**吉林大学学士学位论文（设计）承诺书**

本人郑重承诺：所呈交的学士学位毕业论文（设计），是本人在指导教师的指导下，独立进行实验、设计、调研等工作基础上取得的成果。除文中已经注明引用的内容外，本论文（设计）不包含任何其他个人或集体已经发表或撰写的作品成果。对本人实验或设计中做出重要贡献的个人或集体，均已在文中以明确的方式注明。本人完全意识到本承诺书的法律结果由本人承担。

学士学位论文（设计）作者签名：

2017年5月20日

摘要

基于Qt的卷积神经网络辅助设计系统

以卷积神经网络为代表的神经网络方法近年来在图像识别等领域取得了传统方法难以企及的优秀成绩。神经网络的构成，本身具有模块化的特点，因此已经有数款可以用来搭建神经网络的深度学习框架问世。然而这些框架均采用基于命令行的工具集或者动态、静态库的形式发布，操作直观性差，学习周期较长，另外，由于对神经网络的理论解释尚不完善，尚无一个能够直接指导新网络结构该如何建立的理论，神经网络的设计和测试过程中需要进行频繁的参数调整等人工操作，进一步加重了开发人员的负担。为了解决上述问题，本文提出并设计实现了一套采用C++语言，以Qt作为图形用户接口的集成化、图形化的卷积神经网络辅助设计系统。该系统提供了常用数据格式的转换和读取、神经网络的设计、测试、特征导出等操作，并支持核心组件的替换和更新，能够为相关开发人员减少时间成本。经过测试，系统达到了预期的设计要求。

关键字：卷积神经网络，深度学习框架，Qt，辅助设计，图形用户界面

**Abstract**

Qt Based Design System for Convolutional Neural Network

The neural network method, which is represented by convolution neural network, has achieved excellent results in the field of image recognition in recent years. The composition of the neural network has the characteristics of modularity, so there are several depth learning framework come out，which can be used to build the neural network. However, these frameworks are based on the command line tools and the form of release is dynamic or static library with poor operational intuition and a longer learning cycle. In addition, due to the theoretical explanation of the neural network is not perfect, no one can directly guide the new network structure how to build the theory. The neural network design and testing process requires frequent parameter adjustment and other manual operations, which further increasing the burden on the developers. In order to solve the above problems, this paper presents and designs a set of integrated graphical convolution neural network design system with C + + and Qt user interface. The system that can reduce the time for the relevant developers cost provides common data format conversion and reading, neural network design, testing, feature export and other operations, and support the replacement and update of the core components. After testing, the system meets the intended design requirements.

Keywords：Convolution neural network，Depth learning framework，Qt，Auxiliary design，Graphical user interface

**目 录**

[第1章　绪论 1](#_Toc485196397)

[1.1 研究背景和研究意义 1](#_Toc485196398)

[1.1.1 研究背景 1](#_Toc485196399)

[1.1.2 研究意义 2](#_Toc485196400)

[1.2 国内外研究现状分析 3](#_Toc485196401)

[1.2.1 深度学习框架的研究现状 3](#_Toc485196402)

[1.2.2 深度学习相关的可视化工具 3](#_Toc485196403)

[1.2.3 图像化编程框架 4](#_Toc485196404)

[1.2.4 总结 4](#_Toc485196405)

[1.3 技术路线及方案 5](#_Toc485196406)

[1.3.1 研究过程和技术路线 5](#_Toc485196407)

[1.3.2 重点和难点以及解决方案 5](#_Toc485196408)

[第3章 系统实现 7](#_Toc485196409)

[3.2 需求分析 7](#_Toc485196410)

[3.3 系统设计 7](#_Toc485196411)

[3.3.3 系统处理流程 7](#_Toc485196412)

[3.4 系统实现 8](#_Toc485196413)

[3.4.1实现概述 8](#_Toc485196414)

[3.4.2 具体实现 9](#_Toc485196415)

[第4章　测试和评价 14](#_Toc485196416)

[4.3 部署测试 14](#_Toc485196417)

[第5章 总结与展望 14](#_Toc485196418)

[5.1 总结 14](#_Toc485196419)

[参考文献 15](#_Toc485196420)

[致 谢 18](#_Toc485196421)

第1章　绪论

1.1 研究背景和研究意义

本文描述了一种基于Qt的卷积神经网络辅助设计系统。在系统的开发过程中，涉及到了机器学习、神经网络、可视化编程以及神经网络的设计与调试等问题，因此，深入了解相关的背景以及研究现状对于毕业设计的完成具有实际意义。

### 研究背景

1. 机器学习

机器学习是一项致力于研究如何通过计算的手段，利用数据来增强系统自身性能的学科。以“神经网络”为代表的深度学习方法所获得的模型中，非线性操作的层级数[1]更多，凭借精度、处理速度等多方面的优势，近年来神经网络方法在语音识别、图像处理等多类应用中取得了突破性进展[2-9]。

1. 卷积神经网络
2. 卷积神经网络的简介

神经网络系统的结构是一个有向无环图，信息从输入层通过一个或多个层到达输出层，完成原始信号到高级语义的映射。典型的前馈神经网络有多层感知机[11]和卷积神经网络[12]等。

卷积神经网络通常可以看做是由很多结构类似的“网络模块”首尾相连形成的，每个“网络模块”由卷积阶段、激活阶段和下采样阶段等组成。在若干个“网络模块”之后，通常会连接若干个全连接层和一个输出层，通过分类器得出最终的分类结果，一个完整的卷积神经网络的示意图如图1-1所示。

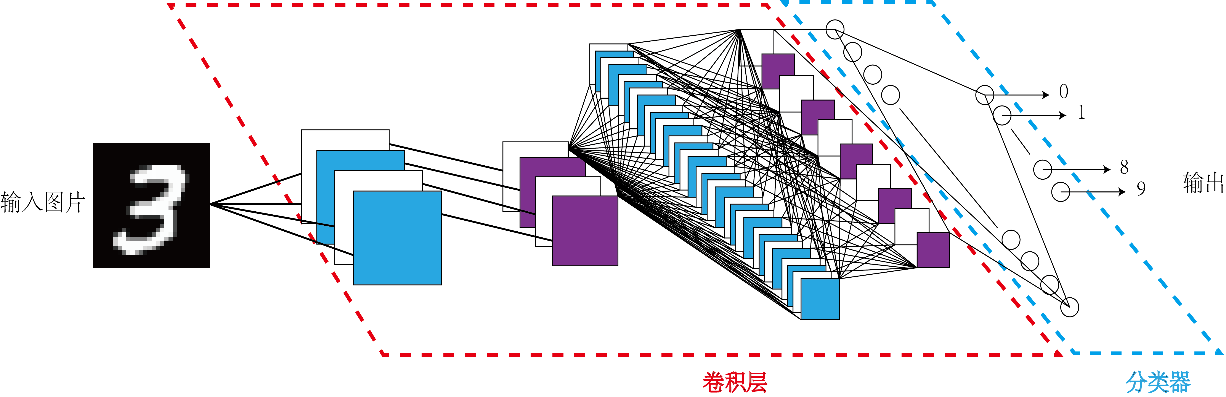


图1-1 卷积神经网络

通常使用反向传播算法以及有监督的训练方式对神经网络进行训练，算法流程如图1-2所示。在训练开始阶段，网络中权值的初值可以通过随机分布函数进行初始化，或者通过加载预训练的网络来获得[17]。网络的误差将会随迭代的进行而逐渐降低，并逐渐逼近极限值，此时称网络已经收敛。

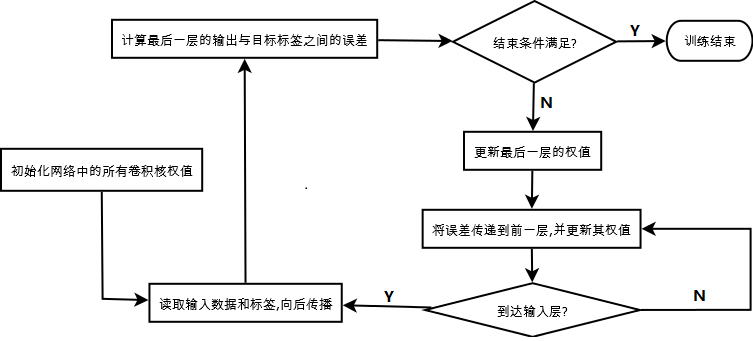


图1-2 反向传播算法

1. 卷积神经网络的设计和测试

更大的数据集、更快的设备和更高效的工具链是卷积神经网络的进一步发展所必备的条件。虽然已有多款深度学习框架问世，但这些框架均是由一系列命令行工具甚至文本组成，掌握他们的使用方法需要一定的学习成本，且集成环境和工具链的缺位，使得这些工具的环境配置较为繁琐，一旦在训练和测试中出现问题，将会给本就漫长的设计流程添加不必要的时间成本。

### 研究意义

深度学习，尤其是神经网络具有较高的研究价值。然而，当前的深度学习工具存在着集成化程度低，学习成本高，环境配置繁琐等问题，这些问题的存在引入了大量时间成本，因此，一个集成化的、图形化的，能够为开发人员节省时间和精力的研究开发环境，对深度学习的发展有一定的意义。

* 1. 国内外研究现状分析

### 深度学习框架的研究现状

目前，常用的深度学习框架有Caffe, TensorFlow, MXNet, Torch, Theano等，表1-1对这些常用的机器学习框架进行了总结。

表1-1 常用机器学习框架对比

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 框架名称 | 开发语言 | 运行速度 | 灵活性 | 文档支持 | 适合模型 | 平台支持 | 上手难易 |
| Caffe | C++/CUDA | 快 | 一般 | 全面 | CNN | 全平台 | 一般 |
| TensorFlow | C++/CUDA/Python | 一般 | 好 | 一般 | CNN/RNN | Linux/OS | 难 |
| MXNet | C++/CUDA | 快 | 好 | 全面 | CNN | 全平台 | 一般 |
| Torch | C/Lua/CUDA | 快 | 好 | 全面 | CNN/RNN | Linux/OS | 一般 |
| Theano | Python/C++/CUDA | 一般 | 好 | 一般 | CNN/RNN | Linux/OS | 易 |

在综合考虑了上手难易度、适用平台、编程语言和编译环境之后，本文拟采用Caffe作为核心框架。

### 深度学习相关的可视化工具

在描述神经网络方面，纽约大学的Matthew等人提出了所谓的“反卷积网络”[18]，采用与卷积神经网络并联的反卷积网络将深层次特征向输入层重映射，给出了一种尝试理解卷积神经网络深层次特征的思路和方案[19]。Caffe的作者贾杨清也编写了网络结构示意图绘制工具[2。

在人机交互方面，目前绝大多数主流的深度学习工具均使用控制台工具，NVIDIA的DIGITS框架自带了一个基于Web的用户界面，通过界面可以进行简单的参数设置和训练监控，但是遗憾的是该框架的使用并不广泛且使用难度较大。

无论是对卷积神经网络的理解、描述，还是人机交互，目前都有较多的工作在做各种尝试，但一个广为接受的理论解释至今没有诞生。另一方面，在深度学习工具的建设方面，工具的通用性很差，它们大多数依赖于某种特定的网络结构和数据集，有些甚至只是显示后台事先渲染好的数据，对于神经网络开发工作的辅助作用也较为有限。

### 图像化编程框架

目前，有很多支持用户界面设计与实现的程序设计框架，常用的有MFC、Java Swing、C# WinForm、WPF和Qt等，表1-2对上述的几种常用[图形用户界面](http://baike.baidu.com/item/%E5%9B%BE%E5%BD%A2%E7%94%A8%E6%88%B7%E7%95%8C%E9%9D%A2" \t "_blank)应用程序开发框架进行了对比。

表1-2 常用图形界面应用程序开发框架对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 框架名称 | 开发语言 | 是否跨平台 | 布局控制 | 是否更新 | 开发复杂度 |
| MFC | C++ | 否 | C++ | 否 | 高 |
| Java Swing | Java | 是 | Java | 是 | 一般 |
| C# WinForm | C# | 否 | C# | 是 | 一般 |
| WPF | C#/VB.Net | 是 | XAML | 是 | 低 |
| Qt | C++/Python | 是 | C++/QML | 是 | 一般 |

本文拟采用Caffe作为核心组件，由于Caffe采用C++编码，为了减少程序开发的复杂度，避免引入交叉编译等额外问题，且综合考虑了技术的先进性和交互效果，本文采用Qt作为[图形用户界面](http://baike.baidu.com/item/%E5%9B%BE%E5%BD%A2%E7%94%A8%E6%88%B7%E7%95%8C%E9%9D%A2" \t "_blank)应用程序开发框架。

### 总结

以神经网络为代表的深度学习方法受到了研究人员的广泛关注。但是存在着理论解释缺乏，新网络的设计和调整依靠经验等问题，同时，仍有诸如减少数据依赖、进一步提高精度和效率、拓展应用领域等问题亟待解决，因此，以神经网络为代表的深度学习方法具有较高的研究价值和研究空间。

目前已经有多款深度学习框架问世。但这些框架多以开源库、工具集等形式出现，没有形成集成环境，存在着搭建难度高、命令复杂、学习成本高等问题，给开发过程增添了额外的成本。集成化、图像化的开发环境，对于神经网络的开发具备实际意义。

在对现有的深度学习框架和图形用户界面应用程序开发框架进行了对比和梳理之后，决定采用Qt和Caffe，开发一款集成化、图形化的卷积神经网络辅助设计系统。

* 1. 技术路线及方案

### 研究过程和技术路线

依照软件工程的理论和实践经验，本文遵循需求分析、概要设计、详细设计、编程，单元测试和部署测试等步骤与顺序，确立了如下技术路线：

1. 文档化管理。参考Microsoft Visual Studio的界面布局方案，将对文件的直接操作入口绑定在文件树的右键上下文菜单上。为了在整个界面内合理排布各种功能，采用“边缘浮动窗口+中心MDI视图”的设计方案。
2. 调用核心组件。本系统采用Caffe深度学习框架作为核心组件，参考集成开发环境（IDE）“编译核心+辅助功能壳”的设计思路，将不需要后期扩展变更的功能封装成独立的可执行文件，而主程序通过启动额外进程的形式对该组件进行调用，采用输入输出重定向的方法实现线程之间的值传递。
3. 图形化的蓝图编辑器。通过对Qt的图形组件进行重载和改写，实现碰撞检测，曲线绘制和上下文响应。
4. 配置管理。配置管理模块负责管理文件，并向需要参数的其他模块提供参数访问服务，用户可以通过图形界面或者直接编辑配置文件的形式更改这些参数。通过修改核心组件的位置，可以实现可新组件的替换。

### 重点和难点以及解决方案

在本毕业设计的开发过程中，有以下重点和难点，相应的解决方案也一并列出。

1. **Qt和Caffe同时使用。**带有“QObject”标记的Qt类的生成工具必须使用Qt的生成工具；另一方面，由于Caffe采用了Google的Protobuf作为数据交换格式，包含或间接包含“caffe.pb.h”的C++文件的生成工具必须是protoc.exe。由于生成工具的冲突，这样的类势必无法编译成功。

**解决方案：**通过引入另一个类隔离Qt类和Caffe类。为了解决上述生成工具不兼容的问题，本文中大量使用了如图1-11所示的类结构，可见，通过中间类的隔离，不需要既含有caffe.pb.h头文件，又带有“QObject”标记的类就可以实现从图形界面调用核心功能。同时，中间类的存在可以封装参数完整性验证等与核心功能关系不大且篇幅比较长、影响业务逻辑的可读性的代码。

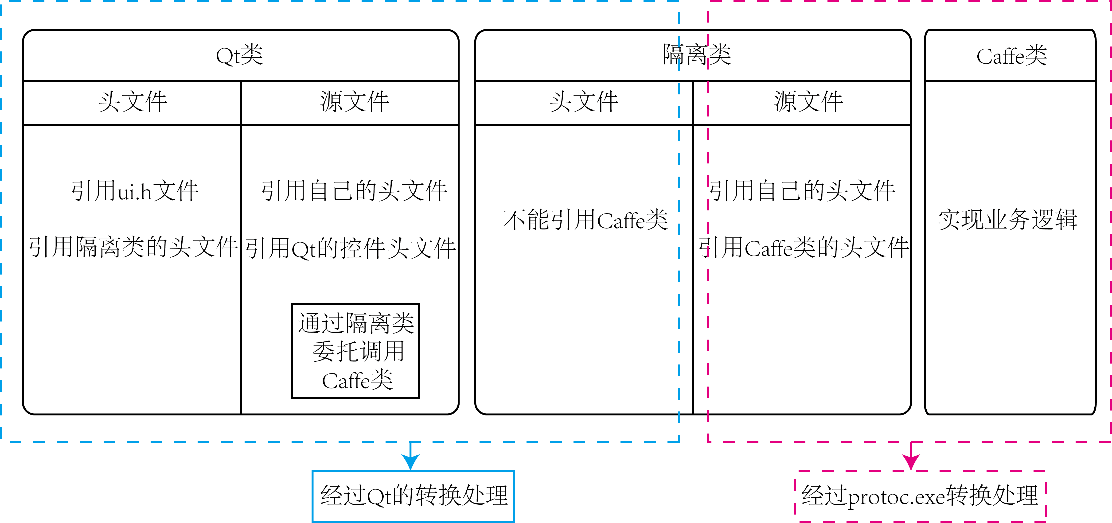


图1-11 通过中间类隔离Qt类和Caffe类

1. **蓝图编辑器中图形的绘制。**参考虚幻编辑器中“模块”之间的链接曲线，需要实现一种能够自动根据起止点变换曲线形状的曲线生成方法。现有的曲线生成函数不能直接实现这个功能。

**解决方案：**分析具体的绘制过程，以贝塞尔曲线为基础，定义“灵敏系数”和“变距因数”，通过一个自定义函数调控曲线控制点的位置，动态生成贝塞尔曲线，从而实现上述功能。

第3章 系统实现

3.2 需求分析

（1）对于模型的训练，测试，提供图形操作界面。

（2）对网络结构的建立提供辅助，实现形式可以是语法高亮等。

（3）对于常用数据形式，提供可视化的格式转换工具和浏览工具。

（4）能够以可视化的方式显示网络模型提取到的特征。

（5）支持核心组件的升级和替换。

3.3 系统设计

### 3.3.3 系统处理流程

系统共分为八个模块，另有三个独立视图和一个核心组件。其中，数据集浏览模块、数据转换模块和特征导出模块采用基于中间层的内部委托调用形式调用核心组件中的相关接口；训练模块和测试模块通过输入输出重定向模块，采用外部进程的形式调用核心组件，并通过控制台视图进行交互；首选项模块可以对该过程中所使用的命令和组件进行配置；文件视图，开始视图是独立的；编辑器模块依赖于系统的图形界面。系统的整体结构见图3-1。

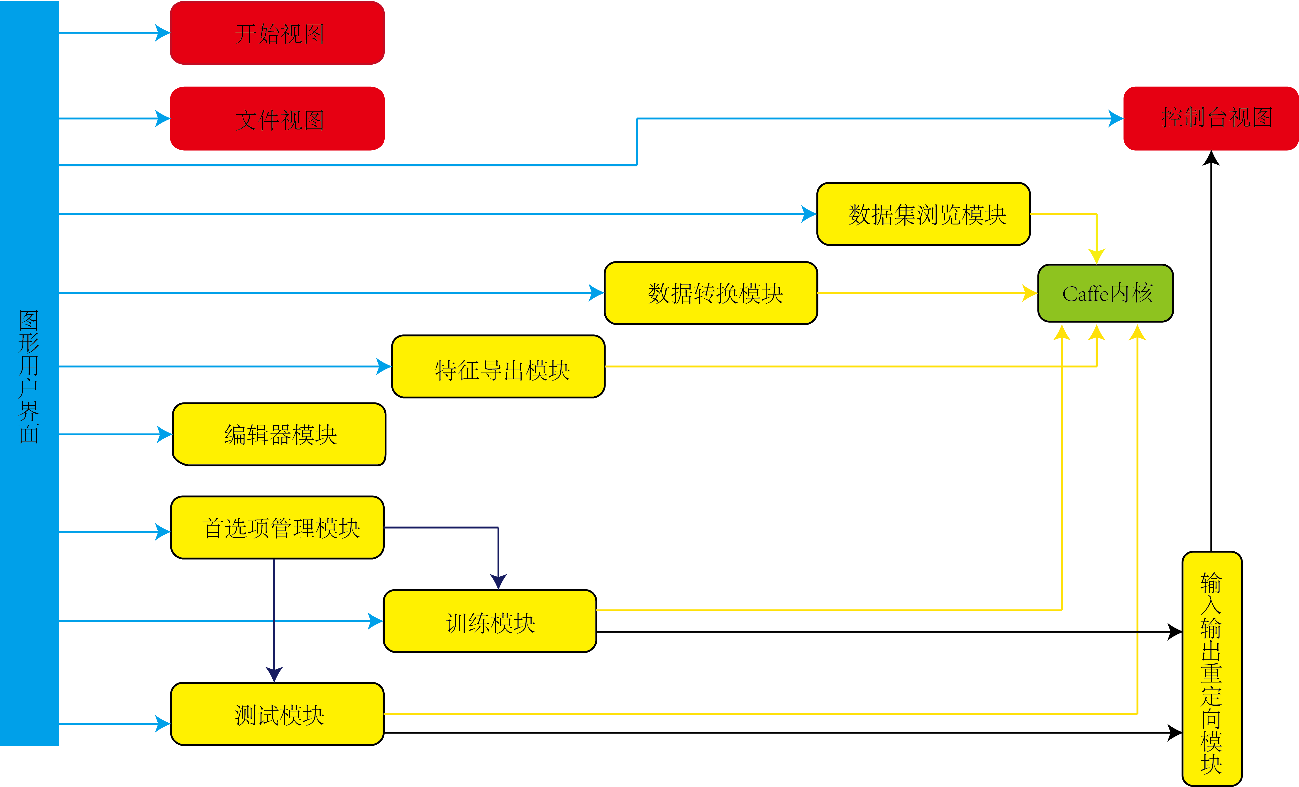


图3-1 “基于Qt 的卷积神经网络辅助设计系统”系统结构图

3.4 系统实现

### 3.4.1实现概述

系统的图形用户界面采用Qt开发，主界面采用MDI文档为核心，浮动窗口为辅助的视图形式，首选项管理器采用模态对话框实现，文件树视图和终端视图使用浮动窗口实现，其余的功能如数据转换，训练、测试等功能均为MDI文档视图，在编辑器视图中，实现了语法高亮和可视化编辑功能。

图3-2展示了系统最终实现的界面效果。



图3-2 系统界面效果

### 3.4.2 具体实现

（1）支持语法高亮的文本编辑器

图3-3展示实现语法高亮文本编辑器的类结构。其中，Ui\_NetEditor类实现了编辑器的界面，具体界面见图3-4，Editor类继承自QPlainTextEdit，具备基本的文本编辑功能，并实现了文本编辑器的保存、另存为、载入、新建等功能，同时使用一个槽函数，连接到内容变更信号上，每当编辑器界面中的内容发生了变更，就触发highlightCurrentLine()函数，该函数根据MyHighLighter中定义的语法规则对当前行使用正则表达式进行匹配，并根据匹配结果使用对应的颜色和字体进行语法标注。另外实现了行号显示条。语法高亮的效果如图3-4所示。

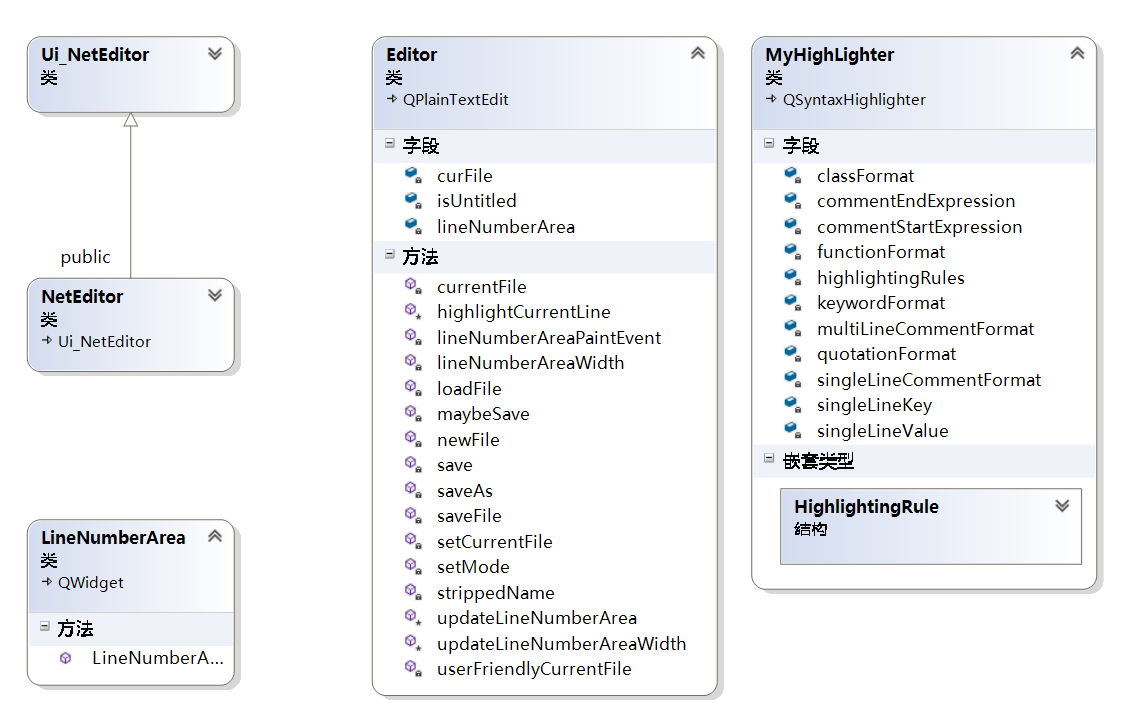


图3-3 实现语法高亮文本编辑器的类结构

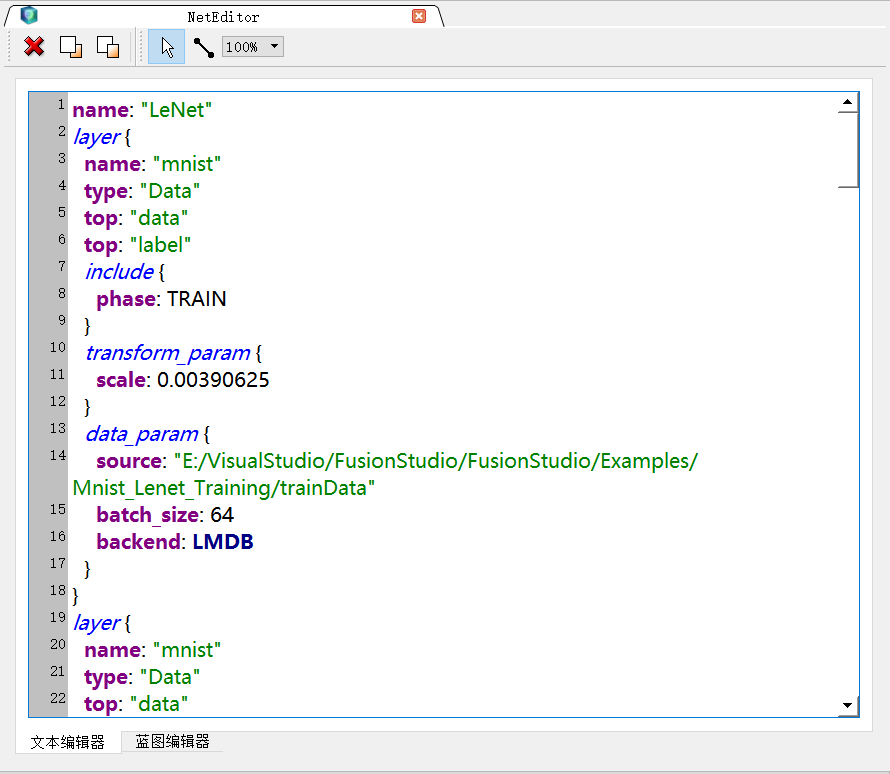


图3-4 文本编辑器的显示效果

（2）图形化网络结构设计器

图3-5展示实现蓝图编辑器的类结构。实现了插入图元、删除图元和图元颜色编辑等操作，图元可以拖动；图元从矢量图加载，能够向绘图区域进行绘制，并支持碰撞检测；继承QGraphicsLineItem实现Arrow类，用于存储和绘制网络块之间的连接曲线，该曲线要连接两个网络块且能够随着网络块的位置变化而自动调整形状，为了实现该功能，对标准曲线绘制函数进行扩展。

Qt提供了贝塞尔曲线的绘制函数cubicTo()，该函数绘制的贝塞尔曲线需要指定起点、终点和两个控制点，其基本形状如图3-6所示，符合系统对曲线形状的要求，但是这样的曲线无法实现根据起止点位置而调整自身形状，为此，我们定义了灵敏度参数*theta*和变距因数*m*。其中，*theta*是一个常数，通过调整这个值来调整曲线变形的速度，*m*的值由以下的公式给出：

其中，点（）是曲线的起点，（）是曲线的终点，变距因数*m*的作用是根据曲线的起点和终点拉开两个控制点之间在X轴上的距离，根据贝塞尔曲线的特点，拉开这两个控制点的距离会使得曲线的两个拐点更加“锋利”。两个控制点的坐标使用以下的方法生成：

,

,

其中，（，）和（，）为曲线的控制点，*p*为纵向灵敏度参数。曲线的绘制结果如图3-7和图3-8所示。

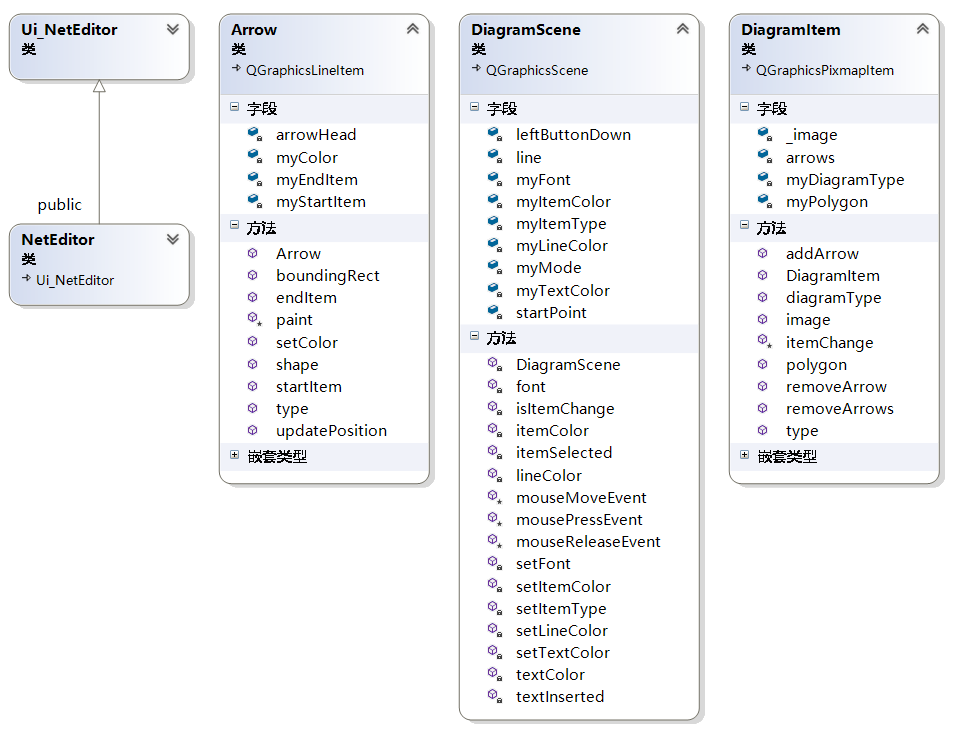
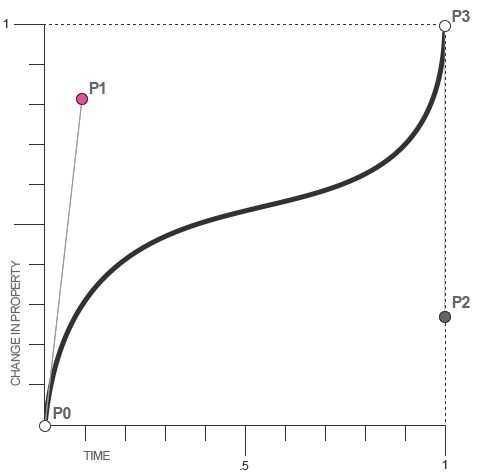


图3-5 图形化网络结构编辑器（蓝图编辑器）的类结构



3-6 贝塞尔曲线

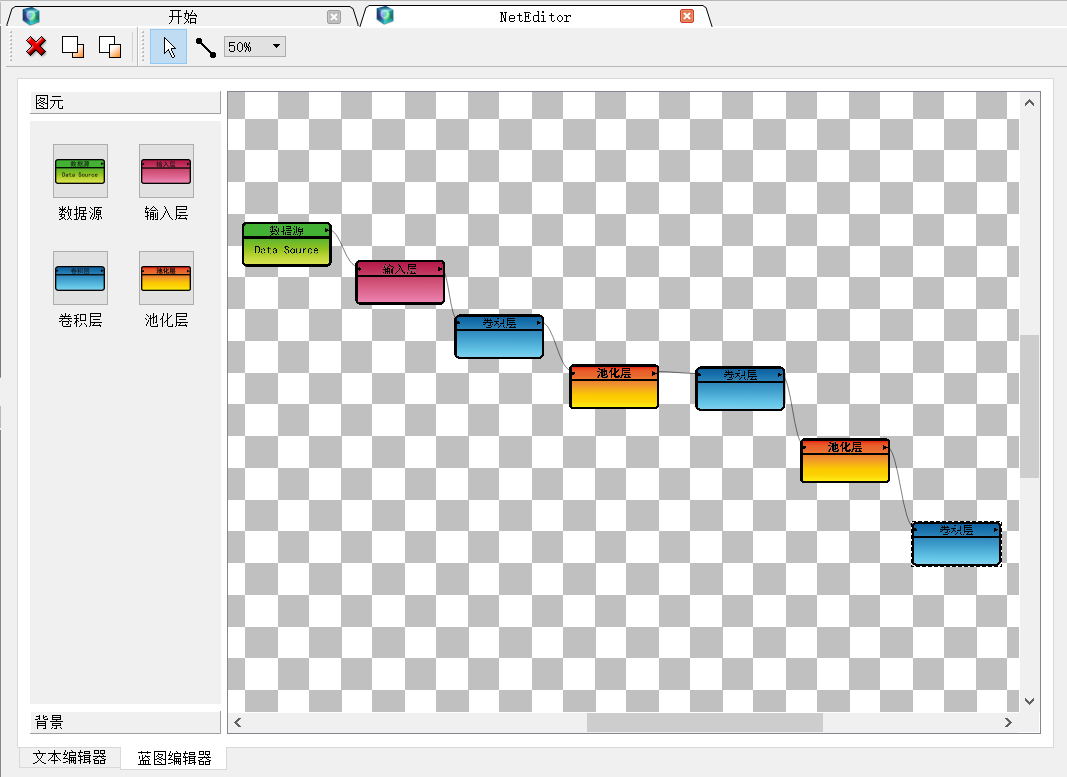


图3-7 蓝图编辑器 （1）

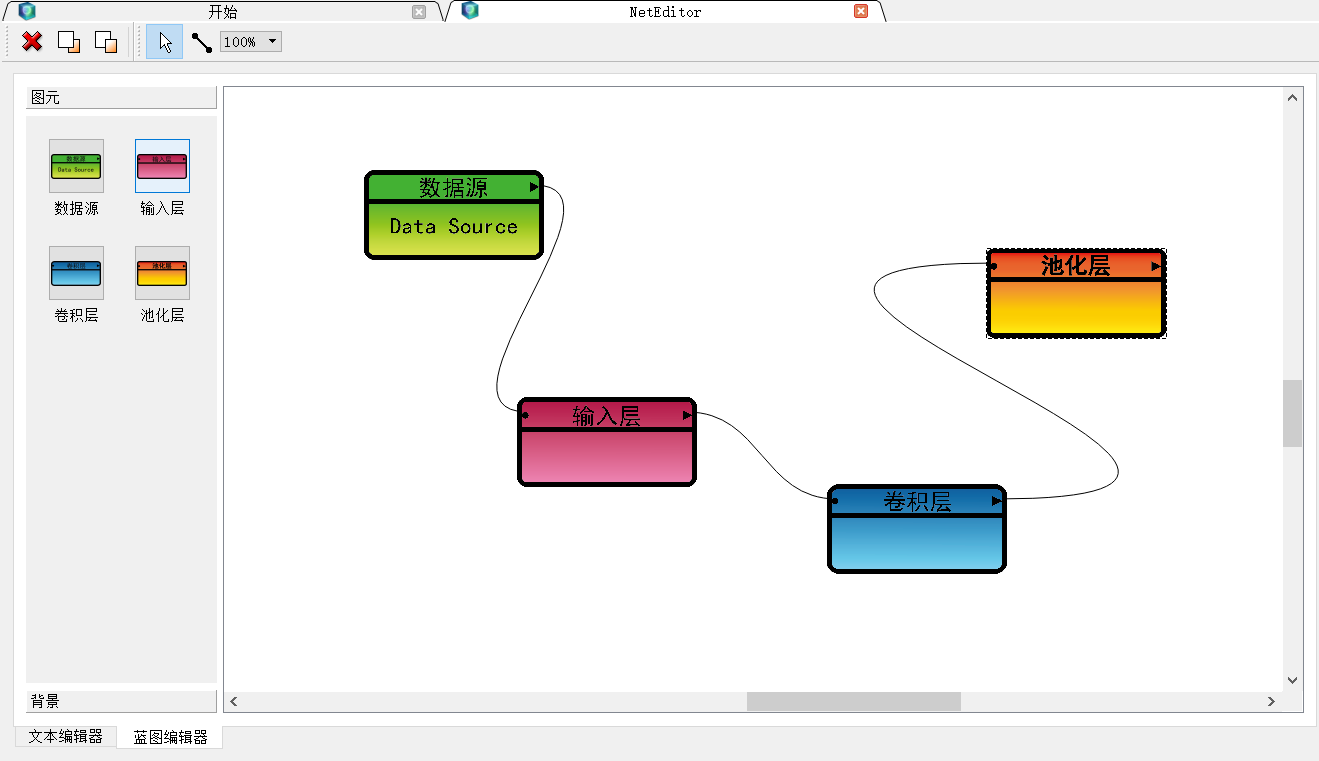


图3-8 蓝图编辑器（2）

第4章　测试和评价

模块测试

按照需求分析所制定大的功能和系统设计所划分的模块对系统进行了模块功能测试，测试结果表明系统的实现已经基本达到了设计要求。

4.3 部署测试

本系统使用Installshield进行打包，并在数台不同的计算机上进行安装测试，参与测试的计算机分别安装了NVIDIA GTX 1060、NVIDIA GTX 1080和AMD显卡，并安装了不同版本的Python环境和CUDA环境，经过对比试验，表明本系统运行的和安装不需要依赖于目标系统上的其他环境，由于已经将所有的依赖库打包，安装到目标计算机上之后，即可立即运行，不需要进行额外的配置。

第5章 总结与展望

5.1 总结

基于Qt的卷积神经网络辅助设计系统采用C++为编程语言，以Qt为图形界面编程框架，以深度学习框架Caffe为核心组件，设计并实现了一款图形化、集成化的卷积神经网络设计环境，通过安装释放，用户可以方便的搭建神经网络开发所需的环境。该系统为神经网络的设计、训练、测试等流程建立了一套工具链，提供了常用的工具，对网络结构的设计进行了优化设计，且支持核心组件的升级替换。通过使用这套环境，可以为相关开发人员节省时间成本。系统基本达成了最初的设计要求。

**参考文献**

[1]Bengio Y. Learning deep architectures for AI[J]. Foundations and trends in Machine Learning, 2009, 2(1): 1-127.

[2]rizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C] Advances in Neural Information Processing Systems. 2012: 1097-1105.

[3]Dahl G E, Yu D, Deng L, et al. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2012, 20(1): 30-42.

[4]Sun Y, Wang X, Tang X. Deep learning face representation from predicting 10,000 classes[C] Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 1891-1898.

[5]Taigman Y, Yang M, Ranzato M A, et al. Deepface: Closing the gap to human-level performance in face verification[C] Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 1701-1708.

[6]Karpathy A, Toderici G, Shetty S, et al. Large-scale video classification with convolutional neural networks[C] Proceedings of theIEEE conference on Computer Vision and Pattern Recognition. 2014: 1725-1732.

[7]Ji S, Xu W, Yang M, et al. 3D convolutional neural networks for human action recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(1): 221-231.

[8]Dong C, Loy C C, He K, et al. Learning a deep convolutional network for image super-resolution[C] European Conference on Computer Vision. Springer International Publishing, 2014: 184-199.

[9]Roth H R, Lu L, Liu J, et al. Improving Computer-Aided Detection Using Convolutional Neural Networks and Random View Aggregation[J]. IEEE Transactions on Medical Imaging, 2016, 35(5): 1170-1181.

[10]Serre T, Kreiman G, Kouh M, et al. A quantitative theory of immediate visual recognition[J]. Progress in brain research, 2007, 165: 33-56.

[11]Hornik K, Stinchcombe M B, White H, et al. Multilayer feedforward networks are universal approximators[J]. Neural Networks, 1989, 2(5): 359-366.

[12]Cun Y L, Boser B E, Denker J S, et al. Handwritten digit recognition with a back-propagation network[C]. Conference and Workshop on Neural Information Processing Systems, 1990: 396-404.

[13]Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C] Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 779-788.

[14]Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks.[J]. Journal of Machine Learning Research, 2010: 249-256

[15]Dahl G E, Sainath T N, Hinton G E, et al. Improving deep neural networks for LVCSR using rectified linear units and dropout[C]. International Conference on Acoustics, Speech, and Signal Processing, 2013: 8609-8613.

[16]Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C] Proceedings of the 27th International Conference on Machine Learning (ICML-10). 2010: 807-814.

[17]Jarrett K, Kavukcuoglu K, Ranzato M, et al. What is the best multi-stage architecture for object recognition[C]. International Conference on Computer Vision, 2009: 2146-2153.

[18]Zeiler M D, Krishnan D, Taylor G W, et al. Deconvolutional networks[C] IEEE Conference on Computer Vision and Pattern Recognition, 2010: 2528-2535.

[19]Zeiler M D, Fergus R. Visualizing and understanding convolutional networks[C] European Conference on Computer Vision. Springer International Publishing, 2014: 818-833.

[20]github: https://github.com/gwding/draw\_convnet，2017-4-25

[21]github: https://github.com/ethereon/netscope，2017-3-1

[22]github: https://github.com/raghakot/keras-vis，2017-3-25

[23]http://auduno.com/post/125362849838/visualizing-googlenet-classes，2017-4-15

[24]http://terencebroad.com/convnetvis/vis.html，2017-4-8

[25]<http://shixialiu.com/publications/cnnvis/demo/，2017-5-15>

[26] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding[C]Proceedings of the 22nd ACM interna

[27] https://www.tensorflow.org,2017-5-19

[28] http://mxnet.io/index.html,2017-5-19

[29] <http://torch.ch,2017-5-19>

[30] http://deeplearning.net/software/theano,2017-5-19

**致 谢**

本论文是在导师王欣老师的细心指导下完成的。导师渊博的专业知识，严谨的治学态度，精益求精的工作作风对我影响深远。本论文从选题到完成，倾注了导师大量的心血。在此谨向导师表示崇高的敬意和衷心的感谢。

本论文写作过程中所使用的场地和设备是由吉林大学计算机科学与技术学院图形学与数字媒体实验室提供的，在此对实验室的老师和学长们致以谢意。

本论文的顺利完成，离不开各位同学和朋友的关心和帮助。在此感谢杨竣翔、徐斌、薛潺涓、孙坤博的帮助，同窗之间的友谊永远长存。