AUTOMATED SCORING OF EXAM ANSWER SHEETS USING ARTIFICIAL NEURAL NETWORKS

Prashant Serai Vatsal Modi Girish Sharma

V.E.S. Polytechnic, Chembur

prashantserai@gmail.com vatsal_m3@yahoo.co.in girish142@hotmail.com

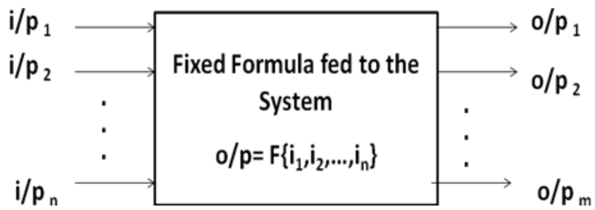
Abstract - ***It was once commented that, machines can't do all what humans can. Gone are those days. Artificial Neural Networks (ANNs) have been developed to emulate the human thinking system; where the advantages of human brains and machines have been combined. ANNs are able to achieve what conventional systems could not. This paper intends to initially provide a short tutorial overview of ANNs. The paper illustrates the basic difference between conventional and intelligent systems. Then, it gives the essence of how ANNs work. Further, it gives an overview on the training aspect and the implementation of ANNs. ANNs are then compared with conventional systems on important aspects. The paper provides the needed introductory knowledge on ANNs to understand their application. Realizing that, the real assessment of a student lies in the correct scoring of his exam answer sheets because of the important role played by marks in determining the future of the students. The present system for scoring involving manual evaluation is faces many problems. With a mission to overcome these, this paper has been developed to propose an automated system for scoring, which can be achieved by the use of ANNs. Then the latter is described, laying a greater emphasis on technical answer sheets, which are estimated to be the most complicated to evaluate automatically. The paper is estimated to be the first one to describe a way to achieve the same. The paper is minimizes the use of 'Jargonistic Terms', for the benefit of engineers and students who are not aware about the field of Neural Networks.***

Introduction

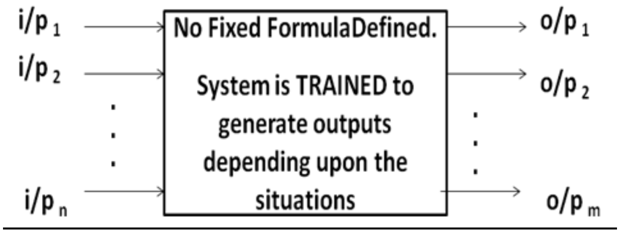
It is rightly said that 'Necessity is the mother of Invention'. Man created machines to reduce his laborious and monotonous work and perform them with unparalleled accuracy. Until recently, machines were not able to do work which required, thinking and analytical skills like humans. However, this scenario has changed; an emulation of the biological neural system has been developed, called Artificial Neural Network (ANN). ANNs, with their perceptional abilities, are able to do work involving such skills.

Intelligent System

The given figures illustrate how, an intelligent system is different from a conventional system.



A Conventional System



An Intelligent System

What is a Neural Network and how does it make for an intelligent system?

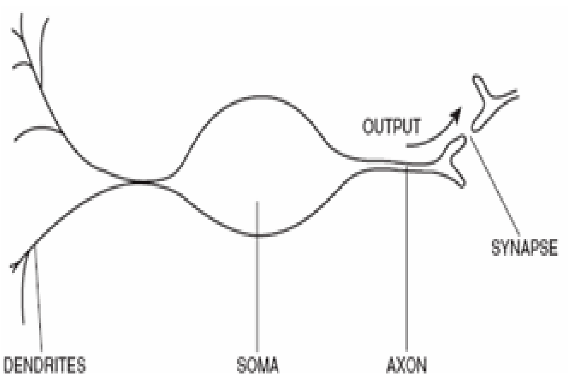
In usual systems, the controller is fed with formulae, one for each output of the system. The equation for each output consists of a formula of those inputs which are significant for that particular output. The degree of significance of each input for a particular output is a result of thinking done by the human brain.[1] Now, the thinking in the brain is done, by a dense network of processing elements or neurons, called a 'Neural Network'. Thus, a system based on the operation of biological neural networks i.e. an emulation of the biological neural system, is called an Artificial Neural Network.[2]

What is the benefit of such a trainable system?

The inputs and the outputs for most of the applications today have a very complex relationship between them and the process of determining the relationship involves many assumptions and approximations. This relationship is often practically quite inaccurate [3] On the other hand, intelligent systems like Neural Networks formulate using the actual conditions thus developing relationships that are much superior and accurate.

A Neuron

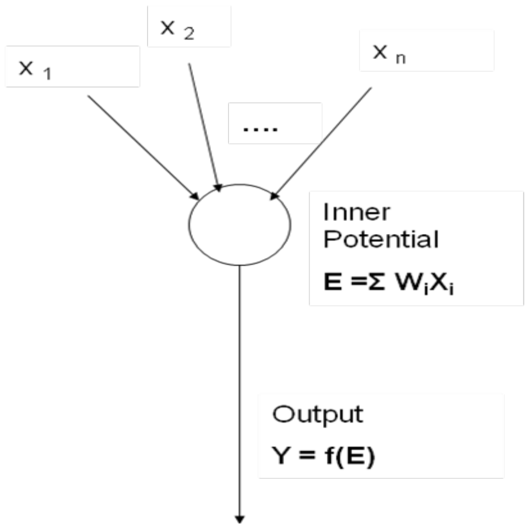
In the biological neuron, the input signal is received along the dendrites.



A Biological Neuron

The soma or the nucleus acts as the summer. When the input exceeds a certain threshold, it causes a chemical reaction to occur, and an electrical pulse to be sent down the axon. The axon acts as an activation function. The connection from one neuron to another goes through the synapses, acting as weights or multiplying factors.[4]

Accordingly the biological neuron gets modeled as,

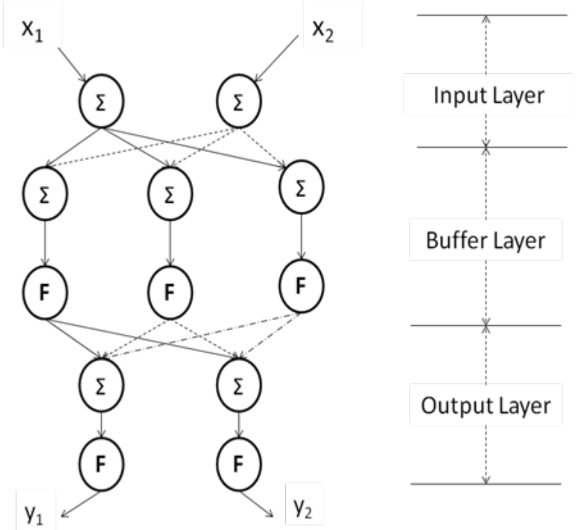


An Artificial Neuron

After taking the weighted sum, E is applied to an activation function f (E). This could be any non-linear function. But, generally, we prefer using the sigmoid function; because it converts inputs in a range of -∞ to +∞ into an activation level between 0 and +1. However, there is no way to calculate beforehand which function will be suitable for given application.

Artificial Neural Network

A neural network is constructed by connecting various neurons in series and parallel. Figure shows a 3 layer neural network,



A 3 layer Neural Network

The input layer acts as a buffer to the hidden layer and may sum inputs. The hidden and output layers are identical and it is these layers that give the network its ability to learn the complex relationships between inputs and outputs. The main processing in the neural network is performed in the hidden and output layers.

ANNs process using the activation function f (E) and the multiplying factors (called weights), which are associated with each input applied to each neuron. The function f (E) remains constant for a particular application, but the required transfer functions can be achieved by the appropriate weights. The process, by means of which, those appropriate weights are obtained is called 'training'.[5]

Training ANNs

Similar to the human brain which learns by experience, the neural networks need to be 'Trained' before they can perform a particular task i.e. producing the desired output for particular kinds of input. For the Training, the neural networks are presented with a varied set of input conditions (known as training examples) such that; if they become successful in handling those situations, they will be able to handle almost all possible situations which they are presented with in similar conditions. After, presenting the ANN with a sample, its weights are varied to achieve the desired output. The same process is repeated with other examples, to achieve weights that satisfy all the training examples. This process might require multiple iterations. For, the process of varying the weights, we make use of 'training algorithms'. The choice of suitable training algorithm should be made according to the application, to achieve the appropriate weights in the minimal number of iterations.[6]

Implementing ANNs

We have seen the mathematical model of artificial neural networks. For hardware implementation, op-amps can be used; also, the entire ANN can be mapped on to a single chip using VLSI technology.

However, with regards to ANNs, there is no way predict before hand: which kind of network, how many number of hidden layers, which training algorithm, etc. will work best for given application. Hence, for the purpose of development, simulation and testing, we use tools like LISP, MATLAB to implement ANNs superficially, though these may be also used as the method for realizing the ANNs for the operation.

ANNs v/s Conventional system

In ANNs, the relationship between the input and output is implicit in nature. These systems are used in applications where the relationship between the inputs and outputs is not clearly definable; and a vast set of input conditions has to be handled.

Whereas, in conventional systems, the relationship between the input and output is explicit in nature. So, these systems are used in applications where the relationship between the inputs and outputs is clearly defined; and the same conditions keep on appearing repetitively.

In conventional systems, we have serial computation i.e.; only 1 thing is processed at a time. So, even if a single activity gets disturbed the entire output gets degraded. In ANN based systems, we have highly parallel computation. So, even if degradation happens, it is graceful, because each neuron has only a small part to play. [8]

Like, conventional systems require programming, Neural Network based systems require training before they can be put to work.

Automated scoring of exam answer sheets

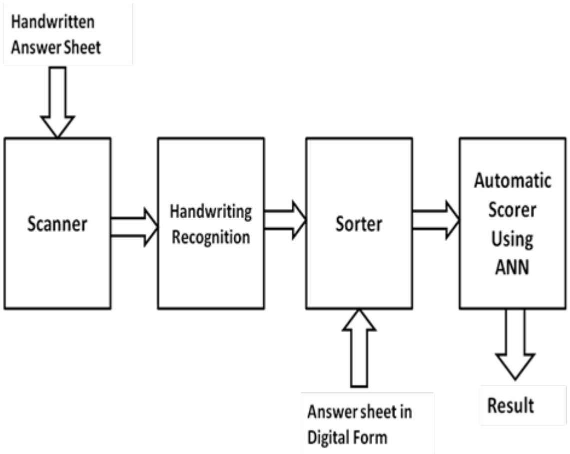
The present paper evaluation system faces many problems in achieving that objective, the prominent ones listed out

- The human evaluators are forced to evaluate a large number of answer sheets in a short period of time. Haste often leads to incorrect evaluation.
- The evaluator might fail to understand the way in which the student has expressed the answer.
- Marks of the student might get affected due to the mood of the examiner or due to mistakes in totaling marks.

With an objective to overcome these issues; we, hereby, propose a system for automatic evaluation of answer sheets, using ANN.

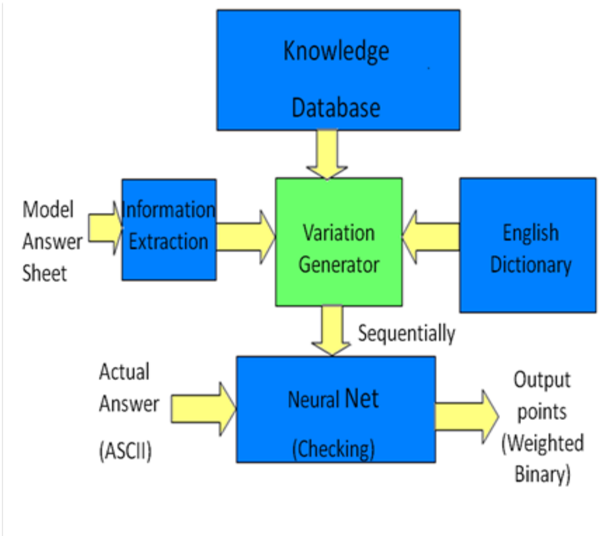
Why use ANN here?

There could be various types of questions asked in question papers, various ways to answer each question, various methods of expressing a point. It would be practically impossible to make a conventional system which can handle such varied inputs. Also it wouldn't be a good idea to place too many restrictions on the paper setter or even on the student. These problems can be effectively avoided using ANNs.



Block Diagram

The answer sheet has to be first scanned using a scanner, the output of which is to be converted into text using suitable handwriting recognition software. If the answer sheet is directly in digital format, then this need not be done. The software will then sort the different answers depending on the numbering given by the student. The theory circuits, programs, block diagrams, derivations, etc. will be sorted differently. Then, it is given to the automated scorer. The ANN in the scorer will be provided with two inputs; one, from the model answer sheet and the other from the actual answer sheet. The model answer sheet has to specify for each answers, what points are expected and how much weightage for each point, diagram, circuit, etc. along with specifying the marks for clarity of expression of point.



Checking of Theory

The figure shows, the process for the checking of Theory. The biggest pain-point in the manual evaluation system is that, the large amounts of theory written by the student, needs to be evaluated. The amount of time required for this is not available.

A theory answer will be scored point wise. A point could be expressed in various ways: a student could use technical synonyms for the terms, he could mention the implied meaning of the term. Also the answer could vary in terms of English vocabulary and grammar. To take care of the technical synonyms and english vocabulary, we have an english dictionary and a knowledge database (both to be updated with time) as resources for a variation generator. The information is extracted out of the model point and given to the variation generator. If even a single variation of the model point is detected in the answer, then, the point has to be marked correct.

The Neural net (checking) will have to be trained using varied sets of data to take care of grammatical variations and detect whether particular points are affirmatively asserted in the given answers.

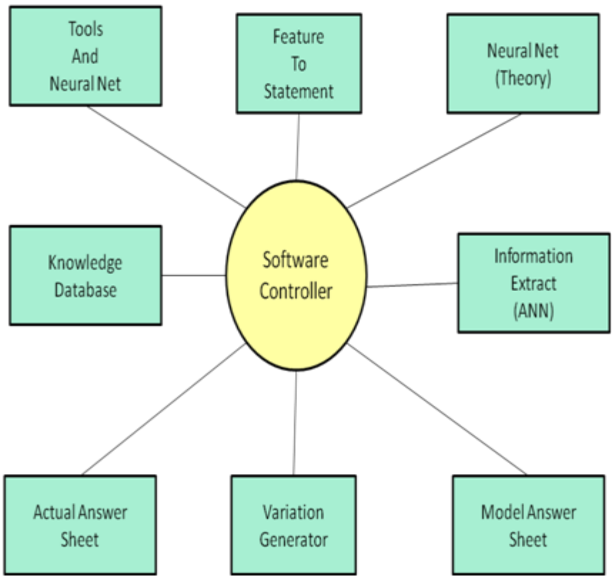
To give an example, consider that, the model point states “The part to the left holds the circuit” and the student writes “The part to the left is called the bench. It holds the PCB.” The neural net in the proposed system will be able to recognize that the point is affirmatively asserted by the student in his answer.[9]

What if the knowledge database doesn’t contain a particular technical term mentioned by the student?

Making of the knowledge database being a one time issue, it would be made by a team of well knowledgable and experienced professors, and with good care that things don’t get left out. Thus, looking at this from another angle, the probability of the knowledge database not knowing about particular terms would be much less than that, of a human evaluator facing such a problem.

Yet, the system can be made more reliable, by defining the range for ambiguity in its output, by determining this during testing. Thus, even if there is an instance of the system’s failure to evaluate a particular answer; the soft copy of the particular answer could be emailed to a human being for evaluation. [10] This person could send the results of the evaluation back to the system in a way that, this reply could be read by the software. Also, every time such problems are detected with the knowledge database, it could be updated.

The previous figure just illustrated the process in which theory evaluation takes place. Actually we have various blocks with a software controller sitting above them and handling the entire system. Let us now see, the actual internal block diagram of the proposed scorer.



Internal Block Diagram of Scorer

Apart from evaluating theory, the other components like block diagrams, circuits, programs, etc. also need to be scored. The software controller manages the entire system, and routes the data through the various blocks as and when required. For the evaluation of such components we need to use certain 'tools', whose output for the model paper version of the component and actual version will be compared using ANNs which will be trained to evaluate the particular component in this manner. For, example, we could use tools like 'P-Spice' for evaluating circuits and ones like 'Keil' for evaluating programs. For the purpose of scoring block diagrams, it is proposed to convert the block diagram into sentence form and score it in a way similar to theory.

Like this various tools could be developed to evaluate various other features like chemical reactions, constructional diagrams and anything else required. This provides for system flexibility.

Conclusion

Artificial Neural Networks are adaptive in nature and are hence more versatile. With their versatility and the ability to provide control customized for specific application, ANNs are set to break all the barriers in automation. Those barriers, which exist because of the inabilities in present day conventional systems, to process like the human brain.

Having covered the introductory parts of ANNs, the paper provides a starting point for those engineers who desire to begin investigating the field of neural networks. While this paper is by no means all inclusive, it provides an initial read for those who are not familiar with ANNs. It has also been intended to bring to light, the necessity and the possibility of having a system for automated scoring of exam answer sheets

References

1) Elements of Artificial Neural Networks
   Kishan Mehrotra, Chilkuri Mohan
   Sanjay Ranka
   MIT Press, 1997

2) 'A neural network approach to artificial Intelligence'
   Circuit Cellar, June-July 1989

3) Applicability of Neural Networks to Industrial and Commercial Power Systems: A Tutorial Overview - S. Mark Halpin, Member, IEEE, and Reuben F. Burch, IV, Senior Member, IEEE

4) 'Introduction to artificial neural networks.' Electronic Technology Directions to the Year 2000, 1995. Proceedings.

5) Artificial neural networks
   Hopfield, J.J.;
   Circuits and Devices Magazine, IEEE
   Volume 4, Issue 5, Sept. 1988

6) NEURAL NETWORKS
   Christos Stergiou, Dimitrios Siganos
   Vol. 4, Journal 94, Documents, Imperial College, London.

7) Introducion to Artificial Neural Networks
   Jacek M. Zurada
   Jaico Publishing House, 1995

8) Artificial neural networks in manufacturing: concepts, applications, and perspectives.Huang, S.H.; Hong-Chao Zhang.
   IEEE Transactions on Components, Packaging, and Manufacturing Technology – Part A, Volume 17, Issue 2, June 1994.

9) Automated Understanding of Financial Statements using Neural Networks and Semantic Grammar, James Markovitch, Search Technologies.

10) Computerized Paper Evaluation Using Neural Network, www.seminarson.com