Assignment 1: Report

Artificial Neural Network.
DVA427 - Lärande System.

Joaquín García Benítez.
Clara Torre García-Barredo.
Mälardalen Högskola.
April 2019.

Explanation of the problem.

This problem consists of creating an Artificial Neural Network (ANN) that can determine if a patient is sick or not based on 19 input parameters that indicate the presence or absence of the disease on the patients.

The Artificial Neural Network is trained with 863 training examples of patients, with their input parameters and if they have the disease or not, and then validated with 115 more examples. At last, it is tested with 173 more patients.

Structure of the ANN.

Our Artificial Neural Network is structured in four different layers:

- **Input layer.** This layer consists only of the inputs. It is formed by 19 nodes that carry one parameter each, composing the diagnostic of the patient. This information is passed onto all the neurons in the first hidden layer, with weighted connections.
- **First hidden layer.** This layer consists of 8 neurons that get the 19 parameters as the weighted inputs, and uses the sigmoid function to calculate its outputs, which are then passed onto the second hidden layer.
- **Second hidden layer.** This layer consists of 5 neurons that get the output of the first hidden layer as a weighted input, and uses the sigmoid function to calculate its outputs, which are then passed onto the output layer.
- **Output layer.** This layer consists of 2 neurons, one for each possible class (0 or 1), that get the output of the second hidden layer as a weighted input, and uses the *softmax* function to calculate its outputs. It returns 2 values between 0 and 1, of which the highest is the ANN's prediction of whether the patient has the disease (1) or not (0).

Equations for updating the weights.

The equations we have used to update the weights are as follows:

$$layW = layW + \eta * layI * layE$$

The parameters of this equation are:

- layW: the weights used by the layer in question.
- η: the learning rate chosen for this problem. In our case, 0.1.
- layI: the inputs passed to the layer in question. For example the output layer, which are the outputs of the second hidden layer.
- layE: the error calculated for the layer in question.

The equations used to calculate layE, or the error of each layer, are two:

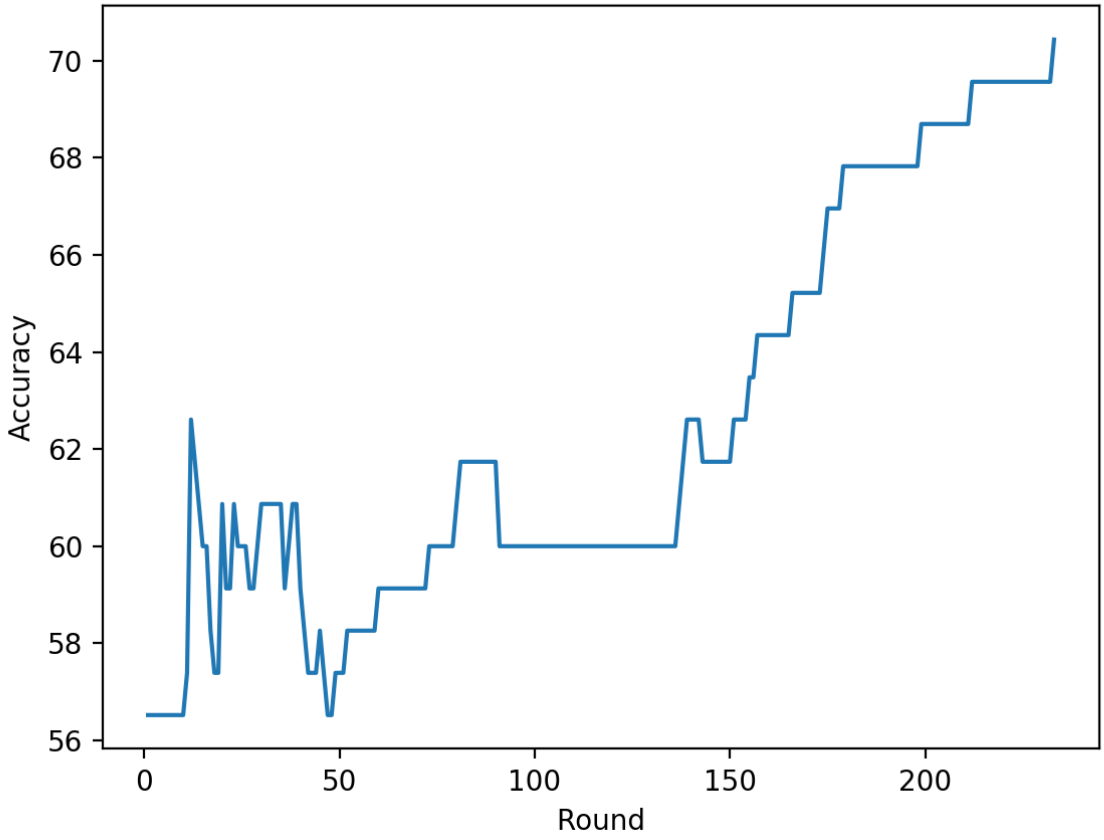
$$outE = targ - sofOut$$

Here, *targ* is the target output (whether the patient has the disease or not), and *sofOut* is the output given by the ANN.

$$hidE = (1 - hidOut) * hidOut * (layE * layW)$$

Here, *hidOut* is the outputs of the hidden layer in question, and *layE* and *layW* are the error and weights, respectively, of the next layer in the ANN. So, if this equation is used for the first hidden layer, *layE* and *layW* would be those of the second hidden layer, and if this second hidden layer were the one in question, *layE* and *layW* would be those of the output layer.

Change of validation accuracy.



As can be seen in the image, the validation percentage changes abruptly at first, while it's starting the learning process, and grows more steadily at last, until it reaches the desired percentage of accuracy.

Percentages of correctness.

The percentages asked for are two:

- The percentage of correctness of the total test data set: 76.16%.
- The percentage of correctness of each of the classes in the data set:
  - 0: 75.28%.
  - 1: 77.10%.