SECURED COMMUNICATION SYSTEM
WITH NEURAL ENCODER

Mahmoud I. Abdalla

Dept. of Electronic and Electrical Communication, Zagazig Univ., Egypt.

Abstract

A secured communication system with neural encoder/decoder is designed and enhanced by providing an Encryption and Decryption algorithm based on a one-dimensional reversible cellular automata. The capabilities of neural networks are used to design a neural encoder/decoder for secured data communications. The generalised delta rule for feed forward with back-propagation technique is used for designing the neural network. Using computer simulation the performance of the system is evaluated for different signal to noise ratios. The design algorithm of the neural encoder/decoder together with the encryption decryption algorithm is given. Comparisons between the performance of neural decoders and that of several conventional ones are discussed. At probabilities of error of $10^{-4}$ and $10^{-6}$, the neural coding gain is found to be about 5 dB.
1. INTRODUCTION

Digital communication systems have three types of coding namely: source codes, secrecy codes and error control codes (also called channel codes). The source codes include codes used to format the data for specialised modulation. Secrecy codes encrypt information so that the information cannot be understood by anyone except the intended recipient. Channel codes are used to increase its immunity to noise. This is accomplished by inserting controlled redundancy into the transmitted information stream[1] that leads to a decrease in the information transmission rate. Such rate can be increased by using the neural coding. As electric communication replaces written communication, security becomes increasingly important, it must be assured that only the intended receiver can extract the message. An artificial neural network can be trained to recognize a number of patterns. If a version of one of these patterns is corrupted by noise, and then presented to a properly trained network, the network can provide the original pattern on which it was trained[2].

Any task that can be done by traditional discriminant analysis can be done at least as well (and almost always much better) by neural networks[2]. Also neural networks can be used as a prediction tool. Neural networks are non-linear parametric models that can approximate any input-output relation. Coding is to convert M possible messages into M possible codewords. The codewords should be selected to achieve objectives such as efficiency and security. Neural networks can be used to satisfy the function of the encryption and the channel codes to form a neural communication system. For various applications, the generalised delta rule for feed-forward with back-propagation technique can be used for designing the neural network[5,6]. The learning paradigm is summarised in reference[6]. The widest application of this code may be in military communication systems.

2. SYSTEM OVERVIEW

The proposed Secure communication system consists of a transmission Subsystem, the Communication Channel and the receiver Subsystem. The transmission Subsystem consists of the following stages:

1- A private key encryptor to receive a frame 21 bytes in length and to produce the Ciphertext frame with the same length.
2- A Dequeue System to receive the Ciphered frame and present a one nibble at a time to
the Neural encoder.

3- The Neural encoder which maps the input nibble into a code vector.

The receiver will consist of equivalent subsysytems to map the transmitted code vector into
the equivalent message nibble, followed by an Encoding system to form the Ciphered frame
which is then decrypted using the Encryption Key

3. DESIGN TECHNIQUE

3.1 The Encryption/Decryption Systems

In the present design, we use an encryption/decryption algorithm based on Time Reversal
Transformation [7] in One-dimension. The advantages of this algorithm are its
computational simplicity, conservation of information, high performance (can be
implemented as a parallel algorithm) and the possibility of having keys as arithmetical
functions which provides astronomical numbers for the possible key combinations. In ref.
[7], such algorithm has been modelled as N-D Reversible Cellular Automata. In the 1-D
case, a simple rule (key function) can be used to encrypt a Symbol (byte) using its
immediate neighbours.

The encryption/decryption Scheme is designed to receive a frame of symbols U and using
the Time Reversible Transformation[7]:

\[ U_i(t-1) = f(\{ U_i(t) \} \cdot U_i(t-1) \mod 2 \] (1)

with the key function operates on the symbol and its immediate neighbours.

\[ f(\{ U_i(t) \}) = W_0 U_i(t) + W_1 U_i(t+1) + W_2 U_i(t-1) \] (2)

where \( W_0, W_1, W_2 \) are arbitrary key weights. The time evolution of the above scheme is
completely reversible and the entropy of the frame increases linearly with time steps(t).
Notice that the Decryption System will follow the same outlined steps since the algorithm is
completely reversible.
3.2 The Neural Transmitter

Fig. (1) illustrates the neural network that is used to design the neural encoder. Different structures of hidden layers with different nodes were tried. The proper structure is found to be two hidden layers with twenty nodes such that the error can be minimised rapidly. The generalized delta rule formulated by Rumelhart, Hinton, and Williams (1986) [9, 10] is used for training the network.

\[ \text{Two hidden layers} \]

Let \( m = [m_1, m_2, m_3, m_4] \) be the input information bits and \( v = [v_1, v_2, v_3, v_4] \) be the encoded-word; the neural network has to map the vector \( m \) into a vector \( v \) at the transmitting side of the communication system. At the receiver, the neural decoder has to map the vector \( v \) into vector \( m \). Fig. (2) shows the block-diagram of a neural data transmitter. The encoder has 16 possible messages. The code vectors are shown in Table (1). The code vectors have been chosen randomly, however the Hamming distance between any pair is 4. It is to be noted that the transmitted signal is in a form similar to that of PAM, thus any unauthorised receiver will not be able to extract the transmitted information.

The design algorithm of neural encoder can be summarised as follows:

1. Choose the number of code messages to be coded and form the code messages.
a- Neural encoder communication system

b- Neural decoder communication system

Fig. 12 Neural encoder/decoder communication system
2. Form code words of the encoder such that the hamming distance between two code words is as large as possible.

3. Design the neural network architecture with the number of input nodes $N$ such that:

$$M=2^N,$$

where $M$ is the number of codewords. The neural network output layer will also have $N$ nodes.

4. The ciphered nibbles will be the input to the neural network while the associated codewords will be the output.

5. Design the neural network to obtain the weights of the neural network.

6. Implement the neural encoder using software or hardware technique.

<table>
<thead>
<tr>
<th>Code-Vector</th>
<th>Code-Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0 1 5 8</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>0 1 4 2</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>0 1 5 3</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>0 1 3 0</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>1 5 7 2</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>1 2 8 5</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>2 9 5 3</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>2 8 7 3</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>3 1 5 9</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>4 8 6 3</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>4 7 2 4</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>5 8 5 1</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>5 7 3 2</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>6 8 2 5</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>6 7 3 9</td>
</tr>
</tbody>
</table>
3.3 Neural Receiver

The neural decoder is built using the same neural architecture (see Fig. (3)). After adjusting the weights, the network was tested. On the basis of experimental results, it has been considered that if the output is greater than .7, then it will be considered one (1 binary) and if it is less than .5 it will be considered zero (0 binary). The network can easily recognize the inputs and map it into the original message. The network maps the vector $v = \{v_1, v_2, v_3, v_4\}$ into the vector $m = \{m_1, m_2, m_3, m_4\}$. The neural receiver is shown in Fig. (2-b).

The design the decoder follows the same steps for the design the neural encoder expects that the input to the neural network will be the code-word while the output will be the deciphered message.

![Neural Decoder Architecture](image)

**Fig. (3) Neural Decoder Architecture**

3. RESULTS AND CONCLUSION

After adjusting the weights, the network can easily recognize the input and map the input messages to the code-words at the transmitting terminal. Four bits are transmitted at the same time and there is no parity check bit. So the coded-word has the same number of bits as the uncoded one. Hence the rate of information transmitted is the same in the two cases with the same bit duration. The neural network maps the input vector $m = \{m_1, m_2, m_3, m_4\}$ into the code-word vector $v = \{v_1, v_2, v_3, v_4\}$ without mistakes. The vector $v$ can be transmitted using four different carriers (carrier for each information bit). Frequency synthesizers can be used to generate the carrier frequencies required. Frequency hopping techniques can be applied to increase the security as well as obtaining its advantages[11].
Computer simulation is used to investigate the performance of the neural receiver. Since the input bits are transmitted at the same time, we assumed that the input bits are subjected to the same noise level. The neural decoder is trained with 192 input patterns with different noise levels.

The probability of error is given by [12, 13]:

\[
P(e) = P(x > a) = \int_{a}^{\infty} \frac{1}{\sqrt{2\pi \sigma^2}} e^{-\frac{(x - 2\sigma)^2}{2\sigma^2}} dx
\]

where \(\sigma\) is the variance of the random variable \(x\).

\[
P(e) = \frac{1}{\sigma} \left( \frac{1}{\sqrt{2\pi}} \right) e^{-\frac{a^2}{2\sigma^2}} \int_{a}^{\infty} \frac{1}{\sqrt{2\pi \sigma^2}} e^{-\frac{(x - 2\sigma)^2}{2\sigma^2}} dx
\]

The value of the variance \(\sigma^2\) is calculated at different signal to noise ratios.

Using this computer simulation the value of \(a\) is calculated. The value of variable \(a\) [in eqn. (5)] is the maximum noise level at which the performance of the decoder is perfect. The probability of error is obtained using eqn. (5). Fig. (4) shows the probability of error for various SNR for the two cases, namely, the neural system, and the conventional one.

Clearly coded transmission results in low probability of error than uncoded transmission that uses ASK modulation technique overall signals to noise ratios. A comparison between the probability of error for coded transmission using neural code and coded transmission using block-code (4, 7) is given in Fig. (5). Clearly the probability of error in the case of neural coding is lower than that of block-code.

Table (2) gives the coding gain for various coding techniques when referred to binary PSK transmission system for the two values of \(P(e)\), namely, \(P(e) = 10^{-5}\) and \(10^{-8}\) [14].

The coding gain is used as the basis of comparison. The coding gain refers to the number of dB that the signal to noise ratio can be reduced from the value required when there are no coding and still provide the same bit error rate. This means that the coding gain \(G\) is:
Fig. (4) Probability of error for neural decoder and uncoded using ASK modulation technique.

- neural coding
- uncoded ASK

Fig. (5) Comparison between the probability of error of neural coding and block coding.

- neural coding
- block coding
The neural communication system is a secret code with high rate of information and a low probability of error with coding gain of 5 dB at $\rho_e=10^{-5}$ and $10^{-8}$. The system parameters were changed step by step and test the neural network by the input code vectors. The tolerance of the input weights is found to be 5.5% for the input and output weights while it is 4.5% for the weights between the two hidden layers.

Table (2) Comparison of coding techniques on a Gaussian channel

<table>
<thead>
<tr>
<th>coding technique</th>
<th>coding gain</th>
<th>$\rho_e=10^{-5}$</th>
<th>$\rho_e=10^{-8}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural codes</td>
<td></td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Block codes (hard decision)</td>
<td></td>
<td>3.4</td>
<td>4.5 - 5.5</td>
</tr>
<tr>
<td>Convolutional coding with sequential decoding</td>
<td></td>
<td>4.5 - 5</td>
<td>5 - 7</td>
</tr>
<tr>
<td>Convolutional coding with viterbi decoding</td>
<td></td>
<td>4.5 - 5</td>
<td>5 - 6.5</td>
</tr>
</tbody>
</table>

4-REFERENCES


