Secure Compression of Bitmap character Using Artificial Neural Network for Transmission

Deepika Gupta[1], M. tech Pursuing, Sobhasaria Engineering College, Sikar
Phone :+91-9460126991, Email :deepika.gupta1218@gmail.com
Mohit Arora[2], M. tech Pursuing, Sobhasaria Engineering College, Sikar
Phone :+91-9602900116, Email :er.m.arora@gmail.com
Anjali Singh[3], M. tech Pursuing, Swami Keshvanand Institute of Tech. , Jaipur
Phone :09610570812, Email :27.anjali01@gmail.com
Priyanka Gaur[4], MBA, Engineering college Bikaner, bikaner
Phone :+91-9261144515, Email : er.priyankagaur@gmail.com

Abstract
The transmission of data across communication paths is an expensive process in respect time and bandwidth. Data compression is usually obtained by substituting a shorter symbol for an original symbol in the source data, which should contain the same information but with a smaller representation in length. The purpose of this paper is to show that neural networks may be promising tools for data compression without loss of information. We combine neural nets, standard statistical compression methods like Huffman coding and arithmetic coding. This paper uses Artificial Neural Network (ANN) based techniques provide new ways for the compression of data at the transmitter and decompression at the receiver with more secure manner. In this paper, security of the data can be obtained along the communication path as it is not in its original form on the communication line.

Key Words*: Artificial Neural Network (ANN), statistical compression method, Back-propagation (BP), Hidden layer architecture, Weight matrix etc.*

Introduction

This paper presents a technique to compress character bitmap type data using neural network. A single learning rate is used throughout the training process in Conventional neural network, but here we proposed a separate learning rate for each output neuron, depending upon the sign of present error gradient and their difference.

We are taking the output of hidden layer as the compressed output of the data and this is send to the remote location along with the weight matrix of hidden to output layer. This weight matrix acts as a key to decode the compressed data. Obviously the information about the number of hidden neuron should be known to reconstruct the original data. We introduce a model that produces better compression than popular Limpel-Ziv compressors (zip, gzip, compress), and is competitive in time, space, and compression ratio with PPM and Burrows- Wheeler algorithms, currently the best known.

2. Secure Compression of data and Transmission

The transmission of data across communication paths is an expensive process in respect time and bandwidth. Data compression provides an option for decrease the number of characters or bits in transmission. It has become increasingly importance of most computer networks, as the volume of information/data traffic has begun to exceed their full capacity for transmission. Artificial Neural Network (ANN) based techniques provide new ways for the compression of data at the transmitter and decompression at the receiver with more secure manner. The security of the data can be obtained along the communication path as it is not in its original form on the communication line. The purpose of this paper is to present application of Artificial Neural network in data compression with unique encryption for transmission.

3. Neural Network

Neural network is composed of a group or groups of chemically connected or functionally associated neurons. A single neuron may be connected to many other neurons and the total number of neurons may be extensive. A hierarchical neural network model is simply model and has some layers of neurons and every neuron links neurons in only next layer .Every neuron has value

$$y = \sum_i x_i W_i$$

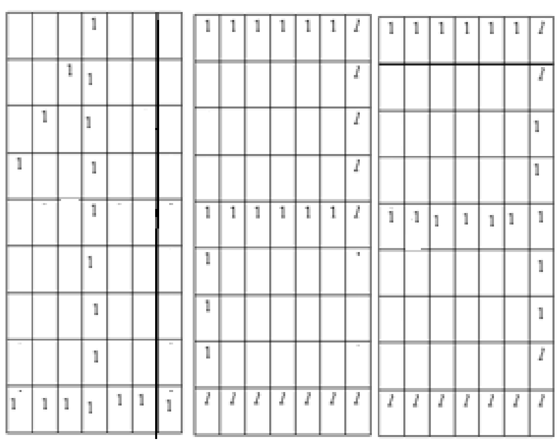
where x_i (i =1,2,L) is value from neuron in previous layer and W_i (i =1,2,L) is weight which mean strength value of links. In the artificial intelligence field, artificial neural networks have been applied successfully to speech recognition, image analysis and adaptive control, in order to construct software agents (in computer and video games) or autonomous robots. Most of the currently employed artificial neural networks for artificial intelligence are based on statistical estimation, optimization and control theory.

The cognitive modeling field involves the physical or mathematical modeling of the behavior of neural systems; ranging from the individual neural level (e.g. modeling the spike response curves of neurons to a stimulus), through the neural cluster level to the complete organism.

In hierarchical neural network model, when the number of neurons of the first and the last are same and the middle layer is fewer than others and learning with output data identity with input data

3.1 Detail architecture of neural network.

We are compressing bitmap of characters having 9 rows and 7 columns. Bit map is made like this



Letter 1 Letter 2 Letter 3

Fig1: matrix in row manner

We give input whole matrix in a row manner. The input neuron of the neural network is 63 (because 9 x 7 =63) after lots of experiment we came to conclusion about using 25 neurons in the hidden layer. We can use any no. of neuron from 25 to 63 in the hidden layer. But at least we have to use 25 in the hidden layer. And at last 63 neurons in the output layer. The algorithm is as follows:-

Step 1 :- We make an identical mapping of the input to the output. Then we store the hidden layer to the output layer weight matrix. The weight training in this layer is done with the variable alpha for each output neuron.

Step 2 :- We have trained 26 letters of capital A-Z. Then the weight matrix 25 x 63 is stored in the remote location. **Step 3** :- Once again we feed the required bitmap-characters to the trained neural Network once and stored the output of the hidden layer in a file.

This is the file which we have to send to the remote location through internet or network. The compression ratio we get is around 25/63.We can vary the no. of hidden layer to get our required compression ratio. But our experience suggests that anything above 25 neuron should be used to get reasonable recovery of the original data. If we use 30 or more than that number of neuron in the hidden layer it would be better in terms of recovery.

Step 4 :- We interchange the 1st 10 column with the last 10 column, to ensure more encryption measure. And also we add the sign (no of char) to every element of the hidden layer output matrix to make it more difficult to understand.

Step 5 :- In the remote location the structure is of 25 x 63. The weights of the output are the weights

Step 6 :- The calculated during training purpose inputs to the neurons are those generated during the compression of the Bitmap characters. At first it is being read in a matrix. It will form a matrix of Order N x 63 where N is the number of characters compressed. We then follow the reverse procedure to get back the original data.

- Subtract sign(N) from every element
- Interchange 1st 10 column with last 10 column
- Finally we get the input to be given to the neural network in the reconstruction process.

Step 7 :- In a single iteration we can reconstruct the desired bitmap characters.

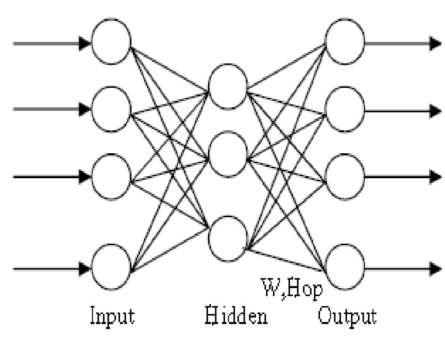


Fig2. *Hidden layer architecture*

We are storing hidden layer weight in the remote location and hidden layer output along with the necessary security conversion is stored in the file which is the desired compressed data.

3.2 Data Compression using Artificial Neural network

Artificial Neural Networks have been applied to many problems, and have demonstrated their superiority over classical methods when dealing with noisy or incomplete data. Neural networks seem to be well suited to this particular function, as they have an ability to preprocess input patterns to produce simpler patterns with fewer components. This Compressed information (stored in a hidden layer) preserves the full information obtained from the external environment. The compressed features may then exit the network into the external environment in their original uncompressed form.

3.3 Back-propagation Neural Network

The Back-propagation (BP) algorithm has been one of the most successful neural network algorithms applied to the problem of data compression. The data compression problem in the case of the BP algorithm is posed as an encoder problem. The data or image to be compressed passes through the input layer of the network, and then subsequently through a very small number of hidden neurons. It is in the hidden layer that the compressed features of the image are stored, therefore the smaller the number of hidden neurons, the higher the compression ratio. The output layer subsequently outputs the decompressed image to the external environment. It is expected that the input and output data are the same or very close. If the image to be compressed is very large, this may sometimes cause difficulty in training, as the input to the network becomes very large. Therefore in the case of large images, they may be broken down into smaller, sub-images. Each sub-image may then be used to train an individual ANN. Experiments have been conducted that have Successfully compressed and decompressed character bitmap data with impressive compression ratios, and little or no loss of data.

4. Network security tool for data encryption/decryption

The problem of security of information is major problem in information and communication system existed since information has been managed. However, as research and invention provide new technology for information management systems become more and more powerful tool for handling this problem The massive use of the communication networks for various purposes in the last two decade has posed new serious security threats and increased the potential damage that violations may cause. Now every industry and institute is increasing their work on computer network environments, they are becoming more critical Issue to secure information on network. Both public and corporate sectors more than ever today depend on the information network. Today many network security tool are using for transmission of information over network but all that tools are not give 100% security Advances in artificial neural networks (ANNs) provide effective solutions to this problem ANNs. The security problem is considered here as the problem of keeping communications over the network private. Thus, protection of information is required against possible violations that can compromise its secrecy (or confidentiality). Secrecy is compromised if information is disclosed to users not authorized to access it. While the encryption scheme used in this work is based on Boolean algebra, the decryption scheme here is based on neural network techniques that use back propagation learning algorithm.

5 Compression of data

Feature of this method is following:

1. The Strength of links of neurons tells us which data influence to compression components.

2. In principal component analysis or factor analysis, we decide the number of components after checking results. But in neural network model, we can decide the number of compression components before analysis, because of we decide the number of neurons in middle layer.

6. Result and Output

Fig3. *Input Pattern displayed*

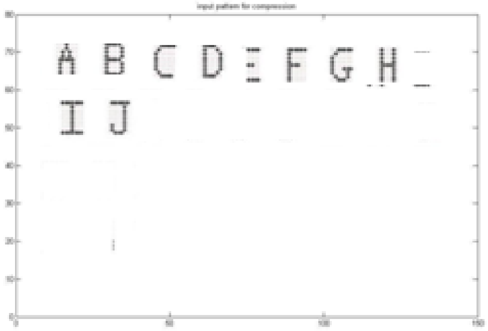


Fig 4.output pattern displayed

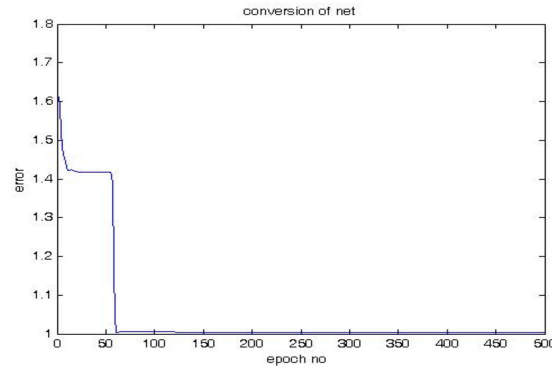


Fig 5. *Learning Rate*

7. Learning rate and momentum

The learning procedure requires that the change in weight is proportional to $\partial E^p / \partial w$.True gradient descent requires that innitesimal steps are taken. The constant of proportionality is the learning rate . For practical purposes we choose a learning rate that is as large as possible without leading to oscillation. One way to avoid oscillation at large , is to make the change in weight dependent of the past weight change by adding a momentum term:

$$\Delta w_{jk}(t+1) = \gamma \delta_k^p y_j^p + \alpha \Delta w_{jk}(t),$$

where t indexes the presentation number and F is a constant which determines the efect of the previous weight change. Although, theoretically, the back-propagation algorithm performs gradient descent on the total error only if the weights are adjusted after the full set of learning patterns has been presented, more often than not the learning rule is applied to each pattern separately, i.e., a pattern p is applied, Ep is calculated, and the weights are adapted (p = 1, 2,..... P). There exists empirical indication that this results in faster convergence. Care has to be taken, however, with the order in which the patterns are taught. For example, when using the same sequence over and over again the network may become focused on the rst few patterns.

Conclusion

In this paper we provide the security by data compression using neural network. It was shown that it is practical to use neural networks for text compression, an application requiring high speed. Among neural models, the best one found combines long and short term learning rates to achieve a balance between using the entire input history and favoring the most recent data to adapt to changing statistics.

References

[1] Fast Text Compression with Neural Networks Source Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference Pages: 230 - 234 Year of Publication: 2000 ISBN:1-57735-113-4 Author: Matthew V. Mahoney

[2] Schmidhuber, Jürgen, and Heil, Stefan 1996, Sequential Neural Text Compression, *IEEE Trans. on Neural Networks*7(1): 142-146.

[3] http://www.learnartificialneuralnetworks.com/