Eli Gabay

Kyle Crowley

CS 491Q Project

5/18/15

This project is based on the kaggle competition titled, “First steps with Julia”. The Julia dataset contains 6,283 20x20 images for training. Because it is a kaggle competition, the test set targets are not given. Thus, the training data needs to be split into 5000 training images and 1283 testing images for supervised learning.

Once images are uploaded, they are converted from RGB to YUV to use luminance information. Then, the model for classification is created. For this project, the model used is a Convolutional Network (ConvNet), which processes image data through a neural network after passing through a convolution layer, a sub-sampling layer, another convolution layer, and another sub-sampling layer. In order to implement this, a filter bank layer, non-linearity layer, and a feature pooling layer is used.

After defining a loss, train, and test function, the model is ready to be used for classification. Initially, the project utilized the tanh() activation function. However, after experimenting with both PReLU and ReLU it was found that ReLU provided the best accuracy. The neural net was initially run with a learning rate of 1e-3 and a learning rate decay of 1e-7, as suggested by the Torch website. Running this code with ReLU returned about 6% accuracy. After experimenting with different values for these parameters, it was found that a learning rate of 0.1 increased performance significantly. The accuracy increased to around 12% in this case. When combined with a momentum of 0.9 as well as adding Spatial Contrastive Normalization, the accuracy of the neural net jumped to around 60%. Another change attempted was with the batch size. Initially a batch size of 64 was used, attempts to lower or raise it provided no significant improvements.

The performance of the ConvNet is significantly better than expected. Training MNist uses a dataset containing over 100,000 images to classify into only 10 classes. Training Julia uses a dataset containing only about 6,000 images to classify into 62 classes (lowercase alphabet, uppercase alphabet, and ten digits). Using a smaller dataset and a larger amount of classes both work against the accuracy of the neural network. Furthermore, there are many more similar symbols in 62 classes than in only 10 classes. These similar symbols can result in a decrease in accuracy because the neural network might not find much of a difference between them.

In conclusion, the Julia dataset was used to create a ConvNet that will classify images of characters to their actual character value. The accuracy of the neural network started at around 6%. Extensive experimentation led ultimately to 60% accuracy.