1. **INTRODUCTION**
   1. **Motivation**

Since the advent of digital computers, there has been a constant effort to expand the domain of a computer, machine perception being one of them. At present, the ability of the machine to perceive their environment is very limited, however there is enormous partial interest associated with the field of machine perception. Automation of tedious jobs such as mail sorting, inspection for the quality control, analysis of complex biomedical photographs for diagnosis providing sensory prostheses for the blind are a few examples. Character recognition is important branch of machine perception.

The principle impetus to the developers of character recognizers have been given by the need to cope with an enormous flood of paper generated by an expanding technology society. Character recognition essentially requires identification of two-dimensional signal structures which comprise of ideal characters imbedded in noise. The noise arises from stylish variations and random combination of dilation, translation, rotation along with superimposed and detected regions of various spatial extents.

* 1. **Purpose**

The purpose of this system is to recognize the printed document which is given as an image file to the system, it also provides a means to recognize handwritten characters. First the system needs samples to acquire knowledge about the patterns to be recognized.

At present scenario, there is growing demand for a software system to recognize characters in a computer system when information is scanned through paper documents. This concentrates on adaptive methods that operate directly on size-normalized images. Compares the relative merits of Neural Network based handwritten digit recognition system and Optical Character Recognizer.

OCR (Optical Character Recognition) translates images of handwritten characters into machine editable format. OCR refers to a process whereby printed documents are transformed into ASCII files for the purpose of compact storage, editing, fast retrieval and other manipulations through the use of a computer.

OCR reads damaged or low-quality codes and returns the best guess at what the code is. It is widely used as a form of information entry from printed paper data records, whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static data, or any suitable documentation. OCR does not deal with quality and sharpness of characters. The recognition stage of OCR process is made difficult by the added noise, image distortion and the various character typefaces, sizes and fonts that a document may have.

The problem of pattern recognition can be solved by using machine learning. In this project a neural network approach is introduced to perform high accuracy recognition on multi-size and multi-font characters. This gives computers the ability to learn without being explicitly programmed. We use supervised learning where a dataset is given and the correct output should look like as it is already known. We take a large number of handwritten digits, known as training examples and then develop a system which can learn from those training examples to automatically infer rules for recognition. The accuracy with which a neural network classifies patterns depends on how well it is trained.

This study consists of two parts. The first part focuses on recognition of handwritten text. Here the neural network is trained with several different forms of handwritten provided by the users. Later when the user presents the characters, it is down sampled to a standard resolution and this down sampled character is recognised.

The second part consists of scanned document recognition. Here a scanned document is stored as a JPG file. The document must be initially pre-processed so that the image is noise free. The user can select a part or the whole document to recognise the text. The recognised text can be modified by an editor program which provides functionalities like cut copy paste search and replace. The training process consists of loading an image file consisting of all of the alphanumeric characters. The important requirement of out project is that the font type of the trained text and that the text to be recognised must be similar. Erroneous results may creep up if the above condition is not satisfied.

Neural networks approach the problem of classification in a different way. The idea is to take a large number of handwritten digits, known as training examples and then develop a system which can learn from those training examples. In other words, the neural networks use the examples to automatically infer rules for recognizing handwritten digits. Furthermore, by increasing the number of training examples, the network can learn more about handwriting and so improve its accuracy. A neural network is trained using stochastic gradient and feed forward back propagation algorithm.

* 1. **Theoretical Background**
     1. **Pattern Recognition**

Character recognition is a subset of pattern recognition field which may be characterized as an information reduction, information mapping and information labelling process. Feature selection is the process of choosing input to the pattern recognition system and must be done with relevance to the task at hand. Pattern recognition approaches may be broadly classified into:

1. **Statistical Pattern Recognition**

Here a set of features are extracted from the input data and are used to assign each feature vector to one of the classes.

1. **Syntactical Pattern Recognition**

Here interconnections interrelationships of features yield important information. It formulates hierarchical description of complex pattern built up from simpler sub patterns.

1. **Neural Pattern Recognition**

In neural networks computational elements or nodes are connected via weights that are adapted during use to improve performance. The ability to adapt and continue learning is essential in speech and pattern recognition.

* + 1. **Character Recognition Methodologies**

1. **Classification based on the nature of applications**

The scheme can be broadly classified into two main classes based on the nature of application. They are:

1. Online Character Recognition
2. Offline Character Recognition
3. **Online Character Recognition**

Online systems employ graphic tables for capturing the image. As a result, the timing information of each stroke is also available which gives an added advantage in the recognition process. However, the applications are limited as writing on the graphic pad is not convenient and practicable. Here recognition is performed as and when the characters are drawn.

1. **Offline Character Recognition**

These systems employ optical scanning device for grabbling the pre-written images. Hence here the recognition is done after the writing is completed. The y finds extensive use in electronic document processing, prime examples being conversion from one language to other, mail sorting.

1. **Classification Based on techniques**
2. **Template matching and correlation techniques**

These directly compare an input character to standards set of prototypes stored. The comparison varies from decision tree structure as that only selected pixels are tested to as simple as one to one-pixel comparison. These techniques are mainly used for recognition of printed text as they are not adaptive to change in writing style inherent in hand written characters.

1. **Feature analysis and matching**

Here significant features are extracted and compared to the feature description of ideal characters. Thus, the original two-dimensional representation of the character image is replaced by the description of the ‘n’ dimensional feature space. The also reduces the storage requirement. The capabilities of human reasoning are better captured by this method rather than template matching.

1. **Classification based on capabilities**

Based on the nature of writing, difficulty of segmentation and the capabilities the following classes have been defined for the problem of handwritten work recognition:

1. Fixed font recognition system
2. Multi font recognition system
3. Hand printed recognition system
4. Handwritten or cursive recognition system.

It differs from the third category in that here the characters may be connected.

* 1. **Paradigm Used**
     1. **Introduction to Neural Networks**

Computers can perform many operations considerably faster than a human being. Yet there are many tasks where the computer falls considerably short of its human counterpart. There are numerous examples of this. Given two pictures a preschool child could easily tell the difference between a cat and a dog. Yet this same simple problem would confound today computers.

Programs can automate repetitive tasks such as balancing check books or calculating the value of an investment portfolio. While a program could easily maintain a large collection of images, it could not tell us what any of those images are of. Programs are inherently unintelligent and uncreative. Ordinary computer programs are only able to perform repetitive tasks.

A neural network attempts to give computer programs human like intelligence. Neural networks are usually designed to recognise patterns in data. A neural network can be trained to recognise specific patterns in data. The term neural network is usually meant to refer to artificial neural network. An artificial neural network attempts to simulate the real neural networks that are contained in the brains of all animals.

1. **Perceptron**

Perceptron is artificial neuron which is developed in 1950’s and 1960’s by the scientist Frank Rosenblatt, inspired by earlier work by Warren McCulloch and Walter Pitts.

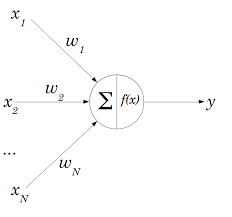


Figure 1.1 A simple perceptron

1. **Working of Perceptron**

A perceptron takes several binary inputs x and produces a single binary output. Rosenblatt proposed a simple rule to compute the output. He introduced weights w real numbers expressing the importance of the respective inputs to the output. The neuron’s output 0 or 1 is determined by the whether the weighted sum is less than or greater than some threshold value. Just like the weights, the threshold is a real number which is a parameter of the neuron.

Output = 0 if <= threshold

1 if > threshold

This is the working of a simple perceptron.

1. **Decision Making**

A perceptron can weigh up different kinds of evidence in order to make decisions and it should seem plausible that a complex network of perceptron could make quite subtle decisions.

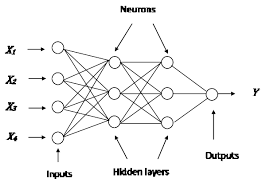


Figure 1.2 Network of perceptron

In this network shown in the figure the first column of perceptron usually called the first layer of perceptron makes three very simple decisions, by weighing the input evidence. The perceptron in the second layer makes a decision by weighing up the results from the first layer of decision making. In this way a perceptron in the second layer can make a decision at a more complex and more abstract level than perceptron in the first layer. In this way, a many layer network of perceptron can engage in sophisticated decision making.

The condition > threshold is cumbersome, and two notational changes are made to simplify it. The first change is to write as a dot product, w.x = , where w and x are vectors whose components are the weights and inputs, respectively. The second change is to move the threshold to the other side of the inequality, and to replace it by what’s known as the perceptron’s bias. Using the bias instead of the threshold, the perceptron rule can be rewritten:

Output = 0 if w.x+b<=0

1 if w.x+b>0

Bias is a measure of how easy it is to get the perceptron to output 1 or to put it in more biological terms, the bias is a measure of how easy it is to get the perceptron to fire. For a perceptron with a really big bias, it’s extremely easy for the perceptron to output 1. But if the bias is very negative, then it’s difficult for the perceptron to output 1.

1. **Implementing Logic Functions**

Another way perceptron can be used is to compute the elementary logical functions such as AND, OR and NAND. For example, suppose a perceptron with two inputs, each with weight -2 and an overall bias of 3

Then input 00 produces output 11, since (-2)\*0+(-2)\*0+3 = 3 is positive. Similar calculations show that the inputs 01 and 10 produce output 1. But the input 11 produces output 0, since it is negative so this perception implements a NAND gate.

The NAND gate implementation show that we can use perceptron to compute simple logical functions. In fact, we can use networks of perceptron to compute any logical function. The reason is that the NAND gate is universal for computation, that is, any computation can be built up out of NAND gates. For example, NAND gates are used to build a circuit which adds two bits x1 and x2 as well as carry bit which is set to 1 when both the inputs are 1.

To get the equivalent network of adder circuit of perceptron all the NAND gates are replaced by perceptron with two inputs, each with weight -2 and an overall bias of 3.

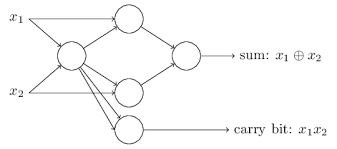


Figure 1.3 Adder circuit using perceptron

It is possible to simply merge the two lines into a single connection with a weight -4 instead of two connections with -2 weights. With that change, the network looks as shown in the figure 1.4 with all unmarked weights equal to -2 and all the biases equal to 3 and a single weight of -4:

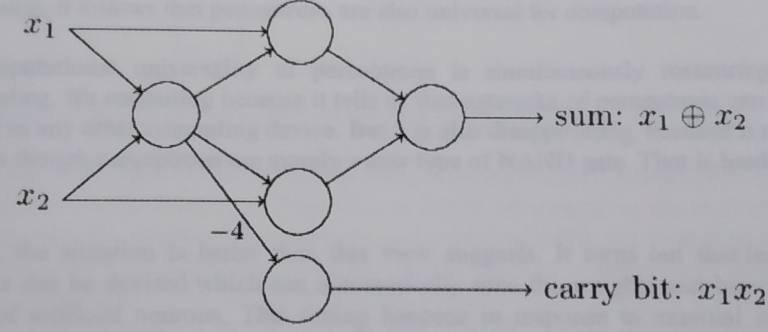


Figure 1.4 Adder circuit with a single connection of weight of -4

In fact, it’s conventional to draw an extra layer of perceptron the input layer to encode the inputs as indicated in figure 1.5

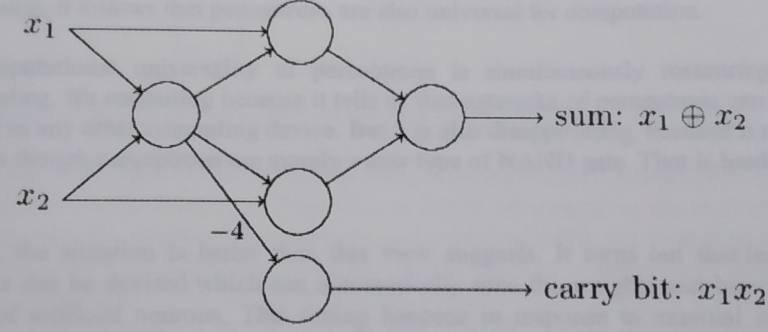


Figure 1.5 Adder circuit with a single connection using extra layer of input perceptron

Figure 1.6 Notation for input perceptron

It has an output but no input. It doesn’t actually mean a perceptron with no inputs. To see this, suppose a perceptron with no inputs is assumed. Then the weighted sum would always be zero, and so the perceptron would output 1 if b>0 and 0 if b<=0. That is, the perceptron would simply output a fixed value, not the desired value (x1, in the example above). It is better to think of the input perceptron as not really being perceptron at all, but rather special units which are simply defined to output the desired values x1, x2….

The adder example demonstrates how a network of perceptron can be used to simulate a circuit containing many NAND gates and because NAND gates are universal for computation, it follows that perceptron are also universal for computation.

The computational universality of perceptron is simultaneously reassuring and disappointing. It’s reassuring because it tells us that networks of perceptron can be as powerful as any other computing device. But it is also disappointing, because it makes it seem as though perceptron are merely a new type of NAND gate. That is hardly big news.

However, the situation is better than this view suggests. It turns out that learning algorithms can be devised which can automatically tune the weights and biases of a network of artificial neurons. This tuning happens in response to external stimuli, without direct intervention by a programmer. These learning algorithms enables to use artificial neurons in a way which is radically different to conventional logic gates. Instead of explicitly laying out a circuit of NAND and other gates, neural networks can simply learn to solve problems, sometimes problems where it would be extremely difficult to directly design a conventional circuit.

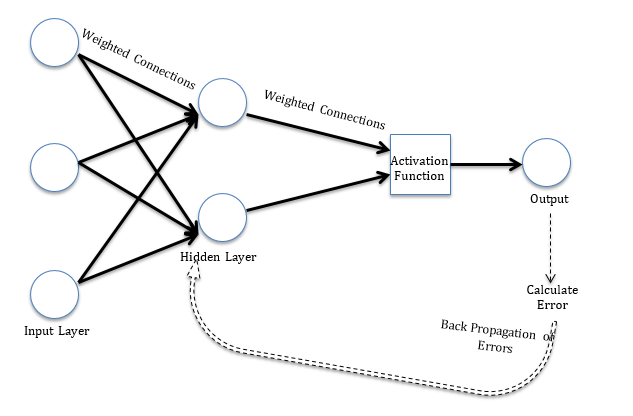
* + 1. **Artificial Neural Network (ANN)**

An artificial neural network is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Artificial neural networks, like people, learn by example. An artificial neural network is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of artificial neural networks as well. A trained neural network can be through of as an expert in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer ‘what if’ questions.

A simple neuron is an artificial neuron is a device with much input and one output. The neuron has two modes of operation, the training mode and the using mode. In the training mode, the neuron can be fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.

* + 1. **Neural Layers**

Neurons are often grouped into layers. Layers are groups of neurons that perform similar functions. There are three types of layers. The input layer is the layer of neurons that receive input from the user program. The layer of neurons that send data to the user program is the output layer. Between the input layer and output layer can are hidden layers. Hidden layer neurons are only connected only to other neurons and never directly interact with the user program.

The input and output layers are not just there as interface points. Every neuron in a neural network has the opportunity to affect processing. Processing can occur at any layer in the neural network. Not every neural network has many layers. The hidden layer is optional. The input and output layers are required, but it is possible to have on layer act as both an input and NeuralNetwork

output layer.

* + 1. **Architecture of Neural Networks**

**Feed-Forward networks**

Feed-Forward ANNs allow signals to travel one way only, from input to output. There is no feedback i.e., the output of any layer does not affect that same layer. Feed-Forward ANNs tend to be straightforward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom up or top down organization.

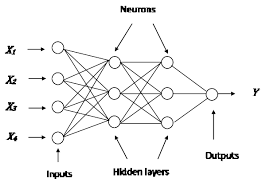


Figure 1.8 Network of Artificial neurons

As mentioned earlier, the leftmost layer in this network is called the input layer, and the neurons within the layer are called input neurons. The rightmost or output layer contains the output neurons, or, as in this case, a single output neuron. The middle layer is called a hidden layer, since the neurons in this layer are neither inputs nor outputs. The term “hidden” perhaps sounds a little mysterious but it really means nothing more than “not an input or an output”. The network above has just a single hidden layer, but some networks have multiple hidden layers. For example, the network shown in the figure has two hidden layers.

The design of the input and output layers in a network is often straightforward. For example, suppose when trying to determine whether a handwritten image depicts a “9” or not, a natural way to design the network is to encode the intensities of the image pixels into input neurons. If the image is a 64\*64 greyscale image, then 4096 = 64\*64 input neurons are present, with the intensities scaled appropriately between 0 and 1. The output layer will contain just a single neuron, with output values of less than 0.5 indicating “input image is not a 9”, and values greater than 0.5 indicating “input image is a 9”.

While the design of the input and output layers of a neural network is often straightforward, there can be quite an art to the design of the hidden layers. In particular, it’s not possible to sum up the design process for the hidden layers with a few simple rules of thumb. Instead, neural networks researchers have developed many design heuristics for the hidden layers, which help people get the behaviour they want out of their nets. For example, such heuristics can be used to help determine how to trade off the number of hidden layers against the time required to train the network.

The neural networks in which the output from one layer is used as input to the next layer. Such networks are called feedforward neural networks. This means there are no loops in the network – information is always fed forward, never fed back. If loops are present, then situations where the input to the sigma function depend on the output occurs. That had to be hard to make sense of, and so such loops are generally not used.

However, there are other models of artificial neural networks in which feedback loops are possible. These models are called recurrent neural networks. The idea in these models is to have neurons which fire for some limited duration of time, before becoming quiescent. That firing can stimulate other neurons, which may fire a little while later, also for a limited duration. That causes still more neurons to fire, and so over time we get a cascade of neurons firing. Loops don’t cause problems in such a model, since a neuron’s output only affects its input at some later time, not instantaneously.

Recurrent neural nets have been less influential than feedforward networks, in part because the learning algorithms for recurrent nets are (at least to date) less powerful. But recurrent networks are still extremely interesting. They are much closer in spirit to how our brains work than feedforward networks and it is possible that recurrent networks can solve important problems which can only be solved with great difficulty by feedforward networks.

**Feedback networks**

Feedback networks can have signals travelling in both directions by introducing loops in the networks. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic their state is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations.

* + 1. **Learning**

There are many different ways that a neural network can learn. Every learning algorithm involves somehow modifying the weights matrixes between the neurons.

Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed. Machine learning is generally classified into supervised and unsupervised learning. In supervised learning, a dataset is given and what the correct output should look like is already known. There is a relationship between input and output. Supervised learning is of two types namely classification and regression. In regression, input variables are mapped into some continuous function. In classification, discrete output results are predicted. The proposed system mainly uses classification. While in unsupervised learning, problems with little or no idea about how the output look like are solved.

Training is a very important process of working with a neural network. There are two forms of training that can be employed with a neural network. Supervised training provides the neural network with training sets and the anticipated output. Unsupervised training supplies the neural network with training sets, but there is no anticipated output provided.

1. **Unsupervised Training**

Unsupervised training is a very common training technique for Kohonen neural networks. This is a general process for training without supervision.

What is meant by training without supervision is that the neural network is provided with training sets, which are collections of defined input values. But the unsupervised neural network is not provided with anticipated outputs.

Unsupervised training is usually used in a classification neural network. A classification neural network takes input patterns, which are presented to the input neurons. These input patterns are then processed, and one single neuron on the output layer fires. This firing neuron can be thought of as the classification of which group the neural input pattern belonged to.

1. **Supervised Training**

The supervised training method is similar to the unsupervised training method in that training sets are provided. Just as with unsupervised training these training sets specify input signals to the neural network.

The primary difference between supervised and unsupervised training is that in supervised training the expected outputs are provided. This allows the supervised training algorithm to adjust the weight matrix based on the difference between the anticipated output of the neural network and the actual output.

There are several popular training algorithms that make use of supervised training. One of most common is the backpropagation algorithm.