Assignment 4

Mukul Sati [msati3@gatech.edu]

April 29, 2016

I used the CAFFE framework [1] for my implementation.

1 MNIST Dataset

I trained two CNN architectures on the MNIST dataset (Fig. 1). For the first architecture, I have a baseline training loss of 2.41 and a baseline test accuracy of 0.1167 and I obtain a final accuracy of 99.03%. Getting good numbers on the first architecture, I wanted to simplify the second architecture. Thus, the second architecture is essentially the first architecture with XXa. For the second architecture, I have a baseline training loss of XX and a baseline test accuracy of XX and obtain a final accuracy of XX. The training progress is shown in Fig. 2. As I did not manually tweak hyper-parameters, I feel it was sufficient have directly using the provided test set during the testing phase instead of performing hold out validation. The kernels learned from the first and second convolutional layers are visualized in Fig. 3. The confusion matrices of the two architectures are shown in Fig. 4, and a few incorrectly classified images are shown in Fig. 5.

The following are the gradient descent equations:

2 Sunset Dataset

I used the CaffeNet pre-trained network that comes with CAFFE. This is pretty similar to AlexNet, but without the relighting data-augmentation and has a difference in the order of the pooling and normalization layers. The primary tweaks I made to CaffeNet are:

1. Editing the last fully connected layer for the binary classification problem at hand. The initial architecture was for the 1000 class ImageNet database.

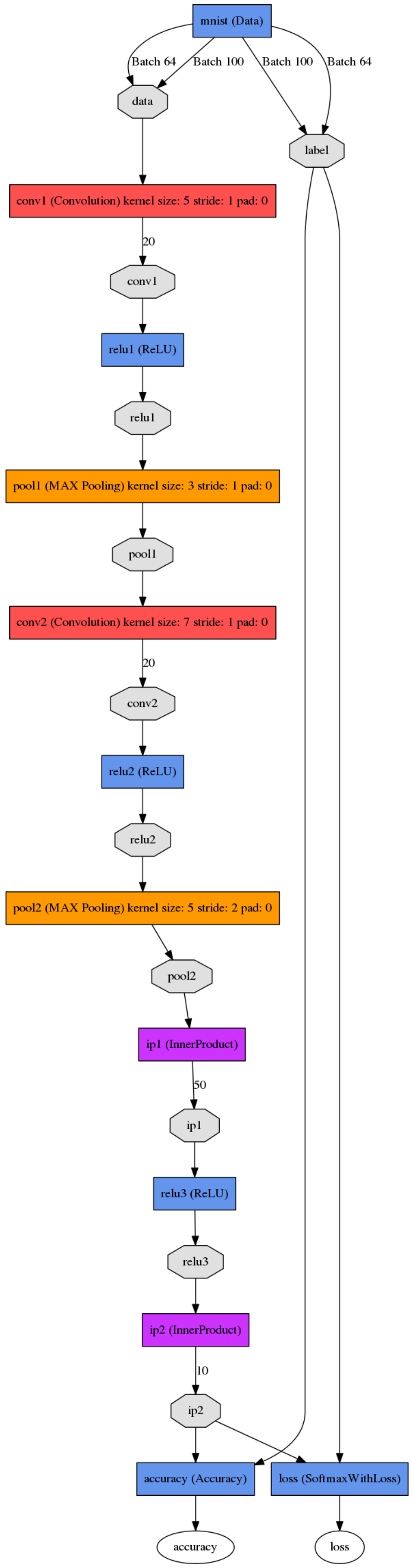
2. Lowering the learning rate of the solver, while using a higher multiplier for the weights of the modified layer.

Using a very small number of training iterations, I quickly arrive to an accuracy of about 89% (Fig. 6). After a longer duration of training, the rate stabilizes at $XX\%$

I use the following schemes for data-augmentation

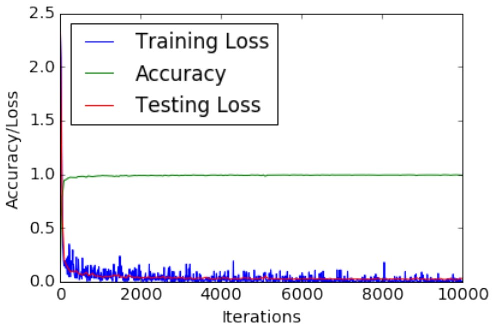
References

[1] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.



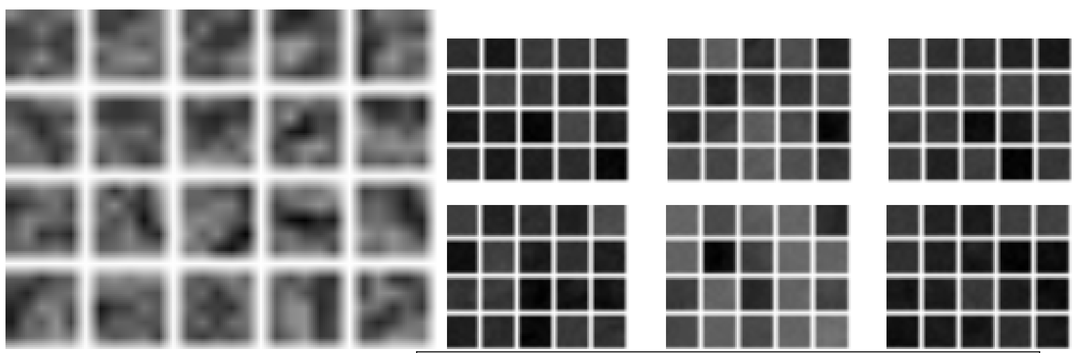
images/mnist_arch2.png

Figure 1: The two architectures trained for the MNIST dataset.



images/mnist_learning2.png

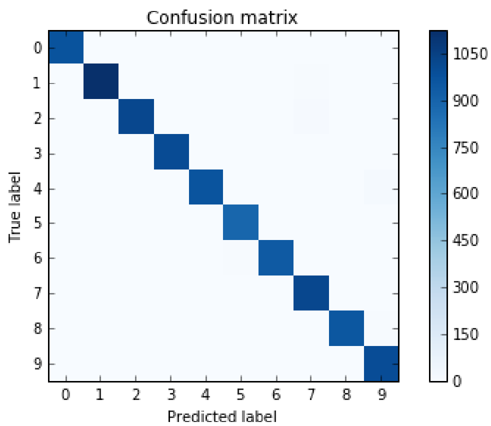
Figure 2: Training loss and testing loss and accuracy versus iterations for architecture 1 (left) and architecture 2 (right).



images/mnist_kernels21.png

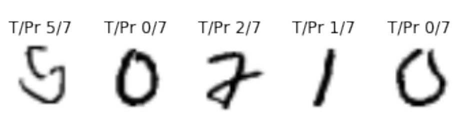
images/mnist_kernels22.png

Figure 3: The learned filters for the first and some of the learned filters for the second convolutional layers of the first (top row) and second (bottom row) architectures.



images/mnist_confusion2.png

Figure 4: Confusion matrix for the first (left) and second (right) architectures.



images/mnist_incorrect2.png

Figure 5: Some incorrectly classified test examples for the first (left) and second (right) architectures, showing also the true (T) and predicted (Pr) labels.

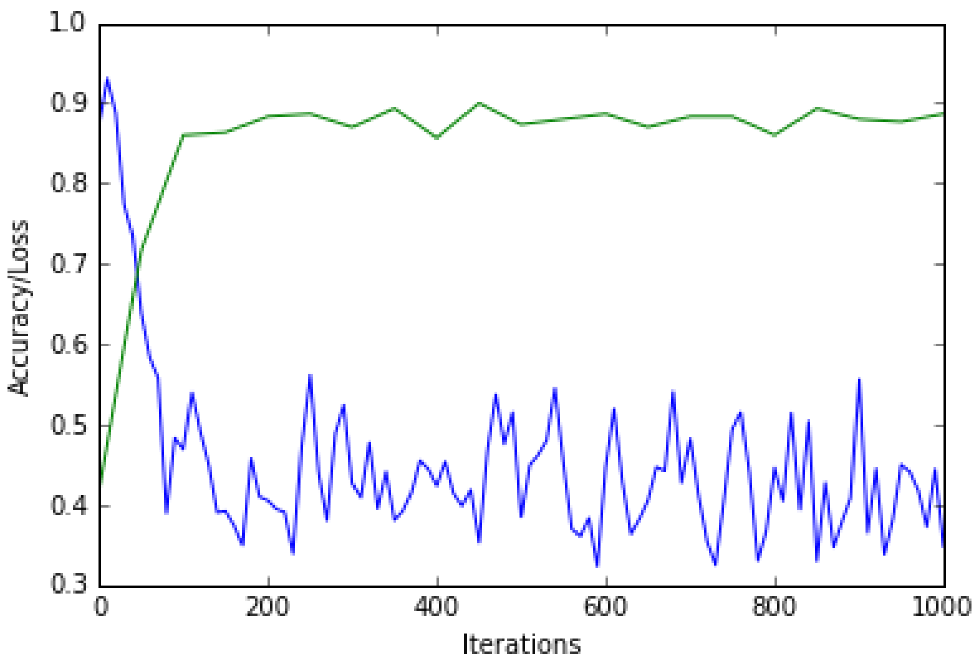


Figure 6: Training loss and testing loss and accuracy versus iterations for the vanilla CaffeNet described above.