PERGAMON

Neural Networks 12 (1999) 145–151

Neural Networks

Contributed Article

On the momentum term in gradient descent learning algorithms

Ning Qian[1]

Center for Neurobiology and Behavior, Columbia University, 722 W. 168th Street, New York, NY 10032, USA

Received 4 November 1997; revised 6 August 1998; accepted 6 August 1998

Abstract

A momentum term is usually included in the simulations of connectionist learning algorithms. Although it is well known that such a term greatly improves the speed of learning, there have been few rigorous studies of its mechanisms. In this paper, I show that in the limit of continuous time, the momentum parameter is analogous to the mass of Newtonian particles that move through a viscous medium in a conservative force field. The behavior of the system near a local minimum is equivalent to a set of coupled and damped harmonic oscillators. The momentum term improves the speed of convergence by bringing some eigen components of the system closer to critical damping. Similar results can be obtained for the discrete time case used in computer simulations. In particular, I derive the bounds for convergence on learning-rate and momentum parameters, and demonstrate that the momentum term can increase the range of learning rate over which the system converges. The optimal condition for convergence is also analyzed.

Keywords: Momentum; Gradient descent learning algorithm; Damped harmonic oscillator; Critical damping; Learning rate; Speed of convergence

1. Introduction

Connectionist neural network models have been successfully applied to a wide range of problems (Rumelhart and McClelland, 1986; McClelland and Rumelhart, 1986; Anderson et al., 1990; Churchland and Sejnowski, 1992). Although there are many different varieties of learning algorithms available, the majority of them—including the popular back-propagation learning algorithm—are of the gradient descent type. For a given network architecture, one usually starts with an error function which is parameterized by the weights (the connection strengths between units) in the network. The gradient of the error function with respect to each weight is then computed and the weights are modified along the downhill direction of the gradient in order to reduce the error. Let $E(\mathbf{w})$ be the error function, where $\mathbf{w}$ is a vector representing all the weights in the network, the simplest gradient descent algorithm, known as the steepest descent, modifies the weights at time step t according to:

$$\Delta\mathbf{w}_t = -\epsilon\nabla_w E(\mathbf{w}_t) \quad (1)$$

where ∇_w represents the gradient operator with respect to the weights, and ϵ is a small positive number known as the learning rate.

It is well known that such a learning scheme can be very slow. The inclusion of a momentum term has been found to increase the rate of convergence dramatically (Rumelhart et al., 1986). With this method, Eq. (1) takes the form:

$$\Delta\mathbf{w}_t = -\epsilon\nabla_w E(\mathbf{w}) + p\Delta\mathbf{w}_{t-1} \quad (2)$$

where p is the momentum parameter. That is, the modification of the weight vector at the current time step depends on both the current gradient and the weight change of the previous step. Intuitively, the rationale for the use of the momentum term is that the steepest descent is particularly slow when there is a long and narrow valley in the error function surface. In this situation, the direction of the gradient is almost perpendicular to the long axis of the valley. The system thus oscillates back and forth in the direction of the short axis, and only moves very slowly along the long axis of the valley. The momentum term helps average out the oscillation along the short axis while at the same time adds up contributions along the long axis (Rumelhart et al., 1986).

Other methods have also been proposed for improving the speed of convergence of gradient descent learning algorithms. For example, the conjugate gradient method has been shown to be superior to the steepest descent in most

[1] Tel.: +1-212-543-5213; Fax: +1-212-543-5161; E-mail: nq6@columbia.edu

0893–6080/99/$- see front matter
PII: S0893-6080(98)00116-6

applications (Press et al., 1992). However, the conjugate method requires more storage of intermediate results than the momentum method, and is non-local in the sense that the information needed to update a weight is not all contained in the pre- and post-synaptic units of the weight. This makes the algorithm less biologically plausible and harder to implement on hardware. In addition, the conjugate gradient method is less robust than the momentum method when the error surface is relatively flat, and when it is very different from a quadratic form in most parts of the parameter space (unpublished observations). Perhaps for these reasons, the momentum method appears to be dominant in the connectionist learning literature. In this paper, I attempt to mathematically analyze the effect of the momentum term on the speed of learning. I will first demonstrate an analogy between the momentum term in gradient descent and the mass of Newtonian particles in a conservative force field. This analogy will help us understand how the momentum term achieves its effect in the continuous time case. I will then examine the discrete time case used in computer simulations. Unlike the continuous time case, the discrete system is not guaranteed to converge to a minimum. I will derive the bounds for convergence on ϵ and p, and demonstrate that the momentum term can increase the range of ϵ over which the system converges. When p is close to one, ϵ can be nearly doubled. The optimal condition for fastest convergence to a minimum is also analyzed.

2. Physical analogy

Consider the continuous version of the steepest descent

$$\frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} = -\epsilon\nabla_{\mathrm{w}}E(\mathbf{w}) \tag{3}$$

where $\mathbf{w}$ is a continuous function of time instead of indexed by discrete time steps. Compare this equation with the Newtonian equation for a point mass m moving in a viscous medium with friction coefficient μ under the influence of a conservative force field with potential energy $E(\mathbf{w})$:

$$m\frac{\mathrm{d}^2\mathbf{w}}{\mathrm{d}t^2} + \mu\frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} = -\nabla_{\mathrm{w}}E(\mathbf{w}) \tag{4}$$

where $\mathbf{w}$ is the coordinate vector of the particle. It is clear that Eq. (3) can be viewed as the special case of Eq. (4) for a massless particle.

The above comparison between the steepest descent and the Newtonian equation prompts us to examine if the mass term in Eq. (4) could play some role in gradient descent. It turns out that Eq. (4) is equivalent to the continuous version of the momentum method specified by Eq. (2). To demonstrate, we discretize Eq. (4) to obtain:

$$m\frac{\mathbf{w}_{t+\Delta t} + \mathbf{w}_{t-\Delta t} - 2\mathbf{w}_t}{\Delta t^2} + \mu\frac{\mathbf{w}_{t+\Delta t} - \mathbf{w}_t}{\Delta t} = -\nabla_{\mathrm{w}}E(\mathbf{w}) \tag{5}$$

After rearrangements, we have:

$$\mathbf{w}_{t+\Delta t} - \mathbf{w}_t = -\frac{(\Delta t)^2}{m+\mu\Delta t}\nabla_{\mathrm{w}}E(\mathbf{w}) + \frac{m}{m+\mu\Delta t}(\mathbf{w}_t - \mathbf{w}_{t-\Delta t}). \tag{6}$$

This equation is identical to Eq. (2) if we let the learning rate ϵ and the momentum p be related to the friction coefficient μ and mass m according to:

$$\epsilon = \frac{(\Delta t)^2}{m+\mu\Delta t}, \tag{7}$$

$$p = \frac{m}{m+\mu\Delta t}. \tag{8}$$

Therefore, the steepest descent with a momentum term is equivalent to a Newtonian particle moving through a viscous medium under the influence of a conservative force field. Note that $p = 0$ implies $m = 0$ and vice versa according to Eq. (8). Thus, the momentum parameter plays the role of mass.

3. Stability and convergence analyses

3.1. Continuous time case

The analogy between the momentum method and the Newtonian mechanics discussed in the previous section also establishes that the momentum method is stable in the continuous time case and is guaranteed to converge to a local minimum for any positive m and μ (or equivalently, ϵ and p). This is because the total energy of the system

$$E_T = \tfrac{1}{2}m\frac{\mathrm{d}\mathbf{w}^T}{\mathrm{d}t}\frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} + E(\mathbf{w}) \tag{9}$$

is a Liapunov function that monotonically decreases due to the presence of friction. Without the momentum term the potential energy $E(\mathbf{w})$ is the error function being minimized. With the momentum term the total energy E_T is the new error function. The two error functions become identical towards the end of training because at the final equilibrium state the weight vector $\mathbf{w}$ will cease to change (or equivalently the velocity of the particle will become zero).

How does the momentum term speed up the convergence of the system to a local minimum? To understand this, we expand the potential energy in Eq. (4) around a minimum at $\mathbf{w}_0$ to obtain:

$$m\frac{\mathrm{d}^2\mathbf{w}}{\mathrm{d}t^2} + \mu\frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} \approx -H(\mathbf{w} - \mathbf{w}_0) \tag{10}$$

where H (the Hessian) is a symmetric and positive definite matrix with the elements:

$$h_{i,j} = \frac{\partial^2 E(\mathbf{w})}{\partial w_i \partial w_j}\Big|_{\mathrm{w}_0} \tag{11}$$

and the first order derivatives of $E(\mathbf{w})$ are omitted because

they vanish at the minimum. Eq. (10) represents a set of coupled and damped harmonic oscillators, with all the oscillators having the same mass m and damping coefficient μ. The coupling between them is determined by the H matrix. Without loss of generality, we let $\mathbf{w}_0 = 0$ in the following discussion ($\mathbf{w}_0$ can always be eliminated with the substitution $(\mathbf{w} - \mathbf{w}_0) \rightarrow \mathbf{w}$).

Since H is symmetric and positive definite, it can always be diagonalized with an orthogonal matrix Q through a similarity transformation:

$$H = QKQ^T,\ QQ^T = I \tag{12}$$

where K is a diagonal matrix with positive entries:

$$K = \begin{pmatrix} k_1 & & & \\ & k_2 & & \\ & & \ddots & \\ & & & k_n \end{pmatrix} \quad (k_i > 0), \tag{13}$$

k_is are the eigenvalues of H, and n is the number of oscillators (or equivalently, the number of weights). Using the transformation:

$$\mathbf{w}' = Q^T\mathbf{w}, \tag{14}$$

Eq. (10) can now be written as a set of uncoupled and damped oscillators:

$$m\frac{\mathrm{d}^2\mathbf{w}'}{\mathrm{d}t^2} + \mu\frac{\mathrm{d}\mathbf{w}'}{\mathrm{d}t} = -K\mathbf{w}' \tag{15}$$

We can now consider each oscillator separately. The ith oscillator is governed by:

$$m\frac{\mathrm{d}^2w'_i}{\mathrm{d}t^2} + \mu\frac{\mathrm{d}w'_i}{\mathrm{d}t} = -k_iw'_i \tag{16}$$

with k_i as the spring constant.Under the special case of no momentum term, which is equivalent to setting $m = 0$, the solution of Eq. (16) is simply:

$$w'_i(t) = ce^{\lambda_{i,0}t} \tag{17}$$

where c is a constant and

$$\lambda_{i,0} = -\frac{k_i}{\mu}. \tag{18}$$

Obviously, different w'_i will converge at a different rate determined by k_i, and the one corresponding to the smallest k_i will have the slowest speed. Since $\mathbf{w}$ is a linear combination of $\mathbf{w}'$, the convergence of $\mathbf{w}$ will be limited by the smallest k_i in the system.

To see how the momentum term helps speed up the convergence, note that when $m \neq 0$, the general solution of Eq. (16) becomes:

$$w'_i(t) = c_1e^{\lambda_{i,1}t} + c_2e^{\lambda_{i,2}t} \tag{19}$$

where c_1 and c_2 are constants, and the eigenvalues $\lambda_{i,1}$ and $\lambda_{i,2}$ are given by:

$$\lambda_{i,\{\substack{1\\2}\}} = -\frac{\mu}{2m} \pm \sqrt{\frac{\mu}{m}\left(\frac{\mu}{4m} - \frac{k_i}{\mu}\right)}. \tag{20}$$

Since μ, m and k_i are all positive, the real parts of the two eigenvalues are always negative so that the convergence of the system is ensured. For a given w'_i, the speed of its convergence is determined by the magnitudes of the real parts of the eigenvalues, with larger magnitudes corresponding to faster convergence. It is easy to show that

$$|\mathrm{Re}\lambda_{i,1}| \leq |\mathrm{Re}\lambda_{i,2}|, \tag{21}$$

and the equality holds when the square root in Eq. (20) is zero or imaginary. Therefore, the speed of convergence of w'_i is limited by $|\mathrm{Re}\lambda_{i,1}|$. To see the effect of the momentum term, we need to compare $|\mathrm{Re}\lambda_{i,1}|$ with $|\mathrm{Re}\lambda_{i,0}| = |\lambda_{i,0}| = k_i/\mu$.

The following result, which is proved in Appendix A, provides the condition under which the momentum term improves the speed of convergence.

Result 1: *For positive m, μ and k_i, the inequality*

$$|\mathrm{Re}\lambda_{i,1}| > \mathrm{Re}\lambda_{i,0}| \tag{22}$$

holds, and therefore the momentum term improves convergence, if and only if

$$k_i < \frac{\mu^2}{2m} \tag{23}$$

That is, given m and μ, for those k_is satisfying Inequality (23), the momentum term improves the convergence of the corresponding w'_is. The convergence of the remaining w'_is is not improved or slowed down. Therefore, to achieve an overall improvement, m and μ should be chosen such that Inequality (23) is satisfied for the majority of k_is. To quantify the degree of improvement, we define a positive parameter α according to:

$$|\mathrm{Re}\lambda_{i,1}| \equiv \alpha|\mathrm{Re}\lambda_{i,0}| = \alpha\frac{k_i}{\mu}. \tag{24}$$

Obviously, an α larger than 1 indicates improvement of convergence, and larger α means greater improvement. We prove the following result in Appendix A.

Result 2. *α reaches the maximum value of 2, and therefore the momentum term is most effective, when*

$$k_i = \frac{\mu^2}{4m}. \tag{25}$$

More specifically, α increases monotonically from 1 to 2 when k_i increases from a very small value to $\mu^2/4m$. It then decreases monotonically from 2 to 1 when k_i increases from $\mu^2/4m$ to $\mu^2/2m$. Finally, when k_i is larger than $\mu^2/2m$, α is smaller than 1 and decreases monotonically to 0.

Thus, given m and μ, the best convergence improvements

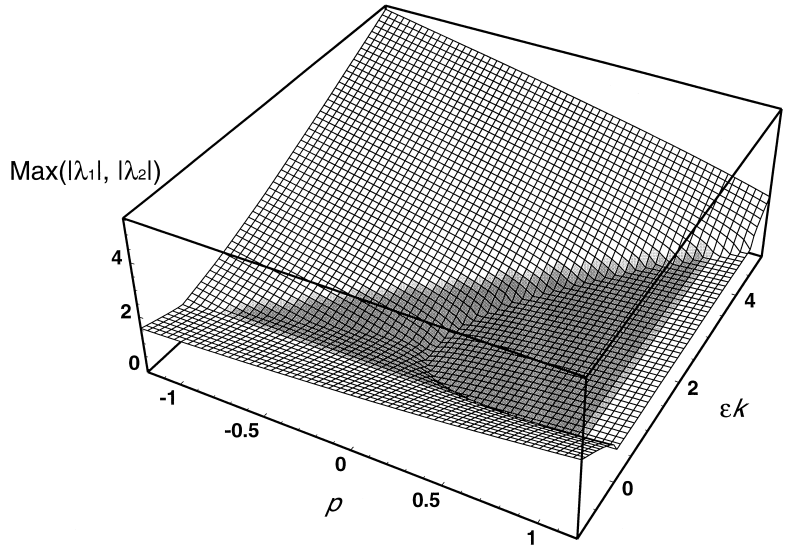

Fig. 1. Using Eq. (31), Max($|\lambda_{i,1}|$, $|\lambda_{i,1}|$) is plotted as a function of p and ϵk_i. The shaded area has values less than 1, and therefore corresponds to the parameter range for convergence. It has the triangular shape predicted by Result 3.

are for those w'_is whose k_is are near the value specified by Eq. (25), at the middle of the range allowed by Inequality (23). Eq. (25) is known as the *critical damping* condition for damped harmonic oscillators in physics (Kleppner and Kolenkow, 1973). Smaller k_is correspond to the so-called heavy damping condition where a relatively weak spring slowly pulls the oscillator, without oscillation along the way, to its equilibrium condition through a very viscous medium. Larger k_is correspond to the light damping condition where a strong spring quickly pulls the oscillator to its equilibrium position, overshoots, and oscillates back and forth, through a relatively less viscous medium, resulting in an overall slow settlement. The critical damping condition is right at the interface between these two cases, and it allows the fastest return of the oscillator to its equilibrium position. It is thus not surprising that the critical damping condition provides the best convergence improvement for the gradient descent learning algorithm.

When k_i is small, a first order expansion of Eq. (19) shows that $\lambda_{i,1} \approx \lambda_{i,0}$ at this limit. That is, small k_i is similar to the no momentum case. Therefore, without the momentum term the system behaves like heavy damping. The momentum term speeds up learning by bringing some eigen components of the system closer to the critical damping.

3.2. Discrete case

While the above results of the continuous case give a clear physical picture of the effect of the momentum term, computer simulations are necessarily discrete. We investigate how the momentum term works in the discrete case in this section. Eq. (6) near a local minimum becomes:

$$\mathbf{w}_{t+1} = [(1+p)I - \epsilon H]\mathbf{w}_t - p\mathbf{w}_{t-1}. \tag{26}$$

Similar to the continuous case, this set of equations can be decoupled by diagonalizing H using Eqs. (12)–(14) to obtain:

$$\mathbf{w'}_{t+1} = [(1+p)I - \epsilon H]\mathbf{w'}_t - p\mathbf{w'}_{t-1}. \tag{27}$$

We can now consider each of the equations separately, with the ith equation given by:

$$w'_{i,t+1} = [1+p-\epsilon k_i]w'_{i,t} - pw'_{i,t-1}. \tag{28}$$

Supplying the dummy equation $w'_{i,t} = w'_{i,t}$, we can rewrite the equation in matrix form:

$$\begin{pmatrix} w'_{i,t} \\ w'_{i,t+1} \end{pmatrix} = A\begin{pmatrix} w'_{i,t-1} \\ w'_{i,t} \end{pmatrix} = A^t\begin{pmatrix} w'_{i,0} \\ w'_{i,1} \end{pmatrix}, \tag{29}$$

where the matrix A is given by:

$$A = \begin{pmatrix} 0 & 1 \\ -p & 1+p-\epsilon k_i \end{pmatrix}. \tag{30}$$

The convergence of w'_i is determined by the eigenvalues of matrix A:

$$\lambda_{i,\{1\atop 2\}} = \frac{1+p-\epsilon k_i \pm \sqrt{(1+p-\epsilon k_i)^2 - 4p}}{2}. \tag{31}$$

To ensure convergence, we require $|\lambda_{i,1}| < 1$ and $|\lambda_{i,2}| < 1$, or equivalently:

$$\mathrm{Max}(|\lambda_{i,1}|, |\lambda_{i,2}|) < 1. \tag{32}$$

We prove the following result in Appendix A.

Result 3: $\mathrm{Max}(|\lambda_{i,1}|, |\lambda_{i,2}|) < 1$, *and therefore the system described by Eq. (28) converges, if and only if* $-1 < p < 1$ *and* $0 < \epsilon k_i < 2 + 2p$.

A graphical demonstration of this result is shown in Fig. 1. Without the momentum term ($p = 0$), the condition for convergence of w'_i becomes $0 < \epsilon k_i < 2$. When positive p is used, the range of learning rate ϵ that ensures convergence clearly increases with p. Since p can be almost as large as 1, the parameter range of convergence can be nearly doubled. It is interesting to note that in most simulations of connectionist learning algorithms, the p values were indeed chosen to be close to 1, typically around 0.9 (for examples see Rumelhart et al., 1986; Qian and Sejnowski, 1988, 1989). Also note that unlike the continuous case, the system is not guaranteed to converge for any positive values of ϵ, p and k_i.

Similar to the continuous case, we are also interested in finding out how the rate of convergence is affected by the momentum term. It is easy to show from Eq. (28) that without the momentum term ($p = 0$), the convergence speed of w'_i is determined by:

$$\lambda_{i,0} = 1 - \epsilon k_i \tag{33}$$

Obviously, if both $|\lambda_{i,1}|$ and $|\lambda_{i,2}|$ are smaller than $|\lambda_{i,0}|$ then the momentum term speeds up the convergence of w'_i; if one of them is larger than $|\lambda_{i,0}|$, then the convergence is slowed down. To avoid tedious calculations with complicated inequalities, we consider the special case of small ϵ typically used in simulations, and expand λ in Eq. (31) to obtain:

$$\lambda_{i,\left\{\begin{smallmatrix}1\\2\end{smallmatrix}\right\}} \approx \begin{cases} 1 - \dfrac{\epsilon k_i}{1-p}, \\ p\left(1 + \dfrac{\epsilon k_i}{1-p}\right). \end{cases} \tag{34}$$

Both are positive for small ϵk_i. For $-1 < p < 0$, we have $\lambda_{i,1} > \lambda_{i,0}$ and therefore the momentum term slows down the convergence for negative p. On the other hand, for $0 < p < 1$, we have $\lambda_{i,1} < \lambda_{i,0}$. We will also have $\lambda_{i,2} < \lambda_{i,0}$ if $p < 1 - \sqrt{\epsilon k_i}$. Therefore, when $0 < p < 1 - \sqrt{\epsilon k_i}$, the momentum term speeds up convergence.

It is easy to see from Eq. (34) that when p increases, $\lambda_{i,1}$ decreases while $\lambda_{i,2}$ increases. The best convergence speed is achieved when $\lambda_{i,1} = \lambda_{i,2}$. Using the original Eq. (31) we find that the best p value is given by

$$p = \left(1 - \sqrt{\epsilon k_i}\right)^2 \tag{35}$$

for small ϵk_i, and the corresponding eigenvalues are

$$\lambda_{i,\left\{\begin{smallmatrix}1\\2\end{smallmatrix}\right\}} = 1 - \sqrt{\epsilon k_i} < \lambda_{i,0}. \tag{36}$$

One might argue that if ϵk_i is not restricted to be small then the best convergence speed could be achieved by excluding the momentum term ($p = 0$), and by setting $\epsilon k_i = 1$ so that $\lambda_{i,0} = 0$. The problem is that there are n (the number of weights in the network) k_is for a given local minimum and they can be of very different magnitudes. To ensure convergence for all w'_is, ϵ has to be smaller than $2/\mathrm{Max}(k_i)$. This results in small ϵk_i for most k_is. We have shown above that the momentum term can not only increase the speed of convergence for small ϵk_i, but can also nearly double the parameter range for convergence.

4. Discussion

In this paper we demonstrated an equivalence between the momentum parameter in the gradient descent learning algorithms and the mass of Newtonian particles that move through a viscous medium under a conservative force field. The behavior of gradient descent near a local minimum is equivalent to a set of coupled and damped harmonic oscillators. Within a reasonable parameter range, the momentum term can improve the speed of convergence for most eigen components in the system by bringing them closer to critical damping. For the discrete time case, the momentum term provides the additional benefit of nearly doubling the parameter range over which the system converges.

The optimal choice of the momentum and learning-rate parameters for the ith eigen component (w'_i) in both the continuous and discrete time cases (see Eqs. (23) and (35)) depends on the value of k_i, which characterizes the ith canonical dimension of the error surface. Since for a given local minimum the k_is characterizing the minimum can cover a wide range of values, it is impossible to make the near optimal choice for all w'_is at the same time. One strategy might be to use different momentums and learning rates for different weights (w_i) in the network, resulting in momentum and learning rate matrices. This approach has been found to speed up training (Jacobs, 1988). However, it may be limited by the fact that each w_i is a linear combination of all w'_i. The convergence of each w_i, therefore, depends on all k_is, and no single optimal set of parameters can be chosen for w_i.

Obviously, one should first decouple the weights by rotating $\mathbf{w}$ into the eigenspace of the Hessian H (see Eq. (14)) and then use Eq. (25) or Eq. (35) to determine the optimal training parameters for each eigencomponent separately. However, this is practically impossible to do because of the huge size of H for networks with large numbers of weights. Diagonal approximation of H, which neglects all off-diagonal terms, has been found to be adequate in the Optimal Brain Damage (OBD) algorithm for removing unimportant weights (LeCun et al., 1990). Under this approximation, we have $k_i = h_{i,i}$, $Q = I$, and the weights are still left coupled. It is an empirical question whether the k_is so determined can be used effectively in Eq. (25) or Eq. (35) for choosing training parameters. An affirmative answer would allow an integration of these equations into the OBD algorithm to improve the rate of convergence without much extra computational cost.

An alternative approach for speeding up training would be to use a single set of momentum and learning-rate parameters for all the weights in the network, but to let them step through ordered sets of values over time during training. A

few iterations at each parameter set would quickly converge those w'_i components in all w_is whose k_is approximately satisfy the optimal condition. Different w'_i components would be converged at different times.

Acknowledgements

The author is supported by NIH grant #MH54125 and a Sloan Research Fellowship.

Appendix A Proof of Results 1 and 2

First, consider the case when

$$0 < k_i \leq \frac{\mu^2}{4m}. \tag{A1}$$

Under this condition, both $\lambda_{i,1}$ and $\lambda_{i,2}$ of Eq. (20) are negative real numbers. It is easy to show that:

$$-\lambda_{i,1} > -\lambda_{i,0} \tag{A2}$$

and therefore Inequality (22) holds. According to the definition of α in Eq. (24), we have

$$\frac{\mu}{2m} - \sqrt{\frac{\mu}{m}\left(\frac{\mu}{4m} - \frac{k_i}{\mu}\right)} = \alpha\frac{k_i}{\mu}. \tag{A3}$$

Since it is straightforward to verify that

$$\frac{\mu}{2m} - \sqrt{\frac{\mu}{m}\left(\frac{\mu}{4m} - \frac{k_i}{\mu}\right)} > \frac{k_i}{\mu} \tag{A4}$$

and

$$\frac{\mu}{2m} - \sqrt{\frac{\mu}{m}\left(\frac{\mu}{4m} - \frac{k_i}{\mu}\right)} \leq 2\frac{k_i}{\mu} \tag{A5}$$

we have

$$1 < \alpha \leq 2. \tag{A6}$$

Rearranging Eq. (A3) to obtain:

$$\frac{mk_i}{\mu^2}\alpha^2 - \alpha + 1 = 0 \tag{A7}$$

we see that when $k_i \to 0$, $\alpha \to 1$; and when $k_i = \mu^2/4m$, $\alpha = 2$. To demonstrate that α increases monotonically for k_i in the interval $(0, \mu^2/4m]$, note that

$$\frac{\partial\alpha}{\partial k_i} = \frac{\alpha^2 m}{\mu^2 - 2\alpha m k_i} > 0 \tag{A8}$$

because

$$\mu^2 - 2\alpha m k_i \geq \mu^2 - 4mk_i > 0. \tag{A9}$$

Next consider the case when

$$\frac{\mu^2}{4m} < k_i < \frac{\mu^2}{2m} \tag{A10}$$

Under this condition, both $\lambda_{i,1}$ and $\lambda_{i,2}$ are complex with negative real parts equal to $(-\mu/2m)$. Therefore,

$$|\mathrm{Re}\lambda_{i,2}| = |\mathrm{Re}\lambda_{i,1}| = \frac{\mu}{2m} > \frac{k_i}{\mu} = |\lambda_{i,0}| \tag{A11}$$

and therefore Inequality (22) holds. According to the definition of α in Eq. (24), we have

$$\frac{\mu}{2m} = \alpha\frac{k_i}{\mu} \tag{A12}$$

Obviously, α decreases monotonically from 2 to 1 for k_i in the interval $[\mu^2/4m,\ \mu^2/2m]$.

The above considerations establish the sufficient condition for Result 1. To prove the necessary condition, we finally consider the remaining case

$$k_i \geq \frac{\mu^2}{2m} \tag{A13}$$

Again, both $\lambda_{i,1}$ and $\lambda_{i,2}$ are complex with a negative real part equal to $(-\mu/2m)$. Therefore,

$$|\mathrm{Re}\lambda_{i,2}| = |\mathrm{Re}\lambda_{i,1}| = \frac{\mu}{2m} < \frac{k_i}{\mu} = |\lambda_{i,0}| \tag{A14}$$

and Inequality (22) does not hold. Eq. (A12) remains valid here and it indicates that α decreases monotonically from 1 to 0 for k_i in the interval $[\mu^2/2m,\ \infty)$.

Appendix B Proof of Result 3

For clarity we drop the subscript i for λ and k. We systematically examine the magnitudes of λ_1 and λ_2 under all possible conditions.

We first show that the condition $\epsilon k_i > 0$ is required for convergence. Assume $\epsilon k_i \leq 0$. Then,

$$\begin{aligned}\lambda_1 &= \frac{1 + p + |\epsilon k| + \sqrt{(1 + p + |\epsilon k|)^2 - 4p}}{2} \\ &\geq \frac{1 + p + |\epsilon k| + \sqrt{(1 + p)^2 - 4p}}{2} \\ &= \frac{1 + p + |\epsilon k| + |1 - p|}{2} \geq 1\end{aligned} \tag{A15}$$

The last step is obvious by considering $p < 1$ and $p \geq 1$ separately. Thus, the convergence of the system requires $\epsilon k_i > 0$. This means the learning rate ϵ should be positive because by definition k_i is always positive.

Let

$$\Delta \equiv (1 + p - \epsilon k)^2 - 4p \tag{A16}$$

in Eq. (31). It can be shown that $\Delta \geq 0$ when $p \leq (1 - \sqrt{\epsilon k})^2$ or $p \geq (1 + \sqrt{\epsilon k})^2$, and that $\Delta < 0$ when $(1 - \sqrt{\epsilon k})^2 < p < (1 + \sqrt{\epsilon k})^2$. We consider these cases of $\Delta \geq 0$ and $\Delta < 0$ separately.

I. $\Delta \geq 0$

Under this condition, $\lambda_{1,2}$ are real. We further divide this

case into two sub-cases corresponding to the two conditions that ensure $\Delta \geq 0$.

A. $p \geq (1+\sqrt{\epsilon k})^2$

Under this condition, we have $1+p-\epsilon k \geq 1+(1+\sqrt{\epsilon k})^2-\epsilon k = 2+2\sqrt{\epsilon k} > 0$. This implies that $\lambda_{1,2} > 0$ and $\lambda_1 \geq \lambda_2$. The convergence of the system requires

$$\lambda_1 = \frac{1+p-\epsilon k+\sqrt{(1+p-\epsilon k)^2-4p}}{2} < 1, \tag{A17}$$

which reduces to:

$$\sqrt{(1+p-\epsilon k)^2-4p} < 1-p+\epsilon k. \tag{A18}$$

However, the right-hand side $1-p+\epsilon k \leq 1-(1+\sqrt{\epsilon k})^2+\epsilon k = -2\sqrt{\epsilon k} \leq 0$ while the left side is greater or equal to zero. This contradiction leads to the conclusion that: If $p \geq (1+\sqrt{\epsilon k})^2$, the system diverges.

B. $p \leq (1-\sqrt{\epsilon k})^2$

We further divide this subcase into two for the purpose of proof.

a. $1+p-\epsilon k \geq 0$

Under these conditions we also have $\lambda_{1,2} > 0$ and $\lambda_1 \geq \lambda_2$. Eq. (A17) gives the stability condition, which leads to the requirements $\epsilon k > 0$, which is already stated, and $1-p+\epsilon k > 0$, which is satisfied because $1-p+\epsilon k \geq 1-(1-\sqrt{\epsilon k})^2+\epsilon k = 2\sqrt{\epsilon k} > 0$. We conclude that: if $p \leq (1-\sqrt{\epsilon k})^2$ and $1+p-\epsilon k \geq 0$ and $\epsilon k > 0$, the system converges. It can be shown that these conditions also imply $|p| < 1$. First, $p \geq \epsilon k - 1 > -1$. To show $p < 1$, assume the opposite that $p \geq 1$. Then $(1-\sqrt{\epsilon k})^2 \geq p \geq 1$. This leads to $\epsilon k \geq 4$. Using this result, $(1-\sqrt{\epsilon k})^2 \geq p$ becomes $\sqrt{\epsilon k}-1 \geq \sqrt{p}$, or $\epsilon k \geq 1+p+2\sqrt{p}$. This contradicts the assumption $\epsilon k < 1+p$. Therefore, $p \leq 1$.

The results can be summarized as: If $p \leq (1-\sqrt{\epsilon k})^2, |p| < 1$, and $0 < \epsilon k \leq 1+p$, the system converges.

b. $1+p-\epsilon k \leq 0$

In this case, $\lambda_{1,2} \leq 0$ and $|\lambda_2| \geq |\lambda_1|$. Stability now requires:

$$\lambda_2 = \frac{1+p-\epsilon k-\sqrt{(1+p-\epsilon k)^2-4p}}{2} > -1 \tag{A19}$$

which reduces to

$$\sqrt{(1+p-\epsilon k)^2-4p} < 3+p-\epsilon k. \tag{A20}$$

This is turn leads to the requirements $3+p-\epsilon k > 0$ and $2+2p-\epsilon k > 0$. We conclude that if $p \leq (1-\sqrt{\epsilon k})^2$, $1+p-\epsilon k \leq 0$, $3+p-\epsilon k > 0$ and $2+2p-\epsilon k > 0$, the system converges. Similar to the case above, the conditions also imply $|p| < 1$. To see this, first note that $1+p \leq \epsilon k < 2+2p$, or $p > -1$. To show $p < 1$, assume the opposite $p \geq 1$. Then $\epsilon k \geq 1+p \geq 2$. Consequently, $p \leq (1-\sqrt{\epsilon k})^2$ implies $\sqrt{p} \leq \sqrt{\epsilon k}-1$. Combining this with $\epsilon k < 3+p$ gives $p < 1$, a contradiction. Therefore, $p < 1$. Because the condition $3+p-\epsilon k > 0$ is contained in the condition $2+2p-\epsilon k > 0$ when $|p| < 1$, we can simplify the convergence requirements to: If $p \leq (1-\sqrt{\epsilon k})^2$, $|p| < 1$ and $1+p \leq \epsilon k < 2+2p$, the system converges.

II. $\Delta \leq 0$

This occurs when $(1-\sqrt{\epsilon k})^2 < p < (1+\sqrt{\epsilon k})^2$. Both λ_1 and λ_2 are complex in this case. Eq. (31) becomes:

$$\lambda_{1,2} = \frac{1+p-\epsilon k \pm i\sqrt{4p-(1+p-\epsilon k)^2}}{2}. \tag{A21}$$

Stability requires that $|\lambda_{1,2}| = \sqrt{p} < 1$, or $p < 1$. We conclude that if $(1-\sqrt{\epsilon k})^2 < p < 1$, the system converges. For p to have a solution, we require $(1-\sqrt{\epsilon k})^2 < 1$, or $0 < \epsilon k < 4$.

Combining all of the above cases together, we arrive at the conclusion that if $|p| < 1$ and $0 < \epsilon k < 2+2p$ the system converges. Since we have exhaustively considered all possible cases in the above derivations, this is not only the sufficient condition but also the necessary condition for convergence.

References

Anderson, J.A., Pellionisz, A., & Rosenfield, E. (Eds.), (1990). *Neurocomputing 2: directions for research*. Cambridge, MA: MIT Press.

Churchland, P.S., & Sejnowski, T.J. (1992). *The computational brain*. Cambridge, MA: MIT Press.

Jacobs, R.A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks, 1*, 295–307.

Kleppner, D., & Kolenkow, R.J. (1973). *An introduction to mechanics*. New York: McGraw-Hill.

LeCun, Y., Denker, J.S., & Solla, S.A. (1990). Optimal brain damage. In D.S. Touretzky (Ed.), *Advances in neural information processing systems 2 (NIPS*89)* (pp. 598–605). Denver, CO: Morgan Kaufman.

McClelland, J.L., & Rumelhart, D.E. (1986). *Parallel distributed processing* (Vol. 2). Cambridge, MA: MIT Press.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., & Flannery, B.P. (1992). *Numerical recipes in C*. Cambridge, UK: Cambridge University Press.

Qian, N., & Sejnowski, T.J. (1988). Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol., 202*, 865–884.

Qian, N., & Sejnowski, T.J. (1989). Learning to solve random-dot stereograms of dense and transparent surfaces with recurrent backpropagation. In D.S. Touretzky, G.E. Hinton, & T.J. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School* (pp. 435–443). San Mateo, CA: Morgan Kaufmann.

Rumelhart, D.E., & McClelland, J.L. (1986). *Parallel distributed processing* (Vol. 1). Cambridge, MA: MIT Press.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart, & J.L. McClelland (Eds.), *Parallel distributed processing* (Vol. 1, pp. 318–362). Cambridge, MA: MIT Press.