Proceedings of the 14th
KACC, Octorber 1999

Forecasting and precision on using multi-layer neural network

Hanxi ZHU, Tomoo AOYAMA, and Ikuo YOSHIHARA
The Faculty of Engineering, Miyazaki University
Gakuen Kibanadai-nishi 1-1, Miyazaki 889-2192, Japan
E-mail: aoyama@esl.miyazaki-u.ac.jp

Abstract

Forecasting and extrapolation for time dependent phenomena by using Multi layer neural network has been studied. We calculated values of a function at short intervals, and made one dimensional vector whose elements were a partial gather of the values. If there is anything same as the future of the functions exists in the fragment set, it is possible for us to have an advanced precision extrapolation. Otherwise, if the approximate function of the primitive function can be constructed by learning the short interval in the network, the precision of extrapolation also can be well realized.

1. Introduction

Many studies of the extrapolation for time dependent phenomena have been published[1]. In the studies, there were some methods that are adopted neural networks. Using neural networks, explicit descriptions (functions) are not necessary. Futures of phenomena are predicted from observed data only. It is very useful and practical, but the usefulness is a defect at same time. If explicit functions are unknown, it is hard to calculate precision of extrapolations. We often find examples of prediction in published papers[1], but hardly discover examinations using neural networks. We consider that the neural network is one of functions. The function is defined at learning data, and it is discrete essentially. But we use it as continuous, and make predictions. It is unreasonable. If the neural network is a continuous function, its differential should be defined. Where we don't consider fractal functions. The explicit differential representation is published [2]. Then, function characters near discrete points of learning data are evaluated by the differential. The fact makes a discrete function into continuous, at least, an index for predictions is got.

2. Multi layer neural networks

We used a neural network[3] without feedback loop, whose structure is three layer. In the layer structure, informed propagations are following.

$\{x1,x2,....xn\} \equiv \chi$
$yj=\sum Vjixi$
$pj=f(yj)$

where a vector {x1,x2,....xn} has an individual information, and its suffix corresponds to numbered neurons in the first layer. The neurons don't include bias ones. [We substitute the bias neurons for threshold-values of actions of neurons.] Suffix "i" and "j" are used for 1st and 2nd layer, respectively. Vij is a matrix of connection weights between neurons in 1st and 2nd layer. "yj" is a variable as temporary defined. And f() is a function simulated a neuron, which must be a differential function. If it is not differential, following learning equations are not defined. We name the function as neuron-function. "pj" is a vector for output of neurons in 2nd layer. Thus, we got following relations for informed propagations between 2nd and 3rd layers.

$\{p1,p2,....,pm\} \equiv \rho$
$qk=\sum Wkjpj$
$ok=g(qk)$
$\{o1,o2,...ok\} \equiv o$

Where Wkj is a matrix of connection weights, and g() is a neuron-function. Suffix "k" is used for 3rd layer, so a vector {o1,o2,...ok} is output for neurons in 3rd layer. In the neural network, an individual information {x1,x2,....xn} is transformed into a vector {o1,o2,...ok}. Where dimension of the vector is converted into different one, therefore, we must take care of the transformations in cases of "n < k" and large "m"-values.

These relations are realized in one individual input/output datum, and the relations also stand up in case of plural data. Thus, we write as following.

$\chi \rightarrow \{\chi\mu\},\ \rho \rightarrow \{\rho\mu\},\ o \rightarrow \{o\mu\}$

In the multi layer neural network, a corresponding relation is organized.

$\{\chi\mu\} \Leftrightarrow \{o\mu\}$

The relation is not a one-to-one correspondence. Followings are allowed.

$\{o\mu\} = \{o\nu\} = ... = \{o\xi\}$

But, next relations are not done.

$\{x\mu\}\neq\{x\nu\}\neq\{x\xi\}$

Differential coefficients between output and input are $\partial ok/\partial xi=\sum Wkjg'(qj)Vjif'(yj)$,
where f' and g' are differentials for f and g functions. By using vector representations, we get $\partial o/\partial x = W g'V f'$.
The neuron-functions f and g are sigmoid functions generally. But it is desirable that the functions are replaced by other ones. We often replace g-function with linear function. Such neural network is called ANN, which is excellent to extrapolate on various phenomena.

3. A recurrent representation for functions

3.1 Takens's embedding theorem

Takens' embedding theorem teaches us possibility for the short range prediction of time series data gotten by chaotic phenomena. The embedding theorem shows that the trace described by multi dimensional variables is calculated by one kind of variable in them. It is impressive and extensions of the theory are broad and general. Principles of the theory are based on the Jacobian matrix defined nearby the last point. It is possible that the principle is extended by using facilities of neural networks. The neural networks evaluate status around learning points from all data, and they are not local but global. Therefore, we believe significance in investigations of the prediction use of neural networks.

3.2 A recurrent representation for functions

We investigate a recurrent representation of functions in order to eliminate periodic conditions for functions. The representation is gotten from following facts,
(1) Sampling values of function are finite vector {x1,x2,....xn},
(2) A set of partial vector of the vector is written as {{x1,x2,...xk},{x2,x3,...xk+1}, ...},
(3) Elements of the set can be corresponded to a scalar series {xk+1,xk+2,...} as following: {{x1,x2,...xk}<-->xk+1,{x2,x3,...xk+1}<-->xk+2,....} They are set of partial vector and scalar series, and fragmentations of functions, which are learning data set for a neural network simultaneously. A property of neural networks combines a vector with another, which includes a scalar. That is {xi,xi+1,...xi+k-1}-->xi+k, where i=1,2,.... The relation is equivalent to a local prediction or fragmented forecasting. If the last partial vector is written as {xj,xj+1,...xj+k-1}, and the corresponding scalar is xj+k. In that case, what kind of character does the vector {xj+1,xj+2,...xj+k} have? The {xj+1,xj+2,...xj+k} is a new undefined vector, but it is also an input datum for neural network. Therefore, a new scalar is calculated from the input datum. Is the scalar written as xj+k+1? In generally, it cannot be allowed. But if the sampled function is periodic and the new vector is included in learning data set, it will be allowed. In such a situation, the calculated scalar is xj+k+1, and a new vector {xj+2,xj+3,...xj+k+1} is also used for advanced prediction. These iterative series describe a periodic function implicitly. We call the iteration as a recurrent representation for the function. When the representation is valid, a multi layer neural network is nearly equal to the sampled function.

4. Precision for recurrent representation

In neural networks, input data are multi dimensional. In order to simplify following discussion, we put the multi dimensional data on a multi dimensional space. From now on, input data are points on the space, so we write learning data as learning points.

4.1 Fractal dimensions

We found an examination for the prediction by using neural networks. It is based on the fractal Brown's function that is a statistical extension of the fractal dimensions.

FB(y)=probability[|1/dx|**H {f(x+dx)-f(x)} < y]

The function "FB()"is a distribution function, and is determined by observed data. Where a variable "H" is related with the fractal dimension, whose details are listed in [2]. The "H" variable is an index that determines reliability for the prediction. It is effective to economic forecasting. However, the index shows linear responses between the linear and random changes. We are sure that the responses are not appropriate for precision index. So we consider a new non-linear index.

4.2 Euclid distance

When sigmoid functions are adopted in neural networks, output responses of the neural network near learning points are roughly equal to that of the points. And Euclid difference increases monotonously with distances from the learning points. The fact suggests implicitly that precision for the recurrent representation can be estimated by the distance. But the distance would not be quantitative index, because of the monotonous increasing only. Because we cannot discover quantitative variance around the learning points, we are sure that Euclid distance is 0th order approximation for the precision. The explicit formula "d" for the 0th approximation is, d=min{D1,D2,....DN}. Where each

"Di" is,
Di={∑j(xj-xi)**2+(xi+1-xj+1)**2+....+(xj+k-1 - xi+k-1)**2}**0.5, and {xi,xi+1,...xi+k-1} (i=1,2,..,N) is the learning data, and {xj,xj+1,.....,xj+k-1} (j=1,2,...,N) is an input point for prediction.

4.3 Differential distance

Revising the defect for the variance around the learning points, we use differential coefficients for neural network. By using analogical derivations for Taylor's expansion, we get following a scheme, {δj,...,δj+k-1}={(xj-xi),(xi+1-xj+1),..,(xj+k-1 - xi+k-1)} where index "i" means the nearest point from the input point. Vector representation is, δ=x'-x, Δ=(∂o/∂x)? δ/1! +(∂**2 o/∂x∂x) [δ×δ]/2! + ..., where symbol "×" means outer (tensor) products for the vector. If the ANN is used, diagonal elements of the second term of Δ are vanished. The scalar "Δ" corresponds with Euclid distance. This scheme is valid on the learning points, but is invalid on far from the points. Such situations will be found during calculations to predict. So we are sure the index Δ is 1st order approximation.

4.4 Invalid for prediction

We defined the nearest point from an input point for prediction, whose index was "i". Similarly, the index of the nearest point for next prediction is defined, and it is "i'" here. The location of the index "i'" should be onward from "i", or be equal to "i". We write the condition as "i' >= i". In learning process of the neural network, such a sequent is not explicitly. However, when we consider the recurrent representation for the function, the sequent is included in the consideration implicitly. If the sequent property is break, it is sure that a new aspect is arisen in the prediction. The prediction includes new information which are not found in learning data. Therefore we should reject the prediction.

In ANN only, the output is out of range [0,1] because of linear "g"-function that is a neuron function in the 3rd layer. If the case happens, it shows that the prediction contravenes the definitions for the neural network. Then, we should regard the prediction as invalid.

4.5 Two conceptions for prediction

When we forecast phenomena by using neural networks, at first we construct a neural network by learning equations. The forecasting are evaluated based on the constructed network, at the time there are two choices for adopting prediction data.

One is used last observed data just before the forecasting. The conception is called as "one[4,5] step ahead prediction" or "short range prediction". It is similar to Takens' embedding theory. But, in the prediction principle, a basic quantity is not corresponded with local information such as the Jacobian. Moreover, the quantity is not got by the last data, but by information from rather past data. Then, it is somewhat extension for Takens'.

The other is used output data calculated only by the neural network. The conception is called as "long range prediction". It is out-of-range of Takens' embedding theory, whose efficiency is only examined by numerical calculations. It is natural that the forecasting from the former are higher precision than that of later.

5. Introducing hypothesis

Above mentioned the forecasting by using neural networks are a reasonable extension based on Takens' embedding theory. But there is a limitation on the prediction, because only observed data are used and is not done the properties of phenomena as a hypothesis. Usually introductions of the property have rejected in the traditional way. We find that arbitrariness is arisen in the introductions, and have avoided that. However, we are convinced of necessity for the prediction, and as far as we know, a hypothesis is introduced without the knowledge. This is a problem we cannot avoid, so we will discuss them mathematically, which a hypothesis is introduced under the multi valued logic.

5.1 One-body operators for multi valued logic

There are many operators to calculate multi valued logic. In them, we consider the one-body operators at first, which are not-operator (~) and rotation (R). The not-operator operates the fragments of input vector, and generates new fragments as

~{xi,xi+1,...,xi+k-1}={1-xi,1-xi+1,...,1-xi+k-1}.

Similarly the scalar value xi+k is translated into 1-xi+k. The operations are accepted when observations are symmetric for the mean value.

The rotation operator operates as, R{xi,xi+1,...,xi+k-1}={S+xi,S+xi+1,...,S+xi+k-1}, where "S+" means the addition of a constant "S" to the x-elements. The results seem to be equal to the value-shift, however the operation is based on Post's not-operation whose character is a kind of rotation. The operations are accepted when similar phenomena are observed on the different bias conditions.

5.2 Two-body operators

We consider the two-body operators now. They are "or-" (|) and "and-" operators (&). The "or-" and

"and-" operators operate the fragments as following,
{xi,xi+1,...,xi+k-1}|{xj,xj+1,...,xj+k-1}=
{max(xi,xj),max(xi+1,xj+1)...,max(xi+k-1,xj+k-1)},
xi+k | xj+k=max(xi+k,xj+k),
{xi,xi+1,...,xi+k-1}&{xj,xj+1,...,xj+k-1}=
{min(xi,xj),min(xi+1,xj+1)...,min(xi+k-1,xj+k-1)}, xi+k & xj+k=min(xi+k,xj+k). These operations are accepted when turndown or turn-up is not observed yet, but they are expected.

6. Conclusion

We investigated followings,

(1) Differential coefficients for multi neural networks
(2) Recurrent representations of functions,
(3) Prediction method on use of neural networks,
(4) Precision of the prediction

We introduced (5) a working hypothesis in order to increase learning data, and discussed that the hypothesis was right or wrong. We test these discussions during calculations for predictions of Lorenz's and Rossler's chaos.

References

1. T.Aoyama, Y.Isu, U.Nagashima, "Extrapolations by using neural-networks and a recurrent representations of functions", IPSJ SIG-Note 95-HPC-59 (1995.12.11).
2. T.AOYAMA and H.Ichikawa, "Backpropagation Algorithm Restricted with some conditions", IPSJ Sig-Note, 91-NA-37, pp.59 (1991.7.17).
3. Rumelhart, D.E., McClelland, N.L., Eds., "Parallel Distributed Processing Exploration in Microstructure of Cognition", Cambridge, MA, 1986, Vols.1, Chapter 8.
4. M.Numata, K.Sugawara, I.Yoshihara and K.Abe, "Time Series Prediction by Genetic Programming", Late Breaking Papers at the Genetic Programming 1998 Conference, pp. 176-179, 1998.
5. M.Numata, K.Sugawara, I.Yoshihara and K.Abe, "Time Series Prediction Modeling by Genetic Programming without Inheritance of Model Parameters", Proc. of the Fourth Int.Symp. on Artificial Life and Robotics(AROB-IV), pp. 500-503, 1999.

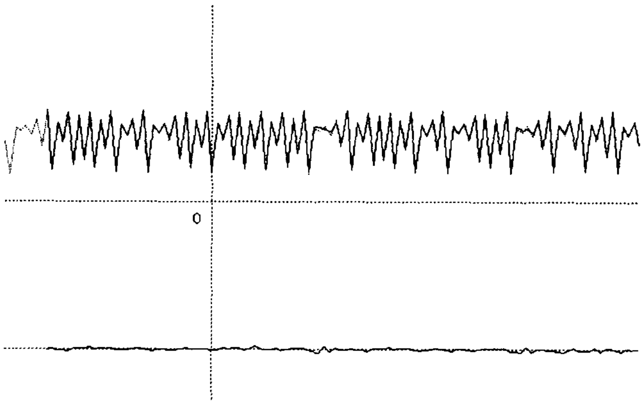

Figure1. forecasting for Lorenz' chaos

The left side of a dotted vertical line means a learning term. The right side is forecasting. Number of the learning data is 32, and forecasting points are 120. In the first graph, the solid curve is results from a neural network, and the plotted curve is true values of the chaos. In the second graph, the solid curve is differences between forecasting values and true ones.

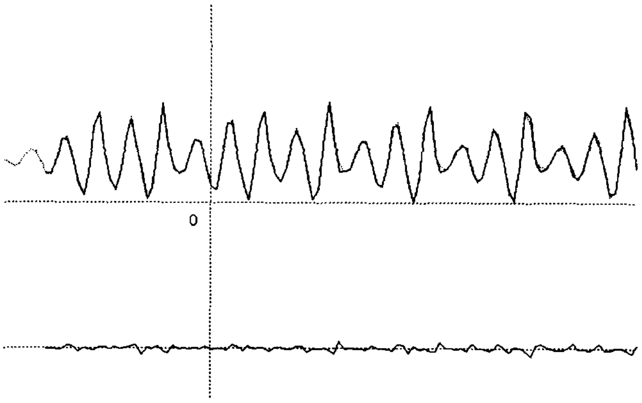

Figure2. forecasting for Rossler' s chaos