NEURAL NETWORKS – exercise 2

In the second exercise, we will cover machine learning. We will use logistic regression - a statistical method that can be considered a form of the simplest (single-layer) neural network. We will use the model for classification, i.e. we want it to approximate the probability of the sample described by vector x belonging to the correct class. For this purpose, we can use a set of training examples in the form of input-class pairs.

We can define the output as:

$$( ) = (+)$$

Where x is the input vector, W and b are the model parameters and the sigma function is:

$$\sigma(n) = \frac{1}{1 + -}$$

This is a convenient function that gives results in the interval [0,1] for every real n, and is also increasing and differentiable.

To train the model in any way, we need a defined optimization task, and for this we need a cost function that we will minimize. We use cross-entropy, for example (x,y) in binary classification:

$$= \ln( ) + (1 -) \ln(1 - ( ))$$

(y will always be zero or one, i.e. only one of the components will be non-zero. The value is summed over all examples in the data set, the sum will be omitted in the formulas.)

The advantage of this cost is that its derivative with respect to the model weights is very simple:

$$\frac{}{} = (- ( ))$$

Model optimization will consist in taking small steps in the direction determined by the gradient - partial derivatives on all weights
– in the training loop until the model converges. In other words, we update the weights by:

$$' = + \frac{}{}$$

Where alpha is a certain learning coefficient, a hyperparameter that we must set in advance.

The task for class 2 is to implement an action on a set*heart disease* classification model, where:

- Model convergence can be defined by a sufficiently small change in the cost function in a given iteration and some maximum number of iterations
- The model can learn by calculating the summary derivative of the cost function over the entire set, one example or a set of examples in an iteration. For the neural network, the next task will require the packing mode, so it is worth practicing operations on entire matrices and not just single data vectors
- Model learning should be verifiable with metrics (e.g. accuracy, fscore, precision - libraries can be used)
- Verification should take into account the division into training and testing data

The exercise is graded on a scale of 0-10 points.