Assignment 2 – Learning

Learning with neural networks

1. Objective

In this assignment we will use a multilayer neural network to learn how to to identify the type of an animal whose characteristics are known. Code is provided that allows training a network to imitate simple binary functions. The code should be adapted to solve this new problem.

2. Some Rules

- The work should be done mostly in the practical classes, in groups of two students.
- It should be implemented using the programming language used in practical classes (Python).
- Evaluation is continuous and done during the practical classes.

3. The Problem

In this work we want to implement neural network based agent capable of identifying the type (bird, fish, etc...) of an animal whose characteristics (if it has feathers, if it produces milk, etc...) are known. Code is provided to define a multilayer neural network and implements the error back propagation training mechanism. Sample code is also provided on how to use a neural network to learn simple Boolean functions such as AND, OR, and XOR.

The students should adapt this code to train a network capable of receiving the characteristics of an animal and correctly identifying its type. For training, an additional file (zoo.txt) is provided, which contains examples of 121 animals, described by a list of 18 attributes. The first

attribute is the animal's name and should not be used in training. The following 16 attributes are numerical and shall be used to create the inputs for the network. The last attribute corresponds to the type of the animal. Attributes are described in a second provided file (info.txt). The network must have 7 outputs, one for each possible type (attribute 18). For example, when the third output of the network has the highest value, it means that the animal whose data was provided is a reptile (reptile).

4. Construction of Training and Test Sets

Before starting this stage of the work, the student must analyse the provided code, which is commented in order to describe its operation. The student should also test running the functions `train_and`, `train_or` e `train_xor` to learn the corresponding Boolean functions.

In this first stage we will implement the `build_sets` function and an auxiliary function called translate which will be called by the previous one. The `build_sets` function creates the training and test sets that will be used in the training and evaluation of the neural network from the data stored in the zoo.txt file. The function must read each line and transform it into a list of values, considering the type of the values being read. The list is then passed as an argument to the translate function, which will construct a training pattern in the format discussed further below. The resulting list will be stored in a list of patterns, whose order must be randomized. Finally, the function should return two lists, the training set with the first 67 patterns and the test set with the rest.

The `translate` function takes each list of values and turns it into a training pattern. Each pattern is a list with the following format `[animal_name, input_pattern, animal_type, output_pattern]`, where the elements of the list are obtained as follows:

- `name_of_animal` is the string contained in the first value of the list received as an argument;
- `animal_type` is a string stored in the last position of the same list;
- `input_pattern` is a list of 0 and 1, containing the attribute values; the number of legs must also be converted into a list of 0 and 1 (10 bits), concatenated with the rest, e.g. `[0 0 0 0 1 0 0 0 0 0]` -> 4 legs;
- `output_pattern` is a list of 0 and 1 that represents the type of animal; it has 7 positions and the position with a 1 corresponds to the animal type. E.g., [0 0 1 0 0 0 0] -> reptile, following the order of attribute values presented in the info.txt file.

In summary, and to exemplify, the first line of the zoo.txt file should result in the following pattern:

"[aardvark,1,0,0,1,0,0,1,1,1,1,0,0,4,0,0,1,mammal]" should result in:

```
['aardvark', [1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1], 'mammal', [1, 0, 0, 0, 0, 0, 0]]
```

5. Training the neural network

In this second stage of the work we will implement three functions. The first function is called `train_zoo`, it receives the training set and must create the neural network and call the iterate function to train it for 300 iterations. This function works in a similar way to the training functions provided in the code, and in each training iteration, it must go through all the patterns stored in the training set, calling the `iterate` function, which receives an input pattern (list of animal attributes) and output (list with binary coding of animal type).

The second function, `test_zoo`, aims to test the performance of the trained network. To do so, it must traverse the test set, and, for each pattern, call the `forward` function, passing the trained network and the list of attributes of the animal as arguments. Next, the function must analyze the list of network outputs (`net['y']`), calling the third function we need to implement (`retranslate`). This function should return the type of the animal corresponding to the network output with the highest value. The `test_zoo` function must compare the value returned by the auxiliary function with the true type of the animal, stored in the training pattern, calculating the percentage of correct answers. For each pattern you should also print a line similar to the following:

```
The network thinks mongoose is a mammal, it should be a mammal
```

After the lines with the result referring to each training pattern, you must also print the value of the final performance of the network (percentage of correct answers), for example:

Success rate: 94.12