Cascade Networks: Another Approach to Function Approximation

Neruda, Roman
1996

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Cascade Networks: Another Approach to Function Approximation

Roman Neruda, Arnošt Štědrý
Institute of Computer Science, Czech Academy of Sciences
Pod vodárenskou věží 2, 182 07, Prague, Czech Republic
roman@uivt.cas.cz, arnost@uivt.cas.cz

Technical report No. V—697

January 13, 1997

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic
phone: (+422) 66414244   fax: (+422) 8585789
e-mail: uivt@uivt.cas.cz

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Cascade Networks: Another Approach to Function Approximation

Roman Neruda, Arnošt Štědrý[1]
Institute of Computer Science, Czech Academy of Sciences
Pod vodárenskou věží 2, 182 07, Prague, Czech Republic
roman@uivt.cas.cz, arnost@uivt.cas.cz

Technical report No. V—697
January 13, 1997

Abstract

A subclass of networks with cascade architecture is presented. We investigate its approximation capabilities by means of a continued fraction framework. It is shown that such network can approximate any meromorphic function arbitrarily well, which in the real domain covers even functions with second order discontinuity. Used techniques imply various alternative learning algorithms that are discussed.

Keywords
Neural Networks, Continue fractions

[1] This research has been partially supported by GA AVČR under grant no. A 203 0602.

1 Introduction

The most widely used and studied types of artificial neural networks have been multilayer perceptrons trained by various modifications of back–propagation algorithm. The learning process of such networks based on gradient descent in high-dimensional spaces is usually very time consuming. That is one of the reasons why cascade architectures of neural networks have appeared recently, together with incremental learning algorithms that gradually add new units to the network in order to achieve a better precision. In each step of these algorithms only the weights corresponding to added unit are changed while the rest of the weights is preserved. A typical example of such network architecture and learning algorithm is Fahlman's Cascor network (Fahlman, 1991). In many applications this network can learn reasonably faster in comparison with multilayer perceptrons. A theoretical background for incremental learning algorithms in multilayer perceptron networks which is based on Jones theorem concerning convergence of iterative approximation in a Hilbert space was laid by Barron, 1993.

It has been shown that multilayer perceptron networks with at least three layers are capable to represent any reasonable input/output function with arbitrary precision. This property, called *universal approximation*, is also true for various other architectures. This is why other criteria for judging the quality of the networks are being investigated. One of them might be the learning time, which seems to be a crucial problem especially in practical applications of neural networks.

In the following we consider architectures with richer topologies than classical multilayer perceptrons, particularly the ones in which connections are not limited only to units in neighboring layers. We call the architectures in which there are lateral connections between the neurons in one layer going from left to right *cascade architectures*.

In this article we introduce a subclass of cascade architectures and describe its approximation capabilities. It is shown that in complex domain any meromorphic function can be approximated arbitrarily well, which in real case includes even functions with the second order discontinuity. Moreover, rational functions are exactly representable by this architecture. The continued fraction calculus used in proofs provides new approaches to learning. In the following we briefly introduce two alternative learning algorithms.

2 Feedforward and Chain Architectures

By *multilayer perceptron* network we mean a network consisting of several layers of units (neurons) connected in a feedforward manner such that each unit in one layer has connections (synapses) with all units in the neighboring layers. It means that synapses between the preceding and the following layer forms a complete bipartite graph. Each connection has a real parameter, called weight, which is a subject of learning. The output y of one unit having inputs x_i with assigned weight values w_i and a threshold b is computed as

$$y = \sigma(\sum_{i=1}^{n} w_i x_i - b),$$

where $\sigma(t)$ is a sigmoidal function, i.e. non-decreasing function with the following limits: $\lim_{t\to\infty}\sigma(t) = 1$ and $\lim_{t\to-\infty}\sigma(t) = 0$. The most popular function—logistic sigmoid—writes as:

$$\sigma(t) = \frac{1}{1+\exp(-t)}$$

In the following, we consider a class of one-hidden-layer networks with a single linear output. The horizontal connections in the hidden layer connect the i-th unit with the $(i+1)$-th one (see Figure 2.1). The hidden units compute functions of the form:

$$o = \delta(z, y) = \frac{az+b}{cz+d+y}, \qquad (2.1)$$

where y corresponds to the output of the preceding unit, z is the actual network input and a, b, c, d are complex parameters. We call a member of this subclass of cascade networks with the activation function of the form (2.1) a *chain network*. On the contrary to real numbers it is not clear which function should be chosen as an activation function of one unit. For example, there is no straightforward complex counterpart of the logistic sigmoid. That is the reason why various activation functions are used (cf. Hirose, 1992a, 1992b)). In our considerations we use the function of the form 2.1 which is widely used in the complex number theory and fits into the framework of continued fractions that will be used in the following. Furthermore, this function—as the simplest possible rational function—is very easy computable but it is still able to approximate discontinuities.

Figure 2.1: Multilayer perceptron and chain network

In order to formulate our results we need to define two classes of functions.

Definition 1 *By $\mathcal{F}$ we denote the set of all functions $f : \mathcal{C} \to \mathcal{C}$ representable by a chain network with any finite number of hidden units.*

Definition 2 *Denote $\mathcal{E}$ the set of functions $f(z) : \mathcal{C} \to \mathcal{C}$ of the form $\frac{p(z)}{q(z)}$, where p and q are polynomials.*

Finally, remind that a *homogeneous polynomial* of n variables $H_n(x_1, \ldots, x_n)$ has a form:

$$H_n(x_1, \ldots, x_n) = \sum_{j=1}^{m} a_j \prod_{i=1}^{n} x^{q_{ij}},$$

such that the sums $\sum_{i=1}^{n} q_{ij}$ are equal for every $j = 1 \ldots m$.

3 Continued Fractions

A function realized by chain network can be straightforwardly represented in the form of a continued fraction which formal definition immediately follows.

Definition 3 *By a* continued fraction *we mean the following formula:*

$$b_0 + \cfrac{a_1(z)}{b_1(z) + \cfrac{a_2(z)}{b_2(z) + \cfrac{a_3(z)}{b_3(z) + \ddots}}} \tag{3.1}$$

where a_i and b_i are complex domain functions of the variable z.

A partial continued fraction *$P_n(z)$ is an expression having only a finite number of coefficients a_i, b_i:*

$$P_n(z) = b_0 + \cfrac{a_1(z)}{b_1(z) + \cfrac{a_2(z)}{b_2(z) + \cfrac{a_3(z)}{\cfrac{\vdots}{b_{n-2}(z) + \cfrac{\vdots}{b_{n-1} + \frac{a_n(z)}{b_n(z)}}}}}}$$

If $lim_{n\to\infty} P_n(z)$ exists and equals v, we say that continued fraction (3.1) has value *v.*

The framework of continued fractions represents a classical part of analytical calculus with basic results dating back to Euler, 1739. We introduce additional definitions and theorems here that are needed in the proof of the main theorem in the following section.

The main relation between a power series and a continued fraction describes the Euler identity (Danilov et al, 1961):

Theorem 4 (Euler) *If expressions on both sides exist then the following equation holds:*

$$\sum_{i=0}^{\infty} c_i z^i = \cfrac{c_0 + c_1 z}{1 - \cfrac{c_2 z}{c_1 + c_2 z - \cfrac{c_1 c_3 z}{\cfrac{\ddots \qquad \vdots}{c_2 + c_3 z - \ldots \qquad - \frac{c_{i-2} c_i z}{c_{i-1} c_i z} - \ddots}}}}$$

The following definition and two theorems from Wall, 1948 show that a power series with negative exponents can be expanded as a continued fraction, as well.

Definition 5 *A sequence of polynomials $B_p(u)(p \leq m)$ is called* orthogonal relative to the sequence *c if*

$$\int B_p(u) B_q(u) d\phi_c(u) \begin{cases} = 0 \ ; p \neq q, p, q \leq m \\ \neq 0 \ ; p = q < m \end{cases}$$

Theorem 6 *A power series $\sum_{i=0}^{\infty}(c_i/z^{i+1})$ can be expanded into a continued fraction if and only if a sequence of polynomials can be constructed which are orthogonal relative to the sequence of coefficients $c_0, c_1, \ldots, c_n, \ldots$.*

Theorem 7 *Let $\sum_{i=0}^{\infty}(c_i/z^{i+1})$ be a power series and Δ_p determinants:*

$$\Delta_p = \begin{vmatrix} c_0 & c_1 & \ldots & c_p \\ c_1 & c_2 & \ldots & c_{p+1} \\ & & \vdots & \\ c_p & c_{p+1} & \ldots & c_{2p} \end{vmatrix},$$

such that:

1. *either $\Delta_p \neq 0$ for every p*

2. *or there is a number m: $\Delta_p \neq 0$ for $p \leq m$ and $\Delta_p = 0$ for $p > m$.*

Then a sequence of polynomials orthogonal relative to the sequence $\{c_0, c_1, \dots\}$ can be constructed.

4 Main results

It is clear from the previous sections that there is a obvious correspondence between functions that are realized by a chain network and continued fractions. Exactly:

Fact 8 *Any function $f \in \mathcal{F}$ has the form of a partial continued fraction P_n, where n is the number of hidden units in the network.*

The following theorem says that functions realizable by chain networks can approximate any meromorphic function. It means that for any function and any precision we can find its approximator $f \in \mathcal{F}$, in another words—there is a network that, in principle, can learn this function arbitrarily well. Due to a limited space we only sketch the main ideas of the proof.

Theorem 9 *A chain network can approximate any meromorphic function with arbitrary precision. Moreover, function of class $\mathcal{E}$ can be exactly represented by a chain network.*

Sketch of proof. The typical way of proving universal approximation property is via the notion of density. In our case we prove that the set of functions $\mathcal{F}$ is dense in the space of meromorphic functions with the uniform metrics.

Any meromorphic function f can be expressed by means of a Laurent series, which is an infinite sum of the form

$$f(z) = \sum_{i=-\infty}^{\infty} a_i z^i = \sum_{i=0}^{\infty} (c_i / z^{i+1}) + \sum_{i=0}^{\infty} c_i z^i.$$

We can divide the sum into two parts according to the sign of exponent i. Now the problem is to realize both sums by continued fractions. Here we use two fundamental theorems that show how to realize the above two power series by continued fractions. The former can be expressed according to the Theorem 7, the latter is rewritten according to the Euler identity (4).

The last technical thing deals with obeying slight conditions of the Theorem 7 and Euler identity. As our series are of a general kind, we have to rewrite them as a sum of three series satisfying necessary presumptions, which finishes the proof. □

So far, we have considered only chain networks with one-dimensional inputs and outputs, but the previous result can be extended for multiple-input networks as well. This is done by means of the dimension reduction, which is a technique based on approximation of multi-variable functions by plane waves. The following theorem is a corollary of a result by Vostrecov and Kreines, 1961.

Theorem 10 *Let W_n is the set of weight vectors of dimension n and $\mathcal{N}(H_n)$ is the set of null points of homogeneous polynomial H_n of n variables. If $W_n \not\subset \mathcal{N}(H_n)$ for any H_n, then there exist continuous functions φ_{ik} of single variable such that the set of functions $\Phi_k(x)$ of the form $\Phi_k(x) = \sum_{i=1}^{k} \varphi_{ik}(w_i \cdot x)$, where $w_i \in W_n$, is dense in the set of all continuous functions.*

Last generalization can be done to multiple outputs, i.e. to vector functions $F : \mathcal{C}^n \to \mathcal{C}^m$, through Cartesian product of one dimensional output spaces. Thus we obtain general networks with multiple inputs and outputs.

5 Learning possibilities

There are various possibilities of learning algorithms for the chain architectures. We just briefly mention an application of three methods here, details can be found in Neruda and Štědrý, 1996.

First, it is possible to derive a variant of the back propagation learning algorithm (Rumelhart et al, 1986) for this architecture. The error propagates from top of the network to the bottom as usual, but it is important that the computation in the hidden layer follows the lateral connections of units. It means that the forward phase goes from left to right (considering the orientation in figure 2.1), while the backward computation proceeds against the direction of horizontal connections.

Imagine an incremental learning algorithm that gradually adds new hidden units placing each one as the leftmost unit in the hidden layer. After adding the unit we keep the old parameters frozen and adapt only the parameters of the new unit. This corresponds to adding a new term to the partial continued fraction while the old terms remain the same. If the adaptation algorithm is reliable[2] this sequence of partial fractions converges to the desirable function. In any case we can rely on some gradient adaptation algorithm with known problems of local minima. Alternatively, the Cascor algorithm can be used since the chain networks create a subclass of cascade networks.

An interesting approach uses a result due to Viskovatov (cf. Danilov et al, 1961) about the continued fraction expansion of rational functions. According to the following formula we can incrementally create a continued fraction based on the coefficients of the rational function:

$$\frac{a_{10} + a_{11}x + a_{12}x^2 + \ldots + a_{1n}x^n + \ldots}{a_{00} + a_{01}x + a_{02}x^2 + \ldots + a_{0n}x^n + \ldots} = \frac{a_{10}}{a_{00} + \frac{a_{20}x}{a_{10} + \frac{a_{30}x}{a_{20} + \ldots}}} \tag{5.1}$$

where

$$a_{mn} = a_{m-1,0}a_{m-2,n+1} - a_{m-2,0}a_{m-1,n+1}. \tag{5.2}$$

Having the training set, we can imagine a gradient learning algorithm that fits the data by a rational function. Using the above formula 5.1 one can directly set the parameters of a chain network. Moreover, it is not necessary to have this interlink,

[2] For instance, suppose quite unrealistic case that the function has a form of a power series. In this case we can use the Euler formula to explicitly compute parameters in each step.

since we can directly derive an adaptive algorithm for the network parameters, which is based on the recursive formula 5.2. So far, this approach works only for one-dimensional case but it would be possible to use the theorem 10 to extend it to a multiple input dimension.

6 Discussion

In our considerations we used the simplest possible activation function of a single unit with respect to the form of continued fraction. It is known from the continued fraction theory, that convergence can be speeded up by the choice of polynomials of higher degree. The concrete usage of such transformation is a question of balance between the complexity of one neuron computations and the number of neurons in the network.

Approximation of functions by perceptron or RBF units can be seen as geometrically intuitive. This is not true in the case of our chain networks, where the activation function, especially in the complex domain, together with the gradual composition does not provide a simple view. On the other hand, it follows from the continued fraction properties that the obtained approximation should be optimal in a sense of the minimization of the number of multiplication and division operations.

Bibliography

[1] Barron, A.R. (1993) Universal approximation bounds for superpositions of sigmoidal function, *IEEE Transactions on information theory*, **39**, 3.

[2] Danilov V.L. et al (1961) *Mathematical analysis*, Moscow, (in Russian).

[3] Euler L. (1739) De fractionibus continuis observatione, *Comm. Acad. Sci. Imper.*, Petropol, **11**.

[4] Fahlman S., Lebiere C. (1991) The cascade correlation learning architecture, In Touretzky D. (ed.) *Advances in Neural Information Processing Systems*, **2**, Morgan–Kaufman.

[5] Hirose, A., (1992a) Dynamics of Fully Complex-Valued Neural Networks, *Electronics Letters*, **16**, 1492–1493.

[6] Hirose, A. (1992b) Continuous Complex-Valued Back-Propagation Learning, *Electronics Letters*, **20**, 1854–1855.

[7] Neruda, R. Štědrý, A. (1995) Approximation Capabilities of Chain Architectures, *Proceedings of the ICANN'95*, EC2 & Cie: Paris, I. 575–580.

[8] Neruda, R. Štědrý, A. (1996) Learning in Cascade Neural Networks, Technical report, Institute of Comp. Sci., Prague.

[9] Rumelhart, D.E., Hinton, G.E., Williams, R.J., (1986) Learning Representations by Back–Propagating Errors. *Nature*, **323**, 533–536.

[10] Vostrecov B. A., Krejnes M. A. (1961) On approximation of continues functions by superpositions of plane waves, *Dokl. Akad. Nauk. USSR.*, **140**, 6, 1237–1240, Moscow.

[11] Wall H.S. (1948) *Analytic Theory of Continued Fractions*, Van Nostrand, New York.