Vojnotehnicki glasnik/Military Technical Courier

ISSN: 0042-8469

vojnotehnicki.glasnik@mod.gov.rs

University of Defence

Serbia

Proti, Danijela D.

FEEDFORWARD NEURAL NETWORKS: THE LEVENBERG-MARQUARDT OPTIMIZATION AND THE OPTIMAL BRAIN SURGEON PRUNING

Vojnotehnicki glasnik/Military Technical Courier, vol. 63, núm. 3, 2015, pp. 11-28

University of Defence

ОРИГИНАЛНИ НАУЧНИ ЧЛАНЦИ
ORIGINAL SCIENTIFIC PAPERS
ОРИГИНАЛЬНЫЕ НАУЧНЫЕ СТАТЬИ

FEEDFORWARD NEURAL NETWORKS: THE LEVENBERG-MARQUARDT OPTIMIZATION AND THE OPTIMAL BRAIN SURGEON PRUNING

Danijela D. Protić
General Staff of the Serbian Army,
Department for Telecommunication and Informatics (J-6),
Centre for Applied Mathematics and Electronics, Belgrade
e-mail: adanijela@ptt.rs

DOI: 10.5937/vojtehg63-7529

FIELD: Telecommunications
ARTICLE TYPE: Original Scientific Paper
ARTICLE LANGUAGE: English

Abstract:

This paper presents the training, testing and pruning of a feedforward neural network with one hidden layer that was used for the prediction of the vowel "a". The paper also describes Gradient Descent, the Gauss-Newton and the Levenberg-Marquardt optimization techniques. Optimal Brain Surgeon pruning is applied to the trained network. The stopping criterion was an abrupt change of the Normalized Sum Squares Error. The structure of the feedforward neural network (FNN) was 18 inputs (four for glottal and 14 for speech samples), 3 neurons in the hidden layer and one output. The results have shown that, after pruning, the glottal signal has no effect on the model for a female speaker, while it affects the prediction of the speech pronounced by a male speaker. In both cases, the structure of the FNN is reduced to a small number of parameters.

Key words: *Levenberg-Marquardt; speech analysis; pruning; feedforward neural networks.*

Introduction

In recent years, artificial neural networks (ANNs) have become one of the most successful structures for solving diverse problems related to artificial intelligence, pattern recognition, classification, curve fitting, time series

prediction, and a wide variety of many practical problems. The multilayer perceptron (MLP) is the oldest and the most frequently used type of ANNs. A typical MLP consists of several layers of neurons. Input layer nodes correspond to the feature vector, the output layer consists of one or several neurons. If the input signal in the ANN propagates from the input layer to the output layer, the networks are called feedforward neural networks (FNNs). In order to find the FNN weights, one must minimize the cost function. In the gradient descent (GD) optimization algorithm, the derivatives of the cost function are calculated and weights are updated in an iterative manner, i.e. the error signal is propagated back to the lower layers. This technique is known as the back-propagation (BP) algorithm. The network weights are moved along the negative of the gradient to find a minimum of the error function (Silva et al., 2008), (Wu et al., 2011), (Protić, 2014). However, the GD is relatively slow and the network solution may trap in one of the local minima instead of the global minimum. The Levenberg-Marquardt (LM) algorithm gives efficient solutions of convergence and better optimization than the GD with the combination of GD and the Gauss-Newton (GN) method (Riecke et al., 2009), (Shahin, Pitt, 2012), (Levenberg, 1944), (Marquardt, 1963). The GN presumes that the error function is approximately quadratic near the optimal solution, which is based on the second order Taylor approximation of the sum of the squared errors. In this way, FNNs and the LM algorithm help address the problems of slow convergence and computation consumption caused by the structures of MLPs.

The experimental analysis, presented in this paper, was based on a prediction of the vowel "*a*", spoken by both a female and a male speaker. All experimental results were obtained by using a FNN with one hidden layer and the LM algorithm. The structure of FNN was 18 inputs, 3 hidden-layer neurons, and the output neuron (18-3-1). Activation functions were tangent hyperbolic for all neurons. Since the outputs from the tangent function were limited to the (-1, 1) interval, the signals were also normalized to [-1, 1]. The training was carried out on 1700 samples of speech signals and corresponding glottal signals. First 14 inputs to the FNN were successive samples of speech: $y_{n\text{-}1}$, ..., $y_{n\text{-}14}$ and the following four inputs corresponded to the glottal signal: $g_{n\text{-}1}$, ..., $g_{n\text{-}4}$. The result of processing in the forward direction was the predicted speech signal sample y_n. The prediction error, or residual, was used for obtaining the Sum Squares Error (SSE). After training, the resulting structure was tested on 1700 samples of independed test sets.

Furthermore, the Optimal Brain Surgeon (OBS) pruning was applied to the trained FNN. This technique reduced the network structure by removing neurons (from the hidden layer) that have not affected the total error rate. The stopping criterion was the abrupt change of Normalized SSE (NSSE). As a demonstration of the aforementioned techniques, the results were presented in figures and summarized in tables.

This paper is organized as follows: the next section describes the GD, the GN and the LM optimization algorithms, as well as the LM optimization technique for the FNN with one hidden layer. Furthermore, the experimental results are presented. The last chapter concludes the paper.

Levenberg-Marquardt optimization

The Levenberg-Marquardt (LM) algorithm can be regarded as a linear combination of the GN method and the GD method. The alternation between these two methods is called a damping strategy, and is controlled by a damping factor. If the damping parameter is large, the LM adjusts parameters like the GD method. If the damping parameter is small, the LM updates parameters like the GN method (Young-tae et al. 2011). GD, GN and LM methods are the optimization algorithms for the basic Least-Squares (LS) problem, i.e. they use LS to fit data. Fitting requires a parametric model that releases the response data to the predictor data with coefficients. The LS method minimizes the summed square of the residuals; a residual being the difference between an observed value and the fitted value provided by a model. See (1).

$$S = \sum_{i=1}^{n} r_i^2 = \sum_{i=n}^{n} \left(y_i - \hat{y}_i \right)^2 \tag{1}$$

S is the sum of squares, y_i and $\hat{y}_i$ are the observed response and its fitted value, respectively, r_i is the residual, and *n* is the number of data points included in the fit. When fitting data, residuals (errors) are assumed random and follow the Gaussian distribution with the zero mean and the constant variance σ^2 (spread of errors is constant).

The LS fitting may be linear, weighted, robust and nonlinear (that fits nonlinear model to data). In general, LS methods involve an iterative procedure for parameter optimization. The nonlinear LS methods involve an iterative improvement to parameter values in order to reduce the sum of the squares of the errors between the function and the measured data points. This is the optimization, i.e. the process of finiding the minimum or maximum value of an objective function.

When using nonlinear optimization algorithms, some choices must be made about the time of stopping the fitting process. Some of possible choices are 1) stop after a fixed number of iteration, 2) stop when the error function falls below the predetermined value, 3) stop when the relative change in the error function falls below some specified value, etc. (Bishop, 1995).

Gradient descent method

The GD method is a general minimization method which updates parameter values in the direction opposite to the gradient of the objective function. It is recognized as a highly convergent method for finding the minimum of simple objective functions (Gavin, 2013). The GD is a first order algorithm because it takes only the first derivative of the function.

Suppose that $f(x)$ is a cost function of a single parameter x. To find a minimum of f, x has to be evolved in such a way that the cost function $f(x)$ decreases. Given some initial value x_0 for x, the value of x can be changed in many directions. In higher dimensions, the analogy of the derivative is the gradient. It is a vector whose entries are partial derivatives of f with respect to each dimension. So, in order to decrease f, evaluate the gradient ∇f in a particular point, take a step in its negative direction, evaluate the gradient to the new point, get a new vector and take a step in its negative direction, and so on until the local minimum of the function is find. This gives a simple gradient descent algorithm:

– initialize x in some way, often just randomly, $x=x_0$,

– find the gradient of ∇f, and update it with a small positive amount of the step size (sometimes it is called the learning rate) in the following way:

$$x_{k+1} = x_k - \lambda \nabla f(x_k),$$

– follow the stopping condition (stopping the procedure either when f is not changing sufficiently quickly, or when the gradient is sufficiently small).

Given stable conditions (a certain choice of λ), it is guaranteed that $f(x_{k+1}) \leq f(x_k)$.

One problem is how to choose the step size. Local minima are sensitive to the starting point, as well as the step size. The step size influences both the convergence rate and the behaviour of the minimization procedure. If the step size is chosen too large, the current fit jumps over the minimum, perhaps jumping back and it may take a long time to converge. On the other hand, if a step size is too small, the algorithm takes very small steps and very little progress at iterations (Ihler, 2013). One option is to choose a fixed step size that will assure convergence wherever the GD starts. Another option is to choose a different step size at each iteration (step size is changed inside the algorithm).

The GD is the simplest convergent algorithm for finding the minimum of $f(x)$. It quickly approaches the solution from a distance, which is its main advantage. The main disvantage of the algorithm is that it aproaches the best fit very slowly, and tends to zigzag along the bottom of long narrow canyons (Zinn-Bjorkman, 2010). This problem is solved by the GN method.

Gauss-Newton method

In the GN method, the SSE is reduced by assuming the LS function is locally quadratic and finding the minimum of the quadratic. It presumes that the objective function is approximately quadratic in the parameters near the optimal solution. For moderately-sized problems, the GN method typically converges much faster than the GD method (Marquardt, 1963). The GD is based on the second order Taylor approximation of the SSE *E* around the optimal solution E_0 (See 2).

$$E \approx E_0 + \left(\frac{\partial E}{\partial \mathbf{u}}\right)^T \delta\mathbf{u} + \frac{1}{2}\delta\mathbf{u}^T \mathbf{H}\,\delta\mathbf{u} \tag{2}$$

where **u** is a parameter vector, $\delta\mathbf{u}$ is the parameter deviation, and **H** is the Hessian symetric matrix of the second derivates of *E*. Intuitively Hessian describes the local *n*-dimensional parabola curvature that approximates *E*. If **H** is positive-definite, the parabola is indeed positively curved, meaning that it goes up in all directions, has a definite minimum, and the parameters are estimated in the following way:

$$\delta\mathbf{u} = \mathbf{u}^* - \mathbf{u} = -\mathbf{H}^{-1}\frac{\partial E}{\partial \mathbf{u}} = 0$$

$$\mathbf{u}^* = \mathbf{u} - \mathbf{H}^{-1}\frac{\partial E}{\partial \mathbf{u}}$$

where $\mathbf{u}^*$ is the estimated parameter vector.

The problem of finding $\mathbf{H}^{-1}$ was solved by Hassibi and Stork (1993) who applied the outer product approximation to develop a computationally efficient procedure for the calculaton of the inverse Hessian. For *N* number of parameters in the data set, and the vector **g** that is the gradient of the error function, the sequential procedure for building up the Hessian is obtained by separating off the contribution from the data point *N*+1 to give:

$$\mathbf{H}_{N+1}^{-1} = \mathbf{H}_N^{-1} - \frac{\mathbf{H}_N^{-1}\mathbf{g}^{N+1}\left(\mathbf{g}^{N+1}\right)^T\mathbf{H}_N^{-1}}{1+\left(\mathbf{g}^{N+1}\right)^T\mathbf{H}_N^{-1}\mathbf{g}^{N+1}}$$

The initial matrix $\mathbf{H}_0$ is chosen to be $\alpha\mathbf{I}$, where α is small quantity.

Given a formula for updating gradient of ∇f with a small positive amount, then $x_{k+1} = x_k - \left(\nabla f(x_k)\right)^{-1}\nabla f(x_k)$.

Levenberg-Marquardt method

As many other algorithms for the minimization of an objective function, the LM algorithm also provides a numerical solution to the problem of minimizing a function, over a space of parameters (Kashyap, 1980), (Ljung, 1987), (Larsen, 1993), (Hansen, Rasmussen, 1994), (Fahlman, 1988). The function update with a small step is $x_{k+1} = x_k - (H + \lambda I)^{-1} \nabla f(x_k)$, which is a blend of the above mentioned algorithm, i.e. a search direction is the solution of the matrix equation

$$(\mathbf{H} + \lambda \mathbf{I})\boldsymbol{\delta} = \mathbf{J}^T \mathbf{E} \tag{3}$$

where λ is the damping factor adjusted at each iteration and guides the optimization process, $\boldsymbol{\delta}$ is the parameter update matrix, **J** is the Jacobian matrix of the first derivatives of **E**, and **I** is the identity matrix. If the reduction of **E** is rapid, a smaller value of λ brings the algorithm closer to the GN algorithm, whereas if the iteration gives insufficient reduction in the residual, λ can be increased, giving a step closer to the GD direction, i.e. the LM method acts more like the GD method when the parameters are far from their optimal value, and acts more like the GN method when the parameters are close to their optimal value.

The update rule goes as follows: if the error goes down following the update, it implies that the quadratic assumption is working and λ has to be reduced (usually by a factor of 10), to reduce the influence of the GD. On the other hand, if the error goes up, λ has to be increased by the same factor. The LM algorithm is thus: do an update in the direction given by the rule above, evaluate the error at the new parameter vector, if the error has increased then retract the step, and go to the first step, otherwise accept this step and decrease λ (Ranganathan, 2004).

Levenberg-Marquardt optimization for the feedforward neural network learning

According to Azimi-Sadjadi and Liou (1992), the FNN with one hidden layer and the sigmoidal-type nonlinearity can approximate any nonlinear function and generate any complex decision region needed for clasification and recognition tasks. Consider the FNN

$$y_i(\mathbf{w}, \mathbf{W}) = F_i\left(\sum_{j=1}^{q} W_{ij} f_j\left(\sum_{l=1}^{m} w_{ij} z_l + w_{j0}\right) + W_{f0}\right)$$

where y_i is the output, **w** and **W** are the synaptic weight matrices and f_j and F_i are the activation functions of the hidden and the output layer, re-

spectively. q and m represent the number of nodes in the input and the hidden layer, respectively. For the FNN with differentiable activation functions of both input variables and weights, each unit computes a weighted sum of inputs

$$a_j = \sum_i w_{ji} z_i$$

where z_i is the activation which sends the connection to the unit j, and w_{ji} is the weight associated with the connection. For a given z_j a nonlinear function $g(\bullet)$ is applied, so $z_i = g(a_i)$. $g(\bullet)$ is known in advance (Protić, 2014).

Also consider the error function

$$E = \sum_b E^n$$

where $E^n = E^n(y_{1,\ldots,} y_c)$. The goal is to evaluate the derivatives of the error E^n with respect to the weights

$$\frac{\partial E^n}{\partial w_{ij}} = \frac{\partial E^n}{\partial a_j} \frac{\partial a_j}{\partial w_{ij}}$$

$$\delta_j = \frac{\partial E^n}{\partial a_j}, \quad \frac{\partial a_j}{\partial w_{ji}} = z_i$$

It follows that

$$\frac{\partial E^n}{\partial w_{ij}} = \delta_j z_i$$

If $g'(a)$ substitutes $\partial E^n / \partial y$, and the errors of the output units (δ_k) and the hidden units (δ_j) are

$$\delta_k = \frac{\partial E^n}{\partial a_k} = g'(a_k) \frac{\partial E^n}{\partial y_k}$$

$$\delta_j = \frac{\partial E^n}{\partial a_j} = \sum_k \frac{\partial E^n}{\partial a_k} \frac{\partial a_k}{\partial a_j}$$

The BP formula gives:

$$\delta_j = g'(a_j) \sum_k w_{kj} \delta_k$$

where δ's can be evaluated backward. In the similar way, this may be used to calculate the other derivatives.

Protić, D., Feedforward neural networks: the Levenberg-Marquardt optimization and optimal brain surgeon pruning, pp. 11–28

Consider the evaluation of the Jacobian matrix, whose elements are given by the derivatives of the network outputs y_k with respect to the network inputs x_i

$$J_{ki} = \frac{\partial y_k}{\partial x_i} = \sum_k \frac{\partial y_k}{\partial a_j}\frac{\partial a_j}{\partial x_i} = \sum_j w_{ji}\frac{\partial y_k}{\partial a_j} = \sum_j w_{ji}\sum_l \frac{\partial y_k}{\partial a_l}\frac{\partial a_l}{\partial a_j}\cdots$$

$$\cdots = \sum_j w_{ji} g'(a_j)\sum_l w_{lj}\frac{\partial y_k}{\partial a_l}$$

The second order derivatives of the error function, i.e. the parameters of the Hessian matrix are given with

$$\frac{\partial^2 E}{\partial w_{ij}\partial w_{lk}} = \sum_n \frac{\partial y^n}{\partial w_{ji}}\frac{\partial y^n}{\partial w_{lk}} + \sum_n \left(y^n - t^n\right)\frac{\partial^2 y^n}{\partial w_{ji}\partial w_{lk}} \tag{4}$$

If the outputs y^n are very close to the target values t^n, the second term can be neglected, which gives a LM formula

$$\frac{\partial^2 E}{\partial w_{ij}\partial w_{lk}} = \sum_n \frac{\partial y^n}{\partial w_{ji}}\frac{\partial y^n}{\partial w_{lk}} \tag{5}$$

The optimization process goes like this: propagate the input signal through the FNN in the forward direction to obtain outputs at each layer. Generate the output signal for each node. Compute the matrices for updating weights. Determine the state of a particular node. Proceed if the input to the node is within the ramp region, otherwise there is no need for weight updating; then examine the next node. Update the weight vector using the recursion.

Pruning is a technique, a tool that helps decide upon the structure of the FNN. It addresses only the neurons in the hidden layer. There are two types of pruning: 1) incremental, starting with input or output layers and then incrementally decrease a size of FNN and retrain after each iteration; 2) selective pruning, which starts with an already trained FNN, with a fixed size, and then removes hidden neurons that will not affect the error rate of the FNN. In this way, unproduced neurons are removed (Hansen, Rasmussen, 1994).

After the network updating is finished, the pruning is carried out in the following way: for a network trained to a local minimum error the linear term as well as higher order terms in Taylor's equations vanish. The goal is to set one of weights (a parameter) to zero (Hassibi and Stork, 1993).

Consider the following equations

$$\delta u_m + u_m = 0$$

$$\mathbf{e}_m^T \delta \mathbf{u} + u_m = 0$$

where u_m is the mth parameter, $\mathbf{e}_m$ is the unit vector of the same dimension as $\delta\mathbf{u}$. The objective of this methodology is to prune the parameter u_m that would cause a minimum increase of an error, as follows

$$\delta \mathbf{u} = -\lambda \mathbf{H}^{-1} \mathbf{e}_m$$

$$\lambda = \frac{u_m}{\mathbf{e}_m^T \mathbf{H}^{-1} \mathbf{e}_m}$$

$$\delta \mathbf{u} = -\frac{u_m}{\mathbf{e}_m^T \mathbf{H}^{-1} \mathbf{e}_m} \mathbf{H}^{-1} \mathbf{e}_m$$

It should be noted that neither **H** nor $\mathbf{H}^{-1}$ have to be diagonal (Hassibi, Stork and Wolff, 1993).

Results

The experiments were carried out on the wovel *„a*“ spoken by a female and a male speaker. Both glottal and vocal signals were used for training and testing. The training was carried out on 1700 samples of speech and the corresponding glottal signals. For the FNN training, the LM method was applied as an optimization algorithm. The network structure was 18-3-1, i.e. 18 inputs (14 inputs for the speech and four inputs for the glottal signal), three neurons in the hidden layer, and one output neuron, with a tangent hyperbolic type for all neurons. According to Protic (2014) the two-pole model for approximately (2**n*+1)*500Hz, *n* = 0, 1 ... poles is sutable for speech prediction. Considering the frequency interval of speech sound [0 - 4] kHz, for the purpose of research the seven two-pole, i.e. 14 inputs, model is used. Four inputs of the glottal signal correspond to two zeros. Since the resulting values of neurons' outputs were inside the interval (-1, 1), the signal samples were normalized to the values at the interval [-1, 1]. Weights were initialized as random values at the same interval. This structure simulated the speech production system (glottal and vocal tract). The NSSE was used for optimization, where NSSE=SSE/*n*; *n* is the size of the error set. The

training was carried out so that the $NSSE_{train}$ was less than 0.001. The resulting structure was tested on an independent test set, and the value of the $NSSE_{test}$ was determined. The OBS pruning algorithm was applied to a determined minimum structure of the FNN so that the rejection of neurons did not affect the overall NSSE. The stopping criterion for pruning was that the abrupt change in $NSSE_{prune}$ was more than 10 times higher than the minimum value of NSSEs calculated in the preceding processes.

The optimal structures were 14-3-1 for the female speaker (Figure 1) and 16-3-1 for the male speaker (Figure 2).

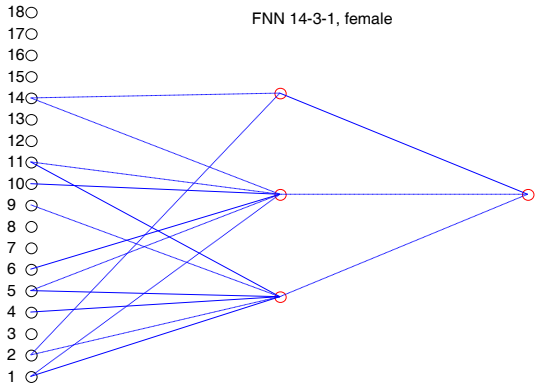

Figure 1 – FNN structure after pruning, 14-3-1, female
Slika 1 – FNN struktura nakon pruninga, 14-3-1, žena
Рис. 1 – структра FNN после обрезки, 14-3-1, женщины

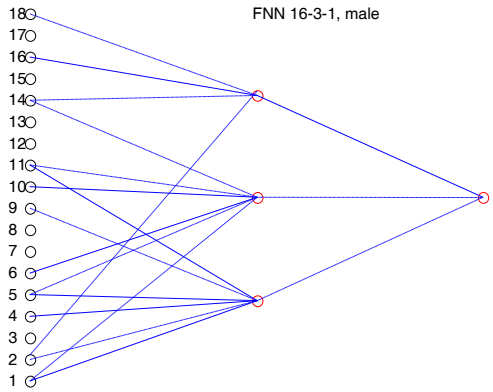

Figure 2 – FNN structure after pruning, 16-3-1, male
Slika 2 – FNN struktura nakon pruninga, 16-3-1, muškarac
Рис. 2 – структра FNN после обрезки, 16-3-1, мужчины

For these structures, the abrupt changes of NSSEs have shown that $NSSE_{prune}$ was more than ten times higher than $NSSE_{train}$ and $NSSE_{test}$. (See Table 1).

Table 1 – NSSE for training, test and pruning
Tabela 1 – NSSE za trening, test i pruning
Таблица 1 – NSSE для тренировки, тестирования и обрезки

	Training	*Testing*	*Pruning*
Error	$NSSE_{train}$	$NSSE_{test}$	$NSSE_{prune}$
$NSSE_{male}$	0.0005	0.0004	0.0040 (16-3-1)
$NSSE_{female}$	0.0009	0.0006	0.0114 (14-3-1)

The fully connected FNNs were reduced in size but the reduction was not significant. As it is presented in Figure 1 and Table 1, the nonlinear structure that takes the form of 14-3-1 is a good predictor of the speech signal for the female speaker. It should be noted that, in this case, the glottal signal has no effect on the prediction, and the FNN behaves as a nonlinear AutoRegressive (AR) model, where there is a grouping of the first six parameters (excluding the fifth node). Grouping is also noticeable for nodes 9-11. It stands out the impact of a sample at the 14^{th} node. For the male speaker there is an impact of the glottal signal which can be seen as an influence of the excitation signal, i.e. a lesser vibration frequency of the vocal cords for the male speaker compared to the female. In this case, the network has the structure of 16-3-1, and the nodes that describe the impact of the speech signal are identically arranged as in the previous case (for the woman), so the FNN behaves as a nonlinear AR model with the eXogeniuos input (ARX).

In both cases, the structure of the FNN is reduced for a small number of parameters. It should be emphasized that if the recognition of speech whose basic frequency is relatively high, the glottal signal does not have to be recorded, which is particularly suitable because the recording technology (electroglottography) is uncomfortable for the speaker (the electrodes are placed externally, on the larynx).

Conclusion

The FNN with one hidden layer that has the structure of 18-3-1 and the hyperbolic tangent transfer functions of all nodes is a good predictor of the speech signal, for the prediction of vowels. The prediction algorithm involves optimization techniques and pruning of neurons that do not have a significant impact on the change of prediction errors. Several optimization algorithms which are described in this paper can be applied to train a neural network. The most popular algorithms are the GD and the GN as well as a

combination of these which is known as the LM algorithm. For the GD algorithm, the optimization is based on the first derivative SSE which is calculated forward from the input layer to the output layer, after which the parameters are set backward, from the output to the input using the BP algorithm. The GN uses the quadrature approximation of the prediction error that is approximated by the Taylor series, so the optimization method includes solving a quadratic equation, as well. The second derivatives of the error are evaluated. In the first case, convergence is not fast enough, while in the second case the processing time and consumption are high, because of the necessity of calculation of inverse Hessian matrices. The LM algorithm is a combination of the two previous algorithms, which is determined with the damping factor (damping strategy) for the adjustment with the trade-off between the GD and the GN methods.

In this paper, the LM optimization algorithm was used to train the network. The network was tested after training, after which OBS pruning was derived. Those neurons that do not affect the change of the error were rejected by pruning, while the stopping criterion was the abrupt change of the NSSE.

The results showed that the LM algorithm provided high quality solutions in a prediction of vowels. The experiments were based on the prediction of the vowel *„a*" that was pronounced by female and male speakers. The amplitudes of signals were normalized to the interval [-1, 1], and adjusted to the tanh transfer functions of neurons. That was also suitable for the comparison of male and female speakers. The optimization algorithm was based on the NSSE, in such a way that the training stopped when the NSSE was less than 0.001. The results showed that this was the proper value for the optimization criterion, considering that the optimized structure gave the minimum test error, and that pruning reduced FNN weights, which were mostly related to the glottal signal. The experiments have proved that OBS pruning can reduce the number of input parameters so that the glottal wave has no influence on the prediction for the female speaker. However, this did not stand for the male speaker, considering the lower basic excitation frequency, i.e. lower speed of opening and closing of the vocal cords. According to the results, it is obvious that the vowel "*a*" can be predicted right and without mistakes for the given FNN, the training based on the LM algorithm and the OBS pruning.

References

Azimi-Sadjadi, M.R., & Liou, R.-. 1992. Fast learning process of multilayer neural networks using recursive least squares method. *IEEE Transactions on Signal Processing,* 40(2), pp.446-450. doi:10.1109/78.124956

Bishop, C. 1995. *Neural Networks for Pattern Recognition*.Clarendon Press. Oxford University Press.

Fahlman, S.E. 1988. Fast-learning variation on back propagation: An empirical study. . In: Proceedings of the 188 Connectionist Model Schools, San Mateo, Pitsburg, USA., pp.38-51

Gavin, H. 2013. *The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems.* Department of Civil and Environmental Engineering Duke University.

Hansen, L.K., & Rasmussen, C.E. 1994. Pruning from Adaptive Regularization. *Neural Computation,* 6(6), pp.1223-1232. doi:10.1162/neco.1994.6.6.1223

Hassibi, B., & Stork, D.G. 1993. Second order derivatives for network pruning: optimal brain surgeon. In S.J. Hanson, J.D. Cowan, & C.L. Giles Eds., *Advances in Neural Information Processing Systems*, pp.164-171.

Hassibi, B., Stork, D.G., & Wolff, G.J. 1993. Optimal Brain Surgeon and general network pruning. *IEEE International Conference on Neural Networks.* Retrieved from http://systems.caltech.edu/EE/Faculty/babak/pubs/conferences/00298572.pdf doi:10.1109/icnn.1993.298572

Ihler, A. 2013. *Linear regression: Gradient descent & stochastic gradient descent*. Irvine: BREN:ICS University of California. Retrieved from https://www.youtube.com/watch?v=WnqQrPNYz5Q

Kashyap, R. 1980. Inconsistency of the AIC rule for estimating the order of autoregressive models. *IEEE Transactions on Automatic Control,* 25(5), pp.996-998. doi:10.1109/TAC.1980.1102471

Larsen, J. 1993. *Design of Neural Networks*. Lyngby: Electronic Institute, DTH.

Ljung, L. 1987. *System Identification: Theory for the User*. Prentice Hall Inc.

Levenberg, K. 1944. A Method for the Solution of Certain Problems in Least Squares. *Quart Appl Math,* 2, pp.164-168.

Marquardt, D.W. 1963. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SLAMJ Appl Math,* 11(2), pp.431-441. doi:10.1137/0111030

Ranganathan, A.(2004). *The Levenberg-Marquardt Algorithm.* Retrieved from http://www.ananth.in/docs/lmtut.pdf.

Protic, D. 2014. A comparative analysis of Serbian phonemes: Linear and non-linear models. *Vojnotehnicki glasnik,* 62(4), pp.7-37. doi:10.5937/vojtehg62-5170

Riecke, L., Esposito, F., Bonte, M., & Formisano, E. 2009. Hearing Illusory Sounds in Noise: The Timing of Sensory-Perceptual Transformations in Auditory Cortex. *Neuron,* 64(4), pp.550-561. doi:10.1016/j.neuron.2009.10.016

Shahin, A.J., & Pitt, M.A. 2012. Alpha activity marking word boundaries mediates speech segmentation. *European Journal of Neuroscience,* 36(12), pp.3740-3748. doi:10.1111/ejn.12008

Silva, L.M., de Sá, J.M., & Alexandre, L.A. 2008. Data classification with multilayer perceptrons using a generalized error function. *Neural Networks,* 21(9), pp.1302-1310. doi:10.1016/j.neunet.2008.04.004

Wu, W., Wang, J., Cheng, M., & Li, Z. 2011. Convergence analysis of online gradient method for BP neural networks. *Neural Networks,* 24(1), pp.91-98. doi:10.1016/j.neunet.2010.09.007

Young-tae, K., Ji-won, H., & Cheol-jung, Y. 2011. A new damping strategy of Levenberg-Marquardt algorithm for Multilayer Perceptrons. *Neural Network World,* 21(4), pp.327-340. doi:10.14311/nnw.2011.21.020

Zinn-Bjorkman, L. 2010. *Numerical optimization using Levenberg-Marquardt Algorithm. EES-16. LA-UR-11-12010. PowerPoint presentation*. PowerPoint presentation.

НЕЙРОННЫЕ СЕТИ С ПРЯМОЙ СВЯЗЬЮ: АЛГОРИТМ ЛЕВЕНБЕРГА – МАРКВАРДТА И ОПТИМАЛЬНЫЙ НЕЙРОХИРУРГИЧЕСКИЙ ПРУНИНГ

ОБЛАСТЬ: телекоммуникации
ТИП СТАТЬИ: оригинальная научная статья
ЯЗЫК СТАТЬИ: английский

Резюме

Данная статья описывает процесс обучения, тестирования и прунинга нейронной сети с прямой связью с одним скрытым слоем, который был использован для прогнозирования гласного «а». Описаны алгоритмы Gradient Descent, Гаусса-Ньютона и Левенберга-Марквардта. Оптимизация методом нейрохирургического прунинга применена к подготовленной сети. Критерием остановки является внезапное изменение нормированной суммы квадратов ошибок. Структура нейронных сетей с прямой связью (FNN) имела 18 входов (четыре для гортанных и 14 для образцов спектра речи). Результаты показали, что после обрезки гортанный сигнал не имеет никакого влияния на модели женской речи, в то время как на модели мужской речи такое влияние зафиксировано. В обоих случаях, структура FNN сводится к небольшому числу параметров.

Ключевые слова: *Левенберг-Марквардт, анализ речи, нейронные сети с прямой связью.*

FEEDFORWARD NEURONSKE MREŽE: LEVENBERG-MARQUARDT OPTIMIZACIJA I OPTIMAL BRAIN SURGEON PRUNING

OBLAST: telekomunikacije
VRSTA ČLANKA: originalni naučni članak
JEZIK ČLANKA: engleski

Sažetak:

U radu su opisani obučavanje, testiranje i pruning feedforward neuronske mreže sa jednim skrivenim slojem koji je korišćen za predikciju vokala a. Opisane su Gradient Descent, Gauss-Newton i Levenberg-Marquardt optimizacione tehnike. Optimal Brain Surgeon pruning je primenjen na treniranu mrežu. Kriterijum zaustavljanja je nagla promena normalizovane sume kvadrata grešaka. Struktura feedforward neuronske mreže (FNN) bila je 18 ulaza (četiri za glotalne i 14 za odbirke govora). Rezultati su pokazali da, nakon pruninga, glotalni signal nema uticaja na model za ženskog govornika, dok utiče na predikciju govora kod muškog govornika. U oba slučaja, struktura FNN je redukovana na mali broj parametara.

Uvod

Veštačka neuronska mreža je jedna od najboljih struktura za rešavanje različitih problema koji se odnose na veštačku inteligenciju, prepoznavanje oblika, klasifikaciju, predikciju vremenskih serija i mnoge praktične probleme. Višeslojni perceptron je najstariji i najčešće korišćeni oblik veštačke neuronske mreže. Tipično, sastoji se od nekoliko slojeva neurona. Na ulazu mreže nalazi se više neurona, a na izlazu je moguće da postoji jedan ili više izlaznih neurona. Ukoliko je signal propagiran od ulaza ka izlazu onda je ovaj tip mreže tzv. feedforward neuronska mreža. Da bi bili određeni parametri ove mreže, potrebno je minimizirati funkciju greške. Kod gradient descent algoritma računaju se prvi izvodi greške i težine se podešavaju iterativnim putem. Signal greške se propagira unazad. Ova tehnika je poznata kao back-propagation algoritam. Težinski parametri u mreži podešavaju se po pravcu negativnog gradijenta funkcije po parametrima. Međutim, ovaj algoritam je relativno spor, pa rešenja mogu biti „zarobljena" u jednom od lokalnih minimuma, umesto da se izračuna globalni minimum.

Levenberg-Marquardt algoritam daje efikasna rešenja za konvergenciju i bolju optimizaciju, jer koristi i Gauss-Newton metod koji podrazumeva da je greška kvadratna u okolini optimalnog rešenja, što je bazirano na Tejlorovoj aproksimaciji drugog reda greške sume kvadrata. Na taj način feedforward neuronska mreža i Levenberg-Marquardt algoritam omogućuju da se lakše reše problemi konvergencije i trajanje računarskih procesa, što je karakteristično za višeslojne perceptrone.

U radu je prikazana predikcija vokala a za govornike oba pola. U eksperimentima je primenjena feedforward neuronska mreža sa jednim skrivenim slojem strukture 18 ulaza, 3 neurona u skrivenom sloju i jedan izlazni neuron (18-3-1). Aktivacione funkcije svih neurona bile su tangens-hiperbolične, a njihove početne vrednosti slučajni brojevi iz opsega [-1, 1]. Obučavanje je izvedeno na 1.700 odbiraka govornog signala i odgovarajućeg glotalnog signala, dok je mreža testirana na 1.700 odbiraka u nepoznatom delu signala. Prvih 14 ulaza u mrežu odgovaraju govornom signalu, a druga četiri odbircima glotalnog signala. Prediktuje se odbirak govora, a greška predikcije računa se kao razlika između tačne i prediktovane vrednosti signala i koristi se za dobijanje sume kvadrata. Reultujuća struktura je testirana na nezavisnom test-skupu, pa je izveden pruning tipa optimal brain surgeon. Ova tehnika redukuje broj neurona u skrivenom sloju tako da to ne utiče na ukupnu grešku. Kriterijum zaustavljanja je nagli skok normalizovane sume kvadrata grešaka.

Levenberg-Marquardt optimizacija

LM algorotam je moguće posmatrati kao linearnu kombinaciju GD i GN metoda. Alternacija ova dva metoda zove se damping strategija koja je kontrolisana damping faktorom. Ukoliko je ovaj faktor veliki, LM se po-

naša kao GD, u suprotnom postaje GN metod. Ove metode su optimizacioni algoritmi za rešavanje problema minimizacije parametara, zasnovane na metodu najmanjih kvadrata. Podešavanje parametara zahteva parametarski model koji minimizira sumu kvadrata reziduala. Rezidual je razlika između tačne i prediktovane vrednosti odbirka signala.

Metod GD je optimizaciona tehnika kojom se minimiziraju vrednosti parametara u pravcu suprotnom gradijentu posmatrane funkcije. To je veoma konvergentan metod za pronalaženje minimuma jednostavnih funkcija. Predstavlja algoritam prvog reda, jer za optimizaciju koristi isključivo prve izvode funkcije grešaka, po parametrima modela.

U GN metodu suma kvadrata grešaka je redukovana uz pretpostavku da je LS funkcija lokalno kvadraturna, pa je i nalaženje minimuma odgovarajuće. Pretpostavljeno je da je optimizaciona funkcija približno kvadratna u okolini optimalnog rešenja. Za probleme srednjeg nivoa GN metod brže konvergira od GD metoda.

Algoritam LM takođe obezbeđuje minimizaciju funkcije greške po vektoru parametara, koji je kombinacija prethodna dva metoda. Damping faktorom je određen izbor metode i načina obučavanja mreže. Za mali damping faktor algoritam je bliži GN optimizacionoj tehnici, dok je kod povećanja damping faktora optimizacioni algoritam sve bliži GD metodu. Podešavanje parametara bazirano je na promeni vrednosti damping faktora, a algoritam se odvija na sledeći način: ulazni signal se propagira ka izlazu, izračunaju se reziduali i primeni optimizacioni algoritam promenom parametara unazad. Ukoliko je optimizacioni kriterijum zadovoljen obučavanje se zaustavlja; u suprotnom se izvodi minimizacija korak po korak do zadovoljavanja optimizacionog kriterijuma.

Po Azimi-Sadjadi and Liou (1992) FNN sa jednim skrivenim slojem i nelinearnosti sigmoidalnog tipa može da aproksimira bilo koju nelinearnu funkciju i generiše svaki kompleksni region odlučivanja za proračune koji se odnose na klasifikaciju ili prepoznavanje. Optimizacioni proces odvija se na sledeći način: propagira signal kroz FNN u pravcu od ulaza ka izlazu, kako bi se odredio izlaz svakog sloja i generiše izlazni signal svakog čvora. Zatim se Izračunaju matrice za promenu parametara i određuje stanje partikularnog čvora. Ako je izlaz u granicama, region se prihvata, u suprotnom se parametri menjaju i posmatra se sledeći čvor. Parametri se podešavaju rekurzijom.

Pruning je tehnika kojom je moguće minimizirati strukturu FNN. Odnosi se isključivo na skriveni sloj mreže. Postoje dva tipa pruninga: 1) inkrementalni koji počinje na ulaznom ili izlaznom sloju, inkrementalno smanjuje veličinu mreže i izvodi ponovni trening nakon svake iteracije i 2) selektivni pruning, koji počinje sa treniranom mrežom, fikse veličine, i zatim uklanja skrivene neurone koji ne utiču na grešku neuronske mreže. Na taj način neproduktivni neuroni su uklonjeni. Nakon što je podešavanje završeno, pruning se izvodi na sledeći način: za mrežu koja je obučena da daje lokalni minimum greške, linearni deo, kao i viši delovi (stepenovi) u Tejlorovoj jednačini nestaju. Cilj je da se težine, tj. parametri postave na nulu.

Rezultati

Eksperimenti su izvedeni na vokalu a koji su izgovarali žena i muškarac. Govorni i glotalni signal korišćeni su za obučavanje i testiranje. To je izvedeno na 1.700 odbiraka vokala a, uključujući i odgovarajući glotalni signal. Algoritam LM je korišćen za obučavanje i optimizaciju. Struktura je bila 18-3-1, što znači 18 ulaza (14 za govorni i 4 za glotalni signal), 3 neurona u skrivenom sloju i jedan izlazni neuron, sa tangens hiperboličnom prenosnom funkcijom svakog neurona. S obzirom na to da su izlazi iz neurona limitirani na vrednosti (-1, 1), i signali su normalizovani u granice [-1, 1]. Početne vrednosti parametara izabrane su slučajno iz istog intervala. Ova struktura simulira sistem za proizvođenje govora (glotalni i vokalni trakt). Normalizovana suma najmanjih kvadrata (NSSE) korišćena je za optimizaciju. Nakon toga, Optimal Brain Surgeon primenjen je kao pruning. Za govornika ženskog pola optimalna je struktura 14-3-1, dok je za muškarca optimalna struktura neuronske mreže 16-3-1. Trening je izveden tako da je $NSSE_{train}$ manja od 0,001. Rezultujuća struktura testirana je na nezavisnom test skupu, i izračunata je vrednost $NSSE_{test}$. Primenjen je OBS pruning na dobijenu strukturu. Kriterijum zaustavljanja pruninga jeste da je nagla promena u $NSSE_{prune}$ 10 i više puta veća od minimalnih dobijenih NSSE grešaka. Nelinearna struktura kod žene pokazala se kao dobar prediktor, kod kojeg nema uticaja glotalnog signala na predikciju. Kod muškarca postoji uticaj glotalnog signala, što je najverovatnije rezultat niske učestanosti pobude.

U oba slučaja, struktura FNN nije bitno redukovana, tako da je, u slučaju da nije moguće primeniti pruning, moguće koristiti potpuno povezanu mrežu, bez bitnih razlika. Posebna pogodnost je što kod visokih osnovnih učestanosti pobude nije neophodno koristiti elektroglotografiju, tehniku snimanja glotalnog signala koja je nekomformna za govornika, jer se, umesto mikrofona, elektrode stavljaju spolja, na larings.

Zaključak

FNN sa jednim skrivenim slojem, strukturom 18-3-1 i tangens hiperboličnim prenosnim funkcijama svih čvorova predstavlja dobar prediktor govornog signala, kada je osnova predikcije vokal. Algoritam predikcije podrazumeva optimizacione tehnike i pruning neurona koji ne utiču bitno na promenu greške predikcije. Nekoliko optimizacionih algoritama koji su opisani u ovom radu može se primeniti za trening neuronske mreže. Najpopularniji od njih su GD i GN, kao i njihova kombinacija LM algoritam. Kod GD algoritma optimizacija je bazirana na prvim izvodima funkcije SSE greške po parametrima modela unapred, a parametri se podešavaju unazad od izlaza ka ulazu, BP algoritmom. GN koristi kvadraturnu aproksimaciju greške parametara koja je razvijena u Tejlorov red, pa i optimizacioni metod podrazumeva rešavanje kvadratnih jednačina. Računaju se drugi izvodi funkcije greške. U prvom slučaju konvergencija je nedovoljno brza, dok je u drugom slučaju za izvršenje algorit-

ma procesno vreme i potrošnja velika, jer je potrebno izračunati inverzne Hesijanove matrice.

Algoritam LM je kombinacija prethodna dva algoritma koja je određena damping faktorom (damping strategija) koji vrši prilagođenje tako da pravi trade-off između GD i GN metoda.

U ovom radu korišćen je LM optimizacioni algoritam za trening mreže. Mreža je nakon treninga testirana, pa je izveden OBS pruning. Pruningom je odbačen višak neurona kada je kriterijum zaustavljanja nagli skok NSSE greške.

Rezultati su pokazali da LM algoritam daje dobra rešenja kod predikcije vokala a. Eksperimentima je, pored toga, dokazano da je pruningom moguće redukovati broj ulaznih parametara, tako da glotalni talas nema uticaja na predikciju kod ženskog govornika, što nije slučaj kod muškarca, s obzirom na nižu osnovnu učestanost pobude, odnosno manju brzinu otvaranja i zatvaranja glasnih žica.

Ključne reči: *Levenberg-Marquardt; analiza govora; pruning; feedforward neuronske mreže.*

Datum prijema članka / Paper received on / Дата получения работы: 08. 01. 2015.
Datum dostavljanja ispravki rukopisa / Manuscript corrections submitted on / Дата получения исправленной версии работы: 01. 03. 2015.
Datum konačnog prihvatanja članka za objavljivanje / Paper accepted for publishing on / Дата окончательного согласования работы: 03. 03. 2015.