

Project Name/Title: TBA

Developers: Aaron Tian, Simon Beyzerov

Project Description + Goals

The goal of this project is to develop a graphical user interface (GUI) for machine learning model development. The software is primarily intended to streamline the process of machine learning development for experienced individuals by reducing boilerplate code, though it may also serve as an educational tool for students as well.

Functional Description

The criteria for the project are described as follows.

highlighted criteria shall be included in the minimum viable product (MVP)

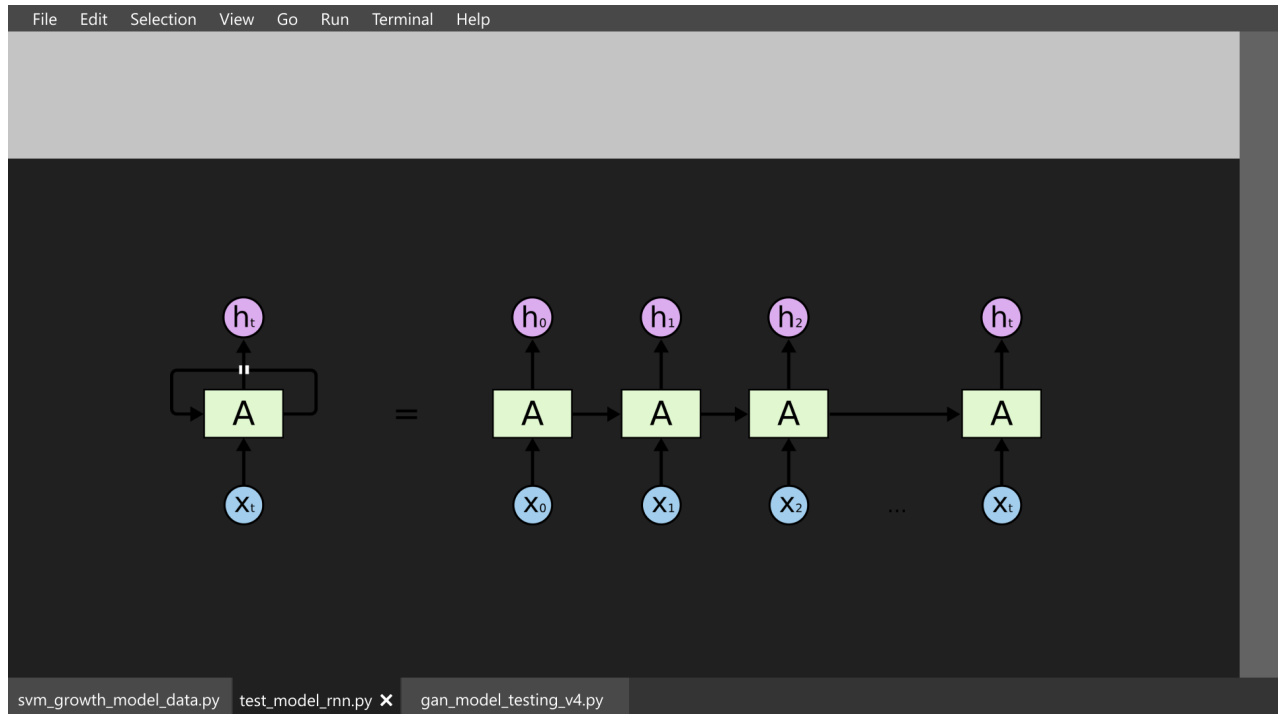
1. The product will have a graphical user interface that displays relevant information to the following subtasks:
 - a. **Data**
 - i. Access to a collection of test datasets (MNIST, CIFAR, IMDB, ArXiv, etc.)
 - ii. Allows users to access locally stored files in the system
 1. Automatically compile data based on filetype (csv, xls, etc.)
 - iii. Allows users to preprocess data (normalization, batching)
 - b. **Model**
 - i. Model creation with intuitive (drag and drop?) interface.
 1. The application will be a wrapper for PyTorch and pytorch_geometric, so all available nn layers in those libraries will carry over.
 - ii. Allows users to export a diagram of the model architecture in a graphical format (png, jpg, svg, pdf, etc.)
 - c. **Train**
 - i. Set initial hyperparameters and train model with button click.
 - ii. Supports visual representation of training progress
 1. Live loss graphs, etc..
 - d. **Test**
 - i. Allows users to import test samples to run the model on.
 - e. **Hyperparameter Tuning**
 - i. Graphical interface displaying specific metrics (loss per epoch, accuracy per epoch, etc)

- ii. Automated hyperparameter tuning methods - random search, grid search.
 - 1. Display performance heatmaps
 - iii. Automatically select the best hyperparameter configuration based on tests and export to the 'train' tab.
 - f. **Analyze**
 - i. In-depth analysis of model with cross-validation
2. Product is lightweight and does not compete with ML model for space in memory
 - a. Lighter models can be run and built on limited hardware resources (low memory, slow CPU, no GPU, etc.)
 3. Product requires minimal setup and accelerates workflow without restricting access to custom aspects of the machine learning library.

Technical and Data Feasibility

The program will be run on a local application on the user's Windows machine (pending future development). Use python/existing PyTorch blah blah blah ---- git? github?

User Interface



(sketch! Not final or representative of final product)

Persistent Storage

Local storage will be supported and data will persist on the user's device/hard drive. Online and cloud-based capabilities are of independent interest and may allow users to share their models with other users or support collaborative projects. Built-in *Git* support is also of interest as an alternative in maintaining collaborative project environments.