

**本科生毕业设计[论文]**

**基于多目标优化的人工神经网络结构设计**

院 系 自动化学院

专业班级 自动化1401班

姓 名 王 壮

学 号 U201414260

指导教师 潘林强

年 月 日

**学位论文原创性声明**

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名： 年 月 日

**学位论文版权使用授权书**

本学位论文作者完 全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于 1、保密囗，在 年解密后适用本授权书

2、不保密囗 。

（请在以上相应方框内打“√”）

作者签名： 年 月 日

导师签名： 年 月 日

摘 要

神经网络作为模拟生物大脑结构和功能的一种模型，在计算机视觉、语音识别、医学医疗等方面得到了广泛的应用。在神经网络中，它的性能不仅仅与权值有关，神经网络的结构也起着至关重要的作用。神经网络的结构过于简单会使得它不足以存储样本中的规律知识，而复杂度过高会则会使得它的泛化能力变差。所以，如何设计一种合适的神经网络结构成为当前的一个研究热点。

在神经网络有足够计算能力的前提下，尽可能的简化神经网络的结构，从这个角度出发，把神经网络的结构设计看成是一个多目标优化问题。多目标优化问题中，多个目标函数之间可能存在冲突，无法做到同时优化所有的目标函数，往往得到是一帕累托最优解解集，解集中的每一个解都是对多个目标进行折衷之后得到的最优解。

本文提出了一种基于多目标优化的神经网络结构设计方法。它以神经网络的隐藏层的层数、神经元个数以及神经网络的误差能量函数作为目标，在NSGA-II算法的基础上进行结构设计。设计了一种长度可变的编码方式，实现了对神经网络层数以及每层神经元个数的同时优化。最终得到一个均匀分布、具有一致性的帕累托前沿，研究人员可以根据这个前沿进行规则提取、模型选择等操作。

**关键词：**神经网络；多目标优化；进化算法；泛化；Pareto

Abstract

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××.

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××.

（Time New Roman小4号，行间距固定1.5倍行距，字符间距为标准）

**Key Words：**××××; ××××; ××××; ××××

（Time New Roman 4号加粗） （Time New Roman小4号）

**目录**

[摘 要 I](#_Toc514699731)

[Abstract II](#_Toc514699732)

[1. 绪论 1](#_Toc514699733)

[1.1. 引言 1](#_Toc514699734)

[1.2. 国内外相关研究情况 2](#_Toc514699735)

[1.3. 研究内容与实现功能 3](#_Toc514699736)

[1.4. 内容安排 3](#_Toc514699737)

[2. 人工神经网络 4](#_Toc514699738)

[2.1. 神经元 4](#_Toc514699739)

[2.2. 激活函数 5](#_Toc514699740)

[2.2.1. Sigmoid函数 5](#_Toc514699741)

[2.2.2. Tanh函数 5](#_Toc514699742)

[2.2.3. Relu函数 5](#_Toc514699743)

[2.3. 前馈神经网络 6](#_Toc514699744)

[2.4. 反馈神经网络 7](#_Toc514699745)

[2.5. Rprop学习算法 7](#_Toc514699746)

[2.5.1. 前向传播 8](#_Toc514699747)

[2.5.2. 反向传播 8](#_Toc514699748)

[3. 单目标学习与多目标学习 11](#_Toc514699749)

[3.1. 单目标学习 11](#_Toc514699750)

[3.2. 多目标学习 11](#_Toc514699751)

[3.2.1. 标量化多目标学习 12](#_Toc514699752)

[3.2.2. 基于Pareto的多目标学习 13](#_Toc514699753)

[4. 多目标进化算法 15](#_Toc514699754)

[4.1. NSGA-II进化算法 15](#_Toc514699755)

[4.1.1. 快速非支配排序算法 15](#_Toc514699756)

[4.1.2. 保留多样性 16](#_Toc514699757)

[4.1.3. 主循环 17](#_Toc514699758)

[5. 实验设计 19](#_Toc514699759)

[5.1. 神经网络模型设置 19](#_Toc514699760)

[5.2. 神经网络编码 20](#_Toc514699761)

[5.3. 交叉 21](#_Toc514699762)

[5.4. 变异 22](#_Toc514699763)

[5.5. 评估 23](#_Toc514699764)

[5.6. 选择 24](#_Toc514699765)

[5.7. 参数设置 24](#_Toc514699766)

[5.8. 实验所用的数据集 24](#_Toc514699767)

[6. 实验结果分析 25](#_Toc514699768)

[6.1. 识别可解释模型 25](#_Toc514699769)

[6.2. 模型选择 28](#_Toc514699770)

[6.3. Dropout的影响 29](#_Toc514699771)

[7. 总结与展望 31](#_Toc514699772)

[8. 致谢 32](#_Toc514699773)

[9. 参考文献 33](#_Toc514699774)

# 绪论

## 引言

人工神经网络是基于生物学中神经网络的结构和功能，在理解并抽象了人脑的结构和外界刺激响应机制之后，以网络拓扑知识为理论基础，模拟生物大脑的神经系统对复杂信息的处理机制的一种计算模型或数学模型。该模型以并行分布的处理能力、高容错性、局部性计算、智能化和自学习等能力为特征，将信息的加工和存储结合在一起，以其独特的知识表示方法和智能化的自适应学习能力，引起各科学领域的关注。它实际上是一个由大量简单元件相互连接而成的复杂网络，具有高度线性化，能够进行复杂的逻辑操作和非线性关系实现的系统。

神经网络的学习算法通常由两部分组成，首先是选择合适的模型，然后使用机器学习算法和已有数据集估计模型的参数包括连接权和偏置值使误差能量最小。大部分情况下，模型的选择只凭直觉和经验一次性完成，然后在此模型上对参数进行优化。在优化时，传统优化方法通常会过分强调模型对当前训练样本的正确分类的的能力，而忽视了网络的泛化能力和抗干扰能力。模型的选择是基础，是神经网络中极其重要的一个步骤，如果选择的模型太简单就会导致训练出来的神经网络无法概括所有知识、拟合能力差，反之如果模型过于复杂网络的拟合能力虽然可以随着网络的复杂性增加而增加，但过于复杂的网络结构也造成了过拟合以及模型理解困难的问题。

神经网络的性能不仅仅与权值优化有关，神经网络的结构大小同样起着至关重要的作用。所以在选择合适的模型时要选择尽可能简单的网络，同时还要保证神经网络具有强大的知识概括能力。然而在训练神经网络时，神经网络的结构越复杂，在训练数据集上的知识概括能力就越强大，这是一对相互矛盾的问题。

可以看出神经网络中存在着一对相互矛盾的目标，我们需要同时对这些目标进行求解，这也就是多目标优化问题。一种解决方案解决是将多个目标汇总为一个标量代价函数，然后使用单目标问题求解方式去求解。另一种方案是基于Pareto的多目标学习算法，相比于其它使用标量代价函数的算法具有更强的解决机器学习中各种类型问题的能力。虽然通过多目标优化方法求解到的解对单一目标函数并不一定是最佳的，但是从整体来看却是一个最佳的解。并且由于多个目标函数之间通常是相互制约和相互影响的，所以通过多目标优化得到的神经网络结构可以解决一般神经网络中出现的训练时间长、局部极小、泛化能力差、可理解性差的问题。

多目标优化问题是工程实践和科学研究等实际问题中的常见问题，也是国内外研究的热点。目前，基于多目标优化的神经网络结构设计问题是国内外的一个研究热点，并且取得了巨大的成功。

## 国内外相关研究情况

在保证神经网络性能的同时，尽可能的使神经网络的结构最简单。这个问题一直以来都备受关注，神经网络方面的专家在这个方向坚持不懈的努力，想要找到一种合适的解决方案。一种最简单直接的方法就是利用专家知识，凭借现有经验，然后通过反复的实验尝试，最终找到一种合适的结构。但是这种方法不仅需要实验人员掌握大量相关的专家知识，而且不同人员的判定方式不同，得到的结果也不尽相同。另一种就是基于多目标优化的神经网络结构设计，这种方法不需要人为干预，也不需要大量的专家知识，算法可以一次得到多个合适的解，提供给实验人员作为选择。

近年来，越来越多的研究人员开始采用多目标进化算法进行神经网络结构设计，经过不断地尝试努力，在理论和实践上都有了突破性的进展。多目标进化算法和神经网络结合在一起时，既可以同时对神经网络的结构和连接权值参数进行优化，也可以只对神经网络的结构进行优化。

意大利经济学家维弗雷多·帕累托(Vilfredo Federico Damaso Pareto)最早提出帕累托最优的概念，之后多目标优化引来了越来越多研究人员的研究关注。1967年，Rosenberg建议采用基于进化的搜索来处理多目标优化问题；1984年，David Schaffer首次在机器学习中实现了向量评估遗传算法(Vector Evaluated Genetic Algorithm, VEGA)。他使用VEGA来在一组覆盖问题中查找和维护多个分类规则。VEGA试图通过使用每个属性(例如成本，可靠性)中的一个为依据来选择一部分下一代来实现这一目标。尽管Schaffer取得了一定的成功，但VEGA只能找到帕累托最前面的极端点，其中一个属性是最大的，因为它从不用根据属性间的平衡而改变选择。1989年，David Goldberg在其著作“Genetic Algorithms for Search, Optimization and Machine Learning”中提出了使用进化算法实现多目标的优化技术。在1993年多目标遗传算法(Multi-objective Genetic Algorithm MOGA)被提出。同年，一种基于Pareto支配定义的锦标赛选择机制算法(Niched Pareto Genetic Algorithm NPGA)被提出。次年，由Kalyanmoy Deb提出了非支配排序遗传算法(Non-dominated Sorting Genetic Algorithm, NSGA)。在1999年强度帕累托进化算法(Strength Pareto Evolutionary Algorithm, SPEA)由E. Zitzler; L. Thiele提出。2001年E. Zitzler等人又提出了SPEA-II(Strength Pareto Evolutionary Algorithm 2)，对SPEA进行了改进。2002年Kalyanmoy Deb在先前提出的NSGA的基础上进行改进提出了NSGA-II称为当前最流行的多目宝进化算法。常用的一些其他进化算法还有MOEA/D、M-MEDA、PESA、PESA-II、PAES等。

## 研究内容与实现功能

在当前领域已有的研究基础上，本文提出了一种基于多目标进化算法NSGA-II的神经网络结构设计方法。算法最终可以实现以下几点功能：

1. 使用一种可变长度的编码方式，对神经网络隐藏层进行编码，可以同时对神经网络的层数和每层的神经元个数进行控制优化。
2. 由于编码长度可变，所以本文创建了一种新的交叉方式，实现了不同长度基因序列的交叉操作，用来产生新的子代。
3. 最终得到一个帕累托前沿，用户可以根据需求选择合适的神经网络结构。

## 内容安排

第一章概述了研究背景和当前已有研究的现状，并大致交代了本文所做的工作内容和最终所达到的效果。

第二章介绍人工神经网络的基础组成部分、结构类型、以及经典Rprop学习算法等相关知识。

第三章对比介绍了单目标学习、标量化多目标学习和基于帕累托的多目标学习，以及一些帕累托的相关知识和概念。

第四章对目前存在多目标进化算法进行对比，并介绍了当前最流行的多目标进化算法NSGA-II的实现过程。

# 人工神经网络

人工神经网络(Artificial Neural Network, ANN)，简称神经网络(Neural Network, NN)是受人类大脑中生物神经网络信息处理方式的启发从而得到的计算模型。由于在语音识别、计算机视觉和文本处理等方面取得了许多突破性成果，人工神经网络在机器学习研究和工业中引起了越来越多的关注。

神经网络有大量的神经元联结组成进行计算，大多数情况下人工神经网络可以根据外界信息进行学习并并改变其内部结构，属于自适应系统。典型的人工神经网络由结构、激活函数、学习规则这三个部分组成。

## 神经元

神经网络中的基本计算单位是神经元，通常称为节点或单元。它接受来自其它神经元或者外部来源的输入，经过计算输出相应的结果。每个输入都有一个与之相关联的权重()，权重是根据当前输入相对于其它输入的重要性进行分配的，输入越重要，权重也就越大。神经元还将函数应用于其输入的加权和，函数称为激活函数。但个神经元模型如下图所示：

|  |
| --- |
|  |
| 图1：单个神经元 |

上述网络中X1、X2、XN为数字输入，w1、w2、wN为输入的连接权重。另外还有另外一个权重为b（偏置）的输入1。神经元的输出Y的计算方式已经在图1中展示，函数f是非线性的将神经元的输入信号按照一定规律转换为输出信号，称为神经元功能函数f（Activation Function），也称激活函数。

## 激活函数

激活函数目的在于将非线性引入到神经元的输出中，这点对于神经网络来说是很重要的，因为现实世界中的绝大部分问题都是非线性的，神经网络想要表示这些问题就必须同样学习这些非线性表示。激活函数形式多样，利用它们的不同特性可以构成功能各异的神经网络。

在神经网络中，网络解决问题的能力与效率除了与网络结构有关外，在很大程度上取决于网络所采用的激活函数。激活函数的选择对网络的收敛速度有较大的影响，针对不同的实际问题，激活函数的选择也应不同。常见的激活函数有下列几种：

### Sigmoid函数

将每个输入的实数映射到（0,1）之间，单调连续，非常适合做输出层，并且求导容易。但是其具有饱和性，一旦输入落入饱和区，一阶导数就变得接近于0，很容易产生梯度消失。函数表示为：

### Tanh函数

将每个输入的实数映射到（0,1）之间，单调连续。输出是以0为中心，收敛速度比sigmoid函数要快。但是同样具有饱和性，无法解决梯度消失的问题。函数表示为：

### Relu函数

是整流线性单位的代表，将输入的实数映射到大于等于0的区间。Relu函数在x<0时是硬饱和。由于当x>0时一阶导数为1，所以，Relu函数在x>0时可以保持梯度不衰减，从而缓解梯度消失问题，还可以更快的去收敛。但是，随着训练的进行，部分输入会落到硬饱和区，导致对应的权重无法更新。我们称之为“神经元死亡”。函数表示为：

|  |  |  |
| --- | --- | --- |
|  |  |  |
| sigmoid | tanh | relu |
| 图2：几种常见的激活函数 | | |
|  | | |

偏置值是为了提供除了节点接收到的正常输入之外的可供训练的常量值，在神经网络中是非常重要的，在此不再赘述。

## 前馈神经网络

前馈神经网络是人工神经网络中最简单的一种，网络中有多个神经元，可以分为若干不同的“层”，相邻层之间的神经元之间存在着连接，每个连接都有相应的权重与之对应。各层按信号传输先后顺序依次排列，第i层的神经元只接受第(i-1)层神经元给出的信号，各神经元之间没有反馈。前馈神经网络中不存在回路，典型的前馈神经网络如下图所示：

|  |
| --- |
|  |
| 图3：典型前馈神经网络 |

一个前馈神经网络通常包含有以下三种类型的节点：

输入节点：输入节点的作用是将外界信息输入到神经网络中，所有的输入节点结合在一起称为“输入层”。输入节点上不进行任何的运算，直接将未经任何处理的信息输入传递给隐藏神经元。

隐藏节点：隐藏节点正如其名字，和外界没有直接的联系，是隐藏起来的。隐藏节点提供计算将输入节点输入的信息转变为输出节点将要输出的信息。隐藏节点组合成隐藏层，隐藏层连接输入层和输出层。一个前馈神经网络只有一个输入层、一个输出层，但是可以有任意个（0个或多个）隐藏层。

输出节点：输出节点将神经网络计算得到的结果输出到外界，所有的输出节点结合在一起称为输出层。输出节点与输入节点不同，输出节点上可以执行相应的计算。

可以看出，输入节点并没有计算功能，它只是为了表征外界信息输入矢量的各元素值。其余各层节点都是具有计算功能的神经元，可以统称为计算单元。每个计算单元可以有任意个输入，但只能有一个输出，同时这个输出可以传递到与之相连的多个单元上作输入。如果称输入层为第一层，计算节点各层从下往上依次称之为第2至第N层，由此构成N层前馈神经网络。BP神经网络就是典型的前馈神经网络之一。

## 反馈神经网络

|  |
| --- |
| 反馈神经网络与前馈神经网络不同，反馈神经网络的每个节点包括输入节点都是一个计算单元，而且同时接受外界输入和其它计算节点的输出作为反馈输入，另外还有自身反馈，即接受自身输出作为输入。典型反馈网络如下图所示： |
| 图4：典型反馈神经网络 |

## Rprop学习算法

神经网络由三部分组成，分别是基本单元神经元、基本结构、连接权值参数。神经网络学习分为参数学习和结构学习。参数学习的目的在于根据输入到神经网络中的数据，调整神经元之间的连接权值参数，使得整个网络的性能达到最优。结构学习的目的是根据输入的数据，调整神经元之间的连接方式，使网络性能达到最优。本小节着重介绍参数学习中的Rprop学习算法。

Rprop学习算法属于基于梯度的优化算法，也是神经网络有监督学习算法中最常用的一类学习算法，也是最简单的一类训练算法。通过不断迭代改变神经元之间的连接权值参数，使得实际输出和期望输出之间的误差最小，被认为是一种快速和鲁棒的学习算法。自适应梯度算法在每次迭代中具有单独的步长大小，即控制学习过程中每个单独的连接权重更新，以便最小化振荡并最大化更新步长，因此可以克服学习速率选择的固有局限性。算法分为前向传播和反向传播两部分。

### 前向传播

初始化时网络中所有的权重都是无规律随机分配的，我们只考虑隐藏神经元V。假设从输入到该节点的权重分别为W1、W2、W3，其中W1是偏置值。我们假设输入值为[35,67]，期望输出为[1,0]。那么隐藏神经元V的输出按照下式计算：  
其中f是一个激活函数，如Sigmoid函数。利用同样的方法可以计算出另一个隐藏神经元的输出，隐藏层两个神经元的输出作为输出层的输入，我们可以计算出输出层输出的结果。假设输出层输出结果为[0.4,0.6]，由于初始权值参数是随机的，所以输出结果预期结果相差很大，因此输出为错误输出。

|  |
| --- |
| Screen Shot 2016-08-09 at 11.52.57 PM.png |
| 图5：前向传播 |

### 反向传播

令表示从神经元j到神经元i权重，令E表示一个随权重变化而变化的任意错误度量。把偏差参数看做是来自外界额外的恒定输入的权重。在每次迭代中，权重按照以下方式进行修改调整：  
其中上标、表示学习迭代的代数，为调整的大小。

学习算法会在满足某个条件是终止，列如迭代次数t超出预定值或者E小于预定值。迭代中每次权重更新的方向是有偏导数正负符号决定的。权重每次更新的步长大小，分别针对每个权重进行单独的调整。则步长计算公式为：  
其中是符号函数，当参数为正数时返回，反之如果参数为负数则返回，当参数等于0是返回0。对于所有的，都初始化为。Rprop算法每次迭代更新都可以分为两个部分，第一部分是调整步长的大小，第二部分为更新权重。

首先介绍第一部分，调整步长的大小。对于每个权重多有一个单独的调整步长，步长的大小可以有下式计算得到：  
其中，为了防止步长变得过大或过小，它们以为界。如果偏导数在连续的迭代中具有相同的符号，则增加步长，而如果它在连续迭代中符号改变了，则减小步长。由前面的公式我们可以推导出公式：  
因此，我们可以知道的调整步长的大小变化的方向与基于网络优化误差得到的梯度的方向是一致的。

算法的第二部分是更新权重。根据上述公式确定步长更新的大小方向之后，有两种不同的情况需要考虑，如果偏导数的符号没有改变，则按照常规方法更新权重，但是当偏导数的符号改变之后，就需要恢复先前的权重：  
将偏导数设置为零可避免在下一次迭代中更新学习速率。在改进的Rprop算法中，认为上式中权重回缩的情况并不总是合理的。只有偏导数变化符号和逼近误差增加时，才恢复先前的权重。因此，权重缩回条件被修改为：。

# 单目标学习与多目标学习

学习算法可以分为两类单目标学习和多目标学习，其中多目标学习还可以分为标量化多目标学习和基于Pareto的多目标学习。

## 单目标学习

所谓单目标学习，是指只对一个目标函数进行优化的算法。比如数据挖掘十大经典算法之一，最常见的聚类算法K-means算法，它属于无监督学习算法。它以空间中所有的点分为k类，通过迭代的方法，逐次更新各聚类中心的值，直到得到最好的聚类结果。评价聚类结果的函数是  
其中是数据节点和类的中心之间的距离测量方法，N数据集中类的数量。所以K-means算法的唯一目标就是最小化函数，属于单目标学习算法。

## 多目标学习

许多现实生活中的问题涉及多个目标的处理。例如在有监督学习中，记忆训练数据不是唯一的目标。还经常需要考虑其他几个目标。在回归和分类中，一个学习模型不仅应该对训练数据具有良好的近似性，还应该对同一问题的不可见数据具有良好的近似性。但是这个目标不能通过最小化单个目标或任何相似误差测量方法实现。事实上，单纯最小化训练数据上的近似误差会产生过拟合现象，这意味着该模型在不可见数据可能表现不佳。换句话说，该模型不能归纳不可见数据。为了防止模型在训练数据上过拟合，必须控制模型复杂度。通常需要考虑的另一个共同目标是学习模型的可理解性或可解释性，当有监督学习用于从数据中提取知识时这点尤为重要。机器学习的可解释性强烈依赖于模型的复杂性，一般来说，复杂度越低，模型越容易理解。在这两种情况下都必须考虑第二个目标模型复杂度。但是这两个目标很有可能是相互矛盾的。当我们想要同时优化多个目标时，先验并不总是清楚哪些目标可能相互关联，以及它们如何相互影响。由于目标冲突，通常不存在单一的最佳解决方案。在这些情况下，我们的目的变为目找到一组决策变量的向量，这组决策变量满足约束而且最优化一组以目标函数为元素的函数向量。这些函数形成了函数之间通常相互冲突的性能标准的一种数学描述。简单地说，我们希望获得一组最佳的折衷解决方案。

多目标学习的数学表示为：  
其中是第个目标函数，分别为不等式约束函数和等式约束函数，是决策变量。

处理多目标问题有两种主要方法。最简单的方法是使用标量化函数，将多目标问题转化为标准的单目标问题，即标量化多目标学习。但是，当标量函数是非线性时，此转换可能无效。这种方法称为单策略算法，因为每次运行都收敛到单个解决方案。为了找到各种折衷解决方案，我们采用了几个参数化的标量化函数，并将它们的结果组合在一起。然而，从权重空间到客观空间的映射不能保证是同构的。这意味着权重的定义对于能否很好地覆盖帕累托政策的前沿的影响并不明显。

另一类算法是基于帕累托的多目标学习算法。与一次只关注单一解决方案相反，基于帕累托的多目标学习算法在单次运行中搜索一组最佳解决方案。进化多目标算法就是应用此类算法的典型算法，如SPEA2和NSGA-II，它们拓展了解决多目标问题的方式。这些进化多目标算法是解决多目标优化问题的最强大技术之一。

### 标量化多目标学习

标量化多目标学习简单来说就是将多目标问题转化为标准的单目标问题，然后利用单目标学习算法的求解方式得到一个解决方案。

在有监督学习中为了防止模型在出现过拟合现象，需要控制模型复杂度，这时候引入一个可以反应模型的函数作为第二个目标。可以将误差函数和模型复杂度这两个目标汇总为一个标量目标函数  
其中E是普通的误差函数，是模型复杂度的量度，例如模型中的神经元的数量，是一个大于零的常数，由用户定义，和分别是模型输出和期望输出，N是训练数据集中数据对的数量。通过这种方式，学习算法可以优化两个目标，尽管目标函数仍然是一个标量函数。

与有监督学习类似，在数据聚类中也经常需要同时考虑多个目标。一方面，很容易看出的是前式中定义的目标函数强烈地偏向球形集群，虽然有些可以通过修改距离测度函数改善，但是对于具有某些类型的聚类结构的数据，仅修改距离测度函数无法解决不了问题。另一方面，也有人提出在开发聚类算法时应考虑存在扰动时聚类方案变化的稳定性。所以数据聚类很多情况下也要作为多目标考虑。

标量化多目标学习算法主要存在两个主要缺陷。首先，确定适当能够反映用户目的的适当超参数λ是很重要的，超参数λ的很小改变有可能引起所求目标向量的显著变化，导致最终解得多样性差。其次，一次运行只能得到单一的解决方案，从中不能获得对问题的深入了解。因为不存在使所有目标同时优化的单一最佳解决方案，所以对于多个目标是相互冲突的情况下，这点尤为重要。这对于多目标学习尤其如此，例如，减小近似误差常常导致模型的复杂性增加。除了上述两个缺点之外，从优化的角度来看，即使超参数被适当地指定，使用标量目标函数也不能实现期望的解决方案。但是请注意，如果超参数在优化过程中动态改变，可以部分解决这个问题。另外经过多次改变超参数λ而得到的帕累托最优解集一般不会均匀分布。

### 基于Pareto的多目标学习

在介绍基于Pareto的多目标学习前我们需要先了解几个概念：

**帕累托支配(Pareto Domination)：**对于m个目标的最小化问题：如果，并且对于存在则称解决方案X支配解决方案Y，解决方案Y被解决方案X支配，记做。

**帕累托最优解(Pareto Optimal Solution)：**解决方案X如果不受任何其它可行的解决方案支配则被称为Pareto最优解。

**帕累托集(Pareto Set)：**由于目标相互矛盾，往往存在着多个帕累托最优解。对于一组给定的最优解集，如果这个集合中的解是相互非支配的，那么则称这个解集为帕累托集。

**帕累托前沿(Pareto Front)**：由帕累托集中的帕累托最优解对应的目标值向量组成的曲线或者曲面被称为帕累托前沿

实际中，我们通常不知真正的全局帕累托前沿在哪里，因此，由多目标进化算法(MOEA)得到的非支配解不一定是帕累托最优解。然而由多目标优化算法得到的非支配解基本可以认为是帕累托最优解。

基于帕累托的多目标学习算法遵循基于基于帕累托的多目标优化来处理学习问题。例如，有监督学习中误差函数和复杂度度量函数这个双目标学习问题可以表示为基于帕累托的多目标优化，如下所示：  
其中E为错误度量函数最常用的是均方误差函数，是神经网络的复杂度度量，比如权重的平方之和，或者权重的绝对值之和，或者神经元个数：

其中和分别是期望输出和模型的实际输出，N是训练数据集中数据对的数量。是神经网络模型中的权重值，M是所有连接的数量，n是神经网络模型中神经元的个数。前面提到的几种神经网络模型复杂度度量常常用于神经网络正则化，其中平方之和被称为高斯正则化，绝对值之和被称为拉普拉斯正则化。

基于帕累托的多目标学习与标量化多目标学习相比较，我们发现在基于帕累托的多目标学习中不在需要指定超参数。这样可以避免用户在学习之前确定超参数的负担。基于帕累托的多目标学习算法一次运行能够得到多个帕累托最优解，从中用户可以提取关于该问题的知识并在选择最终解决方案时做出更好的决策，然后用户在学习之后根据偏好从所得到的帕累托最优解集中挑选一个或多个解决方案。基于帕累托的学习方法的另一个潜在优点是多目标化可以帮助学习算法摆脱局部最优，从而提高学习模型的准确性。

# 多目标进化算法

多目标进化算法自从被提出之后就受到广泛的关注，越来越多的研究人员开始采用多目标进化算法去解决各种问题。在过去几年研究人员相继提出了一些多目标进化算法，如SPEA、PAES、NSGA、SPEA-II、NSGA-II等等，每个算法都有它自身的优劣点。如NSGA算法中优化个数可以是任意个，非支配解的分布也比较均匀。但是它的计算复杂度太高，没有精英保留机制，还需要预设共享参数，这些缺陷在NSGA-II中被改进。

## NSGA-II进化算法

NSGA-II是目前最流行的多目标进化算法。中文名为：一个快速和精英机制的多目标遗传算法(A Fast and Elitist Multi-objective Genetic Algorithm)，由Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan四位在2002年共同提出。它是在NSGA的基础上进行改进，克服了NSGA中的缺点。具有以下优点：

1. 不需要用户指定一些类似与共享参数之类的参数，这些参数的轻微改变都很有可能导致最终结果上很大的变化，所以用户主观指定的参数越少越好，系统也就也就越稳定。
2. 非支配排序（non-dominated sorting）的时间复杂度相对其他算法较低：NSGA的排序方法时间复杂度为，其中M为目标函数个数，N为种群个体数，NSGA-II的排序方法时间复杂度仅为；
3. 密集比较算子(Crowded-Comparison Operator)：既会考虑种群中的非支配解的rank值，也会考虑拥挤系数。
4. 精英保留策略：保留了父代和子代中的精英个体，可以明显地提高多目标遗传算法的效率。

### 快速非支配排序算法

首先，对每个解都要计算两个值：第一个是支配数，也就是支配解的解的个数，第二个是解所支配的解的集合。这一步需要次计算。

所有第一非支配解前沿面的支配数都是0，然后对于每个支配数的解p去依次访问它的所支配的解的集合，对于集合中的每个解q，将其支配数减一。通过这样， 如果任一成员的支配数减为0，就把它放到一个单独的集合Q中，Q中的解成员属于第二支配前沿。然后对Q中所有的解成员继续执行前一步的动作，以确定第三前沿面。直到所有的前沿面被确定。

有上述伪代码可以看出计算每个解的支配数，和其所支配的解的集合的时间复杂度为，其中判断是否被支配需要要次比较，因为有M个目标需要比较。在此基础上判断解属于第几非支配等级的时间复杂度为。这样算法总体的时间复杂度为，由于算法需要对每个解创建一个支配解的集合，所以所需的存储空间则变为。

### 保留多样性

在NSGA-II中使用拥挤比较算法代替了NSGA中的共享函数的方法来保持种群的多样性。共享函数的方法过于依赖共享参数的选择，而的值是由研究人员凭经验设定的。且共享函数的方法时间复杂度比较高，耗费时间。

**拥挤系数：**根据每个单独的目标函数计算当前点距两侧最近的点的平均距离，把这个数值作为当前点的拥挤系数。

为了计算拥挤系数，需要先根据每一个目标函数对种群进行排序。首先对每个目标函数的边界解即拥有当前目标函数最大值和最小值的解指定为无穷大的距离，对于其它剩余的解依次计算拥挤系数。依次变换不同的目标函数，知道所有的目标函数都遍历过一次。将使用每个目标函数计算得到的拥挤距离相加就得到了最终的拥挤系数，最后对拥挤距离做归一化处理。

其中代表集合I中第i个个体的第m个目标函数值，参数和是第m个目标函数的最大值和最小值。可以计算得到上述算法的计算复杂度为。

**拥挤比较算子()：**假设种群中每一个体有两个属性：

1）非支配排名()

2）拥挤系数()

如果()或()并且()则可以说。

也就是说。对于两个有不同非支配排名解中，我们会选择排名更低（更好）的解。如果两个解具有相同的非支配等级，那么我们会选择处于不太拥挤区域的解，以保持种群的多样性。

### 主循环

最初随机创建父代种群P0，作为第一代种群。使用父代种群进行交叉变异生成指定大小的子代种群，然后合并父代种群和子代种群作为一个合成种群，在合成种群中按照非支配等级排名和拥挤系数选择出下一代的父代种群。依次重复只带满足终止条件。

合成种群由父代种群和子代种群组合而成，即，其中种群Rt大小为2N,种群和种群的大小均为N。由于父代和子代种群都在合成种群中，父代和子代中的精英(非支配排名等级低)也就都保留了下来。在合成种群进行非支配等级排序把每个解都分配到对应的支配前沿中，即。如果的大小比N小，我们就明确的将集合所有成员归为新种群中。种群剩余的成员需要在之后的非支配前沿中进行选择，也就是在中进行选择，然后在中进行选择，依次进行，直到所有的前沿面都加入进去或者已选择的前沿面的大小超过所需的大小。实际上，前沿面到前沿面中所有解的数量肯定会比种群所需要的大小要大，即，其中是前沿面中解得数量，所以只需要考虑选取前面几个前沿面。由于选择出的种群中的个体个数是固定的，所以需要通过拥挤比较算子将最后选择的前沿面中的解按降序排列并且选择最好的解来填充新生成的种群中剩余的解。

|  |
| --- |
|  |
| 图6：NSGA-II |

新生成的大小为N的种群重新进行交叉、变异操作，从而重新生成新的大小为N的子代种群。然后继续重复上一步的操作，直到满足终止条件，停止算法。

# 实验设计

实验主要由初始化、交叉变异、评估和选择这几个流程组成，实验的主要流程如下图所示：

|  |
| --- |
|  |
| 图7：基于多目标优化的人工神经网络结构设计主要流程 |

## 神经网络模型设置

在实验中我们使用全连接的前馈神经网络，至少包含一个输入层，一个隐藏层和一个输出层，其中隐藏层神经元是非线性的，输出层神经元是线性的。神经网络中还有一个dropout参数可以用来剔除部分弱连接。

隐藏层神经元采用Relu函数作为激活函数，Relu函数是目前最流行、最常用的激活函数。初始连接权参数随机初始化，偏置值全部设置为0。设置隐藏层数最多4层，最少1层，每层的神经元个数最多10个，最少1个，也就是说最少存在1个隐藏神经元，最多可以存在40个隐藏神经元。神经网络的结构确定之后采用Rprop算法对连接权值和偏置值进行精确调整。

Relu函数和Rprop算法已在前面进了介绍，在此不在展开。

|  |
| --- |
|  |
| 图8：实验所用的隐藏层神经元 |

上图中神经元的输出，神经元的输出最小为0。在使用dropout参数时，一些影响很小的连接会被去掉，具体表现为相应的连接权值被置零，此时神经元的输出公式是不变的。

## 神经网络编码

神经网络的编码方式关系着整个进化神经网络的构建和执行方式、效率。在本次实验中我们采用一种可变长度的直接编码方式对神经网络结构进行编码。直接对神经网络的隐藏层数和隐藏层中每层神经元的个数进行编码，可以直接控制神经网络的层数和神经元个数进行进化和变异。

假设神经网络由输入层、两层隐藏层和输出层组成，每层分别有4个、5个、3个、3个神经元，一共15个神经元。则可以之间按照每层的神经元个数进行编码即，编码的长度就是神经网络的层数。由于数据集中每个数据对的特征维数是固定的，这就意味着所有的数据对需要的输入神经元的数量是固定的。数据集中数据的类别也是可以确定的，所以输出神经元的个数也就可以确定。也就是说，对于一给定数据集，构造神经网络时，网络的输入层和输出层神经元个数是确定的，我们只需要对隐藏层的层数和每层的神经元个数进行调整。所以在编码时可以忽略输入层和输出层，只对隐藏层进行编码，上述神经网路模型的编码也就变为了。这样既可以减小编码的长度节省空间，同时减少了交叉和变异的操作复杂度，不需要考虑当前基因是否属于输入或输出层，可以对编码序列中的任意一个基因进行操作。下图展示了对一个四层的神经网络模型进行编码的过程。

|  |
| --- |
|  |
| 图9：使用层数和每层神经元个数直接对神经网络进行编码 |

## 交叉

由于我们采用的编码方式编码出的基因序列的长度是可变的，并不方便直接用来进行交叉操作，所以在进行交叉操作之前，需要先对所有的基因序列进行“对齐”操作。将所有的基因序列都扩展到相同的长度，比如我们规定神经网络模型最多存在个隐藏层，就把所有的基因序列的长度扩展为，因为所有基因序列的长度都不可能超过。如果基因序列长度小于规定长度，即，则需要对基因序列进行“对齐”。首先生成个从到的不同的随机数，然后将这个不同的随机数按照升序排列，得到一个长度为，升序排列的序列。最后按照序列将基因序列中的基因依次放入扩展基因中。

|  |
| --- |
|  |
| 图10：扩展基因序列 |

将所有的基因序列扩展完成之后，采用Position-based Crossover(PBX)算子对基因序列进行交叉操作。

首先随机选择一对基因序列中的几个基因，选择的个数也是随机的，选择位置可以不连续。按照选择的位置生成一个子代，子代中选中的位置和父代中被选中的基因的位置是一致的，需要注意的是此时子代中没有被选中的基因的内容是空缺的。最后将另一个父代基因序列按照对应的位置填入上一步生成的子代中空缺的位置。两个父代交换位置，其它条件不变，包括第一步中选择的基因的位置，然后按照相同的步骤生成另一个子代。至此交叉操作完成。

|  |
| --- |
|  |
| 图11：交叉操作 |

## 变异

影响多目标进化算法性能的因素很多，除了选择和交叉操作职位，变异算子同样发挥着重要的作用。在算法运行前期，由于初始种群是随机生成的，所以个体之间的差异相对较大，此时交叉算子和选择算法的选择是最重要的，变异操作的影响较小。但是在算法运行后期，个体之间差异减小，此时交叉算子和选择算法的选择相对前期影响减小，相对的变异操作就起到了至关重要的作用。所以变异算子的选择和算法的性能有很大的关系，一个合适变异算子可以是算法稳定，生成的种群具有良好的多样性。

实验中我们采用具有自适应的变异率的变异算子。在个体差异较大时使用较大的变异量和变异率，可以降低算法发生早熟的可能性，使生成的种群的具有良好的多样性；相对的，个体之间差异较小时则采用较小的变异量和变异率，可以使算法在相对较小的范围进行更加精确的搜索，已找到最优解。

变异率：，其中，t为进化的代数，是一个常数，用来控制变异率的变化快慢。

最大变异量：，其中是一个常量，用于定义全局最大变异量，通常取值每层最大节点数的；是一个常数，用来控制最大变异量变化的快慢。

|  |
| --- |
|  |
| 图12：不同值下的变异率变化，从上到下值依次为0.1、0.2、0.5。 |
|  |
| 图13：不同值下的变异量最大值的变化，其中，从上到下值依次为1.0、1.2、1.5。 |

## 评估

本次实验中采用两个目标作为评估函数分别为隐藏层神经元个数和均方误差(MSE)。

隐藏层神经元个数可以直接从基因序列中得到即，其中为种群的基因序列中的第个基因。

在得到结构之后，使用Rprop算法进行学习，精确调整权重，得到训练完成的神经网络。然后计算当前神经网络的均方误差：  
其中和分别为模型输出和期望输出，N是数据集中数据对的数量。

## 选择

使用前文所介绍的NSGA-II多目标进化算法进行选择操作。首先合并父代和子代种群，按照目标函数值即隐藏层神经元个数和均方误差进行非支配度排序，并计算拥挤系数。然后按照计算出的非支配度排名和拥挤系数选择出新的父代种群。

## 参数设置

本次实验所用的参数设置如下表所示：

|  |
| --- |
| 表1：算法参数设置 |
| |  |  | | --- | --- | | 神经网络模型设置 | | | 隐藏层最大层数 | 4 | | 隐藏层每层最多节点数 | 8 | | 进化算法参数设置 | | | 种群大小 | 50 | | 迭代进化次数 | 50 | |

## 实验所用的数据集

本次实验使用Google TensorFlow提供的鸢尾花数据，该数据集包括120个训练实例iris\_training和30个测试实例iris\_test。数据集分为三类，分别为山鸢尾(setosa，使用0表示)、变色鸢尾(versicolor，使用1表示)和维吉尼亚鸢尾(virginica，使用2表示)。每个实例具有4个属性，用于预测鸢尾花的类别。

由于鸢尾花数据相对简单，数据量也比较小在剔除弱节点时效果不太明显。所以我们在做剔除弱节点的影响分析时，用的数据集为MNIST数据集。MNIST相对鸢尾花数据集可用属性和实例的个数都要多很多，所以构建的神经网络相对更加复杂，更适合分析。

# 实验结果分析

根据前文介绍的实验流程和参数的设置，最终得到了由生成的许多个Pareto最优人工神经网络模型组成的一个Pareto前沿。我们可以根据得到的Pareto前沿进行模型选择。并根据前文神经网络模型设置中提到的dropout参数的设置分析剔除弱节点对神经网络模型的影响。

## 识别可解释模型

神经网络的可解释性主要取决于其复杂性。网络越简单，就越容易理解神经网络中所概括的知识。使用基于Pareto的方法进行多目标优化时，最后可以得到许多Pareto最优解，在本实验中就是很多Pareto最优神经网络结构模型。Pareto前沿中的越是简单的模型越是容易从中提取可解释模型。

我们先简单介绍一下我们所采用的提取规则的方法。我们考虑一个最简单的模型，只有一个隐藏神经元，单输入，单输出，隐藏层仍然使用Relu函数作为激活函数，神经网络模型如下图。对于二元分类问题，我们定义如果输出小于0.5，实例被标记为第一类，否则被标记为第二类，即：

|  |
| --- |
|  |
| 图14：用于提取逻辑规则的典型简单网络 |

实例属于第一类的条件为：，且我们知道，其中z为隐藏神经元的输出。因此我们可以把属于第一类的规则变换为：，由此我们可以得到：  
我们假设，因为隐藏层激函数为Relu函数，可以得到：  
因为，上式可以变换为：  
继而得到：  
设，就可以得到下面两条规则：

实验得到的Pareto前沿图如图15所示，其中包含多个Pareto最优结构模型，其中最简单的神经网络结构只有一个隐藏神经元。

|  |
| --- |
|  |
| 图15：实验得到的Pareto前沿 |
|  |
| 图16：只有一个隐藏神经元的神经网络结构模型 |

隐藏层输出为z，从图16我们可以得到：  
属于第一类的条件为：  
即：  
从而我们可以得到：  
由上图中的神经网络中可知：  
从而我们可以得到属于第一类的条件为：

利用同样的方法我们可以提取出属于第二类和第三类的条件。而且我们从模型中还可以发现和的权重相对于和较大，这也说明了和所对应的属性在判断实例具体哪一个类别时较其它属性更加重要。

我们可以发现通过将精度与复杂度进行权衡，使用基于Pareto的多目标优化算法能够找到解决问题最简单的最简单结构。此外，我们可以从简单的Pareto最优网络中提取出神经网络中嵌入的主要知识，从而可以提取可解释的逻辑规则。

## 模型选择

使用基于多目标优化算法进行神经网络模型进行设计时，我们可以得到一个关于精度和复杂度相互权衡的一个Pareto前沿，之后我们可以根据这个前沿进行经验性的模型选择，已选择出在不可见数据上依然具有良好知识概括能力的模型。我们的目标就是寻找一个合适的模型，使得模型的复杂度与数据集的复杂度相匹配。

数据的复杂性可以通过归一化性能增益(NPG)来确定，NPG的定义如下：  
其中其中，和，是训练数据上第和第个Pareto最优解的𝑀𝑆𝐸和神经元个数。当解决方案按照逐渐复杂的顺序排列时，以下关系成立

当模型的复杂度小于数据的复杂度时，模型复杂度的增长会导致神经网络性能的显著提高，可以体现为NPG的值大于0，但是随着模型复杂度和数据复杂度的接近，神经网模型复杂度的增长对神经网络的性能影响逐渐减小，也就是说NPG的值会逐渐降为0。可以认为这时的神经网络模型复杂度和数据集的复杂度相匹配。如果神经网络模型的复杂度继续增加，虽然会进一步提升模型在训练数据集上的性能，但是对于不可见数据则会发生过拟合的现象。

|  |
| --- |
|  |
| 图17：Pareto最优解的准确性与复杂性：“\*”表示训练数据，“o”表示测试数据。 |
|  |
| 图18：鸢尾花数据集上的NPG |

由上图我们可以发现，NPG在隐藏神经元个数为6时首次降为0，也就说明对于鸢尾花数据集神经网络的隐藏神经元个数应该为6。此时的神经网络可以在保证性能的同时具有良好的泛化能力。

## Dropout的影响

Dropout可以在训练模型的时候，随机丢弃掉一部分神经元。也就是神经元再一次训练过程中，有一定的概率是不工作的，不参加神经网络的计算，其权值也不会被更新。示意图如下：

|  |  |
| --- | --- |
|  |  |
| 图19：标准神经网络模型 | 图20：使用dropout后的神经网络模型 |

实验中我们使用TensorFlow中的dropout功能，剔除部分节点，用来防止过拟合。使用dropout之后模型的收敛速度会变慢，在训练数据上的正确率也会相对降低。在没有使用dropout时训练数据集上的正确率可以达到一个很高的值，但是将模型应用于不可见数据即测试数据上的时候模型正确分类的能力却不理想，也就是说发生了过拟合。在使用了dropout时，虽然模型的收敛速度变慢，但是模型在训练数据集和测试数据集上的性能是很相近的。

对于复杂度很高神经网络模型，通过使用Dropout，剔除某些节点，可以很好地防止过拟合。但是对于较为简单的模型，效果可能胡不太明显。

# 总结与展望

基于多目标优化进行神经网络结构的设计方法为研究神经网络的结构设计提供了一种新的观点。通过进化算法对神经网路结构的优化，我们可以更加深入的了解神经网络，从而开发新的算法。

本文介绍了一种人工神经网络结构设计的方法，通过实验说明了如何生成可解释模型以及泛化模型选择的方法。此外我们还简单介绍了Dropout的概念，以及使用之后的对神经网络模型的影响。

在神经网络结构设计方面还有许多的方法值得去尝试，许多问题仍有待解决。例如选种配对机制、最优解的质量评估、种群进化终止的条件等等，这些问题都是当前研究的热点。

目前关于人工神经网络结构设计是国内外的研究热点，其中基于多目标优化的方法具有很大的优势，也受到了很多的关注。到目前为止大部分研究都把重点放在偏差和方差的折衷，但是机器学习和人类记忆系统中另一个重要的想法是塑性和稳定性的折衷，也称在线学习或增量学习。因此关于这方面的研究也许会成为研究者研究的新热点。

# 致谢

四年的时间转瞬即逝，转眼间就要为四年紧张而又充实的大学生活画上句号。在这校园中到处都充满了回忆，由迷茫时的痛苦，也有成功时的喜悦，有悲伤，也有快乐。

首先，我要感谢我的指导老师潘林强老师以及何成、李良昊两位师兄对我的悉心教导。从毕设的选题到最终的论文的撰写定稿，他们都给了我悉心的指导和热情的帮忙，使我的毕业设计能够顺利的完成。

其次，感谢自动化学院的全体领导和老师，由于他们的悉心教导，我学到了专业的知识，掌握了扎实的专业技能。

最后，感谢我的家人在此期间给予我的包容、关爱和鼓励，以及所有陪我一路走来的同学和朋友，正是由于他们的支持和照顾，我才能安心学习，并顺利完成我的学业。

毕业在即，在今后的工作和生活中，我会铭记师长们的教诲，继续不懈努力和追求，来报答所有支持和帮忙过我的人!

# 参考文献

参考文献