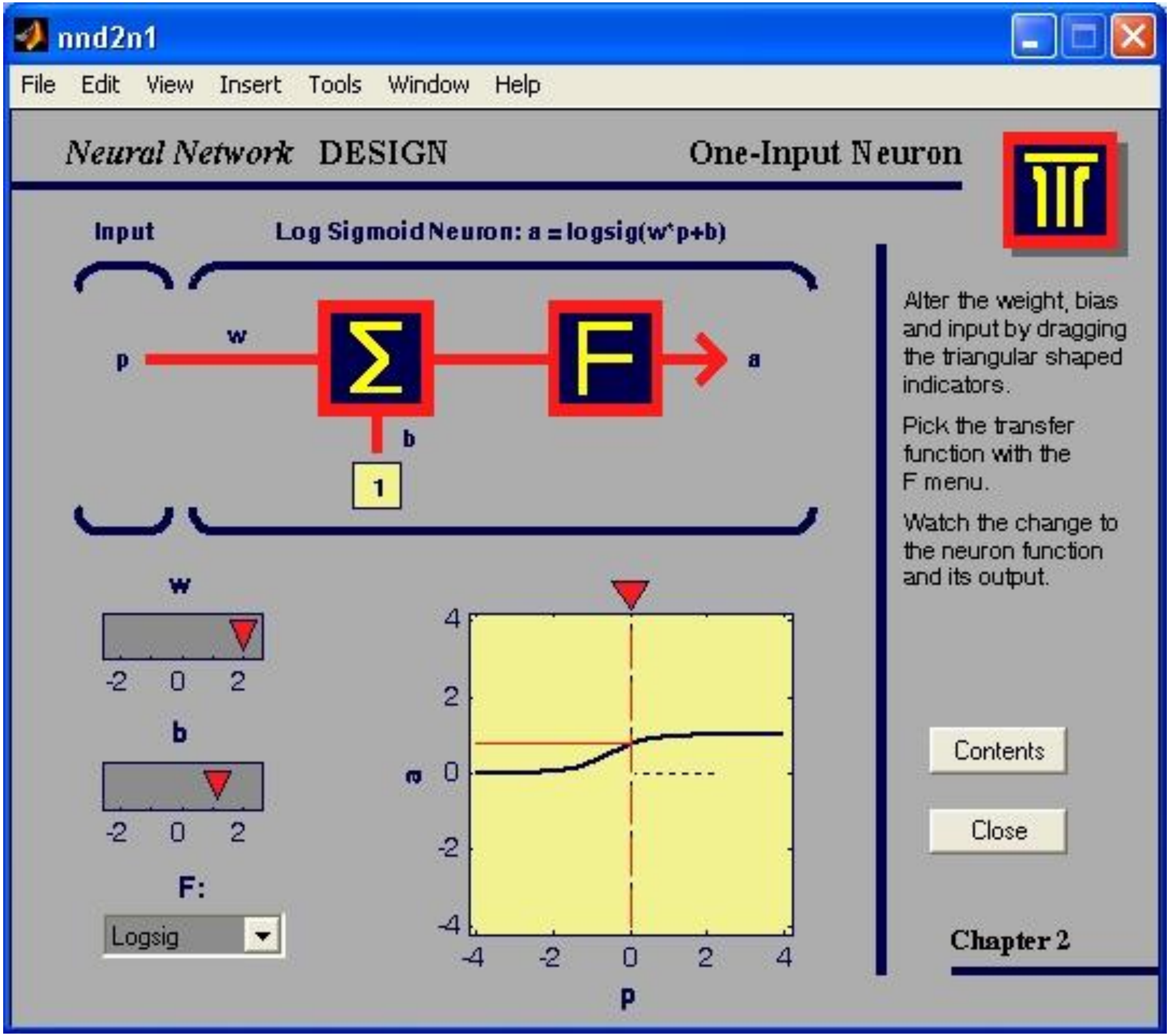
Introduction to Machine Learning

Matlab Neural Network Demos

Barnabás Póczos



Neural Network DESIGN

One-Input Neuron

Log Sigmoid Neuron: a = logsig(w*p+b)

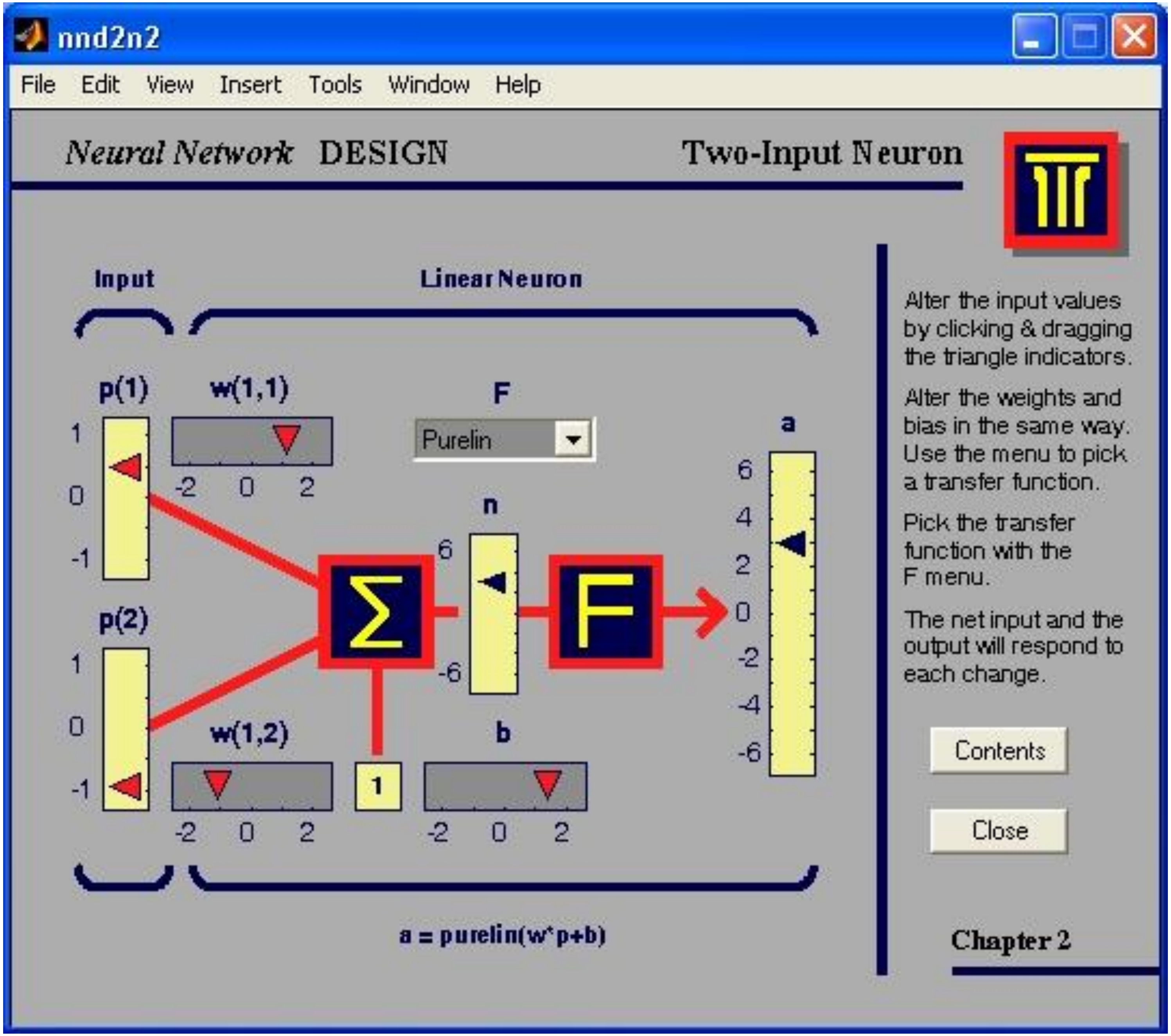
Alter the weight, bias and input by dragging the triangular shaped indicators.

Pick the transfer function with the F menu.

Watch the change to the neuron function and its output.

F: Logsig

Chapter 2



F: Purelin

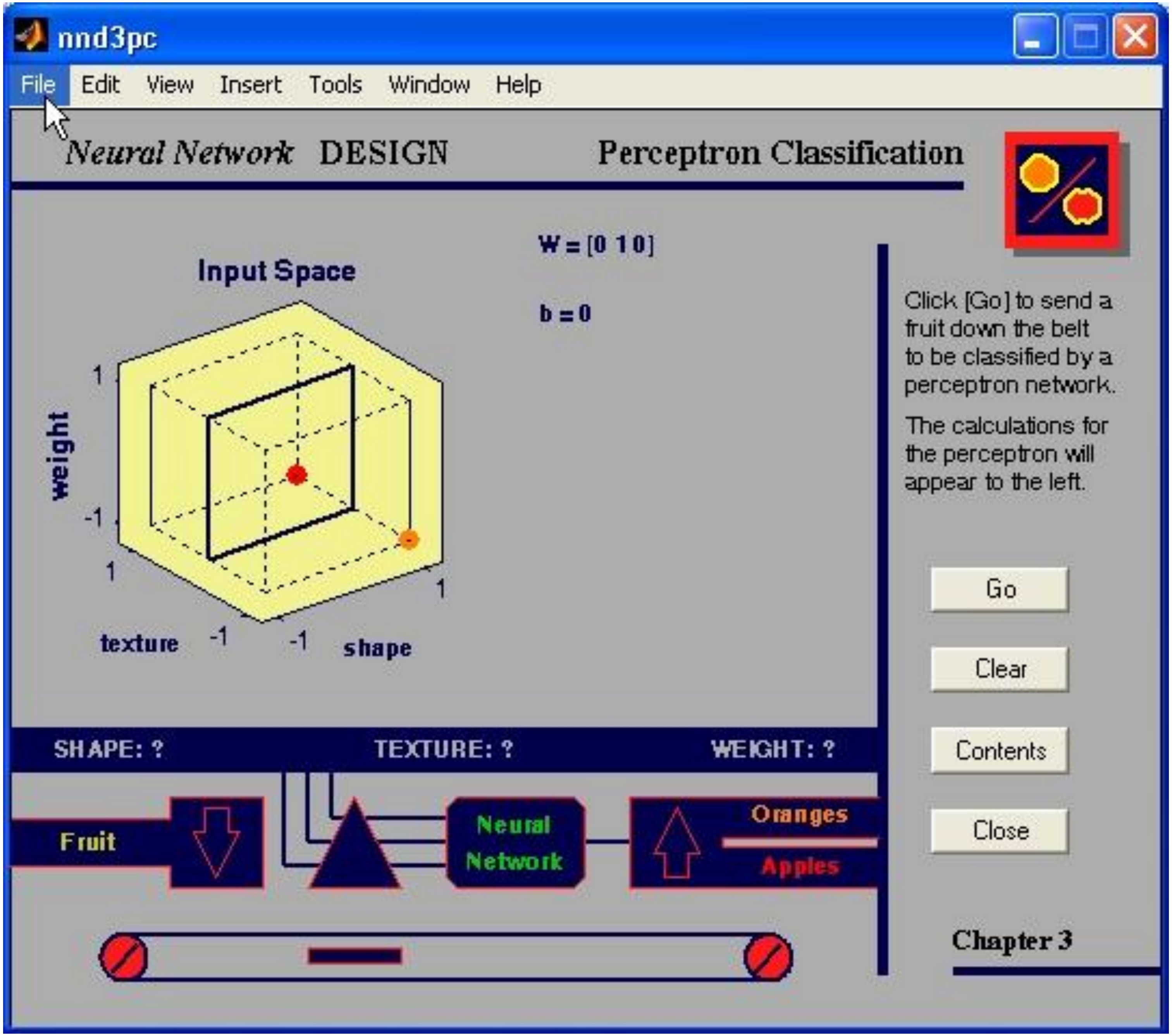
Alter the input values by clicking & dragging the triangle indicators.

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.

a = purelin(w*p+b)



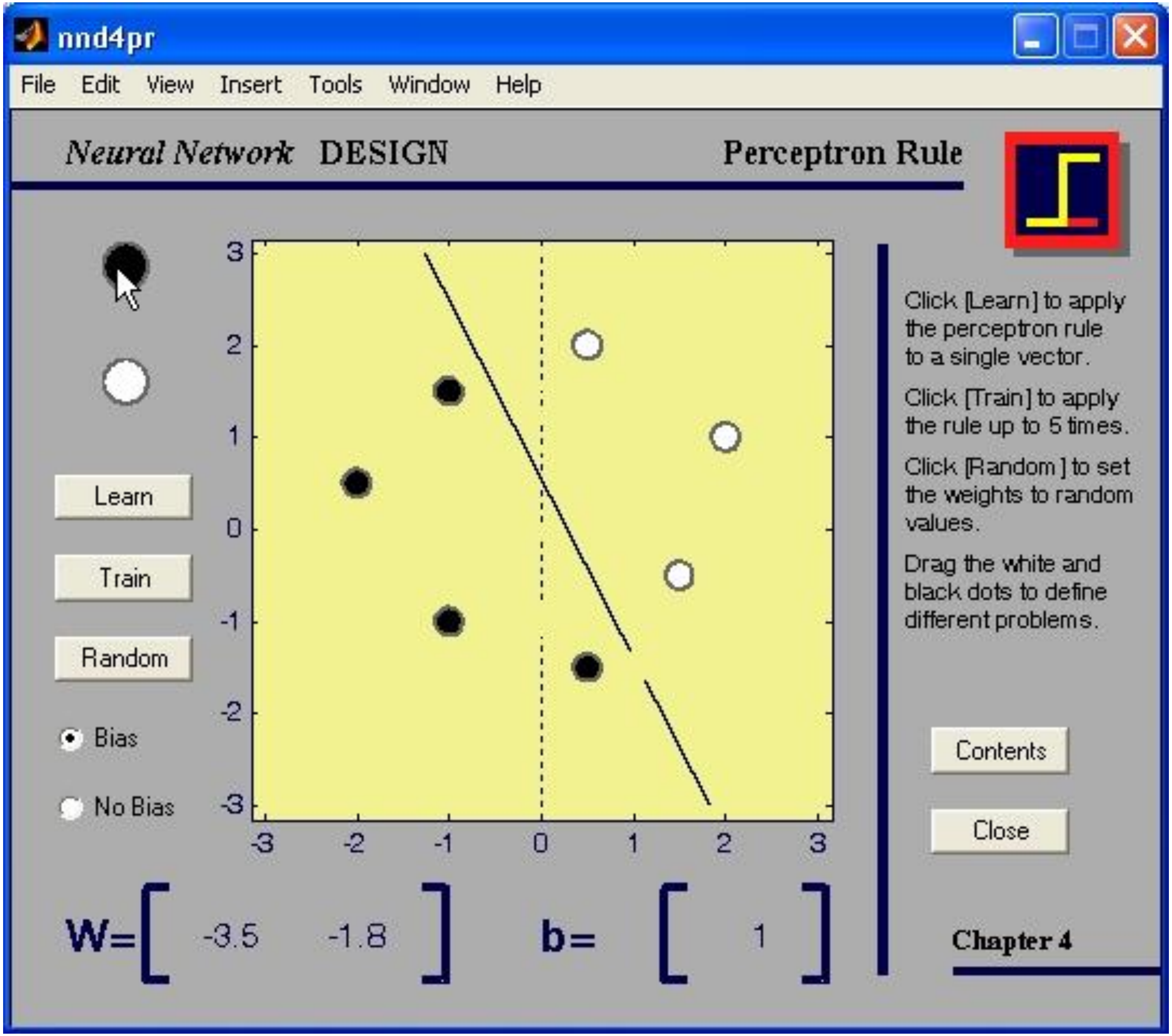
W = [0 1 0]

b = 0

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.

SHAPE: ? TEXTURE: ? WEIGHT: ?



Click [Learn] to apply the perceptron rule to a single vector.

Click [Train] to apply the rule up to 5 times.

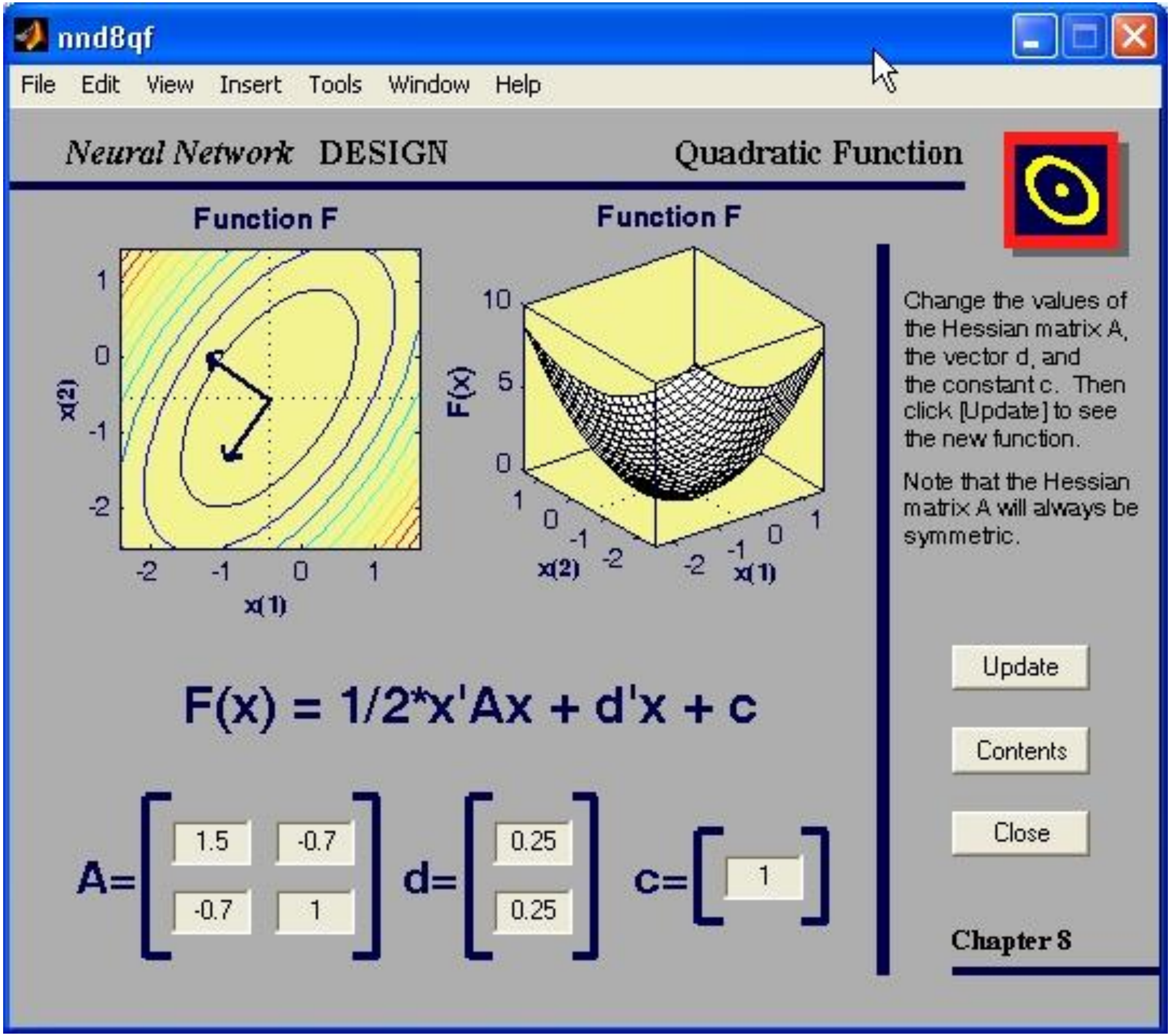
Click [Random] to set the weights to random values.

Drag the white and black dots to define different problems.

● Bias

○ No Bias

$W = [\ -3.5 \quad -1.8\ ]$ $\quad b = [\ 1\ ]$

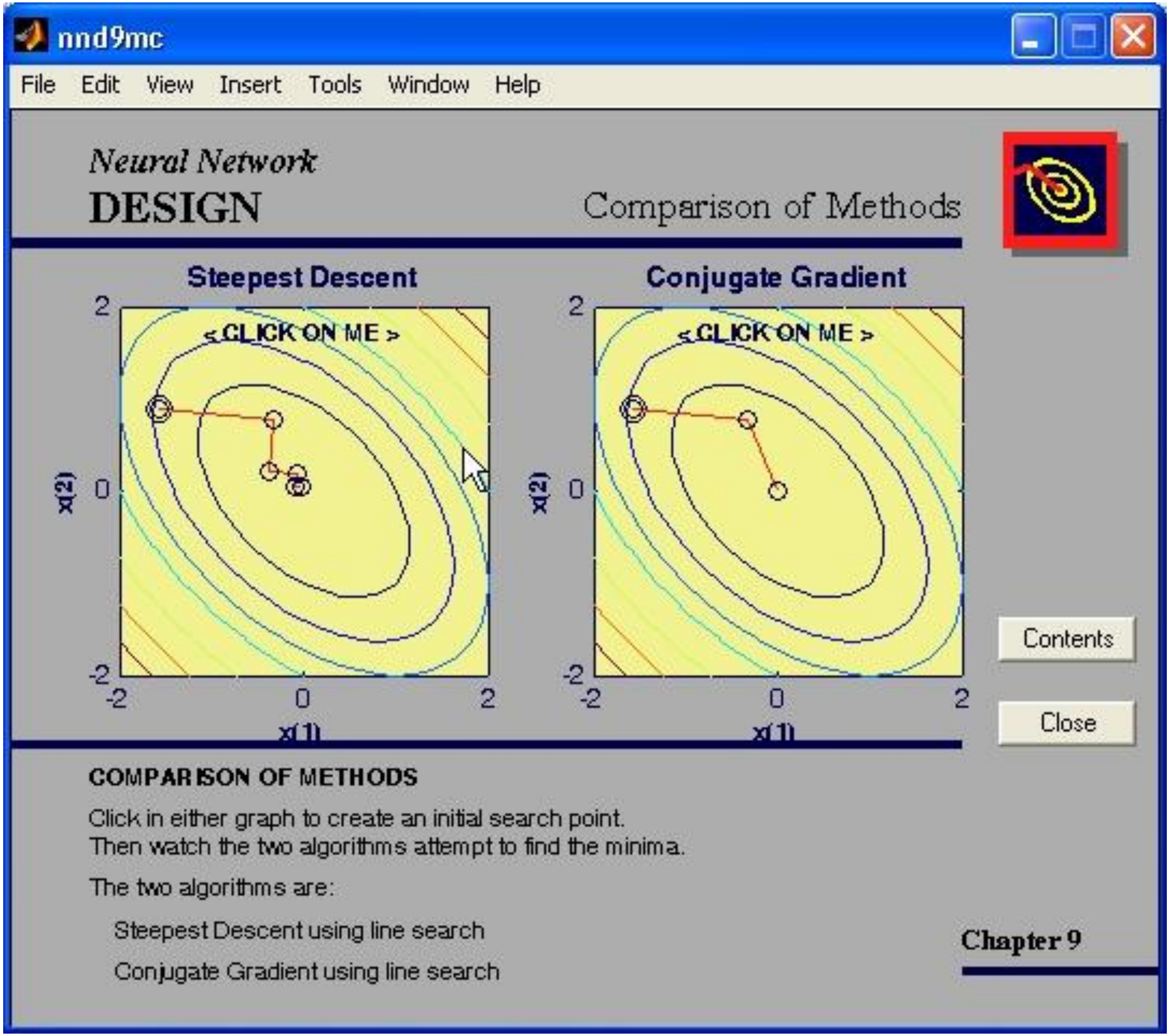


Change the values of the Hessian matrix A, the vector d, and the constant c. Then click [Update] to see the new function.

Note that the Hessian matrix A will always be symmetric.

F(x) = 1/2*x'Ax + d'x + c

$$A=\begin{bmatrix} 1.5 & -0.7 \\ -0.7 & 1 \end{bmatrix} \quad d=\begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix} \quad c=\begin{bmatrix} 1 \end{bmatrix}$$



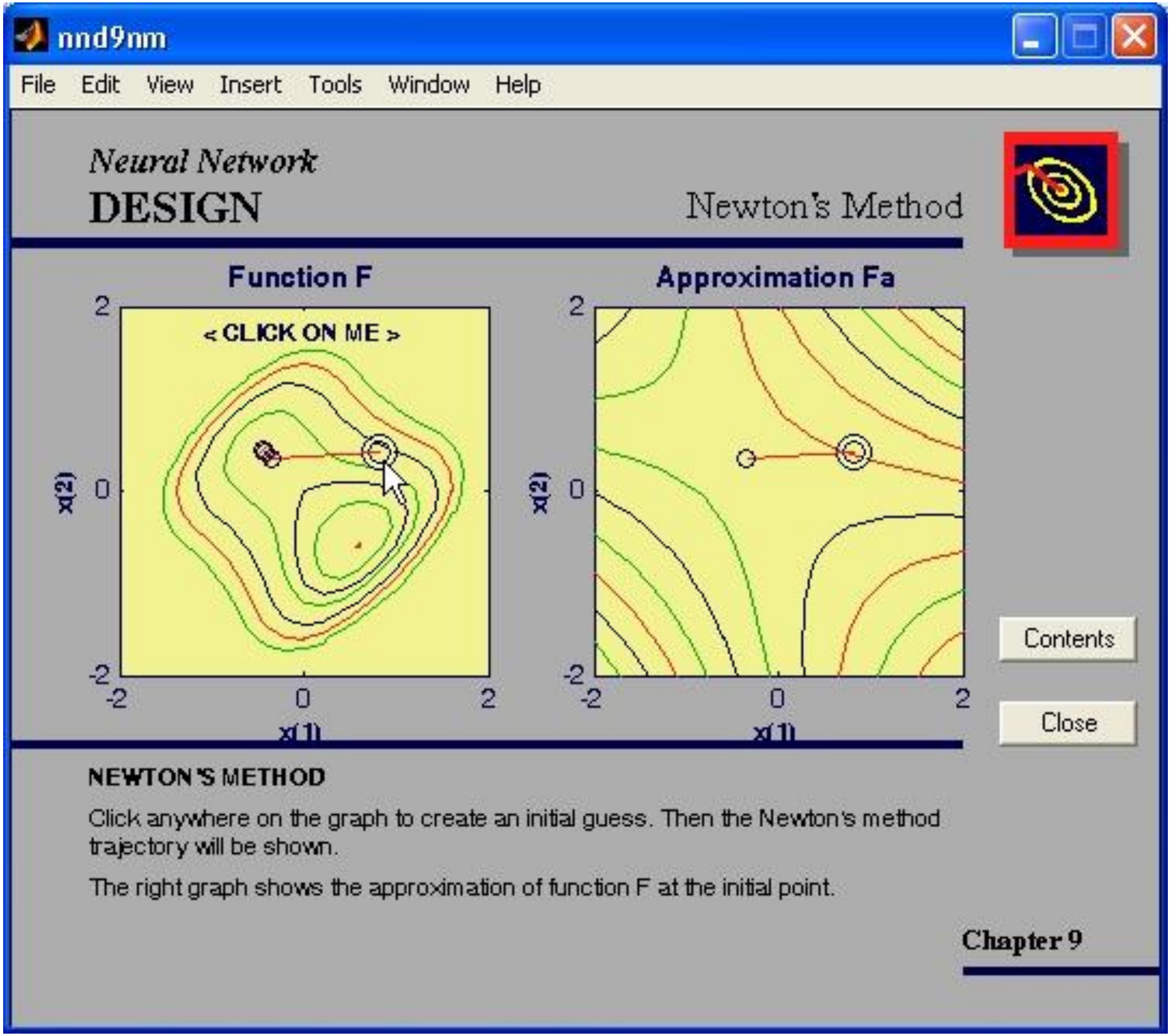
COMPARISON OF METHODS

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

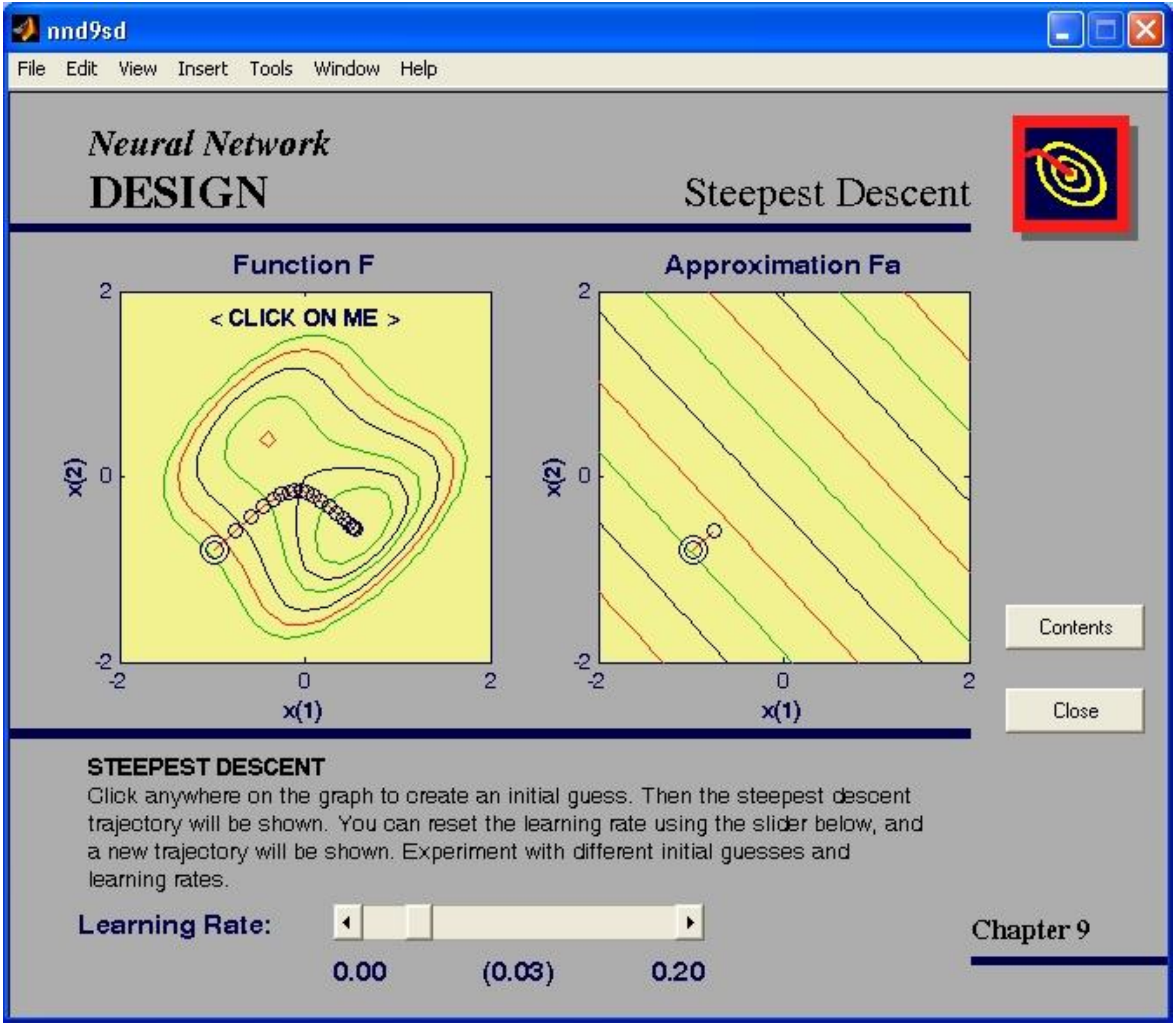
Conjugate Gradient using line search



NEWTON'S METHOD

Click anywhere on the graph to create an initial guess. Then the Newton's method trajectory will be shown.

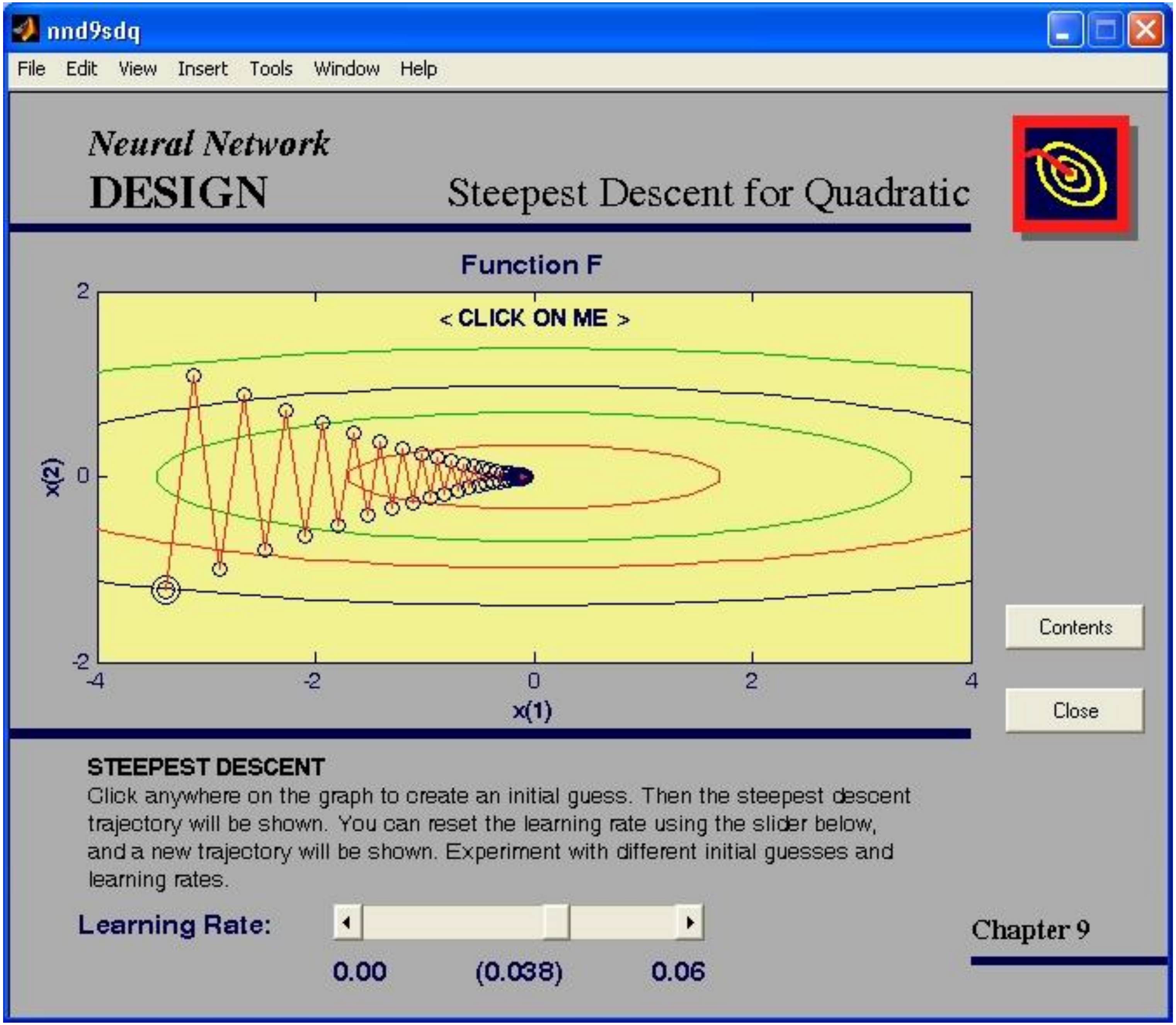
The right graph shows the approximation of function F at the initial point.



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



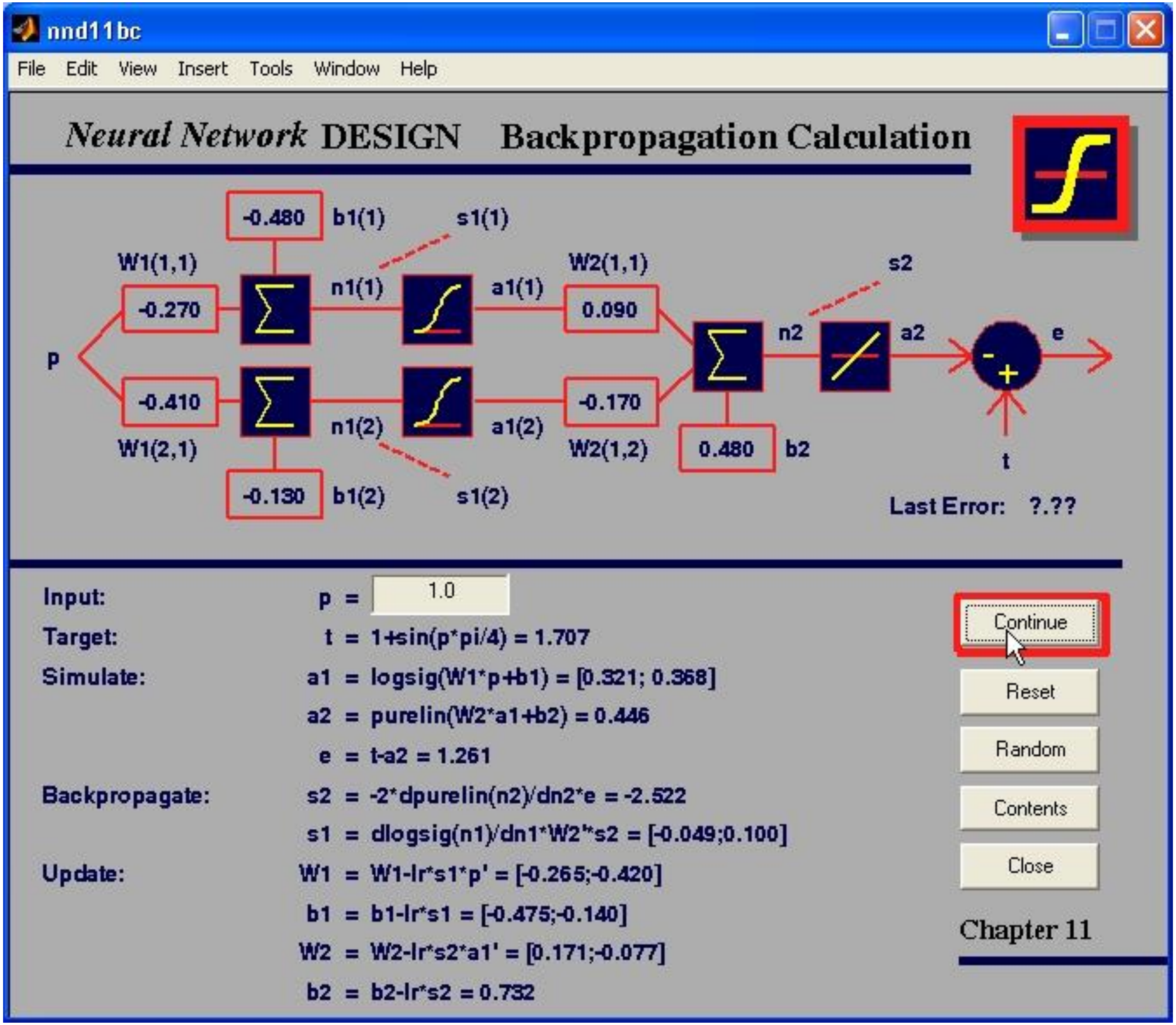
Neural Network
DESIGN

Steepest Descent for Quadratic

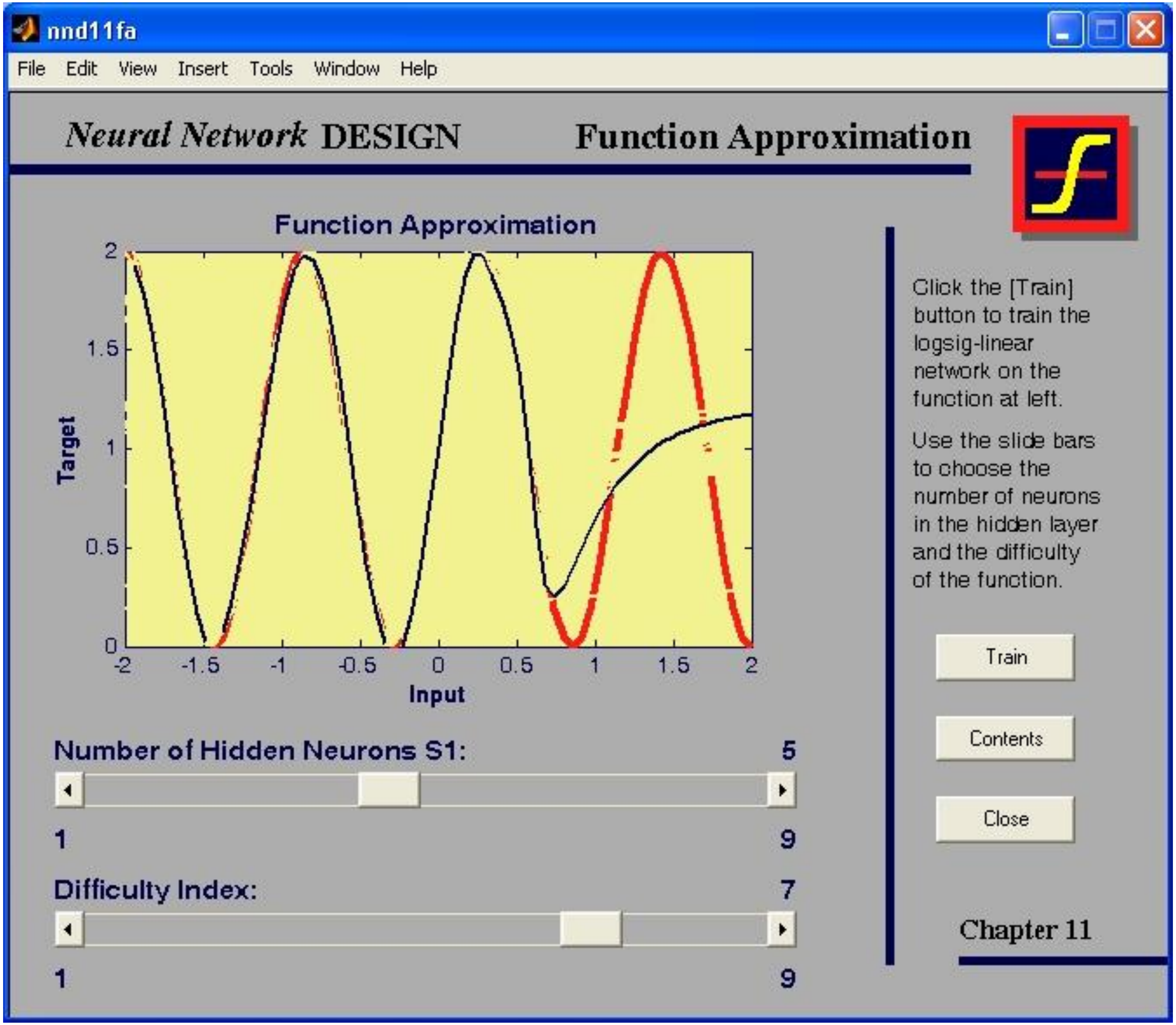
STEEPEST DESCENT

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



```
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
```

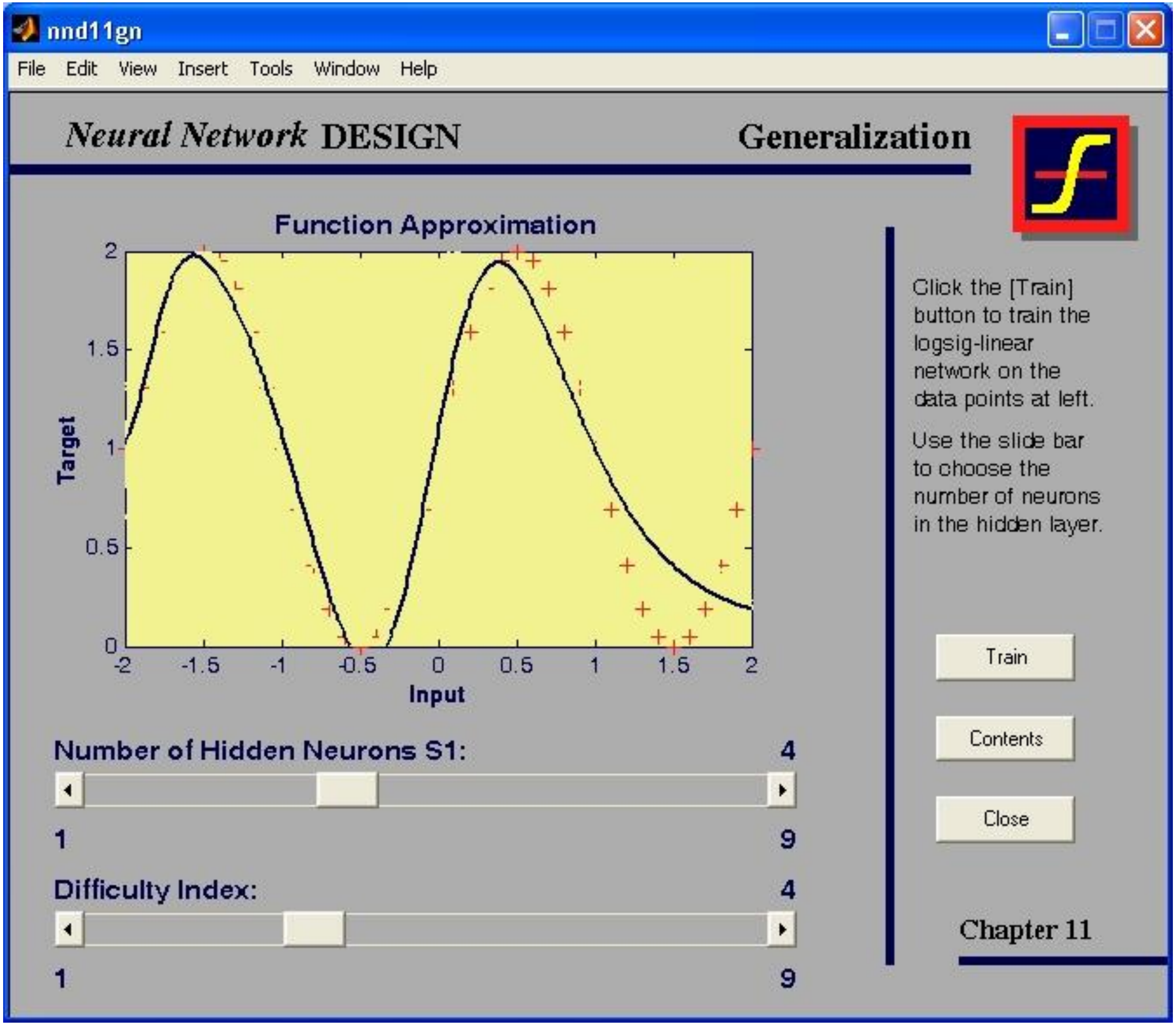


Number of Hidden Neurons S1: 5

Difficulty Index: 7

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.



Click the [Train] button to train the logsig-linear network on the data points at left.

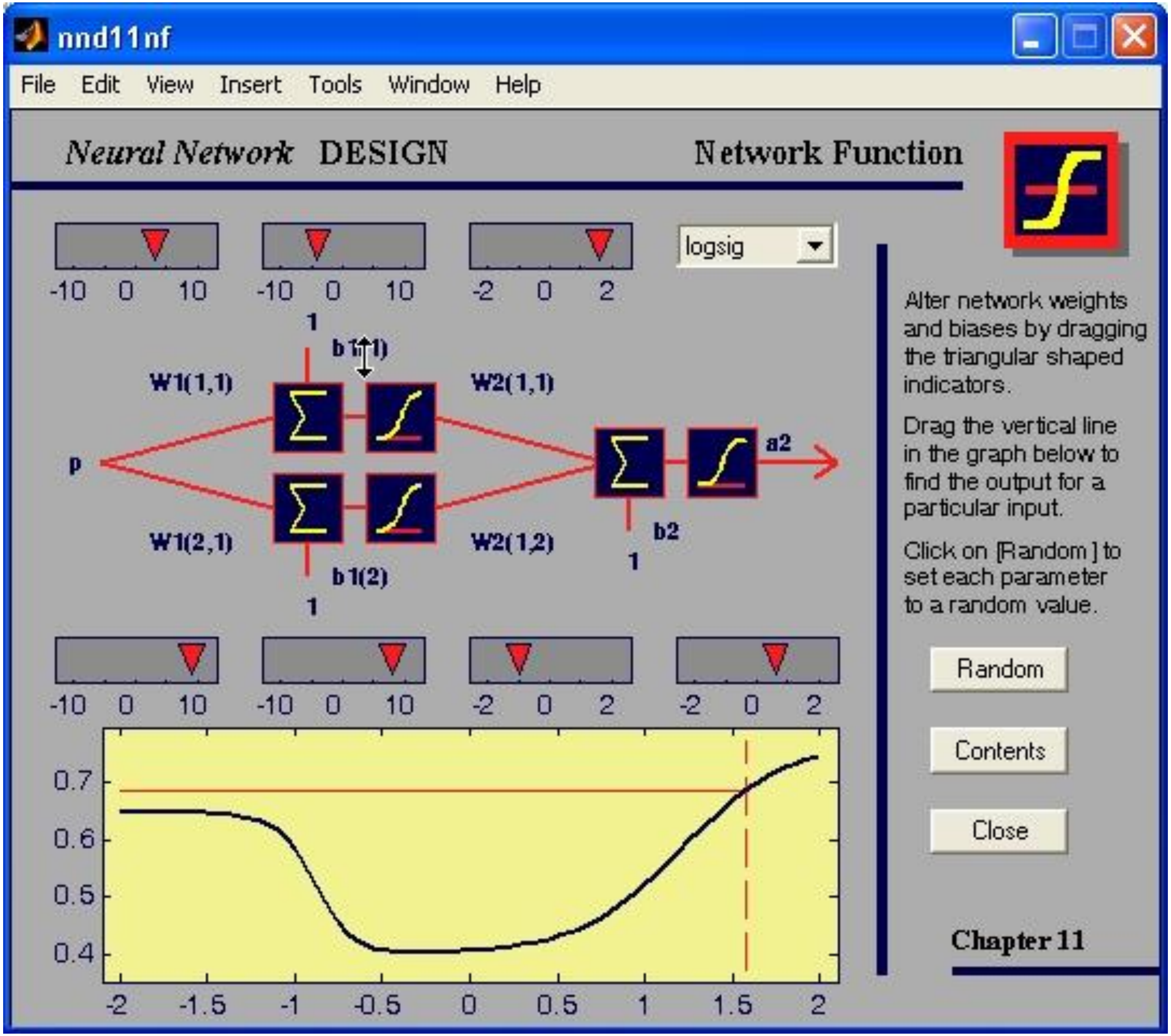
Use the slide bar to choose the number of neurons in the hidden layer.

Number of Hidden Neurons S1: 4

1 9

Difficulty Index: 4

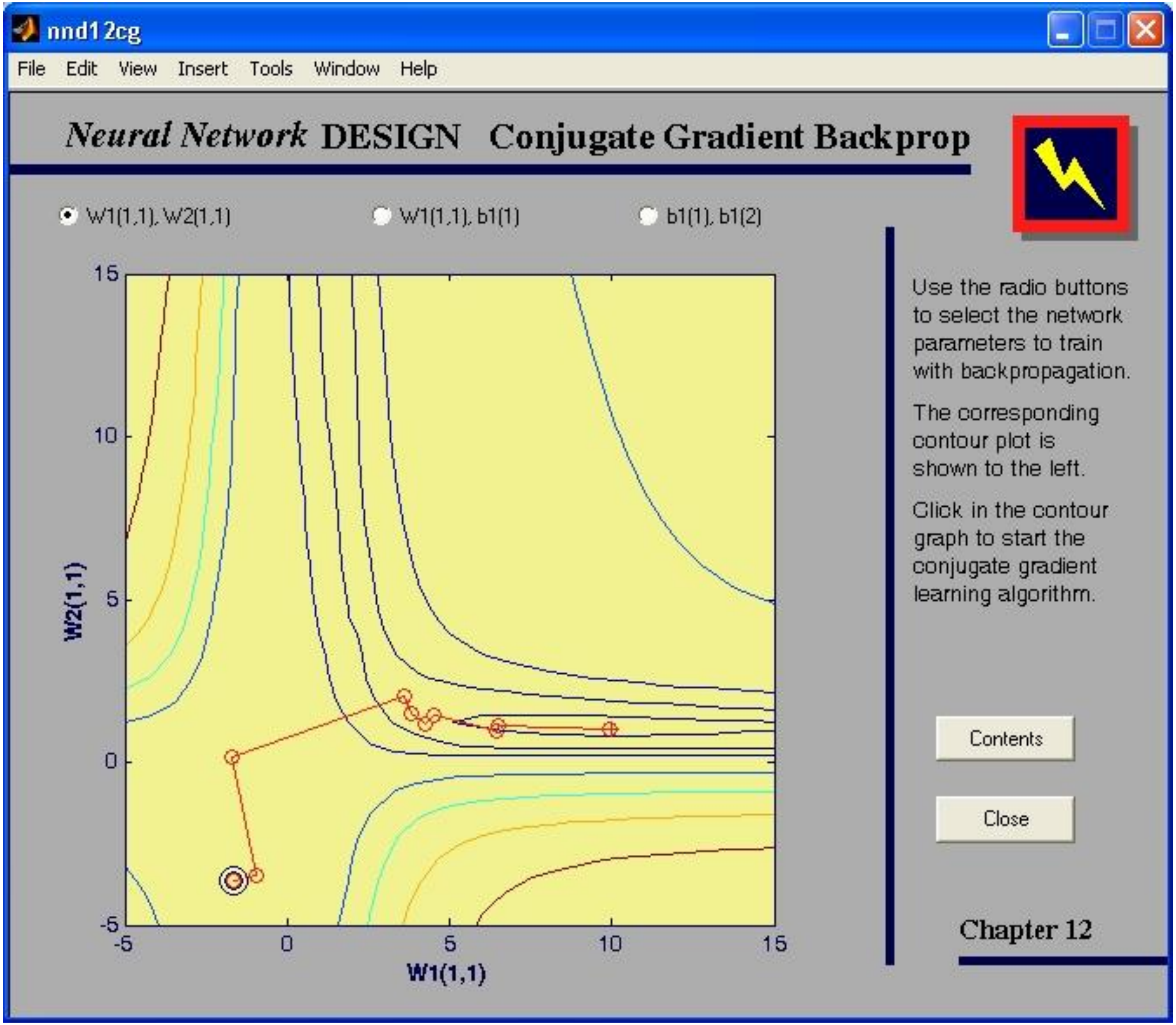
1 9



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.

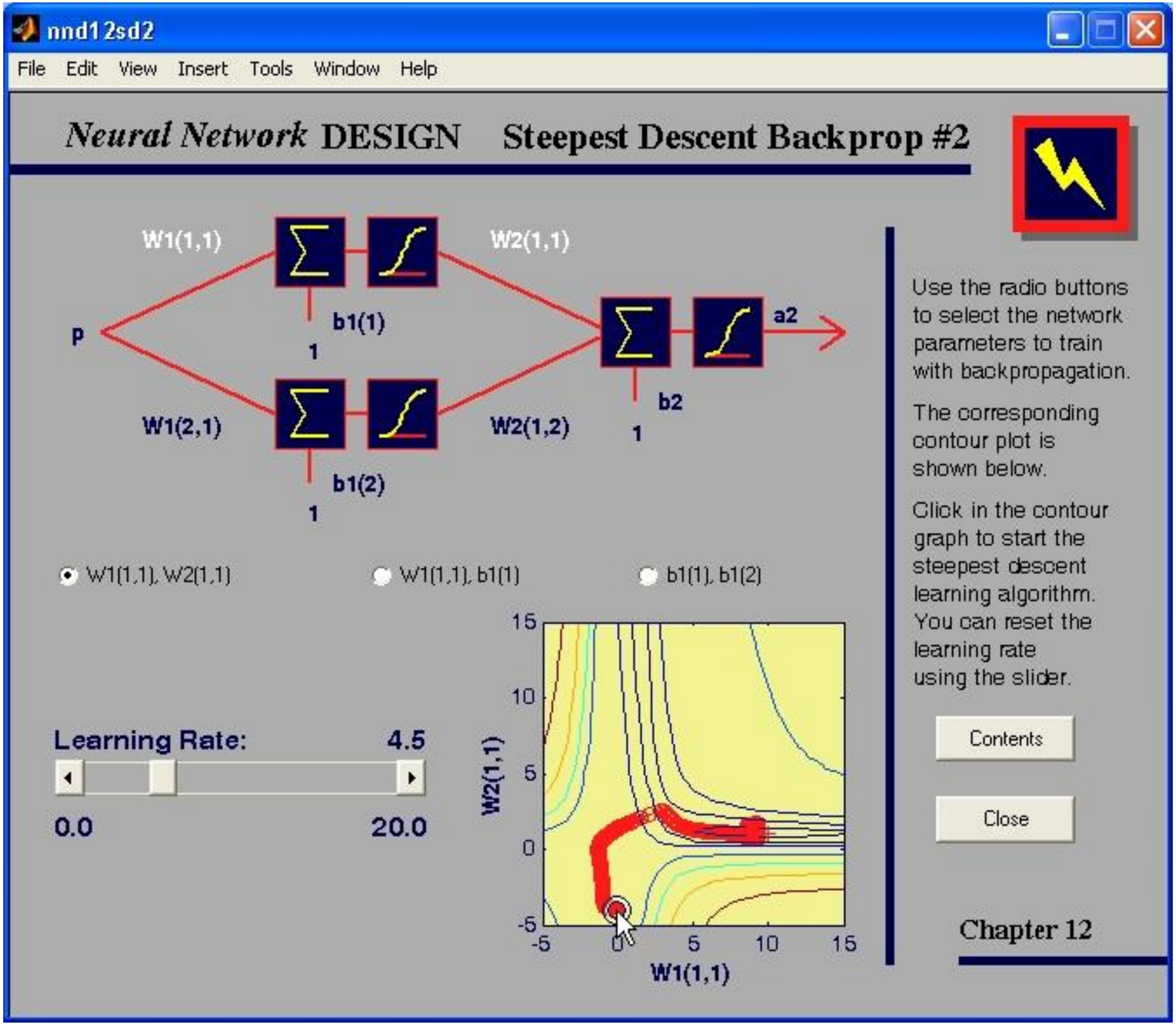
Click on [Random] to set each parameter to a random value.



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.



Neural Network DESIGN Steepest Descent Backprop #2

● W1(1,1), W2(1,1) ○ W1(1,1), b1(1) ○ b1(1), b1(2)

Learning Rate: 4.5

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.