

程序设计实习大作业报告

CodeWings

程序功能介绍

程序是基于 QT 的神经网络可视化与开发项目。

1.网络构建

用户可以选择多种神经网络层类型，可视化搭建自己的神经网络，支持在页面的画布上拖动，网络的计算图由各层在画布上的排列顺序决定。

2.代码生成

随后点击生成代码按钮，就可以在页面下方得到对应的完整 Pytorch 代码。

支持在控制面板上修改用鼠标选中的层的参数。

支持删除所选中的层。

支持点击“复制”按钮复制得到的代码。

3.网络呈现

返回主界面后，可以选择 BlockGenerate 和 NeuronitemGenerate 模式之一，将搭建好的神经网络用相应方式呈现在主界面上，从而能够更直观地查看模块本身数据及模块之间的关联。可根据神经网络类别选择合适的可视化方式表现。

4.历史记录

支持将目前设定保存于历史列表中。

支持在历史中查看已保存的记录，包含保存时间、网络呈现模式、神经网络

层数等信息。

支持对已保存的历史记录进行调取查看（重现于主界面上）操作，同时也可以进行编辑器常见的撤销与重做操作。

5.界面美化

提供多种主界面色彩主题与神经网络线条和方块色彩主题，可自由搭配切换。神经网络线条根据随机权重由粗细表现，可通过重新生成刷新。

6.提供参考资料

在主页面上点击对应按钮，可以查看我们收集的有关深度学习的图片，作为搭建网络的参考。

实现细节

main.cpp:程序入口。

backend.h .cpp：实现 NeuralLayer 类，用于定义各神经网络层。

codegenerator.h .cpp:实现 CodeGenerator 类，用于生成代码。

包括 1.结构 LayerSorter：用于将层按纵坐标排序。

2.函数 generateCodeFromJson：用于将 JSON 数据转化为可解析的网络结构。

3.函数 generatePyTorchCode：生成 Pytorch 代码。

colorthememanager.h .cpp:

包括

1. ColorThemeManager 实现模板存储和记录当前颜色模板
2. struct ColorTheme 记录各组 color 参数

connectionitem.h .cpp:实现线条连接和根据权重决定线粗细

layeritem.h .cpp:对于神经网络生成模块处理

mainwindow.h .cpp .ui:程序主窗口，实现 Mainwindow 类。

包括函数：

1. setupIconButton：创建按钮。
2. setBackground：创建背景图片。
3. on_userGuide_clicked：转到显示资源图像页面，提供软件功能使用参考。
4. on_generateCode_clicked：转到搭建网络的页面。
5. on_generateImage_clicked：生成对应的网络图像

6. on_checkHistory_clicked: 查看已保存的记录。
7. on_startNew_clicked: 重新开始新的网络。
8. on_lastStep_clicked: 撤销上一步操作。
9. on_nextStep_clicked: 重做上一步操作。
10. on_saveCurrent_clicked: 保存当前网络结构。
11. setInstance: 创造实例。
12. showFloatingMessage: 浮动提示框显示操作状态。
13. showWarningMessage: 显示记录保存进程。
14. onHistoryRecordClicked: 显示选择的记录, 重现于主界面上。
15. visualizeNetwork: 可视化呈现网络结构。
16. clearPreviewArea: 清空前后端图像设置。
17. applyTheme: 应用修改后的界面主题。
3. on_generate_code_clicked: 转到搭建网络的页面。
4. on_generate_image_clicked: 生成对应的网络图像。
5. on_history_clicked: 查看网络记录。
6. onHistoryRecordClicked 和 loadHistoryByKey: 加载历史网络结构。
7. showNetworkVisualization: 。
8. on_start_new_clicked: 从新开始新的网络。

- 9. on_turnback_clicked: 转到显示资源图像的页面。
- 10. on_save_clicked: 保存网络记录。
- 11. showFloatingMessage: 浮动提示框显示操作状态。
- 13. showSaveProgressBarMessage: 显示记录保存进程。
- 14. loadNetworkFromJson: 。
- 15. showWarningMessage: 潜在问题提示警告。
- 16. visualizeNetwork: 。
- 17. clearPreviewArea: 清空中间的画布。
- 18. applyTheme: 应用设定的软件色彩主题。

matrial.h .cpp .ui:实现 Matrial 类, 继承自 QMainWindow, 提供资源图片的页面。

neuralnetworkvisualizer.h .cpp:

- 1. createNetwork(const QList<NeuralLayer>& layers): 对于储存的 layer 实现神经网络创建, 以 neuronitemGenerate 的方式实现可视化
- 2.createblockNetwork(const QList<NeuralLayer>& layers): 对于储存的 layer 实现神经网络创建, 以 blockGenerate 的方式实现可视化

3.applyColorTheme(cont QString& themeName):颜色转换

4. QGraphicsItemGroup* createDetailedLayer(const NeuralLayer& layer , int yPos): blockGenerat 的具体实现和细化

5.MovableLayerGroup* createDetailedLayer(const NeuralLayer& layer , int yPos)
实现可移动 block

6. createConnection 创建 block 间连线

7.dragMoveEvent(QDragMoveEvent* event) 允许拖拽

8. dropEvent(QDropEvent* event)放置

9. refreshLayerItem 刷新连线

m_scene 整体图片

m_layers 层

m_dragItem = nullptr;拖拽指针

m_dragStartPos;拖拽位置

QVector<QVector<NeuronItem*>> m_allNeurons: 存储神经元指针以便更新

QList<QGraphicsItemGroup*> m_layerGroups: 存储层指针以便更新

ConnectionLine 连线储存

neuronitem.h .cpp:可视化部分中对于神经元显示的单独处理

addOutgoingConnection/addIncomingConnection 连线存储

itemChange 刷新

codegeneratorwindow.h .cpp .ui:实现 CodeGeneratorWindow 类, 搭建网络的页面。包括以下成员函数

on_return_mainwindow_clicked:返回主页面。

deleteSelectedLayer:删除选中的层。

on_generateCodeButton_clicked:生成代码的按钮。

on_layersList_itemClicked:点击增加画布上的神经网络层。

on_propertiesPanel_parametersUpdated:更新控制面板上的参数。

dropEvent:实现鼠标拖动层的功能。

on_copyCodeButton_clicked:复制代码的按钮

createConnection 创建链接

updateLayerConnections 刷新

clearNetwork:清空网络结构。

onSceneSelectionChanged 参数更新界面部分补充

propertypanel.h .cpp: 实现 PropertyPanel 类, 用于在 CodeGeneratorWindow 实例中实现修改属性的控制面板。

分工情况

姚雨萱：主要负责神经网络可视化 neuroitem 和 block 两种方式的调研、选择、设计、具体实现。可视化部分颜色模板选择以及更新，neuroitem 和 block 拖拽更新实现以及代码界面参数更新。其他文件中和可视化相关部分设置和不同网络层设置参数的补充和 debug。

袁佳逸 主要负责主界面各项操作的实现。具体包括主界面按钮图标排布等各项 ui 设计，以及色彩主题切换、保存及查看记录、前一步和后一步调取及重新开始搭建等软件功能。此外增添提示框、进度条等用户操作时的细节显示，贴近实际软件使用体验。

郝景天 主要负责实现神经网络可视化搭建和展示资源图片的两个页面。具体包括实现网络层的选择，修改参数，拖拽以搭建神经网络层的删除。Pytorch 代码自动生成函数，代码复制功能。设计展示资源图片的页面，搜集一些有关于深度学习的资料。以及录制参与路演选拔的视频。

总结与反思

项目评价

0. 总体上实现了基本功能，达到预期。

1. 负责实现生成图像功能的同学在实现之前做了大量的调查，尝试使用过很多已有的神经网络可视化工具。项目创新方面，这个项目相对于其它神经网络可视化工具，增加了代码生成功能，以及提供了一些诸如提供资料的附加功能。生成图像方面集成了计算图表示与神经元连接表示两种方式，达到了部分此类工具的水平。
2. 实现上一步下一步、保存、新建、生成代码、切换主题背景等多种丰富功能，作为可视化工具使用便利，界面美观。

实现过程中出现的问题以及收获的经验

0. 开工时架构构思还不够完善，造成实现的效率不够高。
1. 初期实现时前后端有些脱节，负责实现后端的同学对前端需求的了解还是不够充分，使得初期写的很多类与函数并没有起到作用。
2. 虽然构想增加与 deepseek 的互动，希望与 AI 课程的大作业相结合。但是未能实现，有些遗憾。也使得项目生成代码的功能的实用性客观上有所降低。
3. 生成图像功能碍于具体实现难易未能针对多种神经网络实现最合适的可视化效果，对除 fcnn 外的可视化实现方式有些笼统
4. 增加了对编写 ui 以及 QT 使用的的经验，对课堂上学到的知识起到了很好的巩固作用。
5. 获得了宝贵的团队开发项目的经验，收益匪浅。