

1 Problem description

The aim of this homework was to create a neural network in C, with two hidden layer. The aim is to predict human satisfaction to be in the input conditions, which are the temperature, the humidity and the air quality index.

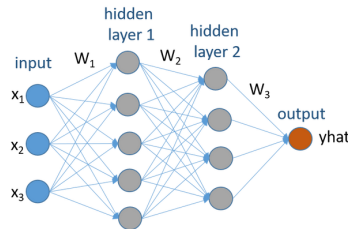


Figure 1: The network we want to implement

2 Description of the solution

1. Initialization

To train our model, we will take a dataset of size 3.

| Temperature | Humidity | Air quality | Satisfaction |
|-------------|----------|-------------|--------------|
| 28 | 45 | 20 | 1 |
| -30 | 0 | 500 | 0 |
| 35 | 41 | 25 | 1 |

All the weights are initialized randomly, using `weights_random_initialization()`. The bias are set to 0.

2. Forward propagation

To implement **forward propagation** we use the function of Lab3. The first and the second **forward propagation** are very similar because the activation function is `relu`.

We compute forward propagation `m` times in the same epoch, for each data in the dataset.

At the end of the **forward propagation**, we can compute the cost.

3. Backward propagation

To implement **backward propagation**, we use the function of Lab4. Contrary to the **forward propagation**, we apply all **backward propagation** at once, using the function for basic matrix operation.

4. Overall functioning

After **backward propagation**, we update parameters for each layers (weights and bias).

Forward propagation, **backward propagation** and **update of parameters** are wrapped inside a **for**, which iterate so we have **NUM_EPOCH** iterations.

3 Results

Using our dataset, we can train the data. Choosing a learning rate of 0.015 and with 500 epochs, we have a pretty good result : we can separate "bad" environment from "good" environment pretty well.

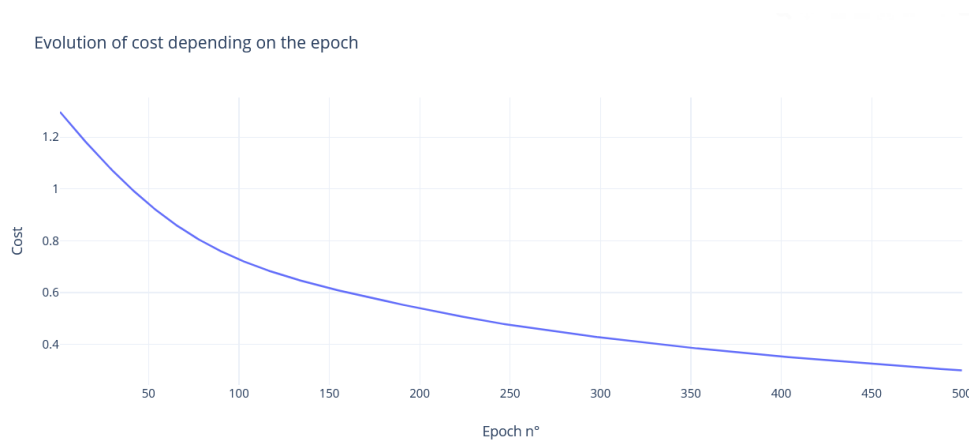


Figure 2: Evolution of cost depending on epoch

4 Issues

The main issue is that we provide really few data to train our model, so the result can be sometimes quite surprising. In addition, the data we provide is totally created, and can lead to some errors. For example, I spend a lot of time trying to figure why my implementation was not working well, when it was actually the data provided which caused problems.