Kungliga Tekniska Högskolan

DD2424 Deep Learning in Data Science

Kévin YERAMIAN

August 12, 2018

1 Introduction

In this assignment I will train and test a multiple layer network with multiple outputs to classify images from the CIFAR-10 dataset. You will train the network using mini-batch gradient descent applied to a cost function that computes the cross-entropy loss of the classifier applied to the labelled training data and an L2 regularization term on the weight matrix.

2 Gradient checking

2.1 Verification of the gradient for the batch normalization

In this part we will generalize the network to add the possibility of changing the number of hidden layer and add the batch normalization.

In order to be sure about the gradient, I compared the value with value of the others gradient function. I limited the dataset size at 100, to first see if it is working. We set h to 1e-6 and lambda to 0.

There is my result for the gradient without the batch normalization, each value is the error for each layer:

hidden layer	W error	b error
(3072, **20**, 10)	(1.6e-06, 1.6e-07)	(2.8e-07, 7e-07)
(3072, **20**, **20**, 10)	(6.4e-05, 6e-06, 1.3e-06)	(1.6e-05, 8.7e-07, 7.3e-07)
(3072, **20**, **20**, **10**, 10)	(0.001, 0.0001, 1.3e-05, 5.5e-07)	(0.001, 2.4e-05, 6.1e-07, 6.8e-07)

There is my result for the gradient with the batch normalization, each value is the error for each layer:

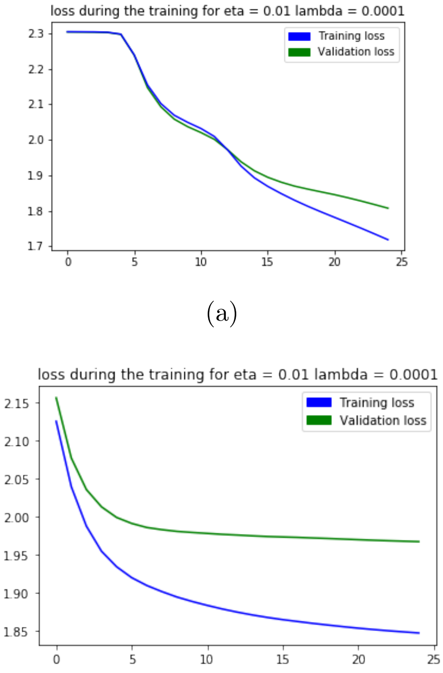
hidden layer	W error	b error
(3072, **20**, 10)	(0.00221, 6.68e-09)	(0.0049, 4.08e-09)
(3072, **20**, **20**, 10)	(0.00532, 0.00361, 6.054e-09)	(0.00827, 0.00497, 6.87e-09)
(3072, **20**, **20**, **10**, 10)	(0.0073, 0.0035, 0.0030, 6.663e-09)	(0.0105, 0.0083, 0.0049, 8.019e-09)

The result is good, we have close similarity. I assumed that the gradient is correct for the rest of the assignment. We notice that the discrepancy between the analytic and the numerical gradients increases for the early layers as the gradient is back-propagated through the network.

3 3-layers (50, 30)

3.1 3-layers (50, 30)

I took 9000 of the 10000 images of the trainset to train each model. I set batch to 100, decay-rate to 0.95, rho to 0.9, epochs to 50 and lamb to 0.0001.
After 25 epoch we have an accuracy of 33.21 without the normalization and 39.7 with the normalization. We still learning as we see with the loss curve:



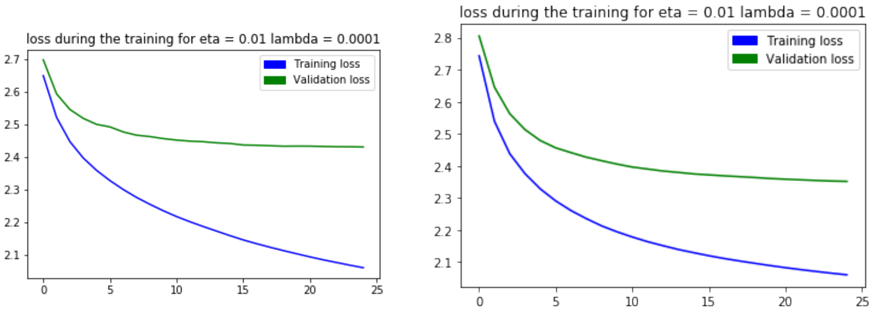
(a)

(b)

Figure 1: (a) without normalization, (b) with normalization

3.2 3-layers (50, 30) with He initialization

I took 9000 of the 10000 images of the trainset to train each model. I set batch to 100, decay-rate to 0.95, rho to 0.9, epochs to 50 and lamb to 0.0001.
After 25 epoch we have an accuracy of 34.44 without normalization and 47.07 with the normalization. We still learning as we see with the loss curve:



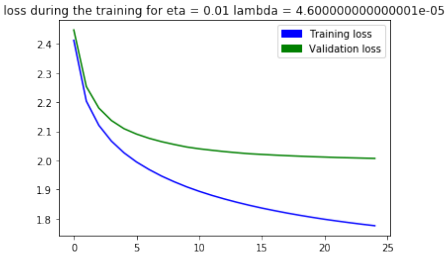
(a) (b)

Figure 2: (a) without normalization, (b) with normalization

4 Grid search

The ranges used for the fine search were [0.001;0.01] for eta and [0.00001;0.0001] for lambda. For my experiments I use 5 epochs and I tried 25 pairs of parameters.

My best model has an accuracy of 46.39.

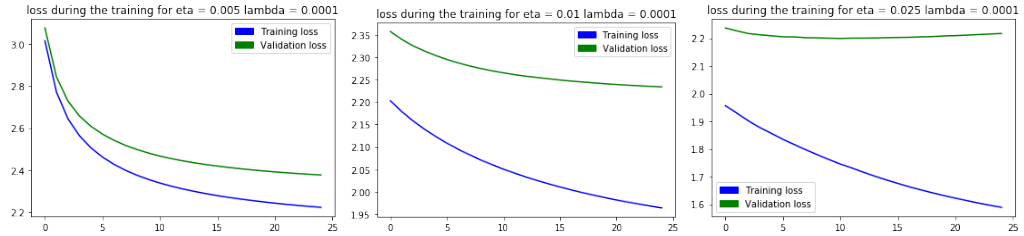


(a)

Figure 3: eta = 0.01 and lambda = 4.6e-05

5 2-layer network with different learning rate

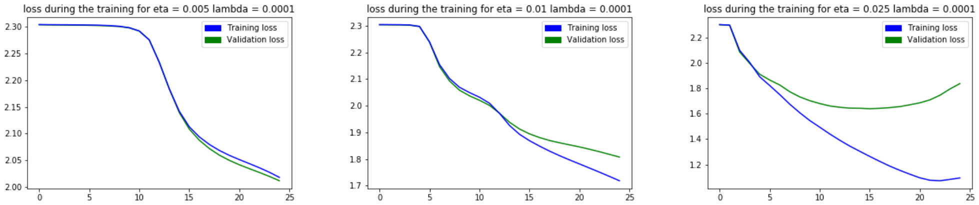
In this part, I trained 3 different networks with small, medium and high eta value. Lambda was set to 0.0001 and I trained the networks for 10 epochs. Here are my result with batch normalization.



(a) (b) (c)

Figure 4: 3 different learning rate for the normalization method

The accuracy for the 3 learning rates: 42.57, 51.27 and 62.37.



(a) (b) (c)

Figure 5: 3 different learning rate without the normalization method