CHAOTIC SIMULATED ANNEALING IN MULTILAYER FEEDFORWARD NETWORKS

D. Shaw and W. Kinsner
Department of Electrical and Computer Engineering,
University of Manitoba
Winnipeg, Manitoba, Canada, R3T-5V6
email: kinsner@ee.umanitoba.ca

ABSTRACT

This paper presents a method of chaotic simulated annealing for avoiding and subsequently escaping from local minima in the training of multilayer feedforward neural networks. A modified form of the standard simulated annealing algorithm is implemented using both Gaussian random numbers and various types of strange chaotic attractors for perturbation of network weight parameters. Specifically, the attractor generated by the logistic equation, Hénon's attractor, Rössler's attractor, and the Lorenz attractor are used at different initial conditions and parametric variations for chaotic perturbations. The variance fractal dimension is used as a quantitative measure of the geometric properties of the strange chaotic attractors. It is shown that, for this application, chaotic simulated annealing using the logistic equation is up to 600 percent faster than conventional simulated annealing with Gaussian random numbers.

1. INTRODUCTION

The existence of simulated annealing algorithms for optimization of multivariate functions has been known and used for several years now. Applications include VLSI circuit design, image processing, Boltzmann machines, and in solving the famous travelling salesman problem. More recently, simulated annealing has been used in the training of multilayer feedforward networks. It is impractical, however, to use it alone to train such a network. Other algorithms, such as gradient descent or conjugate gradients, are vastly superior for convergence to a nearest significant minimum in terms of computational time and efficiency. Simulated annealing, however, has proven to be extremely effective for initializing weight parameters instead of the standard random starting weights. Also, annealing has been used for attempting to break out of the local minima in which the standard training algorithms can easily become stuck.

A practical multilayer feedforward network often has thousands of connection weights which must be optimized in order to achieve favourable results. Within this weight space, there are an infinite amount of possible weight combinations which can exist. Subsequently, a similarly large number of local minima will be present. Training of this type of network with traditional gradient techniques alone will often cause the weights to settle at an unacceptable local minimum in a particular network configuration. Avoiding these false minima requires two separate procedures. First, we must avoid starting in their vicinity by determining the starting weights in a manner which is more deliberate than with the commonly accepted practice of random weight initialization. Second, we require a means to determine whether or not we are at a local minimum, and then to escape from it if we are. Although simulated annealing is an acceptable means with which to solve both of these problems, this work will focus primarily on the use of simulated annealing to initialize weight parameters. This scope has been adopted because experience has shown that random pattern mode training, as opposed to batch mode, is more efficient than annealing for the escape of local minima during training. Also, though, this work will be equally applicable to solution of the latter problem because the identical procedure would be used with only minor parametric changes.

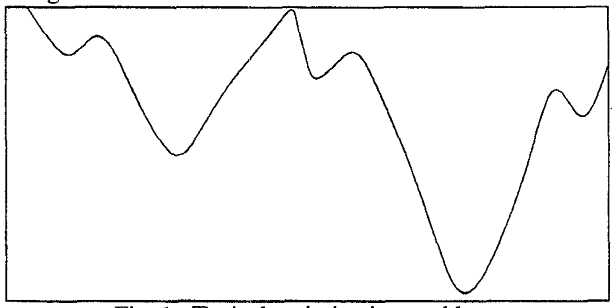
Fig. 1. Typical optimization problem.

For the purpose of this work, simulated annealing can best be described with reference to Fig. 1. Suppose one were to place a ball on this error space and shake it strongly. When the ball finally settles, it can exist anywhere along the curve. If the ball lands near the global minimum in the right half of curve and the strength of the shaking is then reduced, it will be unable to jump high enough to move it to the left half. If it lands in one of the smaller minima on the right side, relatively little shaking will be needed to throw it into the global minimum. Once at the minimum, the weaker shaking will be unable to toss it into a higher minimum.

The amount of shaking, as described above, can be

Authorized licensed use limited to: University of Johannesburg. Downloaded on June 01,2010 at 19:56:00 UTC from IEEE Xplore. Restrictions apply.

controlled by varying a parameter which will be referred to as the annealing temperature. This is simply the standard deviation of the quantity by which the independent variables (weights) will be perturbed during each iteration. Traditional simulated annealing algorithms [LaAa87] employ stochastic acceptance of improvement and usually allow occasional wandering away from improved points based on a separate decreasing temperature parameter. The algorithm used in this work will instead utilize deterministic acceptance parameters, as presented by T. Masters [Mast93]. This type of implementation has been found most appropriate in neural network applications. Further details regarding this procedure can be found in [Mast93].

In the simulated annealing algorithm described by Masters, random numbers are used to determine the weight change for each iteration. This paper explores an alternative to the use of random numbers based on deterministic mathematics. Specifically, various strange chaotic attractors will be used to generate sequences of numbers for weight perturbations. The attractors will be varied in terms of their initial conditions and equation parameters in an attempt to improve the performance of this simulated annealing algorithm. It is conjectured that an optimum level of directional persistence, as exhibited by the attractors, can be found in order to improve the annealing algorithm for this application.

A full description of the various strange chaotic attractors used in this research will follow this section. However, the use of fractal dimensions, as a means to characterize the geometrical structure of the strange chaotic attractors, will first be discussed. Specifically, we will be using the variance fractal dimension for this application.

2. Background Theory

2.1. Fractal Dimension

The dimension of an object is often considered to be the smallest possible integral space onto which the object can be embedded. For example, the dimension of a point is 0, the dimension of a line is 1, and the dimension of a plane is 2. However, ever since the appearance of objects, such as the Cantor set and the Peano space-filling curve, a search for a better definition of dimension has been triggered. For instance, while the space-filling curves are lines by definition, they seemingly occupy the entire 2-dimensional embedding [Pean90, Hilb91]. Therefore, their dimension should be somewhere between 1 and 2. As a result, the concept of dimension was generalized from integers to fractional quantities.

A fractal dimension, D, can be interpreted as the "degree of meandering" (or roughness, brokenness, and irregularity) of an object [Kins94]. For example, although a coastline is immeasurable in terms of length, it has a certain characteristic degree of meandering. Essentially, fractal dimension provides us a means with which to characterize the geometry of any shape or object.

Various methods exist for the calculation of fractal dimensions. In this application, the weight changes will be considered indepently of one another. Thus, we will be required to determine the dimension of a single variable strange chaotic attractor plotted as a temporal signal. A comprehensive exploration of the various ways to calculate fractal dimensions, clearly shows us that the variance dimension, D_σ, is the best choice for this particular problem[Kins94].

2.2. Variance Dimension

Let the time series of interest be defined as a set of points, $B(t)$, which is either continuous or discrete in time t. Then, the variance, σ^2, of its amplitude changes over a time increment and is related to the time increment according to,

$$Var[B(t_2) - B(t_1)] \sim |t_2 - t_1|^{2H} \qquad (1)$$

where H is a value called the Hurst exponent. By setting $\Delta t = |t_2 - t_1|$, and $(\Delta B)_{\Delta t} = B(t_2) - B(t_1)$, we can determine H from,

$$H = \lim_{\Delta t \to 0} \frac{1}{2} \frac{\log[Var(\Delta B)_{\Delta t}]}{\log(\Delta t)} \qquad (2)$$

Finally, the variance dimension can be determined from,

$$D_\sigma = E + 1 - H \qquad (3)$$

where E is the embedding Euclidean dimension and has a value of 1 for this application. Typical values for D_σ are 1.0 for a highly periodic and orderered function such as a sign wave ranging to 2.0 for completely uncorrelated white noise.

The variance dimension is, in fact, so simple to calculate that it can be done directly in real time for most applications as the signal is received. This is of considerable use for fractal analysis of nonstationary processes such as speech and is very immune to noise [Kins94].

2.3. Strange Chaotic Attractors

A strange chaotic attractor will be defined as a set of points which exhibits certain characteristic properties as will be described [PeJS92, Stev89]. Alternatively, an analysis by means of measuring the Lyapunov numbers and exponents can also be used to characterize a strange chaotic attractor as possessing both the attractive and the chaotic properties [PeJS92]. This type of characterization will not be necessary for analysis of the current application.

Firstly, an attractor is characterized by a point in

Authorized licensed use limited to: University of Johannesburg. Downloaded on June 01,2010 at 19:56:00 UTC from IEEE Xplore. Restrictions apply.

phase space to which the solution of a set of equations will converge regardless of initial conditions. In other words, an attractor seems to pull in neighbouring points as generated by the set of equations.

Secondly, the attractor will be defined as being a strange attractor when the equations produce results which follow a fully deterministic path through phase space, yet never shows signs of periodicity or recursion. A quantitative measure for the degree of *strangeness* exhibited by a given strange attractor is the previously described fractal dimension.

Finally, a strange attractor is characterized as being chaotic when the orbits of nearby points diverge from each other due to a sensitive dependence on initial conditions. An important aspect of such a chaotic system is that a simple set of equations can completely describe a very rich and complex nonperiodic behaviour as is necessary in our application.

2.4. The Logistic Equation

The logistic equation produces a simple chaotic strange attractor in a single Euclidean dimension. It is based on an equation which was originally intended to model the size of a population in time intervals, *n*. Calculation of this attractor is done with a simple iterative procedure which is fully described by,

$$p_{n+1} = p_n + rp_n(1-p_n) \tag{4}$$

where *r* represents a constant known as the growth rate. For our purposes, *r* will be set to a fixed value of 3.0. This parameter setting ensures us that chaos will occur when the initial population, p_0, is greater than 0 and less than 1.3. The variance dimension, D_σ, of this type of attractor is 1.999. This is very close to the variance dimension for white noise.

2.5. The Hénon Attractor

Another strange chaotic attractor which is easily calculated is known as the Hénon attractor. It was originally introduced by a French astronomer, Hénon, in 1976 [Héno76] as a simplified model of the Lorenz system which is described in the next section. The Hénon attractor is described by the system of equations given as follows,

$$\begin{aligned} x_{k+1} &= y_k + 1 - ax_k^2 \\ y_{k+1} &= bx_k \end{aligned} \tag{5}$$

for some initial starting point (x_0, y_0).The terms *a* and *b* are constants which are set to 1.4 and 0.3, respectively. Notice that this equation requires the storage of two variables for each iteration as opposed to one for the logistic equation. For our application in simulated annealing, we will use only the x_k component, but y_k will still have to be calculated and stored as x_{k+1} depends on its value. The variance dimension, D_σ, for the x_k component of the Hénon attractor is 1.994. Again, this is close to the variance dimension of white noise.

2.6. The Lorenz Attractor

The Lorenz attractor is generated from a system of differential equations which were originally intended to model thermal convection currents. The differential equations are given as follows,

$$\begin{aligned} x' &= -\sigma x + \sigma y \\ y' &= Rx - y - xz \\ z' &= -Bz + xy \end{aligned} \tag{6}$$

for some initial starting point (x_0, y_0, z_0) at time, $t=0$. The numbers σ, *B*, and *R* are the system's physical parameters which are fixed at 10, 8/3, and 28 respectively. This set of equations now requires the storage of three variables for each iteration. For our application in simulated annealing, we will use only the x_t component, but again the other variables will have to be calculated and stored for each iteration.

Computation of the Lorenz attractor as described by this system of equations is not as simple as the previous two described attractors. To compute this attractor one must solve the above system of differential equations using a numerical technique. A suggested algorithm for this task is the fourth order Runge Kutta technique. This method is a one-step procedure which uses only first-order derivatives to achieve sufficient accuracy for our purposes in a reasonable amount of time. Implementation of this algorithm is a fairly straightforward task [PTVF92].

The variance dimension, D_σ, for the x_t component of this Lorenz attractor was found to vary with the setting of the time step, *dt*, in the solution of the system of differential equations. In fact, by setting *dt* to a relatively low value like 0.0001, it was found that D_σ=1.0 which is the same as for the sine wave. On the other hand, by setting *dt* to a higher value such as 0.1, a variance dimension close to 2 was achieved.

2.7. The Rössler Attractor

The Rössler system of differential equations were designed solely for the purpose of creating a model of a

Authorized licensed use limited to: University of Johannesburg. Downloaded on June 01,2010 at 19:56:00 UTC from IEEE Xplore. Restrictions apply.

strange attractor which uses a simpler chaos generating mechanism. The differential equations are given as follows,

$$x' = -y + z$$

$$y' = x + ay$$

$$z' = b + xz - cz \qquad (7)$$

for some initial starting point (x_0, y_0, z_0) at time, $t=0$. The numbers a, b, and c are system parameters which are fixed at 0.2, 0.2, and 5.7 respectively. This set of equations also requires the storage of three variables for each iteration. For our application in simulated annealing, we will use only the x_t component, but again the other variables will have to be calculated and stored for each iteration.

Computation of the Rössler attractor can be done using the same technique as for the Lorenz attractor. The variance dimension, D_σ, for the x_t component of the Rössler attractor was also found to vary with the setting of the time step, dt, in the solution of the system of differential equations. Ranges for D_σ were found to be from 1.0 for low values of dt to 1.411 for dt=2.2.

3. Experiments and Discussion

3.1. Annealing With Attractors

It is now necessary to describe the special considerations which must be taken into account when implementing the annealing algorithm with the various strange attractors. First, it should be understood that each individual weight parameter must be perturbed by its own distinct sequence of chaotic numbers. In order to do this, it is required that a separate array be established for each time-varying attractor component of size equal to the number of weights in the network. For example, if a given network has 1000 weights and we wish to anneal with the Hénon attractor, then we will require 2 separate arrays of 1000 elements each in order to hold values for x_{k-1} and y_{k-1}. Obviously, to ensure that the chaotic perturbation of each weight follows its own distinct path, it is necessary to initialize each system at a different starting point (x_0, y_0, z_0) by conventional random number generating routines and linear scaling.

Secondly, in the original annealing algorithm described by Masters, it is noted that the random numbers used for weight perturbation have a zero mean and a standard deviation which is equal to the temperature parameter. Thus, the sequences generated by the chaos equations must also be offset and linearly scaled so as to roughly match these criteria. Further conditioning, by means of taking the absolute value and downward shifting was necessary for the sequence generated by the Lorenz equations to satisfy these conditions. It should be noted that the scaling done to fit the strange chaotic attractors to the annealing routine did not have an appreciable effect on the value of the variance dimension. However, to avoid any possible ambiguities, the variance dimensions presented here have been calculated after the above described scaling operations.

Finally, it should be noted that another small modification to the annealing routine was made in order to take advantage of the persistent quality of the numerical sequences taken from the attractors. The modification involves a temporary acceptance of every improvement so that further iterations can be made from that point. Only the best of these temporarily accepted and pursued improvements at each temperature will finally be accepted and carried forward to the next annealing temperature.

3.2. Testing and Results

Two simple test networks were developed in order to assess the usefulness of the new annealing algorithm. The first one is a small network consisting of 50 input nodes, 14 hidden-layer nodes, and 16 output-layer nodes. There is one output node for each of 16 possible training vectors of size 50 samples each. This network has a total of 924 weights (variables) which must be optimized. The objective function which we are trying to minimize with this procedure is the mean squared error, E, of the outputs over the entire training set. For this particular test network, the initial value of E is 0.165 at some arbitrary starting point.

The second test network is closer to what one might find in a practical application of a multilayer feedforward network. It has one output node for each of its 64 possible training vectors of size 200 samples each. It has 200 input nodes and 25 hidden layer nodes for a total of 6600 connection weights. The initial value of E is 0.170 at some arbitrary starting point for this network.

The stopping criterion for both networks has been set at $E=0.03$. First, the networks were both annealed with Master's annealing algorithm to serve as a benchmark for further tests. The first network took 3 minutes and 2 seconds (00:03:02) to reach the stopping criteria. The second network took 6 minutes and 22 seconds (00:06:22). All tests were performed on a dedicated *Sun Sparcstation 5*, using compiled C code.

Next, the networks were reset to their initial starting weights and were trained using the described chaotic annealing algorithm to the same stopping criterion. Excellent results were achieved for these tests. The first network reached $E=0.03$ in about 29 seconds. This is over 600 percent faster than with the benchmark annealing algorithm. The second network reached $E=0.03$ in about 2 minutes and 19 seconds (00:02:19). Though not as dramatic as the results

Authorized licensed use limited to: University of Johannesburg. Downloaded on June 01,2010 at 19:56:00 UTC from IEEE Xplore. Restrictions apply.

obtained for the first network, this still represents a significant improvement of about 275 percent over Master's annealing algorithm. The discrepancy between the rates of improvement between the two test networks can be justified. This is due to the larger network spending more time computing the error function relative to the time spent in computing for the actual annealing routine. In other words, if the proportion of time spent computing the error function to the time spent in annealing could be kept constant for these two test networks, then similar speed improvements would be realized.

Finally, the networks were reset to their initial starting weights and annealing was attempted with the Hénon, Lorenz, and Rössler attractors separately. Unfortunately, no significant results could be obtained with any of these attractors. This means that even though the annealing algorithm still worked, they were all many times slower than even the original benchmark algorithm. None of them reached the stopping criterion of $E=0.03$ in a reasonable amount of time. Possible reasons for this include incorrect scaling of the attractors or general unsuitability for this application due to geometrical properties. However, it is more likely that these functions were scaled incorrectly in time, and perhaps, were too persistent for this application. A larger step size, dt, would have been necessary to solve this problem, but would require a better numerical technique, such as a higher order Euler method, for solution of the differential equations.

SUMMARY

In this paper, we presented a significantly improved method of simulated annealing for initialization of weight parameters in multilayer feedforward networks. The improvement takes advantage of several characteristics of strange chaotic attractors which contrast them to number sequences produced by Gaussian random number generators. The first, and probably most significant of these features is that the strange chaotic attractors exhibit a certain degree of persistence in time. This idea can be paralleled with that of momentum, which has been proven to speed convergence in the standard network training techniques [PlNH86]. Essentially, it has been shown that when a feedforward network is trained by conventional means, improvements are noticed if subsequent weight changes are in the same direction as previous ones. Another significant feature of the attractors is the speed at which successive iterations can be calculated. Even the solution of the systems of differential equations is faster than the necessary Gaussian random number generation routines. Finally, it is important to note that no matter how good a given Gaussian random number generator claims to be, it will always exhibit some type of periodicity at some time, t. The possibility of this periodicity affecting annealing cannot be ignored. However, this problem does not exist in the implementation using strange chaotic attractors as they are completely non-periodic.

Future work includes generation and processing of the other strange chaotic attractors so that they can be used more effectively for this application. It is only in this manner that we will be able to find an optimal degree of *strangeness* as measured by the variance fractal dimension. Pending the success at finding the optimal dimension, it will be possible to find an ideal strange chaotic attractor with which to implement this annealing algorithm.

REFERENCES

[Héno76] M. Hénon, "A two-dimensional mapping with a strange attractor," *Comm. Math. Phys.* 50 (1976) pp. 69-77.

[Hilb91] D. Hilbert, "Über die stetige Abbildung einer Linie auf ein Flächenstück," *Mathematische Annalen 38*, pp. 459-460, 1891.

[Kins94] W. Kinsner, "A unified approach to fractal and multifractal dimensions," *Technical Report*, DEL94-4. Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, Manitoba, Canada, May 1994, 140 pp.

[LaAa87] P. van Laarhoven and E. Aarts, Simulated Annealing: Theory and Applications. Dordrecht, Holland: D. Reidel Publishing Company, 1987, 186 pp.

[Mast93] T. Masters, *Practical Neural Network Recipes in C++*. San Diego, CA: Academic Press, Inc, 1993, 493 pp.

[Pean90] G. Peano, "Sur une courbe, qui remplit toute une air plane," *Mathematische Annalen 36*, pp. 157-160, 1890.

[PeJS92] H. Peitgen, H Jürgens, and D Saupe, *Chaos and Fractals New Frontiers of Science*. New York, NY: Springer-Verlag, 1992, 984 pp.

[PlNH86] D. Plaut, S. Nowlan, and G. Hinton, "Experiments on learning by back propagation," *Technical Report*, CMU-CS-86-126. Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

[PTVF92] W. Press, S Teukolsky, W. Vetterling, and B. Flannery, Numerical Recipes in C, Second Edition. New York, NY: Cambridge University Press, 1992, 994 pp.

[Stev89] R. Stevens, Fractal Programming in C. Redwood City, MA: M & T Publishing, Inc, 1989, 583 pp.

Authorized licensed use limited to: University of Johannesburg. Downloaded on June 01,2010 at 19:56:00 UTC from IEEE Xplore. Restrictions apply.