Assignment 2

Hanlin Sun

April 18, 2018

In the experiments, some hyperparameter are fixed. We set decay_rate = 0.95, rho = 0.9, m = 50 and batch size = 100. When searching best hyperparameters, we set max training epoch = 20.

1 Correctness

We use the algorithm given in lecture to calculate gradients:

$$g = -(y-p)^T \tag{1}$$

$$\frac{\partial L}{\partial b_2} += g, \frac{\partial L}{\partial W_2} += g^T h^T \tag{2}$$

$$g = gW_2 \tag{3}$$

$$g = g\ \mathrm{diag}(\mathrm{Ind}(\mathrm{s}_1 > 0)) \tag{4}$$

$$\frac{\partial L}{\partial b_1} += g, \frac{\partial L}{\partial W_1} += g^T x^T \tag{5}$$

$$\frac{\partial L}{\partial W_i} /= |D^{(t)}|,\ \frac{\partial L}{\partial b_i} /= |D^{(t)}| \tag{6}$$

$$\frac{\partial J}{\partial W_i} = \frac{\partial L}{\partial W_i} + 2\lambda W_i,\ \frac{\partial J}{\partial b_i} = \frac{\partial L}{\partial b_i} + 2\lambda W_i \tag{7}$$

In order to confirm correctness of numerical computation, we compare the numerical outcome with given MATLAB code. We compute the relative error of these two gradients. We mainly test it on b1 and b2, and the relative error is less than 0.1%.

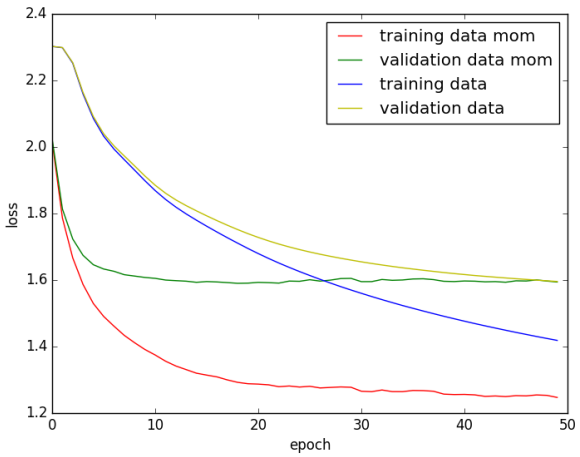

Figure 1: Loss function (without regularization)

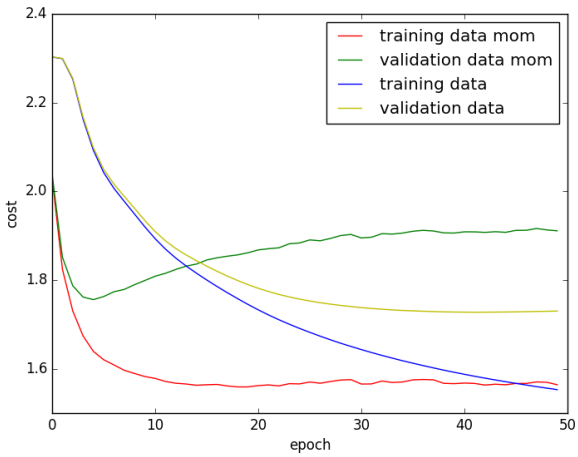

Figure 2: Cost function (with regularization)

2 Speedup effect of Momentum

The figures show the speedup effect of momentum. In this section, all other parameters are set as the same. "Training data mom" the cost or loss on training data and the network is training with momentum. In the figures we can find that when we use momentum, the cost and loss function convergence at approximately 15 epoches, otherwise the function will be convergent after 40 epoches.

3 Hyperparameter – Coarse Search

In coarse search, we use random lambda range from 0.0001 to 0.1 and random learning rate range from 0.001 to 0.5. After searching 50 steps, we pick up 3 hyperparameter from 3 best training network. Finally we pick 3 pairs of learning rate and lambda: lambda = 0.014, eta = 0.019; lambda = 0.015, eta = 0.02 and lambda = 0.023 eta = 0.03. Therefore we change the range of lambda and learning rate for fine search: lambda range from 0.005 to 0.02, learning rate from 0.01 to 0.03.

4 Hyperparameter – Fine Search

In the first fine search, we use lambda range from 0.005 to 0.02, learning rate from 0.01 to 0.03. we pick 3 pairs of learning rate and lambda: lambda = 0.005, eta = 0.028; lambda = 0.0055, eta = 0.028 and lambda = 0.006 eta = 0.015. Based on this data, we perform a second fine search:lambda range from 0.001 to 0.01, learning rate from 0.02 to 0.03. Finally the best outcome we get on validation is lambda =

0.0049, eta = 0.027 with the accuracy on validation data = 0.4536

5 Training-on-all

Using the optimal hyperparameters, lambda = 0.049 and eta = 0.028, we train the network using 49000 training data and 1000 validation data. Finally, we test the network on test data. The accuracy on test data is 0.5098.

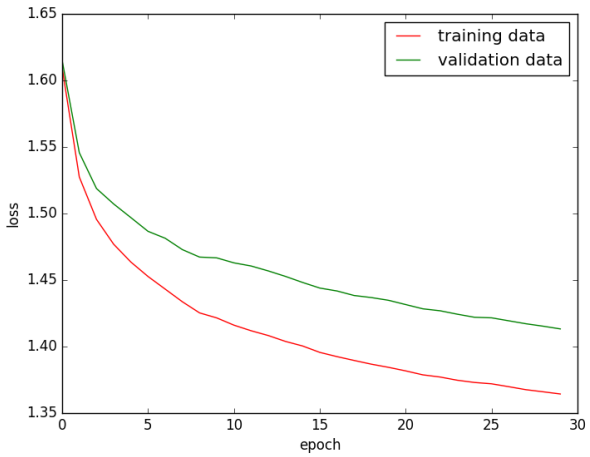

Figure 3: Loss function (without regularization)

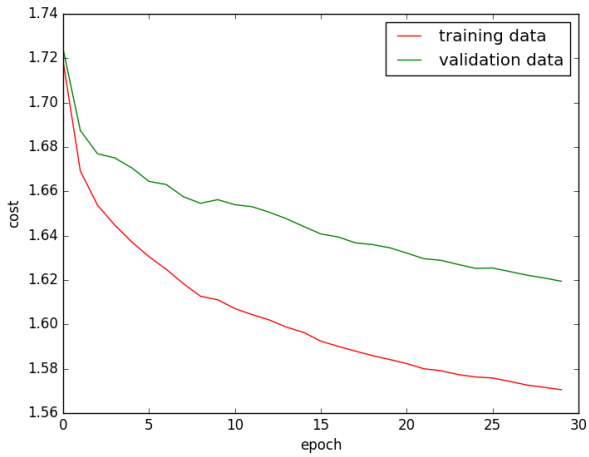

Figure 4: Cost function (with regularization)