Levent Güner
leventg@kth.se

Introduction

This assignment is about implementing k-layer(multi-layer) Neural Network. The two-layer neural network is improved according to k-layer network and batch normalization function added.

Data

CIFAR-10 dataset is used, which has 10 class labels representing different objects and 10,000 different 32x32 images per class.

Methods

There is n-layer neural network generated and tested with different configurations. Learning rate is adjusted by cyclical learning rate. λ is optimized using course and fine search method. Batch normalization option is added to normalize every batch and the data is shuffled in every epoch.

Cyclic learning rate method makes the learning rate changing between η_{min} and η_{max} over determined step number. During the learning process $\eta_{min} = 10^{-5}$ and $\eta_{max} = 10^{-1}$ values are taken.

Results

```
layer 0
-------
grad(W):      Mean absolute diff.: 3.43e-15
grad(b):      Mean absolute diff.: 8.71e-15
grad(gamma):  Mean absolute diff.: 9.88e-12
grad(beta):   Mean absolute diff.: 9.88e-12

layer 1
-------
grad(W):      Mean absolute diff.: 9.68e-12
grad(b):      Mean absolute diff.: 1.11e-11
```

Gradient results of 2-layer network

```
layer 0
-------
grad(W):      Mean absolute diff.: 3.28e-31
grad(b):      Mean absolute diff.: 7.56e-32
grad(gamma):  Mean absolute diff.: 4.11e-17
grad(beta):   Mean absolute diff.: 4.13e-17

layer 1
-------
grad(W):      Mean absolute diff.: 3.20e-18
grad(b):      Mean absolute diff.: 2.44e-17
grad(gamma):  Mean absolute diff.: 7.23e-12
grad(beta):   Mean absolute diff.: 7.88e-12

layer 2
-------
grad(W):      Mean absolute diff.: 7.85e-12
grad(b):      Mean absolute diff.: 1.08e-11
```

Gradient results of 3-layer network

Levent Güner
leventg@kth.se

```
layer 0
-------
grad(W):      Mean absolute diff.: 5.11e-12
grad(b):      Mean absolute diff.: 1.78e-12
grad(gamma):  Mean absolute diff.: 1.02e-11
grad(beta):   Mean absolute diff.: 1.15e-11

layer 1
-------
grad(W):      Mean absolute diff.: 3.27e-12
grad(b):      Mean absolute diff.: 1.78e-12
grad(gamma):  Mean absolute diff.: 1.15e-11
grad(beta):   Mean absolute diff.: 7.55e-12

layer 2
-------
grad(W):      Mean absolute diff.: 6.93e-13
grad(b):      Mean absolute diff.: 8.88e-13
grad(gamma):  Mean absolute diff.: 8.86e-12
grad(beta):   Mean absolute diff.: 7.74e-12

layer 3
-------
grad(W):      Mean absolute diff.: 8.35e-12
grad(b):      Mean absolute diff.: 8.83e-12
```

Gradient results of 4-layer network

Table 1: Gradient results

I have checked the gradients with Centered Difference Formula and got accurate results can be seen in Table 1.

3-Layer Network

Without batch normalization

With batch normalization

Figure 1: Costs over update with 3-layer network

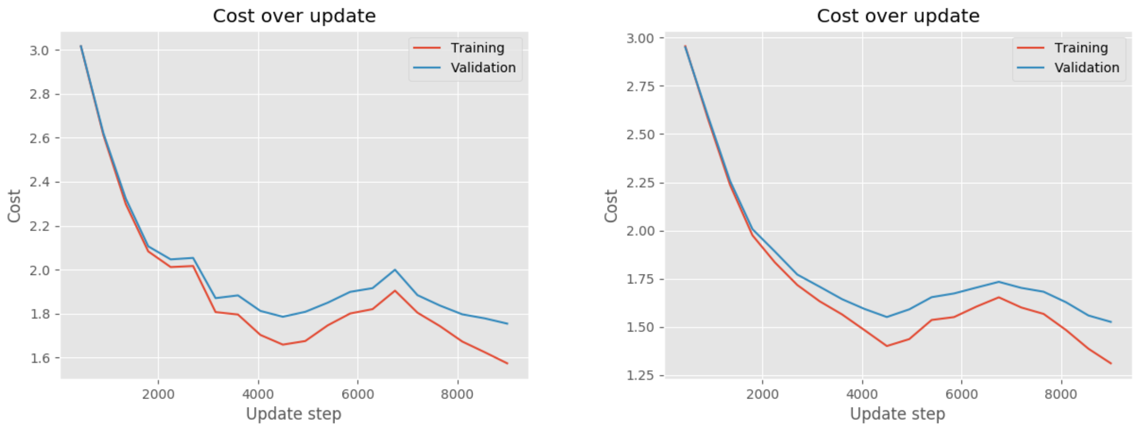
Figure 1 shows the cost results for each update. The learning cycle is adjusted to 2 and η is changing between 10^{-5} and 10^{-1}. λ=0.005.

	Without BN	With BN
Validation accuracy	0.5348	0.5358
Test accuracy	0.5145	0.5219

Table 2: Accuracy results for 3-layer Network

Levent Güner
leventg@kth.se

9-Layer Network



Without batch normalization   With batch normalization

Figure 2: Costs over update with 9-layer network

Figure 2 shows the cost results for each update on a 9-layer network both with and without batch normalization. The learning cycle is adjusted to 2 and η is changing between 10^{-5} and 10^{-1}. λ=0.005.

	Without BN	With BN
Validation accuracy	0.4638	0.5072
Test accuracy	0.4563	0.505

Table 3: Accuracy results for 9-layer Network

Table 3 shows the accuracy results for 9-layer network with and without batch normalization. It is obvious that the accuracy has increased when batch normalization is applied.

Course and Fine Search

λ	Accuracy
1e-1	0.4778
1e-2	0.5226
1e-3	0.5126
1e-4	0.5208
1e-5	0.5198

Table 3: Course search

λ	Accuracy
1e-3	0.5196
2e-3	0.5214
3e-3	0.5226
4e-3	0.5198
5e-3	0.5232
6e-3	0.5328
7e-3	**0.5356**
8e-3	0.5274
9e-3	0.5258
1e-2	0.5302

Table 4: Fine search

Levent Güner
leventg@kth.se

I found the best course search with 1e-2 and done the fine search between 1e-2 and 1e-3. Eventually, the best λ value is 7e-3.

Sensitivity to Initialization

σ	Without BN	With BN
1e-1	0.5155	0.5139
1e-3	0.4761	0.5185
1e-4	0.1	0.5129

Table 5: Constant σ results

Table 5 shows the results when a constant σ value is applied to all layers instead of Xavier/He initialization. While there is no batch normalization, the scores drop quite low. Even with batch normalization, the results are not as good as the previous ones. For those experiments, default values $\eta = [10^{-5}, 10^{-1}]$ and λ=0.005 is used with 2 cycles of learning rate and 3-layer neural network.

Conclusion and Discussions

In this assignment, there is a k-layer neural network model trained with different optimizations, as well as batch normalization. In comparison to single and double-layer models, the model gave better results.

While the number of layers are increased, the model started to give less accurate results without batch normalization. In this case, batch normalization is applied and λ is adjusted according to give the best results. The final result was λ=7e-3. When the 9-layer network trained with this λ, it end up with 0.5102 accuracy.

To sum up, layer number is an important factor on neural network models. However, this does not mean that the more layers give better results. Finding the optimal value with correct hyper parameter values gives us the best results.