PROCEEDINGS
OF THE

FIFTH WORKSHOP ON NEURAL NETWORKS: Academic/Industrial/ NASA/Defense

An International Conference on Computational Intelligence: Neural Networks, Fuzzy Systems, Evolutionary Programming and Virtual Reality

WNN93/FNN93
San Francisco
November 7-10, 1993
San Francisco Airport Marriott, California

Sponsoring Societies:
The Society for Computer Simulation International
NASA: National Aeronautics and Space Administration

Cooperating:
SPIE: The International Society for Optical Engineering
INNS: The International Neural Networks Society
The IEEE Neural Networks Council

SPIE
Volume 2204

WNN'93

Chairs: Walter Karplus, *UCLA*
Mary Lou Padgett, *Auburn University*

FUZZY AND COMPLEX SYSTEMS
Chairs: Hamid Berenji, *NASA/Ames Research Lab*
Lotfi Zadeh, *University of California, Berkeley*
Robert Lea, *NASA/JSC*

Session - Fuzzy Control
Chair: Hamid Berenji, *NASA/Ames Research Lab*

Session - Fuzzy Theory and Applications
Chairs: Bob Shelton, *NASA/JSC*
Lincoln D. Jones, *NASA/JSC*

Session - Fuzzy Logic and Complex Systems
Chair: Jacques Blanc-Talon, *ETCA/CREA/SP*

GENETIC ALGORITHMS
Chair: B. A. Dike, *McDonnell Douglas Aerospace*
Simon Handley, *Stanford University*

Session - Industrial Applications
Chair: John R. Koza, *Stanford University*

Session - Programming Applications
Chair: Charles L. Karr, *U.S. Bureau of Mines*

Session - Theory
Chair: Astro Teller, *Carnegie Mellon University*
V. Vemuri, *Lawrence Livermore National Labs*

NEURAL NETWORKS
Chairs: Robert Shelton, *NASA/JSC*
Rao Vemuri, *LLNL*

Session - Neural Networks Overview
Chair: Michael A. Lehr, *Stanford University*

Session - Architectures and Theory
Chairs: E. Fiesler, *IDIAP*
Marino J. Niccolai, *University of South Alabama*

Session - Controls
Chairs: Robert Shelton, *NASA/JSC*
Bill Fisher, *Lockheed*

Session - Performance
Chairs: Robert W. Green, *University of Massachusetts*
George W. Rogers, *Naval Surface Warfare Center*

Session - Applications and Optimization
Chairs: Camille C. Price, *Stephen F. Austin State University*
M. S. Obaidat, *CUNY, The City College*

Session - Dynamical Systems
Chairs: V. Vemuri, *Lawrence Livermore National Labs*
Bogdan M. Wilamowski, *University of Wyoming*

Session - Signal Process and Pattern Recognition
Chair: George W. Rogers, *Naval Surface Warfare Center*

Session - DB and Hybrid
Chair: Kevin D. Reilly, *University of Alabama, Birmingham*

Session - Time Series
Chairs: Leonard J. Trejo, **Navy Personnel R&D Center**
Eric A. Wan, *Stanford University*

VIRTUAL REALITY
Chairs: Richard A. Blade, *University of Colorado*
Robert Savely, *NASA/JSC*

Session - Aerospace Virtual Reality
Chairs: Richard A. Blade, *University of Colorado*
Tim Sullivan, *LinCom Corporation*
Charles Paul Kosta, *University of Massachusetts, Lowell*

Session - Virtual Reality Applications
Chairs: Andrew W. Self, *University of Brighton*
Brad Bell, *LinCom Corporation*

NASA/ESA Representatives for WNN/FNN
Robert Savely, NASA/JSC
Hamid Berenji, NASA/Ames
Tim Cleghorn, NASA/JSC
Silvano Colombano, NASA/Ames
Robert Lea, NASA/JSC
Joseph Mica, NASA/Goddard
Juan Miro, ESA
Robert Shelton, NASA/JSC

MODIFICATION OF THE BACKPROPAGATION ALGORITHM FOR FASTER CONVERGENCE

(extensive summary)

Lisa M. Torvik and Bogdan M. Wilamowski
Department of Electrical Engineering
University of Wyoming
Laramie, Wy 82071

Introduction

The backpropagation algorithm commonly employed for the training of multilayer neural networks suffers from a slow asymptotic convergence rate. In this algorithm the weight changes are proportional to an error propagating from the output through the slopes of activation functions and through the weights. For large signals (*net* values) the derivatives of the activation functions are very small (Fig. 1) and the back propagating error signal is also very small. Under such conditions the output can be maximally wrong without producing a large error signal. As a consequence, the learning process and weight adjustment can be very slow. During the learning procedure, the competitive patterns can push an output to a maximally wrong value lowering the error signal. In this case it is extremely difficult to recover the proper state during the learning procedure. Basically two different approaches have been developed for improving convergence of the learning process: modification of the activation function and modification of the activation function slope calculation for error propagation.

Modification Of Activation Function

The activation function is modified in such way that even for very large net values the derivative of the activation function never drops to a very small value. Thus an error will always propagate back. This modification was done by adding an additional term to the equation that describes the sigmoidal function.

$$f(net) = \tanh(0.5\,\lambda\, net) + \alpha\, net \qquad (1)$$

The derivative of the activation function is therefore shifted up eliminating small derivative values. The activation function can be altered a number of different ways. In this work a few other cases were also investigated.

Modification Of Backpropagation Algorithm

Convergence of the learning process can be improved by changing how the error propagates back through the network. With a standard sigmoidal activation function, only a small error propagates back when the neuron is in a maximally wrong state. This is a consequence of using the steepest gradient method for calculating the weight adjustments. For the purpose of error propagation, the slope is calculated from the line connecting the output value with the desired value rather than the derivative of the activation function at the output value. This is illustrated in Fig. 2. Note that if the output value is close to the desired value, the calculated slope corresponds to the derivative of activation function, and the algorithm is identical with standard backpropagation formula. Therefore, the "derivative" is calculated differently only for large errors, when the classical approach significantly limits error propagation.

Results Of Computation And Conclusion

This modification of the activation function results in a reduction of learning time by a factor between 1.5 and 2.5. Figures 3 and 4 show the error during the learning procedure for an *"Exclusive OR"* example with a learning constant of 0.3 and $\lambda = 1$. Values of the α parameter used in equation 1 were set 0.0 and 0.05. Note that α=0 corresponds to a standard sigmoidal function used in the backpropagation algorithm.

Modification of the backpropagation algorithm using different methods of determining the slope of the activation function enhances the convergence of the learning procedure. Experiments using the three layer feed forward network (one hidden layer) have shown that substantial improvements can be obtained if the slopes in the last layer are computed using the modified algorithm. Modifying the activation function slope computation in the hidden layer did not significantly affect learning convergence.

Improvement was especially significant when the output neurons were initially set to maximally wrong states (the system was initially trained for maximally wrong values). After setting these unfavorable initial weights, the neural network was trained with the standard and the modified backpropagation algorithms. The results are shown in Figures 5 and 6. It is illustrated that the standard backpropagation algorithm does not converge at all while the modified backpropagation algorithm does converge and gives good results.

Figure 1. Standard binary sigmoidal activation function and its derivative.

Figure 2. Illustration of the modified derivative computation using the slope of the line connecting the points of actual output and desired output.

Figure 3. Global error as a function of iterations for the *Exclusive OR* example using the standard activation function and backpropagation method with $\lambda = 1$ and learning constant $\gamma = 0.3$.

Figure 4. Global error as a function of iterations for the *Exclusive OR* example using the modified activation function with $\alpha = 0.05$ (Equation (1)) and backpropagation method with $\lambda = 1$ and learning constant $\gamma = 0.3$.

Figure 5. Global error a as function of iterations for the *Exclusive OR* example using "maximally wrong" initial weights, standard activation function $\lambda = 1$ and learning constant $\gamma = 0.3$.

Figure 6. Global error as a function of iterations for the *Exclusive OR* example using "maximally wrong" initial weights, standard activation function and modified method for derivative computation (Fig. 2) $\lambda = 1$ and learning constant $\gamma = 0.3$.

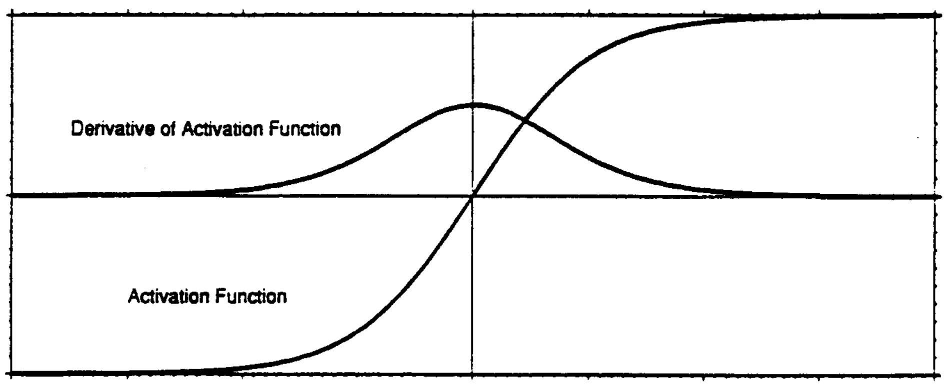

Figure 1. Standard binary sigmoidal activation function and its derivative.

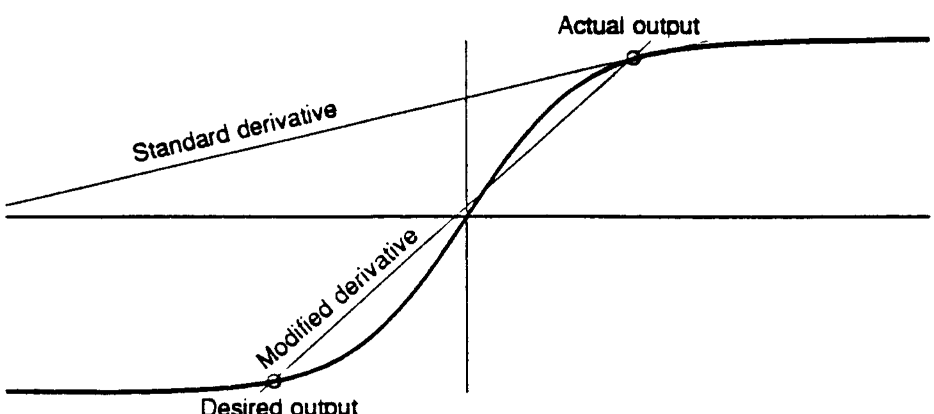

Figure 2. Illustration of the modified derivative computation using the slope of the line connecting the points of actual output and desired output.