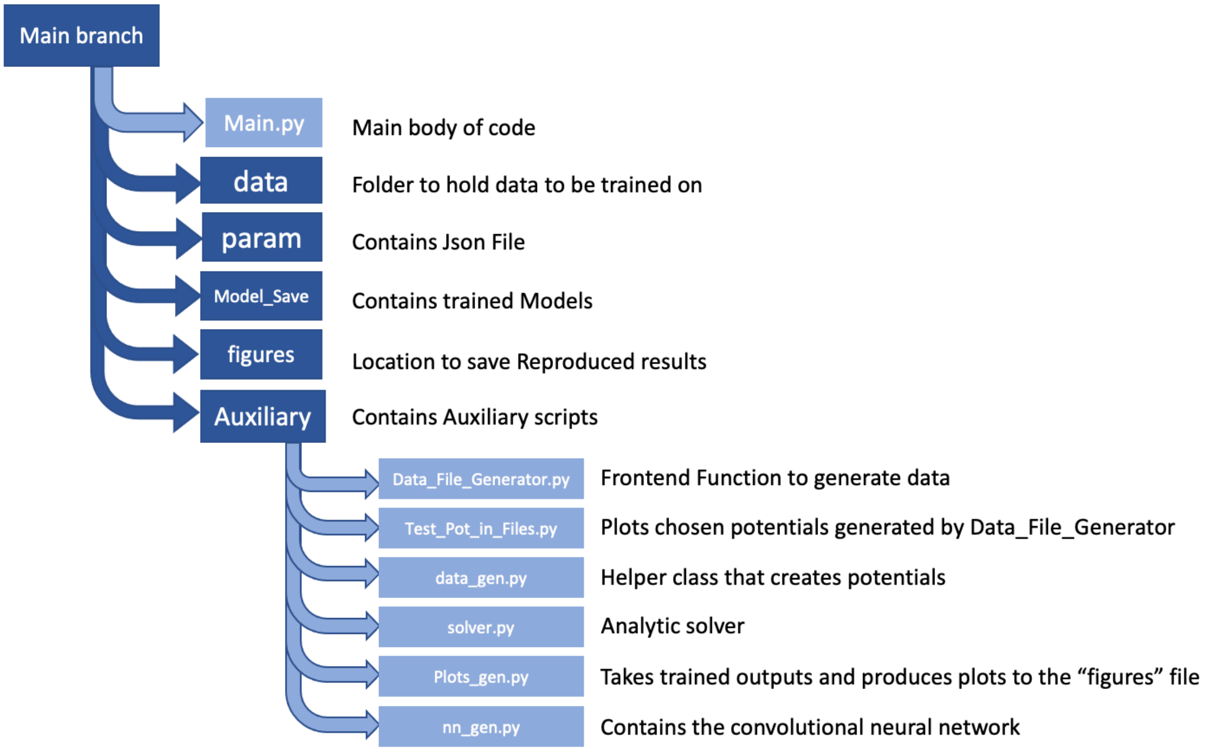
Repository Guide

This is a user guide to the functions a user may need to interact with.

Github Flowchart



Main branch	
Main.py	Main body of code
data	Folder to hold data to be trained on
param	Contains Json File
Model_Save	Contains trained Models
figures	Location to save Reproduced results
Auxiliary	Contains Auxiliary scripts
Data_File_Generator.py	Frontend Function to generate data
Test_Pot_in_Files.py	Plots chosen potentials generated by Data_File_Generator
data_gen.py	Helper class that creates potentials
solver.py	Analytic solver
Plots_gen.py	Takes trained outputs and produces plots to the “figures” file
nn_gen.py	Contains the convolutional neural network

The Main Function.

Usage:

1. The main function is run through command line with the following call:
   - Python3 Main.py
2. Once called, a GUI will ask the user to select a folder containing all data that the user wishes to train or test on.
   - All files ending with “.npz” within the designated folder will be taken as input.
   - To train, set “train” in the corresponding Json file to True, else set to False. Similarly, to test (No learning) set “test” in the corresponding Json file to True, else set to False.,
   - Should both “train” and test” be given as True, the code will first ask for a folder containing the data to train, then once training is completed, ask for a DIFFERENT folder containing data to test on.
3. If Test is set to “True”, both the epoch average loss and the performance of the trained model on a chosen test dataset will be exported to the designated file location. These will be in text form. Plots will NOT be generated.
4. Upon completion, the runtime of the code will be returned.

Other Script Dependencies:

- nn_gen.py

Json File

The main function requires additional arguments, given through a json file labeled “param_filename.json” located in the “param” file. Additional arguments include:

Label	Data type	Description
files_per_batch	Int	Should be left as 1 - How many files to use per batch
learning_rate	float	Learning rate for Adam optimizer
num_epochs	Int	Number of epochs to train on
num_epochs_display	Int	How many epochs do you want displayed updates on
Verbosity	Int	1,2 or 3. Controls output printing verbosity
model_load_path	String	Path to and name of file where the trained weights will be loaded from
model_save_path	String	path to and name of file where the trained weights will be stored
loss_save_path	String (includes .txt)	path to and name of the file to save the loss
performance_save_path	String (includes .txt)	path to and name of the file to save the testing performance of the network on testing data
train	Boolean	To turn off or on training
test	Boolean	To turn off or on testing

Data_File_Generator.py

Description

Used to generate potentials to be run in the main function

Usage:

1. This function is run through command line with the following calls:

Argument	Data type	Default	Description
--num_files	Int	~	Number of Files to be written
--image_size	Int	128	The image size of the potentials. Example : 128
--num_sho	Int	0	Number of SHO potentials to be written
--num_iw	Int	0	Number of IW potentials to be written
--num_dig	Int	0	Number of DIG potentials to be written
--num_rnd	Int	0	Number of RND potentials to be written
--dataset_type	String	~	Create Training or Test Dataset Must be either: Train or Test
-- verbosity	Int	3	1,2 or 3. Controls output printing verbosity

2. Once started, the code will ask for a folder to save the files into. Typical folder is the “data” folder, within a new subfolder named accordingly. I.E “data/train_data”
3. Files will be saved with the following nomenclature:
   - fileNameStart_ImageSize_itterater_potType #potential
4. Runtime will be printed once code is finished execution

Other Script Dependencies:

- Data_gen.py
- Solver.py

Test_Pot_In_Files.py

Description

A simple function to visualize the plots that were generated by the Data_File_Generator.py function

Usage:

1. This function is run through command line with the following call:
   - Test_Pot_In_Files.py
2. Once called, a GUI will ask the user to select a folder containing all data that the user wishes to plot.
   - All files ending with ".npz" within the designated folder will be taken as input.
   - A maximum of 20 potentials within this folder will be plotted.
     - The first 20 potentials to appear in the folder will be plotted.
3. The user can cancel a run of the code by clicking Ctrl + C.