CSCI 5561: Computer Vision, Homework 4 Submission

Ashvin Pidaparti, 5421833

Three methods of deep learning were implemented to classify the MNIST dataset (handwritten, single digit numbers). Mini batches of size 32 were constructed to streamline the training process. The first method implemented used a single fully connected layer with a linear loss function. An accuracy of 67.9% was achieved, and the confusion matrix of this is shown in Figure 1. Learning and decay rates were set to 10^-2 and .99 respectively.

The second method implemented used a single fully connected layer with a cross entropy loss function using softmax. Due to computing limitations and the magnitude of the predictions, computing the softmax led to several overflow errors, and to prevent this, some basic algebra was implemented. The maximum of the predictions was found, and for each computation of exp(xi)/SUM(exp(xi)), the numerator and denominator were multiplied by exp(x_maximum). This is algebraically correct (exp(x_maximum)/exp(x_maximum) = 1), and resulted in no overflow errors. An accuracy of 86.5% was achieved and the confusion matrix is shown in Figure 2. The learning and decay rates were set to .1 and .999 respectively.

The third method implemented used a multilayer perceptron with 2 fully connected layers with a ReLU activation unit in between with the same cross entropy softmax used in the second method. The confusion matrix of this is shown in Figure 3. Unfortunately, the accuracy target of 90% could not be reached, and instead reached approximately 86%. Several configurations of iterations, learning rates, and decay rates were used. A maximum of 10,000 iterations were used and the loss curve was shown to converge (Figure 4). Learning rates spanning from 10^-5 to 10^-1 were used. .5, .8, .9. .95, and .99 were used for decay rate. The optimum was seen to be with 10^-2 and .8 for the values of the learning and decay rates respectively.

The fourth method implemented used a convolutional neural network, maximum pooling, ReLU, and a fully connected layer. To optimize the convolution operation with respect to time, the input image was converted to columns in an array, with each column being the flattened length of a single depth of the filter and containing the values that would be multiplied by the values in the filter. This vastly improved the speed of the operation. This method was provided by Leonard Araujo Santos and was sent out in official communications via the course, with an inline citation. The accuracy of this method was poor, approximately 20%, despite the teaching assistant’s assurances that the code was correct. The confusion matrix is shown in Figure 5. The loss curve converges, however, the accuracy did not reach the target accuracy of 92%. The loss curve is shown in Figure 6. The learning and decay rates were set to 10^-2 and .99 respectively.

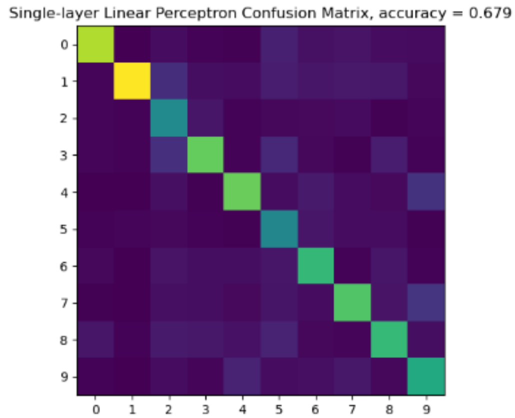

Figure 1. Confusion matrix of Single Layer Perceptron with Linear loss function.

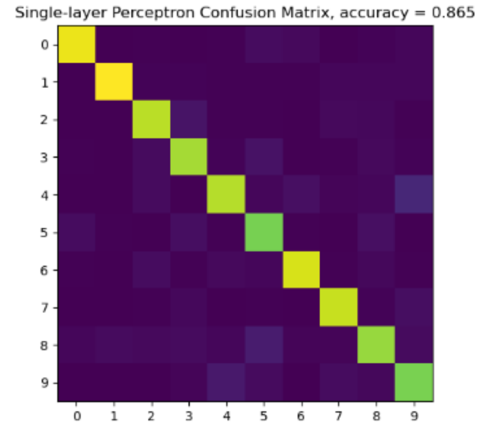

Figure 2. Confusion matrix of Single Layer Perceptron with cross entropy softmax loss function.

Figure 3. Confusion matrix of MultiLayer Perceptron with cross entropy softmax loss function.

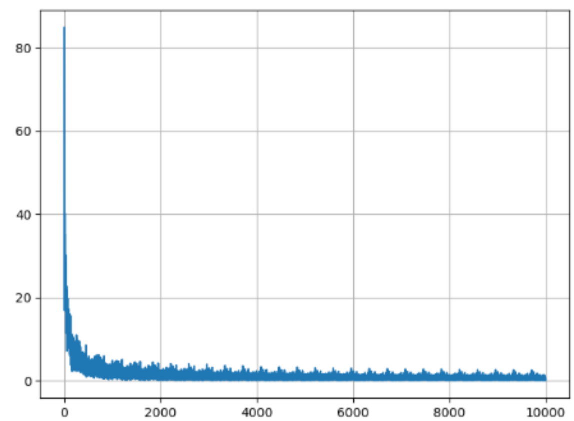

Figure 4. Loss curve of MultiLayer Perceptron with cross entropy softmax loss function.

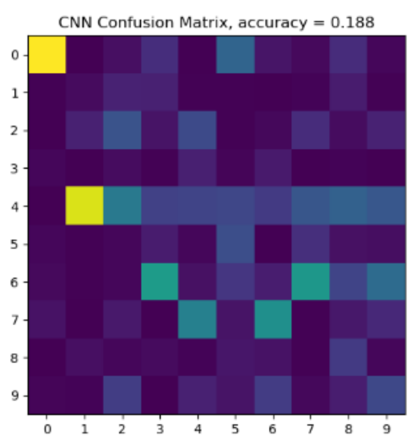

Figure 5. Confusion matrix of Convolutional Neural Network.

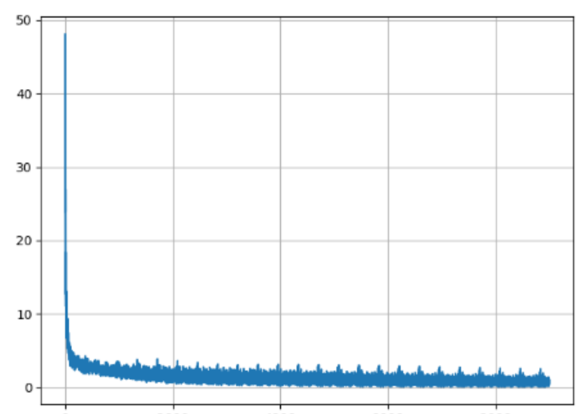

Figure 6. Loss curve of Convolutional Neural Network.