武汉大学 计算机学院

2020级硕士研究生课程



**高级算法设计与分析**

学 院： 计算机学院

学 号： 2020282110206

姓 名： 孙懿

基于模拟退火算法的bp神经网络模型优化

摘 要

神经网络算法是研究目前深度学习的基础，基于bp算法的bp神经网络是运用最广泛和最成功的神经网络模型之一。bp神经网络在函数逼近、模式识别、分类问题、数据压缩等实际运用方面体现出较好的适应性。但是基于梯度下降法的bp神经网络具有收敛速度慢，易陷入局部最小值等缺点。模拟退火算法源于固体退火原理，是一种基于概率的算法。算法的思想是结合[概率](https://baike.baidu.com/item/%E6%A6%82%E7%8E%87)突跳特性在[解空间](https://baike.baidu.com/item/%E8%A7%A3%E7%A9%BA%E9%97%B4)中随机寻找[目标函数](https://baike.baidu.com/item/%E7%9B%AE%E6%A0%87%E5%87%BD%E6%95%B0)的全局最优解，即到达局部最优解时能够概率性地跳出并最终趋于全局最优。模拟退火广泛运用于求解NP问题，而求解神经网络参数的过程是NP完全问题。所以将模拟退火算法运用于bp神经网络的训练优化中可以改善bp神经网络的缺陷。

关键词：bp神经网络；退火算法；全局最优解

# 1 引言

近几年，BP神经网路是在人工智能领域应用最为广泛的关键技术之一。它是一种多层前馈人工神经网络模型，能够学习大量的模式映射关系，并从仿生学的角度模拟人脑的智能行为，广泛应用于模式识别、分类、预测等领域，具有很强的自适应能力。BP神经网络虽然在学术和应用上已经十分成熟．但是BP神经网络也有其缺陷的地方：学习速度慢、网络的超参数难以确定、容易陷入局部最小值、网络可扩展能力有限。对于上述问题研究最多的是如何解决bp神经网络在梯度下降的过程中陷入局部最小值的情况。一种可行的方法是：将模拟退火算法运用到bp神经网络的权值优化过程中。本文将从以下两方面介绍：

1. 在bp神经网络上引入模拟退火算法进行训练优化；
2. 提高模拟退火算法的搜索效率的一些方法。

# 2 相关知识

## 2.1 模拟退火算法

模拟退火算法(Simulated Annealing，SA)最早的思想是由N. Metropolis等人于1953年提出。1983年，S. Kirkpatrick等成功地将退火思想引入到组合优化领域。它是基于Monte-Carlo迭代求解策略的一种随机寻优算法，其出发点是基于物理中固体物质的退火过程与一般组合优化问题之间的相似性。模拟退火算法从某一较高初温出，伴随温度参数的不断下降,结合概率突跳特性在解空间中随机寻找目标函数的全局最优解，即在局部最优解能概率性地跳出并最终趋于全局最优。模拟退火算法是一种通用的优化算法，理论上算法具有概率的全局优化性能，目前已在工程中得到了广泛应用，诸如VLSI、生产调度、控制工程、机器学习、神经网络、信号处理等领域。

模拟退火算法可以分解为解空间、目标函数和初始解三部分。

算法具体步骤如下：

步骤1：初始化：初始温度T(充分大)，初始解状态S(S是算法迭代的起点)，每个T值的迭代次数L。

步骤2：对做第3到第6步。

步骤3：产生新解。

步骤4：计算增量，其中C为评价函数。

步骤5：则接收新解，发生状态转移；否则以概率接受新解。

步骤6：如果满足终止条件则将当前解作为最优解，结束算法，否则进入第7步。

步骤7：T按照一定的策略减小，直至趋近于0。重新进入第2步进行迭代。

其中步骤4和步骤5通过计算新旧状态之差得到可接受概率，并与Random(0,1)进行比较，判断是否接收状态转移。该准则称为Metropolis准则，其中k称作Boltzmann常数。

## 2.2 bp神经网络

多层前馈神经网络结构如下图2.1：

定义：每层神经元与下一层神经元全互联，神经元之间不存在同层连接也不存在跨层连接。

前馈：输入层接受外界输入，隐含层与输出层神经元对信号进行加工，最终结果由输出层神经元输出。

学习：根据训练数据来调整神经元之间的“连接权”以及每个功能神经元的“阈值”。

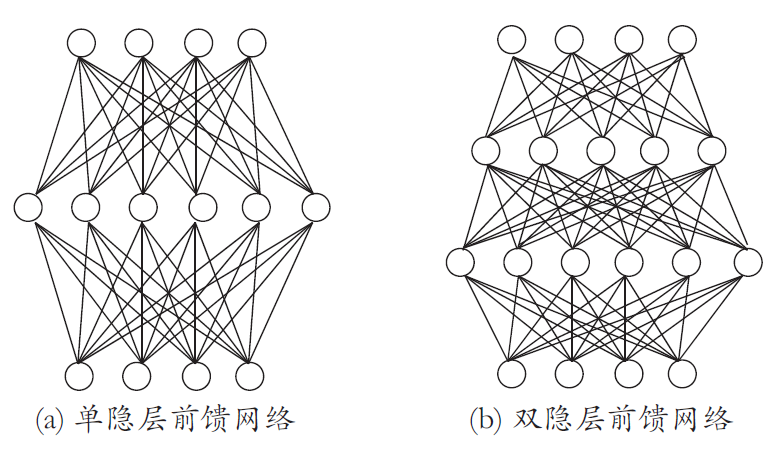


图2.1 bp神经网络的结构

# 3 改进的SABP算法

## 3.1 随机扰动法

为得到神经网络的最优权值，以bp神经网络的实际输出与预测输出的平方型误差函数作为优化函数：

（1）

其中是前向传播结果，是训练样本结果。

定义本问题的解空间和模拟退火状态转移方程：为本问题的解，和表示温度为的情况下，模拟退火状态更新的前后状态。给定以下转移方程：

（2）

其中为扰动参数，为随机扰动变量，可以服从高斯、柯西、均匀分布的其中一种。接收的Metropolis准则为：

（3）

基于SA的bp优化实现具体思想是：根据式（2）产生新解，然后计算能量函数f的值，再根据式（3）判断是否接受新解。不断重复直至满足此温度下迭代次数，降温，再次执行相同操作。

算法如下：

步骤1：初始化参数。设置一个较大的初始温度根据bp神经网络结构随机选择初始权值构成初始状态解；

步骤2：产生新状态解。由式（2）产生新状态解；

步骤3：输入样本，利用bp前向传播算法计算能量函数f也就是旧状态下的损失值和新状态的损失值，即和；

步骤4：根据（3）判断是否接收新解；

步骤5：如果内循环次数到达指定次数，执行步骤6；否则执行步骤2；

步骤6：降温，令。如果温度到达一定阈值，或者当前样本训练迭代次数达到要求，算法结束。否则执行步骤2。

基于随机扰动的算法利用了模拟退火算法搜索全局最优解，有效的解决了陷入局部极小值的问题。但是由于搜索策略是随机的，导致全局收敛较慢。于是下面提出了基于梯度下降和模拟退火结合的bp神经网络改进。

## 3.2 SABP算法

在bp网络中引入SA算法, 既利用了bp算法使得神经网络连接重权的调整具有大致方向的能力, 又利用了SA算法能以一定的概率接受坏的调整, 从而跳出局部极小趋向全局极小的特性, 提高了学习的速度和逼近系统的精度。

改进上述算法，不是立即执行状态转移方程，而是先利用随机梯度下降法快速向局部或全局最优点收敛。执行SGD后的权值矩阵再加入到状态转移方程中进行模拟退火操作。SABP算法如下：

步骤1：初始化参数。设置一个较大的初始温度根据bp神经网络结构随机选择初始权值构成初始状态解；

步骤2：输入样本，利用SGD进行参数更新得到，同时可以得到相应旧状态损失函数值；

步骤3：根据（2）产生新状态解。旧状态为SGD训练后的参数矩阵；

步骤4：利用bp前向传播算法计算能量函数f也就是新状态下的损失值。旧状态损失值已经在SGD训练时得到；

步骤5：根据（3）判断是否接收新解；

步骤6：如果内循环次数到达指定次数，执行步骤6；否则执行步骤3；

步骤7：降温，令。如果温度到达一定阈值，或者当前样本训练迭代次数达到要求，算法结束；否则执行步骤2，。

# 4 结论

从多分类任务和值预测结果来看，SABP可以显著的改善bp神经网络的训练效果，避免陷入局部最小值的情况。SABP将SGD和模拟退火结合起来，降低了训练的复杂度，使得网络可以更快的收敛到最优情况。

# 5 展望

由于模拟退火算法对全局解空间把握不强，容易耗费大量时间去寻找全局最优解或者逼近全局最优解。为了解决这一问题，可以将遗传算法的优点和模拟退火法的优点结合起来。遗传算法往往可以很好把握全局解空间，但是容易陷入局部最优解。利用遗传算法代替SGD进行参数更新，并且使用模拟退火防止解陷入局部最小值，可以进一步改善bp神经网络的性能。

参考文献

[1]宋志宇,李俊杰.基于模拟退火神经网络模型的岩质边坡稳定性评价方法[J].长江科学院院报,2006,23(2):42-45. DOI:10.3969/j.issn.1001-5485.2006.02.012.

[2]尤丽华,吴静静,王瑶, 等.基于模拟退火优化BP神经网络的pH值预测?[J].传感技术学报,2014,(12):1643-1648. DOI:10.3969/j.issn.1004-1699.2014.12.011.

[3]宫珊珊,刘金澳,梅立峰, 等.基于BP模拟退火算法的海上多跳高频无线电传输损耗优化模型[J].新型工业化,2019,9(4):17-21. DOI:10.19335/j.cnki.2095-6649.2019.04.004.

[4]周爱武,翟增辉,刘慧婷.基于模拟退火算法改进的 BP神经网络算法[J].微电子学与计算机,2016,33(4):144-147.

[5]吕琼帅,王世卿.基于遗传模拟退火算法优化的BP神经网络[J].计算机与现代化,2011,(6):91-94. DOI:10.3969/j.issn.1006-2475.2011.06.026.

[6]蒋美云.基于模拟退火算法优化的BP神经网络预测模型[J].软件工程,2018,21(7):36-38. DOI:10.19644/j.cnki.issn2096-1472.2018.07.010.

[7]吴金文,王玉鹏,周海波.基于模拟退火耦合粒子群算法优化BP神经网络的机床主轴热误差补偿研究[J].机床与液压,2019,47(11):92-95. DOI:10.3969/j.issn.1001-3881.2019.11.019.

[8]张坤,甘小艇.模拟退火PSO的神经网络的网络流量预测模型[J].计算机工程与设计,2012,33(5):2013-2016. DOI:10.3969/j.issn.1000-7024.2012.05.068.