Convergence analysis of on-line weight noise injection training algorithms for MLP networks

John Sum, Chi-sing Leung and Kevin Ho

Abstract—Injecting weight noise during training has been proposed for almost two decades as a simple technique to improve fault tolerance and generalization of a multilayer perceptron (MLP). However, little has been done regarding their convergence behaviors. Therefore, we present in this paper the convergence proofs of two of these algorithms for MLPs. One is based on combining injecting multiplicative weight noise and weight decay (MWN-WD) during training. The other is based on combining injecting additive weight noise and weight decay (AWN-WD) during training. Let $m$ be the number of hidden nodes of a MLP, $\alpha$ be the weight decay constant and $S_n$ be the noise variance. It is showed that the convergence of MWN-WD algorithm is with probability one if $\alpha > \sqrt{S_n}/m$. While the convergence of the AWN-WD algorithm is with probability one if $\alpha > 0$.

I. INTRODUCTION

To improve the fault tolerance of a multilayer perceptron (MLP), Murray & Edward [21], [22], [13] modified the conventional backpropagation training by injecting multiplicative weight noise during each step of training. By simulations on character encoder and eye-classifier problems, they found that the resultant multilayer perceptron has better tolerance ability against random weight fault and weight perturbation. Applying the same technique in real-time-recurrent-learning (RTRL), Jim et al [17] have also found that the generalization of a RNN can be improved. Moreover, the convergence speed is faster than conventional RTRL. While on-line weight noise injection training algorithms have succeeded in improving fault tolerance of a MLP, the generalization of a RNN, and convergence speed of training, not much analytical work has been done in regard to the (i) convergence proofs and (ii) objective functions of these algorithms.

Even the authors in [12], [22], [17] have only provided preliminary analyses on the effect of the prediction error of a neural network that is corrupted by weight noise (see Section II.C in [12], [22] for the analysis for MLP and see Section 3 in [17] for the analysis for RNN). G. An in [1] has attempted to solve these problems. In his paper, he considered three different on-line back-propagation training with noise injection. One of them is based on additive weight noise injection (see Section 4 in [1]). While his works in the other two algorithms are correct, his analysis on the case of weight noise injection is questionable. It is because he has not verified if the algorithm based on weight noise injection fulfills the conditions depicted in Bottou’s Theorem [8]. By following the mathematics in [1], one can clearly figure out that the cost function derived by G. An is the prediction error of a MLP if it is corrupted by additive weight noise. It is not the corresponding objective function for on-line weight noise injection training algorithm for MLP.

Even though some other works have been done regarding the prediction error (or sensitivity analysis) of a MLPs [2], [3], [4], [5], [6], [12], [23], none of them worked on their convergence proofs. Until recently, we have showed that the convergence of injecting weight noise during training a RBF is with probability one [14], [16]. Nevertheless, we have showed that the objective function of injecting multiplicative weight noise (or additive weight noise) during training is essentially the mean square errors function. It means, injecting weight noise during training does not help to improve the fault tolerance or the generalization ability of a RBF. Unfortunately, our approach to the proof for RBF [16] cannot be applied to MLPs simply because Gladyshev Theorem is not applicable to MLPs.

After all, for almost fifteen years, the convergence proofs of these weight noise injection-based algorithms for MLP have yet been accomplished and their corresponding objective functions are still unknown.

Therefore, the primary focus of this paper is to analyze the convergence of these weight noise injection-based algorithms with application to MLPs. Two specific algorithms will be analyzed. The first one is based on combining multiplicative weight noise injection and weight decay during training. While the other is based on combining additive weight noise injection and weight decay during training. The main theorem we applied is the classical Doob’s Martingale Convergence Theorem [11], [10].

Clearly, many theorems in stochastic approximation (including Ljung’s Theorem [20] and its variant [27], Kushner & Clark Lemma [18] and the theorems developed by Bottou [8], [9]) have been applied to prove the convergence of different training algorithms for MLPs. But, they are not easily applied for the weight noise injection-based algorithms as some conditions like the infinite often condition [20], [18] and the boundedness [8], [9] are not easy to prove.

For these reasons, we applied the Doob’s Martingale Convergence Theorem. As it is the fundamental theorem that the theory of stochastic approximation relies on. For our convergence proofs, it turns out to be simpler. The condition on the training step size can be showed to be not as restrictive...
as those assumed in some theorems presented in literatures [25], [27], [28], [29].

The rest of the paper will present the main convergence theorems and the corresponding proofs for these weight noise injection-based algorithms for MLPs. In the next section, the background on the network model, the weight decay training algorithm will be described. Then in Section 3, the algorithms based on combining weight noise injection and weight decay during training will be summarized. Their corresponding objective functions will be reviewed. In Section 4, the convergence of the algorithm based on combining multiplicative weight noise and weight decay during training will be proved. The convergence of the algorithm based on combining additive weight noise and weight decay during training will be proved in Section 5. Section 6 will prove that with probability one their weight vectors converge to local minimum of their corresponding objective functions. Finally, a few aspects regarding the conditions on the step size and the extension of the theorems to the convergence of weight decay algorithm are discussed in Section 7. Conclusions are given in the last section.

II. BACKGROUND

We assume that the training data set \( \mathcal{D} = \{(x_k, y_k)\}_{k=1}^N \) is generated by an unknown system, where \( x_k \in \mathbb{R}^n \) is the input vector and \( y_k \in \mathbb{R} \) is the output.

A. Network Model

This unknown system is thus approximated by a MLP with \( n \) input nodes, \( m \) hidden nodes, and one linear output node, defined as follows:

\[
f(x_k, d, A, c) = d^T z(A^T x_k + c),
\]

where \( A = [a_1, \ldots, a_m] \in \mathbb{R}^{n \times m} \) is the input-to-hidden weight matrix, \( a_i \in \mathbb{R}^n \) is the input weight vector associated with the \( i^{th} \) hidden node, \( c = (c_1, \ldots, c_m)^T \in \mathbb{R}^m \) is the input-to-hidden bias vector, \( d \in \mathbb{R}^m \) is the hidden-to-output weight vector, and \( z = (z_1, \ldots, z_m)^T \in \mathbb{R}^m \) is output vector of the hidden layer in which

\[
z_i(x_k, a_i, c_i) = \frac{1}{1 + \exp(-(a_i^T x_k + c_i))}
\]

for \( i = 1, 2, \ldots, m \).

For the sake of presentation, we let \( w_i \in \mathbb{R}^{(n+2)} \) be the parametric vector associated to the \( i^{th} \) hidden node, i.e.

\[
w_i = (d_i, a_i^T, c_i)^T,
\]

and \( w \in \mathbb{R}^{m(n+2)} \) be a parametric vector augmenting all the parametric vectors \( w_1, w_2, \ldots, w_m \). The output is denoted as \( f(x_k, w) \). Throughout the paper, we call \( w_1, w_2, \ldots, w_m \) and \( w \) the weight vectors.

Next, we let \( g(x_k, w) = \nabla_w f(x_k, w) \), where

\[
g(x_k, w) = (\nabla_w f(x_k, w)^T, \ldots, \nabla_w m f(x_k, w)^T)^T.
\]

As \( \nabla_w f(x_k, w) \) depends entirely on \( x_k \) and \( w_i \), we denote it by \( g_i(x_k, w_i) \). Thus,

\[
g(x_k, w) = (g_1(x_k, w_1)^T, \ldots, g_m(x_k, w_m)^T)^T,
\]

and

\[
g_i(x_k, w_i) = \begin{bmatrix} z_i \\ d_i z_i (1 - z_i) x_k \\ d_i z_i (1 - z_i) \end{bmatrix},
\]

where \( z_i = z_i(x_k, a_i, c_i) \).

If we let \( \nabla_w g(x, w) \in \mathbb{R}^{m(n+2) \times m(n+2)} \) be the Hessian matrix of \( f(x, w) \) with respect to the weight vector \( w \), one can readily show that

\[
\nabla_w g(x, w) = \begin{bmatrix} \nabla_{w_1} g_1(x, w_1) & \cdots & 0_{(n+2) \times (n+2)} \\ \vdots & \ddots & \vdots \\ 0_{(n+2) \times (n+2)} & \cdots & \nabla_{w_m} g_m(x, w_m) \end{bmatrix}
\]

where

\[
\nabla_w g_i(x, w) = \nabla \nabla_w f(x, w)
\]

for all \( i = 1, \ldots, m \).

B. Weight Decay Training

In weight decay training, a sample is randomly drawn from the dataset \( \mathcal{D} \) at each update step. We denote the sample being selected at the \( i^{th} \) step as \( \{x_t, y_t\} \). Once the input \( x_t \) has been fed in the MLP, the output is calculated by (1) and (2).

\[
f(x_t, w(t)) = d(t)^T z(t)
\]

\[
z(t) = z(A(t)^T x_t + c(t)).
\]

By replacing \( w_i \) and \( x_k \) in (4) by \( w(t) \) and \( x_t \), respectively, the update equations for the weight vectors \( w_i \) (for \( i = 1, 2, \ldots, m \)) can thus be written as follows:

\[
w_i(t + 1) - w_i(t) = \mu(t) \{ (y_t - f(x_t, w(t))) g_i(x_t, w_i(t)) - \alpha w_i(t) \}.
\]

where \( \mu(t) \) is the step size at the \( t^{th} \) step, and \( \alpha > 0 \) is the decay constant. The last term \(-\alpha w_i(t) \) in (9) is sometimes called forgetting term [19].

It has been proved in [27], [29] that the convergence of (9) is with probability one if \( \sum t \mu(t) = \infty \) and \( \sum t \mu(t)^2 < \infty \). However, we will show later in this paper that these conditions can be replaced by \( \mu(t) \rightarrow 0 \).

III. COMBINING WEIGHT NOISE INJECTION AND WEIGHT DECAY DURING TRAINING

Let \( b_1(t), b_2(t), \ldots, b_m(t) \) be random vectors associated with the weight vectors \( w_1(t), w_2(t), \ldots, w_m(t) \) at step \( t \). Elements in each random vector \( b_i(t) \) are independent mean zero Gaussian distributed random variables with variance denoted by \( S_i \), i.e.

\[
P(b_i(t)) \sim N(0, S_i I_{(n+2) \times (n+2)})
\]

for all \( t \geq 0 \). Furthermore, \( b_1(t_1) \) and \( b_1(t_2) \) are independent for \( t_1 \neq t_2 \).

Finally, we let

\[
\tilde{w}_i(t) = (\tilde{d}_i(t), \tilde{a}_i(t)^T, \tilde{c}_i(t))^T
\]
be the perturbed weight vector associated with the $i^{th}$ hidden node and the perturbed output of the $i^{th}$ hidden node is denoted by
\[
\tilde{z}_i(t) = z_i(x_t, \tilde{a}_i(t), \tilde{e}_i(t)).
\]

The update of $w_i$ based on weight noise injection with weight decay can be written as follows:
\[
w_i(t + 1) - w_i(t) = \mu(t)(y_t - f(x_t, \tilde{w}(t)))g_i(x_t, \tilde{w}(t)) - \mu(t)\alpha w_i(t),
\]
(11)
where $\tilde{w}_i(t)$ is a perturbed weight vector and
\[
g_i(x_t, \tilde{w}_i(t)) = \begin{bmatrix}
\tilde{z}_i(t) \\
\tilde{z}_i(t)(1 - \tilde{z}_i(t))
\end{bmatrix},
\]
(12)
$\mu(t) > 0$ is the step size at the $t^{th}$ step, and $\alpha > 0$ is the decay constant.

A. MWN-WD algorithm

If the weight vector is perturbed by multiplicative weight noise, $\tilde{w}_i(t)$ in (11) is given by
\[
\tilde{w}_i(t) = w_i(t) + b_i(t) \otimes w_i(t),
\]
(13)
where $\otimes$ is the elementwise multiplication operator defined as follows:
\[
b_i(t) \otimes w_i(t) = (b_{i1}(t)w_{i1}(t), \ldots, b_{i(n+2)}(t)w_{i(n+2)}(t))^T.
\]
(14)
As in [3], [4], [21], [22], the noise variance $S_b$ is assumed to be a small positive value. The output $f(x_t, \tilde{w}(t))$ and $g_i(x_t, \tilde{w}_i(t))$ in (11) are approximated by
\[
f(x_t, \tilde{w}) \approx f(x_t, w) + \sum_{i=1}^m g_i(x_t, w_i)^T (b_i \otimes w_i),
\]
(15)
and
\[g_i(x_t, \tilde{w}_i) \approx g_i(x_t, w_i) + \nabla_w g_i(x_t, w_i)(b_i \otimes w_i),
\]
(16)
where $\nabla_w g_i(x_t, w_i)$ is given by (6). In (15) and (16) The parentheses ($t$) attached with $w_i(t)$, $\tilde{w}_i(t)$ and $b_i(t)$ are omitted to save space.

Suppose each sample in the dataset $\mathcal{D}$ has equal probability to be selected. By (11), (15) and (16), the conditional expectation of $w_i(t+1)$ over all random vectors $b_1(t), \ldots, b_m(t)$ on $w(t)$ is given by:
\[
E[w_i(t+1)|w(t)] = w_i(t) + \mu(t)h_i(w(t)),
\]
(17)
where $h_i(w(t))$
\[
= \frac{1}{N} \sum_{k=1}^N (y_k - f(x_k, w(t)))g_i(x_k, w(t))
\]
\[
- \frac{S_b}{N} \sum_{k=1}^N \nabla_w g_i(x_k, w_i(t))v_i(x_k, w_i(t)) - \alpha w_i(t)
\]
(18)
and
\[
v_i(x_k, w_i(t)) = w_i(t) \otimes w_i(t) \otimes g_i(x_k, w_i(t)).
\]
(19)
We have showed that [16]
\[
h_i(w(t)) = -\nabla_w V(w(t)),
\]
(20)
where
\[
V(w) = \frac{1}{2} \left( \frac{1}{N} \sum_{k=1}^N (y_k - f(x_k, w))^2 + \frac{S_b}{N} \sum_{k=1}^N \sum_{i=1}^m (w_i^T g_i(x_k, w_i))^2 \right)
\]
\[
- \frac{S_b}{N} \sum_{k=1}^N \int u(x_k, w)dw + \frac{\alpha}{2}w^Tw
\]
(21)
and
\[
u(x_k, w) = \begin{bmatrix}
w_1 \otimes g_1(x_k, w_1) \otimes g_1(x_k, w_1) \\
w_2 \otimes g_2(x_k, w_2) \otimes g_2(x_k, w_2) \\
\vdots \\
w_m \otimes g_m(x_k, w_m) \otimes g_m(x_k, w_m)
\end{bmatrix}
\]
(22)
The 3rd term in (21) is a line integral. In the later section, we will show that $w(t)$ generated by (11) and (13) converges to a local minimum of this objective function $V(w)$.

B. AWN-WD algorithm

The definition of AWN-WD algorithm is similar to MWN-WD algorithm. The update equation is based on (11) but the perturbed weight vector is now given by
\[
\tilde{w}_i(t) = w_i(t) + b_i(t).
\]
(23)
Thus, the output $f(x_t, \tilde{w}(t))$ and $g_i(x_t, \tilde{w}_i(t))$ in (11) are approximated by
\[
f(x_t, \tilde{w}) \approx f(x_t, w) + \sum_{i=1}^m g_i(x_t, w_i)^T b_i,
\]
(24)
and
\[
g_i(x_t, \tilde{w}_i) \approx g_i(x_t, w_i) + \nabla_w g_i(x_t, w_i)b_i,
\]
(25)
where $\nabla_w g_i(x_t, w_i)$ is given by (6). Again, the parentheses ($t$) attached with $w_i(t)$, $\tilde{w}_i(t)$ and $b_i(t)$ are omitted to save space.

Note that this condition is not required in the convergence proof presented in Section 4.
Suppose each sample in the dataset $\mathcal{D}$ has equal probability to be selected. By (11), (24) and (25), the conditional expectation of $\mathbf{w}_t(t+1)$ over all random vectors $\mathbf{b}_1(t), \cdots, \mathbf{b}_m(t)$ on $\mathbf{w}(t)$ is given by:

$$E[\mathbf{w}_t(t+1)|\mathbf{w}(t)] = \mathbf{w}_t(t) + \mu(t)\mathbf{h}'(\mathbf{w}(t)), \quad (26)$$

where $\mathbf{h}'(\mathbf{w}(t))$

$$
\begin{align*}
\mathbf{h}'(\mathbf{w}(t)) &= \frac{1}{N} \sum_{k=1}^{N} (y_k - f(\mathbf{x}_k, \mathbf{w}(t)))g_i(\mathbf{x}_k, \mathbf{w}(t)) \\
&- \frac{S_b}{N} \sum_{k=1}^{N} \nabla_{\mathbf{w}_i} g_i(\mathbf{x}_i, \mathbf{w}_i)g_i(\mathbf{x}_k, \mathbf{w}(t)) \\
&- \alpha \nabla_{\mathbf{w}_i} \mathbf{w}_i(t).
\end{align*}

We have shown that [16]

$$\mathbf{h}'(\mathbf{w}(t)) = -\nabla_{\mathbf{w}} V'(\mathbf{w}(t)), \quad (28)$$

where

$$
V'(\mathbf{w}) = \frac{1}{2} \left\{ \frac{1}{N} \sum_{k=1}^{N} (y_k - f(\mathbf{x}_k, \mathbf{w}))^2 \\
+ \frac{S_b}{N} \sum_{k=1}^{N} \sum_{i=1}^{m} \|g_i(\mathbf{x}_k, \mathbf{w})\|_2^2 \\
+ \frac{\alpha}{2} \|\mathbf{w}\|_2^2 \right\}. \quad (29)
$$

In the later section, we will show that $\mathbf{w}(t)$ generated by (11) and (23) converges to a local minimum of this objective function $V'(\mathbf{w})$.

IV. CONVERGENCE OF MWN-WD ALGORITHM

The convergence proof is conducted by the following steps. First, we consider the update of the output weight vector $\mathbf{d}(t)$ and apply Doob’s Martingale Convergence Theorem to show that $\lim_{t \to \infty} \|\mathbf{d}(t)\|_2$ exists. As the existence of $\lim_{t \to \infty} \|\mathbf{d}(t)\|_2$ cannot imply the existence of $\lim_{t \to \infty} \mathbf{d}(t)$, we consider the update of the elements in $\mathbf{d}(t)$ and apply Doob’s Martingale Convergence Theorem to show the existence of their limits.

The existence of $\lim_{t \to \infty} \|\mathbf{d}(t)\|_2$ and $\lim_{t \to \infty} d_i(t)$ (for all $i = 1, \cdots, m$), together with the Doob’s Martingale Convergence Theorem are then applied to show the existence of $\lim_{t \to \infty} \mathbf{a}_i(t)$ and $\lim_{t \to \infty} c_i(t)$. Finally, we conclude that $\lim_{t \to \infty} \mathbf{w}(t)$ exists.

Here and after, we let $\mathbf{b}_d(t)$ be the random vector associated with the output vector $\mathbf{d}$. That is,

$$\mathbf{b}_d(t) = (b_{11}(t), b_{21}(t), \cdots, b_{m1}(t))^T, \quad (30)$$

where $b_{11}(t)$ is the first element in $\mathbf{b}_1(t)$. Besides, we use the notation $E_d[|\mathbf{w}(t)|]$ denoting the conditional expectation that is taken over the random vector $\mathbf{b}_d(t)$ only.

A. Existence of $\lim_{t \to \infty} \|\mathbf{d}(t)\|_2$

By (1), (11) and (12), the update of $d_i(t)$ can be expressed as follows:

$$d_i(t + 1) - d_i(t) = \mu(t) \left\{ (y_i - \mathbf{d}_i^T(t)\mathbf{z}(t))\mathbf{z}_i(t) - \alpha d_i(t) \right\}. \quad (31)$$

In vector-matrix form,

$$\mathbf{d}(t + 1) - \mathbf{d}(t) = \mu(t) \left\{ (y_i - \mathbf{d}_i^T(t)\mathbf{z}(t))\mathbf{z}(t) - \alpha \mathbf{d}(t) \right\}. \quad (32)$$

Based on (32), the boundedness of $E[\|\mathbf{d}(t)\|_2]$ and the existence of $\lim_{t \to \infty} \|\mathbf{d}(t)\|_2$ can be stated in the following lemma.

**Lemma 1:** For the algorithm based on (11) and (13), if $0 < \mu(t)(\alpha - \sqrt{S_b}m) < 1$ for all $t \geq 0$, then $\lim_{t \to \infty} \|\mathbf{d}(t)\|_2$ exists and is finite with probability one.

**Proof:** We rewrite the update of $\mathbf{d}(t)$ given by (32) as follows:

$$
\begin{align*}
\mathbf{d}(t + 1) &= (1 - \mu(t)\alpha)\mathbf{d}(t) + \mu(t)g_i\mathbf{z}(t) \\
&- \mu(t)\mathbf{z}(t)\mathbf{z}_i^T(t)\mathbf{d}(t) \\
&- \mu(t)\mathbf{z}(t)\mathbf{z}_i^T(t) (\mathbf{b}_d(t) \otimes \mathbf{d}(t)).
\end{align*}
$$

Here, we let

$$\mathbf{B}(t) = (1 - \mu(t)\alpha)I_{m \times m} - \mu(t)\mathbf{z}(t)\mathbf{z}_i^T(t). \quad (34)$$

Equation (33) can be written as follows:

$$
\begin{align*}
\mathbf{d}(t + 1) &= \mathbf{B}(t)\mathbf{d}(t) + \mu(t)g_i\mathbf{z}(t) \\
&- \mu(t)\mathbf{z}(t)\mathbf{z}_i^T(t) (\mathbf{b}_d(t) \otimes \mathbf{d}(t)).
\end{align*}
$$

Since the elements in $\mathbf{b}_d(t)$ are identical and independent mean zero Gaussian random variables with variance $S_b$,

$$
\begin{align*}
E_d \left[ (\mathbf{d}_d(t) \otimes \mathbf{d}(t)) (\mathbf{b}_d(t) \otimes \mathbf{d}(t))^T | \mathbf{w}(t) \right] &= S_b \begin{bmatrix} d_1(t)^2 & 0 & \cdots & 0 \\
0 & d_2(t)^2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & d_m(t)^2 \end{bmatrix} \\
&= S_b \mathbf{d}(t)\mathbf{d}^T(t).
\end{align*}
$$

Hence,

$$
\begin{align*}
E_d[\|\mathbf{z}(t)\mathbf{z}_i^T(t) (\mathbf{b}_d(t) \otimes \mathbf{d}(t))\|^2_2 | \mathbf{w}(t)] &= S_b \mathbf{r} \left( \mathbf{z}(t)\mathbf{z}_i^T(t) \right)^2 \mathbf{d}(t)\mathbf{d}^T(t) \\
&= S_b \|\mathbf{z}(t)\mathbf{z}_i^T(t)\|^2_2 \mathbf{d}(t)\mathbf{d}^T(t).
\end{align*}
$$

where $\mathbf{r}$ is the trace operator.

Given $\mathbf{w}(t)$, the expectation of the $\|\mathbf{d}(t + 1)\|^2_2$ over the random vector $\mathbf{b}_d(t)$ is given by

$$
\begin{align*}
E_d[\|\mathbf{d}(t + 1)\|^2_2 | \mathbf{w}(t)] &= \|\mathbf{B}(t)\mathbf{d}(t) + \mu(t)g_i\mathbf{z}(t)\|^2_2 + \mu(t)^2 S_b \mathbf{d}(t)\mathbf{d}^T(t) \\
&\leq \|\mathbf{B}(t)\mathbf{d}(t) + \mu(t)g_i\mathbf{z}(t)\|^2_2 + \mu(t)^2 S_b m^2 \|\mathbf{d}(t)\|^2_2 \\
&\leq \left( \|\mathbf{B}(t)\mathbf{d}(t) + \mu(t)g_i\mathbf{z}(t)\|_2^2 + \mu(t)\sqrt{S_b m}\|\mathbf{d}(t)\|_2 \right)^2. \quad (37)
\end{align*}
$$
The last inequality is based on the fact that the eigenvalues of \( \tilde{z}(t)\tilde{z}^T(t) \) are 0 and \( \sum_{i=1}^{m} \tilde{z}_i(t)^2 \).

Further by Jensen Inequality that
\[
E_d[\|\mathbf{d}(t+1)\|_2^2|w(t)] \leq (E_d[\|\mathbf{d}(t+1)\|_2^2]^2)^{1/2},
\]

and then by Triangle Inequality,
\[
E_d[\|\mathbf{d}(t+1)\|_2^2|w(t)] \\
\leq (1 - \mu(t)\alpha - \sqrt{S_{Tm}}\|\mathbf{d}(t)\|_2 + \mu(t)\|\mathbf{y}_{\bar{z}}(t)\|_2) \\
\leq (1 - \mu(t)\alpha)\|\mathbf{d}(t)\|_2 + \mu(t)\|\mathbf{y}_{\bar{z}}(t)\|_2.
\]

The last inequality is based on the fact that the eigenvalues of \( \tilde{z}(t)\tilde{z}^T(t) \) are 0 and \( \sum_{i=1}^{m} \tilde{z}_i(t)^2 \). Hence, the eigenvalue of \( \mathbf{B}(t) \) must be less than or equal to \((1 - \mu(t)\alpha)\).

To save space, we let
\[
\alpha' = \alpha - \sqrt{S_{Tm}}.
\]

As \( y_i \) is generally bounded for all \( t \geq 0 \), \( \|y_\tilde{z}(t)\|_2 \) is bounded by a positive constant. Let it be \( \kappa_d \). Thus,
\[
E_d[\|\mathbf{d}(t+1)\|_2|w(t)] \leq (1 - \mu(t)\alpha')\|\mathbf{d}(t)\|_2 + \mu(t)\kappa_d.
\]

As the right hand side of (39) is independent of the random vector \( \mathbf{b}(t) \),
\[
E[\|\mathbf{d}(t+1)\|_2|w(t)] \leq (1 - \mu(t)\alpha')\|\mathbf{d}(t)\|_2 + \mu(t)\kappa_d.
\]

Equivalently,
\[
E[\|\mathbf{d}(t+1)\|_2^2|w(t)] \leq (1 - \mu(t)\alpha')\|\mathbf{d}(t)\|_2^2 + \mu(t)\kappa_d\alpha'.
\]

Let
\[
\beta(t) = \|\mathbf{d}(t)\|_2 - \frac{\kappa_d}{\alpha'}. \tag{42}
\]

It is clear that \( \{\beta(t)\}_{t \geq 0} \) is a supermartingale and
\[
E[\|\beta(t)\|] \leq E[\|\beta(t-1)\|] \leq \cdots \leq E[\|\beta(0)\|]. \tag{43}
\]

By Doob’s Martingale Convergence Theorem, \( \lim_{t \to \infty} \beta(t) \) exists and is finite with probability one. Then from (42), \( \lim_{t \to \infty} \|\mathbf{d}(t)\|_2 \) exists and is finite with probability one. The proof is completed. Q.E.D.

Lemma 1 is crucial for the following proofs on the existence of \( \lim_{t \to \infty} \mathbf{d}(t) \), \( \lim_{t \to \infty} \mathbf{a}_i(t) \) and \( \lim_{t \to \infty} \mathbf{c}_i(t) \).

The idea can be described in the rest of this subsection.

As \( \lim_{t \to \infty} \|\mathbf{d}(t)\|_2 \) exists with probability one,
\[
\lim_{t \to \infty} E[\|\mathbf{d}(t+1)\|_2|w(t)] = \lim_{t \to \infty} \|\mathbf{d}(t+1)\|_2.
\]

Thus, for any small positive \( \epsilon \), there exists a time \( t^* \) such that
\[
P(\|\mathbf{d}(t)\|_2 < \kappa_d/\alpha' + \epsilon) = 0 \tag{44}
\]

or equivalently
\[
\|\mathbf{d}(t)\|_2 < \kappa_d/\alpha' + \epsilon \tag{45}
\]

is with probability one for all \( t \geq t^* \).

Making use of (44), we can therefore derive inequalities bounding \( E[\mathbf{d}_i(t+1)|w(t)], \mathbf{E}[\mathbf{a}_{ij}(t+1)|w(t)] \) and \( E[\mathbf{c}_{ij}(t+1)|w(t)] \) for \( t \geq t^* \), for \( i = 1, 2, \ldots, m \) and \( j = 1, 2, \ldots, n \).

Then, we can define supermartingales for each \( \mathbf{d}_i(t), \mathbf{a}_{ij}(t) \) and \( \mathbf{c}_{ij}(t) \) respectively and show the existence of their limits by Doob’s Martingale Convergence Theorem.

B. Existence of \( \lim_{t \to \infty} \mathbf{d}(t) \)

To show the existence of the limit of \( \mathbf{d}(t) \), we consider (31) and \( t \geq t^* \).

\[
E_d[\mathbf{d}_i(t+1)|w(t)] = (1 - \mu(t)\alpha)\mathbf{d}_i(t) + \mu(t)\mathbf{y}z(t)\tilde{z}_i(t) \\
\leq (1 - \mu(t)\alpha)\mathbf{d}_i(t) + \mu(t)\kappa_d + \mu(t)\|\mathbf{d}(t)\|_2. \tag{46}
\]

As a result of (44),
\[
E_d[\mathbf{d}_i(t+1)|w(t)] \leq (1 - \mu(t)\alpha)\mathbf{d}_i(t) + \mu(t)\kappa_d + \mu(t)(\kappa_d/\alpha' + \epsilon). \tag{47}
\]

Since the right hand side is independent of other random variables in \( \mathbf{b}(t) \), we can write that
\[
E[\mathbf{d}_i(t+1)|w(t)] \leq (1 - \mu(t)\alpha)\mathbf{d}_i(t) + \mu(t)\kappa_d + \mu(t)(\kappa_d/\alpha' + \epsilon) \tag{48}
\]

for all \( t \geq t^* \).

Hence, we can define a random process \( \{\gamma(t)\}_{t \geq 0} \) in which
\[
\gamma(t) = \mathbf{d}_i(t + t^*) - \frac{\kappa_d + \kappa_d/\alpha' + \epsilon}{\alpha} \tag{49}
\]

and clearly
\[
E[\gamma(t)\|w(t^*)\|] \leq \cdots \leq E[\|\gamma(0)\|w(t^*)].
\]

Therefore, by Doob’s Martingale Convergence Theorem, \( \lim_{t \to \infty} \gamma(t) \) exists and is finite with probability one. We can conclude that \( \lim_{t \to \infty} \mathbf{d}_i(t) \) exists and is finite with probability one. As the same procedure applies to all \( i = 1, 2, \ldots, m \), we can have the following lemma.

Lemma 2: For the algorithm based on (11) and (13), if
\[
0 < \mu(t)(\alpha - \sqrt{S_{Tm}}) < 1 \text{ for all } t \geq 0,
\]
then \( \lim_{t \to \infty} \mathbf{d}(t) \) exists and its elements are finite with probability one.

C. Existence of \( \lim_{t \to \infty} \mathbf{a}_i(t) \)

The proof of the existence of \( \lim_{t \to \infty} \mathbf{a}_i(t) \) is similar to that of the proof of Lemma 2. By (1), (11) and (12), the update of \( \mathbf{a}_i(t) \) can be expressed as follows:

\[
\mathbf{a}_i(t+1) = (1 - \mu(t)\alpha)\mathbf{a}_i(t) + \mu(t)\mathbf{y}_i\tilde{z}_i(t)(1 - \tilde{z}_i(t))\mathbf{d}_i(t)\mathbf{x}_i \\
- \mu(t)\tilde{z}_i(t)(1 - \tilde{z}_i(t))\tilde{d}_i(t)x_i\tilde{z}^T(t)\mathbf{d}(t). \tag{50}
\]

Note from (13) and (30) that
\[
\tilde{d}_i(t) = \mathbf{d}_i(t) + b_{11}(t)d_i(t) \tag{51}
\]

and
\[
\tilde{d}_i(t)\tilde{d}(t) = (\mathbf{d}_i(t) + b_{11}(t)d_i(t))(\mathbf{b}_d(t) \otimes \mathbf{d}(t)). \tag{52}
\]

Let us consider the \( j^{th} \) element in \( \mathbf{a}_i(t) \).

\[
\mathbf{a}_{ij}(t+1) = (1 - \mu(t)\alpha)\mathbf{a}_{ij}(t) + \mu(t)\mathbf{y}_i\tilde{z}_i(t)(1 - \tilde{z}_i(t))\tilde{d}_i(t)x_{ij} \\
- \mu(t)\tilde{z}_i(t)(1 - \tilde{z}_i(t))\tilde{d}_i(t)x_{ij}\tilde{z}^T(t)\mathbf{d}(t). \tag{53}
\]
Lemma 3: For the algorithm based on (11) and (13), if $0 < \mu(t)(\alpha - \sqrt{S_{0}m}) < 1$ for all $t \geq 0$, then for all $i = 1, 2, \ldots, m$, $\lim_{t \to \infty} a_{i}(t)$ exists and its elements are finite with probability one.

Proof: Given $w(t)$ and taking expectation of (53) over $b_{d}(t)$, $E_{d}[a_{ij}(t + 1)|w(t)]$ can be expressed as follows:

$$E_{d}[a_{ij}(t + 1)|w(t)] = (1 - \mu(t)\alpha)a_{ij}(t) + \mu(t)y_{i}\bar{v}_{i}(t)d_{i}(t) - \mu(t)v_{i}(t)d_{i}(t)\bar{z}^{T}(t)(d(t) + S_{0}d_{i}(t)), \quad (54)$$

where $v_{i}(t) = \bar{z}_{i}(t)(1 - \bar{z}_{i}(t))x_{ij}$ and $e_{i} = (0, \ldots, 0, 1, 0, \ldots, 0)^{T}$. \quad (56)

Again, for $t \geq t^{*}$, by (44) and (54),

$$E_{d}[a_{ij}(t + 1)|w(t)] \leq (1 - \mu(t)\alpha)a_{ij}(t) + \mu(t)\kappa_{a}. \quad (59)$$

As the right hand side is independent of other random variables,

$$E[a_{ij}(t + 1)|w(t)] \leq (1 - \mu(t)\alpha)a_{ij}(t) + \mu(t)\kappa_{a}. \quad (60)$$

Similar to Lemma 2, we can define a random process $\{\xi(t)\}_{t \geq 0}$ as follows:

$$\xi(t) = a_{ij}(t + t^{*}) - \frac{\kappa_{a}}{\alpha} \quad (61)$$

for all $t \geq t^{*}$ and clearly

$$E[\xi(t)||w(t^{*})] \leq \cdots \leq E[\xi(0)||w(t^{*})].$$

Therefore, by Doob’s Martingale Convergence Theorem, $\lim_{t \to \infty} \xi(t)$ exists and is finite with probability one. We can conclude that $\lim_{t \to \infty} a_{ij}(t)$ exists and is finite with probability one. Thus, for all $i = 1, \ldots, m$, $\lim_{t \to \infty} a_{i}(t)$ exists and its elements are finite with probability one.

D. Existence of $\lim_{t \to \infty} c_{i}(t)$

By (1), (11) and (12), the update of $c_{i}(t)$ can be expressed as follows:

$$c_{i}(t + 1) = (1 - \mu(t)\alpha)c_{i}(t) + \mu(t)y_{i}\bar{z}_{i}(t)(1 - \bar{z}_{i}(t))d_{i}(t) - \mu(t)\bar{z}_{i}(t)(1 - \bar{z}_{i}(t))\bar{d}_{i}(t)\bar{z}^{T}(t)d_{i}(t). \quad (62)$$

Suppose, we define two augmented vectors as that

$$x_{i}' = \begin{bmatrix} x_{i} \\ 1 \end{bmatrix} \quad \text{and} \quad a_{i}'(t) = \begin{bmatrix} a_{i}(t) \\ c_{i}(t) \end{bmatrix}.$$ 

We can combine (50) and (62) together and come up with the following update equation.

$$a_{i}'(t + 1) = (1 - \mu(t)\alpha)a_{i}(t) + \mu(t)y_{i}\bar{z}_{i}(t)(1 - \bar{z}_{i}(t))d_{i}(t)x_{i}' - \mu(t)\bar{z}_{i}(t)(1 - \bar{z}_{i}(t))\bar{d}_{i}(t)x_{i}'\bar{z}^{T}(t)d_{i}(t). \quad (63)$$

Repeating the same steps as the proof of Lemma 3, we can conclude the existence of $\lim_{t \to \infty} a_{i}'(t)$ and thus $\lim_{t \to \infty} c_{i}(t)$ is with probability one.

Lemma 4: For the algorithm based on (11) and (13), if $0 < \mu(t)(\alpha - \sqrt{S_{0}m}) < 1$ for all $t \geq 0$, then for all $i = 1, 2, \ldots, m$, $\lim_{t \to \infty} c_{i}(t)$ exists and is finite with probability one.

E. Existence of $\lim_{t \to \infty} w(t)$

As a direct implication from Lemma 2-4, we state without proof the following theorem for the weight vector $w(t)$.

Theorem 1: For the algorithm based on (11) and (13), if $0 < \mu(t)(\alpha - \sqrt{S_{0}m}) < 1$ for all $t \geq 0$, then $\lim_{t \to \infty} w(t)$ exists and its elements are finite with probability one.

Let us define a bounded region $\Omega_{w}(w^{*})$ which is centered at $w^{*}$ and $\|w - w^{*}\| \leq \epsilon$ for all $w \in \Omega_{w}(w^{*})$. Theorem 1 implies that for any arbitrary small positive $\epsilon$, there must exist a bounded region $\Omega_{w}(w^{*})$ and a time $t(w^{*})$, such that for all $t \geq t(w^{*})$

$$P(w(t) \in \Omega_{w}(w^{*})) = 1. \quad (64)$$

This final equation is very useful in the subsequent analysis.

V. CONVERGENCE OF AWN-WD ALGORITHM

Basically, the steps of proof for the AWN-WD algorithm are the same as the proof for the MWN-WD algorithm. The only difference is in the definition of $\bar{w}$. Owing to save space, we skip some of the proofs in this section. Only the existence of $\lim_{t \to \infty} \|d(t)\|_{2}$ is proved, as it is the key step to show that noise variance $S_{0}$ does not affect the convergence.

Theorem 2: For the algorithm based on (11) and (23), if $0 < \mu(t)\alpha < 1$ for all $t \geq 0$, then $\lim_{t \to \infty} w(t)$ exists and its elements are finite with probability one.

Proof: Replace $b_{d}(t) \odot d(t)$ in (33) and (35) by $b_{d}(t)$, the update of $d(t)$ is given by

$$d(t + 1) = B(t)d(t) + \mu(t)y_{i}\bar{z}_{i}(t) - \mu(t)\bar{z}_{i}(t)\bar{z}^{T}(t)b_{d}(t). \quad (65)$$

Note that $\bar{z}(t)$ in (65) is now depended on $w(t) + b(t)$ instead of $w(t) + b(t) \odot \bar{w}(t)$.

Given $w(t)$, the expectation of the $\|d(t + 1)\|_{2}^{2}$ over the random vector $b_{d}(t)$ is then given by

$$E_{d}[\|d(t + 1)\|_{2}^{2}|w(t)] = \|B(t)d(t) + \mu(t)y_{i}\bar{z}_{i}(t)\|_{2}^{2} + \mu(t)^{2}E_{d}[b_{d}(t)\bar{z}(t)\bar{z}^{T}(t)]b_{d}(t)|w(t)|. \quad (66)$$
As,
\[
E_d[\|b_d(t)\| \hat{\omega}(t) \hat{\omega}^T(t)]^2 \| b_d(t) \| w(t)]
= \text{Tr} \left\{ E_d[\hat{\omega}(t) \hat{\omega}^T(t)]^2 \| b_d(t) \| b_d^T(t) \| w(t)] \right\}
= S_b \| \hat{\omega}(t) \|^2 \text{Tr} \left\{ \hat{\omega}(t) \hat{\omega}^T(t) \right\}
= S_b \| \hat{\omega}(t) \|^2 \| \hat{\omega}(t) \| \hat{\omega}^T(t)
\leq S_b \| \hat{\omega}(t) \|^2.
\tag{67}
\]

The last inequality is due to the fact that the elements in \( \hat{\omega}(t) \) are all in between 0 and 1. \( \| \hat{\omega}(t) \| \) is bounded by a positive constant. We can then let \( \cdot \)
make three more assumptions on the noise variance without proof.

MWN-WD algorithm will be presented in this section. The both the MWN-WD algorithm and AWN-WD algorithm are locations are at local minimum. As the steps of proofs for
functions. Therefore, it is necessarily to show that their proof for MWN-WD algorithm, their proofs are skipped.

While Theorem 1 and Theorem 2 state the existence of \( w(t) \) when \( t \to \infty \), they could not imply that their limits are located in local minimum of the corresponding objective functions. Therefore, it is necessarily to show that their locations are at local minimum. As the steps of proofs for both the MWN-WD algorithm and AWN-WD algorithm are the same, only the theorem and the proof regarding the MWN-WD algorithm will be presented in this section. The theorem regarding the AWN-WD algorithm will be stated without proof.

Before proceed to the statement of theorem, we need to make three more assumptions on the noise variance \( S_b \) and the step size \( \mu(t) \) as follows:
\[ S_b \ll 1, \tag{70} \]
\[ \mu(t) \to 0 \quad \text{for all } t \geq 0, \tag{71} \]
\[ \sum_{t=1}^{\infty} \mu(t) = \infty \quad \text{for all } t \geq 0. \tag{72} \]

The first assumption on \( S_b \) a common assumption made by other researchers [3], [4], [21], [22]. With this assumption, the approximations for \( f(x_t, w(t)) \) (15) and \( g_t(x_t, w(s)) \) (16) will be making sense. The objective function (21) is in simple close form.

The second assumption is owing to simplify the approximation of \( V(w(t+1)) - V(w(t)) \) by ignoring higher order terms containing \( \mu(t)^2 \). The third assumption is to ensure that \( \sum_{t=1}^{\infty} \mu(t) \| \nabla_w V(w(t)) \|^2 \) diverges if \( \| \nabla_w V(w(t)) \|^2 \) does not converge.

With the assumptions on (70), (71) and (72), the property of \( \nabla_w V(w^*) \) can then be stated as the following theorem.

\textbf{Theorem 3:} For the algorithm based on (11) and (13), if (i) \( (\alpha - \sqrt{S_b}\kappa) > 0 \), (ii) \( S_b \ll 1 \), (iii) \( \mu(t) \to 0 \) for all \( t \geq 0 \) and (iv) \( \sum_{t=1}^{\infty} \mu(t) = \infty \) for any \( t \geq 0 \), then \( w(t) \) converges to the location in which
\[ \nabla_w V(w^*) = \lim_{t \to \infty} \nabla_w V(w(t)) = 0, \tag{73} \]
where \( V(w) \) is a scalar function given by (21).

\textbf{Proof:} First of all, as \( h_1(w) \) is the gradient vector of \( V(w) \) with differentiable functional elements,
\[ V(w) \text{ is differentiable for all } w \quad \text{and} \quad \nabla_w V(w) \text{ is differentiable for all } w. \tag{74} \]

From Condition (i) and (iii), \( \lim_{t \to \infty} \nabla_w V(w(t)) \) and \( \nabla_w V(w(t)) \) exist with finite elements. By virtue of Theorem 1 and (74), \( \lim_{t \to \infty} \nabla_w V(w(t)) \) and \( \nabla_w V(w(t)) \) exist with probability one.

It implies that for any arbitrary \( \epsilon_V \), there exists a time \( t_V \), such that
\[ P(|V(w^*) - V(w(t))| \leq \epsilon_V) = 1 \tag{76} \]
for all \( t \geq t_V \).

By Taylor expansion of \( V(w(t+1)) \) about \( w(t) \),
\[ V(w(t+1)) = V(w(t)) + (\nabla_w V(w(t)))^T \Delta w(t), \tag{77} \]
where
\[ \Delta w(t) = \begin{pmatrix} w(t+1) - w(t) \\ \mu(t) (y_t - \tilde{d}(t) \tilde{z}(t)) g(x_t, w(t)) \end{pmatrix}, \tag{78} \]
where
\[ g = (g_1^T, g_2^T, \ldots, g_m^T)^T. \]
By (17) and (18),
\[ E[V(w(t+1))|w(t)] = V(w(t)) - \mu(t) \| \nabla_w V(w(t)) \|^2. \tag{79} \]
Hence,

\[ E[V(w^*) - V(w(t))] = -\sum_{\tau=tV}^{\infty} \mu(\tau)E \left[ \|\nabla_w V(w(\tau))\|^2 \right] w(tV). \]  

(80)

From (76), the right hand side of (80) must be with probability one smaller than \( \epsilon V \).

\[ \sum_{\tau=tV}^{\infty} \mu(t)E \left[ \|\nabla_w V(w(t))\|^2 \right] w(tV) \leq \epsilon V. \]  

(81)

By virtue of (72) and the inequality that \( E[||q||] \leq (E[||q||^2])^{1/2} \), one can prove by contradiction that

\[ E[\|\nabla_w V(w^*)\|^2] w(tV) = 0. \]  

(82)

As \( \|\nabla_w V(w^*)\| \) is non-negative,

\[ \|\nabla_w V(w^*)\| = 0 \]

Hence

\[ \nabla_w V(w^*) = 0. \]  

(83)

The proof is completed. Q.E.D.

For the AWN-WD algorithm, we let \( w^* \) be the limit \( \lim_{t \to \infty} w(t) \). The property of \( \nabla_w V(w^*) \) is stated without proof as the following theorem.

**Theorem 4:** For the algorithm based on (11) and (23), if (i) \( \alpha > 0 \), (ii) \( S_0 \ll 1 \), (iii) \( \mu(t) \to 0 \) for all \( t \geq 0 \) and (iv) \( \sum_{\tau=0}^{\infty} h(t) = \infty \) for any \( t \geq 0 \), then \( w(t) \) converges to the solution of the objective function given by (29).

**VII. DISCUSSIONS**

The condition on the step size in Theorem 1 and Theorem 2 is much more relax than the conditions for back-propagation algorithm. For instance in [29], the step size \( \mu(t) \) must be a decreasing function. In our proof, this condition is not necessary.

If \( 0 < \mu(t)(\alpha - S_0 m) < 1 \), the MWN-WD algorithm converges with probability one. If \( 0 < \mu(t) \leq \alpha < 1 \), the AWN-WD algorithm converges with probability one. It implies that the step size \( \mu(t) \) can be defined in many ways, including

\[ \mu(t) = \varphi(t), \]

\[ \mu(t) = \varphi(\sin(\omega t) + 1), \]

\[ \mu(t) = \varphi/t, \]

\[ \mu(t) = \text{small positive random number}, \]

where \( 0 < \varphi \leq (\alpha - \sqrt{S_0 m})^{-1} \) and \( \omega \) is a positive constant.

In view of the fact that the algorithm based on (11) and (13) reduces to the original weight decay training algorithm (9) when \( S_0 = 0 \), Theorem 1 implies that the algorithm (9) [29] converges with probability one even if the step size is constant. In contrast to other results like in [29], the step size has to be decreasing with time.

**VIII. CONCLUSION**

In this paper, we have presented two training algorithms based on on-line combining weight noise injection and weight decay. Their algorithms and objective functions have been presented. Their convergence are proved. Apart from the convergence proof, we have also showed that the locations the weight vectors converge to the local minimum of the corresponding objective functions. Finally, a few aspects regarding the conditions on the step size and the extension of the theorems to the convergence of weight decay algorithm are discussed.

**Acknowledgement**

The research work reported in this paper is supported in part by Taiwan National Science Council (NSC) Research Grant 97-2221-E-005-050 and 98-2221-E-005-048.

**REFERENCES**


[16] Ho K., C.S. Leung and J. Sum, Objective functions of online weight noise injection training algorithms for MLP, in submission.


H. Takase, H. Kita and T. Hayashi, A study on the simple penalty term to the error function from the viewpoint of fault tolearnt training, 2004.


**APPENDIX**

The content of this appendix is adapted from Chapter 4, Theorem 4.2, in [10]. A stochastic process \( \{ \eta_t, t \geq 1 \} \) is a supermartingale if \( E[|\eta_t|] < \infty \) for all \( t \) and

\[
E[\eta_{t+1}|\eta_t, \eta_{t-1}, \cdots, \eta_1] \leq \eta_t.
\]  

(85)

**Lemma 5 (Doob’s Martingale Convergence Theorem):** If \( \{ \eta_t, t \geq 1 \} \) is a supermartingale such that for some \( \phi < \infty \) and \( E[|\eta_t|] \leq \phi \) for all \( t \), then \( \lim_{t \to \infty} \eta_t \) exists and is finite with probability one.