4.3.2.5 Defining the Log File

Messages that document the simulation run can be stored in the **log file**. The protocol contains file operations, definitions of values set by clicking the `SET` button in the info panel, as well as a teaching protocol (cycles, parameters, errors). In addition, the user can output data about the network to the log file with the help of the info panel. If no log file is loaded, output takes place only on `stdout`. If no file name is specified when clicking `LOAD`, a possibly open log file is closed and further output is restricted to `stdout`.

4.3.3 Control Panel

With this window the simulator is operated. Figure 4.11 shows this window. Table 4.1 lists all the input options with types and value ranges. The meaning of the 5 learning parameters depends upon the learning function selected with the menu `select learning function` invoked by the button `OPTIONS` of the control panel.

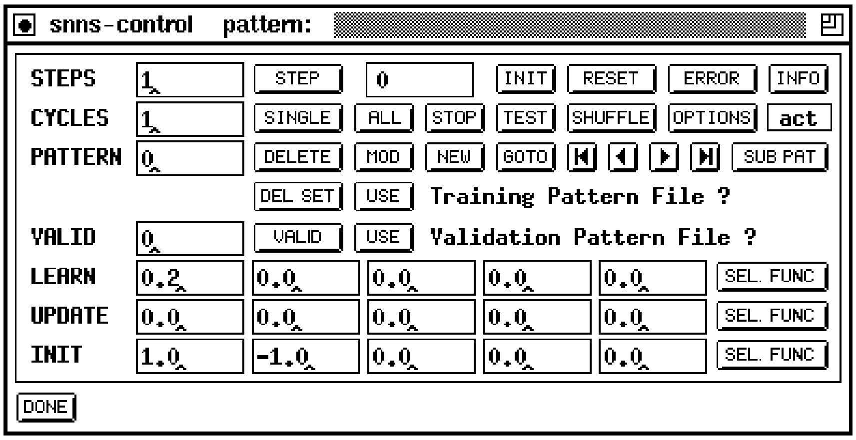

Figure 4.11: Control Panel

Name	Type	Value Range
`STEPS` (update-Steps)	Text	$0 \leq n$
`COUNT` (counter for steps)	Label	$0 \leq n$
`CYCLES`	Text	$0 \leq n$
`PATTERN` (number of current pattern)	Label	$0 \leq n$
`VALID`	Text	$0 \leq n$
`LEARN` (5 parameters: η, α, d, ...)	Text	float
`UPDATE` (5 parameters)	Text	float
`INIT` (5 parameters)	Text	float

Table 4.1: Input fields of the control panel

There are the following text fields, buttons and menu buttons:

1. `STEPS`: This text field specifies the number of update steps of the network. With `Topological_Order` selected as update mode (chosen with the menu `select update function` from the button `OPTIONS` in the control panel) one step is sufficient to propagate information from input to output. With other update modes or with recursive networks, several steps might be needed.

2. `STEP`: After clicking this button, the simulator kernel executes the number of steps specified in the text field `STEPS`. If `STEPS` is zero, the units are only redrawn. The update mode selected with the button `MODE` is used (see chapter 3.2). The first update step in the mode `topological` takes longer than the following, because the net is sorted topologically first. Then all units are redrawn.

3. `COUNT`: The text field next to the `STEP` button displays the steps, executed so far.

4. `INIT`: Initializes the network with values according to the function and parameters given in the last line of the panel.

5. `RESET`: The counter is reset and the units are assigned their initial activation.

6. `ERROR`: By pressing the error button in the control panel, SNNS will print out several statistics. The formulas were contributed by Warren Sarle from the SAS institute. Note that these criteria are for *linear* models; they can sometimes be applied directly to nonlinear models if the sample size is large. A recommended reference for linear model selection criteria is [JGHL80].

   Notation:

n	=	Number of observations (sample size)
p	=	Number of parameters, to be estimated (i.e. weights)
SSE	=	The sum of squared errors
TSS	=	The total sum of squares corrected for the mean for the dependent variable

   Criteria for adequacy of the estimated model in the sample
   Pearson's R^2, the proportion of variance, is explained or accounted by the model:
   $$R^2 := 1 - \frac{SSE}{TSS}$$

   Criteria for adequacy of the true model in the population
   The mean square error [JGHL80] is defined as: $MSE := \frac{SSE}{n-p}$, the root mean square error as: $RMSE := \sqrt{MSE}$.

   The R^2_{adj}, the R^2 [JGHL80] adjusted for degrees of freedom, is defined as:
   $$R^2_{adj} := 1 - \frac{n-1}{n-p} \cdot (1 - R^2)$$

   Criteria for adequacy of the estimated model in the population
   Anemiya's prediction criterion [JGHL80] is similar to the R^2_{adj}:
   $$PC := 1 - \frac{n+p}{n-p} \cdot (1 - R^2)$$

   The estimated mean square error of prediction (J_p) assuming that the values of the regressors are fixed and that the model is correct is:
   $$J_p := (n + p) \cdot MSE/n$$

   The conservative mean square error in prediction [Weh94] is:
   $$CMSEP := \frac{SSE}{n-2p}$$

The generalized cross validation (GCV) is given by Wahba [GHW79] as:

$$GCV := \frac{SSE \cdot n}{(n-p)^2}$$

The estimated mean square error of prediction assuming that both independent and dependent variables are multivariate normal is defined as:

$$GMSEP := \frac{MSE(n+1)(n-2)}{n(n-p-1)}$$

Shibata's criterion $SHIBATA := \frac{SSE(n+2p)}{n}$ can be found in [Shi68].

Finally, there is Akaikes information criterion [JGHL80]:

$$AIC := n \cdot \ln \frac{SSE}{n} + 2p$$

and the Schwarz's Bayesian criterion [JGHL80]:

$$SBC := n \cdot \ln \frac{SSE}{n} + n \cdot \ln p.$$

Obviously, most of these selection criteria do only make sense, if $n >> p$.

7. `INFO`: Information about the net is written to the text window.

8. `CYCLES`: This text field specifies the number of learning cycles. It is mainly used in conjunction with the next two buttons. A cycle (also called an epoch sometimes) is a unit of training where all patterns of a pattern file are presented to the network once.

9. `SINGLE`: The net is trained with a single pattern for the number of training cycles defined in the field `CYCLES`. The text window reports the error of the network every `CYCLES`/10 cycles, i.e. independent of the number of training cycles only 10 numbers are generated. (This prevents flooding the user with network performance data and slowing down the training by file I/O).

   The error reported in the text window is the sum of the quadratic differences between the teaching input and the real output over all output units. The average error per output unit is given behind `ave`.

10. `ALL`: The net is trained with all patterns for the number of training cycles specified in the field `CYCLES`. This is the usual way to train networks from the graphical user interface. Note, that if cycles has a value of, say, 100, the button `ALL` causes SNNS to train all patterns once (one cycle = one epoch) and repeat this 100 times (NOT training each pattern 100 times in a row and then applying the next pattern).

    The error reported in the text window is the sum of the quadratic difference between the teaching output and the real output over all output units summed over the number of patterns presented. The average error per output unit is given behind `ave`.

11. `STOP`: Stops the teaching cycle. After completion of the current step or teaching cycle, the simulation is halted immediately.

12. `TEST`: With this button, the user can test the behavior of the net with all patterns loaded. The activation values of input and output units are copied into the net. (For output units see also button `SHOW`). Then the number of update steps specified in `STEPS` are executed.

13. `SHUFFLE`: It is important for optimal learning that the various patterns are presented in different order in the different cycles. A random sequence of patterns is created automatically, if `SHUFFLE` is switched on.

14. `OPTIONS`: Offers the following menu:

`jog weights`	change all link weights randomly
`edit f-types`	edit/create f-types
`edit sites`	edit/create sites

   If `jog weights` is selected, a popup window appears to specify the value range (`low limit .. high limit`) of the random noise to be added to all links in the network.

15. `act`: With this button, the user specifies the changes to the activation values of the output units when a pattern is applied with `TEST`. The following table gives the three possible alternatives:

`None`	The output units remain unchanged.
`Out`	The output values are computed and set, activations remain unchanged.
`Act`	The activation values are set.

   The label of this button always displays the item selected from the menu.

16. `PATTERN`: This text field displays the current pattern number.

17. `DELETE`: The pattern whose number is displayed in the text field `PATTERN` is deleted from the pattern file.

18. `MOD`: The pattern whose number is displayed in the text field `PATTERN` is modified in place.

   The current activation of the input units and the current output values of output units of the network loaded make up the input and output pattern. These values might have been set with the network editor and the Info panel before.

19. `NEW`: A new pattern is defined that is added behind existing patterns. Input and output values are defined as above. This button is disabled whenever the current pattern set has variable dimensions.

20. `GOTO`: The simulator advances to the pattern whose number is displayed in the text field `PATTERN`.

21. Arrow buttons ⏮, ◀, ▶, and ⏭: With these buttons, the user can navigate through all patterns loaded, as well as jump directly to the first and last pattern. Unlike with the button `TEST` no update steps are performed here.

22. `SUB PAT`: Opens the panel for sub-pattern handling. The button is inactive when the current pattern set has no variable dimensions. The sub-pattern panel is described in section 5.3.

23. `DEL SET`: Opens the menu of loaded pattern sets. The pattern set of the selected entry is removed from main memory. The corresponding pattern file remains

untouched. When the current pattern set is deleted, the last in the list becomes current. When the last remaining pattern set is deleted, the current pattern set becomes undefined and the menu shows the entry `No Files`.

24. `USE`: Also opens the menu of loaded pattern sets. The pattern set of the selected entry becomes the current set. All training, testing, and propagation actions refer always to the current pattern set. The name of the corresponding pattern file is displayed next to the button in the *Current Pattern Set* field.

25. *Current Pattern Set*: This field displays the name of the pattern set currently used for training. When no current pattern set is defined the entry "`Training Pattern File ?`" is displayed.

26. `VALID`: Gives the intervals in which the training process is to be interrupted by the computation of the error on the validation pattern set. A value of 0 inhibits validation. The validation error is printed on the shell window and plotted in the graph display.

27. `USE`: Opens the menu of loaded pattern sets. The pattern set of the selected entry becomes the current validation set. The name of the corresponding pattern file is displayed next to the button in the *Validation Pattern Set* field.

28. *Validation Pattern Set*: This field displays the name of the pattern set currently used for validation. When no current pattern set is defined the entry "`Validation Pattern File ?`" is displayed.

29. `LEARN`: Up to five fields to specify the parameters of the learning function. The number required and their resp. meaning depend upon the learning function used. Only as many widgets as parameters needed will be displayed, i.e. all widgets visible need to filled in. A description of the learning functions that are already built in into SNNS is given in section 4.4.

30. `SEL. FUNC`: in the LEARN row invokes a menu to select a learning function (learning procedure). The following learning functions are currently implemented:

`ART1`	ART1 learning algorithm
`ART2`	ART2 learning algorithm
`ARTMAP`	ARTMAP learning algorithm
	(all ART models by Carpenter & Grossberg)
`BBPTT`	Batch-Backpropagation for recurrent networks
`BPTT`	Backpropagation for recurrent networks
`Backpercolation`	Backpercolation 1 (Mark Jurik)
`BackpropBatch`	Backpropagation for batch training
`BackpropMomentum`	Backpropagation with momentum term
`BackpropWeightDecay`	Backpropagation with Weight Decay
`CC`	Cascade correlation meta algorithm
`Counterpropagation`	Counterpropagation (Robert Hecht-Nielsen)
`Dynamic_LVQ`	LVQ algorithm with dynamic unit allocation
`Hebbian`	Hebbian learning rule
`JE_BP`	Backpropagation for Jordan-Elman networks
`JE_BP_Momentum`	BackpropMomentum for Jordan-Elman networks
`JE_Quickprop`	Quickprop for Jordan-Elman networks
`JE_Rprop`	Rprop for Jordan-Elman networks

`Kohonen`	Kohonen Self Organizing Maps
`Monte-Carlo`	Monte-Carlo learning
`PruningFeedForward`	Pruning algorithms
`QPTT`	Quickprop for recurrent networks
`Quickprop`	Quickprop (Scott Fahlman)
`RCC`	Cascade Correlation for recurrent networks
`RM_delta`	Rumelhart-McClelland's delta rule
`RadialBasisLearning`	Radial Basis Functions
`RBF-DDA`	modified Radial Basis Functions
`Rprop`	Resilient Propagation learning
`SimAnn_SS_error`	Simulated Annealing with SSE computation
`SimAnn_WTA_error`	Simulated Annealing with WTA computation
`SimAnn_WWTA_error`	Simulated Annealing with WWTA computation
`Std_Backpropagation`	"vanilla" Backpropagation
`TimeDelayBackprop`	Backpropagation for TDNNs (Alex Waibel)

31. `UPDATE`: Up to five fields to specify the parameters of the update function. The number required and their resp. meaning depend upon the update function used. Only as many widgets as parameters needed will be displayed, i.e. all widgets visible need to filled in.

32. `SEL. FUNC`: in the UPDATE row invokes a menu to select an update function. A list of the update functions that are already built in into SNNS and their descriptions is given in section 4.5.

33. `INIT`: Five fields to specify the parameters of the init function. The number required and their resp. meaning depend upon the init function used. Only as many widgets as parameters needed will be displayed, i.e. all widgets visible need to filled in.

34. `SEL. FUNC`: in the INIT row invokes a menu to select an initialization function.

See section 4.6 for a list of the init functions available as well as their description.

4.3.4 Info Panel

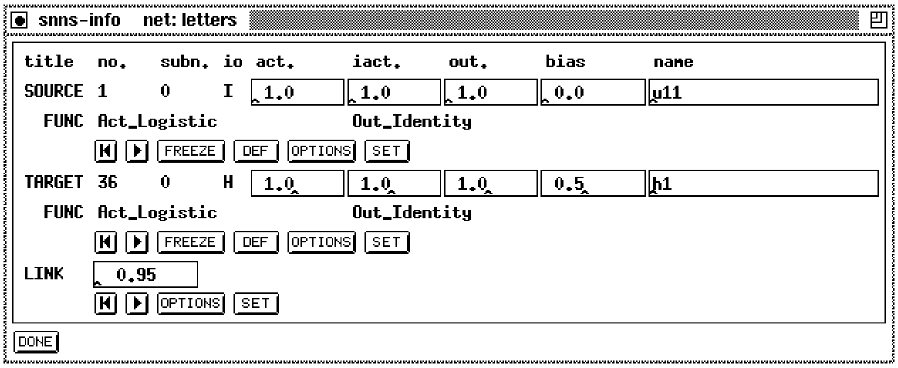

Figure 4.12: Info panel

The info panel displays all data of two units and the link between them. The unit at the beginning of the link is called `SOURCE`, the other `TARGET`. One may run sequentially through all connections or sites of the `TARGET` unit with the arrow buttons and look at the corresponding source units and vice versa.

This panel is also very important for editing, since some operations refer to the displayed `TARGET` unit or (`SOURCE`→`TARGET`) link. A default unit can also be created here, whose values (activation, bias, IO-type, subnet number, layer numbers, activation function, and output function) are copied into all selected units of the net.

The source unit of a link can also be specified in a 2D display by pressing the middle mouse button, the target unit by releasing it. To select a link between two units the user presses the middle mouse button on the source unit in a 2D display, moves the mouse to the target unit while holding down the mouse button and releases it at the target unit. Now the selected units and their link are displayed in the info panel. If no link exists between two units selected in a 2D display, the `TARGET` is displayed with its first link, thereby changing `SOURCE`.

In table 4.2 the various fields are listed. The fields in the second line of the SOURCE or TARGET unit display the name of the activation function, name of the output function, name of the f-type (if available). The fields in the line `LINK` have the following meaning: weight, site value, site function, name of the site. Most often only a link weight is available. In this case no information about sites is displayed.

Unit number, unit subnet number, site value, and site function cannot be modified. To change attributes of type text, the cursor has to be exactly in the corresponding field.

There are the following buttons for the units (from left to right):

1. Arrow button: The button below TARGET selects the first target unit (of the given source unit); the button below SOURCE selects the first source unit (of the given target unit);

Name	Meaning	Type	set by	value range
no.	unit no.	Label		$1..2^{31}$
subn.	subnet no.	Label		$-32736..32735$
io	IO-type	Label	`OPTIONS`	I(nput), O(utput), H(idden), D(ual), S(pecial)
act.	activation	Text	input	float;
iact.	initial act.	Text	input	float;
out.	output	Text	input	float;
bias	bias value	Text	input	float
name	unit name	Text	input	string, starting with letter
activation function		Label	`OPTIONS`	as available
output function		Label	`OPTIONS`	as available
link	weight	Text	input	float
site value		Label		float
site function		Label		as available
site name		Label		as available at `TARGET`

Table 4.2: Table of the unit, link and site fields in the Info panel

2. Arrow button $\boxed{\blacktriangleright}$: The button below TARGET selects the next target unit (of the given source unit); the button below SOURCE selects the next source unit (of the given target unit);
3. `FREEZE`: Unit is frozen, if this button is inverted. Changes become active only after `SET` is clicked.
4. `DEF`: The default unit is assigned the displayed values of `TARGET` and `SOURCE` (only activation, bias, IO-type, subnet number, layer numbers, activation function and output function).
5. `OPTIONS`: Calls the following menu:

`change io-type`	change the IO-type
`change f-type`	change f-type
`display activation function`	graph of the activation function
`change activation function`	change activation function note: f-type gets lost!
`display output function`	graph of the output function
`change output function`	change output function note: f-type gets lost!
`assign layers`	assign unit to layers
`list all sources`	list all predecessors
`list all targets`	list all successors

6. `SET`: Only after clicking this button the attributes of the corresponding unit are set to the specified value. The unit is also redrawn. Therefore the values can be changed without immediate effect on the unit.

There exist the following buttons for links (from left to right):

1. : Select first link of the `TARGET` unit.
2. : Select next link of the `TARGET` unit.
3. `OPTIONS`: Calls the following menu:

`list current site of TARGET`	list of all links of current site.
`list all sites of TARGET`	list all sites of the `TARGET`
`list all links from SOURCE`	list all links starting at `SOURCE`
`delete site`	delete displayed site
	note: f-type gets lost!
`add site`	add new site to `TARGET`
	note: f-type gets lost!

4. `SET`: Only after clicking this button the link weight is set.

4.3.4.1 Unit Function Displays

The characteristic functions of the units can be displayed in a graphic representation. For this purpose separate displays have been created, that can be called by selecting the options `display activation function` or `display output function` in the menu under the `options` button of the target and source unit in the info panel.

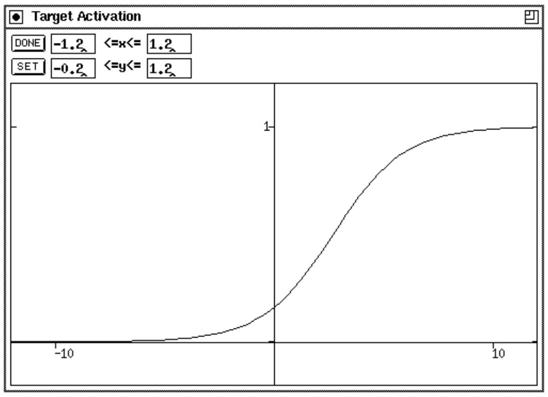

Figure 4.13: The logistic activation function in a unit function display

Figure 4.13 shows an example of an activation function. The window header states whether it is an activation or an output function, as well as whether it is the current function of the source or target unit.

The size of the window is as flexible as the picture range of the displayed function. The picture range can be changed by using the dialog widgets at the top of the function displays. The size of the window may be changed by using the standard mechanisms of your window manager.

If a new activation or output function has been defined for the unit, the display window changes automatically to reflect the new situation. Thereby it is easy to get a quick overview of the available functions by opening the function displays and then clicking through the list of available functions (This list can be obtained by selecting `select activation function` or `select output function` in the unit menu).

4.3.5 2D Displays

A 2D display or simply display is always part of the user interface. It serves to display the network topology, the units' activations and the weights of the links. Each unit is located on a grid position, which simplifies the positioning of the units. The distance between two grid points (`grid width`) can be changed from the default 37 pixels to other values in the setup panel.

The current position, i.e. the grid position of the mouse, is also numerically displayed at the bottom of the manager panel. The x-axis is the horizontal line and valid coordinates lie in the range $-32736 \ldots +32735$ (short integer).

The current version displays units as boxes, where the size of the box is proportional to the value of the displayed attribute. Possible attributes are activation, initial activation, bias, and output. A black box represents a positive value, an empty box a negative value. The size of the unit varies between 16x16 and 0 pixels according to the value of $scaleFactor$. The parameter *scaleFactor* has a default value of 1.0, but may be set to values between 0.0 and 2.0 in the setup panel. Each unit can be displayed with two of several attributes. One above the unit and one below the unit. The attributes to be displayed can be selected in the setup panel.

Links are shown as solid lines, with optional numerical display of the weight in the center of the line and/or arrow head pointing to the target unit. These features are optional, because they heavily affect the drawing speed of the display window.

A display can also be frozen with the button FREEZE (button gets inverted). It is afterwards neither updated anymore[2], nor does it accept further editor commands.

An iconified display is not updated and therefore consumes (almost) no CPU time. If a window is closed, its dimensions and setup parameters are saved in a stack (LIFO). This means that a newly requested display gets the values of the window assigned that was last closed. For better orientation, the window title contains the subnet number which was specified for this display in the setup panel.

4.3.5.1 Setup Panel of a 2D Display

Changes to the kind of display of the network can be performed in the **Setup panel**. All settings become valid only after the button DONE is clicked. The whole display window is then redrawn.

[2]If a frozen display has to be redrawn, e.g. because an overlapping window was moved, it gets updated. If the network has changed since the freeze, its contents will also have changed!

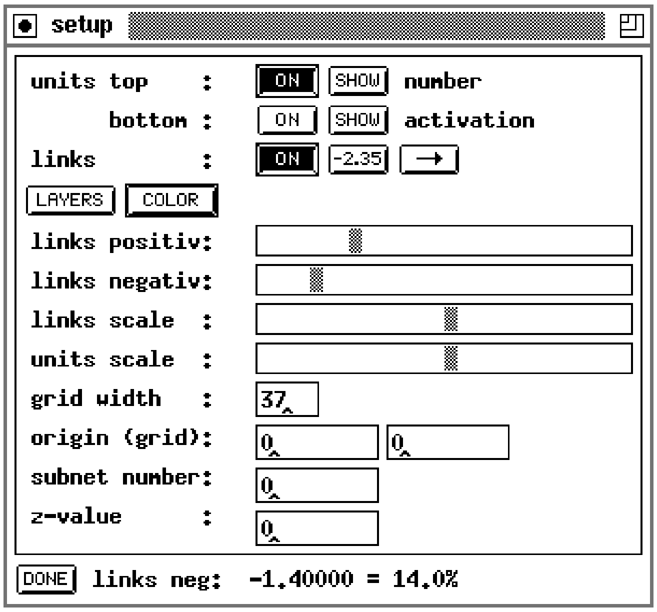

Figure 4.14: Setup Panel of a 2D-display.

1. Buttons to control the display of unit information: The first two lines of the Setup panel (`units top` and `units bottom`) contain two buttons each to set the unit parameter that can be displayed at the top resp. the bottom of the unit.

   The button $\boxed{\texttt{ON}}$ toggles the display of information which can be selected with the button $\boxed{\texttt{SHOW}}$. The unit name, unit number, or the z-value (3D coordinate) can be displayed above the unit, the activation, initial activation, bias, or output of the unit below the unit. The numerical attribute selected with the button $\boxed{\texttt{SHOW}}$ at the bottom of the unit (activation, initial activation, output, or bias) also determines the size of the unit in the graphical representation.

   It is usually not advisable to switch off `top` (number or name), because this information is needed for reference to the info panel. An unnamed unit is always displayed with its number.

2. Buttons to control the display of link information: The third line consists of three buttons to select the display of link data, $\boxed{\texttt{ON}}$, $\boxed{-2.35}$, $\boxed{\rightarrow}$.

   - $\boxed{\texttt{ON}}$ determines whether to draw links at all (then $\boxed{\texttt{ON}}$ is inverted),
   - $\boxed{-2.35}$ displays link weights at the center of the line representing the link,
   - $\boxed{\rightarrow}$ displays arrow heads of the links pointing from source to target unit.

3. $\boxed{\texttt{LAYERS}}$ invokes another popup window to select the display of up to eight different layers in the display window. Layers are being stacked like transparent sheets of paper and allow for a selective display of units and links. These layers need NOT correspond with layers of units of the network topology (as in multilayer feed-forward networks), but they may do so. Layers are very useful to display only a selected subset of the network. The display of each layer can be switched on or off independently.