

Shocking: An Approach to Stabilize Backprop Training with Greedy Adaptive Learning Rates

J. A. Janét, S. M. Scoggins, S. M. Schultz, W. E. Snyder, M. W. White and J. C. Sutton, III

Department of Electrical and Computer Engineering

North Carolina State University

Raleigh, NC 27695-7911, USA

E-MAIL: jajanet@eos.ncsu.edu

Abstract

In general, backprop neural networks converge faster with adaptive learning rates than with learning rates that remain constant or grow or decay without regard to the network error (such as exponentially decaying learning rates). This is because each synapse has its own learning rate that can vary over time by an amount appropriate to that weight. For certain problems however, adaptive learning rates cause neural networks to saturate during training. The rate of occurrence of this problem is increased when the learning rates can grow without limit. When learning rates are permitted to assume values greater than unity, they are considered "greedy." Greedy adaptive learning rates can reduce the training times of networks, but can also compromise the stability of the training process, leading to a network that fails to converge. Most all comparisons of training time are based on neural networks that actually converged. Rarely, if at all, is the failure rate presented; little to no consideration is given to why some neural networks fail to converge or, for that matter, how to reduce the chances of failure. This paper proposes a simple ad hoc approach called "shocking" as a partial solution to the instability problem caused by greedy adaptive learning rates. An analysis based on training times and failure rates for two inherently unstable benchmark problems is used to validate the use of shocking.

1 Introduction

It has been shown that "the method of adaptive learning rates is much faster than steepest descent, generally reducing training time by an order of magnitude, and it is also very dependable. It is not prone to get into trouble and does not require special care...[It] is fast, dependable, and highly automatic..." [14]. The adaptive learning rate modification proposed by Jacobs [3] has become popular for two main reasons: minimal mathematical complexity and numerous reported successes at achieving faster convergences.

The adaptive learning rate model is based on four heuristics that suggest that each weight of a neural network should have its own learning rate and that these rates be allowed to change over time. Qualitatively, the heuristics are: 1) Every parameter of the performance measure should have its own individual learning rate; 2) Every learning rate should be allowed to vary over time; 3) When the derivative of a parameter possesses the same sign for consecutive time steps, the learning rate for that parameter should be increased; and 4) When the sign of the derivative of a parameter alternates for consecutive time steps, the learning rate for that parameter should be decreased.

Despite their success at reducing training times, adaptive learning rate models tend to create instabilities which can cause a neural network to saturate¹ [2]. The standard backprop model estimates the error associated with a synaptic weight to calculate a weight change. Typically, the synaptic weight is updated by only a fraction (less than unity) of the calculated weight change. While this suggests an inherent stability in the overall weight update method, it also causes training times to be slow. On the other hand, updating a synaptic weight by more than the prescribed weight change (learning rate greater than unity) can significantly reduce training times. However, this "greedy" approach increases the likelihood of instabilities.

There are many parameters that determine a neural network's training time and tendency to fail. Some of these include: 1) the type of problem being solved; 2) the architecture size; 3) the initial synaptic weights; 4) the initial learning rates; 5) the maximum learning rate; and so on. In this instance we are interested in the effects of shocking on two random variables: training time $T \in [0, \infty)$ and failure $F = \{0, 1\}$ of an architecture to converge in less than τ_F epochs. In this paper, we use two inherently unstable benchmark problems: 1) the XOR problem; and 2) the XOP problem (a simple character recognition problem with 'x', 'O' and '+'). Many other

¹Saturation puts a neural network on a portion of the error surface that is difficult, if not impossible, to recover from.

large-scale problems have taken advantage of shocking, as well [1, 4, 5, 6, 7, 8, 9].

2 The Modified Adaptive Learning Rate Model

Under the adaptive learning rate model, each synaptic weight in a neural network architecture is allowed to have its own learning rate. We are concerned with the direction in which errors for a synaptic weight decrease over an exponential average f_ω ,

$$f_\omega(0) = 0$$

$$f_\omega(k+1) = \theta \cdot f_\omega(k) + (1-\theta)d_\omega(k) \quad (1)$$

where $d_\omega(k) = \sum_q^Q \frac{\partial \hat{V}_q(k)}{\partial \omega(k)}$, that is the change in weight w prescribed by the current iteration of backprop training, and θ defines the weighting of consecutive error associations ($\theta \approx 0.1$). The signs of d_ω and f_ω give a precise measure of the direction in which the error decreases both currently and recently. The equation for the changing value of the learning rate for a synaptic weight, is

$$\alpha_\omega(k+1) = \begin{cases} \alpha_\omega(k) \cdot \kappa & \text{if } d_\omega(k)f_\omega(k) > 0 \text{ and} \\ & \alpha_\omega(k) < \tau \\ \alpha_\omega(k) \cdot \phi & \text{if } d_\omega(k)f_\omega(k) \leq 0 \end{cases} \quad (2)$$

where typically ($\kappa \approx 1.1$) and ($\phi \approx 0.5$).

One important difference between equation 2 and the adaptive learning rate method in [3] should be noted; instead of *adding* $\kappa \approx 0.1$ to the learning rate, we *multiply* $\kappa \approx 1.1$ [15]. We do this because architectures with many synapses can saturate if their learning rates are too high. Extremely large architectures can have initial learning rates on the order of $\alpha_\omega(0) \approx 0.0001$. Hence, an *addition* of $\kappa \approx 0.1$ to α_ω would have too radical an effect on the back-propagation process. Instead, a subtle 10% increase of the learning rate each epoch reduces the tendency toward saturation regardless of the initial learning rate magnitude [15].

Another detail from equation 2 not mentioned in [3] is that an upper bound τ must be placed on α_ω . Without requiring that $\alpha_\omega(k) < \tau$ when increasing the learning rate, if $d_\omega(k)f_\omega(k)$ is consistently positive the learning rate can grow to infinity. Figure 1 shows the learning rate change according to equation 2 without imposing a learning rate limit, τ , for three different initial learning rates. Typically $\tau \leq 1$ since it is commonly desired that ω not change radically based on the back-propagated error (particularly in later stages of training). However, in the spirit of proportional- and derivative-controllers, learning rates can anticipate future weight changes (i.e., become *greedy*) if ($\tau > 1$). The expected result is an even faster convergence to the global minimum. Again, with

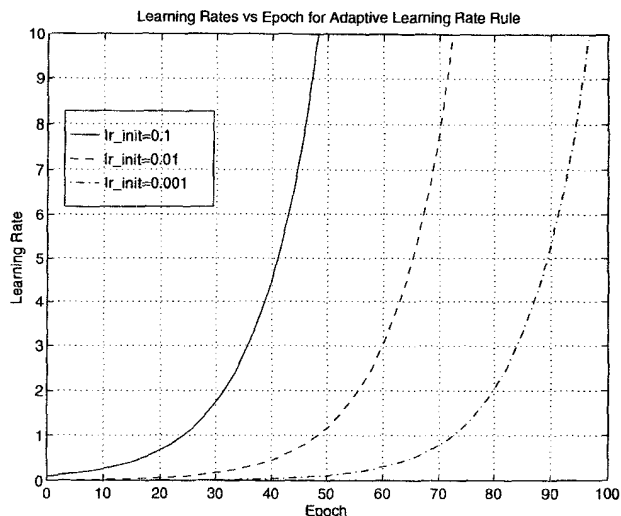


Figure 1: Without an upper boundary, learning rates can grow to infinity.

this greed comes the risk of saturation since excessively large learning rates can make synaptic weights overshoot global minima to a point on the error surface from which the neural network can not recover.

3 Heuristic Conditions for Shocking Adaptive Learning Rates

Our research indicates that it *is* possible to let learning rates become greedy ($\tau > 1$) and still maintain stable convergence. This is accomplished through an *ad hoc* approach called “shocking” [1, 4]. Simply stated, shocking a neural network reduces all synaptic learning rates to small values. The two conditions for shocking are heuristic, yet they can be justified and they have a proven significant impact on the reduction of failure rates. The first heuristic condition stipulates that *if the training error at epoch $k+1$ increases to a value larger than the error at epoch k , the neural net should be shocked*. See Figure 2a. Reverting to small learning rates gives the neural net the opportunity to quickly (re)turn to its original destination or, due to the instability that triggered the shock, locate a better minimum on the error surface.

The second heuristic condition for shocking requires that if the learning rates are large enough to significantly impact training, but the training error is decreasing at a very slow rate $\frac{\Delta \hat{V}}{\Delta k}$, the neural net should be shocked. See Figure 2b. When the learning rates are very large, it is possible for the synaptic weights to overshoot and vacillate over a minimum while still very slowly converging and not causing the neural net to saturate [15]. As a part of this condition we specify that the learning rates should be large because they need time to grow enough to significantly impact the convergence. Hence, we can

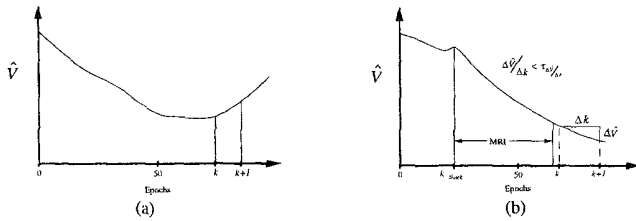


Figure 2: Shock conditions (a) training error increases from minima; and (b) slow convergence.

restate the second condition as: *if a number of epochs have elapsed $\Delta k = k - k_{shock}$ since the last time the net was shocked, and $\Delta k \geq MRI$ (minimum reset interval), and the training error is decreasing at a rate less than a specified minimum reset slope, $\frac{\Delta \hat{V}}{\Delta k} \leq \tau \frac{\Delta \hat{V}}{\Delta k}$, then the neural net should be shocked.*

It has been observed that shocking under these conditions significantly reduces the chances of saturation and, hence, stabilizes greedy adaptive learning. For neural networks that converge, the average training times with shocking will typically not be less than the average training times for neural networks without shocking. This is expected since the conditions for shocking are sensitive to instabilities regardless of magnitude. Hence, the neural network loses some momentum each time the learning rates are reset. But the advantage of shocking lies more in its effectiveness at improving a neural network's probability of convergence even when the learning rates become very large.

4 The Effects of Shocking Adaptive Learning Rates

In this section we: 1) study the effects of placing an upper limit on adaptive learning rate values; and 2) assess the effects of learning rate shocking. To reiterate, shocking is simply the act of resetting all synaptic learning rates to small values when one of the two conditions discussed in Section 3 occurs. The two random variables measured in this comparison are training time $T \in [0, \infty)$ and failure $F = \{0, 1\}$ of an architecture to converge in less than τ_F epochs. While there are many parameters that contribute to a neural network's convergence rate, we allow only three parameters to vary: 1) the maximum learning rate $MLR = \{1, \dots, 20\}$; 2) the minimum reset interval $MRI = \{50, \dots, 200 \text{ epochs}\}$; and 3) the minimum reset slope $MRS = \{0.01, \dots, 0.5 \text{ per epoch}\}$ of the training error per epoch.

4.1 The XOR Problem

The XOR problem is a commonly used benchmark problem due to the inherent challenge it poses to neu-

ral network training. An XOR neural network, despite its relative simplicity, does not always converge. This inherent instability can be attributed to initial weights, learning rates, and weight update algorithms. Our goal here is to determine if shocking helps reduce the number of failed convergences in the XOR network.

The architecture used in this problem has the standard two-input, two-hidden layer neuron, single-output layer neuron configuration (2:2:1). The analysis for this problem is based on 1000 2:2:1 XOR neural nets whose input-to-hidden synaptic weights are randomly initialized to values between ± 0.1 and whose hidden-to-output synaptic weights are randomly initialized to values of ± 1 . The initial learning rate for all synaptic weights is 0.9. These parameters might or might not be "optimal". But, due to their consistency and abundance, they provide a good basis for observing the random variables training time T and failure F of an architecture to converge in less than $\tau_F = 10000$ epochs.

4.1.1 Adaptive Learning Without Shocking

We first look at the collective effects of varying the maximum learning rate $MLR = \{1, \dots, 20\}$. Figure 3 shows that the average training time and variance for XOR networks that converged reduced significantly with increased MLR . The lowest average training time was 240 epochs. But the number of failed convergences ranged from 2 to 33 with a trend that increased seemingly without bound as the maximum learning rate increased.

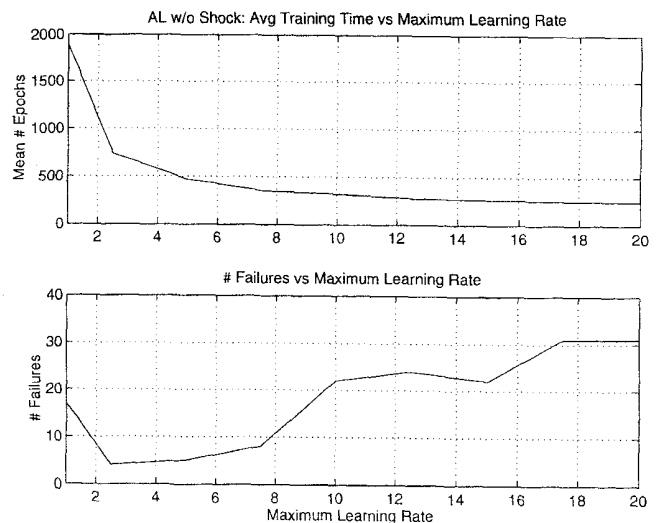


Figure 3: T_{avg} and F per 1000 XOR neural nets without shocking.

4.1.2 Adaptive Learning With Shocking

Looking now at figures 4 through 6 it can be seen that the effects of varying the three parameters $MRI =$

{50, ..., 200 epochs}, $MRS = \{0.01, \dots, 0.5$ per epoch}; and $MLR = \{1, \dots, 20\}$ generally reduced the average training times and variances with increased MLR and MRI . The number of failures ranged from 1 to 21 and, as before, the trend of failures seemed to increase seemingly without bound as all three parameters increased. But what is clearly significant here is that the worst case failure rate for neural networks that were shocked is 36% lower than the worst case failure rate for neural networks that were not shocked. That is, shocking stabilized as much as 36% of the neural nets that would have otherwise failed under training conditions without shocking. Furthermore, shocking allowed for a greater range of maximum learning rates than its rival model without shocking. Finally, it can be seen that all three parameters play a significant role in the training time and failure rate of a neural network.

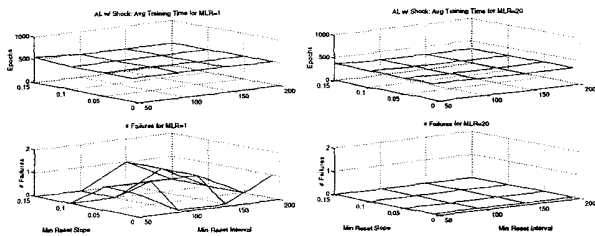


Figure 4: T_{avg} and F per 1000 XOR neural nets with shocking. (a) $MLR = 1$. (b) $MLR = 20$.

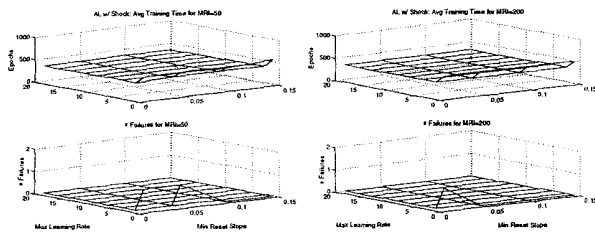


Figure 5: T_{avg} and F per 1000 XOR neural nets with shocking. (a) $MRI = 50$. (b) $MRI = 200$.

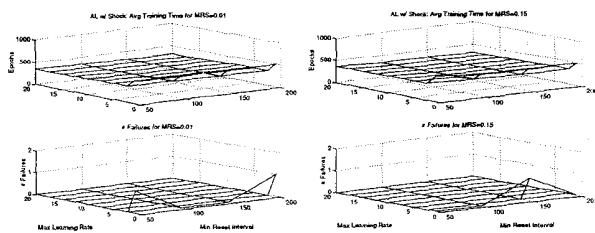


Figure 6: T_{avg} and F per 1000 XOR neural nets with shocking. (a) $MRS = 0.01$. (b) $MRS = 0.5$.

4.2 The XOP Problem

The XOP problem is a simple character recognition problem where the input vector is two-dimensional. Specifically, within a 5×6 pixel binary image one of three 3×3 characters can appear: 1) 'x'; 2) 'O'; and 3) '+'. It is assumed that the character can be anywhere inside the input image and that only one character will be in the input image at a time. The architecture used to solve the XOP problem uses 3×3 receptive fields[11, 12, 13] and 3 output neurons. Figure 7 shows some examples of XOP character configurations.

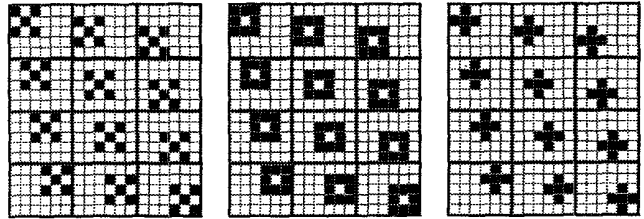


Figure 7: Uncorrupted character configurations in the XOP problem.

Like the XOR problem, the XOP problem is inherently unstable. In fact, we use more than just the shocking heuristic to stabilize training on the XOP problem with the aforementioned architecture. Specifically, it was determined that nearly *all* of the randomly initialized XOP architectures failed to converge unless we coupled the hidden-to-output synaptic weights[10]. The analysis for this problem is based on 50 XOP neural nets² whose input-to-hidden synaptic weights are randomly initialized to values between ± 0.001 and whose hidden-to-output synaptic weights are randomly initialized to values between ± 0.0001 . The initial learning rate for all synaptic weights is 0.4. Again, these parameters might or might not represent “optimal” conditions for XOP neural nets. But, they do provide a good basis for observing and modelling the random variables training time T and failure F of an architecture to converge in less than $\tau_F = 3000$ epochs.

4.2.1 Adaptive Learning Without Shocking

By varying the maximum learning rate $MLR = \{1, \dots, 20\}$, Figure 8 shows that the average training time for neural nets that converged reduced significantly with increased MLR . The lowest average training time in this case was 417 epochs. The number of failed convergences ranged from 2 to 6. No clear trend can be extrapolated from the failure rate plot. However, it is clear that within the range of $MLR = \{1, \dots, 20\}$ at least 4% of the XOP neural nets failed to converge.

²We used fewer neural nets for the XOP problem because the training time is significantly longer than that for the XOR problem.

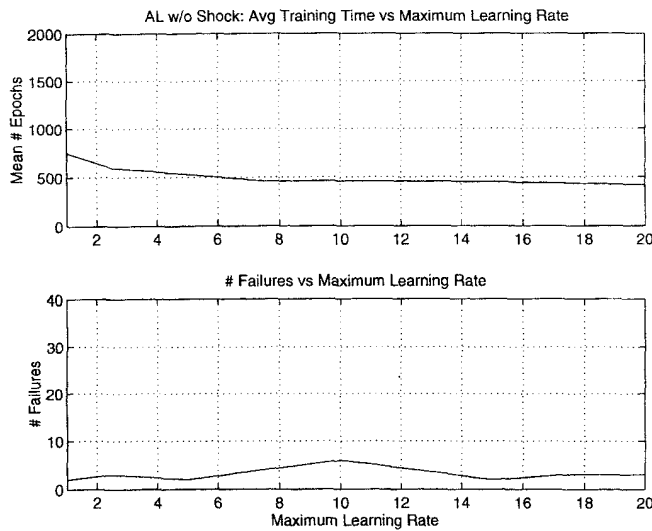


Figure 8: T_{avg} and F per 50 XOP neural nets without shocking.

4.2.2 Adaptive Learning With Shocking

Figures 9 through 11 show the effects of varying the three parameters $MRI = \{50, \dots, 200 \text{ epochs}\}$, $MRS = \{0.01, \dots, 0.5 \text{ per epoch}\}$; and $MLR = \{1, \dots, 20\}$ generally reduced the average training times with increased MLR and MRI . For each combination, there were no failures. In fact, for all of the combinations where $MLR > 1$, there were no failed convergences. So shocking stabilized up to 100% of the XOP neural nets that might have otherwise failed without shocking.

5 Other Applications of Shocking

In addition to increasing performance on benchmark problems, shocking has been shown to benefit real world problem-solving situations using neural networks. Outdoor landmark recognition (OLR) for autonomous vehicles[4], global self-localization (GSL) from sensor data for autonomous robots[4, 5, 7], and the transfer of learned knowledge between robots[5, 6, 8] have all used shocking to add stability to the training process. Specifically, the OLR problem had a 50×50 input data space, up to four hidden layer neurons (each with 5×5 receptive fields), and anywhere from 3 to 8 output neurons. The sensor pattern recognition problem was performed in simulation as well as on actual data generated by two differently configured mobile robots. Architectures were sized to accommodate 30×30 input data spaces, up to four hidden layer neurons, and 10 output neurons. The knowledge transfer problem was addressed in both the OLR and GSL domains. Knowledge transfer involves using previously learned components from one neural network to expedite the training of another network when

new classes are added. All of these applications involved training complex networks on large amounts of multidimensional data. For this reason, training times, and the cost of a network failing to converge, are high. Shocking, while heuristic in nature, seems not only likely to benefit many poor-case problems, but also appears unlikely to compromise the performance of best-case training.

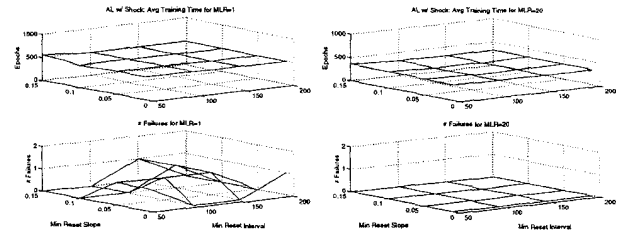


Figure 9: T_{avg} and F per 50 XOP neural nets with shocking. (a) $MLR = 1$. (b) $MLR = 20$.

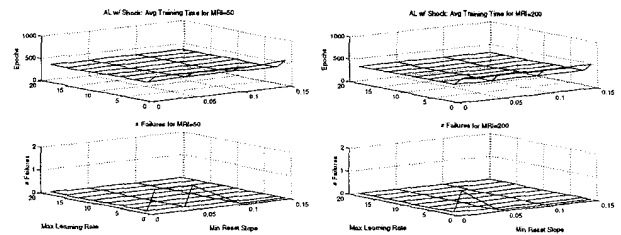


Figure 10: T_{avg} and F per 50 XOP neural nets with shocking. (a) $MRI = 50$. (b) $MRI = 200$.

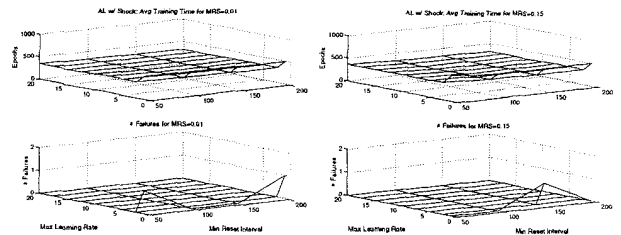


Figure 11: T_{avg} and F per 50 XOP neural nets with shocking. (a) $MRS = 0.01$. (b) $MRS = 0.5$.

6 Summary on the Effects of Shocking

For neural networks that converge, the average training times with shocking will typically not be less than the average training times for neural networks without shocking. This is expected since the conditions for shocking are sensitive to instabilities (increases in error) regardless of magnitude. Hence, the neural network loses some momentum each time the error increases slightly and the shocking heuristic resets the learning rates. The

gains from shocking come in the form of increased likelihood of convergence, even when learning rates become very large. The cost of a few extra epochs usually outweighs the cost of failed convergence. But it is believed that many variations of shocking could be employed that might yield even faster training times and lower failure rates.

References

- [1] M. Azam, H. Potlapalli, J. A. Janét and R. C. Luo, "Outdoor Landmark Recognition Using Segmentation, Fractal Model and Neural Networks." *Proc. ARPA Image Understanding Workshop*, 1996.
- [2] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Co., NY, 1994.
- [3] R. A. Jacobs. "Increased Rates of Convergence Through Learning Rate Adaptation." *Neural Networks* Vol. 1(4), 1988.
- [4] J. A. Janét, T. A. Chase, M. White, John C. Sutton and R. C. Luo, "Pattern Analysis for Autonomous Vehicles with the Region- and Feature-based Neural Network: Global Self-Localization and Traffic Sign Recognition". *1996 IEEE Int'l Conf. on Robotics and Automation*, Minneapolis, MN, April, 1996.
- [5] J. A. Janét, D. S. Schudel, M. White, A. G. England, R. C. Luo, W. E. Snyder, "Global Self-Localization for Actual Mobile Robots: Generating and Sharing Topographical Knowledge Using the Region-Feature Neural Network", *IEEE Int'l Conf on Multisensor Fusion and Integration*, December, 1996.
- [6] J. A. Janét, D. S. Schudel, M. White, A. G. England, E. Grant, W. E. Snyder, "Two Mobile Robots Sharing Topographical Knowledge Generated by the Region-Feature Neural Network", *IEEE Int'l Conf on Robotics and Automation*, Albuquerque, NM, April, 1997.
- [7] J. A. Janét, R. Gutierrez, T. A. Chase, M. White, J. C. Sutton, III, "Autonomous Mobile Robot Global Self-Localization Using Kohonen and Region-Feature Neural Networks", *Journal of Robotic Systems: Special Issue on Mobile Robots*. April, 1997.
- [8] J. A. Janét, D. S. Schudel, A. G. England, J. C. Sutton, III and M. W. White, "Transferring Neural Network-Based Knowledge Between Two Mobile Robots", *IEEE Trans. on Pattern Analysis and Machine Intelligence*. *In Review*.
- [9] J. A. Janét, M. W. White and J. C. Sutton, III, "Neural Network-Based Analogies: Using Previously Learned Features to Expedite the Recognition of Outdoor Traffic Signs," *IEEE Trans. on Pattern Analysis and Machine Intelligence*. *In Review*.
- [10] J. A. Janét, *Pattern Analysis and Control for Autonomous Vehicles with Neural Networks*, PhD thesis, North Carolina State University, Raleigh, NC, 1998.
- [11] Y. LeCun, B. Boser, et. al. "Backpropagation Applied to Handwritten Zip Code Recognition". *Neural Computation*, Vol 1, 1989, pp 541-551.
- [12] Y. LeCun, B. Boser, et. al. "Handwritten digit recognition with a back-propagation network". *Advances in Neural Information Processing Systems 2*, (D. S. Touretsky, ed.) pp 396-404, San Mateo, CA: Morgan Kaufmann, 1990.
- [13] R. C. Luo, Harsh Potlapalli, and D. E. Hislop, "Translation and Shift Invariant Landmark Recognition Using Receptive Field Neural Networks." *1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, North Carolina, July, 1992.
- [14] M. Smith. *Neural Networks for Statistical Modeling*. Van Nostrand Reinhold, NY, 1993.
- [15] T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink and D. L. Alkon, "Accelerating the Convergence of the Back-Propagation Method." *Bio. Cybernetics*, pp. 256-264, Sept. 1988.