2004년도 대한전기학회 하계학술대회 논문집 2004.7.14-16

The solution of single-variable minimization using neural network

손준혁*, 서보혁**
경북대학교 대학원*, 경북대학교**

(Jun-Hyug Son*, Bo-Hyeok Seo**)
Kyungpook Nat. Univ. Graduation School*, Kyungpook Nat. Univ.**

Abstract - Neural network minimization problems are often conditioned and in this contribution way to handle this will be discussed. It is shown that a better conditioned minimization problem can be obtained if the problem is separated with respect to the linear parameters. This will increase the convergence speed of the minimization. One of the most powerful uses of neural networks is in function approximation(curve fitting)[1]. A main characteristic of this solution is that function (*f*) to be approximated is given not explicitly but implicitly through a set of input-output pairs, named as training set, that can be easily obtained from calibration data of the measurement system. In this context, the usage of Neural Network(NN) techniques for modeling the systems behavior can provide lower interpolation errors when compared with classical methods like polynomial interpolation. This paper solve of single-variable minimization using neural network.

1. Introduction

Determination of the minimum and its location of a real-valued function of one variable plays an important role in nonlinear optimization. A one-dimensional minimization routine may be called several times in a multi-variable problem. We show that at the minimum point os a smooth function, the slope is zero. Interval reduction and polynomial fitting methods are presented for finding the minimum. Furthermore, the need to determine the zero of a function occurs frequently in nonlinear optimization. Reliable and efficient ways of finding the minimum or a function are necessary for developing robust techniques for solving multi-variable problems[2-4].
Their main advantages are the ability to generalize results obtained from known situations to unforeseen situations, fast response time in operational phase, due to a high degree of structural parallelism, reliability and efficiency. For these reasons, application of neural networks has emerged as a promising area of research since its adaptive behavior has the potential of conveniently modeling strongly nonlinear characteristics. When compared with polynomial interpolation, interpolation based on neural networks can reach the best curve fitting without the well-known problems of numerical instability caused by a high-degree polynomial interpolation[5-6].

2. Single-variable minimization

We present the minimization ideas by considering a simple example. The first step is to determine the objective function that is to be optimized. One-variable plot of the function over a sufficiently large range of x shows the distinct characteristics. The one-variable plotting routine PLOT1V included on the accompanying dick may be used to get a visual picture on the computer screen. The one-variable problem may be stated as

$$\text{minimize } f(x) \text{ for all real } x \tag{1}$$

The point x^* is a weak local minimum if there exist a $\delta > 0$ such that $f(x)^* \le f(x)$ for all x such thar $|x - x^*| < \delta$, that is $f(x^*) \le f(x)$ for all x in a δ-neighborhood of x^*. x^* is a strong local minimum if there exists a $\delta > 0$ such that $f(x^*) < f(x)$ for all x. These cases are illustrated in Fig. 1. If a minimum does not exist, the function is not bounded below.

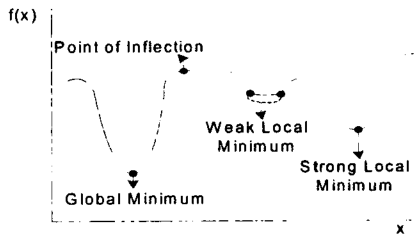

Fig. 1 One-variable plot

2.1 Optimality Conditions

Function f is C^1 continuous, then the necessary condition for x^* to be a local minimum is

$$f'(x^*) = 0 \tag{2}$$

where $f'(x^*)$ represents the derivative $\frac{df}{dx}$ evaluated at $x = x^*$. Note also that $f''(x^*) = \frac{d^2 f}{dx^2}$ evaluated at x^*.

The condition in Eq. (2) is obvious from a graph of the function. In fact, the slope is zero at minimum, maximum, or inflection points.
It is instructive to see how this condition may be derived using Taylor-series expansions about x^*. For a small real number $h > 0$, we use the first-order expansion at $x^* + h$ and $x^* - h$.

$$f(x^* + h) = f(x^*) + hf'(x^*) + O(h^2) \tag{3}$$
$$f(x^* - h) = f(x^*) - hf'(x^*) + O(h^2) \tag{4}$$

where the term $O(h^2)$ is such that $\frac{O(h^2)}{h} \to 0$ as $h \to 0$. For small enough h, the term $hf'(x^*)$ will dominate the remainder term $O(h^2)$. Choosing h sufficiently small and remembering that $f'(x^*)$ is less than $f(x^* \pm h)$, we deduce from these two equations that $f'(x^*) \geq 0$ as well as $-f(x^*) \geq 0$ which gives $f'(x^*) = 0$.

If the function f is C^2 continuous, then the sufficient conditions for x^* to be a strong local minimum are

$$f'(x^*) = 0 \quad (5)$$

$$f''(x^*) > 0 \quad (6)$$

Eq. (6) is easily pictured if we write it as $\frac{d}{dx}f'(x^*) > 0$, which means that the slope $f'(x^*)$ increase with positive x in the vicinity of the minimum. This is indeed the case since $f'(x)$ is negative just to the left of x^* and is positive just to the right of x^*. The condition given in Eq. (6) can be derived using the second-order expansion

$$f(x^*+h) = f(x^*) + hf'(x^*) + \frac{h^2}{2}f''(x^*) + O(h^3) \quad (7)$$

At the minimum, $f'(x^*) = 0$. Upon choosing h sufficiently small, we can ensure that the term $\frac{h^2}{2}f''(x^*)$ dominates the remainder term $O(h^3)$. Thus, at a strong local minimum, we have $\frac{h^2}{2}f''(x^*) > 0$ or $f''(x^*) > 0$.

Convexity can be used in defining optimality. A set S is called a convex set if for any two points in the set, every point on the line jointing the two points is in the set. Alternatively, the set S is convex if for every pair of pair of points x_1 and x_2 in S, and every α such that $0 \leq \alpha \leq 1$, the point $\alpha x_1 + (1-\alpha)x_2$ is in S. As an example, the set of all real numbers R^1 is a convex set. Any closed interval of R^1 is also a convex set.

A function $f(x)$ is called a convex function defined on the convex set S if for every pair of points x_1 and x_2 in S, and $0 \leq \alpha \leq 1$, the follow condition is satisfied

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha)f(x_2) \quad (8)$$

Geometrically this means that the graph of the function between the two points lies below the line segment jointing the two points on the graph as shown in Fig. 2.

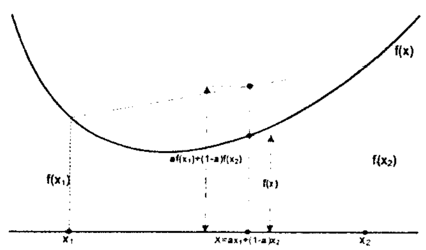

Fig. 2 Convex function

3. The solution using neural network

The neural network emulates the one variable systems associated with the convex function method: an imaginary point with no mass in a time-evolving gradient vector field $y(w, t)$, of the augmented barrier functions $F_{g(t)}$. The evolution of the point is governed by a set of differential equations involving the local gradient. Use the vector of slack variables $w(x) = b - Ax > 0$ to define the vector $w'(x, t)$. It is defined by $w'(x, t) = [g(t) \cdot w_i(x)]^{-1}$. The gradient of $f_{g(t)}(x)$ is computed as $y(x, t) = c - A^t w'(x, t)$.

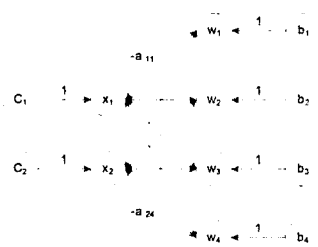

Fig. 3. Linear programming of neural network

Fig. 3 shows a small version of the neural network, in particular, one that 4-constraints and 2-variables inequality constrained one variable problem. There four types of neurons in this architecture; those that output x_j, w_i', c_j and b_i. Let us assume that x^0 denotes once again a known strictly interior feasible point $(Ax^0 < b)$. In the complete system of differential equations for the dynamical system, the x_j neurons are initialized to $x_j = x_j^0$ and governed by the equation

$$\frac{dx_j}{dt} = c_j - \sum_{i=1}^{m} A_{ij} w_i' \quad \text{for all } j. \quad (9)$$

The w_i' neurons are initialized to satisfy $w = b - Ax^0$. They are governed by the equation

$$r\frac{dw_i}{dt} = -w_i + b_i - \sum_{j=1}^{n} A_{ij} x_j \quad \text{for all } i \quad (10)$$

and the nonlinear transformation

$$w_i' = f(w_i, t) \quad (11)$$

More explicitly,

$$f(w_i, t) = \frac{1}{g(t) w_i} \quad (12)$$

where $g(t)$ is a monotonically increasing function of time and r is vary small compared to the scale of $\frac{dg}{dt}$. The c_j and b_i neurons are kept constant at all times. Differential equation (10) is derived form the definition $w(x) = b - Ax > 0$ of the slack variables. The time derivative of $w(x)$ is equated to the deviation $d(t) = -w(x) + b - ax$. When the dynamical system stabilizes, the definition is satisfied with $d(t) = 0$. Otherwise, (10) servos $w(x)$ to reduce $d(t)$. The connection strengths are unity for all c_j and b_i neurons, while the w_i' and x_j neurons are fully interconnected with bidirectional weights $-A_{ij}$. No real-time learning or weigh adaption occurs in this system. The parameters $-A_{ij}$, c_j and b_i, where $1 \le j \le n$ and $1 \le i \le m$, are set at $t=0$ and do not change through the convergence of the system.

4. The comparison solution problem

4.1 Problem

Determine the dimensions of an open box of maximum volume that can be constructed from an A4 sheet 210㎜*297㎜ by cutting four squares of side x from the corners and folding and gluing the edges as in Fig. 4.

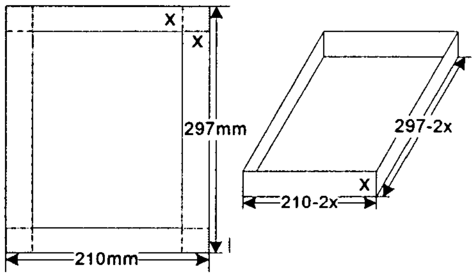

Fig. 4. Open box problem

4.2 Solution comparison

The problem is to maximize

$$V=(297-2x)(210-2x)x3 \qquad (13)$$

we set

$$f=-V=-62370x+1014x^2-4x \qquad (14)$$

Setting $f'(x)=0$, we get

$$f'(x)=-62370+2028x-12x^2=0 \qquad (15)$$

$$x=\frac{-2048\pm\sqrt{2048^2-4(12)(62370)}}{(2)(-12)} \qquad (16)$$

The two roots are $x_1=40.423$㎜ , $x_2=128.577$㎜ .

A physically reasonable cut is the first root $x^*=40.423$㎜ . The second derivative

$$f'(x^*)=2028-24x^*=1057.848. \qquad (17)$$

$f''(x^*)>0$ implies that x^* is a strict minimum of f, or maximum of V.

Table 1. Solution comparison

	calculation	neural network
Height x^*	40.423	40.363
Length	216.154	215.763
Width	129.154	128.958
Maximum volume	1128.5	1123.074

5. Conclusion

In this paper, we described a neural network architecture that can solve minimization problems with a massively parallel algorithm. The algorithm is based on the optimal function approach to minimization. In other to solve the basic of these network, we calculated quadratic equation minimization problems using the approach. Thus far, we have simulated the effects of limited numerical precision of analogue devices and of maximization problem that may be described.

[Reference]

[1] Pereira, J.M.D., Postolache, O., Girao, P.M.B.S., Cretu, M. "Minimizing temperature drift errors of conditioning circuits using artificial neural networks", Instrumentation and Measurement, IEEE Transactions on , Vol. 49, Issue: 5 , pp. 1122 · 1127, 2000.

[2] M. P. Kennedy and L. O. Chua, "Unifying the Tank and Hopfield linear programming circuit and the canonical nonlinear programming circuit of Chau and Lin", IEEE Transactions on Circuits and Systems Vol. CAS 34, no. 2, pp. 210 · 214, 1988..

[3] M. P. Kennedy and L. O. Chua, "Neural networks for nonlinear programming", IEEE Transactions on Circuits and Systems Vol. CAS·35, no. 5, pp. 554 562, 1988.

[4] A. Rodriguez Vazquez, R. Dominguez Castro, A. Rueda, J. L. Huertas, and E. Sanchez Sinencio, "Nonlinear switched capacitor neural networks for optimization problems", IEEE Transactions on Circuits and Systems Vol. 37, no. 3, pp. 384 397, 1990..

[5] C. Y. Maa and M. A. Shanblatt, "Linear and quadratic programming neural network analysis", IEEE Transactions on Neural Networks Vol. NN 3, pp. 580 · 594, 1992.

[6] J. Wang, "On the asymptotic properties of recurrent neural network for optimization", Int. J. Pattern Recognition Artificial Intelligence, vol. 5, no. 4, pp. 581 601, 1991.