"Bodh", BPIT's International Journal of Technology & Management
ISSN: 2454-8421, Volume 1, Issue 1, July-Dec, 2015. *Page 59-64*

Neural Network Activation Functions for Image Compression

Anusha Chhabra
(Dept. of IT, Bhagwan Parshuram Institute ofTechnology, Delhi)
anusha.chhabra@gmail.com

Kanika Mittal
(Dept. of CSE, Bhagwan Parshuram Institute ofTechnology, Delhi)
kkanika_virgo@yahoo.com

***Abstract*- Neural Network has been a fascinating area now-a-days. There are various applications in neural network like system identification and control , game-playing and decision making (backgammon, chess, racing), pattern recognition, sequence recognition, medical diagnosis, financial applications, data mining etc. The activation functions used in neural networks can be non-differentiable or discontinuous functions and Differentiable or continuous functions. In this Paper, discontinuous functions and continuous functions are applied on an image to modify it. Then the simulation results show that the continuous function is more efficient than the discontinuous function. Also, a Lena Image is taken on which the modifications are done.**

I. INTRODUCTION

Neural networks take a different approach to problem solving than that of conventional computers [1]. Conventional Computers restricts the solving capability of the problems that is understandable by the user. The operations are predictable in Conventional Computers and unpredictable in Neural Networks but a large number of tasks, require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform maximum efficiency. A neural network is a massively parallel distributed processor made up of simple processing units (neurons) that has a natural propensity for storing experiential knowledge and making it available for use [8-10]. It resembles the brain in two respects: i) Knowledge is acquired by the network through a learning process, and ii) Interneuron connection strengths, known as synaptic weights, are used to store the knowledge. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques [3]. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze.

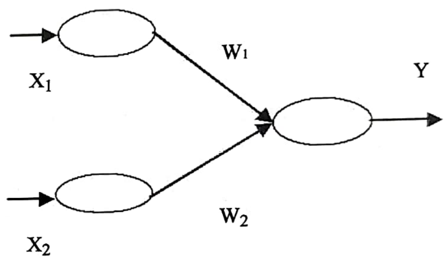

Fig 1Architecture of Simple Artificial Neuron

Figure 1 shows a simplified artificial neural net with two input neurons (X1, X2) and one output neuron(Y). The interconnected weights are given by W1 and W2. The model of ANN is specified by the three basic entities namely: i) model's synaptic interconnections; ii) training or learning rules adopted for updating and adjusting the connection weights; and iii) activation functions.

When creating a functional model of the biological neuron, there are three basic components of importance. First, the synapses of the neuron are modeled as weights. The strength of the connection between an input and a neuron is noted by the value of the weight [6 -8]. Negative weight values reflect inhibitory connections, while positive values designate excitatory connections. The next two components model the actual activity within the neuron cell. An adder sums up all the inputs modified by their respective weights. This activity is referred to as linear combination. Finally, an activation function controls the amplitude of the output of the neuron. An acceptable range of output is usually between 0 and 1, or -1 and 1.

II. RELATED WORK

Many image compression techniques have been proposed in the literature. Image compression is a technique which requires the viewing and storing of images to be standardized. Banerjee and Halder [2] proposed their own algorithm for image compression and compare the results of their proposed algorithm with the JPEG standard and existing BOBC algorithm and report an elegant and simple image compression/decompression algorithm to identify the spatial and spectral redundancy without appreciably sacrificing the quality. Image compression is the application of data compression on digital images. Dhandawate and Joshi [3] discussed the results of image compression with the conventional method for VQ design using SOM and their proposed technique. The main aim was to focus on an efficient VQ design, which will be well applicable to all kind of images in order to improve the quality of reconstructed image. Various quality measures are used for evaluation of performance of compression/decompression. The simulation results in three times less file size when compared with JPEG. Pandian and Anitha [4] proposed a scheme for designing a transform VQ for color image compression using KSOM. The compression of color images is performed by converting color images from RGB to HSV color space. The compression scheme for designing VQ for color image compression using generic codebooks produce reconstructed image with good quality. The image compressed using the DCT transform provide better compression rate with good PSNR values. Tsai, Jhuang and Liu [5] present a new hierarchical SOM to solve the image compression problem. NHSOM uses an estimation function to adjust members of maps dynamically, and reflects the distribution of data efficiently. Kumar, Rai and Shakti [1] show that SOM has been successfully used as a way of dimensionality reduction and feature selection for image compression. SOM may be one dimensional, two dimensional or multidimensional, but most common are either one dimensional or two dimensional maps and the number of input connections depends on the number of attributes to be used in the classification. Wallace [6] proposed the JPEG standard which includes two basic compression methods, each with various modes of operation. A DCT based method is specified for lossy compression and a predictive method for lossless compression. Lu and Shin [7] implemented VQ for image compression based on neural networks. VQ provides high compression ratios and simple decoding processes but implementation of VQ has revealed some major difficulties such as edge integrity and codebook design efficiency. KSOM is known by the ability to form clusters from training samples for pattern classification applications without supervision. Sonal and Dinesh [8] implemented the Image Compression for Self-Organizing Feature Maps which has been used to compress various types of Gray scale images. The PSNR value obtained is 26.89 dB and the time taken for convergence is 320seconds. By adopting the proposed approach the PSNR achieved is 29.79 dB and the time taken for convergence is 185 seconds. The time taken for simulation has been reduced to nearly 50%. The performance of the Self-Organizing Feature Maps has been substantially improved by the proposed approach.

III. LEARNING PROCESS

The main property of an ANN is its capability to learn. Learning or training is a process by means of which a neural network adapts itself to a stimulus by making proper parameter adjustments, resulting in the production of desired response. The learning in an ANN can be generally classified into three categories as:

A. Unsupervised learning

In this process, the learning process is independent and is not supervised by a teacher. In ANN's following unsupervised learning, the input vectors of similar type are grouped without the use of training data to specify how a member of each group looks or to which group a number belongs [9]. In the training process, the network receives the input patterns and organizes these patterns to form clusters. When a new input pattern is applied, the neural network gives an output response indicating the class to which the input pattern belongs. If for an input, a pattern class cannot be found then a new class is generated. The block diagram of unsupervised learning is shown in figure2:

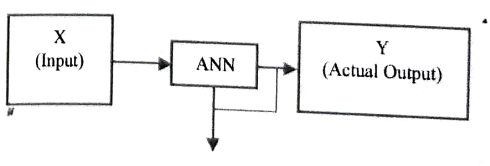

Fig 2 Block Diagram of Unsupervised learning

Features of Unsupervised Learning:

1. No help from the outside.
2. No training data, no information available on the desired output.
3. Learning by doing.
4. Used to pick out structure in the input.
5. Clustering.
6. Reduction of dimensionality and compression.

B. Supervised learning

Supervised learning is fairly common in classification problems because the goal is often to get the computer to learn a classification system that we have created [9] Digit recognition is a common example of classification learning. The block diagram of supervised leaning is shown in Figure3:

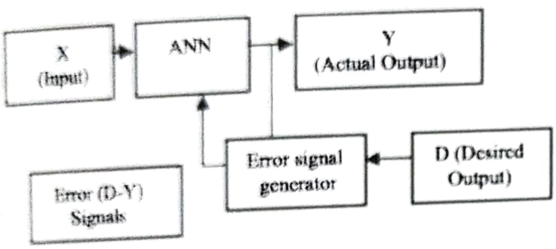

Fig 3 Block Diagram of Supervised learning

Supervised learning is the most common technique for training neural networks and decision trees. Both of these techniques are highly dependent on the information given by the pre-determined classifications. In the case of neural networks, the classification is used to determine the error of the network and then adjust the network to minimize it, and in decision trees, the classifications are used to determine the attributes, provide the most information that can be used to solve the classification puzzle.

C. Reinforcement learning

This learning process is similar to supervised learning. In the case of supervised learning, the correct target output values are known for each pattern. But, in some cases, less information might be available [9]. For example, the network might be told that its actual output is only "50% correct" or so. Thus, here only critic information is available, not the exact information. The learning based on this critic information is called reinforcement learning and the feedback sent is called reinforcement signal. The reinforcement learning is shown in figure4:

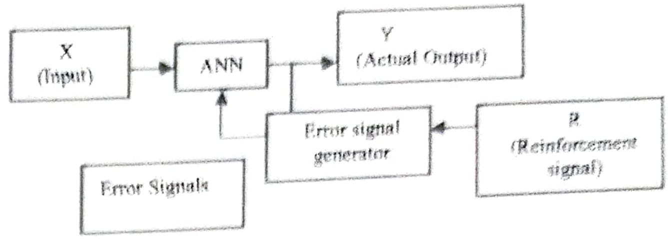

Fig 4 Block Diagram of Reinforcement Learning

Generally, the neural network is formed in three layers, called the input layer, hidden layer, and output layer. Each layer consists of one or more nodes, represented in figure5 by the small circles. The lines between the nodes indicate the flow of information from one node to the next. In this particular type of neural network, the information flows only from the input to the output. Other types of neural networks have more intricate connections, such as feedback paths. The nodes of the input layer are passive, meaning they do not modify the data. They receive a single value on their input, and duplicate the value to their multiple outputs. In comparison, the nodes of the hidden and output layer are active. This means they modify the data . The variables: X1(1)...X1(6) hold the data to be evaluated. For example, they may be pixel values from an image, samples from an audio signal, stock market prices on successive days, etc. Each value from the input layer is duplicated and sent to all of the hidden nodes. This is called a fully interconnected structure.

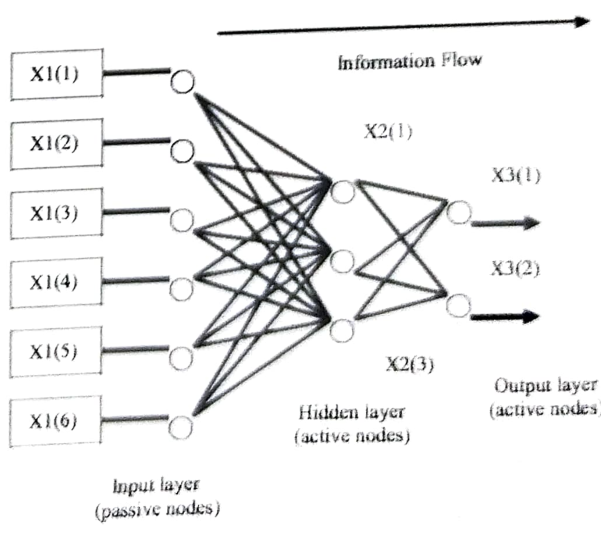

Fig 5 Feed forward Network

The outputs from the hidden layer are represented in the flow diagram of Figure 5, by the variables: X2(1), X2(2), and X2(3). The active nodes of the output layer combine and modify the data to produce the two output values of this network, X3(1) and X3(2). Neural networks can have any number of layers, and any number of nodes per layer. Most applications use the three layer structure with a maximum of a few hundred input nodes. The hidden layer is usually about 10% the size of the input layer. In the case of target detection, the output layer only needs a single node. The output of this node is threshold to provide a positive or negative indication of the target's presence or absence in the input data. As an example, imagine a neural network for recognizing objects in a sonar signal, the values entering a hidden node are multiplied by weights, a set of predetermined numbers stored in the program. The weighted inputs are then added to produce a single number. This number is passed through a nonlinear mathematical function called a sigmoid. This is an "s" shaped curve that limits the node's output. That is, the input to the sigmoid is a value between -∞ and +∞, while its output can only be between 0 and 1.

IV. ACTIVATION FUNCTIONS

To make the work more efficient and to obtain exact output, some force or activation may be given. This activation helps in achieving the exact output. In a similar way, the activation function is applied over the net input to calculate the output of an ANN. The information processing element can be viewed as consisting of two major parts: input and output [10]. An integration function (say f) is associated with the input of a processing element. This function serves to combine activation, information or evidence from an external source or other processing elements into a net input to the processing element. A typical type of activation function like threshold function is shown in figure6:

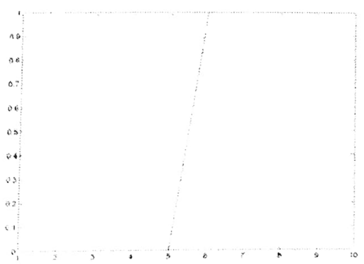

Fig 6 Threshold Function

Activation functions for the hidden units are needed to introduce non-linearity into the networks. The reason is that a composition of linear functions is again a linear function. However, it is the non-linearity (i.e., the capability to represent nonlinear functions) that makes multi-layer networks so powerful. Almost any nonlinear function does the job, although for back-propagation learning it must be differentiable and it helps if the function is bounded.

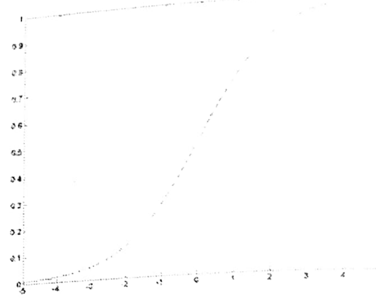

Fig 7 Sigmoid Function

The sigmoid functions shown in figure7, are the most common choices. For the output units, activation functions should be chosen to be suited to the distribution of the target values. For continuous-valued targets with a bounded range, the sigmoid functions are again useful, provided that either the outputs or the targets to be scaled to the range of the output activation function. But if the target values have no known bounded range, it is better to use an unbounded activation function, most often the identity function (which amounts to no activation function). If the target values are positive but have no known upper bound, an exponential output activation function can be used. An Error Signal originates at an output of the network, and propagates backward (layer by layer) through the network. This is said as an error signal because its computation by the network involves an error dependent function in one form to another.

Each hidden or output of a neuron of a multilayer perceptron is designed to perform two computations:

1. The computation of the function signal appearing at the output of a neuron, which is expressed as a continuous nonlinear function of the input signal and synaptic weights associated with that neuron.
2. The computation of an estimate of the vector which is needed for the backward pass through the network.

The error signal at the output of neuron j at iteration n is defined by equation 1:

$$ej(n)=dj(n)-yj(n) \qquad (1)$$

where,
ej(n) is the error energy,
dj(n) is the desired response, and
yj(n) is the actual output.

V. OUTPUTS


Fig 8 Original Image


Fig 9 Conversion of the original Image into Gray Scale


Fig 10 Resultant Image after applying Threshold Function on a Gray Scale Image


Fig 11 Resultant Image after applying Sigmoid Function on a Gray Scale Image

VI. OBSERVATIONS

In this Paper, the desired response d is the mean of all the matrix values obtained from the gray scale image. And the actual output y is the mean of all the matrix values obtained from the resultant image, which is further obtained after applying the functions.

The mean value obtained from the matrix of gray scale image is 5323.34, which is the value of desired response.

<u>Observation 1: Average Error Value by Threshold Activation Function</u>

The mean value obtained from the matrix of resultant image obtained by applying the threshold activation function is 3544.29, which is the value of desired response.

The Error value obtained from this result is

$$5323.34-3544.29 = 1779.05$$

i.e., the Percentage of error is 17.79%.

<u>Observation 2: Average Error Value by Sigmoid Activation Function</u>

The mean value obtained from the matrix of resultant image obtained by applying the sigmoid activation function is 5108.29, which is the value of desired response.

The Error value obtained from this result is

$$5323.34-5108.29 = 215.05$$

i.e., the percentage of error is 2.15%

VII. CONCLUSION

From the implementation, it has been concluded that the Sigmoid Function is good to compress the image than the Activation Function. As the Sigmoid Activation Function is continuous and Differentiable, so the value of Mean Square Error is less in case of Sigmoid Function. As a result, the modification of an image done by the Sigmoid Activation Function gives better Performance than Threshold Activation Function.

REFERENCES

[1] D. Kumar, C.S. Rai, S. Kumar,, " **Face Recognition using Self- Organizing Map and Principal Component Analysis,**" *Proc. IEEE*, 2005.

[2] Banerjee and A. Halder, "**An Efficient Dynamic Image Compression Algorithm based on Block Optimization, Byte Compression and Run-Length Encoding along Y-axis**", IEEE Transaction, 2010.

[3] Y. H. Dandawate and M.A. Joshi, "**Performance analysis of Image Compression using Enhanced Vector Quantizer designed with Self Organizing Feature Maps: The Quality perspective**" IEEE transaction, 2007.

[4] S. Immanuel Alex Pandian and J.Anitha, "**A Neural Network Approach for Color Image Compression in Transform Domain**", International Journal of Recent Trends in Engineering, Vol 2, No. 2, November 2009, 152.

[5] Cheng-Fa Tsai, Chen-An Jhuang, Chih-Wei Liu, "**Gray Image Compression Using New Hierarchical Self-Organizing Map Technique**", IEEE conference, 2008.

[6] Gregory K. Wallace, "**The JPEG Still Picture Compression Standard**", IEEE Transactions on Consumer Electronics, December 1991.

[7] C. Lu and Y. H. Shin, "**A Neural Network based Image Compression System**", IEEE Transaction, Dec. 12, 1991.

[8] Sonal, Dinesh Kumar, "**A Study of various Image Compression Techniques**", IEEE Transaction, 2005.

[9] Aditya, A.F., "**Learning on a General Networks" In Neural Information Processing Systems**", D.Z., Anderson, ed, New York: American Institute of Physics, 1987.

[10] D E. Tamir," **Competitive Neural Network Training: A Multi-resolution Approach**", 2008.