API Guide

There are four major components or phases to solving a machine learning program using this system: the Preprocessor, the Compiler, the Interpreter, and the Weight Evaluation Phase (which uses the interpreter). This document will describe the arguments and syntax for executing each of the four phases. For implementation details, see the header source files.

The Preprocessor

The Preprocessor converts one TenFlang program into another expanded program. The Preprocessor's *main* method requires two arguments, the name of the original program's file, and the name of the file to which the expanded program will be written. The Preprocessor can be run as follows (after "*make preprocessor"* or simply "*make"* has been run):

```
$     ./preprocessor my_program.tf my_expanded_program.tf
```

The Compiler

The Compiler converts one TenFlang program, a Shape Program, into another program, the GCP (Gradient Computing Program). If the user's Shape Program has already been pre-processed, the Compiler's *main* method requires only two arguments, the name of the Shape Program's file, and the name of the file to which the GCP will be written. The Compiler can be run as follows (after "*make compiler*" or simply "*make*" has been run):

```
$     ./compiler my_expanded_shape_program.tf my_gcp.tf
```

If the user's Shape Program has **not** already been pre-processed, the Compiler's *main* method requires an additional argument, the name of the file to which the expanded program will written. (It is the expanded program that is actually interpreted.) The user can specify that his Shape Program has not yet been pre-processed by using the "*-pp*" flag. The Compiler can be run as follows (after "*make compiler*" or simply "*make*" has been run):

```
$     ./compiler my_shape_program.tf my_gcp.tf -pp
my_expanded_shape_program.tf
```

The Interpreter

The Interpreter runs a TenFlang program, reading the program's inputs from a file of input name-value pairs. If the given program has already been pre-processed, the Interpreter's *main* method requires only two arguments, the name of the program to be interpreted, and the name of the input file. The Interpreter can be run as follows (after "*make interpreter*" or simply "*make*" has been run):

```
$     ./interpreter my_program.tf my_inputs.txt
```

If the given program has **not** already been pre-processed, the Interpreter's *main* method requires an additional argument, the name of the file to which the expanded program will be written. (It is the expanded program that is actually interpreted.) The user can specify that his program has not yet been pre-processed by using the "*-pp*" flag. The Interpreter can be run as follows (after "*make interpreter*" or simply "*make*" has been run):

```
$     ./interpreter my_program.tf my_inputs.txt -pp
my_expanded_program.tf
```

The input file passed to the interpreter is a series of lines of the format:
`<input variable name> <input value>`
It is important that each input variable has its own line, and that the name and value are separated by a tab.

The Weight Evaluation Phase

The documentation for the Weight Evaluation Phase is still in progress.