Diagnosis of diabetes using a supervised trained neural network (MLP)

1. Description of network implementation.

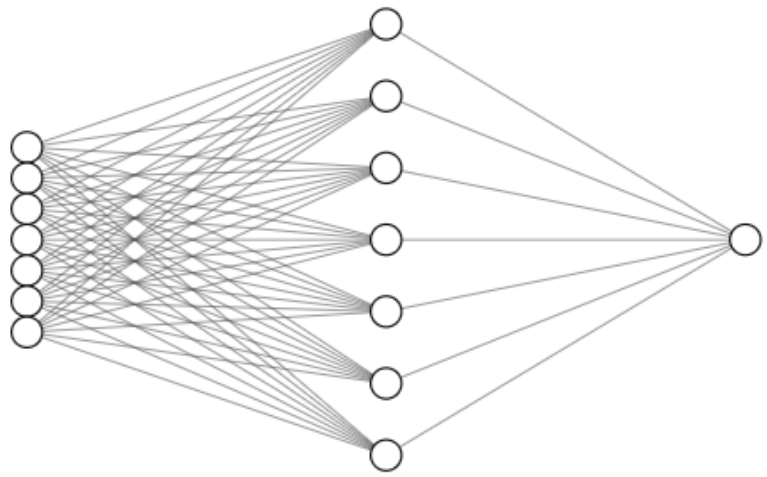
The prepared neural network has been implemented in the Python programming language. The program is divided into three files:

- **funcs.py**, in which functions are written, including data normalization, activation functions, and their derivatives,
- **network.py**, in which the implementation of the network is located, i.e., the Network class along with functions like weight initialization, data propagation, and weight correction,
- **main.py**, which is responsible for training and testing the network.

Libraries used:

- **NumPy** – Used, for example, for creating matrices and performing mathematical operations,
- **Pandas** – Used for data manipulation and importing datasets.

2. Network structure.



a) Input layer
- Number of neurons = 7
- Activation function: Linear, $f(y) = y$

b) Hidden layer
- Number of neurons = 7
- Activation function: LeakyReLU, $f(y) = \max(0.01y, y)$

c) Output layer
- Number of neurons = 1
- Activation function: Sigmoid, $f(y) = \frac{1}{1+e^{-y}}$

The network is used for binary classification. The adopted concept is as follows: a healthy person - value 0, a sick person - value 1. Therefore, to verify the network's performance, following approach applies:

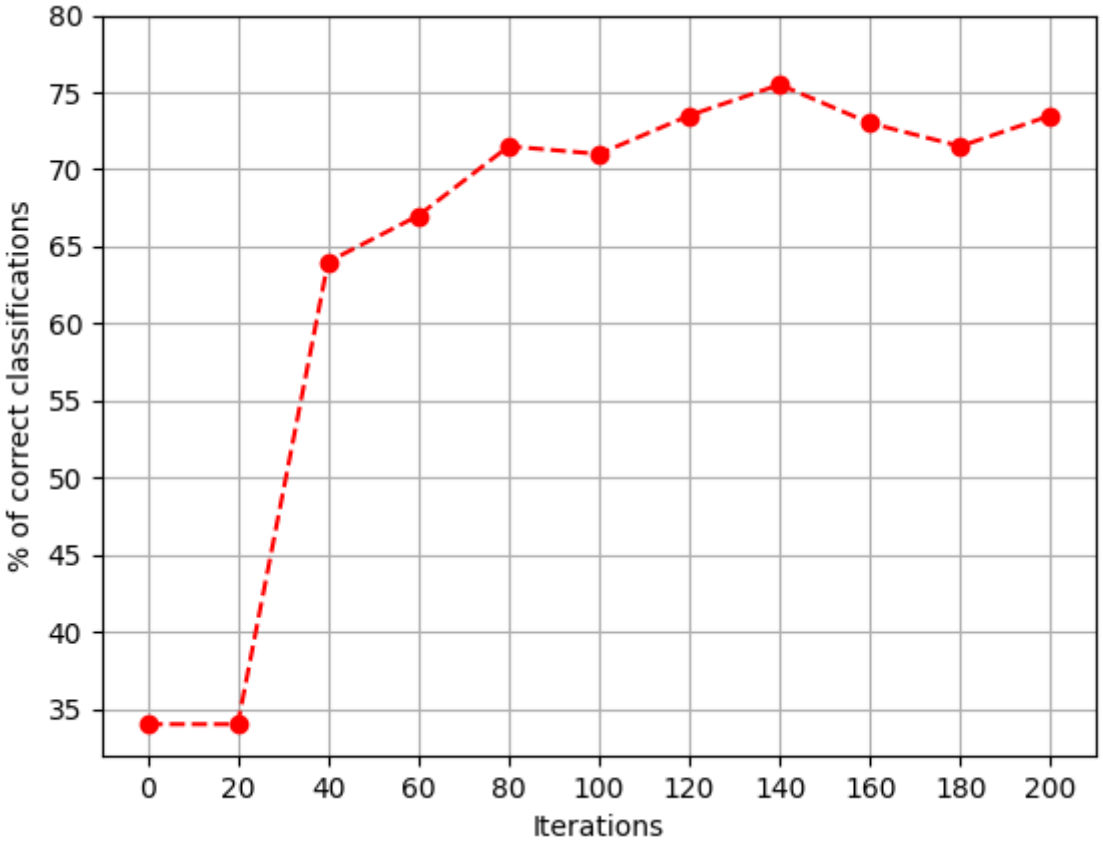
- Output neuron result in the range [0, 0.5) - person classified as healthy.
- Output neuron result in the range [0.5, 1] - person classified as sick.

3. Learning algorithm

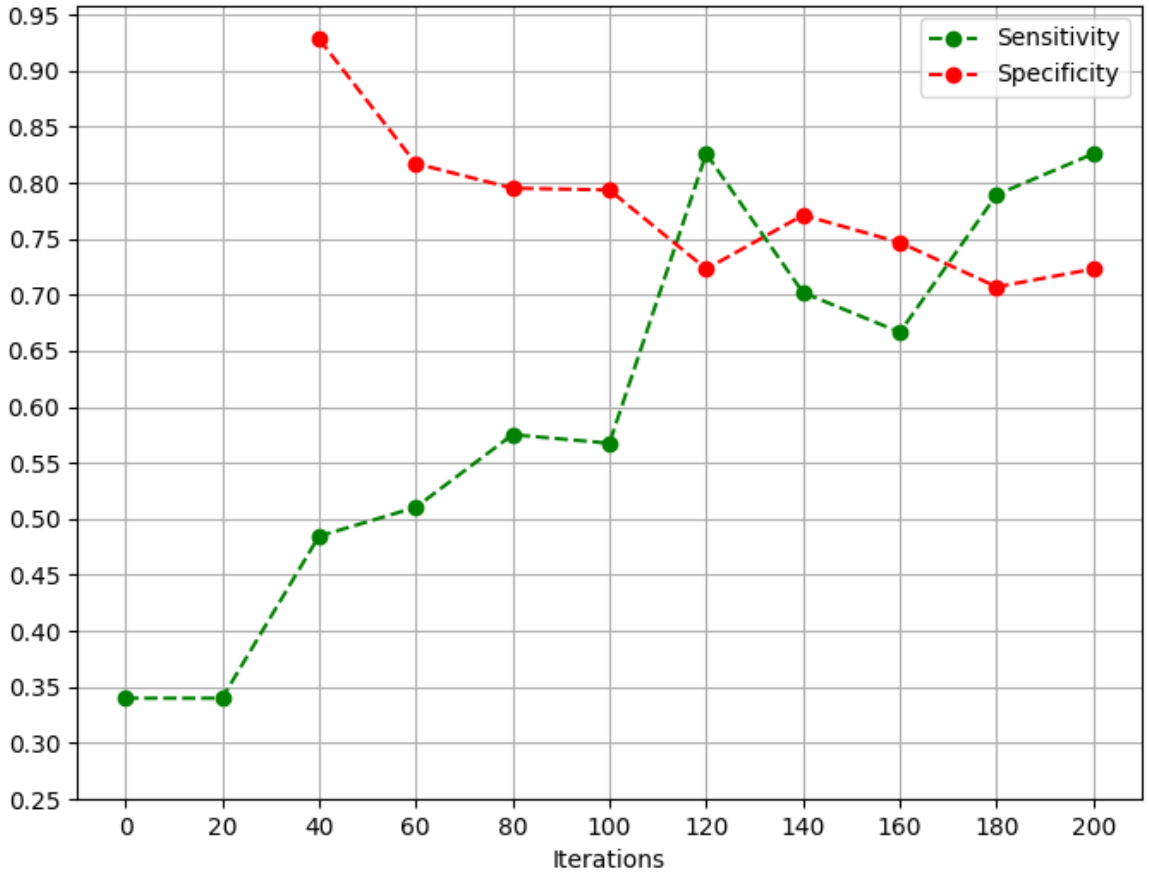
The learning algorithm involves initializing the network with small, randomly chosen weights (after preliminary tests, a weight range of [-0.1, 0.1) is determined). Subsequently, data is fed into the network, propagating through to the output layer of the network. The learning algorithm is based on the method of backpropagation of errors, thus errors of hidden layer neurons and the output neuron are calculated. Using these values, weight adjustments for the hidden and output layer neurons are computed, applied, and the next iteration of the method is performed.

4. The process of training the network.

a) Correct classifications



b) Sensitivity and specificity



5. Example of program operation

```
---
0.0% training complete
Current results from the training dataset
Correct classifications = 43.5 %
Sensitivity = 0.1053
Specificity = 0.5664
----
20.0% training complete
Current results from the training dataset
Correct classifications = 64.0 %
Sensitivity = 0.4841
Specificity = 0.9054
----
40.0% training complete
Current results from the training dataset
Correct classifications = 71.0 %
Sensitivity = 0.5694
Specificity = 0.7891
----
60.0% training complete
Current results from the training dataset
Correct classifications = 71.5 %
Sensitivity = 0.5647
Specificity = 0.8261
----
80.0% training complete
Current results from the training dataset
Correct classifications = 74.5 %
Sensitivity = 0.6977
Specificity = 0.758
----
The results from the training dataset:
Correct classifications = 74.0 %
Sensitivity = 0.7667
Specificity = 0.7353
----
The results from the test dataset:
Correct classifications = 69.88 %
Sensitivity = 0.5634
Specificity = 0.7356
----
```

200 epochs