
Appendix B [Unix]

Introduction

If you are using the Unix, command line version of `tlearn`, you will need to make several adjustments to the exercises and explanations given in the text (which assumes a Macintosh version). The reference manual for the line-oriented version is included here, and we give several examples which illustrate the differences in how the simulator is invoked. There is also one important difference in file formats (the `.teach` file); other files and options are identical across platforms, however.

Configuration Files

The four input files which are described in Appendix A are also used in the Unix, command line version of `tlearn`. With one exception, these files are identical. (Rather than preparing these files within the simulator itself, using `tlearn`'s built-in editor, you would prepare these with a Unix text editor such as `vi` or `emacs`.)

The one input file which differs from the Mac version is the `teach` file. In the Macintosh version of `tlearn`, this file begins with the word “localist” or “distributed”, followed by a line with the number of target patterns, followed by successive lines, each of which has the target pattern for the corresponding input pattern found in the `data` file. In the Unix version, the first two lines are the same as in the Mac version, but the actual patterns must be preceded (on the

same line) by a number which tags that specific target pattern. Patterns are numbered incrementally starting with pattern 0 (zero). The following illustrates the differences between for the XOR problem. Note that in the Unix version, each line now has two numbers; the first is the number of the pattern and what follows is the actual target.

teach files for the XOR problem

Mac version	Unix Version	
distributed	distributed	
4	4	
0	0	0
0	1	0
1	2	1
1	3	1

Invoking the simulator

To invoke **tlearn** on Unix, you give it as a shell command, followed by a list of the options and file names as required (these are documented in the attached Unix on-line manual page).

For example, to run a training session on the XOR problem, we might do the following (assuming % is your shell prompt, and you have used `xor` as the fileroot for your files).

```
% tlearn -f xor -s 5000 -R -r 0.3 -m 0.9 -E 100
```

(If you wish to run this command asynchronously, in the background, add an ampersand (&) to the end of the command-line.)

This command will run the simulator for 5000 training patterns, chosen at random, with a learning rate of 0.3 and momentum of 0.9, and will save the error every 100 patterns in a file called `xor.err`. It will also generate a weights file called `xor.5000.wts`, which can

then be loaded in to **tlearn** on subsequent invocations either in order to test learning, or to continue with additional learning sweeps.

```
% tlearn -f xor -l xor.5000.wts -s 4 -V
```

for testing (“verifying”; notice that there is no **-R** flag, so the 4 patterns will be presented in the order in which they occur in the **data** and **teach** files) or

```
% tlearn -f xor -l xor.5000.wts -s 1000 -r 0.3 -m 0.9 -R
```

to continue training for an additional 1000 sweeps. This would yield an additional weights file called `xor.6000.wts`.

Additional files

In the exercises as described in this handbook, it is assumed that you have access to various data files for many of the exercises. In the Mac version, these are located in folders which come with the simulator. In the Unix version, they have been placed in directories named `Chapter1`, `Chapter2`, etc., within the following directory:

```
~elman/cogs201f/exercises/
```

on various machines (the tilde denotes the path to user `elman`’s home directory; the actual location of this directory varies from machine to machine, so the tilde will find the correct location on any machine on which `elman` has an account).

