Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Control on Landscapes with Local Minima and Flat Regions: A Simulated Annealing and Gain Scheduling Approach

Abraham K. Ishihara and Shahar Ben-Menahem

***Abstract*— Landscapes containing local minima and "flat" regions are frequently encountered in multi-layer neural networks that employ sigmoid-like activation function in the hidden layers. Numerous techniques in the neural network community have been proposed to address these issues. In this note, we extend these ideas to the neural network control of nonlinear systems. We propose a solution which employs simulated annealing and a gain scheduled learning rate.**

I. Introduction

Convergence of the backpropagation algorithm and its variants highly depend on the shape of the landscape[1]. It is well known that multilayer neural networks employing sigmoid-like nonlinearities in the hidden layers result in landscapes with local minima [2]. Another type of local minima, in which we term "flat" regions[2] arise due to the fact that the derivative of sigmoid like activation functions tends exponentially to zero. The occurrence of "flat" regions "is one of the main reasons for the slow convergence of standard backpropagation"[6]. Flat regions are not local minima in the technical sense. As pointed out by [2], with sufficient numerical precision, and sufficient time (or number of iterations), backpropagation is not likely to get stuck here. However, the length of time might in fact be crucial to the application at hand especially if it is the real-time control of an uncertain nonlinear system. Hence, in our view, this represents practical local minima in the sense that it may take an impractical amount of time to converge.

Both simulated annealing and gain scheduled or adaptive learning rates have been used before for neural network pattern recognition and system identification [2]. On the other hand, stochastic stability of neural network control systems has been examined in [4], [3]. However, to the best of our knowledge, these techniques have not been combined in the control of uncertain nonlinear systems using multi-layer neural networks. The primary goal of this paper is to connect these methods with rigorous stability proofs required for closed loop control.

A. Ishihara is with the Department of Electrical Engineering, Carnegie Mellon University Silicon Valley, NASA Research Park, Moffett Field, CA 94035 4035, abe.ishihara@west.cmu.edu

S. Ben-Menahem is with the Department of Physics, Stanford University, Stanford CA 94305, U.S.A and Avago Technologies Inc., San Jose CA USA sanverim@gmail.com

[1]The shape of the landscape is independent of the particular learning algorithm.

[2]Flat regions are also know as flat spots, and the phenomenon of slow convergence due to flat spots is often termed premature saturation [7]

II. A Motivating Example

We pose a system identification problem in which the landscape has a local minimum and flat regions. Consider the following desired signal

$$d^*(v^*,t) = a\sigma_1(v^*x(t)) + b\sigma_2(v^*x(t))$$

where $\sigma_1(y) = e^{-\frac{y^2}{2}}$, $\sigma_2(y) = \frac{1}{1+e^{-y}}$, $a, b \in \mathbb{R}$ are known parameters, and $v^* \in \mathbb{R}$ is an unknown weight. We define the estimated signal as[3] $d = a\sigma_1(vx) + b\sigma_2(vx)$, where v is the weight estimate of v^*. We define the error as $e = d^* - d$ and the objective or cost function as $J = \frac{1}{2}e^2$. This determines a landscape, shown in Fig. 1, that associates the cost and the weight vector.

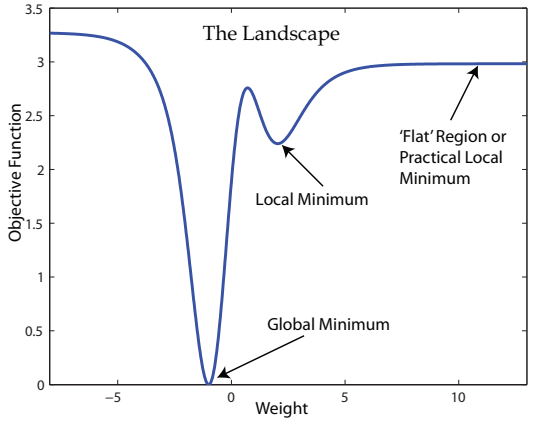

Fig. 1. The Landscape

Computing the gradient of the cost function using the chain rule, we find

$$\frac{\partial J}{\partial v} = -\left(a\sigma_1'(vx) + b\sigma_2'(vx)\right)ex$$

Adjusting the weight, v, according to gradient descent yields

$$\dot{v} = \gamma\left(a\sigma_1'(vx) + b\sigma_2'(vx)\right)ex$$

To illustrate the main idea, we simulate three initial conditions in which the trajectories converge to the global minimum, the local minimum, and a "practical" local minimum. This is illustrated in Fig. 2. This simulation illustrates two

[3]For notational convenience, we will drop the explicit time dependence of $x(t)$ and $v(t)$ and simply write x and v, respectively. It is to be understood that x may depend on time and that v depends on time through a learning rule.

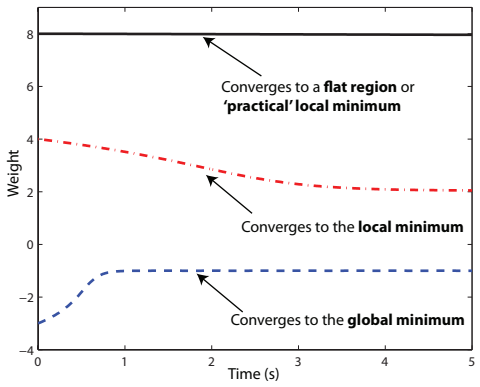

Fig. 2. Trajectories converge to the global minimum, the local minimum, and a 'practical' local minimum using standard backpropagation

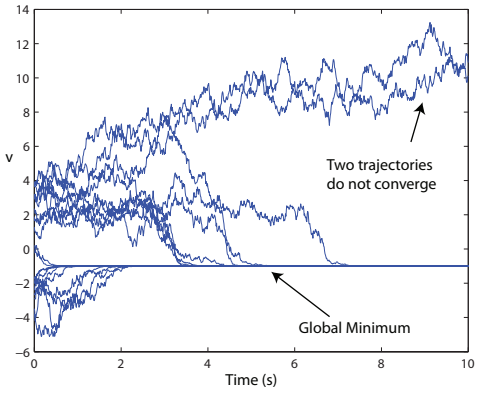

Fig. 3. Learning with Simulated Annealing: Here, we set $\sigma_n = 0.7$. We simulate 20 randomly chosen initial conditions on the interval: $[-4, 4]$.

problems that can arise in high dimensional multilayer neural networks. Under standard gradient descent the trajectory (of the weight vector) may converge to a local minimum depending on its initial condition. If this local minimum is far from the global minima, it is likely that there will be a large performance error. Similarly, if the trajectory is initialized in a "flat" region, then it may become practically stuck there. Since multilayer neural networks have numerous local minima and "flat" regions due to the nonlinearities introduced by activation functions such as sigmoids and RBF's, this scenario is highly likely.

A. Simulated Annealing: Escaping Local Minima

Simulated annealing [1] introduces an additional term in the gradient descent algorithm. This is given by

$$\dot{v} = \gamma \left(a\sigma_1'(vx) + b\sigma_2'(vx) \right) ex + \sigma_n \sqrt{T(t)} \dot{B} \qquad (1)$$

where σ_n represents the noise intensity, $T(t)$ represents temperature, and B represents a Brownian motion process[4]. In the following, we will set $T(t) = e^2(t)$. To illustrate, we simulate 20 randomly chosen initial conditions on the interval: $[-4, 4]$, with $\sigma_n = 0.7$ and $\gamma = 1$. This is illustrated Fig. 3. We observe that most trajectories converge to the global minimum. In fact, all trajectories that went into the local minimum were able to get out and into the global minimum. However, there are two trajectories that appear to be wandering off in the "flat" region. Unlike in the deterministic case where the trajectories in a "flat" region appear to be stuck (see Fig. 2), in this case they appear to be "wandering" away from the global minima. Hence, while it appears that simulated annealing has solved the local minima problem, we now observe that the "flat" regions can be potentially dangerous as the weights not only can be practically stuck but may also go unbounded.

[4]A precise definition of a Brownian motion process is given in Section III-A.

B. Gain Scheduling the Learning Rate: Escaping "Flat" Regions

As discussed previously, there are two trajectories that appear to be wandering off in the "flat" region. In the deterministic case, the trajectory would appear to be stuck as if in a local minimum, and hence we defined this situation as a practical local minimum. When noise is added to the learning dynamics, however, the gradient is essentially zero, and the derivative of the weight evolves according to a scaled Brownian motion process. Hence, we are motivated to make the following transformation which is applicable to both the no noise and noise case. Consider the two transformations $w = f_i(v)$ where

$$f_1(v) = c \tan^{-1}\left(\frac{v}{\alpha}\right) \quad \text{and} \quad f_2(v) = c \tanh\left(\frac{v}{\alpha}\right)$$

The transformations, f_1 and f_2 effectively compress $v = \pm\infty$ to the point given by $w = \pm c\pi$ and $w = \pm c$, respectively. For illustration purposes, we consider only gradient descent, that is $\dot{v} = -\gamma \frac{\partial J}{\partial v}$. Notice that $\left[\frac{\partial f_i}{\partial v}\right]^{-2} \dot{w} = -\gamma \frac{\partial J}{\partial w}$. This motivates the following modified learning rate $\gamma \to \gamma \left[\frac{\partial f_i}{\partial v}\right]^{-2}$ and hence the modified system is given by

$$\dot{\tilde{v}} = -\gamma \left[\frac{\partial f_i}{\partial v}\right]^{-2} \left[a\sigma_1'(vq_d) + b\sigma_2'(vq_d)\right] \tilde{q} q_d + \sigma_n \sqrt{T(t)} \dot{B} \qquad (2)$$

The main idea, is that if v begins to drift away along a "flat" region, the effective learning rate will scale according to $\left[\frac{\partial f_i}{\partial v}\right]^{-2}$. Direct calculation yields

$$\left[\frac{\partial f_1}{\partial v}\right]^{-2} = \left(\frac{\alpha_1}{c_1}\right)^2 \left[1 + \left(\frac{v}{\alpha_1}\right)^2\right]^2 \qquad (3)$$

and

$$\left[\frac{\partial f_2}{\partial v}\right]^{-2} = \left(\frac{\alpha_2}{4c_2}\right)^2 \left(e^{\frac{v}{\alpha}} + e^{-\frac{v}{\alpha_2}}\right)^4$$

In what follows we will set $\alpha_i = c_i$. This ensures that as $\|v\| \to 0$, the effective learning rate tends to the constant,

γ. In Fig. 4, we simulate the gain scheduled learning rates in the deterministic (no-annealing) condition. Of interest is the case when the initial conditions are in the flat region where the effective learning rate in the baseline case is very small. We initialize the weight at $v = 10$ and compare the baseline, f_1 and f_2 cases. We observe that the baseline case converges to a "practical" local minimum in the sense that it will take an impractical amount of time to converge since the gradient in the 'flat' region is essentially zero. Both the gain scheduled learning rates are able to get out of the "flat" region and converge to the local minimum. At this point, they stay there as the gradient is zero. In the simulation we set $\gamma = 1$, $c_1 = \alpha_1 = 1$, and $c_2 = \alpha_2 = 10$.

Remark 1: The gain scheduled learning rules are able to overcome 'flat' regions, but are not able to get out of local minima. On the other hand, simulated annealing is able to overcome local minima but has difficulty in 'flat' regions. This observation is the motivation behind the combined approach proposed in the next section. ◇

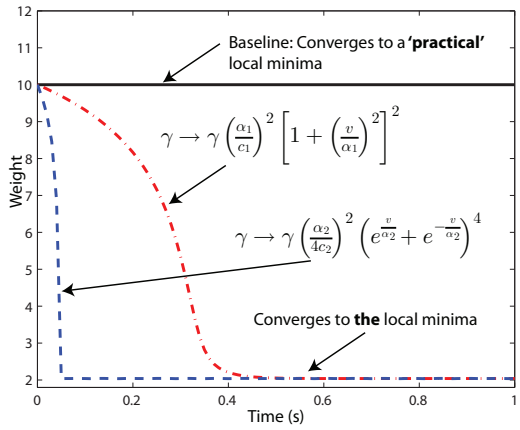

Fig. 4. Comparison of Gain Scheduled Learning Rates

III. Learning and Control

In the previous section, we only considered the learning component, or system identification. In this section, we couple the learning system with a control system.

The Plant: Let the plant be described by the first order nonlinear system:

$$\dot{q}(t) + a\sigma_1(v^* q(t)) + b\sigma_2(v^* q(t)) = \tau(t) \tag{4}$$

where v^* is an unknown parameter. We assume that both a and b are known. The objective of the control input, τ, is to ensure that the state, $q(t)$, track a desired trajectory given by $q_d(t)$. We use the following definitions of the controller, performance error, and parameter error:

$$\begin{array}{ll} \tau = \tau_{ff} + \tau_{fb} & \tilde{q} = q_d - q \\ \tau_{ff} = \dot{q}_d + \bar{\tau}_{ff} & \tilde{v}(t) = v^* - v(t) \\ \tau_{fb} = K_p \tilde{q} & \bar{\tau}_{ff} = a\sigma_1(vq_d) + b\sigma_2(vq_d) \end{array} \tag{5}$$

Note that by Taylor expansion we have

$$\sigma_i(v^* q_d) = \sigma_i(vq_d) + \sigma_i'(vq_d)\tilde{v}q_d + O_i \quad \text{for } i = 1, 2 \tag{6}$$

where $O_i = \mathrm{O}\|\tilde{v}q_d\|^2$. Note that the higher order terms can be bounded as follows:

$$\begin{aligned} O_i &= \sigma_i(v^* q_d) - \sigma_i(vq_d) - \sigma_i'(vq_d)\tilde{v}q_d \\ &\leq 2c_1^{(i)} + c_2^{(i)}\|q_d\|\|\tilde{v}\| \end{aligned} \tag{7}$$

where $c_1^{(i)} = \sup_{y\in\mathbb{R}} \|\sigma_i(y)\|$ and $c_2^{(i)} = \sup_{y\in\mathbb{R}} \|\sigma_i'(y)\|$

Error Dynamics: Using (4), (5) and (6) we have,

$$\begin{aligned} \dot{\tilde{q}} &= -\tau_{fb} + a\left(\sigma_1(v^* q) - \sigma_1(v^* q_d)\right) \\ &\quad + b\left(\sigma_2(v^* q) - \sigma_2(v^* q_d)\right) \\ &\quad + a\sigma_1'(vq_d)\tilde{v}q_d + b\sigma_2'(vq_d)\tilde{v}q_d + aO_1 + bO_2 \end{aligned} \tag{8}$$

A. Simulated Annealing Controller

We now introduce simulated annealing to the learning dynamics as given in (1). The closed loop system becomes:

$$\begin{bmatrix} \dot{\tilde{q}} \\ \dot{\tilde{v}} \end{bmatrix} = \begin{bmatrix} f_1(\tilde{q}, \tilde{v}) \\ f_2(\tilde{q}, \tilde{v}) \end{bmatrix} + H\dot{B} \tag{9}$$

where

$$\begin{aligned} f_1 &= a\left(\sigma_1(v^* q) - \sigma_1(v^* q_d)\right) + b\left(\sigma_2(v^* q) - \sigma_2(v^* q_d)\right) \\ &\quad + a\sigma_1'(vq_d)\tilde{v}q_d + b\sigma_2'(vq_d)\tilde{v}q_d + aO_1 + bO_2 - \tau_{fb} \\ f_2 &= -\gamma\left[a\sigma_1'(vq_d) + b\sigma_2'(vq_d)\right]\tilde{q}q_d + \gamma\kappa\|\tilde{q}\|v \end{aligned} \tag{10}$$

and

$$H = \begin{bmatrix} 0 \\ \sigma_n\sqrt{T(t)} \end{bmatrix} \tag{11}$$

Before proceeding, we state the Ito formula, which is the stochastic version of the chain rule from ordinary calculus. We refer the reader to [5] for further details and background on stochastic differential equations.

Theorem 1 (Ito's Lemma-adapted from [5]): Let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}, P)$ be a complete probability space, and $x(t, \omega)$ be a $\bar{n}$-dimensional Ito process satisfying

$$x(t,\omega) - x(0,\omega) = \int_0^t F(x,s)ds + \int_0^t H(x,s)dB \tag{12}$$

where $F \in \mathcal{L}^1(\mathbb{R}^{\bar{n}} \times \mathbb{R}^+ \times \Omega; \mathbb{R}^{\bar{n}})$, $H \in \mathcal{L}^2(\mathbb{R}^{\bar{n}} \times \mathbb{R}^+ \times \Omega; \mathbb{R}^{\bar{n}\times m})$, and $B : \mathbb{R}^+ \times \Omega \to \mathbb{R}^m$ is an m dimensional Brownian motion. Let $V \in \mathrm{C}^{2,1}(\mathbb{R}^{\bar{n}} \times \mathbb{R}^+; \mathbb{R})$. Then,

$$V(x(t),t) - V(x(0),0) = \int_0^t \bar{F}ds + \int_0^t \bar{H}dB \quad \text{a.s}$$

where $\bar{F} = V_t + V_x F + \frac{1}{2}\mathrm{Tr}\left(H^T V_{xx} H\right)$ and $\bar{H} = V_x H$. The notation, V_x denotes the gradient of V, and V_{xx} denotes the Hessian of V.

For brevity, we define the operator L acting on the Lyapunov function V as

$$LV := V_t + V_x F + \frac{1}{2}\mathrm{Tr}\left(H^T V_{xx} H\right)$$

We will also need the following definition. Let

$$\dot{x} = F(x,t) + H(x,t)\dot{B} \tag{13}$$

Definition 1: The solution to the stochastic differential equation given in (13) is said to be semi-global uniformly

bounded in expectation if for any given (arbitrarily large) set of initial conditions: $x_0 \in B_k$, we have

$$E\|x(t)\| \leq \beta \quad \forall t \geq t_0$$

where β is a constant that may depend on k.

Theorem 2 (SGUBE): Let F and G in (13) satisfy the local Lipschitz and linear growth condition described in [5] (page 51). Let $V \in \mathrm{C}^{2,1}(B_h \times \mathbb{R}^+; \mathbb{R})$, and $\|x(t_0)\| \leq k < h$. Let the radius, $r \in (0, h)$. Denote the exterior of the half cylinder as $\Omega_{r,t_0} := \{(x,t) \in B_h \times \mathbb{R}^+ : \|x(t)\| > r \ \& \ t \geq t_0\}$. Let V be positive definite and decrescent on Ω_{r,t_0}. That is, there exists class K functions, μ_1 and μ_2 such that $\mu_1(\|x\|) \leq V(x,t) \leq \mu_2(\|x\|)$ for all $(x,t) \in \Omega_{r,t_0}$. Suppose, in addition, that μ_1 is convex on $[0,h)$. If $LV \leq 0$ for all $(x,t) \in \Omega_{r,t_0}$, then, the expected value,

$$E(\|x\|) \leq \beta \ \ \forall t \geq t_0$$

where $\beta = \max\{r, \mu_1^{-1}(\mu_2(r)), \mu_1^{-1}(\mu_2(k))\}$.

Remark 2: Due to page limitations, we do not provide the full proof as it follows very closely to the proof of the deterministic version first proposed by [8].

Theorem 3: Let the plant be described by (4) with control input, performance error and parameter error defined in (5). Let the learning dynamics evolve according to

$$\dot{v} = \gamma \left[a\sigma_1'(vq_d) + b\sigma_2'(vq_d)\right] \tilde{q} q_d - \gamma\kappa\|\tilde{q}\| v + \sigma_n \sqrt{T(t)} \dot{B}$$

where $\gamma, \kappa > 0$. Suppose the ideal weight v^* satisfies $\|v^*\| \leq v_b$ where $v_b > 0$ is a known upper bound. Then, if

$$K_p > \left(a c_2^{(1)} + b c_2^{(2)}\right) v_b + \frac{1}{2}\gamma^{-1}\sigma_n^2$$

and the temperature is set to $T(t) = \|\tilde{q}\|^2$, the closed loop system is semi-globally uniformly bounded in expectation (SGUBE).

Proof: The closed loop system is given by

$$\dot{x} = F(x,t) + H(x,t)\dot{B}$$

where $x = [\tilde{q} \ \tilde{v}]^T$, $F = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ and f_i and H are given in (9)-(11). We have

$$LV = V_x F + \frac{1}{2}\mathrm{Tr}\left[V_{\tilde{v}\tilde{v}} H H^T\right]$$

The term on the right hand side becomes

$$\frac{1}{2}\mathrm{Tr}\left[V_{\tilde{v}\tilde{v}} H H^T\right] = \frac{1}{2}\gamma^{-1}\sigma_n^2 T(t)$$

where $T(t) \geq 0$. Since $T(t) = \|\tilde{q}\|^2$ we compute

$$\begin{aligned} LV \leq \ & -\|\tilde{q}\| \left\{ \left(K_p - r_2\|v^*\| - \tfrac{1}{2}\gamma^{-1}\sigma_n^2\right)\|\tilde{q}\| \right. \\ & + \left(\sqrt{\kappa}\|\tilde{v}\| - \tfrac{\kappa\|v^*\| + r_2\|q_d\|}{2\sqrt{\kappa}}\right)^2 \\ & \left. - \left[\left(\tfrac{\kappa\|v^*\| + r_2\|q_d\|}{2\sqrt{\kappa}}\right)^2 + 2r_1\right] \right\} \end{aligned}$$

Hence, if the feedback gain is selected such that

$$K_p > r_2 v_b + \frac{1}{2}\gamma^{-1}\sigma_n^2$$

it follows by Theorem 2, that the closed loop system is SGUBE. ■

B. The Simulated Annealing and Gain Scheduling Controller

We now present the main results of this note in which we combine the simulated annealing controller with a gain scheduled learning rate as describe in Section II-B. As we would like our results to hold for a time-varying desired trajectory, it is necessary to slightly modify the gain scheduler[5] as follows:

$$\left[\frac{\partial f_1}{\partial v}\right]^{-2} = \left(\frac{\alpha_1}{c_1}\right)^2 \left[1 + \left(\frac{v q_d}{\alpha_1}\right)^2\right]^2$$

The only modification is scaling v by q_d. This becomes necessary to obtain a uniform bound as q_d becomes small.

Theorem 4: Let the plant be described by (4) with control input, performance error and parameter error defined in (5). Let the learning dynamics evolve according to simulated annealing and gain scheduling algorithm given by

$$\begin{aligned} \dot{v} \ = \ & \gamma \left(1 + \left[\frac{\partial f_1}{\partial v}\right]^{-2}\right) \left[a\sigma_1'(vq_d) + b\sigma_2'(vq_d)\right] \tilde{q} q_d \\ & - \gamma\kappa\|\tilde{q}\| v + \sigma_n \sqrt{T(t)} \dot{B} \end{aligned}$$

where $\gamma, \kappa > 0$. Suppose the ideal weight v^* satisfies $\|v^*\| \leq v_b$ where $v_b > 0$ is a known upper bound. Then, if

$$K_p > \left(a c_2^{(1)} + b c_2^{(2)}\right) v_b + \frac{1}{2}\gamma^{-1}\sigma_n^2$$

and the temperature is set to $T(t) = \|\tilde{q}\|^2$ the closed loop system is semi-globally uniformly bounded in expectation (SGUBE).

Proof: With the modified learning rule given above, we have

$$\begin{aligned} & V_x F(x) \leq -\left(K_p - r_2\|v^*\|\right)\|\tilde{q}\|^2 + 2r_1\|\tilde{q}\| + r_2\|q_d\|\|\tilde{q}\|\|\tilde{v}\| \\ & + \tilde{v}\left(-\left(1 + \left[\tfrac{\partial f_1}{\partial v}\right]^{-2}\right)\left[a\sigma_1'(vq_d) + b\sigma_2'(vq_d)\right]\tilde{q} q_d \right. \\ & \left. + \kappa\|\tilde{q}\| v + \left[a\sigma_1'(vq_d) + b\sigma_2'(vq_d)\right]\tilde{q} q_d \right) \end{aligned}$$

Using the uniform bound established in (14) in the Appendix, we obtain

$$\begin{aligned} LV \ \leq \ & -\|\tilde{q}\| \left\{ \left(K_p - r_2\|v^*\| - \frac{1}{2}\gamma^{-1}\sigma_n^2\right)\|\tilde{q}\| \right. \\ & + \left(\sqrt{\kappa}\|\tilde{v}\| - \frac{\kappa\|v^*\| + r_2\|q_d\| + C_{f_1}\|q_d\|^2}{2\sqrt{\kappa}}\right)^2 \\ & \left. - \left[\left(\frac{\kappa\|v^*\| + r_2\|q_d\| + C_{f_1}\|q_d\|^2}{2\sqrt{\kappa}}\right)^2 + 2r_1\right] \right\} \end{aligned}$$

[5]For the remaining development, we only consider the case for which $f_i = f_1$. The f_2 case is very similar.

As before, if the feedback gain is selected such that

$$K_p > r_2 v_b + \frac{1}{2}\gamma^{-1}\sigma_n^2$$

it follows by Theorem 2, that the closed loop system is SGUBE. ■

IV. Simulation Examples

In this section, we illustrate the advantages of the simulated annealing and gain scheduling controller.

Example 1: (*Baseline and Simulated Annealing Controller: Performance Comparison*). We first compare the baseline and simulated annealing controller. This is shown in Fig. 5. In the top panel, we plot the performance with respect to tracking a step input. We observe that the baseline controller has a large steady state error. The error in this case is due to the fact that the weight has converged to the local minimum around $v = 2$. This is shown in the bottom panel. The simulated annealing controller is able to 'escape' from the local minima due to the temperature scaled Brownian motion. Once out of the local minimum, both the performance error and the weight estimate converge to the correct solution.

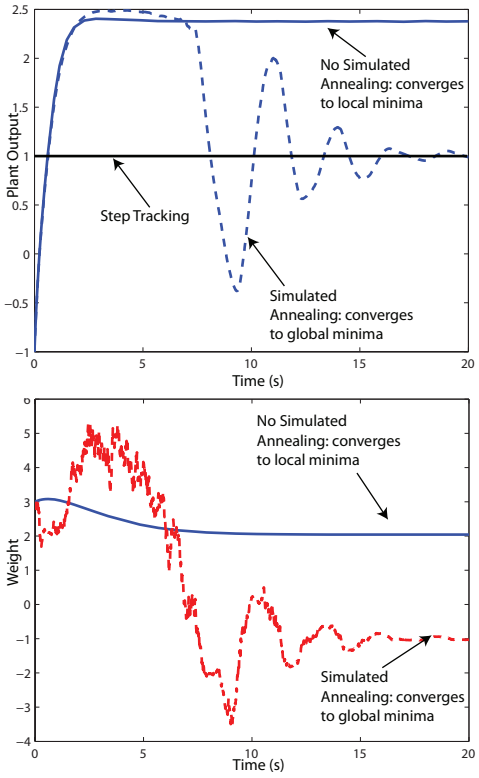

Fig. 5. Nominal and Simulated Annealing Controller

Example 2: (*Baseline, Simulated Annealing - only, Gain Scheduling - only, and Simulated Annealing and Gain Scheduling Controller: Performance Comparison*). In Example 1, the simulated annealing controller was able to get out of the local minimum and converge to the global minimum. This is the main point of simulated annealing. However, as pointed out earlier, this may not be the case if the initial conditions are such that the weights begin in flat regions, or if the initial error in performance drives the weights to a flat region. There are numerous ways in which weights may end up in "flat" regions. Hence, the potential to become practically stuck is an important limitation of using simulated annealing alone. In this example, we initialize the weight in a "flat" region at $v(t_0) = 8$. The simulation is shown in Fig. 6 where the evolution of the performance and weights under the various control scenarios are depicted in the top and bottom panel, respectively. Here, the landscape is very flat, and hence it's gradient is very small. We observe that the baseline controller has converged to *a* practical local minimum in the sense that it would take an impractical amount of time to converge. Furthermore, even if it did converge, it would converge to *the* local minimum, and not the global minimum. The simulated annealing controller also does not converge. In the bottom panel, we observe the weight bouncing around in the flat region. The performance is similar to the baseline controller as the saturation effect of the nonlinearities in this region makes the contribution to the plant dynamics nearly identical. The gain scheduling controller is able to overcome the "flat" region, but becomes stuck in the local minima. The performance is slightly better than the baseline and simulated annealing controllers. Finally, the combined *simulated annealing and gain scheduling* controller is able to not only escape the "flat" region, but also the local minimum as well. We observe that the weight converges to the global minima, and the performance error tends to zero. Notice the coupled error and weight dynamics as the weight begins to descend down to the global minimum. There are times for which $v = v^*$ yet it does not remain there. The reason is due to the effective time constant of the error dynamics that drives the Brownian motion process.

V. Discussion

It is well known that local minima present significant challenges in the control of uncertain nonlinear systems with large scale, multi-layer neural networks. Similarly, "flat" regions, which arise due to the properties of commonly used activation functions such as sigmoids, can significantly degrade performance relegating online neural network controllers to an impractical solution. The model we chose is the simplest in the following sense. The simplest possible neural network is a one parameter net with an activation function. The simplest possible plant is a first order plant. Despite its simplicity, we have shown that standard neural network control methods in this situation will fail. The reason why they fail is due to the fact that the weights become stuck in the local minima and "flat" regions.

The simulated annealing and gain scheduling controller combined the best of both controllers described above. We showed that not only would the controller escape from the flat region, but also would escape the local minima and converge to the global minima. The performance was shown to be far superior to the baseline case. Furthermore,

LEARNING METHOD	Solves *Local Minima?*	Solves *"Flat" Regions?*	Guarantees Boundedness?
Simulated Annealing	yes	no	no
Gain Scheduling	no	yes	no
Regularization	no	no	yes
Proposed Method	yes	yes	yes

TABLE I
COMPARISON OF METHODS

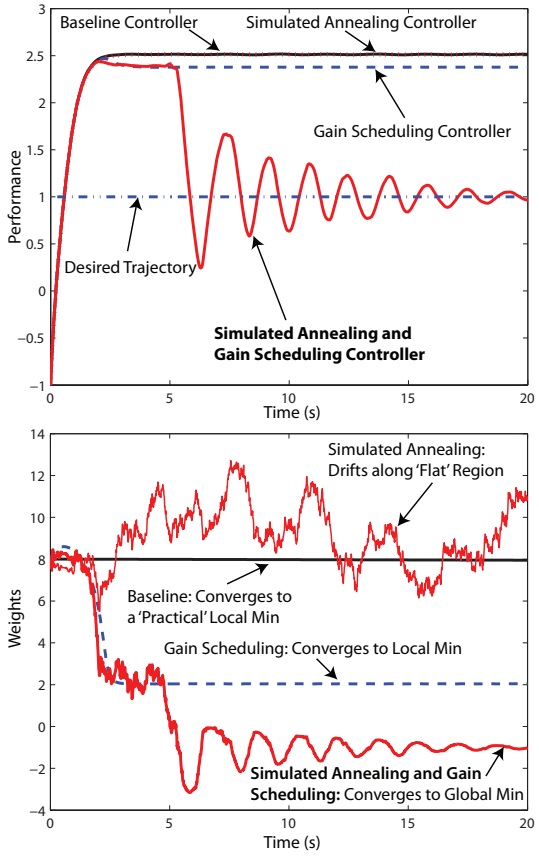

Fig. 6. Performance Comparison of (1) Baseline, (2) Simulated Annealing, (3) Gain Scheduling, and (4) Simulated Annealing and Gain Scheduling Controllers: For cases (1)-(4), we plot the plant performance in the top panel, and the evolution of the weights under the various learning schemes in the bottom panel.

all the methods proposed were encompassed by rigorous stability proofs which are not common in the neural networks literature. It is often the case that neural nets are trained offline, and the resulting network is then fixed and used in the control of a plant. However, in this case, we seek a neural net controller implemented online that has the ability to track time varying or abrupt changes in the plant parameters. This added complexity requires the use of Lyapunov like stability proofs that guarantee the combined dynamics result in a stable system.

Up to this point, we have not discussed regularization. This technique is often employed in neural networks when there exists prior information. In control, this technique is used to overcome the requirements of persistence of excitation. We note that regularization is very different than the gain scheduling method proposed earlier. In all of the proposed controllers, the regularization term was given as $-\gamma\kappa\|\tilde{q}\|v$. This has the effect of driving the weights to the origin for non-zero $\tilde{q}$. Note that this is independent of the gradient. On the other hand, the gain scheduling term 'kicks in' when the weights become large, thus preventing drift in a 'flat' region. However, it does not necessarily drive the weights to the origin. Note that it is possible that the regularization term can lead to an incorrect solution. On the one hand, it ensures boundedness of the weights, but on the other can drive them to an incorrect value resulting in bounded, but poor performance. These observations are summarized in Table I.

APPENDIX

Uniform Bounds on the Product: It can be shown that

$$\left\| \left[\frac{\partial f_1}{\partial v} \right]^{-2} \left[a\sigma_1'(vq_d) + b\sigma_2'(vq_d) \right] q_d \right\| \leq C_{f_1} \|q_d\|^2 \quad (14)$$

where

$$C_{f_1} = a \left[e^{-12} + 2\frac{3^{\frac{3}{2}}}{\alpha_1^2} e^{-\frac{3}{2}} + \frac{5^{\frac{5}{2}}}{\alpha_1^4} e^{-\frac{5}{2}} \right]$$
$$+ b \left[1 + 2\left(\frac{2}{\alpha_1}\right)^2 e^{-2} + \left(\frac{4}{\alpha_1}\right)^4 e^{-4} \right]$$

REFERENCES

[1] S. Geman and C.-R. Hwang. Diffusions for global optimization. *SIAM Journal on Control and Optimization*, 24(1):1031–1043, 1986.
[2] Marco Gori and Alberto Tesi. On the problem of local minima in backpropagation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(1):76–86, 1992.
[3] A.K. Ishihara, J. van Doornik, and S. Ben-Menahem. Feedback error learning with a noisy teacher. *Proceedings of the American Control Conference (ACC)*, pages 4529–4534, 2008.
[4] A.K. Ishihara, J. van Doornik, and S. Ben-Menahem. Neural network robot control with noisy learning. *Proceedings of the 17th International Federation of Automatic Control (IFAC)*, pages 11708 – 11713, 2008.
[5] X. Mao. *Stochastic Differential Equations and Applications*. Horwood, Chichester, 1997.
[6] M. Pfister and R. Rojas. Speeding-up backpropagationa comparison of orthogonal techniques. pages 517–523, 1993.
[7] Javier E. Vitela and Jaques Reifman. Premature saturation in back-propagation networks: Mechanism and necessary conditions. *Neural Networks*, 10:721–735, 1997.
[8] T. Yoshizawa. Stability and boundedness of systems. *Arch. Rational Mech. Anal.*, 6:409421, 1960.