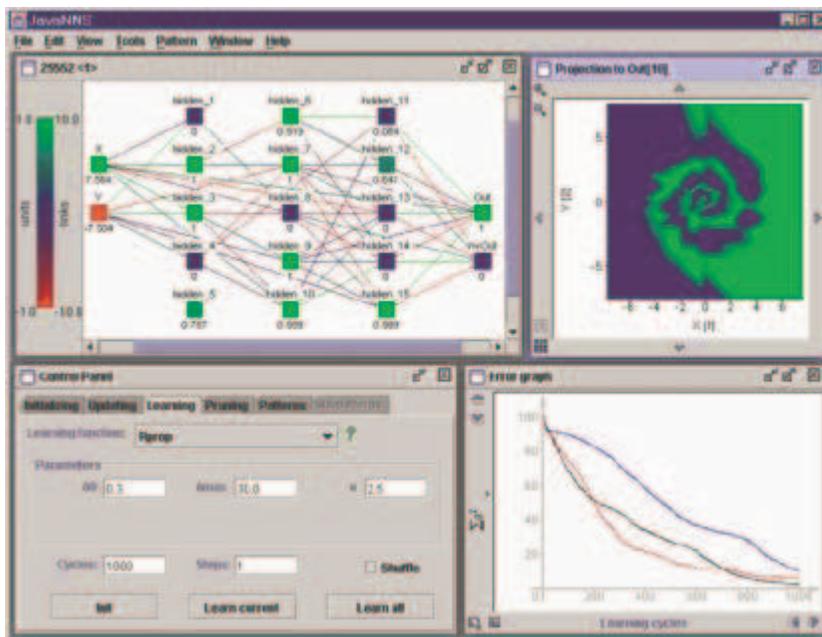


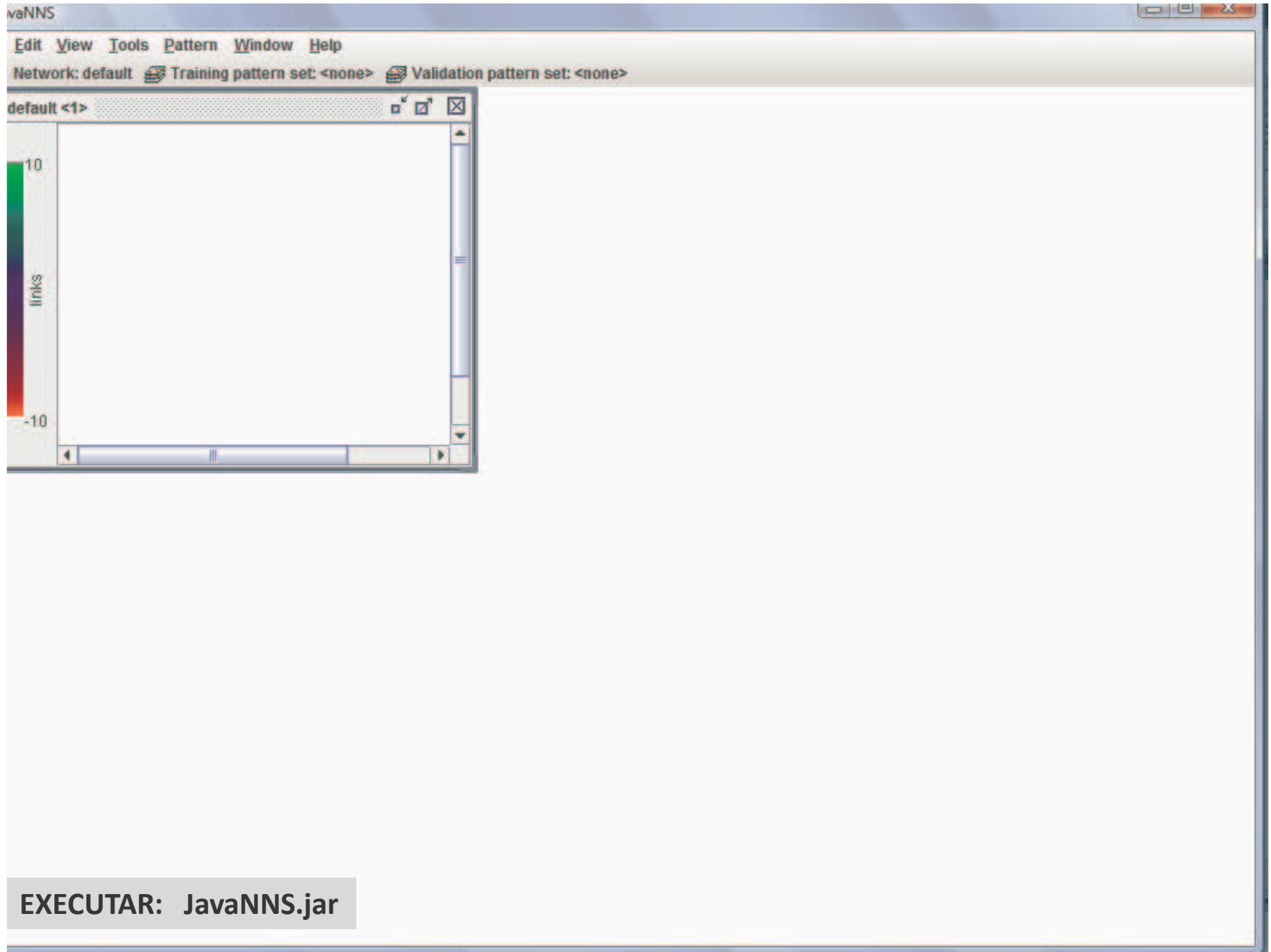
# JAVA NNS

## Passo-a-Passo

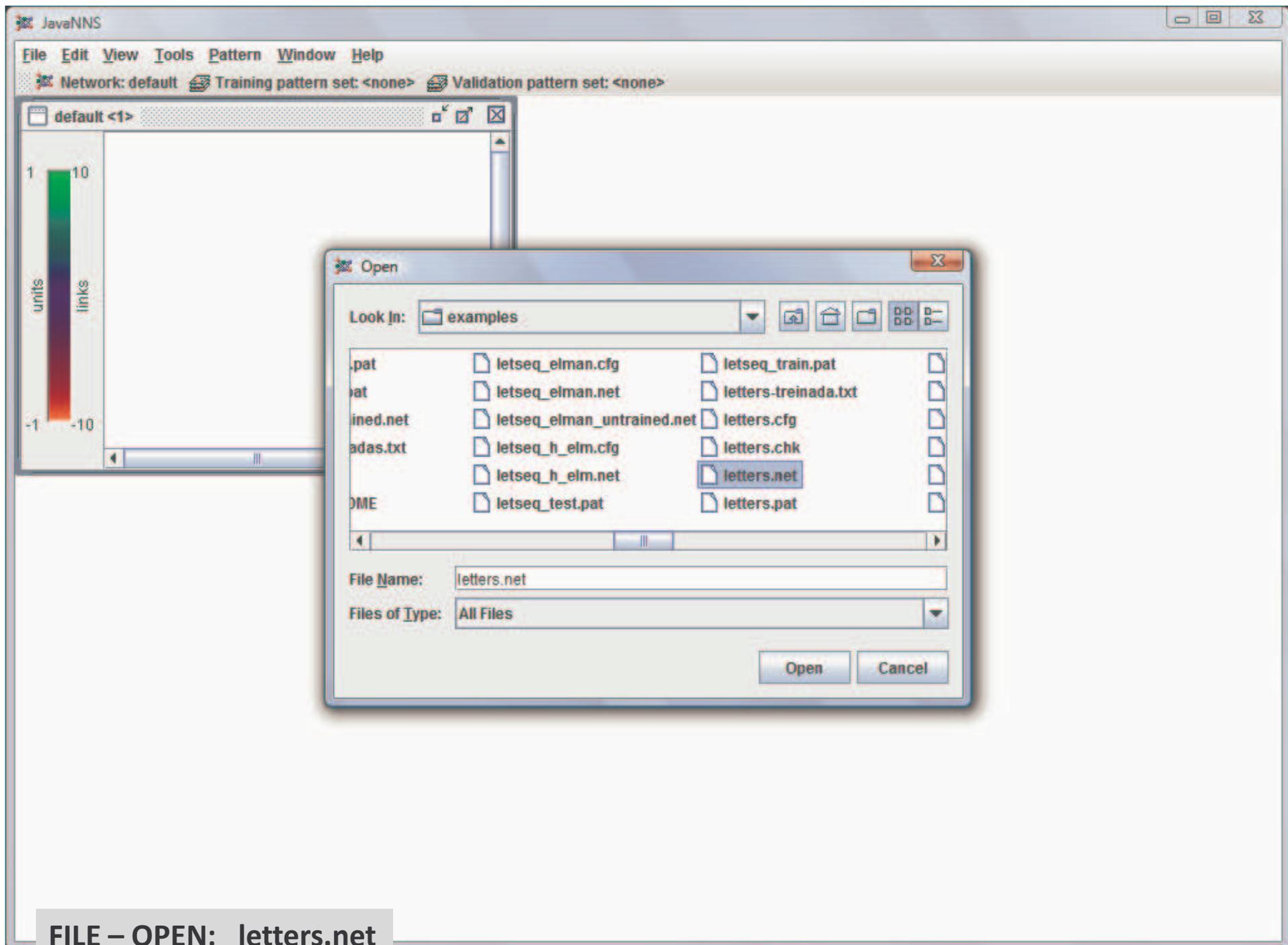


Web Site:

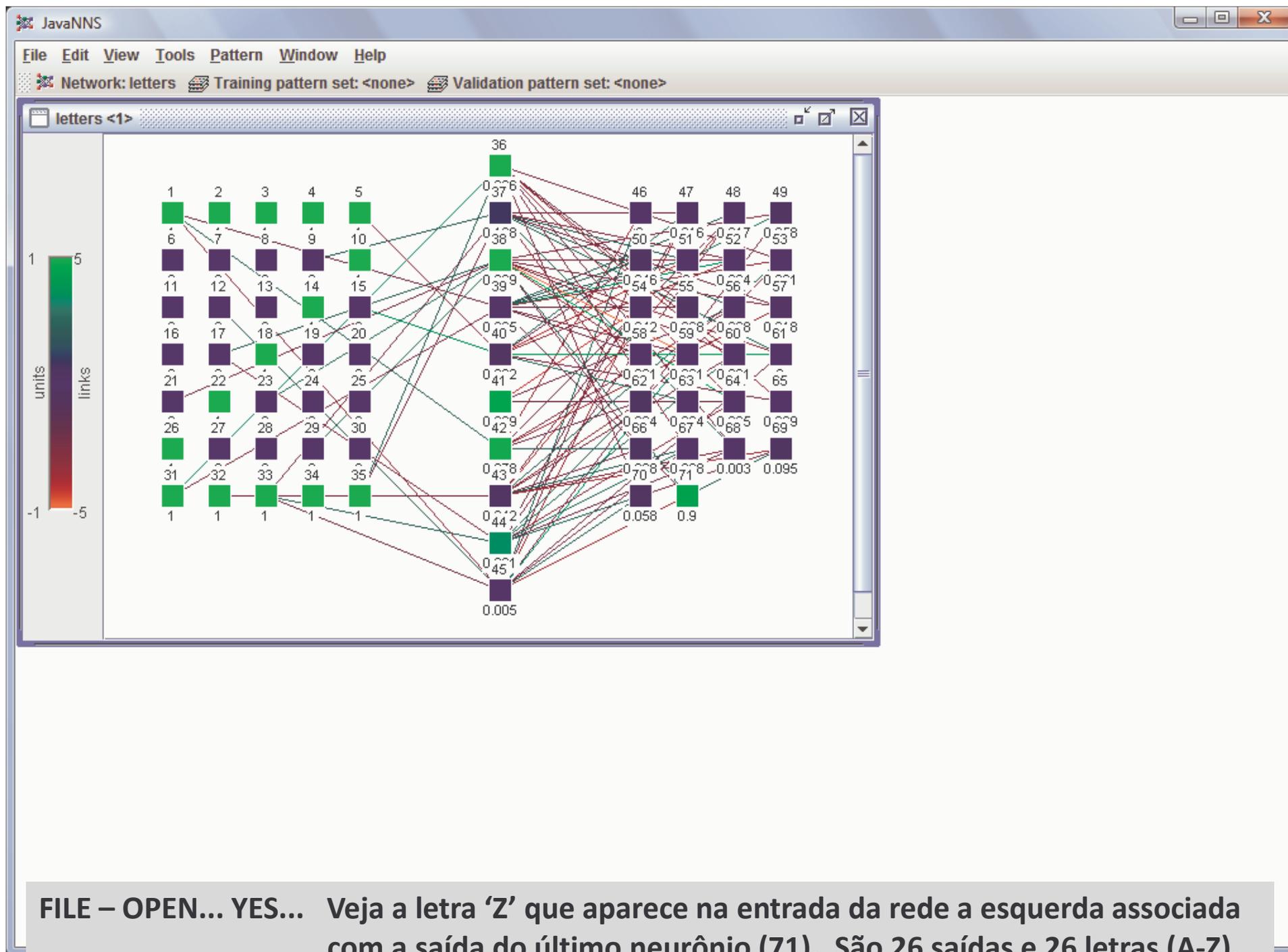
<http://www.ra.cs.uni-tuebingen.de/downloads/JavaNNS/>

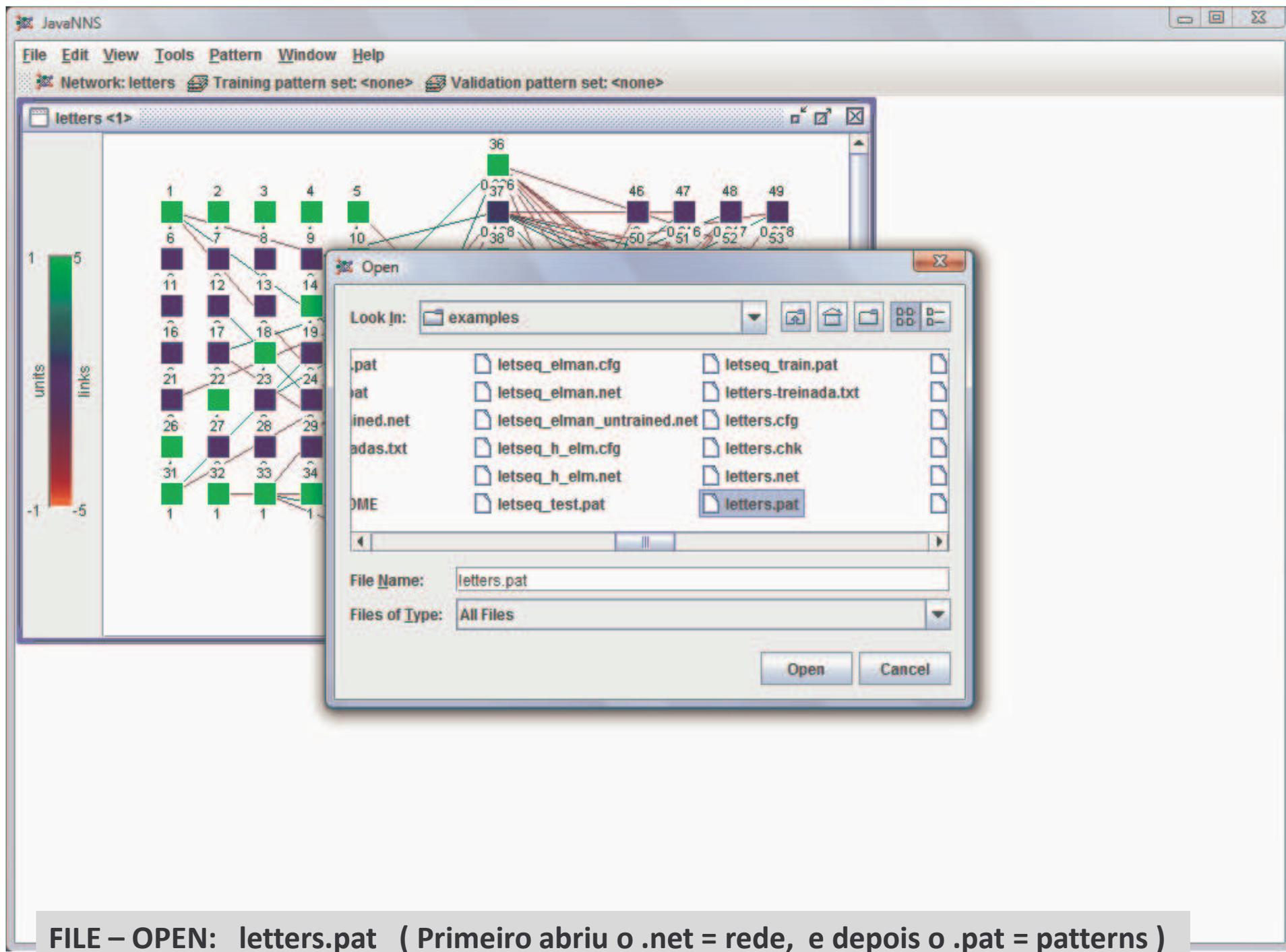


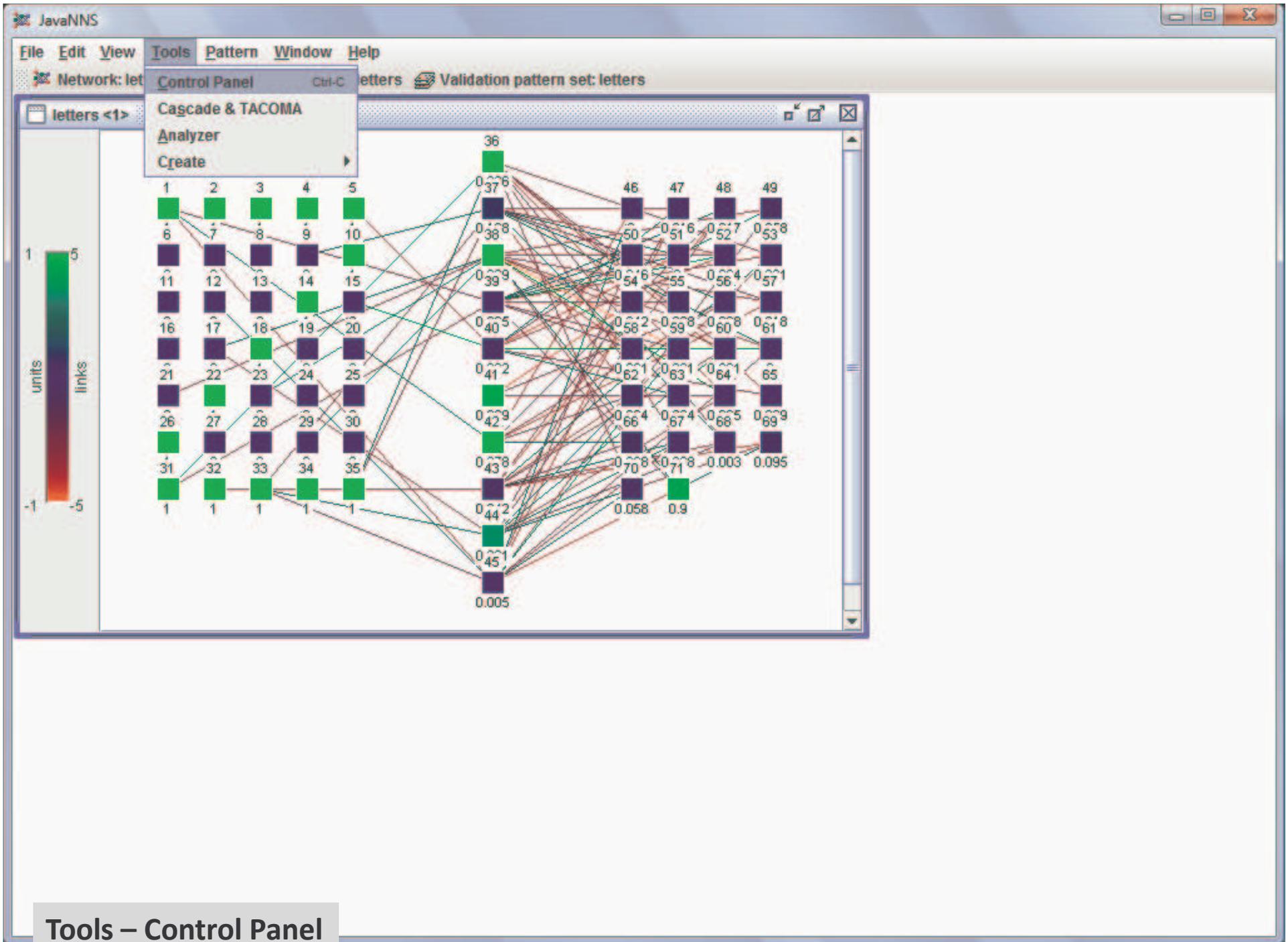
EXECUTAR: JavaNNS.jar

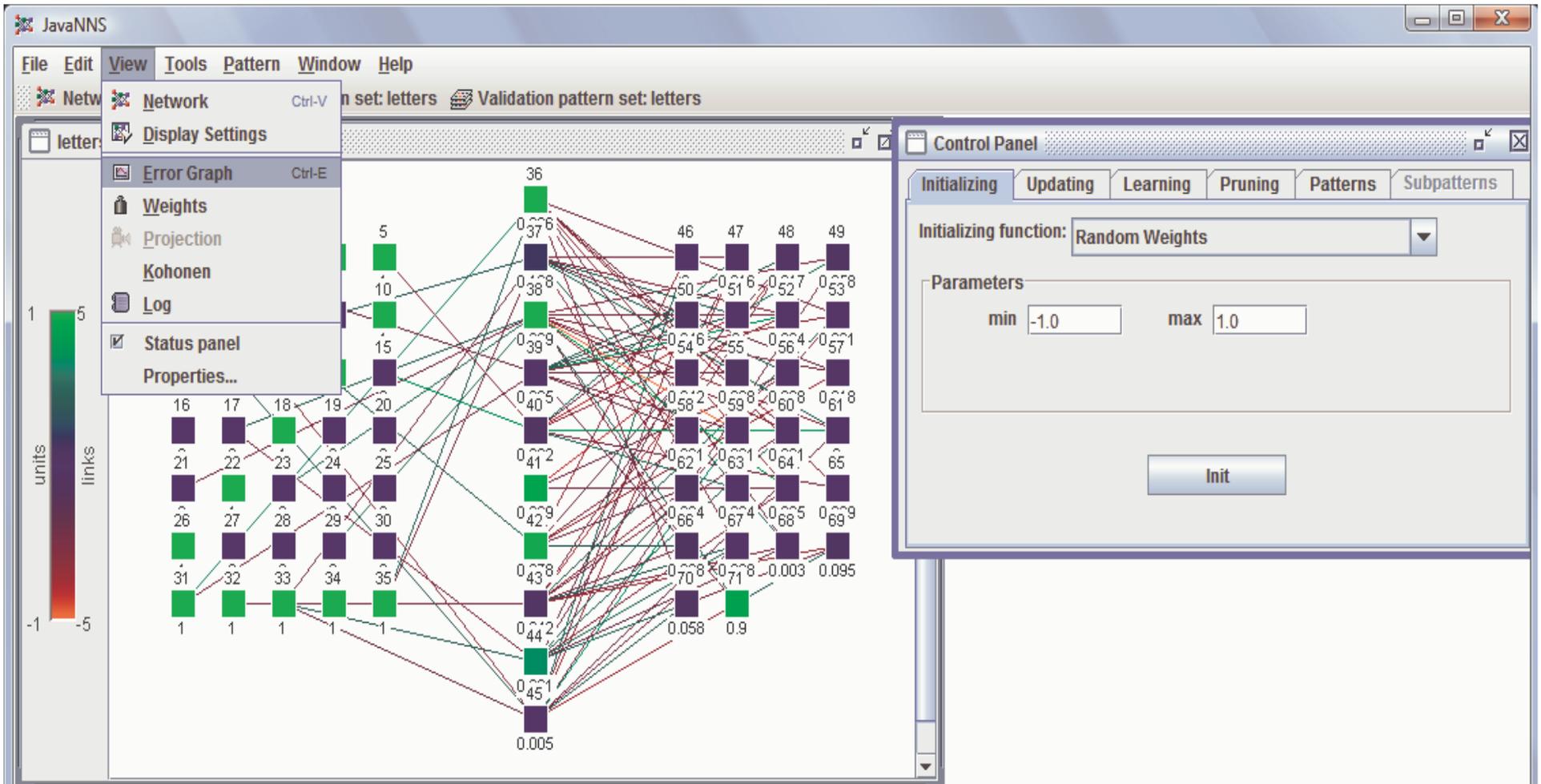


FILE – OPEN: letters.net

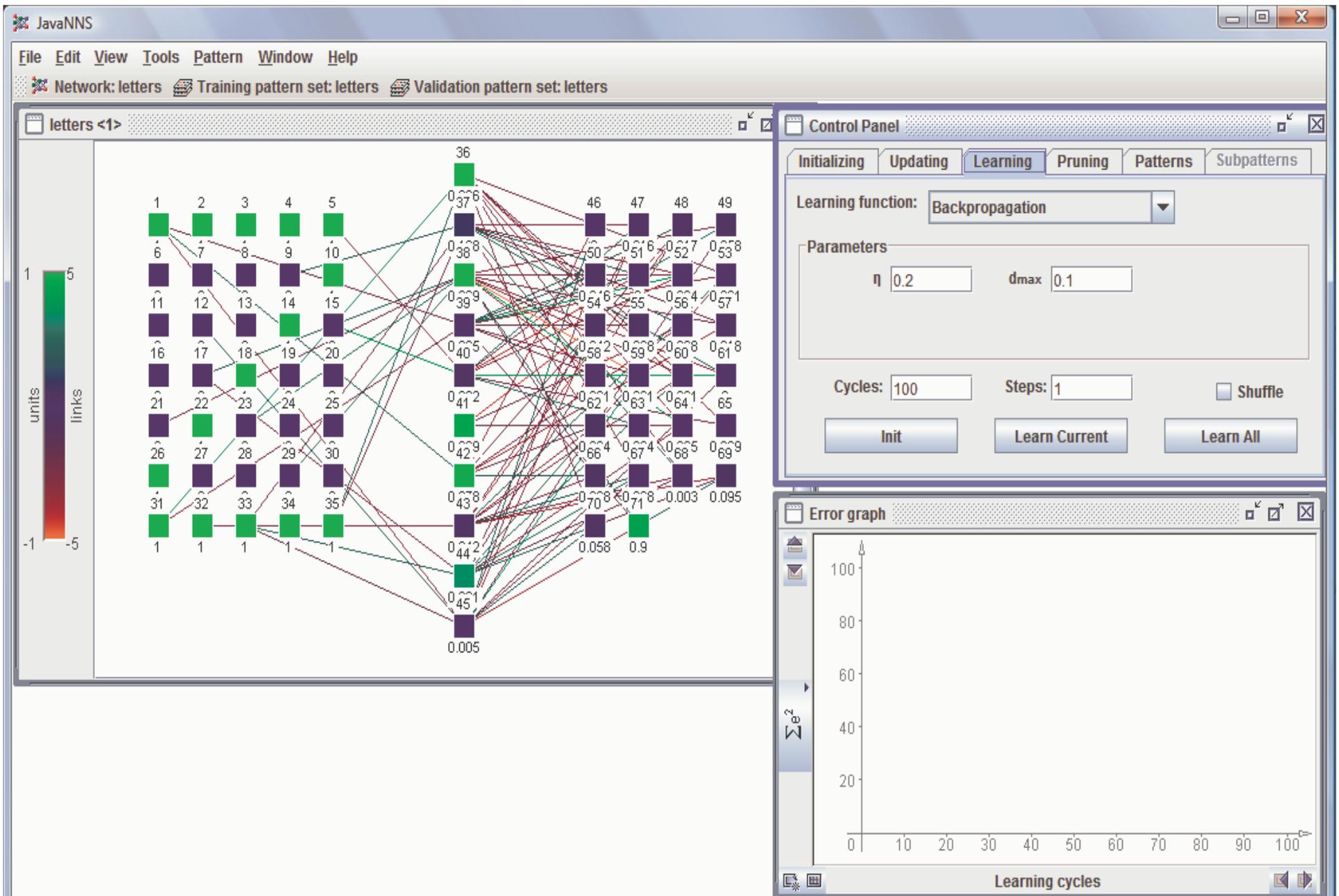






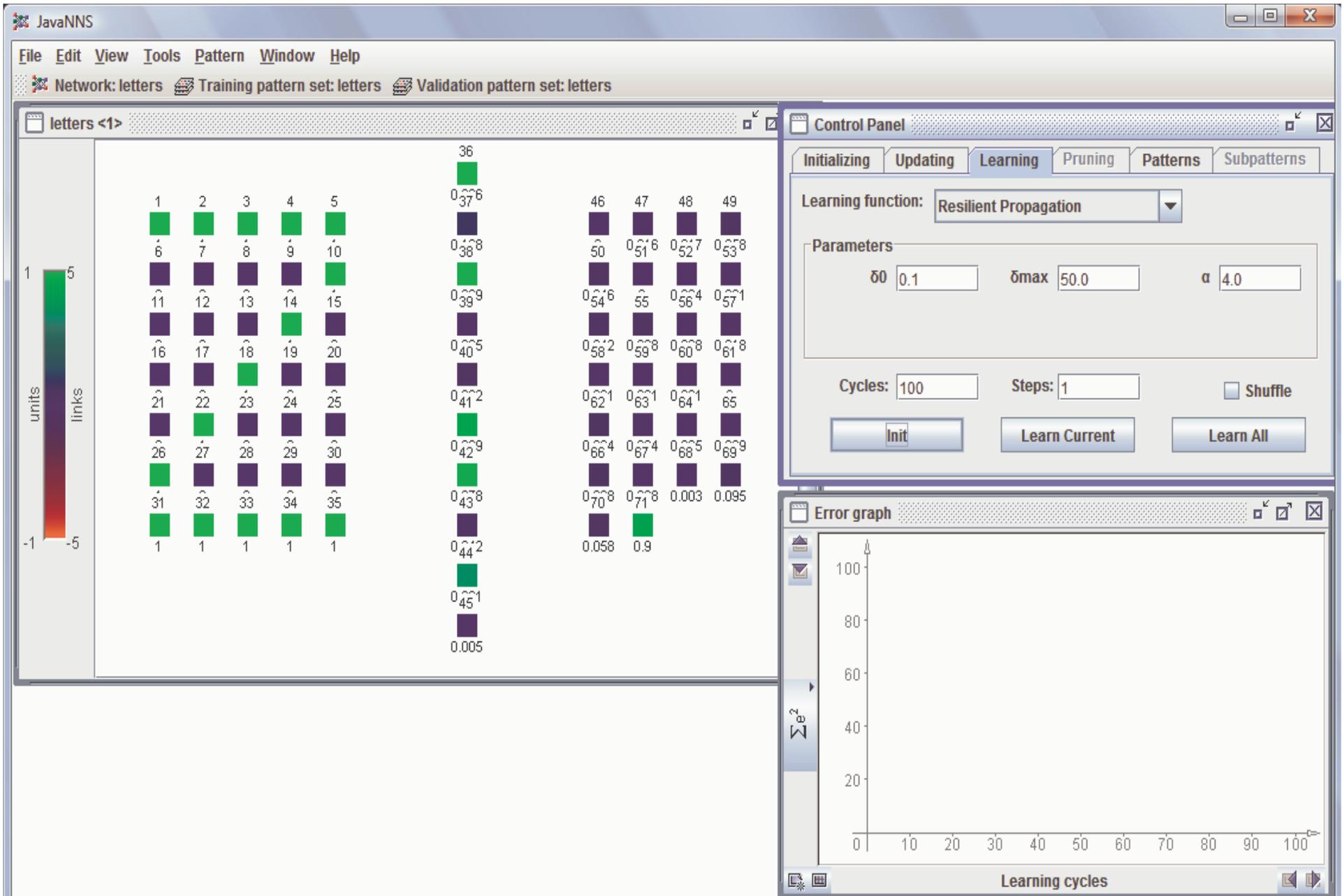


**View – Error Graph**



No Control Panel, vá para a aba "Learning"...

Use o BackPropagation (aprendizado não muito rápido) ou...



“Learning”: Use o Resilient Propagation (aprendizado BEM rápido!)  
 Clicar em “Init” e depois em “Learn All”

JavaNNS

File Edit View Tools Pattern Window Help

Network: letters Training pattern set: letters Validation pattern set: letters

letters <1>

Control Panel

Initializing Updating Learning Pruning **Patterns** Subpatterns

Learning function: Resilient Propagation

Parameters

$\delta$  0.1  $\delta_{max}$  50.0  $\alpha$  4.0

**Ver os patterns**

Cycles: 100 Steps: 1  Shuffle

Init Learn Current Learn All

Error graph

$\Sigma e^2$

Learning cycles

units links

Zoom

Aprende!

Você pode abrir 2 patterns: training / validation

The screenshot displays the JavaNNS application window. The main area shows a neural network diagram with 35 input units (arranged in a 7x5 grid) and 14 hidden units (arranged in two columns of 7). A legend on the left indicates that green units represent 'units' and red units represent 'links'. The network is connected to a control panel on the right. The control panel has tabs for 'Initializing', 'Updating', 'Learning', 'Pruning', 'Patterns', and 'Subpatterns'. The 'Learning' tab is active, showing a 'Learning function' dropdown set to 'Resilient Propagation'. Below this, there are input fields for 'δmax' (50.0) and 'α' (4.0). There is also a 'Steps' field set to 1 and a 'Shuffle' checkbox. Two buttons, 'Learn Current' and 'Learn All', are visible. A 'Save' dialog box is open in the foreground, showing the save location as 'examples', the file name as 'result.txt', and the file type as 'Result files \*.res'. A graph at the bottom right shows 'Learning cycles' on the x-axis (0 to 100) and a y-axis with a value of 20. The graph shows a curve that starts at 20 and decreases towards 0 as learning cycles progress.

Verificando se a rede aprendeu mesmo...  
FILE – Save Data: Result.txt

The screenshot displays the JavaNNS software interface. The main window shows a neural network diagram with 36 units. A color scale on the left indicates the strength of links, ranging from -1 (red) to 5 (green). The network is organized into layers: an input layer with 5 units (1-5), a hidden layer with 10 units (6-15), another hidden layer with 10 units (16-25), and an output layer with 11 units (26-36). A 'Saving details' dialog box is open in the center, with the following settings:

- Start pattern: 1
- End pattern: 26
- Include input patterns
- Include output patterns
- create  append

The 'Control Panel' on the right shows the 'Learning' tab selected. The learning function is 'Resilient Propagation'. Parameters include a learning rate of 0.1, a maximum error ( $\delta_{max}$ ) of 50.0, and a momentum ( $\alpha$ ) of 4.0. The 'Cycles' field is set to 100 and 'Steps' to 1. The 'Shuffle' checkbox is unchecked. Buttons for 'Init', 'Learn Current', and 'Learn All' are visible. Below the control panel, a graph shows the error function  $\sum \delta_i^2$  over 100 learning cycles, with the error decreasing from approximately 100 to near 0.

Clicar em “Include input patterns” e “Include Output Patterns”  
Depois em OK. Abra o arquivo texto gerado... Et voilà!