CS 466/566 Assignment 1 Report

Yasin Serdar Özkanca S001793

Network Designs

Network Layout	
Network 1	***Network 2***
conv 5x5x32-s-1	*conv 3x3x16-s-1*
ReLu	*ReLu*
maxpool 4x4-s-2	*maxpool 2x2-s-2*
conv 5x5x64-s-1	*conv 3x3x32-s-1*
ReLu	*ReLu*
maxpool 4x4-s-2	*maxpool 2x2-s-1*
Fc-dropout	*Fc-dropout*
fc	*fc*

So first, for the network layout we use convolutional layer. Convolutional layer will create features for us for each window size we defined. For the first network its 5x5 for the second its 3x3 for instance. Then we will use Rectified Linear unit as a activation function. After ReLu it is time to use max pooling. Max pooling will down sample our image to scale that we adjusted. After this step we apply the same process again. At the 7^{th} layer we use dropout to avoid overfitting. Overfitting simply means memorizing the training data so it affects our test results in a negative way. At the last layer we are calculating the results we found and then the training part begins.

Training Description

For training first, I used cross entropy to reduce the mean then for minimizing error I used Adam Optimizer instead of gradient descent. After few tries with the parameter, I found the best parameter for the Adam Optimizer is 0.001. The main difference between Adam optimizer and gradient descent is the momentum parameter. This enables Adam to use a larger effective step size, and the algorithm will converge to this step size without fine-tuning.

I used 50 mini batch size for the training and I found the optimal iteration is 12 for the process. After 12 test accuracy drops probably because of the overfitting. So to avoid that I tried different numbers for iteration. Best results I got with 12.

For the fully connected part, I used dropout method to prevent over-fitting.

Data Augmentation

For data augmentation part, I tried three different augmentation techniques;

- *Rotation*
- *Affine transformation*
- *Shifting*

Therefore, for each augmentation we created 9000 new training samples, which increases our training samples for the sake of learning.

Rotation is basically rotating the images for 10 degrees not more than that because we don't want to get nines from a six.

Affine transform makes the image's shape widened or narrowed. And shifting is basically makes a new sample for us by changing the location of the number on the image so that is important.

Experimental Results

Show your experimental results.

Experimental Results	*Performance in %*	
Data Set	***Network 1***	***Network 2***
28x28_dataset	*98.5*	*97.7*
14x14_dataset	*95.8*	*97*
14x14_augmented_dataset	*93.8*	*97.2*

Discussion

As we can clearly see from the results for down sampled data network two works better. I guess that is because for network 2 I used 3x3 window convolution. 5X5 convolution probably missed some data. And also for max pooling in network 2 I used 2x2 size windows so that is another reason why network 2 worked better for 14x14 images. However, for 28x28 images definitely network 1 works better. In addition, data augmenting increased the success rate of our network for network 2. However, for network 1 it did not work well. I guess it is the same reason why network one work better for 28x28 data set.

In general, I increased the success rate with increasing the iteration and the sample size of our data set. The important thing is to increasing the success rate by changing the parameters of the neural network.

Note: I am not familiar with the Python language. I am actually taking the course that teaches Python to learn. That is why I could not manage to make the codes in classes so I zipped them in one folder one by one. It also automatically downloads the mnist data set to the current folder. I am sorry for the inconvenience.