# Homework 2: Convolutional Neural Net Implementation

Alex Faustino - afausti2

September 16, 2019

The attached file, `faustino_hw2.py`, contains the class `ConvNeuralNetwork` which creates a convolutional neural network with a single convolution layer and a single hidden layer. The size of the input, $X$, can be modified, but it is assumed that it is single channel. The number, height, and width of the filter, $K$, can all be modified. The filter parameters determine the dimension of the hidden layer. By default the output of the convolution layer, $Z$, is passed through an element-wise RelU activation function, but this can be changed to sigmoid if desired. The `ConvNeuralNetwork` member function `train` uses SGD to minimize the cross-entropy error and tests the accuracy of the current net after every epoch.

Two additional utility functions are also necessary: `zero_pad` which pads the image's rows and columns with 0's; and the member function `arrange_Conv` which arranges the image's pixel data to speed up the convolution operation.

The default training parameters are:

$$\begin{aligned} X &= 28 \times 28 \\ K &= 16 \times 7 \times 7 \\ \alpha_{init} &= 0.5 \\ \text{epochs} &= 5 \\ \text{batch size} &= 1 \end{aligned}$$

Using the member function `train` with these parameters gives a test accuracy of **98.1%**.