Deep Learning in Data Science
dd2424

Solution to Assignment 3

Yiming Fan
yimingf@kth.se

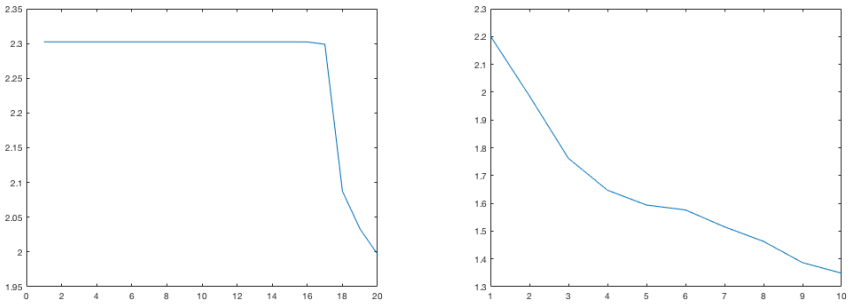

Figure 1: Cost functions with (right) and without (left) batch normalization.

1 Introduction

This assignment aims at training a k-layer network using *gradient descent* method. The dataset used in this assignment is CIFAR-10.

2 Methods & Mechanisms

The network consists of 3 sets of linear classifier $\mathbf{W}$ together with bias vector $\mathbf{b}$. The classifying functions are similar as in assignment 2 and the only addition is that we applied batch normalization:

$$\hat{\mathbf{s}}^{(l)} = \text{BatchNormalize}(\mathbf{s}^{(l)}, \mu^{(l)}, \mathbf{v}^{(l)}) = \left(\text{diag}(\mathbf{v}^{(l)} + \epsilon)\right)^{-\frac{1}{2}} \left(\mathbf{s}^{(l)} - \mu^{(l)}\right) \tag{1}$$

where ϵ is a small number to avoid divide-by-0 problem.

2.1 Gradient descent

At every layer the calculation of gradient descent is exactly the same as in Assignment 2 except for the backward pass for the batch normalization. We are convinced the calculations are correct since the 2-layer network provided around 44% test accuracy on default hyper-parameter settings.

2.2 The effect of batch normalization

We tried training a 3-layer network using default settings with/without batch normalization. Below are plots of the loss functions respectively. The network without batch normalization just start 'to learn' something after 17 epochs, while the other has a steadily decreasing cost function. **Conclusion:** Adding batch normalization will benefit the process of training 3-layer networks.

no.	η	λ	accuracy(%)
1	0.02785	$2.0309 \cdot 10^{-8}$	41.8
2	0.02	$2.0309 \cdot 10^{-8}$	41.57
3	0.02	$2 \cdot 10^{-7}$	41.2

Table 1: Top 3 networks and their respective test accuracy after 10 epochs.

2.3 Coarse and fine search for hyper-parameters

2.3.1 Coarse searches

We have conducted coarse search on hyper-parameters for training 3-layer networks. We tried λ range from 10^{-4} to 10^{-8}, η range from 10^{-2} to 10^{-6} and trained totally $14 \cdot 14 = 196$ combination of $\lambda - \eta$s. We trained each network by 10 epochs and the overall elapsed time was around 7 hours. We have drawn the following conclusions:

1. Since the network is stabilized due to batch normalization the disadvantage of decreasing learning rate is significant. Basically the test accuracy after 10 epochs decreases as the learning rates decrease.

2. The 'ideal' regularization rate λ is found at $\lambda = 2.0309 \cdot 10^{-8}$. The test accuracy is 41.57% when η is set to be 0.02. We will stop testing λ and use this value as the 'pivot' for fine searching.

2.3.2 Fine searches

Based on the prior knowledge that increasing learning rate will benefit learning, we tried fine searches on the learning rate η. We set 20 different values of η ranging from 0.02 to 0.1. The highest test accuracy 41.8% was achieved by setting $\eta = 0.02785$. By now we had our top 3 hyper-parameters:

2.4 2-layer network with(out) batch normalization

We tried the learning rate $\eta = 2 \cdot 10^{-4}, 2 \cdot 10^{-2}$ and $5 \cdot 10^{-2}$ respectively and kept every hyper-parameter else unchanged. The plots are shown on the next page. **Conclusions:** Adding batch normalization could increase the learning ability (the gradients descend faster than those network without batch normalization) and increase the stability (overfitting occurs later than those network without batch normalization).

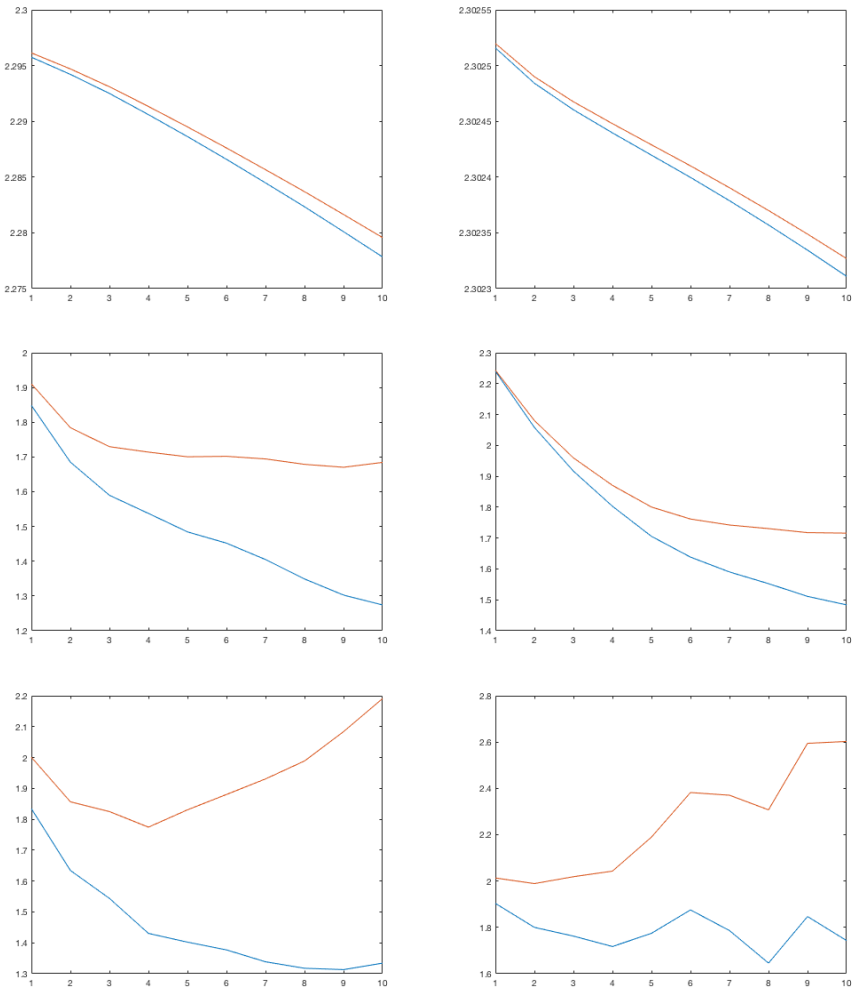

Figure 2: Cost functions with (left) and without (right) batch normalization.