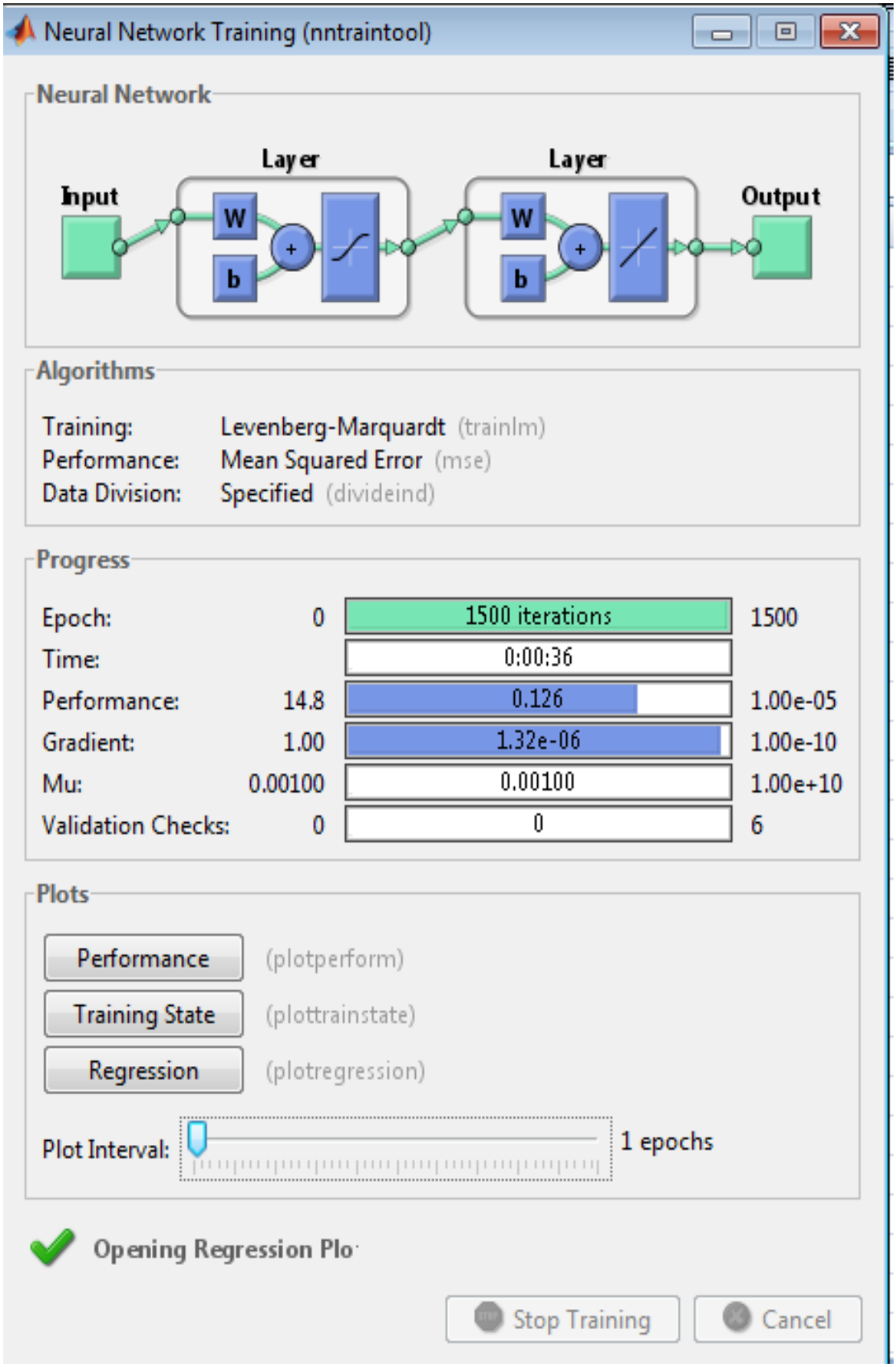
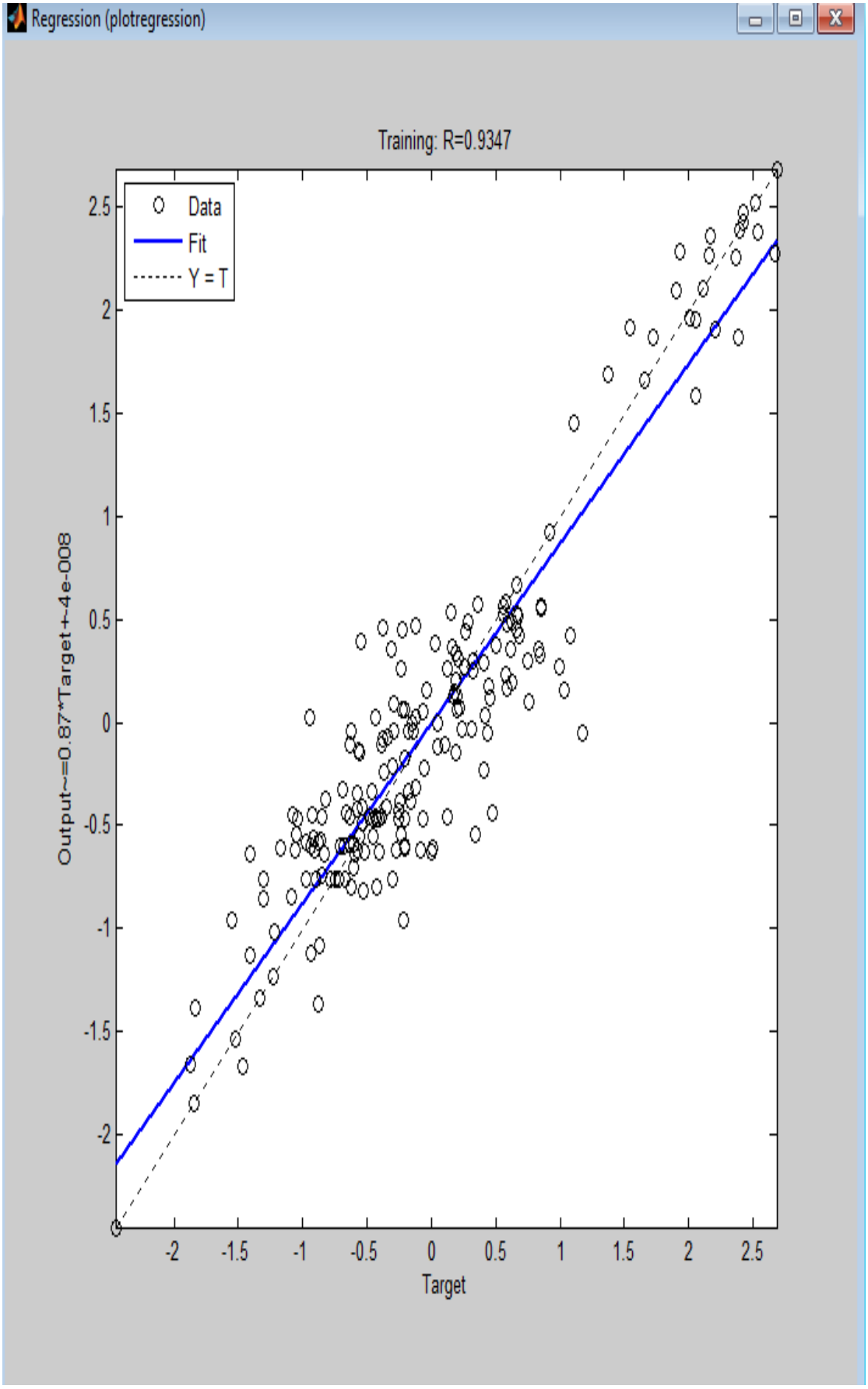
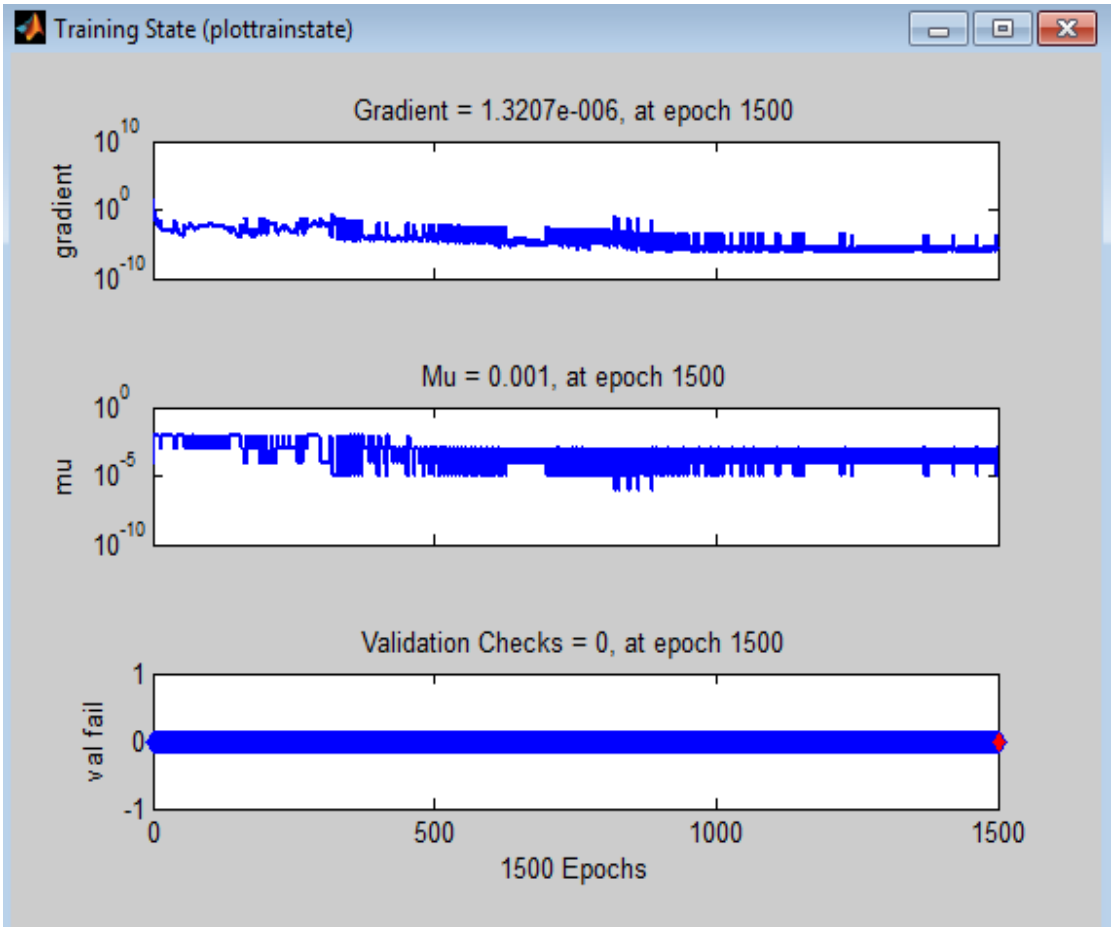
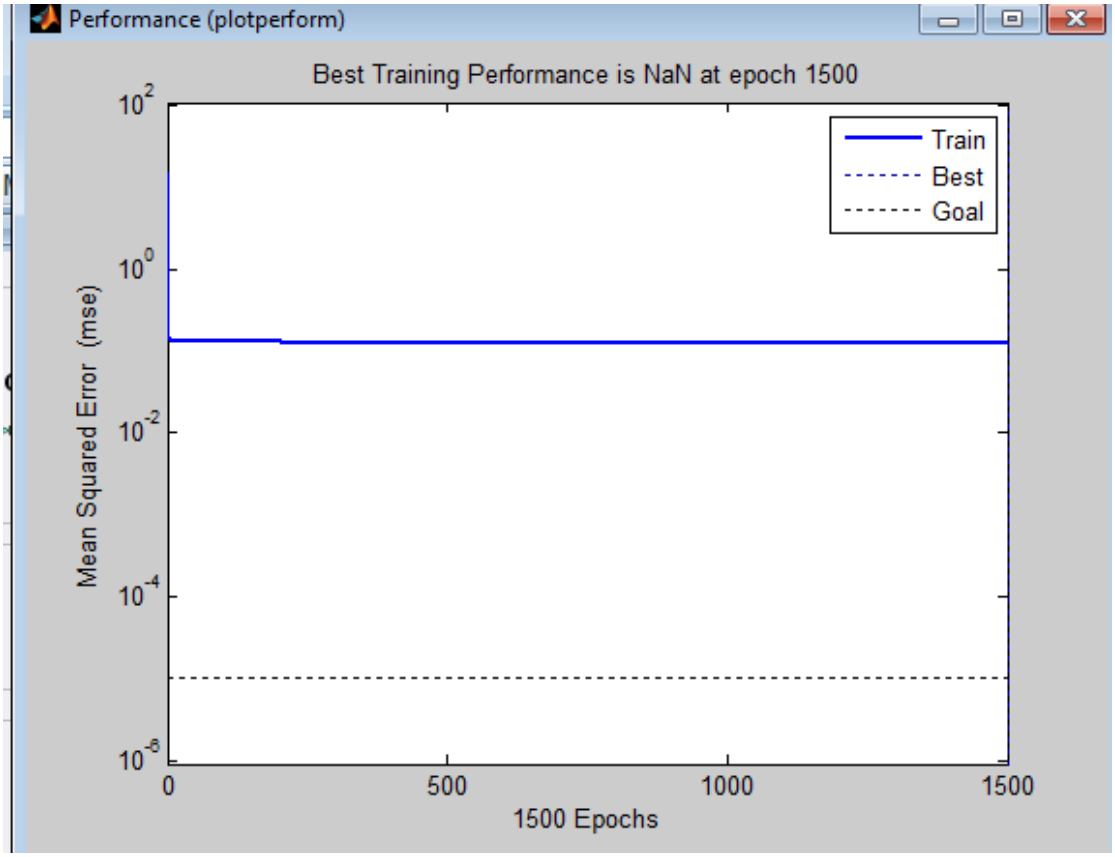
ملاحق الفصل الثالث

شكل (C-333) يمثل تقييم الشبكة العصبية الاصطناعية للتجربة رقم 1

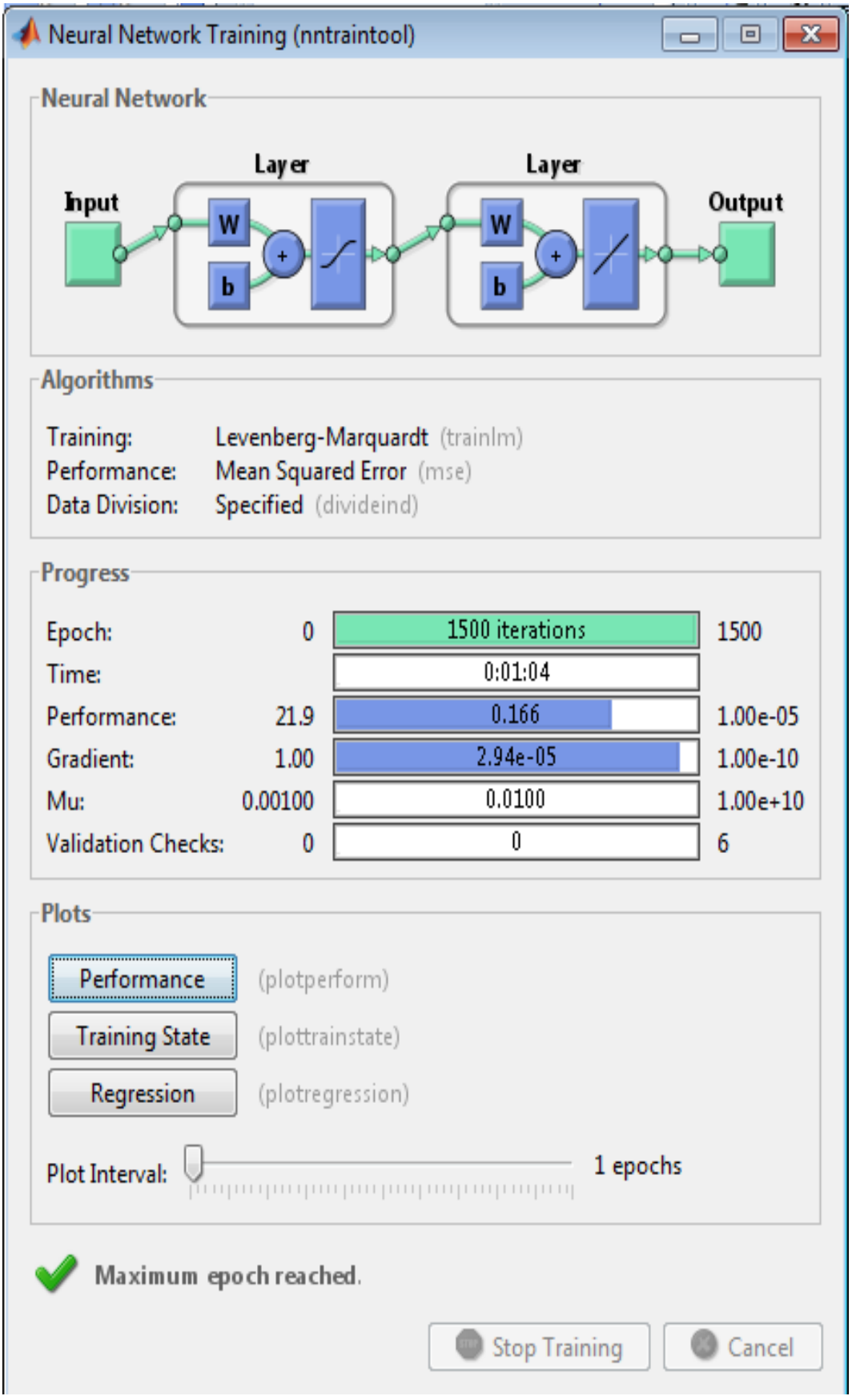

Neural Network Training (nntraintool)
Neural Network
Input
Layer
W
b
Layer
W
b
Output
Algorithms
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Data Division: Specified (divideind)
Progress
Epoch: 0 1500 iterations 1500
Time: 0:00:36
Performance: 14.8 0.126 1.00e-05
Gradient: 1.00 1.32e-06 1.00e-10
Mu: 0.00100 0.00100 1.00e+10
Validation Checks: 0 0 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs
Opening Regression Plo
Stop Training
Cancel

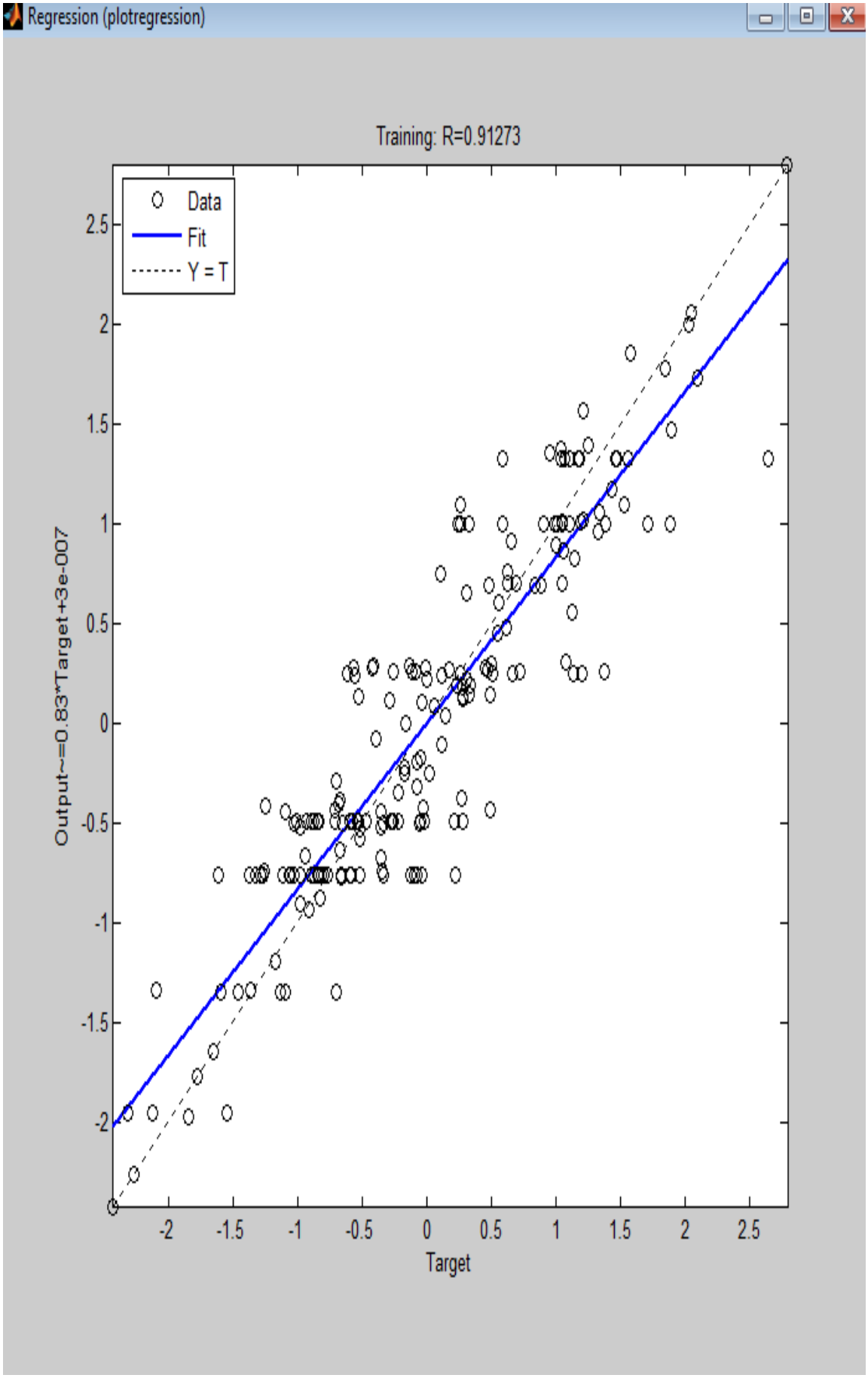

Regression (plotregression)
Training: R=0.9347
Data
Fit
Y = T
Output~=0.87*Target+-4e-008
Target
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

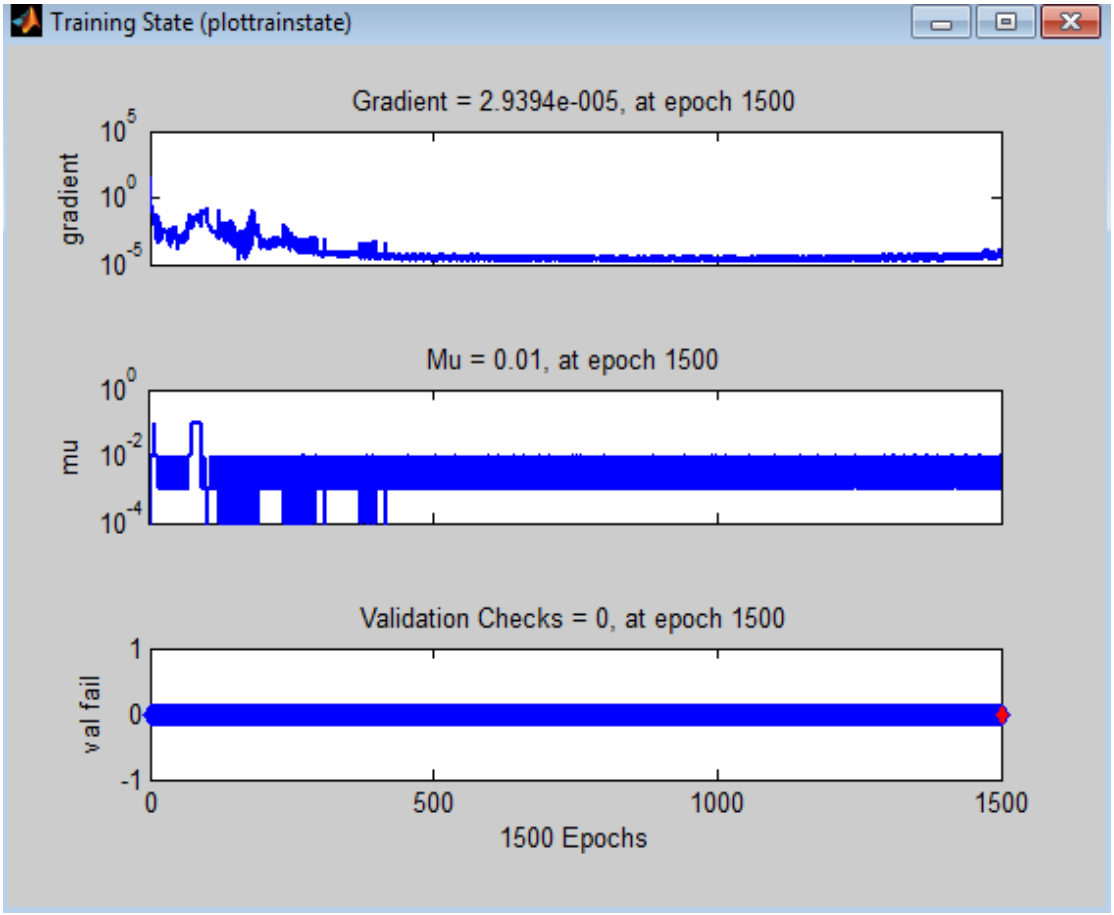

Training State (plottrainstate)
Gradient = 1.3207e-006, at epoch 1500
gradient
10^10
10^0
10^-10
Mu = 0.001, at epoch 1500
mu
10^0
10^-5
10^-10
Validation Checks = 0, at epoch 1500
val fail
1
0
-1
0
500
1000
1500
1500 Epochs

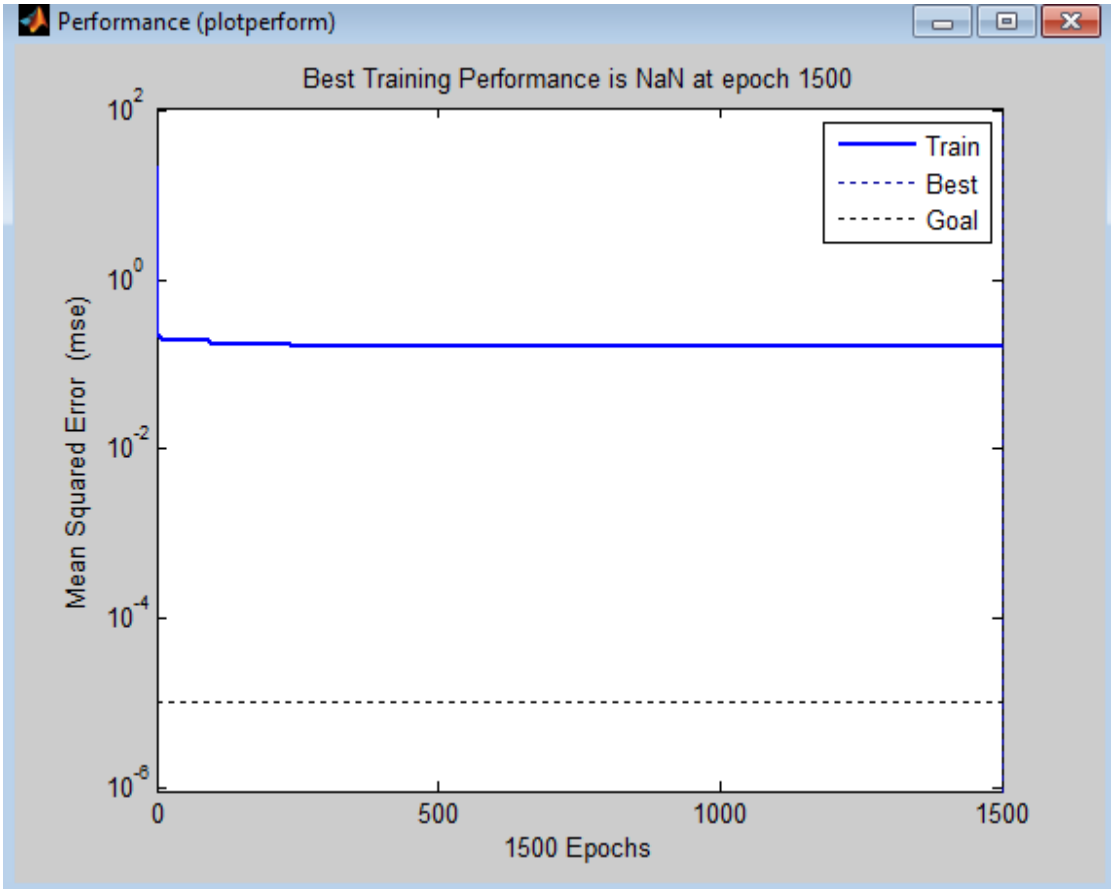


شكل (C34-3) يمثل تقييم الشبكة العصبية الاصطناعية للتجربة رقم 2

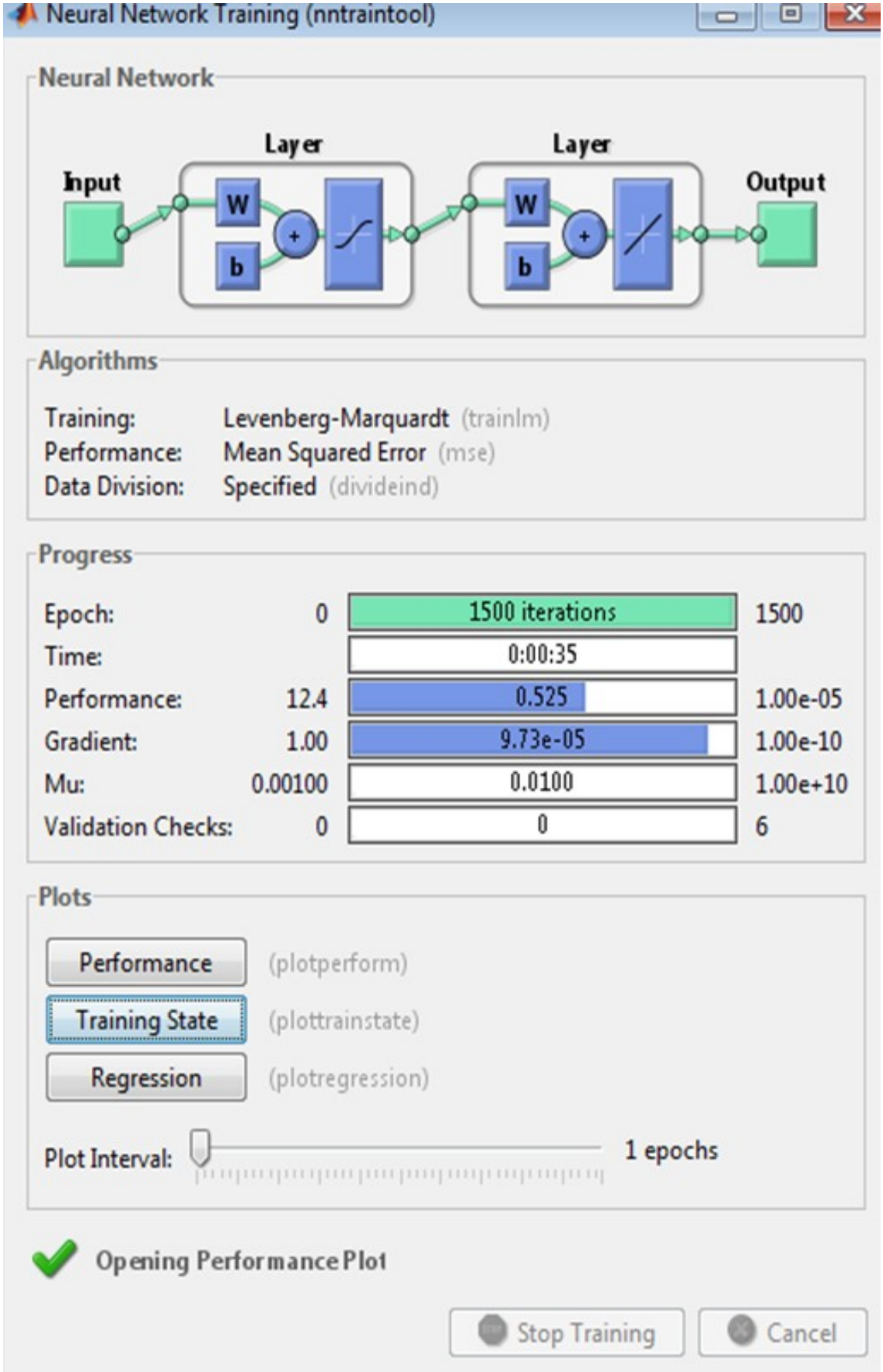

Neural Network Training (nntraintool)
Neural Network
Input
Layer
W
b
Layer
W
b
Output
Algorithms
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Data Division: Specified (divideind)
Progress
Epoch: 0 1500 iterations 1500
Time: 0:01:04
Performance: 21.9 0.166 1.00e-05
Gradient: 1.00 2.94e-05 1.00e-10
Mu: 0.00100 0.0100 1.00e+10
Validation Checks: 0 0 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs
Maximum epoch reached.
Stop Training
Cancel

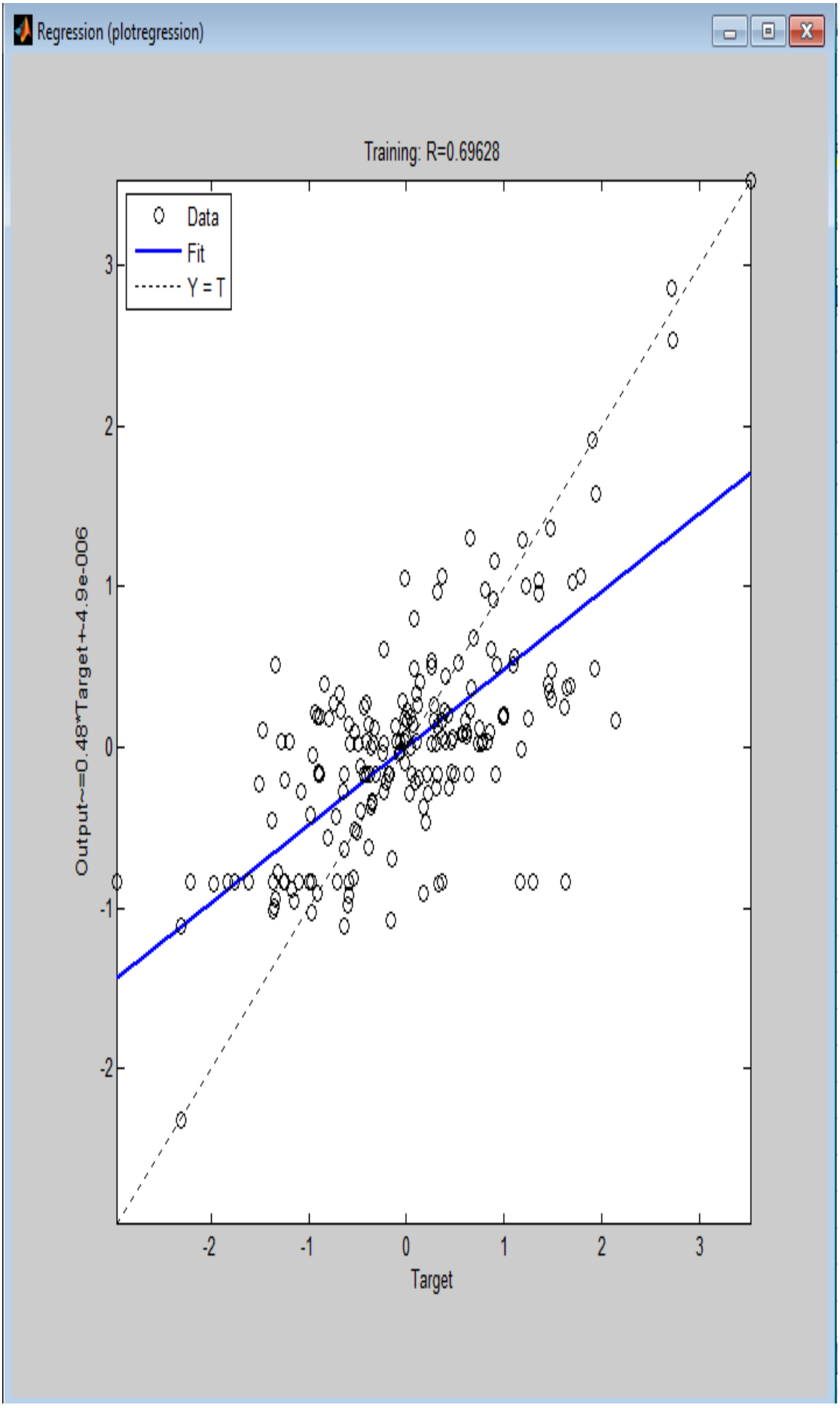

Regression (plotregression)
Training: R=0.91273
Data
Fit
Y = T
Output~=0.83*Target+3e-007
Target
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

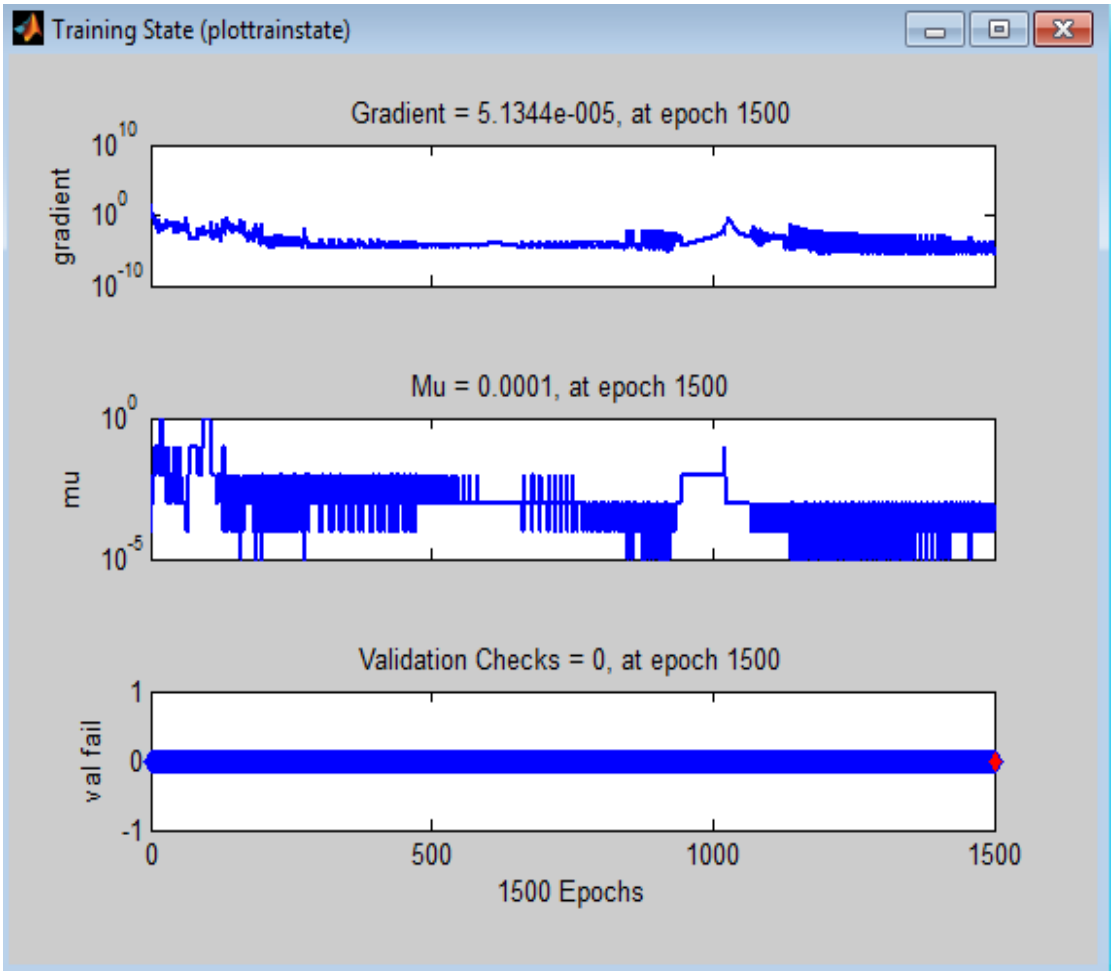

Training State (plottrainstate)
Gradient = 2.9394e-005, at epoch 1500
gradient
10^5
10^0
10^-5
Mu = 0.01, at epoch 1500
mu
10^0
10^-2
10^-4
Validation Checks = 0, at epoch 1500
val fail
1
0
-1
0
500
1000
1500
1500 Epochs

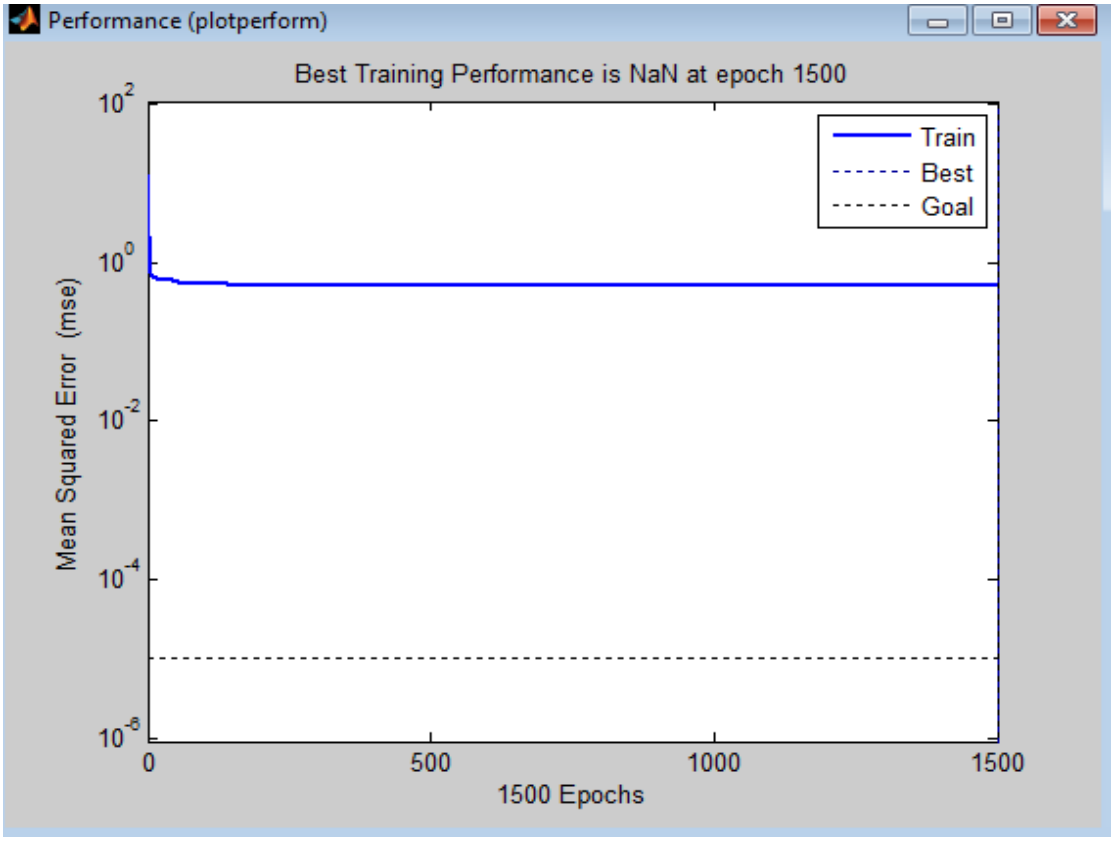


شكل (C35-3) يمثل تقييم الشبكة العصبية الاصطناعية للتجربة رقم 3

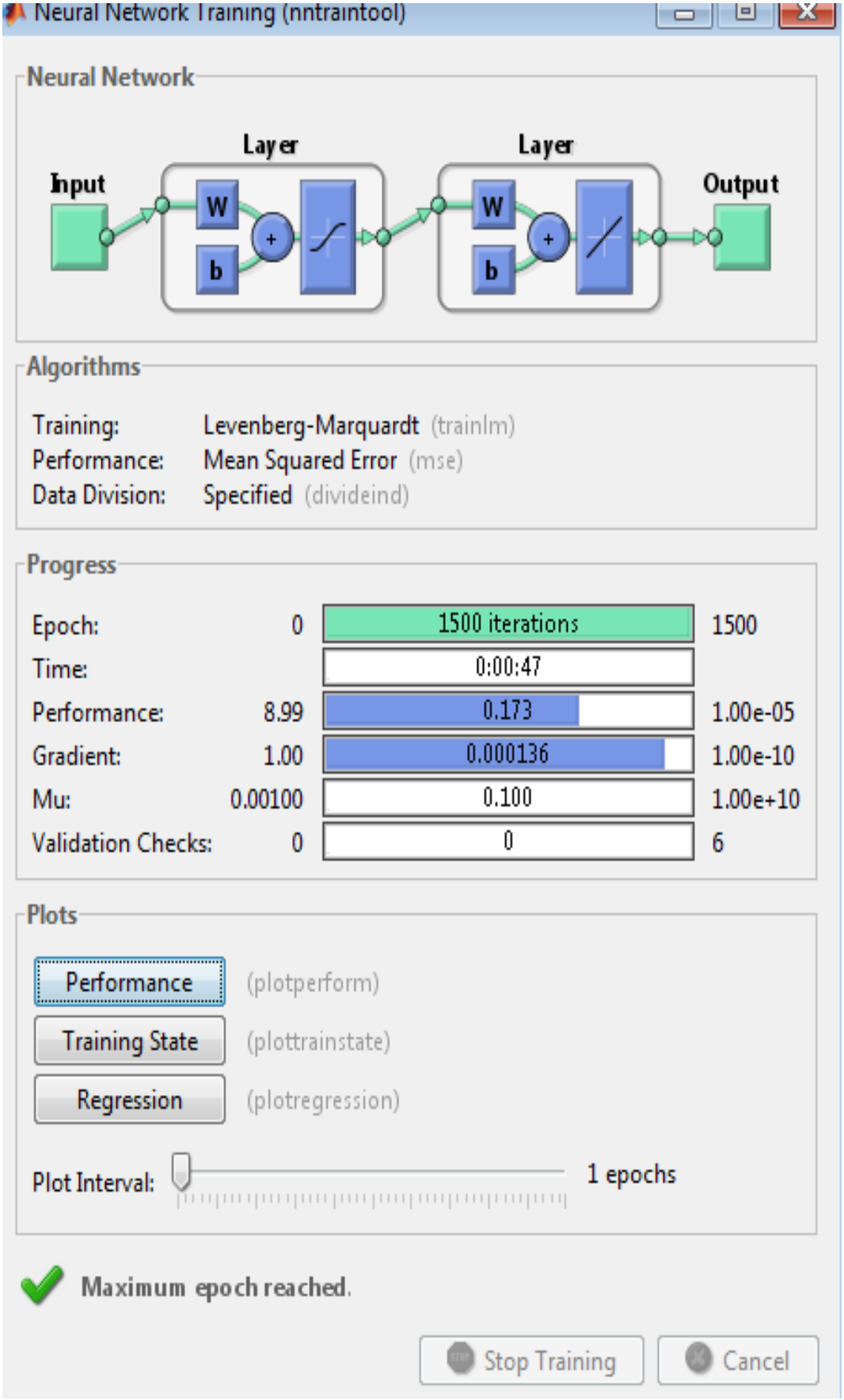

Neural Network Training (nntraintool)
Neural Network
Input
Layer
W
b
Layer
W
b
Output
Algorithms
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Data Division: Specified (divideind)
Progress
Epoch: 0 1500 iterations 1500
Time: 0:00:35
Performance: 12.4 0.525 1.00e-05
Gradient: 1.00 9.73e-05 1.00e-10
Mu: 0.00100 0.0100 1.00e+10
Validation Checks: 0 0 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs
Opening Performance Plot
Stop Training
Cancel

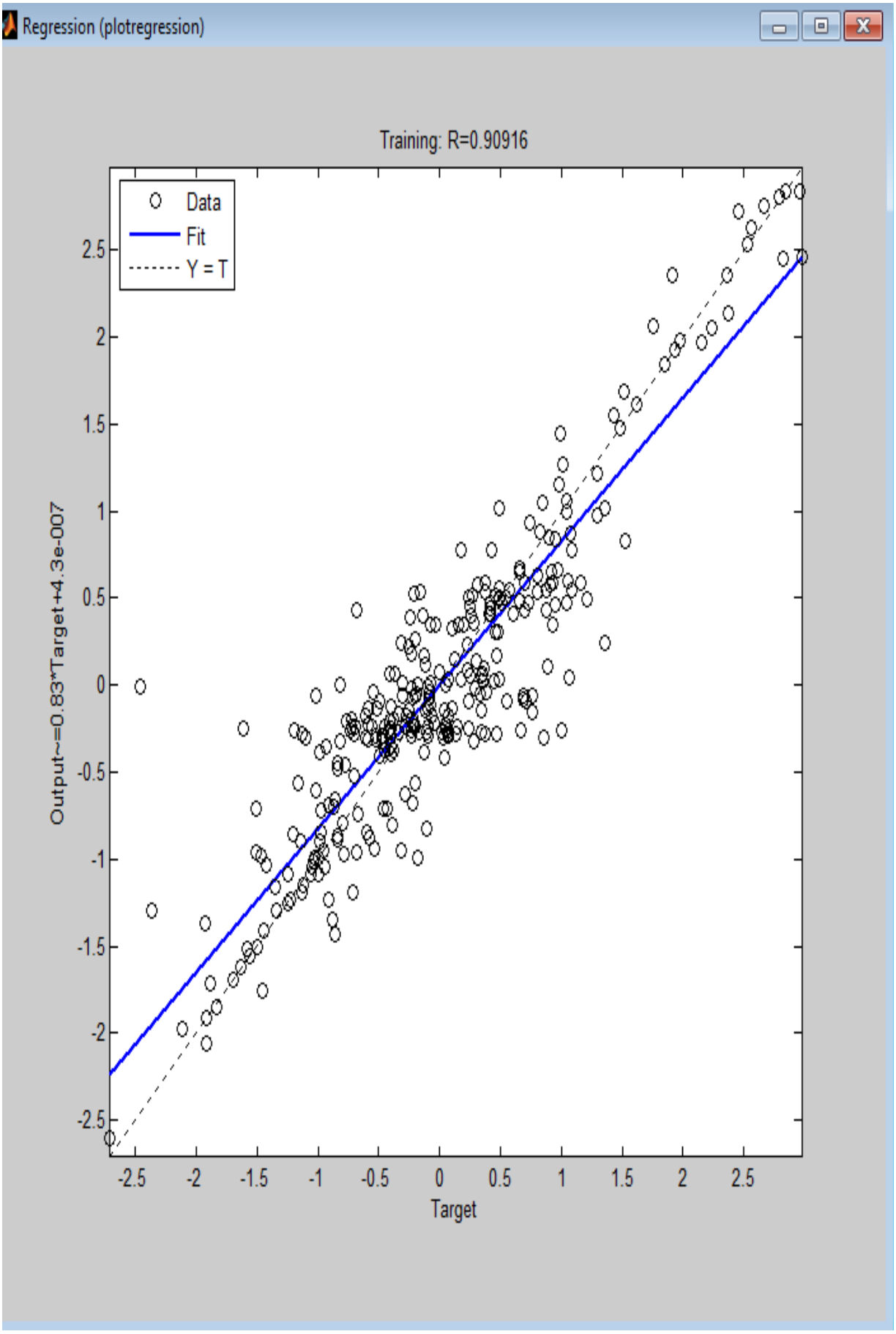

Regression (plotregression)
Training: R=0.69628
Data
Fit
Y = T
Output~=0.48*Target+-4.9e-006
Target
-2
-1
0
1
2
3

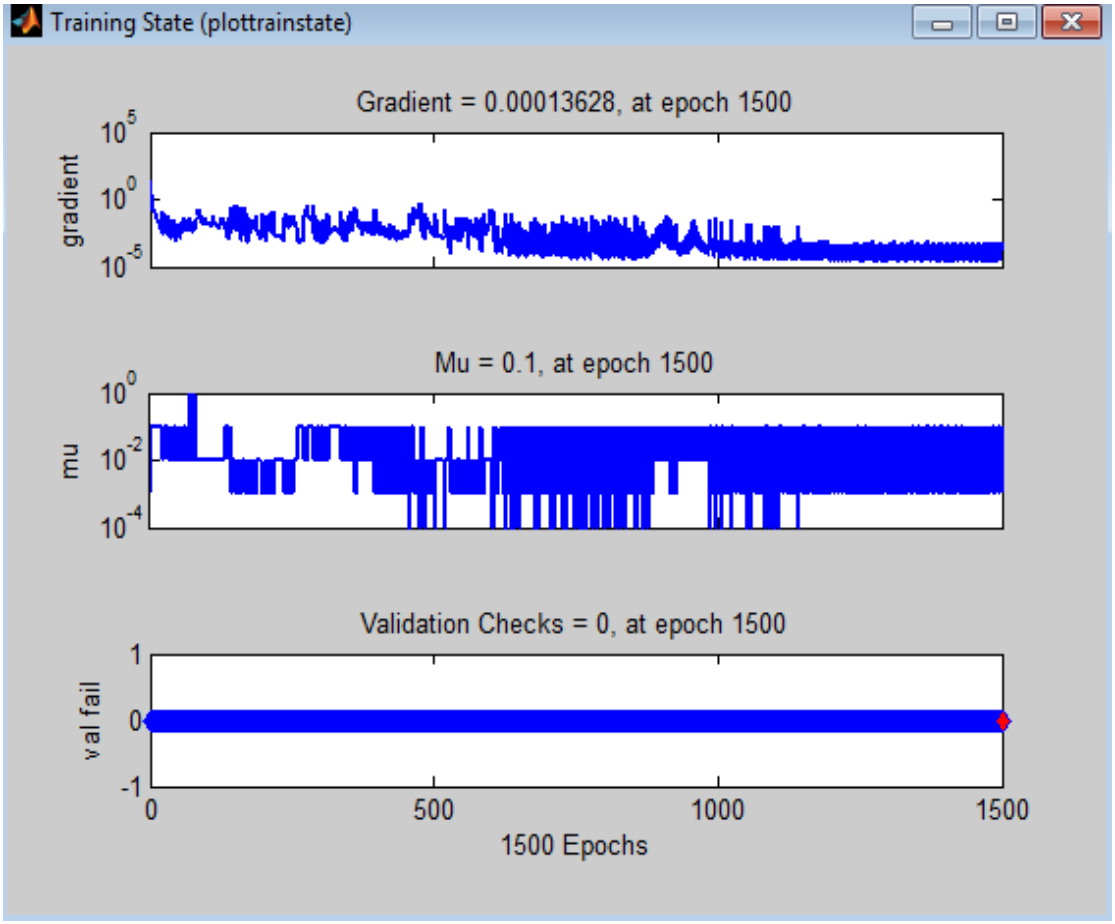

Training State (plottrainstate)
Gradient = 5.1344e-005, at epoch 1500
gradient
10^10
10^0
10^-10
Mu = 0.0001, at epoch 1500
mu
10^0
10^-5
Validation Checks = 0, at epoch 1500
val fail
1
0
-1
0
500
1000
1500
1500 Epochs

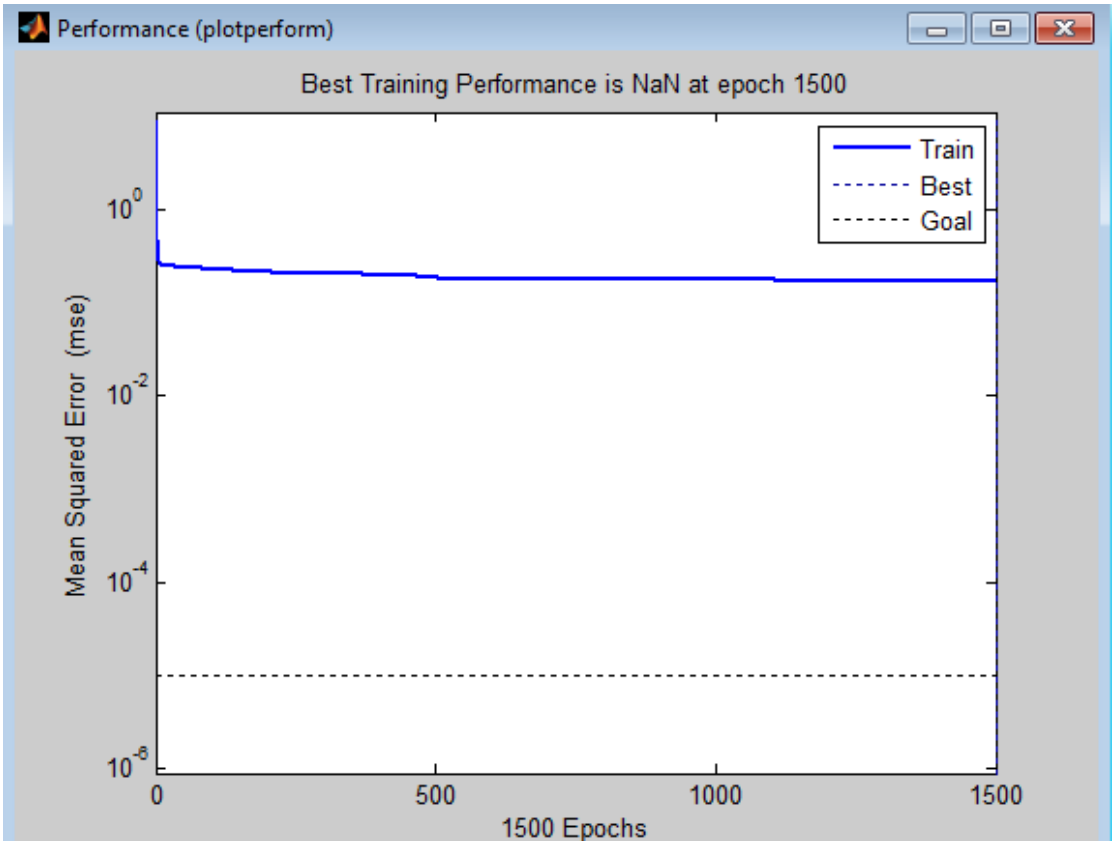


شكل (C36-3) يمثل تقييم الشبكة العصبية الاصطناعية للتجربة رقم 4

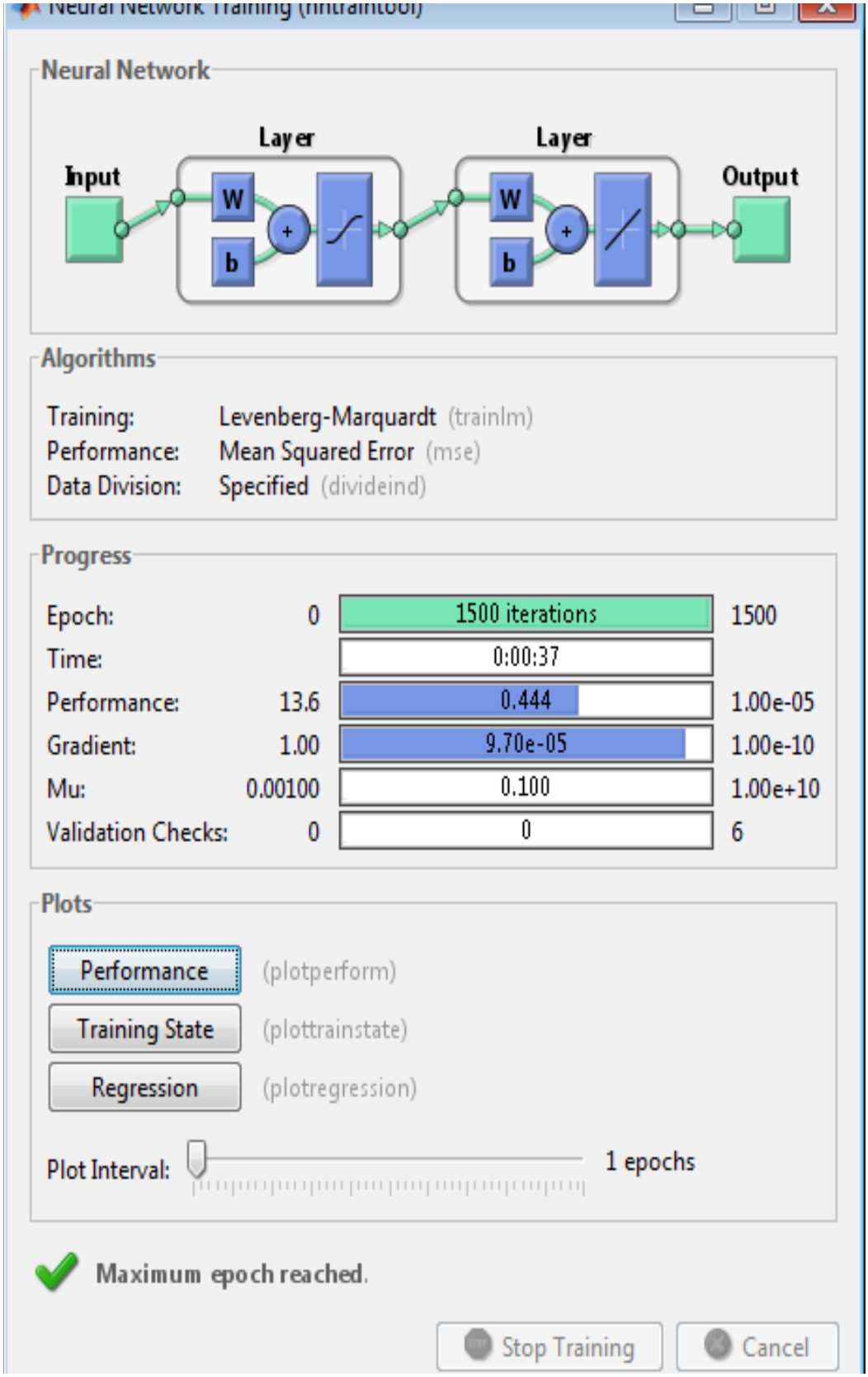

Neural Network Training (nntraintool)
Neural Network
Input
Layer
W
b
Layer
W
b
Output
Algorithms
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Data Division: Specified (divideind)
Progress
Epoch: 0 1500 iterations 1500
Time: 0:00:47
Performance: 8.99 0.173 1.00e-05
Gradient: 1.00 0.000136 1.00e-10
Mu: 0.00100 0.100 1.00e+10
Validation Checks: 0 0 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs
Maximum epoch reached.
Stop Training
Cancel

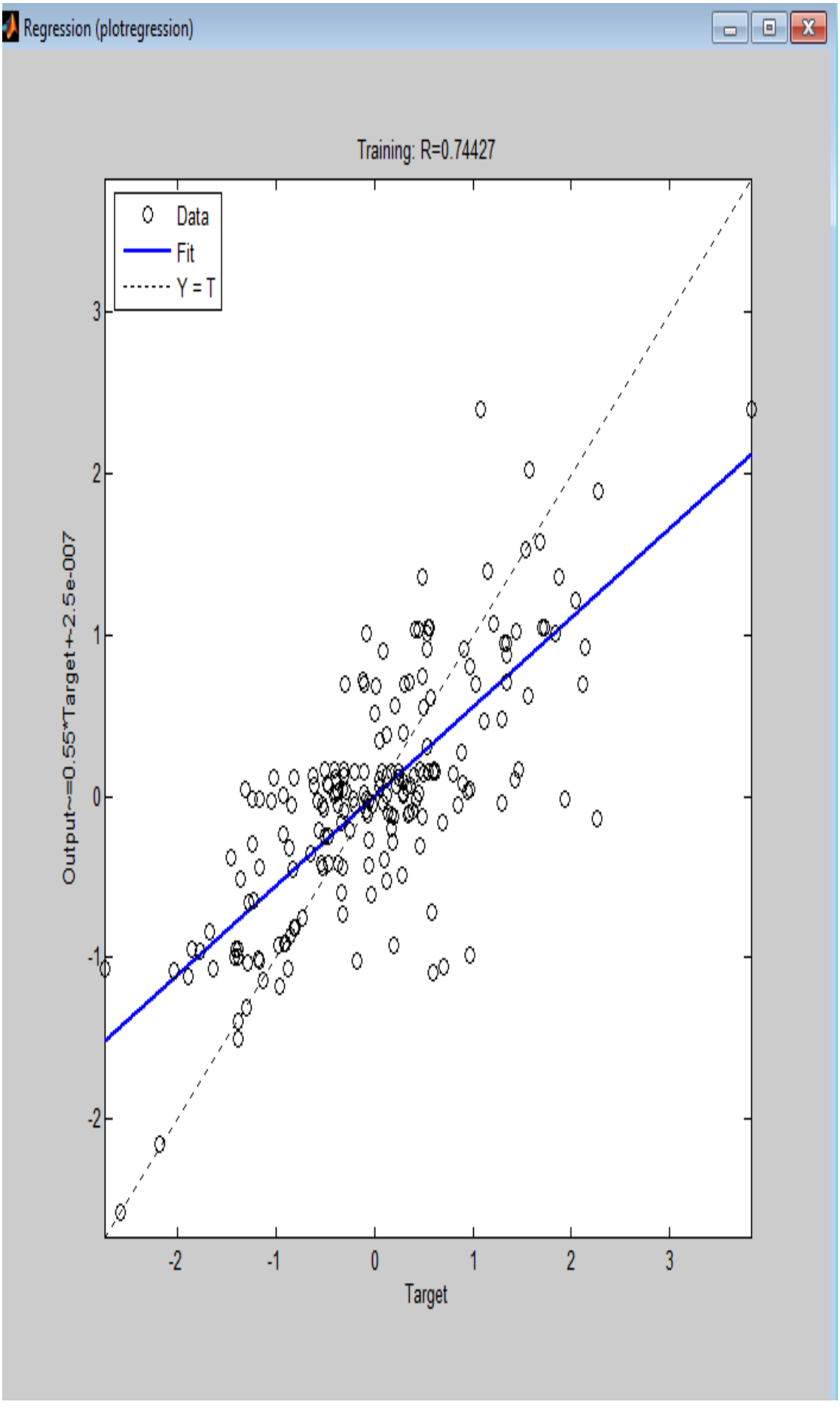

Regression (plotregression)
Training: R=0.90916
Data
Fit
Y = T
Output~=0.83*Target+4.3e-007
Target
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5

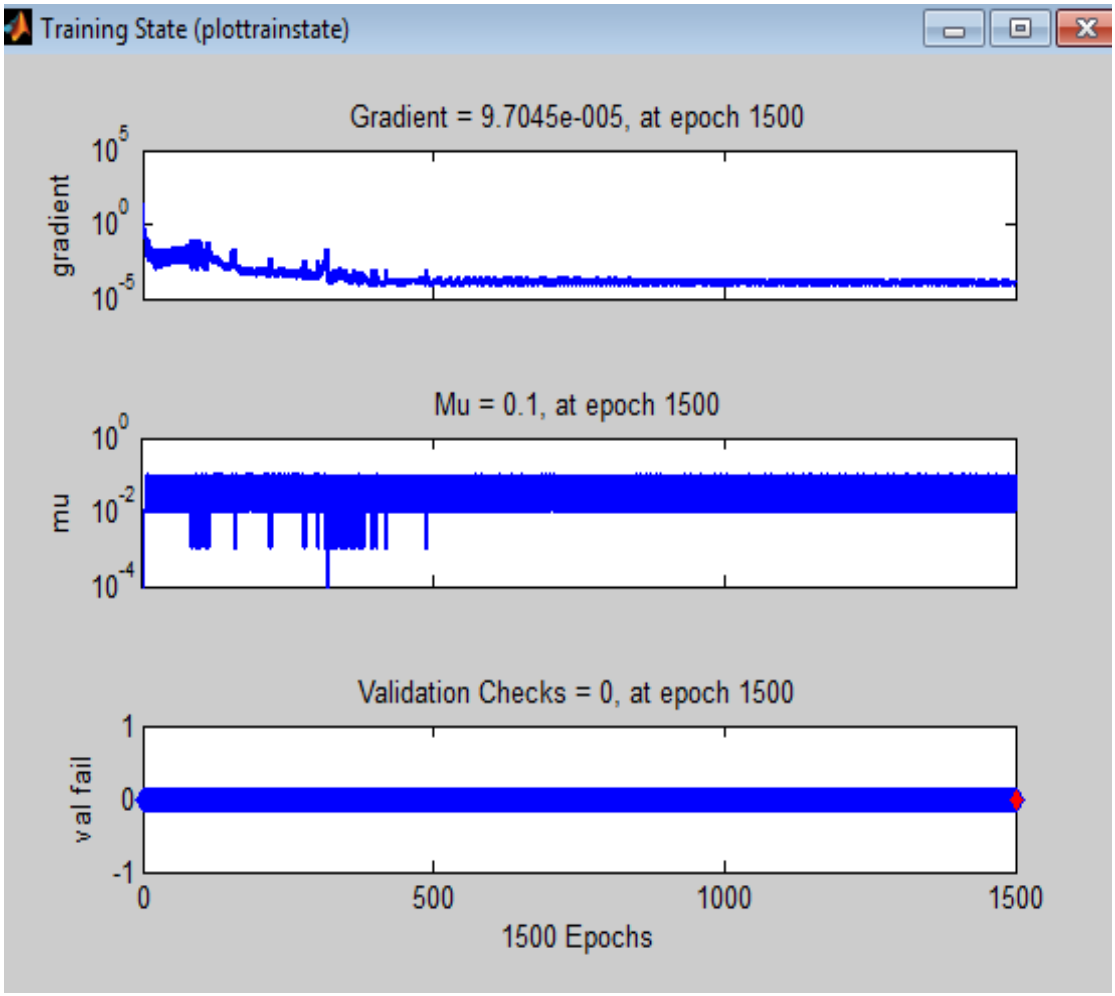

Training State (plottrainstate)
Gradient = 0.00013628, at epoch 1500
gradient
10^5
10^0
10^-5
Mu = 0.1, at epoch 1500
mu
10^0
10^-2
10^-4
Validation Checks = 0, at epoch 1500
val fail
1
0
-1
0
500
1000
1500
1500 Epochs

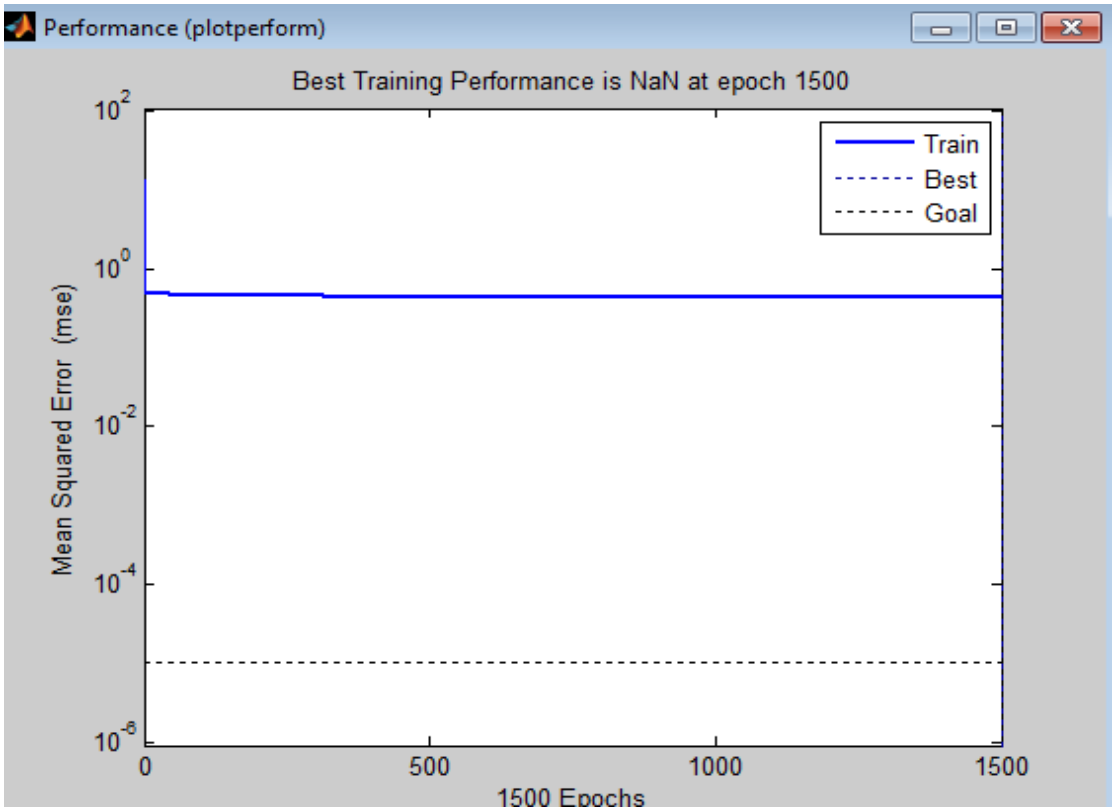


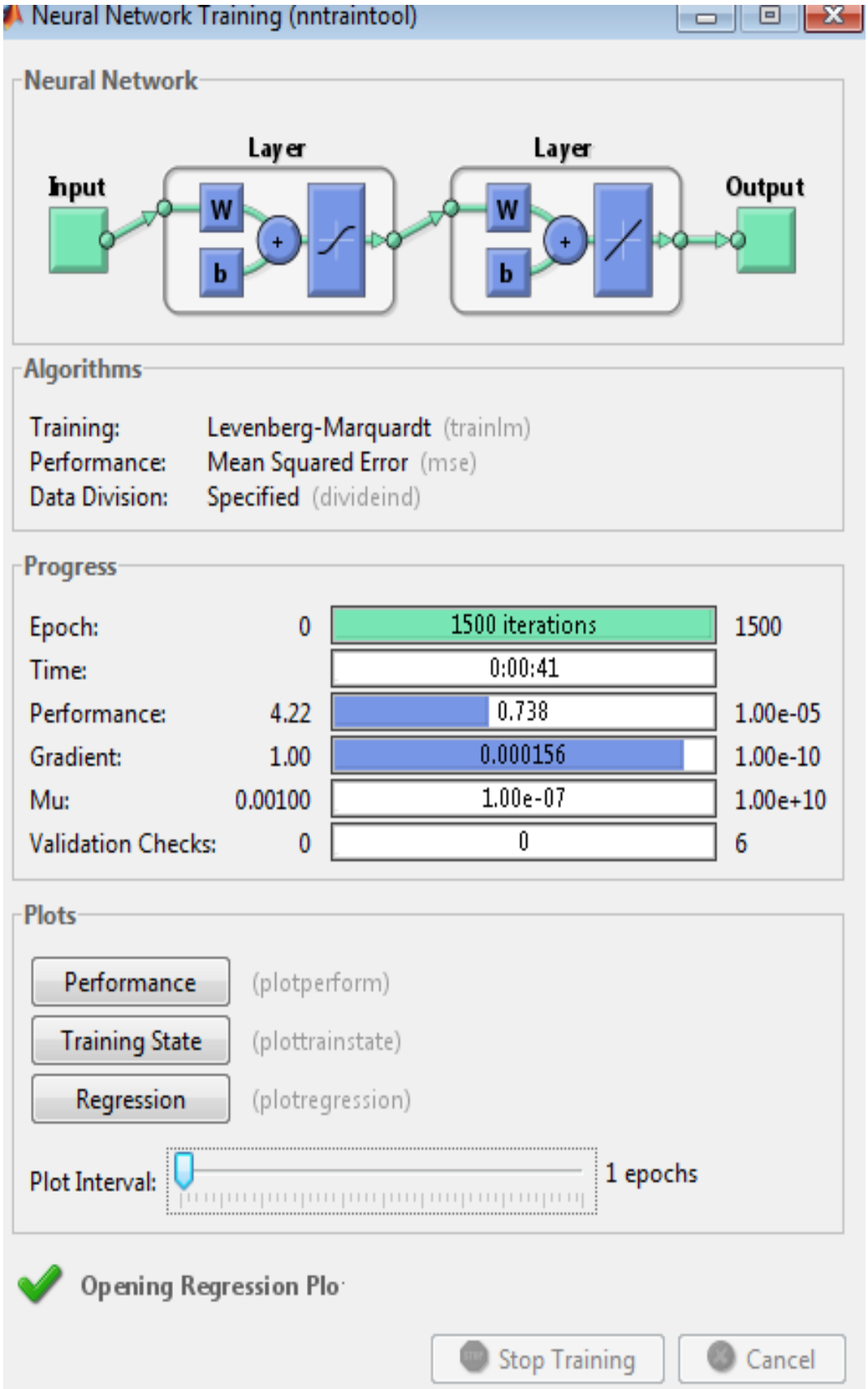
شكل (C37-3) يمثل تقييم الشبكة العصبية الاصطناعية للتجربة رقم 5


Neural Network Training (nntraintool)
Neural Network
Input
Layer
W
b
Layer
W
b
Output
Algorithms
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Data Division: Specified (divideind)
Progress
Epoch: 0 1500 iterations 1500
Time: 0:00:37
Performance: 13.6 0.444 1.00e-05
Gradient: 1.00 9.70e-05 1.00e-10
Mu: 0.00100 0.100 1.00e+10
Validation Checks: 0 0 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs
Maximum epoch reached.
Stop Training
Cancel

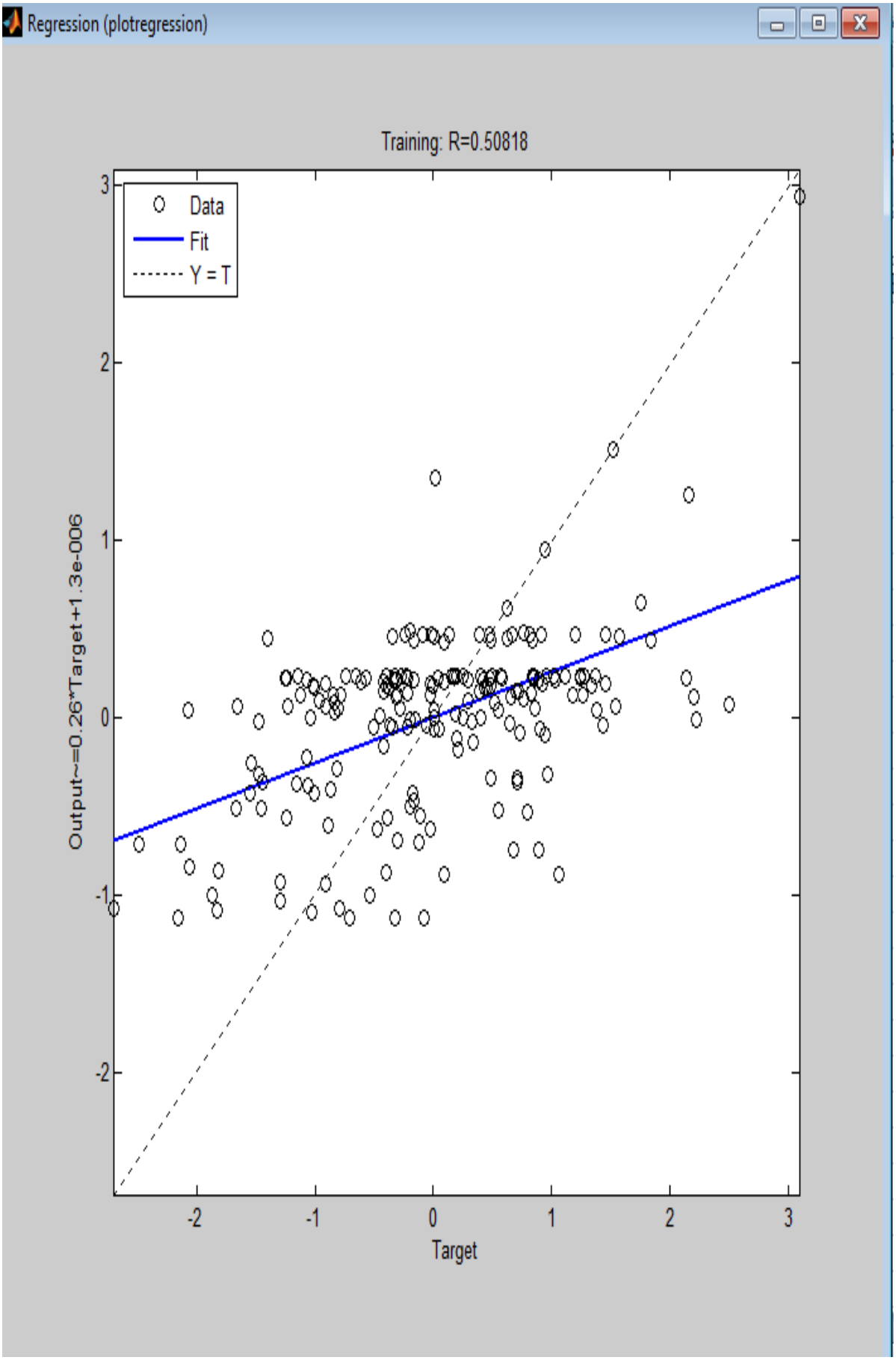

Regression (plotregression)
Training: R=0.74427
Data
Fit
Y = T
Output~=0.55*Target+-2.5e-007
Target
-2
-1
0
1
2
3

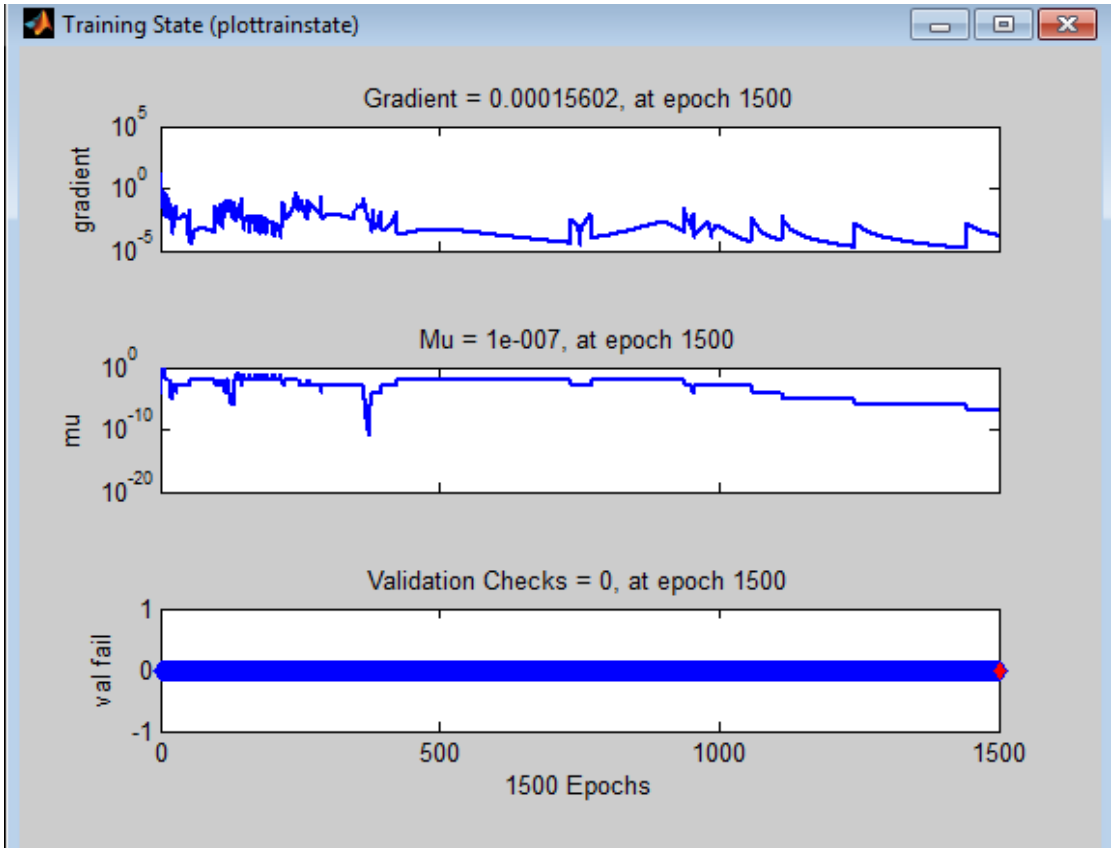

Training State (plottrainstate)
Gradient = 9.7045e-005, at epoch 1500
gradient
10^5
10^0
10^-5
Mu = 0.1, at epoch 1500
mu
10^0
10^-2
10^-4
Validation Checks = 0, at epoch 1500
val fail
1
0
-1
0
500
1000
1500
1500 Epochs

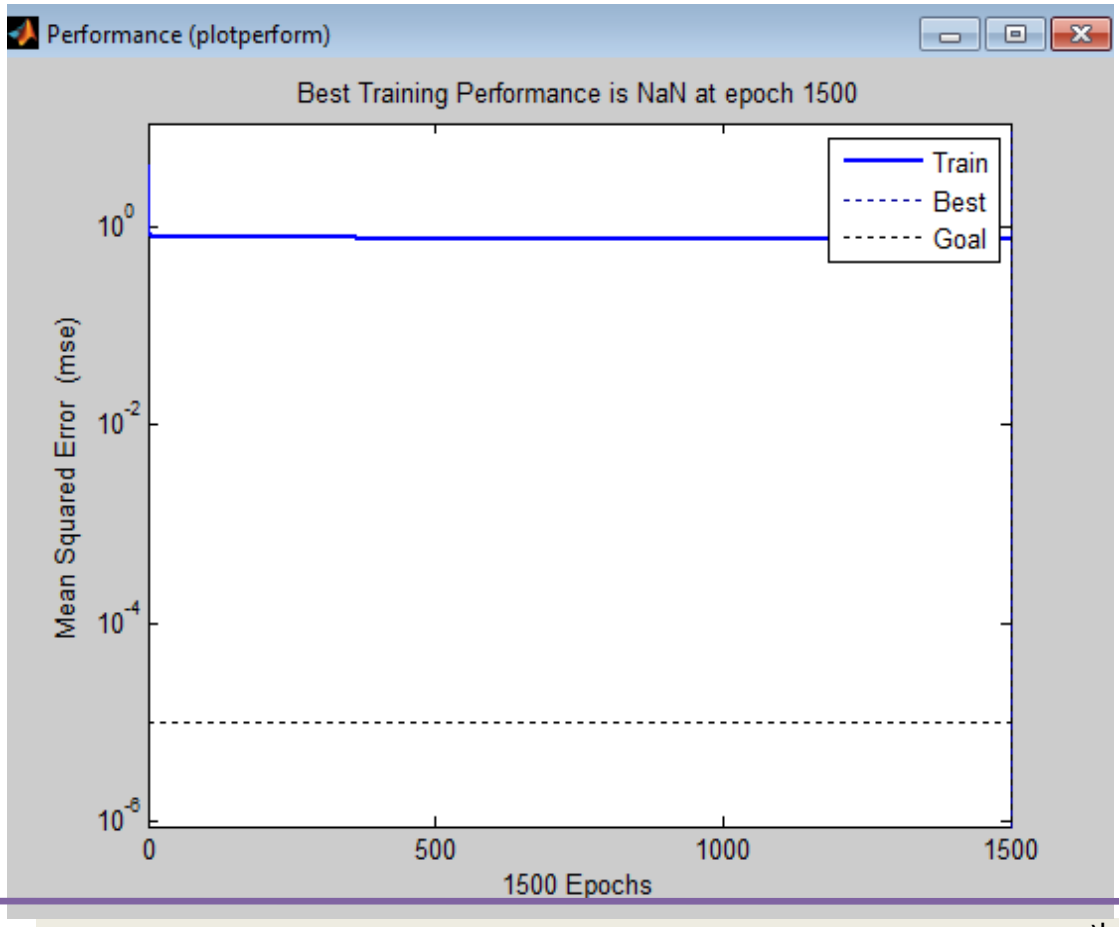

Performance (plotperform)
Best Training Performance is NaN at epoch 1500
Train
Best
Goal
Mean Squared Error (mse)
10^2
10^0
10^-2
10^-4
10^-6
0
500
1000
1500
1500 Epochs

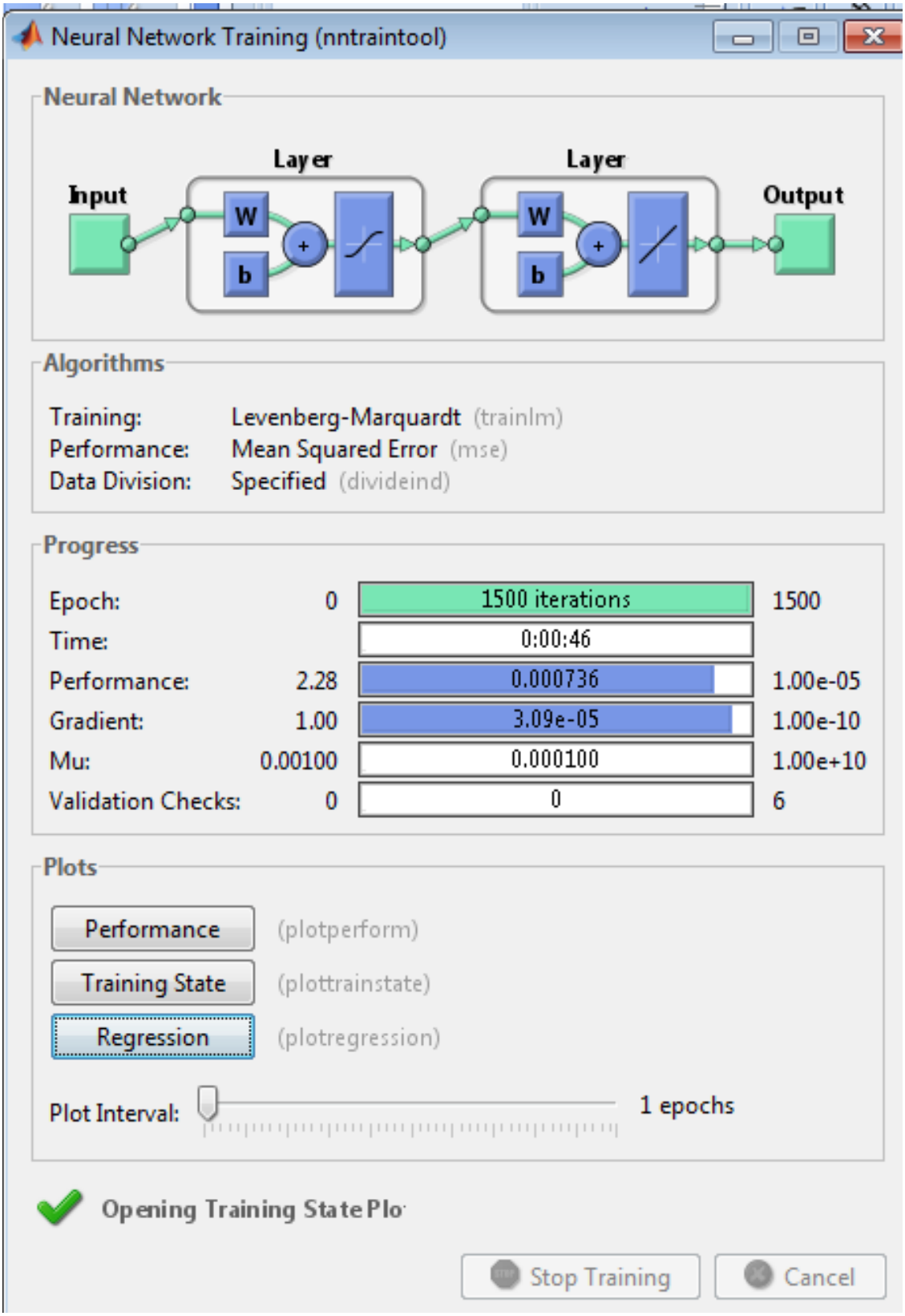


شكل (3-38C) يمثل تقييم الشبكة العصبية الاصطناعية للتجربة رقم 6

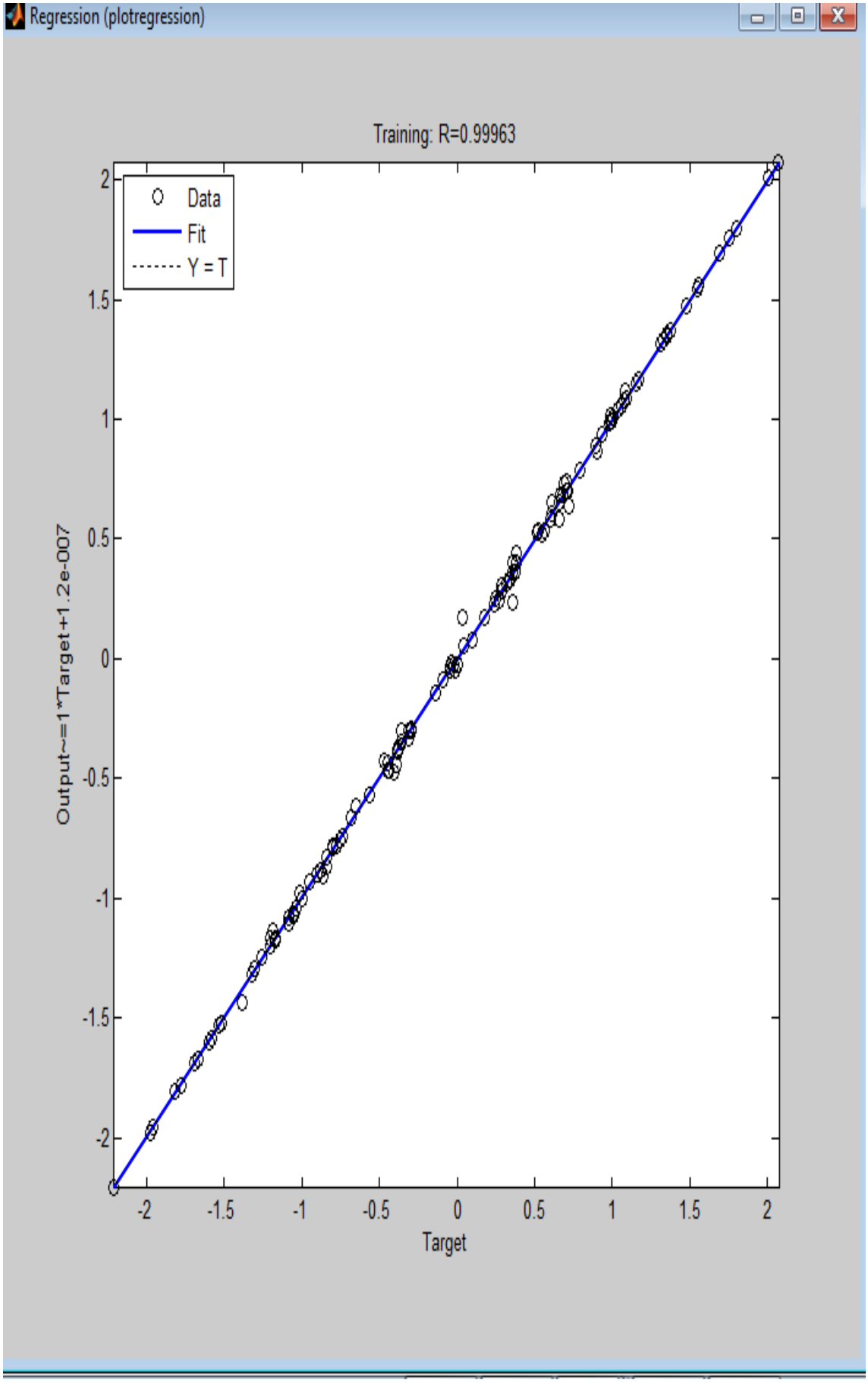

Regression (plotregression)
Training: R=0.50818
Data
Fit
Y = T
Output~=0.26*Target+1.3e-006
Target
3
2
1
0
-1
-2
-2
-1
0
1
2
3

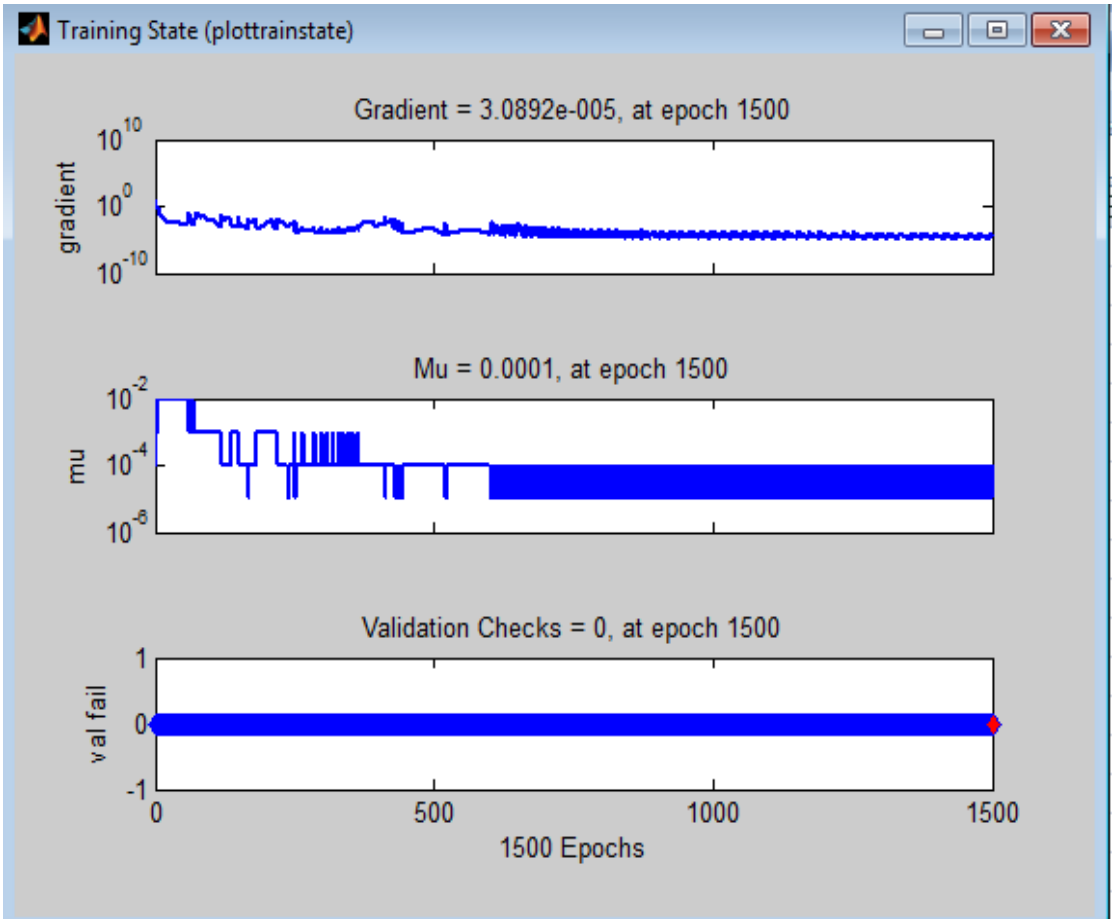

Training State (plottrainstate)
Gradient = 0.00015602, at epoch 1500
gradient
Mu = 1e-007, at epoch 1500
mu
Validation Checks = 0, at epoch 1500
val fail
0
500
1000
1500
1500 Epochs

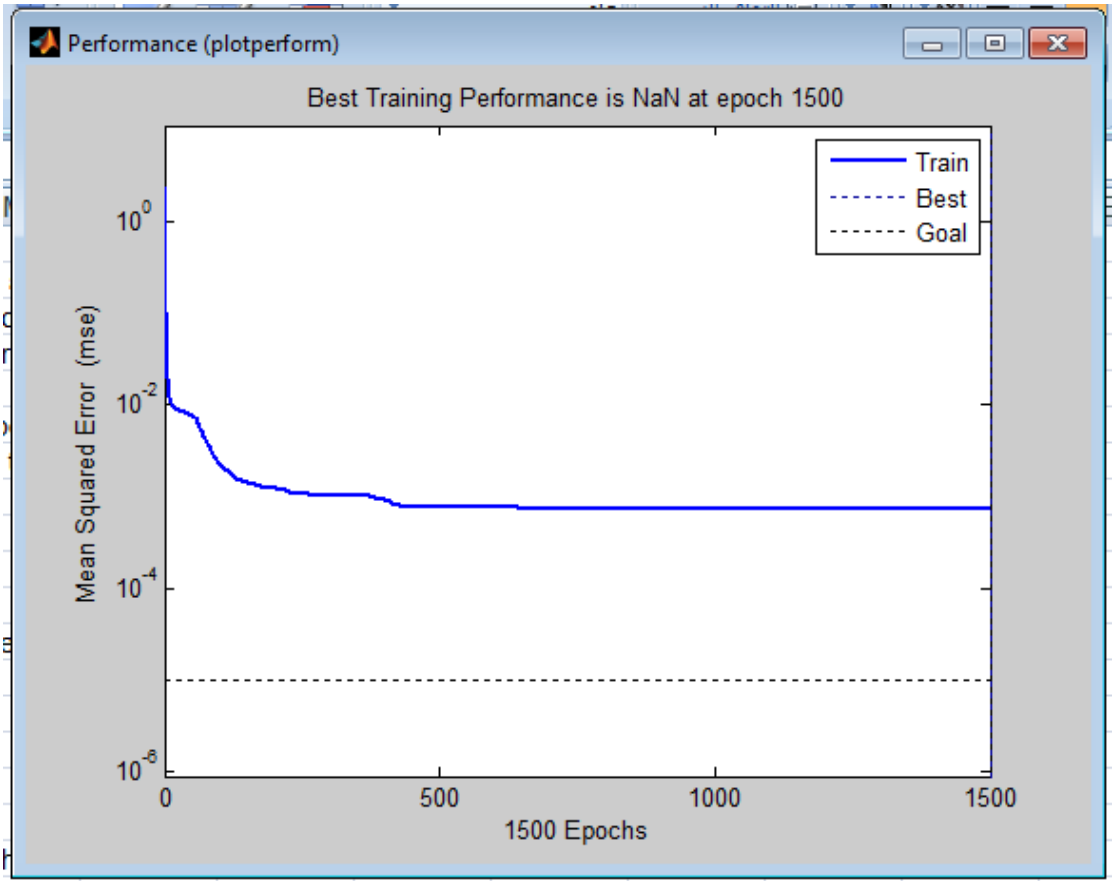

Performance (plotperform)
Best Training Performance is NaN at epoch 1500
Train
Best
Goal
Mean Squared Error (mse)
0
500
1000
1500
1500 Epochs

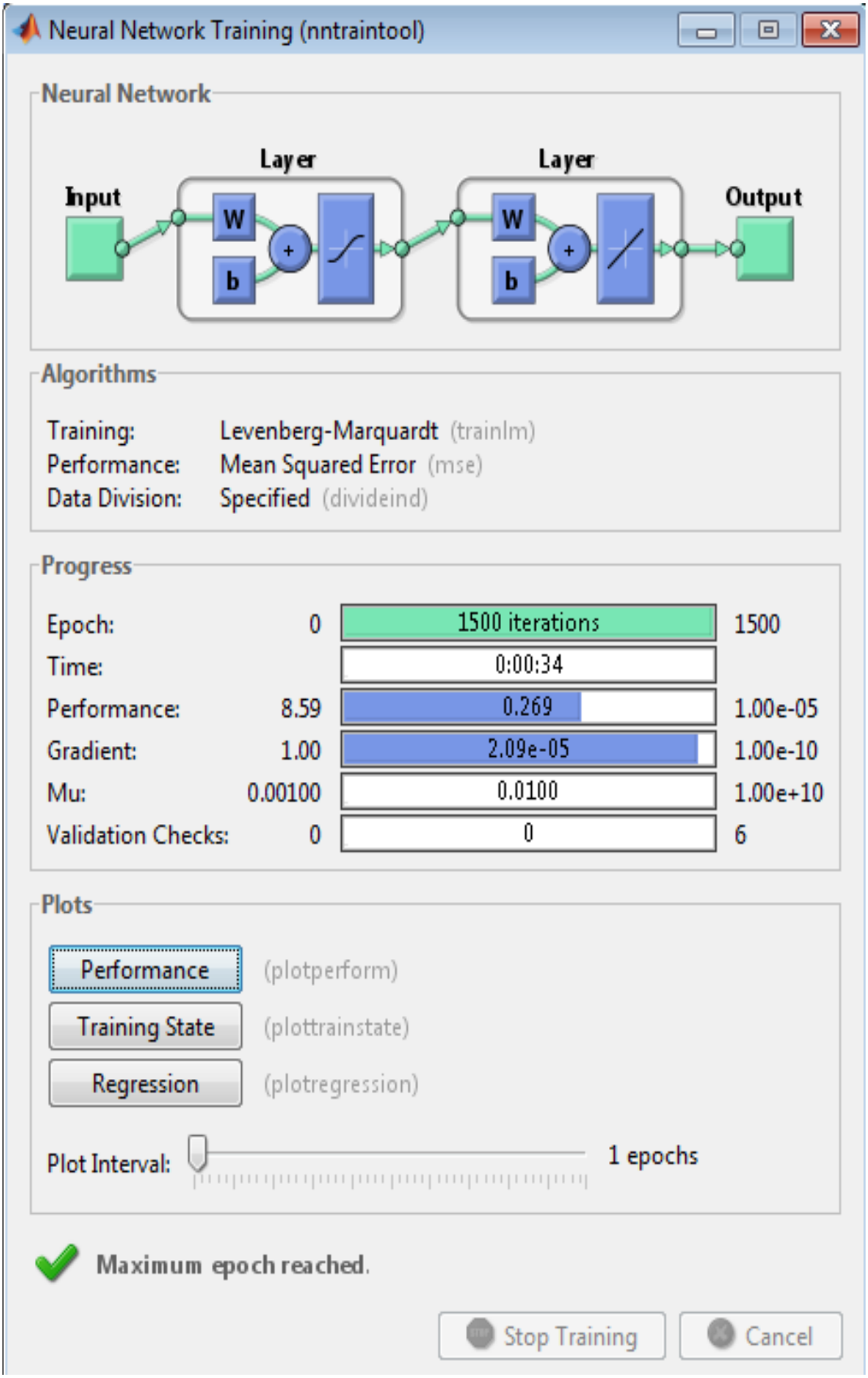


شكل (C39-3) يمثل تقييم الشبكة العصبية الاصطناعية للتجربة رقم 7

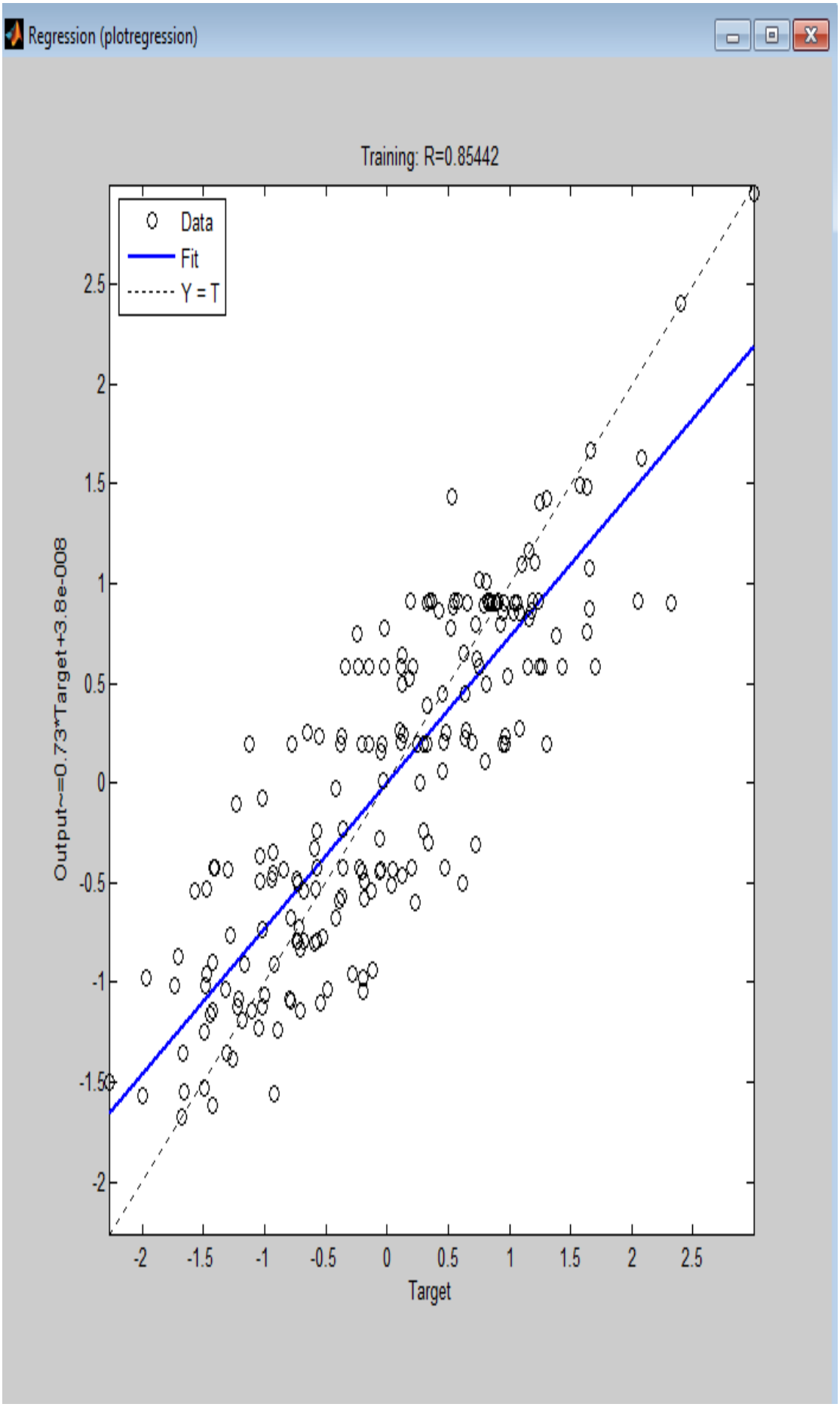

Regression (plotregression)
Training: R=0.99963
Data
Fit
Y = T
Output~=1*Target+1.2e-007
Target
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

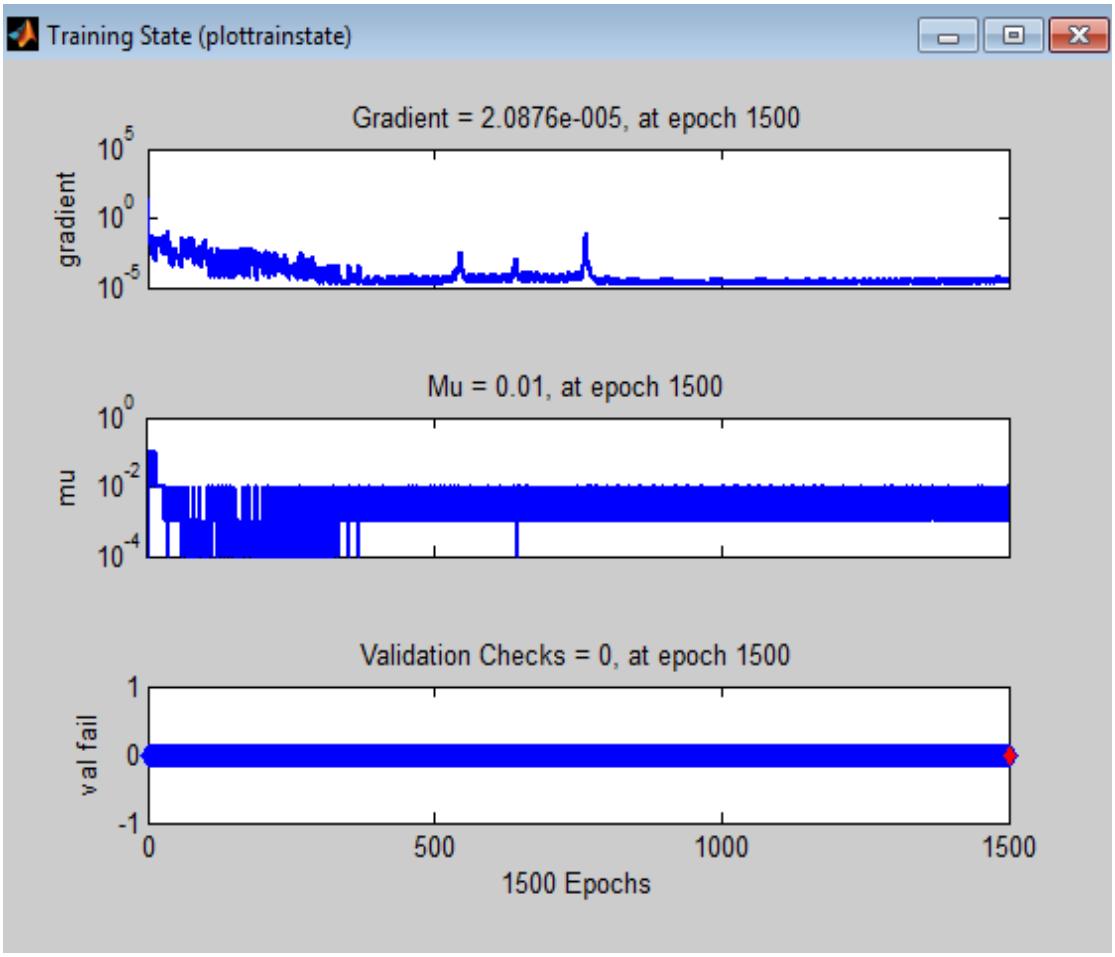

Training State (plottrainstate)
Gradient = 3.0892e-005, at epoch 1500
gradient
10^10
10^0
10^-10
Mu = 0.0001, at epoch 1500
mu
10^-2
10^-4
10^-6
Validation Checks = 0, at epoch 1500
val fail
1
0
-1
0
500
1000
1500
1500 Epochs

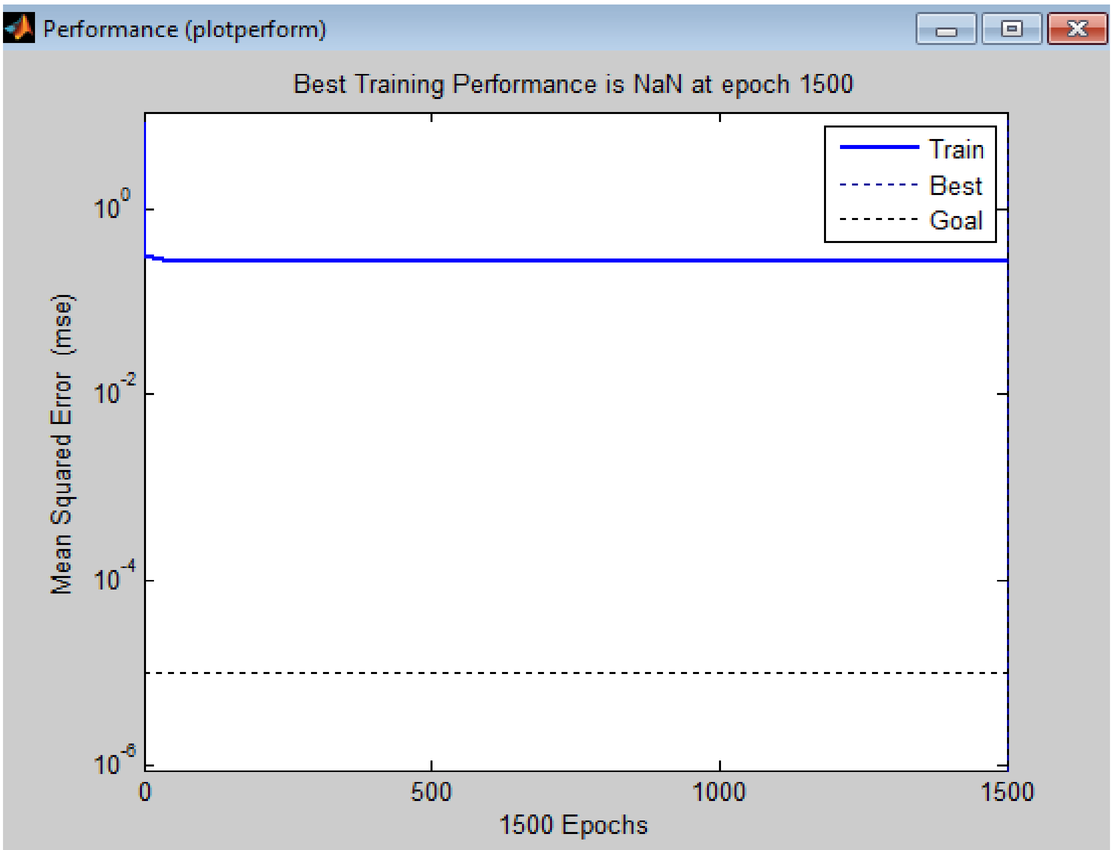

Performance (plotperform)
Best Training Performance is NaN at epoch 1500
Train
Best
Goal
Mean Squared Error (mse)
10^0
10^-2
10^-4
10^-6
0
500
1000
1500
1500 Epochs



شكل (3-40C) يمثل تقييم الشبكة العصبية الاصطناعية للتجربة رقم 8


Regression (plotregression)
Training: R=0.85442
Data
Fit
Y = T
Output~=0.73*Target+3.8e-008
Target
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5


Training State (plottrainstate)
Gradient = 2.0876e-005, at epoch 1500
gradient
Mu = 0.01, at epoch 1500
mu
Validation Checks = 0, at epoch 1500
val fail
0
500
1000
1500
1500 Epochs



مراحل التدريب في الشبكة العصبية الاصطناعية للتجربة رقم 1

```
= net

:Neural Network object


:architecture

numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1  (read-only
(numInputDelays: 0  (read-only
(numLayerDelays: 0  (read-only

:subobject structures
```

```
                inputs: {1x1 cell} of inputs
                layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
    inputWeights: {2x1 cell} containing 1
                                    input weight
    layerWeights: {2x2 cell} containing 1
                                    layer weight

                                    :functions

                                'adaptFcn: 'trains
                                (divideFcn: (none
                            'gradientFcn: 'gdefaults
                                 'initFcn: 'initlay
                                'performFcn: 'mse
                                        plotFcns:
{{'plotperform','plottrainstate','plotregression
                                'trainFcn: 'trainlm

                                        :parameters

                            adaptParam: .passes
                              (divideParam: (none
                             (gradientParam: (none
                                (initParam: (none
                             (performParam: (none
      trainParam: .show, .showWindow,
                  ,.showCommandLine, .epochs
          time, .goal, .max_fail,.
                                    ,.mem_reduc
       min_grad, .mu, .mu_dec,.
                                          ,.mu_inc
                            mu_max.

                            :weight and bias values

  IW: {2x1 cell} containing 1 input
                                    weight matrix
 LW: {2x2 cell} containing 1 layer
                                    weight matrix
   b: {2x1 cell} containing 2 bias
                                          vectors
```

:other

" :name
(userdata: (user information

= net

:Neural Network object

:architecture

numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1 (read-only
(numInputDelays: 0 (read-only
(numLayerDelays: 0 (read-only

:subobject structures

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1
input weight
layerWeights: {2x2 cell} containing 1
layer weight

:functions

'adaptFcn: 'trains
(divideFcn: (none
'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
plotFcns:

```
{'{'plotperform','plottrainstate','plotregression
                    'trainFcn: 'trainlm

                                :parameters

                    adaptParam: .passes
                      (divideParam: (none
                     (gradientParam: (none
                       (initParam: (none
                     (performParam: (none
    trainParam: .show, .showWindow,
                ,.showCommandLine, .epochs
            time, .goal, .max_fail,.
                                    ,.mem_reduc
     min_grad, .mu, .mu_dec,.
                                          ,.mu_inc
                          mu_max.

                       :weight and bias values

  IW: {2x1 cell} containing 1 input
                                    weight matrix
 LW: {2x2 cell} containing 1 layer
                                    weight matrix
   b: {2x1 cell} containing 2 bias
                                      vectors

                                    :other

                                 '' :name
              (userdata: (user information


                                         = net

                        :Neural Network object

                                    :architecture

                              numInputs: 1
                             numLayers: 2
                          [biasConnect: [1; 1
                          [inputConnect: [1; 0
```

```
                    [layerConnect: [0 0; 1 0
                        [outputConnect: [0 1

                (numOutputs: 1  (read-only
              (numInputDelays: 0  (read-only
             (numLayerDelays: 0  (read-only

                            :subobject structures

                inputs: {1x1 cell} of inputs
                layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
      inputWeights: {2x1 cell} containing 1
                                    input weight
      layerWeights: {2x2 cell} containing 1
                                    layer weight

                                    :functions

                                'adaptFcn: 'trains
                                (divideFcn: (none
                            'gradientFcn: 'gdefaults
                                 'initFcn: 'initlay
                                'performFcn: 'mse
                                        plotFcns:
{{'plotperform','plottrainstate','plotregression
                                'trainFcn: 'trainlm

                                        :parameters

                            adaptParam: .passes
                              (divideParam: (none
                             (gradientParam: (none
                                (initParam: (none
                             (performParam: (none

        trainParam: .show, .showWindow,
                        ,.showCommandLine, .epochs
                  time, .goal, .max_fail,.
                                        ,.mem_reduc
         min_grad, .mu, .mu_dec,.
                                            ,.mu_inc
```

```
                                mu_max.

                        :weight and bias values

    IW: {2x1 cell} containing 1 input

                                    weight matrix

    LW: {2x2 cell} containing 1 layer
                                    weight matrix

      b: {2x1 cell} containing 2 bias

                                      vectors

                                    :other

                                  " :name
                (userdata: (user information
```

جدول (C17-3) يمثل مراحل التدريب في الشبكة العصبية الاصطناعية للتجربة رقم 1

مراحل التدريب في الشبكة العصبية الاصطناعية للتجربة رقم 2

```
                                      = net

                        :Neural Network object

                                    :architecture

                                numInputs: 1
                               numLayers: 2
                            [biasConnect: [1; 1
                            [inputConnect: [1; 0
                        [layerConnect: [0 0; 1 0
                            [outputConnect: [0 1

                    (numOutputs: 1  (read-only
                (numInputDelays: 0  (read-only
```

```
            (numLayerDelays: 0  (read-only

                                :subobject structures

                inputs: {1x1 cell} of inputs
                layers: {2x1 cell} of layers
  outputs: {1x2 cell} containing 1 output
  biases: {2x1 cell} containing 2 biases
       inputWeights: {2x1 cell} containing 1
                                        input weight
layerWeights: {2x2 cell} containing 1 layer
                                          weight

                                        :functions

                                'adaptFcn: 'trains
                                (divideFcn: (none
                            'gradientFcn: 'gdefaults
                                 'initFcn: 'initlay
                                'performFcn: 'mse
                                        plotFcns:
 {'{'plotperform','plottrainstate','plotregression
                                'trainFcn: 'trainlm

                                     :parameters

                             adaptParam: .passes
                               (divideParam: (none
                              (gradientParam: (none
                                 (initParam: (none
                              (performParam: (none
          trainParam: .show, .showWindow,
                      ,.showCommandLine, .epochs
               time, .goal, .max_fail,.
                                          ,.mem_reduc
           min_grad, .mu, .mu_dec,.
                                              ,.mu_inc
                                mu_max.

                              :weight and bias values

       IW: {2x1 cell} containing 1 input
                                          weight matrix
```

```
    LW: {2x2 cell} containing 1 layer
                                    weight matrix
      b: {2x1 cell} containing 2 bias
                                        vectors

                                      :other

                                  " :name
            (userdata: (user information


                                        = net

                          :Neural Network object

                                  :architecture

                               numInputs: 1
                              numLayers: 2
                           [biasConnect: [1; 1
                           [inputConnect: [1; 0
                      [layerConnect: [0 0; 1 0
                           [outputConnect: [0 1

              (numOutputs: 1  (read-only
            (numInputDelays: 0  (read-only
            (numLayerDelays: 0  (read-only

                          :subobject structures

              inputs: {1x1 cell} of inputs
              layers: {2x1 cell} of layers
 outputs: {1x2 cell} containing 1 output
 biases: {2x1 cell} containing 2 biases
      inputWeights: {2x1 cell} containing 1
                                    input weight
layerWeights: {2x2 cell} containing 1 layer
                                        weight

                                    :functions

                           'adaptFcn: 'trains
                           (divideFcn: (none
```

'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
plotFcns:
{'{'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters

adaptParam: .passes
(divideParam: (none
(gradientParam: (none
(initParam: (none
(performParam: (none
trainParam: .show, .showWindow,
,.showCommandLine, .epochs
time, .goal, .max_fail,.
,.mem_reduc
min_grad, .mu, .mu_dec,.
,.mu_inc
mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input
weight matrix
LW: {2x2 cell} containing 1 layer
weight matrix
b: {2x1 cell} containing 2 bias
vectors

:other

" :name
(userdata: (user information

= net

:Neural Network object

:architecture

```
                                numInputs: 1
                               numLayers: 2
                              [biasConnect: [1; 1
                              [inputConnect: [1; 0
                           [layerConnect: [0 0; 1 0
                              [outputConnect: [0 1

                    (numOutputs: 1  (read-only
                   (numInputDelays: 0  (read-only
                  (numLayerDelays: 0  (read-only

                                :subobject structures

                     inputs: {1x1 cell} of inputs
                     layers: {2x1 cell} of layers
    outputs: {1x2 cell} containing 1 output
    biases: {2x1 cell} containing 2 biases
          inputWeights: {2x1 cell} containing 1
                                          input weight
  layerWeights: {2x2 cell} containing 1 layer
                                            weight

                                          :functions

                                'adaptFcn: 'trains
                                (divideFcn: (none
                             'gradientFcn: 'gdefaults
                                 'initFcn: 'initlay
                                'performFcn: 'mse
                                       :plotFcns

    {'plotperform','plottrainstate','plotregression'}
                                'trainFcn: 'trainlm

                                       :parameters

                             adaptParam: .passes
                              (divideParam: (none

                             (gradientParam: (none

                                (initParam: (none
                              (performParam: (none
```

```
,trainParam: .show, .showWindow

                ,showCommandLine, .epochs.
          time, .goal, .max_fail,.
                                    ,.mem_reduc
   min_grad, .mu, .mu_dec,.
                                          ,.mu_inc
                        mu_max.

                    :weight and bias values

  IW: {2x1 cell} containing 1 input
                                    weight matrix
LW: {2x2 cell} containing 1 layer

                                   weight matrix
    b: {2x1 cell} containing 2 bias
                                     vectors

                                    :other

                                 " :name
          (userdata: (user information
```

جدول (C19-3) يمثل مراحل التدريب في الشبكة العصبية الاصطناعية للتجربة رقم 2

مراحل التدريب في الشبكة العصبية الاصطناعية للتجربة رقم 3

```
                                    = net

                            :Neural Network object

                                    :architecture

                                 numInputs: 1
                                numLayers: 2
```

```
                    [biasConnect: [1; 1
                    [inputConnect: [1; 0
                [layerConnect: [0 0; 1 0
                    [outputConnect: [0 1

          (numOutputs: 1  (read-only
        (numInputDelays: 0  (read-only
       (numLayerDelays: 0  (read-only

                        :subobject structures

          inputs: {1x1 cell} of inputs
          layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
    inputWeights: {2x1 cell} containing 1
                                  input weight
layerWeights: {2x2 cell} containing 1 layer
                                    weight

                                  :functions

                    'adaptFcn: 'trains
                    (divideFcn: (none
                'gradientFcn: 'gdefaults
                     'initFcn: 'initlay
                    'performFcn: 'mse
                            plotFcns:
{'{'plotperform','plottrainstate','plotregression
                    'trainFcn: 'trainlm

                               :parameters

                 adaptParam: .passes
                   (divideParam: (none
                  (gradientParam: (none
                     (initParam: (none
                  (performParam: (none
    trainParam: .show, .showWindow,
                ,.showCommandLine, .epochs
          time, .goal, .max_fail,.
                                ,.mem_reduc
     min_grad, .mu, .mu_dec,.
```

,.mu_inc
mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input
weight matrix
LW: {2x2 cell} containing 1 layer
weight matrix
b: {2x1 cell} containing 2 bias
vectors

:other

" :name
(userdata: (user information

جدول (C21-3) يمثل مراحل التدريب في الشبكة العصبية الاصطناعية للتجربة 3

مراحل التدريب في الشبكة العصبية الاصطناعية للتجربة رقم 4

= net

:Neural Network object

:architecture

numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1 (read-only
(numInputDelays: 0 (read-only
(numLayerDelays: 0 (read-only

:subobject structures

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers

```
    outputs: {1x2 cell} containing 1 output
    biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input
                                        weight
layerWeights: {2x2 cell} containing 1 layer
                                        weight

                                    :functions

                                'adaptFcn: 'trains
                                (divideFcn: (none
                            'gradientFcn: 'gdefaults
                                'initFcn: 'initlay
                                'performFcn: 'mse
                                        plotFcns:
{{'plotperform','plottrainstate','plotregression
                                'trainFcn: 'trainlm

                                    :parameters

                            adaptParam: .passes
                              (divideParam: (none
                            (gradientParam: (none
                              (initParam: (none
                            (performParam: (none
          trainParam: .show, .showWindow,
                        ,.showCommandLine, .epochs
,time, .goal, .max_fail, .mem_reduc.
 ,min_grad, .mu, .mu_dec, .mu_inc.
                                    mu_max.

                                :weight and bias values

 IW: {2x1 cell} containing 1 input weight
                                        matrix
LW: {2x2 cell} containing 1 layer weight
                                        matrix
  b: {2x1 cell} containing 2 bias vectors

                                    :other

                                    '' :name
                        (userdata: (user information
```

= net

:Neural Network object

:architecture

numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1 (read-only
(numInputDelays: 0 (read-only
(numLayerDelays: 0 (read-only

:subobject structures

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input
weight
layerWeights: {2x2 cell} containing 1 layer
weight

:functions

'adaptFcn: 'trains
(divideFcn: (none
'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
plotFcns:
{'{'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters

```
                                adaptParam: .passes
                                  (divideParam: (none
                                 (gradientParam: (none
                                    (initParam: (none
                                 (performParam: (none
                trainParam: .show, .showWindow,
                                ,.showCommandLine, .epochs
   ,time, .goal, .max_fail, .mem_reduc.
    ,min_grad, .mu, .mu_dec, .mu_inc.
                                        mu_max.

                                :weight and bias values

  IW: {2x1 cell} containing 1 input weight
                                                matrix
 LW: {2x2 cell} containing 1 layer weight
                                                matrix
   b: {2x1 cell} containing 2 bias vectors

                                                :other

                                                " :name
                          (userdata: (user information


                                                = net

                                :Neural Network object

                                                :architecture

                                        numInputs: 1
                                       numLayers: 2
                                    [biasConnect: [1; 1
                                    [inputConnect: [1; 0
                                [layerConnect: [0 0; 1 0
                                    [outputConnect: [0 1

                             (numOutputs: 1  (read-only
                           (numInputDelays: 0  (read-only
                           (numLayerDelays: 0  (read-only

                                :subobject structures
```

```
                    inputs: {1x1 cell} of inputs
                    layers: {2x1 cell} of layers
  outputs: {1x2 cell} containing 1 output
  biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input
                                                weight
layerWeights: {2x2 cell} containing 1 layer
                                                weight

                                            :functions

                                    'adaptFcn: 'trains
                                    (divideFcn: (none
                                'gradientFcn: 'gdefaults
                                     'initFcn: 'initlay
                                    'performFcn: 'mse
                                            plotFcns:
{'{'plotperform','plottrainstate','plotregression
                                    'trainFcn: 'trainlm

                                            :parameters

                                adaptParam: .passes
                                  (divideParam: (none
                                 (gradientParam: (none
                                    (initParam: (none
                                 (performParam: (none
          trainParam: .show, .showWindow,
                          ,.showCommandLine, .epochs
,time, .goal, .max_fail, .mem_reduc.
 ,min_grad, .mu, .mu_dec, .mu_inc.
                                        mu_max.

                                :weight and bias values

 IW: {2x1 cell} containing 1 input weight
                                                matrix
LW: {2x2 cell} containing 1 layer weight
                                                matrix
  b: {2x1 cell} containing 2 bias vectors

                                            :other
```

```
                                                    " :name
                                    (userdata: (user information
```

جدول (C23-3) يمثل مراحل تدريب الشبكة للتجربة رقم 4

مراحل التدريب في الشبكة العصبية الاصطناعية للتجربة رقم 5

```
                                                                = net

                                                  :Neural Network object

                                                          :architecture

                                                        numInputs: 1
                                                      numLayers: 2
                                                  [biasConnect: [1; 1
                                                  [inputConnect: [1; 0
                                              [layerConnect: [0 0; 1 0
                                                  [outputConnect: [0 1

                                          (numOutputs: 1  (read-only
                                        (numInputDelays: 0  (read-only
                                        (numLayerDelays: 0  (read-only

                                                    :subobject structures

                                          inputs: {1x1 cell} of inputs
                                          layers: {2x1 cell} of layers
                                  outputs: {1x2 cell} containing 1 output
                                  biases: {2x1 cell} containing 2 biases
                              inputWeights: {2x1 cell} containing 1 input
                                                                  weight
                              layerWeights: {2x2 cell} containing 1 layer
                                                                  weight

                                                          :functions

                                                  'adaptFcn: 'trains
                                                  (divideFcn: (none
                                              'gradientFcn: 'gdefaults
                                                    'initFcn: 'initlay
                                                  'performFcn: 'mse
                                                        plotFcns:
```

{{'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters

adaptParam: .passes
(divideParam: (none
(gradientParam: (none
(initParam: (none
(performParam: (none
trainParam: .show, .showWindow,
,.showCommandLine, .epochs
,time, .goal, .max_fail, .mem_reduc.
,min_grad, .mu, .mu_dec, .mu_inc.
mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight
matrix
LW: {2x2 cell} containing 1 layer weight
matrix
b: {2x1 cell} containing 2 bias vectors

:other

" :name
(userdata: (user information

= net

:Neural Network object

:architecture

numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

```
                (numOutputs: 1  (read-only
              (numInputDelays: 0  (read-only
             (numLayerDelays: 0  (read-only

                                  :subobject structures

                        inputs: {1x1 cell} of inputs
                        layers: {2x1 cell} of layers
      outputs: {1x2 cell} containing 1 output
      biases: {2x1 cell} containing 2 biases
    inputWeights: {2x1 cell} containing 1 input
                                                weight
    layerWeights: {2x2 cell} containing 1 layer
                                                weight

                                               :functions

                                  'adaptFcn: 'trains
                                  (divideFcn: (none
                             'gradientFcn: 'gdefaults
                                   'initFcn: 'initlay
                                  'performFcn: 'mse
                                        plotFcns:
      {'{'plotperform','plottrainstate','plotregression
                                  'trainFcn: 'trainlm

                                               :parameters

                               adaptParam: .passes
                                 (divideParam: (none
                                (gradientParam: (none
                                   (initParam: (none
                                (performParam: (none
            trainParam: .show, .showWindow,
                                ,.showCommandLine, .epochs
    ,time, .goal, .max_fail, .mem_reduc.
     ,min_grad, .mu, .mu_dec, .mu_inc.
                                        mu_max.

                                  :weight and bias values

    IW: {2x1 cell} containing 1 input weight
                                                matrix
```

```
LW: {2x2 cell} containing 1 layer weight
                                                    matrix
  b: {2x1 cell} containing 2 bias vectors

                                                :other

                                             '' :name
                                (userdata: (user information


                                                 = net

                                       :Neural Network object

                                            :architecture

                                          numInputs: 1
                                         numLayers: 2
                                      [biasConnect: [1; 1
                                      [inputConnect: [1; 0
                                  [layerConnect: [0 0; 1 0
                                      [outputConnect: [0 1

                              (numOutputs: 1  (read-only
                             (numInputDelays: 0  (read-only
                            (numLayerDelays: 0  (read-only

                                       :subobject structures

                                 inputs: {1x1 cell} of inputs
                                 layers: {2x1 cell} of layers
                    outputs: {1x2 cell} containing 1 output
                    biases: {2x1 cell} containing 2 biases
                inputWeights: {2x1 cell} containing 1 input
                                                    weight
                layerWeights: {2x2 cell} containing 1 layer
                                                    weight

                                                :functions

                                      'adaptFcn: 'trains
                                      (divideFcn: (none
                                  'gradientFcn: 'gdefaults
```

'initFcn: 'initlay
'performFcn: 'mse
plotFcns:
{'{'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters

adaptParam: .passes
(divideParam: (none

(gradientParam: (none
(initParam: (none

(performParam: (none

,trainParam: .show, .showWindow

,showCommandLine, .epochs.

,time, .goal, .max_fail, .mem_reduc.

,min_grad, .mu, .mu_dec, .mu_inc.
mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight
matrix
LW: {2x2 cell} containing 1 layer weight
matrix
b: {2x1 cell} containing 2 bias vectors

:other

" :name
(userdata: (user information

جدول (C25-3) يمثل مراحل التدريب في الشبكة العصبية الاصطناعية للتجربة رقم 5

مراحل التدريب في الشبكة العصبية الاصطناعية للتجربة رقم 6

```
= net

:Neural Network object

:architecture

numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1  (read-only
(numInputDelays: 0  (read-only
(numLayerDelays: 0  (read-only

:subobject structures

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1
input weight
layerWeights: {2x2 cell} containing 1 layer
weight

:functions

'adaptFcn: 'trains
(divideFcn: (none
'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
plotFcns:
{'{'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters
```

```
                    adaptParam: .passes
                      (divideParam: (none
                     (gradientParam: (none
                        (initParam: (none
                     (performParam: (none
    trainParam: .show, .showWindow,
                ,.showCommandLine, .epochs
           time, .goal, .max_fail,.
                                  ,.mem_reduc
     min_grad, .mu, .mu_dec,.
                                       ,.mu_inc
                          mu_max.

                     :weight and bias values

  IW: {2x1 cell} containing 1 input
                                  weight matrix
LW: {2x2 cell} containing 1 layer
                                  weight matrix
    b: {2x1 cell} containing 2 bias
                                      vectors

                                     :other

                                 " :name
           (userdata: (user information


                                          = net

                     :Neural Network object

                                  :architecture

                              numInputs: 1
                             numLayers: 2
                          [biasConnect: [1; 1
                          [inputConnect: [1; 0
                      [layerConnect: [0 0; 1 0
                          [outputConnect: [0 1

                (numOutputs: 1  (read-only
               (numInputDelays: 0  (read-only
```

```
            (numLayerDelays: 0  (read-only

                                :subobject structures

                inputs: {1x1 cell} of inputs
                layers: {2x1 cell} of layers
  outputs: {1x2 cell} containing 1 output
  biases: {2x1 cell} containing 2 biases
       inputWeights: {2x1 cell} containing 1
                                        input weight
layerWeights: {2x2 cell} containing 1 layer
                                          weight

                                        :functions

                                'adaptFcn: 'trains
                                (divideFcn: (none
                            'gradientFcn: 'gdefaults
                                 'initFcn: 'initlay
                                'performFcn: 'mse
                                        plotFcns:
 {'{'plotperform','plottrainstate','plotregression
                                'trainFcn: 'trainlm

                                     :parameters

                             adaptParam: .passes
                               (divideParam: (none
                              (gradientParam: (none
                                 (initParam: (none
                              (performParam: (none
          trainParam: .show, .showWindow,
                      ,.showCommandLine, .epochs
               time, .goal, .max_fail,.
                                          ,.mem_reduc
           min_grad, .mu, .mu_dec,.
                                              ,.mu_inc
                                mu_max.

                              :weight and bias values

       IW: {2x1 cell} containing 1 input
                                          weight matrix
```

```
    LW: {2x2 cell} containing 1 layer
                                    weight matrix
      b: {2x1 cell} containing 2 bias
                                        vectors

                                      :other

                                  " :name
            (userdata: (user information


                                        = net

                          :Neural Network object

                                  :architecture

                               numInputs: 1
                              numLayers: 2
                           [biasConnect: [1; 1
                           [inputConnect: [1; 0
                      [layerConnect: [0 0; 1 0
                           [outputConnect: [0 1

              (numOutputs: 1  (read-only
            (numInputDelays: 0  (read-only
            (numLayerDelays: 0  (read-only

                          :subobject structures

              inputs: {1x1 cell} of inputs
              layers: {2x1 cell} of layers
 outputs: {1x2 cell} containing 1 output
 biases: {2x1 cell} containing 2 biases
      inputWeights: {2x1 cell} containing 1
                                    input weight
layerWeights: {2x2 cell} containing 1 layer
                                        weight

                                    :functions

                           'adaptFcn: 'trains
                           (divideFcn: (none
```

'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
plotFcns:
{'{'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters

adaptParam: .passes
(divideParam: (none

(gradientParam: (none

(initParam: (none
(performParam: (none

,trainParam: .show, .showWindow

,showCommandLine, .epochs.

time, .goal, .max_fail,.
,.mem_reduc
min_grad, .mu, .mu_dec,.
,.mu_inc
mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input

weight matrix

LW: {2x2 cell} containing 1 layer

weight matrix
b: {2x1 cell} containing 2 bias
vectors

:other

" :name
(userdata: (user information

جدول (C27-3) يمثل مراحل التدريب في الشبكة العصبية الاصطناعية للتجربة رقم 6

مراحل التدريب في الشبكة العصبية الاصطناعية للتجربة رقم 7

```
                                                    = net

                                        :Neural Network object

                                                :architecture

                                                numInputs: 1
                                               numLayers: 2
                                            [biasConnect: [1; 1
                                            [inputConnect: [1; 0
                                        [layerConnect: [0 0; 1 0
                                            [outputConnect: [0 1

                                    (numOutputs: 1  (read-only
                                (numInputDelays: 0  (read-only
                                (numLayerDelays: 0  (read-only

                                                :subobject structures

                                        inputs: {1x1 cell} of inputs
                                        layers: {2x1 cell} of layers
                    outputs: {1x2 cell} containing 1 output
                    biases: {2x1 cell} containing 2 biases
                        inputWeights: {2x1 cell} containing 1
                                                        input weight
                        layerWeights: {2x2 cell} containing 1
                                                        layer weight

                                                :functions

                                            'adaptFcn: 'trains
                                            (divideFcn: (none
                                        'gradientFcn: 'gdefaults
                                            'initFcn: 'initlay
```

'performFcn: 'mse
plotFcns:
{'{'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters

adaptParam: .passes
(divideParam: (none
(gradientParam: (none
(initParam: (none
(performParam: (none
trainParam: .show, .showWindow,
,.showCommandLine, .epochs
time, .goal, .max_fail,.
,.mem_reduc
min_grad, .mu, .mu_dec,.
,.mu_inc
mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input
weight matrix
LW: {2x2 cell} containing 1 layer
weight matrix
b: {2x1 cell} containing 2 bias
vectors

:other

" :name
(userdata: (user information

net.trainparam.goal=1e-5 <<

= net

:Neural Network object

:architecture

numInputs: 1

numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1 (read-only
(numInputDelays: 0 (read-only
(numLayerDelays: 0 (read-only

:subobject structures

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1
input weight
layerWeights: {2x2 cell} containing 1
layer weight

:functions

'adaptFcn: 'trains
(divideFcn: (none
'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
plotFcns:
{'{'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters

adaptParam: .passes
(divideParam: (none
(gradientParam: (none
(initParam: (none
(performParam: (none
trainParam: .show, .showWindow,
,.showCommandLine, .epochs
time, .goal, .max_fail,.
,.mem_reduc

```
        min_grad, .mu, .mu_dec,.
                                        ,.mu_inc
                            mu_max.

                        :weight and bias values

    IW: {2x1 cell} containing 1 input
                                    weight matrix
   LW: {2x2 cell} containing 1 layer
                                    weight matrix
      b: {2x1 cell} containing 2 bias
                                      vectors

                                     :other

                                  '' :name
              (userdata: (user information

                          'net.performfcn='mse <<

                                       = net

                          :Neural Network object

                                  :architecture

                                numInputs: 1
                               numLayers: 2
                             [biasConnect: [1; 1
                             [inputConnect: [1; 0
                        [layerConnect: [0 0; 1 0
                             [outputConnect: [0 1

                  (numOutputs: 1  (read-only
                 (numInputDelays: 0  (read-only
                (numLayerDelays: 0  (read-only

                          :subobject structures

                  inputs: {1x1 cell} of inputs
                  layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
```

```
    inputWeights: {2x1 cell} containing 1
                                    input weight
    layerWeights: {2x2 cell} containing 1
                                    layer weight

                                    :functions

                              'adaptFcn: 'trains
                              (divideFcn: (none
                         'gradientFcn: 'gdefaults
                               'initFcn: 'initlay

                                  'performFcn: 'mse
                                    :plotFcns

{'plotperform','plottrainstate','plotregression'}

                              'trainFcn: 'trainlm

                                       :parameters

                           adaptParam: .passes

                             (divideParam: (none

                           (gradientParam: (none

                              (initParam: (none
                           (performParam: (none

        trainParam: .show, .showWindow

                    showCommandLine, .epochs. ,
                                                ,
             time, .goal, .max_fail,.
                                    ,.mem_reduc
        min_grad, .mu, .mu_dec,.
                                          ,.mu_inc
                           mu_max.

                           :weight and bias values

    IW: {2x1 cell} containing 1 input
```

weight matrix
LW: {2x2 cell} containing 1 layer
weight matrix
b: {2x1 cell} containing 2 bias
vectors

:other

" :name
(userdata: (user information

جدول (C29-3) يمثل مراحل التدريب لشبكة تجربة رقم 7

مراحل التدريب في الشبكة العصبية الاصطناعية للتجربة رقم 8

= net

:Neural Network object

:architecture

numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1 (read-only
(numInputDelays: 0 (read-only
(numLayerDelays: 0 (read-only

:subobject structures

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1
input weight
layerWeights: {2x2 cell} containing 1
layer weight

:functions

'adaptFcn: 'trains
(divideFcn: (none
'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
plotFcns:
{'{'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters

adaptParam: .passes
(divideParam: (none
(gradientParam: (none
(initParam: (none
(performParam: (none
trainParam: .show, .showWindow,
,.showCommandLine, .epochs
time, .goal, .max_fail,.
,.mem_reduc
min_grad, .mu, .mu_dec,.
,.mu_inc
mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input
weight matrix
LW: {2x2 cell} containing 1 layer
weight matrix
b: {2x1 cell} containing 2 bias
vectors

:other

'' :name
(userdata: (user information

= net

:Neural Network object

```
                                    :architecture

                                      numInputs: 1
                                     numLayers: 2
                                  [biasConnect: [1; 1
                                  [inputConnect: [1; 0
                               [layerConnect: [0 0; 1 0
                                  [outputConnect: [0 1

                            (numOutputs: 1  (read-only
                           (numInputDelays: 0  (read-only
                          (numLayerDelays: 0  (read-only

                                    :subobject structures

                           inputs: {1x1 cell} of inputs
                           layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
      inputWeights: {2x1 cell} containing 1
                                          input weight
      layerWeights: {2x2 cell} containing 1
                                          layer weight

                                    :functions

                                  'adaptFcn: 'trains
                                  (divideFcn: (none
                               'gradientFcn: 'gdefaults
                                    'initFcn: 'initlay
                                 'performFcn: 'mse
                                       plotFcns:
{{'plotperform','plottrainstate','plotregression
                                 'trainFcn: 'trainlm

                                    :parameters

                               adaptParam: .passes
                                (divideParam: (none
                              (gradientParam: (none
                                 (initParam: (none
                              (performParam: (none
```

```
    trainParam: .show, .showWindow,
                ,.showCommandLine, .epochs
          time, .goal, .max_fail,.
                                ,.mem_reduc
     min_grad, .mu, .mu_dec,.
                                      ,.mu_inc
                          mu_max.

                      :weight and bias values

  IW: {2x1 cell} containing 1 input
                                   weight matrix
 LW: {2x2 cell} containing 1 layer
                                   weight matrix
    b: {2x1 cell} containing 2 bias
                                     vectors

                                    :other

                                 " :name
           (userdata: (user information


                                        = net

                      :Neural Network object

                                 :architecture

                              numInputs: 1
                             numLayers: 2
                        [biasConnect: [1; 1
                        [inputConnect: [1; 0
                   [layerConnect: [0 0; 1 0
                        [outputConnect: [0 1

             (numOutputs: 1  (read-only
          (numInputDelays: 0  (read-only
          (numLayerDelays: 0  (read-only

                      :subobject structures

           inputs: {1x1 cell} of inputs
```

```
                layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
      inputWeights: {2x1 cell} containing 1
                                        input weight
      layerWeights: {2x2 cell} containing 1
                                        layer weight

                                        :functions

                                  'adaptFcn: 'trains
                                  (divideFcn: (none
                               'gradientFcn: 'gdefaults
                                    'initFcn: 'initlay
                                  'performFcn: 'mse
                                          plotFcns:
{{'plotperform','plottrainstate','plotregression
                                  'trainFcn: 'trainlm

                                          :parameters

                               adaptParam: .passes

                                (divideParam: (none

                               (gradientParam: (none

                                  (initParam: (none

                               (performParam: (none

       ,trainParam: .show, .showWindow

                         ,showCommandLine, .epochs.

                  ,time, .goal, .max_fail.

                                        ,mem_reduc.

        min_grad, .mu, .mu_dec,.
                                              ,.mu_inc
                                 mu_max.
```

:weight and bias values

IW: {2x1 cell} containing 1 input
weight matrix

LW: {2x2 cell} containing 1 layer

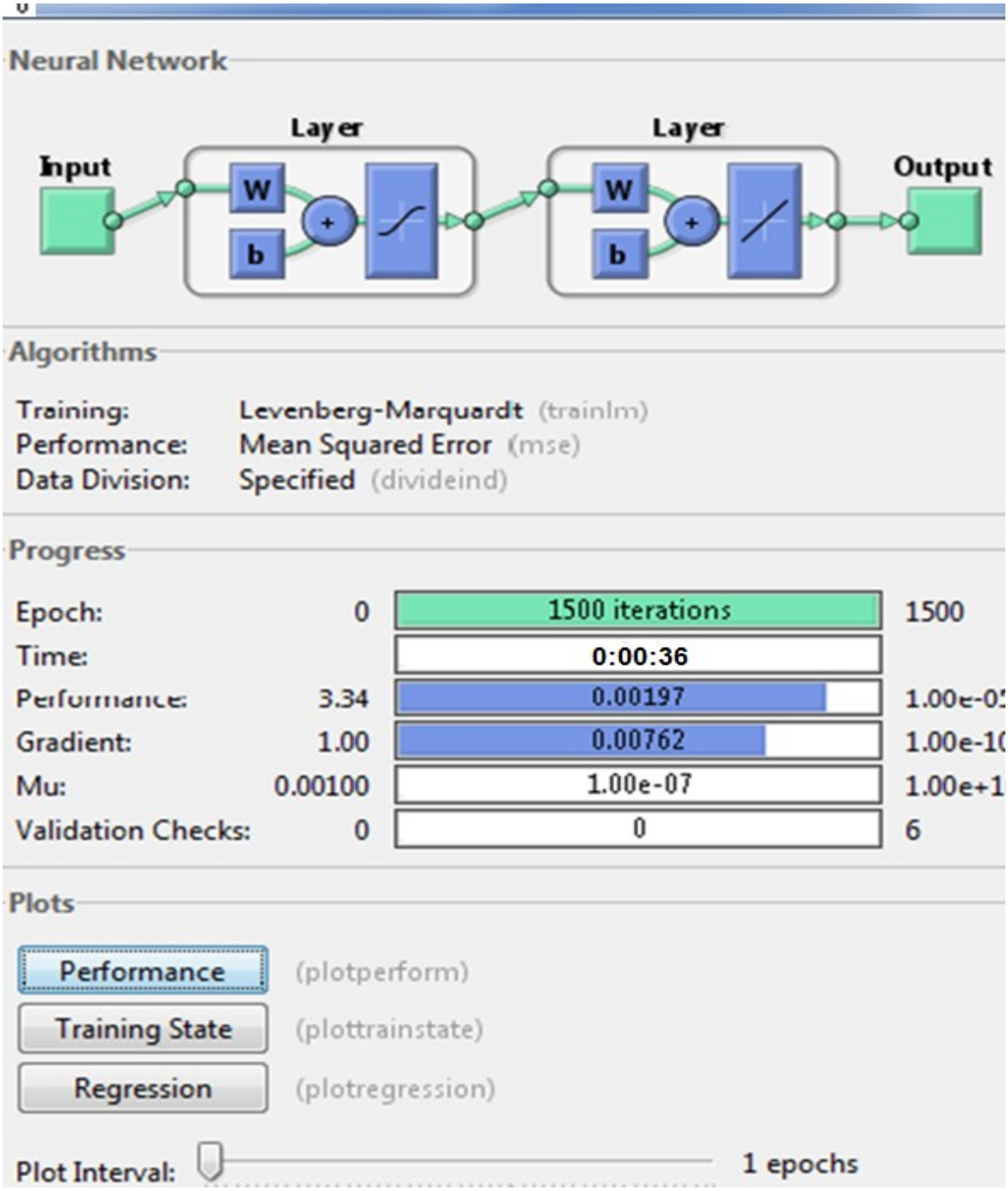
weight matrix
b: {2x1 cell} containing 2 bias
vectors

:other

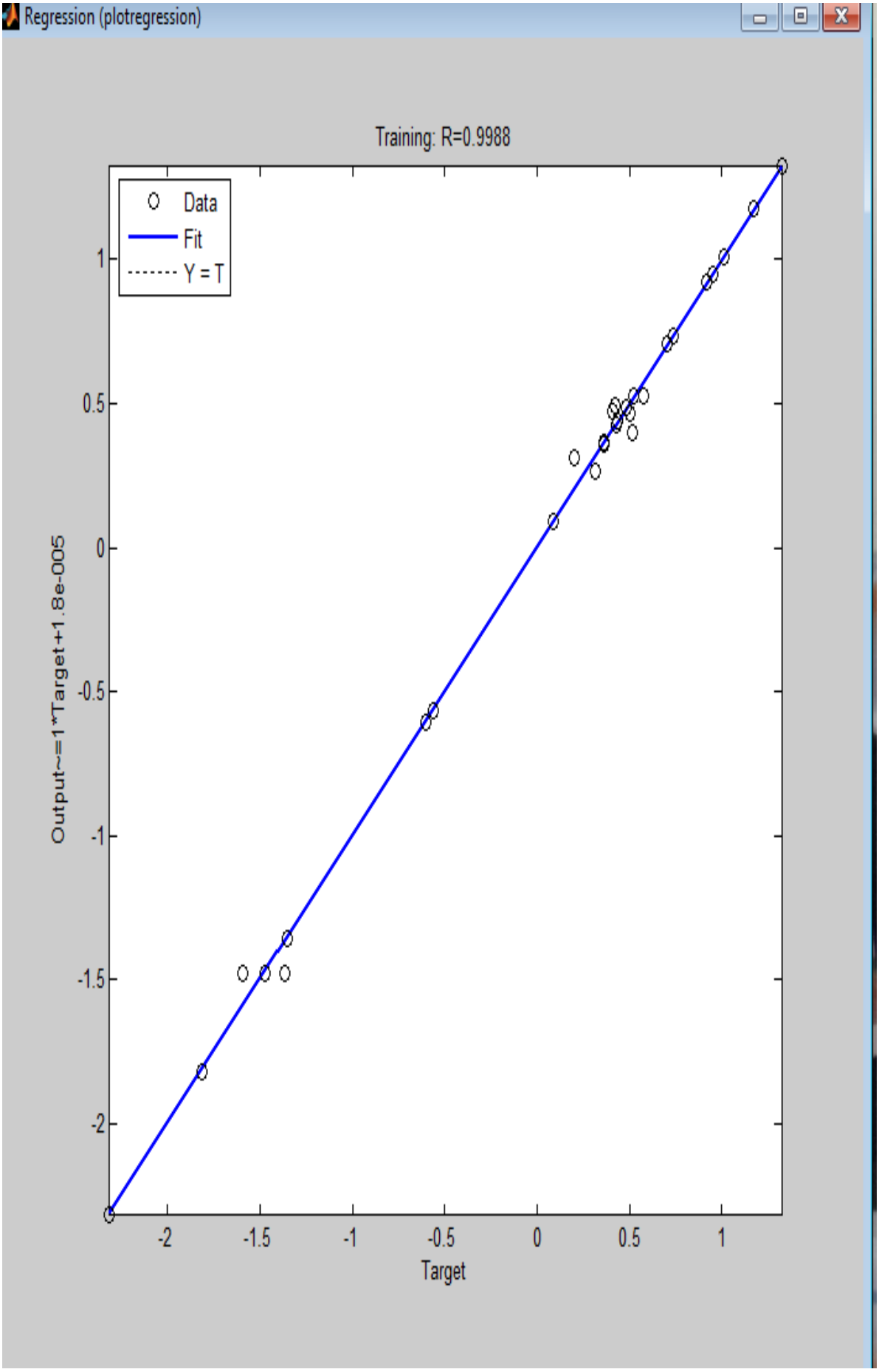
" :name
(userdata: (user information

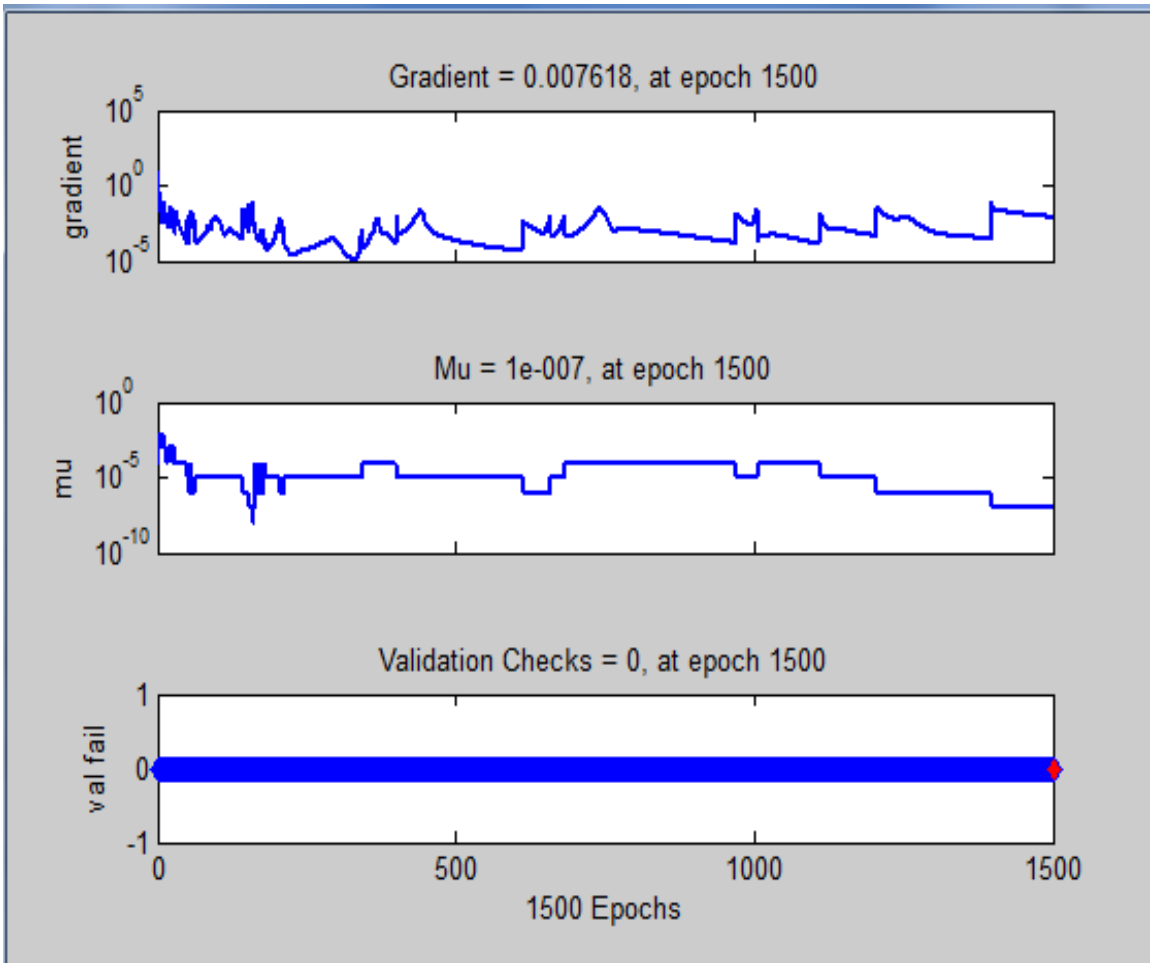
جدول (C31-3) يمثل مراحل تدريب شبكة التجربة رقم 8

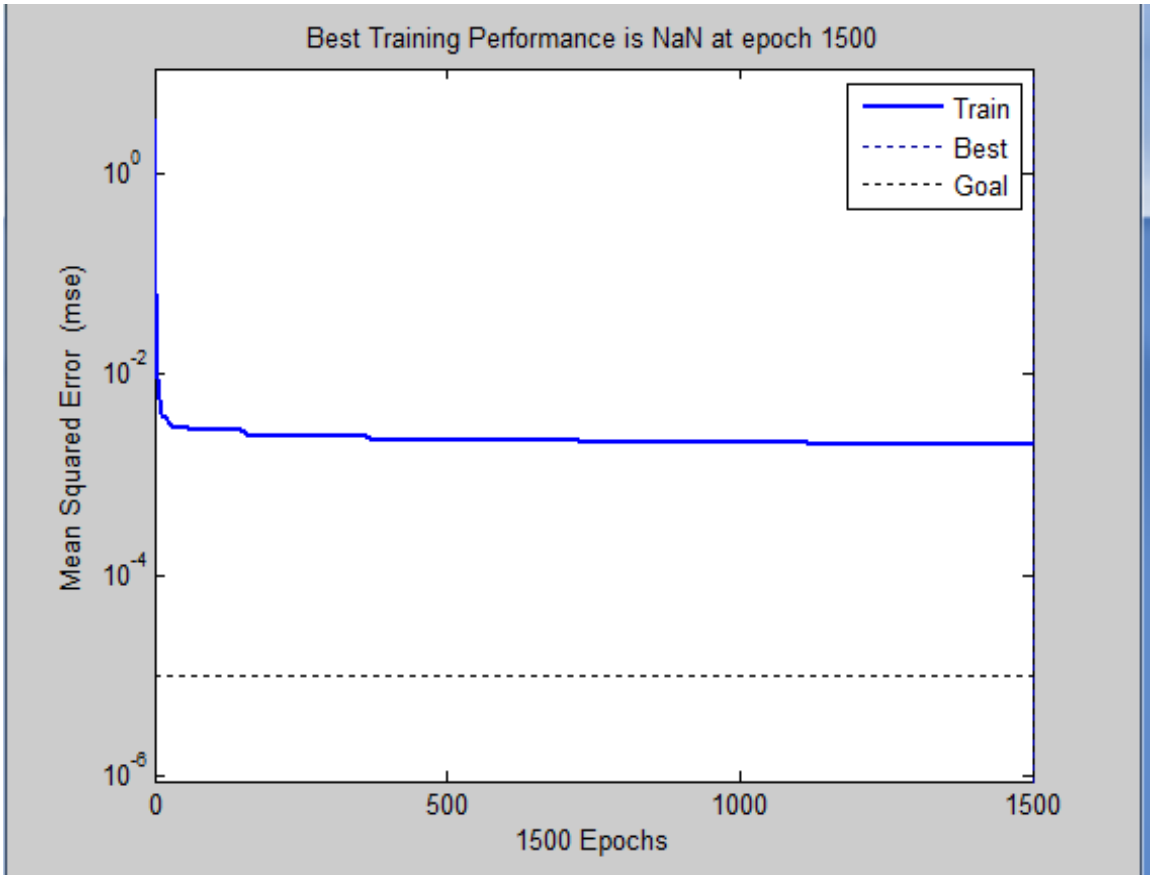
ملاحق الفصل الرابع

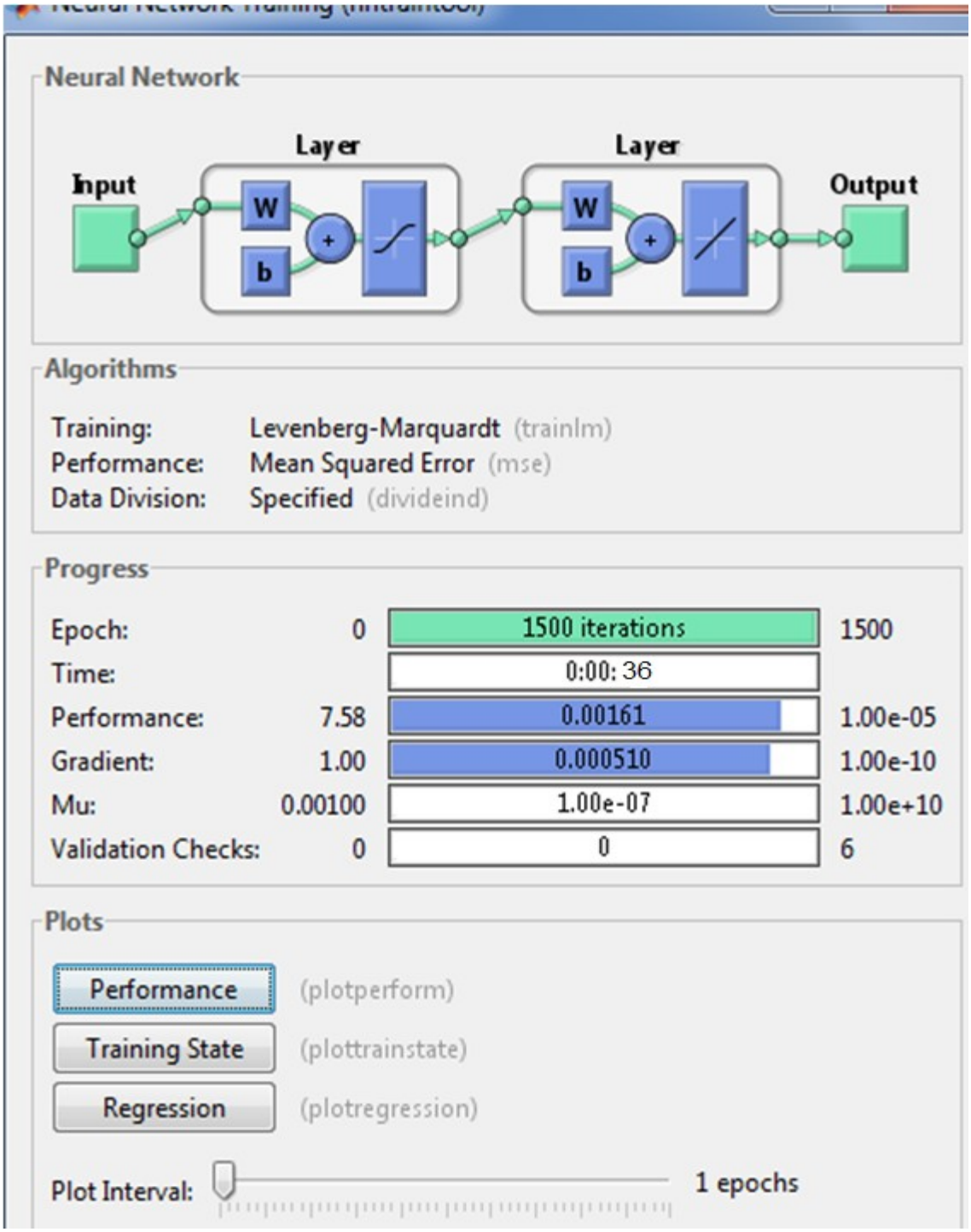


شكل (C8-4) يمثل تقييم الشبكة العصبية الاصطناعية التقليدية للمرحلة 1

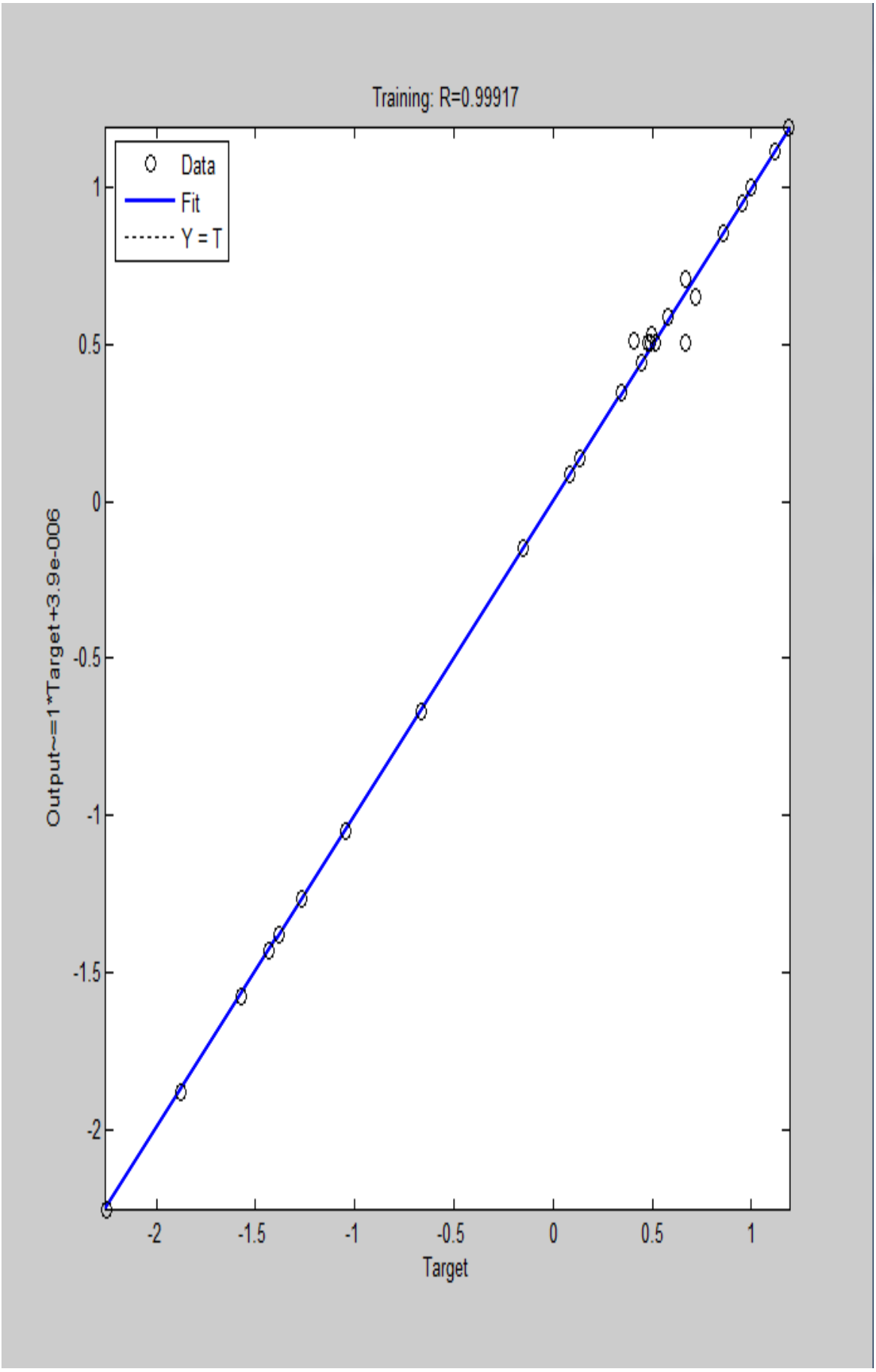

Regression (plotregression)
Training: R=0.9988
Data
Fit
Y = T
Output~=1*Target+1.8e-005
Target
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1

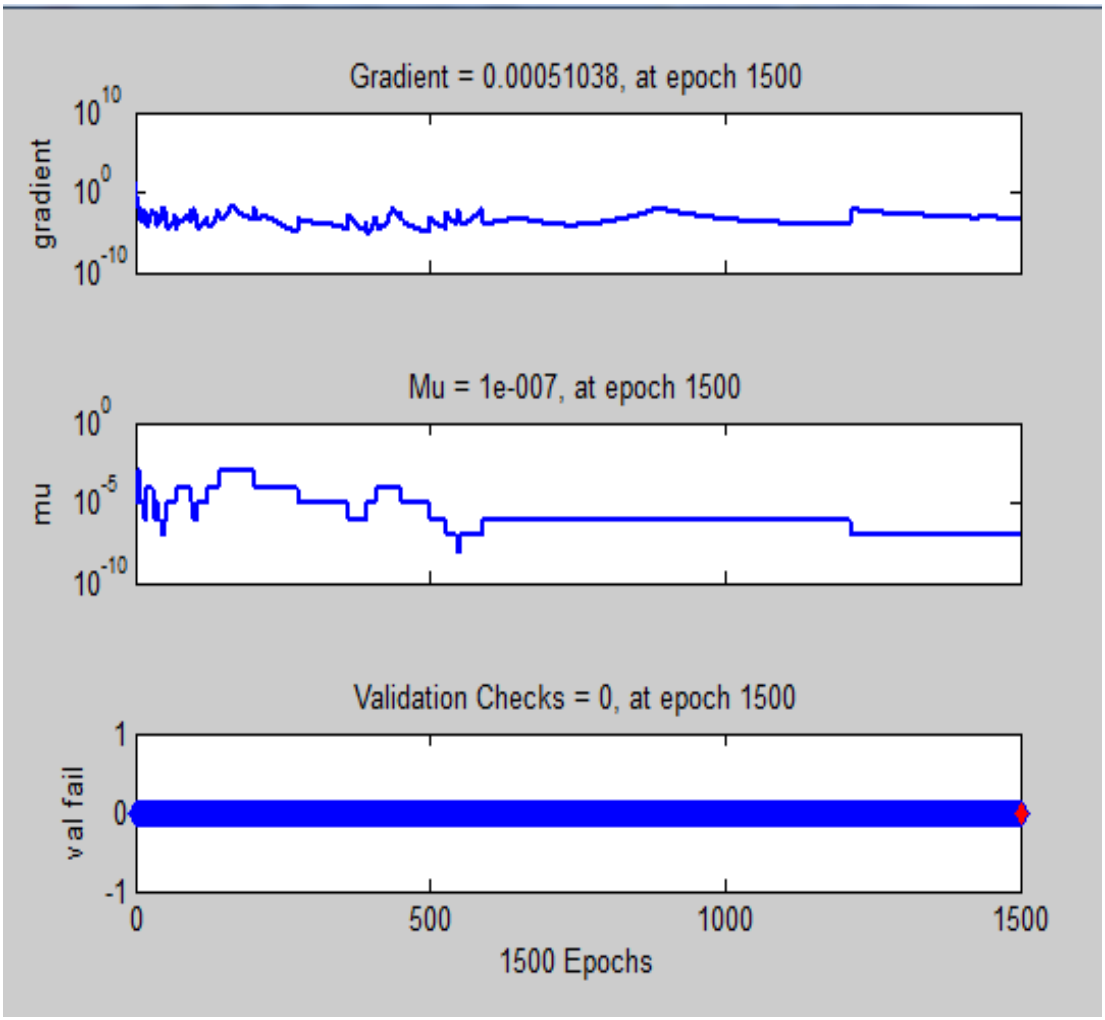

Gradient = 0.007618, at epoch 1500
gradient
10^5
10^0
10^-5
Mu = 1e-007, at epoch 1500
mu
10^0
10^-5
10^-10
Validation Checks = 0, at epoch 1500
val fail
1
0
-1
0
500
1000
1500
1500 Epochs

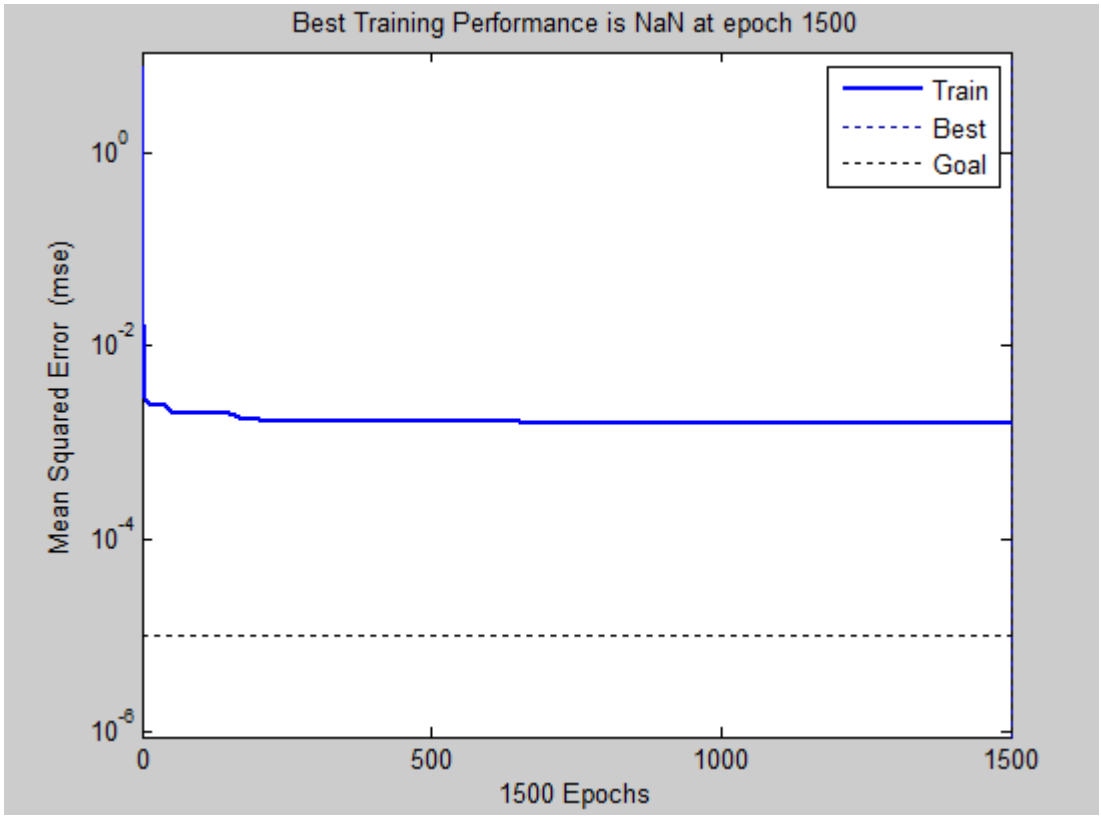

Best Training Performance is NaN at epoch 1500
Train
Best
Goal
Mean Squared Error (mse)
10^0
10^-2
10^-4
10^-6
0
500
1000
1500
1500 Epochs

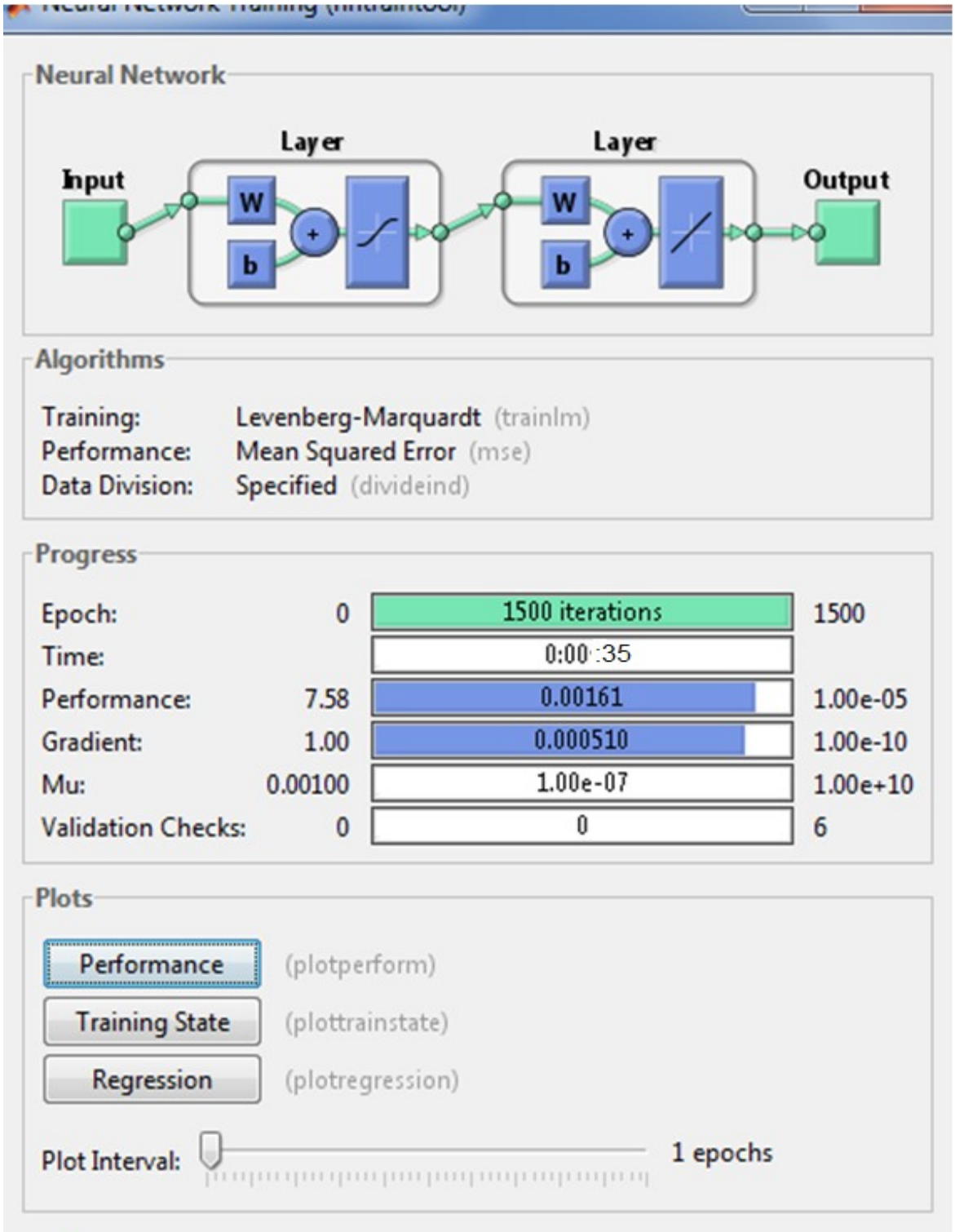


شكل (C9-4) يمثل تقييم الشبكة العصبية الاصطناعية التقليدية للمرحلة الثانية

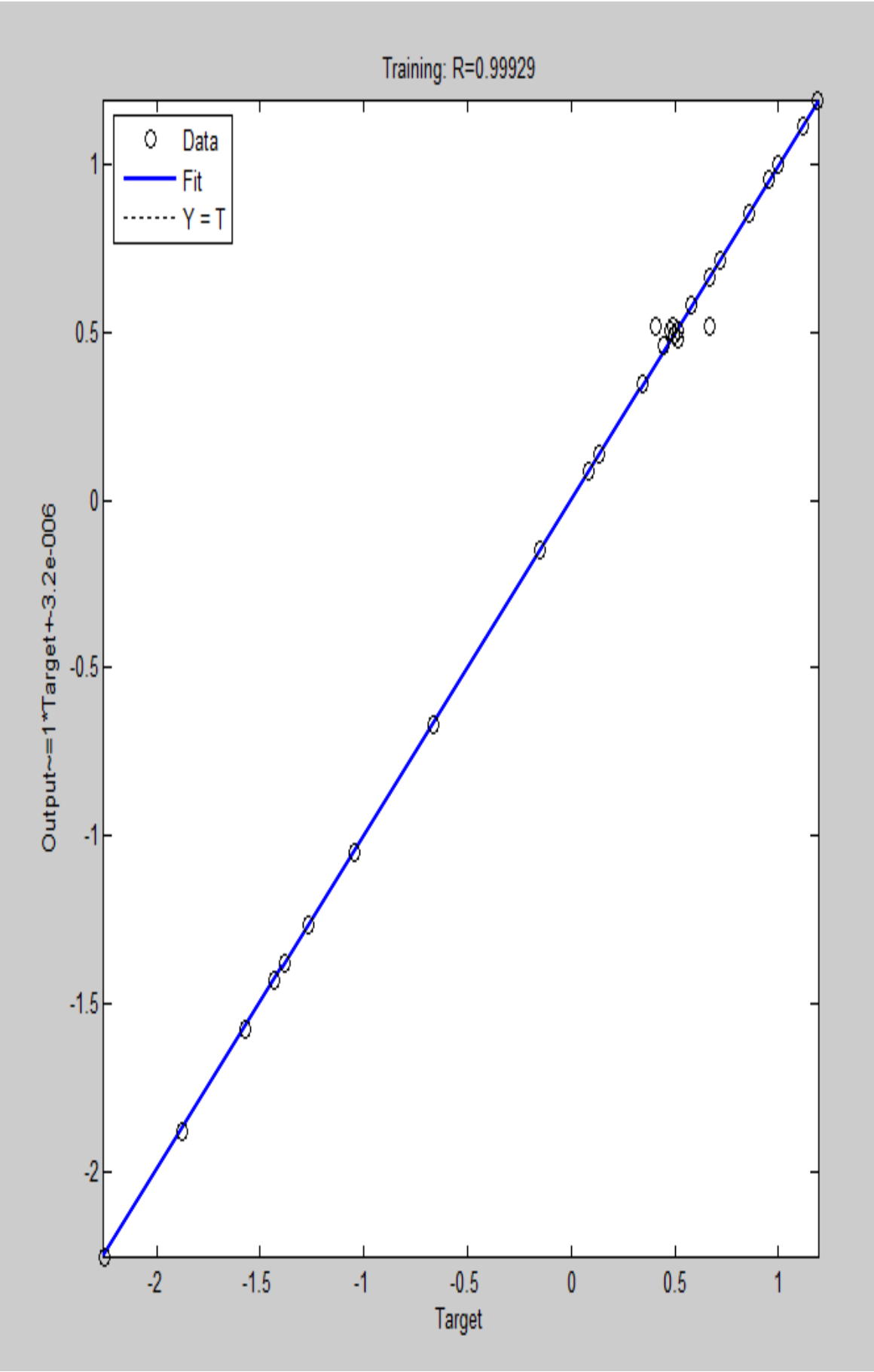

Training: R=0.99917
Data
Fit
Y = T
Output~=1*Target+3.9e-006
Target
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1

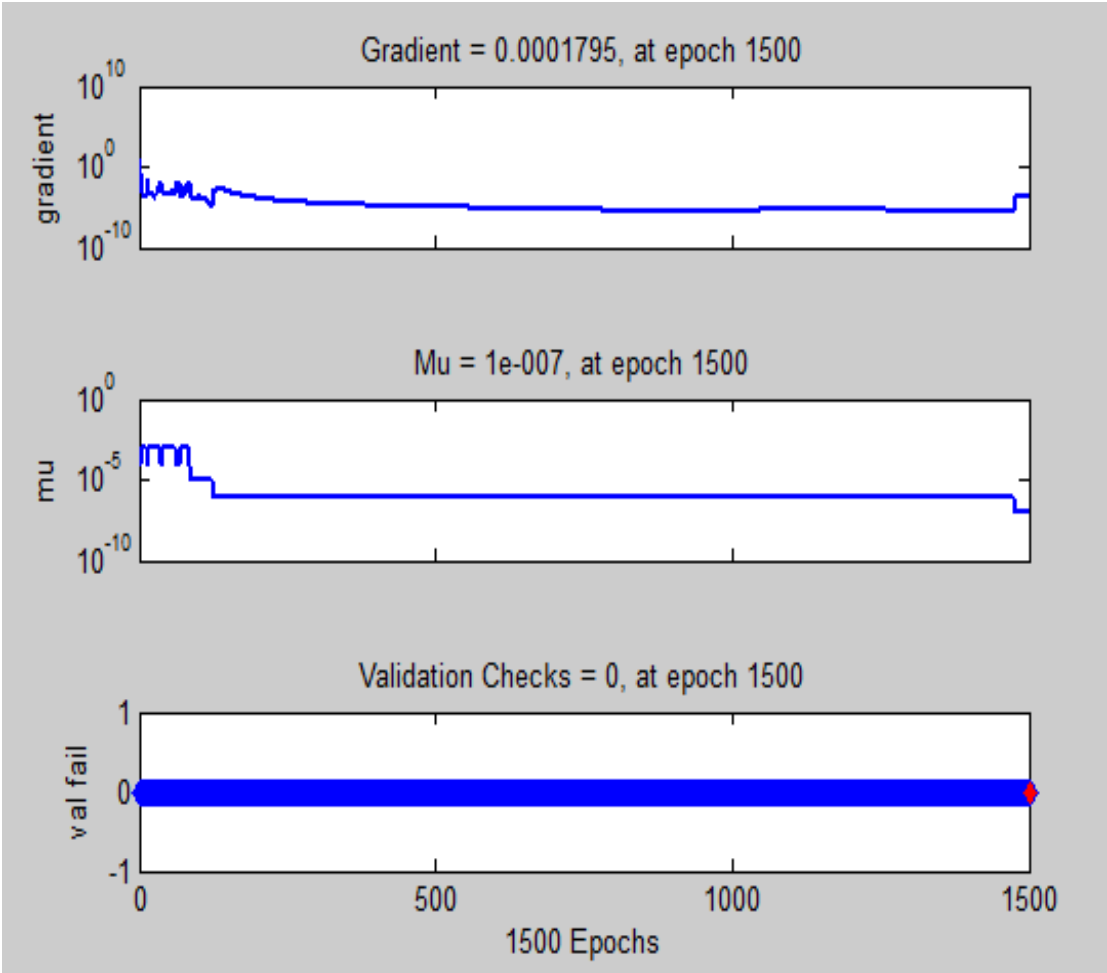

Gradient = 0.00051038, at epoch 1500
gradient
10^10
10^0
10^-10
Mu = 1e-007, at epoch 1500
mu
10^0
10^-5
10^-10
Validation Checks = 0, at epoch 1500
val fail
1
0
-1
0
500
1000
1500
1500 Epochs

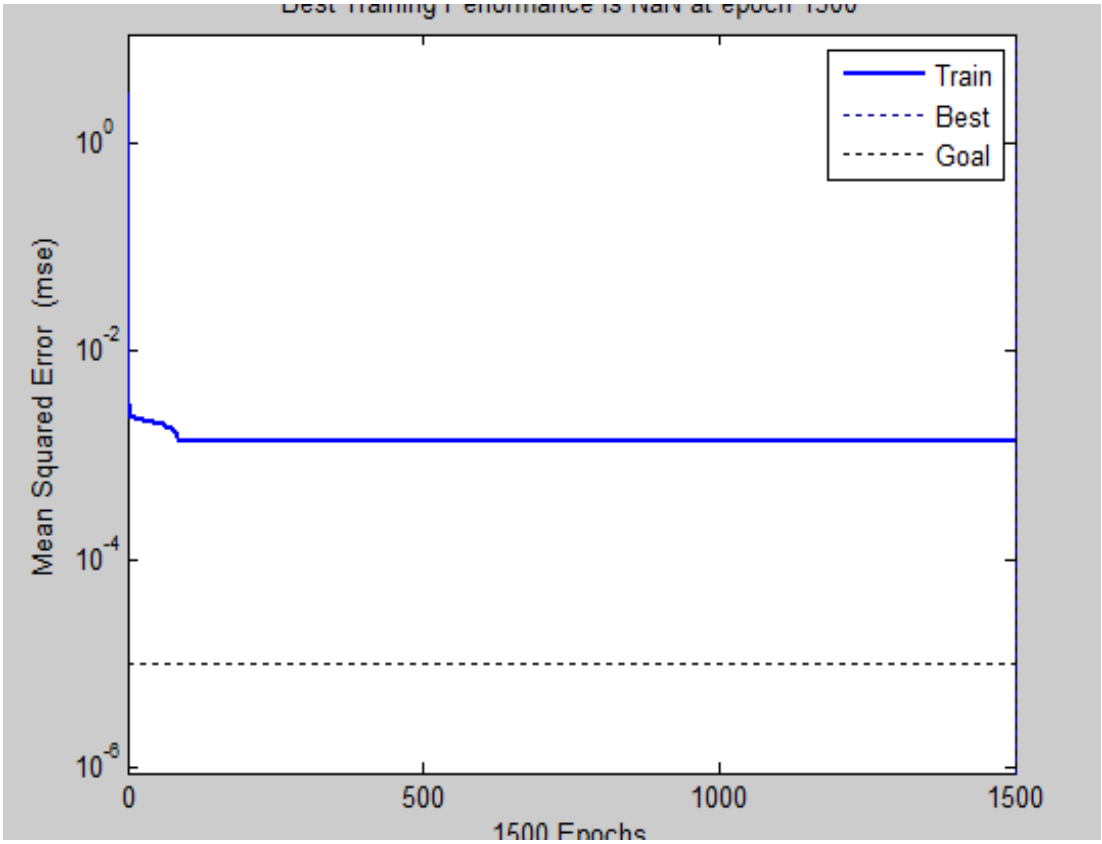

Best Training Performance is NaN at epoch 1500
Train
Best
Goal
Mean Squared Error (mse)
10^0
10^-2
10^-4
10^-6
0
500
1000
1500
1500 Epochs

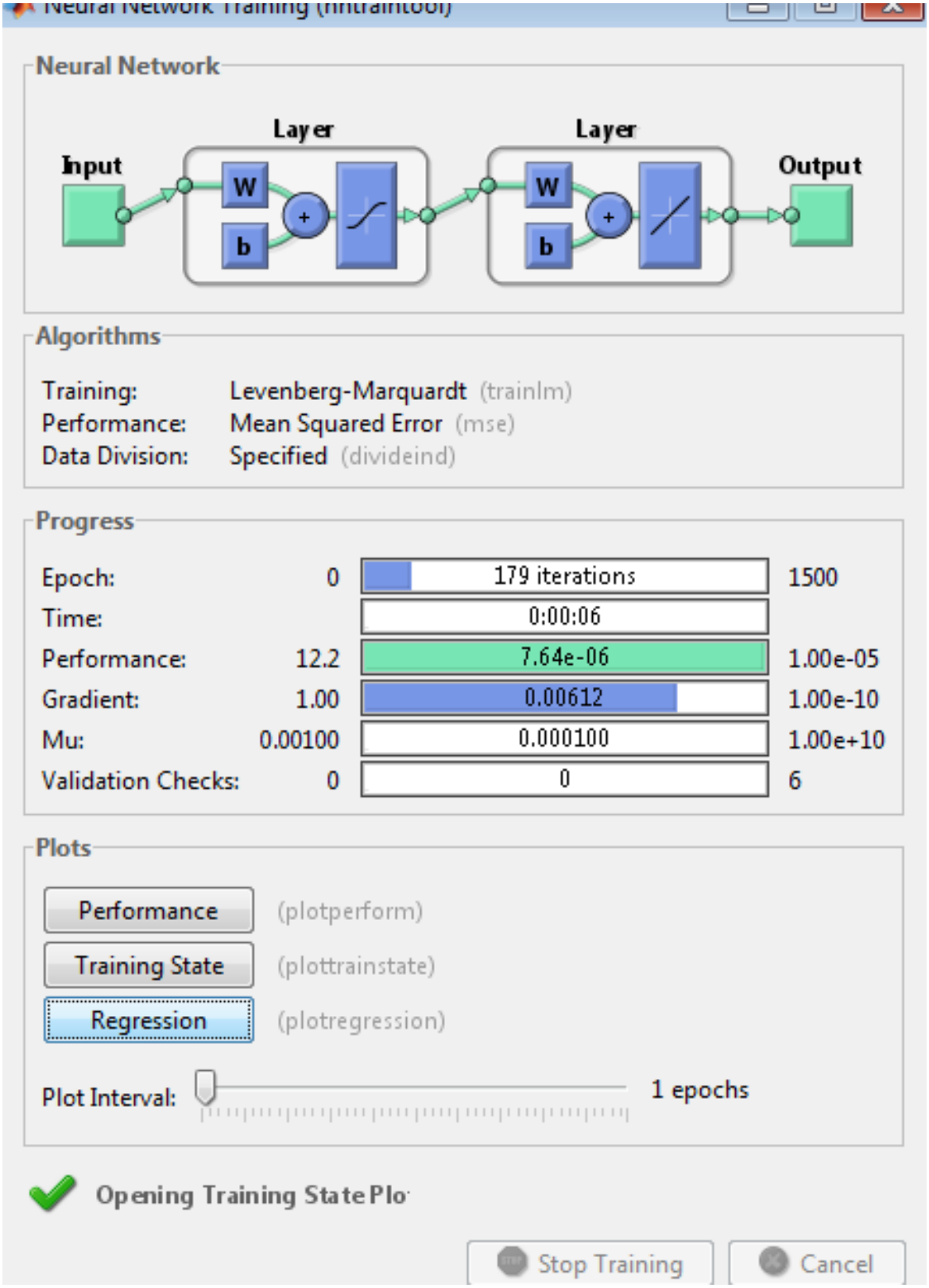


شكل (4-10C) يمثل تقييم الشبكة العصبية الاصطناعية التقليدية للمرحلة الثالثة

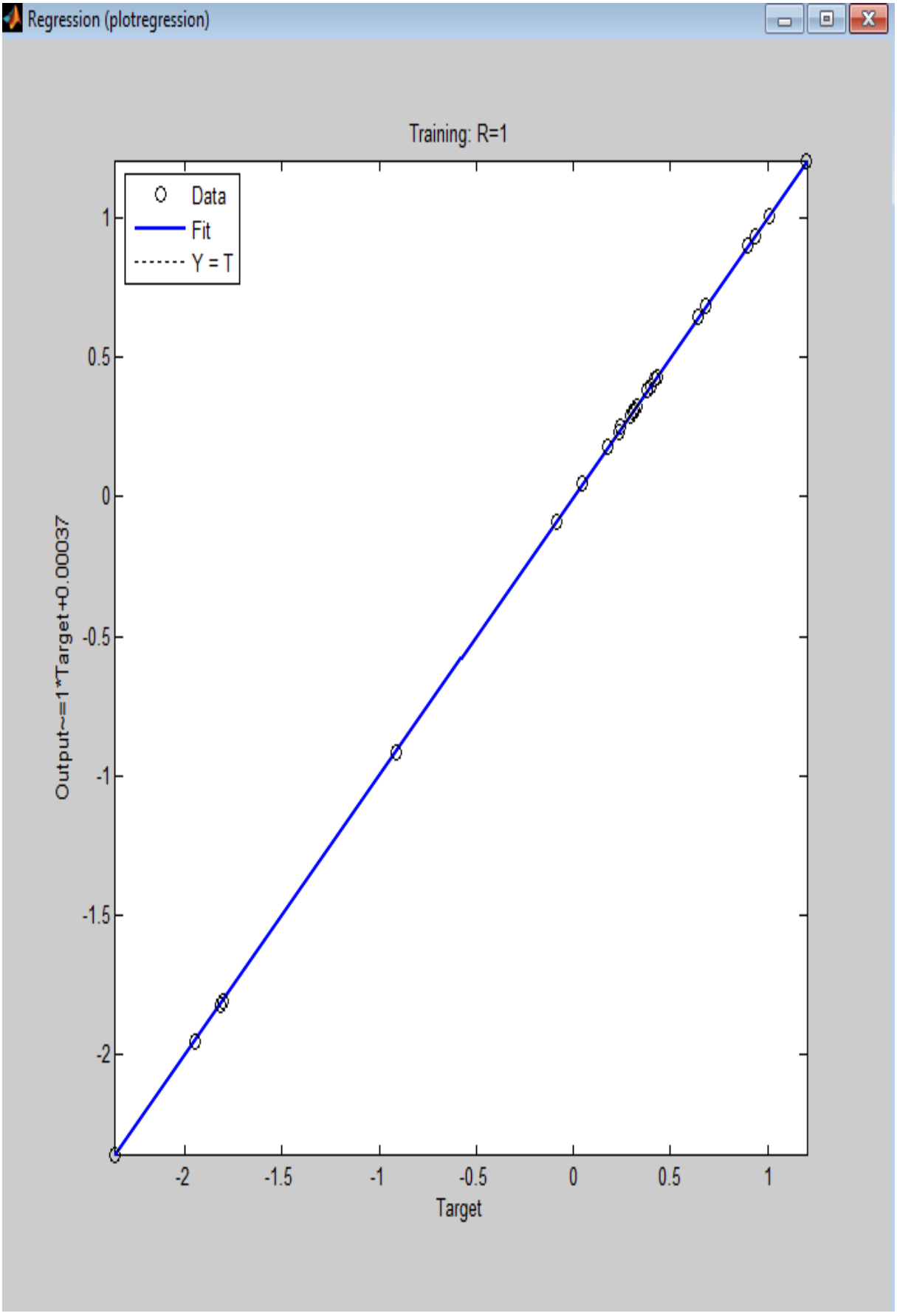

Training: R=0.99929
Data
Fit
Y = T
Output~=1*Target+-3.2e-006
Target
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1

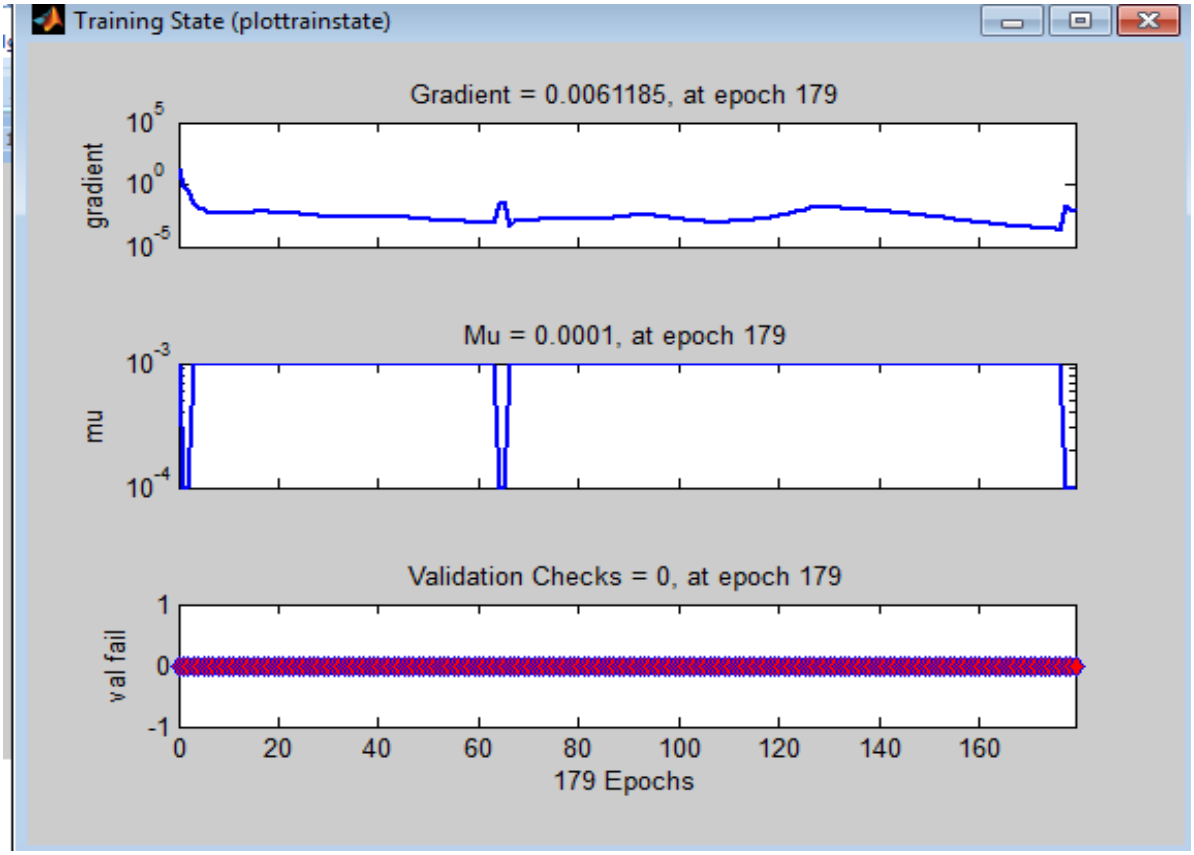

Gradient = 0.0001795, at epoch 1500
gradient
10^10
10^0
10^-10
Mu = 1e-007, at epoch 1500
mu
10^0
10^-5
10^-10
Validation Checks = 0, at epoch 1500
val fail
1
0
-1
0
500
1000
1500
1500 Epochs

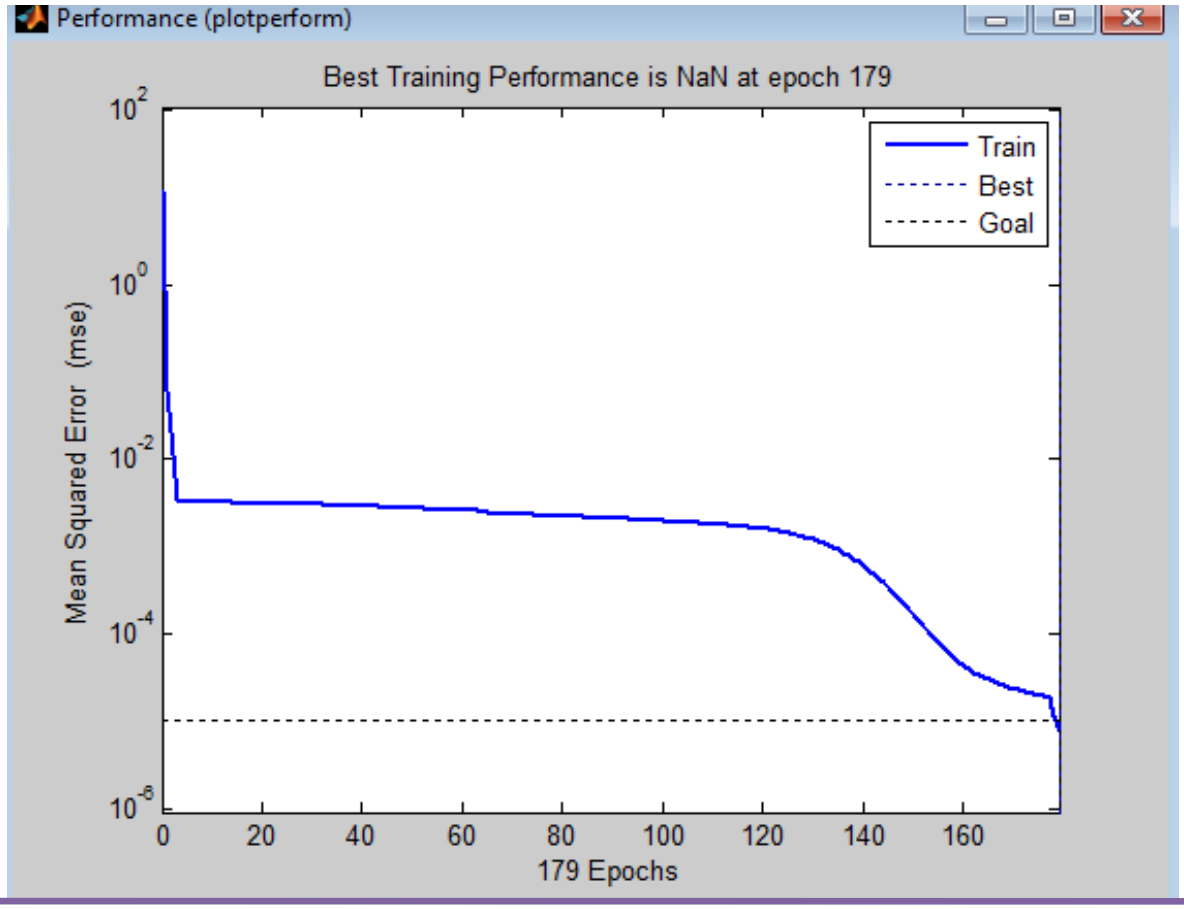

Train
Best
Goal
Mean Squared Error (mse)
10^{0}
10^{-2}
10^{-4}
10^{-6}
0
500
1000
1500
1500 Epochs

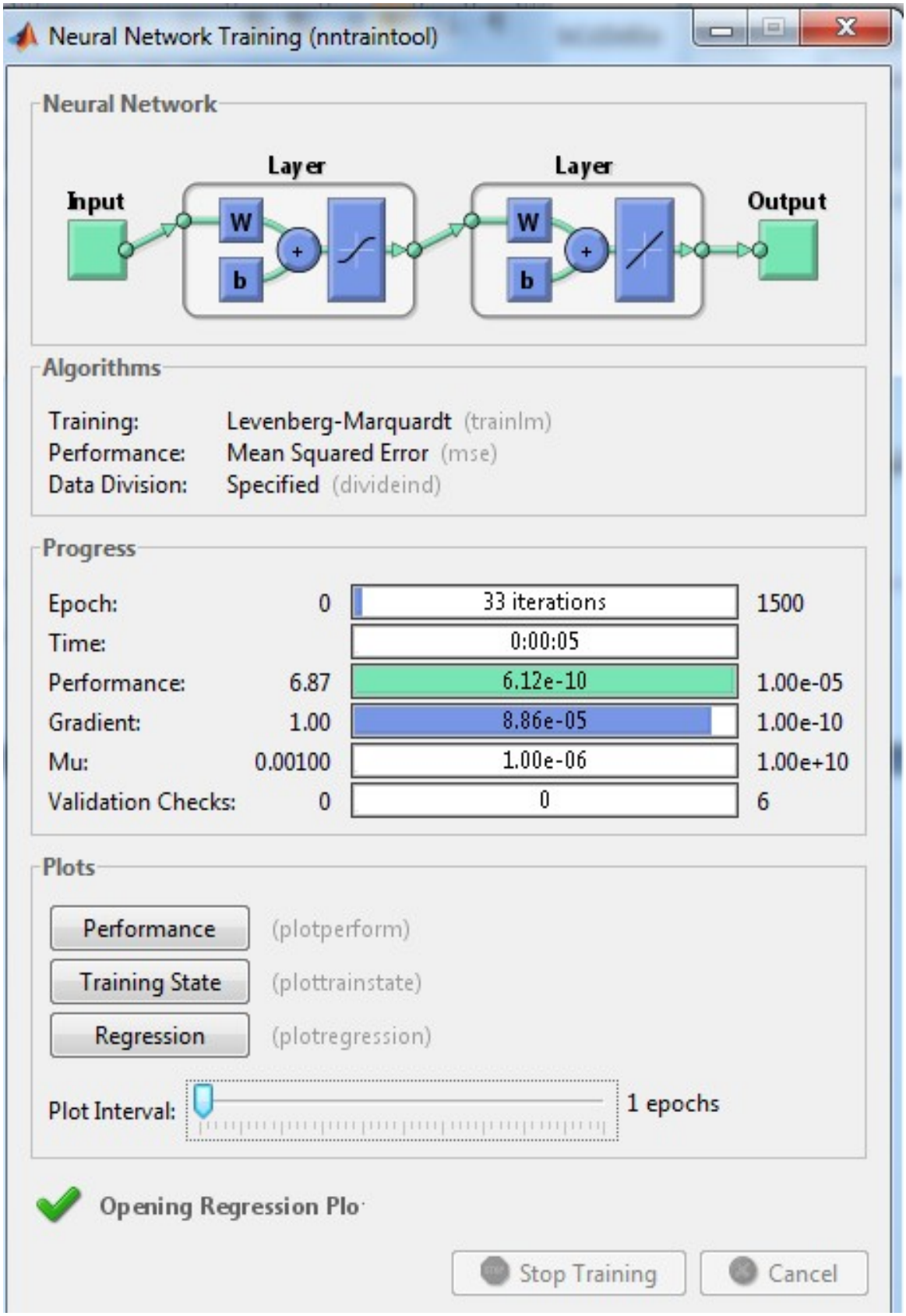


شكل (4-14C) يمثل تقييم الشبكة العصبية الاصطناعية المعدلة للمرحلة الاولى

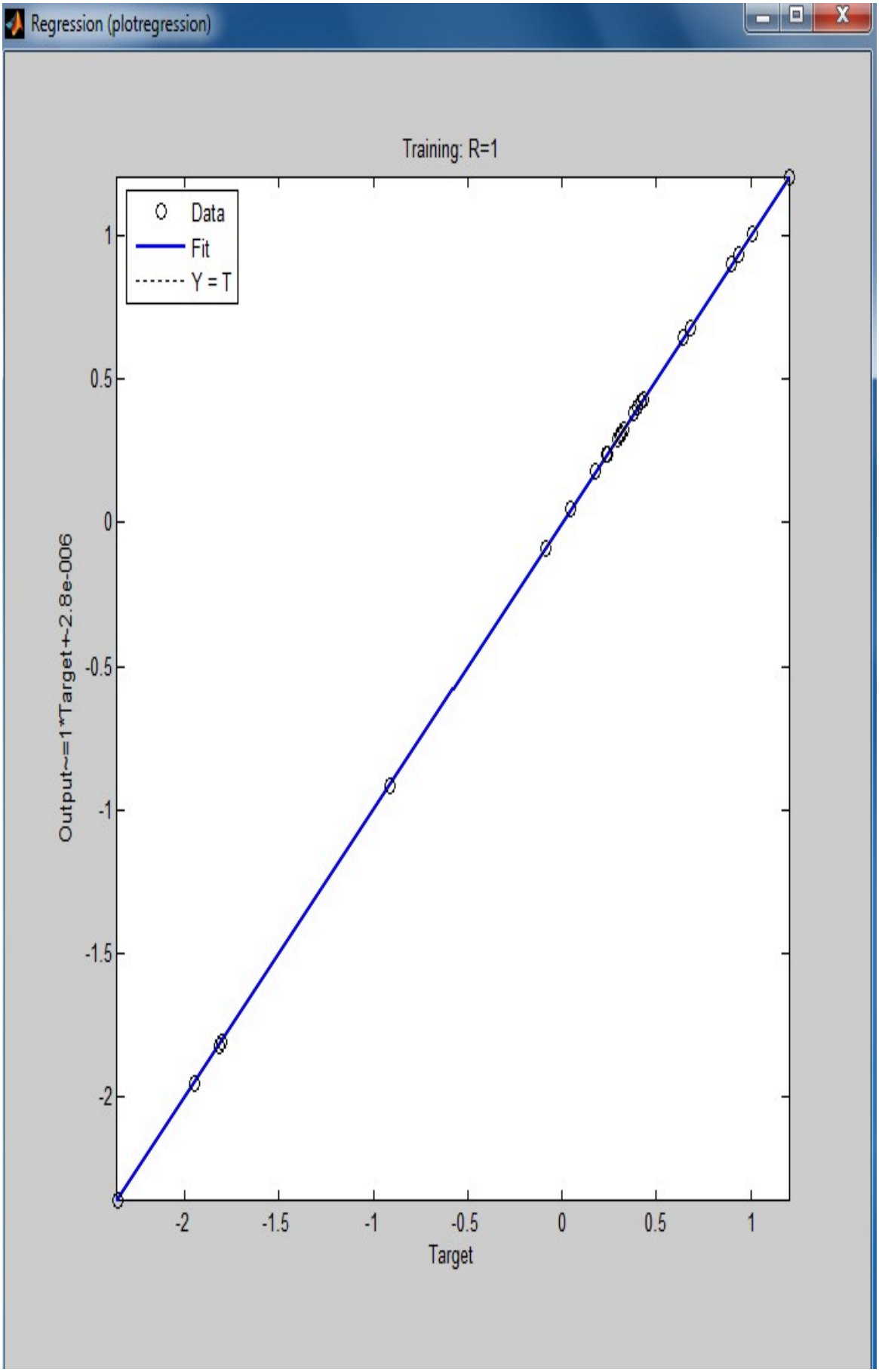

Regression (plotregression)
Training: R=1
Data
Fit
Y = T
Output~=1*Target+0.00037
Target
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1

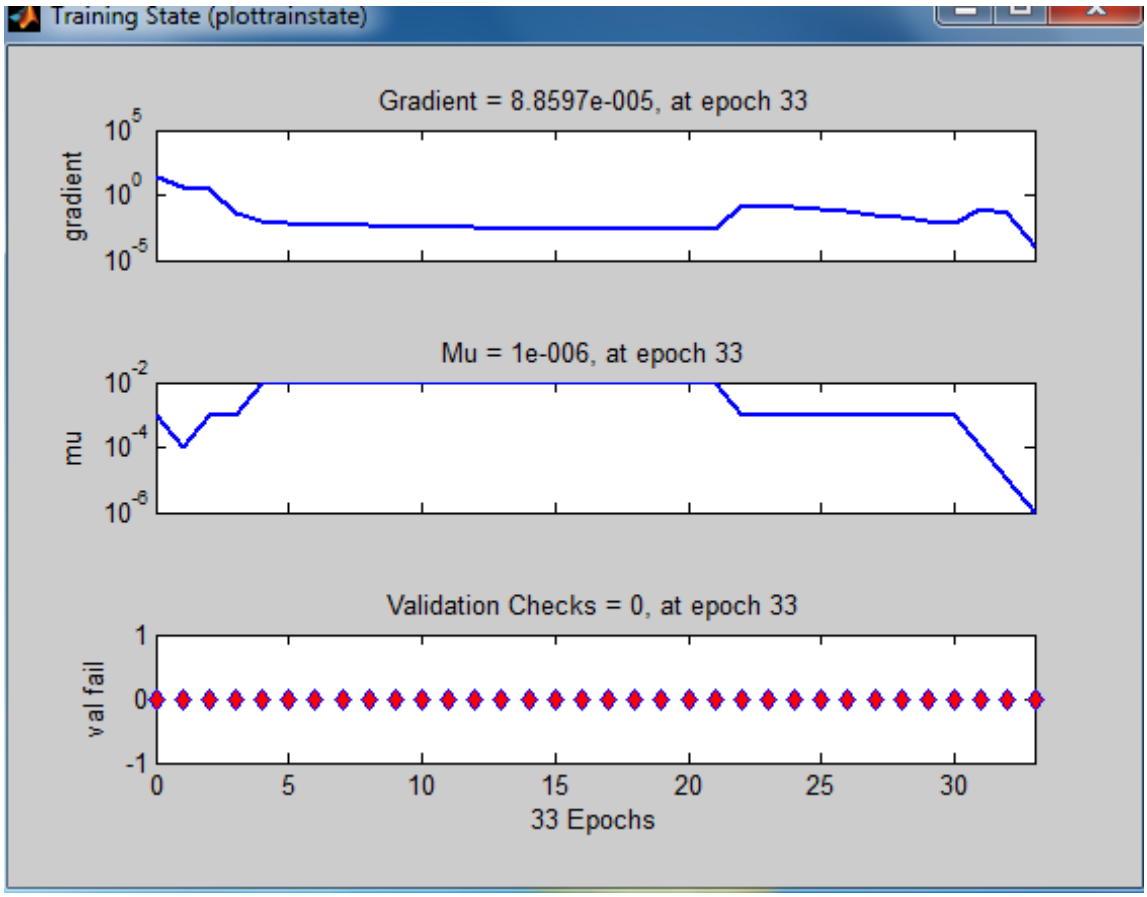

Training State (plottrainstate)
Gradient = 0.0061185, at epoch 179
gradient
Mu = 0.0001, at epoch 179
mu
Validation Checks = 0, at epoch 179
val fail
179 Epochs

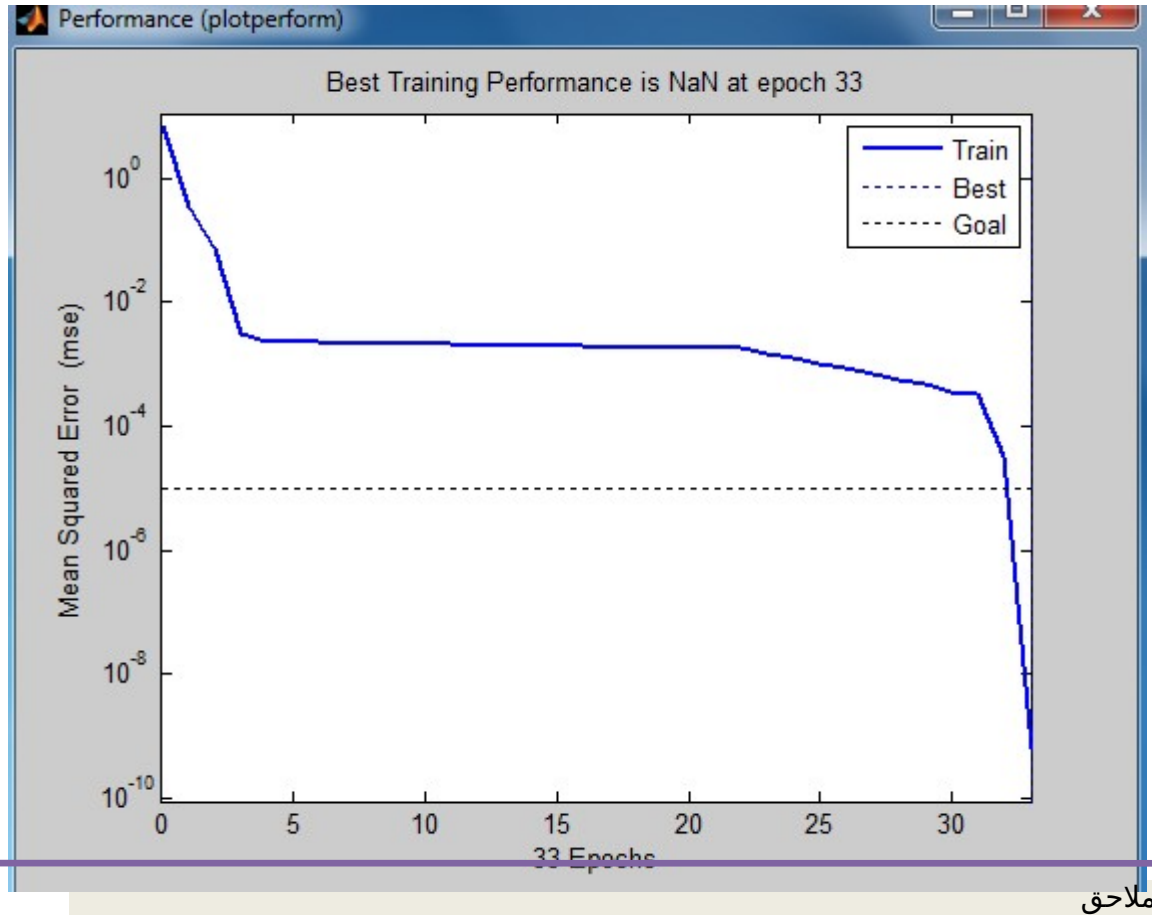

Performance (plotperform)
Best Training Performance is NaN at epoch 179
Train
Best
Goal
Mean Squared Error (mse)
179 Epochs

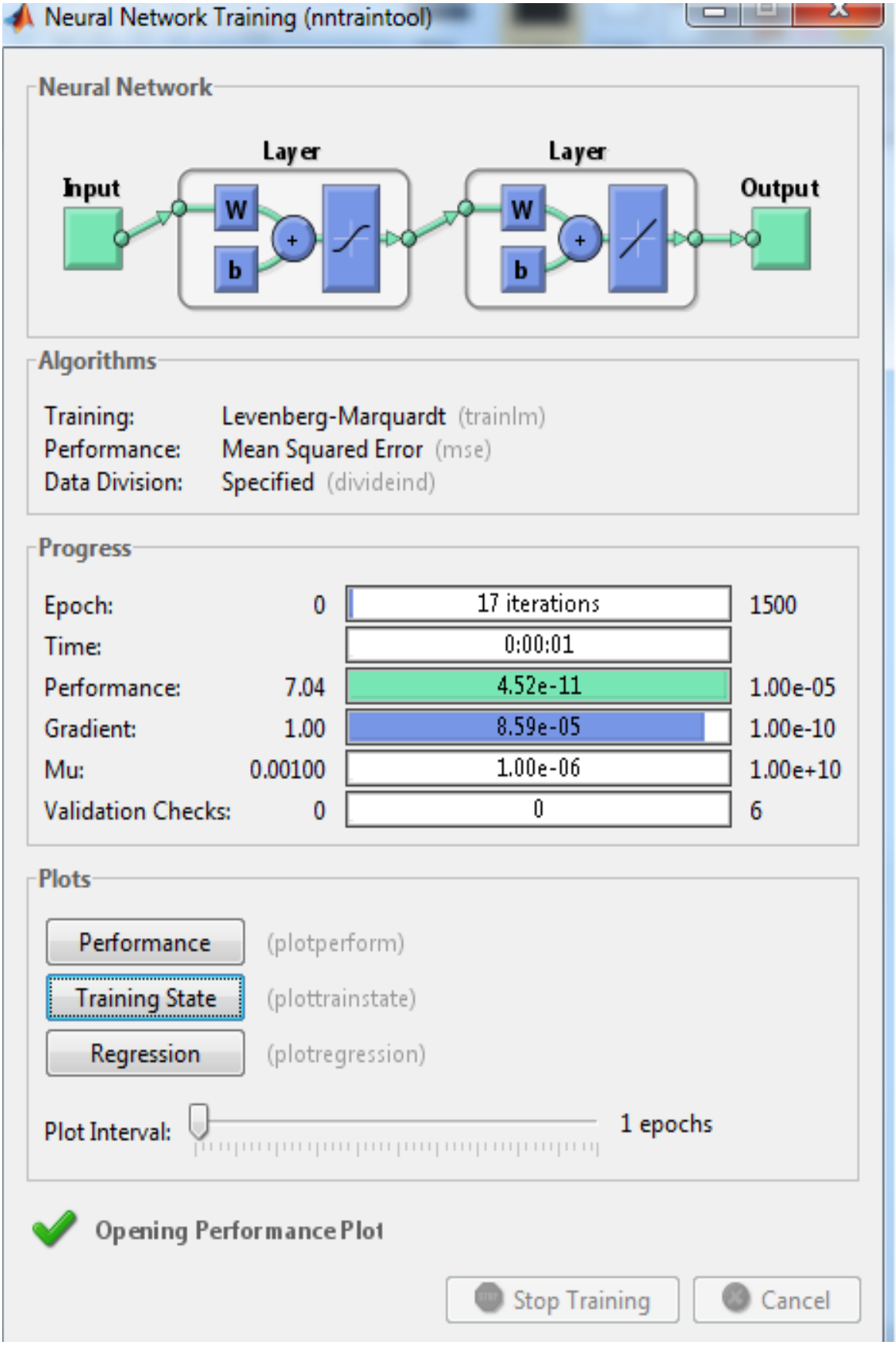


شكل (4-15C) يمثل تقييم الشبكة العصبية الاصطناعية المعدلة للمرحلة الثانية

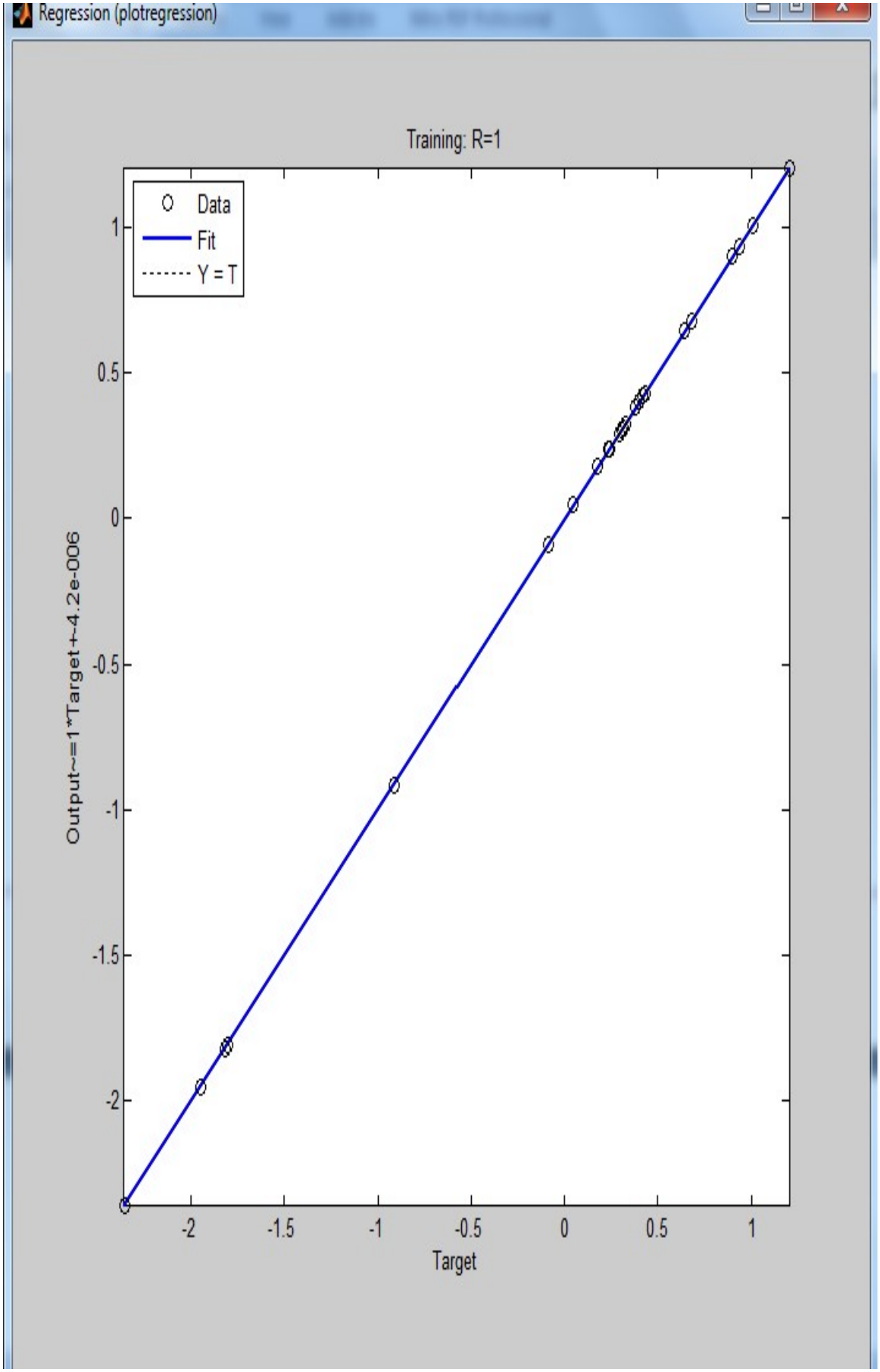

Regression (plotregression)
Training: R=1
Data
Fit
Y = T
Output~=1*Target+-2.8e-006
Target
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1

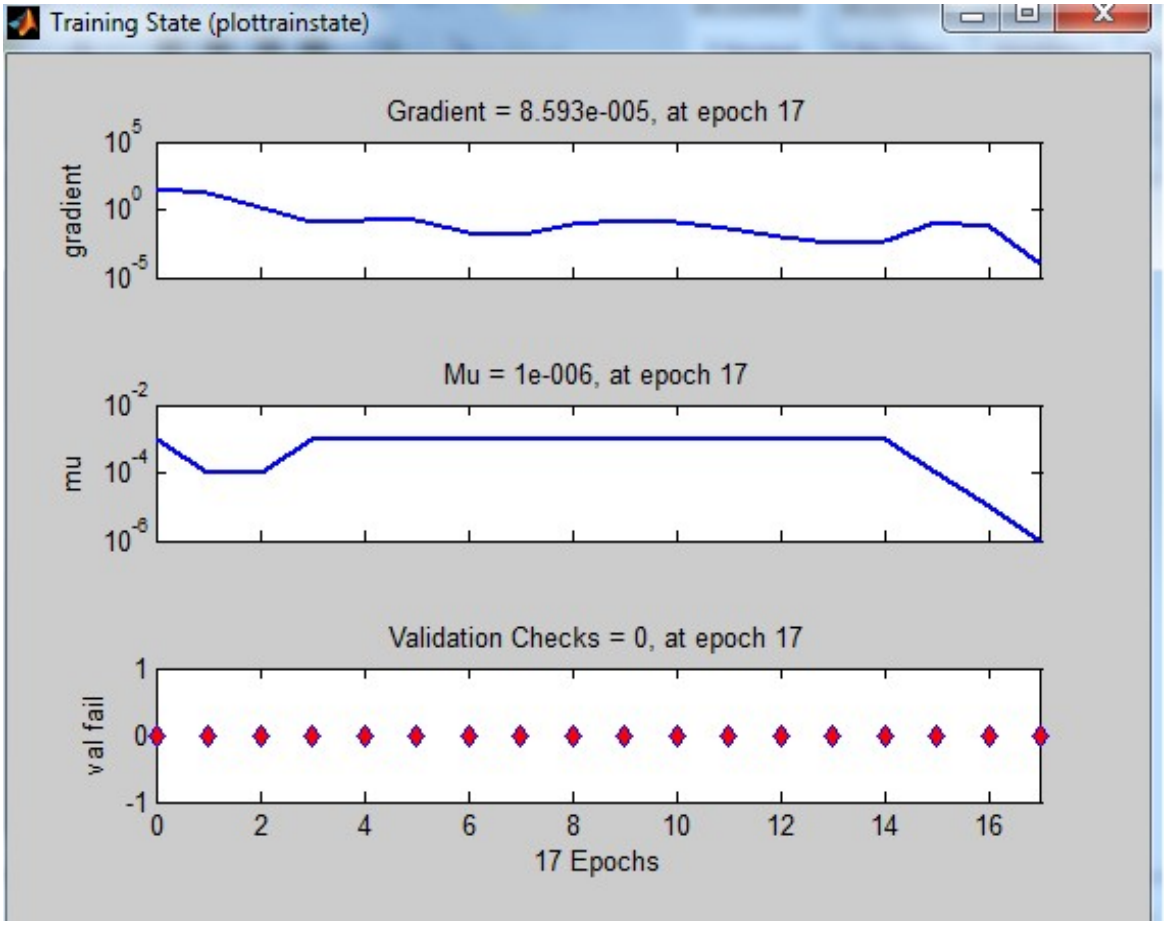

Training State (plottrainstate)
Gradient = 8.8597e-005, at epoch 33
gradient
Mu = 1e-006, at epoch 33
mu
Validation Checks = 0, at epoch 33
val fail
33 Epochs

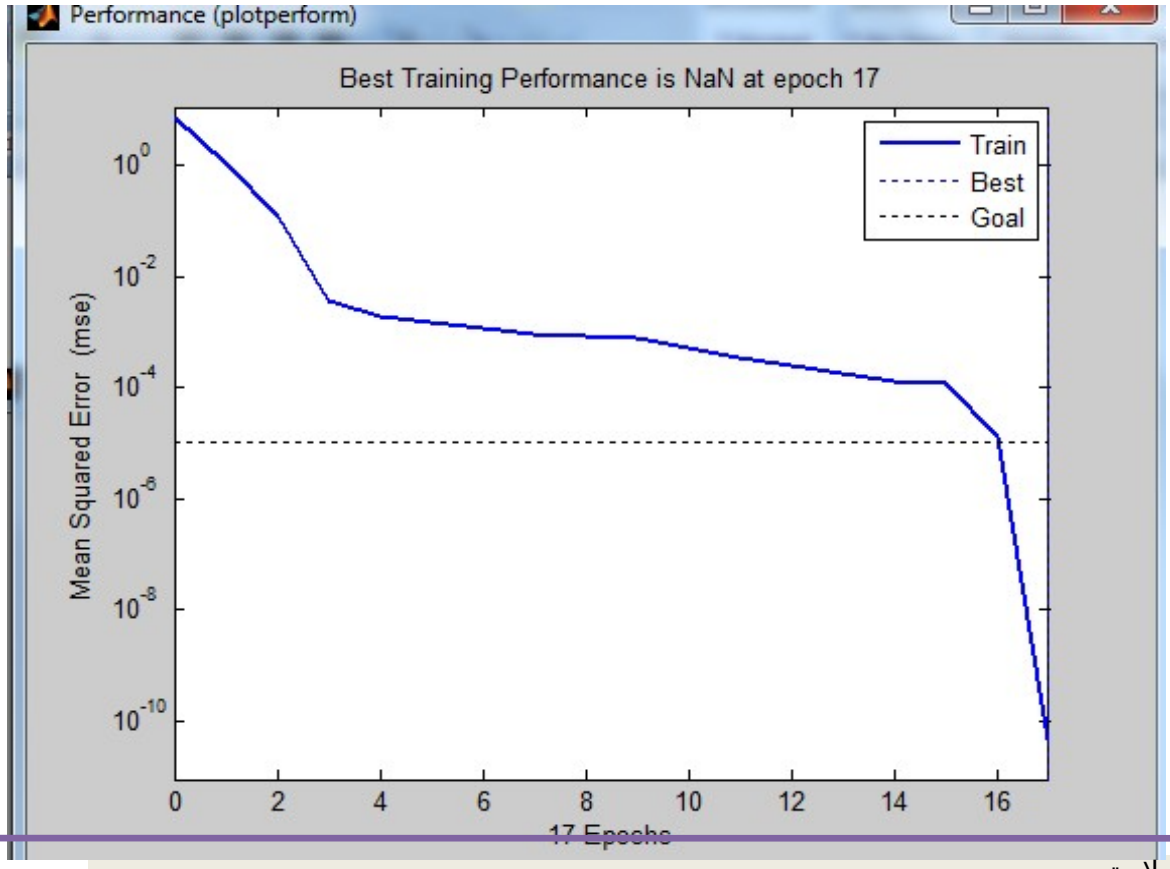

Performance (plotperform)
Best Training Performance is NaN at epoch 33
Train
Best
Goal
Mean Squared Error (mse)
33 Epochs



شكل (C16-4) يمثل تقييم الشبكة العصبية الاصطناعية المعدلة للمرحلة الثالثة


Regression (plotregression)
Training: R=1
Data
Fit
Y = T
Output~=1*Target+-4.2e-006
Target
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1


Training State (plottrainstate)
Gradient = 8.593e-005, at epoch 17
gradient
Mu = 1e-006, at epoch 17
mu
Validation Checks = 0, at epoch 17
val fail
17 Epochs


Performance (plotperform)
Best Training Performance is NaN at epoch 17
Train
Best
Goal
Mean Squared Error (mse)
17 Epochs

مراحل التدريب الشبكة العصبية الاصطناعية التقليدية للمرحلة الاولى

```
= net
    Neural Network object:
    architecture:
        numInputs: 1
        numLayers: 2
        biasConnect: [1; 1]
        inputConnect: [1; 0]
        layerConnect: [0 0; 1 0]
        outputConnect: [0 1]
        numOutputs: 1  (read-only)
        numInputDelays: 0  (read-only)
        numLayerDelays: 0  (read-only)
    subobject structures:
        inputs: {1x1 cell} of inputs
        layers: {2x1 cell} of layers
        outputs: {1x2 cell} containing 1 output
        biases: {2x1 cell} containing 2 biases
        inputWeights: {2x1 cell} containing 1 input weight
        layerWeights: {2x2 cell} containing 1 layer weight
    functions:
        adaptFcn: 'trains'
        divideFcn: (none)
```

```
                                                    'gradientFcn: 'gdefaults
                                                        'initFcn: 'initlay
                                                      'performFcn: 'mse
          {'plotFcns: {'plotperform','plottrainstate','plotregression
                                                      'trainFcn: 'trainlm
                                                             :parameters
                                                    adaptParam: .passes
                                                    (divideParam: (none
                                                    (gradientParam: (none
                                                     (initParam: (none
                                                   (performParam: (none
trainParam:  .show,  .showWindow,  .showCommandLine,
                                                                  ,.epochs
                                ,time, .goal, .max_fail, .mem_reduc.
                                ,min_grad, .mu, .mu_dec, .mu_inc.
                                                        mu_max.
                                                   :weight and bias values
                    IW: {2x1 cell} containing 1 input weight matrix
                   LW: {2x2 cell} containing 1 layer weight matrix
                            b: {2x1 cell} containing 2 bias vectors
                                                               :other
                                                          '' :name
                                                 (userdata: (user information
                                                                = net
```

```
Neural Network object:

architecture:

numInputs: 1

numLayers: 2

biasConnect: [1; 1]

inputConnect: [1; 0]

layerConnect: [0 0; 1 0]

outputConnect: [0 1]

numOutputs: 1  (read-only)

numInputDelays: 0  (read-only)

numLayerDelays: 0  (read-only)

subobject structures:

inputs: {1x1 cell} of inputs

layers: {2x1 cell} of layers

outputs: {1x2 cell} containing 1 output

biases: {2x1 cell} containing 2 biases

inputWeights: {2x1 cell} containing 1 input weight

layerWeights: {2x2 cell} containing 1 layer weight

functions:

adaptFcn: 'trains'

divideFcn: (none)

gradientFcn: 'gdefaults'

initFcn: 'initlay'

performFcn: 'mse'
```

{'plotFcns: {'plotperform','plottrainstate','plotregression

'trainFcn: 'trainlm

:parameters

adaptParam: .passes

(divideParam: (none

(gradientParam: (none

(initParam: (none

(performParam: (none

trainParam: .show, .showWindow, .showCommandLine,

,.epochs

,time, .goal, .max_fail, .mem_reduc

,min_grad, .mu, .mu_dec, .mu_inc

mu_max

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix

LW: {2x2 cell} containing 1 layer weight matrix

b: {2x1 cell} containing 2 bias vectors

:other

'' :name

(userdata: (user information

= net

:Neural Network object

:architecture

numInputs: 1

```
                numLayers: 2
            [biasConnect: [1; 1
          [inputConnect: [1; 0
        [layerConnect: [0 0; 1 0
          [outputConnect: [0 1
      (numOutputs: 1  (read-only
    (numInputDelays: 0  (read-only
    (numLayerDelays: 0  (read-only
                :subobject structures
          inputs: {1x1 cell} of inputs
          layers: {2x1 cell} of layers
       outputs: {1x2 cell} containing 1 output
         biases: {2x1 cell} containing 2 biases
  inputWeights: {2x1 cell} containing 1 input weight
  layerWeights: {2x2 cell} containing 1 layer weight
                    :functions
          'adaptFcn: 'trains

          (divideFcn: (none
        'gradientFcn: 'gdefaults
          'initFcn: 'initlay
          'performFcn: 'mse

{'plotFcns: {'plotperform','plottrainstate','plotregression
```

'trainFcn: 'trainlm

:parameters

adaptParam: .passes

(divideParam: (none

(gradientParam: (none

(initParam: (none

(performParam: (none

trainParam: .show, .showWindow, .showCommandLine,
,.epochs

,time, .goal, .max_fail, .mem_reduc.

,min_grad, .mu, .mu_dec, .mu_inc.

mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix

LW: {2x2 cell} containing 1 layer weight matrix

b: {2x1 cell} containing 2 bias vectors

```
:other

'' :name

)userdata: (user information(
```

جدول (C5-4) يمثل مراحل التدريب الشبكة العصبية الاصطناعية التقليدية للمرحلة الاولى

مراحل التدريب الشبكة العصبية الاصطناعية التقليدية للمرحلة 2

```
= net

:Neural Network object

:architecture

numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1  (read-only
(numInputDelays: 0  (read-only
(numLayerDelays: 0  (read-only

:subobject structures

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

:functions

'adaptFcn: 'trains
(divideFcn: (none
```

```
                                    'gradientFcn: 'gdefaults
                                         'initFcn: 'initlay
                                      'performFcn: 'mse
                                            plotFcns:
{'{'plotperform','plottrainstate','plotregression
                                        'trainFcn: 'trainlm

                                                :parameters

                                     adaptParam: .passes
                                      (divideParam: (none
                                    (gradientParam: (none
                                       (initParam: (none
                                    (performParam: (none
                          trainParam: .show, .showWindow,
                                  ,.showCommandLine, .epochs
                  ,time, .goal, .max_fail, .mem_reduc.
                    ,min_grad, .mu, .mu_dec, .mu_inc.
                                           mu_max.

                                                :weight and bias values

      IW: {2x1 cell} containing 1 input weight matrix
      LW: {2x2 cell} containing 1 layer weight matrix
               b: {2x1 cell} containing 2 bias vectors

                                                        :other

                                                    '' :name
                                  (userdata: (user information


                                                                = net

                                                :Neural Network object

                                                        :architecture

                                                  numInputs: 1
                                                 numLayers: 2
                                               [biasConnect: [1; 1
                                               [inputConnect: [1; 0
```

[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1 (read-only
(numInputDelays: 0 (read-only
(numLayerDelays: 0 (read-only

:subobject structures

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

:functions

'adaptFcn: 'trains
(divideFcn: (none
'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
plotFcns:
{'{'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters

adaptParam: .passes
(divideParam: (none
(gradientParam: (none
(initParam: (none
(performParam: (none
trainParam: .show, .showWindow,
,.showCommandLine, .epochs
,time, .goal, .max_fail, .mem_reduc.
,min_grad, .mu, .mu_dec, .mu_inc.
mu_max.

:weight and bias values

```
    IW: {2x1 cell} containing 1 input weight matrix
    LW: {2x2 cell} containing 1 layer weight matrix
             b: {2x1 cell} containing 2 bias vectors

                                                  :other

                                                   '' :name
                                  (userdata: (user information


                                                  = net

                                    :Neural Network object

                                          :architecture

                                              numInputs: 1
                                             numLayers: 2
                                    [biasConnect: [1; 1
                                    [inputConnect: [1; 0
                                  [layerConnect: [0 0; 1 0
                                    [outputConnect: [0 1

                                 (numOutputs: 1  (read-only
                                (numInputDelays: 0  (read-only
                                (numLayerDelays: 0  (read-only

                                       :subobject structures

                                  inputs: {1x1 cell} of inputs
                                  layers: {2x1 cell} of layers
                        outputs: {1x2 cell} containing 1 output
                         biases: {2x1 cell} containing 2 biases
                    inputWeights: {2x1 cell} containing 1 input weight
                    layerWeights: {2x2 cell} containing 1 layer weight

                                          :functions

                                   'adaptFcn: 'trains
                                    (divideFcn: (none
                                 'gradientFcn: 'gdefaults
```

'initFcn: 'initlay

'performFcn: 'mse

plotFcns:
{'{'plotperform','plottrainstate','plotregression

'trainFcn: 'trainlm

:parameters

adaptParam: .passes
(divideParam: (none
(gradientParam: (none
(initParam: (none
(performParam: (none
trainParam: .show, .showWindow,
,.showCommandLine, .epochs

,time, .goal, .max_fail, .mem_reduc.
,min_grad, .mu, .mu_dec, .mu_inc.
mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix

LW: {2x2 cell} containing 1 layer weight matrix

b: {2x1 cell} containing 2 bias vectors

:other

'' :name

(userdata: (user information

جدول (C7-4) يمثل مراحل التدريب الشبكة العصبية الاصطناعية التقليدية للمرحلة 2

مراحل التدريب الشبكة العصبية الاصطناعية التقليدية للمرحلة 3

```
= net

:Neural Network object

:architecture

numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1  (read-only
(numInputDelays: 0  (read-only
(numLayerDelays: 0  (read-only

:subobject structures

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

:functions

'adaptFcn: 'trains
(divideFcn: (none
'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
{'plotFcns: {'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm
```

:parameters

adaptParam: .passes
(divideParam: (none
(gradientParam: (none
(initParam: (none
(performParam: (none
,trainParam: .show, .showWindow, .showCommandLine, .epochs
,time, .goal, .max_fail, .mem_reduc.
,min_grad, .mu, .mu_dec, .mu_inc.
mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix
LW: {2x2 cell} containing 1 layer weight matrix
b: {2x1 cell} containing 2 bias vectors

:other

'' :name
(userdata: (user information

= net

:Neural Network object

:architecture

numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1 (read-only
(numInputDelays: 0 (read-only
(numLayerDelays: 0 (read-only

:subobject structures

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

:functions

'adaptFcn: 'trains
(divideFcn: (none
'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
{'plotFcns: {'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters

adaptParam: .passes
(divideParam: (none
(gradientParam: (none
(initParam: (none
(performParam: (none
,trainParam: .show, .showWindow, .showCommandLine, .epochs
,time, .goal, .max_fail, .mem_reduc.
,min_grad, .mu, .mu_dec, .mu_inc.
mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix
LW: {2x2 cell} containing 1 layer weight matrix
b: {2x1 cell} containing 2 bias vectors

:other

'' :name
(userdata: (user information

```
= net

:Neural Network object

:architecture

    numInputs: 1
  numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1  (read-only
(numInputDelays: 0  (read-only
(numLayerDelays: 0  (read-only

:subobject structures

  inputs: {1x1 cell} of inputs
  layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
  biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

:functions

'adaptFcn: 'trains
(divideFcn: (none
'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
{'plotFcns: {'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters

adaptParam: .passes
(divideParam: (none
(gradientParam: (none
(initParam: (none
```

(performParam: (none

,trainParam: .show, .showWindow, .showCommandLine, .epochs

,time, .goal, .max_fail, .mem_reduc.
,min_grad, .mu, .mu_dec, .mu_inc.
mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix
LW: {2x2 cell} containing 1 layer weight matrix
b: {2x1 cell} containing 2 bias vectors

:other

" :name
(userdata: (user information

جدول (4-C9) يمثل مراحل التدريب الشبكة العصبية الاصطناعية التقليدية للمرحلة 3

مراحل التدريب الشبكة العصبية الاصطناعية المعدلة للمرحلة الاولى

= net

:Neural Network object

:architecture

numInputs: 1

numLayers: 2

[biasConnect: [1; 1

[inputConnect: [1; 0

[layerConnect: [0 0; 1 0

[outputConnect: [0 1

(numOutputs: 1 (read-only

(numInputDelays: 0 (read-only

(numLayerDelays: 0 (read-only

:subobject structures

inputs: {1x1 cell} of inputs

layers: {2x1 cell} of layers

outputs: {1x2 cell} containing 1 output

biases: {2x1 cell} containing 2 biases

inputWeights: {2x1 cell} containing 1 input weight

layerWeights: {2x2 cell} containing 1 layer weight

:functions

'adaptFcn: 'trains

(divideFcn: (none

'gradientFcn: 'gdefaults

'initFcn: 'initlay

'performFcn: 'mse

{'plotFcns: {'plotperform','plottrainstate','plotregression

'trainFcn: 'trainlm

:parameters

adaptParam: .passes

(divideParam: (none

(gradientParam: (none

(initParam: (none

(performParam: (none

,trainParam: .show, .showWindow, .showCommandLine, .epochs

,time, .goal, .max_fail, .mem_reduc.

,min_grad, .mu, .mu_dec, .mu_inc.

mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix

LW: {2x2 cell} containing 1 layer weight matrix

b: {2x1 cell} containing 2 bias vectors

:other

'' :name

(userdata: (user information

= net

:Neural Network object

:architecture

numInputs: 1

numLayers: 2

[biasConnect: [1; 1

[inputConnect: [1; 0

[layerConnect: [0 0; 1 0

[outputConnect: [0 1

(numOutputs: 1 (read-only

(numInputDelays: 0 (read-only

(numLayerDelays: 0 (read-only

```
:subobject structures
inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight
:functions
'adaptFcn: 'trains
(divideFcn: (none
'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
{'plotFcns: {'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm
:parameters
adaptParam: .passes
(divideParam: (none
(gradientParam: (none
(initParam: (none
(performParam: (none
 ,trainParam: .show, .showWindow, .showCommandLine, .epochs
 ,time, .goal, .max_fail, .mem_reduc.
 ,min_grad, .mu, .mu_dec, .mu_inc.
```

mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix

LW: {2x2 cell} containing 1 layer weight matrix

b: {2x1 cell} containing 2 bias vectors

:other

'' :name

(userdata: (user information

= net

:Neural Network object

:architecture

numInputs: 1

numLayers: 2

[biasConnect: [1; 1

[inputConnect: [1; 0

[layerConnect: [0 0; 1 0

[outputConnect: [0 1

(numOutputs: 1 (read-only

(numInputDelays: 0 (read-only

(numLayerDelays: 0 (read-only

:subobject structures

inputs: {1x1 cell} of inputs

layers: {2x1 cell} of layers

outputs: {1x2 cell} containing 1 output

biases: {2x1 cell} containing 2 biases

inputWeights: {2x1 cell} containing 1 input weight

layerWeights: {2x2 cell} containing 1 layer weight

:functions

'adaptFcn: 'trains

(divideFcn: (none

'gradientFcn: 'gdefaults

'initFcn: 'initlay

'performFcn: 'mse

{'plotFcns: {'plotperform','plottrainstate','plotregression

'trainFcn: 'trainlm

:parameters

adaptParam: .passes

(divideParam: (none

(gradientParam: (none

(initParam: (none

(performParam: (none

,trainParam: .show, .showWindow, .showCommandLine, .epochs

,time, .goal, .max_fail, .mem_reduc

,min_grad, .mu, .mu_dec, .mu_inc

mu_max

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix

LW: {2x2 cell} containing 1 layer weight matrix

b: {2x1 cell} containing 2 bias vectors

:other

'' :name

)userdata: (user information

جدول (C12-4) يمثل مراحل التدريب الشبكة العصبية الاصطناعية المعدلة للمرحلة 1

مراحل التدريب الشبكة العصبية الاصطناعية المعدلة للمرحلة 2

```
= net

:Neural Network object

:architecture

numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1  (read-only
(numInputDelays: 0  (read-only
(numLayerDelays: 0  (read-only

:subobject structures

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

:functions

'adaptFcn: 'trains
(divideFcn: (none
'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
```

{'plotFcns: {'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters

adaptParam: .passes
(divideParam: (none
(gradientParam: (none
(initParam: (none
(performParam: (none
trainParam: .show, .showWindow, .showCommandLine,
,.epochs
,time, .goal, .max_fail, .mem_reduc.
,min_grad, .mu, .mu_dec, .mu_inc.
mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix
LW: {2x2 cell} containing 1 layer weight matrix
b: {2x1 cell} containing 2 bias vectors

:other

'' :name
(userdata: (user information

= net

:Neural Network object

:architecture

numInputs: 1

numLayers: 2

[biasConnect: [1; 1

[inputConnect: [1; 0

[layerConnect: [0 0; 1 0

[outputConnect: [0 1

(numOutputs: 1 (read-only

(numInputDelays: 0 (read-only

(numLayerDelays: 0 (read-only

:subobject structures

inputs: {1x1 cell} of inputs

layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

:functions

'adaptFcn: 'trains

(divideFcn: (none

'gradientFcn: 'gdefaults

'initFcn: 'initlay

'performFcn: 'mse

{'plotFcns: {'plotperform','plottrainstate','plotregression

'trainFcn: 'trainlm

:parameters

adaptParam: .passes

(divideParam: (none

(gradientParam: (none

(initParam: (none

(performParam: (none

trainParam: .show, .showWindow, .showCommandLine,
,.epochs

,time, .goal, .max_fail, .mem_reduc.

,min_grad, .mu, .mu_dec, .mu_inc.

mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix

LW: {2x2 cell} containing 1 layer weight matrix

b: {2x1 cell} containing 2 bias vectors

:other

'' :name
(userdata: (user information

جدول (C13-4) يمثل مراحل التدريب الشبكة العصبية الاصطناعية المعدلة للمرحلة 2

مراحل التدريب الشبكة العصبية الاصطناعية المعدلة للمرحلة 3

```
= net

:Neural Network object

:architecture

numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1  (read-only
(numInputDelays: 0  (read-only
(numLayerDelays: 0  (read-only

:subobject structures

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

:functions

'adaptFcn: 'trains
(divideFcn: (none
'gradientFcn: 'gdefaults
'initFcn: 'initlay
'performFcn: 'mse
plotFcns:
{'{'plotperform','plottrainstate','plotregression
'trainFcn: 'trainlm

:parameters

adaptParam: .passes
(divideParam: (none
(gradientParam: (none
```

(initParam: (none
(performParam: (none
trainParam: .show, .showWindow,
,.showCommandLine, .epochs
,time, .goal, .max_fail, .mem_reduc.
,min_grad, .mu, .mu_dec, .mu_inc.
mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix
LW: {2x2 cell} containing 1 layer weight matrix
b: {2x1 cell} containing 2 bias vectors

:other

'' :name
(userdata: (user information

= net

:Neural Network object

:architecture

numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1

(numOutputs: 1 (read-only
(numInputDelays: 0 (read-only
(numLayerDelays: 0 (read-only

:subobject structures

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output

```
    biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

                                                :functions

                                          'adaptFcn: 'trains
                                           (divideFcn: (none
                                 'gradientFcn: 'gdefaults
                                       'initFcn: 'initlay
                                    'performFcn: 'mse
                                                plotFcns:
{'{'plotperform','plottrainstate','plotregression
                                       'trainFcn: 'trainlm

                                                :parameters

                                        adaptParam: .passes
                                         (divideParam: (none
                                       (gradientParam: (none
                                          (initParam: (none
                                       (performParam: (none
                          trainParam: .show, .showWindow,
                                   ,.showCommandLine, .epochs
                ,time, .goal, .max_fail, .mem_reduc.
                  ,min_grad, .mu, .mu_dec, .mu_inc.
                                            mu_max.

                                        :weight and bias values

 IW: {2x1 cell} containing 1 input weight matrix
LW: {2x2 cell} containing 1 layer weight matrix
          b: {2x1 cell} containing 2 bias vectors

                                                    :other

                                                 '' :name
                                  (userdata: (user information



                                                = net
```

```
                                                    :Neural Network object

                                                              :architecture

                                                                  numInputs: 1
                                                                 numLayers: 2
                                                          [biasConnect: [1; 1
                                                         [inputConnect: [1; 0
                                                    [layerConnect: [0 0; 1 0
                                                        [outputConnect: [0 1

                                                     (numOutputs: 1  (read-only
                                                   (numInputDelays: 0  (read-only
                                                   (numLayerDelays: 0  (read-only

                                                              :subobject structures

                                                          inputs: {1x1 cell} of inputs
                                                         layers: {2x1 cell} of layers
                                                outputs: {1x2 cell} containing 1 output
                                                 biases: {2x1 cell} containing 2 biases
                                    inputWeights: {2x1 cell} containing 1 input weight
                                    layerWeights: {2x2 cell} containing 1 layer weight

                                                                    :functions

                                                                'adaptFcn: 'trains
                                                                 (divideFcn: (none
                                                          'gradientFcn: 'gdefaults
                                                               'initFcn: 'initlay
                                                               'performFcn: 'mse
                                                                     plotFcns:
                                    {'{'plotperform','plottrainstate','plotregression

                                                                'trainFcn: 'trainlm

                                                                    :parameters

                                                              adaptParam: .passes

                                                               (divideParam: (none
```

(gradientParam: (none

(initParam: (none

(performParam: (none

trainParam: .show, .showWindow,
,.showCommandLine, .epochs

,time, .goal, .max_fail, .mem_reduc.

,min_grad, .mu, .mu_dec, .mu_inc.

mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix

LW: {2x2 cell} containing 1 layer weight matrix

b: {2x1 cell} containing 2 bias vectors

:other

'' :name

(userdata: (user information

جدول (4-C15) يمثل مراحل التدريب الشبكة العصبية الاصطناعية المعدلة للمرحلة 3

مصفوفة الاوزان المثلى لطبقة المدخلات-المخفية لنموذج الشبكة العصبية الاصطناعية الامثل

```
{w1=net.iw{1,1
= w1
  3.0284-   1.4177
  2.5515-   0.9838
  3.6646    8.9819-
  1.0219    4.2928
  6.4330    3.3463
  8.4249   10.6813-
  3.2974-   1.2916-
  5.4751    1.4805-
  1.9625    2.8464
  2.8222    1.7705
  0.1749-   1.9583-
  1.4330    2.5915
  0.3726    3.5912
  2.8137    0.2258-
  1.6808-   2.0563
{b1=net.b{1
= b1
  5.0197-
  2.6078-
  4.1707
  3.4822-
  5.9011-
```

```
0.7483
1.1661-
0.4521-
2.3050
3.8107
4.4965-
5.2745
7.4577
3.9941-
6.5992
```

جدول(C16-4) يمثل مصفوفة الاوزان المثلى لطبقـة المـدخلات-المخفيـة لنموذج الشـبكة العصـبية الاصـطناعية الامثـل لسلسـلة صـادرات النفـط الخام للمملكة العربية السعودية.

مصفوفة الاوزان المثلى لطبقـة المخفيـة-المخرجـات لنمـوذج الشبكة العصبية الاصطناعية

```
{w2=net.lw{2,1}

= w2

Columns 1 through 10

0.8295-   3.0692   5.8738-   2.4202-   1.1462-   1.8855

2.9110-   0.1083   1.9801   0.9683-

Columns 11 through 15
```

```
    1.3620-   3.2723-   1.2809    1.0933    2.1086

{b2=net.b{2

= b2

    0.7908-
```

جدول(C17-4) يمثل مصفوفة الاوزان المثلى لطبقة المخفيـة-المخرجـات لنمـوذج الشـبكة العصـبية الاصـطناعية لسلسـلة صـادرات النفـط الخـام للمملكة العربية السعودية.

الشبكة المدربة لاسلوب المعدل للشبكات العصبية الاصطناعية

```
(disp(net

:Neural Network object

:architecture

numInputs: 1

numLayers: 2

[biasConnect: [1; 1

[inputConnect: [1; 0

[layerConnect: [0 0; 1 0

[outputConnect: [0 1
```

(numOutputs: 1 (read-only

(numInputDelays: 0 (read-only

(numLayerDelays: 0 (read-only

:subobject structures

inputs: {1x1 cell} of inputs

layers: {2x1 cell} of layers

outputs: {1x2 cell} containing 1 output

biases: {2x1 cell} containing 2 biases

inputWeights: {2x1 cell} containing 1 input weight

layerWeights: {2x2 cell} containing 1 layer weight

:functions

'adaptFcn: 'trains

(divideFcn: (none

'gradientFcn: 'gdefaults

'initFcn: 'initlay

'performFcn: 'mse

{'plotFcns: {'plotperform','plottrainstate','plotregression

'trainFcn: 'trainlm

:parameters

adaptParam: .passes

(divideParam: (none

(gradientParam: (none

(initParam: (none

(performParam: (none

trainParam: .show, .showWindow, .showCommandLine,

,.epochs

,time, .goal, .max_fail, .mem_reduc.

,min_grad, .mu, .mu_dec, .mu_inc.

mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix

LW: {2x2 cell} containing 1 layer weight matrix

```
b: {2x1 cell} containing 2 bias vectors

:other

'' :name

(userdata: (user information
```

جدول(4-C18) يمثل الشبكة المدربة لشبكة العصبية الاصطناعية المعدلة لسلسلة صادرات النفط الخام للمملكة العربية السعودية.

الشبكة المدربة لاسلوب التقليدي للشبكات العصبية الاصطناعية

```
(disp(net
:Neural Network object
:architecture
numInputs: 1
numLayers: 2
[biasConnect: [1; 1
[inputConnect: [1; 0
[layerConnect: [0 0; 1 0
[outputConnect: [0 1
(numOutputs: 1  (read-only
(numInputDelays: 0  (read-only
(numLayerDelays: 0  (read-only
```

:subobject structures

inputs: {1x1 cell} of inputs

layers: {2x1 cell} of layers

outputs: {1x2 cell} containing 1 output

biases: {2x1 cell} containing 2 biases

inputWeights: {2x1 cell} containing 1 input weight

layerWeights: {2x2 cell} containing 1 layer weight

:functions

'adaptFcn: 'trains

(divideFcn: (none

'gradientFcn: 'gdefaults

'initFcn: 'initlay

'performFcn: 'mse

{'plotFcns: {'plotperform','plottrainstate','plotregression

'trainFcn: 'trainlm

:parameters

adaptParam: .passes

(divideParam: (none

(gradientParam: (none

(initParam: (none

(performParam: (none

trainParam: .show, .showWindow, .showCommandLine,
,.epochs

,time, .goal, .max_fail, .mem_reduc.

,min_grad, .mu, .mu_dec, .mu_inc.

mu_max.

:weight and bias values

IW: {2x1 cell} containing 1 input weight matrix

LW: {2x2 cell} containing 1 layer weight matrix

b: {2x1 cell} containing 2 bias vectors

:other

'' :name

(userdata: (user information

جدول(4-C19) يمثل الشبكة المدربة لشبكة العصبية الاصطناعية التقليدية لسلسلة صادرات النفط الخام للمملكة العربية السعودية.