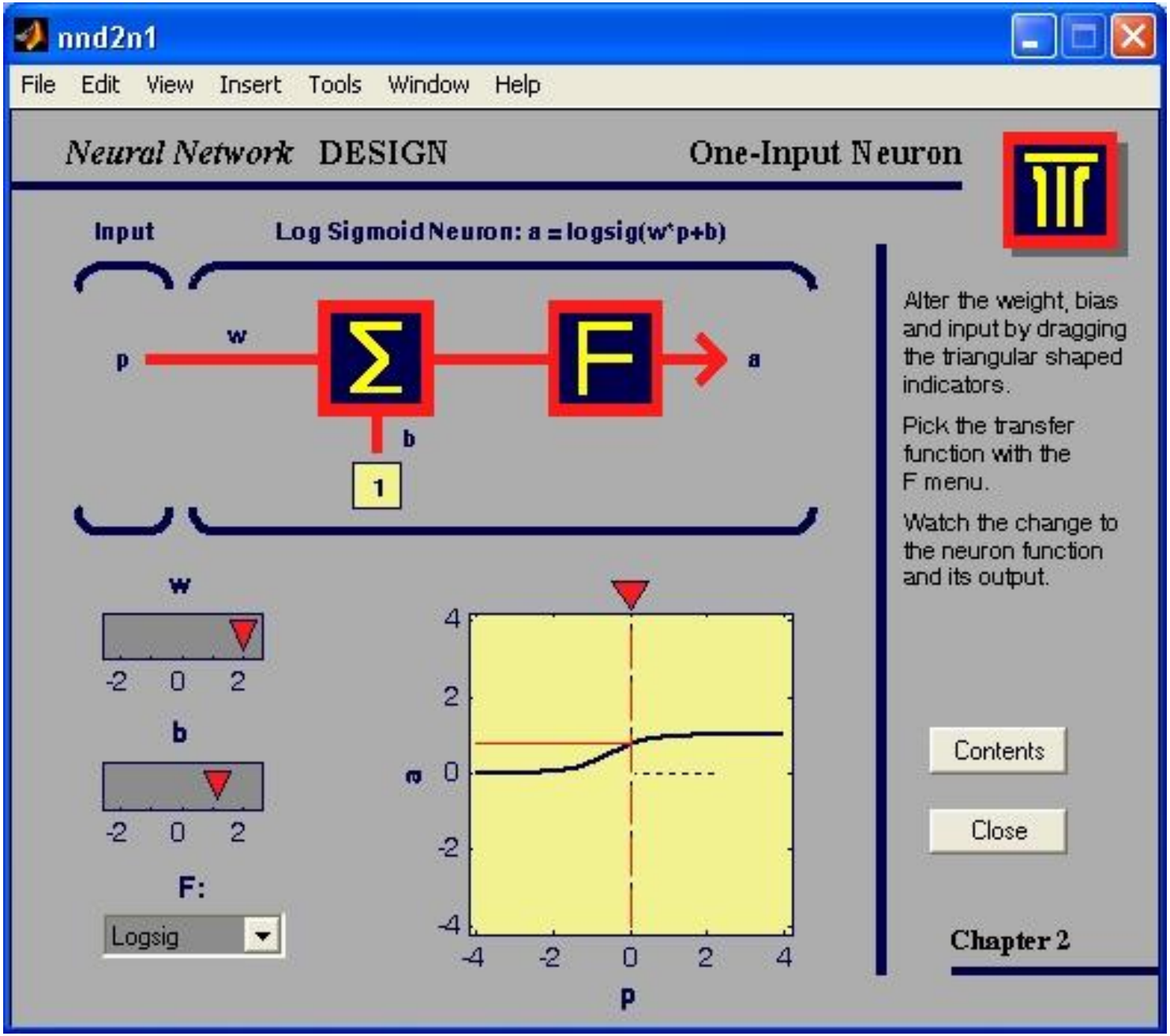
Introduction to Machine Learning

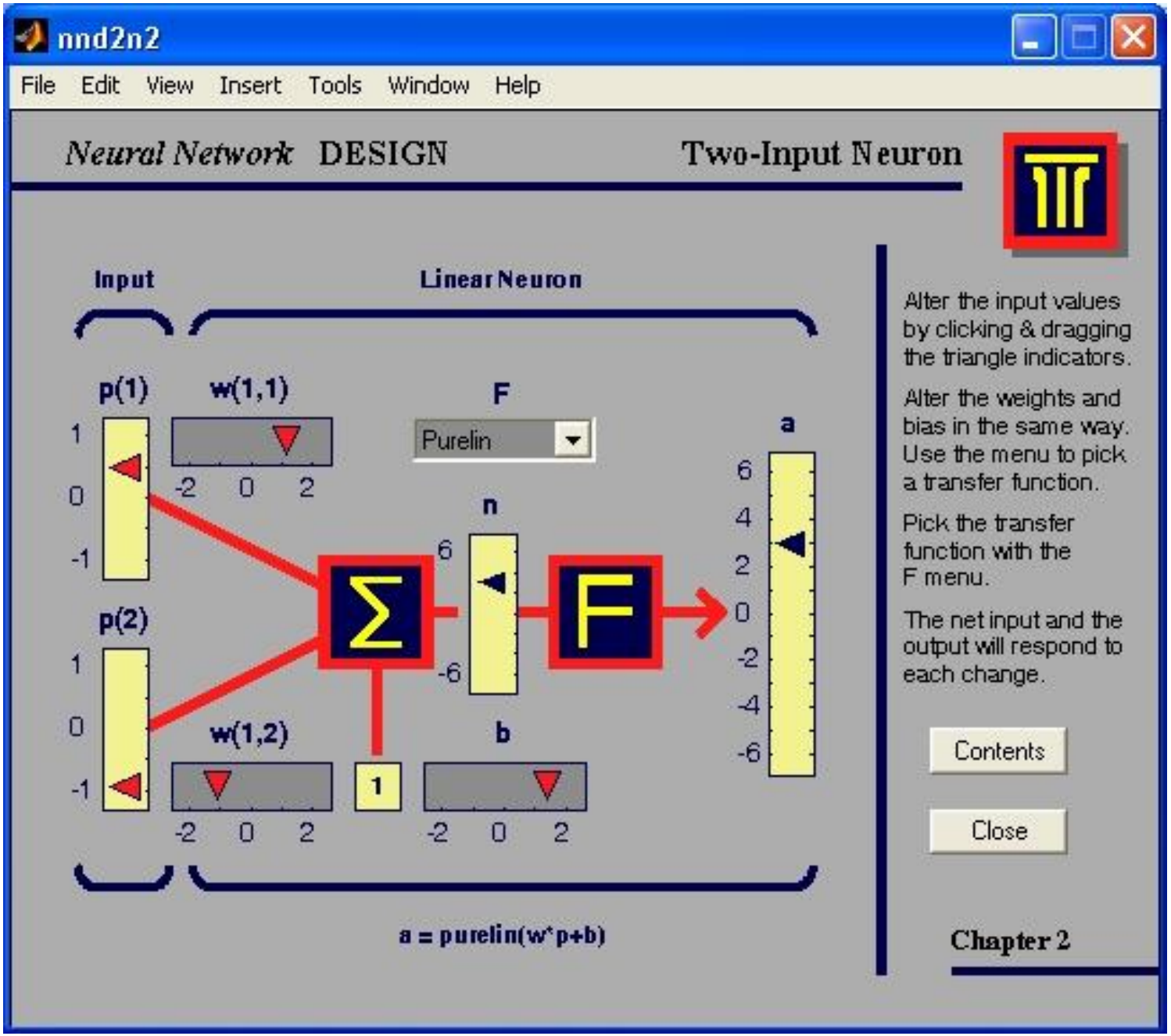
Matlab Neural Network Demos

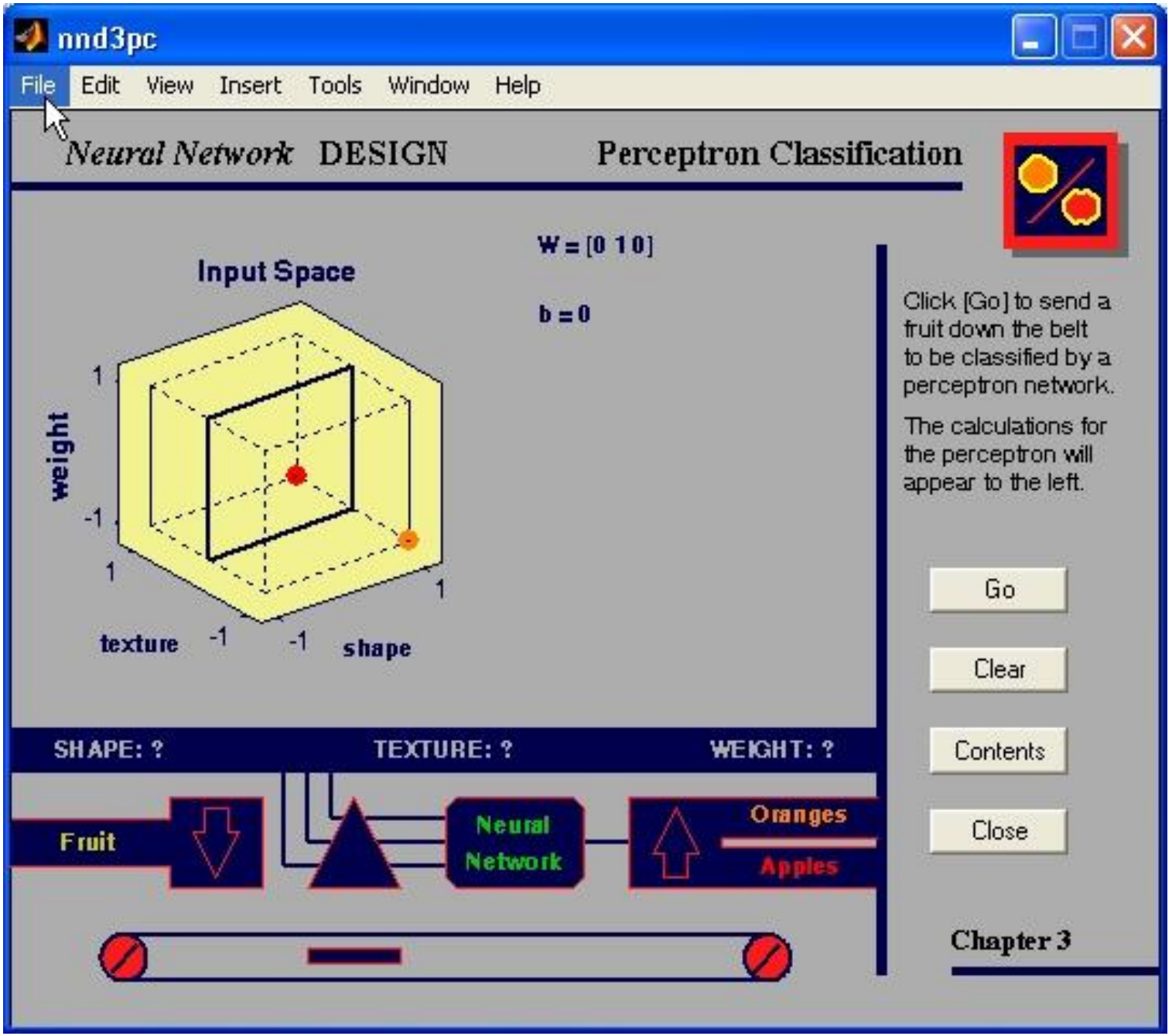
Barnabás Póczos

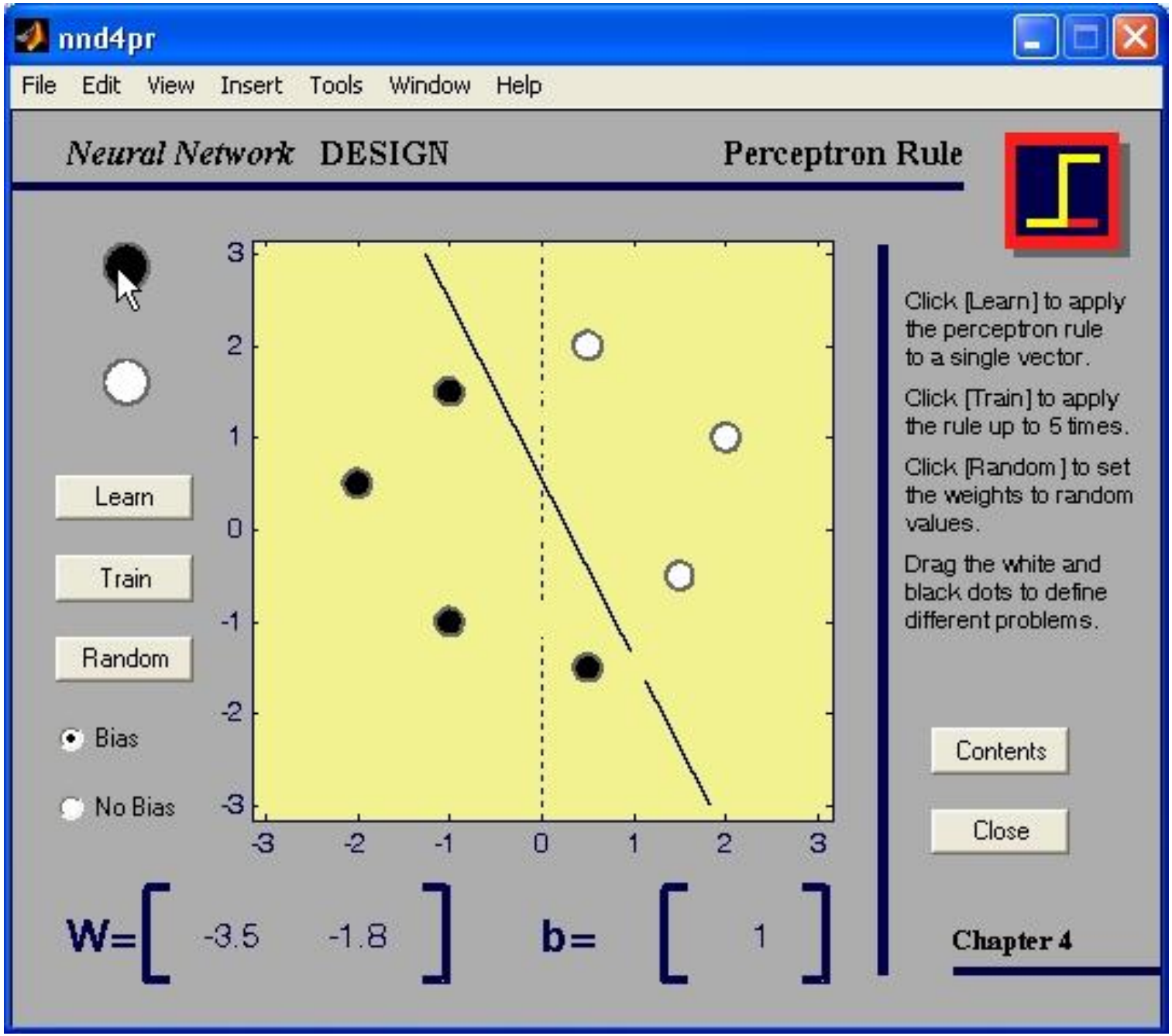
MACHINE LEARNING DEPARTMENT

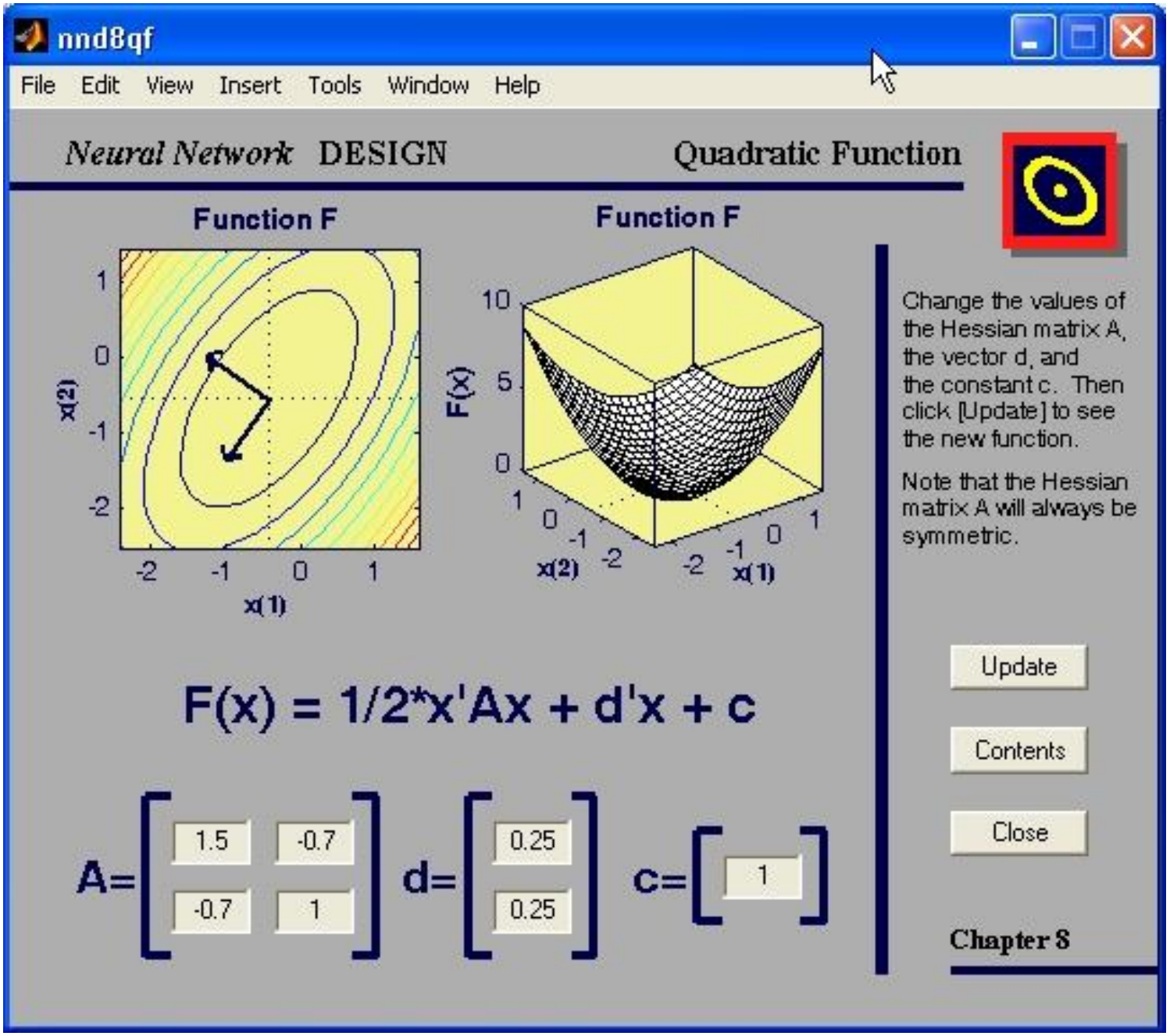
Carnegie Mellon.
School of Computer Science

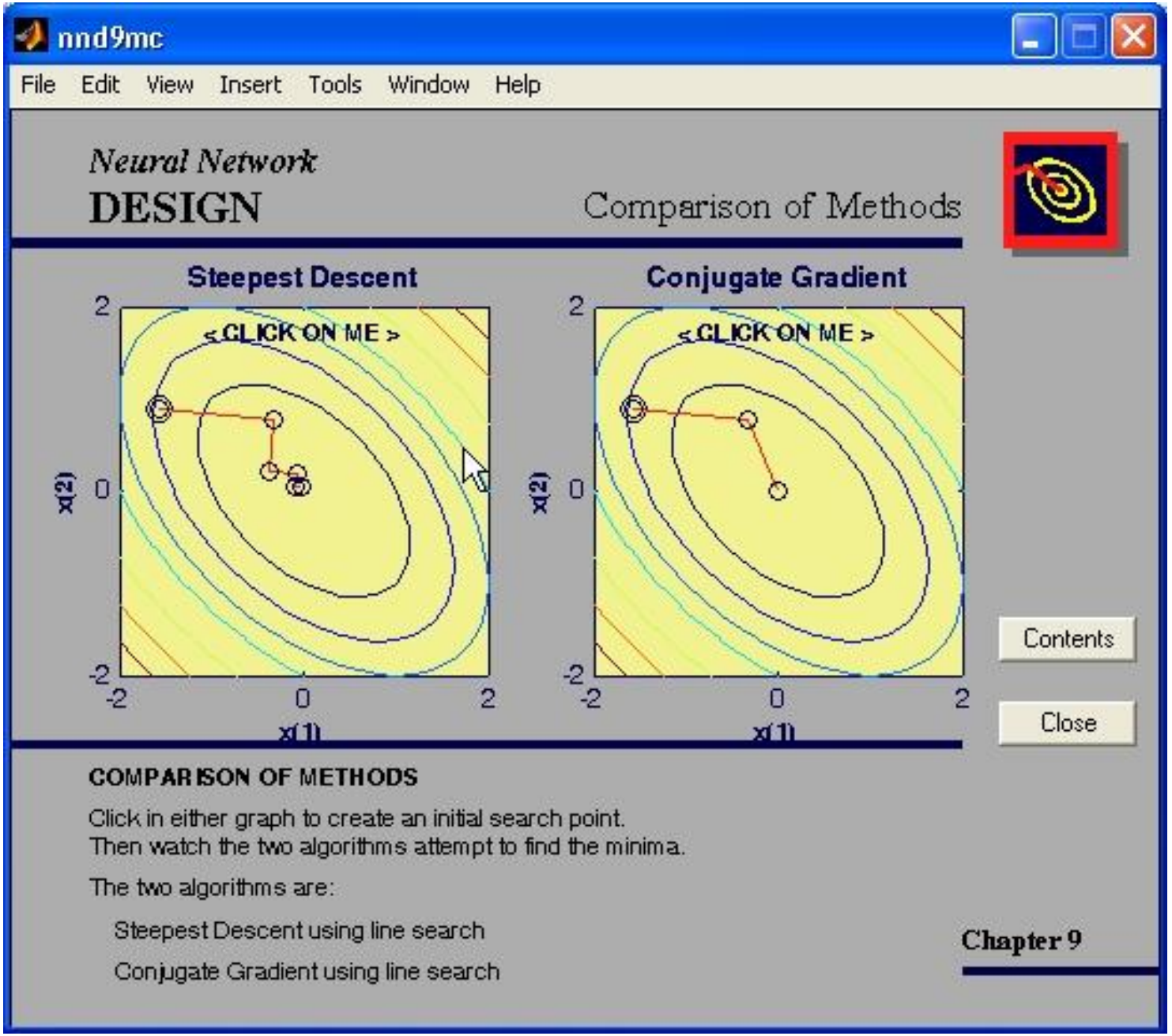

nnd2n1
File Edit View Insert Tools Window Help
Neural Network DESIGN
One-Input Neuron
Input
Log Sigmoid Neuron: a = logsig(w*p+b)
p
w
Σ
F
a
b
1
Alter the weight, bias and input by dragging the triangular shaped indicators.
Pick the transfer function with the F menu.
Watch the change to the neuron function and its output.
w
-2 0 2
b
-2 0 2
F:
Logsig
4
2
0
-2
-4
a
-4 -2 0 2 4
p
Contents
Close
Chapter 2

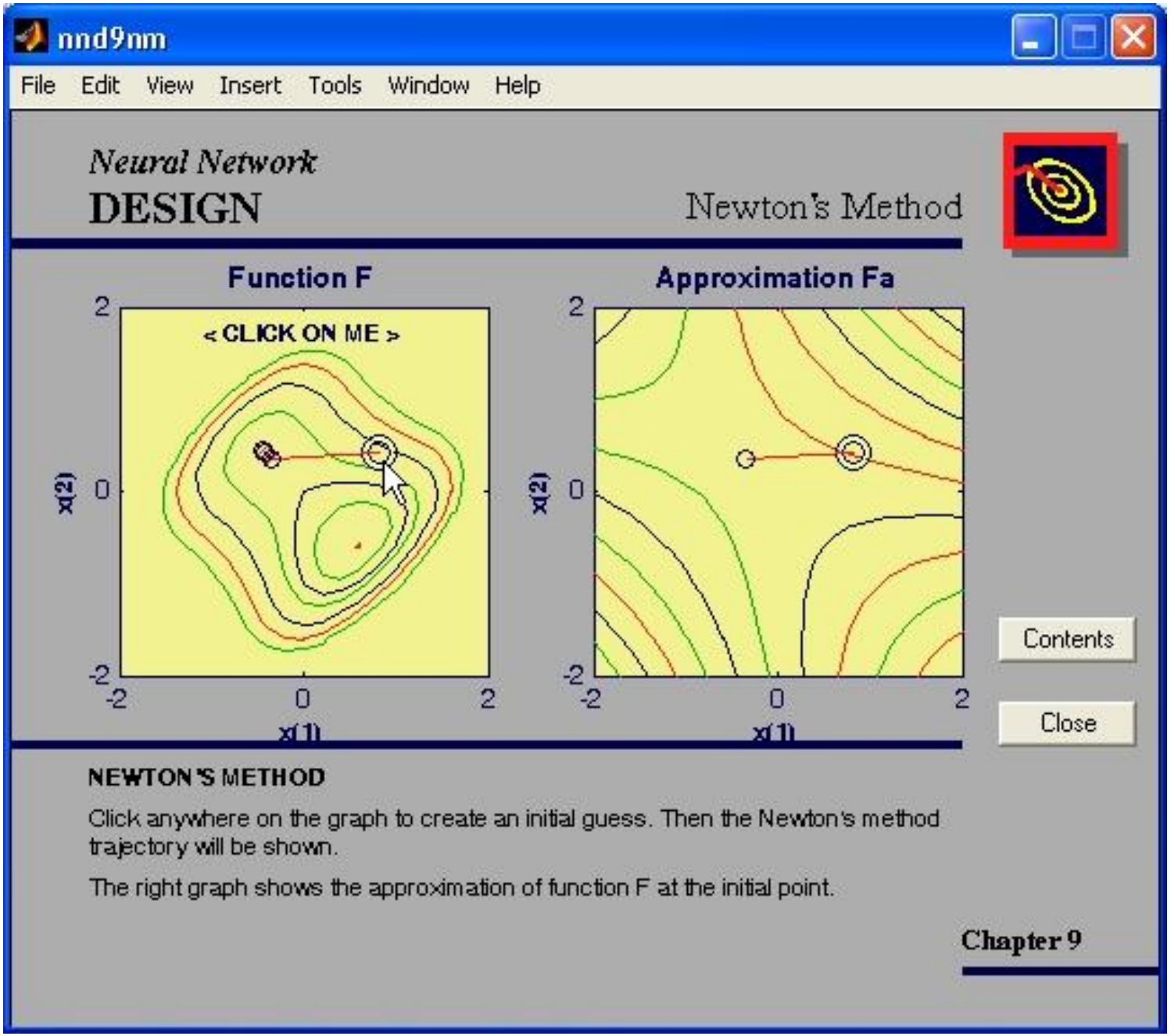

nnd2n2
File Edit View Insert Tools Window Help
Neural Network DESIGN
Two-Input Neuron
Input
Linear Neuron
p(1)
p(2)
w(1,1)
w(1,2)
b
F
Purelin
n
a
1
0
-1
-2
2
6
4
-4
-6
Alter the input values by clicking & dragging the triangle indicators.
Alter the weights and bias in the same way. Use the menu to pick a transfer function.
Pick the transfer function with the F menu.
The net input and the output will respond to each change.
Contents
Close
a = purelin(w*p+b)
Chapter 2

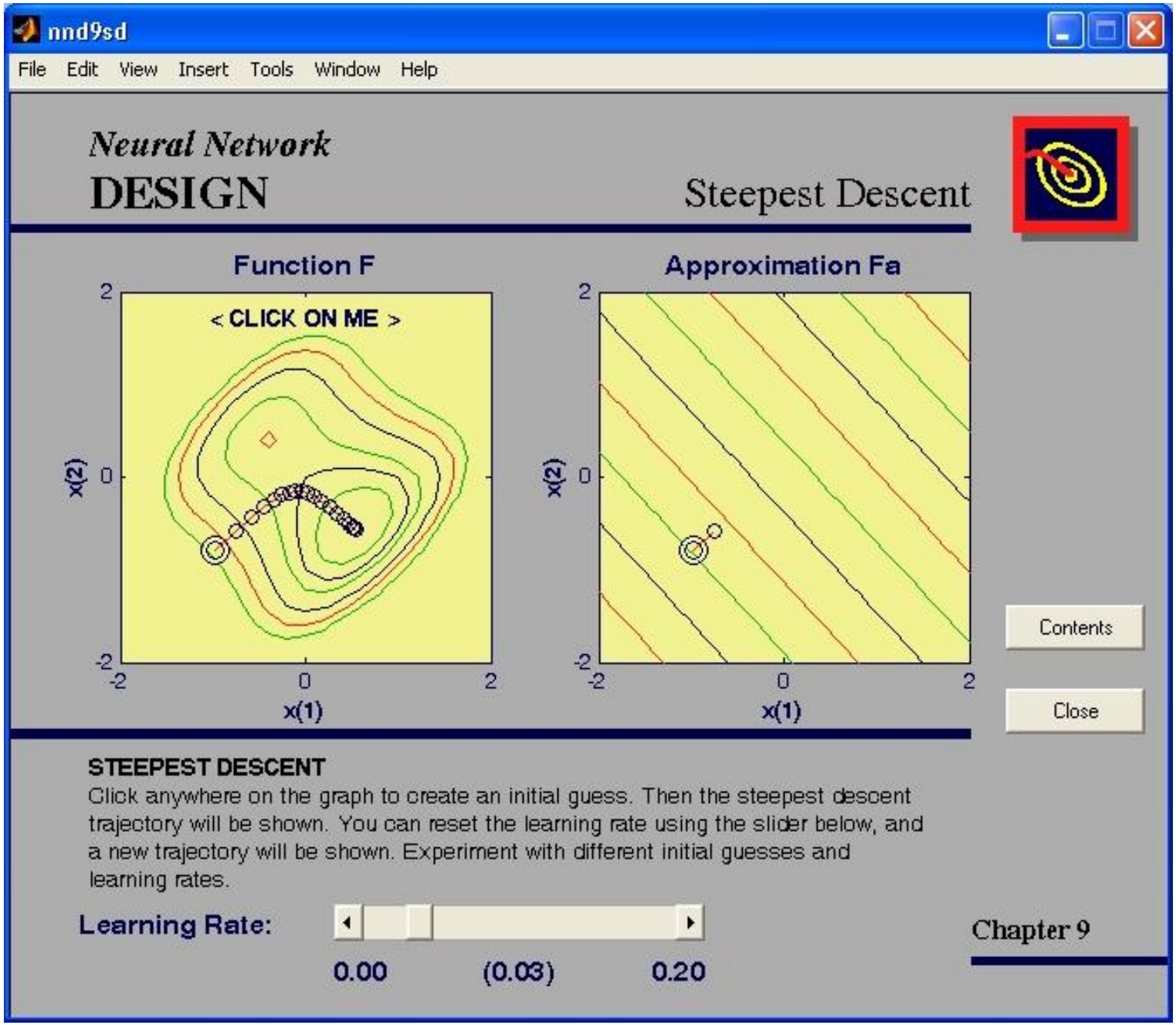

nnd3pc
File Edit View Insert Tools Window Help
Neural Network DESIGN
Perceptron Classification
W = [0 1 0]
b = 0
Input Space
weight
1
-1
1
texture
-1
-1
shape
1
Click [Go] to send a fruit down the belt to be classified by a perceptron network.
The calculations for the perceptron will appear to the left.
Go
Clear
Contents
Close
SHAPE: ?
TEXTURE: ?
WEIGHT: ?
Fruit
Neural Network
Oranges
Apples
Chapter 3

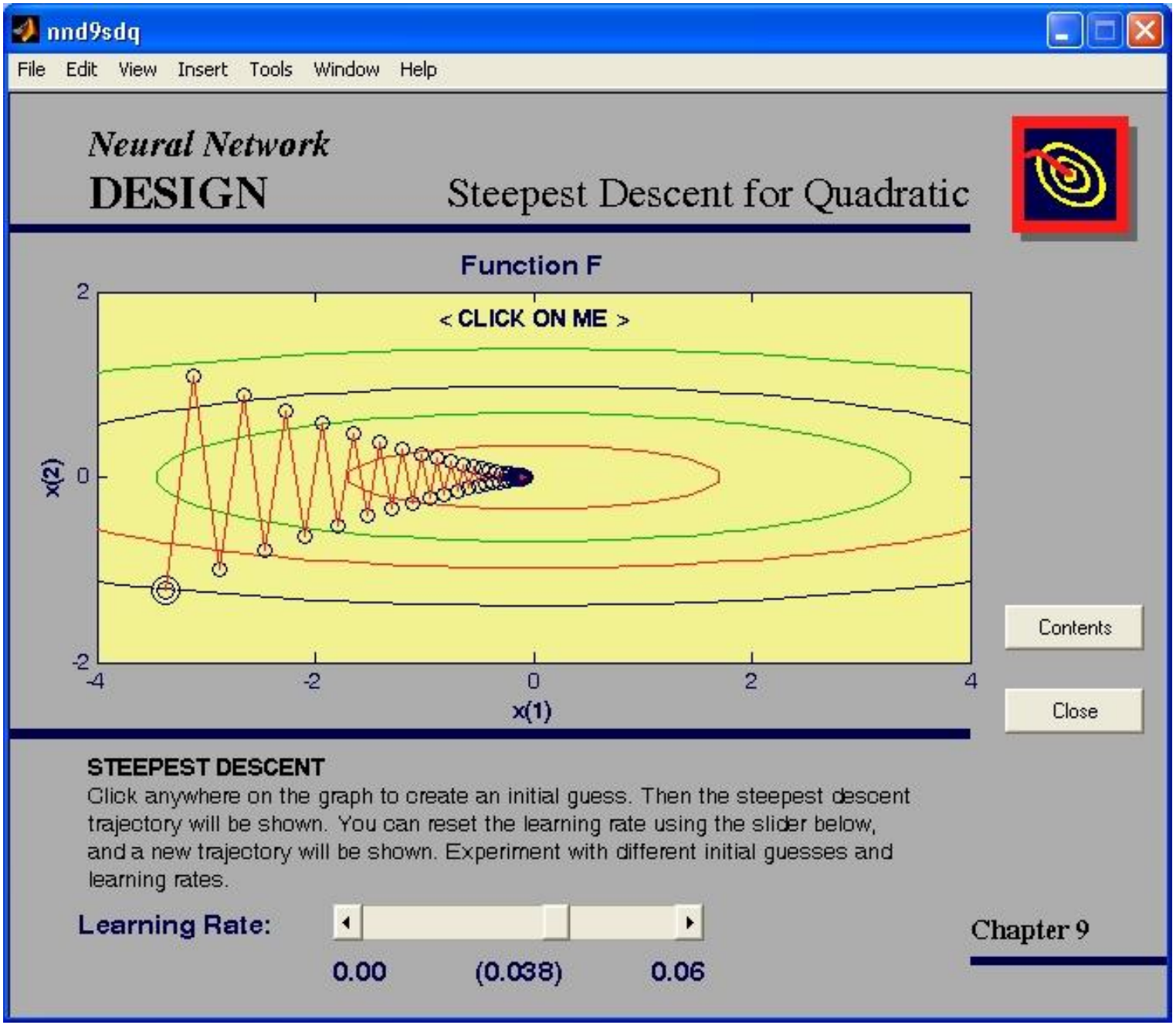

nnd4pr
File Edit View Insert Tools Window Help
Neural Network DESIGN
Perceptron Rule
Learn
Train
Random
Bias
No Bias
3
2
1
0
-1
-2
-3
-3 -2 -1 0 1 2 3
Click [Learn] to apply the perceptron rule to a single vector.
Click [Train] to apply the rule up to 5 times.
Click [Random] to set the weights to random values.
Drag the white and black dots to define different problems.
Contents
Close
W= [-3.5 -1.8]
b= [1]
Chapter 4

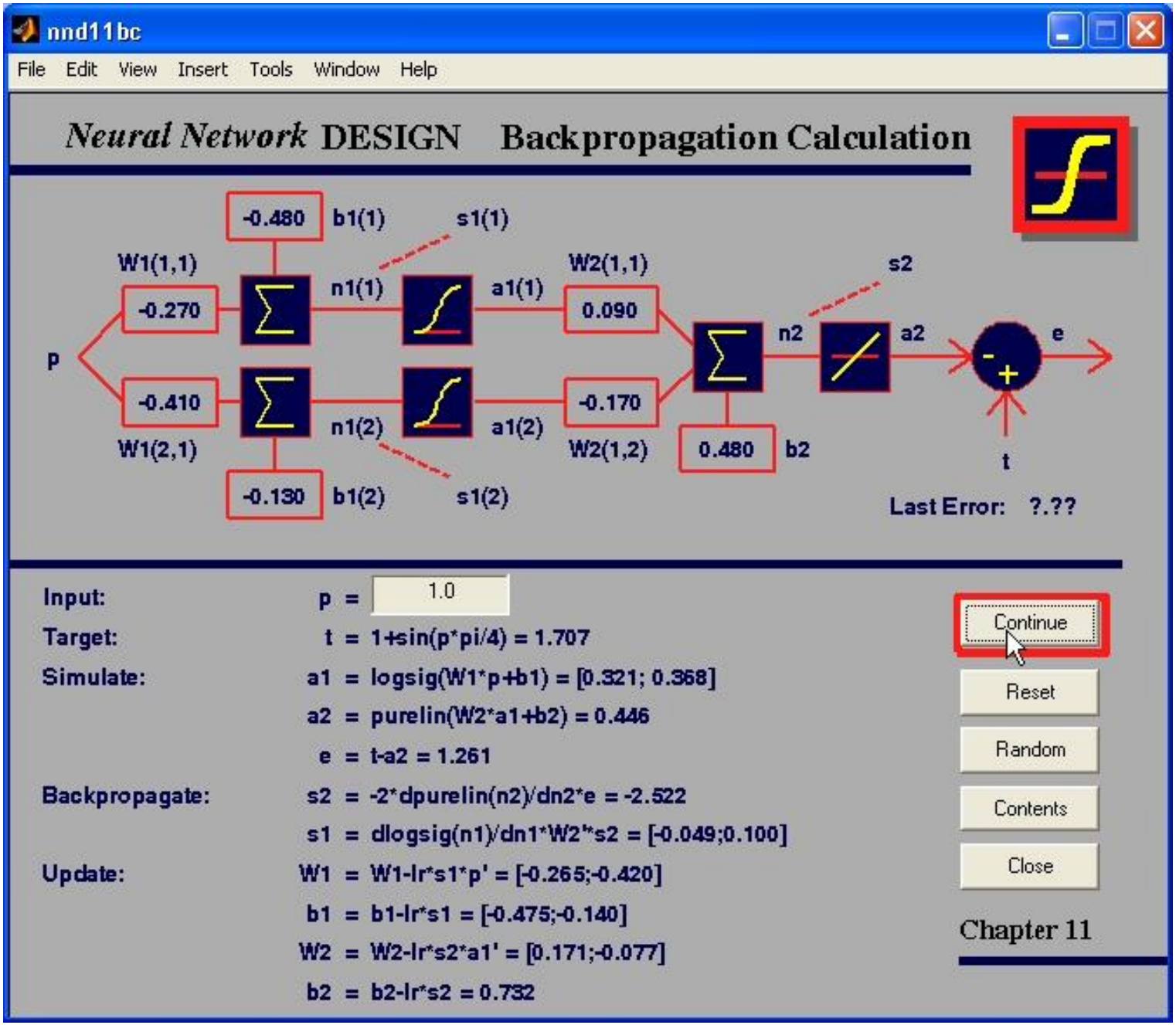

nnd8qf
File Edit View Insert Tools Window Help
Neural Network DESIGN
Quadratic Function
Function F
Function F
x(2)
x(1)
F(x)
Change the values of the Hessian matrix A, the vector d, and the constant c. Then click [Update] to see the new function.
Note that the Hessian matrix A will always be symmetric.
F(x) = 1/2*x'Ax + d'x + c
A=
1.5
-0.7
-0.7
1
d=
0.25
0.25
c=
1
Update
Contents
Close
Chapter 8

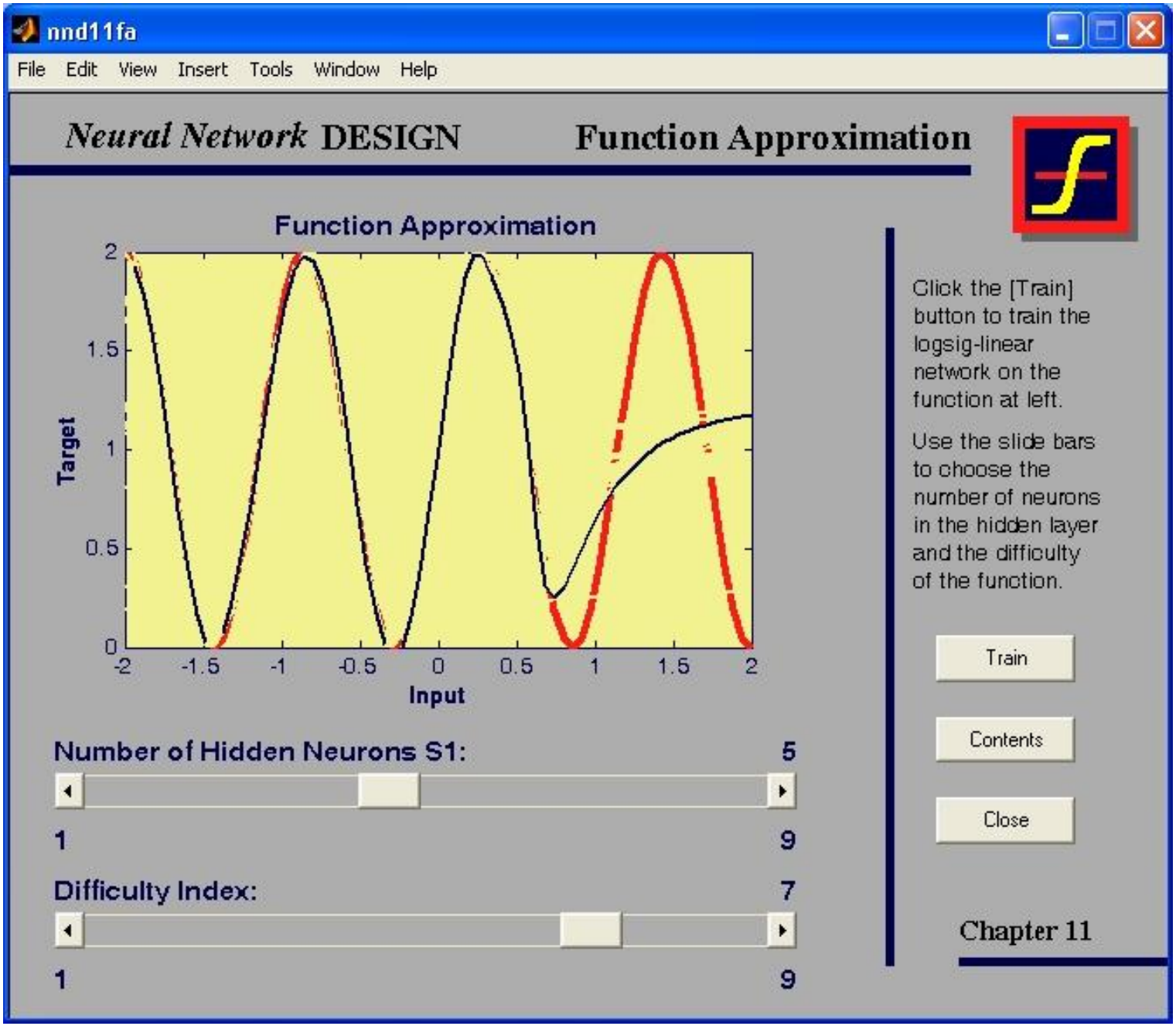

nnd9mc
File Edit View Insert Tools Window Help
Neural Network
DESIGN
Comparison of Methods
Steepest Descent
Conjugate Gradient
<CLICK ON ME>
<CLICK ON ME>
x(2)
x(1)
2
0
-2
Contents
Close
COMPARISON OF METHODS
Click in either graph to create an initial search point.
Then watch the two algorithms attempt to find the minima.
The two algorithms are:
Steepest Descent using line search
Conjugate Gradient using line search
Chapter 9

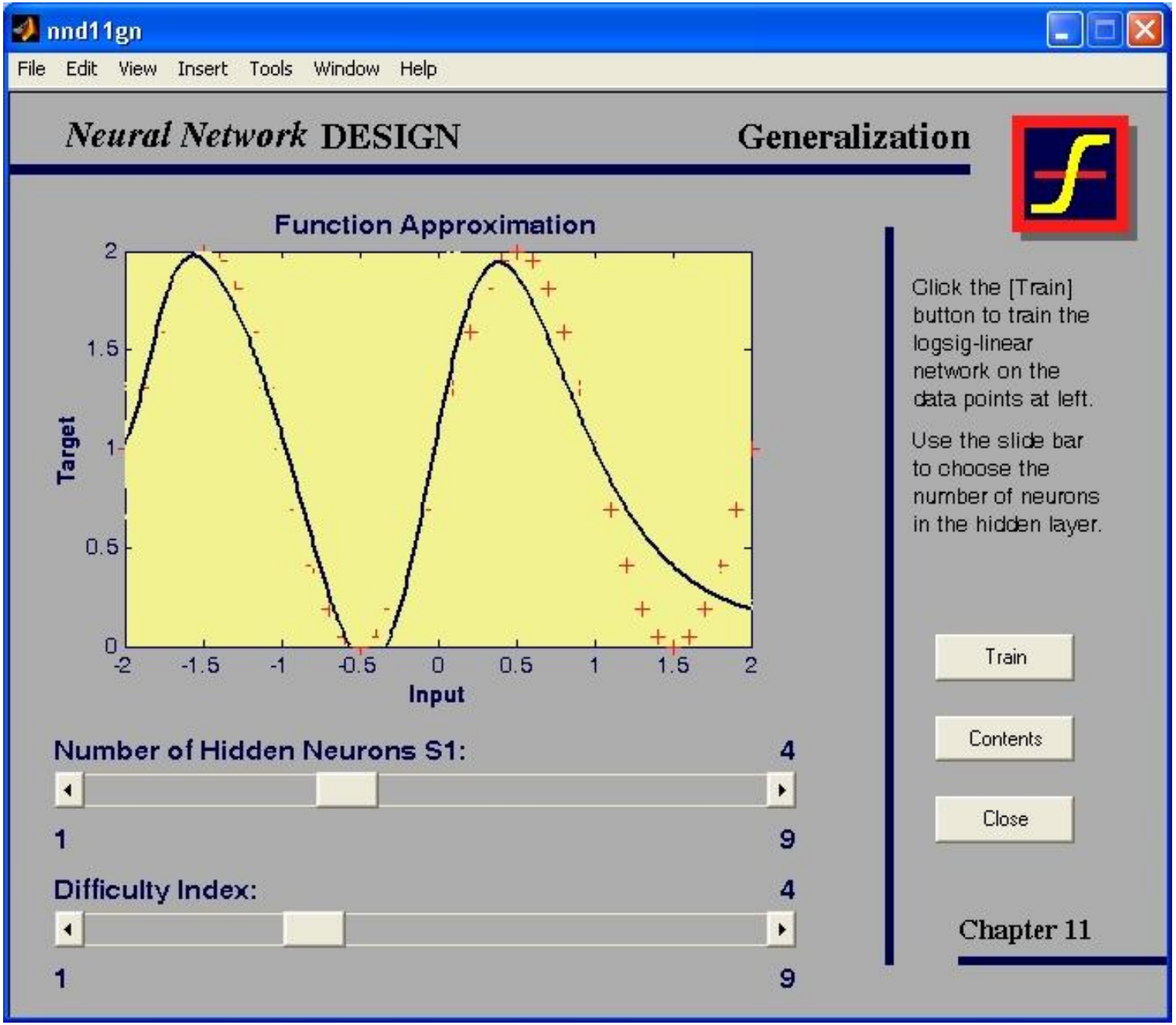

nnd9nm
File Edit View Insert Tools Window Help
Neural Network
DESIGN
Newton's Method
Function F
< CLICK ON ME >
Approximation Fa
x(2)
x(1)
Contents
Close
NEWTON'S METHOD
Click anywhere on the graph to create an initial guess. Then the Newton's method trajectory will be shown.
The right graph shows the approximation of function F at the initial point.
Chapter 9

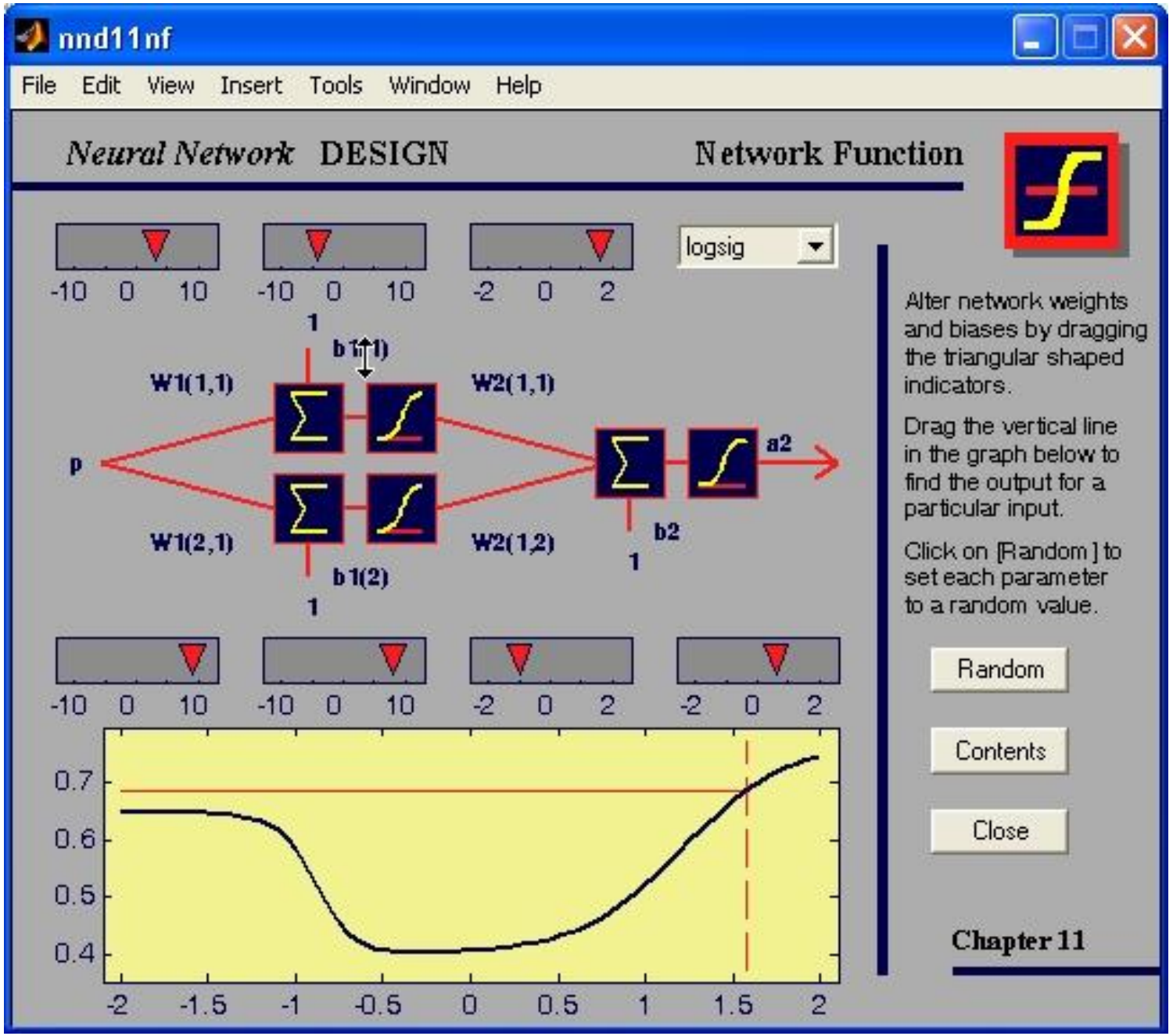

nnd9sd
File Edit View Insert Tools Window Help
Neural Network
DESIGN
Steepest Descent
Function F
< CLICK ON ME >
Approximation Fa
x(2)
x(1)
Contents
Close
STEEPEST DESCENT
Click anywhere on the graph to create an initial guess. Then the steepest descent trajectory will be shown. You can reset the learning rate using the slider below, and a new trajectory will be shown. Experiment with different initial guesses and learning rates.
Learning Rate:
0.00
(0.03)
0.20
Chapter 9

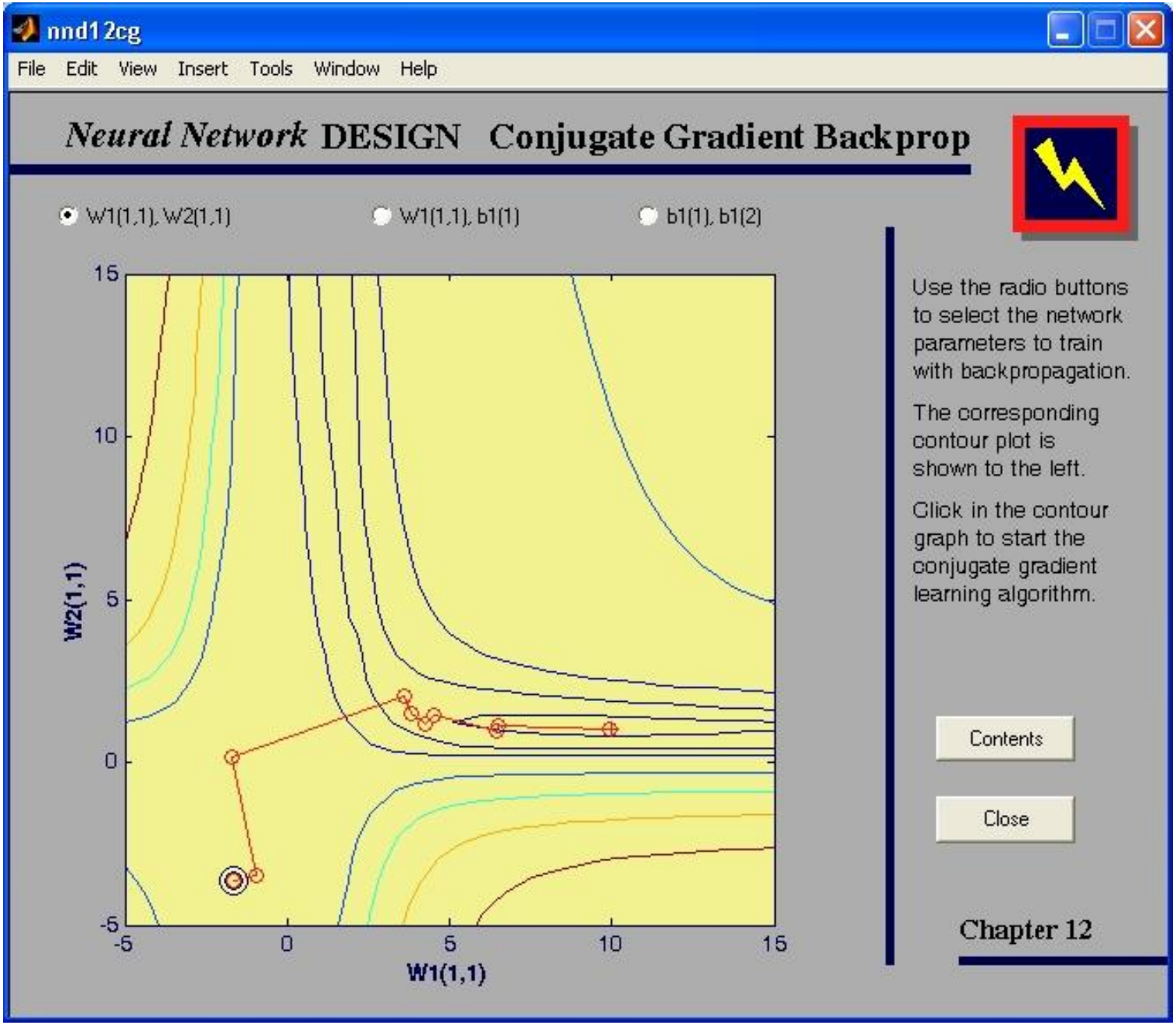

nnd9sdq
File Edit View Insert Tools Window Help
Neural Network
DESIGN
Steepest Descent for Quadratic
Function F
< CLICK ON ME >
x(2)
x(1)
Contents
Close
STEEPEST DESCENT
Click anywhere on the graph to create an initial guess. Then the steepest descent trajectory will be shown. You can reset the learning rate using the slider below, and a new trajectory will be shown. Experiment with different initial guesses and learning rates.
Learning Rate:
0.00
(0.038)
0.06
Chapter 9

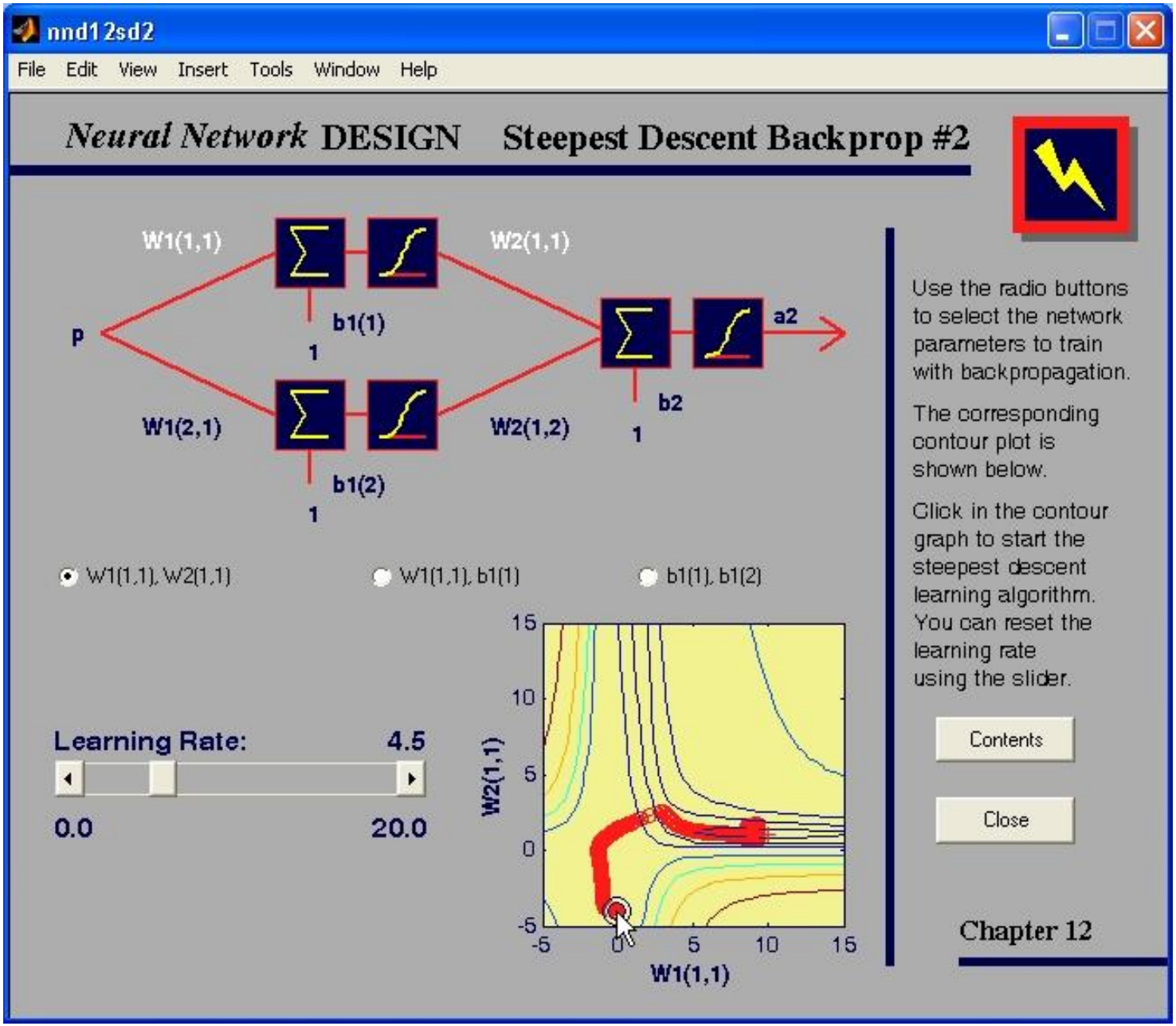

nnd11bc
File Edit View Insert Tools Window Help
Neural Network DESIGN Backpropagation Calculation
-0.480
b1(1)
s1(1)
W1(1,1)
-0.270
n1(1)
a1(1)
W2(1,1)
0.090
s2
n2
a2
e
p
-0.410
W1(2,1)
n1(2)
a1(2)
-0.170
W2(1,2)
0.480
b2
t
-0.130
b1(2)
s1(2)
Last Error: ?.??
Input: p = 1.0
Target: t = 1+sin(p*pi/4) = 1.707
Simulate: a1 = logsig(W1*p+b1) = [0.321; 0.368]
a2 = purelin(W2*a1+b2) = 0.446
e = t-a2 = 1.261
Backpropagate: s2 = -2*dpurelin(n2)/dn2*e = -2.522
s1 = dlogsig(n1)/dn1*W2'*s2 = [-0.049;0.100]
Update: W1 = W1-lr*s1*p' = [-0.265;-0.420]
b1 = b1-lr*s1 = [-0.475;-0.140]
W2 = W2-lr*s2*a1' = [0.171;-0.077]
b2 = b2-lr*s2 = 0.732
Continue
Reset
Random
Contents
Close
Chapter 11


nnd11fa
File Edit View Insert Tools Window Help
Neural Network DESIGN
Function Approximation
Function Approximation
Target
Input
2
1.5
1
0.5
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
Number of Hidden Neurons S1:
5
1
9
Difficulty Index:
7
1
9
Click the [Train] button to train the logsig-linear network on the function at left.
Use the slide bars to choose the number of neurons in the hidden layer and the difficulty of the function.
Train
Contents
Close
Chapter 11


nnd11gn
File Edit View Insert Tools Window Help
Neural Network DESIGN
Generalization
Function Approximation
Target
Input
2
1.5
1
0.5
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
Click the [Train] button to train the logsig-linear network on the data points at left.
Use the slide bar to choose the number of neurons in the hidden layer.
Train
Contents
Close
Number of Hidden Neurons S1:
4
1
9
Difficulty Index:
4
1
9
Chapter 11


nnd11nf
File Edit View Insert Tools Window Help
Neural Network DESIGN
Network Function
logsig
-10 0 10
-10 0 10
-2 0 2
1
b1(1)
W1(1,1)
W2(1,1)
p
a2
W1(2,1)
W2(1,2)
b2
1
b1(2)
1
-10 0 10
-10 0 10
-2 0 2
-2 0 2
0.7
0.6
0.5
0.4
-2 -1.5 -1 -0.5 0 0.5 1 1.5 2
Alter network weights and biases by dragging the triangular shaped indicators.
Drag the vertical line in the graph below to find the output for a particular input.
Click on [Random] to set each parameter to a random value.
Random
Contents
Close
Chapter 11


nnd12cg
File Edit View Insert Tools Window Help
Neural Network DESIGN Conjugate Gradient Backprop
W1(1,1), W2(1,1)
W1(1,1), b1(1)
b1(1), b1(2)
15
10
5
0
-5
W2(1,1)
-5
0
5
10
15
W1(1,1)
Use the radio buttons to select the network parameters to train with backpropagation.
The corresponding contour plot is shown to the left.
Click in the contour graph to start the conjugate gradient learning algorithm.
Contents
Close
Chapter 12


nnd12sd2
File Edit View Insert Tools Window Help
Neural Network DESIGN Steepest Descent Backprop #2
W1(1,1)
W2(1,1)
p
b1(1)
1
a2
b2
1
W1(2,1)
W2(1,2)
b1(2)
1
W1(1,1), W2(1,1)
W1(1,1), b1(1)
b1(1), b1(2)
Learning Rate:
4.5
0.0
20.0
15
10
5
0
-5
W2(1,1)
-5
0
5
10
15
W1(1,1)
Use the radio buttons to select the network parameters to train with backpropagation.
The corresponding contour plot is shown below.
Click in the contour graph to start the steepest descent learning algorithm. You can reset the learning rate using the slider.
Contents
Close
Chapter 12