基于android平台的物流配送系统

姓名：鲍立

指导老师：王东教授

学院：信息科学与工程学院

**摘要**

时代飞速的发展，随之而来的是人们生活方式变得日益便捷。现在，不管是衣食住行等各个方面，人们的需求都能随时随地、十分方便地得到满足。不论是网购、日常点外卖、还是实际购物中，各类商场和商店都提供了送货上门服务。随着配送业务的迅速发展以及需求量的逐渐加大，如何帮助配送人员提高配送的效率便成为了一个很重要的话题。

因而，笔者认为可以通过设计一个物流配送系统来提高配送人员在配送物品的效率，能极大地缩短配送时间，同时更能满足用户的需求。本课题主要完成了以下几个方面的工作：

1. 基于android平台以及百度地图开放平台等，能够提供定位、路线导航等基本功能，最终开发出了物流配送的app；
2. 对实际交通数据获得拥堵状态的评价方法。首先使用了数据挖掘过程中比较高效的聚类方法，针对一般的K-means等聚类方法对初始簇中心严重依赖、聚类结果较不稳定的问题，提出用ROCK算法来进行聚类，最终依据聚类结果得出交通拥堵状态的划分数值。拥堵状态是交通状态分析的重点，通过聚类方法从大量交通数据中处理得到该场景不同状态对应的划分数值，显得更加灵活和可信。
3. 对短时交通拥堵状态进行预测的过程当中，针对常见BP神经网络模型直接预测交通拥堵状态时存在有些情况预测结果不够准确的问题，提出了在神经网络的预测阶段使用BP神经网络结合统计学理论进行联合预测得到最终预测模型的短时交通拥堵状态预测方法。直接通过神经网络模型的预测正确率为95.1%，而在预测阶段结合统计学方法联合校验得到最终预测模型后进行预测的正确率为97.5%。该方法对交通拥堵状态的预测的正确率经试验结果对比证明有了一定的提升。

**关键词：城市道路；交通拥堵状态评价；短时交通拥堵预测；神经网络；统计方法**

**Abstract**

**目录**

摘要..................................................................................................

Abstract............................................................................................

第1章 绪论...................................................................................

1.1选题背景........................................................................

1.2研究的目的和意义........................................................

1.3国内外研究现状............................................................

1.4研究内容和论文结构安排............................................

第2章 关键技术研究....................................................................

2.1 Android系统...................................................................

2.2 Android开发平台简介................................. ..................

2.3百度地图开放平台.........................................................

2.4聚类方法.........................................................................

2.5 神经网络预测及优化.....................................................

第3章 系统需求分析.......................................................................

3.1系统业务需求.................................................................

3.2系统功能需求.................................................................

第4章 系统详细设计及实现...........................................................

第5章 总结与展望...........................................................................

参考文献............................................................................................

致谢....................................................................................................

1. **绪论**

**第1章 选题背景**

* 1. **研究的目的和意义**
  2. **国内外研究现状**
  3. **研究内容和论文结构安排**

1. **关键技术研究**

**2.1 Android系统**

**2.1.1 Android系统平台的架构**

Android系统总的来说可以分成四层架构：Linux内核层、系统运行库层、应用框架层和应用层。

1. Linux内核层

Android系统是基于Linux内核的，也就是说，Android系统的底层基于Linux系统之上。这一层为Android设备的各种硬件提供了底层的驱动，如显示驱动、音频驱动、照相机驱动、蓝牙驱动、Wi-Fi驱动、电源管理等。

但同时Android对于Linux进行了改动，比如它不包含glibc，原先用于一些便携的移动设备并没有采用glibc作为c库，而是goolge自己开发的Bionic Libc来作为代替品等，同时还增强显示驱动、键盘驱动、Flash内存驱动、照相机驱动、音频驱动、蓝牙驱动、Wi-Fi驱动、Binder IPC驱动、Power Management（电源管理），包括硬件时钟，内存分配和共享，低内存管理，kernel调试，日志设备，android IPC机制，电源管理等。

1. 系统运行库层

这一层通过一些C/C++库来为Android系统提供了主要的特性支持。如SQLite库提供了数据库的支持，OpenGL|ES库提供了3D绘图的支持，Webkit库提供了浏览器内核的支持等。

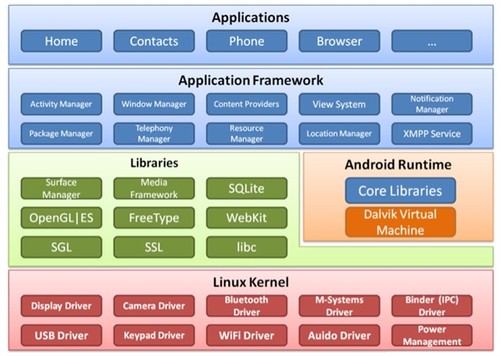
同样在这一层还有Android运行时库提供了一些核心库，能够允许开发者使用Java语言来编写Android应用。还包含了虚拟机Dalvik但之后对ART运行环境进行了部分改变，使得每一个Android应用都有自己的进程，并且都有一个属于自己的Dalvik虚拟机实例，相较于JAVA的虚拟机Dalvik是专门为移动设备定制的，针对内存和CPU性能等方面都有了优化。

1. 应用框架层

这一层主要提供了构建应用程序时可能用到的各种API，Android自带的一些核心应用就是基于这些API完成的。那什么是应用程序框架层？它是一个应用程序的核心，是一个共同使用和遵守的约定，然后在这个约定上共同扩展，但程序保持主体结构的一致，其作用是让程序保持清晰在满足不同需求的同时又不互相影响。而对于Android来说提供给应用开发者的本身就是一个框架，所有的应用开发都必须遵守这个框架的原则，同时在这个基础上进行扩展可以访问核心应用程序所使用的API框架。

1. 应用层

所有安装在手机上的应用程序都是属于这一层的，比如系统自带的联系人、短信等程序，或者从Google Play上下载的小程序，当然还包括自己开发的应用程序。



Android系统架构

**2.1.2 Android系统的优势**

Android系统是一种基于Linux的开源操作系统。从2007年11月诞生以来，Android已经走过了12个年头。在过去的12年里，Android系统发展迅速，并且一度霸占了世界上绝大部分电子设备。Android系统因为开放性、挣脱束缚、丰富的硬件等优势在中国占据越来越多的市场份额。Android系统具备以下几个显著的优点：

1. 开放性。Android的这种开放性允许任何移动终端厂商加入Android，从而可以使其拥有众多的开发者。随着用户资源和应用的日益丰富，一个新平台便会得到迅速的发展。而对于消费者这边而言，最大的受益之处是可以获取到丰富的应用资源。平台的开放同时也意味着更加激烈的竞争，如此一来，消费者便能购买到物美价廉的移动设备。
2. 挣脱运营商的束缚。在过去很长的一段时间里，尤其是在欧美等地区，手机应用等通常会受到运营商等制约，使用什么功能，接入哪种网络等，几乎完全受到运营商的控制。随着3G逐渐过渡到4G网络，手机网络的接入模式已经不再受到运营商的限制。然而，Android系统自出现就具备网络方面的特色，使得手机用户距离互联网更近。
3. 各种丰富的硬件可以选择。Android系统所具有的开放性，使得众多的手机硬件厂商会推出各种具有特色的产品。功能的不同，并不会影响到数据方面的同步，甚至是软件的兼容。
4. 不受任何拘束的开发商。Android平台给第三方开发商提供了一个十分宽泛、自由的开发环境，不会受到各种条条框框的阻扰，给第三方开发商提供及其灵活的开发环境，这样便容易出现更多优秀的应用软件。
5. 联系紧密的Google应用。如今Google已经成立20多年，从搜索引擎到移动互联网的全面渗透，Google推出的一系列服务已然成为了连接用户和互联网的重要纽带，基于Android系统的移动设备也可以无缝连接到Google提供的各种服务。

**2.2 android开发平台简介**

**2.2.1、[Android开发环境中的概念和工具介绍](https://www.cnblogs.com/xiaochao1234/p/3833558.html)**

什么是Android开发环境？Android开发环境当然是用来设计应用于移动设备的系统和软件了，由于Android本身是开源的，那么我们要做的工作就是对其进行定制以及修改，Android手机厂家工作亦是如此，定制自己的驱动（或称BSP，底层）和应用程序（上层）。

开发语言可以使用Java也可以使用C/C++语言，前者Android开发我们称做JDK（Java Development Kit）开发，后者开发我们称之为NDK（Native Development Kit）开发。这么看来的话，Android本身就是C/C++和Java的混合体，所以要学习Android编程，C语言是个基础，Java即使不会也能猜出一二，做个DEMO肯定没有问题。

Android开发环境可以搭建在目前主流系统（Mac、Windows、Linux）的任何一种上，只因为Android相关开发工具的跨平台特性。

1. Java Development Kit（JDK）

Java Development(JDK)是用于开发、编译和测试使用Java语言编写的应用程序、applet 和组件，JDK包含以下几个部分：

开发工具——指工具和实用程序，可帮助您开发、执行、调试和保存以 Java编程语言编写的程序。

运行时环境——由 JDK 使用的 Java Runtime Environment (JRE) 的实现。JRE 包括 Java 虚拟机 (JVM)、类库以及其他支持执行以 Java 编程语言编写的程序的文件。

附加库——开发工具所需的其他类库和支持文件。

演示applet 和应用程序——Java 平台的编程示例源码。

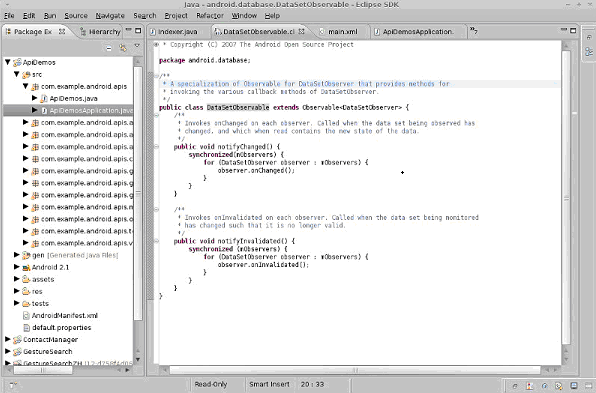
样例代码——某些 Java API 的编程样例源码。

C头文件——支持使用 Java 本机界面、JVM工具界面以及 JavaTM 平台的其他功能进行本机代码编程的头文件。

源代码——组成 Java 核心 API 的所有类的 Java源文件。

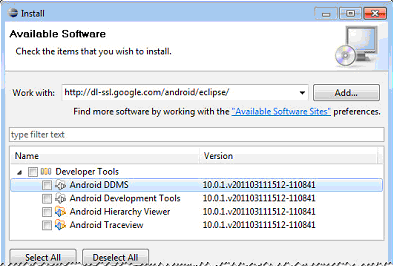
1. [Eclipse](http://www.jizhuomi.com/android/catalog.asp?tags=Eclipse" \t "https://www.cnblogs.com/xiaochao1234/p/_blank)

Eclipse最初由IBM开发的跨平台集成开发环境（IDE），后来贡献给Apache开源软件基金会。最初主要用于Java语言开发，目前可通过C++、Python、PHP等语言插件支持对应语言开发，所以，Eclipse看起来更像一个框架而已，更多工作都是交给插件或上文的JDK来完成，模块化的设计，让Eclipse的定位更清晰。



1. Android Development Tools（[ADT](http://www.jizhuomi.com/android/catalog.asp?tags=ADT" \t "https://www.cnblogs.com/xiaochao1234/p/_blank)）

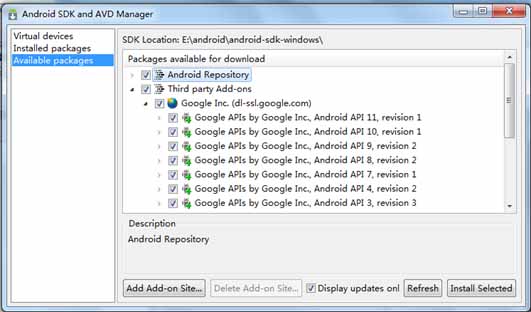
Android开发工具(ADT)，作为Eclipse工具插件，让其支持Android快速入门和便捷开发，可通过Eclipse启动菜单（启动Eclipse后，选择Help->Install New Software）安装。如下图所示。



ADT开发工具，包括Android Dalvik Debug Moniter Server(Android DDMS)和Android Development Tools（ADT），上面的插图中，还有Hierarchy Viewer和TraceView两个查看器，看名称就知道功能了。DDMS可以提供调试设备时为设备截屏，查看线程及内存信息、Logcat、广播信息、模拟呼叫、接收短消息、文件查看器等功能（该功能在安装完ADT之后，点击Eclipse工具栏右上角的那个带箭头的小窗口图标弹出菜单中打开）。 Android Development Tools（ADT）工具应该是为Eclipse支持Android项目快速编程开发和调试插件而已。

1. [Android Software Development Kit](http://www.jizhuomi.com/android/catalog.asp?tags=Android+SDK" \t "https://www.cnblogs.com/xiaochao1234/p/_blank)（SDK）

一般提到SDK就会想到：API接口库、帮助文档和示例源码，Android SDK似乎也不例外，它为开发者提供相关封装API接口库文件、文档资源及一些工具包整合。当然了如果你使用Eclipse作为开发工具，那么只需要安装SDK也可以，ECLIPSE和ADT也一并省略掉。下图是ADK和AVD管理器界面，可从SDK安装目录下执行SDK Manager.exe或在Eclipse中执行Window->Android SDK and AVD manager看到。



安装后SDK目录下的帮助文档相当完善，应有尽有（位置在SDK的docs目录下）。此外，Android SDK还包含一个手机模拟器（Virtual Devices），我们开发时就可以先使用模拟器进行模拟仿真，感觉时机成熟时再下载到真机进行测试。

SDK刚下载解压后，一般还需选择安装一些必须的API接口库及平台工具，下面提到的几项都建议下载安装（至于选择哪个版本，需接口您的开发样机支持Android版本）：

       Android SDK Tools, revision x

       Android SDK Platform-tools, revision x

       Documentation for Android SDK, API xx, revision x

       SDK Platform Android x.x, API x, revision x

       Samples for SDK API x, revision x

       Android Compatibility package, revision x

1. Dalvik虚拟机

Dalvik虚拟机是Android程序的虚拟机，是Android中Java程序的运行基础。其指令集基于寄存器架构，执行其特有的文件格式——dex字节码来完成对象生命周期管理、堆栈管理、线程管理、安全异常管理、垃圾回收等重要功能。它的核心内容是实现库（libdvm.so），大体由C语言实现。依赖于Linux内核的一部分功能——线程机制、内存管理机制，能高效使用内存，并在低速CPU上表现出的高性能。每一个Android应用在底层都会对应一个独立的Dalvik虚拟机实例，其代码在虚拟机的解释下得以执行。

与Dalvik虚拟机关系最密切的非JVM莫属，在Android源码readme文档中有这样一段话：Much of the code under this directory originally came from the Apache Harmony project, and as such contains the standard Apache header comment. Some of the code was written originally for the Android project…

Dalvik VM与Apache Harmony 项目关系源远流长，因此，与JVM关系自然就密切了。然而：Dalvik VM ≠Java VM，dalvik基于寄存器，而JVM基于stack，

Dalvik执行的是特有的DEX文件格式，而JVM运行的是\*.class文件格式。

优势：

（1）在编译时提前优化代码而不是等到运行时；

（2） 虚拟机很小，使用的空间也小；被设计来满足可高效运行多种虚拟机实例；

（3）常量池已被修改为只使用32位的索引，以简化解释器。

1. Android Native Development Kit(NDK)

Android原生开发套件，这里的原生似乎就是指开发语言原生，Android平台基于Linux内核的，所以这里就是指C、C++语言，这对于很多喜欢C/C++的程序员来说或许是个好消息，使用NDK一样可以进行Android开发。

由于NDK开发编译需要GCC编译环境，如果是windows环境，还应该安装Cygwin模拟环境。NDK包含哪些内容：

用于创建基于C/C++源文件的原生代码库；

提供一种将原生库集成到应用程序包，并部署到Android设备的方法；

一系列未来Android平台均会支持的原生系统头文件和库文件；

文档，示例和教程。

**2.2.2、[Android常用集成开发工具Eclipse和Android Studio对比](https://www.cnblogs.com/xiaochao1234/p/3833558.html)**

1、Android Studio的优势

Android Studio 构建程序界面更方便：

Android Studio从一出来的就打着所见得的旗号，一迅雷不及掩耳之势占领了Android项目的开发工具的市场，其实是有道理的。在Eclipse中构建的app的项目，不仅效果和真机上的差别太大，而且速度也不一般。但是Android Studio的界面显示就非常的清楚，并且修改起来也是非常迅速。

Android Studio打印信息更详细：

打印的信息可谓是应有尽有，几乎所有的项目中遇到的问题，包括编写、设计、开发、打包、构建等得错误信息都会在控制台上打印出来，便于问题的准确发下你和定位。反观Eclipse中的打印信息则寒掺得多，除了LogCat之外及时控制台，有事布局文件中的多个逗号“，”都发现不了。

Android Studio编辑历史更详细：

在工作台上修改代码、修改布局文件或者删除文件等，记录得非常细致，每一个操作都有记录，每一个操作都能够撤销。而Eclipse中删除文件后，之前编辑记录就会被清空，及时你恢复文件你就会觉得傻眼的，之前那么多的操作如何回滚是一个问题，除非你使用了版本控制工具。

Android Studio智能识别更强大：

智能识别在Android Studio中是一个非常大的优点，比如只要你输入Fp之后，会自动向你推送含有Fp或者fp、甚至是%F（f）%P（p）的选项，中间不管隔着多少个字符，或者大小不同，系统都能够识别出来并向你推送。

2、Eclipse的优势

Eclipse创建项目更简单:

在Eclipse中创建项目只是点击几下鼠标，然后创建Android项目的问题。但是在Android Studio中可就没有那么简单了，创建一个项目需要超过20分钟的时间不说，创建过程中还可能遇到各种gradle构建的问题，可谓十分麻烦。所以许多的程序习惯从Eclipse中创建项目后导入到Android Studio中。

Eclipse中项目体积小：

在Eclipse中所有的文件必须的，没有多余的配置文件，所有的项目体积很小，毕竟一个项目几十万行的代码，顶多就是30M。但是在Android Studio中就不一样了，各种配置文件，这些文件包含了工具自身的历史文件，还有gradle的构建文件，一个项目超过90M是轻轻松松的问题。

Eclipse中配置文件无需更新：

  创建好一个项目后到项目上线，可能你都无需更新任何Eclipse的文件，这个时间的跨度有可能是一年！而Android Studio中更新gradle文件是家常便饭，昨天才更新的文件，今天启动就需要更新了。甚至在上一次运行时还好好的，这一次运行gradle就出错了，由于不能正常使用Google的原因，更新的速度也不咋滴。

Eclipse中多项目管理很方便：

  因为所有的设置都市针对Eclipse的，添加项目将沿用这些设置，所以管理多个项目是顺理成章的事情。而是Android Studio中每个项目都已每个项目特殊的设置，所以一般是一个打开项目就需要另一个打开一个主窗口。

**2.2.3、本文选择Android Studio开发工具**

总体对比而言，Android Studio更加好用： 1.颜色、图片在布局和代码中可以实时预览 2.string可以实时预览 3.多屏预览、截图带有设备框，可随时录制模拟器视频 4.可以直接打开文件所在位置 5.跨工程移动、搜索、跳转 6.自动保存，无需一直Ctrl + S 7.即使文件关闭依然可以回退N个历史 8.智能重构、智能预测报错 9.每一行文件编辑历史，可追溯到人 10、各种插件 10.例如ADB、.gitignore、sql、markdown、 11.图片可直接转.9图片,并且自带.9编辑 12.在gradle编译时使用依赖超级方便等等。

所以在本文app平台开发中选用了Android Studio开发工具。

**2.3 百度地图开放平台**

**2.3.1 百度地图开放平台概述**

百度地图开放平台面向广大政府、企业、互联网等开发者开放地图服务能力，目前开发者市场占有率约达到75%，每天支撑近50万APP、网站应用运营。

百度地图开放平台拥有六大基础地图服务开放能力：定位、影像、出行、轨迹、数据、分析，并将六大服务能力免费开放给开发者使用。

**2.3.2 百度地图提供的产品与服务**

**定位**

LBS（Location Based Service，基于位置的服务）的三大目标是：你在哪里；你和谁在一起；附近有什么资源。其中“你在哪里”是LBS服务的核心。百度智能定位服务，是为了帮助广大开发者更好解决“你在哪里”这个难题而开放的服务。支持GPS、Wi-Fi、基站融合定位，完美支持各类应用开发者对位置获取的诉求。

百度地图开放平台定位服务是广大开发者定位首选服务，每日定位请求超过300亿次。

基础定位能力——依托百度位置大数据和高精尖定位技术，提供业内最优秀的定位服务产品。具有定位精度高（综合定位精度40m）、覆盖广（完美支持全球定位）、 流量小（小于0.3K）、速度快（首次定位200ms）、内存/CPU占用率低等特点。

离线定位技术——利用百度大数据分析挖掘能力，分析用户常驻点信息，在保证定位精准度的基础之上，提供离线定位能力。 利用离线定位能力，不仅定位速度进一步提升，同时也完美解决无网络环境下的位置获取难题。

室内高精度定位——大型商超内寻找店铺、停车场内反向寻车等业务的发展基础都是室内高精度定位。利用三角定位技术、增强Wi-Fi指纹模型技术、地磁技术、蓝牙技术等， 提供精度1-3m的室内高精度定位服务。结合百度室内地图服务，更好帮助开发者解决室内定位难题。

高精度IP定位——利用定位依据实时建库技术和百度大数据分析挖掘能力，开放高精度IP定位服务。定位综合成功率在65%以上，定位误差可有效控制在350m以内， 定位速度达到5ms。满足Web应用开发者对位置获取的需求。

位置语义化——优质的定位服务，精准性、成功率、覆盖率等是基础。更友好、智能化、人性化的描述当前用户的位置，是未来定位的发展趋势。我们所开放的定位服务， 不仅返回精准坐标，还会返回当前用户的详细地址、周边POI信息，和更人性化的位置描述信息（同时也提供自定义位置描述能力）。

**影像**

提供业内最全最新的矢量地图服务、卫星图、全景图、实时路况图、静态图和个性化地图服务。

**出行**

提供步行、骑行、公交、驾车等多种类型的线路规划方案，各方案还支持多种不同策略的检索，面向移动端设备提供导航SDK等产品。

**轨迹**

百度鹰眼是一套集轨迹追踪、存储、运算、查询的完整轨迹开放服务，可帮助开发者管理多达100万人/车轨迹。使用百度鹰眼，开发者可以轻松开发出适用于车队监控、车联网、专车计费、外业人员监管、儿童防丢领域的轨迹管理产品。

轨迹追踪——鹰眼提供Android SDK、iOS SDK、Web服务API、车载硬件解决方案和智能穿戴设备解决方案，帮助开发者轻松将手机、车联网硬件、智能穿戴设备等终端连接至鹰眼云端，持续回传轨迹。

存储并查询轨迹数据——鹰眼为轨迹数据提供长达1年的存储。同时，使用鹰眼移动端和服务端查询接口，您的应用程序可几乎无延时地查询终端实时位置和历史轨迹。

鹰眼硬件联盟成员系列——鹰眼联合优质硬件厂商组成硬件联盟，为开发者提供多种硬件产品选择。目前已开通购买硬件渠道，并欢迎广大硬件厂商加入百度鹰眼。

**数据**

基于百度地图丰富的地图数据，开放POI检索、热词推荐、地理编码等服务，通过LBS云服务实现结合个性化数据的地图功能开发，满足开发者对各类数据方面的需求。

分析

基于地理大数据、位置大数据、交通大数据和海量行为数据的商业地图产品，以领先的大数据分析和可视化技术，为政企、开发者提供更好的分析能力。

客流来源去向——从省、市、区县、商圈、街道维度精细分析客流来源、客流去向周边的分布。

客群画像——常驻居民与流动客群画像勾勒，从性别、年龄、资产状况、兴趣爱好、消费水平、消费偏好等多重维度立体化勾勒。

位置评估——分析常驻居民数量与密度分布、流动客群数量与密度分布、不同职业、年龄段人群分布、设施及场所分布。

室内客流分析——整体、楼层、店铺等客流分析、新老顾客及到店次数分析、实时客流热力分布。

室内定位——室内数据专人采集。

**2.4 聚类方法**

**2.4.1 聚类方法简介**

聚类是无监督学习中的典型算法，不需要对结果进行任何标记。聚类方法通常用来试图探索和发现一定的模式，进而用于发现共同的群体，按照内部存在的相似性将数据划分为多个类别使得同类内部相似性大，不同分类间相似性小。有时候作为监督学习中稀疏特征的预处理（类似于降维等方法，变成n类后，假设有6类，则每一行都可以表示为类似于000100、010000）。有时候可以作为异常值检测（反欺诈中有用）。

应用场景：新闻聚类、用户购买模式（交叉销售）、图像与基因技术等。

相似度与距离：这个概念是聚类算法中必须明白的，简单来说就是聚类就是将相似的样本聚到一起，而相似度用距离来定义，聚类是希望组内的样本相似度高，组间的样本相似度低，这样样本就能聚成类了。

**2.4.2 常用聚类方法介绍**

1. K-Means(K均值)聚类

算法步骤：

(1) 首先需要选择一些类/组，并随机初始化它们各自的中心点。中心点是与每个数据点向量长度相同的位置。这需要提前预知类的数量(即中心点的数量)；

(2) 计算每个数据点到中心点的距离，数据点距离哪个中心点最近就划分到哪一类中；

(3) 计算每一类中中心点作为新的中心点；

(4) 重复以上步骤，直到每一类中心在每次迭代后变化不大为止。也可以多次随机初始化中心点，然后选择运行结果最好的一个。

优点： 速度快，计算简便 。

缺点： 必须提前知道数据有多少类/组；

2. 基于密度的聚类方法(DBSCAN)

DBSCAN是基于密度的聚类算法。

具体步骤：

（1） 首先确定半径r和minPoints. 从一个没有被访问过的任意数据点开始，以这个点为中心，r为半径的圆内包含的点的数量是否大于或等于minPoints，如果大于或等于minPoints则改点被标记为central point,反之则会被标记为noise point；

（2）重复1的步骤，如果一个noise point存在于某个central point为半径的圆内，则这个点被标记为边缘点，反之仍为noise point。重复步骤1，知道所有的点都被访问过。

优点：不需要知道簇的数量；

缺点：需要确定距离r和minPoints。

3. 层次聚类

层次聚类算法分为自上而下和自下而上两种。凝聚层级聚类(HAC)是自下而上的一种聚类算法。HAC首先将每个数据点视为一个单一的簇，然后计算所有簇之间的距离来合并簇，知道所有的簇聚合成为一个簇为止。

具体步骤：

1. 首先将每个数据点视为一个单一的簇，然后选择一个测量两个簇之间距离的度量标准。例如使用average linkage作为标准，它将两个簇之间的距离定义为第一个簇中的数据点与第二个簇中的数据点之间的平均距离；

2. 在每次迭代中，将两个具有最小average linkage的簇合并成为一个簇。

3. 重复步骤2知道所有的数据点合并成一个簇，然后选择需要多少个簇。

优点：

（1）不需要知道有多少个簇 ；

（2）对于距离度量标准的选择并不敏感。

缺点：效率低。

**2.4.3 本文基于ROCK算法对交通数据进行聚类。**

1.ROCK 算法中用到的四个关键概念：

邻居（Neighbors）：如果两个样本点的相似度达到了阈值（θ），这两个样本点就是邻居。阈值（θ）由用户指定，相似度也是通过用户指定的相似度函数计算。

链接（Links）：两个对象的共同邻居数量。

目标函数（Criterion Function）：最大化下面目标函数以获得最优的聚类结果（最终簇之间的链接总数最小，而簇内的链接总数最大）。Ci：第i个簇，k:簇的个数，ni:Ci的大小（样本点的数量）。一般可使用f (θ) = (1-θ)/(1+θ). f(θ)一般具有以下性质：Ci中的每个样本点在Ci中有nif(θ)个邻居。

相似性的度量（Goodness Measure）：使用该公式计算所有对象的两两相似度，将相似性最高的两个对象合并。通过该相似性度量不断的凝聚对象至k个簇，最终计算上面目标函数值必然是最大的。

2.算法过程：

输入：

需要聚类的个数-k，和相似度阈值-θ

输出：

　　簇和异常值（不一定存在）

算法：

　　开始每个点都是单独的聚类，根据计算点与点间的相似度，生成相似度矩阵；

　　根据相似度矩阵和相似度阈值-θ，计算邻居矩阵-A。如果两点相似度>=θ,取值1（邻居），否则取值0；

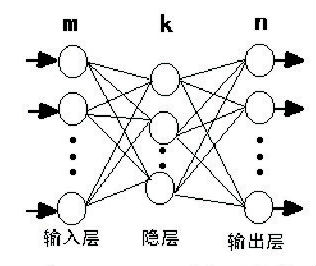
　　计算链接矩阵-L=A x A；

　　计算相似性的度量（Goodness Measure），将相似性最高的两个对象合并。回到第2步进行迭代直到形成k个聚类或聚类的数量不在发生变换。

**2.5 神经网络预测及优化**

**2.5.1 BP网络模型介绍**

BP网络（Back-ProPagation Network）又称反向传播神经网络， 通过样本数据的训练，不断修正网络权值和阈值使误差函数沿负梯度方向下降，逼近期望输出。它是一种应用较为广泛的神经网络模型，多用于函数逼近、模型识别分类、数据压缩和时间序列预测等。



m×k×n 三层BP神经网络

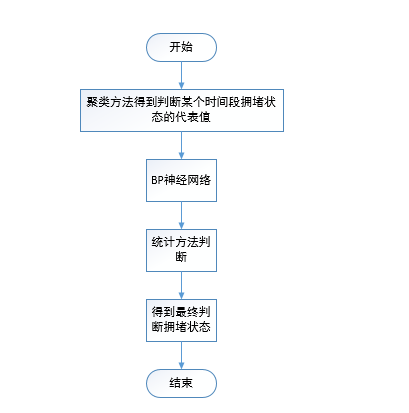
BP网络由输入层、隐层和输出层组成，隐层可以有一层或者多层，图2是m×k×n的三层BP网络模型，网络选用S型传递函数IMG_256， 通过反传误差函数IMG_257 （ （Ti为期望输出、Oi为网络的计算输出），不断调节网络权值和阈值使误差函数E达到极小。

BP网络具有高度非线性和较强的泛化能力，但也存在收敛速度慢、迭代步数多、易于陷入局部极小和全局搜索能力差等缺点。

**2.5.2 基于BP网络模型的优化思路**

1. 方法提出

本文中神经网络简单模拟图示



本文实验框架流程图

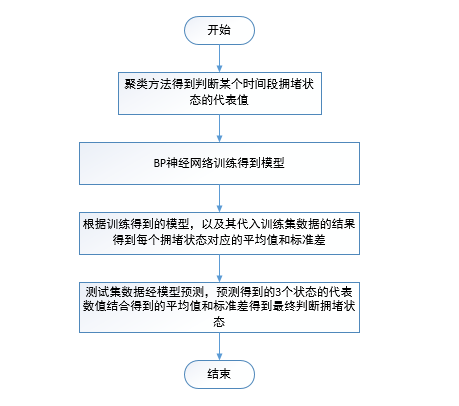
第一阶段

在第一阶段中，通过训练得到经典的BP神经网络模型，如上图所示，输入数据为4个，是4个连续单位时间内通过的车辆数量，预测结果为3个在0到1之间的数值，分别对应拥堵、一般、空闲的代表值，通过这3个数据来判断接下来一个连续单位时间内的交通拥堵状态，通过训练可得到经典的BP神经网络模型。

第二阶段

通过获得的数据集的大部分数据作为训练集，第一阶段训练得出了经典的神经网络模型。如上图所示，模型对输入会得到3个对应的输出。本文在训练出第一阶段的神经网络模型后，再次将训练集数据输入，然后得到所有训练集数据通过模型预测得到的3个代表数值和数据对应交通拥堵状态的对应值。之后再将所有同一拥堵状态下的这3个数值及其对应着的状态进行归类整合在一起。再对所有属于每个分类下的数据进行处理，获得每个状态分类下对应的最大数值的平均值和标准差。最后在对测试集的预测中，再对预测得到的3个数值进行状态归类的环节，结合这两个数据，判断这3个数值相对每一种状态的平均值的偏离程度，并以此作为辅助判断预测结果的手段。

1. 算法提出



算法流程图

第一部分

K-MEANS算法对交通数据进行聚类

输入：

需要聚类的个数-k，和相似度阈值-θ

输出：

　　簇和异常值（不一定存在）

算法：

　　开始每个点都是单独的聚类，根据计算点与点间的相似度，生成相似度矩阵；

　　根据相似度矩阵和相似度阈值-θ，计算邻居矩阵-A。如果两点相似度>=θ,取值1（邻居），否则取值0；

　　计算链接矩阵-L=A x A；

　　计算相似性的度量（Goodness Measure），将相似性最高的两个对象合并。回到第2步进行迭代直到形成k个聚类或聚类的数量不在发生变换。

第二部分

结合神经网络模型和统计学方法对交通流的拥堵状况进行预测的算法

具体过程如下：

输入：连续4个单位时间（5分钟）内的车流辆数据

输出：判断得到的交通拥堵状态，比如拥挤、一般和空闲。

第一步：将所有数据集数据集分为训练集和测试集两部分；

第二步：通过聚类得到的交通拥堵状态判断标准，对训练集中所有数据进行 状态的归类。

第三步：通过第二步中的归类结果，建立一般的BP神经网络模型，该模型包含4个输入和3个输出，3个输出的数值为3种状态的代表数值，并以此判断交通拥堵状态，开始进行训练；

第四步：通过训练得到的神经网络模型反过来对训练集数据进行处理得到各自状态对应的3个数值；并通过3个数值种取决定性作用的那个数值来确定每个状态对应的平均值和标准差。

第五步：最终通过下面的公式对3个数值所属于的交通拥堵状态进行判别

，，y为通过模型后得到的每种状态的代表数值，比如判断是否属于拥堵状态，在公式中通过比较预测得到的数值时候在这个区间可以进行判断，



第六步：重复前面的步骤依次对一般状态以及空闲状态进行判别，通过第五步的辅助判断标准，便更能确定各状态的判别结果。

1. **系统需求分析**
2. **系统详细设计与实现**
3. **总结与展望**

**致谢**

**参考文献**