# Spatial Pooler Implementation for MNIST Dataset

Written by David McDougall, May 13th, 2017.

## Summary

I implemented a spatial pooler (S.P.) and Maximal Likelihood classifier and used them to achieve 77% accuracy on the MNIST dataset. My implementation differs in a few ways from Numenta's. It is written in python3.

## Test Methods

I trained the S.P. for 60,000 cycles on the MNIST training data. There are 60,000 samples of training data. I used synthetic training data. I applied random rotations (max rotation of 15 degrees), and random shifts. The resulting (Encoder + S.P. + M.L. Classifier) system is able to correctly classify approximately 77% of the test dataset.

## Implementation Differences

This implementation is based on but differs from the one described by Numenta's Spacial Pooler white paper, (Cui, Ahmad, Hawkins, 2017, "The HTM Spacial Pooler - a neocortical...") in two main ways, the boosting function and the local inhibition mechanism.

## Logarithmic Boosting Function:

In (Cui, Ahmad, Hawkins, 2017), they use an exponential boosting function (see figure 1D from their paper). Notice that their curve intercepts the boost-factor axis and has an asymptote along the activation frequency axis. The activation frequency is by definition constrained to the range [0, 1].

I use the inverse of their function, which intercepts the activation-frequency axis and asymptotically approaches the boost-factors axis. Then scale the boost factor such that at the desired sparsity it equals 1.0

```
boost_function = -log( activation_frequency )
scale_factor   = 1 / boost_function( target_sparsity )
boost_factor   = boost_function( activation_frequency ) * scale_factor
boost_factor   = log( activation_frequency ) / log( target_sparsity )
```

This mechanism has the advantage of having no parameters.
This mechanism yields an entropy of 97% of the theoretical maximum.

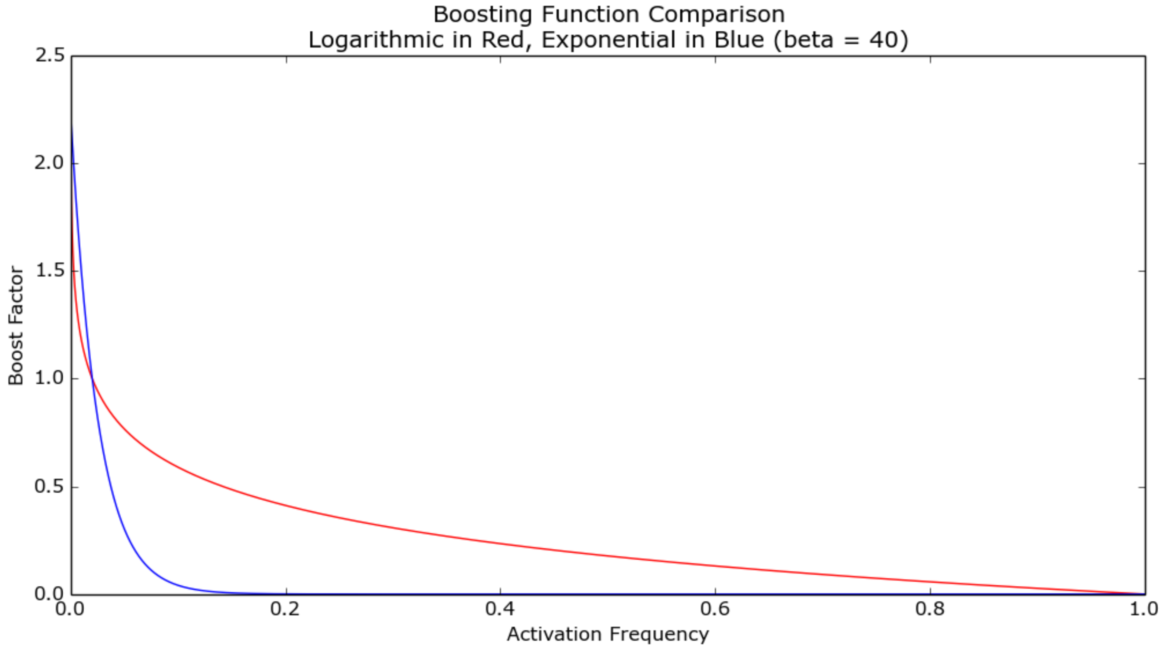

**Figure 1:** Plot comparing the two boosting functions.

## Faster Local Inhibition:

In (Cui, Ahmad, Hawkins, 2017), they activate the top K most excited columns in each area, where K is proportional to the sparsity, and the area is a fixed radius from each column which is proportional to the radius of the receptive field.

I activate the top K most excited columns globally, after normalizing all columns by their local area mean and standard deviation. The local area is a Gaussian filtering and the standard deviation of the Gaussian is proportional to the radius of the receptive field.

In pseudo code:

```
mean_normalized = excitement - gaussian_blur( excitement, radius )
standard_deviation = sqrt( gaussian_blur( mean_normalized ^ 2, radius ) )
normalized = mean_normalized / standard_deviation
activate = top_k( normalized, sparsity * number_of_columns )
```

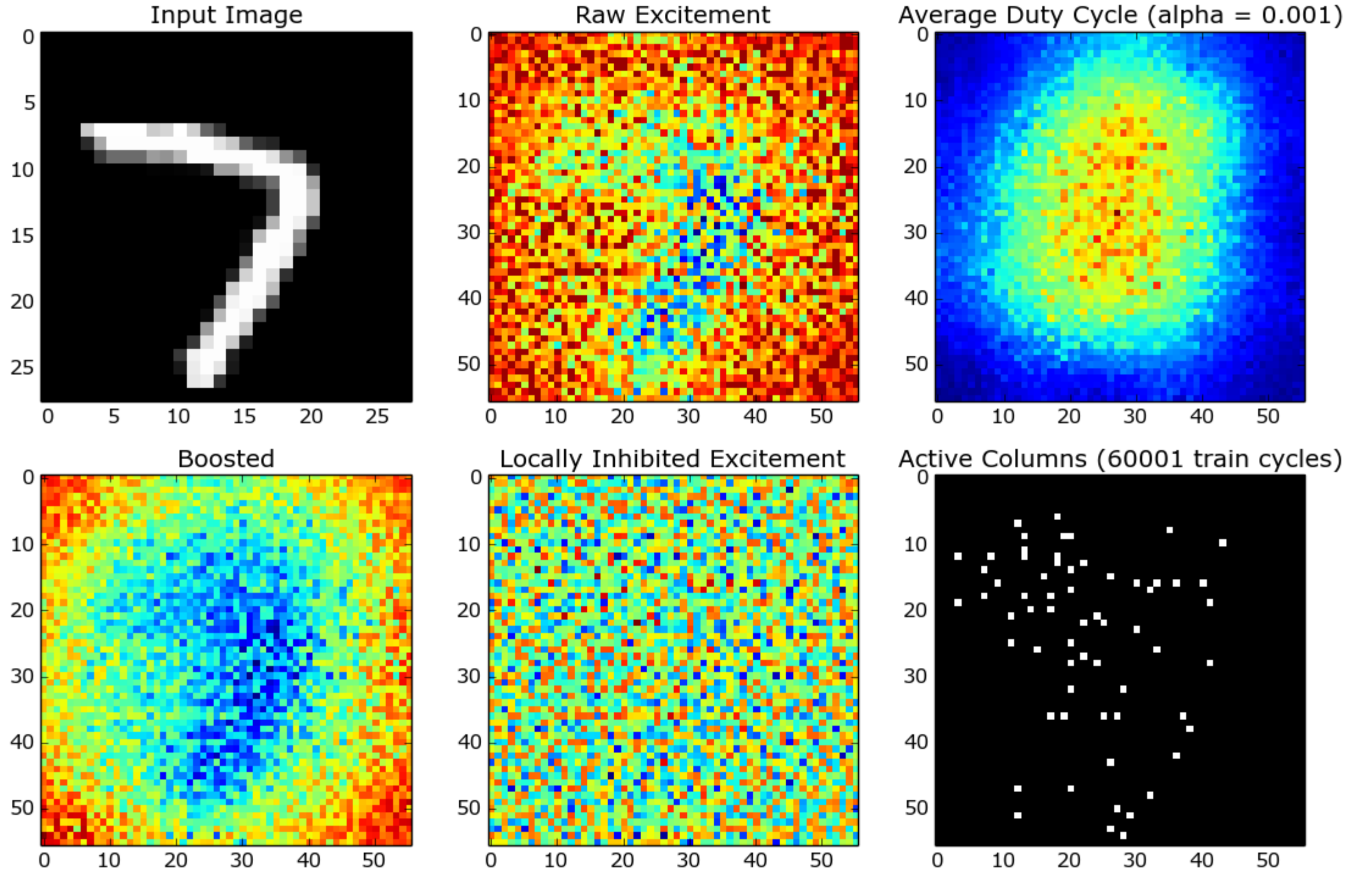

**Figure 2:** The above figure shows the S.P. pipeline. From left to right, top down: Input is accumulated yielding the raw excitement, the activation frequency (aka duty cycle) is used to boost each column (labeled "Boosted"), then that is normalized by each area's mean and std-dev (labeled "Locally Inhibited Excitement"), finally the top 2% of columns are selected.

The following samples of inputs and outputs show that although the active columns overlap with the 'interesting' areas of the input, they are evenly distributed across those areas. This validates my method of local-inhibition followed by global selection which prevents one area from activating disproportionately.

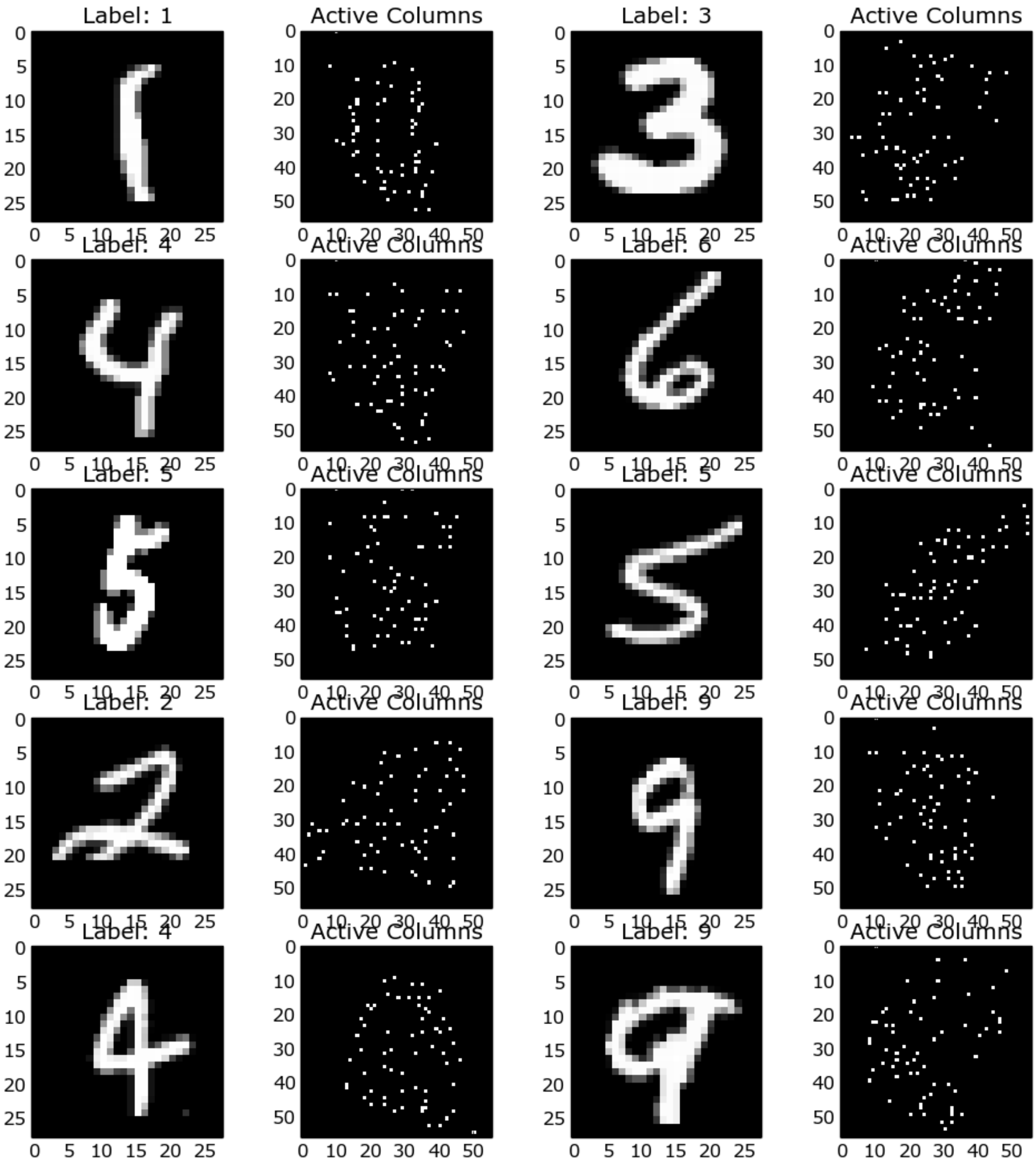

**Figure 3:** samples of inputs and active columns.

Note: The input shape is 28x28, the columns shape is 56x56, and the radius for connecting inputs to columns is 3 units (units from the input space). In these figures the column locations line up with their locations in the input space.

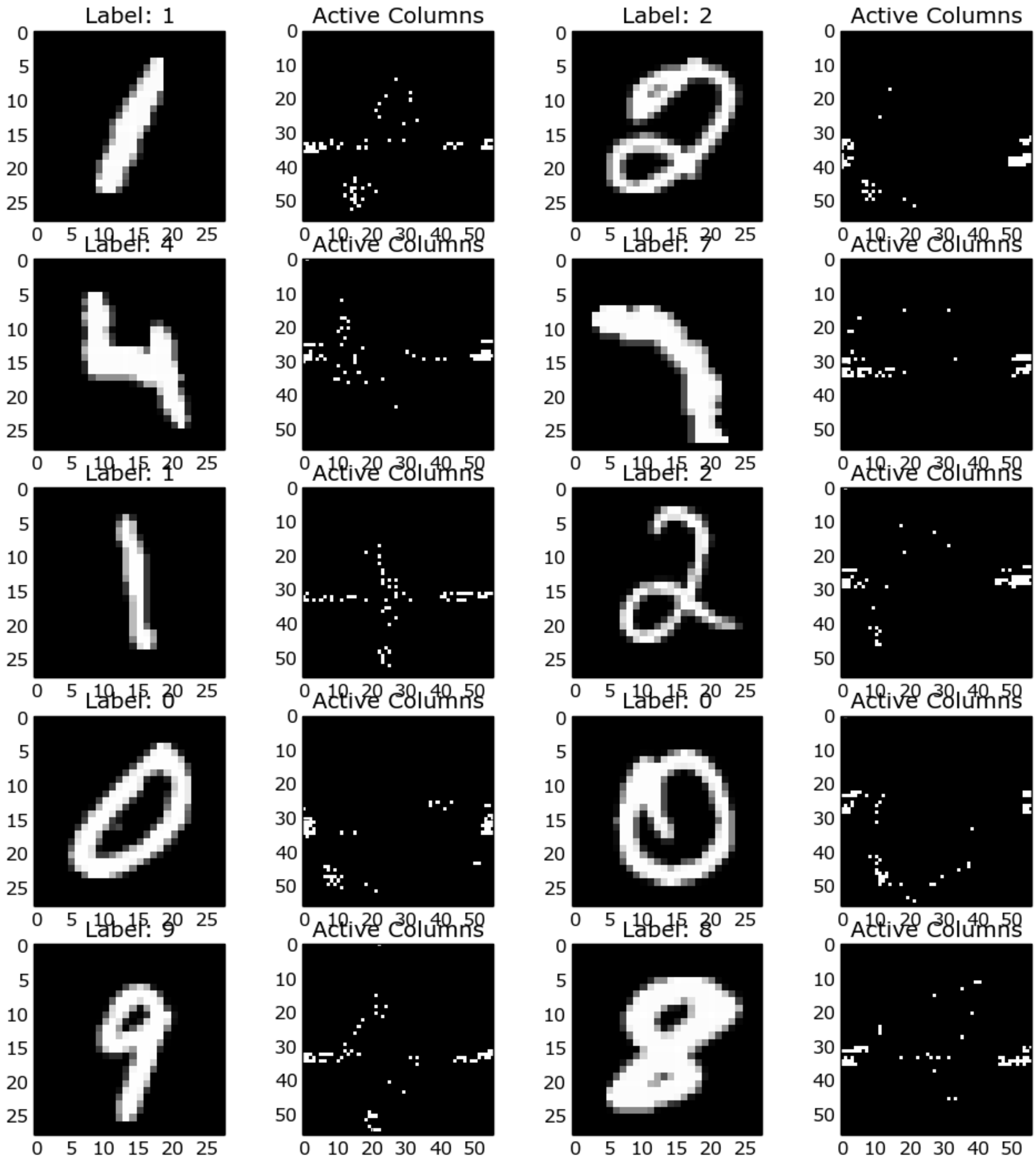

**Figure 4:** In an ablative experiment: I disabled the local inhibition step. Active columns are selected globally and columns still have an input radius. You can see that the active columns still tend to be near the 'interest' areas of the input but that they are clumped. Also, there appear to be areas which are systematically more active.

## Conclusions

I would like to continue working with the MNIST dataset.  I think that with work I could reach my goal of 90% accuracy.  Next I think I will experiment with layering multiple S.P.'s.

An interesting note: there is an interplay between boosting and local inhibition.

**Update:** 11/29/2017, with many fixes and improvements, SP has reached 93% accuracy.