Applying Bayesian Regularization for Acceleration of Levenberg-Marquardt based Neural Network Training

Azizah Suliman¹, Batyrkhan S. Omarov¹,² *

¹ College of Computer Science & Information Technology, Universiti Tenaga Nasional, Kuala Lumpur (Malaysia)
² International Information Technologies University, Almaty (Kazakhstan)

Received 12 November 2017 | Accepted 29 March 2018 | Published 13 April 2018

ABSTRACT

Neural network is widely used for image classification problems, and is proven to be effective with high successful rate. However one of its main challenges is the significant amount of time it takes to train the network. The goal of this research is to improve the neural network training algorithms and apply and test them in classification and recognition problems. In this paper, we describe a method of applying Bayesian regularization to improve Levenberg-Marquardt (LM) algorithm and make it better usable in training neural networks. In the experimental part, we qualify the modified LM algorithm using Bayesian regularization and use it to determine an appropriate number of hidden layers in the network to avoid overtraining. The result of the experiment was very encouraging with a 98.8% correct classification when run on test samples.

II. RELATED WORKS

There are a lot of research to improve Levenberg-Marquardt algorithm to train neural networks, in last years, for example, (An Ru et. al., 2016) in [1] suggested improving the LM algorithm by direct calculation of quasi-Hessian and adopted gradient vector, that does not require storage of the Jacobian matrix. In own case, it causes to reduce the computation time of a neural network training. However this kind of improvement cannot decrease the testing error. Henri P. Gavin, uses LM method for nonlinear least squares curve-fitting problems [2]. In his research, he considers the Levenberg-Marquardt curve-fitting method as a combination of methods for minimization such as the Gauss-Newton method and the gradient descent method.

Thus, Levenberg-Marquardt method improvement comes from improvements of the named two methods. Gauss-Newton method assumes that the least square function is locally quadratic, and it causes to reduce the squared error. The LM method is more similar to the method of gradient descent, when its parameters are far from their optimal value. In contrast, the LM method is more similar to Gauss-Newton method, in case of its parameters are close to their optimum. So, in [2], Henri P. Gavin explains these methods and represents the applying of software to solve problems of approximating of nonlinear least squares. Also, Liyan
Qi proposed LM-method with self adjusting parameters for solving a nonlinear system of equations [3]. By their approach, at each step, the LM parameter \( \mu_k \) is automatically adjusted on the basis of the correlation between actual reduction and predicted reduction. Thus, Under the BD-regular condition, they prove that PSA-LMM is locally superlinearly convergent, for semi-smooth equations and locally quadratically convergent for strongly semi-smooth equations. In [4], Haddout and Rhazi try solving the problem of non-linear least squares, based upon LM and Gauss–Newton methods by minimizing the sum of squares of errors between the data and model prediction.

Some researchers use Levenberg-Marquardt algorithm for problems related to prediction. In [5], Murat Kayri did a comparative analysis to Levenberg-Marquardt and Bayesian regularization algorithms from the point of view of predictive abilities.

Despite its effectiveness, the LM algorithm requires large computational cost. To provide the practical suitability, the method demands to diminish the working time for neural networks of large size.

In our research we propose our approaches to improve the method and illustrate experimental results to prove the improvement of the algorithm.

### III. Architecture

To provide image recognition and classification using neural network training we create a program complex. Flowchart of the program complex is presented in Fig. 1. Firstly, we create a mathematical model of the Levenberg-Marquardt algorithm and its improvement. After, using the developed mathematical model, we construct a neural network that, following, is trained and its performance is evaluated. After getting the result we compare the improved LM algorithm with classical LM method and the improvements of the other researches.

![Flowchart of the program complex](image)

Fig. 1. Architecture of the Image Recognition and Classification Framework.

### IV. Bayesian Regularization Based Improvement of LM Method

The standard LM algorithm copes poorly with the situation where the training set contains elements that stand out from the general population, usually such a situation arises when using data obtained by experiment. It means that, the algorithm is inconvenient for practical training. In order to provide the practical suitability, we propose Applying Bayesian Regularization for Acceleration of Levenberg-Marquardt based Neural Network Training. Moreover, applying Bayesian Regularization allows accelerating the LM based Neural Network Training. The essence of the approach is transition from searching the minimum point of mean square error to searching the minimum point of the function expressed by equation (1).

\[
F(Y) = \alpha E_{\theta} + \beta E_D
\]  

(1)

Here \( E_{\theta} \) – network error, \( E_D \) – the sum of the squares of the network weights, \( \alpha \) and \( \beta \) – hyperparameters.

As a result, the algorithm seeks to minimize the network error and to prevent unlimited growth of its weights. The weight of the neural network can be considered as random variables and their distribution density can be expressed by the formula:

\[
P(\theta | D, \alpha, \beta, M) = \frac{P(D | \theta, \beta, M)P(\theta | \alpha, M)}{P(D | \alpha, \beta, M)}
\]  

(2)

here \( D \) – training set, \( M \) – neural network model (in our case, it is feed forward neural network that learnt on the basis of Levenberg-Marquardt algorithm), \( \theta \) – weight vector of the neural network, \( P(\theta | \alpha, M) \) – priori probability, reflecting our knowledge of the initial weights of the network. \( P(D | \theta, \beta, M) \) – likelihood function, which is the probability that the neural network with weights \( \theta \) correctly responds to a set \( D \). \( P(D | \alpha, \beta, M) \) – normalization factor, that provides the equality of the total probability 1. If we assume the training set is noisy with Gaussian noise and the network weights distribution is Gaussian distribution, then formula (2) will be transformed as:

\[
P(\theta | D, \alpha, \beta, M) = \frac{1}{Z_D(\beta)} e^{-\frac{(\alpha E_{\theta} + \beta E_D)}{2}} = \frac{1}{Z_F(\alpha, \beta)}
\]  

(3)

Here,

\[
Z_D(\beta) = \left(\frac{\beta}{\sigma}\right)^\frac{p}{2}, \quad Z_\theta(\alpha) = \left(\frac{\alpha}{\sigma}\right)^\frac{S}{2}
\]  

(4)

Normalization coefficient can be expressed from the formula (5):

\[
P(D | \alpha, \beta, M) = \frac{Z_D(\alpha, \beta)}{Z_\theta(\alpha)Z_D(\beta)}
\]  

(5)

Coefficient \( Z_F \) remains unknown. However it can be approximated by using the same assumptions as in the Levenberg-Marquardt method as equation (6):

\[
Z_F \approx 2\pi^\frac{S}{2} \sqrt{|H^{-1}|} e^{-F(\theta)}
\]  

(6)

here \( H^{-1} \) - inverse matrix to approximate Hessian matrix. Then

\[
\alpha = \frac{\gamma}{2E_\theta}, \quad \beta = \frac{P-\gamma}{2E_D}
\]  

(7)
Here \( \gamma = S - 2\alpha \text{trace}([H]^{-1}) \), \( \gamma \in [0, S] \)- parameter reflecting the number of weights of a neural network taking part in decreasing a function \( \text{trace}(H^{-1}) \) that is the sum of the diagonal elements of the inverse matrix to approximate Hessian matrix.

In this work to calculate hyperparameters \( \alpha \) and \( \beta \) we use the formula proposed by Jan Poland in the work [6, 7]:

\[
\begin{align*}
\gamma &= S - \alpha \text{trace}([H]^{-1}) \\
\alpha &= \frac{\gamma}{2E_w + \text{trace}([H]^{-1})} \\
\beta &= \frac{p N}{2E_d}
\end{align*}
\]

(8)

As a result, neural network is less susceptible to fluctuation in the training set and accurately approximates the function that is given by the training set.

\[ \text{V. Experiment Results} \]

To test the proposed approach, three well known problems such as face recognition, lung cancer classification and object classification problem, have been applied. To test the problem the data set was divided into two parts, a training set and a test set. The training set data were 70% of instances of each species, and in the test were about 30% of instances data. Fig. 2 demonstrates samples of the each training dataset.

Face recognition experiments were carried out upon the basis of a facial images database by Dr. Libor Spacek [16]. The database contains face images of 395 different people, with 20 images of each person. The test sample consists of 20 images of each person (only \( 20 \times 395 = 7900 \) images).

For image classification, in the presented research we use JSRT database [17] that includes 147 digital images of lung cancer image samples of two types (small-cell and non-small-cell types), with the 512x512 size.

For objective recognition as human, car, and other objects, we use Omnidirectional and panoramic digital images dataset. The dataset includes 30 omnidirectional images for human detection and 50 omnidirectional images for car detection [18].

Fig. 2. Samples of the training dataset.

In particular, the problem of choosing the number of hidden layer neurons was considered in [7-13, 19-21]. In [11] authors state that the optimal number of neurons in the hidden layer \( (N^h) \) should be calculated from the formula (9).

\[
N^h = \sqrt{N^{(i)} N^{(o)}}
\]

(9)

Here \( N^{(i)} \) is number of input layer neurons, \( N^{(o)} \) is the number of output layer neurons.

So, we should check this in practice. In this work, we carried out a practical study of the influence of the number of neurons in the hidden layer of the neural network in the learning rate and recognition quality. As a selection criteria of number of neurons, number of training epochs of neural network and recognition quality were chosen.

Fig. 3. Variable number of hidden neurons and learning rate.

Fig. 3 shows the ratio of neural network training epochs, when the number of neurons in the hidden layer varied from 6 to 24, in increments of 2 neurons. As it can be seen from the ratio, number of teaching epochs for the neural networks with different number of neurons in the hidden layer, the increase of this number reduces the speed of learning. As a result, not only the number of training epochs increase due to the growth of the Jacobian matrix of the neural network, also the total training time increases. However, this does not mean that the neural network with 6 neurons in the hidden layer will give the best results.

The graphs in Fig. 4 confirm that the minimization of the number of neurons in the hidden layer of the neural network does not improve the recognition quality. 9 and 15 neurons in the hidden layer give the best result. However, the number of training epochs in neural network with 18 hidden layers is substantially greater than others. Therefore, we assume that the best results in the “learning rate recognition quality” ratio are given by the neural network with 9 neurons in the hidden layer.

Fig. 4. Testing of neural networks with different number of neurons in the hidden layer.
For objective recognition as human, car, and other objects, we use omnidirectional and panoramic digital images dataset. The dataset includes 30 omnidirectional images for human detection and 50 omnidirectional images for car detection [18].

The predicted method better suits the classical problem, and also supports a trade-off between sensitivity and specificity. The proposed method is also effective in cooperation. Therefore, the proposed method can be a useful tool for classification in the field of data mining.

**VI. CONCLUSION**

A mathematical model of pattern recognition using neural networks was proposed. Modification of Levenberg-Marquardt algorithm using Bayesian regularization was described and tested. Using the modified method feed forward neural network was constructed and tested for classification problems. The proposed method fulfills better in digital image classification, and maintains a good quality trade-off between the classification rate and sensitivity. The proposed method is effective from the point of view of computationally cost. Accordingly, the proposed method can be a valuable tool for medical data analysis and classification.

**References**


Azizah Suliman

Azizah Suliman is an Associate Professor at College of Information Technology, Universiti Tenaga Nasional, Malaysia. She received her BCS from Southern Illinois University, USA in 1985, MCS from Universiti Malaya in 1990 and PhD in Computer Science (AI) from Universiti Putra Malaysia in 2011. Her interests include Soft Computing, Parallel and GPU Programming, Image Processing and Embedded Systems.

Batyrkhan S. Omarov

Batyrkhan S. Omarov is an Assistant Professor at International Information Technologies University, Almaty, Kazakhstan. He received his BCS and MSC from Al-Farabi Kazakh National University, Almaty, Kazakhstan in 2008 and 2010 respectively. From 2014 till 2017 did PhD research at Universiti Tenaga Nasional, Kuala Lumpur, Malaysia. Research Interests include Computer Vision, Machine Learning and Data Science.