On Weight-Noise-Injection Training

Kevin Ho[1], Chi-sing Leung[2], and John Sum[3]*

[1] Department of Computer Science and Communication Engineering,
Providence University, Sha-Lu, Taiwan. ho@pu.edu.tw
[2] Department of Electronic Engineering, City University of Hong Kong
Kowloon Tong, KLN, Hong Kong eeleungc@cityu.edu.hk
[3] Institute of E-Commerce, National Chung Hsing University
Taichung 402, Taiwan pfsum@nchu.edu.tw

Abstract. While injecting weight noise during training has been proposed for more than a decade to improve the convergence, generalization and fault tolerance of a neural network, not much theoretical work has been done to its convergence proof and the objective function that it is minimizing. By applying the Gladyshev Theorem, it is shown that the convergence of injecting weight noise during training an RBF network is almost sure. Besides, the corresponding objective function is essentially the mean square errors (MSE). This objective function indicates that injecting weight noise during training an radial basis function (RBF) network is not able to improve fault tolerance. Despite this technique has been effectively applied to multilayer perceptron, further analysis on the expected update equation of training MLP with weight noise injection is presented. The performance difference between these two models by applying weight injection is discussed.

1 Introduction

Many methods have been developed throughout the last two decades to improve the fault tolerance of a neural network. Well known methods include injecting random fault during training [25, 5], introducing network redundancy [23], applying weight decay learning [9], formulating the training algorithm as a nonlinear constraint optimization problem [10, 22], bounding weight magnitude during training [7, 15, 17], and adding fault tolerant regularizer [2, 19, 27]. A complete survey on fault tolerant learning methods is exhaustive. Readers please refer to [8] and [29] for reference.

Amongst all, the fault-injection-based on-line learning algorithms are of least theoretical studied. By fault injection, either fault or noise is introduced to a neural network model before each step of training. This fault could either be node fault (stuck-at-zero), weight noise or input noise. As many studies have been reported in the literature on input noise injection [1, 4, 24, 13, 14], the primary focus of this paper is on weight noise injection. Our companion paper [28] will be focus on node fault injection.

* Corresponding author.

Suppose a neural network consists of M weights. Let $\theta \in R^M$ be the weight vector of a neural network model and the update equation is given by $\theta(t+1) = \theta(t) - F(x(t+1), y(t+1), \theta(t))$. The idea of weight noise injection is to replace $\theta(t)$ in the factor $F(\cdot, \cdot, \theta(t))$ by $F(\cdot, \cdot, \tilde{\theta}(t))$. Here the elements of $\tilde{\theta}(t)$ is of the form $\tilde{\theta}_i(t) = \theta_i(t) + \Delta\theta_i(t)$. The factor $\Delta\theta_i(t)$ is the weight noise injected. The update equation is thus defined as follows :

$$\theta(t+1) = \theta(t) - F(x(t+1), y(t+1), \tilde{\theta}(t)). \quad (1)$$

Despite injecting weight noise to improve convergence ability, generalization and fault tolerance have long been investigated [20, 21, 16, 11] for MLPs and recurrent neural networks, and theoretical analysis on applying such technique to MLP has been reported [1], little is known about the effect of injecting weight noise during training an RBF network.

In this paper, an analysis on weight-noise-injection-based training will be presented. In the next section, the convergence proof and the objective function of RBF training with on-line weight injection will be analyzed. Section 3 will show the analysis on the case of MLP. The conclusion will be presented in Section 4.

2 RBF training with weight noise injection

2.1 Network model

Let $\mathcal{M}_0$ be an unknown system to be modeled. The input and output of $\mathcal{M}_0$ are denoted by x and y respectively. The only information we know about $\mathcal{M}_0$ is a set of measurement data $\mathcal{D}$, where $\mathcal{D} = \{(x_k, y_k)\}_{k=1}^N$. Making use of this data set, an estimated model $\hat{\mathcal{M}}$ that is *good* enough to capture the *general behavior* of the unknown system can be obtained.

For $k = 1, 2, \cdots, N$, we assume that the true model is governed by an unknown deterministic system $f(x)$ together with mean zero Gaussian output noise :

$$\mathcal{M}_0 \quad : \quad y_k = f(x_k) + e_k, \quad (2)$$

Besides, we assume that the unknown system $f(x)$ can be realized by an RBF network consisting of M hidden nodes, i.e.

$$y_k = \sum_{i=1}^{M} \theta_i^* \phi_i(x_k) + e_k \quad (3)$$

for all $k = 1, 2, \cdots, N$ and $\phi_i(x)$ for all $i = 1, 2, \cdots, M$ are the radial basis functions given by $\phi_i(x) = \exp\left(-\frac{\|x-c_i\|^2}{\sigma}\right)$, where c_is are the centers of the radial basis function and the positive parameter $\sigma > 0$ controls the width of the radial basis functions. In vector form, Equation (3) can be rewritten as follows :

$$y_k = \phi(x_k)^T \theta^* + e_k, \quad (4)$$

where $\phi(\cdot) = (\phi_1(\cdot), \phi_2(\cdot), \cdots, \phi_M(\cdot))^T$ and $\theta^* = (\theta_1^*, \theta_2^*, \cdots, \theta_M^*)^T$.

2.2 Weight noise injection training

While a network is trained by the idea of weight noise injection, the update equation will be given by

$$\theta(t+1) = \theta(t) + \mu_t(y_t - \phi^T(x_t)\tilde{\theta}(t))\phi(x_t), \tag{5}$$

where μ_t is (for $t \geq 1$) the step size at the t^{th} iteration,

$$\tilde{\theta}_i(t) = \begin{cases} \theta_i(t) + \beta_i & \text{for additive noise injection,} \\ \theta_i(t) + \beta_i\theta_i(t) & \text{for multiplicative noise injection.} \end{cases} \tag{6}$$

β_i for all $i = 1, 2, \cdots, M$ are independent mean zero Gaussian noise with variance S_β. Normally, it is assumed that the value of S_β is small. Although the theoretical proof presented later in this paper applies to any bounded value, it is meaningless to consider a large value of S_β.

2.3 Convergence and objective function

Theory of stochastic approximation has been developed for more than half a century for the analysis of recursive algorithms. Advanced theoretical works for complicated recursive algorithms have still been under investigation [18]. The theorem applied in this paper is based on Gladyshev Theorem [12].

Theorem 1 (Gladyshev Theorem [12]). *Let $\theta(t)$ and $M(\theta(t), \omega(t))$ for all $t = 0, 1, 2,$ and so on be m-vectors. $\omega(t)$ for all $t = 0, 1, 2,$ and so on are i.i.d. random vectors with probability density function $P(\omega)$* [4]*. Consider a recursive algorithm defined as follows :*

$$\theta(t+1) = \theta(t) - \mu_t M(\theta(t), \omega(t)). \tag{7}$$

In which, the expectation of $M(\theta, \omega)$ over ω,

$$\bar{M}(\theta) = \int M(\theta, \omega)P(\omega)d\omega, \tag{8}$$

has unique solution θ^* such that $\bar{M}(\theta^*) = 0$.

Suppose there exists positive constants κ_1 and κ_2 such that the following conditions are satisfied :

(C1) $\mu_t \geq 0$, $\sum_t \mu_t = \infty$ *and* $\sum_t \mu_t^2 < \infty$.
(C2) $\inf_{\varepsilon < \|\theta - \theta^*\| < \varepsilon^{-1}} (\theta - \theta^*)^T \bar{M}(\theta) > 0$, *for all* $\varepsilon > 0$.
(C3) $\int \|M(\theta, \omega)\|^2 P(\omega)d\omega \leq \kappa_1 + \kappa_2\|\theta\|^2$.

Then for $t \to \infty$, $\theta(t)$ converges to θ^* with probability one.

[4] In the following convergence proof, $\omega(t) = (x_t, y_t, \beta_t)$. Owing not to confuse the time index t with the element index k, the subscript t is omitted. So that $\omega(t) = (x_t, y_t, \beta)$.

Applying Gladyshev Theorem, the following theorem can be proved for injecting weight noise.

Theorem 2. *For injecting (additive or multiplicative) weight noise during training an RBF network, the weight vector $\theta(t)$ will converge with probability one to*

$$\theta^* = \left(\frac{1}{N} \sum_{k=1}^{N} \phi(x_k)\phi^T(x_k) \right)^{-1} \frac{1}{N} \sum_{k=1}^{N} y_k \phi(x_k). \tag{9}$$

(Proof) For a RBF network that is trained by injecting multiplicative weight noise,

$$\theta(t+1) = \theta(t) + \mu_t(y_t - \phi^T(x_t)\tilde{\theta}(t))\phi(x_t), \tag{10}$$

$$\tilde{\theta_i} = (1+\beta_i)\theta_i, \quad \beta_i \sim \mathcal{N}(0, S_\beta), \;\; \forall\, i = 1, \cdots, M. \tag{11}$$

Suppose S_β is small. Taking expectation of the second term in right hand side of the first equation with respect to β, it can readily be shown that

$$\int_{\Omega_{\tilde{\theta}(t)}} (y_t - \phi^T(x_t)\tilde{\theta}(t))\phi(x_t) d\tilde{\theta}(t) = (y_t - \phi^T(x_t)\theta(t))\phi(x_t).$$

Further taking expectation of the above equation with respect to x_t and y_t, $h(\theta(t))$ will be given by

$$h(\theta(t)) = \frac{1}{N} \sum_{k=1}^{N} (y_k - \phi^T(x_k)\theta(t))\phi(x_k), \tag{12}$$

Therefore, the solution θ^* is $\theta^* = H_\phi^{-1} Y$.

Next, we are going to apply the Gladyshev Theorem for the convergence proof. Normally, the first condition can easily be satisfied. It is because the step size μ_t could be pre-defined. So, we skip the proof of Condition (C1) for simplicity.

To prove Condition (C2), we first note that $\bar{M}(\theta) = -h(\theta)$. We further let $Y = \frac{1}{N}\sum_{k=1}^{N} y_k \phi(x_k)$ and $\omega = (x_t, y_t, \beta)$. Hence, for all $\|\theta - \theta^*\| > 0$, we have $-(\theta - \theta^*)^T h(\theta) = -(\theta - \theta^*)^T (Y - H_\phi \theta)$, which is greater than zero.

To prove Condition $(C3)$, we consider the Equation (10). By triangle inequality,

$$\|M(\theta, \omega)\|^2 \leq \|y_t \phi(x_t)\|^2 + \|\phi(x_t)\phi(x_t)^T \theta\|^2 + \|\phi(x_t)\phi(x_t)^T A_\theta \beta\|^2, \tag{13}$$

where $A_\theta = \mathbf{diag}\{\theta_1, \theta_2, \cdots, \theta_M\}$. Clearly, the first term in the RHS of the inequality is a factor independent of θ. We let it be $K(x_t, y_t)$ as before. The second term is $\theta^T(\phi(x_t)\phi(x_t)^T)^2\theta$. In which the matrix $(\phi(x_t)\phi(x_t)^T)^2$ is symmetric and of bounded elements. Therefore, its largest eigenvalue must also be a bounded nonnegative number, say $\lambda(x_t)$. Taking expectation of the third term with respect to β,

$$\begin{aligned}\|\phi(x_t)\phi(x_t)^T A_\theta \beta\|^2 &= S_\beta \sum_{i=1}^{M} \theta^2 \left(\phi(x_t)\phi(x_t)^T \phi(x_t)\phi(x_t)^T\right)_{ii}, \\ &\leq S_\beta \max_i \{(\phi(x_t)\phi(x_t)^T \phi(x_t)\phi(x_t)^T)_{ii}\} \|\theta\|^2.\end{aligned}$$

As a result,

$$\int \|M(\theta, \omega)\| P(\beta) d\beta \leq K(x_t, y_t) + S_\beta (\lambda(x_t) \max_i \{(\phi(x_t)\phi(x_t)^T)_{ii}^2\}) \|\theta\|^2.$$

Further taking the expectation of the above inequality with respect to x_t and y_t, one can readily show that Condition ($C3$) can be satisfied and the proof is completed.

To prove the convergence of injecting additive weight noise, simply defining $\tilde{\theta}(t)$ in Equation (10) by $\theta(t) + \beta$ and A_θ in Equation (13) by an $M \times M$ identity matrix. It will be clearly that $h(\theta(t))$ will be identical to Equation (12) and the third term will be independent of θ. The proof of Condition (C3) will be accomplished. **Q.E.D.**

As the solution θ^*, by either injecting additive weight noise or multiplicative weight noise, is identical to the solution obtained by the ordinary pseudo-inverse, the following theorem can be implied.

Theorem 3. *The objective function of injecting (additive or multiplicative) weight noise during training an RBF is identical to the mean square errors.*

$$\mathcal{L}(\theta|\mathcal{D}) = \frac{1}{N} \sum_{k=1}^{N} (y_k - f(x_k, \theta))^2. \tag{14}$$

3 MLP training with weight noise injection

3.1 Injecting multiplicative weight noise

Consider a nonlinear neural network $g(x, \theta)$, where both its gradient vector $g_\theta(x, \theta)$ and Hessian matrix $g_{\theta\theta}(x, \theta)$ exist. Similar to that of RBF learning, the online weight noise injection learning algorithm for $g(x, \theta)$ given a dataset $\mathcal{D} = \{(x_k, y_k)\}_{k=1}^N$ can be written as follows :

$$\theta(t+1) = \theta(t) + \mu_t (y_t - g(x_t, \tilde{\theta}(t))) g_\theta(x_t, \tilde{\theta}(t)). \tag{15}$$

$$\tilde{\theta}(t) = \theta(t) + A_\beta \theta(t). \tag{16}$$

Here, $A_\beta = \mathbf{diag}\{\beta_1, \beta_2, \cdots, \beta_M\}, \quad \beta_i \sim \mathcal{N}(0, S_\beta)$. For small S_β, one can assume that $\tilde{\theta}$ is close to θ and then apply Taylor expansion to $g(\cdot, \cdot)$ and $g_\theta(\cdot, \cdot)$ and get that

$$g(x_t, \tilde{\theta}(t)) \approx g(x_t, \theta(t)) + g_\theta(x_t, \theta(t))^T A_\beta \theta(t), \tag{17}$$

$$g_\theta(x_t, \tilde{\theta}(t)) \approx g_\theta(x_t, \theta(t)) + g_{\theta\theta}(x_t, \theta(t)) A_\beta \theta(t). \tag{18}$$

Putting the above approximations into Equation (15) and taking expectation over β, it is readily shown that

$$h(\theta(t)) = \frac{1}{N}\sum_{k=1}^{N}(y_t - g(x_t, \theta(t)))g_\theta(x_t, \theta) - \frac{S_\beta}{N}\sum_{k=1}^{N}\Psi(x_t, \theta(t))\vartheta(t), \qquad (19)$$

where $\vartheta = (\theta_1^2, \theta_2^2, \cdots, \theta_M^2)^T$ and $\Psi(x_t, \theta(t)) = g_{\theta\theta}(x_t, \theta(t))\mathbf{diag}\{g_\theta(x_t, \theta(t))\}$.

Clearly, the first term on the RHS of Equation (19) is proportional to the negative gradient of the MSE term. However, the anti-derivative of the second term is difficult. The corresponding objective function and the convergence proof can hardly be analyzed.

Except the case when the MLP output is linear, i.e. $g(x, w, v) = \sum_i w_i T_i(x, v)$, where w is the output weight vector and v is the input weight vector. $T_i(\cdot, \cdot)$ is the output of the i^{th} hidden unit. In such case, $\frac{\partial^2}{\partial w_i \partial w_j} g(x_t, w, v) = 0$. Therefore, enhancing fault tolerance of a MLP with linear output nodes cannot be achieved by simply adding noise to the output weights during training.

3.2 Injecting additive weight noise

For the case that the injection weight noise is additive, the corresponding $h(\theta)$ can readily be obtained by replacing $A_\beta\theta(t)$ in Equation (16), Equation (17) and Equation (18) to β. Then,

$$h(\theta(t)) = \frac{1}{N}\sum_{k=1}^{N}(y_t - g(x_t, \theta(t)))g_\theta(x_t, \theta) - \frac{S_\beta}{N}\sum_{k=1}^{N} g_{\theta\theta}(x_t, \theta(t))g_\theta(x_t, \theta(t)). \qquad (20)$$

Clearly, the objective function minimized by $h(\theta(t))$ will be given by

$$\frac{1}{2N}\sum_{k=1}^{N}(y_t - g(x_t, \theta(t)))^2 + \frac{S_\beta}{2N}\sum_{k=1}^{N}\|g_\theta(x_t, \theta(t))\|^2. \qquad (21)$$

Suppose the MLP is of linear output and additive weight noise is added only to the output layer, this objective function will become

$$\frac{1}{2N}\sum_{k=1}^{N}(y_t - g(x_t, \theta(t)))^2 + \frac{S_\beta}{2N}\sum_{k=1}^{N}\sum_{i} T_i^2(x, v). \qquad (22)$$

The second term plays the role controlling the magnitude of the output of the hidden nodes.

Remark: One should note that the analysis in this section is purely heuristic, not analytically. Our analysis is focus on the expected updated equation, not the actual update equation. The reason is because the convergence proof for nonlinear system is not straight forward. As mentioned in [18], to prove the convergence of a nonlinear stochastic gradient descent algorithm, one needs to show that either (i) $\theta(t)$ can always be bounded or (ii) $\theta(t)$ can visit a local bound region infinite often. The two conditions are not easy to prove. Although, simulation results can also show that $\theta(t)$ is bounded for all t. Analytical proof has yet to be shown.

4 Conclusions

In this paper, analysis on the behavior of weight-noise-injection training has been presented. In contrast to the approach taken by An [1], we focus on the actual on-line update equation. From this, the convergence of weight-noise-injection training applying to RBF is proved analytically and the true objective function being minimized is revealed. Either for adding multiplicative or additive weight noise, it is found that the objective function being minimized is actually the mean square errors. Therefore, adding weight noise during training a RBF network can neither improve fault tolerance nor generalization.

For MLP, due to its nonlinearity, boundedness on $\theta(t)$ has yet been proven. Therefore, only analysis on the properties of the expected update equations has been presented. For the case of adding additive weight noise during training, it is shown that the objective function consists of two terms. The first term is the usual mean square term. But the second plays a role to regularize the magnitude of the output of the hidden units.

Acknowledgement: The research work reported in this paper is supported in part by Taiwan NSC Research Grant 97-2221-E-005-050.

References

1. An G. The effects of adding noise during backpropagation training on a generalization performance, *Neural Computation*, Vol.8, 643-674, 1996.
2. Bernier J.L. *et al*, Obtaining fault tolerance multilayer perceptrons using an explicit regularization, *Neural Processing Letters*, Vol.12, 107-113, 2000.
3. Bernier J.L. *et al*, A quantitative study of fault tolerance, noise immunity and generalization ability of MLPs, *Neural Computation*, Vol.12, 2941-2964, 2000.
4. Bishop C.M., Training with noise is equivalent to Tikhnov regularization, *Neural Computation*, Vol.7, 108-116, 1995.
5. Bolt G., *Fault tolerant in multi-layer Perceptrons.* PhD Thesis, University of York, UK, 1992.
6. Bottou L., Stochastic gradient learning in neural networks, *NEURO NIMES'91*, 687-706, 1991.
7. Cavalieri S. and O. Mirabella, A novel learning algorithm which improves the partial fault tolerance of multilayer NNs, *Neural Networks*, Vol.12, 91-106, 1999.
8. Chandra P. and Y. Singh, Fault tolerance of feedforward artificial neural networks – A framework of study, *Proceedings of IJCNN'03* Vol.1 489-494, 2003.
9. Chiu C.T. *et al.*, Modifying training algorithms for improved fault tolerance, ICNN'94 Vol.I, 333-338, 1994.

10. Deodhare D., M. Vidyasagar and S. Sathiya Keerthi, Synthesis of fault-tolerant feedforward neural networks using minimax optimization, *IEEE Transactions on Neural Networks*, Vol.9(5), 891-900, 1998.
11. Edwards P.J. and A.F. Murray, Fault tolerant via weight noise in analog VLSI implementations of MLP's – A case study with EPSILON, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol.45, No.9, p.1255-1262, Sep 1998.
12. Gladyshev E., On stochastic approximation, *Theory of Probability and its Applications*, Vol.10, 275-278, 1965.
13. Grandvalet Y., S. Canu, A comment on noise injection into inputs in backpropagation learning, *IEEE Transactions on Systems, Man, and Cybernetics*, 25(4), p.678-681, 1995.
14. Grandvalet Y., S. Canu, S. Boucheron, Noise injection : Theoretical prospects, *Neural Computation*, Vol.9(5), p.1093-1108, 1997.
15. Hammadi N.C. and I. Hideo, A learning algorithm for fault tolerant feedforward neural networks, *IEICE Transactions on Information & Systems*, Vol. E80-D, No.1, 1997.
16. Jim K.C., C.L. Giles and B.G. Horne, An analysis of noise in recurrent neural networks: Convergence and generalization, *IEEE Transactions on Neural Networks*, Vol.7, 1424-1438, 1996.
17. Kamiura N., *et al*, On a weight limit approach for enhancing fault tolerance of feedforward neural networks, *IEICE Transactions on Information & Systems*, Vol. E83-D, No.11, 2000.
18. Lai T.L., Stochastic approximation, *Annals of Statistics*, Vol. 31, No. 2, 391-406, 2003.
19. Leung C.S., J. Sum, A fault tolerant regularizer for RBF networks, *IEEE Transactions on Neural Networks*, Vol. 19 (3), pp.493-507, 2008.
20. Murray A.F. and P.J. Edwards, Synaptic weight noise during multilayer perceptron training: fault tolerance and training improvements, *IEEE Transactions on Neural Networks*, Vol.4(4), 722-725, 1993.
21. Murray A.F. and P.J. Edwards, Enhanced MLP performance and fault tolerance resulting from synaptic weight noise during training, *IEEE Transactions on Neural Networks*, Vol.5(5), 792-802, 1994.
22. Neti C. M.H. Schneider and E.D. Young, Maximally fault tolerance neural networks, *IEEE Transactions on Neural Networks*, Vol.3(1), 14-23, 1992.
23. Phatak D.S. and I. Koren, Complete and partial fault tolerance of feedforward neural nets., *IEEE Transactions on Neural Networks*, Vol.6, 446-456, 1995.
24. Reed R., R.J. Marks II & S. Oh, Similarities of error regularization, sigmoid gain scaling, target smoothing, and training with jitter, *IEEE Transactions on Neural Networks*, Vol.6(3), 529-538, 1995.
25. Sequin C.H. and R.D. Clay, Fault tolerance in feedforward artificial neural networks, *Neural Networks*, Vol.4, 111-141, 1991.
26. Sum J., C.S. Leung and L. Hsu, Fault tolerant learning using Kullback-Leibler Divergence, in *Proc. TENCON'2007* Taipei, 2007.
27. Sum J., C.S. Leung and K. Ho, On objective function, regularizer and prediction error of a learning algorithm for dealing with multiplicative weight noise, *IEEE Transactions on Neural Networks* Vol.20(1), Jan, 2009.
28. Sum J., C.S. Leung, and K. Ho, On node-fault-injection training an RBF network in *Proc. ICONIP'2008*, Springer LNCS, 2009.
29. Tchernev E.B., R.G. Mulvaney, and D.S. Phatak, Investigating the Fault Tolerance of Neural Networks, *Neural Computation*, Vol.17, 1646-1664, 2005.