# Multilayer Perceptrons and Fractals

C. A. Murthy* and Jennifer Pittman

Center for Multivariate Analysis
Department of Statistics
326 Thomas Building
Pennsylvania State University
University Park, PA - 16802, USA

*On lien from the Machine Intelligence Unit, Indian Statistical Institute, 203 B.T. Road, Calcutta - 700035, India.

## Abstract

In this article, a mathematical relationship between the gradient descent technique and contractive maps is examined. This relationship is based upon the observation that the convergence of the gradient descent technique can be proved using results in fractal theory - more specifically, results concerning contractive maps - as opposed to results based on Taylor's series. This proof, involving the eigenvalues of the Hessian matrix of the gradient descent technique's objective function, is presented. A simple example is given in which steps from the aforementioned proof are used to find conditions under which a specific multilayer perceptron is guaranteed to converge. Since the gradient descent technique is used in multilayer perceptrons, and contractive maps give rise to fractals, a theoretical relationship is thus established between multilayer perceptrons and fractals.

**Key Words** :

Gradient descent, Contractive map, Fractal, Hessian matrix, Fixed point, Attractor

# 1 Introduction

Multilayer Perceptrons (MLPs) have been used in numerous applications, many of which have involved either classification of given observations or approximation of the observations' generating function [1]. To arrive at a suitable solution for such problems, MLPs use both back propagation of error [2] and the gradient descent technique to optimize an objective function.

It has been postulated whether the optimization process of MLPs and the generation of fractals are related. Fractals are self-similar mathematical objects which are usually generated by contractive maps [3]. A group of contractive maps, applied repeatedly to any collection of nonempty compact sets, will ultimately force these sets to converge to a single set, called the attractor. The attractor is considered fractal.

The proof of convergence of the gradient descent technique to a local optimum is usually given in terms of results on Taylor's series [6]. However, an examination of the recent literature on fractals and contractive maps raised questions regarding whether this proof could be restated in terms of fractal theory, thus establishing a relationship between MLPs and fractals.

The present article establishes such a relationship by using characteristics of fractal convergence to rewrite the proof of convergence of the gradient descent technique. This entails stating conditions under which the function representing the gradient descent process is a contractive map. Section 2 provides some basic results on fractals which are relevant from the point of view of MLPs. Section 3 provides the relationship between contractive maps and the gradient descent technique. Section 4 describes the connection between contractive maps and MLPs. Section 5 presents some experimental results, and Section 6 contains a final discussion. This work should encourage interaction between researchers in neural networks and those in fractal theory, hopefully leading to positive developments in both fields.

# 2 Mathematical Preliminaries

We shall describe below some of the basic results concerning fractals. The theory behind fractal generation is based on contractive maps; hence we will initially discuss such maps. Although the following results regarding contractive maps hold for any complete metric space, we will state them with respect to $N$-dimensional Euclidean space. Most of the results stated here can be found in Barnsley [3].

Let $\boldsymbol{R}$ denote the real line. Let $\boldsymbol{R}^N$ denote the $N$-dimensional Euclidean plane and let $\boldsymbol{A}$ denote the set of all non-empty compact subsets of $\boldsymbol{R}^N$. $\rho(x, y)$ will denote the euclidean distance between two points $x$ and $y$ in $\boldsymbol{R}^N$.

**Definition 2.1** A function $f$ defined from $\boldsymbol{R}^N$ to $\boldsymbol{R}^N$ is said to be *contractive* if there exists $s$, $0 \leq s < 1$ such that

$$\rho(f(x), f(y)) \ \leq \ s \cdot \rho(x, y) \quad \forall \ x, y \ \in \boldsymbol{R}^N.$$

$s$ is said to be a *contractivity factor* of $f$. ♠

A contractive function (map) $f$ will shrink any nonempty compact subset of $\boldsymbol{R}^N$. If there does not exist any $s < 1$ for which the above holds then $f$ is not contractive.

**Result 2.1** Let $f$ be a contractive map from $\boldsymbol{R}^N$ to $\boldsymbol{R}^N$. Then there exists a unique $x_0 \in \boldsymbol{R}^N$ such that

1. $f(x_0) = \ x_0$

   and

2. $\lim_{n\to\infty} f^n(x) = x_0 \ \forall \ x \in \boldsymbol{R}^N$ where
   - $f^1(x) = f(x)$
   - $f^n(x) = f^{n-1}(f^1(x)) \quad \forall \ n > 1$ and $\forall \ x$.

$x_0$ is said to be the *fixed point* of $f$. Repeated applications of $f$ on any nonempty compact subset of $\boldsymbol{R}^N$ will make it go towards a set containing only $x_0$.

We shall extend Result 2.1 to sets by using a metric between sets, namely, the *Hausdorff metric.*

**Definition 2.2** Let $\rho(x, A) = \inf_{y\in A} \rho(x, y)$. The *Hausdorff distance* between two sets $A$ and $B$ in $\boldsymbol{A}$ is defined as

$$D(A, B) = \max \ [\sup_{x\in A} \rho(x, B), \ \sup_{y\in B} \rho(y, A)].♠$$

It can be easily shown that the above $D$ is a metric in $\boldsymbol{A}$.

**Definition 2.3** Let $f$ be a function from $\boldsymbol{A}$ to itself. Then $f$ is said to be *contractive* if there exists $s, \ 0 \leq s < 1,$ such that

$$D(f(x), f(y)) \ \leq \ s \cdot D(x, y) \quad \forall \ x, y \ \in \boldsymbol{A}.$$

$s$ is said to be a *contractivity factor* of $f$. ♠

**Result 2.2** Let $f$ be a contractive map from $\boldsymbol{A}$ to $\boldsymbol{A}$. Then there exists a unique $x_0 \in \boldsymbol{A}$ such that

1. $f(x_0) = x_0$
2. $\lim_{n\to\infty} f^n(x) = x_0 \;\; \forall \;\; x \in \boldsymbol{A}$ where
   - $f^1(x) = f(x)$
   - $f^n(x) = f^{n-1}(f^1(x)) \quad \forall \;\; n > 1$ and $\forall \; x \in \boldsymbol{A}$.

Here the metric under consideration is the Hausdorff metric $D$. ♠

**Result 2.3** Let $f_1, \; f_2, \; \cdots, \; f_M$ be $M$ contractive maps defined from $\boldsymbol{A}$ to itself Let $s_1, \; s_2, \; \cdots, \; s_M$ be their respective contractivity factors. For any $C \subseteq \boldsymbol{A}$, let

$F_{n+1}(C) = \bigcup_i f_i(F_n(C)) \;\; \forall \;\; n \geq 0$ where $F_0(C) = C \;\; \forall \;\; C \subseteq \boldsymbol{A}$.

Then there exists $A \subseteq \boldsymbol{A}$ such that

1. $F_1(A) = A$
2. $\lim_{n\to\infty} F_n(C) = A \;\; \forall \;\; C \;\; \subseteq \boldsymbol{A}$.

$(\boldsymbol{A} : \; f_1, \; f_2, \; \cdots, \; f_M)$ is said to be an *iterated function system* and $A$ is said to be the *attractor* of the iterated function system. ♠

The word *fractal* is defined by various authors in various ways. Barnsley considers a fractal to be a set in $\boldsymbol{A}$; we will also consider fractals as such. Fractals are usually generated by iterated function systems.

A definition and preliminary result from matrix algebra are stated below. These will be used in developing the relationship between the gradient descent technique and contractive maps.

**Definition 2.4** For a real matrix $B$ of order $N \times N$, the *norm* of $B$, denoted by $||B||$, is defined as

$$||B|| \;\; = \;\; \sup_{(\boldsymbol{x}:||\boldsymbol{x}|| \neq 0)} \;\; ||B\boldsymbol{x}||/||\boldsymbol{x}||.$$

where $||\boldsymbol{x}|| = \sqrt{x_1^2 + \ldots + x_N^2}$ if $\boldsymbol{x} = (x_1, \ldots, x_N)$.

**Result 2.4** If $B$ is a real symmetric positive definite matrix of order $N \times N$ then its norm is $\max[\alpha_1, \; \alpha_2, \cdots, \; \alpha_N]$ where the $\alpha_i$'s are the eigenvalues of the matrix $B$.

**Note** : If $B$ is a real symmetric matrix then the norm of $B$ is the maximum of the modulus of the eigenvalues of $B$.

This completes the mathematical preliminaries. We shall now discuss the relationship between contractive maps and the gradient descent technique.

# 3 Gradient Descent Technique and Contractive Maps

Gradient descent is a technique used to find the minimum of a given function. It is commonly used in neural network applications to find the minimum of the given objective function [6]. We describe the gradient descent technique below.

## 3.1 Gradient Descent Technique

Let $g$ be a continuous, twice differentiable function from $\boldsymbol{R}^N$ to $\boldsymbol{R}$. Let $\partial g(\boldsymbol{y})/\partial x_i$ denote the partial derivative with respect to $x_i$ of the function $g$ at the point $\boldsymbol{y}$ in $\boldsymbol{R}^N$ where $i = 1, 2, \cdots, N$.

Let

$$\nabla g(\boldsymbol{y}) = (\partial g(\boldsymbol{y})/\partial x_1,\ \partial g(\boldsymbol{y})/\partial x_2, \cdots,\ \partial g(\boldsymbol{y})/\partial x_N)^t$$

where $t$ denotes the transpose. $\nabla g(\boldsymbol{y})$ is the *gradient* of $g(\boldsymbol{y})$.

Let $f$, a function from $\boldsymbol{R}^N$ to itself, be such that

$$f(\boldsymbol{y}) = \boldsymbol{y} - \eta \nabla g(\boldsymbol{y}) \tag{1}$$

where $\eta > 0$ is a constant. Equation 1 represents the process of the gradient descent technique. In other words, let

$f^1(\boldsymbol{y}) = \ f(\boldsymbol{y})$ and $f^n(\boldsymbol{y}) = f^{n-1}(f^1(\boldsymbol{y}))$ for all $n > 1$ and for all $\boldsymbol{y} \in \boldsymbol{R}^N$.

The limit of $f^n(\boldsymbol{y})$ as $n$ goes to infinity is taken to be the solution for the minimization of $g$.♠

In terms of neural networks, $g(\boldsymbol{y})$ is usually the error function, i.e., $\sum(obs - exp)^2$, viewed as a function of the network weights. The objective is to find the choice of weights which minimizes $g(\boldsymbol{y})$. Note, however, that the result of the gradient descent technique depends on the choices of $\eta$ and the initial weight vector.

We shall find the relationship between the gradient descent technique and the contractive maps below. Initially, we shall assume that $N = 1$ and later the results will be generalized to any $N$.

Let $N = 1$. Then $g : \boldsymbol{R} \to \boldsymbol{R}$ and Equation 1 can be written as

$$f(y) = y - \eta \frac{dg(y)}{dy} \tag{2}$$

**Theorem 3.1** Let $g$ be a twice differentiable function with $x_0$ as a local minimum. Let $h(x) \ = \ d^2g(x)/dx^2$ be a continuous functionwhere $h(x) \ > \ 0$ at $x = x_0$.
Let $v_2$ be an open interval containing $x_0$ such that $dg(x)/dx \neq 0$ for all $x \neq x_0$ and

$x \in v_2$. Then there exists a closed interval $v$ around $x_0$ such that the function $f$, defined in Equation 2, is a contractive map on $v$ and its fixed point in $v$ is $x_0$.

*Proof*: Note that $h$ is continuous and $h(x_0) > 0$. Then there exists an open interval $v_1$ around $x_0$ such that $h(x) > 0$ for all $x \in v_1$. Let $y_1, y_2$ be in $v_1$ such that $y_1 \neq y_2$. Now

$$\begin{aligned} |f(y_1) - f(y_2)| &= |y_1 - y_2 - \eta(dg(y_1)/dy - dg(y_2)/dy)| \\ &= |y_1 - y_2 - \eta((dg(y_1)/dy - dg(y_2)/dy)/(y_1 - y_2))(y_1 - y_2)| \\ &= |y_1 - y_2||1 - \eta h(y_3)|. \end{aligned}$$

(Here $y_3$ is a convex combination of $y_1$ and $y_2$. This step follows from the Mean Value theorem [4].)

Let $v$ be a closed interval such that (1) $v$ is contained in $v_1$ and $v_2$, (2) $v$ contains $x_0$, and (3) $v$ is bounded. Note that $h$ is bounded on $v$.

Let $a = \inf_{x \in v}(1/h(x))$.
Let

$$0 < \eta < a \tag{3}$$

Then note that $0 < |1 - \eta h(y)| < 1$ for all $y$ in $v$. Then

$$|f(y_1) - f(y_2)| = |y_1 - y_2||1 - \eta h(y_3)| < |y_1 - y_2|$$

Thus $f$ is a contractive map on $v$. Its fixed point is the point $y$ in $v$ such that $f(y) = y$. Now

$$\begin{aligned} & f(y) = y \\ & \quad \Leftrightarrow y - \eta dg(y)/dy = y \\ & \quad \Leftrightarrow \eta dg(y)/dy = 0 \\ & \quad \Leftrightarrow dg(y)/dy = 0 \quad \text{(since } \eta > 0\text{)} \end{aligned}$$

Note that the only point in $v$ for which $dg(y)/dy = 0$ is $x_0$ since $v$ has been chosen in that way. ♠

Generally, the proof for the gradient descent technique is derived using the results on Taylor's series. The above proof, however, is based on the theory of contractive maps.

## Remarks

1. The above theorem indicates that $f$ is a contractive map in the compact interval $v$ and if $f$ is iterated in $v$, it will eventually produce the local minimum $x_0$. The bound for $\eta$ may also be noted in this regard. If $\eta$ does not satisfy the Equation 3, then the function $f$ may take values outside $v$ for some $x$'s in $v$. Also, the selection of $\eta$ is problematic if we want to get the global optimum.

2. The results hold in a certain interval containing the local optimum and the initial point for the iteration needs to be taken in that interval in order to get that particular local optimum. Thus, if we have more than one local optimum and the initial point is taken in the respective interval of one of these local optima then, with the proper choice of $\eta$, the technique will converge to that optimum.

3. If the initial point is outside of the respective intervals of all local optima or $\eta$ is not chosen properly then $f$ may not be contractive and hence the technique may not converge. In other words, the process may diverge to $+\infty$ or $-\infty$ or it may oscillate between $+\infty$ and $-\infty$.

4. Let the function $g$ posess $k$ local optima $x_1,\ x_2,\ \ldots\ ,\ x_k$ for which $h(x_i) > 0\ \forall\ i = 1, \ldots, k$. Let $v_i$ be a closed disk around $x_i$ such that $dg(y)/dy \neq 0$ for all $y \neq x_i$ and $y \in v_i$. Let $\eta_i$ be a choice of $\eta$ which makes $f$ contractive for $i = 1, \ldots, k$. Let $\lambda = \min[\eta_1, \ldots, \eta_k]$ and let

   $$f(y) = y - \lambda \frac{dg(y)}{dy} \tag{4}$$

   Then, if $y$ is an element of $\bigcup_i v_i$, the limit of $f^n y$ as $n$ goes to infinity belongs to the set $A$ where $A = \{x_1, \ldots, x_k\}$. In other words, if $Z$ is a nonempty compact subset of $A$, then

   $$\lim_{n\to\infty} f^n(Z)\ \in\ A.$$

♠

Since neural network applications involve data of higher dimensions, the generalization to the $N$ dimensional case is stated below.

**Theorem 3.2** Let $g$ be a function from $\boldsymbol{R}^N$ to $\boldsymbol{R}$ with a local minimum at $\boldsymbol{x}_0$. Let $b_{ij} = \partial^2 g/\partial x_i \partial x_j$ such that $b_{ij}$ is continuous for each $(ij)$ pair and $b_{ij}\ (\boldsymbol{y})$ denotes the value of $b_{ij}$ calculated at $\boldsymbol{y}$. Let $\boldsymbol{H}$ be the Hessian matrix of $g$. Let $v_2$ be an open disk containing $\boldsymbol{x}_0$ such that $\nabla g(\boldsymbol{x}) \neq \boldsymbol{0}$ for all $\boldsymbol{x} \neq \boldsymbol{x}_0$ and $\boldsymbol{x} \in\ v_2$. Let $v_3$ be an open disc containing $\boldsymbol{x}_0$ such that $\boldsymbol{H}(\boldsymbol{c})$ is positive definite for all $\boldsymbol{c} \in v_3$. Then there exists a closed disc $v$ around $\boldsymbol{x}_0$ and a constant $\eta\ >\ 0$ such that the function $f$, defined in Equation 1, is a contractive map on $v$ and its fixed point in $v$ is $\boldsymbol{x}_0$.

*Proof*: Note that for each $\boldsymbol{c} \in v_3$, the eigenvalues of $\boldsymbol{H}(\boldsymbol{c})$ are all greater than zero since $\boldsymbol{H}(\boldsymbol{c})$ is positive definite. Consider a closed disc $v$ around $\boldsymbol{x}_0$ such that $\nabla g(\boldsymbol{x}) \neq \boldsymbol{0}$ for all $\boldsymbol{x} \neq \boldsymbol{x}_0,\ \boldsymbol{x} \in v$ and $\boldsymbol{H}(\boldsymbol{c})$ is positive definite for all $\boldsymbol{c} \in v$. Note that $v$ is a subset of $v_2 \bigcup v_3$ and such a $v$ exists.

Consider the equation

$$f(\boldsymbol{y}) = \boldsymbol{y} - \eta \nabla g(\boldsymbol{y}) \tag{5}$$

Let $\boldsymbol{a} \neq \boldsymbol{b}$ be two vectors in $v$. Now

$$\begin{aligned}
||f(\boldsymbol{a}) - f(\boldsymbol{b})|| &= ||\boldsymbol{a} - \boldsymbol{b} - \eta(\nabla g(\boldsymbol{a}) - \nabla g(\boldsymbol{b}))|| \\
&= ||\boldsymbol{a}-\boldsymbol{b}|| * ||((\boldsymbol{a}-\boldsymbol{b})/(||\boldsymbol{a}-\boldsymbol{b}||)) - \eta((\nabla g(\boldsymbol{a}) - \nabla g(\boldsymbol{b}))/(||\boldsymbol{a}-\boldsymbol{b}||))|| \\
&= ||\boldsymbol{a}-\boldsymbol{b}|| * ||((\boldsymbol{a}-\boldsymbol{b})/(||\boldsymbol{a}-\boldsymbol{b}||)) - \eta \boldsymbol{H}(\boldsymbol{c})((\boldsymbol{a}-\boldsymbol{b})/(||\boldsymbol{a}-\boldsymbol{b}||))||.
\end{aligned}$$

( Here $\boldsymbol{c}$ is a convex combination of $\boldsymbol{a}$ and $\boldsymbol{b}$, and $\boldsymbol{c}$ also lies in the set $v$. The existence of such a $\boldsymbol{c}$ is guaranteed from the literature [5].)

Let $I$ represent the $N$ dimensional identity matrix. Then

$$\begin{aligned}
||f(\boldsymbol{a}) - f(\boldsymbol{b})|| &= ||\boldsymbol{a} - \boldsymbol{b}|| * ||(I - \eta \boldsymbol{H}(\boldsymbol{c}))((\boldsymbol{a} - \boldsymbol{b})/(||\boldsymbol{a} - \boldsymbol{b}||))||. \\
&\leq ||\boldsymbol{a} - \boldsymbol{b}|| * ||I - \eta \boldsymbol{H}(\boldsymbol{c})||.
\end{aligned}$$

Let $\alpha(\boldsymbol{c}) = ||I - \eta \boldsymbol{H}(\boldsymbol{c})||$.

Now, if $\eta$ is selected in such a way that $\alpha < 1$ for each $\boldsymbol{c}$ in $v$, then $f$ would be a contractive map and thus its fixed point would be $\boldsymbol{x}_0$.
Let $B(\boldsymbol{c}) = I - \eta \boldsymbol{H}(\boldsymbol{c})$. We observe that $B(\boldsymbol{c})$ is symmetric so the norm of $B(\boldsymbol{c})$ is the maximum of the modulus of its eigenvalues (from Section 2). Hence if $\mu_1(\boldsymbol{c})$, $\mu_2(\boldsymbol{c})$, $\cdots$, $\mu_N(\boldsymbol{c})$ are the eigenvalues of $\boldsymbol{H}(\boldsymbol{c})$ then the eigenvalues of $B(\boldsymbol{c})$ are $1 - \eta\mu_i(\boldsymbol{c})$ for $i = 1, 2, \cdots, N$. Note that (1) $\mu_i(\boldsymbol{c}) > 0$ $\forall\ i = 1, 2, \cdots, N$ and $\forall\ \boldsymbol{c} \in v$ since $\boldsymbol{H}(\boldsymbol{c})$ is positive definite and (2) $\mu_i(\boldsymbol{c})$ is bounded for all $i = 1, 2, \cdots, N$ and for all $\boldsymbol{c}$ in $v$ since $v$ is closed. Hence $\eta$ can be selected in such a way that

$$0 < 1 - \eta\mu_i(\boldsymbol{c}) < 1 \text{ for } i = 1, 2, \cdots, N \text{ and } \forall\ \boldsymbol{c} \in v.$$

For this specific $\eta$, $||B(\boldsymbol{c})|| < 1$. This makes $f$ contractive in $v$.
Hence the theorem. ♠

## Remarks

1. Different proofs of the convergence of the gradient descent technique exist in the literature; see [6]. However, this proof is unique in that it establishes a relationship between gradient descent and contractive maps.
2. The continuity of the Hessian matrix in the neighborhood of $\boldsymbol{x}_0$ is one of the assumptions on which the above proof is based. The eigenvalues of $\boldsymbol{H}$ for different $\boldsymbol{c}$'s are bounded because of this assumption.
3. The fixed points of the function $f$ are different for different closed discs, for different $\eta$'s, and for different starting vectors. Note also that for the same $\eta$,

the function $f^n(\boldsymbol{y})$ may oscillate as $n$ goes to infinity for some $\boldsymbol{y}$'s. The values of $\eta$ for which $f^n(\boldsymbol{y})$ oscillates depends on $\boldsymbol{y}$ and the particular problem under consideration.

4. The selection of $\eta$ is crucial to the performance of a neural network and its final result. The current methods for choosing $\eta$ are heuristic, not theoretical. One such method is to choose a value of $\eta$ which is very low. However, which values of $\eta$ are considered low is the subjective choice of the researcher, and may in fact be quite large relative to the problem of interest.

5. Let the function $g$ possess $k$ local optima $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_k$ for which the corresponding Hessian matrices are positive definite. Let $v_i$ be a closed disc around $\boldsymbol{x}_i$ such that the corresponding Hessian matrices are positive definite for all $\boldsymbol{y} \in v_i$ and for all $i = 1, 2, \cdots, k$. Let $\eta_i$ be a choice of $\eta$ which makes $f$ contractive for $i = 1, 2, \cdots, k$. Set $\lambda = \min[\eta_1, \eta_2, \cdots, \eta_k]$. Let

$$f(\boldsymbol{y}) = \boldsymbol{y} - \lambda \nabla g(\boldsymbol{y}) \tag{6}$$

Then the limit of $f^n \boldsymbol{y}$ as $n$ goes to infinity belongs to the set $A$ where $A = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k\}$ if $\boldsymbol{y}$ is an element of $\bigcup_i v_i$. In other words, if $C$ is a nonempty compact subset of $A$, then

$$\lim_{n \to \infty} f^n(C) \in A.$$

♠

In the next section, the connection between MLP and the above theorems is described.

# 4 MLP and Contractive Maps

Multilayer Perceptrons is a neural network model which is commonly used in supervised pattern classification. The connection weights in the network model are updated with the help of back propagation [2]. There are two ways of implementing MLP - batch mode and on line. The following arguments don't depend upon the mode of implementation. However, some authors have considered adding a momentum term to the equation for updating the connection weights in MLP. The following discussion is confined to the MLP as described in [2], and as such does not include models with the momentum term modification.

The connection weights in the MLP are modified with the help of Equation 5. The vector $\boldsymbol{y}$ represents the weight vector while $\eta \nabla g(\boldsymbol{y})$ represents the change in the weight vector (as noted previously, $g(\boldsymbol{y})$ is usually the error function [9]. In the batch mode learning algorithm, the connection weights are changed after **all** the vectors in the training set have been fed to the input layer whereas in the on line learning algorithm, the weights are changed after **each** vector is fed to the input layer. In the back propagation algorithm, initially, the connection weights in the topmost layer are modified.

Then the connection weights in the next lower layer are modified, and so on, until the weights in the bottom-most layer are modified. The modification of connection weights in any layer is done using Equation 5. It is expected that the back propagation algorithm will provide a local optimal solution. It is not difficult to reach a stable value for $g$ by making a few trials with $\eta$.

Theorem 3.2 establishes the relationship between MLP and fractals. It specifies that the gradient descent technique employed by MLP converges for certain values of $\eta$ and certain starting values. For these values $f$ is a contractive map. Systems of contractive maps are often used to generate fractals. Note that the choice of $\eta$ depends upon the eigenvalues of the Hessian matrix. For this reason, given a real life problem, it may be very difficult to use Theorem 3.2 (see Section 5). However, recent developments in neural network theory regarding the computation of the Hessian matrix for certain network models ([7],[9]) may ease its implementation.

# 5 Example

## 5.1 Preliminaries

We determined initially that constructing a hypothetical example involving the use of a MLP to build a mathematical model for minimizing an expression of the form $\sum(obs - exp)^2$ would be extremely difficult [9]. This is due primarily to two facts: first, that such an example would require the creation of a function $g : \mathbf{R}^{weights} \to \mathbf{R}$ where *weights* is the number of weights in the MLP and a choice of training sample points such that the expected results were known and could be compared to the experimental results. Second, the function $\sum(obs - exp)^2$ must have a positive definite Hessian matrix, where the derivatives are taken with respect to the network weights. However, (1) this is often not the case in multilayer perceptron learning ([8], [9]), hence the existence of modified Newton methods, and (2) determining whether the Hessian is positive definite requires the calculation of its eigenvalues, a computationally intensive task if the dimension of the Hessian is large. The extreme difficulties in constructing a realistically complex example is highlighted by the lack of such an example in the existing literature.

For these reasons we chose instead to demonstrate the use of the above theory to minimize a function using a neural network of the form

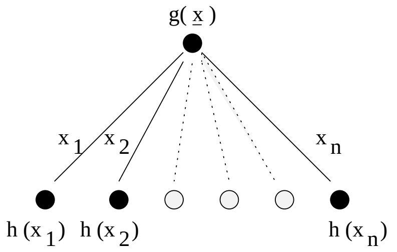

Figure 1: Neural network model

where (1) $x_1, \dots, x_n$ are the network weights as well as the function variables, (2) $h_1(\underline{x}), \dots, h_n(\underline{x})$ are factors of $g(\underline{x})$ (i.e., $g(\underline{x}) = h_1(\underline{x}) * \cdots * h_n(\underline{x})$), and (3) no transfer function is used. Nevertheless, if one has a function to be minimized which meets the assumptions of the above theorems, then our methodology will work for solving problems of the type outlined in the previous section.

## 5.2 Construction

Let $\underline{x} = (x_1, \dots, x_n)$. The function $g(\underline{x})$ to be minimized was required to have a positive definite Hessian matrix, i.e., $\mu_i > 0 \ \forall \ i = 1, \dots, n$, where $(\mu_1, \dots, \mu_n)$ are the eigenvalues of

$$H_g = \begin{bmatrix} \frac{\partial^2}{\partial^2 x_1} g(\underline{x}) & \frac{\partial^2}{\partial x_1 \partial x_2} g(\underline{x}) & \cdots & \cdots & \frac{\partial^2}{\partial x_1 \partial x_n} g(\underline{x}) \\ \frac{\partial^2}{\partial x_2 \partial x_1} g(\underline{x}) & \frac{\partial^2}{\partial^2 x_2} g(\underline{x}) & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_1 \partial x_n} g(\underline{x}) & \cdots & \cdots & \cdots & \frac{\partial^2}{\partial^2 x_n} g(\underline{x}) \end{bmatrix}$$

We chose to minimize the function $g(x_1, x_2) = 5x_1^2 + 8x_1x_2 + 5x_2^2$. In this case, the Hessian matrix is

$$H = \begin{bmatrix} 10 & 8 \\ 8 & 10 \end{bmatrix}$$

with eigenvalues $\mu_1 = 2$ and $\mu_2 = 18$.

Our neural network may be diagrammed as

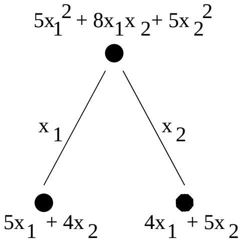

Figure 2: Example neural network

Note that the $x_1$ and $x_2$ are the input values as well as the network weights. Given an initial input/weight vector $(x_1^0, x_2^0)$ and a learning parameter $\nu$, the network will search for the value $(x_1^*, x_2^*)$ which minimizes $g(x_1, x_2)$ by iteratively updating the set of weights. The updating equations are given by

$$\begin{pmatrix} x_1^{i+1} \\ x_2^{i+1} \end{pmatrix} = \begin{pmatrix} x_1^i \\ x_2^i \end{pmatrix} - \eta \begin{pmatrix} \frac{\partial g((x_1, x_2))}{\partial x_1} \\ \frac{\partial g((x_1, x_2))}{\partial x_2} \end{pmatrix}_{(x_1^i, x_2^i)}$$

$$= \begin{pmatrix} x_1^i \\ x_2^i \end{pmatrix} - \eta \begin{pmatrix} 10x_1^i + 8x_2^i \\ 8x_1^i + 10x_2^i \end{pmatrix}$$

where $(x_1^i, x_2^i)$ denotes the value of $(x_1, x_2)$ after the $i^{\text{th}}$ iteration and $\eta$ is chosen such that

$$0 < 1 - \eta\mu_i < 1 \text{ for } i = 1, 2$$

i.e.,

$$0 < 1 - \eta(2) < 1 \text{ and } 0 < 1 - \eta(18) < 1 \quad \Rightarrow \quad 0 < \eta < 1/18$$

Note that

$$\begin{aligned} g(x_1, x_2) &= 5x_1^2 + 8x_1x_2 + 5x_2^2 \\ &= 5x_1^2 + 2 * \sqrt{5} * \tfrac{4}{\sqrt{5}} * x_1x_2 + \tfrac{16}{5}x_2^2 + (5 - \tfrac{16}{5})x_2^2 \\ &= (5x_1 + \tfrac{4}{\sqrt{5}}x_2)^2 + \tfrac{9}{5}x_2^2 \geq 0 \end{aligned}$$

so $(x_1^*, x_2^*) = (0, 0)$. Hence we wish to choose $(x_1^0, x_2^0)$ as a value which lies within an open disk around (0,0).

Given that $g(x_1, x_2)$ has a positive definite Hessian matrix, if we (1) choose $\eta$ such that $0 < \eta < 1/18$, (2) choose $(x_1^0, x_2^0)$ within an open disk around (0,0), and (3) update the network weights using the given formulas, then the above theory ensures that

$$(x_1^n, x_2^n) \rightarrow (x_1^*, x_2^*) \text{ as } n \rightarrow \infty$$

## 5.3 Results

For our experiments, we chose three initial values for $(x_1, x_2)$ and three values for $\eta$. The value $\eta = 0.2$ is outside of (0,1/18) and was chosen for comparison purposes. The program was written so that the algorithm was considered to have converged as soon as the updating values in the weight updating equations dropped below a tolerence value, i.e., if $\tau$ denoted the tolerence value (supplied by the user), the program would stop when

$$10x_1^i + 8x_2^i < \tau \text{ and } 8x_1^i + 10x_2^i < \tau$$

Our experimental results are shown below. Note that n = number of iterations for updating values to fall below tolerence (i.e., until the algorithm had reached desired convergence) and D denotes scientific notation, e.g., 8.05D-2 = $8.05 * 10^{-2}$. No tolerence value was set for $\eta = 0.2$.

| $(\mathbf{x_1^0}, \mathbf{x_2^0})$ | $\eta$ | $\mathbf{x_1^n}$ | $\mathbf{x_2^n}$ | $\mathbf{g(x_1^n, x_2^n)}$ | $\tau$ | $\mathbf{n}$ |
|---|---|---|---|---|---|---|
| (1,1) | 0.05 | 9.99D-26 | 9.99D-26 | 1.8D-49 | 0.05D-5 | $< 25$ |
| | 0.10 | 2.037D-10 | 2.037D-10 | 7.47D-19 | 0.05D-5 | $< 100$ |
| | 0.20 | 3.14D+41 | 3.14D+41 | 1.778D+84 | $*****$ | $< 100$ |
| (2,-3) | 0.05 | 6.64D-05 | -6.64D-05 | 8.818D-09 | 0.05D-3 | $< 100$ |
| | 0.10 | 4.07D-10 | -6.11D-10 | 7.05D-19 | 0.05D-6 | $< 100$ |
| | 0.20 | -1.57D+41 | -1.57D+41 | 4.44D+83 | $*****$ | $< 100$ |
| (-0.5,-1) | 0.05 | 6.64D-06 | -6.64D-06 | 8.81D-11 | 0.05D-4 | $< 100$ |
| | 0.10 | -1.01D-10 | -2.03D-10 | 4.25D-19 | 0.05D-6 | $< 100$ |
| | 0.20 | -2.35D+41 | -2.35D+41 | 1.00D+84 | $*****$ | $< 100$ |

Table 1: Experimental Results

Our method did converge to the minimum function value for the given starting values and the values of $\eta : 0 < \eta < 1/18$. Note that for $\eta = 0.20$, the algorithm diverged instead of converging to the minimum. This is consistent with the above theory.

### 5.4 Comments

1. At first glance, it may appear quite easy to find a function with a positive definite Hessian matrix. However, many simple, 'nice' functions do not have positive definite Hessian matrices. For example, the function $f(x_1, x_2) = x_1^2 x_2^2$ has Hessian matrix

$$\begin{bmatrix} 2x_2^2 & 4x_1x_2 \\ 4x_1x_2 & 2x_1^2 \end{bmatrix}$$

   which is **not** positive definite. Note that with this function it is not possible to chose a starting value $(x_1^0, x_2^0)$ which lies within an open disk of the minimum value. This is because when $x_1 = 0$ (or $x_2 = 0$) there are infinitely many values of $x_2$ ($x_1$) at which the function reaches its minimum. One must be careful when applying the above theory to any problem.

2. For a given starting value $(x_1^0, x_2^0)$, the method outlined above may still lead to the minimum value of a function $g(\underline{x})$ even when the learning parameter $\eta$ does not meet the above specifications. Using the same function as above and a starting value of $(x_1^0, x_2^0) = (2, -2)$, we attained the following results:

| $(\mathbf{x_1^0}, \mathbf{x_2^0})$ | $\eta$ | $\mathbf{x_1^n}$ | $\mathbf{x_2^n}$ | $\mathbf{g(x_1^n, x_2^n)}$ | $\tau$ | $\mathbf{n}$ |
|---|---|---|---|---|---|---|
| (2,-2) | 0.05 | 5.31D-05 | -5.31D-05 | 5.644D-09 | 0.05D-3 | $< 100$ |
| | 0.10 | 4.07D-10 | -4.07D-10 | 3.319D-19 | 0.05D-7 | $< 100$ |
| | 0.20 | 1.306D-22 | -1.306D-22 | 3.414D-44 | 0.05D-20 | $< 100$ |

Table 2: Example with $\eta$ outside specifications

The algorithm converged when $\eta = 0.20$ even though this value is outside of our specifications. However, the algorithm is not guaranteed to converge for such values; only if $\eta$ is chosen appropriately does the above theory guarantee convergence.

# 6 Discussion

A relationship between the gradient descent technique and contractive maps has been derived. Noting that gradient descent is used in multilayer perceptron learning and the application of contractive maps gives rise to fractals, the above relationship is a connection between MLPs and fractals. Given the amount of research and attention being devoted to both neural networks and fractals, we hope that this connection will attract the attention of researchers and lead to advancements in both subjects.

Now that a connection has been made between these two subjects, this connection can be examined with future research. The similarity between gradient descent and fractal generation may lead to improvements in the gradient descent algorithm (hence enhancing MLP performance), as well as improvements in other optimization algorithms which use iterative techniques (such as genetic algorithms).

The utility of Theorem 3.2 in modifying the connection weights for real life problems needs to be explored. We would also like to examine what Theorem 3.2 can tell us about the selection of MLP parameters such as $\eta$ (to avoid oscillation or divergence) and how recent advances in Hessian computation may ease implementation.

The research of the authors is supported by the Army Research Office under the Grant DAAH04-96-1-0082.

# References

[1] P. Antognetti and V. Milutinovic (Eds), *Neural Networks : Concepts, applications and implementations Vols. 1, 2, 3 and 4* , Prentice Hall, New Jersey, 1991.

[2] D. E. Rumelhart, J. L. McClelland and the PDP Research Group (Eds), *Parallel Distributed Processing, Vol. 1*, MIT Press, 1986.

[3] M. F. Barnsley, *Fractals Everywhere*, Academic Press, 1993.

[4] T. M. Apostol, *Mathematical Analysis*, Narosa Publishing House, Bombay, 1992.

[5] J. E. Marsden and A. J. Tromba, *Vector Calculus*, W. H. Freeman and Company, New York, 1988.

[6] Christopher M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1995.

[7] Christopher M. Bishop, "Exact Calculation of the Hessian Matrix for the Multilayer Preceptron", *Neural Computation* **4**, 494-501, 1992.

[8] Roberto Battiti, "First- and Second-Order Methods for Learning: Between Steepest Descent and Newton's Method", *Neural Computation*, **4**, 141-166, 1992.

[9] Jörg Wille, "On the Structure of the Hessian Matrix in Feedforward Networks and Second Derivative Methods", *Proceedings of the ICNN*, pgs. 1851-1855, 1997.