Sunspot Prediction Using Neural Networks

James Villarreal and Paul Baffes
Artificial Intelligence Section/FM72
Mission Planning and Analysis Division
National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas 77058

Introductlon

The earliest systematic observance of sunspot activity is known to have been discovered by the Chinese in 1382 during the Ming Dynasty (1368 - 1644) when spots on the sun were noticed by looking at the sun through thick, forest fire smoke. Not until after the 18th century did sunspot levels become more than a source of wonderment and curiosity. Since 1834 reliable sunspot data has been collected by the National Oceanic and Atmospheric Administration (NOAA) and the U. S. Naval Observatory. Recently, considerable effort has been placed upon the study of the effects of sunspots on the ecosystem and the space environment. This chapter describes the efforts of the Artificial Intelligence Section of the Mission Planning and Analysis Division of the Johnson Space Center involving the prediction of sunspot activity using neural network technologies.

Sunspots

A sunspot is a dark region on the solar disk, indicative of a 2000°K cooler area than the normal photospheric temperature. On the average, sunspots are about 37,000 km in diameter (for comparison, recall that the earth's diameter is 12,740 km) with exceptionally large spots having a diameter of 245,000 km. Essentially, a sunspot is an eruption of a magnetic energy field extending several miles beyond the sun's surface with an accompanying sunspot of reversed polarity acting as a sink.

Therefore, sunspots are a basic measure of solar activity -- the more sunspots, the more active the Sun. Associated with these moments of high activity are increased occurrences of solar flares, which are bursts of electromagnetic energy. An eruption of a solar flare is accompanied by electromagnetic emissions in the microwave radio frequency range. The larger solar flares may emit relativistic charged particles and energetic protons.

The ability to predict sunspot activity plays an increasingly important role in both earth and space endeavors. Among the significant effects of sunspots are: x-ray emissions, energetic photons, ozone density fluctuations, solar wind variations, rainfall and temperature changes, and disturbances of the earth's geomagnetic fields. Such phenomena are important to NASA because of their adverse effect upon space environment. For example, x-ray emissions can disrupt radio communications by altering the electron density in the earth's ionosphere. Communication signals transmitted from radio stations are either refracted or reflected by the earth's atmosphere and returned to receiving stations. The electron density of the atmosphere, called "skin depth," determines the effects of the atmosphere on radio signals -- shorter wavelengths pass through the ionosphere whereas longer wavelengths are reflected. Consequently, any change in ionosphere electron density will disrupt radio communications and may even necessitate changes in transmission paths of navigation signals.

Other adverse effects by sunspots upon the space environment include the release or increased activity of energetic protons closely related to sunspot frequency. Such energetic protons can damage electronic components within satellites. Additionally, sunspots can cause fluctuations of the geomagnetic field resulting in heating of the earth's upper atmosphere. This causes increased drag on space structures and satellites, and complicates predictions of satellite orbits. In fact, increased atmospheric drag due to sunspot activity was the chief cause for the earlier than expected destruction of Skylab in the late 1970s.

Backpropagation Networks

Choosing an appropriate method for sunspot prediction requires a careful analysis of the desired output and the characteristics of the available data. The crux of the problem is to forecast future sunspot activity given a "window," or partial history, of past sunspot measurements. Because a *prediction* is being made, no traditional algorithm will suffice since future events will never exactly duplicate the past. In other words, a simple review of historical data will not work. One must be able to *generalize* from past measurements to have any hope of producing a hypothesis meaningful to an event which has yet to occur. Furthermore, a large amount of data has already been collected which can be brought to bear on the problem. It seems only logical to use as much of that data as possible to bolster the efficacy of the generated results.

In short, the solution method used must be able to digest the available data into patterns which have some significance to forecasting. One neural network paradigm in particular, called the "generalized delta rule" or "backpropagation," fulfills all of the requirements outlined above. Given large sets of input data, backpropagation networks can be used to categorize input patterns *never before presented* to the network. This categorization is not a simple lookup of past events but the result of generalization on the input data based upon a blending of its various features. To show how this can be accomplished, it is instructive to understand the general structure of a backpropagation network.

Background

As the name implies, artificial neural networks are based on concepts borrowed from biological nervous systems. Anatomical evidence of the nervous system indicates that single neurons are highly interconnected to other neurons with which they communicate through the release of variable amounts of neurotransmitters at the synapse. By modelling these properties in computer systems where interconnections are highly distributed and each element is treated as an individual parallel processor, interesting and useful properties have surfaced. Artificial neural networks have the unique property of being able to automatically extract and develop internal features from a given data set and to form generalities from those learned features.

Highlighting the mechanics of artificial neural networks may best be done by comparing the differences between artificial neural networks and the conventional computer system. Conventional computer systems generally consist of a centralized processing unit and an addressable memory. The central processing unit accesses locations of memory where information can be stored or retrieved. This structure is analogous to a postman who stuffs letters into mailboxes. Artificial neural networks, on the other hand, consist of numerous simple processing elements which are highly interconnected. It is in the *connections between* processing elements and *not* in the processing elements themselves where information in a neural network lies. Therefore, a network's memory is not stored in discrete locations as with a conventional computer. Instead, information is *distributed* throughout the entire network and is retrievable only through the interactions of its various processing elements. Unlike conventional computer systems where information is retrieved or fetched from memory, an artificial neural network can best be described as *evoking* its stored information.

Processing Elements

As mentioned earlier, a network is comprised of numerous, independent, highly interconnected processing elements. For backpropagation networks, each element can be characterized as having some *input* connections from other processing elements and some *output* connections to other elements. The basic operation of an element is to compute its *activation value* based upon its inputs and send that value to its output elements. Figure 1 shows a schematic of a processing element. Note that this element has *j* input connections coming from *j* input processing elements. Each connection has an associated value called a *weight*. The output of this processing element is fashioned to nonlinearly transform its summed, continuous-valued inputs by the sigmoid transformation shown by the two formulas in Figure 1. Understanding the details of this transformation is not essential here; the interested reader will find an excellent description of such details provided by Rummelhart et. al.[7]. For the purposes of this discussion it is important simply to note that a processing element's output is calculated solely from the influence of its incoming elements and connections.

When groups of processing elements are arranged in sequential layers, each layer interconnected with the subsequent layer, the result is a wave of activations propagated from the input processing elements, which have no incoming connections, to the output processing elements. The layers of elements between the inputs and outputs take on intermediate values which perform a mapping from the input representation to the output representation. It is from these intermediate or *hidden* elements that the backpropagation network draws its generalization properties. By forming transformations through such intermediate layers, a backpropagation network can arbitrarily categorize the features of its inputs. More importantly, since these categorizations are formed by summing the effects of the inputs, the result is a generalization over the input vector.

$$E_i = \Sigma w_{ij}\, p_j$$

$$p_i = P\,(E_i\,) = \frac{1}{1 + e^{-E_i}}$$

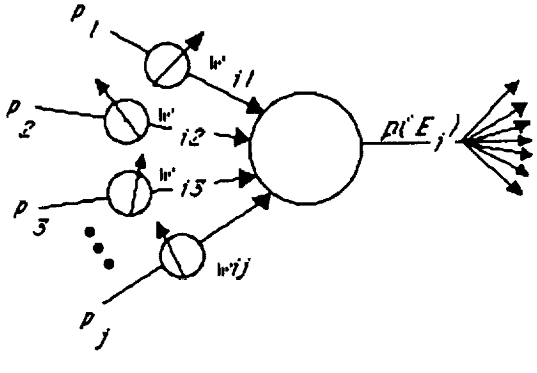

Figure 1: Processing element in a backpropagation network.

The Weights of a Backpropagation Network

The heart of the backpropagation algorithm lies in how the values of its interconnections, or weights, are updated. Initially, the weights in the network are set to some small random number to represent no association between processing elements. Upon being given a set of patterns representing pairs of input/output associations, the network enters what is called a *training* phase. During training, the weights are adjusted according to the learning technique developed by Rumelhart et. al. The training phase is modelled after a behavioristic approach which operates through reinforcement by negative feedback. That is, the network is given an input from some input/output pattern for which it generates an output by propagation. Any discrepancies found when comparing the network's output to the desired output constitute mistakes which are then used to alter the network characteristics.

According to Rumelhart's technique, every weight in the network is adjusted to minimize the total mean square errors between the response of the network, p_{pi}, and the desired outputs, t_{pi} ,to a given input pattern. The indices p and i represent the pattern number and the index to a node respectively. The weights are adjusted according to:

$$\Delta w_{ij}^{(t+1)} = \alpha \Delta w_{ij}^{(t)} + \eta \delta_i^{(n+1)} P_j^{(n)}$$

where $\Delta w_{ij}^{(n)}$ is the error *gradient* of the weight from the jth processing element in layer n to the i th unit in the subsequent layer $(n + 1)$. The parameter α, performs a damping effect through the multi-dimensional error space by relying on the most recent weight adjustment to determine the present adjustment. The overall effect of this weight adjustment is to perform a gradient descent in the error space; however, note that true gradient descent implies infinitesimally small increments. Since such increments would be impractical, η is used to accelerate the learning process. Finally, the error signal, δ_i, is first determined for the output layer, N:

$$\delta_i^{(N)} = (t_i - p_i^{(N)}) P'(E_i^{(N)})$$

and then recursively back propagated through the higher layers:

$$\delta_i^{(n)} = \sum_j \delta_j^{(n+1)} w_{ji}^{(n)} P'(E_i^{(n)}) ,$$

where $P'(E)$ is the first derivative of $P(E)$.

Again, the fine details of the above equations are left to the more thorough discussion provided by Rumelhart, though some important features should be emphasized. First, note that each weight is changed according to a gradient descent technique. This implies that the training process is meant to converge on some minima in the error space. The network is said to have *learned* if the error at this point is below the desired threshold set by the user at which point no further training is performed. Loosely speaking, this implies that the more weights present, the larger the error space and, in general, the larger the number of minima at which the network can be satisfied. The implication, then, is that using larger hidden layers which require more weights will help the network to converge. Unfortunately, added "convergence power" is not the only effect of increasing the numbers of hidden processing elements. Larger hidden layers adversely influence the generalization capabilities of a backpropagation network. In short, the network simply "memorizes" the training patterns. It is only through decreasing the number of hidden processing elements that backpropagation networks can be forced to generalize.

At present, tradeoffs such as these are a typical part of designing neural networks. Setting the number of hidden processing elements, as well as determining values for the constants α and η, is still somewhat of a "black art" best mastered through experience. What follows is our experience setting these parameters for a backpropagation network used to predict sunspot activity.

Sunspot prediction

Several key issues must be considered when designing with a backpropagation network. These issues may include data preprocessing, data format presentation to the network, the network architecture, and the tests for generality. Other concerns include the number of training patterns and training cycles required for successful generalization. Experience has shown that, if possible, neural networks are easier to analyze and manage when the data is processed before it is presented to a neural network. Common forms of processing include normalization of the data, separating the data or system into its constituent forms, and compacting the data into non-redundant formats. Figure 2 illustrates the sunspot data as supplied by the NOAA; monthly sunspot numbers lie along the ordinate and time ranges from years 1834 to 1984 along the abscissa. The mathematics which describe the properties of the backpropagation are not well understood. Throughout the discussion, it will become apparent that neural network designs are primarily empirical. Therefore, we will focus on the issues critical to the development of a successful and usable neural network.

Selecting the training set

An immediate observation is the 11 year cyclic period evident throughout the supplied data. It was this observation that led to the selection of 132 months as the input to the neural network. The neural network had a task which is difficult even by human pattern recognition standards -- to predict future solar activity based solely on historical observations. As is evident in the data, the neural network had to automatically categorize between the very high frequencies apparent on a month by month scale, the middle frequency ranges or the 11 year cycle, and an even closer examination reveals a much lower frequency suppressing or intensifying magnitudes of sunspot activity clusters. The physical underlying phenomena behind these solar characteristics are not well understood by solar experts nor will the neural network be able to explain them.

However, solar science phenomenas were neglected and the data was fashioned so that the neural network's only goal was to predict a future sunspot level. Therefore, the neural network was presented 11 years of data at the input and the output represented the associated future month for a particular input pattern.

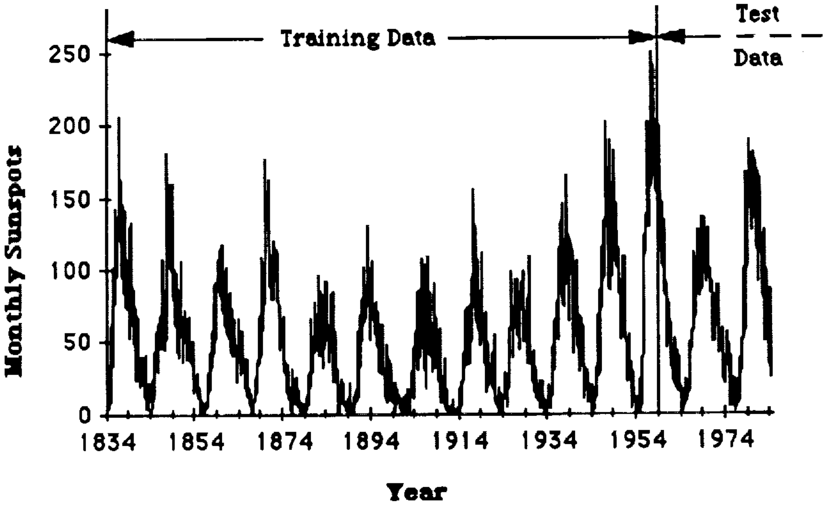

Figure 2: Monthly sunspot numbers from 1834 to 1984 as provided by the NOAA

All processing elements throughout the network used a non-linear activation function. When the upper and lower bounds of the sigmoidal transfer function are set at 1 and 0, respectively, close observation reveals that the most linear region of the processing element's output is exhibited in the range from 0.2 to 0.8. Therefore, the entire sunspot data was normalized between these two ranges; i.e., a value of 0.2 represented no sunspot sightings whereas a value of 0.8 represented a maximum of 254 sunspot sightings (maximum sunspot sightings observed for the provided epoch).

Careful consideration must be given when developing a system representation for a neural network. Sejnowski, in his development of NETtalk, a neural network which learned the relationships between the English language and phonetics, used the 1000 most commonly used words in the English language to train NETtalk. The belief was that a 1000 word set was rich with a sufficient range of English to phonetic translations to cover a large percentage of the rules necessary to read or pronounce a word. With NETtalk, there is the danger of selecting such a small set of words, say 50, that the network would undergeneralize or not pronounce untrained words correctly or to provide such a large training set, say 20,000 words, that the network would have difficulty making distinctions and not adjust its weights correctly. Again, no real specification or rule of thumb exists which can assist in selecting the appropriate training set. Until further advances in neural networks are made, empirical methods seem to be the only solution here. The sunspot prediction neural network was experimented with varying sets of resolution in the training patterns; the input windows were incremented by 1, 4, 8, 10, and 20 month steps. Increasingly better prediction performance was found with increasing step size, maximizing at 10 month steps, and decreasing prediction performance at 20 month steps.

Network Architecture

Another key factor in developing successful neural networks deals with the construction of the appropriate neural network architecture. An earlier project demonstrated that a neural network which generated speech signals from a phonetic type input could only converge with a two hidden layer architecture. Naturally, early efforts in sunspot prediction using neural networks were based on a two hidden layer network architecture. Experiments were conducted with several neural networks architectures which varied the numbers of processing elements in the hidden layers. Even though each neural network converged to an acceptable level, every network exhibited poor generalization capabilities. An interesting observation in the dual hidden layer sunspot prediction neural network architectures was that after satisfactory levels of convergence had been achieved, relatively little activation was present in any of the processing elements in the second hidden layer. This, however, was not the case for the speech generation neural networks. In fact, increased activity in the second hidden layer for the speech generation neural network serves as an indicator for successful generalization.

The next phase of this project experimented with single hidden layer neural networks. Again, the number of processing elements in the hidden layer were varied from 120, 80, 60, and 30 processing elements. Even though each neural network converged, only the neural network with 30 hidden processing elements displayed satisfactory levels of generalization.

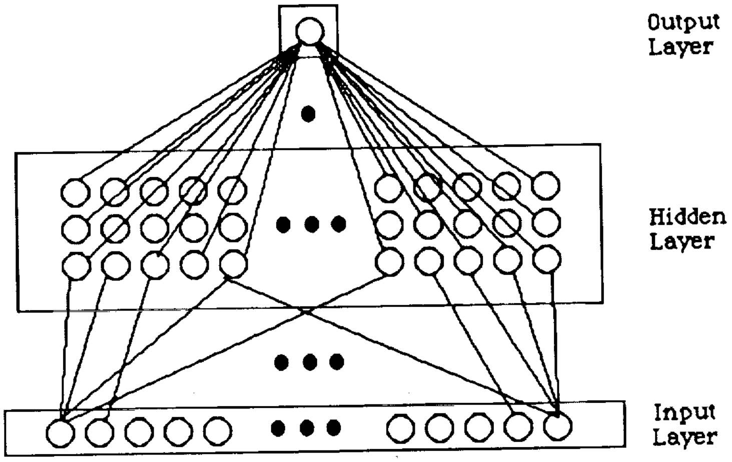

Figure 3: Neural network architecture used to predict a future month sunspot number given 132 (11 years) past sunspot numbers. The number of processing elements in the hidden layer were varied. All connections are not shown.

Testing the network

A neural network's worthiness is not measured by whether it can converge, but instead to how it can draw inferences or generalize to unforeseen stimuli. As noted earlier, each sunspot prediction neural network architecture experiment which varied the number of hidden layers, the number of processing elements in the hidden layers, and the complexity of the training data were all capable of converging. However, significant prediction capabilities were only discovered in a certain neural network architecture with a certain degree of training data complexity.

To test the generalization capabilities of the network, the network was trained with data from 1834 to 1959 and the state of the neural network's performance was tested against the remaining data which ranged from 1959 to 1984; i.e., the weights were not adjusted with the data from 1959 to 1984. Best generalization results were obtained when a 132 input, 30 hidden, and 1 output neural network was trained on a 10 month increment input pattern. Figure 4.0 is the RMS error of the output node with a 10 month step and 30 hidden processing elements. An interesting observation, which did not appear in any other error curve, is the crest found in the neighborhood of 500 passes. Whether the pattern in the error curve has any relevant significance to generalization is not known.

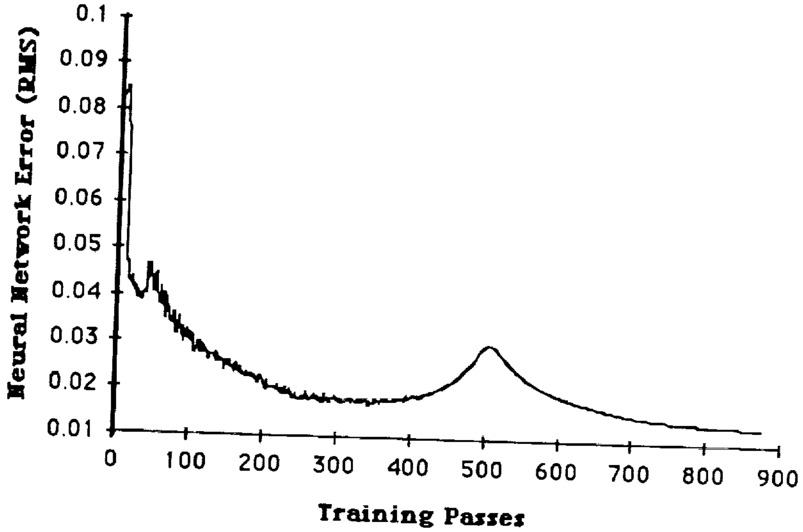

Figure 4: Root Mean Square (RMS) error for the output of a 132-30-1 neural network where the 11 year input window is incremented by 10 months as it traverses through the training data.

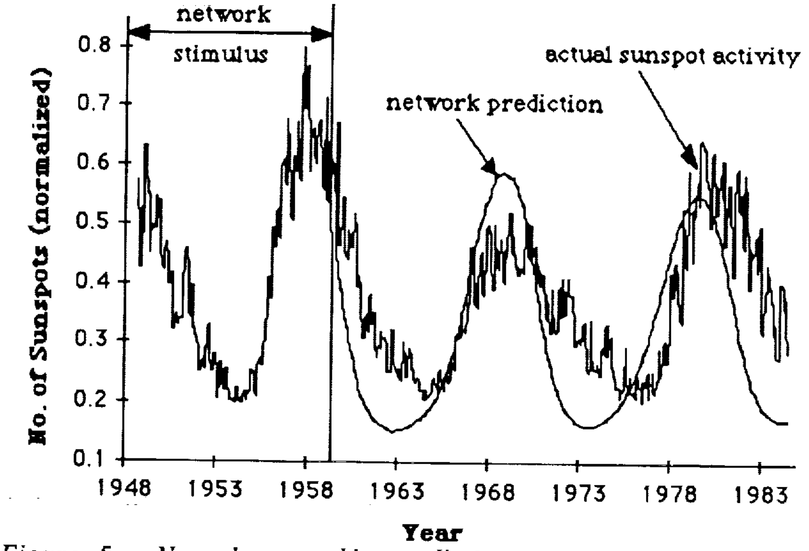

Figure 5a: Neural network's prediction performance after 25 passes (average error is 9.4% and RMS error is 11.3% for test segment).

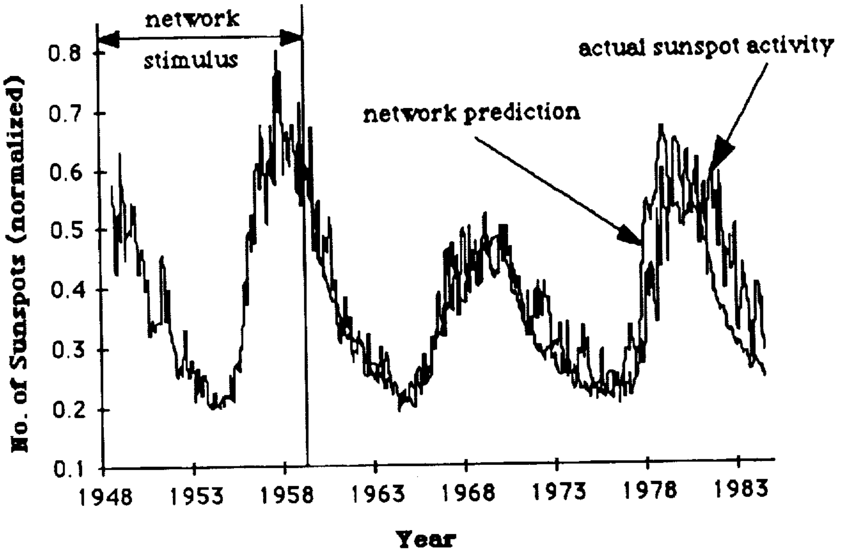

Figure 5c: Neural network's prediction performance after 610 passes (average error is 3.35% and RMS error is 4.58% for test segment).

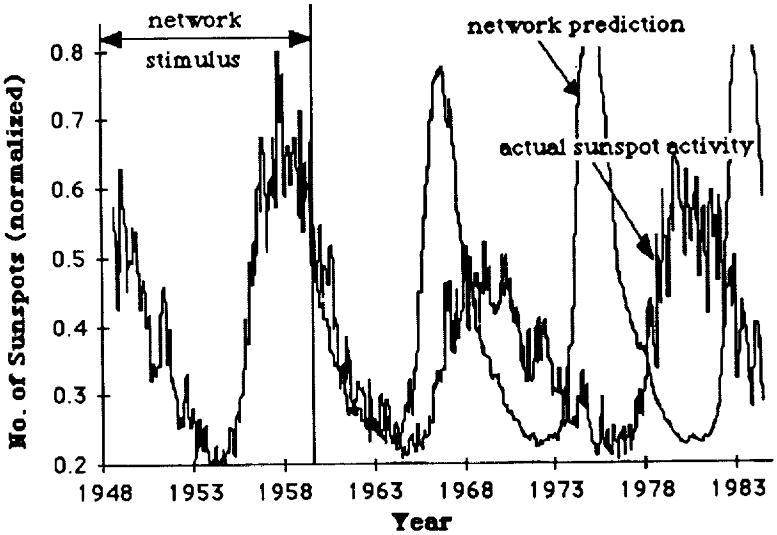

Figure 5b: Neural network's prediction performance after 260 passes (average error is 12.24% and RMS error is 18.16% for test segment).

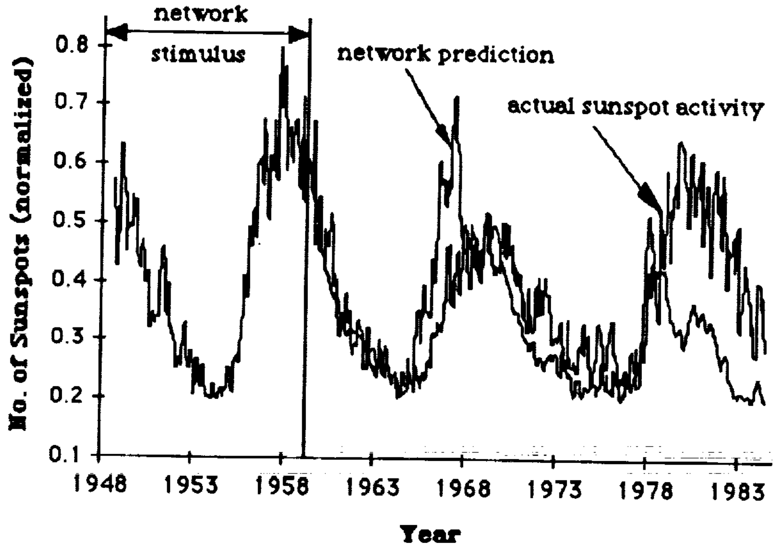

Figure 5d: Neural network's prediction performance after 810 passes (average error is 5.9% and RMS error is 8.6% for test segment).

Figure 5 illustrates several performance generalization states of the 132-30-1 neural network during the training process. The neural network's prediction capabilities were determined by suspending the "learning" process and then evoking the neural network with the last 132 months in the training set. The neural network's "prediction" or output was then appended to the tail end of the excitation source. This process was repeated for all the data points necessary to cover the remaining test portion of the actual sunspot number data.

The graphs show that, as expected, the neural network performed poorly in the early stages of training, peaked in the intermediate stages, and then returned to relatively poorer performances with continued training. For actual sunspot prediction uses, this neural network would be retrained with the original training data, while again monitoring its prediction performance against the test data. Having achieved a satisfactory level of confidence in its predictability, the neural network would be invoked with the actual test data and prompted to produce future "unknown" sunspot activity.

Conclusion

In summary, this work shows that neural networks are indeed a very useful tool for developing system models. Several key concerns have been pointed out which are necessary in developing useful neural network based systems.

References

[1] Bray, R. J., and Loughhead, R. E., [1964] "Sunspots", Dover Publications, Inc, New York, New York.

[2] Johnson, G. G., and Newman, S. R., [1980]"Solar Activity Prediction of Sunspot Numbers - Predicted Solar Radio Flux", JSC-16390, Houston, Texas.

[3] Herman, J. R., and Goldberg, R. A., [1977] "Sun, Weather, and Climate", Dover Publications, Inc., New York, New York.

[4] McNish, A. G. and Lincoln, J. V., [1949] "Prediction of Sunspot Numbers", Transactions, American Geophysical Union, Vol. 30, Number 5, pp. 673-685.

[5] Newman, S. R., [1980] "Solar Activity Prediction of Sunspot Numbers (Verification) - Predicted Solar Radio Flux - Predicted Geomagnetic Indices A_p and K_p", JSC-16762, Houston, Texas.

[6] Pepin, R. O., Eddy, J. A., and Merrill, R. B., [1979] "Proceedings of the Conference on The Ancient Sun - Fossil Record in the Earth, Moon, and Meteorites", Boulder, Colorado.

[7] Rumelhart, D. E., and McClelland, J. L., [1986] "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", MIT Press, Cambridge, Massachusetts.

[8] Sawyer, C., Warwick, J. W., and Dennett, J. T., [1986] "Solar Flare Prediction", Colorado Associated University Press, Boulder, Colorado.

[9] Sejnowski, T. J. and Rosenberg, C. R., [1986] " NETtalk: A Parallel Network that Learns to Read Aloud", Johns Hopkins University, Technical Report JHU/EECS-86/01.

[10] Villarreal, J. A., [1988] "Artificial Neural Network Directions Within the Artificial Intelligence Section/ NASA", Proceedings of the ISA-88 International Conference and Exhibit, Houston, Texas.

[11] Vitinskii, Y. I., [1962] "Solar Activity Forecasting", Translated from Russian, U. S. Department of Commerce, Clearinghouse for Federal Scientific and Technical Information, Springfield, VA.

[12] White, H., [1988] "Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Predictions", Proceedings of the IEEE International Conference on Neural Networks - 1988, Institute of Electrical and Electronics Engineers, Inc., New York, New York, pp. II-451 - II-458.

[13] Wang, J. C. H., [1980] "A note on sunspot records from China", Proceedings of the Conference on The Ancient Sun - Fossil Record in the Earth, Moon, and Meteorites, Boulder, Colorado.