

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

Кафедра комп'ютерних наук

**Теорія розпізнавання образів та класифікації в системах
штучного інтелекту**

Лабораторна робота №8

Виконав:

Студент групи КН-20002Б

Кропивка Анатолій Анатолійович

Київ 2023

Тема: Дослідження багатошарового нелінійного персептрону і алгоритму зворотнього розповсюдження помилки.

Мета: Дослідження можливостей багатошарового персептрону, як універсального апроксиматора і класифікатора.

Підготовка до роботи: Вивчити й уяснити призначення і зміст завдання до лабораторної роботи.

Алгоритм GD

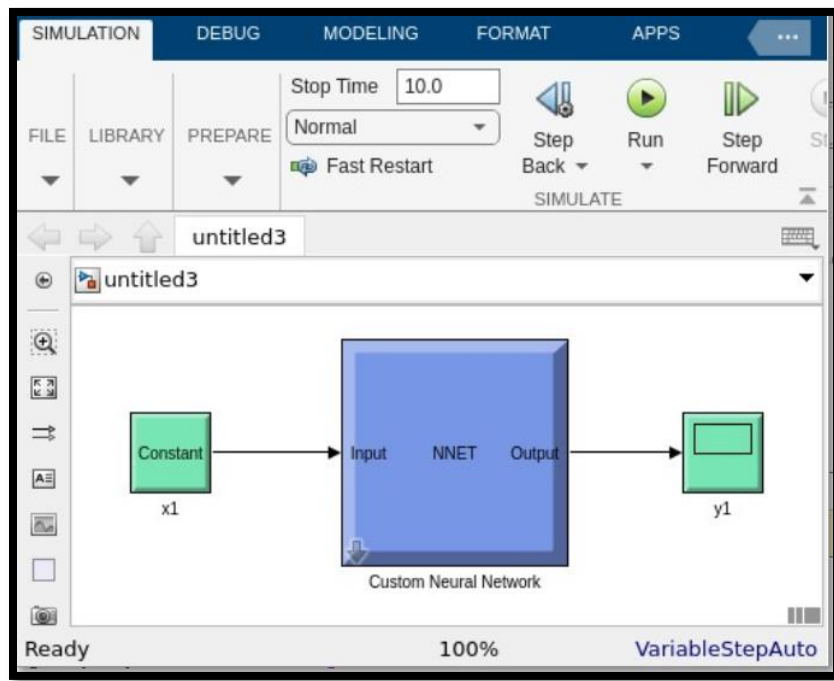
Створення нейронну мережу, щоб забезпечити наступне відображення послідовності входу P у послідовність цілей T:

```
P = [0.10 0.31 0.51 0.72 0.93 1.14 1.34 1.55 1.76 1.96 2.17 2.38 2.59 2.79 3.00];
```

```
T = [0.1010 0.3365 0.6551 1.1159 1.7632 2.5847 3.4686 4.2115 4.6152 4.6095 4.2887  
3.8349 3.4160 3.1388 3.0603];
```

```
net = newff([0 3],[4 1], {'tansig' 'purelin'});
```

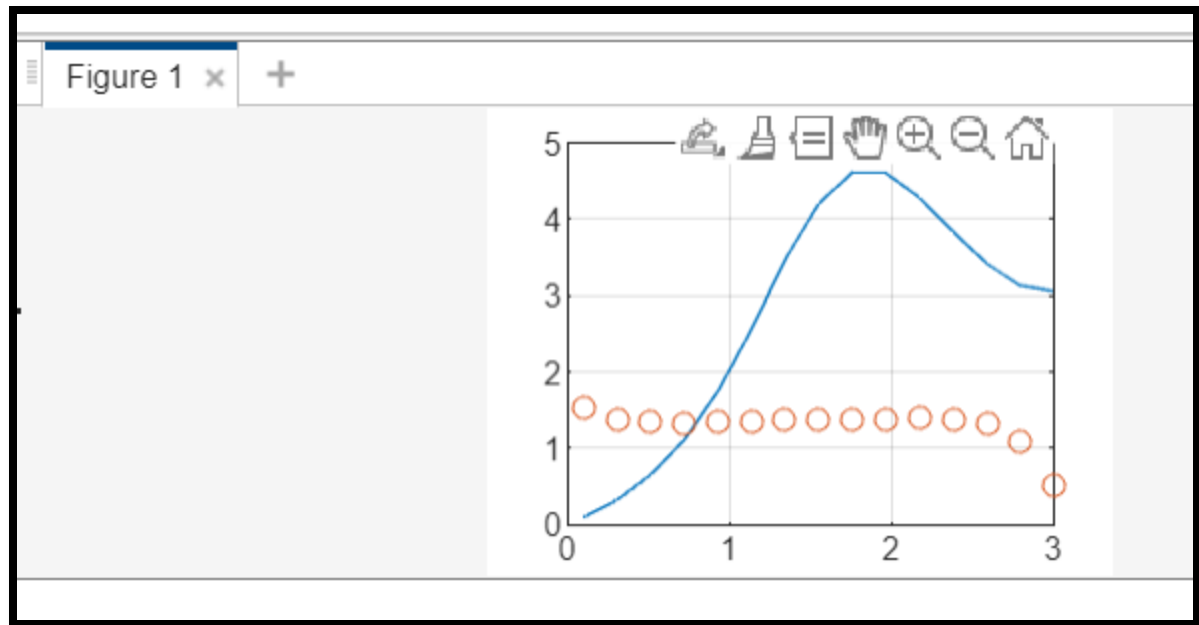
```
gensim(net)
```



Виконаємо моделювання мережі та побудуємо графіки сигналів виходу і цілі:

```
Y = sim (net, P); figure(1), clf
```

```
plot(P, T, P, Y, 'o'), grid on
```



Навчимо мережу протягом 50 циклів:

```
et.trainParam.epochs = 50;
```

```
net = train(net, P, T);
```

Network Diagram

Training Results

Training finished: Reached maximum number of epochs ✓

Training Progress

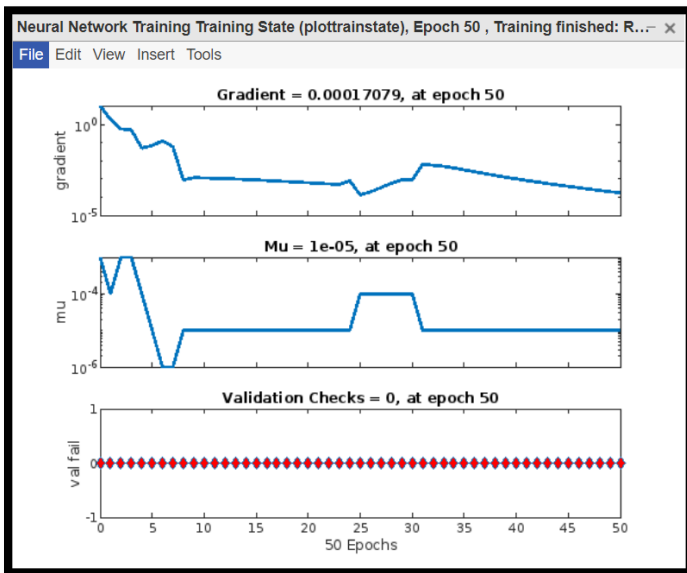
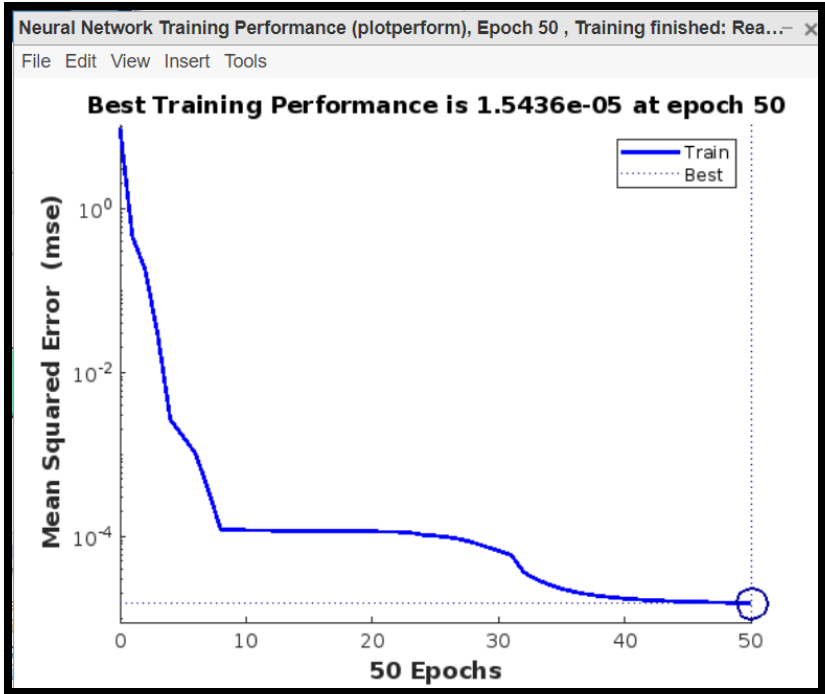
Unit	Initial Value	Stopped Value	Target Value
Epoch	0	50	50
Elapsed Time	-	00:00:01	-
Performance	7.57	2.59e-05	0
Gradient	9.08	0.00912	1e-07
Mu	0.001	1e-05	1e+10
Validation Checks	0	0	6

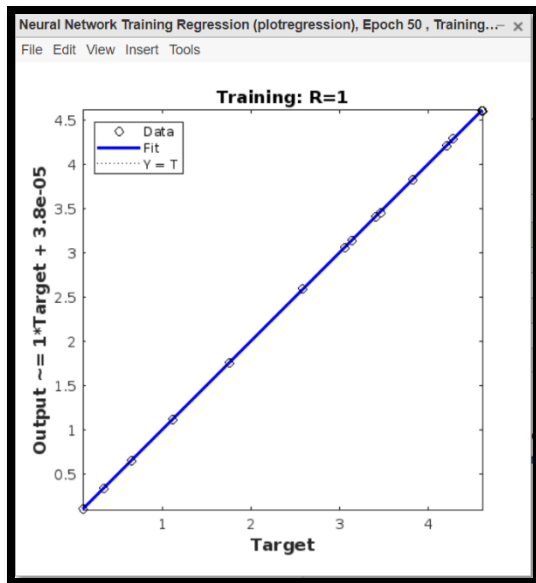
Training Algorithms

Data Division: Levenberg-Marquardt trainlm
 Performance: Mean Squared Error mse
 Calculations: MEX

Training Plots

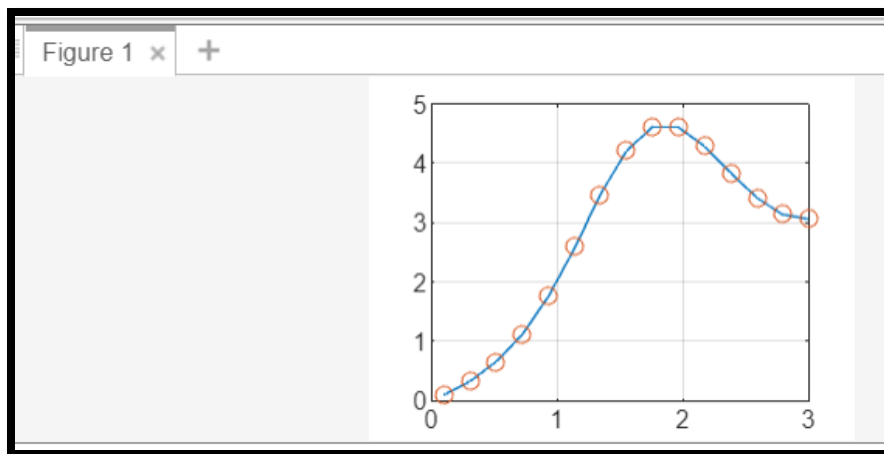
Performance Training State
 Regression



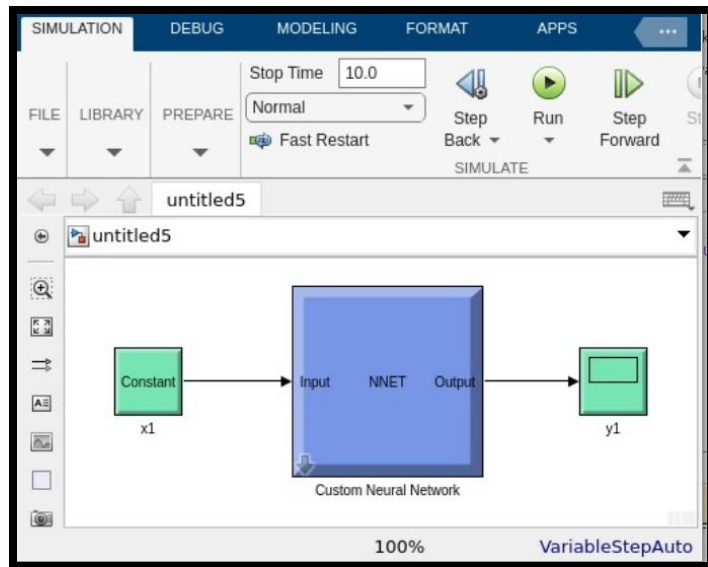


Виконаємо моделювання сформованої двошарової мережі, використовуючи навчальну послідовність входу

```
Y = sim(net, P); plot(P, T, P, Y, 'o'), grid on
```



```
net = newff([-1 1;-1 1],[5,1],{'tansig','purelin'},'traingd');
gensim(net)
```



```

net = newff([-1 1;-1 1],[5,1],{'tansig','purelin'},'traingd');
gensim(net)
x=-1:0.1:1;
y=x;
p = [x; y];
p=num2cell(p,1);
t = sin (x.^ 2)./cos (y.^2);
t=num2cell(t,1);
net.adaptParam.passes=300;
tic, [net, a, e] = adapt (net, p, t); toc
a = sim (net, p)
e=t;
mse(e)
a = sim (net, p)

```

```
untitled3.m x untitled4.m x untitled5.m x untitled6.m x + Figure 1 x +
/MATLAB Drive/untitled5.m
1 net = newff([-1 1;-1 1],[5,1],{'tansig','purelin'},'traingd');
2 gensim(net)
3

Command Window
New to MATLAB? See resources for Getting Started.

'untitled5'
>> net.biases{1,1}.learnFcn='learnkd';
>> net.layerWeights{2,1}.learnFcn='learnkd';
163
net.inputWeights{1,1}.learnFcn='learnkd'; net.layerWeights{2,1}.learnParam.lr = 0.2;

ans =

163
```

```
a =

1x21 cell array

Columns 1 through 12

{[0.9478]} {[0.9221]} {[0.8832]} {[0.8390]} {[0.8017]} {[0.7751]} {[0.7531]} {[0.7235]} {[0.6700]} {[0.5752]} {[0.4365]} {[0.2736]}

Columns 13 through 21

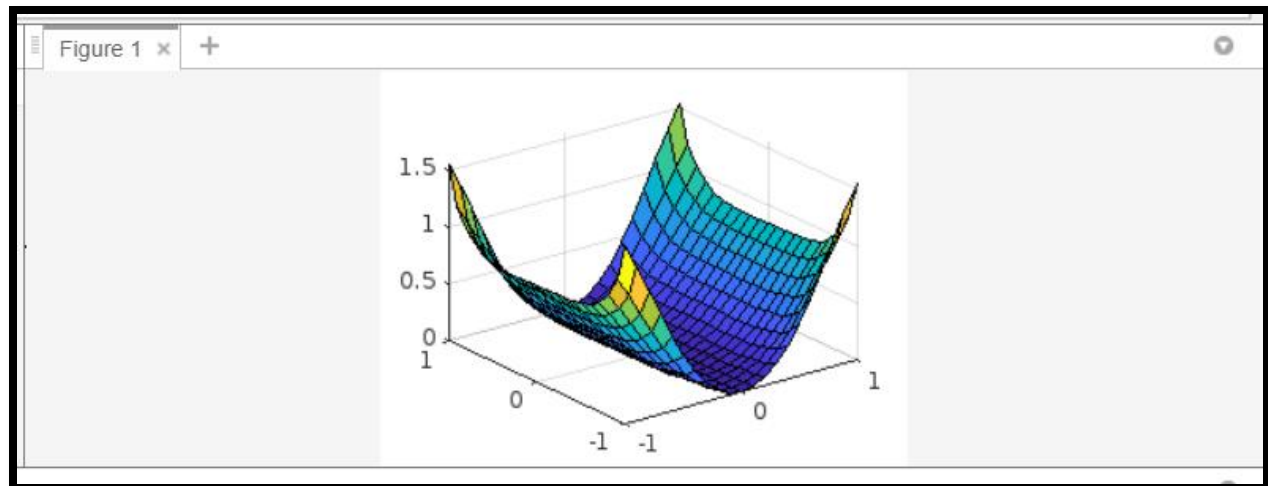
{[0.0967]} {[0.1079]} {[0.3321]} {[0.5225]} {[0.6380]} {[0.6844]} {[0.6871]} {[0.6673]} {[0.6380]}
```

```
>> mse(e)

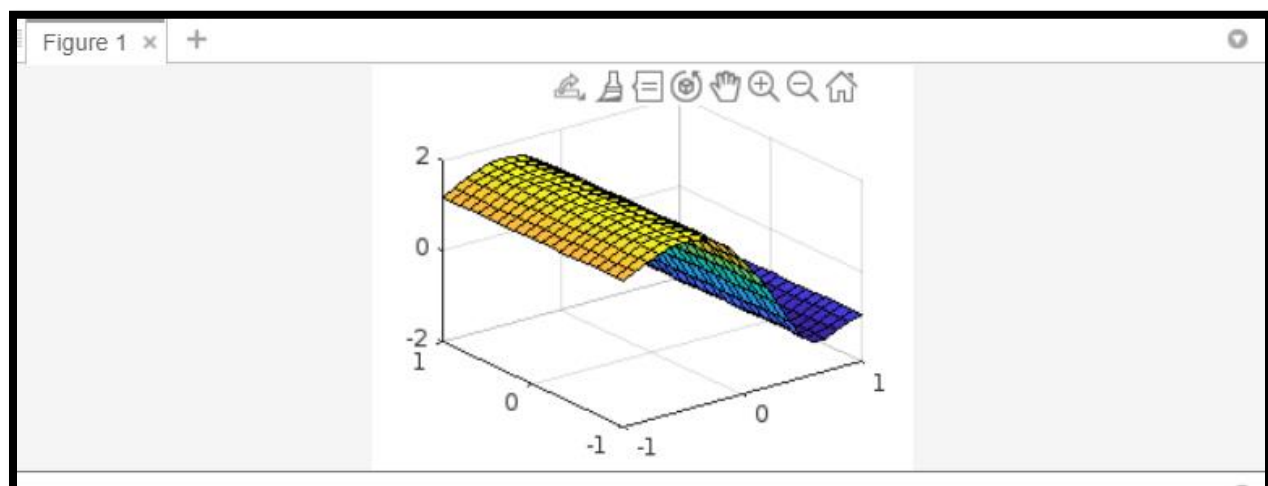
ans =

0.4391
```

```
[x, y] = meshgrid (x, y);
z = sin (x.^2)./ cos (y.^2);
surf(x,y,z)
```

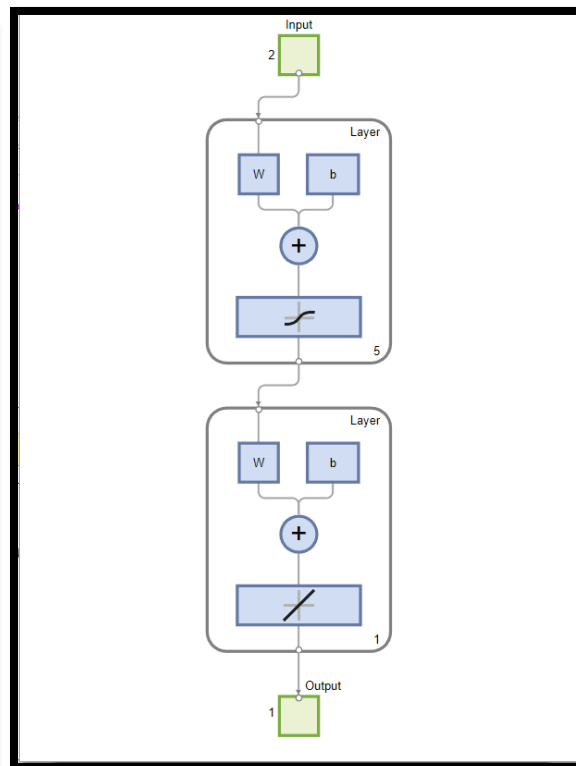
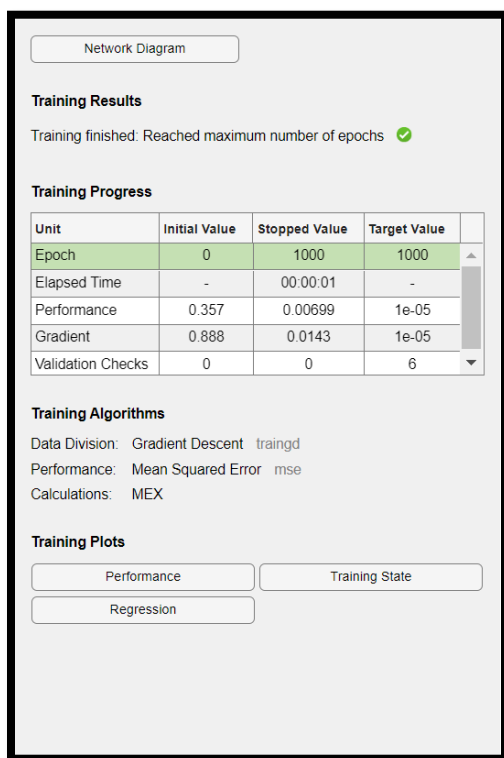
```
a=cat(1,a{:});  
[a]=meshgrid(a);  
surf(x,y,a)
```

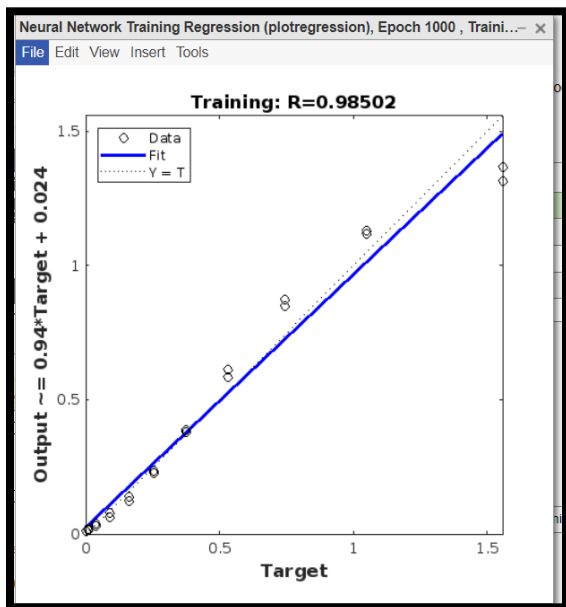
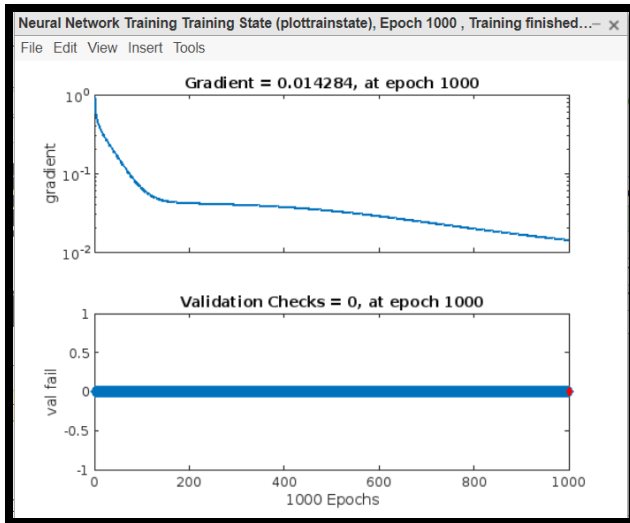
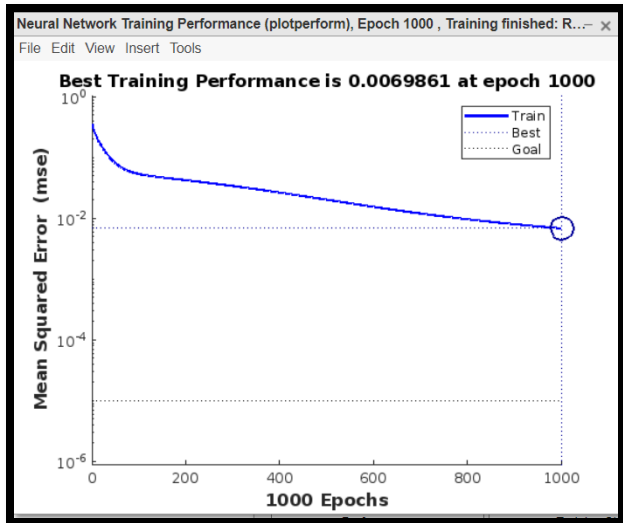


Групове навчання

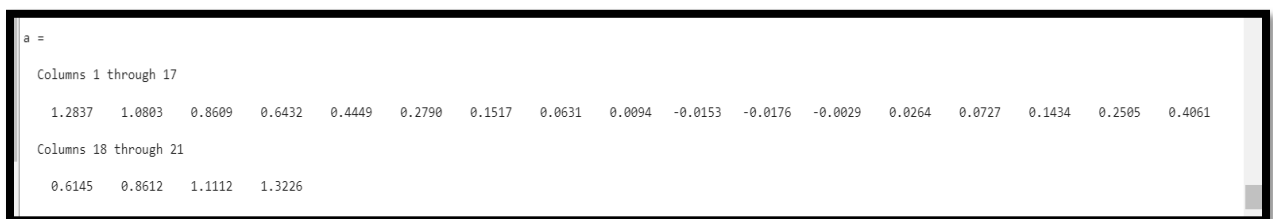
```
net = newff([-1 1;-1 1],[5,1],{'tansig','purelin'},'traingd');  
net.trainParam.show = 50;  
net.trainParam.lr = 0.05;  
net.trainParam.goal = 1e-005;  
x=-1:0.1:1;  
y=x;  
p = [x; y];  
t = sin (x.^ 2)./ cos (y.^2);  
tic, net =train (net, p, t); toc
```

Elapsed time is 1.750972 seconds.





$$a = \text{sim}(\text{net}, p)$$



$$e=t$$

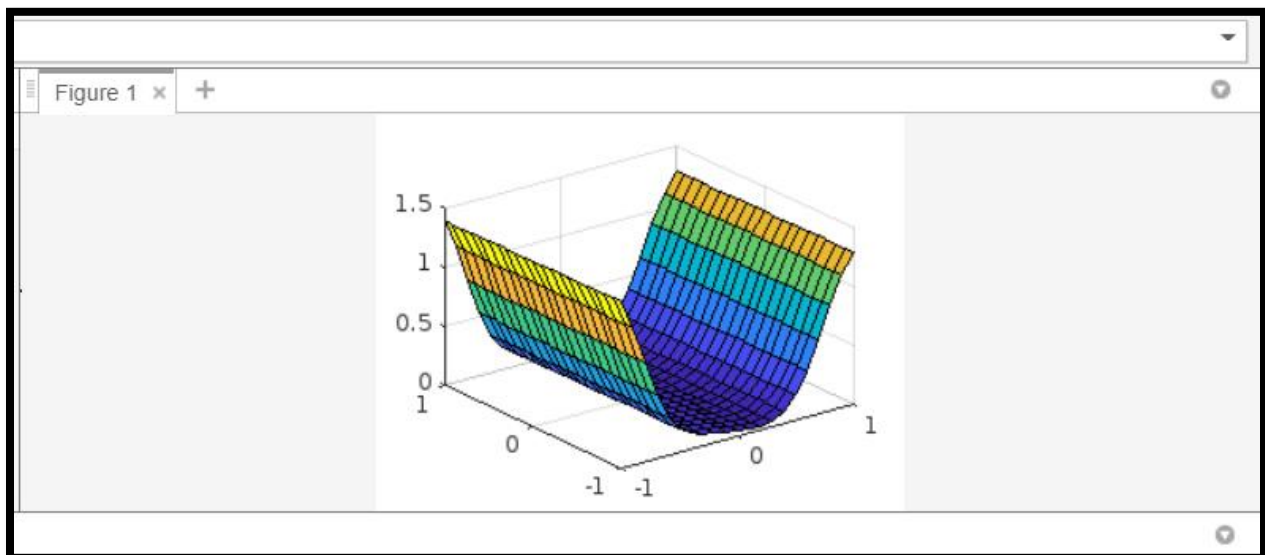
```
>> e=t
e =
Columns 1 through 17
1.5574    1.0505    0.7445    0.5334    0.3764    0.2553    0.1614    0.0902    0.0400    0.0100         0    0.0100    0.0400    0.0902    0.1614    0.2553    0.3764
Columns 18 through 21
0.5334    0.7445    1.0505    1.5574
```

mse(e)

```
>> mse(e)

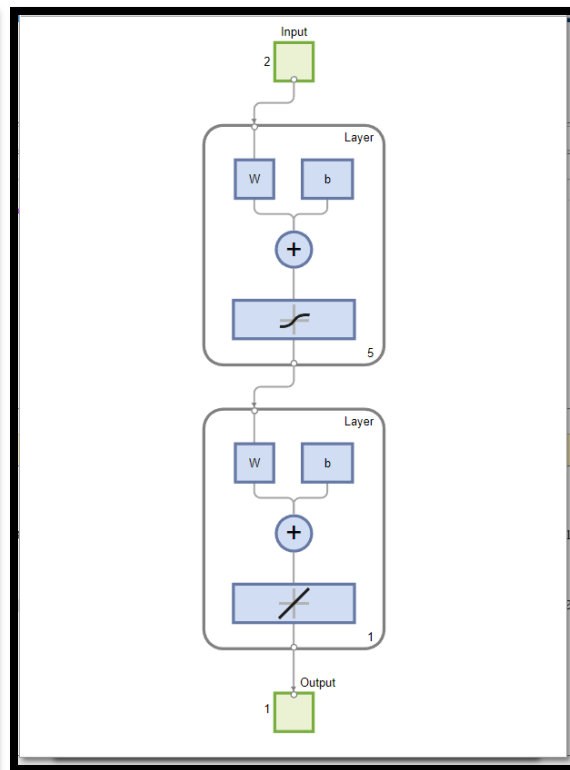
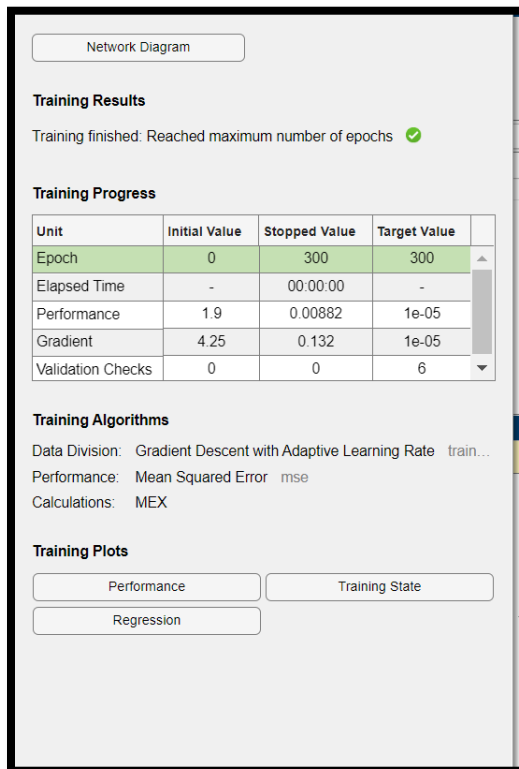
ans =

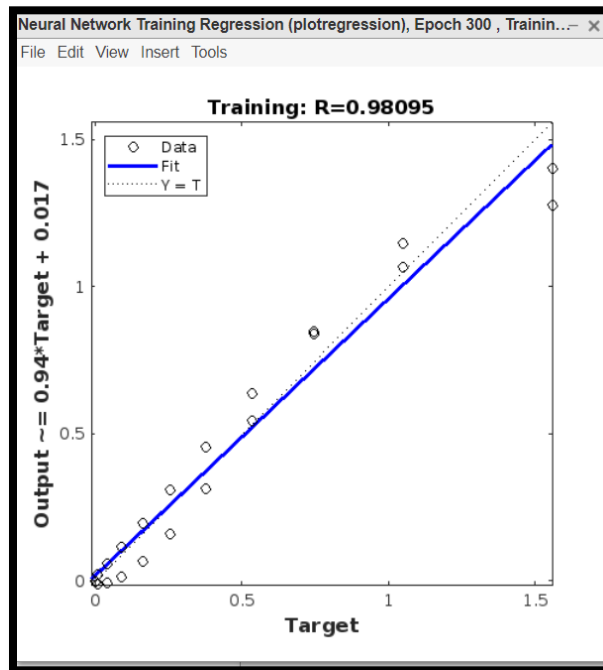
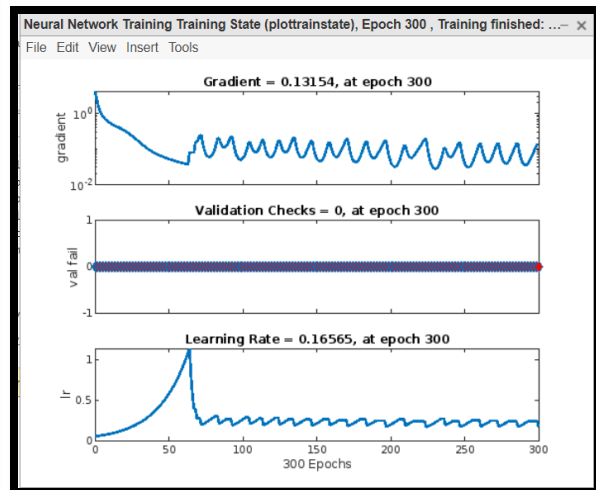
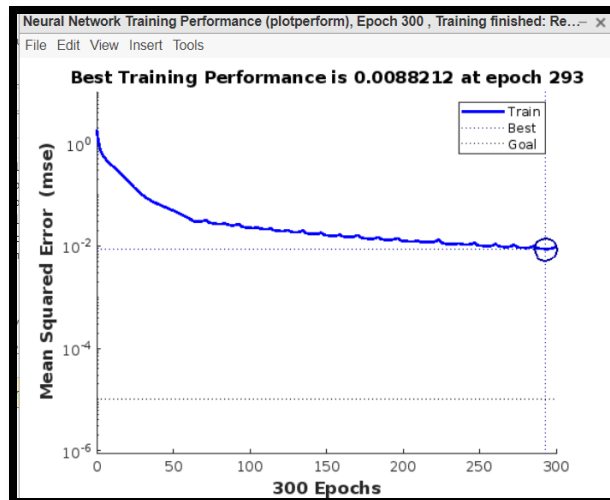
    0.4391
```



Алгоритм GDM

```
net = newff([-1 1;-1 1],[5,1],{'tansig','purelin'},'traingda');  
net.trainParam.epochs=300;  
net.trainParam.goal=1e-5;  
net.trainParam.lr=0.05;  
net.trainParam.mc=0.9;  
net.trainParam.show=50;  
x=-1:0.1:1;  
y=x;  
p=[x; y];  
t = sin (x.^ 2)./ cos (y.^ 2); tic,  
net = train (net, p, t); toc
```





$a = \text{sim}(\text{net}, p);$

```
>> a
a =
Columns 1 through 17
    1.4035    1.1362    0.8410    0.5683    0.3554    0.2104    0.1216    0.0722    0.0480    0.0397    0.0417    0.0515    0.0690    0.0970    0.1421    0.2167    0.3397
Columns 18 through 21
    0.5342    0.8102    1.1369    1.4453
```

$e=t;$

$\text{mse}(e)$

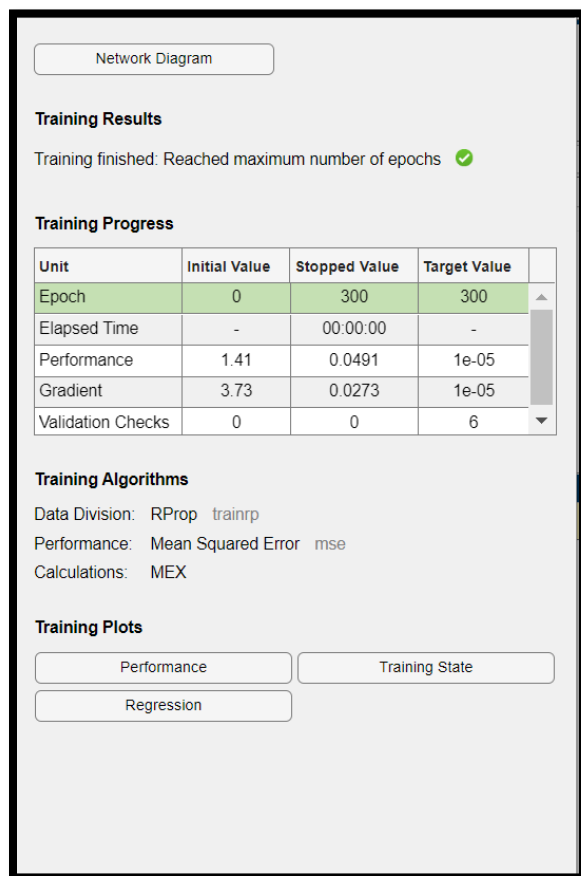
```
>> mse(e)
```

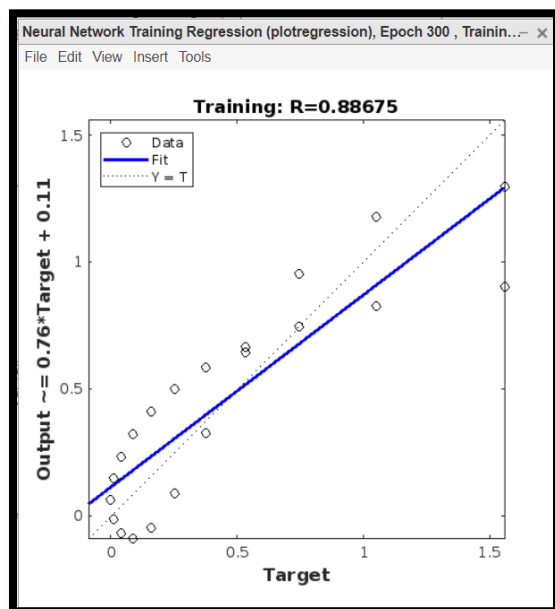
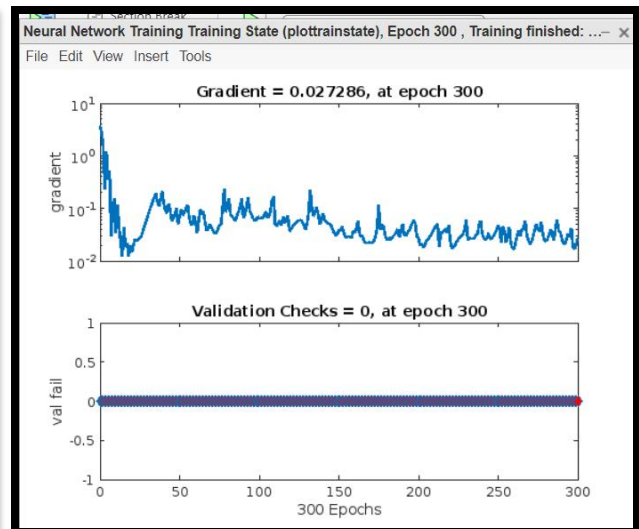
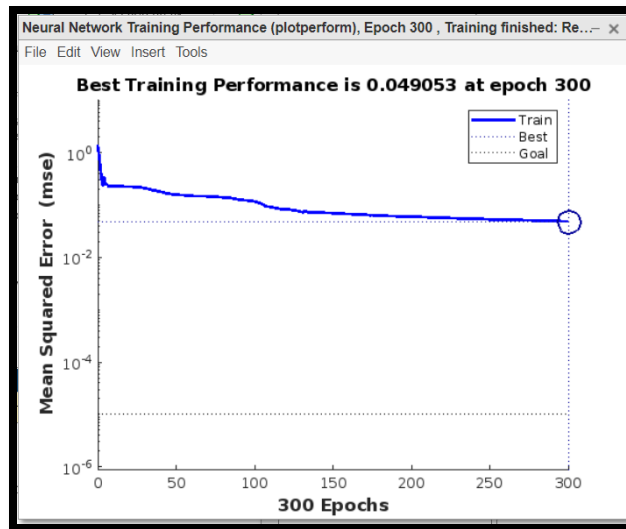
```
ans =
```

```
0.4391
```

Алгоритм Rprop

```
net=newff([-1 2;0 5],[3,1],{'tansig','purelin'},'trainrp');  
net.trainParam.show = 10;  
net.trainParam.epochs =300;  
net.trainParam.goal = 1e-5;  
x=-1:0.1:1;  
y=x;  
p = [x; y];  
t = sin (x.^ 2)./ cos (y.^ 2); tic,  
net =train (net, p, t); toc
```





`a = sim (net, p);`

```
>> a
a =
Columns 1 through 17
    1.2961    1.1771    0.9512    0.6421    0.3267    0.0865   -0.0464   -0.0875   -0.0674   -0.0118    0.0625    0.1461    0.2336    0.3222    0.4105    0.4976    0.5829
Columns 18 through 21
    0.6663    0.7475    0.8264    0.9028
```

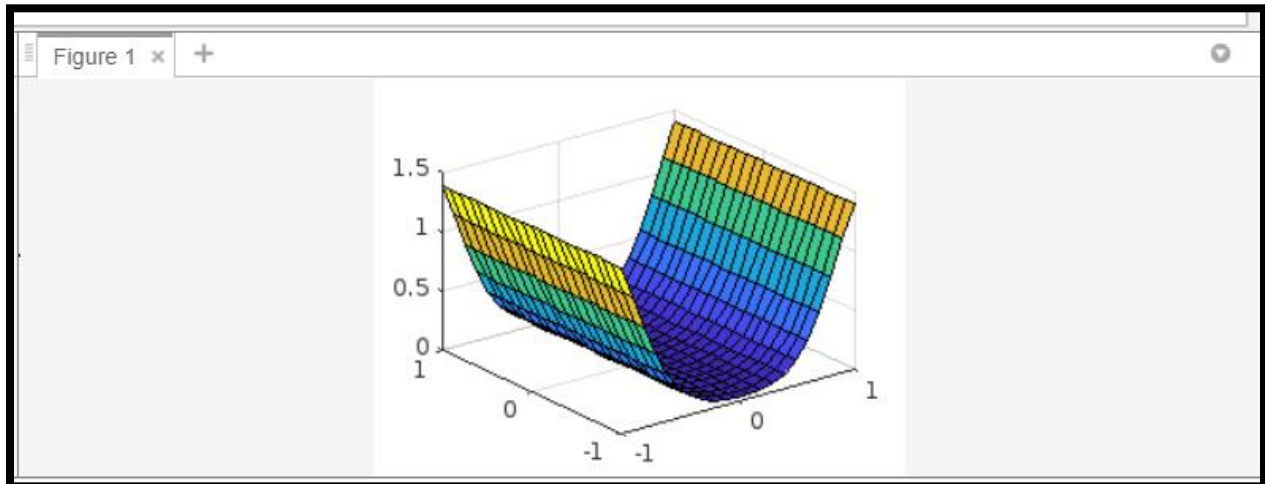
`e=t;`
`mse(e)`

```
>> mse(e)
```

```
ans =
```

```
0.4391
```

```
>>
```



Висновок : При виконанні даної роботи було досліджено багатошарового нелінійного персептрону і алгоритму зворотнього розповсюдження помилки. Розглянули та виконали зазначені завдання.