Spiral Example Guide

Gabriele Pernici

May 2024

Contents

1 Introduction

This document depicts how to use this example to reproduce the data regarding the spiral example. Bear in mind that a terminal capable of supporting the *make* and *g++* commands must be used, moreover this guide is written from a linux user's point of view.

2 Setup

The first thing to do is to download the GitHub repository and extracting it from its zipped state. After that, you must use the *make* command from terminal, while being in the repository's first folder. This will produce the executable file named *main* inside the same folder.

3 Training

In order to start the training the following command must be used, while being in the same folder:

```
./main SpiralExample/ComandFiles/Train.cmd
```

This will start the training algorithm and save every output in the *Results* folder.

4 Results files

In the results folder there will be different files:

- 3 log files: *log.txt* (with every iteration result), *test_log.txt* (with every test result that is performed every 1000 iterations) and *moves_log.txt* (with every move and decision taken).
- The struct files (all the *.json*): they contain a loadable structure of the network after every bit increase in the training process. The *struct.json* file contains the final structure.
- The results files (all the *.exa*): they contain the actual result classification performed by the network on each input sample. The *res.exa* file contains the final result.

5 Evaluation

After the training, in order to do the evaluation the following command must be used in the usual folder:

```
./main SpiralExample/ComandFiles/Eval.cmd
```

This will produce the results used for the graphical representation of the classification. The results will be placed in the *Results* folder as *eval.exa*.

6 Python scripts

There are 4 python scripts in the repository (inside the *PythonScripts* folder). What each of them is used for the following:

- *GenerateSamplesForImage.py* is used to generate the input file (*Spiral_evaluation.exa*) for the evaluation described in section 5.
- *GenerateSamplesForTrain.py* is used to generate the input file (*Spiral_input.exa*) for the training described in section 3.
- *PlotNumericalResults.py* is used to analyze the progress of the training and its generalization (described in the *log.txt* and *test_log.txt*).
- *PlotSpiralImage.py* is used to plot the results of the evaluation process (described in section 5) and saves the image inside the *Results* folder as *eval.jpg*.