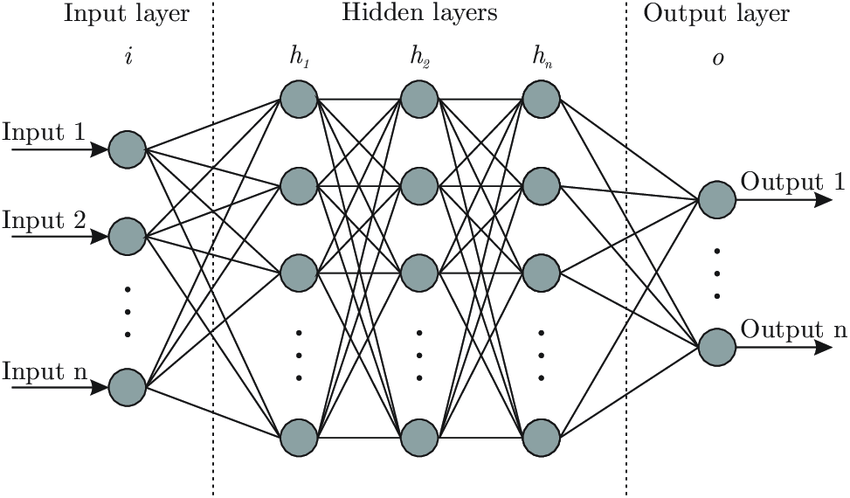
# Introduction

In this paper, we will discuss about Neural network technique, Multi-layer perceptron(MLP). We will use handwritten number dataset to train the model. Data set have pixel data (28\*28) of handwritten letter between 0 to 9. We will experiment different attribute of MLP classifier to find the optimum model to predict the class of letter image.

## Neural Network

Humans have cognitive ability to identify object observed through our eyes based on our experience and information available in our mind with high degree of accuracy. We can recognize person by their face because we have learned over a period how one-person’s facial characteristics differs from others. Artificial neural network is modern machine learning attempt to imitate how human brain work, hence the name Neural network.

Imitating how human’s nervous system consisting of billions of neurons works, artificial neural network attempt to implement multiple hidden layers with will consist neuron like object. A “neuron” in a neural network is a mathematical function that receives the input, decides based on classification structure and pass it to another neuron in next layer. Based on size of hidden layer, number of neuron and complexity of dataset, thousands of network path between nodes are created. This network resembles statistical method such as curve fitting and regression analysis, different cost functions and implementation technique can be used to maximize accuracy of the model.

MLP is a supervised learning neural network algorithm that learns a function by using training dataset, where m and o is the number of dimension of input and output. It can used to train non-linear function approximator for either classification or regression. Unlike in logistic regression, it can have one or more non-linear layers in between input and output layer called hidden layer.

It executes in two phases: Feed-Forward and Back Propagation.

**Feed-Forward**

In this step, weights and bias are accounted on values received from input layer. The resultant unique values are sent to neurons in first hidden layer where they are operated by activation function. The output of activation function is then multiplied by weights of second hidden layer and sent to each neuron. This continues until the final layer is reached, the final value output is produced by algorithm.

**Back Propagation**

Back propagation is a feedback loop system where algorithm attempt to improve the accuracy by minimizing cost function. To minimize the error calculated at the end of the output node, partial derivative of the error function is calculated with respect to all the weights and biases. The error is reduced gradually based on the learning rate “alpha”, this process is called gradient decent. The derivate of cost is used to calculate the slope of error function, slope indicates if the weight needs to be increased or decreased. This process provides feedback to reduce error in next iteration.

One cycle of feed-forward and back propagation is called one epoch. Based on the complexity of model, one can decide or experiment different number of cycles to optimize neural network model.

# Implementation and analysis

To implement artificial neural network with given dataset, we will use MLP classifier package offered by Scikit-learn. MLP offered by scikit-learn is only meant to be used for small scale application as it does not offer GPU support.

We will start by exploring provided dataset. The dataset given has the shape of 42000 row and 785 column, 1st column is the label of number and 784 columns contains the array of 28 \* 28-pixel data.

As part of checking dataset for cleanliness and outliers, we will first list the number of null values in each column, there were no null value found in this data. Since, each pixel reading is expected to be number between 0 and 255, we will check if there is any outlier or typo in the dataset by writing a function to search of any record which lies outside the boundary between 0 and 255. No outliers or typos were found in pixel data.

MLP classifier let us choose from 4 activation functions identity, logistics, tanh and relu. In this model, we are going to use “relu” as our activation function as out target variable is a positive categorical variable with 10 class. There are more than 20 parameters for MLPclassifier function, however we are only going to define few important parameters.

1. mlp = MLPClassifier(max\_iter= 100, activation='relu', learning\_rate= "adaptive", learning\_rate\_init = 0.001 , shuffle = True)

In above model configuration, we have defined activation function will be relu for this model. Along with that, we have also defined that we will let feed forward and back propagation to re-iterate for maximum 100 epoch to achieve optimum loss. Model has been configured to use adaptive learning rate, it will start with initial learning rate of 0.001, it can increase, or decrease based on requirement. Shuffle parameter when set to true will enable sample shuffling in each interaction. The above configured model is then fitted with training set, X\_train and y\_train. This will train the neural network model with adaptive learning rate with maximum 100 epoch to achieve best accuracy score. Although, all the feature has same range, between 0 to 255, we will still scale the numerical value of pixel to assist the model training.

To evaluate the accuracy of the model, we tested model by predicting the class of number based on pixel data in training set. Since, this data was not included in training set, this gives the practical accuracy of model. When tested with testing set, **model yielded the accuracy of 96.66%.**

By reviewing normalized confusion matrix, we can say that model was able to predict the class of dependent variable with very good accuracy. However, looking into accuracy of each class, we can evaluate individual class accuracy. We can see, number 0,1 and 6 were predict with accuracy of 98%, whereas prediction of number 3 and 8 was only correct 95% of time. It could be due to the complexity of the structure of number. However, this can be improved with more samples.

We can also observe that number 9 was predicted 7 and number 3 were predicted 5 very often compared to any other inaccuracies. This tell us now neural networks and how a small similarity can influence wrong prediction. This also indicates that this requires huge sample size to prefect the prediction accuracy score.

**KNN Classifier**

We will train a K-nearest neighbor classification model with the same data. The we will evaluate the accuracy of KNN model at different value of K to compare it with the accuracy of neural network model.

We have trained KNN model at K= 1, 2, 5, 10, Below are the accuracy of KNN model at different K value.

Accuracy score is : 0.937 when value of k is 1

Accuracy score is : 0.924 when value of k is 2

Accuracy score is : 0.935 when value of k is 5

Accuracy score is : 0.930 when value of k is 10

In the above accuracy scores of KNN model at different K value, we can see that accuracy was found best at K=1. KNN classification model is quite basic compared to the working of neural network, yet KNN was able to classify dependent variable with accuracy of 93.7%

# Conclusion

In this paper, we discussed about working of neural network and experimented artificial neural network by building a classification model based on neural network. Neural network was able to predict the classification of data with accuracy of 96.66% .

Most important thing what we notice in above two model was time constraint. Although neural network is more complex and yields better accuracy than KNN model, KNN model took a lot of time to process the calculation and produce result.

KNN model took almost 5 mins for each value of K to train and predict class for training set. The same job was done by neural network in less than 10 seconds. This difference in time is due to the working of each algorithm, KNN model use Euclidean distance between test case and the whole training set. Unlike any other algorithm, KNN model does not remember actual fit line or best fit but remember each record in n-dimensional space. While predicting test case, it calculates distance between test case and each record, evaluates the label of specified number of K-nearest neighbor and take vote to select the class. This to such working nature of KNN algorithm it uses a lot of CPU and takes a lot of time to produce result.

Neural network in other hand passes the value through input node to multiple notes account wights and bias, it iterates the feedback loop to minimize the cost function to reach optimum point, once reached, it remember the structure. The fitted structure can predict significantly fast it does not have to remember all the training set like KNN model does.

Since the feature variable we are dealing with is the array of 28 \* 28 pixel reading, all the features are equally important to train the entire picture of a image. Although the corner pixels of most of the records are mostly 0, because the letter are placed in middle, but it require all 784 pixel information to form a complete image.

We conclude that, neural network is significantly faster than KNN model and can achieve better accuracy.

# References

Scikit-Learn (n.d), Neural network models (supervised), Retrieved from <https://scikit-learn.org/stable/modules/neural_networks_supervised.html#multi-layer-perceptron>

Scikit-Learn (n.d), MLPClassifier, Retrieved from <https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier>

Shukla L (2019), Designing Your Neural Networks, Retrieved from <https://www.kdnuggets.com/2019/11/designing-neural-networks.html>

McCOrmick C (2014), Gradient Descent Derivation, Retrieved from <https://mccormickml.com/2014/03/04/gradient-descent-derivation/>

Robinson S (n.d), Introduction to Neural Networks with Scikit-Learn, Retrieved from <https://stackabuse.com/introduction-to-neural-networks-with-scikit-learn/>