

**Лабораторна робота No 5**  
**МОДЕЛЮВАННЯ НЕЙРОННИХ МЕРЕЖ В MATLAB. КОМАНДНИЙ РЕЖИМ. GUI**  
**ИНТЕРФЕЙС. ПАКЕТ SIMULINK**

Виконала: студентка групи МІТ-41  
Півторак Каріна

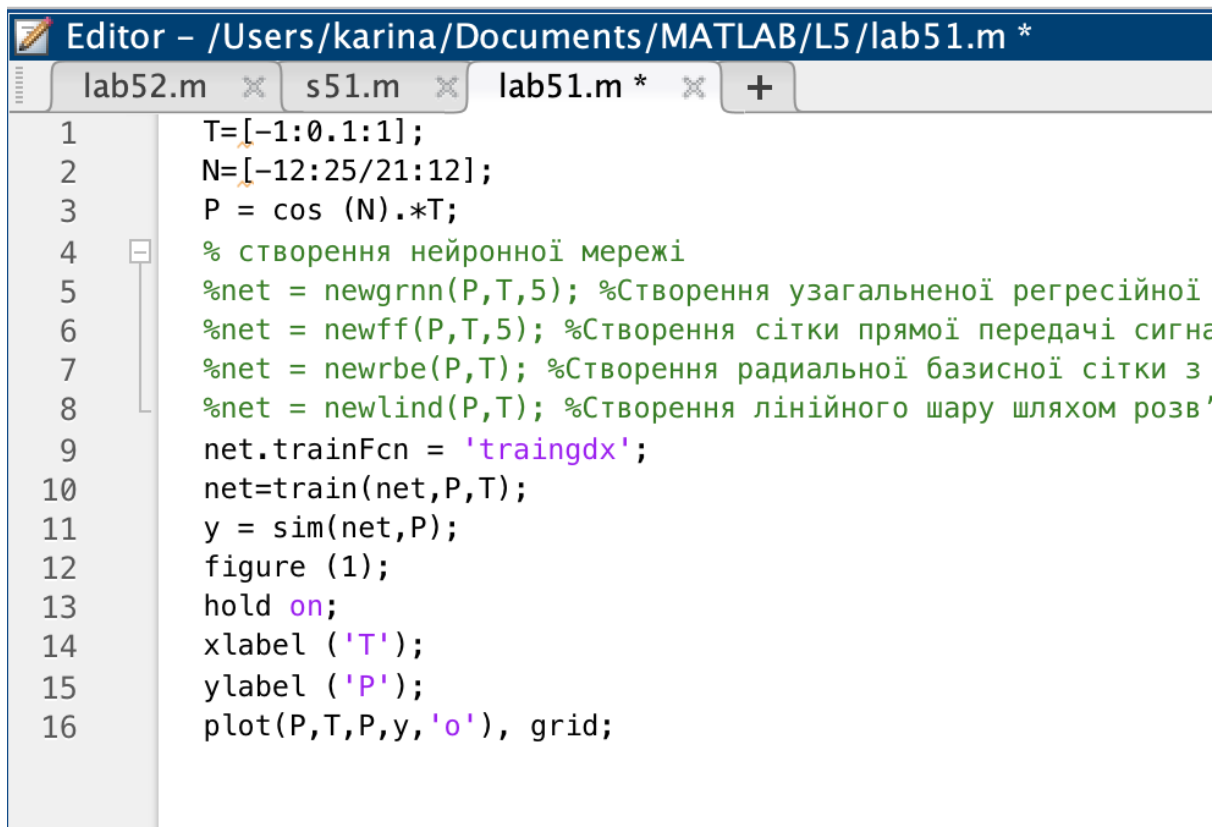
Хід роботи

Варіант 17

17	$y = x_1 \cdot \cos x_2, \quad x_1 \in [1;10],$ $x_2 \in [-90^\circ; 90^\circ]$
----	--

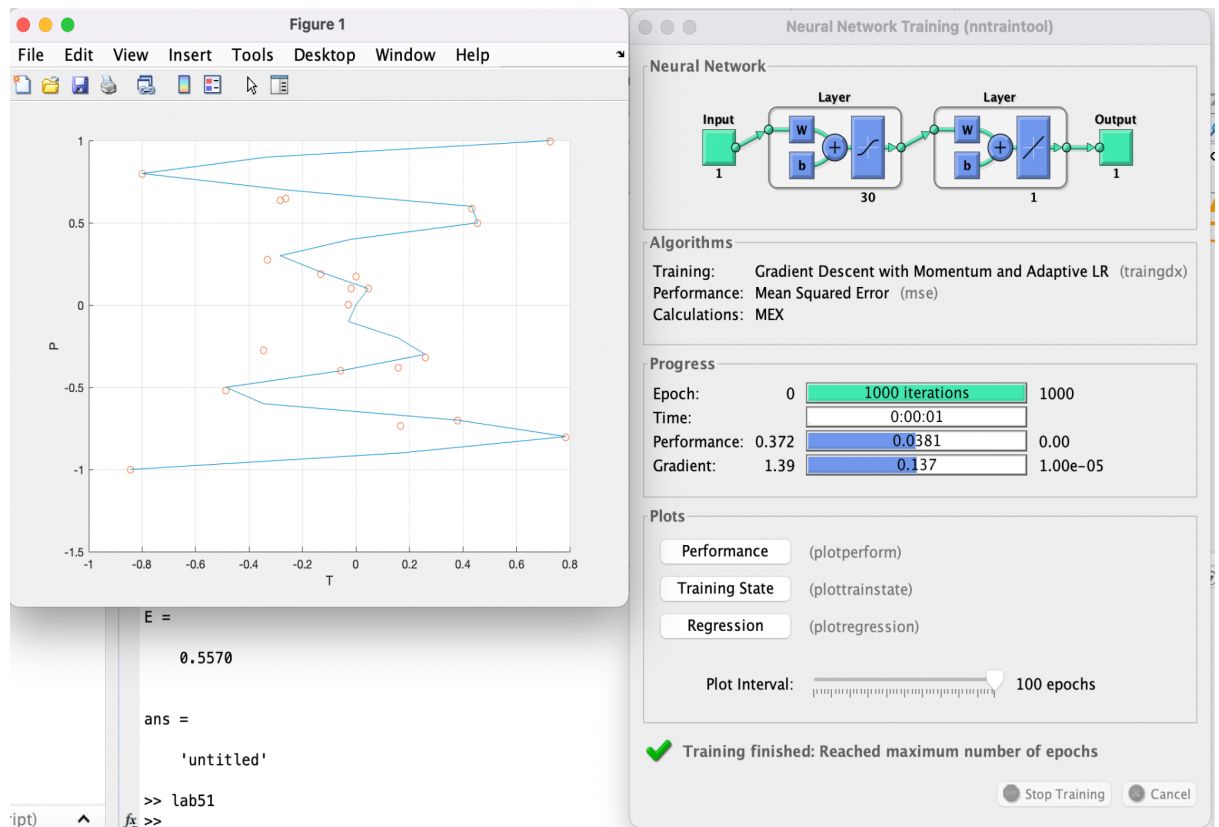
Завдання 1.

1. Побудувати нейронну мережу для апроксимації заданої за варіантом функції (таблиця 5.1), використовуючи функцію `newgrnn` пакета Neural Networks Toolbox (Приклад 5.1). Розрахувати максимальну відносну похибку апроксимації. 2. Поліпшити якість апроксимації шляхом побудови нейронної мережі використовуючи функцію `newtbe`. Розрахувати максимальну відносну похибку апроксимації. 3. Створити лінійну нейронну мережу використовуючи функцію `newlind`. Розрахувати максимальну відносну похибку апроксимації та порівняти з пунктом 1 та 2.



```
Editor - /Users/karina/Documents/MATLAB/L5/lab51.m *
lab52.m x s51.m x lab51.m * x +
1      T=[-1:0.1:1];
2      N=[-12:25/21:12];
3      P = cos (N).*T;
4      % створення нейронної мережі
5      %net = newgrnn(P,T,5); %Створення узагальненої регресійної
6      %net = newff(P,T,5); %Створення сітки прямої передачі сигна
7      %net = newrbe(P,T); %Створення радіальної базисної сітки з
8      %net = newlind(P,T); %Створення лінійного шару шляхом розв'
9      net.trainFcn = 'traingdx';
10     net=train(net,P,T);
11     y = sim(net,P);
12     figure (1);
13     hold on;
14     xlabel ('T');
15     ylabel ('P');
16     plot(P,T,P,y,'o'), grid;
```

25/21 – це кількість точок у  $N(25, \text{від } -12 \text{ до } 12)$  поділити на кількість точок у  $T(21, \text{від } -1 \text{ до } 1)$



## Завдання 2.

Створити нейронну мережу і виконати апроксимацію функції відповідно варіанту (табл. 5.1) з урахуванням впливу шуму, розподіленого за нормальним законом. Вибір варіанта завдання здійснюється за номером по списку групи. (Приклад 5.4)

Код проекту:

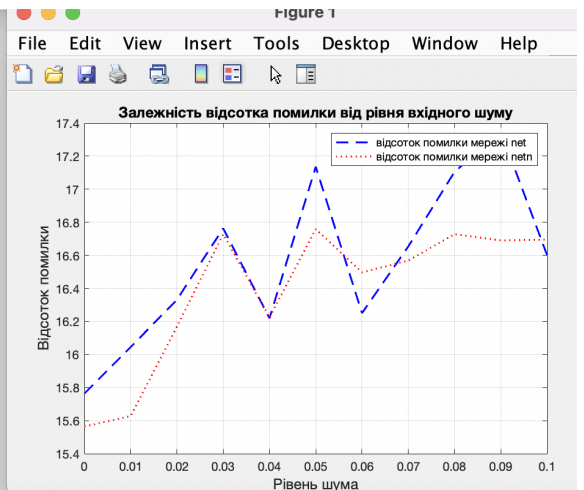
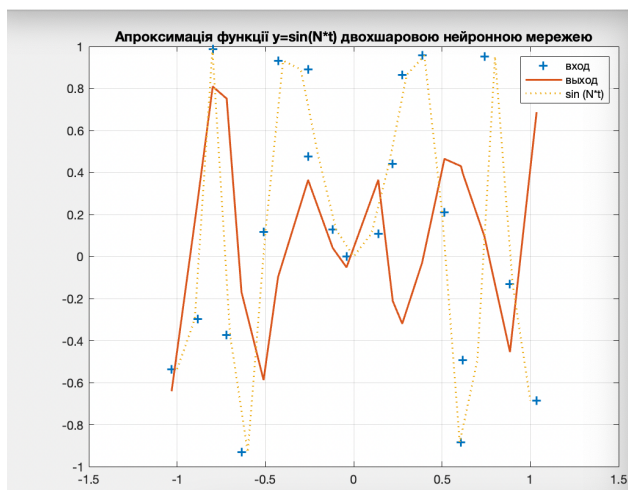
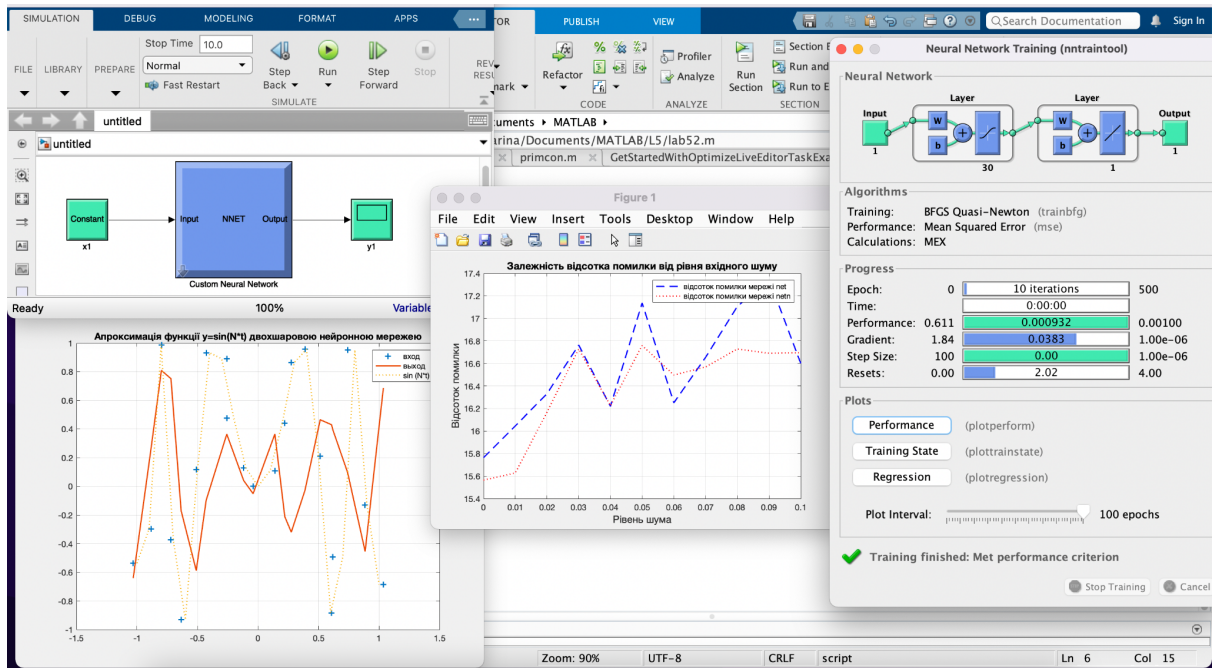
```

Editor - /Users/karina/Documents/MATLAB/L5/lab52.m
lab52.m x s51.m x lab51.m x +
1 clear
2 clc
3 %----- Створення і навчання мережі при відсутності шуму -----
4 p = [-1:0.1:1]; %вхідний вектор мережі
5 N=[-12:25/21:12];
6 t = cos (N).*p;
7 % вихідний вектор мережі
8 net = newff(minmax(p), [30,1], {'tansig', 'purelin'}, 'trainbfg'); %створення
9 net.trainParam.epochs = 500; % завдання кількості циклів навчання
10 net.trainParam.show = 50; %кількість циклів для показу проміжних
11 net.trainParam.goal = 1e-3; %цільова помилка навчання
12 [net, tr] = train(net, p, t);
13 netn = net;% ініціалізація мережі
14 netn.trainParam.epochs = 500; % завдання кількості циклів навчання
15 netn.trainParam.show = 50; % кількість циклів для показу проміжних
16 netn.trainParam.goal = 1e-3; % цільова помилка навчання
17 t1 = [sin(N.*p) sin(N.*p)]; %вихідний вектор мережі
18 p1 = [p, (p+randn (size (p))*0.02)]; %вхідний вектор мережі з шумом
19 [netn, tr]=train(netn, p1, t1); % навчання мережі за наявності шуму
20 %----- Повторне навчання мережі при відсутності шуму -----
21 netn.trainParam.goal=1e-3; % цільова помилка навчання
22 [netn, tr] = train (netn, p, t); % повторне навчання мережі при відсутності шуму
23 %----- Оцінка ефективності функціонування системи -----
24 tic % встановлюємо таймер
25 noise = 0:0.01:0.1; % різні рівні шуму

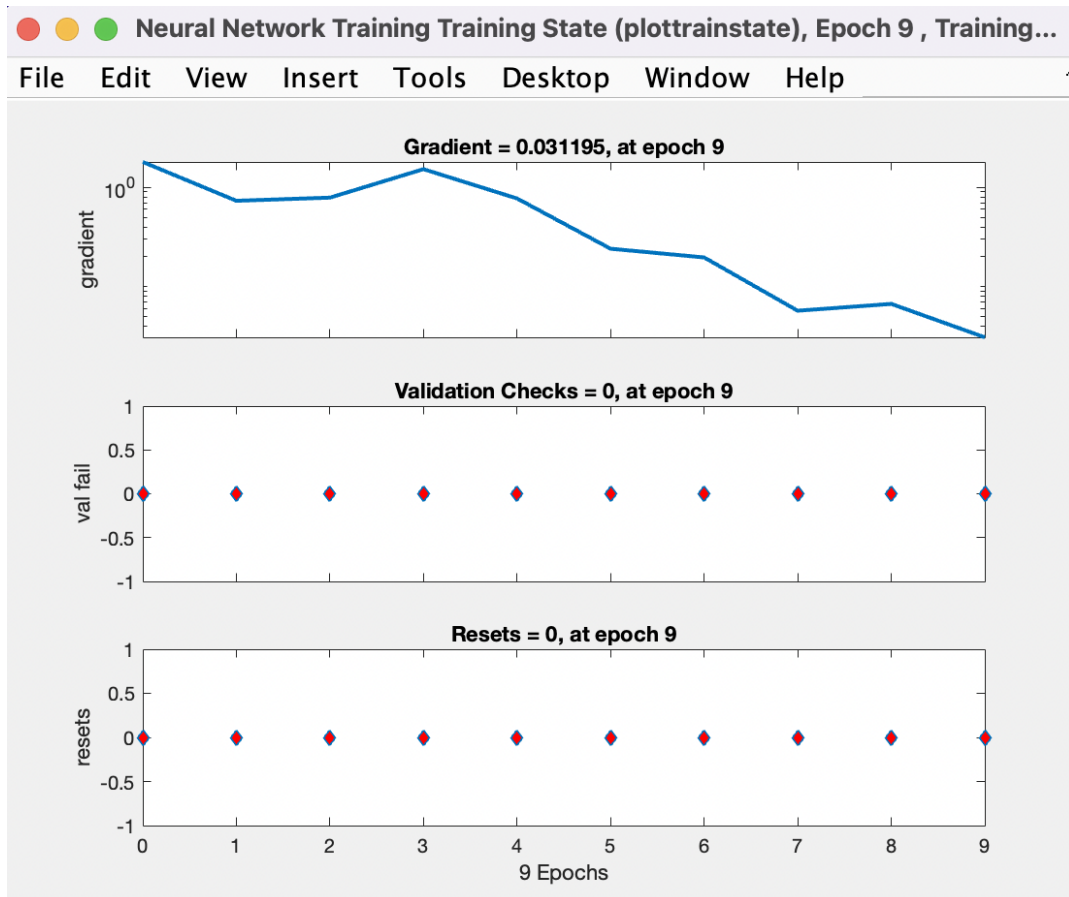
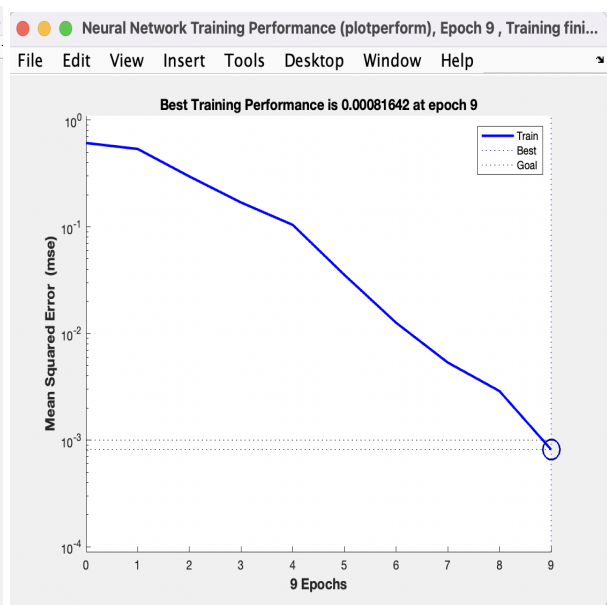
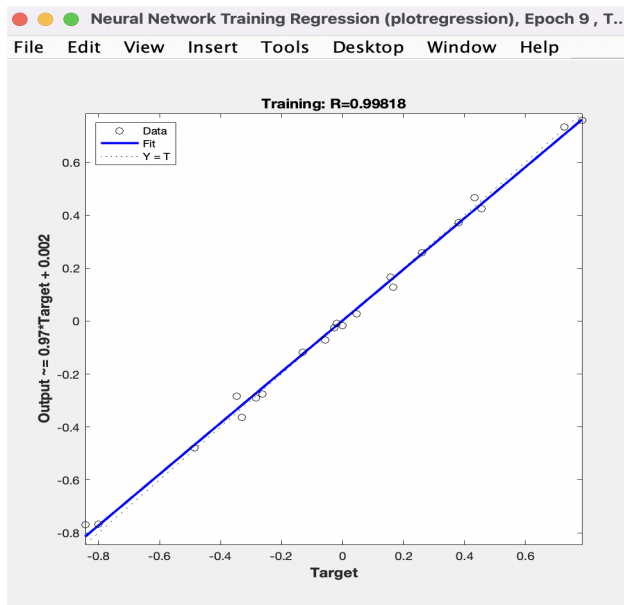
network2 = [network2 errors2/41/20];
end
network1 ;
network2;
toc
figure(1)
plot (noise, network1*100, '--b', noise, network2*100, ':r', 'LineWidth', 1.5);
% графіки
% залежності відсотка помилки від рівня шуму
legend('Відсоток помилки мережі net', 'Відсоток помилки мережі netn');
%легенда
xlabel ('Рівень шуму','FontSize',12);
ylabel ('Відсоток помилки','FontSize',12);
title('Залежність відсотка помилки від рівня вхідного шуму', 'FontSize', 12, 'FontWeight', 'bold');
grid on
%----- Використання нейронної мережі -----
p2=randn (size (p))*0.03 + p; % вхідний вектор з шумом
[an,E]=sim(net, p2); % використання нейронної мережі
figure(2);
plot (p2, t, '+', p2, an, '-', p, t, ':', 'LineWidth', 1.5) % графіки
legend('вход', 'выход', 'sin (N*t)'); %легенда
title ('Апроксимація функції y=sin(N*t) двошаровою нейронною мережею', 'FontSize', 11.5, 'FontWeight', 'bold'); %заголовок
grid on %координатна сітка
E=mse (an-t) % середньоквадратична помилка використання нейронної мережі
gensim (net) % структурна схема нейронної мережі

```

Результат програми:



Регресия и перформанс



Помилка:  $E = 0.3002$

Висновок: На даній лабораторній роботі я навчився побудові нейронної мережі для апроксимації обраної функції, використовуючи функції пакета Neural Networks Toolbox.