Assignment 3 Report

Max Dauber (dauber@kth.se)

DD2424 Deep Learning in Data Science

Introduction

A k-layer neural network was implemented from scratch using Python. The neural network is trained using backpropagation and mini-batch gradient descent. As a data set CIFAR-10 is used, which consists of 32x32 images corresponding to 10 classes of objects. The model makes use of cyclical learning rate and batch normalization. The hyperparameter λ is optimized using course and fine search.

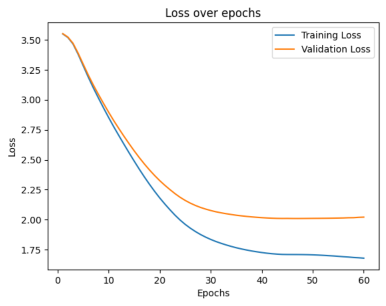
A k-layer neural network was implemented from scratch using Python. The neural network is trained using MiniBatch gradient descent and a cyclical learning rate and course/fine search methods to optimize the value for the regularization parameter λ. Batch normalization is also used to stabilize the training

Gradient Checking

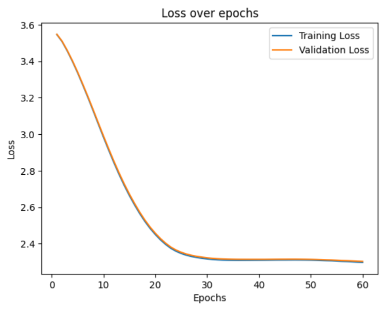
Testing is done in the `CheckGradients` method to compare the results from computing the gradients analytically using `ComputeGradients` and numerically using `ComputeGradientsNum` by using the finite difference method. This is after refactoring both functions to be compatible with the k-layer network in this assignment. The analytically commputed gradients were found to be within $\pm 10^{-5}$ of the numerically computed gradients, meaning they are likely to be bug free.

Loss Curves for 3 and 9 Layer Networks

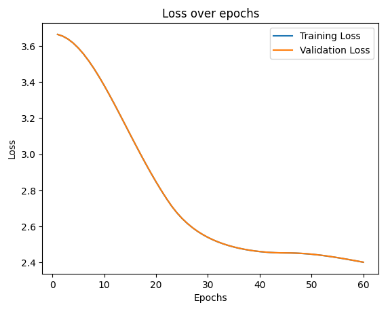
Figure 1 shows the loss curve for a 3-layer network with 50 nodes in both hidden layers, with and without batch normalization. The parameters were created using He initialization and the model was trained for 40 epochs/2 cycles (along with hyperparameters $\eta = 1^{-5}$ to 1^{-1}, lr = 0.01, $n_s = 5 * 45000 / \textit{batch}$, $\lambda = 0.005$) . Figure 1 also shows the loss curve for a 9 layer neural net with hidden layer sizes [50, 30, 20, 20, 10, 10, 10, 10] with He initialization and the same hyperparameter configuration. The results show that batch normalization has a positive effect on the loss, decreasing as the epochs increase.



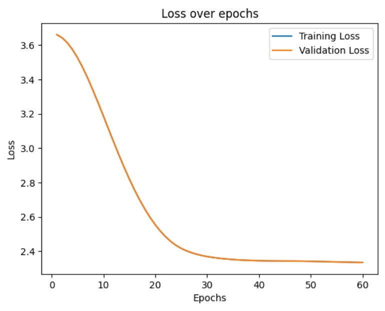
(a) 3-Layer Default



(b) 3-Layer Batch Norm



(c) 9-Layer Default



(d) 9-Layer Batch Norm

Figure 1: Loss Curves with and without Batch Normalization

Parameter Search (λ)

In the coarse search for the perfect λ, 5 values were tested in the range $[1e^{-5}, 1e^{-1}]$. Using these values, in conjunction with the other static parameters (batch = 100, $n_s = 800$, epochs = 20) the results displayed in Table 1 were obtained. The bolded results were the best performers overall.

λ	Train Accuracy	Validation Accuracy
1e-5	56.53%	52.40%
1e-4	56.57%	50.70%
1e-3	55.12%	50.40%
1e-2	47.54%	47.41%
1e-1	36.75%	37.70%

Table 1: Coarse Search Results

After finding the best values for the coarse search, the best values were used as an indication for the fine search range. Two fine passes were done over number ranges $[1e^{-5}, 1e^{-4}]$ and $[5e^{-6}, 1.5e^{-5}]$. Using these values, in conjunction with the other static parameters (batch = 100, $n_s = 800$, epochs = 20) the results displayed in Table 2 were obtained. The bolded results were the best performers overall.

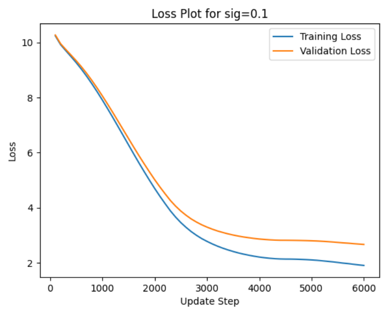
λ	Train Accuracy	Validation Accuracy	Test Accuracy
5e-6	56.59%	51.10%	50.04%
7.5e-6	56.66%	50.80%	50.85%
1e-5	56.53%	52.40%	49.66%
1.25e-5	56.38%	51.20%	49.49%
1.5e-5	56.25%	51.70%	50.18%
2.5e-5	56.57%	50.70%	50.72%
5e-5	56.41%	51.70%	49.97%
7.5e-5	56.43%	51.70%	49.58%
1e-4	56.33%	50.80%	50.08%

Table 2: Fine Search Results (First and Second Pass)

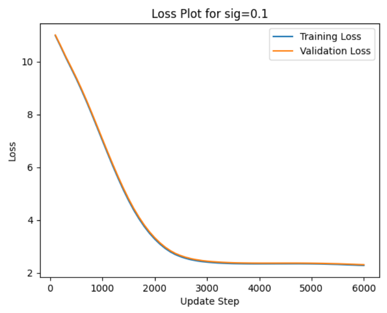
After finding an optimal λ values, the network was trained on all the training data for roughly 3 cycles with $\lambda = 7.5e^{-6}$. A training accuracy of **56.19%** and a test accuracy of **53.09%** were achieved at the end of the process.

Sensitivity to Initialization

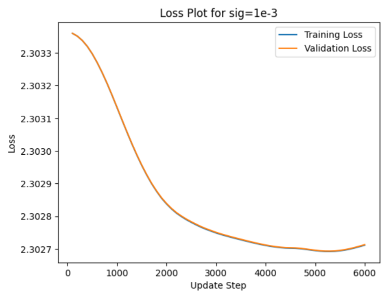
The results of the experiment to test the sensitivity of initialization with and without batch normalization can be seen in Figure 2. Six models were created, each with zero mean and a different initialization of the weights according to the given sig values, and each was tested with and without batch normalization. Each model was run for 60 epochs using the standard parameter settings as mentioned in the standard loss curve runs for 3-layer networks. It can be seen that the models with batch normalization are less sensitive to initialization.



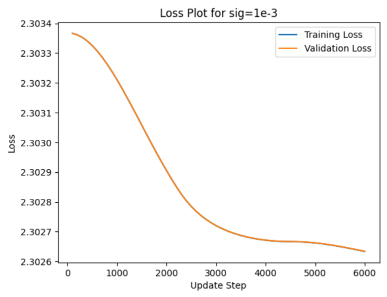
(a) sig=1e-1 Default



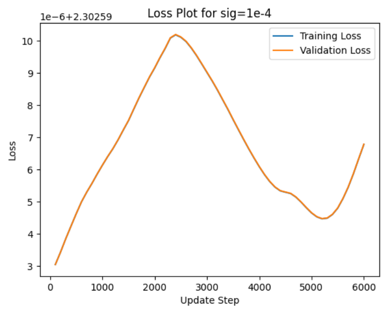
(b) sig=1e-1 Batch Norm



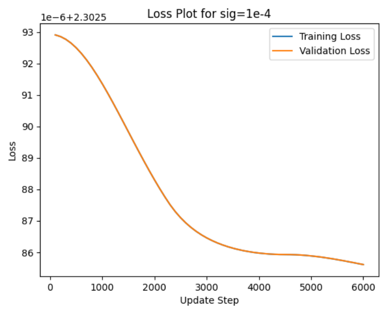
(c) sig=1e-3 Default



(d) sig=1e-3 Batch Norm



(e) sig=1e-4 Default



(f) sig=1e-4 Batch Norm

Figure 2: Sensitivity to Initialization (with and without BN)