Table 1: Network with 25 Hidden Nodes Improvement over 50 Epochs

Epoch	Batch	Train Accuracy	Test Accuracy
1	10	51.2	50.0
13	10	61.2	50.0
19	10	91.0	61.2
50	10	98.2	61.4

HW1

Class: CS519 - Deep Learning
Name: Ammar Kothari
Date: 02/13/17

Forward and Gradient Equations

Forward

$x_b = x.append(b)$
$L_1 = W_1 * x_b$
$NL_1 = max(L_1, 0)$
$NL_{1b} = NL_1.append(b)$
$L_2 = W_2 * NL_{1b}$
$NL_2 = 1/(1 + exp^{-L_1})$
$L = Y * log(NL_2) + (1 - Y) * log(1 - NL_2)$

Gradient

$\delta_{out} = (Y - NL_2)$
$\partial L/\partial W_2 = \delta_{out} * NL_{1b}$
$\partial L/\partial W_1 = min(1, max(0, NL_1)) * W_2 * \delta_{out} * x_b$

Stochastic Mini-batch Gradient Descent

```
1. Randomize order of test set
    while test samples remain:
        for 1 set of examples in minibatch:
            2. Run one set of test samples through network forward
            3. accumulate $\partial L/\partial W_2$ and $\partial L/\partial W_1$
        4. determine step size
        5. Update weights
```

Results

I trained several networks using between 100 and 500 samples from the CIFAR set. However, I still tested against the full test set. This saved significant computation time. Figure 1 is a small 5 node network that is unable to learn much. Although it does improve, it quickly plateaus at basically randomly guessing. Figure 2 shows worse performance. This is likely due to the fact that it is only looped through the data twice. Given minimal learning time, the network is unable to learn anything meaningful. The test accuracy for both of these is 50%.

The performance of the same network (25 Hidden Units) over many more epochs creates significantly better results. This network was trained on only 500 samples (5% of the total data) yet was able to achieve a test accuracy of 61.35%. The success of this network can be seen in Figure 3.

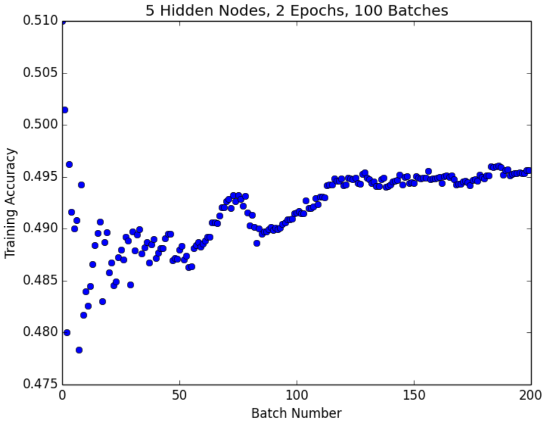

Figure 1: Accuracy on Training set for a small network over only 2 Epochs

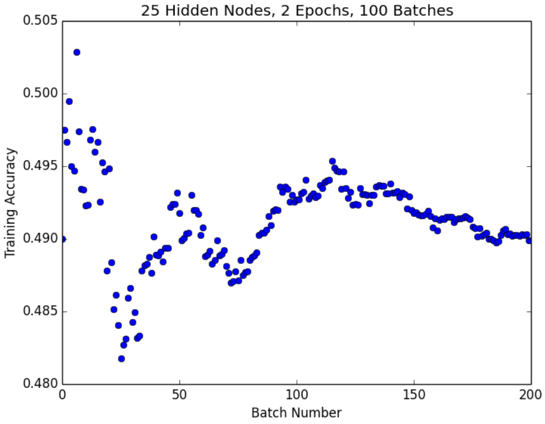

Figure 2: Accuracy on Training set for a moderate network over only 2 Epochs

Discussion About Neural Network Quality

This networks does not work very well. Although it is eventually capable of classifying the images with some accuracy and consistency, it is not capable of classification outside of these two classes. It also does not generalize very well.

This type of network does not take advantage of relationship amongst the data. A convolutional network is able to exploit the spatial relationship in images to extract more interesting features and as a result is more powerful.

As expected, increasing the number of hidden nodes improves the ability of the network. 5 is an exceptionally low number of hidden nodes. However, it is worth testing with such a limited number to ensure that a simple network cannot solve the problem. Increasing the number of hidden nodes to 25 shows significant improvement.

Lack of Results

I ran into some issues with creating the network. I wrote the entire code twice thinking I had missed something. In the end, my computer was so slow and the computation required was substantial enough to make it seem like the network was not learning.

I have verified that my network is learning since I have evaluated on several simpler sets of data with

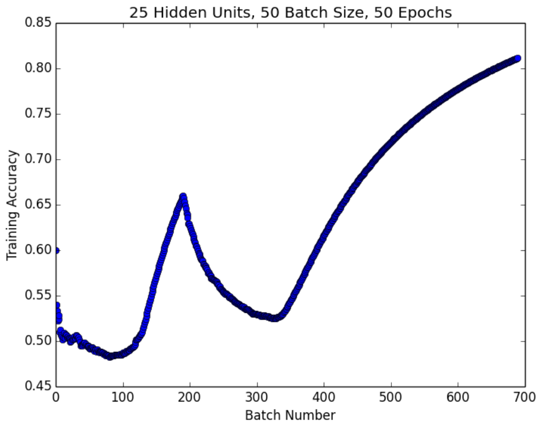

Figure 3: Accuracy on Training set for a moderate network over only 50 Epochs

success. I am turning in a report and whatever data I have available. I will continue to run the test cases and submit a more complete assignment prior to next Monday.