An Alternate Approach for Decoding of Convolutional Codes

Shalini Bahel
Department of Electronics Technology
G.N.D. University, Amritsar

Karan S Narang
Tata Consultancy Services,
Gurgaon

ABSTRACT
Convolutional Codes are used in a variety of areas from computers to communications. Ideally one simply looks at a received message, which may contain errors, and decodes it into the error-free message. Unfortunately, this decoding process can be quite complicated and might not exploit the maximum error correction capabilities of the code. For these reasons neural networks have been widely used as decoders. A neural network approach for decoding of convolutional codes is studied. Here sample neural network uses simple perceptron model with one hidden layer. The training of the neural network is done using Back-propagation. A sequentially programmed Viterbi decoding algorithm is used to generate training patterns for training the neural network decoder. The performance of the trained neural network is compared with Viterbi decoding solution. The comparisons indicate that the neural network approach perform with comparable error-correcting accuracy as the Viterbi decoding algorithm.

General Terms

Neural network, coding, information theory.

Keywords
Convolololutional codes, back propogation, viterbi decoding.

1. INTRODUCTION
Neural networks are proving to be useful in variety of applications and their full potential is far from realization. A neural network can be trained to realize complicated non-linear mappings from the input space to output space and it can be designed to serve as an associative memory capable of correct recall when the input is noisy or incomplete. These features facilitate the use of neural networks for modeling, analyzing, forecasting and optimization.

Neural networks have been used extensively in continuous speech recognition and synthesis, image processing and coding, pattern recognition and classification, power load forecasting, interpretation and prediction of financial trends for stock market etc. The present paper deals with the application of neural networks in the decoding of convolutional codes for error control purpose.

Error control coding is necessary for reliable digital data transmission. The advancement of large-scale, high-speed data networks in military and commercial applications increases the demand of error control and error correction coding. The objective of an error correcting code is to add redundancy to the message in an analytical fashion so that the original message can be recovered if it has been garbled during transmission. The major concern of design is the control of errors to obtain reliable reproduction of data.

In 1967 Viterbi proposed a maximum likelihood scheme that was relatively easy to implement for convolutional codes with small memory order. The Viterbi algorithm is used in this paper to generate training patterns for the neural network decoder. The performance of the trained neural network decoder is compared for different input datasets with Viterbi convolutional decoder [1].

2. CONVOLUTIONAL CODES
In a convolutional code, the block of n code digits generated by the encoder in a particular time unit depends not only on the block of k message digits within that time unit but also on the block of data digits within a previous span of N-1 time units. Convolutional codes can be devised for correcting random errors, burst errors or both. These codes are commonly specified by three parameters: (n,k,K), where n is the number of encoder output bits, k is the number of input bits, and K is constraint length. The constraint length represents the number of bits in the encoder memory that affect the generation of the n output bits. The quantity k/n, called the code rate, is a measure of the efficiency of the code. Throughout this paper the convolutional encoder (2,1,3) shown in figure 1 is used [2].
The encoder introduces redundant bits into the data stream through the use of linear shift registers. The information bits are input into shift registers and the output bits are obtained by modulo-2 addition of the input information bits and the contents of the shift registers. The encoder can be represented in several different but equivalent ways. They are Tree Diagram Representation, State Diagram Representation and Trellis Diagram Representation.

2.1 Tree diagram Representation
The tree diagram representation shows all possible information and encoded sequences for the convolutional encoder. Figure 2 shows the tree diagram representation for encoder of figure 1 for four input bit intervals. In the tree diagram a red line represents input information bit '0' and a blue line represents input information bit '1'. The corresponding output encoded bits are shown on the branches of the tree. An input information sequence defines a specific path through the tree diagram from left to right.

2.2 State Diagram Representation

The state diagram shows the state information of a convolutional encoder. The state information of a convolutional encoder is stored in the shift registers. Figure 3 shows the state diagram of the encoder in figure 1. In the state diagram, the state information of the encoder is shown in the circles. Each new input information bit causes a transition from one state to another. The path information between the states, denoted as x/c, represents input information bit x and output encoded bits c.

2.3 Trellis Diagram Representation

The trellis diagram is basically a redrawing of the state diagram. It shows all possible state transition at each time step. Frequently, a legend accompanies the trellis diagram to show the state transition and the corresponding input and output mappings(x/c). Figure 4 shows the trellis diagram for the encoder in figure 1. In general, the trellis is repeated after depth L is reached. Among the various decoding methods for convolutional codes, Viterbi's maximum likely-hood algorithm is one of the best decoding technique evolved. The Viterbi decoder examines an entire received sequence of a given length. The decoder computes a metric for each path and makes a decision on this metric. All paths are followed until two paths converge on one node. Then the path with the higher metric is kept and the one with lower metric is discarded. The Viterbi algorithm utilizes the trellis diagram to compute the path metric. The decoder advances through the trellis and make decisions by removing the least likely paths [3,4].

For an N bit sequence, the total numbers of possible received sequences are 2^N. Of these only 2^{kL} are valid. The Viterbi algorithm applies the maximum likely-hood principles to limit the comparison to 2 to the power of kL surviving paths instead of checking all paths.

3. NEURAL NETWORKS

The basic aim of Neural Network is to have a machine which has the abilities of both computers and humans. In its most general form, a neural network is a system that is designed to model the way in which the brain performs a particular task or function of interest. The network is usually implemented using electronic components or simulated in software on a digital computer [5]. Neural network systems can be used where

- An algorithmic solution cannot be formulated.
- Lots of examples of the behavior are available.
- There is a need to pick out the structure from existing data.

A neural network is a massively parallel-distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

- Knowledge is acquired by the network through a learning process.
- Inter-neuron connection strength known as synaptic weights are used to store knowledge.

The procedure used to perform the learning process is called a learning algorithm, the function of which is to modify the synaptic weights of the network in an orderly fashion so as to attain a desired design objective.

Back-propagation is a learning algorithm for multilayer feed-forward neural network. The network can be trained to capture the mapping implicit in the given set of input-output patterns pairs. The approach followed is basically a gradient descent along the error surface to arrive at the optimum set of weights. The error is defined as the squared difference between the desired output (i.e., given output pattern) and the actual output obtained at the output layer of the network due to application of an input pattern from the given input-output pattern pair. The output is calculated using the current setting of the weights in all the layers. The optimum weights may be obtained if the weights are adjusted in such a way that the expected error between the desired and the actual output is minimum [6].

4. DESIGN APPROACH

4.1 Neural Network Architecture Selection

Since one of the goals is to improve the speed of the decoding process of convolutional code, it is necessary to find a neural network with an optimum structure to accomplish this task. A simple perceptron with one hidden layer (figure 5) is chosen as the basic structure to keep the network as simple as possible and the back propagation method is used to train the network.

Because the decoding of convolutional code is dependent on the Historical data, the input to the neural network decoder has to contain previously received code words. Even though the output of the neural network should be the decoded result of the given input it should also be able to keep track of previously received code words. In order to achieve this, the neural network maps the most recently received code words, excluding the least recently received code words (the obsolete one) to the output of the network and feeds this information back to the input of the network for decoding the next code word as shown in figure 6.

4.2 Training Pattern Generation

It is essential to train the network with a set of patterns which result in the correct decoding of any sequence of code words. The simplest approach to this task is to utilize the conventional encoding and decoding procedures in the pattern generating process. Training patterns can be generated by passing all possible combinations of data sequences through the encoder to generate the code word sequences which should be received by the decoder without transmission errors.

4.3 Program Description

A set of programs were written for the simulation of neural network convolutional decoder – A training pattern generation program, A network training program and a network decoder program.

The training pattern generation program utilizes the encoding object which is used to generate those patterns which do not contain transmission error. The network training program does the construction, training and storage of the weight matrices file of a network based on the training pattern file. The network is trained by adjusting weights matrices after each pass of the training pattern file. Convergence is achieved when maximum error at the output neurons becomes less than the specified amount given in the training pattern file.

The network decoder program simulates the network described by the weight matrices file which decodes on a given input code word data file. The outputs of the network decoder program can then be used to compare the performance of a neural network decoder with that of a conventional decoder.

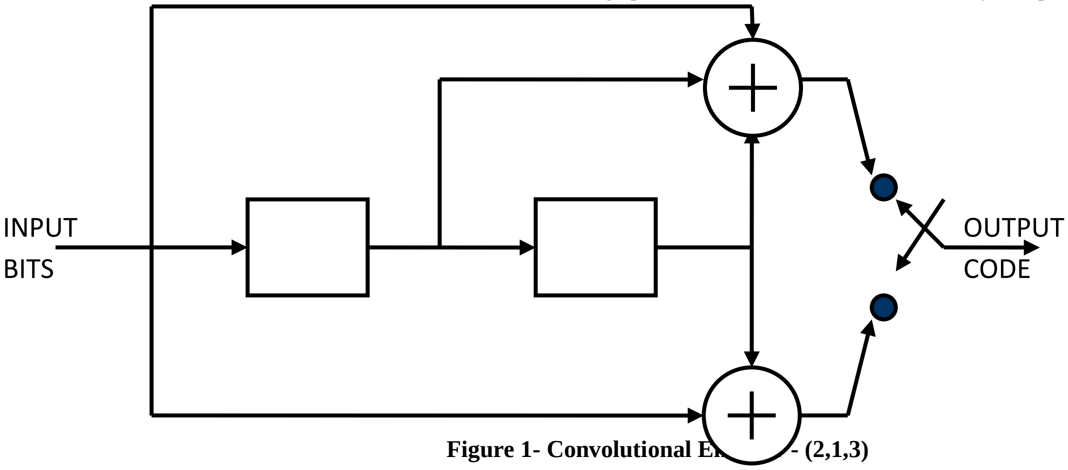

Figure 1- Convolutional En[illegible] - (2,1,3)

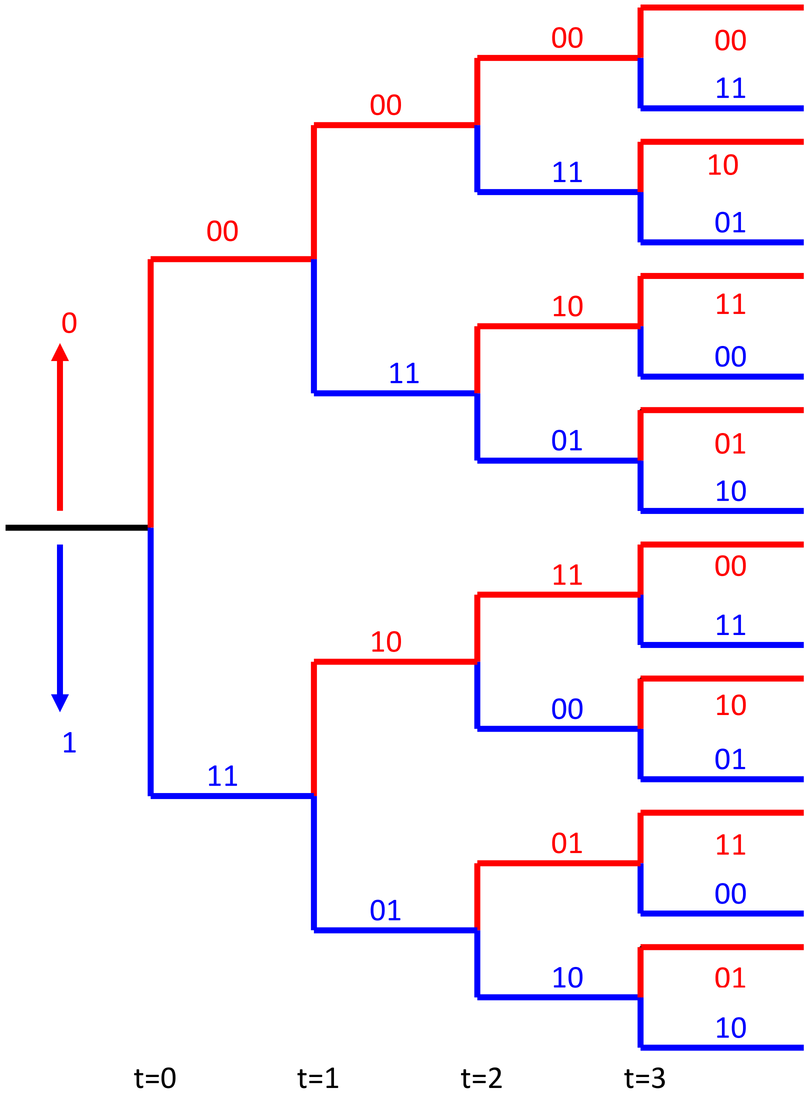

Figure 2 Tree Diagram representation of the encoder shown in figure 1

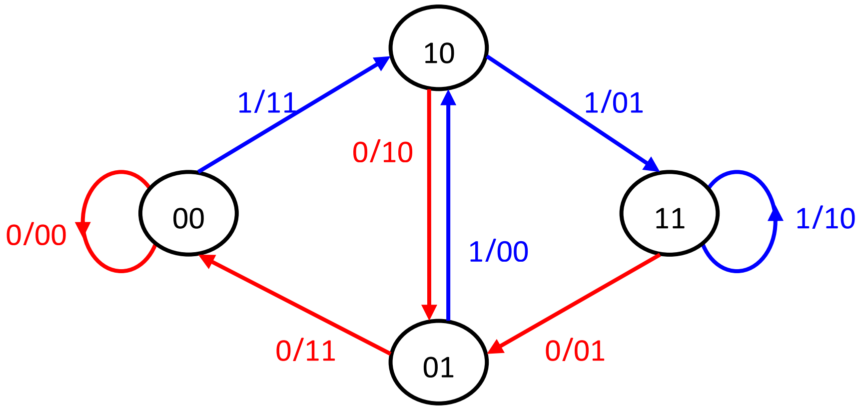

Figure 3 State Diagram representation of the encoder shown in figure 1

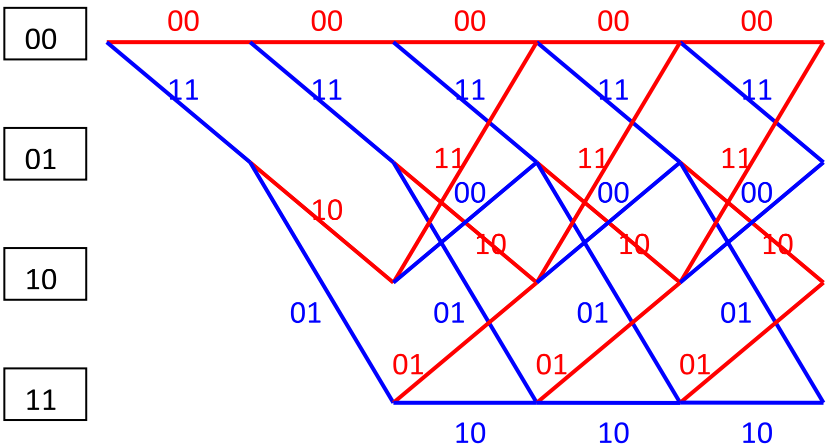

Figure 4 Trellis Diagram representation of the encoder shown in figure 1

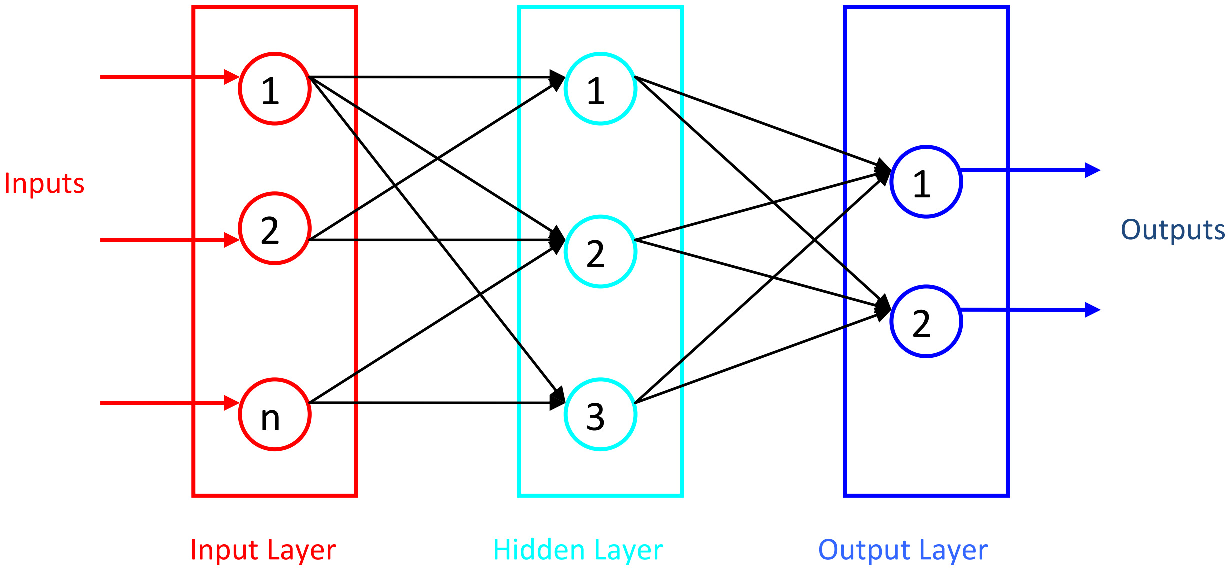

Figure 5 Simple perceptron model

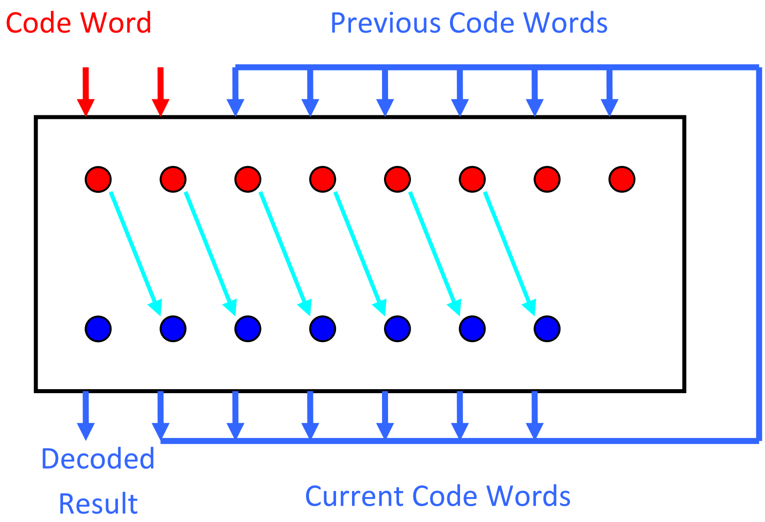

Figure 6 Neural network convolutional decoder

5. CONCLUSIONS

The neural network approach to decoding of convolutional codes has been examined. The decoded code words were compared to the conventional decoder using Viterbi decoding algorithm. Sample neural network is constructed using the simple perceptron with one hidden layer. The training of the neural network is done by Back-propagation. The training patterns were generated to map the most recently received code words to the output which simulates an external register for keeping track of previously received code words.

The comparison of the neural network decoders to the Viterbi decoding algorithm of convolutional encoder (2,1,3) showed comparable error correcting capabilities. The advantages of using neural network decoders are that the network can be updated continuously with new data to optimize its performance at any instant, the networks ability to handle a large number of input variables rapidly and the networks ability to filter noisy data and interpolate incomplete data. The disadvantage of the neural network decoders is the extra time required in training before entering into the performance stage.

6. REFERENCES

[1] Viterbi, A. J. and Omura, J. K. 1979. Principles of Digital Communication and Coding. McGraw-Hill, Inc.

[2] Pless, V. 2003. Introduction to the theory of Error-Correcting Codes. John wiley and sons, Inc.

[3] Bouloutas, A., Hart, G.W. and Schwartz, M. 1991. Two extensions of the Viterbi algorithm. IEEE Transactions on Information Theory. 37[2], 430-436.

[4] Fettweis, G. and Meyr, H. 1989. Parallel Viterbi algorithm implementation: breaking the ACS-bottleneck. IEEE Transactions on Communications. 37[8], 785-790.

[5] Yegnanarayana, B. 2003. Artificial Neural networks. PHI, New-Delhi.

[6] Huang, S.-C. and Huang, Y.-F. 1991. Bounds on the number of hidden neurons in multilayer perceptrons. IEEE Transactions on Neural Networks. 2[1], 47-54.