Project Development Phase
Sprint – 2

Date	21 October 2022
Team ID	PNT2022TMID24826IBM
Project Name	Project - A novel Method for Handwritten Digit Recognition System

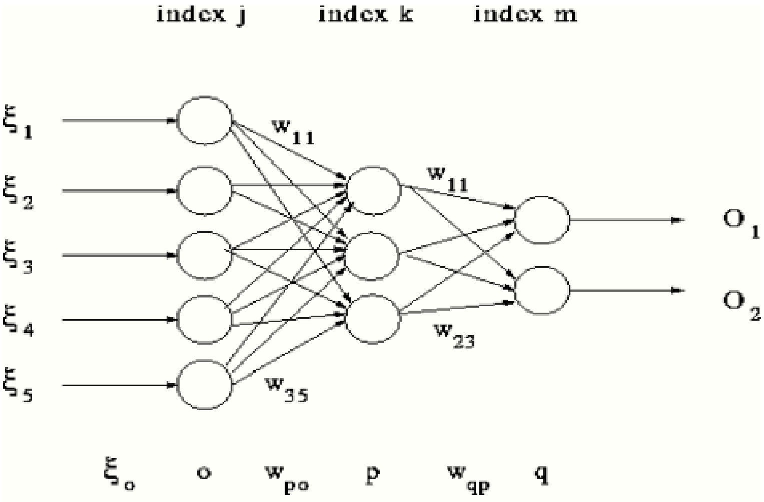
HANDWRITTEN CHARACTER RECOGNITION IN PATTERN RECOGNITION

Linear Classification is a useful method to recognize handwritten characters. The background basis of Artificial Neural Network (ANN) can be implemented as a classification function. Linear Classification works very similar to Artificial Neural Network because the mapping of the ANN cell or one layer of the ANN cell is equivalent to the linear discrimination function. Therefore, if the ANN is a two-layer network, which is consisting of an input and an output layer, it can act as a linear classifier.

Neural Network

A Neural Network (NN) [2]is a function with adjustable or tunable parameters. Let the input to a neural network be denoted by x. This is a real-valued or row vector of length and is typically referred to as input or input vector or regressor or sometimes pattern vector. The length of the vector x is the number of inputs to the network. So let the network output be denoted b Y. This is an approximation of the desired output y, which is also a real-valued vector having one or more components and the number of outputs from the network. The data sets often contain many input and output pairs. The x and y denote matrices with one input and one output vector on each row.

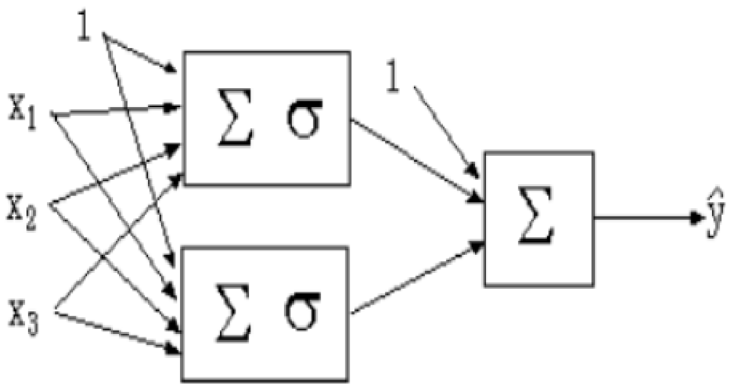
A neural network is a structure involving weighted interconnections between neurons or units. They are often non-linear scalar transformations but can also be linear scalar transformation. The following figure shows an example of a one-hidden-layer neural network with three inputs, x = {x1,x2 ,x3}.



The three inputs, along with a unity bias input, are fed each of the two neurons into the hidden layer. The two outputs from this layer and from a unity bias are then fed into the single output layer neuron. This produces the scalar output Y. The layer of neurons is called hidden layer because the outputs are not directly seen in the data. Each arrow in the diagram corresponds to a real-valued parameter, or a weight, of the network. The values of these parameters are tuned in the training network. A neuron is structured to process multiple inputs. This includes the unity bias in a non-linear way. Then, this produces a single output. All inputs to the neuron are first augmented by multiplicative weights. These weighted inputs are summed and then transformed via a non-linear activation function and as indicated from the neurons in the first layer of the network are non-linear. The single output neuron is linear because no activation function is used. The information in an ANN is always stored in a number of parameters. These parameters can be pre-set by the operator or trained by presenting the ANN with example.

Artificial Neural Network

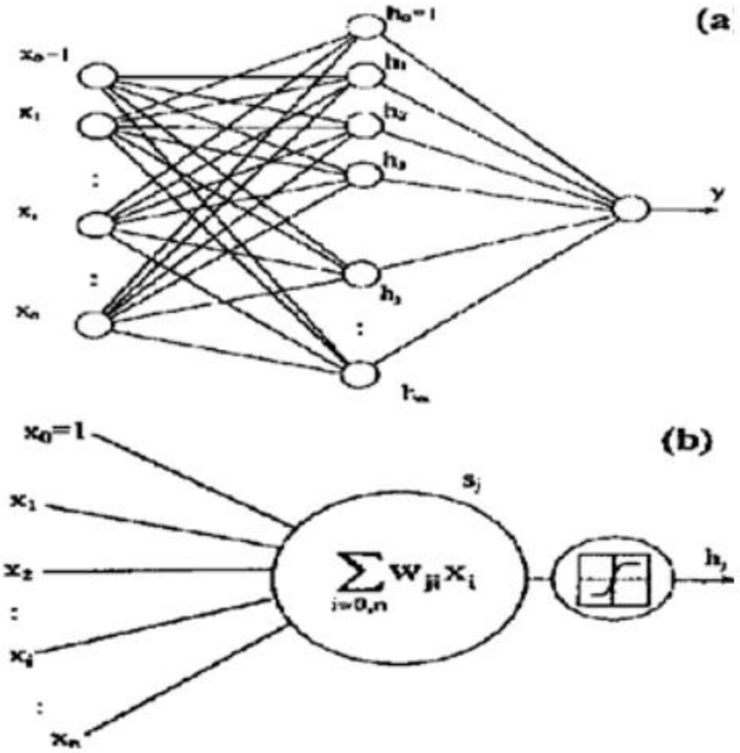
Artificial Neural Network (ANN) has been around since the late 1950's. But it Was not until the mid-1980 that they became sophisticated enough for applications. Today, ANN is applied to a lot of real- world problems. These problems are considered complex problems. ANN‘s are also a good pattern recognition engines and robust classifiers. They have the ability to generalize by making decisions about imprecise input data. They also offer solutions to a variety of classification problems such as speech, character and signal recognition. Artificial Neural Network (ANN) is a collection of very simple and massively interconnected cells. The cells are arranged in a way that each cell derives its input from one or more other cells. It is linked through weighted connections to one or more other cells. This way, input to the ANN is distributed throughout the network so that an output is in the form of one or more activated cells. es of input and also possibly together with the desired output. The following figure3.2 is an example of a simple of ANN:



Examples of types of Neural Networks.

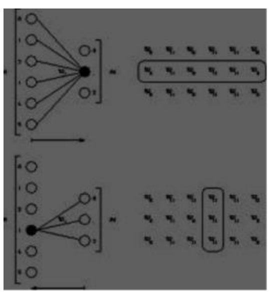
- **Multi-Layer Feed-forward Neural Networks**

Multi-Layer Feed-forward neural networks (FFNN)[9] have high performances in input and output function approximation. In a three-layer FFNN, the first layer connects the input variables. This layer is called the input layer. The last layer connects the output variables. This layer is called output layer. Layers between the input and output layers are called hidden layers. In a system, there can be more than one hidden layer. The processing unit elements are called nodes. Each of these nodes is connected to the nodes of neighboring layers. The parameters associated with node connections are called weights. All connections are feed forward; therefore they allow information transfer from previous layer to the next consecutive layers only. For example, the node j receives incoming signals from node i in the previous layer. Each incoming signal is a weight. The effective incoming signal to node j is the weighted sum of all incoming signals. The following figures are an example of a usual FFNN and nodes:



- **Back-propagation algorithm**.

Back-propagation algorithm[11] consists of two phases. First phase is the forward phase. This is the phase where the activations propagate from the input layer to the output layer. The second phase is the backward phase. This is the phase where then the observed actual value and the requested nominal value in the output layer are propagated backwards so it can modify the weights and bias values. The following figure 3.4 is an example of the forward propagation and backward propagation.



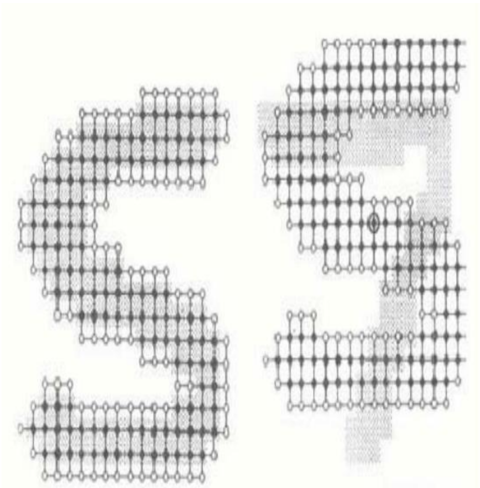
In forward propagation, the weights of the needed receptive connections of neuron j are in one row of the weight matrix. In backward propagation, the neuron j in the output layer calculates the error between the expected nominal targets. The error is propagated backwards to the previous hidden layer and the neuron i in the hidden layer calculates this error that is propagated backwards to its previous layer. This is why the column of the weight matrix is used. n the actual output values. This output value is known from both the forward propagation and backward propagation.

- **Handwritten Character Recognition**

There are many different types of recognitions in the modern time, which can really solve complex problems in the real world today. Examples of recognitions are: face recognition, shape recognition, handwritten character recognition, such as handwritten Chinese character recognition and handwritten digit recognition.

Shape Recognition

Shape describes a spatial region. Most shapes are a 2-D space. Shape recognition works on the similarity measure so that it can determine that two shapes correspond to each other. The recognition needs to respect the properties of imperfect perception, for example: noise, rotation, shearing, etc. One of the techniques used in shape recognition is elastic matching distance. Here we use a binary-valued image X on the square lattice S as an example. The value of X at pixel belonging to S is denoted X(s). The images we are interested in this example are the images of the handwritten numerals. Pixel with value 1 stands for ”black‘ or ”numeral‘ and pixel with value 0 stands for ”white‘ or ”background‘. There are ten numeral classes numbered 0 to 9. These ten numeral classes come in different shapes. The goal is to provide a space of images on S with an alternative metric (X, X‘) that can reduce this intra-class spread as much as possible. Matching problems are not easy tasks. Satisfactory matches can sometimes be obtained reliably and rapidly under two general conditions:



1. Objects to be matched should be topologically structured.

2. Initial conditions should provide a rough guess of the map to be constructed.