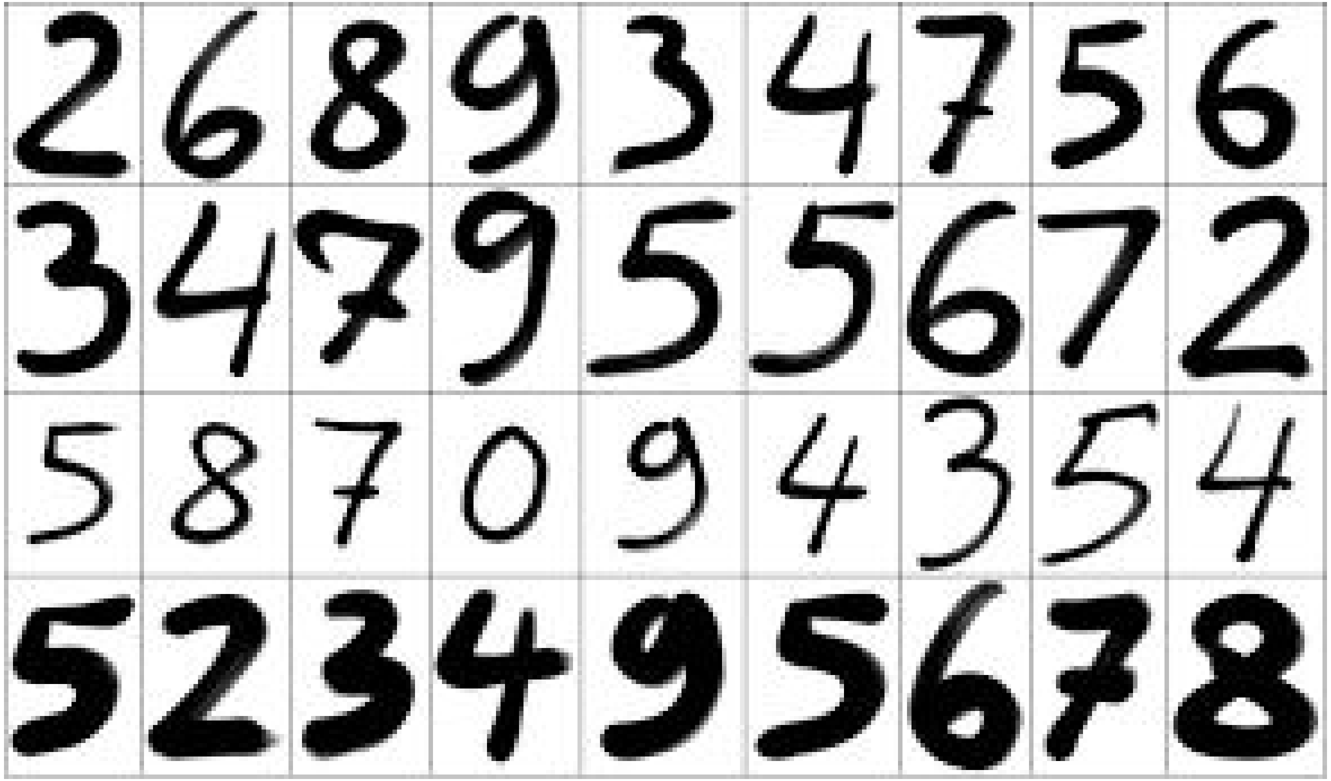
Convolution Neural Network on Handwritten Digit Recognition

Xiangguang Zheng (Sherwood)

handwritten digit recognition(MNIST data set)

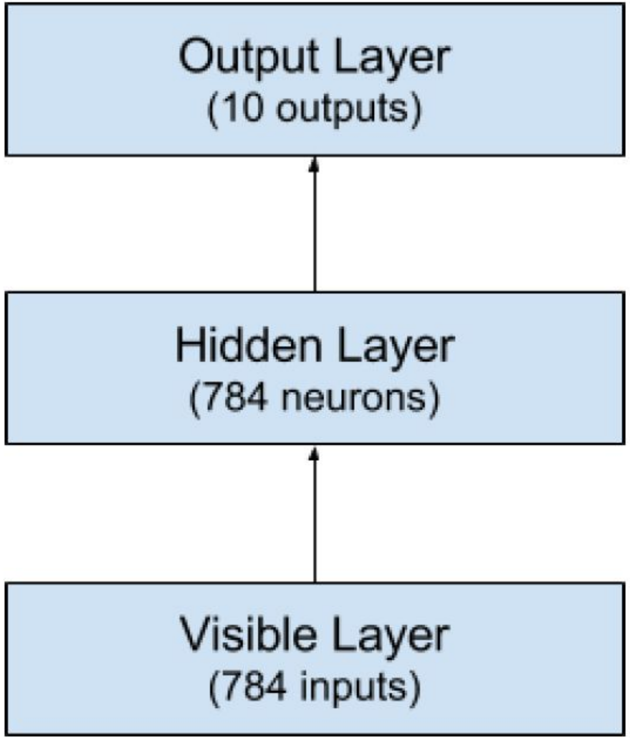


Experiment Environment

- Keras using TensorFlow backend with only CPU support
    - Keras is is a high-level neural networks library, written in Python and capable of running on top of either TensorFlow or Theano

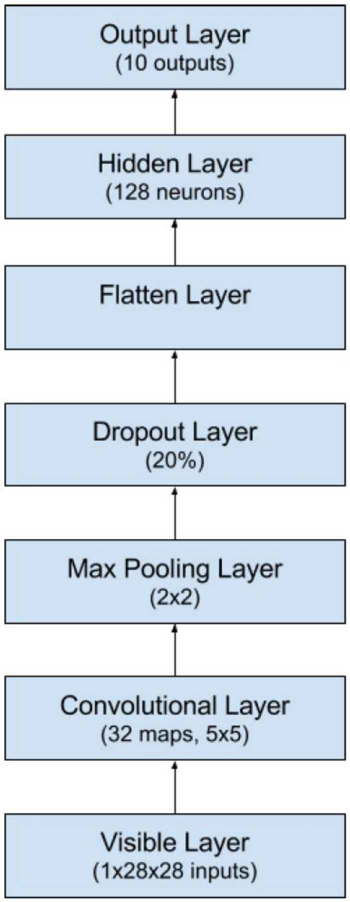
Generic neural network

- using perceptron neural network with one hidden layer
    - 28 * 28 size image = 784 pixel input
    - input are gray scaled
    - 784 neurons in hidden layer
    - batch size of 200 with 10 epoches



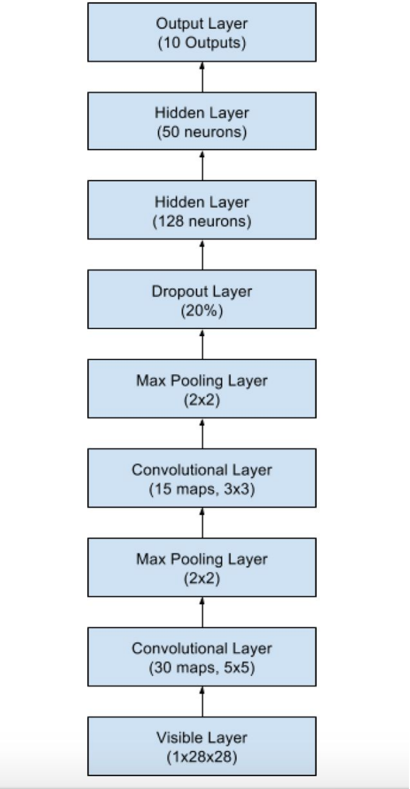
Basic CNN

- Basic CNN with one convolution layer and one fully connected layer
    - input size 28 * 28 * 1 with only 1 channel (grayscale)
    - convolutional layer:
        - 32 filters with stride 1
        - create 24 * 24 * 32 activation maps
    - max pooling layer 2*2 with stride 2
        - output is (24 - 2) / 2 + 1 = 12
        - output volume: 12 * 12 * 32
    - flatten layer : 4608 * 0.8 = 3686 neurons

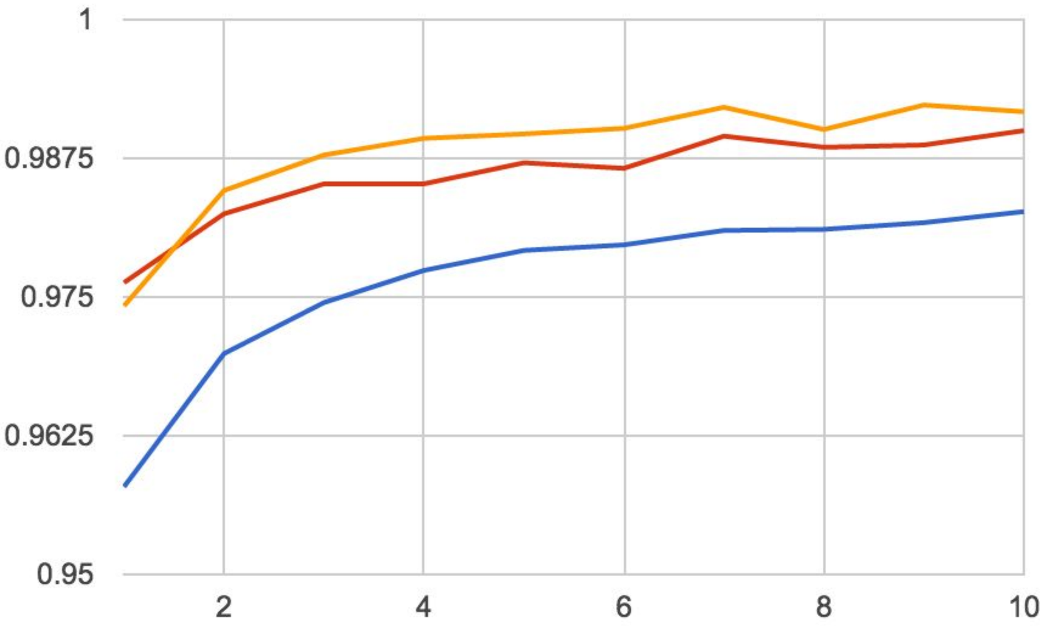


larger CNN

- larger CNN with two conv layer and 2 fully connected hidden layer
  - input size 28 * 28 * 1
  - first convolution layer 24 * 24 * 30
  - first max pooling layer 12 * 12 * 30
  - second convolution layer 10 * 10 * 15
  - second max pooling layer 5 * 5 * 15
  - flattern layer: 375 * 0.8 = 300 neurons



Performance Comparison



🟦	NN	1.73%
🟥	CNN	1.03%
🟧	large CNN	0.83%

How the improvement is achieved

- From generic to CNN
  - capture structural feature
  - blockify the image through each convolution layer to capture each feature individually
  - using multiple filters to capture each feature in depth
  - customized filters for convolution layer
- From CNN to larger CNN
  - two convolutional layer to transfer low-level feature to higher level feature
  - adding hidden layer adds the depth of flattened fully connected layer
  - improvement is not significant in this case

Question