文本复制检测报告单(全文标明引文)

№:ADBD2017R_20170525161401430278270672 检测时间：2017-05-25 16:14:01

检测文献： 基于CUDA的遗传算法和神经网络在股票趋势问题中的应用研究

作者： 王劭阳

检测范围：
中国学术期刊网络出版总库
中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库
中国重要会议论文全文数据库
中国重要报纸全文数据库
中国专利全文数据库
互联网资源(包含贴吧等论坛资源)
英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)
港澳台学术文献库
优先出版文献库
互联网文档资源
图书资源
CNKI大成编客-原创作品库
大学生论文联合比对库
个人比对库

时间范围： 1900-01-01至2017-05-25

指导教师： 任健

检测结果

总文字复制比： 2.3% **跨语言检测结果：** -

去除引用文献复制比：2.3% **去除本人已发表文献复制比：2.3%**

单篇最大文字复制比：0.8%（基于自适应遗传算法的机组排班方法研究和应用）

重复字数：	[574]	总字数：	[25098]	单篇最大重复字数：	[210]
总段落数：	[3]	前部重合字数：	[212]	疑似段落最大重合字数：	[445]
疑似段落数：	[2]	后部重合字数：	[362]	疑似段落最小重合字数：	[129]

指　标： ☐ 疑似剽窃观点 ☑ 疑似剽窃文字表述 ☐ 疑自我剽窃
☐ 一稿多投 ☐ 过度引用 ☐ 疑似整体剽窃 ☐ 重复发表

表格：**0** 脚注与尾注：**0**

4.5%	(445)	基于CUDA的遗传算法和神经网络在股票趋势问题中的应用研究_第1部分（总9940字）
0%	(0)	基于CUDA的遗传算法和神经网络在股票趋势问题中的应用研究_第2部分（总10023字）
2.5%	(129)	基于CUDA的遗传算法和神经网络在股票趋势问题中的应用研究_第3部分（总5135字）

（注释： 无问题部分 文字复制比部分 引用部分）

1. 基于CUDA的遗传算法和神经网络在股票趋势问题中的应用研究_第1部分 总字数：9940

相似文献列表 **文字复制比：**4.5%(445) **疑似剽窃观点（0）**

1	基于自适应遗传算法的机组排班方法研究和应用 赵红竹(导师：夏洪山) - 《南京航空航天大学硕士论文》- 2008-01-01	2.1%（210） 是否引证：否
2	WDM波长网络优化设计和动态路由的研究 - 豆丁网 -《互联网文档资源（http://www.docin.com）》- 2012	1.7%（171） 是否引证：否
3	图像增强模型及算法研究 卢丽敏(导师：周海银) - 《中国人民解放军国防科学技术大学硕士论文》- 2002-11-01	1.5%（148） 是否引证：否
4	基于机组实时性能的火电机组负荷优化调度 彭兴(导师：徐治皋) - 《东南大学硕士论文》- 2006-03-23	1.3%（134） 是否引证：否
5	基于机组实时性能的火电机组负荷优化调度 - 豆丁网 -《互联网文档资源（http://www.docin.com）》- 2016	1.3%（134） 是否引证：否
6	基于遗传算法的Web服务组合优化 毛一梅;乐嘉锦; - 《计算机应用与软件》- 2008-11-15	0.8%（79） 是否引证：否
7	基于服务质量的Web服务关键技术研究 毛一梅(导师：乐嘉锦) - 《东华大学博士论文》- 2008-10-01	0.8%（79） 是否引证：否
8	基于无人机的飞推综合控制研究 周小平(导师：朱金陵) - 《西南交通大学硕士论文》- 2008-06-01	0.6%（61） 是否引证：否
9	锅炉给水泵运行方式优化及其控制的研究 邓春(导师：何秀华) - 《江苏大学硕士论文》- 2007-06-01	0.5%（54） 是否引证：否
10	人工神经网络在农村土地利用分类中的应用 郭小英;何东健; - 《农机化研究》- 2011-01-01	0.4%（44） 是否引证：否
11	村域循环农业结构分析、评价与优化 陈冬梅(导师：卞新民) - 《南京农业大学硕士论文》- 2006-07-01	0.3%（30） 是否引证：否

原文内容

基于CUDA的遗传算法和神经网络在股票趋势问题中的应用研究

学生： 王劭阳

指导老师： 任健

摘要

在本论文中，我们针对股票趋势问题开展研究，使用基于CUDA的并行化遗传算法、人工神经网络这两种算法通过舆情数据进行股票趋势的预测。首先通过对问题的抽象建模，将问题转化为合适的数学形式，再设计具体的算法，然后使用历史数据将两种模型训练出来，最终得到能够根据当日舆情预测下日股票涨跌的遗传算法和人工神经网络。本论文还进行了遗传算法和人工神经网络在此问题上效果的优劣对比，最后使用CUDA对这两种算法进行并行化的优化，使训练性能提高，并对比了优化前后的性能。本论文为基于CUDA的遗传算法和神经网络在股票趋势问题中的应用提供了解决方案。

本论文实现的算法使用Python语言开发，在人工神经网络的开发上借助了Keras和Theano框架来实现，在遗传算法和人工神经网络部分都借助了Nvdia公司研发的CUDA框架进行使用GPU并行计算的优化。

关键词：股票趋势预测，遗传算法，人工神经网络，CUDA加速

Application and Research of CUDA based Genetic Algorithm and Artificial Neural Network on Stock Trends Problem

Author: WANG Shaoyang

Tutor: REN Jian

Abstract

In this thesis, we focus on how to make predict of stock trends using CUDA paralleled genetic algorithm and artificial neural network with public opinion data.

First of all, we build an abstract model of the problem to convert the problem into an appropriate mathematical form. Then we design the specific algorithm and use historical data to train these two models. And ultimately,

we are able to predict stock trend of the next day under the public opinion of the stock market today using genetic algorithm and artificial neural networks.

In this thesis, the advantages and disadvantages between genetic algorithm and artificial neural network on this problem are compared. Then we use CUDA to optimize these two algorithms, which improves the performance of the training, and we also do the compare of the performance between algorithms before and after optimization. This thesis provides a solution for the application of CUDA based genetic algorithm and neural network on stock trends prediction.

The algorithms implemented in this thesis are developed by Python. Keras and Theano framework are used in the development of artificial neural network. In the development of both genetic algorithm and artificial neural network, we use the CUDA framework developed by Nvdia to optimize the training speed.

Key words: Stock trends，Genetic Algorithm，Artificial Neural Network，CUDA acceleration

1 绪论

1.1 研究背景

金融产品（股票、外汇等）价格的趋势问题一直以来都是众多百姓、学者、企业、政府非常关心的问题之一，吸引了大量研究者兴趣。许多百姓希望能够精准地把握股票市场的每个走向，在股票的涨跌之间捞取财富，让自己赚个盆满钵满；学者们希望早日攻破股票趋势预测这一难题，总结出理论，建立起科学研究和实际问题间的联系；企业希望尽早获取股票趋势的情报，在竞争中先别人一步，获取优势；政府希望根据预测，提前制定方案，更有效地进行市场的调控。

近些年，诸如“微博”、“微信”等SNS网站和服务迅速崛起，“腾讯新闻”、“华尔街日报”等资讯类网站纷纷开辟用户评论的版块，“知乎”、“百度知道”等知识分享类的网站也开始流行。大量网民在这些网站上获取别人留下的信息，有时也分享自己的观点。在这网络时代，政府和企业也跟随时代的潮流，在互联网上公布政策和企业经营状况的信息。从互联网上这些大量的信息、大量的数据中，可以梳理出民众对于一些股票的想法、对于股票涨跌的判断，而民众依据自己的判断买入或卖出股票，反过来又影响了这些股票的涨跌。

在计算机领域内，各种与机器学习算法相关的研究正在进行，一部分研究将这些算法用来解决实际的问题。许多机器学习算法的目标是通过在大数据集的训练，使计算机可以模拟人类的学习行为，建立知识。由于训练的数据集经常很大，基于CUDA的并行计算模型也常用于机器学习的算法中。

本论文希望通过对舆情大数据和基于CUDA的遗传算法、人工神经网络这两种机器学习算法的研究，得出一套能够用于预测股票趋势的模型，用户可以据此以预测股票市场的动向，并为根据舆情数据来预测股票涨跌的问题提供解决方案。

1.2 国内外研究现状

1.2.1 国外研究及应用现状

由于股票市场预测在商业上有着极具吸引力的好处，所以在近几年成为了最热门的研究领域之一[1]。不幸的是，股票市场在本质上是复杂的、动态的和混乱的[2]。一般来说有三种关于这种预测的学术观点：

1）第一种学派认为，基于历史和现在的信息，投资者不可能稳定地获得超过交易获利平均值的利益。这种学派的支持理论包括Random Walk Hypothesis和Efficient Market Hypothesis[3]，这篇文献[4]也对这些理论提供了证据。

2）第二种学派认为，可以通过作基础分析，对各种宏观经济因素进行深入研究，研究有关行业的财务状况和业绩，从而发现各因素与股票趋势之间的联系[1]

3）第三种学派为各种金融数据建立模型来做预测。许多基于人工神经网络的模型使用历史和现在的数据来预测未来的股票趋势，人工神经网络由于其能以高精度逼近任意非线性函数而越来越受到欢迎[6]

目前已经存在许多方案来预测股票趋势，比如Clements和Hendry等在2009年实现的线性回归算法[7]、Kara等在2011年实现的人工神经网络[8]、Nayak等在2012年实现的遗传算法[9]、Q. Wen等在2010年实现的支持向量机[10]等等。

1.2.2 国内研究及应用现状

在国内，对股票趋势问题的研究也不少，吉根林等在2002年实现了基于人机交互的遗传算法用来挖掘股票投

资的风险规则[11]，卢琇泽等在2010年提出了一种后向传播的仍公司将网络模型并应用在了股市分析上[12]，张炜等在2015年提出了一种基于遗传算法的粗糙集属性约简方法的股票预测模型[13]。

1.3 研究目标与内容

1.3.1 研究目标

本论文旨在针对股票趋势问题开展研究，即使用基于CUDA的并行化遗传算法、人工神经网络这两种机器学习算法通过舆情数据进行股票趋势的预测。首先通过对问题的抽象、建模，将问题转化为容易应用两种算法的形式，再设计具体的算法，然后使用历史数据将模型训练出来，最终得到能够根据当日舆情预测下日股票涨跌的遗传算法和人工神经网络，并进行遗传算法和人工神经网络在此问题上的优劣对比，最后使用CUDA对这两种算法进行并行化的优化，使训练性能（训练速度）提高，并对比优化前后的性能。

1.3.2 研究内容

本课题的研究内容如图 1.1所示，具体分为以下几点：

1) 针对通过舆情数据进行股票趋势预测问题进行抽象。这是本论文的第一阶段，主要是将论文研究的问题抽象为数学模型，用数学定义清楚，方便之后的研究，也消除用语言表达常会产生的歧义性。

2) 建立使用遗传算法解决通过舆情数据进行股票趋势预测问题的模型，设计遗传算法的交叉、变异等操作的方式，并编程实现，调整参数，训练并验证模型。

3) 建立使用人工神经网络通过舆情数据进行解决股票趋势预测问题的模型，设计网络结构、神经元、损失函数等涉及到的组成元素，并编程实现，调整参数，训练并验证模型。

4) 对比遗传算法和人工神经网络两种算法在通过舆情数据进行解决股票趋势预测问题上的效果

5) 使用CUDA对遗传算法和人工神经网络进行并行化的优化，改进两种算法的训练性能，并对比优化前后的性能。

本论文主要使用python进行开发，使用Keras和Theano框架进行人工神经网络部分的开发，使用CUDA和PyCUDA进行算法并行化的开发。

4210053038475图 STYLEREF 1 \s 1. SEQ 图 * ARABIC \s 1 1 研究内容图 STYLEREF 1 \s 1. SEQ 图 * ARABIC \s 1 1 研究内容421005192405

1.4 论文结构

本论文的主要内容是使用基于CUDA的并行化遗传算法、人工神经网络这两种算法解决通过舆情数据进行股票趋势预测的问题，同时对比两种算法的效果，并对比无CUDA优化和有CUDA优化的算法训练效率。

论文的总体结构如下：第一章是绪论，说明了基于CUDA的遗传算法和人工神经网络解决通过舆情数据进行股票趋势预测问题的背景、研究现状和研究目标与内容。第二章是相关技术，对论文中用到的遗传算法、人工神经网络、Theano、Keras、GPGPU年计算、CUDA等技术作了简要的介绍。第三章是算法设计，这章作为本论文的核心内容，从问题定义与抽象到数据格式说明、从遗传算法和人工神经网络的问题建模到算法细节再到并行化处理，详细地介绍了本论文如何设计算法来解决问题。第四章是算法实现，展示了本论文的开发环境和整体系统及系统各模块的具体实现方式。第五章是实验验证，设计了算法的验证方式，展示了遗传算法和人工神经网络的正确率及两者之间的对比，还分别展示了两种算法使用CUDA优化前与优化后的性能及性能对比。最后是结论，总结了本论文做的全部工作。

2

相关技术

2.1 遗传算法

在自然界中，有大量的不同物种的生物生存繁衍，存活在自然界中的生物显示出了它们对自然环境的优秀的适应力。人们受到这些现象的启发，投身于研究和模拟生物的各种特性，研究其如何保证种群的生存，如何从大量灭亡的种群中个脱颖而出，为设计和开发人工自适应系统提供了强有力的参考。遗传算法（Genetic Algorithm）就是这种用计算机模拟生物行为的算法中令人瞩目的重要成果。基于模拟生物遗传与进化的过程，遗传算法使各类人工系统具有很好的自适应能力和优化能力。通过使用交叉、变异、优胜劣汰等启发式算子，遗传算法经常被用来解决优化问题和大空间搜索问题，解的质量通常很高[14]。

在遗传算法中，包含候选解（一般称作个体）的种群会随着算法的进行向更优的方向进化。每个候选解都有一

系列的属性（基因序列），编码为计算机可理解的数据，通常是二进制的01字符串，但也有其他的编码方式。遗传算法以编码后的基因作为运算对象，而传统的优化算法往往直接优化计算编码前的实际值本身。这种处理方式使得我们可以应用遗传算法来解决一些无数值概念只有代码概念的优化问题。

演化过程通常从一个充满随机生成的个体的种群开始，每次迭代会产生一个新一代种群，称做一次进化。在每一代种群中，需要计算每个个体的适应度。适应度通常是基因的一个函数，适应度越高，离问题的最优解越近。进化时要从当前种群选择适应度较高的个体，通过合并原基因并且做随机变异的方式据此生成新的个体，重复此过程产生出新的种群。新一代种群又被用于下一次迭代来产生又一个新种群。一般在达到最大迭代次数或种群中某个体适应度已经足够高时，算法就会停止。

典型的遗传算法需要（1）在候选解和基因序列之间编解码的方式；（2）用于评估基因的适应度函数。基因序列的表示一般是01数组，也可能是整数数组、浮点数数组、字符数组、自定义结构的数组等。这样表示基因序列的主要目的是使它们的基因在长度上对齐，并且具有固定的大小，有利于方便地进行交叉变异等操作。可变长度的基因有时也会被使用，但实现起来更加复杂。

种群的规模取决于实际问题，通常包含成百上千的个体。初始化时，一般会选择随机生成的种群，允许随机生成的个体遍布整个解空间（又称搜索空间）中，有时随机“播种”的个体刚好会出现在最优解的附近。

遗传算法通常用来解决全局最优化问题，特别是目标函数非常复杂的问题。因为不依赖目标函数梯度的交叉变异等混合操作可以有效地让种群跳出局部最优，而爬山法和梯度下降就容易陷在局部最优中。又因为遗传算法不依赖目标函数的梯度，直接以目标函数值作为搜索信息，所以非常适合目标函数不可微不可导或者很难求导的问题，泛用性很高。

传统的优化算法一般都会从单个初始值开始迭代求解问题，而遗传算法是从多个初始值开始搜索，从单初始值搜索容易误入局部最优解，而遗传算法的搜索覆盖面大，有利于全局择优，这是遗传算法与传统优化算法相比的优势。

由于遗传算法中的交叉变异操作都是具有随机性的，与传统的优化算法通常使用人工制定的规则不同，所以遗传算法属于随机化搜索的一种。传统的优化算法通常使用确定性的搜索方法，从搜索空间中的一个点到另一个点有确定的转移方法和转移关系，这种确定性往往也有可能使得搜索永远达不到最优点，因而也限制了传统优化算法的应用范围，遗传算法则不然。虽然这种随机的特性也会在种群中产生一些适应度不高的个体，但是随着进化过程的进行，新的群体中总会更多地产生出许多优秀的个体。算法中选择的目的是不断逼近最优解，交叉的目的是组合并保留可能产生最优解基因，变异的目的是使基因向解空间中随机的区域延伸、扩大搜索的范围。

2.2 人工神经网络

人工神经网络（Artificial Neural Network），简称神经网络，是一种用于机器学习、计算机科学和其他学科研究的计算模型，基于大量简单单元（称作人造神经元）的连接来模拟生物大脑中的神经系统。在神经元之间的连接上传递有不同强度的激活信号，如果组合的输入信号足够强，神经元将被激活，并且信号传播到与其连接的其他神经元。这种系统可以不用显式地编程，而是通过样本训练出来，并且能很好地解决某些使用传统算法时不好找到问题解的模型的问题。像其他机器学习方法一样，神经网络已经被用于解决使用普通的基于规则的编程难以解决的各种各样的任务，如计算机视觉和语音识别。

如图 2.1，通常，神经元会被组织成层的形式，有输入层（input）、隐层（hidden）和输出层（output），隐层可以多于一层，神经元在层间连接。现代的神经网络项目通常包括几千到几百万个神经元和数百万个连接，这样的网络的计算能力类似蠕虫的大脑，比人类大脑简单几个数量级。神经元的状态和神经元间传递的信号通常是实数。在每个连接和神经元上可以存在阈值函数，使得传播的信号的大小不能超过限制。反向传播算法用来根据已知数据训练神经网络，使用正向的刺激来修改连接权重。然而，神经网络的每次训练是否成功是难以预知的，训练结束后，有些网络在特定问题上表现良好，有些1259205611505125920541230 55图 STYLEREF 1 \s 2. SEQ 图 * ARABIC \s 1 1 人工神经网络的层次图 STYLEREF 1 \s 2. SEQ 图 * ARABIC \s 1 1 人工神经网络的层次则不然，顺利的训练通常要花费很多个交互周期。

尽管有些神经网络的类型十分抽象，但大多神经网络的目标都是希望网络能以与人类相同的方式来解决问题。生物领域内对大脑的新研究经常会衍生出神经网络的一些新的模式，比如让神经元动态跨层连接等。

2.2.1 Theano

Theano是一个用于定义、优化和计算各种数学表达式的Python框架[15]，尤其擅长多维数组相关的计算。只需要切换一项配置，就可以使计算运行在CPU或GPU（使用CUDA）上。Theano的在大数据集上的性能与用C语言手工实现的性能相近。并且Theano支持对用户透明的使用CUDA优化计算的功能，可以使用户节省大量的开发时间。Theano同时还提供了建立和训练各种神经网络能力。

2.2.2 Keras

Keras是一个支持Python的高级的神经网络编程接口，提供简洁而一致的API，能使一般的神经网络开发的工作量极大减少，使用Keras的API建立神经网络简单而快速。Keras的模型设计高度模块化，包含正则化模块、网络层模块、代价函数模块、优化器、初始化模块等等，在提供许多功能强大的模块同时也可以让用户自定义模块，这种便利性提高了Keras的用户体验，使其适用于各种工程或研究工作。同时Keras还支持很方便的CPU与GPU的切换。Keras后端目前支持Theano和Tensorflow两种框架。

2.3 GPGPU并行计算

从20世纪末到现在，计算机技术不断发展，越来越多的行业开始使用计算机进行特定的计算。虽然CPU等硬件性能在不断的提高，但是应用上对计算效率的要求提高的更快，传统的串行程序在性能上已经无法满足计算密集型任务的需求。而并行计算的出现使计算密集型任务有了更好的解决方案。近些年，人们从使用多核CPU，到使用多块多核CPU，到现在使用GPU（图形处理器）甚至多块GPU来做并行计算。原本GPU是专门用来渲染图形的，后来可以做通用计算的GPU出现，让GPU更加泛用，于是基于GPU的并行计算模型也成为了一个研究的新方向。

GPGPU（General-purpose computing on graphics processing units）的意思就是使用通用GPU来完成应用程序传统上在CPU上完成的计算[16][17][18]。由于GPU的核心数通常很多，即使是一个单一的GPU-CPU框架也可以提供比多块CPU更高的性能[19]。基本上，现代GPGPU管道会像传输图形一样在GPU和CPU之间传输要分析的数据，所以现在不需要显式把数据转换为图形的形式再用GPU做计算，不会影响计算的效率。涉及到矩阵和向量的问题可以容易地从CPU算法改为GPU算法，特别是涉及二至四维数组的问题，GPU对这些数据类型做了优化。

2001年，科学计算社区开始在GPU上做矩阵乘法的实验；2005年，LU因式分解算法成为第一个在GPU上运行比CPU快的常用科学计算算法[20]。在这些早期使用GPU做通用计算的工作中都要使用OpenGL或DirectX的图形接口定义的源语重新设计计算问题。后来Brook、Accelerator等通用编程API出现，才避免了这种繁琐的翻译[21][22]。接着又出现了Nvidia的CUDA框架，允许程序员忽略图形的概念，促进了GPGPU的发展。

2.3.1 CUDA

CUDA是由Nvidia发布的并行计算框架和编程接口，使开发人员能够用支持CUDA的GPU进行通用计算。CUDA框架原生支持C、C++和Fortran语言，支持OpenACC和OpenCL等计算库。CUDA与传统的GPGPU框架相比有如下优点：（1）代码可以从内存中的任意地址读取；（2）支持整数和位运算；（3）提供线程间的快速共享内存[14]等。CUDA的架构如图 2.2和图 2.3所示，线程层次分为多个维度，方便用户编程。多个线程（Thread）先按一定的形式分配到各个线程块（Block）上，然后多个线center4280535图 STYLEREF 1 \s 2. SEQ 图 * ARABIC \s 1 2 CUDA架构图图 STYLEREF 1 \s 2. SEQ 图 * ARABIC \s 1 2 CUDA架构图center12744450086741075647555图 STYLEREF 1 \s 2. SEQ 图 * ARABIC \s 1 3 CUDA Grid示意图图 STYLEREF 1 \s 2. SEQ 图 * ARABIC \s 1 3 CUDA Grid示意图center4779645程块组织为各个线程格（Grid）。

2.3.2 PyCUDA

由于CUDA原生不支持Python语言，本论文需要使用PyCUDA库。PyCUDA是由第三方开发的CUDA API在Python语言上的一个封装，使用户可以通过Python调用CUDA的接口。

3

算法设计

3.1 问题定义

本节将详细介绍本论文研究的问题、研究的数据集、预处理数据的方式和问题抽象。

3.1.1 问题描述

本论文研究的目标是，根据舆情数据来预测股票的涨跌情况。研究的数据集由欧洲一家企业提供。本论文实现的算法将通过一定天数的舆情数据和股票收盘价，训练出模型，之后用户就可以根据此模型和当天的舆情数据，来

预测下一天的股票涨跌。

3.1.2 数据格式

我们得到的每份数据分为两部分：

6902452598420图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 1 股票价格数据样例图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 1 股票价格数据样例center779145一部分是2014~2016年每日该股票的价格（见图 3.1），五列分别是日期、开盘价、最高价、最低价、收盘价。

另一部分是2014~2016年每日发生的与该股票相关的舆情（见图 3.2）。其中每天的每条舆情数据有四个属性(event_type_id, scope_id, polarity, count)，分别代表：事件类型ID、领域类型ID、极性（积极、中性或消极）、该事件在当天的发生次数。

6699252695575图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 2 股票相关舆情数据样例图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 2 股票相关舆情数据样例center47625

3.1.3 问题抽象

要解决问题首先要对实际问题进行抽象。首先对数据进行预处理，目的是把每条数据处理成从特征映射到下一天股票涨跌情况的格式，即（特征1，特征2，...）->（涨跌情况）。

对于价格数据部分，计算出每日的股票收盘价相对于前一天是上涨还是下跌，作为涨跌情况。

对于舆情数据部分，统计所有数据中出现过的所有不同的三元组(event_type_id, scope_id, polarity)，将每一组不同的三元组编号，给予一个从1开始的正整数ID，称作feature_id，作为一个舆情特征。

指　标

疑似剽窃文字表述

1. 基于模拟生物遗传与进化的过程，遗传算法使各类人工系统具有很好的自适应能力和优化能力。
2. 优化算法通常使用确定性的搜索方法，从搜索空间中的一个点到另一个点有确定的转移方法和转移关系，这种确定性往往也有可能使得搜索永远达不到最优点，因而也限制了传统优化算法的应用范围，遗传算法则不然。虽然这种随机的特性也会在种群中产生一些适应度不高的个体，但是随着进化过程的进行，新的群体中总会更多地产生出许多优秀的个体。算法

2. 基于CUDA的遗传算法和神经网络在股票趋势问题中的应用研究_第2部分　　总字数：**10023**

相似文献列表　　**文字复制比：**0%(0)　　**疑似剽窃观点（0）**

原文内容

再把每天每个舆情特征的发生次数记录下来作为该特征的权重。举个例子，比如说2014年1月1日，某股票对应公司的微博（scope_id）下出现大量用户的评论（event_type_id），评论内容为对该公司某产品投诉（polarity），这一条舆情数据就被我们抽象成一个特征，赋予一个特征ID。这样，把每个特征ID当作特征向量的每个维度，该特征的权重为向量分量的大小，就得到了每日舆情数据的特征向量。

设：m为样本总数（总天数），n为特征总数。

对于每日的数据：

y(j)=1, if第j+1天股票价格上涨或不变0, if第j+1天股票价格下跌

x(j)=(x0, x1, x2, …,xn)，其xi(j)为第j天feature_id = i的特征权重，设置x0恒为1。

称一组 (xj, y(j)) 为一个样本。

对于全体数据：

y= y(1)y(2)…y(m) X= x(1)x(2)…x(m)= x0(1)…xn(1)……x0(m)…xn(m)

需要提到的一点是，X和y是从总体Xtotal和ytotal（从古至今每一天的数据）抽样出的一组样本。

现实中存在一个函数f，满足y=f(X)，并且ytotal=f(Xtotal)。

14700253429000图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 3 算法思路图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 3 算法思路center619125如图 3.3，我们的目标是找到一个函数h，使得h与f尽量相近，使我们可以通过函数h来预测股票的趋势。

3.2 遗传算法

本节介绍使用遗传算法来解决问题的算法设计。包括对问题的建模、遗传算法中基因的定义、交叉和变异操作的算法和整个遗传算法的流程。

3.2.1 问题建模

设参数 θ=(θ0, θ1, θ2, …,θn) 为一个n维向量。

我们选定函数h的模型为 hθx=gθTx=11+ e-θTx。

其中gz=11+ e-z 为sigmoid函数，形状如图 3.4所示。当z < 0时，gz∈(0, 0.5)；当z > 0时，gz∈(0.5, 1)。

7937503975735图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 4 sigmoid函数部分图像图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 4 sigmoid函数部分图像center642013这个模型把特征向量x根据θ线性映射为一个实数，再将这个实数通过sigmoid函数映射到(0, 1)区间内。

由于x，y是随机变量，并且hθx∈(0, 1)，所以我们做出如下假设：在给定x和θ的条件下，y = 1的概率为hθx。

即：

Py=1x, θ)=hθx

Py=0x, θ)=1-hθx

即Pyx, θ)满足参数为hθx的伯努利分布。

将两个等式写在一起得到：

Pyx, θ)=(hθx)y(1-hθx)1-y

如果能求出θ，那么函数h就求出来了。我们已经有了样本 (X,y)，要根据(X,y)来反求出满足样本分布的参数θ，我们使用极大似然估计的方法。假设m个样本相互独立，设L为参数θ的似然度，则根据似然函数的定义，有：

Lθ=PyX,θ)

=j=1mPyjxj,θ

=j=1m(hθxj)yj(1-hθxj)1-yj

根据极大似然原理，最可能的θ的值应为使Lθ取到最大的值。那么我们把Lθ作为遗传算法中的适应度函数，随着算法的进行，算法最终会找出使Lθ足够大的θ，则问题就得到解决了。

3.2.2 基因定义

如果要使用遗传算法来解决问题，按照遗传算法的工作流程，需要在要求解的θ和遗传算法中个体的基因建立双射，也就是确定编码和解码的方式。常用的编码解码方式有很多种，在本论文中，由于θ∈Rn，即θ有n个分量，各分量均为实数，于是很自然的把θ编码为一个长度为n的实数数组就可以了。这种基因的定义方式有便于大空间搜索，并且结果的精度很高的优点。

3.2.3 遗传算法流程

遗传算法的流程如图 3.5所示。

首先随机生成一个种群，种群中包括随机生成的初始个体，每个个体有自己的基因序列，然后不断迭代产生新的种群，直到达到迭代次数上限或者种群中的某个个体的适应度足够高。个体适应度是基因的函数，也就是上文提到的似然函数Lθ。

每次迭代产生新种群的过程包括两个步骤：交叉、变异。

首先进行交叉过程，每次随机选出两个较优秀的原种群个体交叉产生新的个体，直到充满新种群。然后进行变异过程，遍历新种群内的所有个体，按一定概率对其中的基因做突变。

11461752737485图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 5 遗传算法流程图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 5 遗传算法流程center146685

3.2.4 交叉操作

遗传算法在繁衍新一代种群时，要不断根据上一代的两个个体（下称A和B）基因交叉产生一个新一代的个体。

right2948940算法 STYLEREF 1 \s 3. SEQ 算法 * ARABIC \s 1 1 个体选择算法0算法 STYLEREF 1 \s 3. SEQ 算法 * ARABIC \s 1 1 个体选择算法right1211580下面介绍我们选择A和B让其繁衍的方式。遗传算法的核心思路是优胜劣汰、适者生存，我们要在交叉操作中体现这个启发式的优化。我们在老种群中先完全随机地选出一些个

体（数量为tournament_size，是算法的一个参数），然后在这些个体中取适应度最高的那个作为A去繁衍，再用同样的方式选出B。如下是个体选择的算法代码描述：

然后我们将A和B的基因交叉，交叉方法是从头到尾遍历产生新个体的每一个基因，这个基因以一定的概率（概率为uniform_rate，是遗传算法的一个参数）从A中相应位center2356485算法 STYLEREF 1 \s 3. SEQ 算法 * ARABIC \s 1 2 交叉算法0算法 STYLEREF 1 \s 3. SEQ 算法 * ARABIC \s 1 2 交叉算法center352425置复制或从B中相应位置复制。如下是交叉的算法代码描述：

经过这一步骤，我们就得到未经变异的新一代种群的所有个体了。

3.2.5 变异操作

接下来要对新种群中的每个个体进行变异操作。与最常见的二进制编码的遗传算法不同，由于我们的基因使用实数编码，所以变异操作要复杂一些。

对每个个体的变异操作流程如下：

遍历该个体的每个基因，以一定的概率（概率为mutation_rate，是遗传算法的一个参数）决定这个基因是否需要变异。

如果需要变异，那么新的基因值 θi'=θi+γ·d·(1-r(1-tT)b) ，其中

θi为该位置原有的基因

γ为以50%概率随机产生的1或-1，用于控制新基因向上增长还是向下缩减

d是一个设置好的实数步长，是遗传算法的参数

t为当前种群的代次

T为算法的总迭代数

b是一个常量，用来控制算法的学习速度，是遗传算法的参数

r为[0, 1)的随机数。

公式中的(1-r(1-tT)b)因子根据参数b和迭代的进度，控制了每个基因变异的速度，算法开始时变异速度很快，随着迭代次数增加，变异速度会逐渐减小。

3.2.6 适应度计算

对于每个新产生的个体，我们都要计算它基因的适应度。根据上文提到的问题抽象和问题建模，每个个体的基因都是不同的参数向量θ=(θ0, θ1, θ2, …,θn)，而该基因的适应度就是似然度Lθ=PyX,θ)=j=1m(hθxj)yj(1-hθxj)1-yj。其中的y和X就是我们的样本，也就是训练数据。

3.2.7 其他优化

在自然界中，生物繁衍到下一代后，老一代并不会全部死亡，会有一些精英个体由于适应度很高、生命力强盛等原因保留了下来。为了进一步优化算法，我加入了精英保留这个启发式的策略来模拟自然界的这个现象。策略是将上一代作为精英的适应度最高的个体不经交叉、不经变异，直接保留到新一代。这样就避免了由简单的杂交破坏掉较好的基因组合，达到优秀基因组合逐渐累积起来的目的，同时也可以加快最优解的收敛速度。

具体算法是每次产生新一代时，在交叉过程之前，遍历选择上一代适应度最大的个体，复制进下一代的种群，同时将需要交叉产生的个体数目减一，变异操作时也跳过这个个体。

3.3 遗传算法的并行化

我们希望通过使用GPU大规模并行计算来提高遗传算法的运行效率。由于GPU的核心数量比CPU多得多，并行度就会高很多。我们的遗传算法中对每一个新产生的个体都要计算其适应度，在计算适应度时为了计算hθxj (1≤j≤m)，由于hθxj=11+ e-θTxj，我们要计算m个θTxj，而这刚好是XθT的m个分量。那么这个频繁进行的矩阵乘法就成为了我们优化的重点。

在矩阵乘法中，因为乘积的每个分量的计算都是独立的，之间互不相干，如果把乘积的每个分量放在一个单独的核上做计算，各个核没有互斥资源的竞争，不需要锁，而且运算所需要的资源都是一致的，只需要从内存拷贝一个副本到显存上，所以矩阵乘法天生具有容易并行的性质。

X是一个m*n的矩阵，θT是一个n*1的矩阵，乘积是一个m*1的矩阵。所以这样一个矩阵相乘的并行度就是m。现在家用CPU内核数量一般都小于8个，而GPU的内核数量普遍破千，如Nvidia GTX1080Ti的内核数就有3584个之多。如果m远大于8，那么使用GPU并行加速的优势就会很大。

并行化XθT的计算算法如下：

算法初始化

在显存中分配三块空间，一块用来存储X，一块用来存储θ，一块用来存储乘积，并记录下三块空间的地址

将矩阵X从内存拷贝到显存中

每次计算XθT

将向量θ从内存拷贝到显存中

根据m的大小设置线程块数量和线程数量

启动所有的计算线程，并行计算乘积的每个分量，结果存储在预先分配好的空间中

将乘积从显存拷贝到内存中

由于每次计算时矩阵X都是不变的，所以每次计算都只需要把θ从内存拷贝到显存，X只需要使用最初拷贝的那份就可以，这样可以节省大量的IO时间。

3.4 人工神经网络

本节介绍使用人工神经网络来解决问题的算法设计。包括对问题的建模、神经网络结构的设计、神经元的设计、损失函数的设计、训练方法和其他优化。

3.4.1 问题建模

在3.1.3小节中我们讨论了问题的抽象：现实中存在一个函数f，满足y=f(X)，并且ytotal=f(Xtotal)。我们的目标是从假设空间H中找到一个函数h∈H，使得h与f尽量相近。我们将建立一个人工神经网络，并使用样本集训练出网络的各个参数，该网络就是函数h，将特征向量x映射到预测值h(x)。

3.4.2 网络结构

算法采用全连接的人工神经网络。如图 3.6，网络包括一个输入层，多个隐层（隐层数量为算法的参数）和一个输出层。输入层包含n个神经元，每个神经元输出样本xj的一个分量。每个隐层包含一定量的神经元（单隐层神经元个数为算法的参数）。输出层包含1个神经元，输出对当前输入数据的预测结果。各层之间的神经元全连接。

4603753930650图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 6 神经网络结构示意图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 6 神经网络结构示意center260985

3.4.3 神经元设计

在每个神经元内设置一个参数θ=(θ0, θ1, θ2, …,θn)。如图 3.7，神经元的输入是上一层的所有输出，设为向量x'，然后把x'和θ相乘得到一个实数，再把这个数做一个sigmoid函数映射σz=11+ e-z，映射到(0, 1)区间内，将这个值输出。

可以看出，我们人工神经网络算法中的每个神经元的设计与3.2.1小节讨论的遗传算法问题模型十分相似，上文讨论的遗传算法模型实际上可以看做是一个无隐层、只有输入层和输出层的人工神经网络。

对于每一层而言，设x'和y'分别是该层的输入输出向量，Θ是该层的参数矩阵（将每个神经元的参数θ拼在一起），A是矩阵为Θ的线性变换，则y'=σ(Θx'+b)=σ(Ax'+b)。可以看出神经网络的每一层对输入向量先进行一个线性变换（升降维、放大缩小、旋转），然后平移，然后再进行一个非线性变化（弯曲），将输入空间投向另一个空间。多层神经网络组合，可以产生十分复杂的函数模型，使假设空间H很大。

3.4.4 5651503451225图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 7 神经元示意图 STYLEREF 1 \s 3. SEQ 图 * ARABIC \s 1 7 神经元示意center190500损失函数

我们的神经网络的训练目的是最小化每个神经元的损失函数lossθ。在3.2.1小节中，我们希望最大化似然函数Lθ=PyX,θ)=j=1m(hθxj)yj(1-hθxj)1-yj。同样的道理，在这里对于输出神经元，我们希望最大化似然函数Lθ=Pyx',θ)=j=1m(y'j)yj(1-y'j)1-yj。将Lθ取对数后得到

lθ=log Lθ=j=1m[yjlogy'j+1-yjlog1-y'j]

将损失函数设置为对数似然函数的相反数：lossθ=-lθ。那么随着算法的进行，各神经元的损失函数值会越来越小，最终就会得到我们想要的函数h。

在信息论中，常用交叉熵来衡量两个概率分布p、q之间的相似性。交叉熵越小，说明p和q越相似。从这个角度来考虑的话，根据定义，样本分布y和预测分布y'的交叉熵

CrossEntropyy, y'=-1mj=1m[yjlogy'j+1-yjlog1-y'j]

显然，我们的损失函数lossθ刚好正比于y和y'的交叉熵CrossEntropyy, y'。

3.4.5 后向传播

我们使用后向传播的方法来训练出每个神经元的参数θ。

首先初始化所有θ为随机值。每次迭代中，首先算出输出层的损失函数值lossθ，然后更新该神经元的参数θ=θ+η*Δlossθ，其中η为设置好的参数，用于控制学习速率，Δlossθ为loss对于θ的梯度。

对于隐层，我们定义函数

lossкθ(k)=y'*klossk+1*θi(k+1)

其中k为当前层数，θi(k+1)为k+1层与k层该神经元连接的那个分量。接着从后向前迭代每一层，按照θ=θ+η*Δlosskθ(k)更新每个神经元的参数。直到达到最大的迭代次数。

3.4.6 L2正则化

由于人工神经网络需要大量的数据来训练，而我们的训练数据不够多，比较容易导致过拟合。为了防止这一现象，我们在损失函数上添加了L2正则项：

loss'θ=lossθ+λ2ni=1nθi2

其中λ为设置好的控制正则化力度的参数。可以看出，L2正则项其实是一个惩罚项，θ的各分量越大，惩罚越严重。这样可以使模型变得更简单，而简单的模型一般更有范化的能力。

算法思路很朴素：通过最小化损失函数来拟合训练集，又通过加入正则项来防止过分拟合训练集。

3.4.7 Dropout优化

为了进一步的防止过拟合，算法使用了dropout的优化策略，也就是说，在算法的每次迭代过程中，都暂时丢弃一部分（比例为算法的参数dropout_rate）随机的神经元。也就相当于我们训练出了很多个部分结构的神经网络，每一个经过dropout剩下的神经网络的部分（以下简称部分）都可以给出一种分类结果，分类结果有些正确，有些错误。不断让网络进行训练，大多数部分的分类结果都是比较优秀的，少数的错误分类的部分对整个网络的分类结果影响不大，同时又因为每次迭代减少了部分神经元，也就是减少了参数的维度，而减少了算法过拟合的风险。

3.5 人工神经网络的并行化

与遗传算法一样，人工神经网络中也需要进行大量的矩阵运算，我们使用theano框架提供的功能，将数据转为32精度的浮点数，然后把矩阵运算等操作挪到GPU中做，来优化算法的执行效率。

4 算法实现

4.1 开发环境

编程语言：Python 2.7

操作系统：Windows 10 64bit

包管理工具：conda、pip

IDE：Spyder 3

CPU：Intel Core i5 6500

GPU：NVIDIA GeForce GTX 1060

依赖库：NVIDIA CUDA Toolkit 8.0、NumPy 1.11.3、Keras 1.0.7、Theano 0.9.0、PyCuda

4.2 系统架构及实现

3752855107305图 STYLEREF 1 \s 4. SEQ 图 * ARABIC \s 1 1 实验系统架构图0图 STYLEREF 1 \s 4. SEQ 图 * ARABIC \s 1 1 实验系统架构图38290599250500实验系统整体的架构如图 4.1所示，我们依照之前的设计，把需要实现的程序分为了三个文件：ga.py用于实验基于CPU的遗传算法的效果和性能；ga_cuda.py用于实验基于GPU的遗传算法的效果和性能；dl.py用于实验人工神经网络算法的效果和性能。

我们使用NumPy库来支持矩阵和向量的计算；在遗传算法部分使用PyCuda来编译c++的CUDA并行计算代码并插入到我们的Python程序中；在人工神经网络部分利用Keras和Theano框架方便地建立神经网络、训练并验证数据集，也可以通过改变配置，使程序方便地切换到GPU上运行。

4.2.1 数据解析和预处理模块

该模块对数据进行解析和预处理。本论文研究的数据集是以csv格式提供的，所以要使用python内部的csv库

来解析数据文件。首先解析价格数据，逐行读取文件，从第二行开始，读取该行的日期和收盘价，然后判断该日期相较于前一天价格是上涨还是下跌来得出该日趋势，再以日期为键、趋势为值存入字典trends里面。接着解析舆情数据，逐行读取文件，对每一行读取该行的日期、类型ID、领域ID、极性和次数，把类型ID、领域ID、极性组成三元组，若还未给此三元组赋予特征ID就生成一个特征ID并把关联信息存入一个字典中，然后在二维数组sentiments中保存该日该特征的权重为次数。

数据解析完之后开始建立矩阵和向量。初始化矩阵X和向量y为空的numpy对象，然后按日期对trends和sentiments排序，对每个日期把趋势和价格写入X和y中。

4.2.2 种群模块

该模块把种群概念抽象为类class Population。支持如下方法：

__init__：用于建立新种群，如果need_init参数为真，则不断添加随机生成的个体来填满这个种群。

get_individual：按序号获取种群中的某个个体。

get_individuals：获取种群中的所有个体。

append_individual：在种群的末尾添加个体。

size：获取种群的大小。

get_fittest：获取种群中适应度最高的个体。

4.2.3 个体模块

该模块把个体概念抽象为类class Individual。支持如下方法：

__init__：用于初始化新个体，生成空的基因序列，将适应度置为0。

generate_individual：使用均一分布的±10000间的随机值填充个体的基因序列。

set_default_genes_len：静态方法，用于设置默认的基因长度。

get_gene：获取特定位置的单个基因。

set_gene：设置特定位置的单个基因。

get_genes：获取整条基因。

get_fitness：获取个体的适应度。首先检查该个体的适应度之前有没有计算过，没有的话进行计算并把结果保存下来。

4.2.4 演化模块

该模块class Alogorithm实现了各种演化过程需要的所有算法，包括交叉算法、变异算法等。支持如下静态方法：

evolve_population：根据传入的种群，进行一次完整的繁衍过程。先不断的调用tournament_selection来选出两个个体进行交叉产生新个体填入新种群中，然后遍历新种群的每个个体进行变异操作，返回新产生的种群。

tournament_selection：用于选出要交叉的个体。传入种群，随机选出该种群中一定量的个体，返回这些个体中适应度最高的个体。

crossover：用于交叉两个个体。首先初始化一个新个体，然后遍历生成新个体的基因序列，每次生成一个[0,1)的随机数，判断其是否小于设置好的交叉率，如果小于则该基因从个体1复制，否则从个体2复制。

non_uniform_degree：用于计算突变的程度。

mutate：用于让个体进行变异。遍历传入的个体基因序列，每次生成一个[0,1)的随机数，判断其是否小于设置好的突变率，如果小于则计算突变的程度并对该单个基因进行变异。

4.2.5 基于CPU的适应度计算模块

该模块class FitnessCalc实现了基于CPU的适应度计算的算法。支持如下静态方法：

h：用于计算hθx。输入的x既可以是全体样本的矩阵，也可以是单个样本的向量。

l：用于计算似然函数Lθ。

get_fitness：用于计算个体的适应度。输入一个个体，调用方法l来计算似然度作为该个体的适应度。

4.2.6 基于GPU的适应度计算模块

该模块class FitnessCalc实现了基于GPU的适应度计算的算法。

-120653552190算法 STYLEREF 1 \s 4. SEQ 算法 * ARABIC \s 1 1 在GPU单核上运行的算法算法 STYLER

EF 1 \s 4. SEQ 算法 * ARABIC \s 1 1 在GPU单核上运行的算法right1251585如算法 4.1，首先该模块用c++实现了在单线程上计算矩阵乘法中的一个元素并做sigmoid映射的算法：先根据当前线程块号和线程号计算出要计算的元素在乘积矩阵中的偏移，然后读取在显存中的乘数矩阵和乘数向量，计算相应位置的元素值，顺便做sigmoid映射，之后填在显存相应的位置中。

在模块初始化时，会调用PyCuda来把这个c++程序编译并作为该模块的python函数h加载到算法进程中。模块还支持如下静态方法：

I：用于计算似然函数Lθ。与基于CPU的适应度计算模块中的方法I不同，这里先判断是否已经在显存中分配好了空间，若没有则先分配空间并把指针保存起来，然后如果显存中不存在矩阵X则把X复制到显存相应空间中。再复制θ到显存相应空间中，建立大量GPU线程运行函数h计算出结果，再把结果从显存拷贝回内存中。

get_fitness：用于计算个体的适应度。输入一个个体，调用方法I来计算似然度作为该个体的适应度。

4.2.7 遗传算法训练模块

该模块调用种群、个体、演化、适应度计算等模块，串联出整个遗传算法的流程，在样本子集上做训练，最终得到训练出的参数θ。同时此模块使用计时器记录了训练过程所使用的时间。

模块首先设置好总迭代数、种群大小、基因长度等参数，然后把训练集交付给适应度计算模块，再使用种群模块初始化生成一个随机的种群，不断迭代。迭代时记录每次繁衍新种群需要的时间。最终记录下最末代种群中最优个体的基因序列作为返回值。

4.2.8 人工神经网络训练模块

该模块建立并设置整个人工神经网络，并使用训练集将网络训练完成。

此模块利用Keras和Theano框架实现。首先实例化一个Keras的Sequential对象，然后构建网络，添加输入层、隐层和输出层，添加dropout层，设置神经元使用sigmoid作为激活函数，使用L2正则化的方法，使用后向传播的训练方法，使用交叉熵作为损失函数，使用二分类来评估结果。

---

3. 基于CUDA的遗传算法和神经网络在股票趋势问题中的应用研究_第3部分 总字数：**5135**

相似文献列表 **文字复制比：**2.5%(129) **疑似剽窃观点（0）**

1	德国DAX指数 -《网络（http://www.zaxue.com）》- 2008	2.5%（126） 是否引证：否
2	德国DAX指数创历史新高启示 肖玉航 -《中华工商时报》- 2013-10-21	2.4%（122） 是否引证：否
3	德国GER30指数是什么？|理财讲堂-地宝网 -《网络（http://www.tiboo.cn/）》- 2016	2.4%（122） 是否引证：否
4	大众成德国Dax指数上最赚钱公司 收益佳 -《网络（http://www.caam.org.）》- 2013-03-18	2.0%（102） 是否引证：否
5	义隆金融德国DAX指数交易规则|谈股论市-地宝网 -《网络（http://www.tiboo.cn/）》- 2016	2.0%（102） 是否引证：否
6	各国指数计算方法_金钱与爱情 -《网络（http://blog.sina.com）》- 2014	2.0%（102） 是否引证：否
7	鼎金投资：交易德国DAX指数期货如何选取正规平台？_鼎金投资-外盘开户-外盘融资 -《网络（http://blog.sina.com）》- 2016	1.9%（99） 是否引证：否
8	鼎金投资：德指DAX的交易代码是什么_鼎金投资-外盘开户-外盘融资 -《网络（http://blog.sina.com）》- 2016	1.9%（99） 是否引证：否
9	德国指数开户有什么条件|理财讲堂-地宝网 -《网络（http://www.tiboo.cn/）》- 2016	1.9%（99） 是否引证：否
10	德国DAX指数开户有什么要求？_股指期货(gzqh)股吧 -《网络（http://guba.eastmone）》- 2016	1.9%（99） 是否引证：否
11	德国GER30指数的代码是什么？|理财讲堂-地宝网	1.9%（99）

	-《网络（http://www.tiboo.cn/）》- 2016	是否引证：否
12	法兰克福DAX指数的正规平台有哪些？|理财讲堂-地宝网 -《网络（http://www.tiboo.cn/）》- 2016	1.9%（99） 是否引证：否
13	论人民币汇率形成机制改革 袁利勇(导师：李晓) -《吉林大学硕士论文》- 2006-04-25	1.8%（93） 是否引证：否
14	交易全球8大股票指数 尽在兴业投资-E起理财-理财论坛-财经世界-和讯论坛 -《网络（http://bbs.hexun.com）》- 2015	1.8%（93） 是否引证：否
15	金融术语- EMBA百科 - EMBA百科全书|经管百科|智库百科 -《网络（http://wiki.chinaemb）》- 2010	1.8%（91） 是否引证：否
16	金融学- 科技中国——欢迎光临全球最大的互联网博物馆 -全球第一互联网博物馆，你我的知识加油站 -《网络（http://wiki.chinalab）》- 2011	1.8%（91） 是否引证：否
17	[转载]富时A50、H股指数、标普500、德国DAX—交易外盘必知的股票指数期货（_独孤志筠 -《网络（http://blog.sina.com）》- 2016	1.5%（79） 是否引证：否
18	德国指数的交易时间是什么时候？|理财讲堂-地宝网 -《网络（http://www.tiboo.cn/）》- 2016	1.5%（79） 是否引证：否
19	GER30指数是什么意思？|理财讲堂-地宝网 -《网络（http://www.tiboo.cn/）》- 2016	1.5%（79） 是否引证：否
20	法兰克福股市DAX指数-商业评论百科-专业的管理百科 -《网络（http://wiki.ebusines）》- 2011	1.3%（69） 是否引证：否
21	法兰克福DAX指数-名词解释-龙头股票网 -《网络（http://www.lootou.cn）》- 2009	1.3%（69） 是否引证：否
22	资金安全 Fxsol官网 Fxsol中国官网 环球金汇网 Fxsol外汇开户 Fxsol外汇公司 Fxsol手机移动MT4平台 MT4下载 原油交易 外汇投资 黄金投资 MT4平台 外汇交易平台 外汇 -《网络（http://www.hqjhw.com）》- 2011	1.3%（69） 是否引证：否
23	广州国际期货 全球股指 恒生指数开户_希西家投资理财资讯 -《网络（http://blog.sina.com）》- 2013	1.3%（69） 是否引证：否
24	国际证券指数精选—— 安吉丽娜——东方财富网博客 -《网络（http://blog.eastmone）》- 2012	1.3%（67） 是否引证：否
25	德国公司治理立法的最新进展及其借鉴_第3页_中国论文下载中心_经济法论文 -《网络（http://www.studa.net）》- 2009	0.7%（37） 是否引证：否

原文内容

然后调用Keras模型的fit方法，输入训练集，设置迭代次数，自动训练网络。训练时使用计时器计时，保存所用的时间。

center655320left2341245配置 STYLEREF 1 \s 4. SEQ 配置 * ARABIC \s 1 1 theanorc文件0配置 STYLEREF 1 \s 4. SEQ 配置 * ARABIC \s 1 1 theanorc文件Keras底层采用Theano框架实现，要使用GPU优化性能只需要在配置文件中做如配置4.1的设置。

4.2.9 测试模块

该模块用于测试算法产生的模型的正确率。输入测试集和训练好的模型（对于遗传算法是θ，对于人工神经网络是网络对象），使用模型对测试集的每条数据做预测，再与正确结果作比较，得到预测正确的天数，最后输出正确率（正确率=预测正确的天数总天数×100%）。

5 实验验证

5.1 验证方法

为了保证结果的可靠性，本论文采用了两个数据集进行实验验证：

1． 2014年8月11日至2016年8月9日间德国DAX股票指数（下称DAX）的价格和相关的舆情数据，共包含505

天的价格和51654条舆情。DAX（德语： Deutscher Aktienindex）是德国重要的股票指数，是由德意志交易所集团（Deutsche Börse Group）推出的一个蓝筹股指数。该指数中包含有30家主要的德国公司。DAX指数是欧洲的重要证券指数，也是世界证券市场中的重要指数之一。

2． 2014年6月2日至2016年6月2日间国际黄金价格（下称GLD）和相关的舆情数据，共包含518天的价格和38770条舆情。

对于每个数据集，按照3:1:1的比例划分为训练集、验证集和测试集。使用训练集进行模型训练，使用验证集对算法进行调参，使用测试集进行正确率的测试。单独划分验证集是为了确保测试的可靠性，防止为了拟合测试集而调参。

在对比基于CPU的（无CUDA优化）遗传算法训练性能和基于GPU的（有CUDA优化）遗传算法训练性能，以及基于CPU的（无CUDA优化）人工神经网络训练性能和基于GPU的（有CUDA优化）人工神经网络训练性能时，采用DAX数据集来分析。

5.2 算法评估

5.2.1 遗传算法的正确率

表格 5.1 遗传算法参数设置

可调参数值

总迭代数（产生种群数） 1000

种群内个体数 100

交叉率uniform_rate 0.85

变异率mutation_rate 0.05

个体选择数tournament_size 30

变异速度b 3

变异步长d 10000

按表格 5.1设置可调参数，分别使用两个数据集训练模型并测试。

1905001671955图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 1 遗传算法DAX测试结果图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 1 遗传算法DAX测试结果left73596500如图 5.1，算法在DAX数据集上测试了104个数据点，其中预测正确66个，正确率为63.4615%。

1828803011805图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 2 遗传算法GLD测试结果图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 2 遗传算法GLD测试结果left209994500如图 5.2，算法在GLD数据集上测试了117个数据点，其中预测正确79个，正确率为67.5214%。

5.2.2 人工神经网络的正确率

按表格 5.2设置可调参数，分别使用两个数据集训练模型并测试。

表格 5.2 人工神经网络参数设置

可调参数值

总迭代数 10000

η 0.01

隐层数 1

隐层神经元数 8

center1101090图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 3 人工神经网络DAX测试结果图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 3 人工神经网络DAX测试结果center792480如图 5.3，算法在DAX数据集上测试了104个数据点，其中预测正确59个，正确率为56.7308%。

如图 5.4，算法在GLD数据集上测试了117个数据点，其中预测正确77个，正确率为65.8120%。

5.2.3 center506730图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 4 人工神经网络GLD测试结果图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 4 人工神经网络GLD测试结果center190500遗传算法和人工神经网络正确率的对比目录

两种算法在两个测试集上的正确率对比见表格 5.3。

表格 5.3 算法正确率

DAX GLD

遗传算法 63.4615% 67.5214%

人工神经网络 56.7308% 65.8120%

可以看出，遗传算法的表现要比人工神经网络更加优秀。

687705363093068008544119 80图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 5 人工神经网络对GLD训练集的预测正确率图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 5 人工神经网络对GLD训练集的预测正确率7029453265170图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 6 人工神经网络对DAX训练集的预测正确率图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 6 人工神经网络对DAX训练集的预测正确率6877052520315不难理解，从直觉上考虑，舆情和股票价格应当是线性相关的。比如说在微博上对某公司的负面评价越多，该公司的股票价格越可能下跌，并且评价数和下跌概率直觉上应该呈线性。遗传算法的问题模型就使用了一个简单的线性模型θTx，sigmoid函数只是把结果映射到(0,1)区间内，不影响模型的线性。而人工神经网络的问题模型很复杂，上文提到多层多神经元的网络可以表示相当复杂的函数，而复杂的函数模型就容易过分拟合训练集。如图 5.5和图 5.6，使用人工神经网络训练时，在最后的迭代中，对训练集的预测正确率高达79.33%和73.67%，然而在测试集上正确率却只有56.7308%和65.8120%，说明此模型的确过拟合了。

5.2.4 遗传算法CPU与GPU训练性能的对比

DAX训练集共包括300条数据。分别使用基于CPU和基于GPU的遗传算法训练该训练集，并记录花费的时间。结果如图 5.7和图 5.8，基于CPU的算法花费了86.9秒，基于GPU的算法花费了73.4秒2152651068705200025135 8265图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 7 CPU遗传算法训练DAX数据的耗时0图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 7 CPU遗传算法训练DAX数据的耗时。通过使用CUDA的优化，在训练集大小为300时，遗27984451365885图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 8 GPU遗传算法训练DAX数据的耗时0图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 8 GPU遗传算法训练DAX数据的耗时28136851068705传算法的训练性能提高了18.4%。

5.2.5 人工神经网络CPU与GPU训练性能的对比

2762251367790图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 9 CPU神经网络训练DAX数据的耗时0图 STYL

EREF 1 \s 5. SEQ 图 * ARABIC \s 1 9 CPU神经网络训练DAX数据的耗时27984451352550图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 10 GPU神经网络训练DAX数据的耗时0图 STYLEREF 1 \s 5. SEQ 图 * ARABIC \s 1 10 GPU神经网络训练DAX数据的耗时28365451070610337185105537O分别使用基于CPU和基于GPU的人工神经网络训练DAX训练集，并记录花费的时间。结果如和，基于CPU的算法花费了6.8秒，基于GPU的算法花费了13.4秒。我们意外地发现使用GPU比使用CPU还多花了接近一倍的时间。

在使用CUDA优化算法性能时，theano框架把矩阵运算等计算挪到了GPU上做，以矩阵乘法为例，乘法的并行度是min{GPU核心数，m*1}。实验环境中虽然GPU的核心数是1280个，而训练集却只有300条数据，乘法的并行度就只有300。由于训练集小，并行度不够大，并行计算节约的时间还没有内存显存间拷贝数据带来的额外开销多，所以造成使用GPU比CPU还要慢的情况。

我们使用复制的手段扩大训练集，发现当训练集大小为1700时，基于CPU和GPU的算法耗时十分接近，均为23秒左右。当训练集大小为2400时，基于CPU的算法训练耗时27.4秒，基于GPU的算法训练耗时25.7秒。

结论

本论文成功地对使用舆情数据进行股票趋势预测问题的抽象和建模，设计并实现了遗传算法和人工神经网络这两种算法，成功地使用历史数据将模型训练出来，得到了能够根据当日舆情预测下日股票涨跌的遗传算法和人工神经网络。本论文还进行了遗传算法和人工神经网络在此问题上的优劣对比，最后成功使用CUDA对这两种算法进行并行化的优化，并对比了优化前后的性能，为基于CUDA的遗传算法和神经网络在股票趋势问题中的应用提供了解决方案。

相较于人工神经网络，遗传算法更好地解决了通过舆情数据来预测股票趋势的问题。对于遗传算法，我们成功地使用CUDA优化了它的训练性能，提高了训练速度；对于人工神经网络，我们了解了在本课题的实验环境下，300条数据的训练集并不能发挥CUDA并行计算的优势，但如果扩大训练集到1700条数据以上，使用CUDA优化的算法就会比只使用CPU的算法性能高、速度快。

致谢

在这毕业设计论文即将完成的时刻，我希望给对我的毕业设计提供过帮助的人和单位以诚挚的感谢。

首先要感谢的是我的导师任健老师。任健老师是一个充满活力、热情开朗、思维敏捷、极具人格魅力的人，虽然年轻，但工作认真负责一丝不苟。从我毕业设计的选题、题目分析，到开题、中期，再到最终的答辩，在每个环节上，一直认认真真地指导我毕业设计的工作，对我做的不好的地方提出改正的指导和意见，从没有不耐烦；在我思路阻塞的时候总能三两句话就拨开了我思路上的迷雾，使我的毕业设计顺利地进行下去。任健老师身上的学者气度也常常打动我，使我也按照学者的标准来要求自己。“高山仰止，景行行止。虽不能至，然心向往之”，可以说，能选择任健老师作为我毕业设计的导师是我的幸运。

【感谢某单位提供数据】

然后我要感谢同系的硕士研究生胡京徽学长。胡学长是计算机系的老学长，在我毕业设计中涉及到的许多具体的细节技术问题上指导了我。小问题像如何选择框架、配置环境的技巧，大问题像某种算法的设计为什么行不通，胡学长都能给出自己的见解，作为我解决问题的重要参考。

接着我要感谢答辩组的所有老师，在我毕业设计开题答辩、中期答辩和即将到来的最终答辩里一针见血地指出我的问题所在，让我能尽早改正毕业设计中的错误，不在歧途上越走越远。

我还要感谢我的前室友王逸翔、室友丁蔚然、室友刘艺华、室友刘一等朋友，在我因为遇到困难而灰心丧气时给我鼓励、给我力量。在与他们平时的闲聊中，偶尔还能激发出奇思妙想，应用在我的毕设中。

最后我要感谢我的父母，感谢他们对我的教育，感谢他们给我提供了良好的资源环境，让我无后顾之忧地求学。如果没有他们，我现在不可能在这里撰写这篇毕业设计论文。

再次感谢以上提到的和由于篇幅所限未能提到的所有对我进行过帮助的人们，谢谢你们！

附录A

本实验所实现的算法均可在https://github.com/hacker94/BiShe中找到。参考文献

[1] Majhi R, Panda G, Sahoo G, et al. Stock market prediction of S&P 500 and DJIA using Bacterial Foraging Optimization Technique[C]// Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. IEEE Xplore, 2007:2569-2575.

[2] Tan T Z, Quek C, Ng G S. Brain-inspired genetic complementary learning for stock market prediction[C]// Evolutionary Computation, 2005. The 2005 IEEE Congress on. IEEE, 2005:2653-2660 Vol. 3.

[3] Noll T A. Chaos and Order in the Capital Markets: A New View of Cycles, Prices, and Market Volatility. [J]. Journal of Finance, 1993, 48(5):2041.

[4] Taylor S J. Modelling Financial Time Series(2nd Edition)[M]. WORLD SCIENTIFIC, 2007.

[5] Petr Pospichal J J. GPU-based Acceleration of the Genetic Algorithm[J]. Gecco Competition, 2009.

[6] Han Y, Xiu L, Wang Z, et al. Artificial neural networks controlled fast valving in a power generation plant [J]. IEEE Transactions on Neural Networks, 1997, 8(2):373.

[7] Clements M P, Hendry D F. Forecasting Economic Processes--A Reply.[J]. 1998.

[8] Kara Y, Boyacioglu M A, Ömer Kaan Baykan. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange[M]. Pergamon Press, Inc. 2011.

[9] Nayak S C, Misra B B, Behera H S. Index prediction with neuro-genetic hybrid network: A comparative analysis of performance[C]// International Conference on Computing, Communication and Applications. IEEE, 2012:1-6.

[10] Wen Q, Yang Z, Song Y, et al. Automatic stock decision support system based on box theory and SVM algorithm[J]. Expert Systems with Applications, 2010, 37(2):1015-1022.

[11] 吉根林, 邢乃宁, 孙志挥. 应用人机交互的遗传算法挖掘股票投资风险规则[J]. 小型微型计算机系统, 2002, 23(12):1492-1495.

[12] 卢琇泽, 叶德谦, 南敏. 基于遗传算法和神经网络的股票价格预测[J]. 电脑开发与应用, 2010, 23(2):61-62.

[13] 张炜. 基于遗传算法的属性约简方法在股票预测中的应用研究[D]. 湖南大学, 2013.

[14] Mitchell M. An introduction to genetic algorithms[M]. MIT Press, 1996.

[15] Team T D, Alrfou R, Alain G, et al. Theano: A Python framework for fast computation of mathematical expressions[J]. 2016.

[16] Fung J, Tang F, Mann S. Mediated Reality Using Computer Graphics Hardware for Computer Vision [C]// International Symposium on Wearable Computers. IEEE, 2002:83-89.

[17] Aimone C, Fung J, Mann S. An EyeTap video-based featureless projective motion estimation assisted by gyroscopic tracking for wearable computer mediated reality[J]. Personal and Ubiquitous Computing, 2003, 7(5):236-248.

[18] Fung J, Mann S. Computer vision signal processing on graphics processing units[C]// IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. IEEE, 2004:V-93-6 vol.5.

[19] Mittal S, Vetter J S. A Survey of CPU-GPU Heterogeneous Computing Techniques[M]. ACM, 2015.

[20] Du P, Weber R, Luszczek P, et al. From CUDA to OpenCL: Towards a performance-portable solution for multi-platform GPU programming ☆, ☆☆[J]. Parallel Computing, 2012, 38(8):391-407.

[21] Tarditi D, Puri S, Oglesby J. Accelerator:using data parallelism to program GPUs for general-purpose uses[C]// ACM, 2006:325-335.

[22] Che S, Boyer M, Meng J, et al. A performance study of general-purpose applications on graphics processors using CUDA[J]. Journal of Parallel & Distributed Computing, 2008, 68(10):1370-1380.

[23] Zhao X, Gao X S, Hu Z C. Evolutionary programming based on non-uniform mutation[J]. Applied Mathematics & Computation, 2007, 192(1):1-11.

[24] 田小梅, 龚静. 实数编码遗传算法的评述[J]. 湖南生态科学学报, 2005, 11(1):25-31.

[25] Rocke D M. Genetic Algorithms + Data Structures = Evolution Programs, by Z. Michalewicz[M]// Genetic algorithms + data structures = evolution programs. Springer-Verlag, 1996:347-348.

[26] Harman M, Mansouri S A, Zhang Y. Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications[J]. 2009, 2003.

[27] 李宏毅. Deep Learning Tutorial[EB/OL]. http://people.tamu.edu/~gengxbtamu/pdf/Deep_Learning_Tutorial_Hungyi_Lee.pdf, 2015

指　标

疑似剽窃文字表述

1. 该指数中包含有30家主要的德国公司。DAX指数是欧洲的重要证券指数，也是世界证券市场中的重要指数之一。