Stability of Back Propagation Training Algorithm for Neural Networks

*Luma. N. M. Tawfiq**

Received 22, December, 2009
Accepted 19, February, 2012

Abstract*:*

In this paper, we derive and prove the stability bounds of the momentum coefficient μ and the learning rate η of the back propagation updating rule in Artificial Neural Networks .The theoretical upper bound of learning rate η is derived and its practical approximation is obtained .

Key words : BP training algorithm , Ann , Gradient Descent Training Algorithm , Gradient Descent with Momentum and Lyapunov Stability .

Introduction

Back propagation (BP) process can train multilayer Artificial Neural Networks **(** Ann's). With differentiable transfer functions, to perform a function approximation to continuous function $f \in R^n$, pattern association and pattern classification. The term of back propagation to the process by which derivatives of network error with respect to network weights and biases, can be computed. This process may be used with a number of different optimization strategies. The BP architecture was developed in the early of 1970 by several independent sources (Werbor; Parker; Rumelhart, Hinton and Williams). The central idea is that the errors of the units of the hidden layer are determined by back-propagating the errors of the units of the output layer. For this reason the method is often called the back-propagation learning rule. Back-propagation can also be considered as a generalization of the delta rule for non-linear activation functions and multilayer networks. The application of the generalization delta rule thus involves two phases [1] : During the first phase the input x is presented and propagated forward through the network to compute the output values y for each output unit. This output is compared with its desired value d, resulting in an error signal E for each output unit. The second phase involves a backward pass through the network during which the error signal is passed to each unit in the network and appropriate weight changes are calculated.Gradient (Steepest) Descent Training AlgorithmA standard back propagation algorithm is a gradient descent algorithm (as in the Widrow-Hoff learning rule) .For the basic gradient (steepest) descent algorithm, the weights and biases are moved in the direction of the negative gradient of the performance function .For the method of gradient descent, the weight update is given by :

$$w(k+1)=\square w(k)+\eta_k(-g_k) \quad ….(1)$$

where η_k is a parameter governing the speed of learning, named learning rate, controlling the distance between $W_{(t+1)}$ and $W_{(t)}$ and g_k is the gradient of the error surface at w(k) [2] .The

*Department of Mathematics, College of Education Ibn Al-Haitham, Baghdad University .

convergence condition is satisfied by choosing : $0 < \eta_k < \frac{1}{2\lambda_{max.}}$

where $\lambda_{max.}$ is the largest eigen value of weight matrix [3].

Gradient Descent With Momentum Training Algorithm [4]

There is another training algorithm for Ann that often provides faster convergence. The weight update formulas for gradient descent with momentum is given by :

$w(k+1)=\square w(k) \; +\eta_k \; (-g_k) \; +\mu \; (w(k) - \square w(k-1))$

that is :

$w(k+1) =\square w(k) + \eta_k (-g_k) + \mu \, \Delta w(k)$

i.e.

$$\Delta W(k+1) =\eta_k (-g_k) +\mu\Delta w(k) \quad \ldots(2)$$

where the momentum parameter μ is constrained to be in the range (0, 1). Momentum allows the Ann to make reasonably large weight adjustments, while using a smaller learning rate to prevent a large response to the error from any one of training pattern .

The gradient is constant (g_k = const) . Then, by applying iteratively (2) :

$\Delta W = - \eta \; g_k \; (1+ \mu + \mu^2 + \ldots) \; -\square \; - \frac{\eta}{1-\mu} g_k$

(because $\mu \in (0, 1)$ and then $\lim_{n\to\infty} \mu^n = 0$), i.e. the learning rate effectively increases from η_k to $\frac{\eta}{(1-\mu)}$

Stability and Convergence

Stability refers to the equilibrium behavior of the activation state of a neural network whereas convergence refers to the adjustment behavior of the weight during training, which will eventually lead to minimization of error between the desired and actual outputs [3]. Thus convergence is typically associated with supervised training, although it is relevant in all cases of training, both supervised and unsupervised. In this section we will discuss the global behavior of Ann's whose activation dynamics is described by the following set of equations [5] $\dot{c}_i = x_i(c_i)[d_i(c_i) - \sum_{k=1}^{N} w_{ik} z_k(c_k)]$ i= 1,2,.. ,M …. (3)

where $c_i = c_i(t)$ activation value of the i^{th} neuron and it is a function of time and the coefficients $[w_{ik}]$ form a symmetric matrix of weights, $\dot{c}_i$ (t) which gives the rate of change of the activation value of the i^{th} neuron of Ann, $X= (x_1,x_2,\ldots,x_N)^T$ is the input vector with components x_i, i =1 ,2,…,N , $d = (d_1,d_2,\ldots,d_M)^T$ is the desired output vector with components d_j , j = 1,2, … , M.These equations represent a class of N-dimensional competitive dynamical systems. In general, the activation state of the network starts from an initial state and follows a trajectory dictated by the dynamics of the equations. A network will be useful only if a trajectory leads eventually to an equilibrium state at which point there is no further change in the state.Such a state is also called a stable state, when a small perturbation of the state settles to the same state.Different initial states may follow different trajectories, all of which should terminate at some equilibrium states. There may be several trajectories that may terminate at the same equilibrium state.The existence of such equilibrium states enables global pattern formation possible in a network.That is, an input pattern corresponding to a starting state will eventually lead to one of the global patterns, which can be interpreted as storage of the input pattern in long term memory. The global pattern thus formed will only change if there is a different external input. In some cases the network parameters such as weight may slowly change due to learning or self-organization. If the global pattern formation still occurs for choice of these parameters, then the resulting pattern is said to be absolutely stable or globally stable.The set of equations (3) describing activation dynamics do exhibit stable states which are also called fixed point equilibrium states. Such a network then can form global patterns at these states, and hence can be used for pattern storage. One of the conditions is that the weights $\{w_{ik}\}$ should be symmetric ($w_{ik} = w_{ki}$).If the weights are not exactly symmetric, then the network may exhibit periodic oscillations of states in certain regions of the state space. These oscillatory regions are also stable, and hence can be used for pattern storage. Oscillatory stable states may also arise when there is some delay in the feedback of the outputs from other processing units to the current unit, even though the weights are exactly symmetric. In general, it is difficult to know whether a network will have stable points, and if so, how many. It is even more difficult to determine the behavior of the network near the stable points to examine the nature of stability. However, in a few cases it is possible to predict the global pattern behavior, if it is possible to show the existence of an energy function (error function , objective function) called Lyapunov function [6]. It is a scalar function of the parameters of the network, denoted by V(x), where x is the activation state vector of the network V(x) is said to be a Lyapunov function if $V(x) \leq 0$ for all x . [6]It is sufficient if we can find a Lyapunov function for a network in order to demonstrate the existence of stable equilibrium states. It is not a necessary condition, as the network may still have stable points, even thought a Lyapunov function could not be found. The existence of Lyapunov function makes it easy to analyze the stability of the network .If the Lyapunov function is interpreted as an energy function, then the condition that $V(x) \leq 0$ means that any change in the energy due to change in the state of the network results in lowering the total energy. Eventually the trajectory leads to a state from where there is no further decrease in the energy due to changes in the state. Such a state corresponds to the energy minimum, at which V(x) = 0. Normally there will be many states at which $\dot{V}(x) = 0$. All such states corresponds to equilibrium points or stable states. All trajectories in the state space will eventually lead to one of these stable states.

Discussion on Equilibrium

Normally the term equilibrium is used to denote the state of a network at which the network settles when small perturbations are made to the state. In the deterministic models (The activation model considered so far are deterministic models), the equilibrium states are also steady states. Hence these states satisfy the equations $\dot{c}_i(t) = 0$, for i = 1,…, N .

Note that : $\dot{c}_i(t) = 0$ is a necessary condition for a state to be an equilibrium state, but not a sufficient condition.

Definition [7]

x^* is said to be an equilibrium point (fixed point, stationary point, steady state) it $x(t) = x^*$ implies equality for all future time.

Lyapunov Stability Theory [7]

There are several mathematical definitions of the term " stability ". The one due to Lyapunov is most useful here.

Definition 1

The equilibrium state x = 0 is stable if, for any $\epsilon > 0$ there exists $\delta(\epsilon) > 0$, such that $|x(0)| < \delta$ implies $|x(t)| < \epsilon$ for all $t \geq 0$.

In other words, the system trajectory can be kept arbitrarily close to the origin by stating sufficiently close to it .

Note

An equilibrium state is unstable if the above condition is not satisfied. It's a bit tricky to negate the quantifiers, but here goes. There exists at least one Є such that for every δ > 0 , there exists a trajectory with | x(0) | < δ and | x(t) | ≥ Є for some t . For a linear system, in stability is equivalent to blowing up . In a nonlinear system , this is not the case .

Bounds of Momentum Coefficien(µ) of Back Propagation

For an energy function E(W) to be minimized, the updating rule for adjusting weights in neural network by using Back propagation with momentum is expressed by :

$$W(k+1) = W(k) + \eta\left(-\frac{dE}{dw}\right) + \mu\,(W(k) - W(k-1)) \quad \ldots(4)$$

where ŋ is the learning rate and µ is the momentum coefficient. It is well known in the literature [8] that ŋ is positive and µ is in [0,1). No proof is given however.Suppose that the structure of a multilayer Artificial neural network is determined and it can represent the unknown nonlinear mapping exactly with proper weight W^*. That is, for a given input pattern x_p , the desired output d_p can be obtained from the neural network output $y_p = f(W, x_p)$ with weight $W = W^*$, or $d_p = f(W^*, x_p)$ .The energy function E(W) is usually defined as the sum of squared error between the actual output y_p of neural network and the desired output d_p for all training patterns:

$$E = \frac{1}{2}\sum_{p=1}^{L}(d_p - y_p)^T(d_p - y_p) \quad \ldots(5)$$

Then the updating rule (4) will have the following form :-

$$W(k+1) = W(k) + \sum_{p=1}^{L} M_p(k)^T e_p(k) + \mu\ W(k) - W(k-1) \quad \ldots.(6)$$

Where

$M_p(k) = \dfrac{df(W(k), x_p)}{dW(k)}$ and $e_p(k) = d_p - y_p(k)$

$= f(w^*, x_p) - f(w(k), x_p)$

In this section, the necessary condition of ŋ and µ we will derive which ensure the stability and convergence of the updating rule (6).Now to find the bounds of µ :Updating rule (6) can be written as :

$$W(k+1) = W(k) + \eta\,h(k) + \mu\,(W(k) - W(k-1)) \quad \ldots(7)$$

Where $h(k) = \sum_{p=1}^{L} M_p(k)^T e_p(k)$.

Defining v(k) = W(k) - W(k-1), we get from (7)

$$v(k+1) = \mu\,v(k) + \eta\,h(k) \quad \ldots..(8)$$

The solution of (8) is :

$$v(k+1) = \mu^K v(1) + \eta \sum_{i=1}^{k} \mu^{k-i} h(i) \quad \ldots..(9)$$

Recalling the assumption that $d_p = f(W^*, x_p)$, we get the following lemma

Lemma 1

If $W(k) \longrightarrow W^*$, then $v(k) \longrightarrow 0$ and $e_p(k) \to 0$.

Proof

$v(k) = W(k) - W(k-1) \longrightarrow 0$ as $W(k) \longrightarrow W^*$, $e_p(k) = d_p - y_p(k) = f(W^*, x_p) - f(W(k), x_p)$, and f(W, X) is a continuous function of W.

Hence, $e_p(k) \longrightarrow 0$ for all p = 1,2,…, L , as $W(k) \to W^*$.

Theorem 1

Anecessary condition for the convergence of the updating rule (6) is that | µ | < 1 .

Proof

By Lemma (1), $v(k) \to 0$ and $e_p(k) \to 0$ as $W(k) \to W^*$.

That is, for any ϵ > 0, there exists an integer K such that $\| v(k) \| < \epsilon$ and $\| e_p(k) \| < \epsilon$ for all k > K.

For k = K+1, Equation (9) can be written as :

$$\mu^{K+1} v(1) + \eta \sum_{i=1}^{K} \mu^{K+1-i} h(i) = v(K+2) - \eta\,h(K+1) \ldots(10)$$

In view of (10), the following inequality can be easily derived :

$$|\mu|^{K+1}\| v(1) + \eta \sum_{i=1}^{K} \mu^{-i} h(i) \| \le \| v(K+2) \| + \eta \| h(K+1) \| \ \ldots(11)$$

As $W(K+1) \longrightarrow W^*$, $v(K+2) \longrightarrow 0$, and $h(K+1) \longrightarrow 0$ because $e_p(K+1) \longrightarrow 0$, W(K+1) is bounded, and all element of $M_p(K+1)$ are bounded. Therefore, the right hand side of (11) goes to zero. So the left hand side also goes to zero.

However, $\|v(1) + ŋ \sum_{i=1}^{K} \mu^{-i}\, h(\ i\ ) \|$ can not be zero in general, therefore, $| \mu |^{K+1}$ should go to zero as $K \longrightarrow 0$. Consequently $| \mu | < 1$ is required.

This theorem shows that $| \mu | < 1$ is a necessary condition. If $| \mu | \geq 1$, then the updating rule is unstable. The momentum coefficient μ is thus bounded by $-1< \mu < 1$.

Theoretical Lower and Upper Bounds of Learning Rate ŋ

The bounds for ŋ We will establish in two cases: $\mu = 0$ and $0 < | \mu | < 1$.

The case μ = 0

When $\mu = 0$, (6) can be written as :

$$W(\ k+1\ ) = W(\ k\ ) + ŋ\, h(\ k\ ) \quad ….(12)$$

Define $u(\ k\ ) = W^{*}- W(\ k\ )$, (12) is the same as : $u(\ k+1\ ) = u(\ k\ ) - ŋ\, h(\ k\ )$.

One way to show the convergence of W(k) is to require that :

$\|\ u(k+1)\ \|^2 \leq \|\ u(k)\ \|^2$. Then the following inequality must be true ;

$$\|\ u(\ k+1)\ \|^2 = \|\ u(k)\ \|^2 + ŋ^2\, h^T(\ k)\, h(\ k) - 2ŋ\, u^T(\ k)\, h^T(\ k) \leq \|\ u(\ k)\ \|^2 \quad …\ (13)$$

That is :

$$ŋ^2\, h^T(\ k\ )\, h(\ k\ ) \leq 2ŋ\, u^T(\ k\ )\, h(\ k\ ) \quad , \text{for all } k \quad ..(14)$$

Theorem 2

A necessary condition for the convergence of the updating rule (12) is that ŋ be positive.

Proof

When $ŋ = 0$, the updating rule is trivial because there will not be any weight updating in (12).When ŋ is negative, we will show it is impossible by contradiction.For the simple case of m =1 and L =1, the inequality (14) becomes (when both sides of (14) are divided by ŋ):

$$ŋ e^T(k)M(k)M^T(k)e(k) \geq 2\, u^T(k)M^T(k)e(k) \quad …..(15)$$

where $M(k) = \dfrac{df(W(k),X)}{dW(k)}$ is 1×M vector, and $e(k) = d - y(k) = f(W^{*},X) - f(W(k),X)$ is a scalar.

Then from (15), we get :

$$ŋ \geq \frac{2u^T(k)M^T(k)e(k)}{e^T(k)M(k)M^T(k)e(k)} \quad ….(16)$$

By taken the limit $W(k) \longrightarrow W^{*}$ on both sides of (16) becomes :

$$\lim_{w(k)\to W^{*}} ŋ \geq \frac{2N}{MM^T},$$

where $M = \dfrac{df(W^{*},X)}{dW^{*}}$,

and $N = u^T(\ k)\, M^T(\ k)\, e(\ k)$.

That means $ŋ \geq 0$, which contradicts the assumption that ŋ is negative. Therefore, ŋ must be positive.

Note

Theorem (2) determine the lower bound of ŋ is zero when $\mu = 0$, and the upper bound of ŋ for $\mu = 0$ can be derived from (14) as follows :

$$ŋ \leq \frac{2u^T(k)h(k)}{h^T(k)h(k)} \quad , \quad \text{when } \mu = 0$$

The case 0 < | μ | < 1

Use of the definition $u(\ k) = W^{*}- W(\ k)$, we get from (6) :

$$u(\ k+1) = u(\ k) - ŋ\, h(\ k) + \mu\, (\ u(\ k) - u(\ k-1)\ ) …..(17)$$

Multiplying both sides of (17) by $h^T(k)$ yields :

$$h^T(\ k)\, u(\ k+1) = h^T(\ k)\, u(\ k) - ŋ\, h^T(\ k)\, h(\ k) + \mu\, h^T(\ k) - \mu\, h^T(\ k)\, u(\ k-1).$$

$$ŋ\, h^T(\ k)\, h(\ k) = (1+ \mu\ )\, h^T(\ k)\, u(\ k) - \mu\, h^T(\ k)\, u(\ k-1) - h^T(\ k)\, u(\ k+1) \quad ….(18)$$

$$|ŋ|\, h^T(\ k)\, h(\ k) \leq (1+\mu\ )\, |\ h^T(\ k)u(\ k)\ | + |\ \mu\, h^T(\ k)\, u(\ k-1) + h^T(\ k)\, u(\ K+1)\ | …(19)$$

$$|ŋ| \leq \frac{(1+\mu)\,|\,h^T(k)u(k)\,| + |\,\mu h^T(k)u(k-1) + h^T(k)u(k+1)\,|}{h^T(k)h(k)}$$

Therefore, the upper bound of | ŋ | is given by :

| ŋ | ≤ B(k ,µ) for 0 < | µ | < 1 where B(k,µ) =

$$\frac{(1+\mu)\,|\,h^T(k)u(k)\,|+|\,\alpha h^T(k)u(k-1)+h^T(k)u(k+1)\,|}{h^T(k)h(k)}$$

, when 0< | µ | < 1

when ŋ = 0, (6) becomes W(k+1) = (1+ µ)W(k) - µ W(k-1), which is a system with eigenvalues 1 and µ. The system is marginally stable. Hence, ŋ should be positive for 0 < | µ | < 1.

The above results can be summarized by the following theorem :

Theorem 3

Anecessaryconditionforthe convergence of updating rule (6) is that ŋ should be 0 < ŋ < B(k ,µ) for -1 < µ < 1.

Computable Upper Bound of Learning Rate ŋ

Theorem(3) gives the theoretical upper bounds of ŋ which cannot be easily evaluated because W^* is not known a priori so that u(k) = W^* - W(k) cannot be computed as required by B(k,µ). Thus it would be helpful for us to find computable bounds which are approximations to the theoretical bounds.

As W(k) $\longrightarrow$ W^*,the limit of B(k,µ) is 2b for µ = 0, and the limit of B(k,µ) is 2b(1+µ) for 0< |µ| < 1. Because 2b (1+ µ) is equal to 2b when µ = 0, we can say that for all µ in (-1,1), the limit of B(k ,µ) is 2b(1+ µ) at convergence.

b(k) can be used as an approximation to b at each iteration step k in the learning process. In summary, we have the following theorem :

Theorem 4

An approximation of the upper bound of ŋ is Min [2 b(k) (1+ µ)].

The upper bound can be computed at each step k as the weights are updated.

Theorem (4) states that the upper bound of ŋ is proportional to 1+ µ. Since the upper bound is only a necessary condition, a more conservative value of ŋ should be used in applications.

References:

1. B. Krose and P. van der Smagt, 1996, An introduction to Neural Networks, Eighth edition, published inAmsterdam
2. L.N.M. Tawfiq, 2004, On Design And Training of Artificial Neural Networks For Solving Differential Equations , PhD. Thesis, College of Education Ibn Al-Haitham, Baghdad University .
3. B.Yegnanarayana , 2000, Artifical Neural Networks , Second edition , published in New Delhi .
4. T.Poggio and F.Girosi, 1989, A Theory of Networks for Approximation and Learning ,MASSACHUS-ETTS INSTITUTE OF TECHNOLOGY ARTIFICIAL INTELLIGENCE LABORATORY, A.I.Memo ,1140 (31) : 88 - 102 .
5. Z.Sen-Lin and L.Mei-Qin , 2005, Stability Analysis of discrete - time BAM neural networks based onstandard neural network models , Journal of Zhejiang University Science , 6A (7), : 689-696.
6. Z.Yi, P.A.Heng and A.W.C.Fu, 1999, Estimate of Exponential Convergence Rate and Exponential StabilityFor Neural Networks , IEEE , Vol. 10 (6) : 105 - 117 .
7. S.Sebastian , 2000 , Lyapunov Stability theory , published in El paso , Texas.
8. D.E.Rumelhart ,G.E.Hinton and R.J.Williams, 1986, Learning internal representations by error propagation ,parallel Distributed processing J., I : 318 – 363 .

أستقراريـة خـوارزميـة التدريب المرتد للشـبكات العصبيـة

لمى ناجي محمد توفيق

*قسم الرياضيات – كلية التربية أبن الهيثم – جامعة بغداد

الخلاصة :

يتضمن البحث اشتقاق وبرهان حدود أو الفترة التي ينتمي إليها ثابت العزم μ ومعدل التعلم η لقاعدة التدريب المرتد في الشبكات العصبية الصناعية كذلك يتضمن البحث اشتقاق القيد الأعلى لمعدل التعلم η نظريا والقيمة التقريبية عمليا.