American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-7, Issue-7, pp-219-227
www.ajer.org
Research Paper Open Access

Predicting Hurricane Intensity *Via*AFast Convergent Artificial Neural Network

Dr. Dolores De Groff[1], Dr. Perambur Neelakanta[2]
[1,2](Department of Computer & Electrical Engineering & Computer Science/ Florida Atlantic University, USA)
CorrespondingAuthor: Dr. Dolores De Groff

ABSTRACT:*Motivated by the persistent need to know the status of a hurricane instant-by-instant (almost on real-time basis) as regard to its intensity profile, with a fair degree of accuracy, proposed in this study is a novel method towards forecasting hurricane wind speed/intensity using a fast convergent, artificial neural network (ANN). A compatible architecture is indicated for the test ANN with ad hoc suites of training/prediction. Relevantly prescribed design on training and prediction schedules relies on a dynamic learning-rate algorithm deduced in terms of eigenvalues of a Hessian matrix associated with the input-data changing temporally with the progression of the hurricane. The underlying exercise includes the use of a traditional multilayer ANN architecture with feed-forward and backpropagation techniques plus a newly proposed fast-convergence algorithm; and, the test ANN forecasts fairly accurateresults of predicted wind speeds. The proposed strategy is applied to the data of Hurricane Wilma (2005, USA), after the neural network has been trained with subsets of historical hurricane data. The predicted results are cross-verified with actual wind speeds availed from the weather-center. This neoteric approach (exemplified via its potential application) has not been hitherto explored to the best of authors' knowledge.*

***KEYWORDS** –Artificial neural network, Backpropagation algorithm, Eigenvalues, Fast learning rate, Hurricane profile forecasting*

Date of Submission: 04-07-2018 Date of acceptance: 19-07-2018

I. INTRODUCTION

The scope of this study is to forecast hurricane intensity *via* an artificial neural network (ANN) that enables a quick convergence with accurate predictions.

An ANN is a mathematical model evolved as a computational tool based on the image of the biological neural complex as elaborated in [1] by Neelakanta and De Groff. A supervised training is based on the Hessian matrix of the relevant input data. In the present study such data consists of details supplied by the National Hurricane Center (NHC). That is, training the ANN involves inputting a plurality of parameters about several historical hurricane tracks. Hurricane Frances (2004, USA) , Hurricane Charley (2004,USA), Hurricane Jeanne (2004, USA) were used for training with the goal of predicting wind speeds of Hurricane Wilma (2004,USA).All of these hurricanes have a common connection in that theyalloccurred in near geographical regions [2,3,4,5]. The desired output is sought comparable to a supervisory teacher standard. A weight set is then determined based on the interconnections between the neuronal layers. That is, the artificial neural network refers to a mathematical technique that maps from an input space to an output space. The goal of supervised training is to update the weights iteratively to minimize the error which is the difference between the actual output vector of the network and the desired output vector.

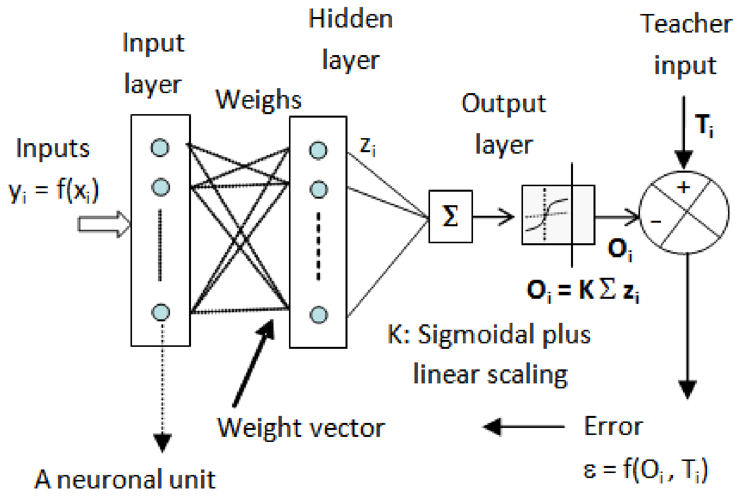

Figure 1: Test ANN architecture constructed with 9 input neuron units (NU), 1 hidden layer with 9 NUs, 1 output unit, hyperbolic tangent sigmoid

The test ANN simulation described here refers to a multilayered feedforward perceptron (MLFP) made of an input layer, a single hidden layer and a single output (Figure 1). The values addressed at the input units progress *via* interconnected inner and hidden layers and the summed output is squashed by a nonlinear sigmoid to assume a limited level; and, the sigmoid-compressed output is finally compared against a teacher value (representing the desired output objective). The resulting error is sensed and applied to the interconnection weights by a backpropagation gradient algorithm. That is, , the sigmoid-compressed value indicates an output, **O** which is compared against a teacher/supervisory (reference) value, **T** (representing the desired output objective). The resulting error corresponding to (**O** − **T**) expressed in terms of an error function such as ε denoting the mean-squared value of (**O** − **T**) is then sensed and backpropagated. The *backpropagation* (BP) algorithm adopted facilitates typically a steepest-descent based gradient that modifies the weight vector values (either to increase or decrease), when the error function is applied to the interconnection weights, $\mathbf{W}_{ij}$.

In Figure 1, the summed value ($\Sigma \mathbf{z}_i$) at the output unit corresponds to r^{th} ensemble ofinputs $\{y_i\}$, weighted across the input-layer (with i = 1, 2, 3, ..., I = 9 units) and the hidden-layer (with j = 1, 2, 3, ..., J = 9 units). Its linearly-scaled value, $K\times \Sigma \mathbf{z_i}$ (with K denoting a linear-scaling constant) is subsequently squashed by a hyperbolic tangent (sigmoidal) function, f(.), yielding an output, $\mathbf{O_i} = \mathbf{f}(K\times \Sigma \mathbf{z_i})$. Further, the coefficients of weighting of interconnections between input- and hidden-layers are specified by the set: $\{\mathbf{W}_{ij}\}$. The topology of the ANN, in addition includes supervised learning and backpropagation facilitated by a teacher value T_i as illustrated. An error ε, which is a function of ($\mathbf{O}_i$,$\mathbf{T}_i$) is deduced and its gradient is specified by: ($\pm \Delta W_{ij}$): and, the entity, $\alpha \times (\pm \Delta W_{ij})$ is then applied iteratively so as to modify the existing value of W_{ij} until, the error (ε) reduces to zero or a prescribed low, 'stop' value. Here, α denotes a *learning coefficient* that can be chosen to achieve a desirable (fast) convergence rate of the iteration imposed.

In summary, sensing the output error and applying it on the interconnection weights of the test ANN (as shown in Figure 1) is done iteratively, until the error approaches zero (or to a specified stop-value). That is, to begin with the coefficients of interconnection (or, the weights) are prescribed with a random set of (uniformly-distributed) values, in the range (– 1 to + 1). Then, the interconnection weights are updated *via* backpropagation using several ensemble of input-sets, r = 1, 2, 3, ..., R. For each ensemble input, the ANN is rendered to a state of convergence *via* iterative backpropagation of the error. This procedure is called *training* or *learning* phase. When all the ensembles of the training (input) set are exhausted, the net is conformed to a final set of "trained/learned" interconnection weights specified by the matrix, $[\mathbf{W}_{ij}]_{tr}$. That is, the net is now presumably "trained" and ready to take a fresh set of input data so as provide a (sub) optimal solution or an output comparable to the desired target depicting the objective value, set forth by the supervisory reference. In other words, subsequent to training phase, the test ANN with its trained interconnection weights (specified by the matrix, $[\mathbf{W}_{ij}]_{tr}$) is ready to accept a set of new input values, for which the prediction output is sought. Hence, with a new input set, the test ANN can be run with the $[\mathbf{W}_{ij}]_{tr}$ values; and, using the same teacher value, $\mathbf{T}_i$ (adopted as a supervisory reference in the training phase), under the converged state, the test ANN yields an output ($\mathbf{O}_i$) value depicting the output being predicted for the new input set addressed. This refers to the *prediction phase* of the test ANN.

Typically backpropagation algorithm is considered as a powerful tool for training feedforward neural networks. However, since it applies the steepest descent method to update the weights, it suffers from slow convergence rate and may yield suboptimal solutions only [6]. As such, in this work, an alternative procedure is suggested which increases the rate of convergence. Applying this speed-up technique results in the number of iterations required to train the net being much smaller [7]. The objective of this learning process is to adjust the weights of the network so as to minimize the average squared error, ε. For a net of input size L, the associated T being the teacher (desired) output, O the network output, y the network input, the squared error (cost) function is given by [8]:

$$\varepsilon = \frac{1}{2L}\sum_{n=1}^{L}\left(T^{(n)} - O^{(n)}\right)^2 \tag{1}$$

where $[\mathbf{O}]_i$= [f(K×Σ$\mathbf{z_i}$)], (Σ$\mathbf{z_i}$) depicts the sum of the outputs from the hidden neuronal units and, K is a prescribed linear scaling-constant on the computed sum. Further, [f(K × Σ$\mathbf{z_i}$)] implies a squashing transfer function imposed on the linearly-scaled sum, **(**K × Σ$\mathbf{z_i}$**)**], so that the resulting output remains limited (typically between ± 1). Among various possibilities of choosing this squashing function, f(.), depicting a simple hyperbolic tangent (sigmoidal) function [1] is adopted in the present study.w denotes the weight of interconnection between the input and hidden layer. Further, relevant to each iteration of backpropagated error, changing the value of w, the following relation holds good:

$$w(new) = w(old) - \mu\frac{d\varepsilon}{dw} \tag{2}$$

with μ being a constant of proportionality.

$$\frac{d\varepsilon}{dw} = -\frac{1}{L}\sum_{n=1}^{L}\left(T^{(n)} - O^{(n)}\right)y(n) \tag{3}$$

Equation (3) can be further rewritten:

$$\frac{d\varepsilon}{dw} = -\frac{1}{L}\sum_{n=1}^{L}T^{(n)}y^{(n)} + \frac{1}{L}w\sum_{n=1}^{L}y^{(n)}y^{(n)} \equiv -\rho + Hw \tag{4}$$

where

$$\rho = \frac{1}{L}\sum_{n=1}^{L}T^{(n)}y^{(n)} \tag{5}$$

$$H = \frac{1}{L}\sum_{n=1}^{L}y^{(n)}y^{(n)} \tag{6}$$

The derivative of e is zero at the minimum yields the optimum weight as:

$$w_{opt} = \frac{\rho}{H} \tag{7}$$

From equation (4) and applying the analysis to multidimensional case, the Hessian is specified as the average over all inputs of y^Ty,wherey^T is the transpose of y. It signifies the shape of the cost surface. The eigenvalues of H denote a measure of the steepness of the surface along the curvature directions. A large eigenvalue would signify steep curvature and that a small learning rate is needed. That is, the learning rate should be inversely proportional to the eigenvalue. Since a single learning rate is required, it is chosen such that it will not cause divergence along the steep directions (pertinent to large eigenvalue directions). Thus, a learning rate is chosen that is of the order of ($1/\lambda_{max}$), where λ_{max} is the largest eigenvalue of the Hessian matrix.

In summary, the supervised training proposed here for the test MLFP architecture is based on specifying a Hessian matrix of the relevant data applied on the input neurons interconnected to equal number of neurons of a hidden layer (I = J in Figure 1). That is, in Figure 1, there are I=9 input units y= {y_1,y_2,…y_I},. The Hessian matrix corresponds to y^Ty which is an I x I (9 x 9 in the present example) square matrix; and, this Hessian matrix can be put into a diagonal form [HD]. Because of the symmetry of the Hessian matrix, it has a unique, single eigenvalue, λ_{II} in the diagonal form as shown below (all other eigenvalues are zero):

$$[\mathbf{HD}] = \begin{bmatrix} 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & & 0 \\ & & & & \\ 0 & 0 & 0 & & \lambda_{II} \end{bmatrix} \tag{8}$$

In the proposed backpropagation model, the learning-rate (α) applied on the test ANN, would correspond to the largest, single eigenvalue, λ_{II} of the Hessian matrix as above so that, a fast convergence towards a desired extent of output prediction is feasible; as such, in the present study, this algorithmic suite is prescribed to the test ANN with the input data pertinent to each ensemble of the test hurricane profile being studied. The hurricane data considered refers to the details supplied by the NHC and gathered while tracking Hurricanes Frances, Charley, Jeanne, and Wilma [2,3,4,5]. Thus, the efficacy of the study performed here using the proposed ANN-based method and the results predicted on hurricane forecasts are cross-verified against details of an actual hurricane-tracking and prediction records of a past hurricane (namely, Hurricane Wilma).

In this paper, hurricane status of intensity profile is examined by using the proposed neural network model. This method includes the novel approach of calculating the learning rate by finding the largest eigenvalue of the Hessian matrix. It is shown that convergence towards accurate output prediction is realized quickly.

II. ANN HURRICANE INTENSITY PREDICTION METHOD

2.1 Description of the Data Set and Motivation for the ANN to Predict Hurricane Intensity

As is well known, a hurricane is a tropical cyclone with a wide range of motional classifications with stratified levels of maximum sustained wind speed.Thus, hurricane refers to a product of the tropical ocean and atmosphere steered by easterly trade winds, temperate westerlies, intensely powered by heat from the sea, manifesting as winds of growing velocity and ferocious energy. Adjunctly, as the winds grow the seas assume violent states; and, the sweeping wind may spawn into tornadoes and produce torrential rains and floods.

Systematic studies on such hurricanes have led to classifying hurricane winds. Known as Saffir-Simpson Hurricane Intensity Scale [9] relevant details indicate the level of damage caused by the underlying category of hurricane. On the lower side, the scale referred to as category 1 (74-95 mph) specifies the hurricane intensity that would cause some damage. Likewise, the dangerous wind speeds are specified in the increasing scale of 2, 3, 4, and 5. The category 5 refers to wind speeds of 156 mph or higher with associated plausible damages being catastrophic. Thus, hurricane manifests as a certain category of status and proliferates across a vast geographical area covering the sea/ocean and the land.

It is an indicated scientific interest and societal need to track continuously the hurricane status spatially and temporally. Relevant prediction or forecast of hurricane movement and the associated status parameters, mainly the wind speed, is crucial so that proactive measures can be taken towards minimal damages and life-saving.

Given the data availed across the proliferating progression of a hurricane , the strategy of predicting the hurricane status conforms to a stochastic forecasting effort. That is, the *ex post* details obtained in the yester-hours can be judiciously adopted to formulate an *ex ante* forecast track on the hurricane progression.

Hence, proposed here is a methodology to predict the hurricane status *ex ante*, with the past data availed *ex post*. Relevant approach conforms to a novel artificial neural network (ANN) based technique. The study pursued thereof emulates an ANN that has fast convergent capabilities.

This ANN method overcomes certain shortcomings in the existing hurricane prediction systems. Currently, satellites are used to obtain the data needed to track a hurricane. Data generated by the satellites is used to create physical models for forecasting purposes. Statistically based techniques are used to generate the forecasts. Several problems exist with this approach. First, initial information provided by the remote sensors on the satellites is sparse, making the prediction process more formidable in the early stages of the hurricane's formation and movement. Second, the measured parameters like wind speed and location have a great deal of uncertainty. Third, forecasting has a subjective component making the requirement for skill and experience a necessity. Using the ANN method to predict wind speed is an improved method because it has an inherent capability to incorporate historical data in the prediction process.

2.2Prescription of the Teacher Value

The data used to train the neural network comes from the information about historical hurricanes compiled from the National Hurricane Center [2,3,4,5]. The information which is inputted into the input layer of the neural network comprises historical information first obtained by air crafts about the location of the center of the hurricane (latitude and longitude), the pressure in the center of the hurricane, the average wind speed of the hurricane, the corresponding sea surface temperature, the direction of movement of the hurricane (e.g., north, south, east, west) . In signifying direction, for example, if the hurricane traveled only westerly, the West direction would be set to 1, other directions set to 0.

Historical data from hurricanes in the same nearby geographical regions compiled by the National Hurricane Center was used to train the ANN. That is, for this research work the goal is to attempt to predict wind speeds of Hurricane Wilma (2005, Florida region, USA). To accomplish this task, data from Hurricane Frances (2004, Florida region, USA), Hurricane Charley (2004, Florida region, USA), Hurricane Jeanne (2004, Florida region, USA), and twenty beginning data points for Hurricane Wilma (Data #1 through Data #20) were inputted into the ANN in an effort to train it.

Table 1 summarizes the initial data procured on Hurricane Wilma Measurements #1 to #20 *(ex post* data):

Table 1: Initial Data Procured on Hurricane Wilma

Hurricane Parameters									
Data #	Location		Pressure (mb)	Wind Speed (kt)	Sea-Surface Temp (Deg. C)	Direction			
	Lat	Long				S	N	E	W
1	17.6	-78.5	1004	25	31	0	0	0	1
2	17.6	-78.8	1004	25	31	1	0	0	1
3	17.5	-79	1003	30	31	0	0	0	1
4	17.5	-79.2	1003	30	31	0	0	0	1
5	17.5	-79.4	1002	30	31	1	0	0	1
6	17.4	-79.6	1001	30	31	1	0	0	0
7	16.9	-79.6	1000	35	31	1	0	0	1
8	16.3	-79.7	999	40	31	1	0	0	1
9	16	-79.8	997	45	31	1	0	0	1
10	15.8	-79.9	988	55	31	1	0	0	0
11	15.7	-79.9	982	60	31	0	1	0	1
12	16.2	-80.3	979	65	31	0	1	0	1
13	16.6	-81.1	975	75	31	0	0	0	1
14	16.6	-81.8	946	130	31	0	1	0	1
15	17	-82.2	892	150	31	0	1	0	1
16	17.3	-82.8	882	160	31	0	1	0	1
17	17.4	-83.4	892	140	31	0	1	0	1
18	17.9	-84	892	135	31	0	1	0	1
19	18.1	-84.7	901	130	31	0	1	0	1
20	18.3	-85.2	910	130	31	0	1	0	1

The data was then normalized to the ranges of 0 to 1. This was done having in mind the training procedure. The latitude and the longitude were normalized by adding 180 degrees and then dividing by 360 degrees. The pressure was normalized by dividing by the standard atmospheric pressure. The wind speed was normalized by dividing by the highest wind speed that hurricanes can theoretically achieve. The sea surface temperature was normalized by dividing by the warmest gulf sea-surface temperature recorded. Hurricane normalized parameters are shown in Table 2 for Measurements #1 to #20 (*ex post* data).

Table 2: Initial Normalized Data Procured on Hurricane Wilma

Hurricane Parameters (Normalized)									
Data #	Location y_1, y_2		Pressure (mb) y_3	Wind Speed (kt) y_4	Sea-Surface Temp (Deg. C) y_5	Direction y_6,y_7,y_8,y_9			
	Lat	Long				S	N	E	W
1	0.5489	0.2819	0.9909	0.1438	0.96875	0	0	0	1
2	0.5489	0.2811	0.9909	0.1438	0.96875	1	0	0	1
3	0.5486	0.2806	0.9899	0.1726	0.96875	0	0	0	1
4	0.5486	0.2800	0.9899	0.1726	0.96875	0	0	0	1
5	0.5486	0.2794	0.9889	0.1726	0.96875	1	0	0	1
6	0.5483	0.2789	0.988	0.1726	0.96875	1	0	0	0
7	0.5470	0.2789	0.987	0.2014	0.96875	1	0	0	1
8	0.5453	0.2786	0.986	0.2302	0.96875	1	0	0	1
9	0.5444	0.2783	0.984	0.2589	0.96875	1	0	0	1
10	0.5439	0.2781	0.9751	0.3165	0.96875	1	0	0	0

11	0.5436	0.2781	0.9692	0.3452	0.96875	0	1	0	1
12	0.5450	0.2769	0.9662	0.374	0.96875	0	1	0	1
13	0.5461	0.2747	0.9623	0.4315	0.96875	0	0	0	1
14	0.5461	0.2728	0.9337	0.7481	0.96875	0	1	0	1
15	0.5472	0.2717	0.8804	0.8631	0.96875	0	1	0	1
16	0.5481	0.2700	0.8705	0.9206	0.96875	0	1	0	1
17	0.5483	0.2683	0.8804	0.8055	0.96875	0	1	0	1
18	0.5497	0.2667	0.8804	0.7768	0.96875	0	1	0	1
19	0.5503	0.2647	0.8893	0.748	0.96875	0	1	0	1
20	0.5508	0.2633	0.8981	0.748	0.96875	0	1	0	1

The test ANN architecture shown in Figure 1 needs a teacher value (Ti) in both training as well as in prediction phase. It is prescribed as follows: It is surmised in this study that the teacher value can be taken commensurate with the dynamics of the hurricane parameters. For example, considering the wind-speed of the hurricane, it represents an emphasized hurricane parameter that changes almost instant-by-instant epochs. Therefore, given the set of nine normalized hurricane-decisive parameters available for training the test ANN (latitude y_1, longitude y_2, pressure at center y_3, wind speed y_4, sea-surface temperature y_5, directions y_6, y_7, y_8, y_9 pertinent to each ensemble), the teacher value can be taken as equal to the wind-speed in the next epoch (for example, six hours later). This parameter constitutes the fourth item of the nine inputs and it depicts a salient representative reference of the hurricane profile at any epochal instant. (The same procedure of adopting the wind-speed parameter as the teacher value is pursued in the prediction phase also). Thus, in the proposed test ANN, the teacher value would change dynamically with respect to each ensemble considered.

2.3Prescription of the Learning Coefficient and Adjustment of the Weights

As described earlier, in order to implement the backpropagation of the output error, the test ANN needs a learning-rate coefficient α, which can be incorporated into the constant, μ (of equation 2). Proposed here thereof is to consider an optimal learning-rate (α) so as to realize a fast convergence of the net. That is, while envisaging the backpropagated error applied to update the interconnection weights, as per the schedule, w(new) = w(old) $\pm$ $\mu \times$ ($d\varepsilon/dw$), the prescribed learning-rate should enable the weight coefficients $\{w_{ij}\}$ to assume values such that, the output error, ε converges rapidly to zero (or a prescribed very small, "stop" value). Relevant learning rate α is deduced as outlined earlier and summarized below:

In the present study, the above algorithmic suite is prescribed to the test ANN; and, considering the input data pertinent to each ensemble of the test hurricane profile being studied with reference to NHC data on the Hurricane Wilma [5] and, constructing a corresponding transpose $[y_{1,} y_{2, \ldots,} y_9]^T$, a [9 × 9] Hessian matrix [**H**] is specified as follows: [**HD**] = $[y_{1,} y_{2, \ldots,} y_9]^T$. $[y_{1,} y_{2, \ldots,} y_9]$. Since each input data set consists of nine (normalized) hurricane parameters (as listed in Table 2), corresponding [**HD**] refers to a symmetric square [(I = 9) × (I = 9)] matrix yielding one non-vanishing eigenvalue as a diagonal element. That is, for each ensemble data set there is one non-vanishing eigenvalue that remains as listed in Table 3 for the input data of Hurricane Wilma.

Table 3: Computed Hessian eigenvalues, λ_{99} for the ensemble data sets {r = 1, 2, 3, ..., 20}

r	1	2	3	4	5	6	7	8	9	10
λ_{99}	3.3219	4.3214	3.3279	3.3276	4.3254	3.3228	4.3301	4.3386	4.3477	3.3626
r	11	12	13	14	15	16	17	18	19	20
λ_{99}	4.3699	4.3857	3.4244	4.7424	4.8317	**4.9171**	4.7351	4.6902	4.6617	4.6774

As indicated earlier, the desired learning-rate can be set inversely proportional to the maximum eigen-value of the data set. From Table 3, this maximum eigenvalue, λ_{max}, occurs for the ensemble, r =16 and has a value, λ_{max} = 4.9171. Hence, corresponding learning-rate α is: $(1/\lambda_{max})$ = 0.20337.
Similarly, as discussed previously, the learning-rate changes dynamically depending on historical data fed into the neural network. Data from Hurricanes Frances, Charley, and Jeanne led to learning-rates of 0.2140, 0.2126, and 0.2232, respectively (during the training phase of the ANN).

As discussed earlier, the learning rate adopted in the aforesaid training schedules refer to $\alpha = 1/\lambda_{max}$. The efficacy of using ($\alpha = 1/\lambda_{max}$) for the learning-rate, (*in lieu* of traditional, arbitrarily specified value say, α = 0.001) and changing it on trial-and-error basis towards realizing a fast-convergence, is demonstrated by deducing relevant learning-curves. Examples of such comparisons *via* learning-curves obtained for an exemplar training ensembles of, r = 1, 2, are presented in Figure 2 considering training phase for predictions on Hurricane Wilma.

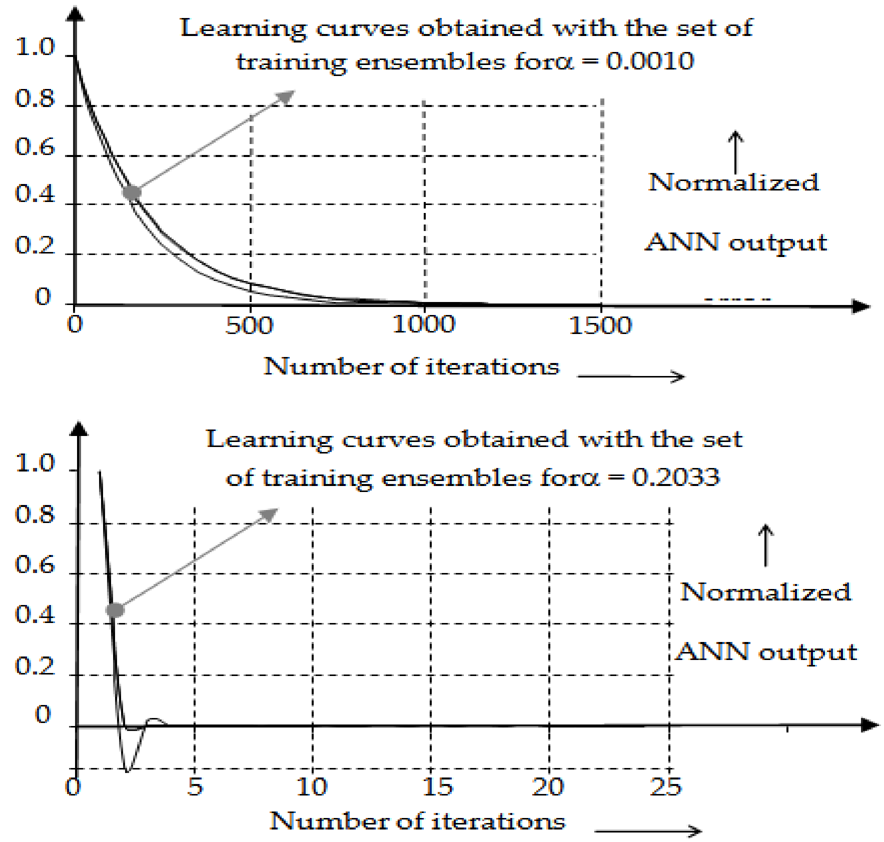

Figure 2. Learning curves obtained with the set of training ensembles r = 1, 2using an arbitrary learning coefficient, α = 0.0010 and adopting an optimum value of α = 0.2033 (as deduced by the proposed method)

III. RESULTS

The following Table 4 shows the predicted hurricane wind speed (output of the ANN) for Hurricane Wilma for Data #21 through Data #29:

Table 4: Predicted Wind Speed Compared to Actual Wind Speeds of Hurricane Wilma

Data #	Actual Wind Speed (kt)	Predicted Wind Speed (kt)	Percentage Error
21	130	130.5434	0.42%
22	130	131.0913	0.84%
23	130	131.5663	1.20%
24	125	126.9270	1.54%
25	120	122.1154	1.76%
26	130	131.4840	1.14%
27	120	122.4217	2.02%
28	115	117.9913	2.60%
29	110	113.4681	3.15%

These predicted results are very close to actual historical data. The ANN has been trained with historical hurricane data for hurricanes in the nearby geographical regions and also trained with beginning data (Data #1 through Data #20) of Hurricane Wilma. The ANN was then successfully able to predict wind speeds for Wilma (Data #21 to Data #29). The data supplied to the neural network is updated at predetermined intervals. The information from the output neuron represents a value of predicted hurricane wind speed several hours (for example, six hours) in advance of the hurricane.

In this study, nine inputs, nine hidden layer neurons, and one output were used. However, the number of input, output, and hidden layer neurons can be varied to improve the forecasting ability of the neural network.

IV. CONCLUSION

The current study has demonstrated a promising application of ANN for predicting hurricane wind speed that is particularly essential so that the public can be accurately warned against life-threatening conditions. Results obtained (shown in Table 4) are very promising. As the data verifies, the ANN is quite accurate in forecasting for future time intervals of wind speed. Referring to Table 4, ANN prediction of wind speed is very close to measured historical values for data ensemble set #21 through data ensemble set #29.

Pursuant to a summary of details on the study performed as outlined above, relevant to the major queries posed earlier, the conclusive remarks and response can be listed as follows:

- ANN can be used for hurricane intensity on prediction. The ANN method has the advantage that there is an inherent capability to incorporate historical data in the prediction process.
- It was found that the convergence rate can be speeded up by first calculating an appropriate value for the learning rate based on ($\alpha = 1/\lambda_{max}$)where λ_{max} is the maximum eigenvalue of the corresponding Hessian matrices of the input data.
- The ANN also converges to slightly more accurate values when λ_{max} is used.
- In this work, the standard Backpropagation algorithm for training feedforward neural networks has been improved based on the concept of using efficient learning rate based on λ_{max}.
- Experimental results show that the new algorithm offers much higher speed of convergence than the conventional algorithm.
- The new algorithm also causes convergence to slightly more accurate values.
- The improvement presented here can be considered as a valuable and viable alternative to using arbitrary learning rate.

V. APPENDIX: ALGORITHM ON ANN COMPUTATION

Training the neural network

a. For this purpose, sets of normalized data taken from the historical record of Hurricanes Frances, Charley, Jeanne, and Wilma are input into the neural net. The Hessian matrix is found for each data set pertaining to each hurricane. Then the learning rate is computed as the inverse of λ_{max}, where λ_{max} is the maximum eigenvalue of all of the Hessian matrices. Initially assume a set of uniformly distributed random weights (-1 to + 1). For 9 inputs and 9 neurons one requires a total weight matrix of 9×9. Assume zero bias input.

b. The output of a neuron is calculated using the following algorithmic steps:

- $Y = \Sigma W_i X_i$ where i ranges from 1 to 9.
- Multiply and calculate for each of the 9 neurons. The result will then be an output vector of 9×1 matrix with each entry corresponding to the output of one neuron.
- Apply the nonlinear (hyperbolic tangent) activation function to each of the outputs thus generating another vector of 9×1 size.
- Sum all elements in this vector and compare the result with the teacher value.
- Calculate the error.
- Adjust the weights proportional to the value of error and learning coefficient (inverse of eigmax).
- Repeat steps above till the error reduces to less than 0.001.
- Store the final weight values in an array
- Repeat steps above for all the sets of inputs. Sets of weight values (9×9) arrays are obtained after this step
- Use the weight matrix obtained after all the data sets (pertaining to Frances, Charley, Jeanne, and Wilma) have been used to train the neural net.
- Plot the change of error with respect to the number of iterations for any one input.

c. Start the testing phase

- Apply the randomized and normalized data sets 21 through 29 to the ANN.
- Calculate the outputs of each neuron by using the final weight obtained after the training phase.
- Apply the activation function to each of the outputs and sum the outputs.
- Calculate the error using the desired value being the teacher value.
- Display the output for each test as well as the error in each set.

d. End

REFERENCES

[1]. P.S. Neelakanta, D. De Groff ,*Neural network modeling: statistical mechanics and cybernetic perspectives*(Boca Raton, Fl: CRC Press, 1994).
[2]. J.L. Beven II, *Tropical cyclone report Hurricane Frances*, National Hurricane Center, 2005.
[3]. Pasch, Brown, Blake, *Tropical cyclone report Hurricane Charley*, National Hurricane Center, 2004.

[4]. Lawrence, Cobb, *Tropical cyclone report Hurricane Jeanne*, National Hurricane Center, 2005.

[5]. Pasch, Blake, *Tropical cyclone report Hurricane Wilma*, National Hurricane Center, 2005.

[6]. Magoulas, Androulakis, Vrahatis, Improving the convergence of the backpropagation algorithm using learning rate adaptation methods, *Neural Computation, 11(7*), 1999, 1769-1796.

[7]. D. De Groff, P.S. Neelakanta, Faster convergent neural networks*, International Journal of Computers and Technology*, *17(1)*, 2018, 7126-7132.

[8]. S. Haykin, *Neural networks: a comprehensive foundation* (Englewood Cliffs, NJ: Macmillan Publishing Company, 1994).

[9]. R. Rowlett, *How many? A dictionary of units of measurement* (Chapel Hill, R. Rowlett and University of North Carolina, 1998).

Dr. Dolores De Groff"Predicting Hurricane Intensity Via A Fast Convergent Artificial Neural Network.“American Journal of Engineering Research (AJER), vol. 7, no. 07, 2018, pp. 219-227.