

AIGENT_JS

Contributers

Phan Trung Duc : Backend developer

Nguyen Dinh Ngoc Tri: Frontend developer

Le Truong Tho: Frontend developer

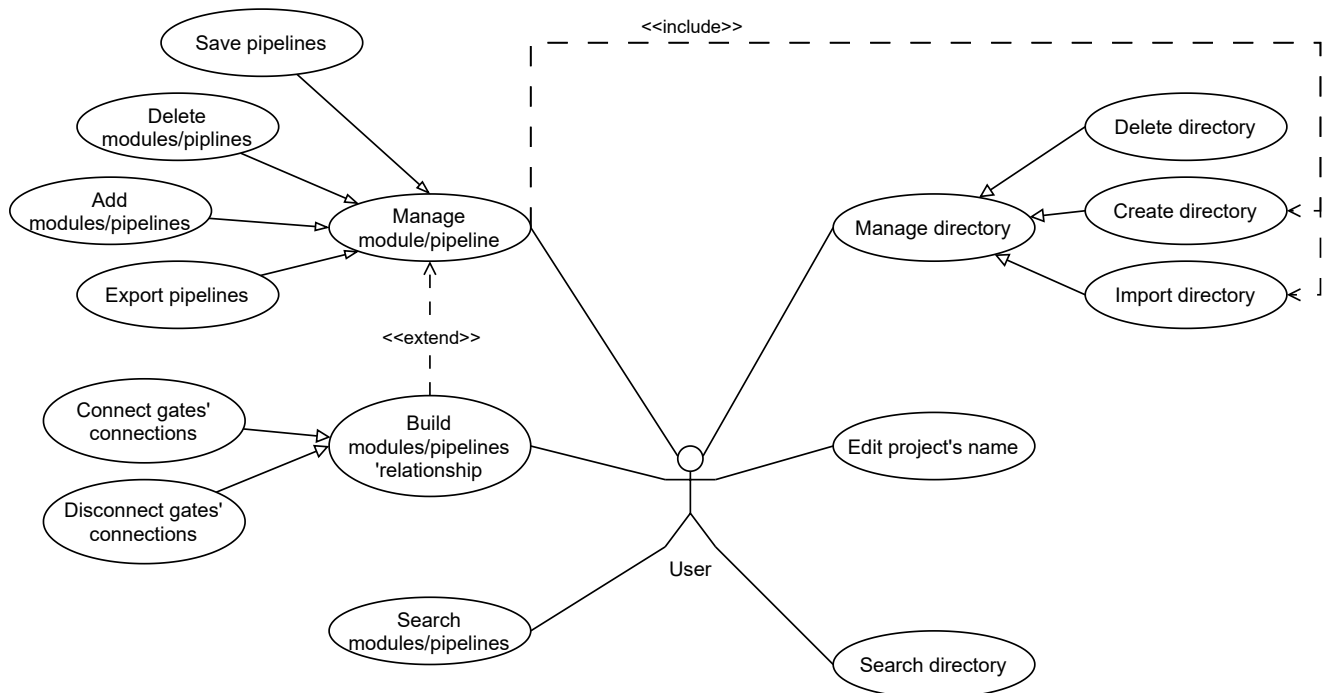
Introduction

AIGENaTor is an application designed for creating AI or programs in general with drag-and-drop method. This application is currently created for educational purpose, so the performance of programs created with this application might have quite bad performance.

Problem statement

AI and Machine Learning (ML) have rapidly evolving recently and become a common and useful tool in our daily life. However, these technologies seem to be quite complicated, especially for newbie. We suffer these problems ourselves. Thus, we decide to implement an application to create AI pipelines so that everyone can take use of the basic level of AI as well as visualize them. We hope AIGENT can be a useful tool to make learning, building, and using AI become an easy and enjoyable task for everyone.

Usecase Diagram



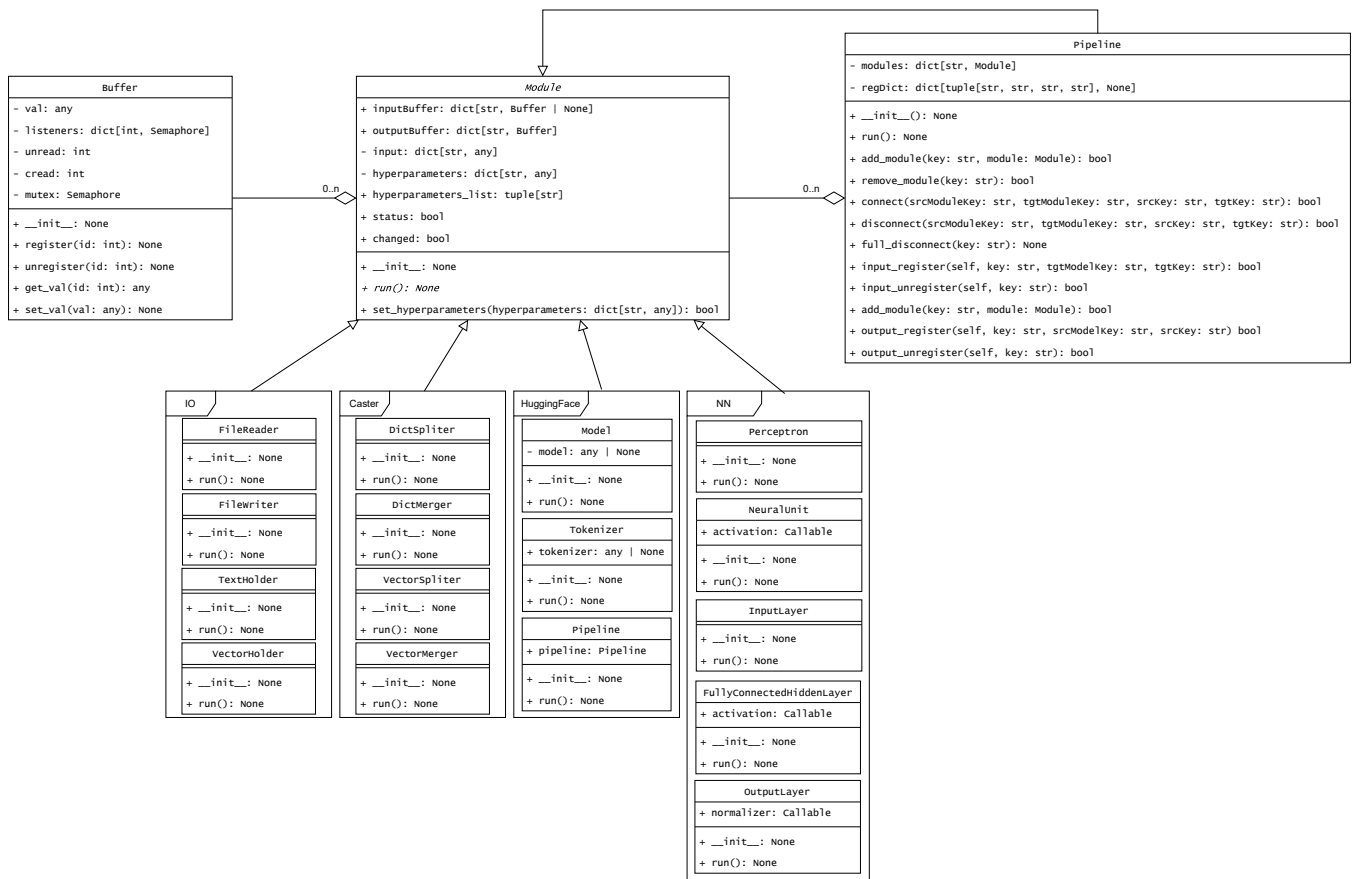
The diagram visually depicts:

User Interactions: How users (actors) can interact with the system to manage pipelines.

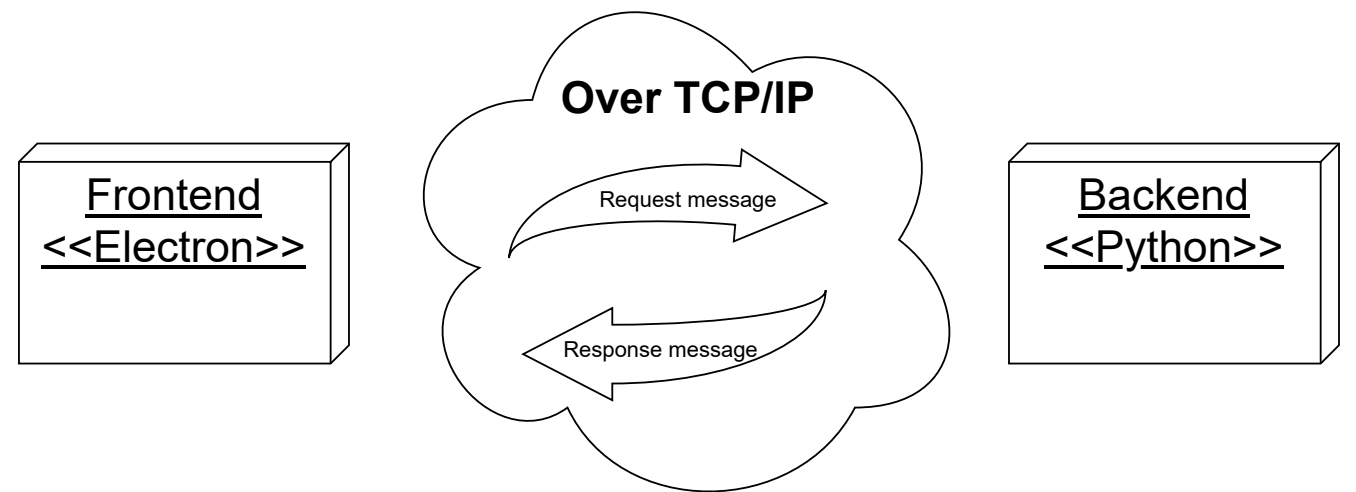
Functionalities: Key functionalities available to users, including creating directories, searching pipelines, managing projects, building/deleting pipelines, and exporting/importing data.

Relationships: Connections between functionalities, such as nested actions and optional extensions. This use case diagram provides a clear understanding of user interaction capabilities within the pipeline management system.

Class Diagram



Architecture



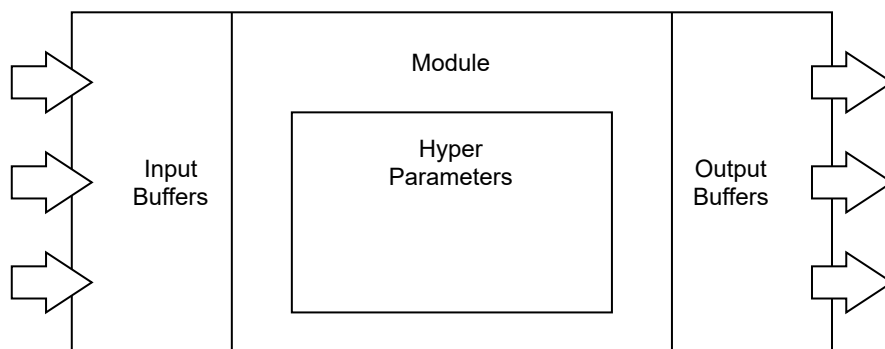
Our program consist of two parts, which can work independently, frontend and backend. While backend is implemented with Python, frontend is coded with Electron, a popular Node.js framework for desktop development that have web like syntax - which not seem to help at all. These two parts communicate with each other purely by json strings sended through TCP/IP sockets.

Backend

To create a drag-and-drop AI pipelines, we split them into modules, which is anything can work independently with some given input and return some output, do something, or both.

Modules

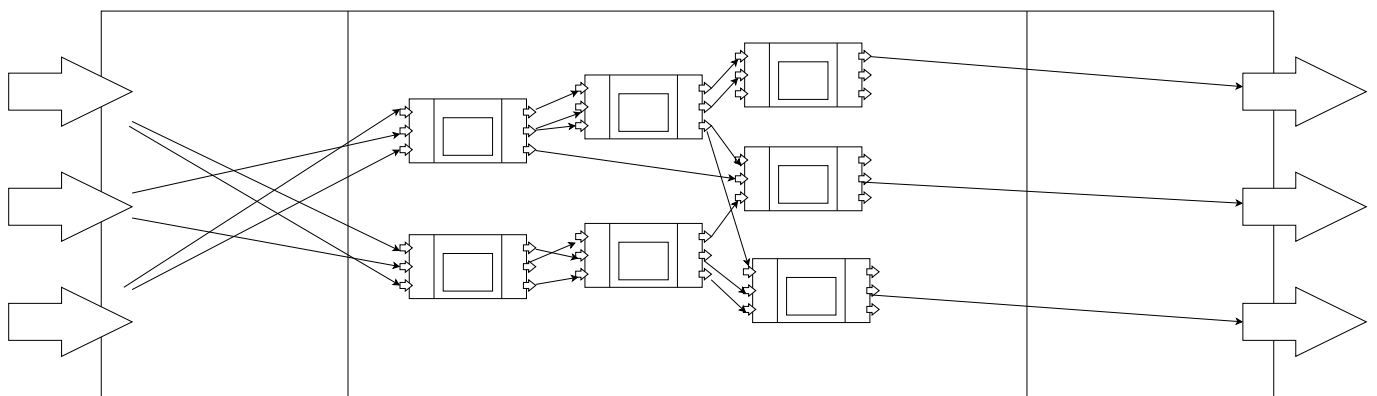
Modules consist of three main components: output buffer, which is designed to hold and maintain value for asynchronous accesses; input buffer, which holds pointers to the correspond source: an output buffer of another module; and a run method, define how a module compute, act, and map input value to output value. This design allows modules to work all by themselves, provides good modularity.



Our backend is designed to allow developers to easily scale built-in modules. Just implement your own module following the implemented modules we provide in to the available files or a new python file and place it in python folder and it will work fine. However, our frontend has not support this feature yet due to difficulties in file reading and text processing.

Pipeline

To make it easier for users to build bigger and bigger applications, we implement a Pipeline as if it is a module. Pipelines inherit all nature of modules with additional information about the modules that the pipeline has. Pipelines run by activating all modules and letting them run in multi-thread. This allows us to maximize the flexibility of the pipeline with a simple design. However, this design heavily affects the performance of the pipeline, especially for feed-forward pipelines consisting of many modules.



A pipeline can be saved, and because pipelines are also modules, it can be added into other pipelines. Hence, our program allows users to build up large pipeline by gradually building smaller pipelines.

Frontend

Our program features a user-friendly frontend comprising two main pages: the Homepage and the Workspace.

Homepage



The Homepage serves as the control center for directories, which contain the user interface, backend, and connections of the modules. On this page, users have access to several essential functions:

Create New Directory: Users can create a new directory to navigate to the Workspace page so as to start organizing their modules and workflows.

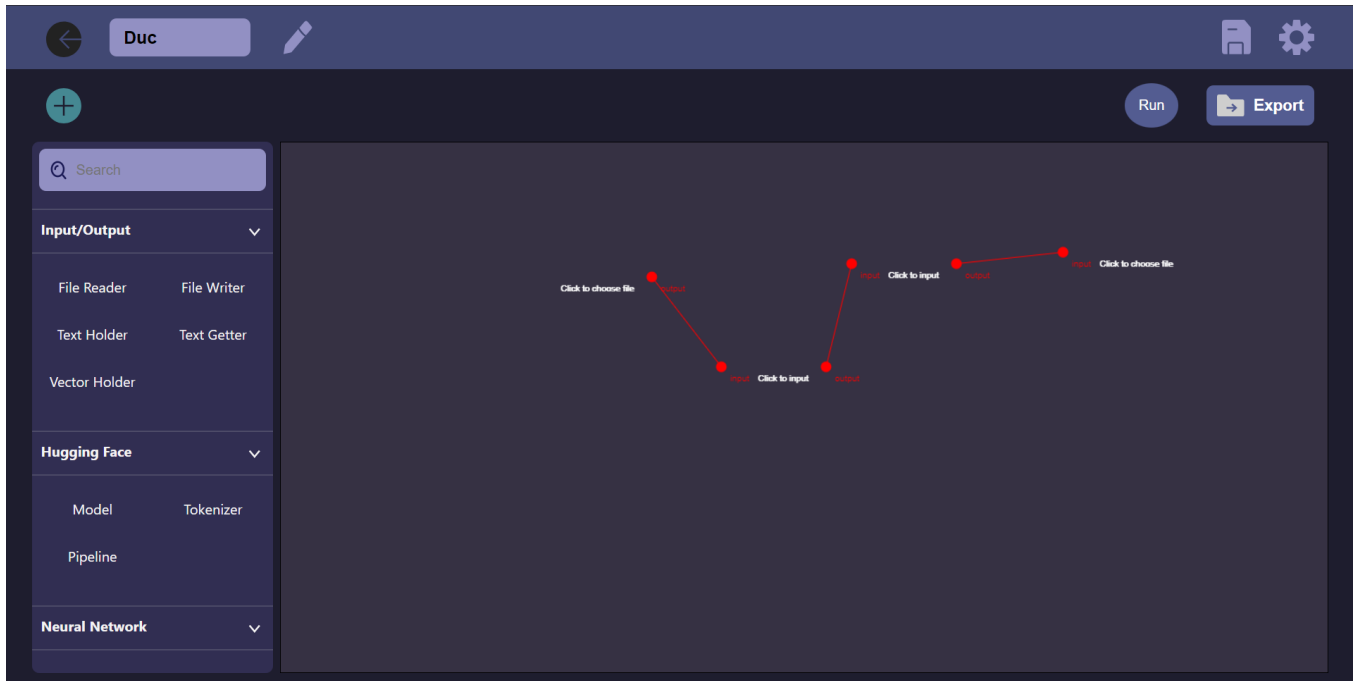
Search Directory: A search function allows users to quickly find existing directories.

Delete Directory: Users can remove directories they no longer need.

Import Directory: This special feature enables users to import their directories, provided they adhere to the program's format requirements.

Access Workspace: Users can click on an existing directory to navigate to the Workspace page, where they can edit pipelines, modules, and connections between gates within that directory.

Workspace



The Workspace is where the core module interactions and pipeline creations occur. It features a menu that provides access to both pre-built modules and user-created modules. Key functionalities on this page include:

Drag and Drop Modules: Users can easily drag and drop modules from the menu into the workspace.

Connect Gates: Users can connect the gates of various modules to form a larger, interconnected pipeline.

Set Hyperparameters: For added flexibility, users can click on individual modules to set hyperparameters, tailoring each module's behavior to their specific needs.

Delete Connections and Modules: Within the workspace, users can delete connections between gates and remove modules that are no longer needed, ensuring the workspace remains clean and efficient.

Export and Save: The Workspace also offers functions to export the entire pipeline in a predefined format and save the current state of the pipeline for future use.

Key Features

- 1. Drag and Drop Modules:** Our platform offers an intuitive drag-and-drop interface that allows users to seamlessly add modules to their workspace. Users can effortlessly move these modules around the workspace, arranging them in any configuration that suits their workflow.
- 2. Zoom and Pan:** To enhance user experience, the workspace supports zooming in and out, providing users with the flexibility to focus on specific modules or get an overview of the entire project. Additionally, users can pan across the workspace to navigate their modules easily.
- 3. Flexible Module Positioning:** Users can position modules anywhere within the workspace, giving them complete control over the layout. This feature ensures that users can organize their workspace in the most efficient and personalized manner.
- 4. Connectable Gates:** Each module comes equipped with gates that can be connected to gates of other modules. This allows users to create complex workflows by linking modules together to perform

specific tasks or functions. The connectivity between modules enables the execution of customized operations tailored to the user's requirements.

5. **Hyperparameter Setting:** Users can click on modules to set hyperparameters, allowing for precise control and customization of each module's behavior. This feature enables users to fine-tune their modules to achieve optimal performance for their specific tasks.

How to Use the Program

Starting on the Homepage:

1. **First Time Setup:**

Click the "Add New" button to create a new directory. This will take you to the next page.

2. **Editing an Existing Directory:**

If you already have a directory and want to edit it, click on the directory frame to move to the Workspace page.

On the Workspace Page:

1. **Editing Project Name:**

To edit the project name, click the edit icon and enter the new name.

2. **Adding Modules:**

Click and hold a module from the side menu. Drag it to the workspace and release the mouse button to place it.

3. **Zooming and Panning:**

Use the scroll wheel on your mouse to zoom in and out of the workspace.

4. **Connecting Gates:**

Each module has a different number of input and output gates.

To connect gates: Hover the cursor over a gate until it changes to a hand icon. Click and hold the gate, drag to another gate (which will also show a hand icon), and release the mouse button to create the connection.

5. **Deleting Connections and Modules:**

To delete a connection: Right-click on the connection line. To delete a module: Right-click on the module (ensure the hand icon is over the module before right-clicking).

6. **Saving and Running the Project:**

To save your work, click the save icon. To run the project and see the output, click the run button. To export the project as a JSON file, click the export button.