

w251_homework04_MNIST_Bona_Lee

February 26, 2020

2. ConvnetJS MNIST demo

In this lab, we will look at the processing of the MNIST data set using ConvnetJS. This demo uses this page: <http://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html> The MNIST data set consists of 28x28 black and white images of hand written digits and the goal is to correctly classify them. Once you load the page, the network starts running and you can see the loss and predictions change in real time. Try the following:

- Name all the layers in the network, make sure you understand what they do.
 1. **Input layer:** It holds the raw pixel values of the image.
 2. **2 pairs of Convolutional/ReLU layer and Pooling layer:**
 - **Convolutional/ReLU layer:** Its function is to extract the high-level features such as edges, from the input image.
 - **Pooling layer:** Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. It is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.
 3. **Fully-Connected layer:** Its function is to learn non-linear combinations of the high-level features as represented by the output of the previous layer. It classifies the image using the Softmax Classification technique.
- Experiment with the number and size of filters in each layer. Does it improve the accuracy?
 - Smaller filter sizes result in a higher accuracy because with smaller filter sizes, highly local features are extracted without much image overview; therefore, it captures the basic components in the image. Also, slow reduction in the image dimension can make the network deep.
- Remove the pooling layers. Does it impact the accuracy?
 - I expected that pooling layer removal would result in a high training accuracy and a low validation accuracy but didn't observe any significant change in the accuracy after removing the pooling layers. However, removing the pooling layers increased the time it takes to train and validate the same amount of data, compared to having the pooling layers.
- Add one more conv layer. Does it help with accuracy?
 - With a small number examples, training accuracy tended to be high, and validation accuracy tended to be low. In other words, it resulted in overfitting. However, as the number of examples seen increased, these numbers were flipped - training accuracy being lower than validation accuracy.
- Increase the batch size. What impact does it have?
 - Large batch size resulted in very large gradient updates and very small gradient updates.

The size of the update depends heavily on which particular samples are drawn from the dataset. On the other hand, using small batch size had almost constant updates.

- What is the best accuracy you can achieve? Are you over 99%? 99.5%?
 - An increase in the size of dataset (examples seen) increased the training accuracy to about 0.98 and the validation accuracy to about 0.95.