Tasks on the course "Neural Networks in Machine Learning"

[Task No. 1 2](#_Toc72231193)

[Acquaintance with modern software that implements neural networks 2](#_Toc72231194)

[Task No. 2 3](#_Toc72231195)

[Testing neural networks to solve a binary classification problem 3](#_Toc72231196)

[Task No. 3 6](#_Toc72231197)

[Perceptron research 6](#_Toc72231198)

[Task No. 4 8](#_Toc72231199)

[Application of neural networks for data processing tasks 8](#_Toc72231200)

## Task No. 1

# Acquaintance with modern software that implements neural networks

**A task:**

Choose a suitable technology / programming language / environment for performing laboratory work in the course "Neural Networks".

**Operating procedure:**

Consider the implementation of neural network algorithms in (at least 4-5)

* Python
* R
* Matlab
* Wolfram m athematica
* Statistica
* AI Experiments
* Etc. (I hope you are not limited to the list above)

To do this, select (develop) the criteria by which the products will be evaluated. And appreciate.

**Reporting** :

* Esse

**Content of the report:**

The report should include the following sections (but are not limited to ):

* list of criteria,
* summary table with comparison,
* List of sources,
* explanations, opinions, opinions of authoritative and not particularly authoritative sources and authors are possible,
* conclusion (on what exactly you, based on your research, will do the tasks of the laboratory workshop,
* ...

## Task No. 2

# Testing neural networks to solve a binary classification problem

**A task:**

Build, train and conduct testing for neural STI solutions for the garden bianrnoy classification, as well as to study the dependence of the results of learning and prediction on the input data.

**Work order:**

1. Simulate "different" inputs belonging to two classes:
   1. samples of various lengths,
   2. different balance of observations in classes,
   3. different class separability,
   4. data with a more complex structure.
2. Divide the input data into training and test samples.
3. Build and train a neural network on a training set.
4. Investigate the dependence of training and forecasting (on a test sample) for various input data.
5. Prepare a report.

**Report:**

The report should contain:

* description of various model data (model and graphically , if possible ) ,
* experimental results.

**Examples of "different" inputs:**

|  |  |
| --- | --- |
| Two well separable classes of Gaussian observations of the same size (100 observations each) with the same variation matrix and different mathematical expectations: |  |
| 1000 observations: |  |
| Classes of various sizes (50 and 300 observations) |  |
| Unbalanced classes, with closer centers |  |
| Even closer and matched with covariance |  |
| And so far .. |  |

## Task No. 3

# Perceptron research

**A task:**

Investigation of the principle of construction, training and functioning of a neural network that implements Rosenblatt's single-layer perceptron in the problem of decimal digits recognition.

**Work order:**

1. Generate training and test samples (you can share with each other, you can use pictures from the Internet).
2. Build and train a neural network. Explore problems and subtleties.

**Reporting:**

1. Report in "text format"
2. Trained neural network (so that you can immediately check its performance)
3. "Blanks" for "testing" are empty white drawings on which you can draw a number in a graphical editor and submit this drawing to the input of the neural network for testing.

**Content of the report :**

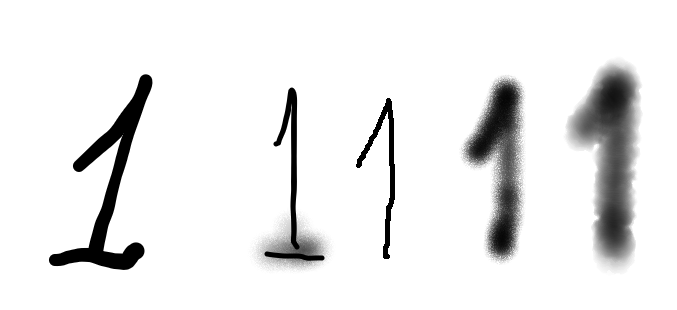
1. Diagram of the constructed neural network
2. Collection of educational pictures
3. Network parameters
4. Results on the test sample and conclusions on the work

**Initial data:**

The initial data are figures of the same size, white background, black numbers. For example:



The figures may contain noise (in this case, vertical and horizontal lines), the thickness of the lines may be different, the numbers are handwritten in different handwriting.



## Task No. 4

# Application of neural networks for data processing tasks

**A task:**

Build and study a multilayer neural network for

* approximation of a function of several variables
* predicting time series values

**Work order:**

1. Simulate raw data
2. Visualize them
3. Simulate a multilayer perceptron
4. Train, validate and test the resulting network
5. Analyze learning outcomes

**Reporting:**

1. Trained neural network
2. Report

**Content of the report:**

1. Diagram of the constructed neural network
2. Weights
3. Learning and Testing Results
4. Dependence of the accuracy and topology of the network on the input data and training duration

**Initial data:**

1. One of the tasks is selected: either the approximation of functions, or forecasting.
2. A basic type of function or time series is selected and training is carried out, for example:

– some well-known constants

1. We complicate and / or "clog" the input data
   1. instead of training, a "noisy" - Gauus random variable with zero expectation and some variance is used (how does the value of variance affect training and the accuracy of a neural network?)
   2. we complicate the type of "functional" dependence, for example:

where are the new model parameters:

* + 1. comparable to the original,
    2. small compared to the original
  1. injecting noise into data with a "complicated dependency"
  2. etc