**Introduction**

For this project, we have been given a dataset of images containing digits from 0 to 9, which has already been divided into training and test set and we are required to find the most optimum model to train on our dataset. We are required to use two techniques for training, namely, Deep Convolutional Neural Networks and Auto-Encoders.

**Conclusion (CNN)**

The kind of model we choose is dependent on our dataset and the distribution of the data in our dataset greatly impacts the accuracy and efficiency of our model. Hence, it is essential to shuffle the data randomly before training to reduce the chances of having an unfortunate split or of having entire mini-batches of highly correlated examples.

We experiment with many parameters like the learning rate, mini-batch size, momentum and decay term in order to find the optimum model for our dataset. Each parameter has a different purpose and can impact our accuracy on the training and test set in different ways. We can also extend this project to experiment with more parameters like the ratio in which the dataset is divided into train set and test set, the dimensions of the kernel, dimensions of the pooling regions, function used for pooling and so on. However, we do not experiment with these parameters for this project as they are outside the scope of this project.

**Conclusion (Autoencoders)**

We must experiment with a lot of hyper parameters to generate the best results, such as the sparsity constraint parameters, sparsity proportion and sparsity regularization; number of hidden neurons; and transfer functions. All in all, increasing the epochs will allow the autoencoders to learn the features better. Choosing the appropriate transfer functions are also important in order for the autoencoder to learn the features that best represent the input images. From our experiment, we get that logsig-logsig and satlin-satlin transfer functions in the encoder and decoder layer outperforms the combination of other transfer functions. For classification problem, using deepnet is recommended because it has a better performance than training each layer separately.

**Discussions and Challenges**

Ideally, we should try all combinations of all the parameters while trying to determine the best model to use for training on the training data. However, due to the high computational power required for such a task and the time constraints for this project, we first experimented with individual or sets of two parameters while keeping the other parameters constant, and tried to determine the best parameter values from each experiment. We then again experimented with these parameters in the same way, but we set the constant parameters to the values obtained from the first time. The plots shown in this report are the ones we obtained on the second time, using which we again try to find the optimum values of the parameters. Although we might not be able to get the best model, we hope that this process would help us to get closer to the optimum model. It is much more feasible to try all possible models with all possible combinations of parameters in a distributed environment, where each model can be run separately on different machines, which will help to reduce the time taken to train all the models.