Neural Network Methods for a Class of Nonsmooth Nonconvex Optimization Problems (Postprint)

Authors: Yu Xin, Chen Zhaorong

Date: 2018-05-24T00:00:00+00:00

Abstract

A neural network model is proposed for solving a class of nonsmooth nonconvex optimization problems with equality and inequality constraints. It is proven that when the objective function is lower bounded, the solution trajectory of the neural network converges to the feasible region in finite time. Meanwhile, the equilibrium point set of the neural network coincides with the critical point set of the optimization problem, and the neural network ultimately converges to the critical point set of the optimization problem. Unlike traditional penalty function-based neural network models, the proposed model does not require calculation of penalty parameters. Finally, the effectiveness of the proposed model is validated through simulation experiments.

Full Text

Preamble

Neural Network Optimization Method for a Class of Nonconvex Nonsmooth Optimization Problems

Yu Xin, Chen Zhaorong†
(School of Computer and Electronic Information, Guangxi University, Nanning 530004, China)

Abstract: This paper proposes a novel neural network model for solving nonsmooth nonconvex optimization problems with equality and inequality constraints. It is proved that when the objective function has a lower bound, the solution trajectory of the neural network converges to the feasible domain in finite time. Meanwhile, the equilibrium point set of the neural network coincides with the critical point set of the optimization problem, and the neural network ultimately converges to the critical point set. Unlike traditional penalty-function-based neural network models, the proposed model does not require calculation of penalty parameters. Finally, simulation experiments verify the effectiveness of the proposed model.

Keywords: neural network; nonconvex nonsmooth optimization; finite-time convergence

0 Introduction

In 1986, Tank and Hopfield first introduced analog neural network methods for solving constrained optimization problems, with further elaboration in subsequent work. The core idea involves using dynamical neural networks to simultaneously simulate the objective function and constraint functions, leveraging the analog and parallel processing capabilities of neural networks to compute optimal solutions. Building upon this foundation, Kennedy and Chua improved the approach by constructing a neural network with finite penalty parameters to solve nonlinear programming problems.

Lagrange multiplier methods have also been adapted into recurrent neural networks for solving nonlinear convex optimization problems. These Lagrange neural networks consist of variable neurons and Lagrange multiplier neurons. Variable neurons are responsible for finding minima of the objective function and providing equilibrium solutions, while Lagrange multiplier neurons rapidly guide the dynamic trajectory into the feasible domain. Detailed studies on Lagrange neural networks can be found in references [4-6].

In the early development of optimization using neural networks, models were designed primarily for smooth nonlinear programming problems. To address nonsmooth nonlinear programming, Forti et al. proposed the G-NPC neural network model—a differential inclusion-based gradient system derived from NPC (Nonsmooth Programming Circuit). This model requires only that the objective and constraint functions be regular, not necessarily smooth. To enhance generality for nonsmooth programming, references [8,9] proposed penalty-function and subgradient-based recurrent neural networks for solving nonsmooth convex and nonsmooth nonconvex programming problems.

Neural network convergence behavior can be characterized as either global or local. A neural network model is globally convergent if the solution trajectory from any initial point in the domain converges to an optimal solution (or approximate optimal solution). For instance, the neural network model in reference [10] is globally convergent under certain conditions. However, if convergence requires that the initial state be selected from within the feasible domain, the convergence is only local.

Penalty-function-based neural network models are largely non-globally convergent because finding an interior point of the feasible domain typically requires selecting a special domain containing the center of the feasible region. Therefore, to guarantee convergence, the initial solution set must be chosen within the feasible domain, and the penalty parameter must be sufficiently large. Moreover, the effectiveness of penalty-function neural network models depends on deterministic or non-deterministic penalty parameters, which are difficult to estimate in practical applications.

To address this challenge, Zheng et al. proposed a new recurrent neural network for nonsmooth convex optimization that does not require any penalty parameters. However, the structure of this neural network is exceptionally complex. To reduce model complexity, references [12-16] proposed several single-layer recurrent neural networks. Gou et al. developed a one-layer recurrent neural network for pseudoconvex optimization with linear equality constraints, proving that the solution trajectory converges to the feasible domain in finite time under equality constraints. Liu et al. proposed a new one-layer recurrent neural network for pseudoconvex optimization with linear equality constraints based on the penalty function method. Liu also earlier proposed a subgradient-based neural network for linear programming problems, proving global convergence within its domain.

Traditional neural network models for solving engineering optimization problems mostly require appropriate penalty functions, but these parameters are difficult to compute for certain objective functions, creating computational difficulties. This paper combines traditional recurrent neural network models to propose a new recurrent neural network model for solving optimization problems with both equality and inequality constraints. The advantages of this model are: (a) unlike traditional penalty-function-based neural network models, the proposed model does not require calculation of penalty factors; (b) many existing models restrict initial points to a bounded sphere, whereas this model allows arbitrary selection of the initial point; and (c) most existing models can only solve convex optimization problems, whereas this model can solve a class of nonconvex optimization problems.

---

1 Preliminary Knowledge

This paper considers the following nonsmooth nonconvex constrained optimization problem:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0, \\ & h(x) = 0, \end{aligned}$$

where $x \in \mathbb{R}^n$ is the variable vector, $f : \mathbb{R}^n \to \mathbb{R}$ is a possibly nonsmooth nonconvex objective function, $g : \mathbb{R}^n \to \mathbb{R}^m$ is a vector of inequality constraint functions, and $h : \mathbb{R}^n \to \mathbb{R}^p$ is a vector of equality constraint functions. We assume that g and h can be nonsmooth convex functions.

Definition 1 [19]: A function f is said to be locally Lipschitz near x if there exists a positive integer k such that for any x_1, x_2 in a neighborhood of x, $|f(x_1) - f(x_2)| \leq k\|x_1 - x_2\|$. If f is locally Lipschitz near every point in its domain, then f is called locally Lipschitz.

If f is locally Lipschitz on $\mathbb{R}^n$, the generalized directional derivative of f at x in direction v is defined as:

$$f^0(x; v) = \limsup_{y \to x, t \downarrow 0} \frac{f(y + tv) - f(y)}{t}.$$

The subdifferential of f at x is defined as:

$$\partial f(x) = \{\xi \in \mathbb{R}^n : f^0(x; v) \geq \langle \xi, v \rangle, \forall v\}.$$

Lemma 1 (Chain Rule [20]): If $x(t)$ is differentiable and locally Lipschitz, then:

$$\frac{d}{dt} f(x(t)) = \langle \xi, \dot{x}(t) \rangle, \quad \forall \xi \in \partial f(x(t)).$$

This paper proposes the following differential inclusion-based neural network dynamics model:

$$\dot{x}(t) \in -\partial f(x(t)) - \partial G(x(t)) - A^T \partial h(Ax(t) - b) + P(Ax(t) - b),$$

where $G(x) = \sum_{j=1}^{m} \max\{0, g_j(x)\}$, $P = I - A^T (AA^T)^{-1} A$ is a projection matrix, $A \in \mathbb{R}^{p \times n}$, and $b \in \mathbb{R}^p$.

---

2 Neural Network Convergence Analysis

Definition 2 [8]: A solution on interval $[0, T)$ satisfying initial condition $x(0) = x_0$ is an absolutely continuous function $x(t)$ such that:

$$\dot{x}(t) \in -\partial W(x(t)), \quad \text{for a.e. } t \in [0, T),$$

where $W(x) = f(x) + G(x) + \frac{1}{2}\|Ax - b\|^2$.

The feasible domain satisfying the constraint functions is defined as $S = \{x \in \mathbb{R}^n : g(x) \leq 0, h(x) = 0\}$. Let $S_1 = \{x \in \mathbb{R}^n : g(x) \leq 0\}$ and $S_2 = \{x \in \mathbb{R}^n : h(x) = 0\}$, so $S = S_1 \cap S_2$.

For the optimization problem studied in this paper, we assume there exists $\tilde{x} \in \mathbb{R}^n$ and $\delta > 0$ such that $\tilde{x} \in \operatorname{int}(S_1) \cap S_2$, where $\operatorname{int}(S_1)$ denotes the interior of S_1, and $B(\tilde{x}, \delta) \subseteq S_1$ denotes an open ball centered at $\tilde{x}$ with radius δ.

Lemma 2 [8]: For any $x \in \mathbb{R}^n$, $\partial G(x)$ is a nonempty, compact, convex, and upper semicontinuous set-valued mapping.

Lemma 3: The neural network model (2) has a local solution.

Proof: Since the right-hand side of (2) is a nonempty, compact, convex, and upper semicontinuous set-valued mapping, neural network (2) has at least one solution satisfying the initial condition.

Theorem 1: For any initial point x_0, the trajectory of neural network model (2) enters the equality constraint region S_2 in finite time and remains therein.

Proof: Define the energy function $H(x) = \frac{1}{2}\|Ax - b\|^2$. By the chain rule, there exists $\eta(t) \in \partial h(Ax(t) - b)$ such that:

$$\frac{d}{dt}H(x(t)) = \langle A^T\eta(t), \dot{x}(t)\rangle.$$

For any $x(t) \notin S_2$, we have $\|\eta(t)\| \geq \lambda_{\min}(AA^T) > 0$, where $\lambda_{\min}(AA^T)$ is the minimum eigenvalue of AA^T. Combining this with the neural network dynamics yields:

$$\frac{d}{dt}H(x(t)) \leq -\lambda_{\min}(AA^T)\|Ax(t) - b\|^2.$$

Integrating both sides from 0 to t gives:

$$H(x(t)) \leq H(x_0) - \lambda_{\min}(AA^T)\int_0^t \|Ax(s) - b\|^2 ds.$$

Clearly, the neural network trajectory enters S_2 in finite time. To prove it remains in S_2, we use contradiction. Assume the trajectory leaves S_2 at time t_1 and stays outside for $t \in (t_1, t_2)$. However, since $H(x(t))$ is decreasing and nonnegative, this leads to a contradiction, proving Theorem 1.

Assumption 1: The objective function $f(x)$ has a lower bound.

Theorem 2: If Assumption 1 holds, then for any initial point x_0, neural network model (2) has a global solution.

Proof: From Theorem 1, the neural network state vector $x(t)$ enters S_2 in finite time and remains therein. Thus, for $t \geq T$, the model simplifies to:

$$\dot{x}(t) \in -\partial f(x(t)) - \partial G(x(t)) + P(Ax(t) - b).$$

Define the energy function $W(x) = f(x) + G(x) + \frac{1}{2}\|Ax - b\|^2$. By the chain rule, for any $t \geq T$, there exist $\gamma(t) \in \partial f(x(t))$ and $\xi(t) \in \partial G(x(t))$ such that:

$$\frac{d}{dt}W(x(t)) = \langle \gamma(t) + \xi(t), \dot{x}(t) \rangle.$$

Since $f(x)$ is locally Lipschitz and has a lower bound, and $G(x)$ is bounded below, $W(x(t))$ is bounded. This implies the existence of a global solution.

Theorem 3: If Assumption 1 holds, for any initial point x_0, the trajectory of neural network model (2) enters the feasible domain S in finite time and remains therein.

Proof: From Theorem 1, $x(t)$ enters S_2 in finite time and stays there. We need only prove that for any initial point $x_0 \in \mathbb{R}^n \quad S_1$, $x(t)$ enters S_1 in finite time and remains there. Define $J^+(x) = \{j \in \{1, \dots, m\} : g_j(x) > 0\}$ and $G(x) = \sum_{j \in J^+(x)} g_j(x)$. For $x(t) \notin S_1$, we have $G(x(t)) > 0$.

From the neural network dynamics and chain rule, there exists $\mu > 0$ such that:

$$\frac{d}{dt}G(x(t)) \leq -\mu G(x(t)).$$

Integrating yields $G(x(t)) \leq G(x_0)e^{-\mu t}$, which shows $x(t)$ enters S_1 in finite time. The proof that it remains in S_1 follows similarly to Theorem 1 by contradiction.

Definition 3: The equilibrium point set of the neural network is $\mathcal{E} = \{x \in \mathbb{R}^n : 0 \in -\partial f(x) - \partial G(x) - A^T \partial h(Ax - b) + P(Ax - b)\}$.

Definition 4: The critical point set of the optimization problem is $\mathcal{C} = \{x \in S : 0 \in \partial f(x) + N_S(x)\}$, where $N_S(x)$ is the normal cone to the feasible domain S at x.

Theorem 4: If Assumption 1 holds, for any initial point x_0, every cluster point of the neural network trajectory is an equilibrium point.

Proof: From Theorem 3, $x(t) \in S$ for all $t \geq T$. Since $W(x(t))$ is non-increasing and bounded below, $\lim_{t \to \infty} \frac{d}{dt}W(x(t)) = 0$. This implies $0 \in \partial W(x(t))$, meaning $x(t)$ converges to the equilibrium set $\mathcal{E}$.

Theorem 5: If Assumption 1 holds, for any initial point x_0, any cluster point of the neural network trajectory is a critical point of optimization problem (1).

Proof: Let x^* be a cluster point. From Theorem 4, $x^* \in \mathcal{E} \cap S$. By the definition of equilibrium and the properties of subdifferentials, there exist $\gamma \in \partial f(x^*)$, $\xi \in \partial G(x^*)$, and $\eta \in \partial h(Ax^* - b)$ such that:

$$0 = -\gamma - \xi - A^T\eta + P(Ax^* - b).$$

Since $x^* \in S$, we have $h(Ax^* - b) = 0$ and $G(x^*) = 0$. Moreover, $\xi \in N_{S_1}(x^*)$ and $-A^T\eta + P(Ax^* - b) \in N_{S_2}(x^*)$. Therefore, $0 \in \partial f(x^*) + N_S(x^*)$, which means $x^* \in \mathcal{C}$.

---

3 Experimental Simulation

Neural network optimization models differ from conventional optimization algorithms primarily in their parallel computing capability. This parallelism enables efficient computation. To verify the effectiveness of the proposed differential inclusion neural network model (2), we use MATLAB 2012a to simulate optimization problem (1) and confirm that the solution trajectory $x(t)$ converges to an optimal solution in S. Since MATLAB 2012a's default data type is 'short' with four decimal places, all simulation results are presented with four decimal places.

Example 1 (Nonsmooth objective function, smooth constraint functions): Consider the problem:

$$\begin{aligned} \min \quad & f(x) = \|x\|_1 + \cos(x_3) \\ \text{s.t.} \quad & x_1^2 + x_2^2 - x_3 \le 0, \\ & x_1 + x_2 + x_3 = 1, \end{aligned}$$

where the objective function is nonsmooth and nonconvex, while the constraints are smooth convex functions. Five groups of random initial points are selected: $(1.5, 1.5, 1.5)^T$, $(3, 2, 3)^T$, $(0.5, 0.5, 2.5)^T$, $(2, 3, 1)^T$, and $(1, 1, 2)^T$. The trajectories of the state vector $x(t)$ are shown in [Figure 1: see original paper]. All solution trajectories converge to the unique equilibrium point $x^* = (1.7921, 0.2156, 3.0038)^T$.

Example 2 (Both objective and constraint functions nonsmooth): Consider:

$$\begin{aligned} \min \quad & f(x) = \|x\|_1 - \cos(x_1) \\ \text{s.t.} \quad & |x_1| + x_2^2 - x_3 \le 0, \\ & x_1 + x_2 + x_3 = 1, \end{aligned}$$

where the objective function is nonsmooth convex and one inequality constraint is nonsmooth convex. Five initial points are selected: $(1, 0, 3)^T$, $(2, 1, 2)^T$, $(3, 2.5, 0.5)^T$, $(4, 1, 1.5)^T$, and $(1.5, 1.5, 1)^T$. The solution trajectories of $x(t)$ are shown in [Figure 2: see original paper], all converging to the unique solution $x^* = (0.0043, 0.2640, 3.7377)^T$.

[Figure 1: see original paper] shows the convergence trajectories for Example 1, and [Figure 2: see original paper] shows the convergence trajectories for Example 2. Both examples demonstrate that the proposed neural network model effectively solves nonsmooth nonconvex optimization problems.

---

4 Conclusion

This paper proposes a novel recurrent neural network model for solving nonsmooth nonconvex optimization problems with equality and inequality constraints. Compared with traditional neural network models, the proposed recurrent neural network model requires no penalty functions, thereby resolving the practical difficulty of computing penalty parameters. Additionally, the network convergence speed is significantly improved. Two examples verify the correctness of the theoretical results for solving nonsmooth convex optimization problems. Future work will explore other neural network models (particularly chaotic neural networks) for solving nonsmooth nonconvex optimization problems.

References

[1] Tank D W, Hopfield J J. Simple neural optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit[J]. IEEE Trans on Circuits Systems, 1986, 33(5): 533–541.

[2] Kennedy M P, Chua L O. Neural networks for nonlinear programming[J]. IEEE Trans on Circuits Systems, 1988, 35(5): 554–562.

[3] Zhang Shengwei, Constantinides A G. Lagrange programming neural networks[J]. IEEE Trans on Circuits and Systems II: Analog and Digital Signal Processing, 1992, 39(7): 441-452.

[4] Feng Ruibin, Leung ChiSing, Constantinides A G, et al. Lagrange programming neural network for nondifferentiable optimization problems in sparse approximation[J]. IEEE Trans on Neural Networks and Learning Systems, 2017, 28(10): 2395-2407.

[5] Liang Junli, Leung Chising, So Hingcheung. Lagrange programming neural network approach for target localization in distributed MIMO radar[J]. IEEE Trans on Signal Processing, 2016, 64(6): 1574-1585.

[6] Liang Junli, So Hingcheung, Leung Chising, et al. Waveform design with unit modulus and spectral shape constraints via lagrange programming neural network[J]. IEEE Journal of Selected Topics In Signal Processing, 2015, 9(8): 1377-1386.

[7] Forti M, Nistri P, Quincampoix M. Generalized neural network for nonsmooth nonlinear programming problems[J]. IEEE Trans on Circuits and Systems I: Regular Papers, 2004, 51(9): 1741–1754.

[8] Bian Wei, Xue XiaoPing. Subgradient-based neural networks for nonsmooth nonconvex optimization problems[J]. IEEE Trans on Neural Networks, 2009, 20(6): 1024-1038.

[9] Bian Wei, Xue XiaoPing. Subgradient-based neural networks for nonsmooth convex optimization problems[J]. IEEE Trans on Circuits and Systems I: Regular Papers, 2008, 55(8): 2378-2391.

[10] Hu XiaoLin, Wang Jun. Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network[J]. IEEE Trans on Neural Networks, 2006, 17(6): 1487-1499.

[11] Cheng Long, Hou Zengguang, Lin Yingzi, et al. Recurrent neural network for non-smooth convex optimization problems with application to the identification of genetic regulatory networks[J]. IEEE Trans on Neural Networks, 2011, 22(5): 714-726.

[12] Qin Sitian, Feng Jiqiang, Song JiaHui, et al. A one-layer recurrent neural network for constrained complex-variable convex optimization[J]. IEEE Trans on Neural Networks and Learning Systems, 2018, 29(3): 534-544.

[13] Liu Xiaolan, Zhou Mi. A one-layer recurrent neural network for non-smooth convex optimization subject to linear inequality constraints[J]. Chaos, Solitons and Fractals, 2016(87): 39-46.

[14] Guo Zhishan, Liu Qingshan, Wang Jun. A one-layer recurrent neural network for pseudoconvex optimization subject to linear equality constraints[J]. IEEE Trans on Neural Networks, 2011, 22(12): 1982-1900.

[15] Liu Qingshan, Guo Zhishan, Wang Jun. A one-layer recurrent neural network for constrained pseudoconvex optimization and its application for dynamic portfolio optimization[J]. Neural Networks, 2012, 21(1): 99-109.

[16] Liu Qingshan, Cao Jinde, Chen Guangrong. A novel recurrent neural network with finite-time convergence for linear programming[J]. Neural Computation, 2010, 22(11): 2962-2978.

[17] Huang Yuancan, Yu Chuang. Improved lagrange nonlinear programming neural networks for inequality constraints[C]// Proc of International Joint Conference on Neural Networks. 2007: 962-966.

[18] Yu Xin, Xu Zhijian, Chen Zhaorong, et al. Lagrange neural network for nonsmooth nonconvex optimization problems with equality and inequality constrains[J]. Journal of Electronics & Information Technology, 2017, 39(8): 1950-1955.

[19] Aubin J P, Cellina A. Differential inclusion: set-valued maps and viability theory[M]. Berlin: Springer-Verlag, 1984.

[20] Clarke F H. Optimization and nonsmooth analysis[M]. New York, USA: Wiley, 1983.

[21] Yu Xin, Li Chenyu, Xu Zhijian, et al. Novel lagrange neural network for nonsmooth optimization problems[J]. Application Research of Computers, 2016, 33(11): 3261-3264, 3269.

[22] Yu Xin, Yu Yan, Xie Mian. Smoothing Lagrange neural network for constrained nonsmooth optimization problems[J]. Application Research of Computers, 2014, 31(5): 1349-1352.

[23] Yu Xin, Xie Mian, Li Chenyu. Solving non-Lipschitz optimization problems by smoothing neural networks[J]. Computer Engineering and Science, 2015, 37(12): 2262-2269.

Note: Figure translations are in progress. See original paper for figures.