CmdLine

This class provides a parameter parsing framework which is very useful when one needs to run several experiments that rely on different parameter settings that are passed in the command line. This class will also override the default print function to direct all the output to a log file as well as screen at the same time.

A sample `lua` file is given below that makes use of `CmdLine` class.

```
cmd = torch.CmdLine()
cmd:text()
cmd:text()
cmd:text('Training a simple network')
cmd:text()
cmd:text('Options')
cmd:option('-seed',123,'initial random seed')
cmd:option('-booloption',false,'boolean option')
cmd:option('-stroption','mystring','string option')
cmd:text()

-- parse input params
params = cmd:parse(arg)

params.rundir = cmd:string('experiment', params, {dir=true})
paths.mkdir(params.rundir)

-- create log file
cmd:log(params.rundir .. '/log', params)
```

When this file is run on the th command line as follows

```
# th myscript.lua
```

It will produce the following output:

```
[program started on Tue Jan 10 15:33:49 2012]
```

```
[command line arguments]
booloption  false
seed    123
rundir  experiment
stroption   mystring
[----------------------]
booloption  false
seed    123
rundir  experiment
stroption   mystring
```

The same output will also be written to file `experiment/log`. Whenever one of the options are passed on the command line and is different than the default value, the `rundir` is name is produced to reflect the parameter setting.

```
# th myscript.lua -seed 456 -stroption mycustomstring
```

This will produce the following output:

```
[program started on Tue Jan 10 15:36:55 2012]
[command line arguments]
booloption  false
seed    456
rundir  experiment,seed=456,stroption=mycustomstring
stroption   mycustomstring
[----------------------]
booloption  false
seed    456
rundir  experiment,seed=456,stroption=mycustomstring
stroption   mycustomstring
```

and the output will be logged in `experiment,seed=456,stroption=mycustomstring/log`

addTime([name] [,format])

Adds a prefix to every line in the log file with the date/time in the given format with an optional name argument. The date/time format is the same as `os.date()`. Note that the prefix is only added to the

log file, not the screen output. The default value for name is empty and the default format is '%F %T'.

The final produced output for the following command is:

```
> cmd:addTime('your project name','%F %T')
> print('Your log message')
```

```
2012-02-07 08:21:56[your project name]: Your log message
```

log(filename, parameter_table)

It sets the log filename to `filename` and prints the values of parameters in the `parameter_table`. If filename is an open file descriptor, it will write to the file instead of creating a new one.

option(name, default, help)

Stores an option argument. The name should always start with '-'.

[table] parse(arg)

Parses a given table, `arg` is by default the argument table that is created by `lua` using the command line arguments passed to the executable. Returns a table of option values.

silent()

Silences the output to standard output. The only output is written to the log file.

[string] string(prefix, params, ignore)

Returns a string representation of the options by concatenating the non-default options. `ignore` is a table `{dir=true}`, which will ensure that option named `dir` will be ignored while creating the string representation.

This function is useful for creating unique experiment directories that depend on the parameter settings.

text(string)

Logs a custom text message.