Journal of Engineering and Applied Sciences 15 (6): 1337-1340, 2020
ISSN: 1816-949X
© Medwell Journals, 2020

Approximation of Fractal Interpolation using Artificial Neural Network

Rashad A. Al-Jawfi
Department of Mathematics, Faculty of Sciences and Arts, Najran University, KSA

Abstract: A new fractal interpolation method called NNA (Neural Network Algorithm) based on IFS is proposed to interpolate the self-affine signals with the expected interpolation error. Experiments on the theoretical data and real field seismic data show that the proposed interpolation schemes can not only get the expected point's value but also get a great accuracy in reconstruction of the seismic profile, leading to a significant improvement over other trace interpolation methods.

Key words: Fractal interpolation, neural network

INTRODUCTION

The approximation of natural complex shapes constitutes an important research area for reconstruction and representation of problems in several application domains (medical imaging (Zair and Tosan, 1996), multimedia data representation (Zair and Tosan, 1997a, b), computer assisted design). In fractal theory, the determination of an Iterated Function System (IFS) Model for approximating natural data, is called "the inverse problem" (Wang, 2000). The fractal image coding techniques first introduced by Jacquin (1992) constitute an efficient example for this approach. Fractal approximation technique, although, it has the advantage of describing self-similar objects, suffer from an important drawback consisting in a lack of control on the fractal figures to be described.

This is essentially due to the use of the contraction affine operators defined in a reduced space, i.e., for images. In contrast, free form approximation methods allow to control the objects with high flexibility via. a set of control points. But only smooth objects can be reconstructed.

Guerin *et al*. (2000) used the concept of control point for approximation fractal interpolation function. Cocharan *et al*., introduced approximated method based on a Hough transform of fractal function transformation parameters. Cheng and Zhu, constituted a multiresolution approximation on an alternative multiresolution approximation to the wavelets and a theoretical basis for the successful applications of the fractal transform algorithm in signal/image compression. In this study, approximation of fractal interpolation functions is introduced by using artificial neural network.

Iterated function systems: Following Barnsley (1986) and Gayatri (2006), under fractals, we mean compact subsets of a complete metric space.

An Iterated Function System (IFS), Gayatri (2006) is a structure $(X, F) = \{f_1, f_2, ..., f_N\})$ consisting of continuous function $f_i: X \rightarrow X$, $i = 1, 2, ..., N$, acting from a complete metric space (X, d) with metric d, into the space itself. In the case when functions f_i are contractions, there is a set $A \subseteq X$, such that $f_i(A) \subseteq A$, $\forall i \in \{1, 2, ..., N\}$ which is called the attractor of the IFS.

The induced operator $F = \bigcup_{i=1}^{N} f_i$ acts on another metric space H(X) of compact subsets of X, (H(X), h) being complete with respect to the Hausdorff metric h, $F: H(X) \rightarrow H(X)$. Associated with every IFS F is a unique compact set A_F, the attractor of IFS:

$$\exists A_F \in H(X): A_F = F(A_F)$$

The Hausdorff metric is defined as follows:

$$h(A, B) = \max(d_s(A, B), d_s(B, A)), \quad A, B \in H(X)$$

Where:

$$d_s(A, B) = \max_{x \in A} \min_{y \in B}(d(x, y))$$

We are dealing here with hyperbolic IFSs with all f_i affine transforations. To build the fractal described by and IFS F, we take an initial $A_0 \in H(X)$ and define $A_{n+1} = F(A_n)$, $n = 0, 1, ..., \infty$.

The sequence $\{A_n\}_{n=0}^{\infty}$ converges to A_F in the Hausdorff metric. After the great enough number N of iterations, we have $F^N(A_0) \cong A_F$.

Definition (Barnsley 86): Let (M, d) be a metric space and let F be the family of all closed subsets of M. For $r>0$ and A in F, let $V_r(A) = \{m: d(m, F)<r\}$ and definite for members A and B of F, d' (A, B) = inf $\{r: A \subset V_r(B)$ and $B \subset V_r(A)$. Here, d' is the well-known Hausdorff metric.

Let M be a compact metric space and H be the set of all nonempty closed subsets of M. Then, H is a compact metric space with the Hausdörff metric.

Corresponding Author: Rashad A. Al-Jawfi, Department of Mathematics, Faculty of Science, Ibb University, Yemen

Let $w_n = M \to M$ for $n \in \{1, 2, ..., N\}$ be continuous. $\{M, w_n: n = 1, 2, .., N\}$ is called an Iterated Function System (IFS).

Consider $w_n(A) = w_n(x)$, $x \in A$. Define $W:H \to H$ by $W(A) = \bigcup_n w_n(A)$ for $A \in H$. Any set $G \subset H$ such that $W(G) = G$ is called an attractor for the IFS. An IFS is called hyperbolic if, for some s, $0 \leq s < 1$ and $n \in \{1, 2, ..., N\}$, $d(w_n(x), w_n(y)) \leq s.d(x, y) \forall x, y \in M$. In this case W is a contraction mapping which obeys $h(W(A), W(B)) \leq s.h(A, B)$, $\forall A, B \in H$.

Also, W admits a unique attractor. Barnsley (1986) explains how to find this unique attractor.

Iterated function systems describe fractal interpolation functions: Let us give a set $\{xi, f(xi)| i = 0, 1, 2, ..., N\}$ of data points and we have to find a continuous function $S:[x_0, x_N] \to R$ which interpolates the data, i.e., $S(x_i) = f(x_i)$, $i = 1, ..., N$.

In contrast to the ordinary case where we look for a smooth interpolation function, here, we are interested in finding a fractal interpolation function S. In the special case when an IFS $F = \{R^2; f_1, f_2, ..., f_N\}$ consists of the so-called shear transformations (Zair and Tosan, 1997a, b), F describes a fractal interpolation function F in the sense that its attractor A_F is the graph of the function F. A shear transformation is an affine transformation of the special form:

$$f_i\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}a_i & 0\\c_i & d_i\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}e_i\\f_i\end{pmatrix}$$

constrained by the data according to:

$$f_i\begin{pmatrix}x_0\\f(x_0)\end{pmatrix}=\begin{pmatrix}x_{i-1}\\f(x_{i-1})\end{pmatrix}, f_i\begin{pmatrix}x_N\\f(x_N)\end{pmatrix}=\begin{pmatrix}x_i\\f(x_i)\end{pmatrix}$$

for i = 1, 2, ..., N. It maps lines parallel to the Y-axis into lines parallel to the Y-axis. Parameter d_i serves as the vertical scaling factor. Parameters $\{d_i|i = 1, 2, ..., N\}$ determine the fractal dimension of the attractor A_F of the IFS F.

Basic concepts of artificial neural network: Neural networks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'. Patterns are presented to the network via. the 'input layer' which communicates to one or more 'hidden layers' where the actual processing is done via. a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is output as shown in Fig. 1.

Most ANNs contain some form of 'learning rule' which modifies the weights of the connections according to the input patterns that it is presented with. In a sense, ANNs learn by example as do their biological counterparts (Bressloff and Stark, 1991).

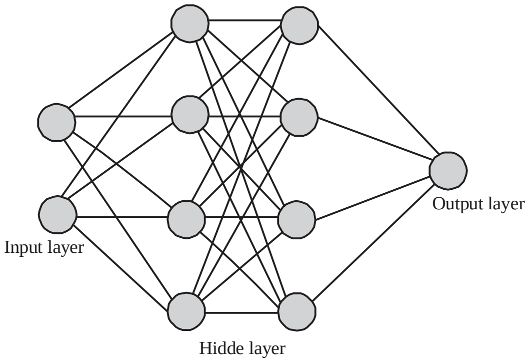

Fig. 1: Neural network

Although, there are many different kinds of learning rules used by neural networks, this demonstration is concerned only with one; the delta rule. The delta rule is often utilized by the most common class of ANNs called 'Backpropagational Neural Networks' (BPNNs). Backpropagation is an abbreviation for the backwards propagation of error.

With the delta rule as with other types of backpropagation, 'learning' is a supervised process that occurs with each cycle or 'epoch' (i.e., each time the network is presented with a new input pattern) through a forward activation flow of outputs and the backwards error propagation of weight adjustments. Once a neural network is 'trained' to a satisfactory level it may be used as an analytical tool on other data.

In comparison, ANNs are not sequential or necessarily deterministic. Neural networks are universal approximators and they work best if the system you are using them to model has a high tolerance to error.

Depending on the nature of the application and the strength of the internal data patterns you can generally expect a network to train quite well. This applies to problems where the relationships may be quite dynamic or non-linear. ANNs provide an analytical alternative to conventional techniques which are often limited by strict assumptions of normality, linearity, variable independence, etc.

Iterated function systems describe neural networks: The base element of a neural network model, a Formal Neuron (Jacquin, 1992), computes $y(x, w) = g(x. w)$ the inner product of its input vector $x = (x_0, x_1, ..., x_n)^T$ and its synaptic weight elector $w = (w^0, w^1, ..., w^n)^T$ which then goes through activation function to yield neuron's output. The input $x_0 = 1$ and the corresponding synaptic weight w_0 is called the threshold of the neuron. As activation function g, either Heaviside (step) function or a sigmoid function is commonly used (Fig. 2).

In mathematics, interpolation process is the computation of values between the ones that are known or tabulated using the surrounding points or values (Gayatri,

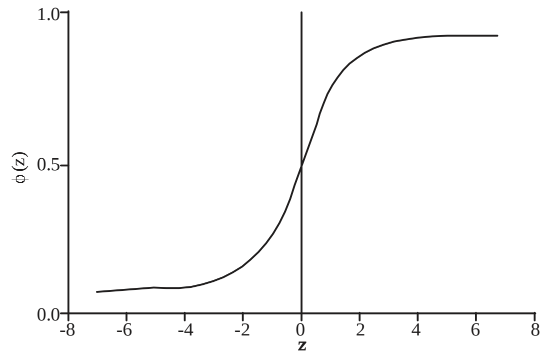

Fig. 2: Sigmoid function

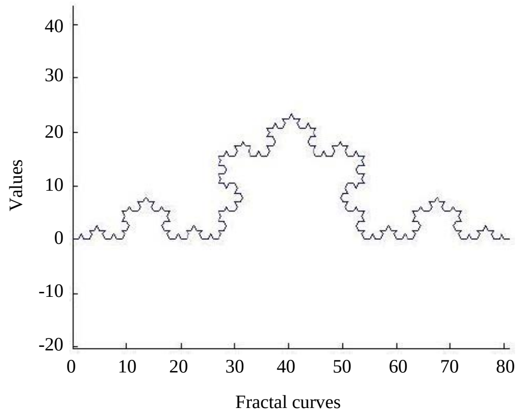

Fig. 3: Snowflake

2006). According to Springer online reference works, "Interpolation is a process of obtaining a sequence of interpolation functions $\{f_n(z)\}$ for some indefinitely growing number of interpolation conditions. The aim of the interpolation process is to approximate by means of interpolation functions $\{f_n(z)\}$ n of an initial function f(z) about which there is incomplete information or which is complicated to deal with directly. Some of the famous interpolation processes are Newton's divided difference interpolation, Aitken's interpolation, Lagrange interpolating polynomial, Bessel's interpolating formula and Gauss's interpolating formula. Computing the parameters of each transformations g_1 and g_2 if $\delta \leq a$ for small real number a.

A neural network model (Bressloff and Stark, 1991) is an interconnection of neurons. IFSs describe recurrent models. To be more precise, they describe binary recurrent asymmetric neural networks.

Koch snowflake is one of the earliest mathematical fractal curves introduced by Helge von Koch in 1904 (Barnsley, 1986). The curve is constructed in this way (Fig. 3):

MATERIALS AND METHODS

The proposed technique for approximation neural curve use the artificial neural network can be introduce as the flowing: for the two affine transformation:

$$f_1\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_1 \\ f_1 \end{bmatrix}$$
$$f_2\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_2 \\ f_2 \end{bmatrix}$$

Reading three points from the original curve denoted by p_1, p_2 and p_3. For flexibility one can put p_1 as the original point. Letting $g_1(p_1) = p_1$ and $g_2(p_1) = p_2$, Then:

$$\begin{bmatrix} e_1 \\ f_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} e_2 \\ f_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

Taking $g_1(p_3) = p_2$ and $g_2(p_3) = p_3$, Then:

$$\begin{bmatrix} e_1 \\ f_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} e_2 \\ f_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

then:

$$\begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix}\begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix}\begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}$$

Therefore:

$$\begin{aligned} a_1x_3 + b_1y_3 &= x_2 \\ c_1x_3 + d_1y_3 &= y_2 \\ a_2x_3 + b_2y_3 + x_2 &= x_3 \\ c_2x_3 + d_2y_3 + y_2 &= y_3 \end{aligned}$$

Transforming the system into the artificial neural network diagram, n = 2 and m = 4 (Fig. 4 and 5). Letting $b_1 = 0$ and $b_2 = 0$, then, n = 2, m = 2 and:

$$a_1 = \frac{x_2}{x_3},\ a_2 = \frac{x_3 - x_2}{x_3}$$

And then, we have four unknown variables and two equations. If we take $(x_i, y_i)_{i=1}^{N}$ to be the original data and $(x'_i, y'_i)_{i=1}^{N}$ to be the approximation data, so the total error δ is given by Fig. 6:

$$\delta = \sqrt{\sum_{i=1}^{N}\left(y'_i - y_i\right)^2}$$

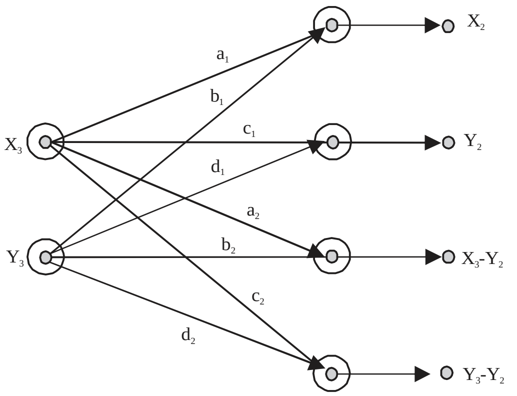

Fig. 4: Neural network for fractal interpolation

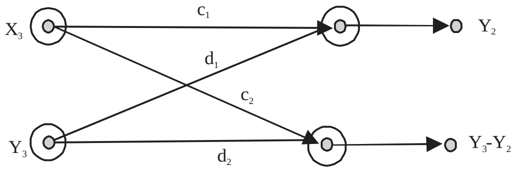

Fig. 5: Neural network

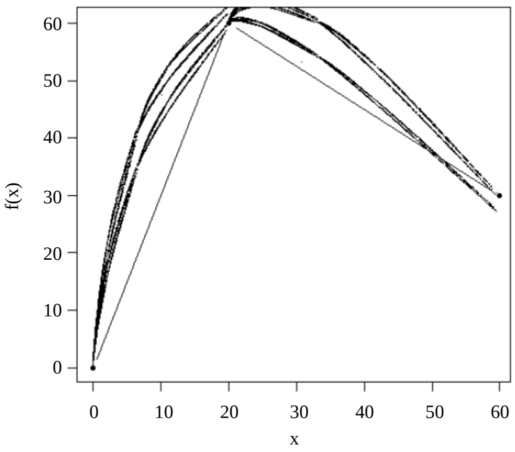

Fig. 6: Fractal interpolation

CONCLUSION

In this study, we have presented a new artificial neural network method to finding the parameters of fractal interpolation functions. For approximation fractal interpolation functions in two dimension, an artificial neural network technique was used and represented, the results illustrate that the proposed interpolation algorithm yields a more significant improvement in precision than the methods in the previous literature.

ACKNOWLEDGEMENT

This research is supported by the Scientific Research Deanship at Najran University, Kingdom of Saudi Arabia (KSA), research project number NU/ESCI/16/094.

REFERENCES

Barnsley, M.F., 1986. Fractal functions and interpolation Constructive Approximation, 2: 303-329.

Bressloff, P.C. and J. Stark, 1991. Neural Networks, Learning Automata and Iterated Function Systems. In: Fractals and Chaos, Crilly A.J., R.A. Earnshow and H. Jones (Eds.). Springer, New York, USA., ISBN: 978-1-4612-7770-5, pp: 145-190.

Gayatri, R.S., 2006. Fractal interpolation. M.Sc. Thesis, University of Tennessee, Knoxville, Tennessee.

Guerin, E., E. Tosan and A. Baskurt, 2000. Fractal coding of shapes based on a projected IFS model. Proceedings of the 2000 International Conference on Image Processing (Cat. No.00CH37101), September 10-13, 2000, IEEE, Vancouver, Canada, pp: 203-206.

Jacquin, A.E., 1992. Image coding based on a fractal theory of iterated contractive image transformations. IEEE Trans. Image Process., 1: 18-30.

Wang, J.J., 2000. Invertibdity in fractal geometry. Ph.D, Thesis, Foreign Language Training Center, National Sun Yat-sen University, Kaohsiung, Taiwan.

Zair, C.E. and E. Tosan, 1996. Fractal modeling using free form techniques. Comput. Graphics Forum, 15: 269-278.

Zair, C.E. and E. Tosan, 1997b. Computer Aided Geometric Design with IFS Techniques. In: Fractals Frontiers, Novak, M.M. and T.G. Dewey (Eds.)., World Scientific Publishing, Singapore, pp: 443-452.

Zair, C.E. and E. Tosan, 1997a. Unified IFS-based model to generate smooth or fractal forms. Surf. Fitting Multiresolution Methods, 1: 345-354.