Artificial Neural Network to Forecast Short-Term Cloud Cover

Final Report

Prepared for:
NASA
Kennedy Space Center
Under Contract NAS10-11844

11 July 1994

Prepared By:
Applied Meteorology Unit

ENSCO, Inc.
445 Pineda Court
Melbourne, Florida 32940
(407) 853-8205
(407) 254-4122

Attributes and Acknowledgments:

NASA/KSC POC:
Dr. Frank Merceret, Chief, Applied Meteorology Unit
TM-LLP-2A

Applied Meteorology Unit (AMU)

Robin Schumann, Primary Author
John Manobianco
Greg Taylor
Mark Wheeler
Ann Yersavich

Table Of Contents

List of Tables

List of Figures

Executive Summary

During the analysis for the *Shuttle Landing Facility Cloud Cover Study: Climatological Analysis and Two Tenths Cloud Cover Rule Evaluation* (Atchison et al. 1992), we noticed some regularities in the data that led us to believe artificial neural network technology could be used to develop potential forecasting tools. Artificial neural network models have the ability to resolve many nonlinear relationships and handle highly correlated data making them very appropriate for application to meteorological problems.

Subsequently in 1993, the AMU was tasked to develop a prototype forecast tool using artificial neural network technology. The priority and level of the AMU's other taskings left very limited resources and time to devote to the artificial neural network development and implementation. Since we had already assimilated five plus year surface observation and upper air data bases during the cloud cover study, we estimated we would be able to develop an operational prototype neural network model within the limited resource constraints. We began development of the ANN in September 1993 and continued the effort through the middle of January 1994 at about one third of a full time equivalent effort.

We began the project by attempting to develop an artificial neural network using the surface data only as inputs. Selection of the surface data training set consisted of first dividing the data into subsets according to the observed change in cloud cover over two hours, then randomly selecting a uniform distribution of training records from the subsets. Initial selection of the actual input variables to use in the training set consisted of polling the user community for their suggestions of surface observations which may be indicative of near term cloud cover changes. Trial and error methodology was used thereafter to adjust the input variables.

Two different artificial networks were generated, one trained and tested with data spanning the entire year and another trained and tested with summer time (May - September) data only. Briefly, the models were generally able to differentiate between significant cloud cover increases and decreases, but they did not do well predicting the magnitude of the change.

Rather than exert more effort performing a detailed analysis of the surface observations in an attempt to improve the networks' performance, we felt it would be more beneficial to include the upper air data as input to the network.

Addition of the upper air data did not, however, have the desired effect. The temporal resolution of the rawinsondes was much less than that of the surface observations. The amount of cloud cover would change dramatically in both directions while the measured upper air data did not change at all. This resulted in the neural network tending to output a zero change in cloud cover all the time. The upper air data were incorporated into the network towards the end of the project and time constraints

did not allow us to further adjust the upper air representations in the neural network model.

We underestimated the effort necessary to develop an operational neural network prototype. The surface observations alone did not provide easily detectable patterns for the neural network model to recognize and associate with near-term cloud cover changes. Consequently, the neural network model did not perform as well as expected. The temporal resolution of the upper air data prevented it from being exploited by the neural network model within the allotted time schedule. Due to the greater priority of other tasks, we did not pursue the work further as it was unclear how much effort would be required to improve the model's performance to the level where it would be a useful forecast tool.

Despite the failure to develop an operational prototype within the given time schedule, this project did demonstrate that neural networks may have potential as forecasting tools. Even when provided the limited data already available within the AMU, the neural network did exhibit some ability to learn. The conclusion within this report suggests some directions for continuation of this project should the resources become available.

1.0Introduction

This brief report describes the Applied Meteorology Unit's development and evaluation of an artificial neural network for predicting cloud cover at the Shuttle Landing Facility (SLF). This first section describes the motivation for the project and provides an introduction to artificial neural network models. Section 2 describes the development of the neural network model and summarizes its effectiveness. Section 3 summarizes the evaluation analysis performed on the neural network model; and finally, Section 4 summarizes the project's successes and failures.

1.1 Project Background and Motivation

During our analysis for the *Shuttle Landing Facility Cloud Cover Study: Climatological Analysis and Two Tenths Cloud Cover Rule Evaluation* (Atchison et al. 1992), we noticed some regularities in the data that led us to believe artificial neural network technology could be used to develop potential forecasting tools. Artificial neural network models have the ability to resolve many nonlinear relationships and handle highly correlated data making them appropriate for application to meteorological problems.

The shuttle landing forecast is a challenging forecast given the very specific time and spatial constraints associated with the operation. The decision to land at a given site must be made prior to the de-orbit burn, approximately 90 minutes before the actual landing time. Because of Florida's dynamic weather, NASA developed what is known as the KSC two-tenths cloud cover rule to reduce the risk of attempting a shuttle landing in unacceptable weather. The rule states:

> For scattered cloud layers below 10K feet, cloud cover must be observed to be less than or equal to 0.2 at the de-orbit burn go/no-go decision time (JSC Flight Rules).

The two-tenths cloud cover rule is designed to minimize the likelihood of landing the shuttle during weather constraint violations related to cloud cover (i.e. ceilings, thunderstorms, and precipitation.). The AMU recommended a tasking to implement an operational prototype neural network to forecast cloud cover that upon evaluation could eventually be transitioned to operational use.

Subsequently in 1993, the AMU was tasked to develop a prototype forecast tool using artificial neural network technology. The priority and level of the AMU's other taskings left very limited resources and time to devote to the artificial neural network development and implementation. Since we had already assimilated five plus year surface observation and upper air data bases during the cloud cover study, we estimated we could develop an operational prototype neural network model within the limited resource constraints. The rest of the report documents the development of the neural network model and presents the evaluation of its performance.

1.2 Description of Artificial Neural Network Models

This subsection provides a brief introduction to artificial neural network models. It is included to provide interested readers unfamiliar with artificial neural network techniques a brief summary of the concepts behind artificial neural network techniques. Traditional computing techniques take advantage of the computer's architecture to solve problems well understood but not easily solved by human calculation. On the other hand, some tasks, such as pattern recognition and motor control which are not well understood, are easily handled by the brain and nervous system yet elude traditional computer procedures. Artificial neural networks attempt to model these poorly understood problems by employing a mathematical model of the brain's structure.

The brain consists of billions of densely interconnected neurons. The premise behind artificial neural network models is that mimicking the brain's structure of many highly connected processing elements will enable computers to tackle tasks they have not as of yet performed well. Artificial neural networks are mathematical models derived from this structure. Though biological plausibility is sometimes applied to artificial neural network models, they are not intended to model the actual workings inside the brain or nervous system.

In general, a neural network model consists of neurons or processing elements, each of which is connected to other elements according to some schema by connection weights. The connection weights between processing elements contain the knowledge stored in the artificial neural network model. Usually, the processing elements are classified as input units, output units, or hidden units. (Some neural network models which perform tasks such as optimization do not have specific input or output units.) Model input is supplied through the input units, and model output is shown on the output units. The hidden elements are necessary to enable the system to learn relationships which are not linearly separable. Figure 1 illustrates a typical neural netwwork model. The model learns by adjusting its connection weights in response to the input-output pairs presented to it during training. Neural networks are trained by example, they are not usually programmed with a priori knowledge.

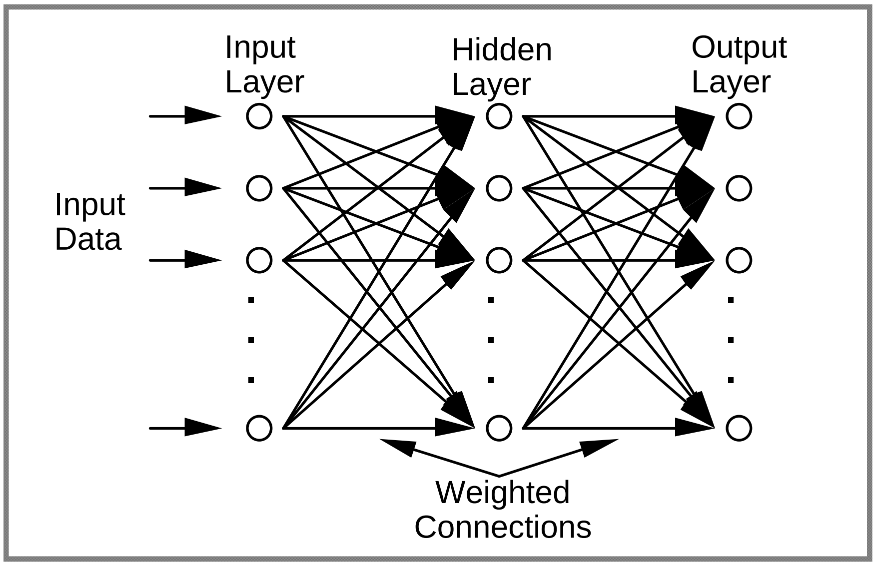

Figure 1. Neural Network Model Example

Though much of the motivation driving neural computing research has been geared towards development of specialized hardware, the mathematical models have been coded into software and proven to be valuable tools in the areas of signal processing, system modeling, pattern recognition, and classification.

In neural network models for prediction, inputs and their associated outputs are presented to the network's input and output processing elements, respectively. The connection weights are adjusted after the input - output pair of vectors is presented to the network until the network is able to produce the desired output within some pre-determined error bounds. The algorithm for adjusting the weights depends upon the type of network model used. In this application, the backward error propagation model was used. Backward error propagation is described in detail in nearly all neural network text books (see NeuralWare 1991, Hertz, Krough, and Palmer 1991, Rumelhart and McClelland 1986, and Aleksander and Morton 1991).

1.3 Planned Approach and Methodology

As stated previously, the AMU's resources available for the neural network development and implementation were limited. Since the surface observation and upper air data bases were already compiled for the two-tenths cloud cover study, we believed we could develop a viable prototype neural network with the limited resources allocated to the project.

We planned to develop a neural network training set using the surface observations as the primary input and to include upper air data if necessary to resolve conflicts in the surface observations. The neural network would be trained to recognize surface conditions indicative of future cloud cover changes and to predict the amount of cloud cover change over the next two hours.

Our evaluation plan for the neural network model included testing the model under a variety of initial conditions and documenting its performance under those conditions. It also included determining the probability of detection and false alarm rates for significant increases and decreases in cloud cover.

2.0 ANN Development and Evaluation Summary

We began development of the ANN in September 1993 with the transfer of the surface observation data base resident on a PC to SAS on an IBM RS/6000 UNIX workstation. The work continued through the middle of January 1994 at a level of about one third of a full time equivalent. This section describes in detail the data available for the project and the neural network model development effort.

2.1 Database Description

The surface observation database developed for the AMU's *Shuttle Landing Facility Cloud Cover Study: Climatological Analysis and Two Tenths Cloud Cover Rule Evaluation* was used to train and test the neural network models. The data base contains hourly estimates of the following items:

- Observations of
  - thunderstorms,
  - precipitation,
  - fog, and
  - haze,
- Surface pressure,
- Surface dew point,
- Surface wind direction,
- Surface wind speed,
- Surface temperature,
- Total cloud cover,
- Tenths of cloud cover below 10 000 feet,
- Ceiling height, and
- Visibility.

During the preparation of the data base for the cloud cover climatological study, the tenths of cloud cover below 10 000 feet were estimated manually from information on the Form 10's filled out by the Range contractor weather observers located at the weather station adjacent to the SLF. The tenths of cloud cover were not estimated for the hours when a weather constraint was violated in order to reduce the effort involved.

Ceilings below 10 000 feet, however, are weather constraint violations, and there were no actual tenths of cloud cover estimates for initial times or forecast verification times for these conditions. Training data records containing ceilings were essential for network utility (it was important to know if the cloud cover increased to over five tenths) so we assigned a value of seven tenths to any data record indicating a ceiling. (This left weather constraint violations where a ceiling was not in effect as the only records systematically omitted from the training set.) Seven tenths was used as a compromise since the actual cloud cover could range from six to ten tenths. This provided the network with training vectors containing conditions leading to a ceiling as well as conditions indicating the break up of a ceiling. The training vectors could not, however, indicate whether a ceiling condition improved or worsened (i.e. cloud cover increased or decreased between six and ten tenths). This may have introduced some bias to the network model.

In addition to the surface observations, the AMU had previously developed a data base containing Cape Canaveral (station 74794) rawinsonde data from 1986 through 1992 for the two-tenths cloud cover climatological study. The rawinsonde data consisted of the following items for each record:

- Year,
- Month,
- Day,
- Altitude,
- Wind direction,
- Wind speed,
- Wind shear,
- Temperature,
- Dew point,
- Pressure,
- Relative humidity, and
- Index of refraction.

These data were merged with the surface observations in order to provide a complete picture of the local atmosphere. The upper air data were not as temporally dense as the surface observations, so upper air elements were repeated for several hourly data records. To ensure the neural network was not provided any data during training that would not be available in real-time, data were repeated for the hours after the sounding and preceding the next sounding rather than for the hours surrounding the actual sounding time which would be a closer representation of the actual conditions.

2.2 Development Chronology

We began the project by attempting to develop an artificial neural network using only the surface data as inputs. Selection of the surface data training set consisted of dividing the data into subsets according to the observed change in cloud cover over two hours, then randomly selecting a uniform distribution of training records from the subsets. The training data set consisted of a uniform distribution of cloud cover changes in order to force the neural network model to learn the surface conditions predictive of cloud cover changes. If the input data reflected the true distribution of cloud cover changes, where the vast majority of two hour cloud cover changes were zero, the neural network could easily have reduced the RMS error on its output by predicting a zero change all of the time rather than learning the conditions indicative of a near-term cloud cover change.

Initial selection of the actual input variables to use in the training set consisted of polling the user community for their suggestions of surface observations which may be indicative of near term cloud cover changes. Trial and error methodology was used thereafter to adjust the input variables. We tried several different input scenarios. In some cases, input data which would have a significant impact on an operational forecaster's cloud cover prediction did not necessarily have the desired effect on the neural network. For example, when initial cloud cover was used as an input to the model, the neural network learned that if the initial cloud cover is zero, then the only possible change is to increase. Conversely, it learned that if the initial cloud cover is a ceiling, then the only possible change is to decrease. Though the neural network model had a near 100 percent probability of detection for large cloud cover changes when the initial cloud cover was included in the input, its false alarm rate was also extremely high.

After attempting several input scenarios, we developed a model which exhibited some limited success. The following items were used as inputs to the artificial neural network:

- Dew point depression,
- Wind direction,
- Change in dew point depression over last three hours,
- Change in temperature over last three hours,
- Time of day,
- Season,
- Change in cloud cover over last hour, and
- Change in cloud cover over last two hours.

Each of the above variables was scaled to the interval [-1,1]. Two different artificial networks were generated, one trained and tested with data spanning the entire year and another trained and tested with summer time (May - September) data only.

Section 3 presents a detailed analysis of these networks' performance. Briefly, the models were generally able to differentiate between significant cloud cover increases and decreases, but they did not do well predicting the magnitude of this change. As will be shown in Section 3, neither model performed as well as persistence. This was probably due to the common practice in training neural networks of using uniform distributions in the training data. In this case, the change in cloud cover was uniformly distributed in the training data. This forced the networks to learn changes rather than allowing them to reduce their RMS errors by responding with a zero change in cloud cover all of the time which is the most frequent observed result.

Rather than exert more effort performing a detailed analysis of the surface observations in an attempt to improve the networks' performance, we felt it would be more beneficial to include the upper air data as input to the network. In addition to the surface observations listed previously, the following new data elements were scaled to the interval [-1,1] and input to the neural network:

- Wind directions at 850, 700, and 500 millibars,
- Relative humidities at 850 and 700 millibars,
- Change in wind direction between 800 and 700 millibars,
- Change in wind direction between 700 and 500 millibars, and
- Change in temperature between 800 and 500 millibars

Addition of the upper air data did not have the desired effect. The temporal resolution of the rawinsondes was much less than that of the surface observations. Within the training set, the cloud cover would change dramatically in both directions while the measured upper air data did not change at all (a consequence of the poor temporal resolution of the upper air data). This resulted in the neural network tending to output a zero change in cloud cover all the time.

The upper air data were incorporated into the network towards the end of the project and time constraints did not allow us to further adjust the upper air representations in the neural network model. More innovative approaches to incorporating the data (i.e. applying the upper air data only at observation time and using 0's other times or using upper air data output by a mesoscale model output as input) may significantly improve the network's performance. Since the upper air data were not fully exploited in the neural network model, only the performance of the surface observation only neural network model was assessed.

3.0 Analysis of ANN Results

As described above, the artificial neural networks had trouble predicting the magnitude of the change in cloud cover. Table 3.1 shows the actual number of correct responses generated by the neural network for the two hour cloud cover forecast compared to persistence. Any neural network output within one tenth of the desired response was considered correct and any change less than or equal to one tenth in the observed data was considered to be characterized by persistence.

As is evidenced by the data in Table 3.1, the neural network did not perform as well as persistence. The data set used to train the neural network contained a uniform distribution of changes in cloud cover. That is, the number of records where the observed change in cloud cover was one tenth was the same as the number of records where the change was observed to be zero tenths, two tenths, and etc. This forced the network to learn changes. Had the neural network been trained with data actually representing the true distribution of the observed two hour cloud cover change, it may have been able to match persistence without really learning anything about the indicators of actual cloud cover change.

Table 3.1. ANN Output VS Persistence Over 2 Hours (Entire Year)

Initial Cloud Cover	# of Samples	% Correct by Persistence	% Correct Output by ANN
0	4701	87	78
1	3869	84	67
2	2910	74	55
3	2244	66	46
4	1288	51	45
5	631	32	44
6+	3719	70	60
Total	19362	75	62

The artificial neural network was more successful in correctly identifying whether the cloud cover increased or decreased over two hours than it was in predicting the actual cloud cover. It did not, however, handle situations where the observed change in cloud cover was one tenth or less in either direction. In those cases, the neural network's responses centered about 0, but were not reliable in direction and often indicated large increases or decreases in cloud cover when none were reported. Table 3.2 provides the probabilities the neural networks would detect an increase or decrease in the amount of cloud cover. The percentages provided in Table 3.2 are defined as follows:

POD^P: The probability the neural network would predict an increase in cloud cover greater than or equal to delta when the observed increase is greater than delta. (If delta equals 0, then assume delta = 1/10.)

FAR^P: The probability the neural network would predict an increase in cloud cover when the observed change in cloud cover was a decrease of delta or more. (In this case, do not assume delta = 1/10 if delta equals 0.)

POD^M: The probability the neural network would predict a decrease in cloud cover greater than or equal to delta when the observed decrease is greater than delta. (If delta equals 0, then assume delta = 1/10.)

FAR^M: The probability the neural network would predict a decrease in cloud cover when the observed change in cloud cover was an increase of delta or more. (In this case, do not assume delta = 1/10 if delta equals 0.)

Table 3.2. Ability of ANN to Distinguish Between Increases and Decreases in Cloud Cover Over 2 Hours (Entire Year)

Delta	POD^P (%)	FAR^P (%)	POD^M (%)	FAR^M (%)
0	62	78	32	65
1/10	62	34	32	23
2/10	63	22	44	23

Operational forecasters adhere to different sets of rules depending upon the time of year in order to accommodate seasonal effects in their analysis of the weather conditions and forecasting. Artificial neural networks tend to generalize and in doing so average seasonal effects over the entire year. To see how much this affected the results, we retrained the artificial neural network with daytime summer data only. Summer, in this case, consisted of the months May through September.

Table 3.3 compares the performance of the summer time only network to persistence. As with the artificial neural network for the entire year, the training data set consisted of a uniform distribution of cloud cover change rather than a distribution dominated by persistence. For the most part, Table 3.3 does not indicate a significant improvement in performance.

Table 3.3. ANN Output VS Persistence Over 2 Hours (Summer Only)			
Initial Cloud Cover (Tenths)	# of Samples	% Correct by Persistence	% Correct Output by ANN
0	1826	86	80
1	1914	85	71
2	1586	80	59
3	1176	72	49
4	587	57	55
5	217	37	49
6+	647	54	49
Total	7953	75	62

Table 3.4, however, does suggest the summer only artificial neural network performed a little better than the entire year artificial neural network. This is especially evidenced by the reduction in the False Alarm Rates.

Table 3.4. Ability of ANN to Distinguish Between Increases and Decreases in Cloud Cover Over 2 Hours (Summer Only)				
Delta	POD^P (%)	FAR^P (%)	POD^M (%)	FAR^M (%)
0	62	62	40	42
1/10	62	32	40	18
2/10	63	19	56	16

4.0 Conclusions

We underestimated the effort necessary to develop an operational neural network prototype. The surface observations alone did not provide easily detectable patterns for the neural network model to recognize and associate with near-term cloud cover changes and the neural network model did not perform as well as expected. The temporal resolution of the upper air data prevented it from being exploited by the neural network model within the allotted time schedule. Due to the greater priority of our other tasks, we did not pursue the work further as it was unclear how much effort would be required to improve the model's performance to the level where it would be a useful forecast tool.

Despite the failure to develop an operational prototype within the given time schedule, this project did demonstrate that neural networks may have potential as forecasting tools. Even when provided the limited data already available within the AMU, the neural network did exhibit some ability to learn. The following paragraphs identify some of the "lessons learned" and suggest some possible directions for continuation of the project should the resources become available.

First, data more predictive of cloud cover changes should be incorporated into the neural network. More innovative approaches could make it feasible to include the upper air data in the network training. Instead of using the same upper air data throughout the entire day (i.e. when there is only one sounding per day), forecast values from mesoscale models could be used when the latest sounding is no longer representative of the atmosphere. Also, more spatial data could be incorporated into the network. The neural network was not provided any information regarding cloud cover which may be advecting towards the SLF.

Also, developing different neural networks for the different times of the year should improve performance. (The summer only neural network performed slightly better than the neural network trained with data from the entire year.) Forecasters use entirely different sets of rules based on the time of year. In their tendency to generalize, neural networks average the seasonal effects over the entire year. Developing different neural networks for the different seasons would better allow the networks to develop their own sets of rules for the different seasons. The same principle applies to diurnal effects, and different neural network models for different times of the day may also be beneficial.

Finally, closer inspection of the training data, especially when only surface observations are used, may improve performance. The training data records were selected at random for this project. In some cases, surface observations which would normally indicate a decrease in clouds may result in an increase due to a feature not contained in the surface conditions. This confuses the network during training since two data records containing nearly the same input data items have entirely different desired results. Just a few of these anomalous records can inhibit the network's ability to learn the standard forecasting rules.

If the anomalous cases are removed from the training set, the network would still perform poorly for the anomalous cases, but its ability to learn the standard rules would be significantly improved. Removing anomalous cases is both labor intensive and risky since it would be easy to remove data records containing important information for the network. If the training data contain anomalous data records, it is best to add additional input elements that can differentiate between the apparent inconsistencies. If the necessary input elements are unavailable, then removing the anomalous data records may be necessary. Careful inspection of data along with documentation of the features the network does not handle well should be included as part of the input data editing process.

Artificial neural networks are similar to other models in that they can perform only as well as their input data allow. Surface observations are not customarily considered the only indicators of cloud cover changes and should not be the only input to a neural network attempting to provide operational cloud cover forecasts. As a short term experiment, however, the surface data allowed us to evaluate the feasibility of developing artificial neural network forecasting aids. Though artificial neural networks may prove useful in short-term cloud cover forecasting, the development of operational forecast tools requires significantly more resources than the AMU had allocated for this effort.

5.0 References

Alexander, Igor and Helen Morton. 1991. *An Introduction to Neural Computing*. Chapman and Hall 240 pages.

Atchison M. K., et al. 1993. *Shuttle Landing Facility Cloud Cover Study: Climatological Analysis and Two-Tenths Cloud Cover Rule Evaluation*

Johnson Space Center Flight Rules, paragraph 4-64, page 4-58.

Hertz, J., A. Krogh, and R. G. Palmer. 1991. *Introduction to the Theory of Neural Computation.* Addison-Wesley Publishing Company. Redwood City, CA.

NeuralWare. 1991. *Neural Computing. NeuralWare,* Inc.

Rumelhart, D. E. and J. L. McClelland, editors. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume I, Foundations.* MIT Press.