

Lecture 7

Νευρωνικά Δίκτυα

(Artificial Neural Networks)

Ενότητες που θα καλύψουμε

- Perceptrons
- Πολυστρωματικά νευρωνικά δίκτυα (multilayer neural networks) και πως λειτουργούν
 - Ο αλγόριθμος Backpropagation
 - Gradient Descent
- Πως αναπτύσσουμε σε python
 - perceptron classifier
 - single layer neural network
 - multilayer neural network
 - σύστημα οπτικής αναγνώρισης χαρακτήρων (OCR engine)

Εισαγωγή στα νευρωνικά δίκτυα

- Ένα τεχνητό νευρωνικό δίκτυο (artificial neural network ή ANN) είναι ένα μοντέλο σχεδιασμένο να μιμείται τη διαδικασία μάθησης του ανθρώπινου εγκεφάλου.
- Τα ANN μπορούν να αναγνωρίσουν μπορεί να εντοπίσει τα υποκείμενα μοτίβα στα δεδομένα και να μάθει από αυτά.
- Χρησιμοποιούνται για
 - Ταξινόμηση (classification)
 - Παλινδρόμιση (regression)
 - Διαχωρισμό (segmentation) ή ομαδοποίηση
 - και άλλα

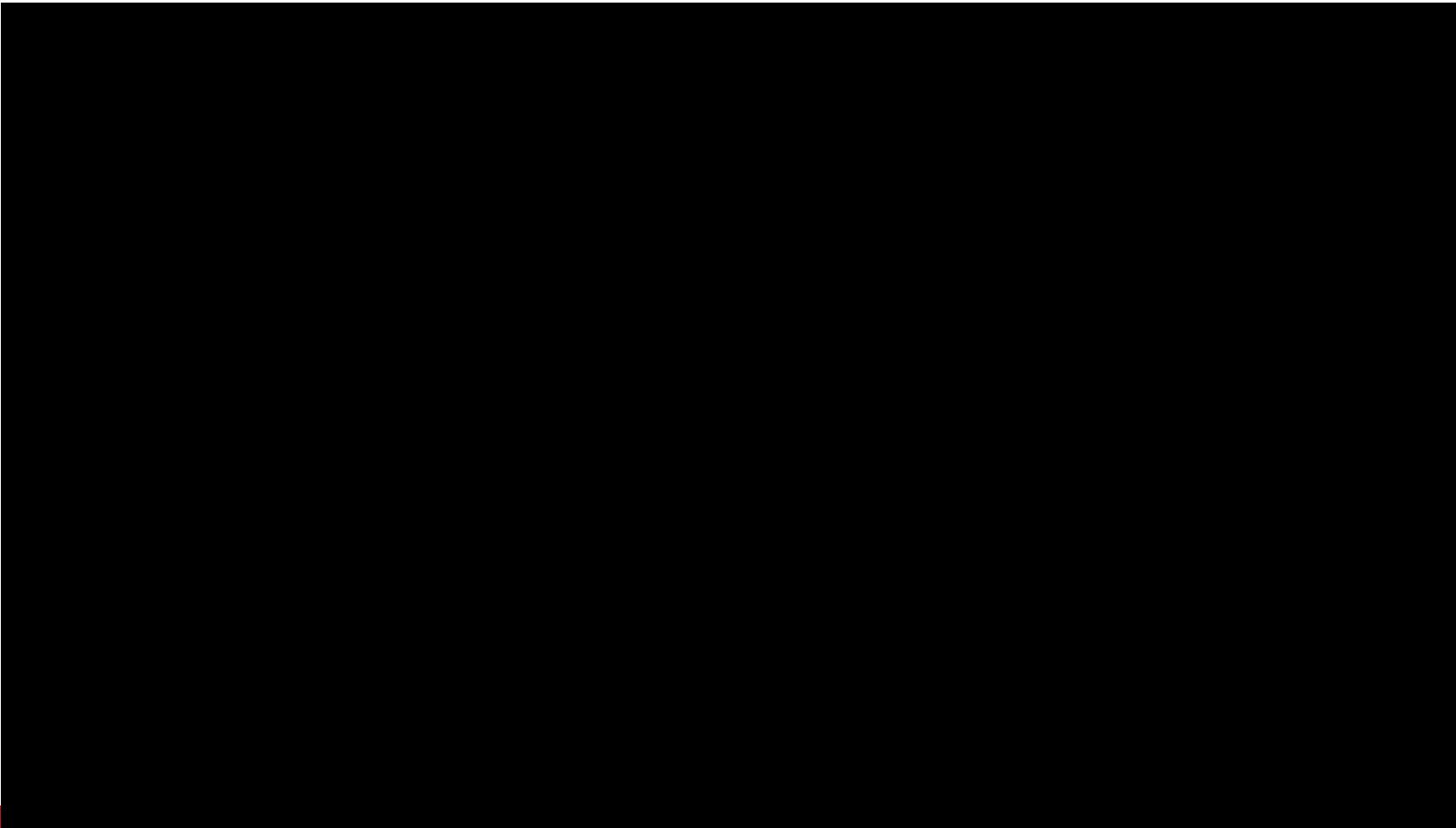
Εισαγωγή στα νευρωνικά δίκτυα

- Πρέπει να μετατρέψουμε οποιαδήποτε δεδομένα σε αριθμητική μορφή πριν τα τροφοδοτήσουμε στο νευρωνικό δίκτυο.
- Πολλοί διαφορετικοί τύποι δεδομένων
 - Οπτικό, κειμένου, χρονολογικές σειρές και ούτω καθεξής.
- Πρέπει να καταλάβουμε πώς να αναπαραστήσουμε τα προβλήματα με τρόπο που να είναι κατανοητός από τα τεχνητά νευρωνικά δίκτυα.

Κατασκευή νευρωνικού δικτύου

- Η ανθρώπινη διαδικασία μάθησης είναι ιεραρχική.
 - Διάφορα στάδια στο νευρωνικό δίκτυο του εγκεφάλου μας, κάθε στάδιο αντιστοιχεί σε διαφορετική λεπτομέρεια (επίπεδο λεπτομέρειας / *granularity*)
- Π.χ. οπτική αναγνώριση ενός κουτιού
 - Το πρώτο στάδιο προσδιορίζει απλά πράγματα, όπως γωνίες και άκρες,
 - το επόμενο στάδιο προσδιορίζει το γενικό σχήμα,
 - το επόμενο στάδιο προσδιορίζει το είδος του αντικειμένου που είναι
- Για να προσομοιωθεί η διαδικασία μάθησης του ανθρώπινου εγκεφάλου, δημιουργείται ένα τεχνητό νευρωνικό δίκτυο χρησιμοποιώντας στρώματα νευρώνων.
 - οι νευρώνες είναι εμπνευσμένοι από τους βιολογικούς νευρώνες
 - κάθε στρώμα ή επίπεδο (*layer*) είναι ένα σύνολο ανεξάρτητων νευρώνων
 - κάθε νευρώνας σε ένα επίπεδο συνδέεται με νευρώνες στο παρακείμενο επίπεδο

Κατασκευή νευρωνικού δικτύου



Δείτε το βίντεο [online](#)

Κατασκευή νευρωνικού δικτύου μάθηση και gradient descent

⟨/Recap⟩

Δείτε το βίντεο [online](#)

Κατασκευή νευρωνικού δικτύου

backpropagation

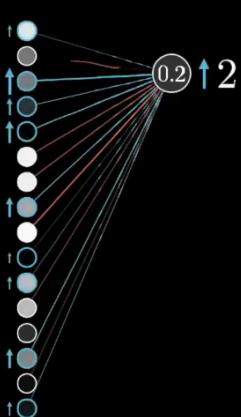
Plan

- Recap
- Intuitive walkthrough
- Derivatives in computational graphs

Increase b

Increase w_i
in proportion to a_i

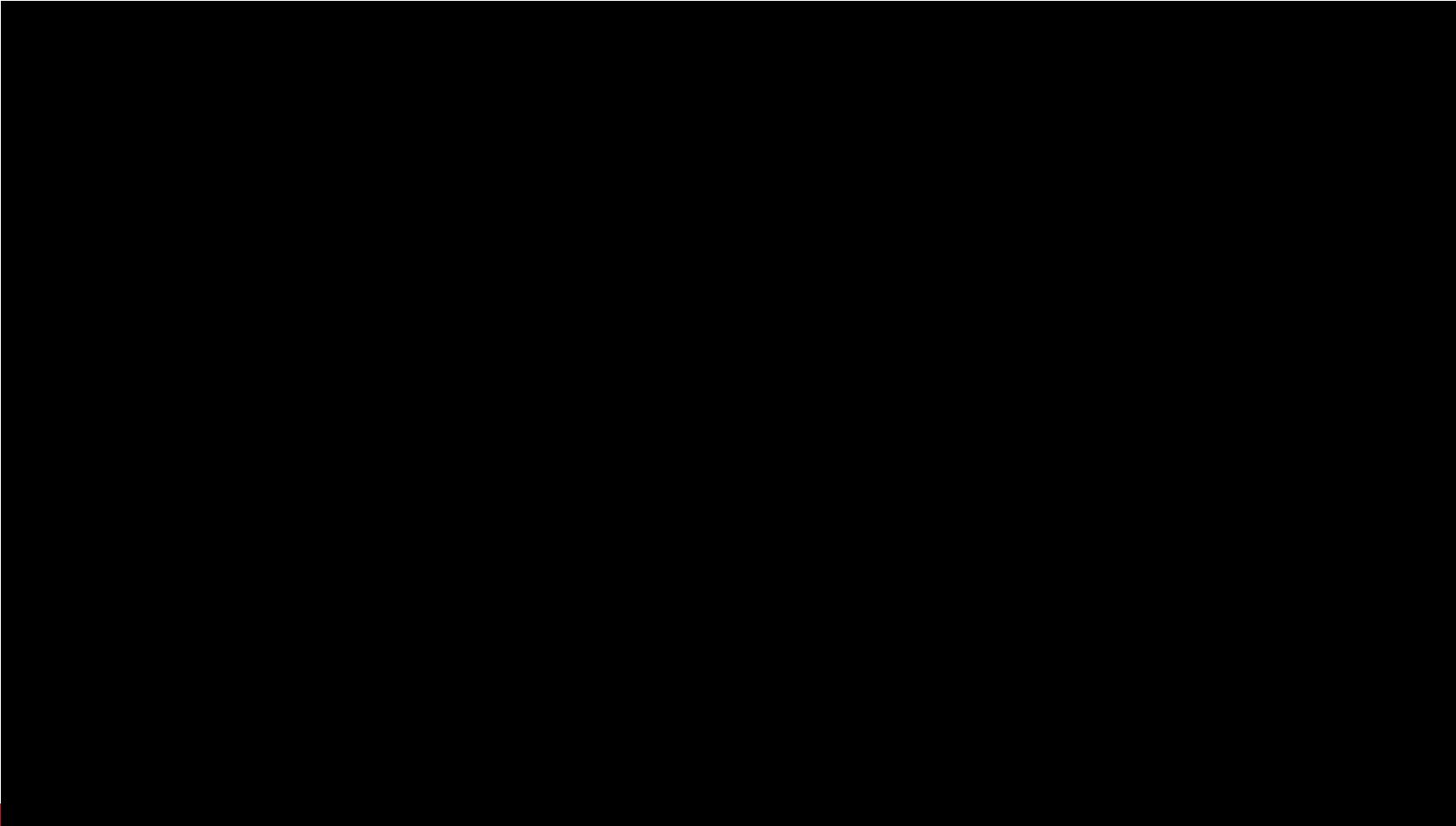
Change a_i



Δείτε το βίντεο [online](#)

Κατασκευή νευρωνικού δικτύου

backpropagation calculus

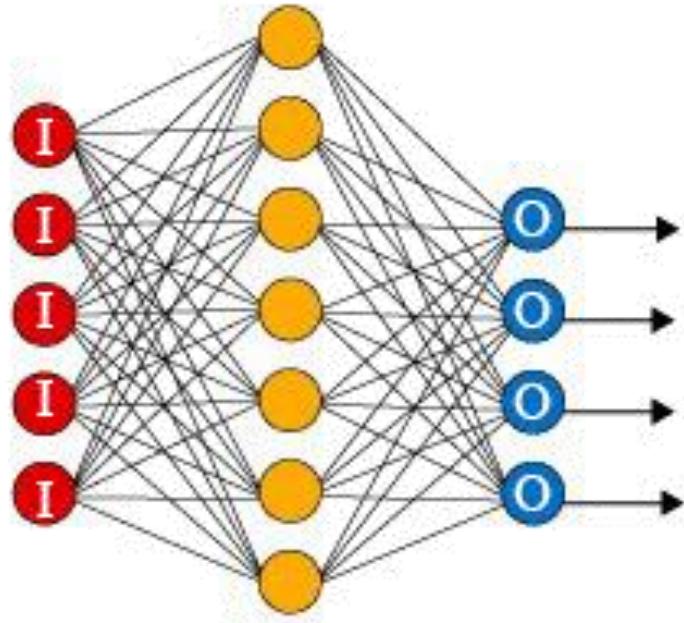


Δείτε το βίντεο [online](#)

Εκπαιδεύοντας ένα νευρωνικό δίκτυο

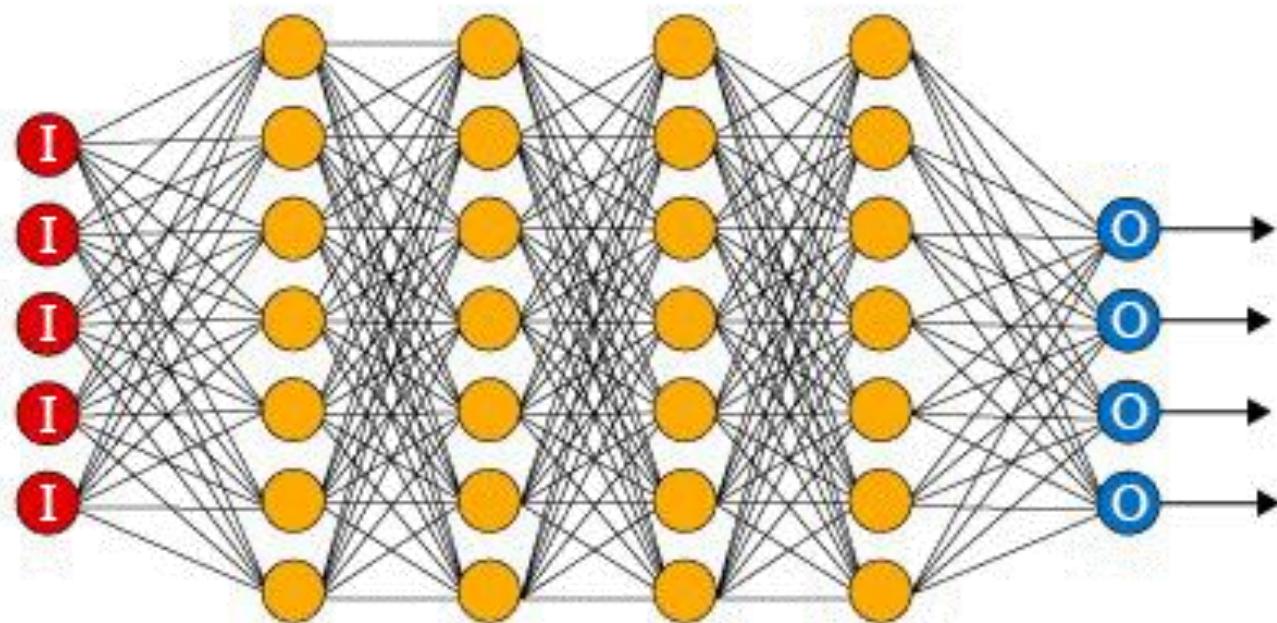
- Αν έχουμε να κάνουμε με δεδομένα εισόδου N -διάστατων, τότε το επίπεδο εισόδου (input layer) θα αποτελείται από N νευρώνες.
- Εάν έχουμε M διακριτές κατηγορίες/τάξεις στα δεδομένα εκπαίδευσης, τότε το επίπεδο εξόδου (output layer) θα αποτελείται από M νευρώνες.
- Τα επίπεδα μεταξύ των επιπέδων εισόδου και εξόδου ονομάζονται κρυφά επίπεδα (hidden layers).
 - Ένα απλό νευρωνικό δίκτυο θα αποτελείται από λίγα επίπεδα
 - Ένα βαθύ νευρωνικό δίκτυο (deep neural network) θα αποτελείται από πολλά επίπεδα.

Simple Neural Network



I Input Layer

Deep Learning Neural Network



● Hidden Layer

○ Output Layer

Εκπαίδευση νευρωνικού δικτύου

- Πρώτο βήμα: συλλέξτε τα κατάλληλα δεδομένα εκπαίδευσης και επισημάντε τα.
- Κάθε νευρώνας λειτουργεί ως μια απλή συνάρτηση και το νευρωνικό δίκτυο εκπαιδεύεται έως ότου το σφάλμα πέσει κάτω από μια ορισμένη τιμή.
- Σφάλμα: η διαφορά μεταξύ της προβλεπόμενης και της πραγματικής εξόδου.
- Με βάση το σφάλμα, το νευρωνικό δίκτυο προσαρμόζεται και επανεκπαιδεύεται μέχρι να πλησιάσει πιο κοντά στη λύση.
- Εδώ θα χρησιμοποιήσουμε μια βιβλιοθήκη που ονομάζεται [NeuroLab](#).

Εκπαίδευση νευρωνικού δικτύου

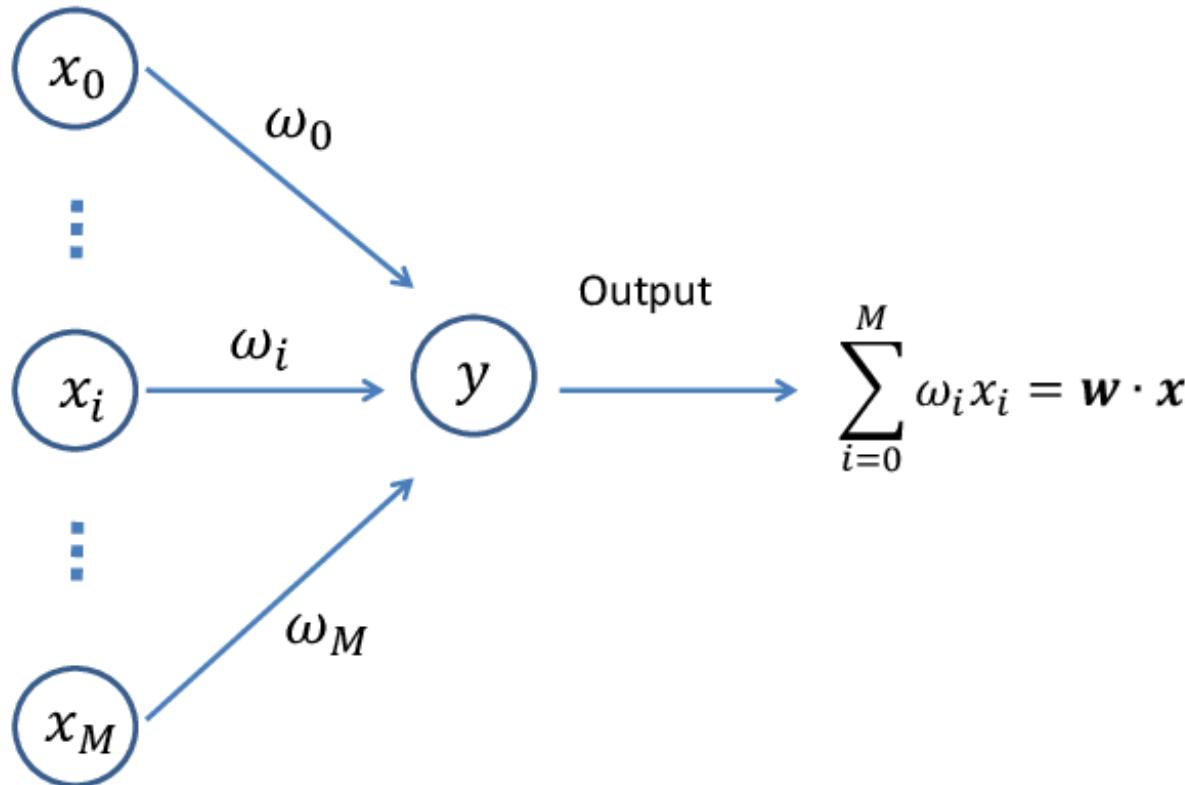
- Στη γραμμή εντολών πληκτρολογήστε:
 - pip3 install neurolab

```
C:\[REDACTED]\WPy64-31050\scripts>pip3 install neurolab
```

Δημιουργία ταξινομητή Perceptron

- Ένα **Perceptron** είναι το δομικό στοιχείο ενός τεχνητού νευρωνικού δικτύου.
- Είναι ένας μεμονωμένος νευρώνας που λαμβάνει εισόδους, εκτελεί υπολογισμούς σε αυτές και στη συνέχεια παράγει μια έξοδο.
- Χρησιμοποιεί μια απλή γραμμική συνάρτηση.
- Με ένα σημείο δεδομένων εισόδου N -διάστασης, υπολογίζει το σταθμισμένο άθροισμα της εισόδου και στη συνέχεια προσθέτει μια σταθερά (bias) για να παράγει την έξοδο.
- Αυτά τα απλά Perceptrons χρησιμοποιούνται για να σχεδιάσουν πολύ περίπλοκα βαθιά νευρωνικά δίκτυα.

Δημιουργία ταξινομητή Perceptron



Δημιουργία ταξινομητή Perceptron

- Τώρα κατεβάστε το αρχείο "perceptron_classifier.py" από τη σελίδα του μαθήματος στο moodle.
- Φορτώστε το αρχείο από το Spyder IDE.

**Warning : If you have problems when running file
perceptron_classifier.py with neurolab library try it running python
3.7 because there may be compatibility issues with later versions**

Before running perceptron_classifier.py execute the following in command line

pip3 install numpy

pip3 install matplotlib

pip install neurolab

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import neurolab as nl
4
5 # Load input data
6 text = np.loadtxt('data_perceptron.txt')
7
8 # Separate datapoints and labels
9 data = text[:, :2]
10 labels = text[:, 2].reshape((text.shape[0], 1))
11
12 # Plot input data
13 plt.figure()
14 plt.scatter(data[:,0], data[:,1])
15 plt.xlabel('Dimension 1')
16 plt.ylabel('Dimension 2')
17 plt.title('Input data')
18
19 # Define minimum and maximum values for each dimension
20 dim1_min, dim1_max, dim2_min, dim2_max = 0, 1, 0, 1
21
22 # Number of neurons in the output Layer
23 num_output = labels.shape[1]
24
25 # Define a perceptron with 2 input neurons (because we
26 # have 2 dimensions in the input data)
27 dim1 = [dim1_min, dim1_max]
28 dim2 = [dim2_min, dim2_max]
29 perceptron = nl.net.newp([dim1, dim2], num_output)
30
31 # Train the perceptron using the data
32 error_progress = perceptron.train(data, labels, epochs=100, show=20, lr=0.03)
33
34 # Plot the training progress
35 plt.figure()
36 plt.plot(error_progress)
37 plt.xlabel('Number of epochs')
38 plt.ylabel('Training error')
39 plt.title('Training error progress')
40 plt.grid()
41
42 plt.show()
```

Εισαγάγετε τα απαραίτητα πακέτα

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import neurolab as nl
4
5 # Load input data
6 text = np.loadtxt('data_perceptron.txt')
7
8 # Separate datapoints and labels
9 data = text[:, :2]
10 labels = text[:, 2].reshape((text.shape[0], 1))
11
12 # Plot input data
13 plt.figure()
14 plt.scatter(data[:,0], data[:,1])
15 plt.xlabel('Dimension 1')
16 plt.ylabel('Dimension 2')
17 plt.title('Input data')
18
19 # Define minimum and maximum values for each dimension
20 dim1_min, dim1_max, dim2_min, dim2_max = 0, 1, 0, 1
21
22 # Number of neurons in the output Layer
23 num_output = labels.shape[1]
24
25 # Define a perceptron with 2 input neurons (because we
26 # have 2 dimensions in the input data)
27 dim1 = [dim1_min, dim1_max]
28 dim2 = [dim2_min, dim2_max]
29 perceptron = nl.net.newp([dim1, dim2], num_output)
30
31 # Train the perceptron using the data
32 error_progress = perceptron.train(data, labels, epochs=100, show=20, lr=0.03)
33
34 # Plot the training progress
35 plt.figure()
36 plt.plot(error_progress)
37 plt.xlabel('Number of epochs')
38 plt.ylabel('Training error')
39 plt.title('Training error progress')
40 plt.grid()
41
42 plt.show()
```

Εισαγάγετε τα απαραίτητα πακέτα

Παρατηρήστε την εισαγωγή του νέου πακέτου που εγκαταστήσαμε.

Τεκμηρίωση μπορείτε να βρείτε [εδώ](#).

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import neurolab as nl
4
5 # Load input data
6 text = np.loadtxt('data_perceptron.txt')
7
8 # Separate datapoints and labels
9 data = text[:, :2]
10 labels = text[:, 2].reshape((text.shape[0], 1))
11
12 # Plot input data
13 plt.figure()
14 plt.scatter(data[:,0], data[:,1])
15 plt.xlabel('Dimension 1')
16 plt.ylabel('Dimension 2')
17 plt.title('Input data')
18
19 # Define minimum and maximum values for each dimension
20 dim1_min, dim1_max, dim2_min, dim2_max = 0, 1, 0, 1
21
22 # Number of neurons in the output Layer
23 num_output = labels.shape[1]
24
25 # Define a perceptron with 2 input neurons (because we
26 # have 2 dimensions in the input data)
27 dim1 = [dim1_min, dim1_max]
28 dim2 = [dim2_min, dim2_max]
29 perceptron = nl.net.newp([dim1, dim2], num_output)
30
31 # Train the perceptron using the data
32 error_progress = perceptron.train(data, labels, epochs=100, show=20, lr=0.03)
33
34 # Plot the training progress
35 plt.figure()
36 plt.plot(error_progress)
37 plt.xlabel('Number of epochs')
38 plt.ylabel('Training error')
39 plt.title('Training error progress')
40 plt.grid()
41
42 plt.show()
```

Φορτώστε τα δεδομένα εισόδου από το αρχείο κειμένου data_perceptron.txt που σας παρέχεται στη σελίδα του μαθήματος.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import neurolab as nl
4
5 # Load input data
6 text = np.loadtxt('data_perceptron.txt')
7
8 # Separate datapoints and labels
9 data = text[:, :2]
10 labels = text[:, 2].reshape((text.shape[0], 1))
11
12 # Plot input data
13 plt.figure()
14 plt.scatter(data[:,0], data[:,1])
15 plt.xlabel('Dimension 1')
16 plt.ylabel('Dimension 2')
17 plt.title('Input data')
18
19 # Define minimum and maximum values for each dimension
20 dim1_min, dim1_max, dim2_min, dim2_max = 0, 1, 0, 1
21
22 # Number of neurons in the output Layer
23 num_output = labels.shape[1]
24
25 # Define a perceptron with 2 input neurons (because we
26 # have 2 dimensions in the input data)
27 dim1 = [dim1_min, dim1_max]
28 dim2 = [dim2_min, dim2_max]
29 perceptron = nl.net.newp([dim1, dim2], num_output)
30
31 # Train the perceptron using the data
32 error_progress = perceptron.train(data, labels, epochs=100, show=20, lr=0.03)
33
34 # Plot the training progress
35 plt.figure()
36 plt.plot(error_progress)
37 plt.xlabel('Number of epochs')
38 plt.ylabel('Training error')
39 plt.title('Training error progress')
40 plt.grid()
41
42 plt.show()

```

Αυτά είναι τα δεδομένα σας.

0.38	0.19	0
0.17	0.31	0
0.29	0.54	0
0.89	0.55	1
0.78	0.36	1

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import neurolab as nl
4
5 # Load input data
6 text = np.loadtxt('data_perceptron.txt')
7
8 # Separate datapoints and labels
9 data = text[:, :2]
10 labels = text[:, 2].reshape((text.shape[0], 1))
11
12 # Plot input data
13 plt.figure()
14 plt.scatter(data[:,0], data[:,1])
15 plt.xlabel('Dimension 1')
16 plt.ylabel('Dimension 2')
17 plt.title('Input data')
18
19 # Define minimum and maximum values for each dimension
20 dim1_min, dim1_max, dim2_min, dim2_max = 0, 1, 0, 1
21
22 # Number of neurons in the output Layer
23 num_output = labels.shape[1]
24
25 # Define a perceptron with 2 input neurons (because we
26 # have 2 dimensions in the input data)
27 dim1 = [dim1_min, dim1_max]
28 dim2 = [dim2_min, dim2_max]
29 perceptron = nl.net.newp([dim1, dim2], num_output)
30
31 # Train the perceptron using the data
32 error_progress = perceptron.train(data, labels, epochs=100, show=20, lr=0.03)
33
34 # Plot the training progress
35 plt.figure()
36 plt.plot(error_progress)
37 plt.xlabel('Number of epochs')
38 plt.ylabel('Training error')
39 plt.title('Training error progress')
40 plt.grid()
41
42 plt.show()

```

0.38	0.19	0
0.17	0.31	0
0.29	0.54	0
0.89	0.55	1
0.78	0.36	1

Διαχωρίστε το κείμενο σε σημεία δεδομένων και ετικέτες.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import neurolab as nl
4
5 # Load input data
6 text = np.loadtxt('data_perceptron.txt')
7
8 # Separate datapoints and labels
9 data = text[:, :2]
10 labels = text[:, 2].reshape((text.shape[0], 1))
11
12 # Plot input data
13 plt.figure()
14 plt.scatter(data[:,0], data[:,1])
15 plt.xlabel('Dimension 1')
16 plt.ylabel('Dimension 2')
17 plt.title('Input data')
18
19 # Define minimum and maximum values for each dimension
20 dim1_min, dim1_max, dim2_min, dim2_max = 0, 1, 0, 1
21
22 # Number of neurons in the output Layer
23 num_output = labels.shape[1]
24
25 # Define a perceptron with 2 input neurons (because we
26 # have 2 dimensions in the input data)
27 dim1 = [dim1_min, dim1_max]
28 dim2 = [dim2_min, dim2_max]
29 perceptron = nl.net.newp([dim1, dim2], num_output)
30
31 # Train the perceptron using the data
32 error_progress = perceptron.train(data, labels, epochs=100, show=20, lr=0.03)
33
34 # Plot the training progress
35 plt.figure()
36 plt.plot(error_progress)
37 plt.xlabel('Number of epochs')
38 plt.ylabel('Training error')
39 plt.title('Training error progress')
40 plt.grid()
41
42 plt.show()

```

0.38	0.19	0
0.17	0.31	0
0.29	0.54	0
0.89	0.55	1
0.78	0.36	1

Αποτυπώστε τα σημεία σε ένα γράφημα.

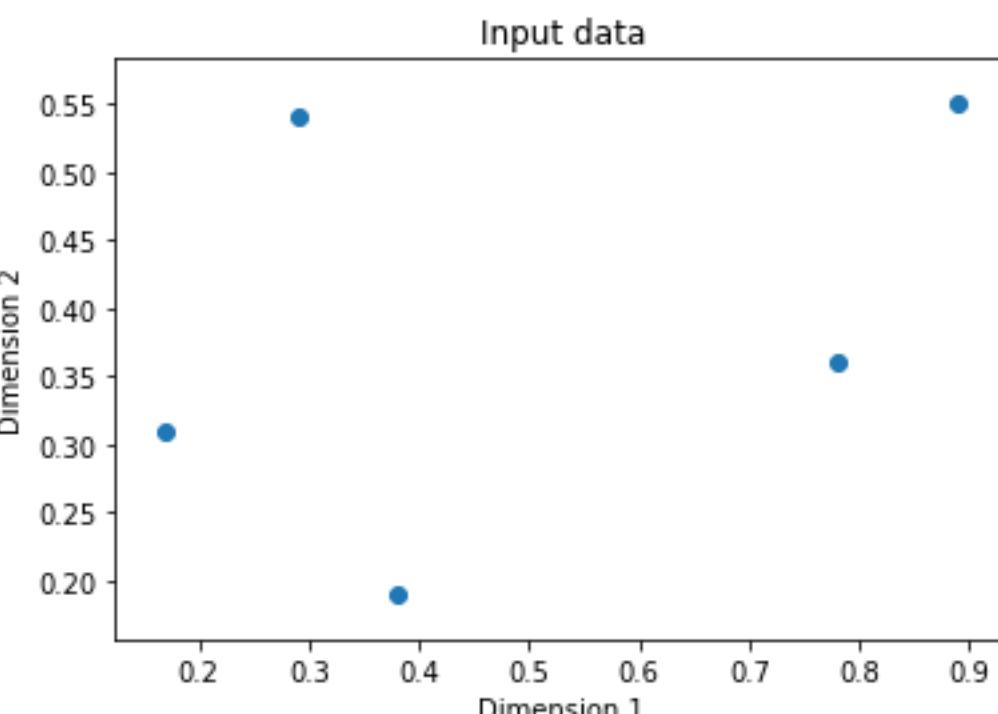
```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import neurolab as nl
4
5 # Load input data
6 text = np.loadtxt('data_perceptron.txt')
7
8 # Separate datapoints and labels
9 data = text[:, :2]
10 labels = text[:, 2].reshape((text.shape[0], 1))
11
12 # Plot input data
13 plt.figure()
14 plt.scatter(data[:,0], data[:,1])
15 plt.xlabel('Dimension 1')
16 plt.ylabel('Dimension 2')
17 plt.title('Input data')
18
19 # Define minimum and maximum values for ea
20 dim1_min, dim1_max, dim2_min, dim2_max = 0
21
22 # Number of neurons in the output Layer
23 num_output = labels.shape[1]
24
25 # Define a perceptron with 2 input neurons
26 # have 2 dimensions in the input data)
27 dim1 = [dim1_min, dim1_max]
28 dim2 = [dim2_min, dim2_max]
29 perceptron = nl.net.newp([dim1, dim2], num
30
31 # Train the perceptron using the data
32 error_progress = perceptron.train(data, la
33
34 # Plot the training progress
35 plt.figure()
36 plt.plot(error_progress)
37 plt.xlabel('Number of epochs')
38 plt.ylabel('Training error')
39 plt.title('Training error progress')
40 plt.grid()
41
42 plt.show()

```

0.38	0.19	0
0.17	0.31	0
0.29	0.54	0
0.89	0.55	1
0.78	0.36	1

Θα πρέπει να πάρετε αυτό το αποτέλεσμα.



```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import neurolab as nl
4
5 # Load input data
6 text = np.loadtxt('data_perceptron.txt')
7
8 # Separate datapoints and labels
9 data = text[:, :2]
10 labels = text[:, 2].reshape((text.shape[0], 1))
11
12 # Plot input data
13 plt.figure()
14 plt.scatter(data[:,0], data[:,1])
15 plt.xlabel('Dimension 1')
16 plt.ylabel('Dimension 2')
17 plt.title('Input data')
18
19 # Define minimum and maximum values for each dimension
20 dim1_min, dim1_max, dim2_min, dim2_max = 0, 1, 0, 1
21
22 # Number of neurons in the output Layer
23 num_output = labels.shape[1]
24
25 # Define a perceptron with 2 input neurons (because we
26 # have 2 dimensions in the input data)
27 dim1 = [dim1_min, dim1_max]
28 dim2 = [dim2_min, dim2_max]
29 perceptron = nl.net.newp([dim1, dim2], num_output)
30
31 # Train the perceptron using the data
32 error_progress = perceptron.train(data, labels, epochs=100, show=20, lr=0.03)
33
34 # Plot the training progress
35 plt.figure()
36 plt.plot(error_progress)
37 plt.xlabel('Number of epochs')
38 plt.ylabel('Training error')
39 plt.title('Training error progress')
40 plt.grid()
41
42 plt.show()

```

0.38	0.19	0
0.17	0.31	0
0.29	0.54	0
0.89	0.55	1
0.78	0.36	1

Καθορίστε τις μέγιστες και ελάχιστες τιμές που μπορεί να λάβει κάθε διάσταση.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import neurolab as nl
4
5 # Load input data
6 text = np.loadtxt('data_perceptron.txt')
7
8 # Separate datapoints and labels
9 data = text[:, :2]
10 labels = text[:, 2].reshape((text.shape[0], 1))
11
12 # Plot input data
13 plt.figure()
14 plt.scatter(data[:,0], data[:,1])
15 plt.xlabel('Dimension 1')
16 plt.ylabel('Dimension 2')
17 plt.title('Input data')
18
19 # Define minimum and maximum values for each dimension
20 dim1_min, dim1_max, dim2_min, dim2_max = 0, 1, 0, 1
21
22 # Number of neurons in the output layer
23 num_output = labels.shape[1]
24
25 # Define a perceptron with 2 input neurons (because we
26 # have 2 dimensions in the input data)
27 dim1 = [dim1_min, dim1_max]
28 dim2 = [dim2_min, dim2_max]
29 perceptron = nl.net.newp([dim1, dim2], num_output)
30
31 # Train the perceptron using the data
32 error_progress = perceptron.train(data, labels, epochs=100, show=20, lr=0.03)
33
34 # Plot the training progress
35 plt.figure()
36 plt.plot(error_progress)
37 plt.xlabel('Number of epochs')
38 plt.ylabel('Training error')
39 plt.title('Training error progress')
40 plt.grid()
41
42 plt.show()

```

0.38	0.19	0
0.17	0.31	0
0.29	0.54	0
0.89	0.55	1
0.78	0.36	1

Τα δεδομένα χωρίζονται σε δύο κατηγορίες, οπότε χρειαζόμαστε μόνο ένα bit (0 ή 1) για να αναπαραστήσουμε την έξοδο.

Το επίπεδο εξόδου θα περιέχει έναν μόνο νευρώνα.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import neurolab as nl
4
5 # Load input data
6 text = np.loadtxt('data_perceptron.txt')
7
8 # Separate datapoints and labels
9 data = text[:, :2]
10 labels = text[:, 2].reshape((text.shape[0], 1))
11
12 # Plot input data
13 plt.figure()
14 plt.scatter(data[:,0], data[:,1])
15 plt.xlabel('Dimension 1')
16 plt.ylabel('Dimension 2')
17 plt.title('Input data')
18
19 # Define minimum and maximum values for each dimension
20 dim1_min, dim1_max, dim2_min, dim2_max = 0, 1, 0, 1
21
22 # Number of neurons in the output Layer
23 num_output = labels.shape[1]
24
25 # Define a perceptron with 2 input neurons (because we
26 # have 2 dimensions in the input data)
27 dim1 = [dim1_min, dim1_max]
28 dim2 = [dim2_min, dim2_max]
29 perceptron = nl.net.newp([dim1, dim2], num_output)
30
31 # Train the perceptron using the data
32 error_progress = perceptron.train(data, labels, epochs=100, show=20, lr=0.03)
33
34 # Plot the training progress
35 plt.figure()
36 plt.plot(error_progress)
37 plt.xlabel('Number of epochs')
38 plt.ylabel('Training error')
39 plt.title('Training error progress')
40 plt.grid()
41
42 plt.show()

```

0.38	0.19	0
0.17	0.31	0
0.29	0.54	0
0.89	0.55	1
0.78	0.36	1

$x_1 \quad x_2$

Τα σημεία δεδομένων είναι δισδιάστατα (x_1, x_2).

Ορίζουμε ένα Perceptron με δύο νευρώνες εισόδου, όπου εκχωρούμε έναν νευρώνα για κάθε διάσταση.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import neurolab as nl
4
5 # Load input data
6 text = np.loadtxt('data_perceptron.txt')
7
8 # Separate datapoints and labels
9 data = text[:, :2]
10 labels = text[:, 2].reshape((text.shape[0], 1))
11
12 # Plot input data
13 plt.figure()
14 plt.scatter(data[:,0], data[:,1])
15 plt.xlabel('Dimension 1')
16 plt.ylabel('Dimension 2')
17 plt.title('Input data')
18
19 # Define minimum and maximum values for each dimension
20 dim1_min, dim1_max, dim2_min, dim2_max = 0, 1, 0, 1
21
22 # Number of neurons in the output Layer
23 num_output = labels.shape[1]
24
25 # Define a perceptron with 2 input neurons (because we
26 # have 2 dimensions in the input data)
27 dim1 = [dim1_min, dim1_max]
28 dim2 = [dim2_min, dim2_max]
29 perceptron = nl.net.newp([dim1, dim2], num_output)
30
31 # Train the perceptron using the data
32 error_progress = perceptron.train(data, labels, epochs=100, show=20, lr=0.03)
33
34 # Plot the training progress
35 plt.figure()
36 plt.plot(error_progress)
37 plt.xlabel('Number of epochs')
38 plt.ylabel('Training error')
39 plt.title('Training error progress')
40 plt.grid()
41
42 plt.show()

```

0.38	0.19	0
0.17	0.31	0
0.29	0.54	0
0.89	0.55	1
0.78	0.36	1

Εκπαιδεύουμε το perceptron με τα δεδομένα από το training σετ.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import neurolab as nl
4
5 # Load input data
6 text = np.loadtxt('data_perceptron.txt')
7
8 # Separate datapoints and labels
9 data = text[:, :2]
10 labels = text[:, 2].reshape((text.shape[0], 1))
11
12 # Plot input data
13 plt.figure()
14 plt.scatter(data[:,0], data[:,1])
15 plt.xlabel('Dimension 1')
16 plt.ylabel('Dimension 2')
17 plt.title('Input data')
18
19 # Define minimum and maximum values for each dimension
20 dim1_min, dim1_max, dim2_min, dim2_max = 0, 1, 0, 1
21
22 # Number of neurons in the output Layer
23 num_output = labels.shape[1]
24
25 # Define a perceptron with 2 input neurons (because we
26 # have 2 dimensions in the input data)
27 dim1 = [dim1_min, dim1_max]
28 dim2 = [dim2_min, dim2_max]
29 perceptron = nl.net.newp([dim1, dim2], num_output)
30
31 # Train the perceptron using the data
32 error_progress = perceptron.train(data, labels, epochs=100, show=20, lr=0.03)
33
34 # Plot the training progress
35 plt.figure()
36 plt.plot(error_progress)
37 plt.xlabel('Number of epochs')
38 plt.ylabel('Training error')
39 plt.title('Training error progress')
40 plt.grid()
41
42 plt.show()

```

0.38	0.19	0
0.17	0.31	0
0.29	0.54	0
0.89	0.55	1
0.78	0.36	1

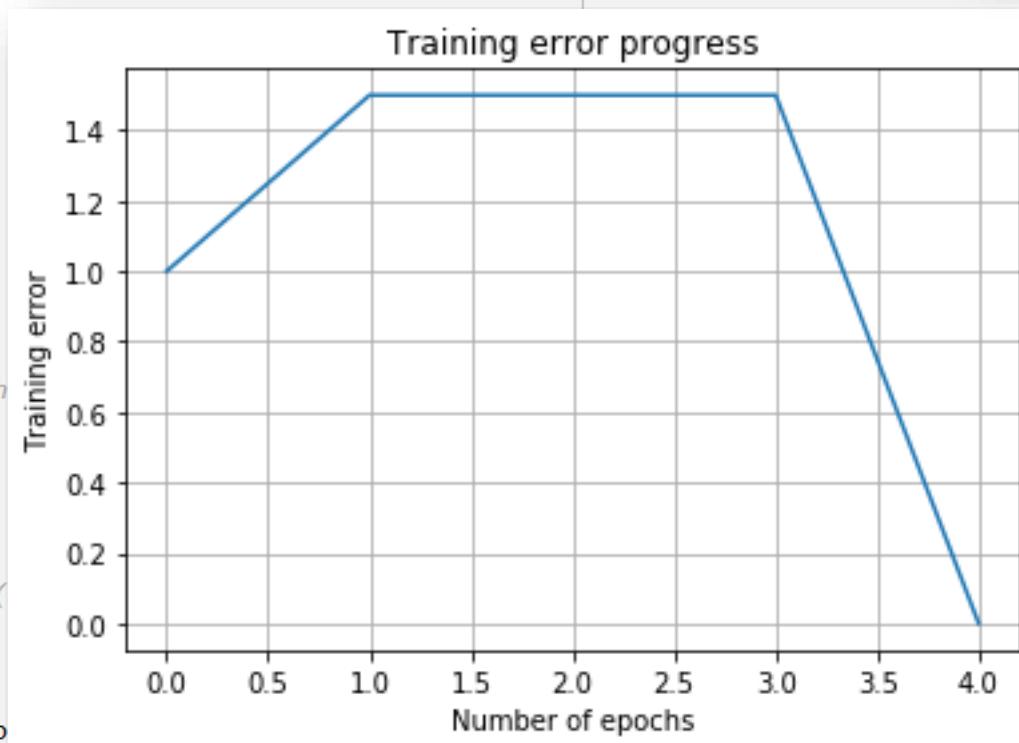
Σχεδιάστε την πρόοδο της εκπαίδευσης χρησιμοποιώντας τη μέτρηση σφάλματος (error_progress).

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import neurolab as nl
4
5 # Load input data
6 text = np.loadtxt('data_perceptron.txt')
7
8 # Separate datapoints and labels
9 data = text[:, :2]
10 labels = text[:, 2].reshape((text.shape[0]),
11
12 # Plot input data
13 plt.figure()
14 plt.scatter(data[:,0], data[:,1])
15 plt.xlabel('Dimension 1')
16 plt.ylabel('Dimension 2')
17 plt.title('Input data')
18
19 # Define minimum and maximum values for each
20 dim1_min, dim1_max, dim2_min, dim2_max = 0,
21
22 # Number of neurons in the output Layer
23 num_output = labels.shape[1]
24
25 # Define a perceptron with 2 input neurons (
26 # have 2 dimensions in the input data)
27 dim1 = [dim1_min, dim1_max]
28 dim2 = [dim2_min, dim2_max]
29 perceptron = nl.net.newp([dim1, dim2], num_o
30
31 # Train the perceptron using the data
32 error_progress = perceptron.train(data, labels, epochs=100, show=20, lr=0.03)
33
34 # Plot the training progress
35 plt.figure()
36 plt.plot(error_progress)
37 plt.xlabel('Number of epochs')
38 plt.ylabel('Training error')
39 plt.title('Training error progress')
40 plt.grid()
41
42 plt.show()

```

0.38	0.19	0
0.17	0.31	0
0.29	0.54	0
0.89	0.55	1
0.78	0.36	1



Θα πρέπει να πάρετε περίπου αυτό
το αποτέλεσμα. Τι παρατηρείτε;