CS 4780 Kaggle Competition Report

Team name: EXCITED!
Team member: Ayon Borthakur (ab2535) , Ling Zeng (lz455) , Yi Chen (yc2329) , Yuan Xiong (yx292)

Our team name is EXCITED, and we achieved 100% on public leaderboard.
This project is about classifying digits with different fonts, and the datasets is similar to MNIST datasets. We are given a large number of handwritten digits as training examples and need to predict new incoming testing examples.
Preprocessing: We split the data into training (3600 samples) and validation parts(400 samples). We rescaled the input pixel values by 255 units and reshaped the image to width(28, height(28) and pixel (1). We also randomly shuffled the train data for each train - test run.
Model description: Since images have higher order pixel correlations, we implemented Convolutional Neural Networks (CNNs / ConvNets) algorithm which are proven to support translational invariance. CNN has 3 architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer. Through a differentiable function, the 3-D input is transformed into 3-D output. We used 'Keras' with 'Tensorflow' backend combined with sklearn for the project.
Layer 1: The first hidden layer has 30 feature maps of size 5 x 5 and activation function is 'ReLu'.
Layer 2: Does maximum pooling from 2x2 units.
Layer 3: 15 feature map convo-net of size 3 x 3 with activation function 'ReLu'.
Layer 4: Does maximum pooling from 2x2 units.
Layer 5: It's a dropout layer which makes the network robust to noise by neglecting the activation of 20 % units. Dropout worked better than 'batch normalization' here.
Layer 6: Flattens the 2D data for feeding it to the subsequent hidden layers.
Layer 7, 8 and 9 are all to all connected layers with 128, 50 and 10 (num classes) units respectively. Layer 9, the output has a 'softmax' activation function.

We trained the network with batch sizes of 200 and 400 epochs. Since, the task was to classify digits with different fonts, we also used data augmentation techniques in 'Keras' : essentially varying the 'zoom_range' , 'horizontal_shift', and 'Vertical shift' of the images using 'ImageDataGenerator'.

In the class, Killian taught us that CNN is a stochastic gradient descent, and it outputs differently every time. Thus, running the same model for several times and comparing the output, we can improve the output by getting the prediction mode of each digit. I first tried voting classifier and found that I cannot using multiple input features and multiple output classes. Therefore I wrote a function to get the mode of the output. And our accuracy improved from 0.9825 to 0.9875 using 3 models. After using data augmentation, we got 100% accuracy.