

带Group Lasso正则项的Pi-Sigma神经网络在线梯度算法研究

刘 乐, 范钦伟

西安工程大学, 陕西 西安

收稿日期: 2022年2月21日; 录用日期: 2022年3月17日; 发布日期: 2022年3月24日

摘 要

Pi-Sigma神经网络是隐层带有求和神经元, 输出层带有求积神经元的一种前馈神经网络, 该网络具有较强的非线性映射能力。在误差函数中添加正则项是神经网络常用的优化方法, 和传统的 L_2 、 $L_{1/2}$ 正则项相比Group Lasso正则项可以在组级别上消除不必要的权值, 具有良好的稀疏效果。众所周知, 利用梯度法进行权值更新的学习方式有两种: 一种是批处理学习算法, 另一种是在线学习算法。本文提出带Group Lasso正则项的在线梯度学习算法来训练Pi-Sigma神经网络。最后, 数值实验结果表明改进后的算法收敛速度更快并且具有较好的泛化性能。

关键词

Pi-Sigma神经网络, 在线梯度算法, Group Lasso正则项

Online Gradient Algorithm for Pi-Sigma Neural Networks with Smooth Group Lasso Regularizer

Le Liu, Qinwei Fan

Xi'an Polytechnic University, Xi'an Shaanxi

Received: Feb. 21st, 2022; accepted: Mar. 17th, 2022; published: Mar. 24th, 2022

Abstract

Pi-Sigma neural network is a feedforward neural network with summation neurons in hidden layer and quadrature neurons in output layer, which has strong nonlinear mapping ability. Adding

regular terms to the error function is a common optimization method for neural networks. Compared with the traditional L_2 and $L_{1/2}$ regular terms, the Group Lasso regular terms can eliminate unnecessary weights at the group level and have a good sparse effect. As is known to all, there are two learning methods for weight update using gradient method: one is batch learning algorithm; the other is online learning algorithm. This paper proposes an online gradient learning algorithm with Group Lasso regularized terms to train Pi-Sigma neural networks. Finally, numerical results show that the improved algorithm converges faster and has better generalization performance.

Keywords

Pi-Sigma Neural Network, Online Gradient Algorithm, Group Lasso Regular Term

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

人工神经网络已经被广泛应用于各种领域[1]。其中前馈神经网络以其结构灵活性, 具有与不同训练算法的兼容性的特点成为最受欢迎的体系结构之一。Pi-Sigma 神经网络(简称 PSNN)由 Shin 和 Ghosh 在 1991 年首先提出, 它是前馈神经网络中的一种, 具有良好的收敛性和泛化性能, 该网络也具有较为规则的结构, 并且具有更快的学习速度, 能够通过增量添加单元来达到所需的复杂度水平[2], 在分类实验和函数逼近实验中得到了良好的收敛性和准确性。该网络利用乘积神经元作为输出单元, 在使用较少的权值和处理单元的同时, 间接结合了高阶网络的能力, 并且成功地应用到了许多领域[3] [4] [5]。但该网络隐层中的求和神经元个数不易太多, 否则会因为使用的权值太多而使该网络的学习速率降低, 因此网络中隐层神经元的个数的多少很重要。合适大小的网络可以提高其泛化性能, 因此寻找最合适的网络体系结构一直是一个重要的问题。正则化方法可以用来确定具有好的泛化性能的网络结构。

各种正则项已经被广泛应用, 并且不同的正则项具有不同的作用, L_0 正则项能够得到稀疏解, 但它面临着 NP 难题且很难求解; 对于 L_1 正则项来说, 它的求解较为容易并且具有稀疏性; L_2 正则项是很常用的一个正则项, 它可以抑制权值过大的增长, 但是没有稀疏性[6]; $L_{1/2}$ 正则项具有稀疏性并且可以使网络具有好的泛化能力[7], 但它只能删除接近于零的个体权值, 而不能删除连接权值都接近于零的隐藏神经元; Group Lasso 正则项可以在组水平上修剪不必要的权值和节点, 使网络结构更稀疏[8]。综上所述, L_2 正则项不具有稀疏性, $L_{1/2}$ 正则项虽然具有稀疏性, 但是稀疏效率不高, 只能去掉个别稀疏的权值但不能去掉整个的神经元节点, 所以针对这一关键问题, 在这里我们借用 Group Lasso 的思想在训练 PSNN 的过程中消除隐藏层的分组权重。Group Lasso 正则项具有稀疏性, 在网络结构的稀疏方面具有良好的效果, 它的优势是具有组水平上的稀疏性, 当保留必要的节点, 同时将冗余的节点作为一个组进行惩罚时, 修剪将更加直观和有效。

2. 相关研究

PSNN 由于其良好的特性已经被许多学者研究, 并且得到成功应用。熊焱[9]研究了 PSNN 的几种梯度学习算法, 同时给出了相应算法学习能力和收敛性理论上的分析, 并用仿真实验进行了验证; 在喻昕等人[10]的论文中, 为了克服权值过小导致的网络收敛速度慢的缺点, 给出了基于乘子法随机单点在线梯

度算法; 聂勇等人[11]将混合遗传算法运用到 PSNN 中, 并且证明了这种算法比遗传算法收敛更快; 亢喜岱[12]给出了 PSNN 的随机输入在线梯度算法, 对相应的收敛性进行了证明。在这篇文章中, 我们给出带 Group Lasso 正则项的 PSNN 的在线梯度学习算法, 来进一步优化网络结构。

3. Pi-Sigma 神经网络算法描述

令 PSNN 输入层的神经元个数为 n , 隐层(求和层)的神经元个数为 D , 输出层(求积层)的神经元个数为 1, 记连接输入层和求和层第 l 个神经元的权值向量为 $w_l = (w_{l1}, \dots, w_{ln})^T \in R^n (1 \leq l \leq D)$, 并记 $w = (w_1^T, \dots, w_D^T)^T \in R^{nD}$ 。对于连接求和层和输出层的权值都为 1。有一个输入为 $x_n = -1$, 对应的偏置为 w_{Dn} 。激活函数为 $g: R \rightarrow R$ 。结构图如图 1 所示。

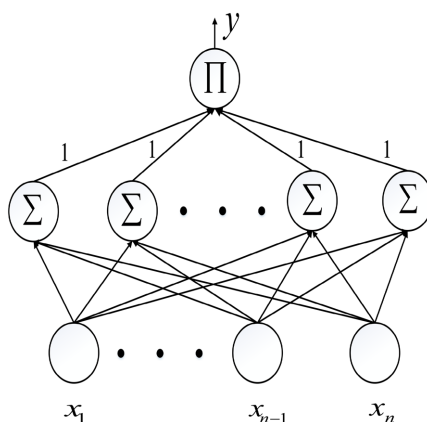


Figure 1. Structure diagram of Pi-Sigma neural network
图 1. Pi-Sigma 神经网络结构图

通过给定一个输入向量 $x \in R^n$, 网络的实际输出为

$$y = g\left(\prod_{l=1}^D \left(\sum_{i=1}^n (w_{li} x_i)\right)\right) = g\left(\prod_{l=1}^D (w_l \cdot x)\right) \quad (1)$$

其中 $w_l \cdot x$ 代表着 w_l 和 x 的内积。

3.1. 带 L_2 正则项的在线梯度算法

将训练样本 $\{x^m, O^m\}_{m=1}^M \subset R^n \times R$ 输入给网络, 其中 O^m 是输入向量 x^m 的理想输出。将 L_2 正则项加入误差函数中, 最终的误差函数形式为:

$$\begin{aligned} E(w) &= \frac{1}{2} \sum_{m=1}^M \left(O^m - g\left(\prod_{l=1}^D (w_l \cdot x^m)\right) \right)^2 + \lambda \sum_{l=1}^D \|w_l\|^2 \\ &= \sum_{m=1}^M g_m \left(\prod_{l=1}^D (w_l \cdot x^m) \right) + \lambda \sum_{l=1}^D \|w_l\|^2 \end{aligned} \quad (2)$$

其中, λ 是正则项的参数, $g_m(t) = \frac{1}{2} (O^m - g(t))^2$ 。

因此可以得到(2)式关于 w_k 的偏导:

$$E_{w_k}(w) = \sum_{m=1}^M g'_m \left(\prod_{l=1}^D (w_l \cdot x^m) \right) \prod_{l=1, l \neq k}^D (w_l \cdot x^m) x^m + 2\lambda w_k \quad (3)$$

其中 $l=1,2,\dots,D$ 。

给定一个随机初始权值 w_0 , 在线学习算法的权值更新公式为:

$$w^{iM+m} = w^{iM+m-1} - \eta_i \Delta_m^i w^{iM+m-1} \quad (4)$$

其中 $i=0,1,2,\dots; m=1,2,\dots,M$ 。

$$\Delta_m^i w_k^{iM+m-1} = g'_m \left(\prod_{l=1}^D (w_l^{iM+m-1} \cdot x^m) \right) \prod_{\substack{l=1 \\ l \neq k}}^D (w_l^{iM+m-1} \cdot x^m) x^m + \frac{2\lambda}{M} w_k^{iM+m-1} \quad (5)$$

3.2. 带 $L_{1/2}$ 正则项的在线梯度算法

将一组训练样本 $\{x^m, O^m\}_{m=1}^M \subset R^n \times R$ 输入到网络, 其中 O^m 是输入向量 x^m 的理想输出。将 $L_{1/2}$ 正则项加入误差函数中, 误差函数最终的形式为:

$$\begin{aligned} E(w) &= \frac{1}{2} \sum_{m=1}^M \left(O^m - g \left(\prod_{l=1}^D (w_l \cdot x^m) \right) \right)^2 + \lambda \sum_{l=1}^D \sum_{i=1}^n |w_{li}|^{\frac{1}{2}} \\ &= \sum_{m=1}^M g_m \left(\prod_{l=1}^D (w_l \cdot x^m) \right) + \lambda \sum_{l=1}^D \sum_{i=1}^n |w_{li}|^{\frac{1}{2}} \end{aligned} \quad (6)$$

其中, λ 是正则项的系数, $g_m(t) = \frac{1}{2} (O^m - g(t))^2$ 。

因此可以得到(6)式关于 w_{ki} 的偏导:

$$E_{w_{ki}}(w) = \sum_{m=1}^M g'_m \left(\prod_{l=1}^D (w_l \cdot x^m) \right) \prod_{\substack{l=1 \\ l \neq k}}^D (w_l \cdot x^m) x_i^m + \frac{\lambda \operatorname{sgn}(w_{ki})}{2|w_{ki}|^{\frac{1}{2}}} \quad (7)$$

其中 $l=1,2,\dots,D$ 。

给定一个随机初始权值 w_0 , 在线学习算法的更新公式为:

$$w^{iM+m} = w^{iM+m-1} - \eta_i \Delta_m^i w^{iM+m-1} \quad (8)$$

其中 $i=0,1,2,\dots; m=1,2,\dots,M$ 。

$$\Delta_m^i w_{ki}^{iM+m-1} = g'_m \left(\prod_{l=1}^D (w_l^{iM+m-1} \cdot x^m) \right) \prod_{\substack{l=1 \\ l \neq k}}^D (w_l^{iM+m-1} \cdot x^m) x_i^m + \frac{\lambda \operatorname{sgn}(w_{ki}^{iM+m-1})}{2M |w_{ki}^{iM+m-1}|^{\frac{1}{2}}} \quad (9)$$

3.3. 带 Group Lasso 正则项的在线梯度算法

将一组训练样本 $\{x^m, O^m\}_{m=1}^M \subset R^n \times R$ 输入到网络, 其中 O^m 是输入向量 x^m 的理想输出。将 Group Lasso 正则项加入误差函数中, 误差函数最终的形式为:

$$\begin{aligned} E(w) &= \frac{1}{2} \sum_{m=1}^M \left(O^m - g \left(\prod_{l=1}^D (w_l \cdot x^m) \right) \right)^2 + \lambda \sum_{l=1}^D \|w_l\| \\ &= \sum_{m=1}^M g_m \left(\prod_{l=1}^D (w_l \cdot x^m) \right) + \lambda \sum_{l=1}^D \|w_l\| \end{aligned} \quad (10)$$

其中, λ 是正则项的系数, $g_m(t) = \frac{1}{2} (O^m - g(t))^2$ 。

因此可以得到(10)式关于 w_k 的偏导:

$$E_{w_k}(w) = \sum_{m=1}^M g'_m \left(\prod_{l=1}^D (w_l \cdot x^m) \right) \prod_{\substack{l=1 \\ l \neq k}}^D (w_l \cdot x^m) x^m + \lambda \frac{w_k}{\|w_k\|} \quad (11)$$

其中 $l = 1, 2, \dots, D$ 。

给定一个随机初始权值 w_0 , 在线学习算法的更新公式为:

$$w^{iM+m} = w^{iM+m-1} - \eta_i \Delta_m^i w^{iM+m-1} \quad (12)$$

其中 $i = 0, 1, 2, \dots; m = 1, 2, \dots, M$ 。

$$\Delta_m^i w_k^{iM+m-1} = g'_m \left(\prod_{l=1}^D (w_l^{iM+m-1} \cdot x^m) \right) \prod_{\substack{l=1 \\ l \neq k}}^D (w_l^{iM+m-1} \cdot x^m) x^m + \frac{\lambda w_k^{iM+m-1}}{M \|w_k^{iM+m-1}\|} \quad (13)$$

4. 数值模拟

为了验证该算法的有效性, 这里使用一个函数逼近实验为例, 利用带 L_2 正则项, $L_{1/2}$ 正则项, Group Lasso 正则项的算法来逼近函数: $G(x) = x \cdot \text{csc}\left(\frac{\pi}{2}x\right)$ 。输入样本维数为 3, 输入样本个数为 101 个, 3 个隐层神经元, 1 个输出, 在这里惩罚项系数 $\lambda = 0.0001$, 学习率为 $\eta = 0.0004$, 最大迭代步数为 5000, 误差标准为 10^{-4} 。图 2, 图 3 和图 4 分别为带 L_2 正则项, 带 $L_{1/2}$ 正则项以及带 Group Lasso 正则项的算法逼近函数 $G(x)$ 的图像, 在这三幅图中, 实线为函数 $G(x)$ 函数图, 另一条为相应的逼近图。图 5 和图 6 分别给出了这三种算法的误差函数图和梯度范数图。

在这三种算法的逼近图中, 实线为要逼近的函数的曲线, 另一条为相应算法逼近该函数的曲线, 两条曲线的重合度越高, 则说明相应的算法的逼近效果越好。从图 2, 图 3 和图 4 中我们可以看到, 在相同的参数条件下, 带 Group Lasso 正则项的算法逼近 $G(x)$ 的曲线与 $G(x)$ 本身的曲线重合度最高, 因此说明提出的新算法比带 L_2 正则项和带 $L_{1/2}$ 正则项的算法的逼近效果都好, 同时也说明了新算法具有更好的泛化性能。三种算法的误差函数和梯度范数是单调递减并且收敛的, 收敛前越靠近 y 轴, 说明收敛的速度越快, 收敛后越接近 x 轴说明误差或梯度范数越小, 从图 5 和图 6 中可以看到, 带这三种正则项的算法的误差函数和梯度范数单调都是单调递减趋于 0 的, 同时可以看出带 Group Lasso 正则项的算法收敛速度最快, 误差最小。

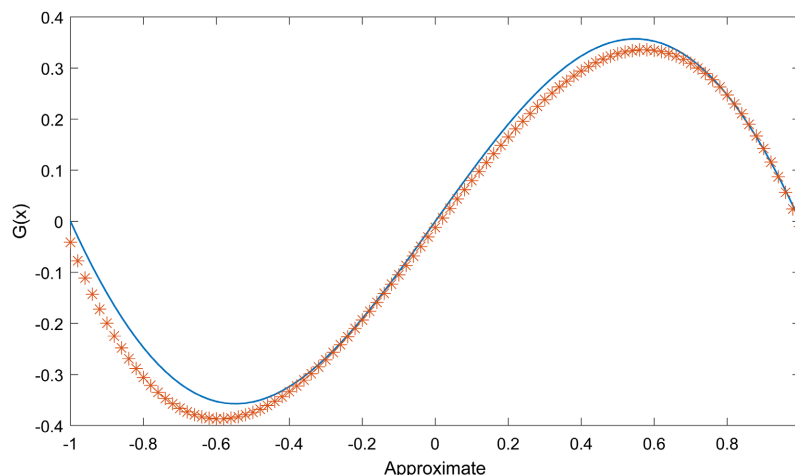


Figure 2. Algorithm approximation diagram with L_2 regular term
图 2. 带 L_2 正则项的算法逼近图

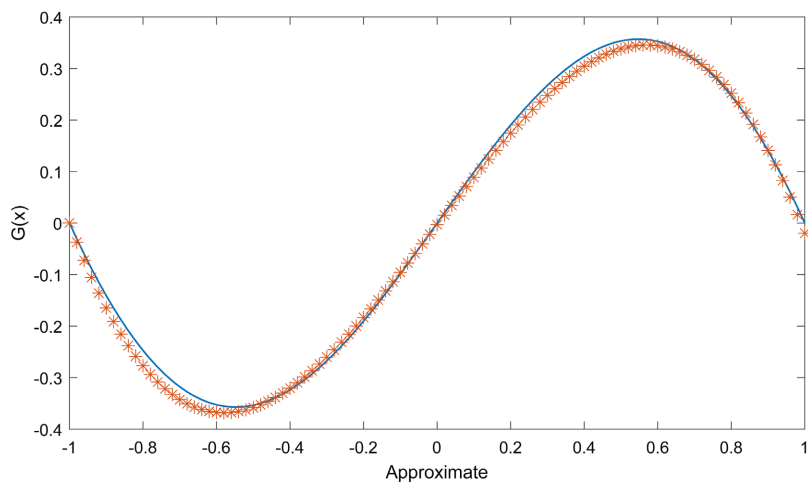


Figure 3. Algorithm approximation diagram with $L_{1/2}$ regular term

图 3. 带 $L_{1/2}$ 正则项的算法逼近图

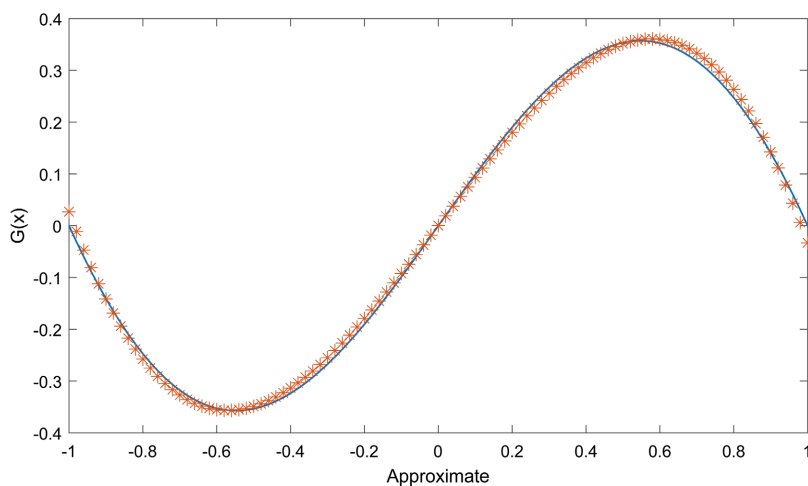


Figure 4. Algorithm approximation diagram with Group Lasso regular term

图 4. 带 Group Lasso 正则项的算法逼近图

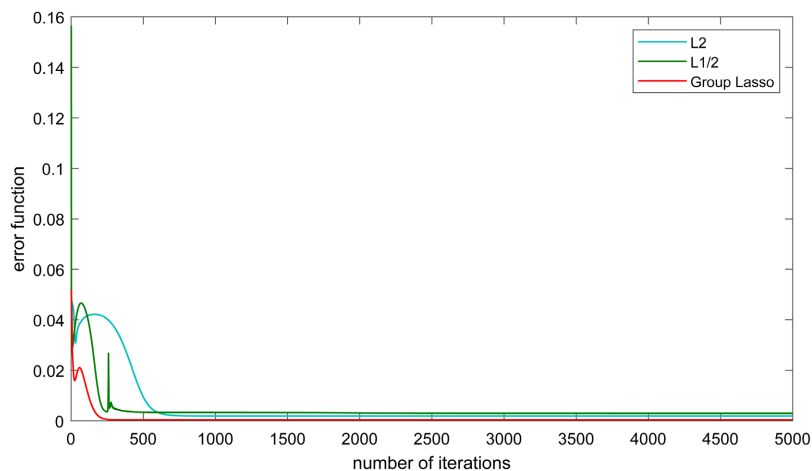


Figure 5. Error function diagram of three algorithms

图 5. 三种算法的误差函数图

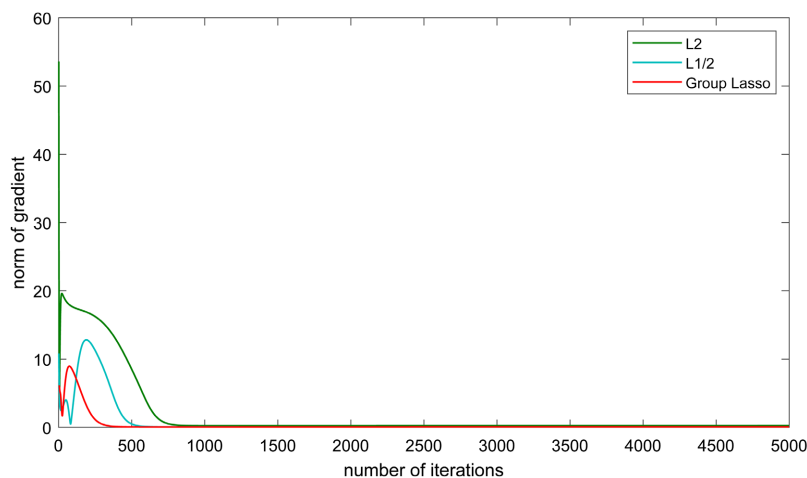


Figure 6. Norm of gradient graph of three algorithms

图 6. 三种算法的梯度范数图

5. 结束语

本文中, 我们考虑了 Group Lasso 惩罚项的良好稀疏性能, 提出了带 Group Lasso 正则项的 Pi-Sigma 神经网络的在线梯度学习算法。通过函数逼近实验, 根据三种算法的逼近效果图, 误差函数图以及梯度范数图验证了该算法的有效性。

参考文献

- [1] Hsieh, W.W. (2009) Machine Learning Methods in the Environmental Sciences. Cambridge University Press, Cambridge. <https://doi.org/10.1017/CBO9780511627217>
- [2] Shin, Y. and Ghosh, J. (1991) The Pi-Sigma Network: An Efficient Higher-Order Neural Network for Pattern Classification and Function Approximation. *International Joint Conference on Neural Networks*, **1**, 13-18.
- [3] 蒋立军, 许枫, 朴胜春, 杜文萍. Pi-Sigma 神经网络在实时底质分类中的应用[J]. *应用声学*, 2005, 24(6): 346-350.
- [4] 杜拴平, 宋明凯, 宫先仪. Pi-Sigma 网络在水声目标分类中的应用[J]. *声学学报*, 1997(4): 345-351.
- [5] 聂永, 邓伟. 一种基于 Pi-Sigma 神经网络图像压缩的方法[J]. *苏州大学学报(工科版)*, 2007, 27(4): 8-10.
- [6] Mohamed, K.S., et al. (2016) Batch Gradient Method for Training of Pi-Sigma Neural Network with Penalty. *International Journal of Artificial Intelligence and Applications*, **7**, 11-20. <https://doi.org/10.5121/ijaiia.2016.7102>
- [7] Mohamed, K.S., Wu, W., et al. (2017) A Modified Higher-Order Feed forward Neural Network with Smoothing Regularization. *Neural Network World*, **27**, 577-592. <https://doi.org/10.14311/NNW.2017.27.032>
- [8] Tibshirani, R. (2011) Regression Shrinkage and Selection via the Lasso: Aretrospective. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, **73**, 273-282. <https://doi.org/10.1111/j.1467-9868.2011.00771.x>
- [9] 熊焱. Pi-Sigma 神经网络的几种梯度学习算法[D]: [博士学位论文]. 大连: 大连理工大学, 2007.
- [10] 喻昕, 邓飞, 唐利霞. Pi-Sigma 神经网络的乘子法随机单点在线梯度算法[J]. *计算机应用研究*, 2011, 28(11): 4074-4077.
- [11] 聂永, 邓伟. Pi-Sigma 神经网络混合学习算法及收敛性分析[J]. *计算机工程与应用*, 2008, 44(35): 56-58.
- [12] 亢喜岱. Pi-Sigma 神经网络随机输入在线梯度法的收敛性[D]: [硕士学位论文]. 大连: 大连理工大学, 2007.