MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
National University "Zaporizhzhia Polytechnic"

LABORATORY GUIDELINES

in the discipline

"Modern methods and models of Intelligent Systems"

for students of specialty 123 "Computer Engineering"
according to the educational program "Specialized Computer Systems"

for all forms of education

2023

Laboratory guidelines in the discipline "Modern methods and models of Intelligent Systems" for students of specialty 123 "Computer Engineering" according to the educational program "Specialized Computer Systems" for all forms of education / Compl.: M. Tiahunova – Zaporizhzhia: NU "Zaporizhzhia Polytechnic", 2023. – 36 p.

Compiler: M. Tiahunova, Ph.D., Associate Professor

Reviewer: H. Kyrychek, Ph.D., Associate Professor

Responsible
for issue: M. Tiahunova, Ph.D., Associate Professor

Approved at the meeting of the Computer Systems and Networks Department
(protocol № 5 of January 30, 2023)

Recommended for publication by the Educational and Methodological Commission of Computer Sciences and Technologies Faculty
(protocol № 6 of January 31, 2023)

CONTENT

1 LABORATORY WORK № 1
NEURAL NETWORKS IN MATLAB® ENVIRONMENT

The purpose of the work: to acquire skills in the software environment MATLAB when working with neural networks.

1.1 Theoretical information

Neural networks are widely used to solve a variety of problems. Among them are the tasks of classification, clustering, forecasting, management, approximation, optimization and management. *MATLAB®* provides the ability to simulate the operation of neural networks of various types, using a convenient interface.

Usingneurons with certain activation functions (transmitting characteristics), neural networks are a universal approximator. The latter means that in a given range of changes in input variables, neuron networks can reproduce (simulate) an arbitrary continuous function of these variables with a given accuracy.

In order to work with MATLAB®, you must register on the official website of the developer of this software at the link https://www.mathworks.com/. After that, you must either download this software environment to your computer, or work in the online version of this product.

The details of installing software MATLAB®2022 on your computer are shown in Fig. 1.1.

In this laboratory work, neural networks of direct propagation will be considered, that is, without feedback, a feature of which is their stability.

After the structure of the neural network is selected, the parameters that will be at the input and output of the network are specified, the percentage of data that is intended for training, control and testing of the neural network is determined, it is necessary to train the neural network.

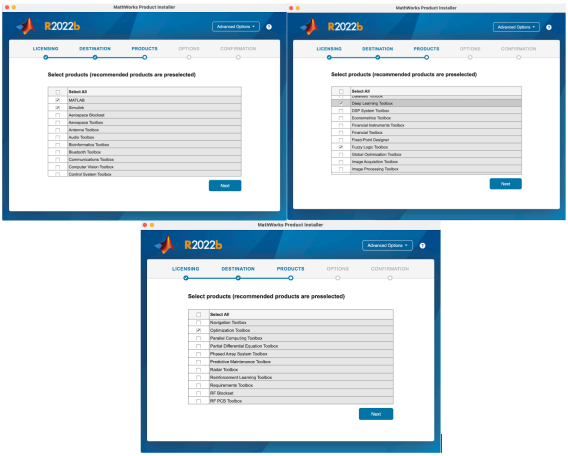

Figure 1.1 – Features when installing MATLAB®2022

And the MATLAB® interface for solving problems with neural networks can be launched in several ways:

1 Method: entering command *nnstart* in the command window, as shown in Fig. 1.2;

2 Method: by going to the APPS and selecting the appropriate network type in MACHINE LEARNING AND DEEP LEARNING, as shown in Fig. 1.3.

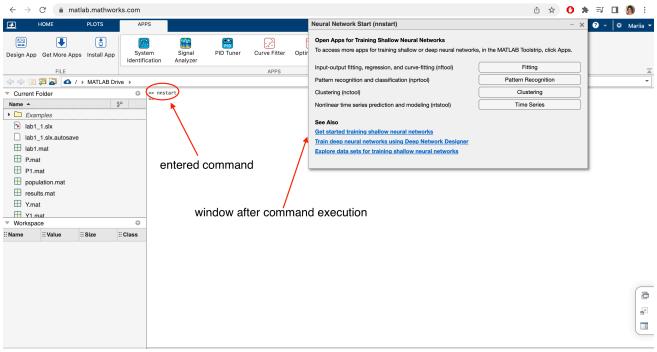

Figure 1.2 – Entering the command nnstart

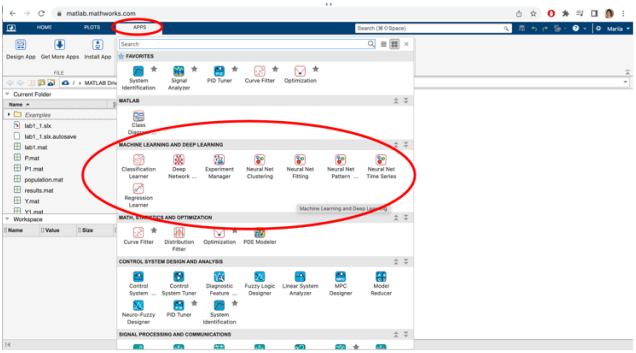

Figure 1.3 – Selection in the APPS

The Neural Net Fitting application allows you to create, visualize and train a two-layer direct communication network to solve the problems of its data.

Using this subsection of the program, it is possible to:

- import data from the file, the MATLAB® workspace, or use one of the above examples of data sets;
- to allocate data to kits for training, verification and testing.
- to identify and teach neural networks;
- to define network performance using standard error and regression analysis;
- to analyze of results using visualization graphs, such as regression or a histogram of errors;
- with the creation of its functions, suitable for deployment using the tools MATLAB Compiler™ and MATLAB Coder, and exporting to Simulink for use with Simulink® Coder™.

1.2 Progress

In this laboratory work, we will simulate a neural network, which will determine the number of the population of Ukraine depending on the year entered.

The necessary data for processing will be taken from Table 1.1, which is presented below.

Table 1.1 – Population of Ukraine in different years

Year	Population	Year	Population	Year	Population	Year	Population
1950	36588000	1987	51260900	2000	49115000	2010	45782600
1951	37223000	1989	51452000	2001	48663600	2011	45598200
1959	41869000	1992	51708200	2002	48240902	2012	45453300
1960	42468600	1993	51870400	2003	47823100	2013	45372700
1970	47118200	1994	51715400	2004	47442100	2014	45245900
1973	48274400	1995	51300400	2005	47100500	2015	42759661
1976	49151000	1996	50874100	2006	46749200	2016	42590879
1979	49752200	1997	50400000	2007	46465700	2017	42414900
1980	49952500	1998	49973500	2008	46192300	2020	41732779
1984	50678600	1999	49544800	2009	45963300	2022	40997699

After that, this data must be entered into the MATLAB® environment. To do this, you need to create an array of years "years" and an array of number of population "population", as shown in Fig. 1.4 list. 1.1.

Listing 1.1 – Initial data

```
years = [1950 1951 1959 1960 1970 1973 1976 1979 1980 1984 1987 1989
1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005
2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2020 2022]

population = [36588000 37223000 41869000 42468600 47118200 48274400
49151000 49752200 49952500 50678600 51260900 51452000 51708200 51870400
51715400 51300400 50874100 50400000 49973500 49544800 49115000 48663600
48240902 47823100 47442100 47100500 46749200 46465700 46192300 45963300
45782600 45598200 45453300 45372700 45245900 42759661 42590879 42414900
41732779 40997699]
```

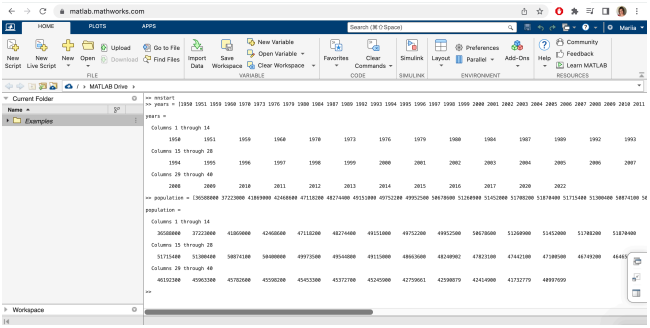

Figure 1.4 – Creating data sets in the MATLAB® Workspace

Next, in one of these ways (Fig. 1.2 and Fig. 1.3), go to the window for creating a neural network. And we choose Fitting or Neuron Net Fitting depending on the chosen method of creating a neural network (Fig. 1.5 – 1.6).

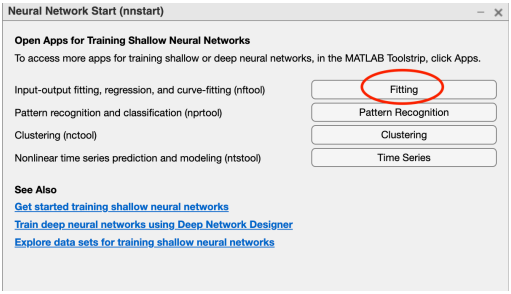

Figure 1.5 – Choosing the type of neural network

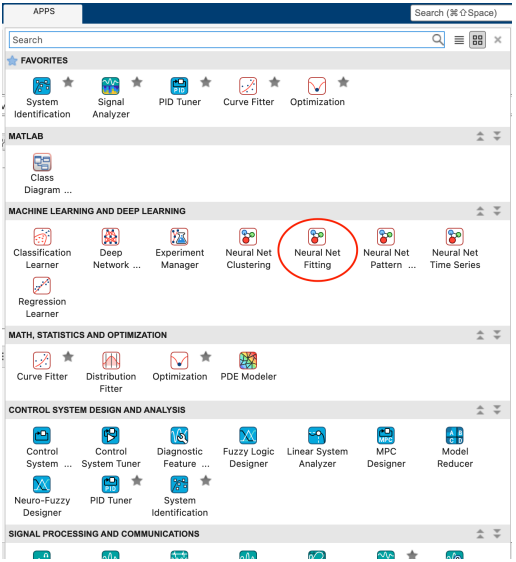

Figure 1.6 – Choosing the type of neural network in the APPS

After selecting the appropriate neural network, a window will open with its structure and settings. We select the data to download as shown in Fig. 1.7.

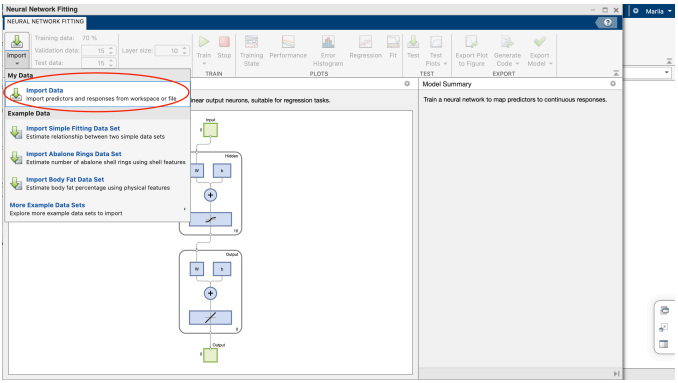

Figure 1.7 – Selecting data to import

After clicking on the corresponding field Import Data, we select the necessary data of the years and the number of people that were entered into the environment earlier, as shown in Fig. 1.8, and click OK.

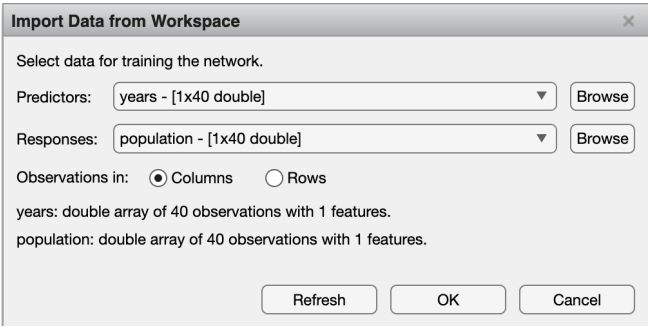

Figure 1.8 – Import Data

In the window that opens, you can change the percentage of data for validation and testing, as well as the number of layers of the neural network Changing parameters circled in red in Fig. 1.9. For our example, there is no need to change these values, so we leave them as default.

Next, it is necessary to train the created neural network and analyze the results. To start the learning process, click the Train button, as shown in Fig. 1.9.

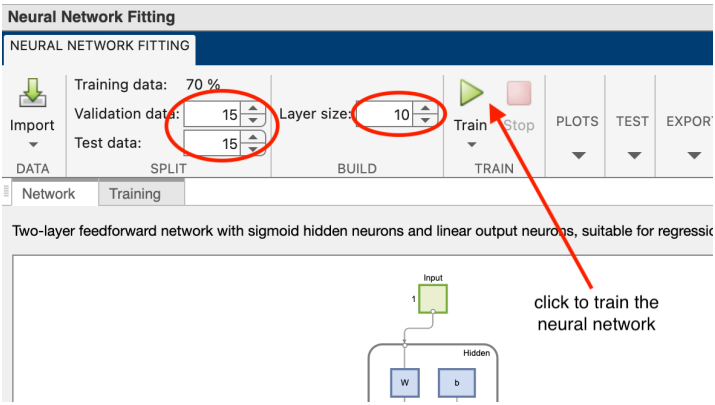

Figure 1.9 – Neural network parameter settings

After completing the learning process, a window will open with the results obtained, similar to Fig. 1.10. However, the values of the parameters that will be obtained from each student may differ from those shown in the screenshot.

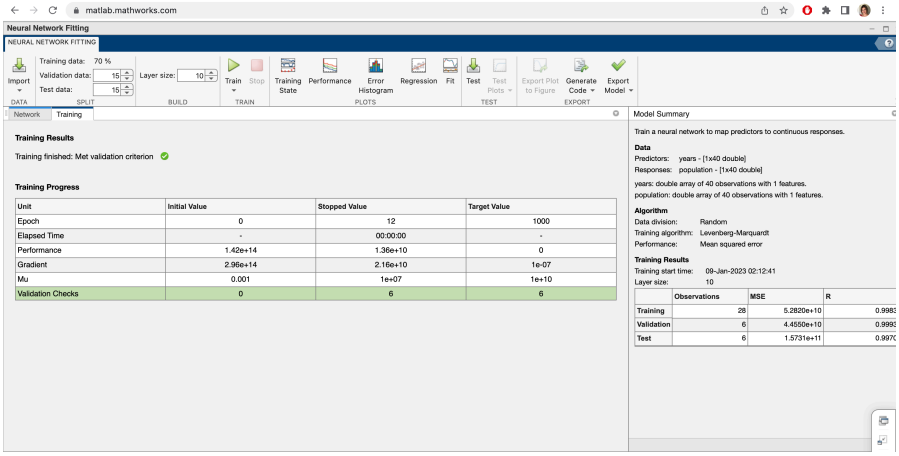

Figure 1.10 – Neural Network Learning Outcomes

By default, the Levenberg-Marquardt teaching method is chosen. Optionally, you can choose other teaching methods by clicking on the black small triangle under the Train button, as shown in Fig. 1.11.

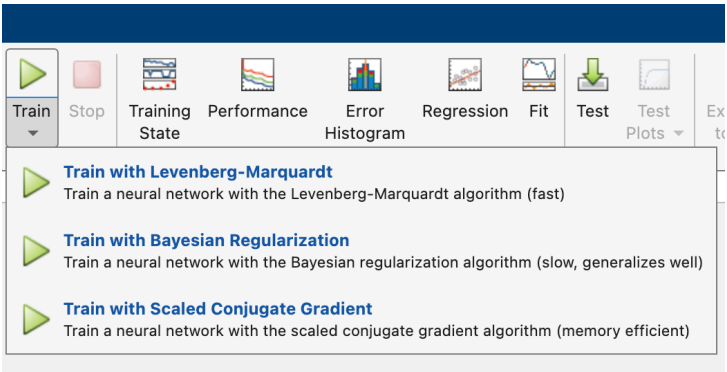

Figure 1.11 – Choosing method of a neural network training

Further, behind each of the buttons on the PLOTS section panel (circled in red in Fig. 1.12), you can view and analyze the results of neural network training.

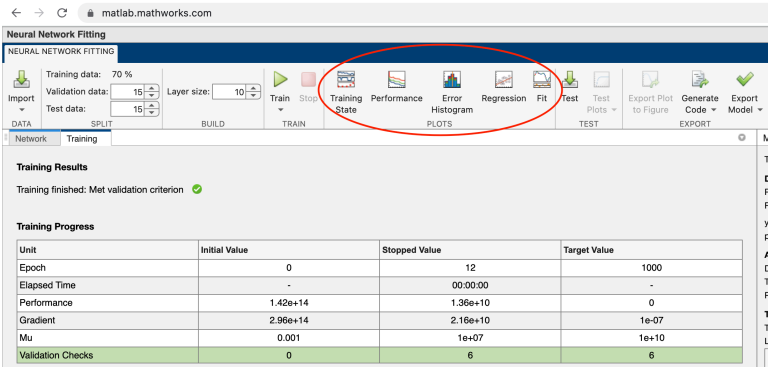

Figure 1.12 – Panel of visual analysis of the results obtained

Especially visualis the display of the real (planned according to Table 1.1) and the result obtained by the neural network using the Fit button (Fig. 1.13).

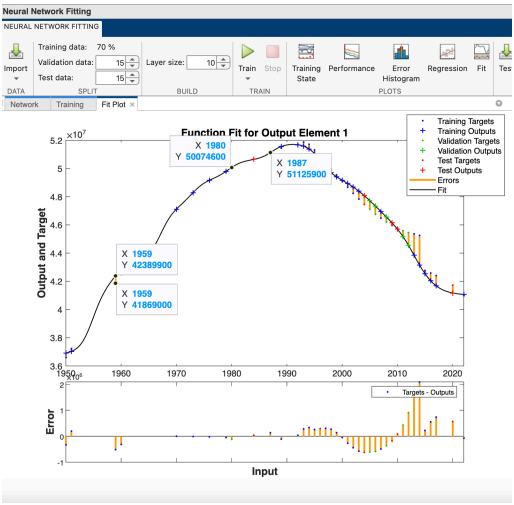

Figure 1.13 – The results obtained by the Fit

In order to view what value is at the input of the system, what will be the reaction of the trained neural network, you need to export it to the Simulink environment. To do this, click on the triangular to the bottom of the inscription "Export Model" and select "Export to Simulink" from the drop-down menu, as shown in Fig. 1.14.

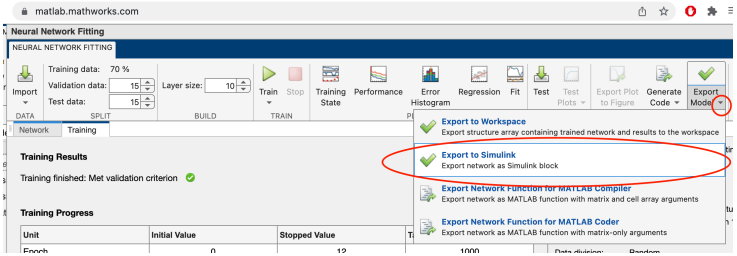

Figure 1.14 – Exporting neural network to Simulink environment

As a result, a model will be generated in Simulink, as shown in Fig. 1.15. You can switch the necessary windows for work using the navigation buttons in the lower right corner (Fig.1.16).

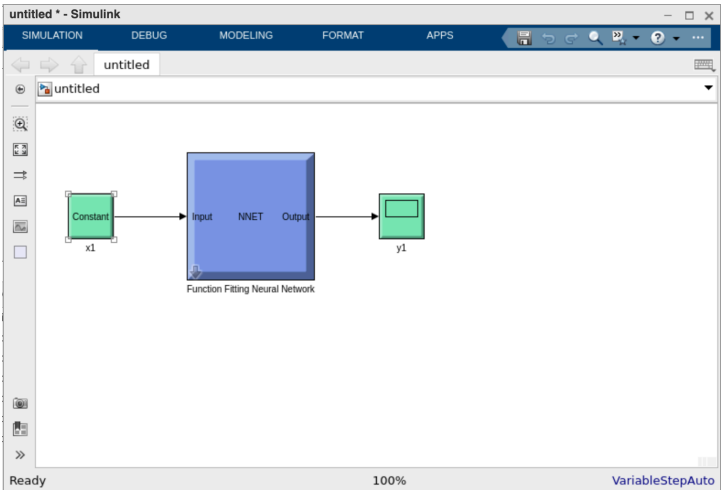

Figure 1.15 – Neural network model in environment Simulink

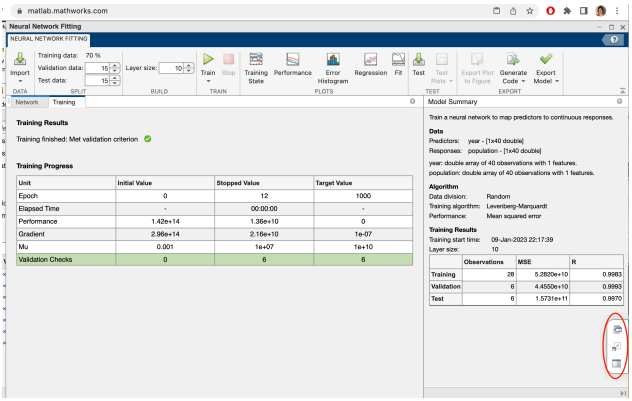

Figure 1.16 – Navigation Panel

For the convenience of displaying the results in the Simulink window, you need to change the existing output block "y1" to the "display" block. To do this, select the LIBRARY subsection in the SIMULATION tab and select Library Browse in it, as shown in Fig. 1.17.

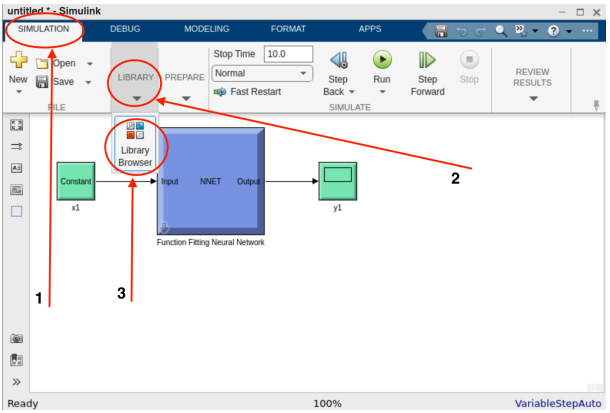

Figure 1.17 – Selection of the element base library

After that, you need to enter "display" in the search field and press the search button. In the provided list, select the element "Display", as shown in Fig. 1.18.

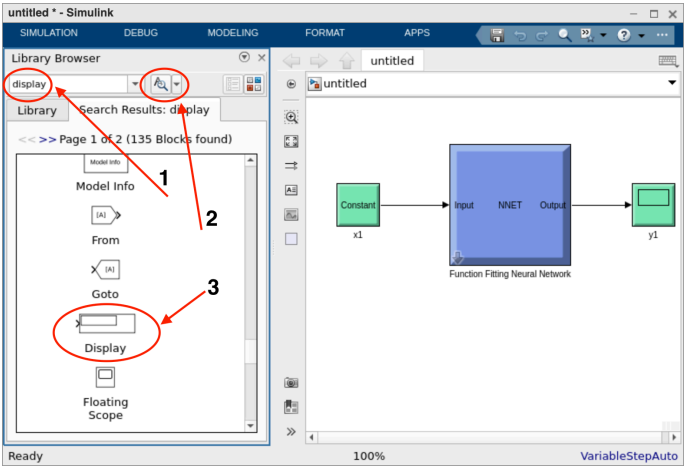

Figure 1.18 – Search for an item to display results

Next, we transfer the found block "Display" to the modeling window and replace the block "y1" with it, as shown in Fig. 1.19.

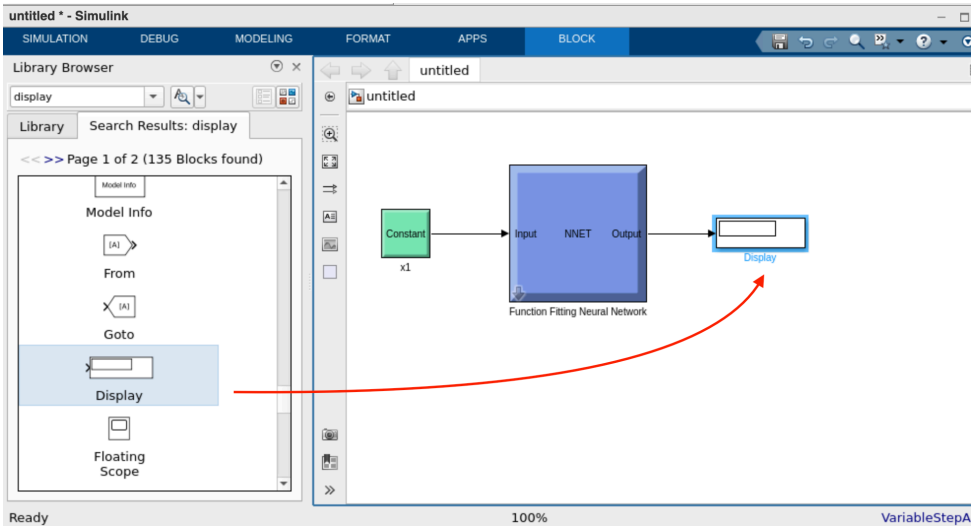

Figure 1.19 – Adding a "Display"

After that, right-click on the block "x1", select "Block Parameters" and change the value in the field "Constant value" for any year from table 1.1 (Fig. 1.20). Click OK.

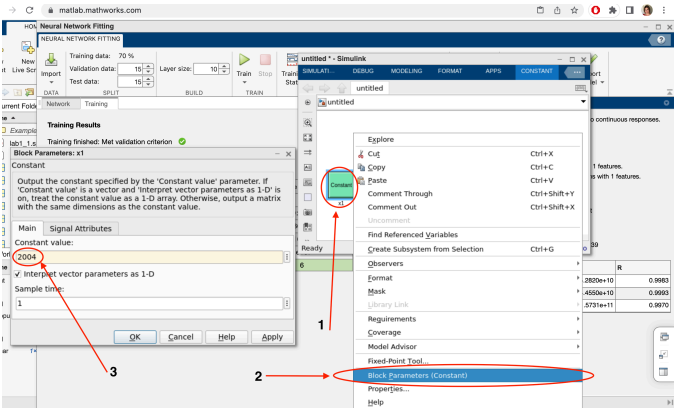

Figure 1.20 – Change of parameters on block "x1"

After that, start the modeling process. To do this, in the SIMULATION tab, press the Run button, or the F5 key. The simulation results are displayed in the Display block (Fig. 1.21).

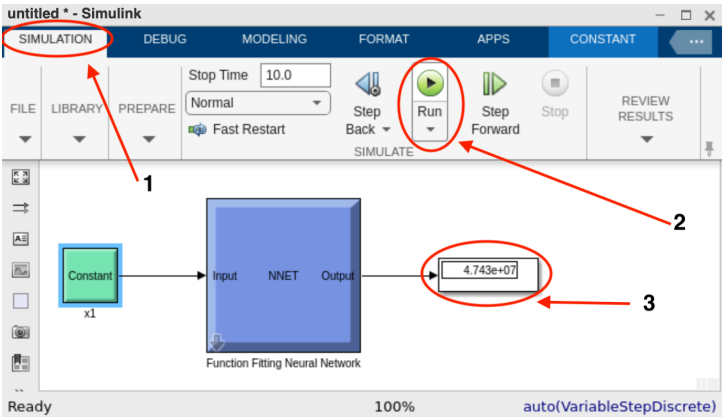

Figure 1.21 – Simulink Simulations

To change the format of the display of the source data, you can right-click on the Display block, select "Block Parameters" and select the "bank " format in the drop-down list. Next, click OK and data in block DISPLAY will change the display format to a more familiar one. Consistently, all this is displayed in Fig. 1.22.

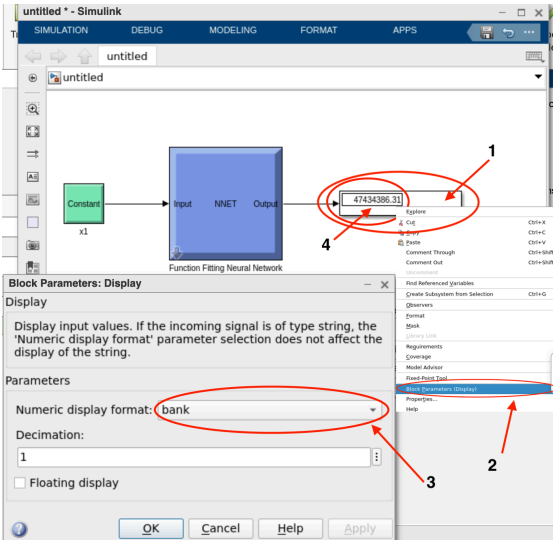

Figure 1.22 – Change format display results

In order to view the details of the simulated network, click on the arrow of the main block of the model (Fig.1.23), then you get the result, as in Fig. 1.24. For a more detailed view of individual blocks, double-click on a specific block. The result is presented in Fig. 1.25.

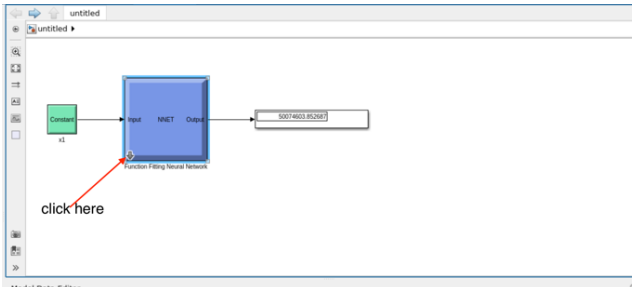

Figure 1.23 – Select a detailed view of the item

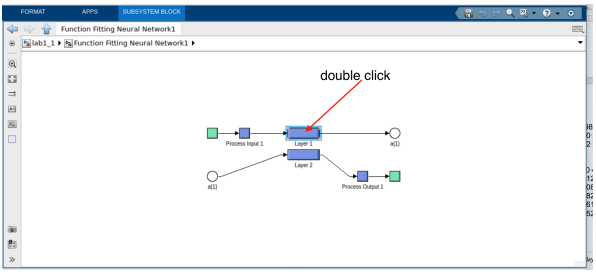

Figure 1.24 – View the approximation function

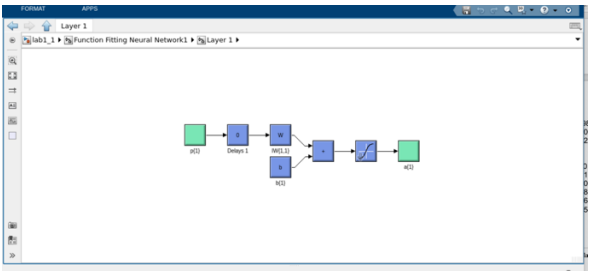

Figure 1.25 – View the approximation function of a certain layer

Next, save the results. To do this, in the SIMULATION tab, click Save, select the storage location and file name (Fig. 1.26).

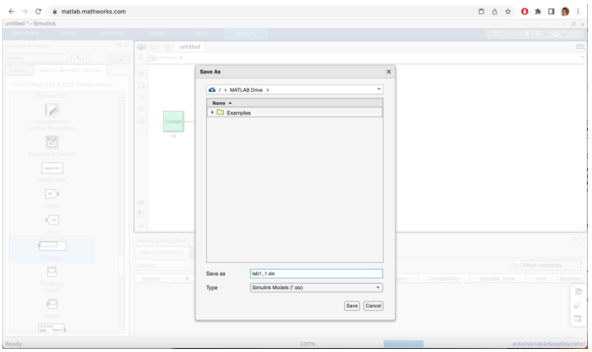

Figure 1.26 – Saving results in Simulink

In order to save the data entered into the working space, you need to save them to a separate file with the extension .mat. To do this, click Save Workspace, select a storage location and set a name, for example, lab 1_1. mat (fig.1.27).

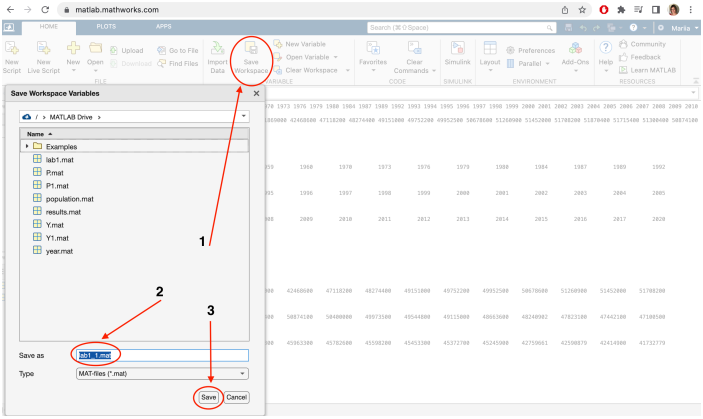

Figure 1.27 – Saving data from Workspace

Analyzing the errors in learning and the results in Simulink, we conclude that the error is quite large. One option to reduce the error can be to increase our set of initial data. To increase the data, enter the following operations into the MATLAB command window as shown below:

Y = [years years]

P = [population population]
Y1 = [Y Y Y]
P1 = [P P P]

1.3 Tasks to complete

1. Perform all the actions described in paragraph 1.2;
2. Train the neural network on an enlarged set of initial data Y1 and P1 and analyze the results;
3. Check the operation of a neural network trained on an enlarged dataset in the Simulink environment;
4. Show the results to the teacher and explain the analysis;
5. Do a similar task, but using the population of China in different years.
6. It is necessary to approximate the function of the following form:

$$cos\left(\frac{5\pi x}{N} + sin\left(\frac{12\pi x}{N}\right)\right)$$

To do this, enter the target function `target = cos function (5 * pi * [1:100] / 100 + sin (12 * pi * [1:100] / 100))` and the training data set `data = [1:100]` in the command line.

7. Analyze the results.

1.4 Contents

The report should have the following components:
– title page;
– the purpose of the work;
– the result of the tasks.

1.5 Review questions

1. What tasks are solved by neural networks, examples of the use of neural networks.
2. The general principle and purpose of the study of neural networks.
3. Features of using the MATLAB® environment for modeling neural networks.

2 LABORATORY WORK №2
FACE RECOGNITION

The purpose of the work: to learn how to create programs for facial recognition in Python using the OpenCV library.

2.1 Theoretical information

Automatic identification of a person's personality by his image in a photo or video stream has wide commercial and scientific applications. This topic appeared in the early 80th, however, its rapid development began in the 90th, after the creation of new technologies in the field of image processing and computers. This technology is of particular interest due to the fact that it can contactless.

Currently, interest in various methods of identifying individuals has increased. To solve this problem, it is necessary to perform two stages: identify the face in the photo, highlight and recognize it.

With all the variety of different algorithms and methods of image recognition, a typical recognition method consists of three components:

– transformation of the original image into the initial representation (may include both preprocessing and mathematical transformations, for example, the calculation of the main components);

– selection of key characteristics (for example, the first n main components or coefficients of discrete cosine transformation are taken);

– classification mechanism (modeling): cluster model, metric, neural network, etc.

In addition, the construction of the recognition method is based on a priori information about the subject area (in this case, the characteristics of a person's face), and is corrected by experimental information that appears during the development of the method.

2.2 Setting up the software environment

First you need to download and install the PyCharm IDE. After that, you need to create a new project. Next, you need to install the opencv library. To do this, in the program terminal enter the command:

```
pip install opencv-python
```

Next, in the program window, import this library with the command:

```
import cv2
```

Given that opencv is a free and open source library, many classifiers can be found on github. Follow the link https://github.com/opencv/opencv and open the data folder, in the same place we select existing classifiers.

For our task, select the face definition classifier from the haarcascade folder:

```
haarcascade_frontalface_default.xml
```

Next, you need to download this file and transfer it to the current project by path:

```
Lib\site-packages\cv2\data
```

After that, you need to create an object in Python (the path to the selected file is written in brackets):

```
face_cascade_db =
cv2.CascadeClassifier(cv2.data.haarcascades+"haarcascade
_frontalface_default.xml")
```

Next, you need to upload to the project any photo with a face in the current project directory.

After that, you need to create an object for this image:

```
img = cv2.imread("img.jpg")
```

In order for our image to correctly recognize the existing classifier, it is necessary to convert it into shades of gray:

```
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

We display this image on the screen. To do this, enter the imshow command (window name, image name) and the delay cv2.waitKey():

```
cv2.imshow('rez', img_gray)
cv2.waitKey()
```

And run the project for execution (right click - Run). As a result, we will see the image in gray tints (Fig. 2.1).

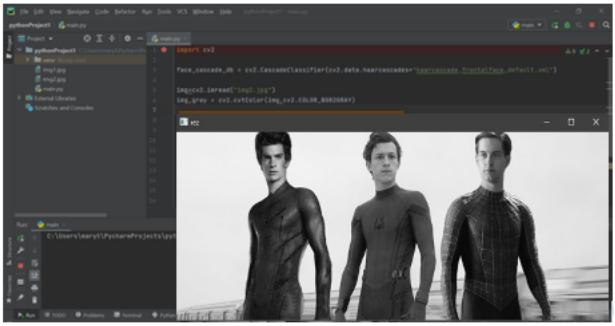

Figure 2.1 – Display the image in gray tints

Next, we will create an object faces, in which we will write the recognized faces using the classifier:

```
faces = face_cascade_db.detectMultiScale(img_gray, 1.1, 19)
```

With the help of the face object, we will select the list of found coordinates in this image.

Next, we display a square on the found faces and not a black and white, but a color, image. The full program code is listed in listing 2.1.

Listing 2.1 – Photo facial recognition

```
import cv2

face_cascade_db =
cv2.CascadeClassifier(cv2.data.haarcascades+"haarcascade_fron
talface_default.xml")

img = cv2.imread("family.jpg")
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

faces = face_cascade_db.detectMultiScale(img_gray, 1.1, 19)
for (x, y, w, h) in faces:
```

```
    cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)

cv2.imshow('rez', img)
cv2.waitKey()
```

After that, we will launch the project and as a result we get a program that recognizes the face in the photo (fig. 2.2).

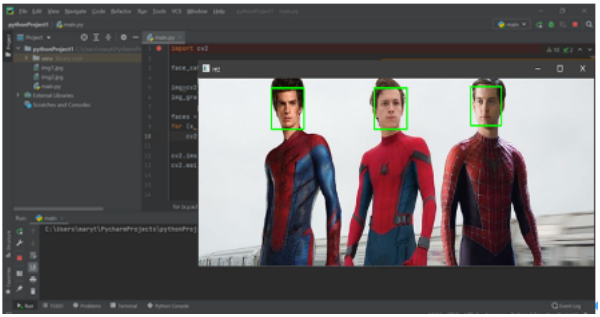

Figure 2.2 – The result of face recognition in the photo

2.3 Facial recognition from the video stream

You need to write a program that is able to recognize faces from a webcam stream.

To do this, create a new file in the project. As a source of information, we enter a stream from a webcam:

```
    cap = cv2.VideoCapture(0)
```

Next, in the loop, we get an image from a webcam:

```
while True:
  success, img = cap.read()
  img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

  faces = face_cascade_db.detectMultiScale(img_gray, 1.1, 19)
  for (x, y, w, h) in faces:
      cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

Function for exiting the cycle by pressing the "q" button:

```
    cv2.imshow('rez', img)
    if cv2.waitKey() & 0xff == ord('q'):
    break
```

The full program code is listed in listing 2.2.

Listing 2.2 – Facial recognition program code from webcam stream

```
import cv2

face_cascade_db =
cv2.CascadeClassifier(cv2.data.haarcascades+"haarcascade_fron
talface_default.xml")

cap = cv2.VideoCapture(0)
while True:
  success, img = cap.read()
  img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

  faces = face_cascade_db.detectMultiScale(img_gray, 1.1, 19)
  for (x, y, w, h) in faces:
      cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)

  cv2.imshow('rez', img)
  if cv2.waitKey() & 0xff == ord('q'):
      break

cap.release()
cv2.destroyAllWindows()
```

As a result, frames from the webcam with a circled face on it will be displayed alternately if successful (Fig. 2.3).

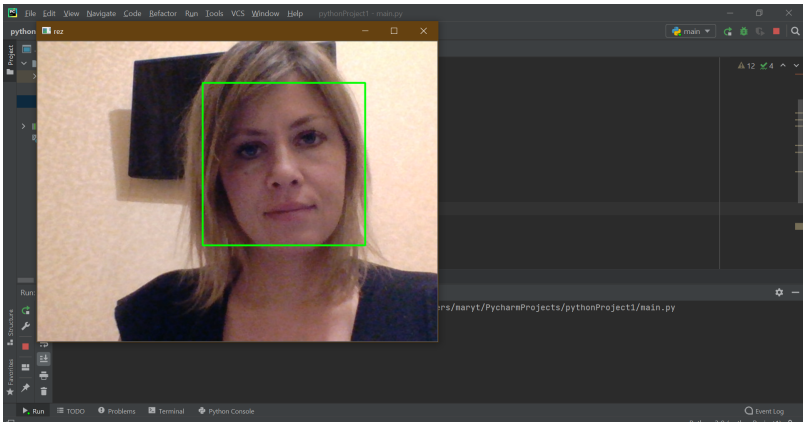

Figure 2.3 – The result of face recognition from a webcam stream

As you can see from the results, the program will read the information from the webcam stream, recognize the face on it and wrap it around the deno with a green frame.

2.4 Tasks to complete

Run examples 2.2, 2.3. Demonstrate the results of the performance to the teacher.

An additional (optional) task is:

- Create a face recognition app from a recorded video.

Demonstrate the results of the performance to the teacher.

Based on the results of the tasks, conclusions should be drawn and cited in the report.

2.5 Contents

The report should have the following components:

– title page;

– the purpose of the work;

– the result of the tasks.

2.6 Review questions

1. Very briefly describe the software products that exist for facial recognition.
2. Briefly describe the Python libraries that are designed for recognition.
3. Describe Python libraries for working with neural networks.
4. Features of convolutional neural networks.

3 LABORATORY WORK №3
APPLICATION OF GENETIC ALGORITHMS

The purpose of the work: to learn how to apply genetic algorithms in practice.

3.1 Theoretical information

The foundations of the theory of genetic algorithms were formulated by Holland in his fundamental work and were subsequently developed by a number of researchers. The most famous and cited at present is the monograph of D. Goldberg, which systematically outlines the main results and areas of practical application of genetic algorithms.

Genetic algorithms use principles and terminology borrowed from biological science – genetics. In genetic algorithms, each individual presents a potential solution to a problem. In the classical genetic algorithm, an individual is encoded by a string of binary symbols – a chromosome, each bit of which is called a gene. The plurality of individuals – potential solutions make up the population. The search for (sub) an optimal solution to the problem is performed in the process of population evolution – the sequential transformation of one finite set of solutions into another using genetic operators of reproduction, crossover and mutation.

Evolutionary computing uses the following three mechanisms of natural evolution:

– the first principle is based on the concept of survival of the strongest and natural selection according to Darwin which was formulated by him in 1859 in the book "The Origin of Species by Natural Selection". According to Darwin, individuals who are better able to solve problems in their environment survive and multiply more (reproduce). In genetic algorithms, each individual is a solution to a certain problem by analogy with this principle, individuals with the best values of the target (fitness) function have a great chance of surviving and reproducing. Formalization of this principle is given by the reproduction operator;

– The second principle is due to the fact that the descendant chromosome consists of parts obtained from the chromosomes of the parents. This principle was discovered in 1865 by Mendel. Its formalization provides the basis for the crossing operator (crossover);

– the third principle is based on the concept of mutation, discovered in 1900 by de Vre. Initially, this term was used to describe significant (abrupt) changes in the properties of descendants and their acquisition of properties that are absent from parents. By analogy with this principle, genetic algorithms use a similar mechanism to dramatically change the properties of offspring and thereby increase the diversity of individuals in a population (set of solutions).

These three principles form the core of evolutionary computing. Using them, the population (the set of solutions to this problem) evolves from generation to generation.

The genetic algorithm takes a set of parameters of the optimization problem and encodes them with sequences of finite length in some finite alphabet (in the simplest case, the binary alphabet "0" and "1").

A simple genetic algorithm randomly generates an initial population of thongs (chromosomes). The algorithm then generates the next generation (population), using three main genetic operators:

– reproduction operator (CA);

– crossing operator (crossover, OK);

– mutation operator (OM).

Genetic operators are a mathematical formalization of the above three fundamental principles of Darwin, Mendel and de Vre of natural evolution.

The genetic algorithm works until the specified number of generations (iterations) of the evolution process is fulfilled or a given quality is obtained at some generation or due to premature convergence when it enters some local optimum.

In each generation, a set of artificial individuals is created using old ones and adding new ones with good properties. Genetic algorithms are not just a random search, they effectively use the information accumulated in the process of evolution. In the process of finding a solution, it is necessary to maintain a balance between the "operation" of the currently obtained best solutions and the expansion of the search space. Different search methods solve this problem in different ways.

3.2 Task to minimize the function

It is necessary to minimize the function of one variable:

$$f(x) = 8x - 16 - 12\sqrt[3]{(x+4)^2}$$

To do this, in the *MATLAB*® environment, you need to create two scripts ex2.m and ex2_func.m. The first contains the function described above (listing 3.1), the second contains an algorithm for calculating fitness function, finding the best individual and the average inidstan between individuals (listing 3.2).

To create a script, click the button “New Script” in the HOME tab. In the window that opens, enter the required script (List 3.1). To create another script, just click "+" to the right of the script name, as shown in Fig. 3.1.

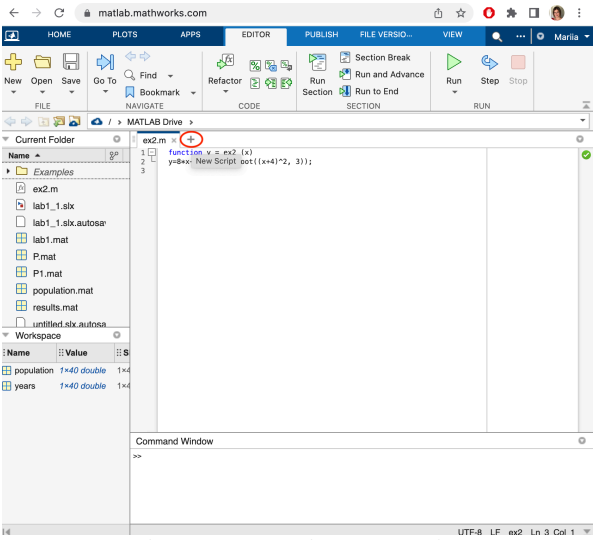

Figure 3.1 – Creating a new script

Listing 3.1 – Minimization Feature

```
function y = ex2 (x)
y=8*x-16-12*(nthroot((x+4)^2, 3));
end
```

Listing 3.2 – Genetic algorithm function

```
function [X,FVAL,REASON,OUTPUT,POPULATION,SCORES] = ex2_func
fitnessFunction = @ex2;
nvars = 1;
```

```
options = optimoptions(@ga,'PopInitRange',[-4 ; 1]);
options = optimoptions(options,'PopulationSize',10);
options =
optimoptions(options,'MutationFcn',{@mutationgaussian 1 1});
options = optimoptions(options,'Display','off');
options = optimoptions(options,'PlotFcns',{@gaplotbestf,
@gaplotbestindiv, @gaplotdistance});
[X,FVAL,REASON,OUTPUT,POPULATION,SCORES] =
ga(fitnessFunction,nvars,options);
end
```

To save the created files, press Save – Save as..., as shown in Fig.3.2, and save the scripts with the appropriate name.

After that, you need to run the created scripts. To do this, in the command line you need to enter the command `optimtool('ga')` – Fig.3.3.

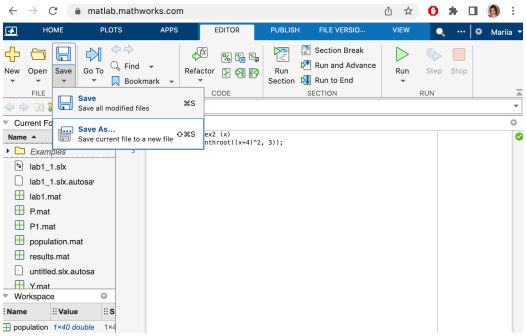

Figure 3.2 – Saving the script

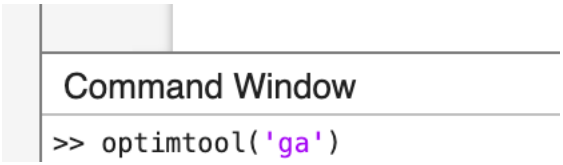

Figure 3.3 – Working in a command window

Next, you need to run the created scripts by clicking Run in the EDITOR tab (Fig.3.4 – circled in red), as a result of which we will get the result in calculating the fitness function, searching for the best individual and the averagedistance between individuals (Fig. 3.4).

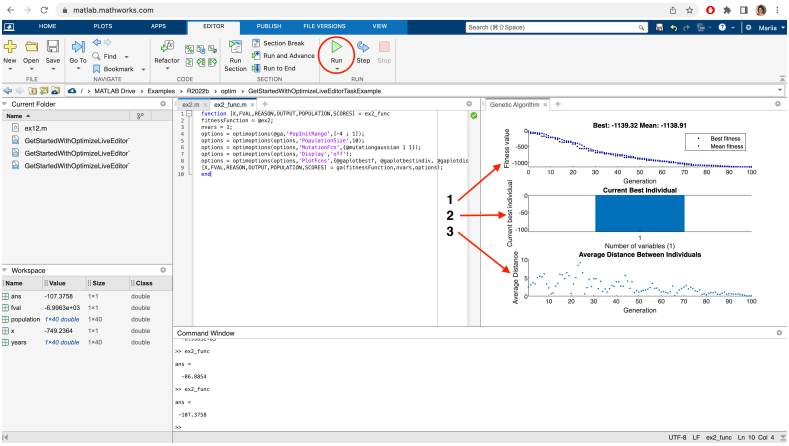

Figure 3.4 – The result of the genetic algorithm

The first figure shows the change in the value of the target function. The second figure shows the best individual. The third figure corresponds to the shifting of the distance between individuals in generations. Individuals become the same (hemming distance = 0) in the last 20 generations.

The genetic algorithm must be run several times, and then choose the optimal solution. This is due to the fact that the initial population is formed using a random number generator.

3.3 The task to maximize the function

It is necessary to maximize the function of two variables:

$$z(x, y) = \exp(-x^2 - y^2) + \sin(x + y)$$

An application with genetic algorithms solves only minimization problems, therefore, to find the maximum function f(x), the function -f (x) should be minimized. This is because the minimum point -f(x) is some point f(x) at which the maximum is reached.

To do this, in the *MATLAB®* environment, you must re-create two scripts. The first contains the function described above (listing 3.3), the second contains an algorithm for finding the fitness function, the best individual and the average inthe distance between individuals (listing 3.4).

Listing 3.3 – Maximization function

```
function z = ex13 (x)
z=-(exp(-x(1)^2-x(2)^2)+sin(x(1)+x(2)));
end
```

Listing 3.4 – Genetic algorithm function

```
function [X,FVAL,REASON,OUTPUT,POPULATION,SCORES] = ex13_func
fitnessFunction = @ex13;
nvars = 2;
options = optimoptions(@ga,'PopInitRange',[-1 ; 3]);
options = optimoptions(options,'PopulationSize',10);
options =
optimoptions(options,'MutationFcn',{@mutationgaussian 1 1});
options = optimoptions(options,'Display','off');
options = optimoptions(options,'PlotFcns',{@gaplotbestf,
@gaplotbestindiv, @gaplotdistance});
[X,FVAL,REASON,OUTPUT,POPULATION,SCORES] =
ga(fitnessFunction,nvars,options);
end
```

We check the result of calculating fitness function, searching for the two best individuals and the average distance between individuals (Fig. 3.5).

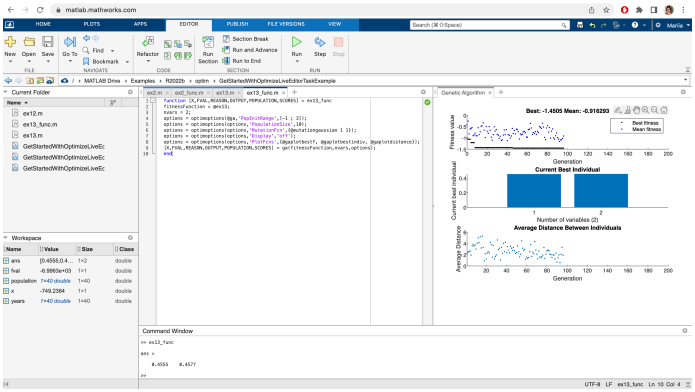

Figure 3.5 – The result of the genetic algorithm for a function with two variables

3.4 Tasks to complete

1. Run examples 3.2, 3.3. Demonstrate the results of the performance to the teacher.
2. Additional tasks are:

– find maximum function:

$$y(x) = 3\sqrt[3]{(x+4)^2} - 2x - 8$$

– find the minimum function:

$$z(x, y) = (y - 3) * \exp(-x^2 - y^2)$$

Demonstrate the results of the performance to the teacher.

Based on the results of the tasks, conclusions should be drawn and cited in the report.

3.5 Contents

The report should have the following components:
– title page;
– the purpose of the work;
– the result of the tasks.

3.6 Review questions

1. Fundamental principles of evolutionary calculations.
2. What is the genetic algorithm?
3. The main operators of genetic algorithms.
4. Areas of use of genetic algorithms.
5. Expand the concept of "parallel genetic algorithms."

LITERATURE

1. Shakhovska N.B. Artificial Intelligence Systems / Shakhovska N.B., R.M. Kaminsky, O.B. Vovk - Lviv: Lviv Polytechnic, 2018. - 392 c.

2. Official MATLAB® user support page / Access mode: https://www.mathworks.com/help/matlab/index.html

3. Wilson H. Artificial intelligence. / H. Wilson. - Grey House Publishing. 2018.