Decryption of Chaotically Encrypted Signals Using Neural Networks

Craig Tainter

Faculty Sponsor: Jeffrey Baggett, Department of Mathematics

ABSTRACT

A possible data encryption scheme is to add a small amplitude message to a larger amplitude chaotic carrier signal. The primary objective of this research was to see if a neural network could approximately recover the hidden message if was given the intercepted signal. It was found that even with a reasonably complex carrier signal, the message was, to an extent, recoverable. The remainder of the research focused in improving and quantifying the quality of the approximately recovered message.

INTRODUCTION

Suppose you wanted to send a secret message to someone else. The person receiving the message would have to know how the secret message was created so they could decrypt the content. When a secret message is being encrypted, the encryption should be as complex as possible so that if anyone were to intercept the message, they would not be able to decrypt it.

In mathematics, chaos is characterized by long term non-periodic behavior in a deterministic system which exhibits sensitive dependence on initial conditions. Two chaotic signals that are initially very similar will eventually diverge from each other. This makes the exact long term behavior of a chaotic signal difficult to predict, though the overall pattern of the signal may be quite easily understood. A good example is the weather, we know in advance that the overall pattern will include four seasons, but we cannot possibly predict whether or not it will rain on a particular day three weeks from now.

Because of this dependence on initial conditions, a chaotic signal is a perfect candidate for the carrier signal. A carrier signal is what is added to the message being sent to "hide" the message. The person receiving the message would know the exact initial conditions of the chaotic carrier and could exactly replicate it. However, if a person was to intercept this signal, it should be nearly impossible to guess the initial conditions exactly correct making it nearly impossible to figure out the carrier signal.

Edward Lorenz was a meteorologist and mathematician from MIT. In 1963, he published a paper called, "Deterministic Non-Periodic Flows" in which he introduced the Lorenz Equations. They were originally used to model some of the unpredictable behavior which we normally associate with the weather. The Lorenz equations produce chaotic solutions…they have sensitive dependence on initial conditions. Since the Lorenz equations are widely studied, they were used to generate the chaotic carrier signals. The Lorenz equations are the system of differential equations found in Figure 1. A three dimensional solution can be seen in Figure 2a. Figure 2b shows a plot of each of the coordinates plotted against time.

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = x(r - z) - y$$
$$\dot{z} = xy - \beta z$$

Figure 1: The Lorenz Equations. σ, r, and β are positive constants.

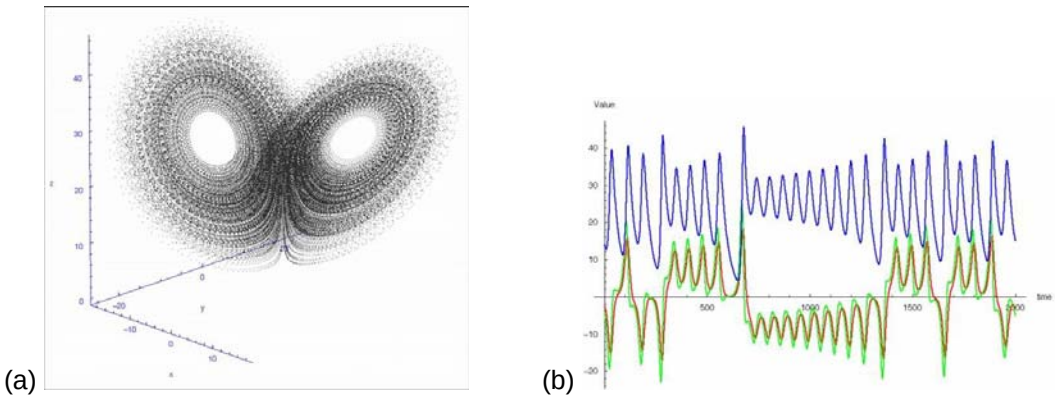

Figure 2: A solution to the Lorenz equations with r = 28, σ = 10, β = 8/3. The left figure (a) shows a three dimension view of the solution, while the right figure (b) shows the x (red), y (green), and z (blue) coordinates of the solution versus time.

Figure 3 shows the general scheme for how chaotic encryption would work. In this example, the "carrier" signal is a sine curve and the message is just a randomly generated signal. Here you can tell just by looking at the graph that there is something extra added in. If this were to be actually used, the amplitude of the message would be so small compared to that of the carrier that the graphs of the carrier and the carrier+message would look identical to the naked eye.

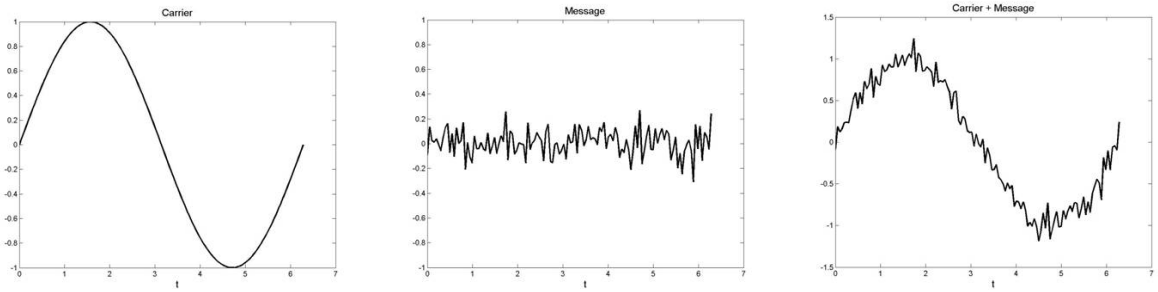

Figure 3: The carrier and the message are added together to give the carrier+message signal which is the signal that would be sent.

Neural networks are complex mathematical functions that can be used for pattern recognition and time series prediction. In this work we are particularly interested in time series prediction. A time series is a sequence of vectors, $\mathbf{x}(t), t = 0,1,...,$ where t represents elapsed time. For simplicity, we will discuss the case in which the sequence is of scalar values, thought the techniques generalize easily to vector sequences. Theoretically, x may be any value which varies continuously with time t, such as temperature. In practice, x will be sampled at regular time intervals to give a series of discrete data points. Work in neural networks has focused on forecasting future development of the time series from values of x up to the current time. Formally, this can be stated as: find a function $f : R^N \rightarrow R$ to obtain an estimate of x at time $t+1$, from the data N time steps back from time t, so that: $x(t+1) = f(x(t), x(t-1),..., x(t-N+1))$. See Figure 4 for a graphical view of this neural network.

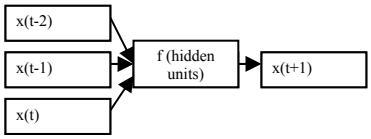

Figure 4: The standard method of performing time series prediction using a sliding window of, in this case, three time steps.

Details about the structure of the approximating function f (in our case, as is standard, f is a linear combination of hyperbolic tangent functions) can be found in many textbooks on neural networks. Throughout this work, we have used the Neural Network System Identification Toolbox (NNSYSID) software (Nørgaard, Ravn, Poulsen and Hansen, 2000) to find the approximating function f.

Because neural networks have been shown to be capable of learning and reproducing chaotic signals (Frank, Davey, and Hunt, 2001) our idea is to use a neural network to identify the carrier signal in an encrypted message. While no neural network can exactly predict the future behavior of a chaotic signal, a neural network can be used to find the overall pattern of a chaotic signal. In our case, we are not interested in predicting the future behavior of a chaotic signal, but instead we have a signal which is composed of two parts, a chaotic carrier signal and a message and we wish to separate the two. In this work, we assume that the message signal is essentially random noise without structure. The message signal can be regarded as high frequency noise and the chaotic carrier signal is a low frequency signal (varies slowly in time – see Figure 2b). The neural network attempts to identify or "learn" the chaotic part of the encrypted signal. Thus the output of the neural network is a predicted version of the carrier signal. In this way, the neural network can be thought of as a "low pass" filter – slowly varying, low frequency parts of the signal are passed through, while quickly varying, high frequency parts of the signal are removed or filtered out. Using the approximated carrier signal that is predicted from the neural network, we can subtract it from the encrypted signal to get an approximation to the original message.

METHOD

A lot of this research was trial and error. It is difficult to understand and especially hard to predict how well a certain neural network will be able to predict the carrier signal. For this reason, much of the first portion of the research was simply trying different structures of neural networks (such as how many past input points were used to predict the next point) to see what worked the best. It seemed there were certain network structures that were better than others. Obviously, better results were obtained the more inputs that were used. For example, if only one past input was given to the neural network, it did a poor job of predicting the next point. If you gave it say five points, however, it did a much better job of predicting the next point. The more past input points that were used, the more complex the structure of the network. Increasing the number of past inputs indefinitely eventually results in a network that does not predict the carrier signal as well. It was found that 5-8 past inputs yielded neural networks that best predicted the chaotic carrier signal. As we show in the Results section below, the approximately decrypted message was of medium quality using this approach.

In an attempt to understand why the neural network was not doing a better job of approximating the chaotic carrier signal, we conducted some experiments in which the chaotic carrier signal was allowed to have multiple components. At each point in time, the carrier signal consisted of a vector of data from two or three coordinates of the chaotic signal generated by the Lorenz equations (see Figure 3). This does not correspond to a real encryption scheme in which the carrier signal would consist of a single chaotic signal, rather this was a scientific exercise to see how well the neural network could approximate the chaotic carrier signal if the time series contains more information about the chaos. In some cases, as is shown below, the neural network is able to nearly perfectly recover the encrypted message when the carrier signal is multi-dimensional.

Another direction in this research was to see how the content of the message affects the ability of the neural network approach to decrypting it. Two kinds of messages were used, the first was a random message in which the message consisted of random numbers chosen from a normal distribution, the second type of message was an actual digital audio signal. The digital audio signal has some structure (it is, after all, sampled sound waves), but the sound oscillations are of much higher frequency (shorter time scales) than the oscillations of the chaotic carrier signal. In spite of the seemingly poor performance of the neural network at approximating the chaotic carrier signal and decrypting the message, the digital audio message was recovered and intelligible. Reasons for this discrepancy are discussed below.

RESULTS

For the chaotic carrier signal, the Lorenz equations were simulated (as in Figure 2b) and the x-coordinate was saved at time intervals of 0.125. To measure the amplitudes of the signals, standard deviation was used as a measure of a typical oscillation in the signal.

The first portion of this research was done using a message consisting of normally distributed random numbers with zero mean. The standard deviation of the carrier signal is about 7.7 while the standard deviation of the random message is normalized to be about 0.031. Thus the message to carrier signal ratio was 0.4%. The time series

consists of 1600 numbers. Typical results of a neural network approximation to the message are shown in Figure 6 below. In this case, the past five values in the time series are used to predict the next one.

To measure the error in the approximation of the message, two quantities were measured. The first is Pearson's correlation coefficient, r, between the approximated message and the actual message. Two signals are perfectly correlated if $r = +1$ or $(r = -1)$. The second was the square of the relative error, $error = \sum (x(t) - y(t))^2 / \sum x(t)^2$. If two graphs are identical, $x(t) - y(t)$ will be zero so the numerator of this fraction will be zero and the error will be zero.

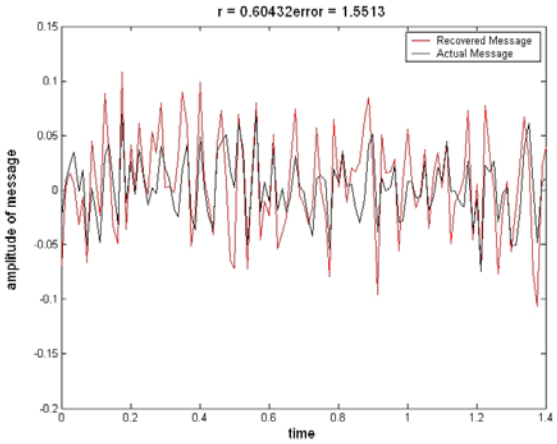

Figure 5: A typical neural network result with a randomly generated signal. The black line is the actual message and the red line is the predicted message. Here, r = 0.60432 and error = 1.5513. Only the first 100 values in the time series are shown.

The neural network approximation to the message shown in Figure 5 indicates that the recovered message has some of the same features as the original message, but the correlation coefficient value of $r \approx .60$ is not especially high.

The neural network functions as a low-pass filter that removes the high frequency components from the encrypted signal (which consists of carrier+message). A simpler low-pass filter is to take a "sliding" average of neighboring points in the time series. We call this the "average method". To get the averaged time series, in this case the predicted carrier signal, we use $\bar{x}(t) = (x(t-1) + x(t) + x(t+1))/3$. The average method filters out high frequency components from the encrypted signal to get a prediction of the carrier signal as shown in Figure 6.

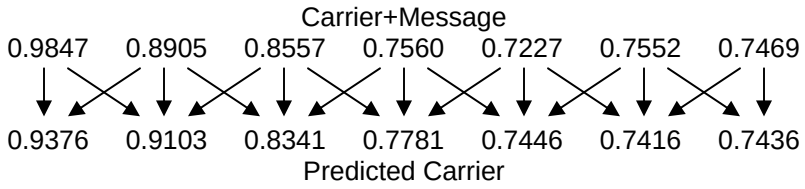

Figure 6: The top set of data is the carrier+message. To get the predicted carrier, an average is taken of the point and the point on either side of it.

Using the average method to predict the carrier signal and recover the encrypted message gives surprisingly good results that are comparable to those obtained by the neural network approach; see Figure 7.

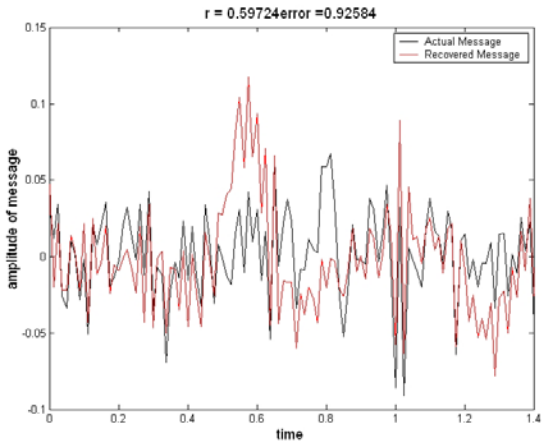

Figure 7: A typical average method result with a randomly generated signal. The black line is the actual message and the red line is the predicted message. Here, r = 0.59724 and error = 0.92584.

Since neural networks are usually good at predicting chaotic behavior, it was a bit surprising that the neural network message recovery did not work appreciably better than the far simpler average method. We hypothesized that the neural network was unable to learn enough about the chaotic signal from the one-coordinate time-series to completely predict the chaotic carrier signal. As an experiment we tried allowing each value of the carrier signal to be not just a simple scalar, but rather a vector coordinates from the simulated Lorenz equations. It was found that if each element in the time series was a vector consisting of both the x- and y-coordinate values (see Figure 2b) and that if the message to be hidden was added to the x-coordinate, then the neural network approximation to the message was nearly perfect; see Figure 8. Again, the past 5 elements in the time series were used to predict the next element.

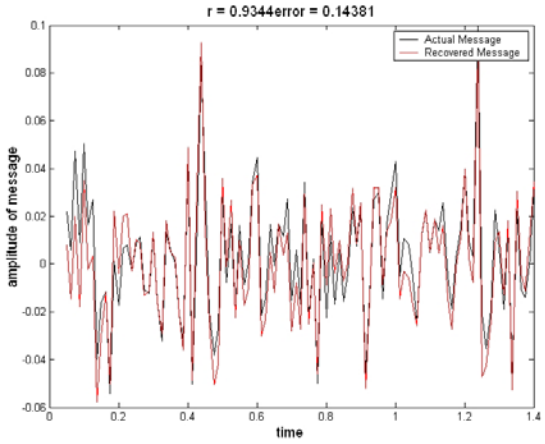

Figure 8: A typical result when the neural network was given both the x and y coordinates with the message only added to the x coordinate. The black line is the actual message and the red line is the predicted message. Here, r = 0.9344 and error = 0.14381.

To investigate how the ability of the neural network to decrypt a message depends on the structure of the message, digital audio clips were encrypted by adding them to a chaotic carrier signal as described above. Digital audio, in its simplest form, is a time series of sampled values that describe oscillations at different frequencies which we hear as different pitches. Again, the message was added to a time series of chaotic values from the x-coordinate of the Lorenz equations and a neural network, using 5 past values, was trained to approximate the carrier signal. Results are shown in Figure 9. The correlation coefficient of $r \approx .21$ is significantly lower than when the message

was randomly generated. The squared relative error is also lower. This would seem to indicate that the neural network is doing a poor job of recovering the digital audio message. However, the approximated message is in fact clearly audible with only a slight buzzing noise.

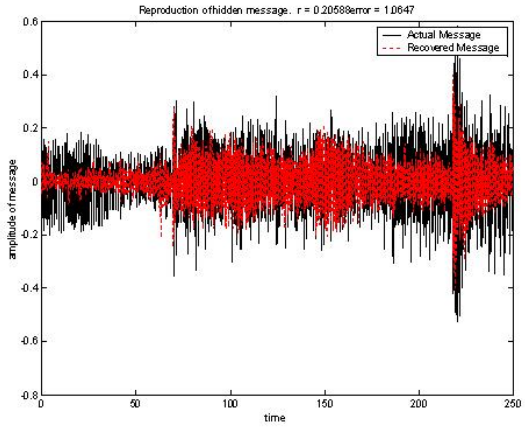

Figure 9: Actual audio results from the neural network.
The black line is the actual message and the red line is the predicted message.
Here, r = 0.20588 and error = 1.0647.

For complete comparison, the average method was employed to approximately recover the encrypted digital audio clip. Surprisingly, the correlation improves to $r \approx .33$ and the squared relative error decreases to .91. However the quality of the sound recovered from the encrypted digital audio using the average method is much less than the sound quality from the neural network. This indicates that the quantities we are using to quantify the quality of the message approximation do not reflect the right things. The recovered message can be poorly correlated with the actual message yielding a low correlation coefficient and yet an audio message is still perfectly intelligible.

To analyze the performance of the neural network in decrypting an audio message, power spectra were used. Power spectra are obtained using the Fourier Transform. They are used in analyzing audio message. The power spectra itself is a graph of all the frequencies (measured in Hz – vibrations per second) on the x axis, and the power (amplitude) of each frequency is on the y axis. Therefore, an audio signal that contains a lot of a certain frequency would have a high peak at the corresponding frequency whereas a frequency that was not as prevalent would have a small peak at its respective frequency. The power spectra for the original message and neural network decrypted message are shown in Figure 10.

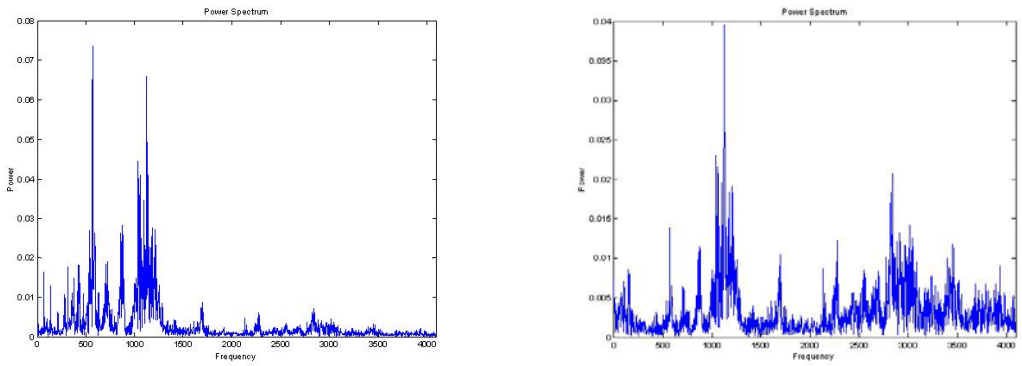

Figure 10: Power spectrum of original message (left) and power spectrum of decrypted message (right).

The shape of the power spectrum of the decrypted message agrees with that of the original message at frequencies from approximately 700 Hz to 1700 Hz. Human hearing is most acute at frequencies between 500 Hz

and 2000 Hz. Thus the neural network does a good job of decrypting the important frequencies in the audio message. Notice however, that the neural network decrypted message has a lot of "noise" at frequencies above 2000 Hz that was not in the original audio message. This extra noise is, at least partially, responsible for the low correlation coefficient between the original and decrypted messages. By comparing only the content of the original and decrypted messages between 500 Hz and 2000 Hz, it was found that the correlation coefficient for these "cleaned" messages was 0.57 (while being only 0.44 for the average method).

CONCLUSIONS

More research should be done to understand how well a neural network can decrypt messages hidden by chaotic encryption. Mathematically, it is known that chaotic signals can be reconstructed from time series, but there is less work on how this is to be done in practice. This work shows that the ability of the neural network to decrypt messages will depend on the structure of the message. Furthermore, simple measures of the quality of the decrypted message such as the correlation coefficient and the squared relative error fail to reflect ability of the neural network to decrypt the hidden message – the neural network did a good job of decrypting an audio message in spite of the low mathematical correlation with the original message.

ACKNOWLEDGEMENTS

I would like to thank the UW-L Undergraduate Research Grants program for the funding of this project. I would also like to thank Dr. Jeff Baggett for all the extra time he put into this project helping me with everything.

REFERENCES

Frank RJ, Davey N & Hunt SP. Time Series Prediction and Neural Networks, *Journal of Intelligent and Robotic Systems* 31:91-103, 2001.

Nørgaard M, Ravn O, Poulsen NK, & Hansen LK. Neural Networks for Modeling and Control of Dynamic Systems, Springer-Verlag, London, 2000.