**— Training ReadME**

**—**This document serves as the foundational guide to using the Training repository. Specifically it includes the resources necessary to train your own version of the Prime Number MLP or play around with the ML cpp based library.

Upon viewing the repo there are several major components that makeup the MLP itself, first, the utilities.cpp/.h are one of the backbone components that assist in file/weight management. The layers.cpp/.h include each individual layer utilized in the architecture, if you wanted to create your own architecture, I’d suggest digging into this file to understand the format. The activate.cpp/.h are pretty straight forward, it holds the necessary activation functions for the model to utilize, there are a few different functions made, however it can surely be expanded upon. Last but not least there is the MLP.cpp/.h, this is the central component that ties in all of the pieces and constructs the model with each layer, weight retrieving methods, forward, and prediction behavior defined.

-Getting yourself, I’d suggest utilizing Visual Studio as this is my native environment, but it should be fine to run in any C/C++ supported IDE.

-After downloading the repo, and opening it in your suited environment, navigate to the main.cpp script.

-This is the top level component that orchestrates the training for the MLP class, it follows a familiar flow not too dissimilar from pytorch/tensorflow… the syntax of course, but a tradeoff for computation.

-If you wanted to run training, ensure the other testbenches that are included are disabled, as there can only be one main at a time before you build-compile-run, the testbenches are to ensure functionality, however if you needed or wanted to confirm a functional component just run any one of these testbenches along with whatever source file that is tied to that testbench.