**可视化集成式深度学习开发环境**

目录

**1. 简介1**

**2. 软件的总体结构和核心方法2**

软件的总体结构3

键入章标题(第 3 级)4

核心方法5

键入章标题(第 3 级)6

**3. 案例7**

案例18

案例29

案例310

案例410

案例510

**4. 结论11**

**致谢12**

**参考文献13**

# 1. 简介

[从深度学习切入,讲深度学习工具,然后其局限性,然后我们的工具,最后模仿Caffe论文中那一段写明关于开源社区和基础项目的问题,目前这一部分写的太少了.]

如我们所知,机器学习,尤其是深度学习是近年来的研究热点,被认为是非常有前景和应用价值的人工智能方法.经过多年的研究,深度学习已经拥有了很多理论基础和算法,例如自动求导,梯度下降,ADAM等这样的方法成为了研究者们常用的方法.随着大量的高校研究人员和百度,谷歌,微软这样的大企业进入该领域,深度学习迅速积累起大量的基础算法和诸多行之有效的经验,并在对弈,自动驾驶,搜索,图像处理等领域取得了应用效果.随着Caffe和Tensorflow这样的深度学习框架的推出,深度学习的研究和部署也有了实用的解决方案,他们的出现将研究人员从诸如自动求导这样的基本操作的编程实现中解放出来,使他们能够将精力集中于解决更前沿的问题.根据某大赛的统计目前大量的论文中的方法采用了这些框架.

[参考文献]

尽管我们已经有了这样的工具,但是他们都是控制台程序,虽然借助TensorBoard这样的工具和其他的一些可视化工具powered by python,可以实现一定程度的可视化,但是这些工具本身具有一定的使用门槛,另外,在进行与图像相关的深度学习研究中,对数据进行预处理是很常见的操作,这就需要operator同时必须掌握一定程度的图像处理方法的编程实现.在需要大量实验时,冗余重复的编程工作会拖慢工作进度.

深度学习和与之相关的数据预处理工作具有面向数据流的特点,在较高层次(接近用户)的层次上不适合使用面向对象的编程范式,而适合使用基于函数的过程化编程,这就为使用类似蓝图脚本(in Unreal Engine 4)的Visual Scripting奠定了基础.

本程序是一个完全开源的面向对象的交互式图形软件,设计实现了支持数据读取,预处理,深度学习模型的搭建和可视化等功能的图形化编程环境,可以使用”拖拽”方式进行操作,并支持用户扩展.程序使用C#和python开发,在构思,研究数据预处理,调整网络模型参数和观察网络行为等方面应用本程序,只需要编写少量代码甚至完全不需要编程,可以降低对编程技能的需求,提高工作效率.

程序是在AutoDesk公司Dynamo项目(which is design for参数化建模)的基础上编写的,without its Community,本程序will be nothing,在此感谢Dynamo项目的开发组和所有在(Dynamo连接)上贡献过代码的Contributors.

[文章结构]

本文的其余部分组织如下:第二部分介绍了软件的整体结构,以及特色.第三部分介绍了几个深度学习的例子,以说明该软件对深度学习模型的表现能力并展示可视化表示,第四部分是结论.

(

重点词汇辨析:

1.”图形化编程”,是否会和Visual C++这种所谓的’’可视化编程’’混淆?

2.参考词汇: Visual Scripting/ node-based interface

3.汉语的版本中使用”图形化编程”

)

# 2. 相关工作

Caffe,Tensorflow,的简介,并说明他们的使用特点.根据统计,目前Tensorflow等深度学

框架的使用率非常高.深度学习模型带有模块化的特征,无论多么复杂的深度学习模型,都能分解为卷积层,全连接层,激活函数,损失函数,正则化方法等block,对2013年以来的比较有代表性的深度学习论文的统计和研究发现,在对深度学习方法本身的研究中,除了对已知方法的拆解组合和微调外,还有很多工作提出了新的结构和方法,例如残差网络首次应用跨层链接和残差学习,…首次应用动量方法解决局部最优解等.

数据预处理方面,由于深度学习需要使用大量标注数据,对数据进行预处理和简单分析是开展工作的前提.图像处理是数据预处理中最常用的手段.常用的针对图像进行预处理有分割,缩放,旋转,平移,形态学处理,边缘检测等方法或者这些方法的组合.对论文公开的源代码和深度学习相关的开源项目的不完全统计表明,OpenCV和python是最常用的图像预处理手段.

封装度和易用性的矛盾体现在深度学习工具和图像处理上,为了便于使用并降低学习成本,就要提高封装度,向用户提供高级API,但是这样系统的可扩展性就会下降;如果要向用户提供足够多的选择,那么用户将不得不面对更复杂的编程问题和更高的学习成本.

图形化编程方面,麻省理工学院开发的Scratch可以进行简单的面相过程编程,他的building block对应的是高级语言中的基本语句结构,如变量,语句块等,主要面向青少年教育领域; ArduBlock是开源硬件Arduino的第三方图形化编程工具,他的building block类似于Scratch,添加了一些与硬件对应的例如引脚查询等功能.和这两种类似的图形化编程方法类似于对编程语言的直接翻译,没有提供面向对象功能,且IO和扩展能力有限;BluePrint是虚幻4游戏引擎中的脚本系统,该系统能够实现游戏设计中几乎全部的脚本编程需求,蓝图的building block对应的是高级语言中的方法,变量和对象,其中内置block封装了引擎的API和常用功能,并提供了扩展接口,用户可以使用C++编写自己的building block.蓝图是设计比较完善的图形化编程工具,但是他依托于虚幻,无法应用到游戏以外的其他领域.

[统计数据有参考文献,是一个调查机构写的]

# 3. Node-based 图形化编程模型

由于本文主要面相深度学习和图像数据处理,基于这些任务面相数据(data based)

# 4.Untouchable扩展接口

# 5.应用案例

案例1:使用蓝图进行简单计算,要设计到文件IO和标准输入输出IO

案例2:使用蓝图进行图像处理,实现某篇论文中的图像预处理方法.

案例:3使用蓝图进行神经网络设计,挑选具有代表性的例如AlexNet,残差网络,SSD,RNN等进行设计表示并导出代码进行运行测试,并与直接编写的对应的网络进行性能对比和代码量对比,如果可能,演示从代码转换成蓝图的能力.并说明限制条件

案例4:使用蓝图绘制曲线,曲面,查看卷积核等,此处可以是用一个简单的神经网络或者简单计算进行,

案例5:进行用户自定义扩展,目前有两种扩展:用节点组合成一个复杂节点/基于dll的编扩展.

[所有的案例配截图,简要说明操作,]

# 6.讨论与结论

本文介绍了图形化编程方法在深度学习模型的设计和调试中的应用,有效的使用了代码转换和MVVM设计模式,设计并实现了一种基于节点的图形化编程方法,使用了几个特定的示例来演示功能,所提出的的用户界面已被证明是直观且灵活的,可以通过节点的连接完成深度学习模型的设计,并使用调试工具进行模型的调试,为快速简洁的深度学习应用开发提供了一个有效的工具,他还提供了扩展接口,能够被用户简单的定制为更适合自身开发需要的灵活而强大的平台.

# 致谢

在文章结尾部分的单独部分中，在引用之前整理确认，因此，不要将它们包括在标题页上，作为标题或其他的脚注。这里列出那些在研究期间提供帮助的人（例如，提供语言帮助，写作帮助或者阅读文章的证明等）。  
以这种标准方式列出资金来源：

资助：这项工作得到了国立卫生研究院的资助[grant number xxxx，yyyy]; 西雅图比尔和梅林达·盖茨基金会[授权号码zzzz]; 和美国和平研究所[grant number aaaa]。

没有必要对计划或赠款和奖励的类型进行详细的说明。当拨款来自大学，学院或其他研究机构可获得的整笔拨款或其他资源时，请提供提供资助的机构或组织的名称。

如果没有为研究提供资金，请包括以下句子：

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

# 参考文献

2018年6月2日09:35:30更新:

1. 目前已出现了一款在线软件TensorEditor, <https://www.tensoreditor.com/>,但是他是闭源的,但是他真的能生成tensorflow代码.

2. 可以加一个例子,肺癌检测中形态学处理那8步,引用那篇论文:grt123

3. 要说明程序是基于AutoDesk Dynamo开源软件编写的,其中核心部件是DesignScript.

4. 可以说一下,在深度学习中,进行数据预处理经常是比较麻烦但是也比较重要的,程序的存在可以在构思期间让开发者专注于思路和工作流的搭建而不是具体的编程操作中.

6. Python的绘图功能能不能想办法集成进来

7.考虑加载已经训练好的模型,然后导出某些层的特征并可视化

2018年6月11日10:43:55更新

【根据Apache 2.0开源协议,我可以修改源代码,但是必须注明我对哪些文件进行了修改,发布时必须附带Apache 2.0 Licence,必须保留原有的商标,专利和版权声明,我可以在任何需要的时候修改源代码并将其用于其他开源项目或者商品进行发布甚至销售】