Assignment 3

Max Junestrand

May 5, 2021

1 Figures, plots and comments

Each subsection covers a question of the assignment.

1.1 Gradient Checking

I used extremely similar code from last week to check the analytical gradients with the numerical, the only real differences is that we check the gradients for both ALL Ws, Bs, Gammas, and Betas, compared to last week. The relative error rate between the analytical and numerical gradients with lambda = 0 was

For the numbers here I just checked W1, W2, b1, b2, gamma1, gamma2 and beta1 and beta2, but for a k-layered network we could check as many of these as we want. I checked them when I did networks with more layers as well to make sure that the analytical code was bug free.

Figure 1: relative error rate for gradients

W1	W2
6.16E-09	3.14E-09
b1	b2
3.15E-13	5.88E-13
gamma1	gamma2
2.67E-10	4.11E-11
beta1	beta2
2.27E-10	2.63E-11

I conclude that the error rate is very low and let's continue with the assignment.

1.2 Evolution of loss function 3-layer netowrk

Here we look at the evolution of the loss function when training a 3-layered network with and without batch-normalization.

I used 45.000 images for training, and evaluate the network on 5.000 images. Lambda is set to 0.005, and eta is 0.001. The number of nodes in the layer is [50, 50, 10].

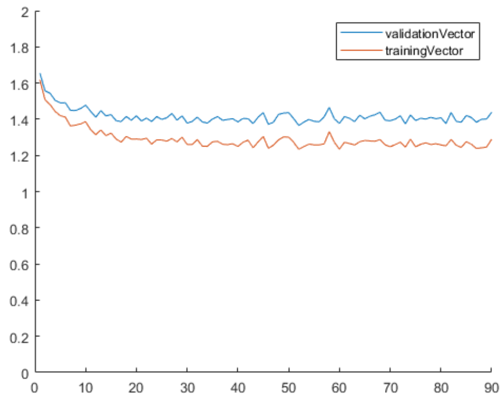

Figure 2: Loss when using batch normalization

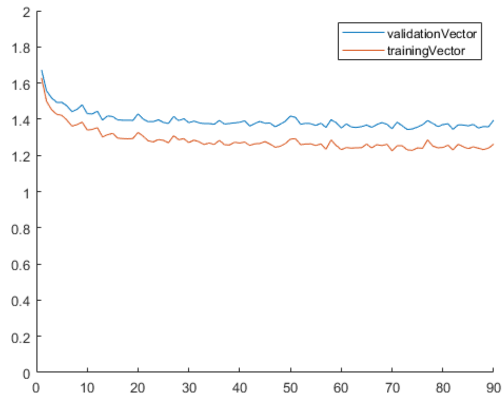

Figure 3: Loss without batch normalization

1.3 Evolution of loss function 9-layer network

Here we look at the evolution of the loss function when training a 9-layered network with and without batch-normalization.

I used 45.000 images for training, and evaluate the network on 5.000 images. Lambda is set to 0.005, and eta is 0.001. The number of nodes in the layer is [50, 30, 20, 20, 10, 10, 10, 10]

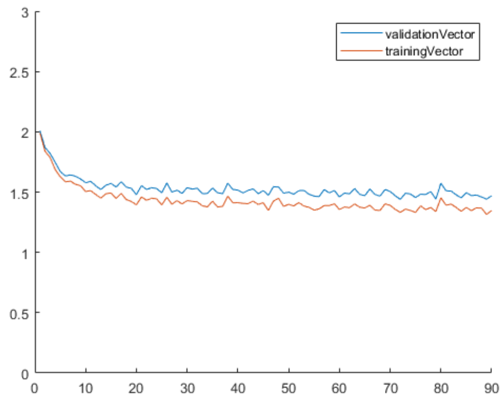

Figure 4: Loss when using batch normalization

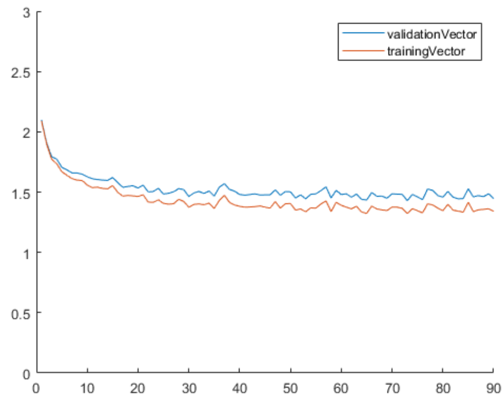

Figure 5: Loss without batch normalization

1.4 Search for Lambda

The search for Lambda begins with "testing" different lambdas by generating different values between 10e-5 and 10e-1. I generated 8 different values for lambda and noted their accuracy on the validation test. The validation set was 10% of the original available training data (out of 50.000).

I used the same algorithm as for assignment 2 to generate random values of lambda between 1e-5 and 1e-1.

The table for the different generated values of Lambda and their corresponding accuracy is found below.

Figure 6: Coarse search for a good lambda

			Lambda Search					
Accuracy	0.46	0.4934	**0.5166**	**0.5184**	0.4918	0.3868	0.4818	0.397
Lambda	0.020211	0.000343	**0.002362**	**0.002931**	3.78E-05	0.095162	1.12E-05	0.047143

We can note that the bold values of lambda and their accuracy on the validation set are the two best. Meaning that lambda = 0.002361615 and 0.002930824 were the best. With an accuracy between 51.66 and 51.84%. We will thus use these values to continue our search for a better Lambda in the next step.

1.5 Fine Search for Lambda

I evaluated the network with lambas values around the two best I found previously. None of the results were better, so we will use lambda = 0.0029 as our best. Which gave an accuracy of 51.84%

1.6 Sensitivity to initialization

According to the assignment, I shall investigate a new approach to the initialization of W, with different parameters both for batch and no batch.

There is a trend of getting worse accuracy when not using batch normalization with lower sigmas. When using sigma = 1e-4 we get extremely poor accuracy, and the loss is constant. Due to the randomness its hard to come to a conclusion about which is the best, but it seems to be a safer bet to use batch normalization.

I also used the best lambda that I had found from the previous experiments, and interesting to note is that an even higher accuracy on the validation data was found here for sigma = 1e-3: 53.04%. Again, might be due to randomness.

Here are the accuracy values for the different configurations:

Figure 7: Coarse search for a good lambda

	With batch normalization	sigma
acc	0.526	1.00E-01
acc	0.5148	1.00E-03
acc	0.52	1.00E-04
	With no batch normalization	sigma
acc	0.5304	1.00E-01
acc	0.5172	1.00E-03
acc	0.0986	1.00E-04

The loss asked for in the assignment functions are plotted below.

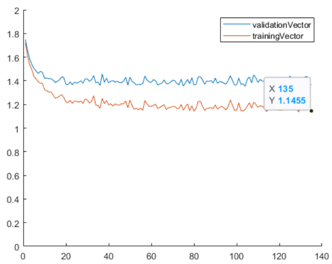

Figure 8: Loss with batch, sigma = 1e-1

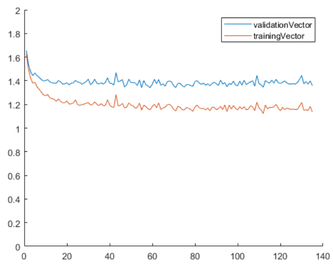

Figure 9: Loss with batch, sigma = 1e-3

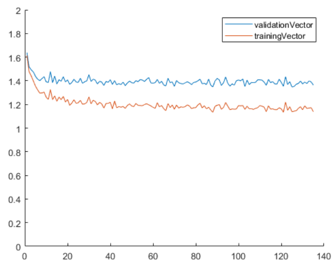

Figure 10: Loss with batch, sigma = 1e-4

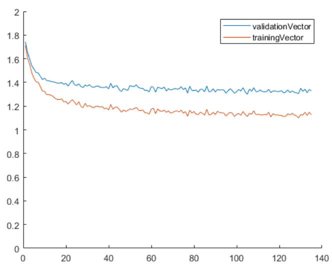

Figure 11: Loss without batch, sigma = 1e-1

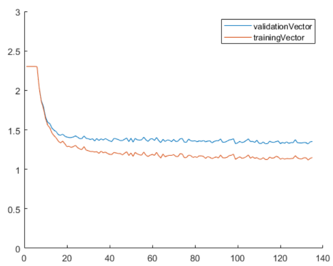

Figure 12: Loss without batch, sigma = 1e-3

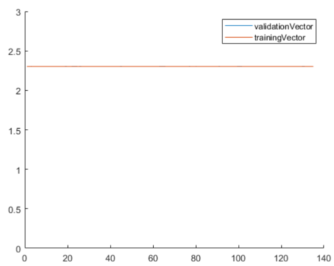

Figure 13: Loss without batch, sigma = 1e-4