

# MLP user's guide

June 12, 2024

## Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introduction</b>                                  | <b>3</b> |
| <b>2</b> | <b>Architecture of the software</b>                  | <b>3</b> |
| <b>3</b> | <b>Solving multi-class problems</b>                  | <b>3</b> |
| 3.1      | Simple examples . . . . .                            | 3        |
| 3.2      | Structure of the files containing the data . . . . . | 3        |
| 3.3      | Training and testing the MLP . . . . .               | 4        |
| <b>4</b> | <b>General comments</b>                              | <b>5</b> |

## 1 Introduction

This application is an implementation of the multi-layer perceptron (MLP) [1]. The program is written in C ANSI, and thus can be used under the various releases of UNIX, Linux, IRIX, etc.

## 2 Architecture of the software

This application is made up of one single program, named `mlp`, which can be used both for training and in test. It can be compiled thanks to the command:

```
compile_mlp
```

(the corresponding makefile is in the subdirectory `make`).

## 3 Solving multi-class problems

### 3.1 Simple examples

A simple way to become familiar with the use of the software consists in running it on some of the benchmarks provided. They include the nine data sets used to evaluate the *isotropic kernel machine* (IKM) [2]. In order to select any of them, it suffices to use the corresponding script, named `configure.name`, where `name` is the name of the data set as used in the directory `Data` (`arcene`, `bull`, `car`, ...). Once this is done, the files `Fichcom/train_mlp.com` and `Fichcom/eval_mlp.com` (see below) contain the appropriate parameters. Suffice it to use the commands `execute_train_mlp` and `execute_eval_mlp` to start training and evaluate the network respectively.

### 3.2 Structure of the files containing the data

The files containing the data must be text files, with a specific structure. We illustrate this structure on one of the five training sets of the "Car Evaluation" problem. The name of the corresponding file is `Data/car0.app`.

1383  $\leftarrow$  number of points in the set

6  $\leftarrow$  number of components of the vectors coding the input data (descriptions)

4  $\leftarrow$  number of categories

2.0 2.0 4.0 4.0 2.0 2.0 2 ← description of the first example example plus the label of its category (here 2).  
 1.0 2.0 5.0 4.0 0.0 2.0 2 ← description of the second example plus the label of its category  
 ...

### 3.3 Training and testing the MLP

Training is initiated with the command

```
execute_train_mlp
```

In order to specify the nature of the problem to be solved, the file

```
Fichcom/train_mlp.com
```

must preliminary be filled. It is made up of ten lines. Its structure, illustrated on the aforementioned "Car Evaluation" problem, is as follows:

```
on ← 'o': mlp used for training, 'n': weights initialized randomly ('o'
would correspond to a warm start)
4 ← number of categories
Data/car.app ← name of the file where the training set is stored
matrix/car.mat ← name of the file containing the initial values of the
weights
matrix/car.mat ← name of the file where the new values of the weights
will be stored
5 ← number of units in the hidden layer
0.0001 ← initial value of the gradient step
0.01 ← value governing the intervals for the random initialization of
the weights
10000 ← number of epochs
Data/car.output ← file where the output of the network will be stored
(used in test only)
```

Testing is initiated with the command

```
execute_eval_mlp
```

The structure of the file

```
Fichcom/eval_mlp.com
```

containing the parameters used to test the MLP is the same as the structure of the file `Fichcom/train_mlp.com`. The two files differ on two lines only. The first character of `Fichcom/eval_mlp.com` is 'n' (instead of 'o'). Since

it is of little interest to test a network which is not trained, the second character should logically be 'o'. A priori, the third line of the file should be the name of a file containing a test set (although it is utterly possible to test the training performance).

## 4 General comments

Please, feel free to report any suggestions you could have to improve the program or this document to the following address: `Yann.Guermeur@cnrs.fr`.

## Acknowledgments

## References

- [1] M. Anthony and P.L. Bartlett. *Artificial Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.
- [2] Y. Guermeur and N. Wicker. Isotropic kernel machine. 2024. (in revision).