International Journal of Engineering and Applied Sciences (IJEAS)
ISSN: 2394-3661, Volume-2, Issue-4, April 2015

Different Hybrid Neural Network in Inverse Design

K.Thinakaran, Dr.R.Rajasekar

Abstract— **Here, we investigate a different hybrid neural network method for the design of airfoil using inverse procedure. The aerodynamic force coefficients corresponding to series of airfoil are stored in a database along with the airfoil coordinates. A feedforward neural network is created with input as a aerodynamic coefficient and the output as the airfoil coordinates. In existing algorithm as an FNN training method has some limitation associated with local optimum and oscillation. The cost terms of the first algorithm are selected based on the activation functions of the hidden neurons and first order derivatives of the activation functions of the output neurons. The cost terms of the second algorithm are selected based on the first order derivatives of the activation functions of the hidden neurons and the activation functions of the output neurons. Results indicate that optimally trained artificial neural networks may accurately predict airfoil profile.**

Index Terms— **nonlinear error, airfoil design, neural networks, backpropagation, hybrid, inverse design.**

I. INTRODUCTION

This paper focuses on the comparative analysis of hybrid methods. In order to develop aircraft components and configurations, automated design procedures are needed. At present, there are two main approaches to the design of aircraft configurations. The first is direct optimization, where an aerodynamic object function, such as the pressure distribution is optimized computationally by gradually varying the design parameters, such as the surface geometry. The second is the inverse design methods. Here, a two dimensional airfoil profile is obtained for the given coefficient of lift (CL) and the coefficient of drag(CD).

But the back propagation takes long time to converge the computational effort can therefore be excessive. Some focused on better function and suitable learning rate and momentum (18-22). Details on the use of numerical optimization in aerodynamic design can be found in works by Hicks and other authors. (1-3). in inverse design methods, the aim is to generate geometry for airfoil. There are many inverse techniques in use, for example, hodo-graph methods for two dimensional flows (4-6) and other two dimensional formulations using panel methods (7-8). The above methodologies have been extended to the three dimensional case (12-14). To design fast algorithm, Abid et al. proposed a new algorithm by minimizing sum of squares of linear and nonlinear errors for all output (22). Kathirvalavakumar proposed new efficient learning algorithm for training ANN (21). The hidden layer and output layer was trained separately to speed up the convergence. Many constrained learning algorithm with functional constraints into neural networks have been proposed (24). Jeong et al. proposed learning algorithm based on first and second order derivatives of neural activation at hidden layers (23).

Han et al. proposed two modified constrained to obtain faster convergence (25). The additional cost terms of the first algorithm are selected based on the first order derivatives of the activation functions of the hidden neurons and second order derivatives of the activation of the output neurons. Second one are selected based on the second order derivatives of the activation functions of the hidden neurons and first order derivatives of activation functions of the output neurons. High order techniques have one goal in mind; to increase the speed with which back propagation converges to optimal weights(16). HONNs lead to faster convergence, reduced network size and more accurate curve fitting, compared to other types of more complex NNs.
The objective of this work is to show that different hybrid neural network method for the design of airfoil using inverse procedure. The cost terms of the first algorithm are selected based on the activation functions of the hidden neurons and first order derivatives of the activation functions of the output neurons. The cost terms of the second algorithm are selected based on the first derivatives of the activation functions of the hidden neurons and the activation functions of the output neurons. In existing algorithm as an FNN training method has some limitation associated with local optimum and oscillation. Results indicate that optimally trained artificial neural networks may accurately predict airfoil profile

II. ANN TRAINING METHOD

$$n_i := g\left(\sum_j w_{ij} n_j - \mu_i\right)$$

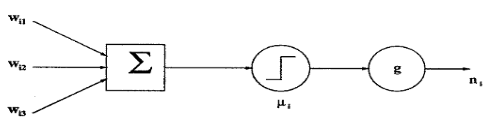

Figure 1.Schematic Diagram for a Simple Neuron

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it. Here, ni is called the state or activation of the neuron i.g() is a general non linear function called variously the activation-function. The weight wij represents the strength of the connection between neurons i and j. μi is the threshold value for neuron i, the general architecture of a two layer neural network with feed-forward connections and one hidden layer is shown in Figure3. The input layer is not included in the layer count because its nodes do not correspond to neural elements. The weighted sum of the inputs must reach the threshold value for the neuron to

K.Thinakaran, Associate Professor, Sri Venkateswara College of Engineering & Technology, Thiruvallur
Dr.R.Rajasekar, Professor, Excel Engineering College, kumarapalayam ,Erode

transmit. One drawback associated with neural networks is that it is normally very difficult to interpret the values of the connecting weights wij in terms of the task being implemented.

Neural networks offer a very powerful and general framework for representing nonlinear mappings from several input variables to several output variables. Since the goal is to produce a system which makes good predictions for new data. Training generally involves minimization of an appropriate error function defined with respect to the training set. Learning algorithms such as the back-propagation algorithm for feed-forward multilayer networks (16) help us to find such a set of weights by successive improvement from an arbitrary starting point.An airfoil profile can be described by a set of x- and y-coordinates, as illustrated in figure2.

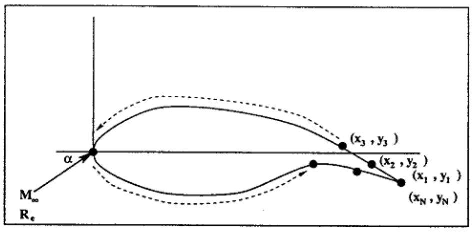

FIGURE 2. FLOW FIELD AND AIRFOIL DATA

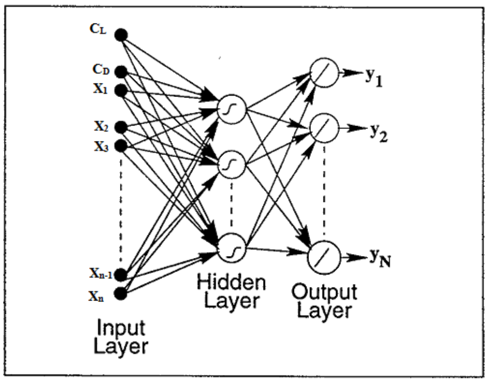

Figure3. The Neural-Network Trained to Predict Y-coordinates and Cl, Cd and X Coordinates are the inputs

The aerodynamic force coefficients corresponding to series of airfoil are stored in a database along with the airfoil coordinates. A feedforward neural network is created with input as a aerodynamic coefficient and the output as the airfoil coordinates. This is then trained to predict the corresponding surface y-coordinates We used the below sigmoidal activation function to generate the output.

$$f(x) = \frac{1}{1+e^{-x}} \quad (1)$$

For a neuron j at output layer L, the linear outputs given by Abid et al. is

$$u_j^L = \sum w_{ji}\, y_i^H \quad (2)$$

Where wji is the weight connection between the output neuron j and hidden neuron i. And y_i^H is the output of neuron i at hidden layer H.And the non linear output given by Abid et al. is

$$f(u_j^L) = \frac{1}{1+e^{-u_j^L}} \quad (3)$$

The non linear error is given by

$$e_{1j}^L = d_j^L - y_j^L \quad (4)$$

Where d_j^L and y_j^L respectively is desired and current output for jth unit in the Lth layer.

To achieve low input and output mapping the error must be reduce by derivative of cost function When the value of μ_j^L becomes larger. This procedure, the dependence of the learning function is on the instantaneous value of the total error thereby leading to faster convergence.

Cost function for first algorithm

Now, the weight update rule for the output layer is derived by applying the gradient descent method to Ep. Hence we get weight update rule for output Layer L as

$$\Delta w_{ji} = -\mu_L \frac{\partial E_p}{\partial w_{ji}}$$

Where μ_L is the network learning paramater

$$\Delta w_{ji} = -\mu_L e_{1j}^L \frac{\partial y_j^L}{\partial w_{ji}}$$

$$\Delta w_{ji} = -\mu_L e_{1j}^L \frac{\partial y_j^L}{\partial u_j} \frac{\partial u_j^L}{\partial w_{ji}}$$

$$\Delta w_{ji} = -\mu_L e_{1j}^L f'(u_j^L) y_i^H \quad (5)$$

And we get the weight update rule for the hidden layer H as

$$\Delta w_{ji} = -\mu_H \frac{\partial E_p}{\partial w_{ji}}$$

$$\Delta w_{ji} = -\mu_L e_{1j}^H f(u_j^H) y_i^F \quad (6)$$

Cost function for second algorithm

$$\Delta w_{ji} = -\mu_L e_{1j}^L f(u_j^L) y_i^H \quad (7)$$

And we get the weight update rule for the hidden layer H as

$$\Delta w_{ji} = -\mu_H \frac{\partial E_p}{\partial w_{ji}}$$

$$\Delta w_{ji} = -\mu_L e_{1j}^H f'(u_j^H) y_i^F \quad (8)$$

The network learning parameter μ is initialized, which plays an important role in minimizing the error. Then the network is trained with corresponding change of weight for both hidden and output layer.

Proposed Algorithm

In the Proposed Algorithm the network learning parameter μ is first initialized. Here, the change of weight for output layer and hidden layer is determined using new cost function equation (5)and (6) respectively.

Step1: Initialize the parameter μ to some random values
Step2: Assign Threshold value to a fixed value based on the sigmoid function.
Step3: Calculate linear output using equation (2)
Step4: Calculate Non-Linear output using sigmoid function as in the equation (3)
Step5: Calculate the below values for output layer
Calculate weight change for output layer using the equation (5)
Step6: Calculate the below values for hidden layer
Calculate the weight change for hidden layer using the equation (6)
Step7: Calculate the mean square error
Step8: If the mean square error value is greater than threshold value, then the above steps from 3 to 7 is repeated
Step9: If the mean square error value is less than threshold value, then declare that the network is trained

IV. RESULTS AND DISCUSSION

All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified. In our investigation of neural network models for inverse design, we found that satisfactory results were obtained by the cost terms selected based on the activation functions of the hidden neurons and first order derivatives of the activation functions of the output neurons. In our case, it was found that twenty hidden nodes could adequately capture the nonlinear relationship between the airfoil profiles. As mentioned previously we have a database comprised of 26 upper and lower-surface x and y coordinates, together with the corresponding coefficient of lift(CL) and the coefficient of drag(CD). There were 78 patterns in total. The main goal is to determine the airfoil profile for a given conditions. This is the "inverse" problem.

The network was trained to minimum error (using 60 training patterns) on a test set (comprising 18 patterns) which was not used in the training process. The computed profiles show good agreement with the actual profiles. The new airfoil is tested again for the same flow conditions in XFoil tool to compare Cl, Cd. In Table I, we have given the values of stored y coordinates, the values of calculated y coordinates for a pattern and also the difference between these values.

Just for sample, we have given 7 coordinates out of 26 coordinates for a pattern. From this table, we can say that the computed profiles generated during the test process show good agreement with the actual profiles.

Table I - Profile Comparison between Calculated & Stored Y Coordinates

Y coordinate in database	**Y coordinate calculated using proposed algorithm in test phase**	**Difference**
0.003391	0.00351	-0.00012
0.009775	0.00934	0.000435
0.018689	0.0182	0.000489
0.03154	0.02907	0.00247
0.046256	0.0408	0.005456
0.057878	0.05215	0.005728
0.068702	0.06176	0.006942
0.076115	0.06817	0.007945

Next we compute the convergence rate at training phase. To do this, we noted down the MSE error at each epoch and plotted it in the graph in Figure4. The red line indicates the errors of first algorithm converge. It is clear that first algorithm converges quickly and in this approach the error is less at the converging stage. It shows how the training decreases mean square Error (MSE) with the epoch. From this figure it is obvious that the first algorithm increase the converges speed and without oscillation of learning.

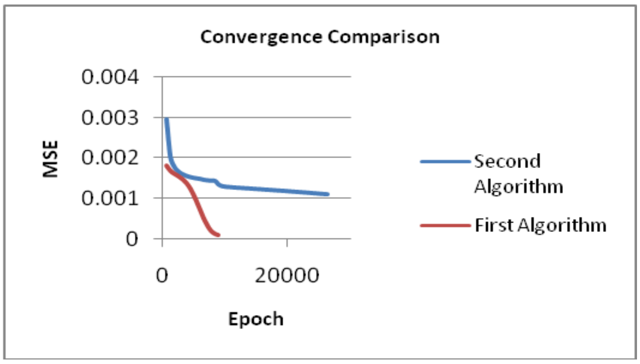

Figure4. Convergence Comparison

Next the error in each epoch obtained from first algorithm and second algorithm are compared and tabulated in Table-II. From this table, it is clear that the first algorithm predicted approximately correct airfoil without oscillation in learning. This proves that the proposed approach results in less error and takes less time to predict the airfoil for the given CL & CD.

Table II - Comparison Table

First Alg.		**Second Alg**	
Epoch	**MSE**	**Epoch**	**MSE**
0	3.086162	0	3.606326
600	0.001808	600	0.002952
1200	0.001684	1200	0.002036
1800	0.001618	1800	0.00179
2400	0.001562	2400	0.001674
3000	0.0015	3000	0.001609

3600	0.001422	3600	0.001564
4200	0.001313	4200	0.001533
4800	0.001157	4800	0.001511
5400	0.000963	5400	0.001494
6000	0.000752	6000	0.00148
6600	0.000537	6600	0.001456
7200	0.000363	7200	0.001444
7800	0.000226	7800	0.001435
8400	0.000155	8400	0.001428
9000	0.000117	9000	0.001335
		9600	0.001298
		10200	0.001285
		10800	0.001277
		11400	0.00127
		12000	0.001263
		12600	0.001257
		13200	0.001251
		13800	0.001245
		14400	0.001238
		15000	0.001232
		15600	0.001226
		16200	0.001219
		16800	0.001213
		17400	0.001207
		18000	0.0012
		18600	0.001193
		19200	0.001187
		19800	0.00118
		20400	0.001173
		21000	0.001166
		21600	0.001159
		22200	0.001152
		22800	0.001145
		23400	0.001138
		24000	0.00113
		24600	0.001123
		25200	0.001115
		25800	0.001108
		26400	0.0011

Figure5 contains the airfoils naca2013, naca2012 and naca1017 which are generated by proposed algorithm in test phase. From this figure, we can say that profile generated from proposed algorithm in test phase matches with that of stored database profiles. A measure of the accuracy of the results obtained can be inferred from examination of error which is defined as

$$Error = \frac{y_{i(actual)} - y_{i(computed)}}{airfoil\ thickness\ ratio} * 100$$

Table III - Maximum Error for Airfoil Profiles Generated by Proposed Algorithm.

Airfoil	Maximum error (%)
NACA2013	0.001798
NACA2012	0.001495
NACA1017	0.003779

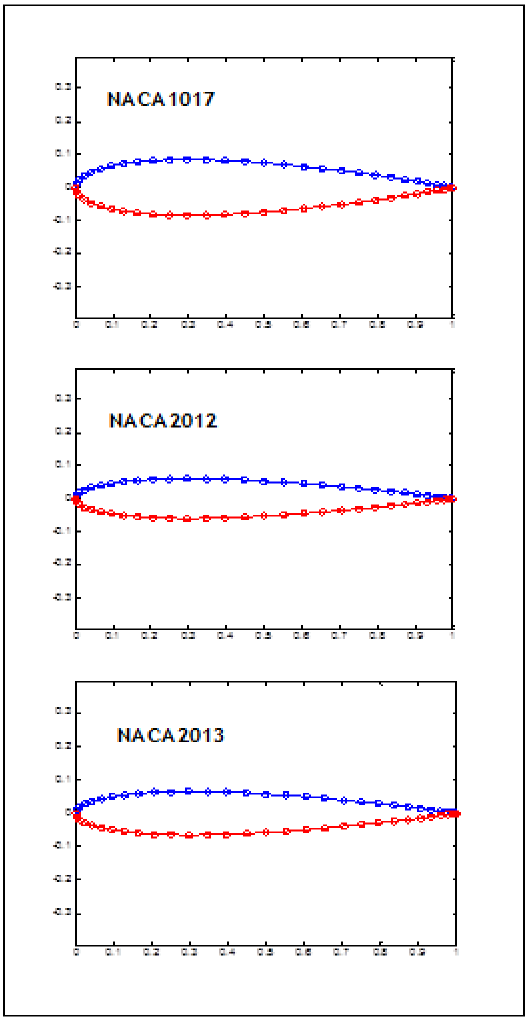

Figure5. Proposed Alg. generated Airfoils

Where $y_{i(actual)}$ is the actual y-coordinate of the section at location i, $y_{i(computed)}$ is the computed y-coordinate. Table-III shows the maximum error in percentage for the airfoil profiles naca2013, naca2012 and naca1017 which are generated by proposed algorithm in test phase. From this table III, we can conclude that our approach predicated comparatively the correct airfoil profiles.

V. CONCLUSIONS

In this paper, we have used an inverse design methodology using artificial neural networks which is used for the design of airfoil profiles. The results indicate the cost function of first algorithm increase the convergence speed. In the proposed algorithm the cost terms are selected based on the activation functions of the hidden neurons and first order derivatives of the activation functions of the output neurons. Results indicate that optimally trained artificial neural networks may accurately predict airfoil profile without oscillation in learning

.

[1] Hicks, R.M. and Vanderplaats, G.N., "Design of Low-Speed Airfoils by Numerical Optimization," SAE Business Aircraft Meeting, Wichita, April 1975

[2] Hicks, R.M. and Henne, P.A., "Wing Design by Numerical Optimization," AIAA Paper 77 1247,1977[3]. Vanderplaats, G.N., "CONMIN A FORTRAN Program for Constrained Function Minimization," NASA TMX- 62282, 1973

[3] Nieuwland, G.Y., "Transonic Potential Flow around a Family of Quasi-Elliptical Airfoil Sections,"National Luchten Ruimtevaar Laboratorium Amsterdam, NLR-TR- T.172, 1967

[4] Garabedian, P.R. and Korn, D.G.,"Numerical Design of Transonic Airfoils," Numerical Solution of Differential Equations - II, Academic Press, 1971

[5] Hoist, T.L., Sloof, J.W. Yoshihara, H. and all-haus Jr.,W.F.,z" Applied computational Transonic Aerodynamics" AGARD-AG-266,1982

[6] Sloof, J.W., "Computational Procedures in Transonic Aerodynamic Design," Lecture presented at ITCS Short Course on Computational Methods in potential Aerodynamics, Amalfi,Italy, 1982 (also NLR MP 82020 U)

[7] Ormsbee, A.I. and Chen, A.W., "Multielement Airfoils Optimized for Maximum Lift Coefficient, "AIAA Journal, Volume 10, December 1972, pp. 1620-1624.

[8] Fray, J.M.J. and Sloof, J.W., "A Constrained Inverse' Method for the Aerodynamic Design of Thick Wings with Given Pressure Distribution in Subsonic Flow," AGARD-CP.

[9] Tranen, T.L., "A Rapid Computer Aided Transonic Design Method," AIAA , June 1974, 74-501.

[10] Carlson, L.A., "Transonic Airfoil Design using Cartesian Coordinates," NASA-CR-2578, April 1976

[11] Henne, P.A., "An Inverse Transonic Wing Design Method,"AIAA , Pasadena, CA.,January 1980, Paper 80-0330.

[12] Shankar, V., Malmuth, N.D. and Cole, J.D.," Computational Transonic Design Procedure for Three-Dimensional Wings and Wing-Body Combinations," AIAA Paper 79-0344, January 1979

[13] Garabedian, P., McFadden, G. and Bauer, B.,"The NYU Inverse Swept Wing Code,"NASA CR-3662, January 1983

[14] Shankar, V., "A Full-Potential Inverse Method based on a Density Linearization Scheme for Wing Design," AIAA Paper 81-1234, 1981

[15] Hertz, J., Krogh, A. and Palmer, R.G., Introduction to the Theory of Neural Computation, Addison-Wesley Publishing Co., California, 1991

[16] Riedmiller, M. and Braun, H., "A Direct Adaptive Method for Faster Backpropagation Learning: the RPROP Algorithm," Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, March 28-April 1, 1993

[17] L.Behera, S. Kumar, A. Patnaik "On adaptive learning rate that guarantees convergence in feedforward networks", IEEE Trans. Neural Networks 17, 1116-1125, (2006)

[18] R. A. Jacobs "Increased rates of convergence through learning rate adaptation". Neural Networks 1, 295-307, (1988).

[19] G. D. Magoulas, V. P. Plagianakos, M. N. Vrahatis A. Van Ooyen, B.Nienhuis "Improving the convergence of the backpropagation algorithm". Neural Networks. 5, 465-471,(1992).

[20] X. H. Yu, G. A. Chen, S. X. Cheng "Acceleration of backpropagation learning using optimized learning rate and momomentum", Electron. Lett. 29, 1288-1289, (1993).

[21] T.Kathirvalavakumar, P. Thangavel "A modified backpropagation training algorithm for feedforward neuralnetworks", Neural Processing Letters 23, 111-119, (2006).

[22] S. Abid, F. Fnaiech, M. Najim "A fast feedforward training algorithm using a modified form of the standard backpropagation algorithm". IEEE Trans. Neural Networks 12, 424-430, (2001).

[23] R. Battiti "First and second order methods for learning: Between steepest descent and Newton's method", Neural Computation 4, 141-166, (1992).

[24] S. Y. Jeong, S. Y. Lee "Adaptive learning algorithms to incorporate additional functional constraints into neural networks", Neuro computing 35, 73-90, (2000).

[25] F. Han, Q. H. Ling, D. S. Huang "Modified constrained learning algorithms incorporating additional functional constraints into neural networks", Information Sciences 178, 907-919, (2008).

[26] K. Levenberg, "A method for the solution of certain non-linear problems in least squares", Quarterly of Applied Mathematics 2, 164-168, (1994)

AUTHOR PROFILES

Mr.K. Thinakaran received M.E., degree in computer science from Mahendra Engineering College which is affiliated to Anna University, Coimbatore, Tamilnadu in 2009. He is currently working toward the Ph.D. degree at the Anna University. He is currently a Assistant Professor in Computer Science Engineering, Sri Venkateswara College of Engineering & Technology, Thiruvallur, India. His current research interests include Neural Network and Data Mining.

Dr.R.Rajasekar received his doctorate from Department of Aeronautics, Imperial College, London, UK. His aeronautical masters' degree was from IIT, Madras. He is currently working as the Professor and Head of Aeronautical Engineering Department, Excel Engineering College, Kumarappalayam, India. (an affiliated college under Anna University, Chennai). His specialization and research interests are aerodynamics and its applications.