NEURAL NETWORK APPLICATION TO TELEMETRY FRAME SYNCHRONIZATION

David E. Massey
Goddard Space Flight Center
Wallops Flight Facility
Wallops, Va 23337

ABSTRACT

This paper looks into the use of neural network software as applied to the classical signal to noise concern when dealing with space to ground data communications. Use of a digital neural network to extend the correlation range of Pulse Code Modulation (PCM) down into noise is investigated. Conventional synchronization pattern correlation is done with digital logic comparisons on a sliding window with a set number of bit mismatch errors allowed. Correlation with a neural network does pattern recognition with a weighted network of artificial neurons that have been trained to recognize the sync pattern within noise. The output of such a neural network will produce a best guess of the correct pattern.

INTRODUCTION

The initial results of this work deals with the application of neural network simulation software to sounding rocket telemetry data in the non-realtime. As rocket payloads with telemetry systems go to higher and higher altitudes the RF link margins become less and less and more noise is introduced into the received data at the ground station. Some of the techniques in use today to solve this problem are; Higher power transmitters, larger ground station antenna’s, modulation methods, and data encoding methods (convolutional etc). This paper investigates the use of neural networks to do frame sync pattern correlation. This pattern recognition is done on the frame sync pattern in the telemetry frame, therefore the data fields still could contain noise, but frame synchronization will be maintained longer than that of a conventional frame synchronizer. Until neural network hardware evolves to the point where this technique can be used in realtime, the data must be played back in a reduced rate in the non-realtime into a computer for this neural network pattern recognition to be accomplished.

PCM TELEMETRY 101

A quick overview of Pulse Code Modulation (PCM) telemetry is in order here to illustrate how noise will effect the received data at the ground station. At the output of the radio receiver on the ground, the data is presented in a one/zero (high or low) format. A predefined number of these ones and zeros make up a word of data (typically 8 or 10 bits). A predetermined number of words make up a frame of data. A predetermined number of frames make up a major frame, but this paper only deals with data at the frame level. Within this frame of data there exist a frame sync pattern that will remain the same for all frames transmitted (there does exist other types of frame sync patterns that do change, but those are beyond the scope of this paper) this frame sync pattern will typically be two or three words in size and be located in the same place within the frame. Figure #1 shows three frames of data with the frame sync pattern FA F3 20 (hexadecimal) being the frame sync pattern.

FA F3 20 01 34 87 45 99 DE 2A 12 BB CC 3E
FA F3 20 02 88 67 23 BC AD E3 71 90 34 12
FA F3 20 03 45 02 01 86 9D B2 C7 E5 2A 92

FIGURE #1 Typical PCM frames

This example shows a frame being made up of 14 words each being 8 bits in length. These frames of data will just be stacked together one after another. In the presence of noise on the communication link, some of these bits will be altered from their original value. This causes bit errors in the system, if the error is in the data portion of the frame ‘Lock’ (Lock refers to the state of knowing the bit boundary of the words and where the frame sync pattern is) is not lost, but if errors appear in the frame sync pattern FA F3 20 then lock may be lost. Current ground station frame synchronizers employ several techniques to deal with bit errors in the sync pattern. The first would be to just allow a certain number of errors before the sync pattern is called no good, but this increases the chances of false synchronization. Another method would be to allow bit slippage, which would allow the sync pattern to be found intact but not exactly where it is suppose to be. If frame lock is lost then the synchronizer must begin to search for it again, and in the presence of noise this is difficult. While searching for the sync pattern much valuable data is lost.

NEURAL NETWORK MODELING

A neural network consist of artificial neurons connected together with outputs routed to inputs in successive layers (See figure #2). A neuron can be thought of as a switch with one or many inputs and one output. The output is triggered when the inputs weighted by some value exceeds a learned threshold. The outputs of these neurons will then go into inputs of other neurons in other layers. Several types of neural topology were investigated, backpropagation, adaline, madaline, and hopfield. These four types of networks were modeled but the backpropagation neural network was selected to be used throughout this investigation (Figure #3 shows the actual Backpropagation neural network model used. Note the 24 bit inputs to the left and the single output to the right; see reference #1). In this case the network was modeled to correlate the telemetry frame in figure #1, which has 24 frame synchronization bits as the input to the neural network. The output is simply a one or zero to indicate a sync match or not. These neural networks can be realized in hardware or in software. For the purpose of this investigation a software neural network is used.

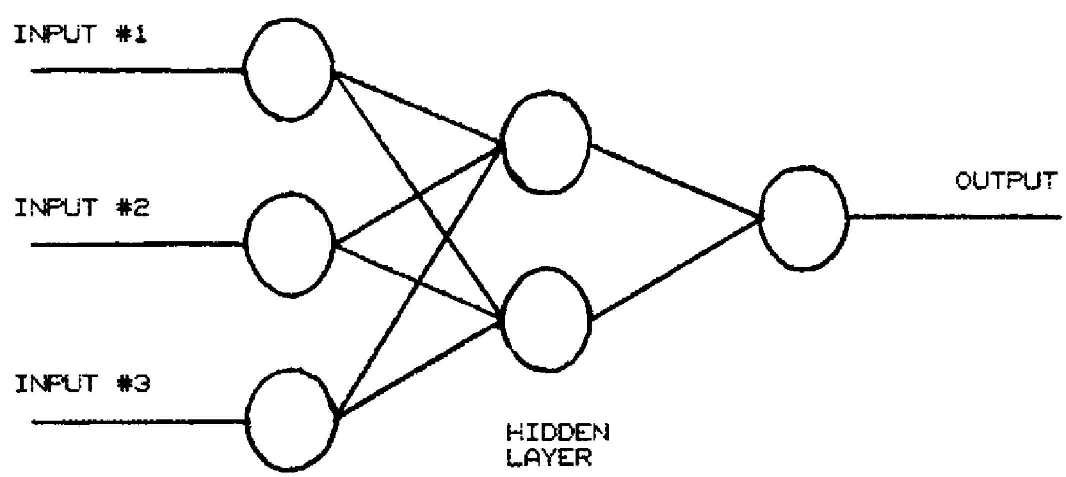

FIGURE # 2 Basic Neural Network

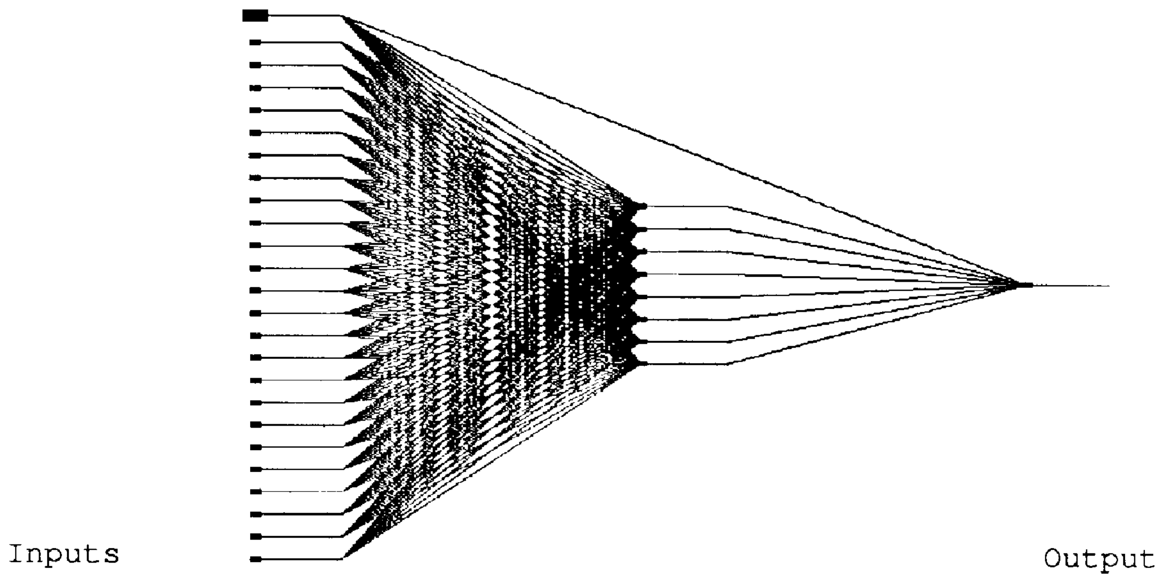

FIGURE #3 Software implemented Backpropagation network

NEURAL NETWORK TRAINING

A neural network can not be used until it is trained (educated) to recognize the sync pattern. Training consist of playing several hundred variations of the sync patterns into it. Of these sync patterns some must be correct and some must have varying amounts of noise in them (white noise) . This training file is created by using a PCM simulator (setup to simulate the space to ground link) to modulate an RF generator which is received by a telemetry receiver (figure #4 shows the serial data capture block diagram). The data is then recorded (by a bit serial recording board in a personal computer; see reference #2) at various RF levels to create noise within the data. The frame sync patterns are then extracted and put into a file that can be played into the software neural network for training (figure #5 shows an excerpt of data that the training file was made from, this data is in hexadecimal format the highlighted area’s are the frame sync patterns that are used to train the neural network).

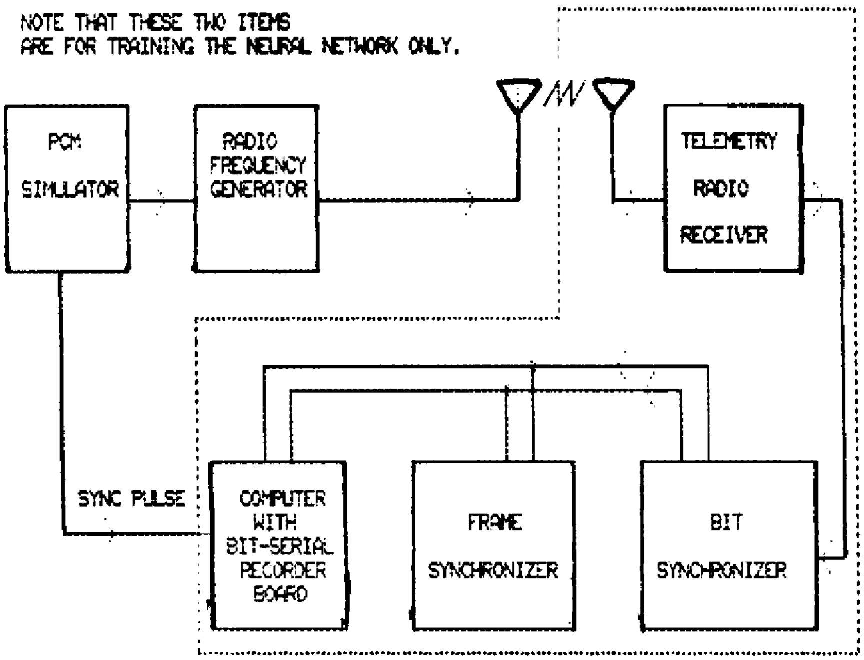

FIGURE #4 Block diagram of serial data capture

AAAAAAAA**BD7990**05AABEAAAAAAAA
AAAAAAAA**AD7990**062ABB55555555
55555555**FAF320**0D5556D5555555
55555555**FAF320**0E555662AAAAAA
AAAAAAAB**F5E640**1EAAAAAA00AAAA
AAAAAAAB**F5E640**00AAAAB1555555
55555557**EBCC80**05555505555555
55555555**EBCC80**09555520AAAAAA
AAAAAAA**BD7990**01AAAAAAAA2AAAA
AAAAAAA**FD7990**022AAA855555555
5555555**FAF320**055155555555555
5555555**BAF320**0655540AAAAAAAA
AAAAAAA**F5E640**0EAAA0AAAAAAAAA
AAAAAAB**75E640**10AAAA155555555
5555557**FAF320**25555055555555
5555555**EBCC80**295506AAAAAAAAA

FIGURE #5 Actual noise embedded training data file excerpt
Frame sync pattern used to train is in bold.

SYNC CORRELATION

To actually run the neural network software on the actual data, the telemetry data must be played into the computer (as in figure #4). This is done by the same serial bit recording board that was used to create the training file; see reference #3. The telemetry data is then recorded into a disk file. A program to shift bits (sliding windows 24 bits each) at a time is run on the recorded file to prepare it for the neural network. This disk file is then presented to the neural network for sync pattern recognition (Figure #6 shows a graphical block diagram of the way the data is handled and presented to the neural network, note that this process is done entirely in software once the serial data has been captured into a file). The neural network will then trigger its output whenever it sees a sync pattern (even within a certain level of noise) . This then establishes the boundaries of the words and frames. The data in the frame can then be extracted (even though there will be some noise in this data it may still be very useful).

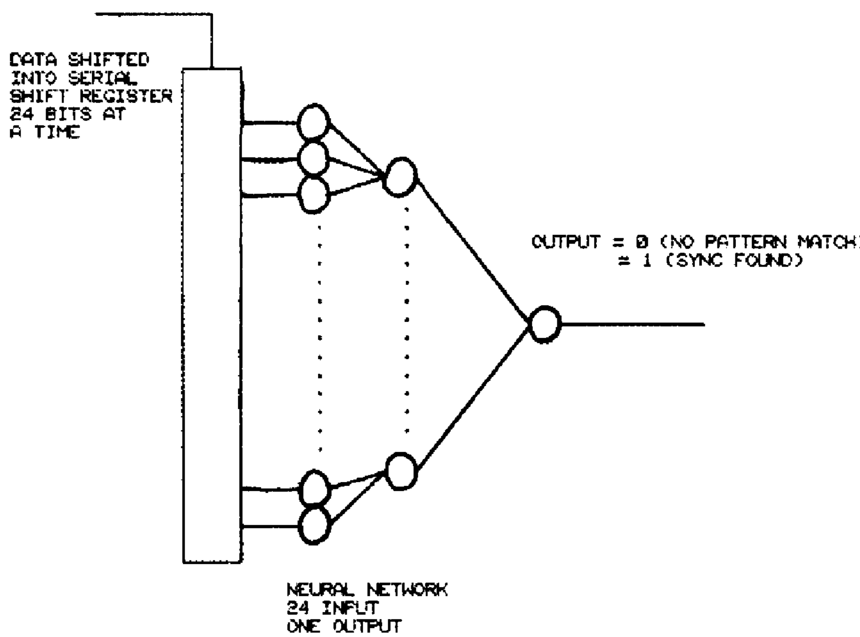

FIGURE #6 Block diagram of data presentation to the neural network

CONCLUSION

Results from this investigation indicate that the neural network software was able to extract correct frame synchronization much longer than that of a standard frame synchronizers. The cost of this improvement is that much more effort must be expended in post processing. This paper is a preliminary work on this subject and only scratches the surface, but this method does seem to show some promise for extracting valuable information from a noisy radio frequency data link. The data may still contain some noise but the boundaries (data is synchronized) of the data words are known and this is useful even if one half of the data is noisy.

ACKNOWLEDGEMENT

The author wishes to thank Dr. Andrew Veronis for his support and guidance.

REFERENCES:

1. Wasserman, Philip D. , “Neural computing, Theory and practice”, ISBN 0-442-20743-3, Van Nostrand Reinhold, 1989.

2 . Massey, David E. “Pulse Code Modulation (PCM) data storage and analysis using a Microcomputer”, American Institute of Aeronautics and Astronautics (AIAA) 7th annual conference, 1986.

3. Lawrence, Jeannette, “Introduction to Neural Networks”, California Scientific Software, 1991.