Defence Science Journal, Vol. 59, No. 6, November 2009, pp. 670-674

SHORT COMMUNICATION

Highly Accurate Multi-layer Perceptron Neural Network for Air Data System

H.S. Krishna

Aeronautical Development Agency, Bangalore–560 017
E-mail: krishna@jetmail.ada.gov.in

ABSTRACT

The error backpropagation multi-layer perceptron algorithm is revisited. This algorithm is used to train and validate two models of three-layer neural networks that can be used to calibrate a 5-hole pressure probe. This paper addresses Occam's Razor problem as it describes the *adhoc* training methodology applied to improve accuracy and sensitivity. The trained outputs from 5-4-3 feed-forward network architecture with jump connection are comparable to second decimal digit (~0.05) accuracy, hitherto unreported in literature.

Keywords: Back propagation, calibration, curve-fitting, error, inner product, logistic function, neuron, perceptron, pressure probe, training network, synaptic weights

NOMENCLATURE

a_j	Gain parameter of activation function
d_i	Target output values of neural network
$e_j(l)=(d_j(l)-y_j(l))$	Error
$E = ½\sum[e_i(l)]^2$	Sum of the square error
l	Iterative step, epoch
O_i	Output value of neuron in outer layer
$P=$ f(α,β,U)	Input pressure to network
U	Velocity of flow field
W	Synaptic weights
y_j	Net output from hidden layer neuron

Greek Symbols

α	Angle of attack
β	Angle of side slip
ϕ	Roll angle
δ	Backpropagation error
ε	Machine epsilon
η	Learning rate, 0.01 to 1

Subscript

k	Input layer
j	Hidden layer
i	Output layer

1. INTRODUCTION

A neural network may be trained to classify patterns in input data through feature extraction process. The training of the network model is a learning process in which the error is minimised to recognise the test pattern data as in a curve-fitting problem. The complexity of the neural network model is determined by the order of the polynomial curve to be fitted. Experience has shown that optimal network represented by third order polynomial gives better systematic representation than highly complex models with exact representation of data called Occam Razor[1,2]. In a network, the neurons are arranged in the form of layers and interlinked by synapses carrying weights either in a feed-forward or feedback network. The nonlinear activation function in a neuron induces heuristics to the learning process through statistical processing of input. Any non-polynomial (NP) function such as logistic sigmoid or hyperbolic tangent, that is differentiable and bounded, may be used as the activation function. A single neuron is limited to performing pattern classification with only two classes. A hidden layer with atleast two-neurons is required for solving the classic XOR problem[3]. Further, a three-layer network can generate arbitrarily complex boundaries such as the U-shaped curves[1] commonly occurring in many engineering disciplines, as a bound on the generalisation error. The intermediary hidden layer accomplishes this task by classifying the input patterns into linearly separable forms. Also, according to universal approximation theorem[1] for nonlinear input-output mapping, a single hidden layer is sufficient for multi-layer perceptron (MLP) to compute a uniform ε approximation to a given training set.

The foregoing discussion has important bearing on the selection of number of network layers. In the light of this context, a three-layer network architecture with a single hidden layer of neurons is implemented and its efficacy in training the neural network highly accurately for calibrating 5-hole pressure probe[4] with near linear variation in input pressure data is illustrated. The multi-hole probes have found wide application[5-7], especially, in aircraft air data system for measuring flow velocities

Received 3 December 2008, Revised 2 April 2009

at high angles of attack and each of them need to be calibrated separately[8].

2. BACKPROPAGATION ALGORITHM

The flow chart of the (EBP_MLP) algorithm[3,9] is presented in Fig. 1. Typically, a neural network model consists of input, hidden and output layers. In the l^{-th} iteration of a feedforward network, a neuron *i* driven by input signal *p* produces an output y_i that differs from actual or desired target output, d_i by a small amount, the error, $e_i(l) = d_i(l) - y_i(l)$. This error signal is then back propagated through the network from output to input layers of neurons as a sequence of corrective adjustments to the synaptic weights $\eta\Delta\delta_j p_k$ in successive iterative steps. The cost function, $E(l) = ½\ \Sigma[e_i(l)]^2$, usually referred to as batch error-correction or delta rule varies as a paraboloid with weights, W_{ij} and W_{jk} in three dimensional space, and the bottom of this bowl locates the steepest descent gradient or global minimum of cost function. To increase the speed of convergence, momentum factor, α, is multiplied by the fraction of weight change, ΔW between successive iterations, and then added as an additional term to weight update equation in each iterative step. The momentum correction ensures monotonic weight change. The computational complexity of this algorithm is O(*N*) where *N* is the number of weights and biases.

3. TRAINING DATA

It is a known fact that the error backpropagation network (BPN) generally gives good results for interpolation problems but only approximate results for extrapolation or structured prediction of data. Therefore, in order to increase accuracy, the training data should represent the entire input space or population. The training pattern is usually identified by pattern recognition techniques such as feature extraction by means of histogram or scatter plot, by solving decision boundary problems and through classification process. The training vector should cover both the higher and lower sides of the training data with minimum error output. For example, the training data for the calibration of 5-hole pressure probe here was fixed at roll angle of 2° to cover both the lower side of target, 0° as well as higher side of target of 4°, 6°, 8° and 10°.

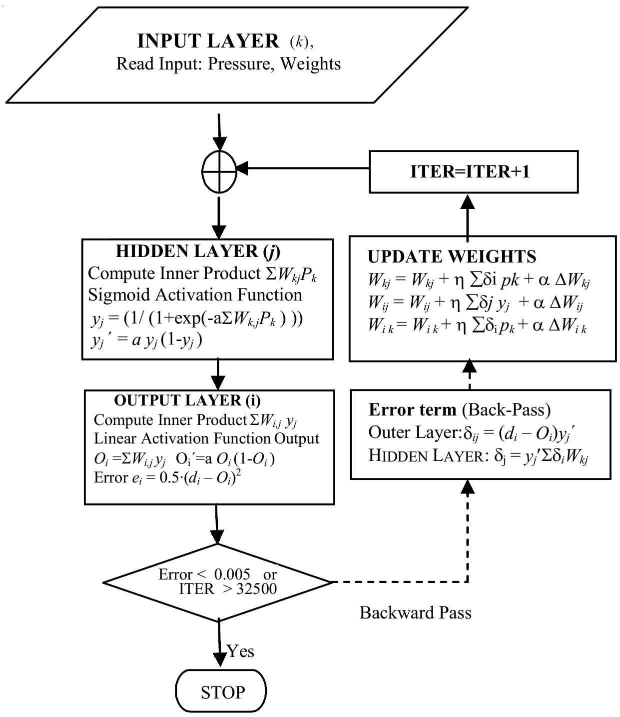

Figure 1. Flow chart of error backpropagation multilayer perceptron algorithm.

3.1 Training Methodology

The most important concern here is the overfitting of training data which may happen if the given target values are not non-dimensionalised for the required sensitivity to changes in weight, i.e., $\partial P/\partial W$, $\partial \phi/\partial P$, $\partial P/\partial \alpha$, and $\partial P/\partial U$. The sigmoidal function varies from 0 to 1 asymptotically. For this reason, the output training data is generally normalised to lie between 0.1 and 0.9. However, for maximum sensitivity and to avoid over-fitting of data, the output training values should preferably be normalised to lie in the range 0.05 to 0.27 (Table 1). Since the input and the expected output values of training vector are known apriori, supervised learning[3,9] was carried out by adjusting weights and slope of the activation function of neurons or gain parameter, a_j. Scaling can be affected through non-dimensionalisation of the input signal in small ratio, say 1:2 to 1:10 and the output values by large numbers, of the order of hundreds and even thousands (1:1000). Also, fine tuning for the range of output values can be affected by varying gain a_i of each neuron in the hidden layer. This technique greatly simplifies the task of weight adjustments and aid in smooth convergence of output pattern without having to laboriously control learning rate, momentum factor, and weights. Although, the BPN algorithm stochiastically determines the true direction of the global error minimum, sometimes it has the tendency to approximate a local minimum for global, in which case the error induced will be unusually high due to large increase in cost function for a small change in synaptic weights. This problem may be overcome by early stopping or by executing fewer iterations with acceptable error in solution.

Table 1. Linear and nonlinear regions of logistic function

X	e^{-X}	$O_I = 1/(1+e^{-X})$
-3.0	20.08	0.0474
-2.0*	7.389*	0.1192*
-1.0*	2.718*	0.2680*
-0.5	1.648	0.3775
0	1.0	0.5000
0.5	0.606	0.6224
1.0	0.367	0.7310
2.0	0.135	0.8802
3.0	0.0497	0.9525

*Highly nonlinear values

4. AIR DATA PROBE CALIBRATION

Flush air data system (FADS) of aircraft usually employs omni-directional probes that may be calibrated using wind tunnel test data as target. These are either 5-hole or 7-hole probes that serve as pressure ports. Local flow angles upto 75° have been measured accurately by using calibrated probes in separated and recirculating flows. The pressure measured at the port is related to the velocity and flow angles by the functional relationship: (U, α, β) = $f(p)$. The unknown functional relationship f was ingenuously implemented by training the neurons of the neural network. The calibrated probe is then used to measure the flow properties in an unknown flowfield.

4.1 Test cases

Two architecture models of 3-layer MLP neural networks are trained and validated as under:

(i) 5-5-3 MLP network model[10] comprising of 5 input linear neurons and a 3-output linear neurons layer

(ii) 5-4-3 MLP network model similar to previous one but with additional jump connections as shown in Fig. 2.

The input and target (output) data used here for calibration have been adapted from published work of Crowther and Lamont[4]. The input training data to the EBP_MLP program consists of a set of pressure values (P_1-P_5) as shown in Table 2. The initial weights used for batch training of 5-4-3 network model are listed in Tables 3 and 4, and program parameters are tabulated in Table 5. The ingenuity of the model permits initial training weights, W_{kj} and W_{ij}, for forward path to remain the same for the entire α range as shown in Table 3 and only the initial jump connection weights, W_{ik} need be varied with as given in Table 4.

5. RESULTS AND DISCUSSIONS

The results from 5-4-3 model of neural network trained by error back propagation using multi-layer perceptron algorithm tabulated in Table 6 comprises of roll angle ϕ, velocity U and angle of attack α. It is an optimum configuration and can be considered as an improvement over 5-5-3 model because of the greater accuracy of the results and the ease of control over training target values through jump connection weights W_{ik}. The calibration accuracies of 5-hole probe reported by Crowther and Lamont[4]; and Rediniotis and Vijayagopal[8] are between 0.3 and 0.8 units of exact target values for the output flow parameters. Crowther and Lamont[4] achieved same order of accuracies of 0.08 m/s and 0.09° as that reported here but with the larger number of pressure tappings and larger sensor network (22-11-3). In this paper, accuracies correct to second decimal place, i.e., 0.001°–0.04° and 0.07 m/s have been achieved with comparatively

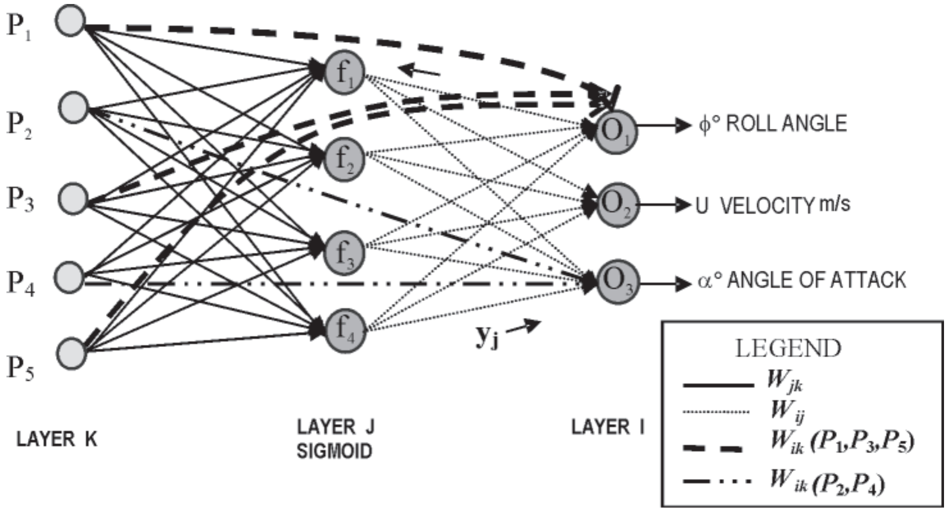

Figure 2. 5-4-3 neural network architecture model.

Table 2. Input (P) and target output (φ,U,α) values for 5-4-3 network

Data set	Target φ°	Target : α = 5°, U=32 m/s					Target: α=15°, U=32 m/s					Target: α = 25°,U=32 m/s				
		P_1	P_2	P_3	P_4	P_5	P_1	P_2	P_3	P_4	P_5	P_1	P_2	P_3	P_4	P_5
1.	0	175	85	70	60	55	180	110	56	55	54	175	140	40	39	38
2.**	2	175	85	70	70	65	178	115	70	60	55	175	141	45	40	22
3.	4	169	83	78	65	60	177	110	80	60	49	174	141	55	44	14
4.	6	170	80	85	65	60	178	105	85	60	36	172	140	65	50	7
5.	8	166	80	90	65	60	175	102	98	59	35	170	137	73	57	- 5
6.	10	161	80	100	60	50	175	100	102	50	20	169	136	82	58	-10

(*Adapted from Crowther and Lamont[4], **Training Set)

Table 3. Initial weights for input-hidden and hidden-output layers of 5-4-3 neural network

Target values : $\phi = 2°$ U = 32 m/s α= 5°, 15°, 25°					
Input-hidden layer Weight matrix W_{kj} x 99 (k=1,2,3,4,5; j =1,2,3,4)	-8.0	-2.0	-5.5	8.2	0.1
	-1.1	2.0	6.6	1.0	-1.0
	0.1	3.0	5.2	2.8	9.0
	0.03	7.4	1.1	9.0	-6.8
	-.008	0.7	-1.1	9.9	12.1
Hidden-output layer weight matrix W_{ij}x17.24 (i =1,2,3; j = 1,2,3,4)	5.9	12.3	-5.0	-6.6	5.3
	15.1	19.9	12.9	-2.1	-9.9
	5.8	10.9	1.40	-3.0	-3.8

Table 4. Input-output layer W_{ik} initial weight matrix for jump connection

Target $\alpha°$	W_{ik} : Weight values (i =1,3; k =1,2,3,4,5)				
	1-1	1-3	1-5	3-2	3-4
5°	-166	74.5	-30.1	35.3	-9.15
15°	17.9	10.5	56.0	-27.3	-22.6
25°	9.97	3.5	46.0	-22.3	-25.6

Table 5. Scaling, learning, momentum and gain parameters used in EBP_MLP program for 5-4-3 network

Non-dimensionalisation of input pressure:	$P_k = (P_k / n\text{p})$, np = 203.;
Scaling ratio for target output:	Roll angle (ϕ/nd_1), velocity U/nd_2, angle of attack, α/nd_3; nd_1 = 62.1; nd_2 = 960; nd_3 = 601; ϕ = 0°,2°,4°,6°,8°,10°; U = 32m/s; α = 25°.
Learning rate parameters:	Input-hidden layer, η= 0.001; hidden-output layers, η_o = 0.001; input-output(jump) layers, η_1= 0.091
Momentum parameters:	Input-hidden layers, α= 0.01; hidden-output layers, α_o = 0.01; input-output (jump) layers, α_1= 1.0
Gain parameter for logistic function of hidden neurons:	a_1=0.5; a_2=0.35; a_3=1.39 ; a_4 = 0.17; a_5 = 0.993
Stoping criterion or convergence criterion	Sc = 0.000082; number of iterations = 32500

Table 6. Trained results from EBP_MLP program for 5-4-3 neural network architecture

Data set	Target $\phi°$	Target: α = 5° and U=32m/s			Target: α = 15° and U=32m/s			Target: α = 25° and U=32m/s		
		$\phi°$ (Output)	U (Output)	$\alpha°$ (Output)	$\phi°$ (Output)	U (Output)	$\alpha°$ (Output)	$\phi°$ (Output)	U (Output)	$\alpha°$ (Output)
1.	0	0.0796	31.932	4.953	0.0489	31.91	14.94	0.024	31.906	24.936
2.	2	2.005	31.923	4.951	2.0052	31.92	14.95	2.004	31.924	24.950
3.	4	3.992	31.927	4.952	4.0419	31.93	14.95	4.059	31.934	24.958
4.	6	6.067	31.924	4.952	6.0262	31.93	14.96	5.996	31.942	24.964
5.	8	8.032	31.925	4.952	8.0039	31.93	14.97	8.028	31.954	24.973
6.	10	10.038	31.937	4.954	10.007	31.94	14.96	10.01	31.963	24.981

smaller size network architecture (5-4-3) and fewer tappings, mainly due to improvised training method.

6. CONCLUDING REMARKS

A single hidden layer was employed to train the 3-layer neural networks models by error back propagation algorithm. The nearly–linear wind tunnel data facilitated the use of single non–linear (logistic function) neuron hidden layer to calibrate a 5-hole probe. Accuracies correct to second decimal digit in the target output of flow angles and flow velocities were obtained for 5-4-3 model with jump connection (feed forward network). This appears to address the problem of Occam's razor which hinges on the principle that simpler model should be preferred to a complex one. The improvised training method describes threadbare, the procedure for rapid convergence and greater flexibility in control over the target outputs.

REFERENCES

1. Haykin, Simon Neural networks–A comprehensive foundation. Ed.2. PHI, 2006. ISBN-81-203-2373-4. p. 207, 101, 208.
2. Bishop, C.M. Neural networks for pattern recognition. Oxford University Press, 1999. ISBN 0 19 853864 2; p.14.
3. Freeman, J.A. & Skapura, D.M. Neural networks algorithms,

applications and programming techniques. Addison-Wesley Publishing Company, Inc, 1991. ISBN 0-201-51376-5.
4. Crowther, W.J. & Lamont, P.J. A neural network approach to the calibration of a flush air data system. *Royal Aeronau. J.*, 2001.
5. Gonsalez, J.C. & Arrington, E.A. Five-hole flow angle probe calibration for the NASA glenn icing reasearch tunnel. NASA Report No. NASA-CR 1999–202330. AIAA Paper No. AIAA-96-2201, May 1999.
6. Fingersh, L.J. & Robinson, M.C. Wind tunnel calibration of 5-hole pressure probe for applications to wind turbines. AIAA Paper No. AIAA-97-0854, 1997.
7. Parameswaran, V. Jategaonkar, R.V. & Press, M. Calibration of five-hole probe for flow angles from dynamic and tower fly-by maneuvers. *In* Atmospheric and Flight Mechanics Conference and Exhibit, 5-8 August, Monterey, California. AIAA Paper No. AIAA-2002-4627.
8. Rediniotis, O.K. & Vijayagopal, R. Miniature multihole pressure probes and their neural-network-based calibration. *AIAA J*ournal, 1999, **7**(6), 666-68.
9. Zurada, J.M. Introduction to artificial neural systems. West Publishing Co, 1992. pp. 186, 206-16.
10. Krishna, H.S. EBP_MLP Neural network algorithm and calibration of pressure probe. ADA Report No. ADA/ARD/TR/147, Nov. 2008.

Contributor

Mr H.S. Krishna obtained his BE(Mech Engg) from Bangalore University, ME (Aerospace Engg) from IISc, Bangalore, and MSc(IT) from Kuvempu University. His has worked in the areas of heat transfer, CFD, parallel computers and HPC, image processing, and computer science.