Convergence analyses on on-line weight noise injection-based training algorithms for MLPs

John Sum, Senior Member, IEEE, Chi-Sing Leung, Member, IEEE, 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 its convergence behavior. Therefore, we present in this paper the convergence proofs of two algorithms based on weight noise injection for MLPs. One is based on combining injecting multiplicative weight noise and weight decay (AWN-WD) during training. The other is based on combining injecting additive weight noise and weight decay (AWN-WD) during training. Let $\alpha$ be the weight decay constant and $S_0$ be the variance of the noise injected. It is showed that the convergence of MWN-WD algorithm is with probability one if $\alpha > 0$ and $S_0 < 1$. While the convergence of the AWN-WD algorithm is with probability one only if $\alpha > 0$. By computer simulations on (i) a simple mapping problem, (ii) the Mackey-Glass time series prediction problem and (iii) a nonlinear time series prediction problem, we find that the on-line learning algorithm based on purely multiplicative weight noise injection does not converge. Moreover, the MLPs generated by combining weight noise injection and weight decay during training show better performances than their pure weight noise injection counterparts.

I. INTRODUCTION

To improve the fault tolerance of a multilayer perceptron (MLP), Murray & Edward [28], [29], [15] modified the conventional back-propagation 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 [21] have also found, by simulation, that the generalization of a RNN can be improved and the algorithm converges faster than the conventional RTRL does. While on-line weight noise injection training algorithms have demonstrated, by simulations, 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 [14], [29], [21] have only provided preliminary analyses on the effect of the prediction error of a neural network that is corrupted by weight noise (refer to Section II.C in [14], [29] for the analysis for MLPs and Section 3 in [21] for the analysis for RNNs). G.An in [1] has attempted to these learning algorithm applying to MLPs. But the author failed to provide the convergence proofs for these algorithms$^1$.

For multiplicative weight noise case, as commented in our recent paper [20] that the objective function$^2$ derived in [1] is essentially the prediction error of a MLP if it is corrupted by multiplicative weight noise. It is not the corresponding objective function of the on-line multiplicative weight noise injection training algorithm.

Even though some other analyses have been done regarding the effect of weight noise on a neural network [2], [3], [4], [5], [6], [14], [31], they worked on the prediction error of a neural network if it is corrupted by weight noise. None of them worked on the convergence proof, nor the objective function of the weight noise injection-based training algorithms.

Until recently, we have showed in [18], [19] that the convergence of injecting weight noise during training a RBF is with probability one. The objective function of injecting multiplicative weight noise (or additive weight noise) during training a RBF 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. For the weight noise injection-based algorithms for MLPs, only the objective functions have recently been derived [20].

After all, for almost two decades, the convergence proofs of these weight noise injection-based algorithms for MLPs have yet been accomplished. Therefore, the primary focus of this paper is to analyze the convergence of these weight noise injection-based algorithms for MLPs. Specifically, we analyze an algorithm based on combining multiplicative weight noise injection and weight decay during training (a.k.a. MWN-WD), and an algorithm based on combining additive weight noise injection and weight decay during training (a.k.a. AWN-WD). The main theorem we applied is the classical Doob’s Martingale Convergence Theorem [13], [10]. Computer simulations based on three datasets: (1) an artificial 2D mapping, (2) the Mackey-Class time series data, (3) a nonlinear time series data from Chen [12], are conducted to investigate the convergence behaviors of the learning algorithms and the performance of the MLPs that are generated by these algorithms. The contributions of the paper are three folds.

1) Theoretically, we show that the convergence of the algorithm based on combining, either multiplicative or additive, weight noise injection with weight decay during training MLPs is with probability one.

2) It is demonstrated by simulations that the on-line learning algorithm based on pure multiplicative weight noise injection during training does not converge.

$^1$In [1], G.An applied a theorem from Bottou [8] to make claim on the convergence. The claim is not valid as the author overlooked that these on-line weight noise injection-based algorithms do not fulfill the conditions depicted in Bottou’s Theorem.

$^2$In G.An’s paper, the objective function is called cost function.
3) It is demonstrated by simulations that the MLPs attained by the algorithms combining weight noise injection and weight decay show better performance than the MLPs attained by pure weight noise injection during training.

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 of the network model and the weight decay training algorithm will be described. Then in Section 3, the algorithms MWN-WD and AWN-WD will be delineated. Their corresponding objective functions will be reviewed. In Section 4, the convergence of the MWN-WD algorithm will be proved. The convergence of the AWN-WD is proved in Section 5. Section 6 will prove that with probability one the weight vectors obtained by both MWN-WD and AWN-WD algorithms converge to a local minimum of their corresponding objective functions. The simulations and the dataset results are elucidated in Section 7. 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 8. Conclusions are given in the last section.

II. BACKGROUND

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

A. Network Model

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

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

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. It is a vector function of the input \( x_k \), the weight matrix \( A \) and the bias vector \( c \). The \( i \)th element of \( z \) is defined as

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

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, let \( g(x_k, w) \) be \( \nabla_w f(x_k, w) \) and \( g_i(x_k, w_i) \) be \( \nabla_w f(x_k, w) \). We get that

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

in which

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

(4)

Furthermore, we let \( \nabla_w g(x, w) \) be \( \nabla \nabla_w f(x, w) \) and \( \nabla_w g_i(x_k, w) \) be \( \nabla \nabla_w f(x_k, w) \) for all \( i = 1, \ldots, m \), it can be shown that

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

(5)

B. Weight Decay Training Algorithm

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

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

\[
z(t) = z(x_t, A(t), c(t)).
\]

(6)

(7)

By replacing \( w_t \) and \( x_t \) in (4) by \( w(t) \) and \( x_t \) respectively, we have

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

(8)

Here, we denote \( z_i(x_t, a_i(t), c_i(t)) \) by \( z_i(t) \) owing to save space.

The update equations for the weight vectors \( w_i \) (for \( i = 1, 2, \ldots, m \)) can thus be written as follows:

\[
w_i(t+1) = w_i(t) + \mu(t) \left( [y_t - f(x_t, w(t))] g_i(x_t, w_i(t)) - \alpha w_i(t) \right),
\]

(9)

where \( \mu(t) > 0 \) is the step size at the \( t \)th step, and \( \alpha > 0 \) is the decay constant. The objective function being minimized by (9) is given by

\[
V_{mse}(w) = \frac{1}{N} \sum_{k=1}^{N} (y_k - f(x_k, w))^2 + \alpha \|w\|^2.
\]

(10)

The last term \(-\alpha w_i(t)\) in (9) is called the forgetting term [23]. It has been proved in [35], [37] that the convergence of (9) is with probability one if \( \sum_t (1 + \mu(t)) = \infty \) and \( \sum_t \mu(t)^2 < \infty \).

III. COMBINING WEIGHT NOISE INJECTION AND WEIGHT DECAY DURING TRAINING

Let \( b_1(t), b_2(t), \ldots, b_m(t) \in \mathbb{R}^{(n+2)} \) 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 \( S_b \), i.e.

\[
P(b_1(t)) \sim \mathcal{N}(0, S_b I_{(n+2) \times (n+2)})
\]

(11)

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

\[
\tilde{w}_i(t) = (\tilde{d}_i(t), \tilde{a}_i(t)^T, \tilde{c}_i(t))^T
\]

\(3\)We will show later in this paper that the conditions on step size can be relaxed to \( \mu(t) \to 0 \).
be the perturbed weight vector associated with the \(i^{th}\) hidden node. The perturbed output of the \(i^{th}\) hidden node is denoted by
\[
\tilde{z}_i(t) = z_i(x_t, \tilde{a}_i(t), \tilde{c}_i(t)).
\]

The update of \(w_i\) based on combining weight noise injection and weight decay during training can be written as follows:
\[
w_i(t + 1) = w_i(t) + \mu(t) \left\{(y_k - f(x_t, \tilde{w}_i(t)))g_i(x_t, \tilde{w}_i(t)) - \alpha w_i(t)\right\},
\]
where
\[
g_i(x_t, \tilde{w}_i(t)) = \begin{bmatrix}
\tilde{d}_i(t) \tilde{z}_i(t)(1 - \tilde{z}_i(t)) x_t \\
\tilde{d}_i(t) \tilde{z}_i(t)(1 - \tilde{z}_i(t))
\end{bmatrix},
\]
(13)
\[\mu(t) > 0\] 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 (12) is given by
\[
\tilde{w}_i(t) = w_i(t) + b_i(t) \otimes w_i(t),
\]
(14)
where
\[
b_i(t) \otimes w_i(t) = (b_{i1}(t)w_{i1}(t), \ldots, b_{i(n+2)}(t)w_{i(n+2)}(t))^T.
\]
(15)

As in [3], [4], [28], [29], the noise variance \(S_0\) is assumed to be a small positive value\(^4\). Thus, \(f(x_t, \tilde{w}_i(t))\) and \(g_i(x_t, \tilde{w}_i(t))\) in (12) can be approximated as follows:
\[
f(x_t, \tilde{w}) \approx f(x_t, w) + \sum_{i=1}^{m} g_i(x_t, w_i) \sum_{i=1}^{m} (b_i \otimes w_i)
\]
and
\[
g_i(x_t, \tilde{w}_i(t)) \approx g_i(x_t, w_i) + \nabla w_i g_i(x_t, w_i) \sum_{i=1}^{m} (b_i \otimes w_i).
\]
(17)

In (16) and (17), 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 \(D\) has equal probability to be selected. By (12), (16) and (17), the conditional expectation of \(\tilde{w}_i(t + 1)\) over random vectors \(b_i(t), 1 \leq i \leq m\), is given by
\[
E[\tilde{w}_i(t + 1) | w(t)] = w_i(t) + \mu(t)h_i(w(t)),
\]
where
\[
h_i(w(t)) = \frac{1}{N} \sum_{k=1}^{N} \left(y_k - f(x_k, w(t))\right) g_i(x_k, w(t))
\]
\[
- \frac{S_0}{N} \sum_{k=1}^{N} \nabla w_i g_i(x_k, w(t)) v_i(x_k, w(t))
\]
\[- \alpha w_i(t)
\]
and
\[
v_i(x_k, w(t)) = w_i(t) \otimes w_i(t) \otimes g_i(x_k, w_i(t)).
\]
(20)

We have showed in [20] that
\[
h_i(w(t)) = -\nabla w_i V(w(t)),
\]
where
\[
V(w) = \frac{1}{2} \left\{ \left(\frac{1}{N} \sum_{k=1}^{N} \left(y_k - f(x_k, w)\right)^2 \right) + \frac{S_0}{N} \sum_{k=1}^{N} \sum_{i=1}^{m} \left\| g_i(x_k, w) \right\|_2^2 + \alpha \left\| w \right\|_2^2 \right\}
\]
(30)
In the later section, we will show that \(w(t)\) generated by (12) and (24) converges to a local minimum of \(V(w)\).
IV. CONVERGENCE OF MWN-WD ALGORITHM

In the section, the convergence proof is presented. First, we consider the update of the output weight vector $d(t)$ and apply Doob’s Martingale Convergence Theorem to show that $\lim \to_\infty \|d(t)\|_2$ exists. As the existence of $\lim \to_\infty \|d(t)\|_2$ cannot imply the existence of $\lim \to_\infty d(t)$, we consider the update of the elements in $d(t)$ and apply Doob’s Martingale Convergence Theorem to show the existence of their limits. The existence of $\lim \to_\infty \|d(t)\|_2$ and $\lim \to_\infty d_i(t)$, $1 \leq i \leq m$, together with the Doob’s Martingale Convergence Theorem are then applied to show the existence of $\lim \to_\infty a_i(t)$ and $\lim \to_\infty g_i(t)$. Finally, we conclude that $\lim \to_\infty w(t)$ exists.

Here and after, we let $b_d(t)$ be the random vector associated with the output vector $d$. That is,

$$b_d(t) = (b_{11}(t), b_{21}(t), \ldots, b_{m1}(t))^T,$$

(31)

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

A. Existence of $\lim \to_\infty \|d(t)\|_2$

By (1), (12) and (13), the update of $d_i(t)$ can be expressed as follows:

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

(32)

In vector-matrix form,

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

(33)

Based on (33), the boundedness of $E[\|d(t)\|_2]$ and the existence of $\lim \to_\infty \|d(t)\|_2$ can be stated in the following lemma.

**Lemma 1:** For the algorithm based on (12) and (14), if $0 < \mu(t) \alpha < 1$ and $S_b < 1$ for all $t \geq 0$, then $\lim \to_\infty \|d(t)\|_2$ exists and is finite with probability one.

**Proof:** We rewrite the update of $d(t)$ given by (33) as follows:

$$d(t + 1) = (1 - \mu(t)\alpha) d(t) + \mu(t) y_i \tilde{z}_i(t) - \mu(t) \tilde{z}_i(t) \tilde{z}_i^T(t) d(t) - \mu(t) \tilde{z}_i(t) \tilde{z}_i^T(t) (b_d(t) \otimes d(t)).$$

(34)

Here, we let

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

(35)

Equation (34) can be written as follows:

$$d(t + 1) = B(t) d(t) + \mu(t) y_i \tilde{z}_i(t) - \mu(t) \tilde{z}_i(t) \tilde{z}_i^T(t) (b_d(t) \otimes d(t)).$$

(36)

Since the elements in $b_d(t)$ are identical and independent mean zero Gaussian random variables with variance $S_b$,

$$E_d[ (b_d(t) \otimes d(t)) (b_d(t) \otimes d(t))^T | w(t) ]$$

$$= S_b \begin{bmatrix} d_1(t)^2 & 0 & \cdots & 0 \\ 0 & d_2(t)^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_m(t)^2 \end{bmatrix}$$

$$= S_b d(t) d^T(t).$$

Hence,

$$E_d[ \|\tilde{z}(t) \tilde{z}_i^T(t) (b_d(t) \otimes d(t)) \|^2 | w(t) ]$$

$$= S_b \text{Tr} \left\{ (\tilde{z}(t) \tilde{z}_i^T(t))^2 d(t) d^T(t) \right\}$$

$$= S_b \|\tilde{z}(t) \tilde{z}_i^T(t) d(t)\|_2^2,$$  

(37)

where $\text{Tr}$ is the trace operator. 

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

$$E_d[ \|d(t + 1)\|_2^2 | w(t) ] = \| B(t) d(t) + \mu(t) y_i \tilde{z}(t) \|_2^2 + \mu(t)^2 S_b \| \tilde{z}(t) \tilde{z}_i^T(t) d(t)\|_2^2.$$  

(38)

Let

$$\Gamma_1 = B(t) d(t) + \mu(t) y_i \tilde{z}(t),$$

$$\Gamma_2 = \mu(t)^2 \sqrt{S_b} \tilde{z}(t) \tilde{z}_i^T(t) d(t).$$

Equation (38) can be rewritten as follows:

$$E_d[ \|d(t + 1)\|_2^2 | w(t) ] = \frac{1}{2} \left\{ \|\Gamma_1 + \Gamma_2\|_2^2 + \|\Gamma_1 - \Gamma_2\|_2^2 \right\}.$$  

(39)

Hence,

$$E_d[ \|d(t + 1)\|_2^2 | w(t) ] \leq \max \left\{ \|\Gamma_1 + \Gamma_2\|_2^2, \|\Gamma_1 - \Gamma_2\|_2^2 \right\}.$$  

(40)

Note by (35) that

$$B(t) d(t) + \Gamma_2$$

$$= \left( (1 - \mu(t)\alpha) I_{m \times m} - \mu(t)(1 - \sqrt{S_b}) \tilde{z}_i \tilde{z}_i^T(t) \right) d(t)$$

(41)

and

$$B(t) d(t) - \Gamma_2$$

$$= \left( (1 - \mu(t)\alpha) I_{m \times m} - \mu(t)(1 + \sqrt{S_b}) \tilde{z}_i \tilde{z}_i^T(t) \right) d(t).$$

(42)

Since $S_b < 1$, and by the fact that the eigenvalues of $\tilde{z}_i \tilde{z}_i^T(t)$ are zero and $\sum_{i=1}^m \tilde{z}_i^2$, the following inequalities hold.

$$\|B(t) d(t) + \Gamma_2\|_2 \leq (1 - \mu(t)\alpha) \|d(t)\|_2$$

(43)

$$\|B(t) d(t) - \Gamma_2\|_2 \leq (1 - \mu(t)\alpha) \|d(t)\|_2.$$  

(44)

By Triangle Inequality, we get that

$$\|\Gamma_1 + \Gamma_2\|_2 \leq (1 - \mu(t)\alpha) \|d(t)\|_2$$

(45)

$$+ \mu(t) \|y_i \tilde{z}(t)\|_2$$

and

$$\|\Gamma_1 - \Gamma_2\|_2 \leq (1 - \mu(t)\alpha) \|d(t)\|_2$$

(46)

$$+ \mu(t) \|y_i \tilde{z}(t)\|_2.$$

By (40), (45) and (46),

$$\left( E_d[ \|d(t + 1)\|_2^2 | w(t) ] \right)^{1/2} \leq (1 - \mu(t)\alpha) \|d(t)\|_2 + \mu(t) \|y_i \tilde{z}(t)\|_2.$$  

(47)

As $y_i$ is generally bounded for all $t \geq 0$, $\|y_i \tilde{z}(t)\|_2$ is bounded by a positive constant. Let it be $\kappa_d$. Further by Jensen Inequality, we get that

$$E_d[ \|d(t + 1)\|_2 | w(t) ] \leq \left( E_d[ \|d(t + 1)\|_2^2 | w(t) ] \right)^{1/2}.$$  

(48)
Thus by (47) and (48),
\[ E_d[\|d(t+1)\|_2 | w(t)] \leq (1 - \mu(t)\alpha) \|d(t)\|_2 + \mu(t)\kappa_d. \] (49)
As the right hand side of (49) is independent of the random vector \(b_d(t)\),
\[ E[\|d(t+1)\|_2 | w(t)] \leq (1 - \mu(t)\alpha) \|d(t)\|_2 + \mu(t)\kappa_d. \] (50)
Equivalently,
\[ E[\|d(t+1)\|_2 - \kappa_d/\alpha | w(t)] \leq (1 - \mu(t)\alpha) \|d(t)\|_2 - \kappa_d/\alpha. \] (51)
Let
\[ \beta(t) = \|d(t)\|_2 - \frac{\kappa_d}{\alpha}. \] (52)
It is clear that \(\{\beta(t)\}_{t \geq 0}\) is a supermartingale and
\[ E[|\beta(t)|] \leq E[|\beta(t-1)|] \leq \cdots \leq E[|\beta(0)|]. \] (53)
By Doob’s Martingale Convergence Theorem, \(\lim_{t \rightarrow \infty} \beta(t)\) exists and is finite with probability one. Then from (52), \(\lim_{t \rightarrow \infty} \|d(t)\|_2\) 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 \(\lim_{t \rightarrow \infty} d(t)\), \(\lim_{t \rightarrow \infty} a_i(t)\) and \(\lim_{t \rightarrow \infty} c_i(t)\). The idea is described in the rest of this subsection.

As \(\lim_{t \rightarrow \infty} \|d(t)\|_2\) exists with probability one,
\[ \lim_{t \rightarrow \infty} E[\|d(t+1)\|_2 | w(t)] = \lim_{t \rightarrow \infty} \|d(t+1)\|_2. \]
Thus, for any small positive \(\epsilon\), there exists a time \(t^*\) such that
\[ P(\|d(t)\|_2 - \kappa_d/\alpha + \epsilon > 0) = 0 \] (54)
or equivalently
\[ \|d(t)\|_2 < \kappa_d/\alpha + \epsilon \] (55)
is with probability one for all \(t \geq t^*\).

Making use of (54), we can therefore derive inequalities for the bounds of \(E[d_i(t+1)|w(t)]\), \(E[a_i(t+1)|w(t)]\) and \(E[c_i(t+1)|w(t)]\) for \(t \geq t^*\), for \(i = 1, 2, \ldots, m\) and \(j = 1, 2, \ldots, n\).

Then, we can define supermartingales for \(d_i(t), a_{ij}(t)\) and \(c_{ij}(t)\) respectively and show the existence of their limits by Doob’s Martingale Convergence Theorem.

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

To show the existence of the limit of \(d(t)\), we consider (32) and \(t \geq t^*\).
\[ E_d[d_i(t+1)|w(t)] = (1 - \mu(t)\alpha)d_i(t) + \mu(t)(y_t - d^T(t)\hat{z}(t))\hat{z}_i(t) \leq (1 - \mu(t)\alpha)d_i(t) + \mu(t)\kappa_d + \mu(t)\|d(t)\|_2. \] (56)
As a result of (55),
\[ E_d[d_i(t+1)|w(t)] \leq (1 - \mu(t)\alpha)d_i(t) + \mu(t)\kappa_d + \mu(t)(\kappa_d/\alpha + \epsilon). \] (57)
Since the right hand side is independent of \(b_d(t)\), we can write that
\[ E[d_i(t+1)|w(t)] \leq (1 - \mu(t)\alpha)d_i(t) + \mu(t)\kappa_d + \mu(t)(\kappa_d/\alpha + \epsilon) \] (58)
for all \(t \geq t^*\).

Hence, we can define a random process \(\{\Gamma(t)\}_{t \geq 0}\) in which
\[ \Gamma(t) = d_i(t + t^*) - \frac{\kappa_d + \kappa_d/\alpha + \epsilon}{\alpha}. \] (59)
and clearly
\[ E[\|\Gamma(t)\|w(t^*)] = \cdots \leq E[\|\Gamma(0)\|w(t^*)]. \]
Therefore, by Doob’s Martingale Convergence Theorem, \(\lim_{t \rightarrow \infty} \Gamma(t)\) exists and is finite with probability one. We can conclude that \(\lim_{t \rightarrow \infty} d_i(t)\) 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 (12) and (14), if \(\alpha > 0\) and \(S_b < 1\), and \(\mu(t)\) is small for all \(t\), then \(\lim_{t \rightarrow \infty} d(t)\) exists and its elements are finite with probability one.

C. Existence of \(\lim_{t \rightarrow \infty} a_i(t)\)

The proof of the existence of \(\lim_{t \rightarrow \infty} a_i(t)\) is similar that of the proof of Lemma 2. By (1), (12) and (13), the update of \(a_i(t)\) can be expressed as follows:
\[ a_i(t + 1) = (1 - \mu(t)\alpha)a_i(t) + \mu(t)\hat{z}(t)(1 - \hat{z}(t))d_i(t)x_i(t) \] (60)
Note from (14) and (31) that
\[ \hat{d}_i(t) = d_i(t) + b_{i1}(t)d_i(t). \] (61)
and
\[ \hat{d}_i(t)d_i(t) = (d_i(t) + b_{i1}(t)d_i(t)) (b_d(t) \otimes d(t)). \] (62)
Let us consider the \(j^{th}\) element in \(a_i(t)\).
\[ a_{ij}(t + 1) = (1 - \mu(t)\alpha)a_{ij}(t) + \mu(t)\hat{z}_i(t)(1 - \hat{z}_i(t))d_i(t)x_{ij}(t) \] (63)
**Lemma 3:** For the algorithm based on (12) and (14), if \(0 < \mu(t)\alpha < 1\) and \(S_b < 1\) for all \(t \geq 0\), then for all \(i = 1, 2, \ldots, m\), \(\lim_{t \rightarrow \infty} a_i(t)\) exists and its elements are finite with probability one.

**Proof:** Given \(w(t)\) and taking expectation of (63) 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)v_1(t)d_i(t) (y_t - \hat{z}(t)(d(t) + S_b d_i(t)e_i) \) (64)
where
\[ v_1(t) = \hat{z}_i(t)(1 - \hat{z}_i(t))x_{ij} \] (65)
and
\[ \epsilon_i = \underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0}_{m-i}. \] (66)
Again, for \(t \geq t^*\), by (54) and (64),
\[ E_d[a_{ij}(t+1)|w(t)] \] \[ \leq (1 - \mu(t)\alpha)a_{ij}(t) + \mu(t)|d_i(t)||x_{ij}| + \mu(t)|x_{ij}| (|d_i(t)||d(t)||_2 + S_b d_i(t)^2). \] (67)
Since $|x_{ij}|$ is bounded, say by $\kappa_x$, we can replace the second term and the third term by $\mu(t)\kappa_x$, where
\[ \kappa_a = (\kappa_d/\alpha + \varepsilon)^2 \kappa_x (1 + S_0). \] (68)

Thus,
\[ E_d[a_{ij}(t+1)|w(t)] \leq (1 - \mu(t)\alpha)a_{ij}(t) + \mu(t)\kappa_a. \] (69)

As the right hand side is independent of $b(t)$,
\[ E[a_{ij}(t+1)|w(t)] \leq (1 - \mu(t)\alpha)a_{ij}(t) + \mu(t)\kappa_a. \] (70)

Similar to Lemma 2, we can define a random process $\{\xi(t)\}_{t \geq 0}$ as follows:
\[ \xi(t) = a_{ij}(t + t^*) - \frac{\kappa_a}{\alpha} \] (71)

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 $i = 1, \cdots, m$, $\lim_{t \to \infty} a_i(t)$ exists and its elements are finite with probability one.

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

By (1), (12) and (13), the update of $c_i(t)$ can be expressed as follows:
\[ c_i(t+1) = (1 - \mu(t)\alpha)c_i(t) + \mu(t)\bar{z}_i(t)(1 - \bar{z}_i(t))d_i(t)\left(y_i - \bar{z}^T(t)d(t)\right). \] (72)

Suppose, we define two augmented vectors as that
\[ \bar{x}_i = \begin{bmatrix} x_i \\ 1 \end{bmatrix} \quad \text{and} \quad \bar{a}_i(t) = \begin{bmatrix} a_i(t) \\ c_i(t) \end{bmatrix}. \]

We can combine (60) and (72) together and come up with the following update equation.
\[ \bar{a}_i(t+1) = (1 - \mu(t)\alpha)\bar{a}_i(t) + \mu(t)\bar{z}_i(t)(1 - \bar{z}_i(t))d_i(t)\bar{x}_i\left(y_i - \bar{z}^T(t)d(t)\right). \] (73)

Repeating the same steps as the proof of Lemma 3, we can conclude the existence of $\lim_{t \to \infty} \bar{a}_i(t)$ and thus $\lim_{t \to \infty} c_i(t)$ is with probability one.

Lemma 4: For the algorithm based on (12) and (14), if $0 < \mu(t)\alpha < 1$ and $S_0 < 1$ for all $t \geq 0$, then for all $i = 1, 2, \cdots, m$, $\lim_{t \to \infty} c_i(t)$ 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 (12) and (14), if $0 < \mu(t)\alpha < 1$ and $S_0 < 1$ 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 $\Omega_\epsilon(w^*)$ which is centered at $w^*$ and $\|w - w^*\| \leq \epsilon$ for all $w \in \Omega_\epsilon(w^*)$. Theorem 1 implies that for any arbitrary small positive $\epsilon$, there must exist a bounded region $\Omega_\epsilon(w^*)$ and a time $t(w^*)$, such that for all $t \geq t(w^*)$
\[ P(w(t) \in \Omega_\epsilon(w^*)) = 1. \] (74)

This final equation is very useful in the subsequent analysis.

V. CONVERGENCE OF AWN-WD ALGORITHM

Basically, the proof steps for the AWN-WD algorithm are the same as the steps for the MWN-WD algorithm. The only difference is the definition of $w$. Owing to save space, we only show the proof for the existence of $\lim_{t \to \infty} \|d(t)\|_2$, 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 (12) and (24), 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 $b_d(t) \otimes d(t)$ in (34) and (36) by $b_d(t)$, the update of $d(t)$ is given by
\[ d(t+1) = B(t)d(t) + \mu(t)y \bar{z}(t) - \mu(t)\bar{z}(t)\bar{z}^T(t)b_d(t). \] (75)

Note that $\bar{z}(t)$ in (75) now depends on $w(t) + b(t)$ instead of $w(t) + b(t) \otimes w(t)$.

Given $w(t)$, the expectation of $\|d(t+1)\|_2^2$ over the random vector $b_d(t)$ is then given by
\[ E_d[\|d(t+1)\|_2^2] \] (76)

\[ = \|B(t)d(t) + \mu(t)y \bar{z}(t)\|_2^2 + \mu(t)^2 E_d[\bar{a}_d(t)(\bar{z}(t)\bar{z}^T(t))^2 b_d(t)|w(t)|]. \]

In (76),
\[ E_d[b_d^T(t)(\bar{z}(t)\bar{z}^T(t))^2 b_d(t)|w(t)|] = \text{Tr} \left\{ E_d[\bar{z}(t)\bar{z}^T(t))^2 b_d(t)b_d^T(t)|w(t)|] \right\} \]
\[ = S_0 \text{Tr} \left\{ \bar{z}(t)\bar{z}^T(t) \right\} \]
\[ = S_0 \|\bar{z}(t)\|_2^2 \]
\[ \leq S_0 \bar{m}^2. \] (77)

As the elements in $\bar{z}(t)$ are bounded by 0 and 1, $\|\bar{z}(t)\|_2 \leq \sqrt{m}$.

\[ E_d[b_d^T(t)(\bar{z}(t)\bar{z}^T(t))^2 b_d(t)|w(t)|] \leq S_0 \bar{m}^2 \] (78)

and then
\[ E_d[\|d(t+1)\|_2^2|w(t)|] \leq \|B(t)d(t) + \mu(t)y \bar{z}(t)\|_2^2 + \mu(t)^2 S_0 \bar{m}^2. \] (79)
By Jensen Inequality and Triangle Inequality,
\[ E_d[\|\mathbf{d}(t+1)\|_2^2w(t)] \]
\[ \leq \|\mathbf{B}(t)d(t)\|_2 + \mu(t)\|y(t)\|_2 + \mu(t)\sqrt{s_b}m \]
\[ \leq (1 - \mu(t)a)\|d(t)\|_2 + \mu(t)\|y(t)\|_2 + \mu(t)s_b. \] (80)

As \( y_t \) is bounded for all \( t \geq 0 \), \( \|y(t)\|_2^2 \) is bounded by a positive constant. We can let \( \kappa_d = \|y(t)\|_2 + \sqrt{s_b}m \) and replace \( \kappa_d \) by \( \kappa_d' \) in the steps from (49) to (53), to prove that if \( 0 < \mu(t) < 1, \lim_{t \to \infty} \|d(t)\|_2 \) exists and is finite with probability one.

As the proofs for the existence of \( \lim_{t \to \infty} d(t) \), \( \lim_{t \to \infty} a_i(t) \) and \( \lim_{t \to \infty} c_i(t) \) follow the same proof steps for MWN-WD algorithm, they are skipped. Q.E.D.

VI. CONVERGENCE OF \( V(w(t)) \) AND \( \bar{V}(w(t)) \)

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 at a local minima of the corresponding objective functions. Therefore, it is necessarily to prove that their locations are at local minimum.

Before proceed to prove, we need to make three more assumptions on the noise variance \( S_b \) and the step size \( \mu(t) \) as follows:

\[ S_b \ll 1, \quad \mu(t) \to 0 \quad \text{for all} \quad t \geq 0, \quad \sum_{\tau = t}^\infty \mu(\tau) = \infty \quad \text{for all} \quad t \geq 0. \] (81) (82) (83)

The first assumption (81) is a common assumption made by other researchers [3], [4], [28], [29]. With this, the approximations of \( f(x_t, w(t)) \) in (16) and \( g_i(x_t, w(t)) \) in (17) for the MWN-WD algorithm, and its corresponding objective function in (22) are reasonable. Similarly, assumption (81) ensures that the approximations of \( f(x_t, w(t)) \) in (25) and \( g_i(x_t, w(t)) \) in (26) for the AWN-WD algorithm, and its corresponding objective function in (30) are reasonable. The second assumption (82) 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 \( \sum_{\tau = t}^\infty \mu(\tau)\|\nabla_w V(w(\tau))\|_2^2 \) diverges if \( \|\nabla_w V(w(t))\|_2 \) does not converge.

A. MWN-WD Algorithm

With the assumptions in (81), (82) and (83), the property of \( \nabla_w V(w^*) \) can then be stated as the following theorem.

**Theorem 3:** For the algorithm based on (12) and (14), if (i) \( \alpha > 0 \), (ii) \( S_b \ll 1 \), (iii) \( \mu(t) \to 0 \) for all \( t \geq 0 \) and (iv) \( \sum_{\tau = t}^\infty \mu(\tau) = \infty \) for any \( t \geq 0 \), then

\[ \nabla_w V(w^*) = \lim_{t \to \infty} \nabla_w V(w(t)) = 0, \] (84)

and \( V(w^*) \) is a local minima of \( V(w) \).

**Proof:** Since \( z_i(\cdot) \) in (4) is differentiable, \( g_i(x_t, w_i) \) is differentiable. Hence, \( h_i(w) \) given by (19) is differentiable. As \( h_i(w) \) is the gradient vector of \( V(w) \), we get the following properties: For all \( w \),

\[ V(w) \] is differentiable and \( \nabla_w V(w) \) is differentiable. (85) (86)

Furthermore, from Condition (i) and (iii) given in Theorem 3, we can prove by Theorem 1 that with probability one \( \lim_{t \to \infty} w(t) \) exists and with finite elements. As both \( V(w) \) and \( \nabla_w V(w) \) are differentiable and \( \lim_{t \to \infty} w(t) \) exists with probability one, \( \lim_{t \to \infty} V(w(t)) \) and \( \lim_{t \to \infty} \nabla_w V(w(t)) \) exist with probability one. The existence of \( \lim_{t \to \infty} V(w(t)) \) implies that for any arbitrary \( \epsilon_V \), there exists a time \( t_V \), such that

\[ P(\|V(w^*) - V(w(t))\| \leq \epsilon_V) = 1 \] (87)

for all \( t \geq t_V \).

By Taylor expansion of \( V(w(t+1)) \) about \( w(t) \),

\[ V(w(t+1)) = V(w(t)) + (\nabla_w V(w(t)))^T \Delta w(t), \] (88)

where

\[ \Delta w(t) = w(t+1) - w(t) \]
\[ = \mu(t)(y_t - \bar{d}(t)\tilde{z}(t)) \]
\[ \leq \frac{g_i(x_i, w(t))}{g_m(x_i, w(t))} \] (89)

By (18) and (19),

\[ E[V(w(t+1))|w(t)] = V(w(t)) - \mu(t)\|\nabla_w V(w(t))\|^2. \] (90)

Hence,

\[ E[V(w^*) - V(w(t_V))] \]
\[ = -\sum_{\tau = t_V}^{t} \mu(\tau)E[\|\nabla_w V(w(\tau))\|^2 w(t_V)] \] (91)

From (87),

\[ \sum_{t=t_V}^{\infty} \mu(t)E[\|\nabla_w V(w(t))\|^2 w(t_V)] \leq \epsilon_V. \] (92)

By virtue of (83) and the inequality that \( E[|q|] \leq (E[|q|^2])^{1/2} \), one can prove by contradiction that

\[ E[\|\nabla_w V(w^*)\| w(t_V)] = 0. \] (93)

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

\[ \nabla_w V(w^*) = 0. \] (94)

Hence,

\[ \nabla_w V(w^*) = 0. \] (95)

Also, the existence of \( \lim_{t \to \infty} V(w(t)) \) implies that

\[ E[V(w^*)|w(t_V)] = V(w^*). \]

By (91), we get that

\[ V(w^*) \leq V(w(t_V)). \] (95)

Therefore, we can conclude from (94) and (95) that \( V(w^*) \) is a local minima of \( V(w) \). The proof is completed. Q.E.D.
B. AWN-WD Algorithm

For the AWN-WD algorithm, we let \( w^{**} \) be the limit \( \lim_{t \to \infty} w(t) \). The property of \( \nabla_w \hat{V}(w^{**}) \) is stated in the following theorem.

**Theorem 4:** For the algorithm based on (12) and (24), if (i) \( \alpha > 0 \), (ii) \( S_b \ll 1 \), (iii) \( \mu(t) \to 0 \) for all \( t \geq 0 \) and (iv) \( \sum_{\tau=1}^{\infty} \mu(t) = \infty \) for any \( t \geq 0 \), then
\[
\nabla_w \hat{V}(w^{**}) = \lim_{t \to \infty} \nabla_w \hat{V}(w(t)) = 0, \tag{96}
\]
and \( \hat{V}(w^*) \) is a local minima of \( \hat{V}(w) \).

**Proof:** The proof is similar to the proof of Theorem 3. Since \( z_t(. \) in (4) is differentiable, \( g_t(x_t, w_t) \) is differentiable. Hence, \( h_t(w) \) given by (28) is differentiable. As \( h_t(w) \) is the gradient vector of \( \hat{V}(w) \), for all \( w \),
\[
\hat{V}(w) \text{ is differentiable (and also continuous) and } \nabla_w \hat{V}(w) \text{ is differentiable (and also continuous).} \tag{97}
\]
Further from Condition (i) and (iii) given in Theorem 4, we can prove by Theorem 2 that with probability one \( \lim_{t \to \infty} w(t) \) exists and with finite elements. As both \( \hat{V}(w) \) and \( \nabla_w \hat{V}(w) \) are differentiable and \( \lim_{t \to \infty} w(t) \) exists with probability one, \( \lim_{t \to \infty} \hat{V}(w(t)) \) and \( \lim_{t \to \infty} \nabla_w \hat{V}(w(t)) \) exist with probability one.

Then replacing \( \hat{V}(w) \) by \( \hat{V}(w^*) \), the proof can be accomplished by following the same steps as in the proof of Theorem 3. Q.E.D.

VII. SIMULATIONS

Three datasets are used to examine the convergence properties of the algorithms and the performance of the MLPs generated by these algorithms. The datasets include (i) 2D mapping, (ii) Mackey-Glass and (iii) NAR.

A. Datasets

**2D Mapping:** It is an artificial dataset which consists of 200 data points generated from the following equation.
\[
y_k = \sin(x_{k1}) \sin(x_{k2}) + e_k, \tag{99}
\]
where \( x_{k1}, x_{k2} \) are the \( k \)th sample inputs, \( y_k \) be its target output and \( e_k \) is a mean zero Gaussian noise with variance 0.01. Amongst these 200 data points, 100 of them are randomly selected to be the training data points, and the other 100 data points are the testing dataset.

**Mackey-Glass:** It is a benchmark time series dataset available on the Internet. The data is generated numerically from the differential equation given by
\[
\frac{dx(t)}{dt} = 0.2 \cdot \frac{x(t - \tau)}{1 + x(t - \tau)^{10}} - 0.1x(t), \tag{100}
\]
with \( x(0) = 1.2 \) and \( \tau = 17 \). In our simulations, we use 1000 points in this time series dataset. We denote the data in the time series by \( y(k) \) for \( k = 1, 2, \cdots, 1000 \). A MLP is trained to predict the current value \( y(k) \) based on the past observations \( y(k-1), y(k-2), y(k-3), y(k-4) \). That is to say, the MLP consists of four inputs and one output. The first 500 data are picked for training and others are for testing.

**NAR:** We consider the following nonlinear autoregressive (NAR) time series [12], given by
\[
y(k) = (0.8 - 0.5 \exp(-y^2(k - 1)))y(k - 1) - (0.3 + 0.9 \exp(-y^2(k - 1)))y(k - 2) + 0.1 \sin(\pi y(k - 1)) + \epsilon(k), \tag{101}
\]
where \( \epsilon(k) \) is a mean zero Gaussian random variable with variance equals to 0.09. One thousand samples were generated by \( y(0) = y(1) = 0.1 \). The first 500 data points were used for training and the other 500 points were used for testing. The neural network is used to predict \( y(k) \) based on the past observation \( y(k - 1) \) and \( y(k - 2) \).

B. Methodology

We use linear output MLPs for the simulations. For the 2D mapping, the network is with structure 2 - 10 - 1. For the Mackey-Glass series, the structure is 4 - 10 - 1. The structure for the NAR data is 2 - 10 - 1.

In the training process, five different weight noise variance 0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, and four different weight decay constants 0, 10^{-5}, 10^{-4}, 10^{-3} are defined. In total, 20 MLPs are generated for each dataset. The step size is set to 0.1 and the total number of epochs is 10^{5}.

For the MLPs that are generated by multiplicative weight noise injection during training, multiplicative weight noise of variances \([0, 0.002, 0.004, ..., 0.04]\) are added to the networks for testing. For each noise variance, 100 networks are generated and their testing MSEs are recorded. For the MLPs that are generated by additive weight noise injection during training, the evaluation method is the same except that the noise injected is additive instead of multiplicative.

C. Results

Here, we only present the key results. For the results of all settings, readers can refer to [25]. For all three datasets, we have also found that their results are quite similar.

Regarding the convergence behavior, it is clear from Fig 2, Fig 5, Fig 8, Fig 3, Fig 6 and Fig 9 that the MSE value converges in all combinations of \((\alpha, S_b)\). However, it is not the case for the weights. For multiplicative noise case, the weights do not converge when \( \alpha = 0 \). The weights diverge when \( \alpha = 0 \) and \( S_b = 10^{-2} \) and the weights converge when \( \alpha > 0 \), as showed in Fig 2, Fig 5 and Fig 8. Similar behavior has observed in the case additive noise, as showed in Fig 3, Fig 6 and Fig 9.

Regarding the performance of the trained MLP, for multiplicative weight noise case, the MLP generated by MWN-WD gives the best performance in comparison with other three algorithms, as showed in Fig 1a, Fig 4a and Fig 7a. For additive noise case, the MLP generated by AWN-WD also gives the best performance in comparison with other three algorithms, as showed in Fig 1b, Fig 4b and Fig 7b.

VIII. DISCUSSIONS

In this section, we would like to discuss three issues related to the results presented in this paper. First, we will discuss about the step size. Second, we will show that the convergence of weight decay learning with constant step size is with probability one. Third, we will discuss on the convergence behaviors and the objective functions of our algorithms if it
Theorem 3. We state these as the following corollaries of Theorem 1 and converges with probability one even if the step size is constant. when reduces to the original weight decay training algorithm (9) this condition is not necessary.

A. On the step size \( \mu(t) \)

The condition on the step size in Theorem 1 and Theorem 2 is much more relax than the condition of some convergence theorems for back-propagation algorithm. For instance in [37], the step size \( \mu(t) \) must be a decreasing function. In our proofs, this condition is not necessary.

If \( 0 < \mu(t)\alpha < 1 \) and \( S_0 < 1 \), the MWN-WD algorithm converges with probability one. If \( 0 < \mu(t)\alpha < 1 \), the AWN-WD algorithm converges with probability one. It implies that the step size \( \mu(t) \) can be defined in many ways, including \( \mu(t) = \varphi \), \( \mu(t) = \varphi(\sin(\omega t) + 1) \), \( \mu(t) = \varphi/t \), or \( \mu(t) \) is a small positive random number, where \( \omega \) is a positive number.

B. Weight decay learning with constant step size

In view of the fact that algorithm based on (12) and (14) reduces to the original weight decay training algorithm (9) when \( S_0 = 0 \), Theorem 1 implies that the algorithm (9) [37] converges with probability one even if the step size is constant. We state these as the following corollaries of Theorem 1 and Theorem 3.

Corollary 1: For the algorithm based on (9), 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.

Corollary 2: For the algorithm based on (9), if (i) \( \alpha > 0 \), (ii) \( \mu(t) \to 0 \) for all \( t \geq 0 \) and (iii) \( \sum_{t=1}^{\infty} \mu(t) = \infty \) for any \( t \geq 0 \), then

\[
\lim_{t \to \infty} \nabla_w V_{WD}(w(t)) = 0,
\]

and \( \lim_{t \to \infty} V_{WD}(w(t)) \) is a local minima of \( V_{WD}(w) \), where

\[
V_{WD}(w) = \frac{1}{2} \left\{ \frac{1}{N} \sum_{k=1}^{N} (y_k - f(x_k, w))^2 + \alpha \|w\|_2^2 \right\}.
\]

In [37], the conditions for the step size are \( \sum_{t=1}^{\infty} \mu^2(t) < \infty \) and \( \sum_{t=1}^{\infty} \mu(t) = \infty \). Our results show that the convergence is guaranteed even if the step size is a small constant.

C. MLPs with sigmoid output node

In this paper, the MLPs being discussed is with linear output node. As a matter of fact, the approach to convergence proof can also been applied to the MLPs with sigmoid output given by

\[
\phi(x_t, w(t)) = \frac{1}{1 + \exp(-f(x_t, w(t)))}
\]

\[
f(x_t, w(t)) = d(t)^T z(t),
\]
where the elements of \( z(t) \) is defined in (2).

For classification problem, the original update of \( w_i \) (for \( i = 1, 2, \cdots, m \)) is defined as follows:

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

(106)

where \( \mu(t) > 0 \) is the step size and \( \alpha > 0 \) is the decay constant\(^5\). The update of \( w_i \) based on combining weight noise injection and weight decay during training can be written as follows:

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

(108)

where \( \tilde{w}_i(t) \) is the weight vector which is corrupted by weight noise.

\(^5\)The objective function being minimized by (106) is the cross entropy error function.

\[
    V_{ce}(w) = \frac{1}{N} \sum_{k=1}^{n} \{y_k \ln \phi(x_k, w) + (1 - y_k) \ln (1 - \phi(x_k, w))\} + \alpha \| w \|^2.
\]

(107)

Using the same technique applied in the convergence proof for (12), we are able to prove that the convergence of the algorithm (108) is with probability one. As the proof steps are almost the same as those in the previous convergence proofs, we simply sketch the steps here.

From (108), the update of the \( d_i, a_{ij}, \) and \( c_i \) are expressed as follows:

\[
    d_i(t + 1) = (1 - \mu(t)\alpha)d_i(t) + \mu(t)\chi(t),
\]

(109)

\[
    a_{ij}(t + 1) = (1 - \mu(t)\alpha)a_{ij}(t)
    + \mu(t)\chi(t)\tilde{z}_i(t)(1 - \tilde{z}_i(t))x_{ij},
\]

(110)

\[
    c_i(t + 1) = (1 - \mu(t)\alpha)c_i(t)
    + \mu(t)\chi(t)\tilde{z}_i(t)(1 - \tilde{z}_i(t)),
\]

(111)

where

\[
    \chi(t) = (y_t - \phi(x_t, \tilde{w}(t)))\phi'(x_t, \tilde{w}(t))\tilde{z}_i(t),
\]

\[
    \phi'(x_t, \tilde{w}(t)) = \frac{\partial \phi}{\partial f} \bigg|_{f = f(x_t, \tilde{w}(t))}
    = \phi(x_t, \tilde{w}(t))(1 - \phi(x_t, \tilde{w}(t))).
\]

Since \( y_t, \phi(x_t, \tilde{w}(t)) \) and \( \tilde{z}(t) \) are bounded, it is ready showed
that

\[ E[d_i(t + 1)|w(t)] \leq (1 - \mu(t)\alpha)d_i(t) + \mu(t)\bar{\kappa}_d, \]
\[ E[a_{ij}(t + 1)|w(t)] \leq (1 - \mu(t)\alpha)a_{ij}(t) + \mu(t)\bar{\kappa}_a, \]
\[ E[c_i(t + 1)|w(t)] \leq (1 - \mu(t)\alpha)c_i(t) + \mu(t)\bar{\kappa}_c. \]

Hence, we can define the supermartingales \( \{d_i(t) - \bar{\kappa}_d/\alpha\}_{t \geq 0}, \{a_{ij}(t) - \bar{\kappa}_a/\alpha\}_{t \geq 0} \) and \( \{c_i(t) - \bar{\kappa}_c/\alpha\}_{t \geq 0} \). By Doob’s Martingale Convergence Theorem, the limits of these supermartingales exist and are finite with probability one. Therefore, we can state without proof in the following theorem in regard to the existence of \( \lim_{t \to \infty} w(t) \) that are generated by (108).

**Theorem 5:** For the algorithm based on (108), 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.

Moreover, the condition for which the convergence is ensured is simply \( \alpha > 0 \).

However, the objective function being minimized by (108) has not yet been found. We can only stop short for the convergence analysis. Further investigation on this objective function and the location in which \( w(t) \) converges are two of our future works.

**IX. Conclusion**

In this paper, we follow our previous work in [20] to present the convergence analyses on two on-line algorithms which combine the idea of weight noise injection and weight decay. Their convergence analyses are presented and simulation results on the convergence of the weights are elucidated. By applying Doob’s Martingale Convergence Theorem, we proved in Theorem 1 and Theorem 2 that under mild conditions the convergence of these algorithms are with probability one. Besides, we have also showed in Theorem 3 and Theorem 4 that the locations the weight vectors converge to a local minimum of the corresponding objective functions. Owing to study the convergence behaviors of these algorithms and their performance, intensive computer simulations are conducted on (i) a simple mapping problem, (ii) the Mackey-Glass time
with applications to classification problems as well. Similar results are
years, there is an increasing interest in the study of the effect
2, 6, 21, 28, 29], our works presented in this paper and
the pure weight noise injection-based algorithms
tolerance abilities) than the neural networks attained by
results have showed that the algorithms combining weight noise injection and weight
during training is able to alleviate the divergence effect due to
Moreover, our simulation
injection-based algorithms, the benefit of adding weight decay
a few issues regarding the conditions on the step size and
the convergence behaviors of the algorithms if they are applied to train a MLP
with sigmoid output node are discussed.

While most of the previous works are analyzing the effect
of noise in our brains and our human behaviors [11], [16], [17], [24], [27], [30]. Extend our work to study the effect
of noise (either brain noise or mental noise) on lifespan learning should be a valuable future direction.

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
[1] An G. The effects of adding noise during backpropagation training on a
and generalization ability of MLPs, Neural Computation, Vol.12, 2941-2964, 2000.
[5] Bernier J.L. et al, Improving the tolerance of multilayer perceptrons by

Fig. 7. Performance of MLPs in the NAR problem. The vertical axis
corresponds to the testing MSE and the horizontal axis corresponds to the
variance of noise $S_b$ injected during training.

Fig. 8. Change of the MSE and weights against time for the NAR problem
with multiplicative weight noise injection. From top to bottom, the parameters
$(\alpha, S_b)$ are $(0, 0)$, $(0, 10^{-2})$, $(10^{-5}, 0)$ and $(10^{-5}, 10^{-2})$. Note that the
time scale (horizontal axis) for the MSE is in log scale. While the time scale
for others are in linear scale and the data displayed are captured in every $10^5$
steps.

series prediction problem and (iii) a nonlinear time series
prediction problem. It is demonstrated that the algorithms
combining weight noise injection and weight decay ex-
hibit better convergence behaviors than their pure weight
noise injection counterparts. For multiplicative weight noise
injection-based algorithms, the benefit of adding weight decay
during training is much clear. Adding weight decay during
training is able to alleviate the divergence effect due to
multiplicative weight noise injection. Moreover, our simulation
results have showed that the neural networks attained by
the algorithms combining weight noise injection and weight
decay could have better performances (as well as fault
tolerance abilities) than the neural networks attained by
the pure weight noise injection-based algorithms 6. Finally,
a few issues regarding the conditions on the step size and
the extension of the theorems to the convergence of weight
decay algorithm with constant step size, and the convergence
behaviors of the algorithms if they are applied to train a MLP
with sigmoid output node are discussed.

Actually, we have conducted simulations for the sigmoidal output MLPs
with applications to classification problems as well. Similar results are
obtained. Readers who are interested in those results can refer to [25] for
the details.
Fig. 9. Change of the MSE and weights against time for the NAR problem.