Assignment 1 - Deep Learning DD2424

Erifili Ichtiaroglou

April 2017

1 Exercise 1

For this exercise we had to correctly calculate the gradients of the softmax function so that we could learn the parameters of our model (W, b). We implemented the mini-batch algorithm and we tested the model using different values for the regularization term λ and the learning rate η. Below we present the graphs from the results of our model for the different parameter settings that were given to us. We observe that when we use a dig value for the learning rate the model is not stable, as it jumps through the minima. In addition when we use high regularization factor the accuracy of the model drops, as regularization part becomes more important than the learning part.

Figure 1: Mini-batch learning with $\lambda = 0$, $\eta = 0.1$, $epochs = 40$, $batch = 100$
accuracy at the validation set 30.56%

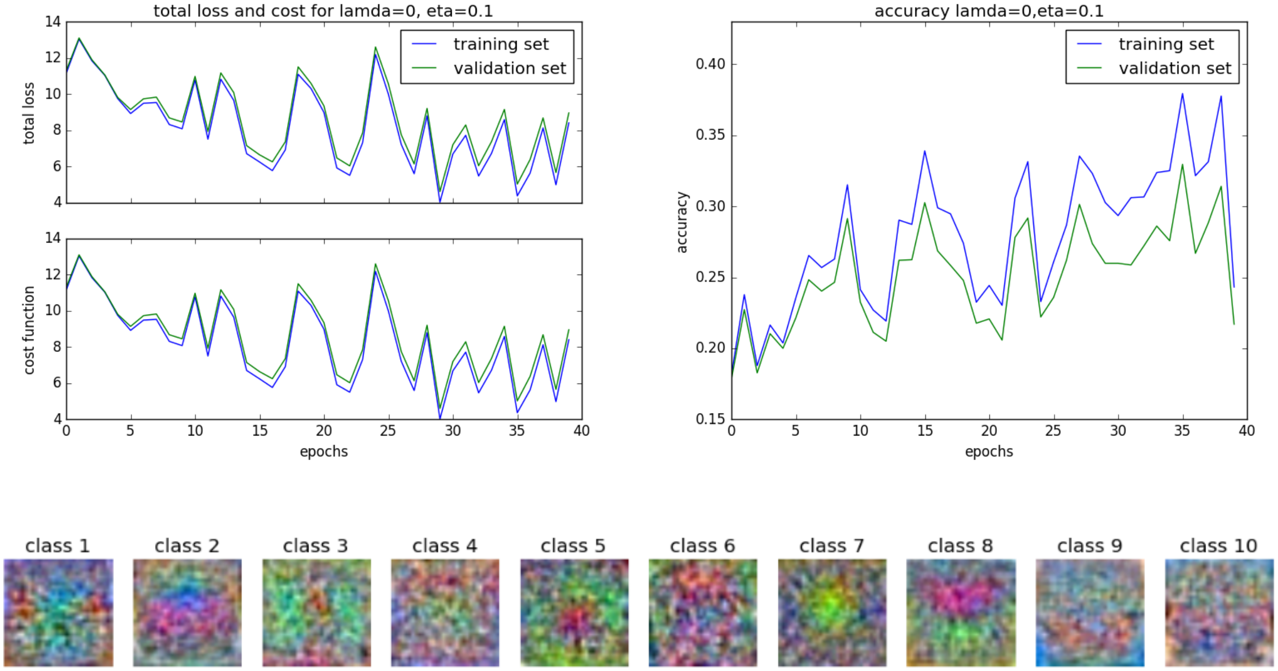


Figure 2: Mini-batch learning with $\lambda = 0$, $\eta = 0.01$, $epochs = 40$, $batch = 100$
accuracy at the validation set 37.53%

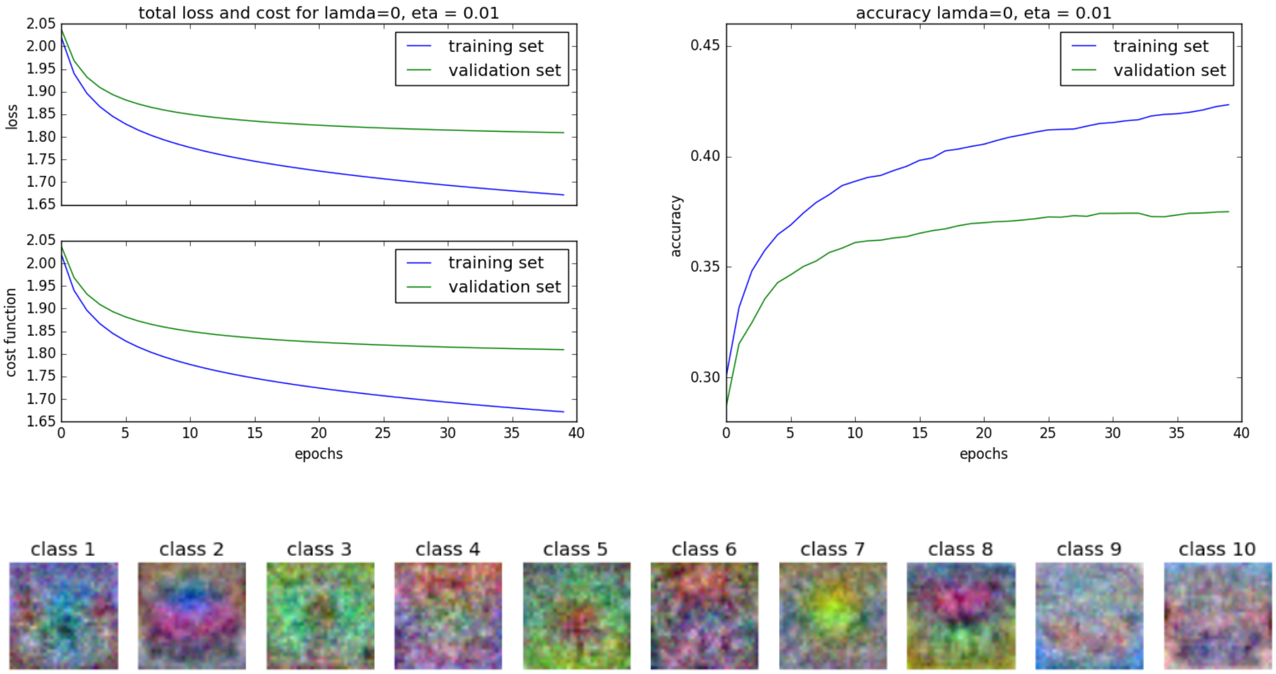


Figure 3: Mini-batch learning with $\lambda = 0.1$, $\eta = 0.01$, $epochs = 40$, $batch = 100$ accuracy at the validation set 34.09%

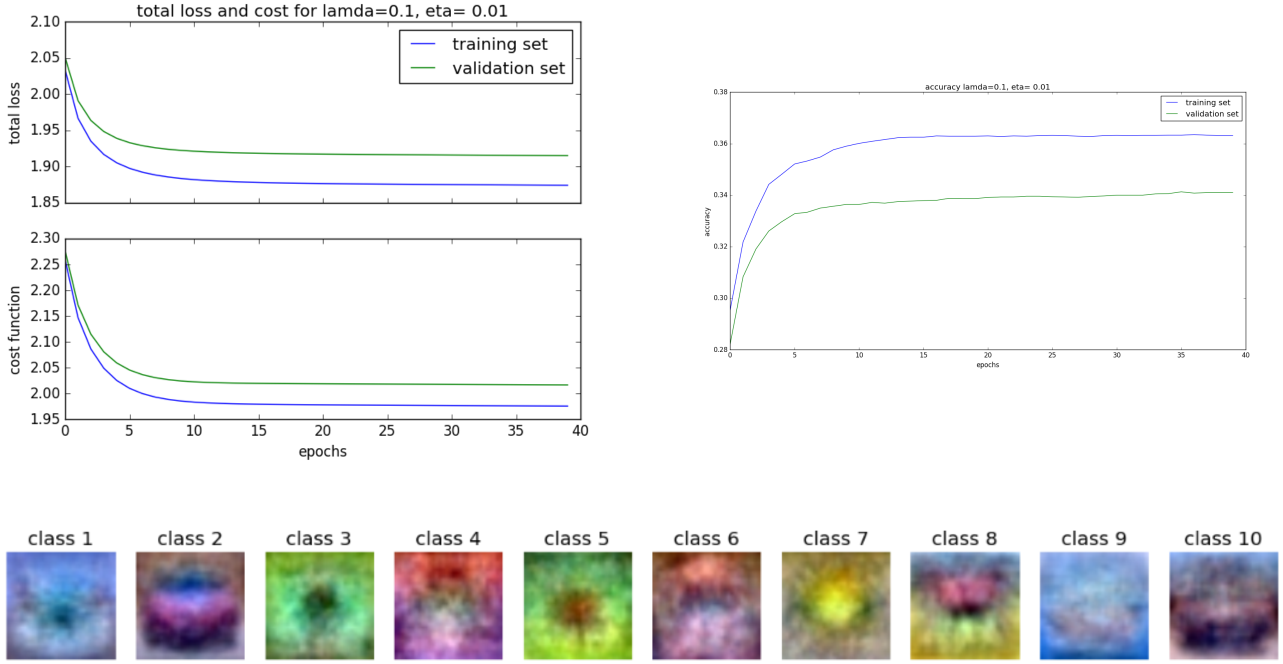
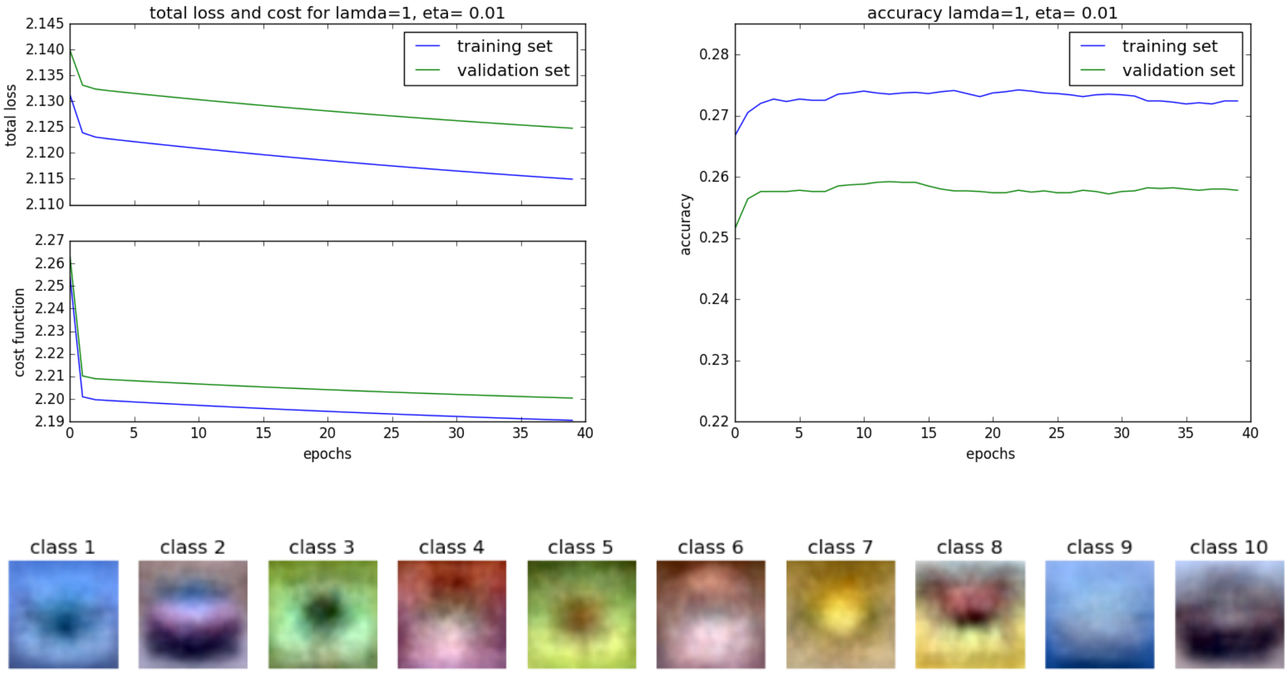


Figure 4: Mini-batch learning with $\lambda = 1$, $\eta = 0.01$, $epochs = 40$, $batch = 100$
accuracy at the validation set 25.78%



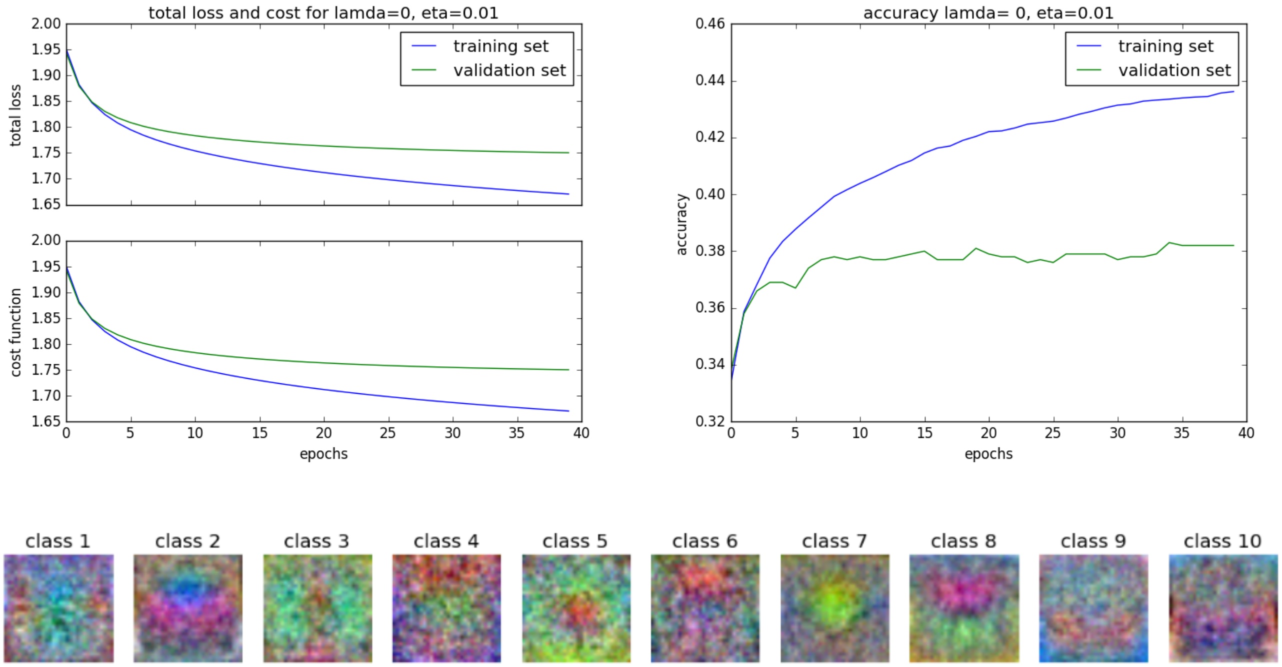
2 Exercise 2

For this part we had to try some methods to optimize the performance of our model.Below we will describe the methods that were implemented.

2.1 Use more training data

For this trick we trained the model using 1900 images as inputs and validated it using only 1000 images. That means we had more information for the model to learn. The parameters of the model were set as $\lambda = 0$, $\eta = 0.01$, $epochs = 40$, $batch = 100$. Doing that we manage to achieve an accuracy of 38,2%, instead of 37.53% when using 10000 images as inputs.

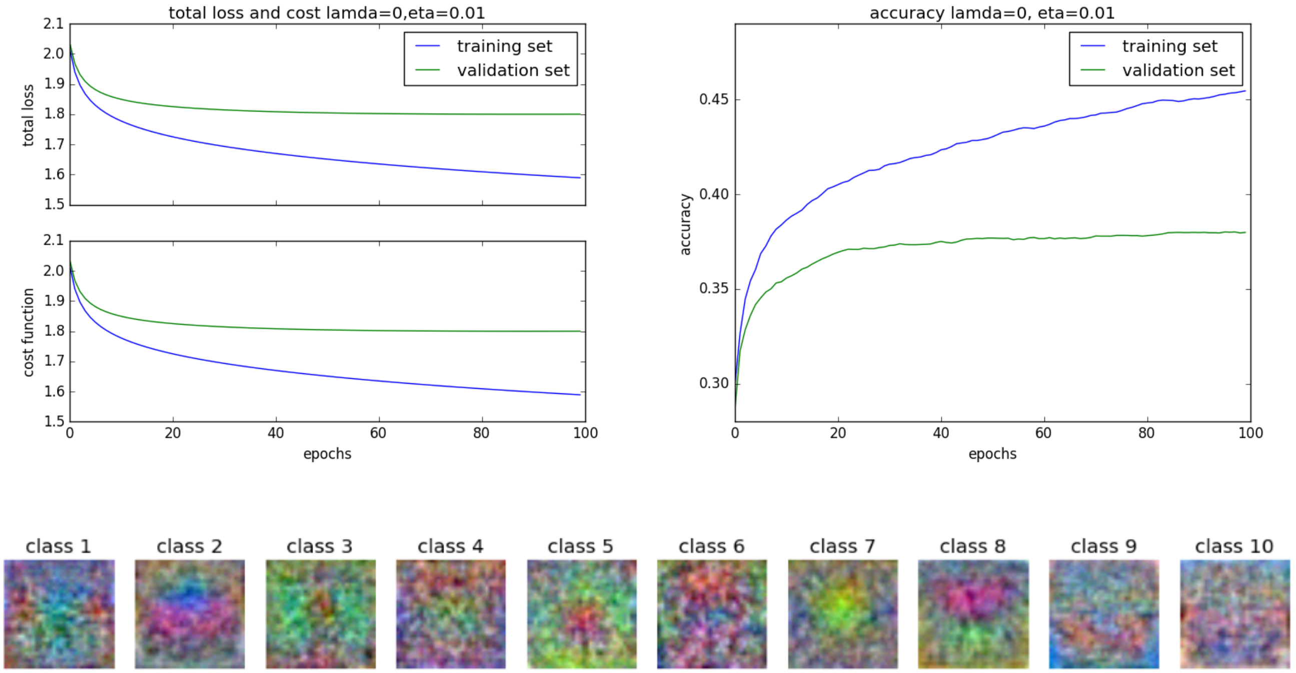
Figure 5: Mini-batch learning with $\lambda = 0$, $\eta = 0.01$, $epochs = 40$, $batch = 100$
accuracy at the validation set 38.20%



2.2 Train for a longer time

For this trick we trained the model for 100 epochs instead of 40 and we return the set of parameters W, b with the best accuracy for the validation set. The parameters of the model were set as $\lambda = 0$, $\eta = 0.01$, $epochs = 100$, $batch = 100$. Doing that we achieved an accuracy of 38,01%, instead of 37.53% when we were training for 40 epochs.

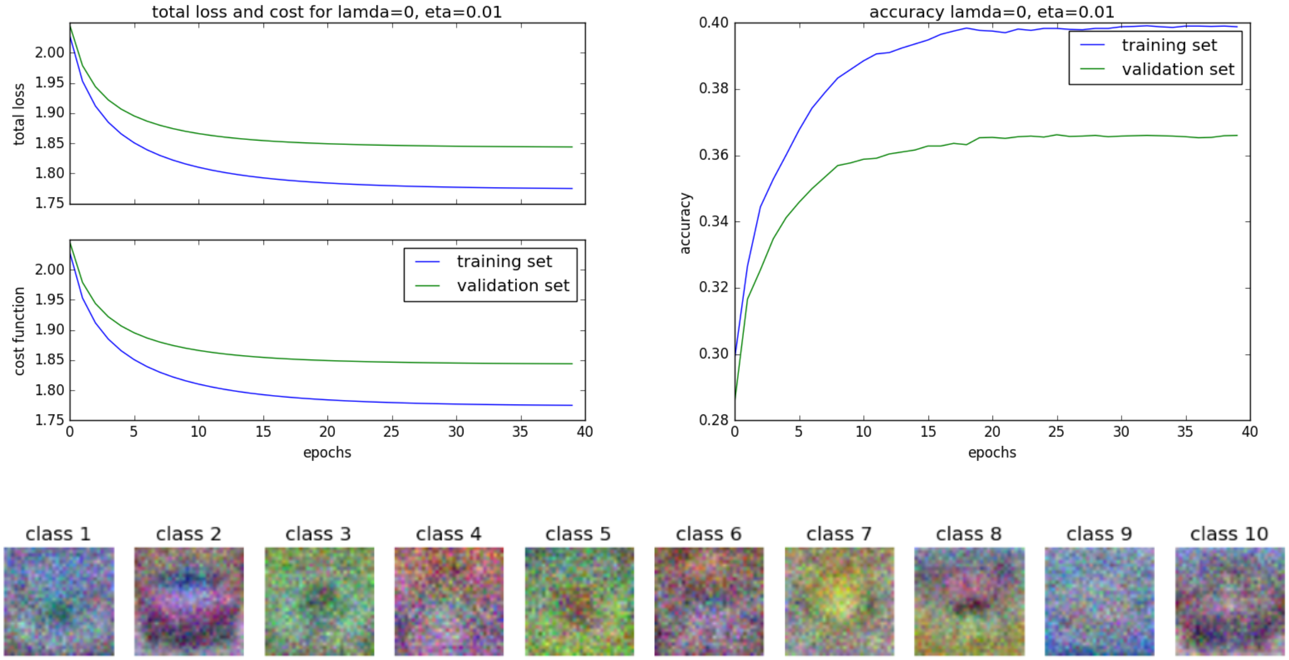
Figure 6: Mini-batch learning with $\lambda = 0$, $\eta = 0.01$, $epochs = 100$, $batch = 100$
accuracy at the validation set 38.01%



2.3 Decaying the learning rate

For this trick after each epoch we tried to reduce the value of learning rate by a factor of 0.9. Using this trick the model learned more quickly, as the cost function seems to reach low values after the first 25 iterations. The parameters of the model were set as $\lambda = 0$, $\eta = 0.01$, $epochs = 40$, $batch = 100$. Doing that we achieved an accuracy of 36,8%, instead of 37.53%.

Figure 7: Mini-batch learning with $\lambda = 0$, $\eta = 0.01$, $epochs = 100$, $batch = 100$
accuracy at the validation set 36.8%


total loss and cost for lamda=0, eta=0.01
total loss
cost function
training set
validation set
epochs
accuracy lamda=0, eta=0.01
accuracy
class 1
class 2
class 3
class 4
class 5
class 6
class 7
class 8
class 9
class 10