Overcoming the Local-Minimum Problem in Training Multilayer Perceptrons with the NRAE Training Method*

James Ting-Ho Lo[1], Yichuan Gui[2], and Yun Peng[2]

[1]Department of Mathematics and Statistics
[2]Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, Maryland 21250, USA
{jameslo,yichgui1,ypeng}@umbc.edu

Abstract. A method of training multilayer perceptrons (MLPs) to reach a global or nearly global minimum of the standard mean squared error (MSE) criterion is proposed. It has been found that the region in the weight space that does not have a local minimum of the normalized risk-averting error (NRAE) criterion expands strictly to the entire weight space as the risk-sensitivity index increases to infinity. If the MLP under training has enough hidden neurons, the MSE and NRAE criteria are both equal to nearly zero at a global or nearly global minimum. Training the MLP with the NRAE at a sufficiently large risk-sensitivity index can therefore effectively avoid non-global local minima. Numerical experiments show consistently successful convergence from different initial guesses of the weights of the MLP at a risk-sensitivity index over 10^6. The experiments are conducted on examples with non-global local minima of the MSE criterion that are difficult to escape from by training directly with the MSE criterion.

Keywords: Neural network, Training, Normalized risk-averting error, Global optimization, Local-minimum, Mean squared error, Hessian matrix

1 Introduction

The local-minimum problem has plagued the development and application of the neural network approach based on the multilayer perceptron (MLP) and has attracted much attention since its inception [1–9]. A promising method to alleviate the problem was proposed in [10, 11]. The method employs a new type of risk-averting error (RAE) criterion, which is a transformation of the standard mean squared error (MSE) criterion for training the MLP. By gradually increasing the risk-sensitivity index, the convexity region of the RAE criterion expands

* This material is based upon work supported in part by the National Science Foundation under Grant ECCS1028048, but does not necessarily reflect the position or policy of the Government.

strictly, thereby creating tunnels or wormholes for a local search method such as the conjugate gradient and quasi-Newton algorithm to escape non-global minima. However, the method has two shortcomings. First, the RAE is a sum of exponential functions of the risk sensitivity index. Computer overflow occurs in evaluating the RAE at a large risk sensitivity index. Second, it is not always easy to select an appropriate value of the risk-sensitivity index to start the gradual convexification. In the following, a remedy is discussed.

A standard formulation of training a multilayer perceptron (MLP) under supervision follows: A set of pairs, (x_k, y_k), $k = 1, ..., K$, of which the vectors x_k and the vectors y_k are related by an unknown function f

$$y_k = f(x_k) + \xi_k$$

where ξ_k are random noises. Find the weight vector w of a MLP $\hat{f}(x, w)$ such that the mean squared error (MSE) criterion,

$$Q(w) = \frac{1}{K} \sum_{k=1}^{K} \left\| y_k - \hat{f}(x_k, w) \right\|^2 \tag{1}$$

is minimized. If the MLP $\hat{f}(x_k, w)$ is nonlinear in w, the MSE criterion $Q(w)$ is usually nonconvex and has non-global local minima.

It is proven in [11] that the convexity region of $J_\lambda(w)/\lambda$, where $J_\lambda(w)$ is a new type of risk-averting error criterion,

$$J_\lambda(w) := \sum_{k=1}^{K} \exp\left(\lambda \left\| y_k - \hat{f}(x_k, w) \right\|^2 \right) \tag{2}$$

expands strictly as λ increases, and that $\lim_{\lambda \to 0} \frac{1}{\lambda} \ln \left[\frac{1}{K} J_\lambda(w) \right] = Q(w)$. Here $:=$ means "denote" or "be defined to be". These properties confirmed the effectiveness of the adaptive training method reported in [10] for avoiding poor local minima. However, note that the RAE is a sum of exponential functions of the risk sensitivity index λ.

The normalized RAE (NRAE)

$$C_\lambda(w) := \frac{1}{\lambda} \ln \left[\frac{1}{K} J_\lambda(w) \right]$$

is a strictly increasing function of $J_\lambda(w)/\lambda^2$, whose Hessian matrix is $H_\lambda(w)/\lambda^2$. A formula (i.e., equation (8) in [11]) for the Hessian matrix $H_\lambda(w)$ shows that $\lim_{\lambda \to \infty} H_\lambda(w)/\lambda^2$ is a positive semi-definite matrix. It follows that the convexity region of $J_\lambda(w)/\lambda^2$ expands to nearly the entire weight space as λ increases to nearly infinity. Moreover, $J_\lambda(w)/\lambda^2$ does not have a non-global local minimum in the convexity region for λ sufficiently large, although $\lim_{\lambda \to \infty} J_\lambda(w)/\lambda^2$ does not exist.

Since $C_\lambda(w)$ is a strictly increasing function of $J_\lambda(w)/\lambda^2$, it does not have a non-global local minimum in the convexity region of $J_\lambda(w)/\lambda^2$ for λ sufficiently

large. As the convexity region of $J_\lambda(w)/\lambda^2$ expands to nearly the entire weight space as λ increases to nearly infinity, $C_\lambda(w)$ does not have a non-global minimum in nearly the entire weight space for λ sufficiently large. This is the first and primary reason for using $C_\lambda(w)$ as the training criterion.

As will be seen later in this paper, for $\lambda >> 1$, $C_\lambda(w)$ and its gradient vector $g_\lambda(w) := \partial C_\lambda(w)/\partial w_j$ and Hessian matrix $H_\lambda(w) := \left[\partial^2 C_\lambda(w)/\partial w_i \partial w_j\right]$ can be computed without evaluating the exponential function $\exp\left(\lambda\left\|y_k - \hat{f}(x_k, w)\right\|^2\right)$, for $\lambda >> 1$, $k = 1, \ldots, K$. This is the second reason for using $C_\lambda(w)$ as the training criterion.

If the MLP has enough hidden neurons to approximate the target function nearly perfectly, global minima of $C_\lambda(w)$ and the MSE criterion $Q(w)$ are both nearly 0. This is the third reason for using $C_\lambda(w)$ as the training criterion.

The method of training the MLP with $C_\lambda(w)$ is called the NRAE training method. The method is numerically tested for a number of large values of λ. Both $C_\lambda(w)$ and $Q(w)$ for the resultant MLP consistently converge to 0 for λ in the range 10^6 and 10^{11}. When λ exceeds 10^{11}, round-off errors occur and the NRAE training method could not be carried out. We expect to fix this numerical problem in the near future.

2 Evaluating NRAE and Its Derivatives

For notational simplicity, let

$$\begin{aligned}\hat{y}_k(w) &:= f(x_k, w)\\ \varepsilon_k(w) &:= y_k - \hat{y}_k(w)\ .\end{aligned}$$

For a vector w, let $S(w) = \arg\max_{k\in\{1,\ldots,K\}} \|\varepsilon_k(w)\|^2$, which set may contain more than one elements if a tie exists, and $M(w) = \min_k \{k | k \in S(w)\}$. It follows that

$$\|\varepsilon_k(w)\|^2 \le \left\|\varepsilon_{M(w)}(w)\right\|^2\ .$$

Let

$$\eta_k(w) := e^{\lambda\left(\|\varepsilon_k(w)\|^2 - \left\|\varepsilon_{M(w)}(w)\right\|^2\right)}$$

then

$$\begin{aligned}\eta_k(w) &\le 1\\ \ln\left[\sum_{k=1}^{K}\eta_k(w)\right] &\le \ln K\,.\end{aligned}$$

Hence

$$\begin{aligned} C_\lambda(w) &= \frac{1}{\lambda}\ln\left[\frac{1}{K}e^{\lambda\left\|\varepsilon_{M(w)}(w)\right\|^2}\sum_{k=1}^{K}\eta_k(w)\right] \\ &= \frac{1}{\lambda}\ln\frac{1}{K}+\left\|\varepsilon_{M(w)}(w)\right\|^2+\frac{1}{\lambda}\ln\left[\sum_{k=1}^{K}\eta_k(w)\right] \\ &\le \left\|\varepsilon_{M(w)}(w)\right\|^2 \end{aligned} \tag{3}$$

and the terms in (3) are bounded by functions independent of λ and no register overflow occurs for $\lambda >> 1$.

Consider the first-order derivative,

$$\begin{aligned} \frac{\partial C_\lambda(w)}{\partial w_j} &= \frac{1}{\lambda J_\lambda(w)}\frac{\partial J_\lambda(w)}{\partial w_j} \\ &= \frac{1}{\lambda J_\lambda(w)}\left[-2\lambda\sum_{k=1}^{K}e^{\lambda\|\varepsilon_k(w)\|^2}\varepsilon_k^T(w)\frac{\partial\hat{y}_k(w)}{\partial w_j}\right] \\ &= \frac{-2\sum_{k=1}^{K}\eta_k(w)\,\varepsilon_k^T(w)\frac{\partial\hat{y}_k(w)}{\partial w_j}}{\sum_{k=1}^{K}\eta_k(w)} \end{aligned} \tag{4}$$

where

$$\sum_{k=1}^{K}\eta_k(w)\le K$$

$$\sum_{k=1}^{K}\eta_k(w)\,\varepsilon_k^T(w)\frac{\partial\hat{y}_k(w)}{\partial w_j}\le\sum_{k=1}^{K}\varepsilon_k^T(w)\frac{\partial\hat{y}_k(w)}{\partial w_j}$$

which is independent of λ. Hence, both the numerator and denominator of (4) can be handled without register overflow for $\lambda >> 1$.

The Hessian matrix can be evaluated in a similar way.

3 Numerical Experiments

In this section, a function approximation task is implemented to demonstrate the effectiveness of the proposed NRAE training method. Before each training session starts, some parameters for MLPs are selected as follows. First, each synaptic weight in a weight vector is randomly selected from a uniform distribution between $-2.4/F_i$ and $2.4/F_i$, where F_i is the number of input neurons of the connected unit. Second, all input and output values defined in the training data are normalized into $[-1, 1]$. Third, the activation function in each training neuron is chosen as the hyperbolic tangent function $\varphi(v) = a\tanh(bv)$, where $a = 1.7159$ and $b = 2/3$.

3.1 Function Approximation

A function with three notches is defined by

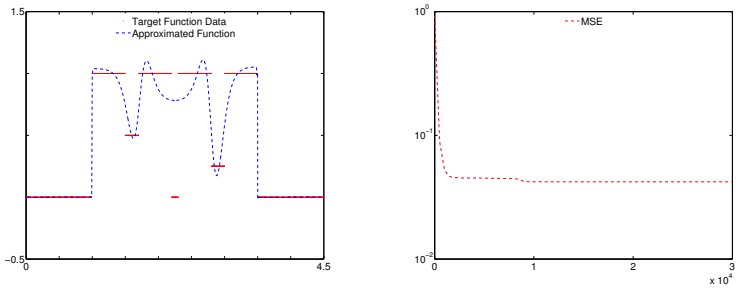
$$y = f(x) = \begin{cases} 0 & \text{if } x \in [0, 1.0] \cup [2.2, 2.3] \cup [3.5, 4.5] \\ 0.25 & \text{if } x \in [2.8, 3.0] \\ 0.5 & \text{if } x \in [1.5, 1.7] \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

where $x \in X = [0, 4.5]$. The input values x_k are obtained by random sampling 2000 non-repeatable numbers from X with a uniform distribution, and the corresponding output values y_k are computed by (5). The training data with 2000 (x_k, y_k) pairs is chosen to perform the three-notch function approximation. In our experiment, we randomly select five different initial weight vectors to start five training groups. In each training group, one standard MSE training session and six NRAE training sessions with the BP and BFGS algorithm are performed with the same initial weight vector. The values of λ are set respectively as $10^6, 10^7, 10^8, 10^9, 10^{10}$ and 10^{11} in all NRAE training sessions. MLPs with 1:16:1 architecture are initiated to both the MSE and NRAE training sessions. All training results are obtained when the deviation of objective function values between two consequent training epochs is less or equal to 10^{-15}.

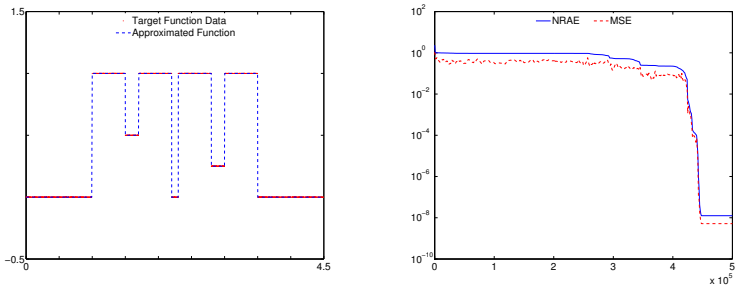
3.2 Discussion

In this section, experimental results are demonstrated and discussed. First, as an example to visually show the training results, two approximated functions and learning curves separately obtained by the MSE and NRAE training method in one training group are selected and plotted in Fig. 1. Although six different values of λ are chosen to perform the NRAE training sessions in this selected training group, only one plot of the approximated function with $\lambda = 10^6$ is shown. Those approximated function plots achieved by other five NRAE training sessions with $\lambda = 10^7, 10^8, 10^9, 10^{10}$ and 10^{11} are exactly the same as Fig. 1(c), and learning curves for them are shown in Fig. 2. At last, a comparison of training errors achieved by different guesses of five initial weight vectors between the MSE and NRAE training sessions is illustrated in Fig. 3.

Since the three-notch function is intended to have typical non-global minima, the observations in our experimental results demonstrate that the NRAE training with a sufficiently large λ has the capability to avoid non-global local minima comparing with the MSE training. First, approximated function plots in Fig. 1(a) and Fig. 1(c) show that the NRAE training with a sufficiently large λ captures all significant features located in the target three-notch function, but the MSE training only finds few parts of those features. Second, learning curves in Fig. 1 and Fig. 2 present similar patterns for the NRAE training with sufficiently large values of λ to reach the global or nearly global minimum. Third, results in Fig. 3 indicate that the NRAE training sessions with sufficiently large values of λ consistently lead all trained MLPs to achieve satisfactory training errors, which are lower than the MSE training.



(a) The MSE Training (b) Learning Curve

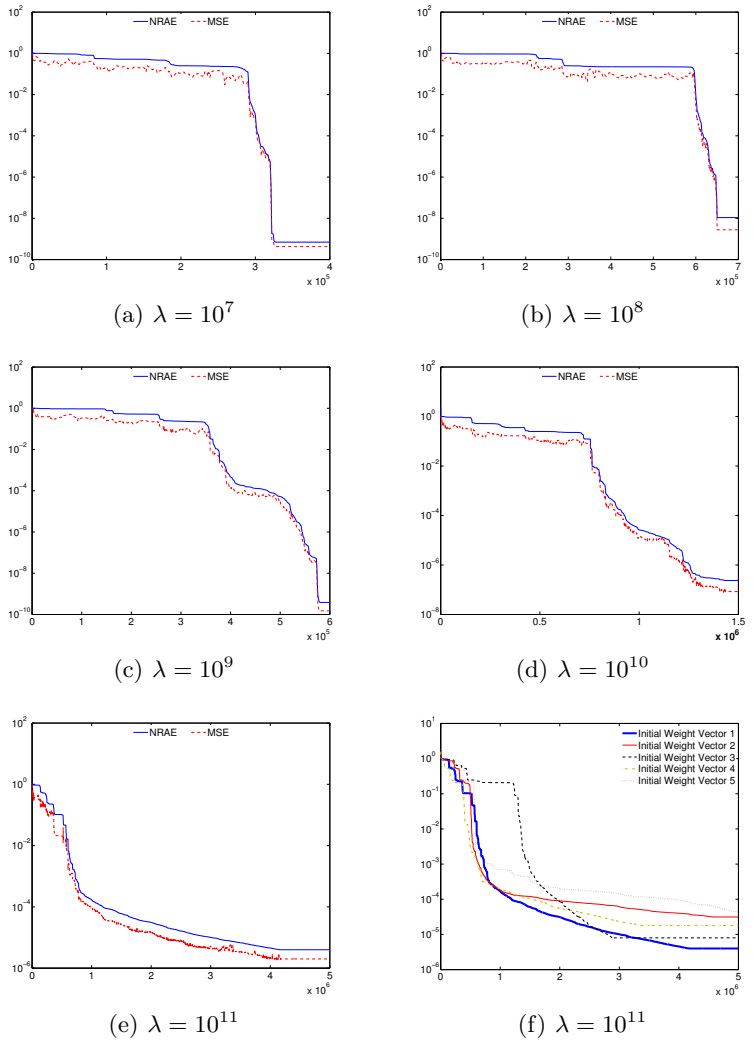


(c) The NRAE Training (d) Learning Curve

Fig. 1. Results of the three-notch function approximation with the MSE and NRAE training. Figures on the left side column are function plots, and numbers on the horizontal and vertical axes in each subfigure denote the input and output of the function, respectively. Figures on the right side column are learning curves for the corresponding training criteria, and numbers on the horizontal and vertical axes in each subfigure denote the values of training epochs and errors, respectively. Here, the actual values of training errors are converted to the logarithmic numbers with respect to base 10. The represented NRAE training result and learning curves concerning the NRAE and MSE criteria are obtained when $\lambda = 10^6$.

4 Conclusion

The NRAE training criterion does not have a non-global local minimum in nearly the entire weight space, provided that the risk sensitivity index λ of the NRAE is sufficiently large. We propose to use the NRAE criterion to train an MLPs that has enough hidden neurons to approximate the target function nearly perfectly. To select a sufficiently large λ, we start with a large number, say 10^6, as long as the computer can handle the NRAE with this λ and the local search optimization method (e.g., the BFGS and conjugate gradient method) applied to minimize this NRAE. If the NRAE criterion does not converge to zero, we increase the risk sensitivity index by multiplying it by 10. We continue increasing λ in this manner, if necessary, until the NRAE and MSE are nearly zero.



(a) $\lambda = 10^7$ (b) $\lambda = 10^8$

(c) $\lambda = 10^9$ (d) $\lambda = 10^{10}$

(e) $\lambda = 10^{11}$ (f) $\lambda = 10^{11}$

Fig. 2. Learning Curves for the three-notch function approximation with the NRAE training. Figures from Fig. 2(a) to Fig. 2(e) illustrate different trends of the NRAE training with respect to both the NRAE and MSE criteria as increasing of λ. Fig. 2(f) shows only the learning curves concerning the NRAE criterion for five different initial weight vectors when $\lambda = 10^{11}$. Numbers on the horizontal axis are the values of training epochs. Numbers on the vertical axis are the values of training errors which are converted to the logarithmic numbers with respect to base 10.

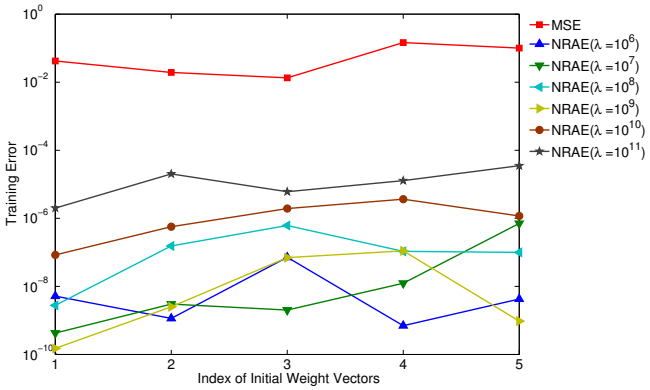


Fig. 3. Training errors of five different initial weight vectors for the three-notch function approximation with the MSE and NRAE training. Colors and symbols in the showed lines are used to distinguish the MSE and NRAE training methods, or describe independent NRAE training sessions with different values of λ. Here, all actual values on the vertical axis are converted to the logarithmic numbers with respect to base 10.

References

1. Aarts, E., Korst, J.: The Neuron. Oxford University Press (1989)
2. Zurada, J.M.: Introduction to Artificial Neural Networks. West Publishing Company, St. Paul, MN (1992)
3. Hassoun, M.H.: Fundamentals of Artificial Neural Networks. MIT Press, Cambridge, Massachusetts (1995)
4. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, third edition. Springer, New York (1999)
5. Principe, J.C., Euliano, N.R., Lefebvre, W.C.: Neural and Adaptive Systems: Fundamentals through Simulations. John Wiley and Sons, Inc., New York (2000)
6. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York, NY (2006)
7. Du, K.L., Swamy, M.: Neural Networks in a Softcomputing Framework. Springer, New York, NY (2006)
8. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C: The Art of Scientific Computing. Third edn. Cambridge University Press, New York, NY (2007)
9. Haykin, S.: Neural Networks and Learning Machines. Third edn. Prentice Hall, Upper Saddle River, New Jersey (2008)
10. Lo, J.T.-H., Bassu, D.: An adaptive method of training multilayer perceptrons. In: Proc. International Joint Conference on Neural Networks (IJCNN'01). Volume 3. (2001) 2013–2018
11. Lo, J.T.-H.: Convexification for data fitting. Journal of Global Optimization **46**(2) (2010) 307–315