Nonlinear Iterated Function System Coding Using Neural Networks

Rashad A. Al-Jawfi
Department of Mathematics, Facuilet of Science, Ibb University, Ibb, Yemen
Email: algofi@yemen.net.ye

Abstract—In this paper we attempt to form a neural network to coding non linear iterated function system. Our approach to this problem consists of finding an error function which will be minimized when the network coded attractor is equal to the desired attractor. First we start with a given iterated function system attractor; with a random set of weights of the network. Second, we compare the consequent images using this neural network with the original image. On basis of the result of this comparison we can update the weight functions and the code of the non-linear iterated function system (NLIFS). A common metric or error function used to compare between the two image fractal attractors is the Hausdorff distance. The error function gives us good means to measurement the difference between the two images.

I. INTRODUCTION

The basic concept of fractals is that they contain a large degree of self-similarity. This means that they usually contain small copies of themselves buried deep within the original.

On the other hand, neural networks have been hailed as the paradigm of choice for problems which require "Human Like" perception. A network could be performing its function perfectly, responding correctly to every input that it is given. However, its internal workings could still be construed as a black box, leaving its user without knowledge of what is happening internally.

We are interested in different ways to tease neural networks open, to analyze what they are representing, and how they are "thinking".

In this work we present a novel algorithm to introduce the code of non linear iterated function system which generates a fractal image. Its features being that it is exact fully describing a network's function, as concise, and not an incremental collection of approximations and direct mapping of a network's input to its output.

II. FRACTAL CODING

A. Nonlinear iterated function systems

An IFS is specified by n contractive, affine transformations $T_i$, $1 \leq i \leq n$. Each transformation $T_i$ has the form $T_i(p) = A_i p + E_i$, where

$$A_i = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \quad \text{and} \quad E_i = \begin{pmatrix} e_i \\ f_i \end{pmatrix}.$$ Each $T_i$ has a fixed point $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ that is mapped to itself under $T_i$, i.e. $T_i\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = A_i \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + E_i = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$.

Solving the equation defining $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ gives

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \frac{(b_i f_i + c_i(1 - d_i))/\delta}{(e_i f_i + c_i(1 - a_i))/\delta}.$$ Where $\delta = (1 - a_i)(1 - d_i) - b_i c_i$.

The effect of $T_i$ on two points $p$ and $q$ is to map them to two points $p'$ and $q'$ that are closer together than $p$ and $q$. The contractivity factor of $T_i$ is the least number $s_i$ such that $p' q' \leq s_i \overline{pq}$.

The attractor of an IFS can be obtained as follows[11]:

Let $D$ be any disk with $T_i(D) \subseteq D$, $1 \leq i \leq n$. The attractor is given by $\cap_{i=1}^n T^m(D)$ Where $T(A) = \bigcup_{i=1}^n T_i(A)$.

The nonlinear iterated function systems which we will coding can be introduced as

$$T_i(p) = A \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} + B \begin{pmatrix} x \\ y \end{pmatrix} + C [x y] + D.$$ Where

$$A = \begin{bmatrix} a_i & b_i \\ a_2 & b_2 \end{bmatrix}, \quad B = \begin{bmatrix} c_i & d_i \\ c_2 & d_2 \end{bmatrix}, \quad C = \begin{bmatrix} e_i \\ e_2 \end{bmatrix}, \quad D = \begin{bmatrix} f_i \\ f_2 \end{bmatrix}.$$

B. The inverse problem of fractals

The inverse problem of generating fractals, that is, For a given set in $\mathbb{R}^n$, construct a suitable IFS whose attractor is the given set (to a certain desired degree of accuracy)[1].

The tackling of this inverse problem, as it stands, is difficult, if it is not impossible. However, if the given set is self-similar, then the required construction is almost
straightforward. The IFS can be found easily by making mathematical translation of the property of self-similarity.

C. Collage theorem [2]

Let \((X, d)\) be a complete metric space, let \(F = \bigcup_{m=1}^{n} f_m\) be an IFS with contraction factor \(r\), and fixed point \(T_0\). Let \(T\) be a closed subset of \(X\). Let \(\varepsilon > 0\) be any positive number, and suppose that the \(\{f_i\}\) are chosen such that
\[
d(T, F(T)) < \varepsilon
\]
Then
\[
d(T, T_0) < \frac{\varepsilon}{1-r} \quad \forall x \in X
\]

III. USING NEURAL NETWORKS TO CODING NONLINEAR ITERATED FUNCTION SYSTEM

A. Design of Neural Network

B. The Hopfield network uses the fixed points of the network dynamics to represent memory elements. Networks studied by Pollack [8], and Giles [5] use the current activation of the network as a state in a state machine while using the dynamics of the network which is treated as an Iterated Function system that is coding for its fractal attractor[2]. Melnik [6], applies one of the transforms on a random point for a number of steps, until it converged. Al-Jawfi[10], introduce a neural network to coding linear iterated function system using the whole shape of fractal image.

There is still no general algorithm for nonlinear iterated function system attractor coding, the problem we want to solve in this paper, which is given a nonlinear fractal attractor, find a set of weights for the neural network which will approximate the attractor.

A neural network consists of seven input units and two output units for all nonlinear transform (IFS) which represent a scalar function.

The transform is selected randomly, and all input neurons receive a coordinate of each point of the fractal image, one neuron for \(x\) coordinate and the other for \(y\) coordinate for each transform. The result of this operation is \(x\) and \(y\) output which consists of the sigmoid function [6],[7] with a bias.

\[
\begin{align*}
    X_{out} &= (1 + e^{-W_{in}})^{-1} \\
    Y_{out} &= (1 + e^{-W_{in}})^{-1}
\end{align*}
\]

where

\[
WX_{in} = w_{1x} \cdot X^2 + w_{1y} \cdot Y^2 + w_{xx} \cdot X + w_{xy} \cdot Y + w_{y}
\]

And

\[
WX_{in} = w_{1x} \cdot X^2 + w_{1y} \cdot Y^2 + w_{xx} \cdot X + w_{xy} \cdot Y + w_{y}
\]

Where \(w_{something}\) is the weights showed in the following figure.

At the end of this operation for a large number of points with random iterations, we get an image. Of course this image is different in general from the image we want to find its nonlinear iterated function system (NLIFS).

Furthermore, we must update the weight functions of the neural network \(w_{ijk}\) to get better approximation to the target image.

This change of the weight function depends with the measure of the difference between the two images. This difference is known as the error function, which must minimize with every update of the weight functions.
The error function, used to compare fractals attractors, is the Hausdorff distance [2],[4],[5].

The distance between the two images A and B is calculated as follows:

We first calculate the distance between the element \( a \in A \) and the set \( B \) which is the smallest distance between \( a \) and each element \( b \in B \).

\[
d(a, B) = \min \{ \| b - a \| ; b \in B \}
\]

Then the distance between A and B is the largest distance for each element \( a \in A \).

\[
d(A, B) = \max \{ d(a, B) ; a \in A \}
\]

Also the distance between B and A is equal to

\[
d(B, A) = \max \{ d(b, A) ; b \in B \}
\]

And then the distance between the two sets is the largest of the tow distances \( d(A, B) \) and \( d(B, A) \)

\[
H(A, B) = \max \{ d(A, B), d(B, A) \}
\]

Hence, our error function is defined as:

\[
E(T, A) = \sum_i d(T_i(x, y), A) + \sum_{x,y} d((x, y), T(A))
\]

Where \( T_i(x, y) \) is the image of the point \( (x, y) \) with respect to the transform \( T_i \). And \( T(A) \) is the whole image of A.

C. The procedure of non linear iterated function system coding

1. Input: fractal image, random weights, \( n \) the number of (IFS), \( \varepsilon \) and \( \mu \).
2. Compute the error function \( E(A, T(A)) \).
3. If \( E > \varepsilon \) then
   - Compute \( \frac{\partial E}{\partial W_{ij}} \), \( \Delta W_{ij} = -\mu \frac{\partial E}{\partial W_{ij}} \)
   - Update the weights \( W_{ij} = W_{ij} + \Delta W_{ij} \)
   - Go to 2
4. Else: stop

D. Numerical result

The value of the error function with respect to the iteration of the neural network is shown in table (3.1) below.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.84945</td>
</tr>
<tr>
<td>2</td>
<td>2.76882504902661</td>
</tr>
<tr>
<td>3</td>
<td>1.16086447780231</td>
</tr>
<tr>
<td>51</td>
<td>4.84525451424714 ( \times 10^{-2} )</td>
</tr>
<tr>
<td>52</td>
<td>4.72791469254315 ( \times 10^{-2} )</td>
</tr>
<tr>
<td>100</td>
<td>2.35102508927172 ( \times 10^{-2} )</td>
</tr>
<tr>
<td>101</td>
<td>2.33305036520368 ( \times 10^{-2} )</td>
</tr>
<tr>
<td>127</td>
<td>2.00198962949984 ( \times 10^{-2} )</td>
</tr>
<tr>
<td>128</td>
<td>1.99307784767636 ( \times 10^{-2} )</td>
</tr>
</tbody>
</table>

Moreover, some of the nonlinear IFS attractor is shown below in figure 3.3 with iteration 3, 70 and 128.

The attractor with 128 iteration is so closed to the original attractor.

With another form of nonlinear polynomial, we can reformed the neural network to be apropos with respect to the number of coefficient of polynomial.
IV. CONCLUSION AND FUTURE WORK

This paper focused on the inverse problem of fractals with related to nonlinear iterated function systems with 2-dimension. Solving the inverse problem of linear and nonlinear 3-dimension fractals remains an open problem.

REFERENCES


