

# Air Temperature Prediction Using Evolutionary Artificial Neural Networks

Raj Yadav

*Assistant Professor/ Software Engineer  
Kautilya Institute of Technology & Engineering  
Email : yadavrajc@gmail.com*

**Abstract: Evolutionary neural networks have been applied successfully in the past to many prediction problems. In this paper I describe an evolutionary neural network, which attempts to predict the maximum air temperature given the day and month of the year.**

## I. INTRODUCTION

As scientists and philosophers ponder human intelligence, several profound questions arise: what is intelligence and is it measurable, does intelligence even exist, and can it be reproduced in a machine? We immediately go to the best empirical source about what gives humans the capacity to be intelligent, the brain. While trying to classify and understand this vital organ, early researchers attempted to partition the brain into smaller pieces until they arrived at the brain cell and neurons. They found that there existed many neurons in the brain, which were all interconnected and formed a sort of network, a Neural Network (NN).

As these seemed like simple enough constructs when looked at in a micro scale compared to the brain, researchers in the 1940's [1] attempted to model this construct of interconnected nodes in a computer to improve computing power. The Neural Network model researchers agreed upon was a series of connected nodes, each of which was a simple calculator. Every connection had a weight associated with it so that the influence of one node over another could be determined and controlled.

In a similar light, as human understanding of the natural laws of evolution and survival-of-the-fittest grew scientists successfully used them to create a new model of computation, Genetic Algorithms (GAs). The basis of GAs is that those of a population not suited for an environment (solution) will die off leaving the strongest to procreate. These progeny will then be allowed to mutate and evolve towards the fitness that best suits the environment; as the environment changes, so the population evolves to fit the new environment. Possibly the best product of GAs is the ability to converge quickly to a solution in a large search space.

When these two models, NNs and GAs, were brought together they formed an Evolutionary Artificial Neural Network (EANN). The necessity for this relationship came as researchers realized the benefits of searching for the optimal training set, topology, thresholds, and weights to increase the generalization, accuracy, and performance of a network. For a more through discussion about EANNs see [2].

The NN model seems to be perfectly suited to pattern recognition and inductive reasoning. For this reason EANNs have been used heavily in many applications where these problems are found, such as River Flow Prediction [3], Sun Spot Prediction [4], Image Processing Tasks [5], Classifying Cancer Cells [6], Classifications of Sonar Targets [7], and many more.

## II. WEATHER PREDICTION

As it is well known, weather prediction and meteorology is a very complex and imprecise science. The chief reason for this complexity is that the atmosphere of the Earth is essentially a chaotic system. Currently, to get a reasonable accurate prediction of weather patterns, supercomputers are used to model the atmosphere using as many known atmospheric variables as possible [8].

EANNs fit this problem well for 5 reasons: they would reduce the computational power required to accurately predict atmospheric variables from a supercomputer to a single NN, a large database is available of historical weather data which can be used as training sets, EANNs find the best generalized network to solve patterns outside their training sets (in comparison to ANNs), EANNs can detect and utilize hidden patterns in data to arrive at a solution, and finally, EANNs have been shown to accurately predict irregular and complex variables in past work [3,4,5,6,7].

To show whether this conjecture is true and atmospheric variables can be predicted to within a reasonable range using EANNs, an EANN will be designed using historical daily weather data to predict the daily maximum temperature of a future date.

## III. IMPLEMENTATION

### Data Sets

The data collected by the University of California Statewide Integrated Pest Management Project in the UC IPM California Weather Database [9] was used to provide training, test, and validation sets for the EANN. The data set selected was collected at a Brentwood, California (BRNTWOOD.A) weather station while some data from Tracy (TRACY.A) and Davis (DAVIS.A) was used to fill in the missing values. Together, these weather stations provided the date, daily precipitation, max temperature, min

temperature, max soil temperature, min soil temperature, max relative humidity, min relative humidity, solar radiation, and wind speed between November 18, 1985 and November 18, 2001 (5845 days). While disregarding days with incomplete data (missing values), a training set was created from years 1985-November 18, 1993 (2920 days), a test set was created from years November 19, 1993 – November 18, 1997 (1462 days), and a final validation set was created from years November 19, 1997 – 2001 (1463 days).

### Input and Output

The inputs for the network were month, day, daily precipitation, max temperature, min temperature, max soil temperature, min soil temperature, max relative humidity, min relative humidity, solar radiation, and wind speed. Although each of these may or may not have a direct correlation with maximum daily temperature, the EANN determines exactly how much influence each of these variables has over temperature and assigns weights to their connections accordingly. The output of the network was its predicted value for the maximum temperature that given day. All inputs and outputs were, as is with all neural networks, normalized to [-1,1] using the function:

$$\text{normalized} = \frac{(\text{maximum\_value} - \text{actual\_value})}{(\text{maximum\_value} - \text{minimum\_value})}$$

This normalization allows for the most regular topology to be evolved by the EANN and thereby the best generalization of the network.

### Network Representation

Because a EANNs convergence time is determined mainly by the method chosen to encode the NN representation in the GA, Kitano Grammar Encoding [10] was chosen over Direct Encoding for a faster convergence time [11]. This encoding method has the advantage of shortening the GAs chromosome length and still discovering a solution in the

search space very fast while representing the full connectivity matrix. Unlike Direct Encoding where each chromosome connection has genetic operators applied to it, Kitano Grammar Encoding uses the GAs power to evolve replacement rules to develop a correct grammar for the network. These replacement rules are then translated into replacement constants, which are not evolved, and thereby into the connection matrix. The primary difference in the methods is that, if  $n$  is the number of nodes in the network, Direct Encoding uses a matrix of size  $2^n$  and Kitano Encoding evolves a matrix of size  $n^2$ . Since chromosome size is the key in convergence time for a GA, the smaller chromosomes of the Kitano Encoding will allow the population to converge at a faster rate [1].

### Network Training Algorithm

The well-known NN training algorithm, backpropagation (BP), was used to identify and correct the weights of the network. The BP algorithm was chosen for its simplicity. However better choices would have been QuickProp (QP), or Rprop (RP) because of their faster convergence time, and better performance on noisy, complex and deceptive surfaces [2].

### Network Parameters

**Table 1.** Network Parameters

Parameter	Value
Neuron transfer function	Tanh
Weight initialization function	Gaussian; mean = 0.0; std = 5.0;
Stopping criteria	network_error = 0.01; max_iter = 500; conv = 200; eps = $10^{-4}$ ;
BP error function	mean square error (MSE);
Training Epochs	total_epoch = 30;

## IV. PERFORMANCE EVALUATION

Because generalization of the network is its highest valued property, as is with most prediction networks, performance on the test and validation data sets was used to evaluate the fitness of each network. After the network had been trained, and each of the test and validation sets were evaluated by the resulting network, the number of results that were within the allowed prediction error bounds were returned to the GA environment for fitness evaluation and population modification.

### Reproduction Operators

The genetic operator crossover was chosen as the means of chromosome reproduction within the genetic population. From the chromosomes that were not eliminated because they were nonfunctioning or did not meet performance criteria, two were chosen randomly. The first randomly chosen chromosome created a new chromosome using the first half of its production rules; the second chromosome finished the new chromosome by supplying the second half of its production rules. This method was successful, however a better reproduction operator could be produced to guarantee the resultant chromosome (network) from the pairing be functional, as is not the case with this specified crossover method.

### Algorithm

The algorithm that was used is very simple.

1. Randomly create a population of chromosomes
2. Build each chromosome as a network
  - a. Train network
  - b. Test network
  - c. Validate network
3. Fitness quantified by number of tests that returned results within error bounds
4. Eliminate chromosomes (networks) that do not meet fitness bounds
5. Apply genetic operator crossover to non-eliminated chromosomes

6. Apply genetic operator mutation to population
7. If population does not meet fitness requirements return to step 2

## V. TESTS AND RESULTS

### Runs

There were three runs made, each having their genetic chromosome length and error bounds varied to gain a better understanding of the ability of an EANN to predict the maximum daily temperature. Each run used the same network parameters, training, test, and validation data sets. The results are below. The error bound is the maximum number of degrees in temperature by which the network can be incorrect and still be considered a valid prediction. The genetic chromosome size is the number of bits in the genetic chromosome when the Kitano Grammar is translated into a connection matrix. The generations to convergence is how many genetic generations were required to evolve the best network given network parameters. The number of predictions within error bounds (correct) is the number of validation dates (1463 days) in the validation data set that were predicted within the error bounds.

#### Run One

**Table 2.** Run One Data

Error bounds	+/- 2 degrees
Genetic chromosome size	25 bits
Generations to convergence	48
Number of predictions within error bounds	1111, 75.93%;

The first run shows that only 75.93% prediction accuracy is attained when the error bounds are restricted to 2 degrees. Although this is a tight prediction requirement, it is still a low accuracy rate.

#### Run Two

**Table 3.** Run Two Data

Error bounds	+/- 3 degrees
Genetic chromosome size	25 bits
Generations to convergence	45
Number of predictions within error bounds	1199, 81.95%

This run was designed to see how strong the correlation is between the error bound variable and the accuracy rate. Compared with run one, these parameters do very well; attaining a 81.95% accuracy rate with a 3 degree error bound, that is a 6.02% improvement with only losing 1 degree of accuracy. However, the resulting accuracy rate of 81.95% is lower than expected, and unreasonable given other methods of computational weather prediction.

#### Run Three

**Table 4.** Run Three Data

Error bounds	+/- 2 degrees
Genetic chromosome size	50 bits
Generations to convergence	130
Number of predictions within error bounds	1163, 79.49%

Given that we have a standard for our network to predict the maximum air temperature within 2 degrees from the first run, the third run was designed to see how the genetic population size (and its resulting network) affected the prediction accuracy. With a 200% increase in chromosome size, the accuracy rate rises 3.56%. However, the accuracy rate attained with 50 bits is nearly that when we lose a degree of accuracy.

The reason for the chromosome length being involved in the prediction accuracy is that the larger the connection matrix, the greater the number of networks that can be created; this is because there can now be 50 nodes in the network where as previously there could be only 25.

## VI. CONCLUSION

Meteorology is a very difficult science because of the complex and chaotic systems involved. At times these systems make forming predictions nearly impossible, as shown with severe storm prediction [12]. However, it is this author's belief that reasonable maximum temperature predictions within 2 degrees should occur with at least a 90% accuracy rate to rival other meteorological prediction models. It was shown that given the specifications of this system and the provided data, with a 2-degree error bound only a 79.49% accuracy rate was achieved.

The author believes that a reasonable prediction accuracy rate could be achieved with this methodology given a larger training set, using faster and better training algorithms, and more known atmospheric values. This objective will obviously be the goal of future work in this area of research.

## VI. REFERENCES

- [1] Branke, Jürgen. Evolutionary Algorithms for Neural Network Design and Training, 1995, on-line, accessed on November, 15 2001.
- [2] Yao, Xin. A Review of Evolutionary Neural Networks, 1992, on-line, accessed on November, 15 2001.
- [3] Prudêncio, Ricardo Bastos Cavalcante; Ludermir, Teresa Bernarda. Evolutionary Design of Neural Networks: Application to River Flow Prediction, 1999, on-line, accessed on.
- [4] Hakkarainen, J.; Jumppanen A.; Kyngäs, J.; Kyyrö, J. An Evolutionary Approach to Neural Network Design Applied to Sunspot Prediction, 1996, on-line, accessed on November, 15 2001.
- [5] Mayer, Helmut A.; Schwaiger, Roland; Huber, Reinhold. Evolving Topologies of Artificial Neural Networks Adapted to Image Processing Tasks. *In Proc. Of 26<sup>th</sup> Int. Symp. On Remote Sensing of Environment*, pp. 71-74, Vancouver, BC, 1996.
- [6] Mangasarian, O.; Wolberg, W. Cancer Diagnosis via Linear Programming. *SIAM News*, 1990.
- [7] Gorman, R.P.; Sejnowski, T.J. Learned Classification of Sonar Target Using A Massively-Parallel Network. *IEEE Trans. On Acoustics, Speech, and Signal Processing*, Vol. 36, pp. 1135-1140, 1998.
- [8] Baillie, C.; Michalakes, J.; Skalin, R. Regional Weather Modeling on Parallel Computers, 1997, on-line, accessed on November, 15 2001.
- [9] UP IPM California Weather Database, on-line, accessed on December 5, 2001, <http://www.ipm.ucdavis.edu/WEATHER/weather1.html>.
- [10] Kitano, H. Designing Neural Networks Using Genetic Algorithms with Graph Generation System. *Complex Systems*, Vol. 4, pp. 461-476, 1990.
- [11] Kusçu, Ibrahim; Thorton, Chris. Design of Artificial Neural Networks Using Genetic Algorithms: Review and Prospect, 1994,
- [12] Chrisochoides, Nikos; Droegemeier, Kelvin; Fox, Geoffrey; Mills, Kim; Xue, Ming. A Methodology For Developing High.