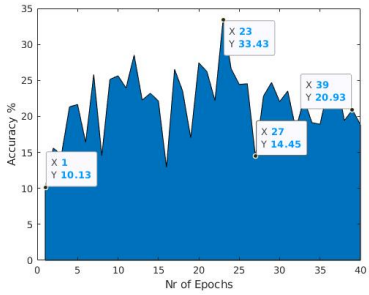
Results

The training was performed using data batch 1 and testing using the test batch from the CIFAR-10 dataset. Looking at figure 1 which uses $\lambda = 0$ and $\eta = 0.1$ we can see that the graph fluctuates a lot between iterations. The η parameter is the learning rate of the model. If η is set too low then the training might not be effective enough, which results in the model having to train for a long time. If the η is set too high however then it may cause the weights to shift too much between steps. Figure 1 displays this last problem quiet well, the high η parameter causes the model to shift its beliefs too much on each training examples, causing the accuracy and cost to fluctuate which in turn makes it hard to find and converge towards a local minimum, which is what we are after.

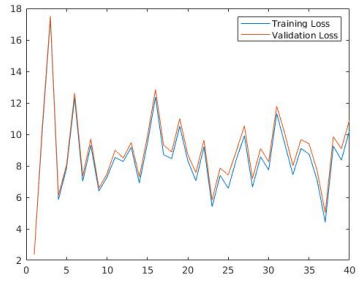
Looking at the cost graph in figure 2 and 4 visualizes the differences the regularization term λ has on the training. When regularization is introduced, the model converges towards a local minimum faster. The idea with regularization in training is to make the model learn from training examples, but not so much that it overfits its parameters based on the examples. What we want is for the model to be able to generalize what it learns and apply it to new data. If we set the λ variable too high however, then the model converges to quickly and makes the mode too general which prevents it from improving its performance from further training, an example of this can be seen in figure 3.

The trick with training a good model is setting good η and λ parameters during the process. The ideal goal is to get a model which can get good performance without having to train for too long. In this assignment we achieved a pretty good learning rate with *ta* set to 0.1, the curves as can be seen are a lot smoother then. We would also of course like the model to be a good fit and be able to perform well on previously unseen data.

(a) Trained Image with new W



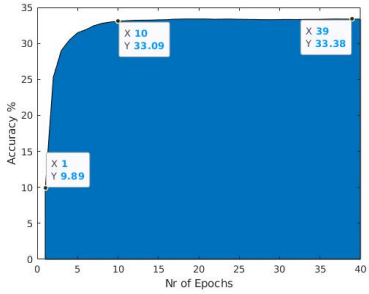
(b) Accuracy



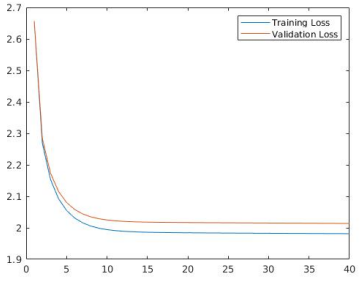
(c) Training and Validation Cost

Figure 1: $\lambda = 0, numberepochs = 40, numberbatches = 100, \eta = 0.1$

(a) Trained Image with new W



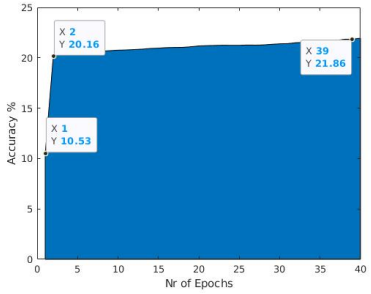
(b) Accuracy



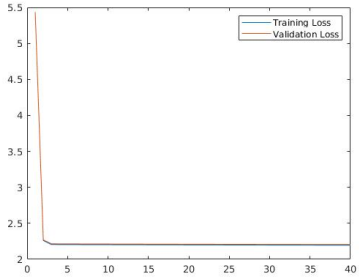
(c) Training and Validation Cost

Figure 2: lambda = 0.1, number epochs = 40, number batches = 100, eta = 0.01

(a) Trained Image with new W

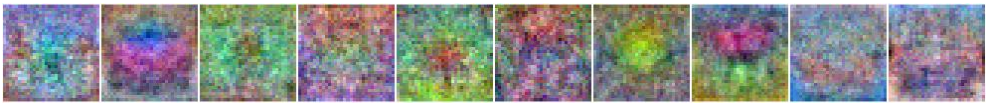


(b) Accuracy

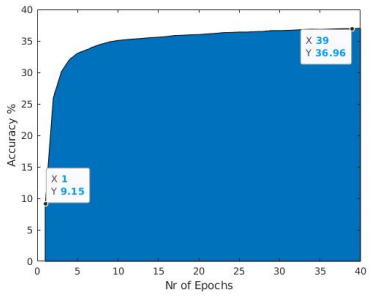


(c) Training and Validation Cost

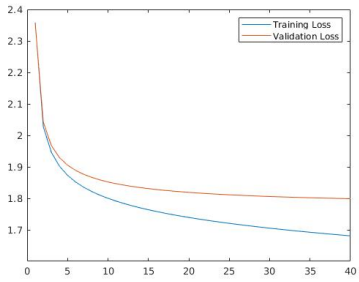
Figure 3: lambda = 1, number epochs = 40, number batches = 100, eta = 0.01



(a) Trained Image with new W



(b) Accuracy



(c) Training and Validation Cost

Figure 4: lambda = 0, number epochs = 40, number batches = 100, eta = 0.01