The final published version of this paper is available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5565614
Please cite this article as follows:
M. Forouzanfar, H.R. Dajani, V.Z. Groza, M. Bolic, and S. Rajan, "Comparison of feed-forward neural network training algorithms for oscillometric blood pressure estimation," 4th Int. Workshop Soft Computing Applications (SOFA 2010), Arad, Romania, July 2010, pp. 119-23.

Comparison of Feed-Forward Neural Network Training Algorithms for Oscillometric Blood Pressure Estimation

M. Forouzanfar, H. R. Dajani, V. Z. Groza, M. Bolic
School of IT and Engineering, University of Ottawa
Ottawa, ON, Canada
{mforo040, hdajani, groza, mbolic}@site.uottawa.ca

S. Rajan
Defence Research and Development
Ottawa, ON, Canada
sreeraman.rajan@drdc-rddc.gc.ca

***Abstract*— Feed-Forward Neural Network (FFNN) has recently been utilized to estimate blood pressure (BP) from the oscillometric measurements. However, there has been no study till now that consolidated the role played by the different neural network (NN) training algorithms in affecting the BP estimates. This paper compares the estimation errors in the BP due to ten different training algorithms belonging to three classes: steepest descent (with variable learning rate, with variable learning rate and momentum, resilient backpropagation), quasi-Newton (Broyden-Fletcher-Goldfarb-Shanno, one step secant, Levenberg-Marquardt) and conjugate gradient (Fletcher-Reeves update, Polak-Ribiére update, Powell-Beale restart, scaled conjugate gradient) that are used to train two separate NNs: one to estimate the systolic pressure and the other one to estimate the diastolic pressure. The different training algorithms are compared in terms of estimation error (mean absolute error and standard deviation of error) and training performance (training time and number of training iterations to reach the optimal weights). The NN-based approach is also compared with the conventional maximum amplitude algorithm.**

Keywords- blood pressure (BP); estimation; oscillometric waveforms; neural network (NN); training algorithms

I. INTRODUCTION

Blood pressure (BP) is an important component of the diagnostic tests as it contains significant information about the physiological state of a person. Among BP measurement techniques, oscillometric method offers the best opportunity for automation [1]. In this technique, a cuff with a pressure sensor is usually placed around the subject's biceps or wrist, and is slowly released from a suprasystolic to a subdiastolic pressure. As the occluding cuff is deflated, pulsatile oscillations appear on the sensed cuff pressure (CP) waveform. The amplitudes of these oscillations increase to a peak and then decrease with further deflation. It is generally accepted that the oscillation amplitudes (OAs) embedded in the CP carry most of the BP information [1].

A variety of algorithms has been used for estimation of the diastolic, mean and systolic pressures from OAs, among which the maximum amplitude algorithm (MAA) is the most common one [1, 2]. The MAA uses fixed amplitude ratios based on the oscillometric peak to determine the time points at which the CP coincides with the systolic and diastolic pressures, respectively. It has been shown [3], that the mean BP may be estimated accurately by MAA. However, due to the sensitivity of the method to variations in BP waveform, pulse pressure and arterial compliance, the systolic and diastolic pressures cannot be precisely determined. Moreover, the MAA is not capable of modeling and extracting the complex and nonlinear relationship that may likely exist between BP and the OAs [4].

A possible solution to overcome these limitations of conventional oscillometric algorithms is the application of neural networks (NNs). NNs do not require an explicit mathematical model and are thus suitable for physiological systems that defy modeling due to their non-linear nature. In this context, Baker [5] proposed a two-layer feed-forward neural network (FFNN) with steepest descent (SD) back propagation training algorithm for estimation of BP from the superficial temporal artery. In a similar effort, Narus et al. [6] trained a three-layer FFNN using SD back propagation training with momentum for estimation of BP at the supraorbital artery. Although the SD back propagation algorithm is easy to implement, it has several disadvantages such as slow learning, requiring a good training dataset, getting stuck in local minima, and providing little or no robustness to noise [7].

In this paper, we employ the NN approach to estimate BP from the wrist oscillometric measurements. An occluding cuff is placed around the subject's wrist and is deflated from a suprasystolic to a subdiastolic pressure. The oscillations within the CP signal are extracted and the OAs are represented as a function of CP. Then, two separate FFNNs are trained by using OAs samples as the input and nurse measurements as the corresponding targets to estimate systolic and diastolic pressures. Ten different training algorithms belonging to three classes: SD (with variable learning rate, with variable learning rate and momentum, resilient backpropagation), quasi-Newton (Broyden–Fletcher–Goldfarb–Shanno, one step secant, Levenberg–Marquardt) and conjugate gradient (Fletcher-Reeves update, Polak-Ribiére update, Powell-Beale restart, scaled conjugate gradient) are evaluated using a dataset collected from 85 subjects. BP estimation results using different training algorithms are compared in terms of estimation error (mean absolute error and standard deviation of error) and training performance (training time and number of training iterations). The results are also compared with the conventional MAA.

The rest of the paper is organized as follows. Section II presents various aspects of the methodology. Section III briefly reviews the different training algorithms used in this paper. Section IV contains experimental results while Section V concludes the paper.

II. METHODOLOGY

A. Oscillometric Measurement

To record the oscillometric waveforms, an occluding cuff is placed around the subject's wrist and is deflated from a suprasystolic to a subdiastolic pressure. A bandpass filter is then utilized to extract the oscillations within the CP signal. A sample wrist CP waveform along with the extracted CP oscillations is shown in Fig. 1.

B. Oscillation Amplitude (OA) Detection

As the CP oscillations contain information about the BP, in the next step, the positive and negative peaks of the oscillations are detected using a local maxima detection technique. Afterwards, the peak-to-peak values are computed and interpolated. The resultant signal after the previously mentioned processing (OA) is plotted as a function of the CP, as shown in Fig. 2. In [5] and [6], the OAs were evenly sampled at 3 mmHg increments of CP and directly fed to two separate NNs to estimate each of the systolic and diastolic pressures. In our approach, we perform an additional lowpass filtering step using a moving average filter with span of five to smooth OA before further processing. For more details on oscillation extraction and peak detection, the reader is referred to [2].

C. Feed-Forward Neural Network (FFNN)

Neural network (NN) can be considered as non-linear statistical data modeling tool that can model almost any nonlinear relationship that may exist between inputs and outputs or find patterns in data. These computational models are characterized by their architecture, learning algorithm, and activation function [7].

The feed-forward NN (FFNN) architecture is selected in this study. The FFNN consists of one or more nonlinear hidden layers along with a linear output layer. The hidden layers' activation functions are sigmoidal functions that empower the network to learn the complex and nonlinear relationship between the inputs and the targets, while the linear output layer makes it possible to have outputs of any range. In this architecture, a unidirectional weight connection exists between each two successive layers. A two-layer FFNN with sigmoidal functions in the hidden layer, and a linear output layer can potentially approximate any function with finite number of discontinuities, provided a sufficient number of neurons exists in the hidden layer [8]. In our application, two separate two-layer FFNNs with one output neuron were designed: one to estimate the systolic pressure and the other one to estimate the diastolic pressure. The hyperbolic tangent sigmoid transfer function was used in the hidden layer and linear transfer function in the output. The number of neurons in the hidden layer was set to five which was found to yield the best BP estimation results in terms of mean absolute error [9].

III. TRAINING ALGORITHMS

Training is the process of determining the optimal weights of the NN. This is done by defining a performance function (which is usually the mean square error between the network's output and the desired target) and then minimizing it with respect to weights. The minimization is performed by

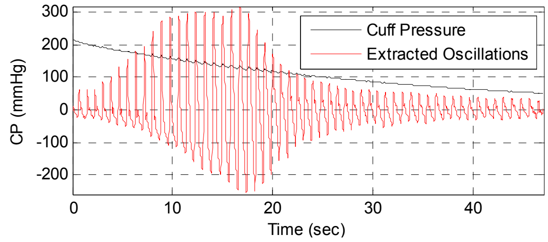

Figure 1. Sample cuff pressure (CP) waveform (black curve) along with the extracted oscillations (red curve). The oscillations are magnified for better visualization.

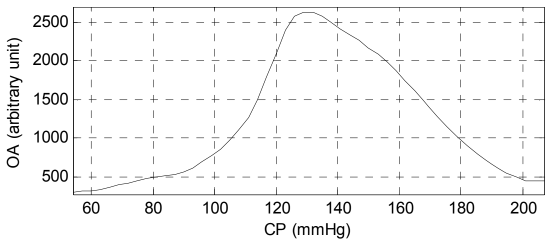

Figure 2. Sample oscillation apmplitude (OA) as a function of cuff pressure (CP)

calculating the gradient using a technique called backpropagation which can be done in batch or incremental styles [7]. In this paper, we used the batch training style. The NNs were trained using different training algorithms belonging to three backpropagation classes described as follows.

A. Steepest Descent (SD):

The SD algorithm is derived based on the first-order Taylor series expansion of the performance function. In this algorithm, the update is done in the negative gradient direction. One iteration of this algorithm can be written as follows [7]:

$$\Delta \mathbf{w}_i = -\mu_i \mathbf{g}_i \tag{1}$$

where $\Delta \mathbf{w}_i$ is the vector of weights changes, $\mathbf{g}_i$ is the vector of gradients and μ_i is the learning rate that determines the length of the weight update. Although the SD algorithm is easy to implement, it has several disadvantages such as slow learning, requiring a good training dataset, getting stuck in local minima, and having little or no robustness to noise. Heuristic modifications can be applied to improve the performance of SD algorithm.

1) *Steepest Descent with Variable Learning Rate (SD-VLR):* The performance of the NN is very sensitive to the initial choice of learning rate, especially when the rate is fixed during the whole learning process. A solution to overcome the sensitivity issue is the use of a variable (adaptive) learning rate. In the adaptive learning technique, the step size is chosen as large as possible while keeping learning stable. In each iteration, if the new error is greater than the old one by a predefined ratio (here set to 1.04), the new parameters (weights) are discarded and the learning rate is decreased (here by multiplying by 0.7). Otherwise, the new parameters

are kept. If the new error is less than the old error, the learning rate is increased (here by multiplying by 1.05). The initial value of learning rate was set to 0.01.

2) *Steepest Descent with Variable Learning Rate and Momentum (SD-VLRM):* In order to reduce the sensitivity of the network to fast changes of the error surface, a fraction of the previous weight change (called momentum term) can be added to the gradient decreasing term, as follows [7]:

$$\Delta \mathbf{w}_i = -\mu_i \mathbf{g}_i + p\Delta \mathbf{w}_{i-1} \tag{2}$$

where $p(0 \leq p \leq 1)$ is the momentum parameter. The momentum parameter p was set to 0.9 in our application.

3) *Resilient Backpropagation (RBP):* For the FFNNs with sigmoidal activation functions, the gradient can be of very small magnitude even though the weights are far from their optimum values. This is due to the slope of sigmoidal functions that approaches zero as the input magnitude increases [10]. A solution to this problem is to use only the direction of the gradient to update the weights, while the amount of the update is determined by another update factor (here initially set to 0.07). When the gradient has the same sign for two successive iterations, the update factor is increased by a ratio (here set to 1.05). The update factor is decreased (here by multiplying by 0.8) when the gradient changes sign from the previous iteration. When the derivative is zero, the update value is not changed [10].

B. Quasi-Newton (QN):

The Newton's method is based on the second-order Taylor series expansion. The iterative procedure of the Newton's algorithm is obtained as [7]:

$$\Delta \mathbf{w}_i = -\mathbf{A}_i^{-1}\mathbf{g}_i \tag{3}$$

where $\mathbf{A}_i$ is the Hessian matrix of the performance function at iteration i. Computation of Hessian matrix in each iteration is computationally expensive for FFNNs. To solve this problem, some algorithms have been proposed which are based on Newton's method but do not compute the second derivatives at each step.

1) *Broyden–Fletcher–Goldfarb–Shanno (BFGS):* In BFGS algorithm, the inverse of Hessian matrix is updated as a function of successive gradients of the performance function. The method is fully described in [11].

2) *One Step Secant (OSS):* This algorithm is same as BFGS, but based on the assumption that each iteration the preceding Hessian was the identity matrix [12]. Therefore, the OSS method does not need to store the large Hessian matrix.

3) *Levenberg-Marquardt (LM):* The LM method assumes that the performance function has a quadratic form in a region around the current search point, called trusted region. Based on this assumption, the Hessian matrix is approximated by the Jacobian matrix and the weight update is obtained as [13]:

$$\Delta \mathbf{w}_i = -[\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda \mathbf{I}]^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{e} \tag{4}$$

where $\mathbf{J}$ is the Jacobian matrix containing the first derivatives of the errors with respect to weights, $\mathbf{e}$ is a vector of network errors, and λ is a constant (here set to 0.001). When the performance function decreases, this parameter is multiplied by a factor (here set to 0.1) and when the performance function increases it is multiplied by another factor (here set to 10).

Same as the SD algorithms, in most of the QN algorithms, the step size is adjusted at each iteration. A search is performed along the new search direction to determine the step size that minimizes the performance function along that line. Here, we used the backtracking search method in all the QN algorithms [11].

C. Conjugate Gradient (CG):

The SD algorithm produces rapid function decrease along the negative of the gradient. The Newton's method is more accurate and much faster in convergence but it needs the computation and storage of Hessian matrix which may be impractical. The CG algorithm searches along the conjugate directions, which provides generally more robust convergence than the SD while it does not need the Hessian matrix. The new search direction $\boldsymbol{p}_i$ is determined as a combination of the new SD direction $\boldsymbol{g}_i$ with the previous search direction $\boldsymbol{p}_{i-1}$ [7]:

$$\mathbf{p}_i = -\mathbf{g}_i + \beta_i \mathbf{p}_{i-1} \tag{5}$$

where β_i is a constant that can be computed by various methods.

1) *Fletcher-Reeves Update (FRU):* This method updates β_i as the norm squared of the current gradient vector divided by the norm squared of the previous gradient vector [14]:

$$\beta_i = \frac{\mathbf{g}_i^{\mathrm{T}}\mathbf{g}_i}{\mathbf{g}_{i-1}^{\mathrm{T}}\mathbf{g}_{i-1}} \tag{6}$$

2) *Polak-Ribiére Update (PRU):* This method updates β_i as the dot product of the previous change in the gradient vector with the current gradient vector divided by the norm squared of the previous gradient vector [14]:

$$\beta_i = \frac{\Delta\mathbf{g}_{i-1}^{\mathrm{T}}\mathbf{g}_i}{\mathbf{g}_{i-1}^{\mathrm{T}}\mathbf{g}_{i-1}} \tag{7}$$

3) *Powell-Beale Restarts (PBR):* In all CG algorithms the search direction is periodically reset to the negative direction of the gradient. This is done usually when the number of iterations reaches the number of network parameters. In PBR approach, orthogonality between the current and the previous gradient vectors is checked at each iteration. If there is not enough orthogonality left, then the search direction is reset. The orthogonality is checked using the following inequality [15]:

$$\left|\mathbf{g}_{i-1}^{\mathrm{T}}\mathbf{g}_{i-1}\right| \geq 0.2\|\mathbf{g}_i\|^2 \tag{8}$$

4) *Scaled* Conjugate *Gradient (SCG):* The introduced CG algorithms so far are based on a line search in each iteration

Table I: Comparison of NN-based approach using different training algorithms with the MAA in terms of estimation error (MAE and SDE).

		MAA	Steepest Descent (SD)			Quasi-Newton			Conjugate Gradient (CG)			
			SD-VLR	SD-VLRM	RBP	BFGS	OSS	LM	FRU	PRU	PBR	SCG
Systolic Estimation	MAE (mmHg)	8.18	7.19	7.00	**6.90**	6.91	6.98	7.76	7.28	7.04	7.26	7.02
	SDE (mmHg)	12.03	10.71	10.10	**9.90**	10.24	10.21	10.81	10.61	10.43	10.56	10.36
Diastolic Estimation	MAE (mmHg)	7.61	6.44	6.29	**5.83**	6.32	6.02	6.54	5.95	5.88	5.91	6.24
	SDE (mmHg)	9.08	8.17	8.40	**7.34**	8.03	7.58	8.38	7.59	7.50	7.64	7.96

Table II: Comparison of different NN training algorithms in terms of training performance (training time and number of iterations to reach the optimal weights).

		Steepest Descent (SD)			Quasi-Newton			Conjugate Gradient (CG)			
		SD-VLR	SD-VLRM	RBP	BFGS	OSS	LM	FRU	PRU	PBR	SCG
Systolic Estimation	Avg. no. iterations	597	273	194	379	137	**110**	132	129	124	140
	Avg. train time (sec)	15.16	6.65	**5.53**	28.60	9.61	6.85	8.84	8.93	8.78	8.32
Diastolic Estimation	Avg. no. iterations	689	351	163	264	163	**107**	136	140	133	150
	Avg. train time (sec)	17.82	8.52	**4.82**	20.68	11.30	6.68	8.97	9.65	9.23	8.90

which is computationally expensive. To overcome this shortcoming, the SCG method combines the trust region approach (same as LM) with CG algorithm. For more details, the reader is referred to [16].

Same as the SD and QN algorithms, in most of the CG algorithms, the step size is adjusted at each iteration. Here, the Charalambous' search method [17] was utilized in all the CG algorithms except the SCG to adjust the step size.

IV. EXPERIMENTAL RESULTS

A. Dataset

Our oscillometric waveform dataset was obtained from 85 subjects aged from 12 to 80 using a digital BP monitor (Biosign Technologies Inc.). Five sets of oscillometric BP measurements were obtained from each volunteer (425 total measurements). Corresponding to each pressure waveform, two reference readings were recorded using the auscultatory method by two nurses. The average value of these two measurements was used as the reference pressure of each subject.

B. Normalization

As mentioned in Section II-B, the obtained OAs were evenly sampled at 3 mmHg increments of the CP. This led to 41 samples per each OA. Next, the OA samples were scaled in the interval [-1, 1] to make the NNs train faster, to reduce the chances of getting stuck in local minima and to avoid saturation of the network. The resultant data was then used as the input to the NNs.

C. Evalutation Criteria

The BP estimation results were compared in terms of estimation error and training performance. Two metrics were chosen to compute the estimation error: *i)* mean absolute error (MAE) that shows the overall accuracy in estimating the BP and *ii)* standard deviation of error (SDE) that is a measure of error variability. The training performance of the NNs was compared in terms of *i)* training time and *ii)* number of training iterations to reach the optimal weights.

D. Train and Test Strategy

In order to train networks to find the necessary input-output relationships without over-fitting the training dataset, the early-stopping technique was utilized [18]. The dataset was divided into three sets, called training, validation and testing data. One subject (including five measurements) was selected for the test, which led to the largest possible set of data for training (and validation). The rest of the subjects were randomly divided into training (80%) and validation (20%). The networks were trained with the training data and the validation error was checked at every iteration. When the validation error increased for a specified number of successive iterations (here set to 100), the networks' parameters that lead to the best validation performance were saved. The process was then repeated such that each subject in the dataset was used once for the test.

E. Results and Discussion

The estimation errors in terms of MAE and SDE for the FFNN using different training algorithms are shown in Table I. The estimation errors are also compared with the conventional MAA [2]. The NNs' training performance in terms of training time and number of training iterations to reach the optimal weights are listed in table II. These values were averaged over the 85 runs of the algorithms. The results are obtained on a Pentium IV 2.8 GHz processor with 2.0 GB of RAM.

Comparing the NN-based approach with the conventional MAA, it is observed that the NN-based method outperforms the MAA in terms of estimation error. For the systolic pressure estimation, the MAE and SDE are decreased up to

1.28 mmHg and 2.13 mmHg, respectively. For the diastolic pressure estimation, the MAE and SDE are decreased up to 1.78 mmHg and 1.74 mmHg, respectively. These results are achieved using the RBP training algorithm.

Among the SD training algorithms, the SD-VLR has the largest estimation error with the worst training performance. The SD-VLRM significantly improves the training performance while the estimation error is only slightly improved. The best result in terms of both estimation error and training performance is obtained using RBP.

Among the QN training algorithms, the BFGS and OSS obtain the least estimation errors. However, the BFGS is the worst in training performance with the longest training time and the maximum number of iterations. The BFGS training time is even longer than SD-VLR, but it needs less number of training iterations. This is due to the fact that QN algorithms make advantage of the second-order Taylor series expansion of the performance function which makes the training faster in terms of the number of iterations. However, since each iteration is computationally more expensive than that of the SD algorithms, the whole training time is longer. The training time has been significantly improved in the OSS algorithm with almost estimation errors similar to BFGS algorithm. The best training performance is achieved using the LM algorithm but with the cost of slight increase in estimation errors. The LM is almost the second fastest algorithm after the RBP, with the least number of training iterations among all the training algorithms.

Among the CG training algorithms, the FRU, PRU, and PBR have almost similar results both in estimation error and in training performance. They are faster than the SD-VLR, BFGS and OSS algorithms but not as fast as the advanced SD (SD-VLRM, RBP) and QN (LM) methods. The CG based algorithms have estimation errors that are almost the same as the other algorithms. The training performance is improved in SCG by employing the trust region approach while achieving comparable estimation errors to the other CG algorithms.

Comparing all the ten different training algorithms together, it is found that the estimation errors are very close while the training performances significantly differ. However, the best estimation errors and fastest training are obtained using the RBP algorithm.

V. CONCLUSION

In this paper, we utilized the NN approach to estimate BP from the oscillometric readings. After recording the oscillations within a deflating cuff, their amplitudes were represented as a function of CP. Then, two separate FFNNs were trained by samples of OAs that were evenly spaced in specific increments of the CP, with nurse measurements as the corresponding network targets to estimate systolic and diastolic BPs. Ten different training algorithms belonging to three classes: SD (SD-VLR, SD-VLRM, RBP), QN (BFGS, OSS, LM) and CG (FRU, PRU, PBR, SCG) were evaluated using a dataset collected from 85 subjects with five sets of oscillometric BP measurements for each subject. It was found that the NN-based approach achieves substantial improvements in terms of MAE and SDE compared to the conventional MAA. Among different training algorithms used to train the networks, the RBP was found to be superior in terms of both estimation error and training performance.

ACKNOWLEDGMENT

The authors would like to thank Ontario Centres of Excellence (OCE) and Biosign Technologies Inc. for supporting this work. They are also grateful to Prof. Emil M. Petriu, Prof. Wail Gueaieb, and Mr. Silu Chen for their contributions.

REFERENCES

[1] K.-G. Ng and C.F. Small, "Survey of automated noninvasive blood pressure monitors," J. Clin. Eng., vol. 19, pp. 452-475, 1994.

[2] S. Chen, V.Z. Groza, M. Bolic, and H.R. Dajani, "Assessment of Algorithms for Oscillometric Blood Pressure Measurement," in proc. IEEE Instrumentation and Measurement Technology Conference (I2MTC), Singapore, May 2009, pp. 1763-1767.

[3] F.K. Forster and D. Turney, "Oscillometric determination of diastolic, mean and systolic blood pressure–a numerical model," J. Biomech. Eng., pp. 359-364, 1986.

[4] P.D. Baker, J.A. Orr, D.R. Westenskow, and T.P. Egbert "Method for determining blood pressure utilizing a neural network," U. S. Patent 5,339,818, Aug. 23, 1994.

[5] P.D. Baker, "Neural network processing of oscillometric waveforms and blood pressure estimation from the superficial temporal artery," Ph.D. dissertation, University of Utah, Salt Lake City, UT, USA, 1990.

[6] S. Narus, T. Egbert, T. Lee, J. Lu, and D. Westenskow, "Noninvasive blood pressure monitoring from the supraorbital artery using an artificial neural network oscillometric algorithm," J. Clin. Monitor., vol. 11, pp. 289-297, Sep. 1999.

[7] J.-S.R. Jang, C.-T. Sun, and E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, NJ: Prentice-Hall, 1997.

[8] G. Cybenko, "Approximation by superpositions of a sigmoidal function," Math.Control Signals Systems, vol. 2, pp. 303-314, 1989.

[9] M. Forouzanfar, H.R. Dajani, V.Z. Groza, M. Bolic, and S. Rajan, "Adaptive Neuro-Fuzzy Inference System for Oscillometric Blood Pressure Estimation," accepted. To be published in proc. IEEE International Workshop on Medical Measurements and Applications (MeMeA), Ottawa, Canada, Apr. 2010.

[10] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in proc. IEEE International Conference on Neural Networks, San Fransisco, USA, Mar. 1993, pp. 586 – 591.

[11] J.E. Dennis and R.B. Schnabel, "Numerical Methods for Unconstrained Optimization and Nonlinear Equations," Englewood Cliffs, NJ: Prentice-Hall, 1983.

[12] R. Battiti "First and second order methods for learning: Between steepest descent and Newton's method," Neural Computation, vol. 4, pp. 141–166, Mar. 1992.

[13] M.T. Hagan and M. Menhaj, "Training feed-forward networks with the Marquardt algorithm," IEEE Transactions on Neural Networks, vol. 5, pp. 989–993, Nov. 1994.

[14] R. Fletcher and C.M. Reeves, "Function minimization by conjugate gradients," Computer Journal, vol. 7, pp. 149–154, 1964.

[15] M.J.D. Powell, "Restart procedures for the conjugate gradient method," Mathematical Programming, vol. 12, pp. 241–254, Dec. 1977.

[16] M.F. Moller "A scaled conjugate gradient algorithm for fast supervised learning," Neural Networks, vol. 6, pp. 525–533, 1993.

[17] C. Charalambous "Conjugate gradient algorithm for efficient training of artificial neural networks," IEEE Proceedings, vol. 139, pp. 301–310, June 1992.

[18] M.T. Hagan, H.B. Demuth, and M.H. Beale, Neural Network Design, Boston: PWS Pub. Co., 1995.