

## Hand written digit classifier using Convolutional Neural Network

Our goal is to correctly identify digits from a dataset of tens of thousands of handwritten images. The data contains gray-scale images of hand-drawn digits, from zero through nine. Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255, inclusive.

**We will use following three codes:**

**1\_CNN\_tuning\_hyperparameters:** In the first code, we will start with a CNN with randomly selected hyper parameters. Thereafter we will tune these hyper parameters in three steps.

We will start with a CNN, having 4 CNN maps, each followed by BatchNormalization. We also used 2 MaxPool2D layers & 2 Dropout layers. Then we used 3 layers of dense network.

We will then tune the hyper parameters in following three steps:

1. Tuning the activation function.
2. Tuning no. of CNN maps, no. of filters in each CNN map, kernel size & no. of nodes in dense layers.
3. Tuning batch size, epochs & dropout percentage.

**2\_BatchNormalisation\_and\_EarlyStopping:** In this code, we will see the impact of batch normalisation and early stopping callback on CNN. Without batch normalisation, training time is much higher. Early stopping helps in preventing overfitting.

**3\_Image\_Augmentation\_and\_final\_model:** In this code, we will use ImageDataGenerator to generate new images from train data images (using image augmentation technique). This technique helps in increasing the training data, which will help in building more accurate models.