

MLP for protein sequence processing

User's guide

October 13, 2025

Contents

1	Introduction	3
2	Architecture of the software	3
3	Solving problems	3
3.1	Simple examples	3
3.2	Training and testing the MLP	3
4	General comments	4

1 Introduction

This application is an implementation of the multi-layer perceptron (MLP) [1] dedicated to protein sequence processing. This dedication was made with the aim to ensure that minor changes should also make the application suitable for DNA or RNA sequence processing problems (recognition of protein-coding segments, exon recognition, etc.).

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 problems

3.1 Simple examples

A simple way to become familiar with the use of the software consists in running it on the data sets provided, which are located in the directory `MSVMpred/Data`. In order to select any of the corresponding problems, it suffices to use the corresponding script, named `configure.name`, where `name` is the name of the problem. 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 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 eleven lines. Its structure, illustrated on the IPM problem, is as follows:

```

on ← 'o': mlp used for training, 'n': weights initialized randomly ('o'
would correspond to a warm start)
22 ← number of symbols in the alphabet
2 ← number of categories
../Data/IPM1.app ← name of the file where the training set is
stored
matrix/IPM1.mat ← name of the file containing the initial values of
the weights
matrix/IPM1.mat ← name of the file where the new values of the
weights will be stored
20 ← number of units in the hidden layer
0.01 ← initial value of the learning rate
0.01 ← value governing the intervals for the random initialization of
the weights
10000 ← number of epochs
Data/IPM1.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 fourth 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.

References

- [1] M. Anthony and P.L. Bartlett. *Artificial Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.