Deep Learning (DD2424) Assignment 2

Badi Mirzai

July 2021

1 Intro

In this assignment, we train a two-layer perception on the CIFAR-10 data set, which contains 10000 images of labeled objects. Furthermore, we use cyclic learning rates and a grid search for the optimal lambda regularization in order to optimize the hyper-parameter settings.

The input layer is 3072 pixeles values from each image with softmax as activation function for the output layer and 50 nodes on the hidden layer, while the output layer is 10 nodes. The classification of the image is done by choosing the output with the highest predicted probability. The network is trained with gradient descent through back propagation of the cross entropy loss. The trained network achieved an accuracy of 63.5% train dataset and a test accuracy of 53.1%.

2 Gradient checks

In this assignment, the gradients of the neural network was checked with numerical approximations. The gradients was checked similar to how I did it in the first assignment, checking with the first 50 parts of the input dimensions of the 0th image, which yielded an maximum absolute error in the magnitude of 10^{-8} (for the weights and bias derivatives) and an maximum relative error in the order of 10^{-7}. These were the comparisons with the fast numerical gradients. For the slow numerical gradients that was much more precise, the absolute error descried to the order 10^{-10}.

The error for the entire 100th image was also computed. The absolute error was in the order or 10^{-7} and 10^{-8} for the weights and bias respectively (with the fast approximator). The maximum relative error was in the magnitude of 10^{-10} for both weights and biases. Similarly for the slow-grad check, all the maximum errors was in the order of $2.5 * 10^{-6}$ or lower.

I am quite certain than the gradients were correctly calculated, based on the tests and results obtained.

The Network was also checked by training the two-layer netowrok on the first 50 images for 200 epoches, where it was seen that overfitting occured. eta = 0.001, batch size was set to 10. See Figure 1 for this demonstration.

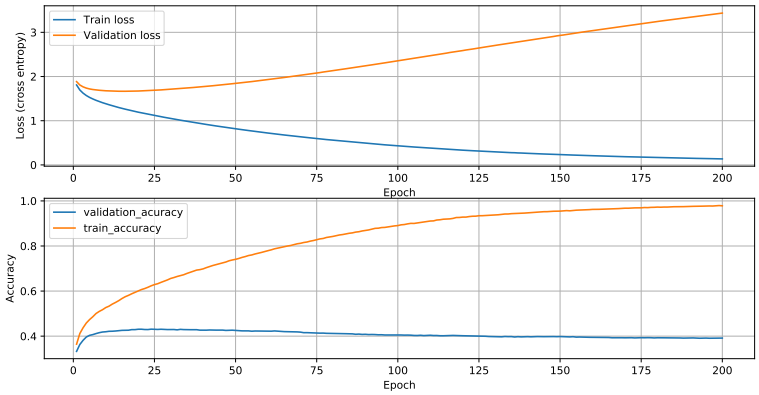

Figure 1: Loss functions and accuracy over 200 epoches on the 50 first images in the CIFAR 10 dataset. This demonstrates overfit which implies that the calculated gradients seems correct. for last epoch, train loss= 0.13603, train accuracy=0.9783. $\lambda = 0$, $n_{epochs} = 200$, $n_{batch} = 10$, $\eta = 0.1$.

3 Trained Network results

The models was trained on the CIFAR-10 data-set, where mini-batches of size 100 was used. The training data was shuffled for each epoch, where the loss and accuracy was calculated each epoch for the training and validation data sets. The input data was pre-processed by first normalized to one (division by 255). The mean and variance of the training data was calculated and used as a way to center the training, test and validation input data for increasing the performance.

For the course random search, I uniformlly sampled diferent lambda values between 10^{-5} and 10^{-1} (see figure 4) and the corresponding best values is in Table 1. Here (for both the course search and fine search), we trained on 45000 images and used 5000 images for validation. We ran for 2 cycles, with $n_{epoch} = 10$, $\eta_{min} = 10^{-5}$, $\eta_{max} = 10^{-1}$, $n_{batch} = 100$, $n_s = 900$. After the course search, we repeated the lambda search on a narrower range of lambda values, see figure 5. The three best lambda values can be seen in Table 2. Finally, we trained the model on the best λ for 3 cycles, see Figure 6. The results showed that the validation accuracy reached up to 53%.

4 Discussion and conclusion

After training a two-layer perception, the results show the importance of using cyclical learning rates and varying different regularization terms in order to find the best hyperparameter settings. For the coarse search, we can see that a too

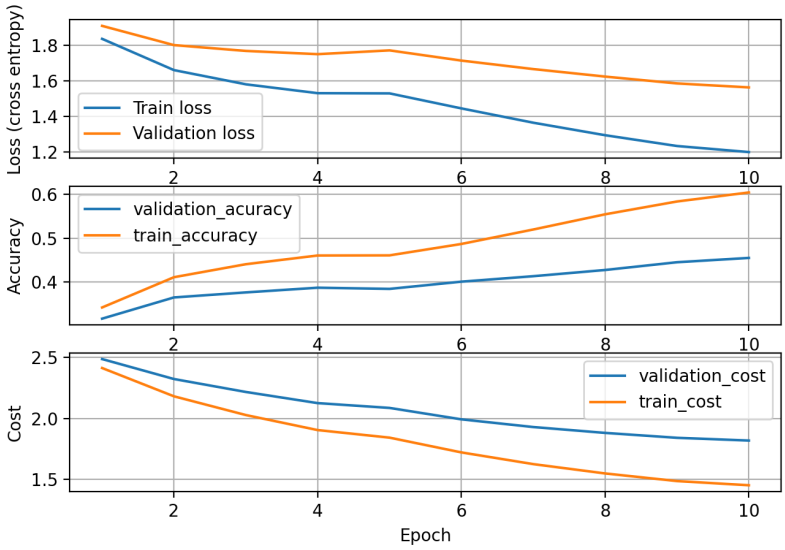

Figure 2: Curves with cyclical learning rates. $\lambda = 0.01$, $n_{epochs} = 10$ (1 cycle), $n_{batch} = 100$, $\eta_{min} = 1 * 10^{-5}$, $\eta_{max} = 0.1$, $n_s = 500$. test accuracy = 46.17%, train accuracy = 60.37%

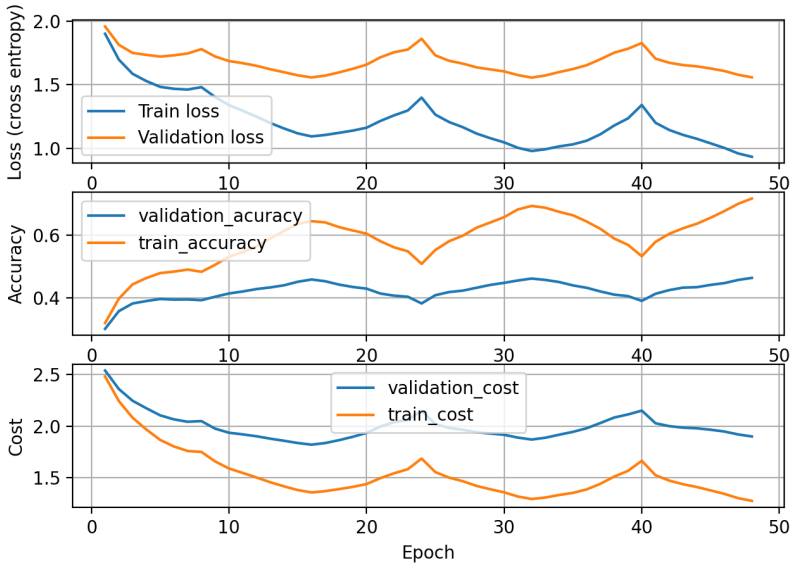

Figure 3: Curves with cyclical learning rates. $\lambda = 0.01$, $n_{epochs} = 48$ (3 cycles), $n_{batch} = 100$, $\eta_{min} = 1 * 10^{-5}$, $\eta_{max} = 0.1$, $n_s = 800$. test accuracy = 46.9%, train accuracy =71.67%.

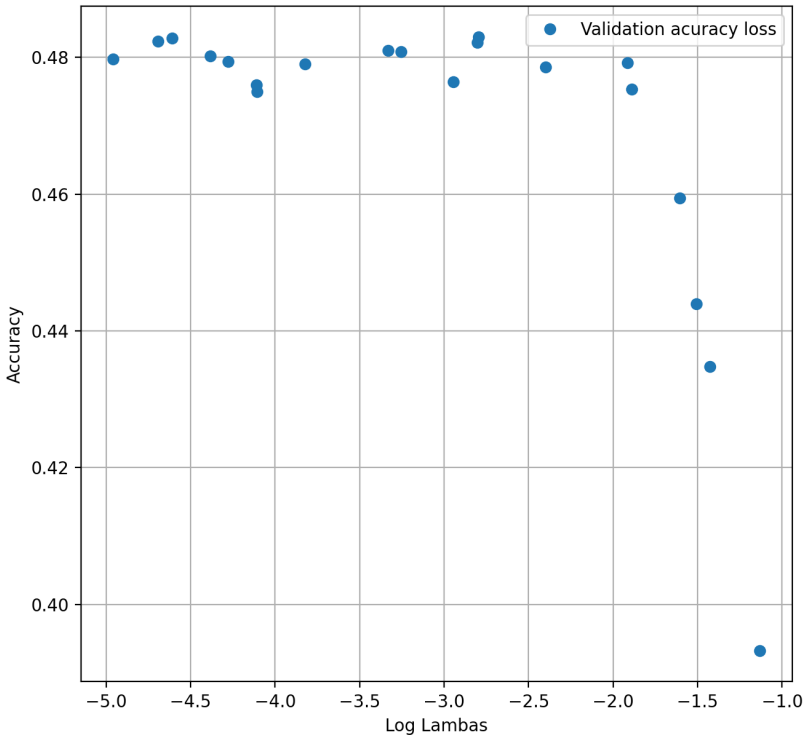

Figure 4: Validtion Accuracy over different values of the lambda regularization in the course search. Network was trained on all 5 datasets, with 5000 images as validation dataset.

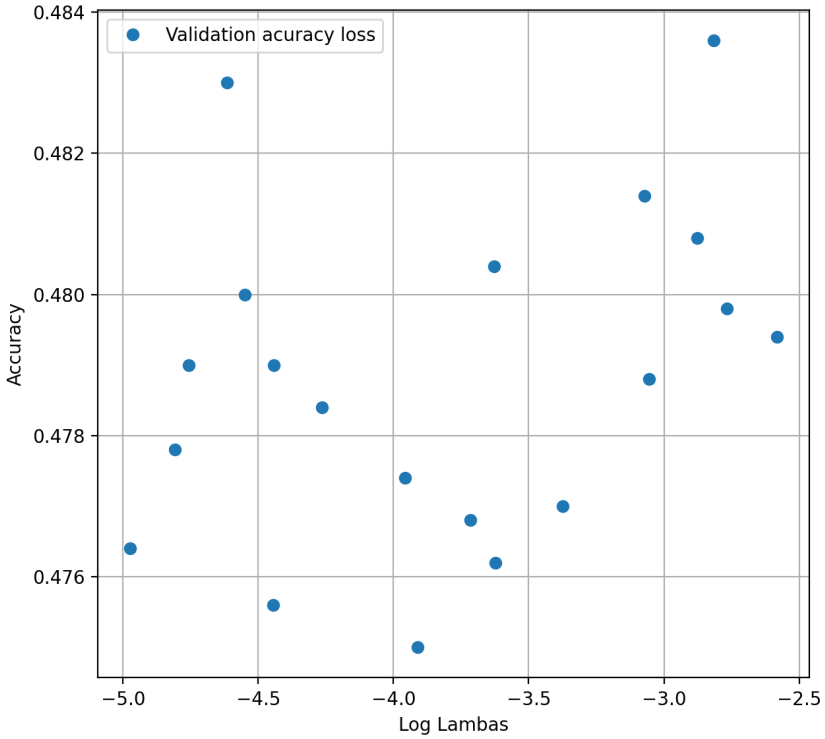

Figure 5: Validtion Accuracy over different values of the lambda regularization in the fine search. Network was trained on all 5 datasets, with 5000 images as validation dataset.

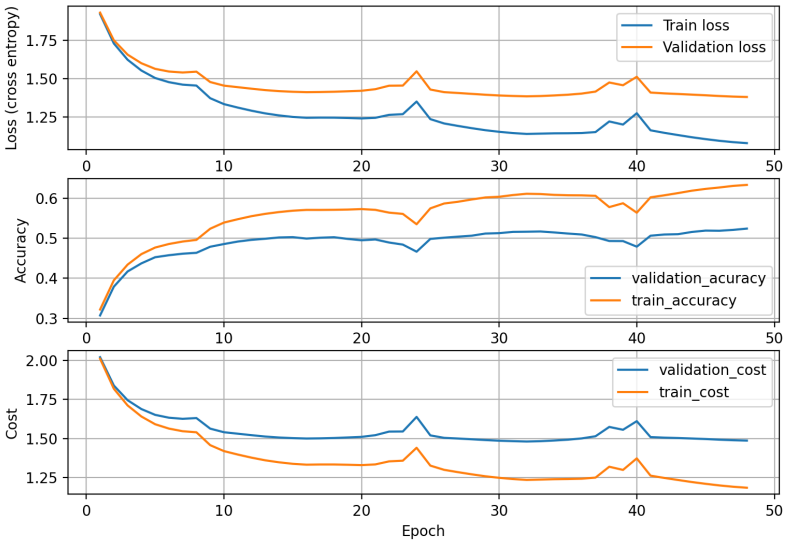

Figure 6: Curves with cyclical learning rates with the best $\lambda = 1.526 * 10^{-3}$. Here, $n_{epochs} = 48$ (3 cycles), $n_{batch} = 100$, $\eta_{min} = 1 * 10^{-5}$, $\eta_{max} = 0.1$, $n_s = 900$. Test accuracy = 53.1%, train accuracy =63.5%.

Table 1: 3 best lambdas for the course search.

$\lambda(log)$	λ	accuracy (%)
-4.610	$2.455*10^{-5}$	48.28
-4.695	$2.019*10^{-5}$	48.24
-2.797	$1.597*\ 10^{-3}$	48.30

Table 2: 3 best lambdas for the fine search.

$\lambda(log)$	λ	accuracy (%)
-2.816	$1.526*10^{-3}$	48.36
-4.615	$2.429*10^{-5}$	48.30
-3.072	$8.478*\ 10^{-4}$	48.14

small regularization term will not impact the system too much, while the a too large might cause it to be unable to learn the true model. See figure 4.

First of all having a too small learning rate will result in a slower convergence of the trained Neural Network with can be unpractical. A second search for the best lambda regularization was done in Figure 5. The results by using lambda search is clear by comparing figures 6 and 3, where the test accuracy has been increased. It is worth noting that the train accuracy has been decreased, but the gap between the test and train accuracy has decreased. This demonstrates the property of a correctly chosen λ, where over-fitting is reduced. The data was also shuffled as a result as a way to generalize the training.