Manual for PMNEURO 1.0a free version June 1996
IMPORTANT : USE PMNEURO AT YOUR OWN RISC, NO WARRANTY
THIS MANUAL IS NO EXPLANATION ABOUT NEURONAL NETWORKS

Creating and working with neuronal networks, using PMNEURO 1.0a -- what do you need ?

1. The program PMNEURO 1.0a
2. Data files for training
3. Data files for propagation
4. Read the manual and read the examples at the end of this text

Content of program package :

PMNEURO.EXE	
MANUAL.TXT	(ascii file)
MANUAL.LWP	(WORKS file)
MB0607R2.DAT	example data file
MB0607R2.DOT	example propagation data file
MB0607R2.REA	example real result data file
REGIST.TXT	registration text
License.txt	

Next version :
Dear user,
if you send me a registration (and money), I'll send back a new version of the program.
I'm planing in next program versions :
Help-function, proofing and testing user input for plausibility, better graphical output, user choosen formulas for backpropagation, user choosen connecting of neurons, finding good error/learning rate values by the program itself, an interface to other input data sending or output data using programs(program - program comunication), bug fixes of version 1.0 and ideas of users.

My adress : Jochen Herwig
Eisenacher Str. 71 /N
36208 Wildeck
Germany

My E-mail : PMNEURO@TORA.ROBIN.DE

What does the program ?

1. It creates the structure/weight file from the training data file
2. It propagates data through the structure/weight file and creates a result file

Installing and starting the program :

make a path and copy the program to it.

doubleclick on the icon, or call it from an OS/2 window with PMNEURO
(using PMNEURO d makes a german version available)

Standard features :

After training, the structure/weights can be saved with 'Dialog-->Save'
Train an existing net with other values use 'Dialog-->Open'
(for example using another training set or error values)
Propagate data with a trained net use 'Dialog-->prop. load'
Propagate immediately after training or DURING training use 'Dialog-->propagate'

Advanced features :

First :

In the example above a neuronal net was trained and data were propagated thru the net.
The result is a new file with the propagation data and the result value.

May be in a time series there is asked for more than one value after the last real value;
the user can build a new propagate data set including the result value from the last propagation.
But the program can do this automatically, if there is a value greater 1 in the field 'Number of propagation' , the program builds a new virtuel propagation data set.
The values in the original propagation data file are shifted and replaced with the last result value.

But is it right only using the last propagation value while the net weights are still the same old values !?

Second :

The first propagated result value can be used to train the old neuronal network again and after training ends a new propagation trial starts !!!

After the first propagation 'Dialog-->Virtuell' shifts the propagation data file and replaces the last propagation data value with the propagation result value.

'Dialog-->Virt.Stru' creates a new virtuell training data set and trains the new network.

'Dialog-->Virt.Prop' propagates the new propagation data thru the new network.

'Dialog-->Virt.Cycle' does this process automatically !! It uses the value in 'Option-->Default-->Nr of Virt.Propagations' and repeats the process until this value is reached!

For each virtuell propagation the program creates a result file.
The names of this files are created by the program : VIRTnA.ERG
n = current number of the propagation

But wouldn't it be better changing error and learning rate values in this process, too !?

Third :

For each virtuell training, the error value and the learning rate value are changed by subtracting 'Option-->Default-->Subtract from Learn./Error' field values.
This stops, if the values will become negativ !

What happens, if the user types in negative values in this fields?
The program aborts, if the values will become greater 1 .

Fourth :

In propagating values for time series not all known values are used for training.
Some values are needed for testing the propagation results.
These values are called 'real results'.

The program can read a file with these values and shows them in a diagramm.
May be there are 25 values. The values are integers between 1 and 50 .

This diagramm has the dimensions of 'Option-->Default-->Presentation', for the example
the 'Range from to' should be 1 to 50 and the difference should be 1.

Presentation in 5 and 10 shows a rectangle with 5 columns and 10 lines.
Presentation in 50 and 1 shows a rectangle with 50 columns and 1 line.
The values are the numbers from 1 to 50.
Values equal to values in the 'real results' file are painted in another color.

Propagation results will be multiplicated with 100 and equal values will be marked in the
diagram with a 'X'.

Format of training data files

	Example
(1 .. N) Input values	0.01 0.02 0.03 0.04 (N=4)
(1 .. N) corresponding Output values	0.05 (N=1)
	0.02 0.03 0.04 0.05 (N=4)
	0.06 (N=1)

...this combination (trainingset, observation) repeats (1..N) times There are 2 trainingsets (N=2)

Data values are separated by blanks. The range of value : 0 .. 1.0
The program does no normalization !

Format of data files for propagation

	Example
(1 .. N) Input values	0.03 0.04 0.05 0.06 (N=4)

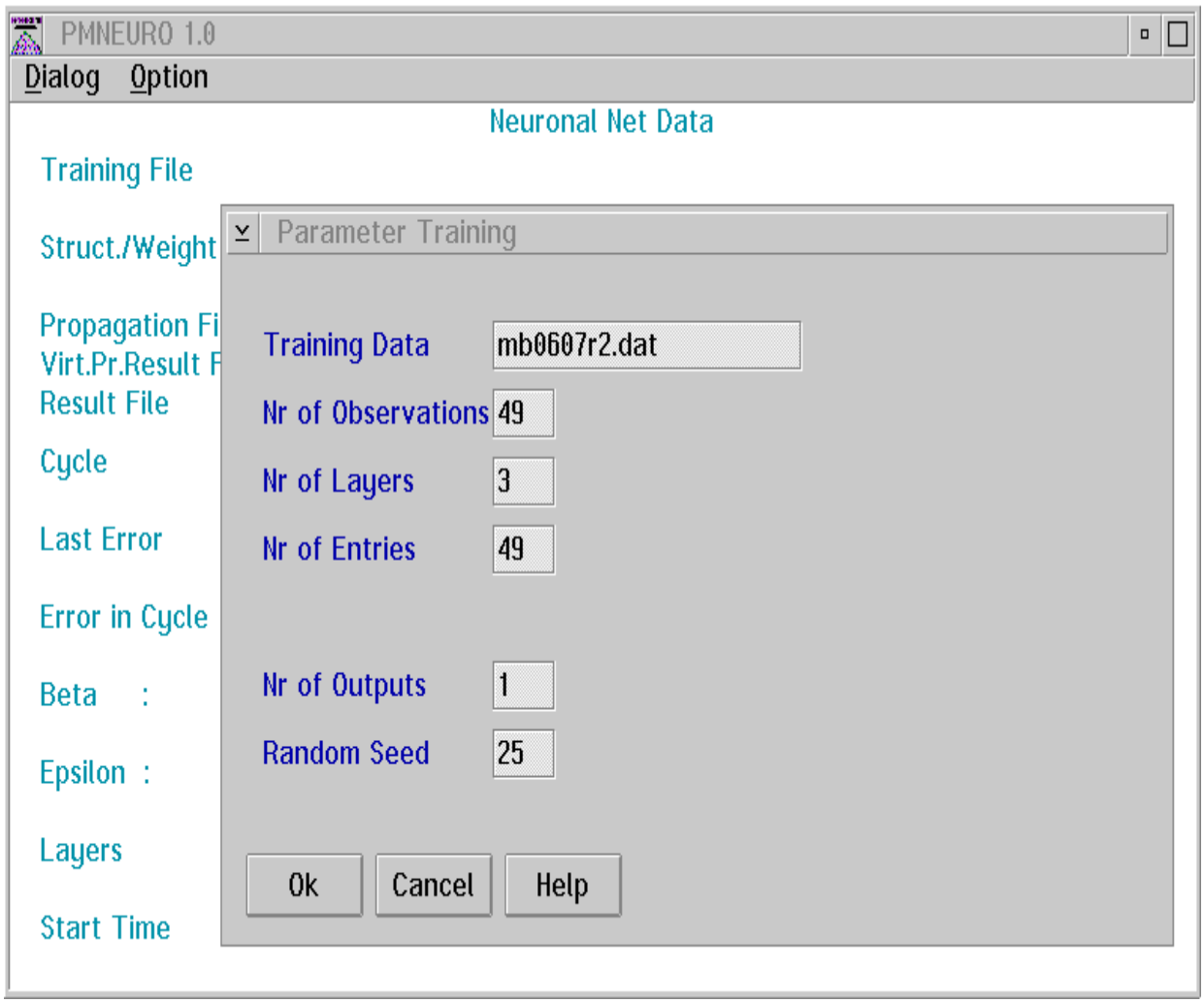
There are only 1 set of propagation input values.
The number of input values in a propagation data file must be the same as in the used training data file.

Format of result file

	Example
(1 .. N) Input values	0.03 0.04 0.05 0.06 (N=4)
(1 .. N) propagated values	0.06785 (N=1) == result

Format of structure/weights files	Example
special sign :	'N' (always the same)
number of input values	4
number of output values	1
values for layer 1 .... layer n	

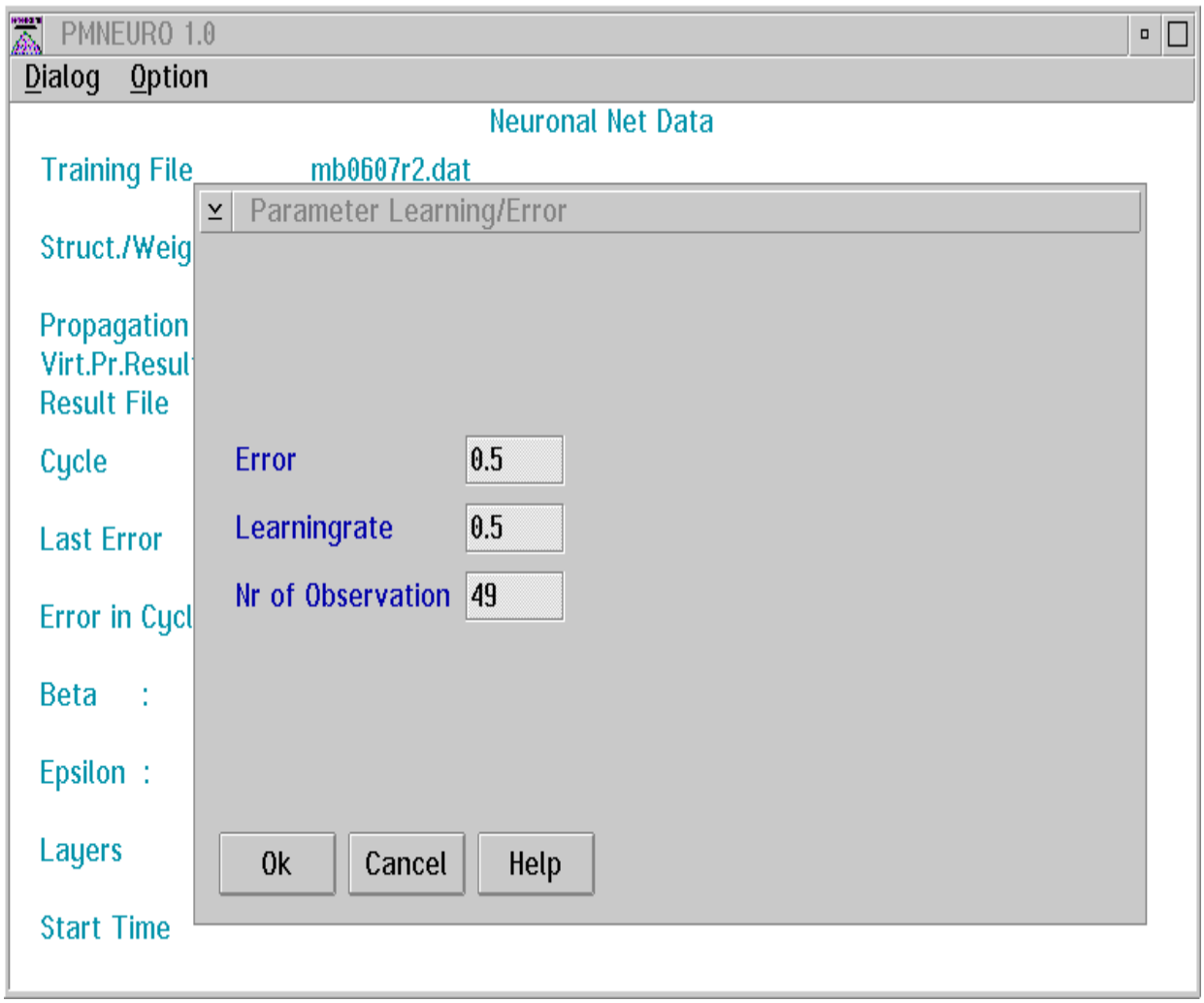
Method : Backpropagation



1. Description for : creating and traing a neuronal network
In pulldown 'Dialog' chose '1. New' then type in :

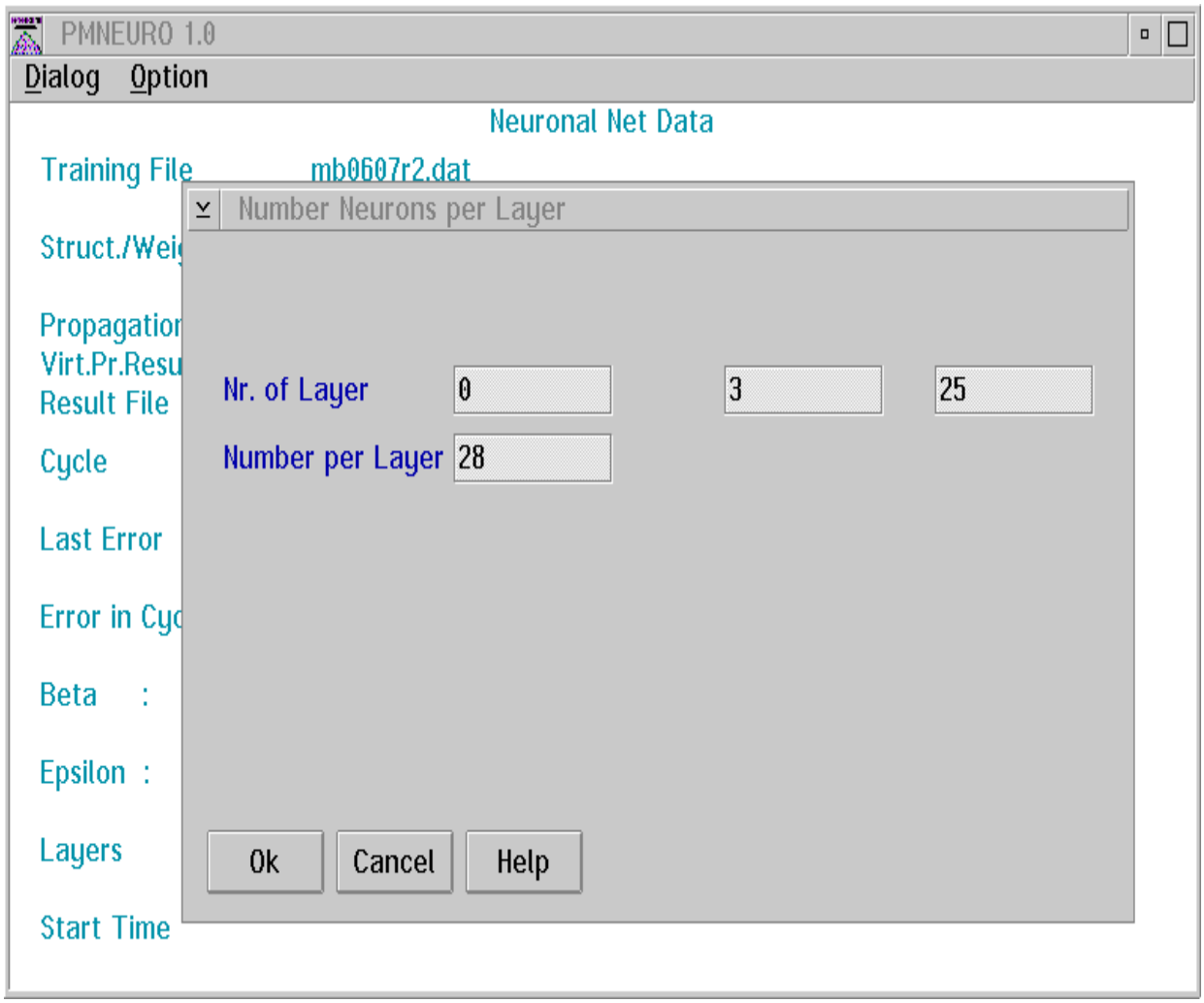
Training Data	name of the file with training data
Observations	number of combinations of trainingsets
Layers	number of layers including first and last layer
Entries	number of input values of an observation, it is the number of neurons in the first layer
Outputs	number of output values of an observation, it is the number of neurons in the last layer
Random	the weights in the network will be randomly initialised

After pushing 'OK' button, the next window pops up :



Error	value between 0.01 and 0.99
Learningrate	value between 0.01 and 0.99
Observations	number of observations/trainingsets in the data file

After pushing 'OK' button, the next window pops up :



Nr of Layer

field 1 value 0 means layer after the first layer (remember number of neurons in first layer are equal to number of input values / input layer)
value 1 means second layer after the first ..... value 2 means third layer after the first .....

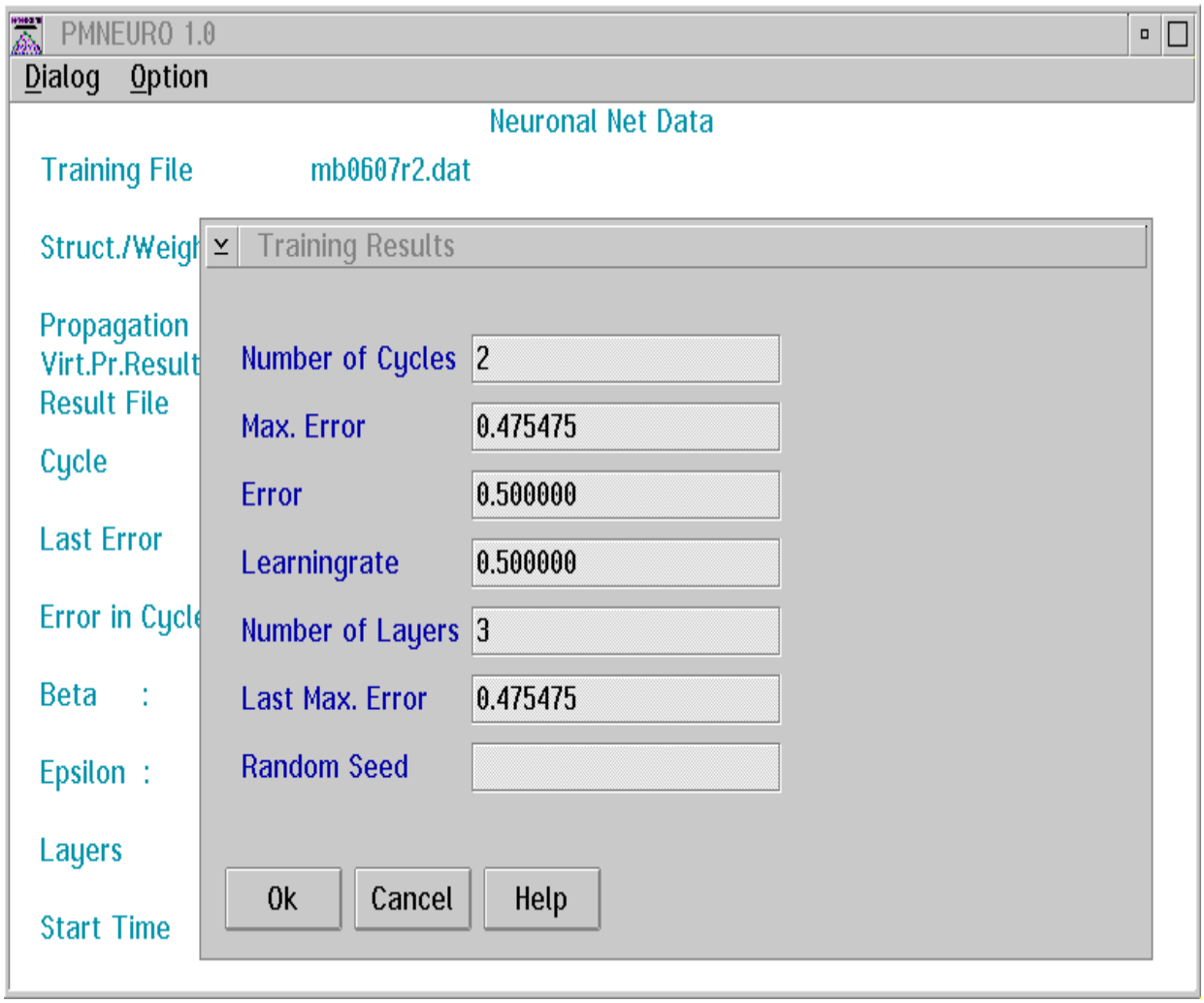
field 2 value 3 shows number of layers (input+output+middlelayer)
field 3 value 25 is the random value(in the example)

User choosen Number per Layer 28 neurons in this layer, in this field you type in the number of neurons in this layer

The structure of the network in this example :

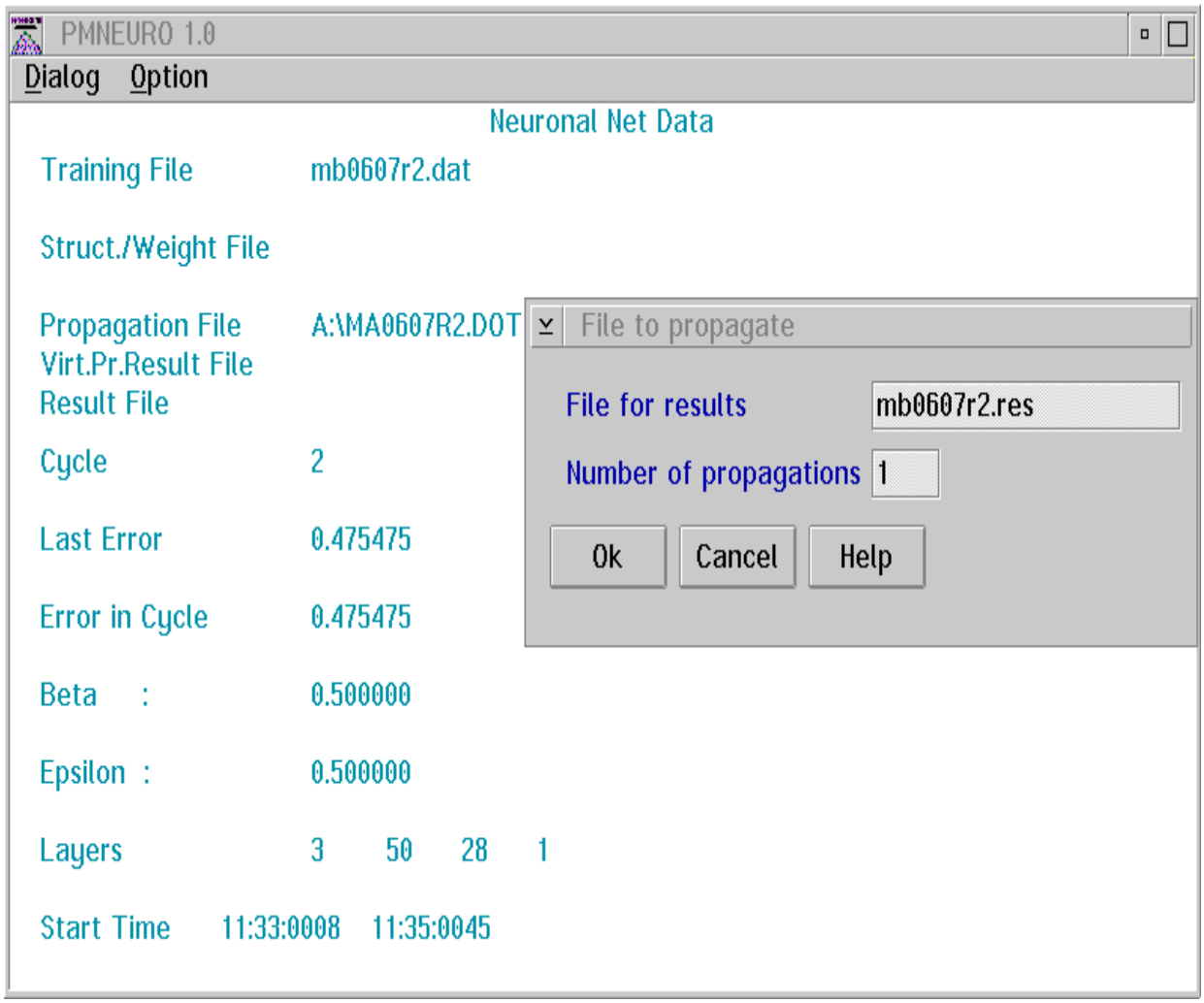
1 (Number of output in the outputlayer/last layer)
28 (Number of neurons in layer(x)
49 (Number of neurons in inputlayer/first layer)
3 is the number of layers !

the program asks for number of input neurons for all layers between first and last layer; this popup will be repeated after pushing 'OK' button for every layer .
After the last layer the training starts. Training progress can be seen in the chart window.



The training ends when

- the calculated error is smaller than 'Error' input value
- the number of cycles is greater than the 'abort after cycle' in 'Option'
  (trick : to stop calculating, make the value in abort after cycle smaller)

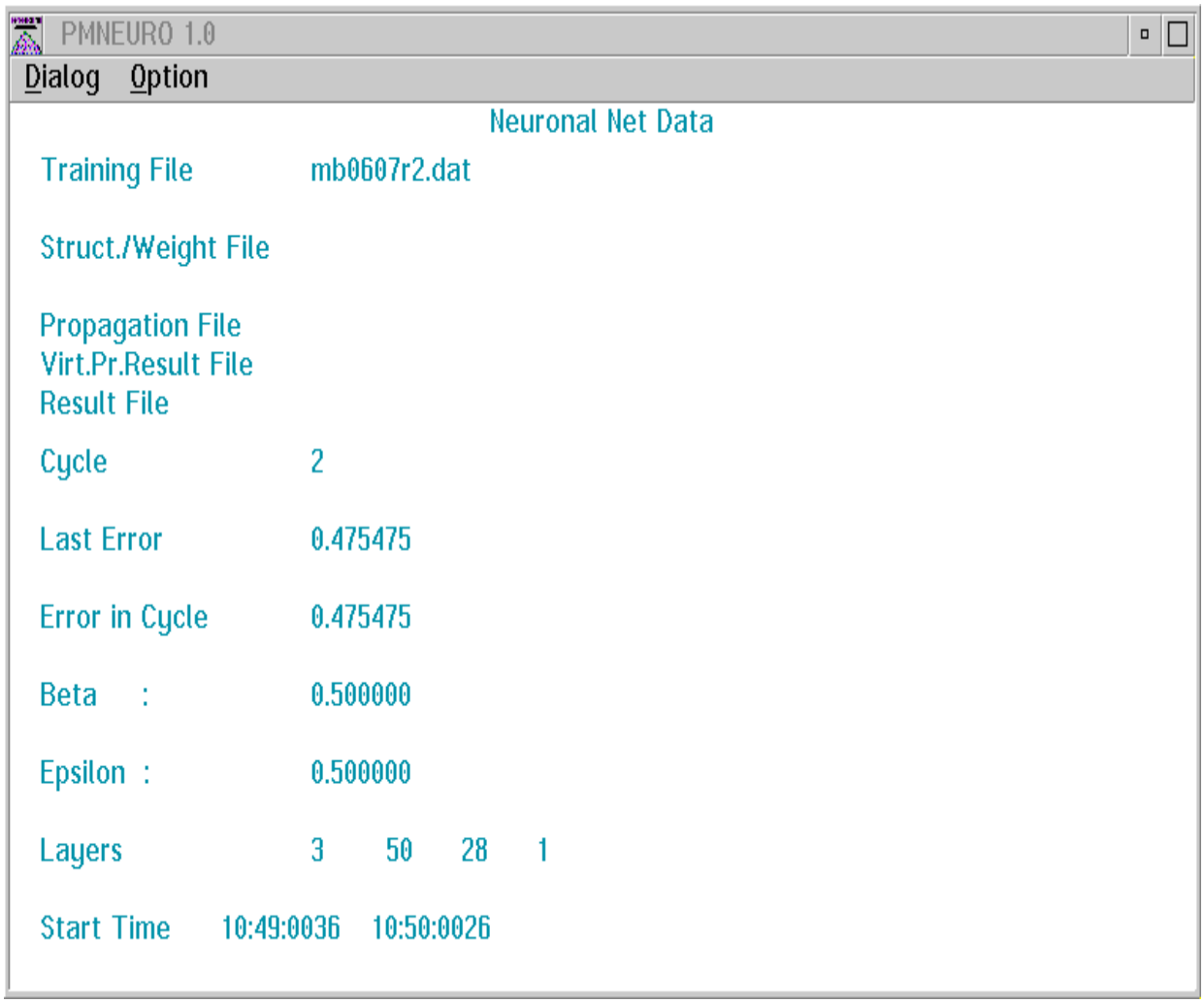


To propagate data, chose 'Dialog--> Propagate' after/during training. There will be a file dialog, asking for
the propagation data file and this little data window.

File for results result data (should be a non existing file, otherwise it will be overwritten!)
Number of prop. 1 means one propagation
> 1 means one propagation AND NOW the programs builds a new propagation
data set with the last propagation result and uses it for a new propagation until it
reaches the number of propagation !

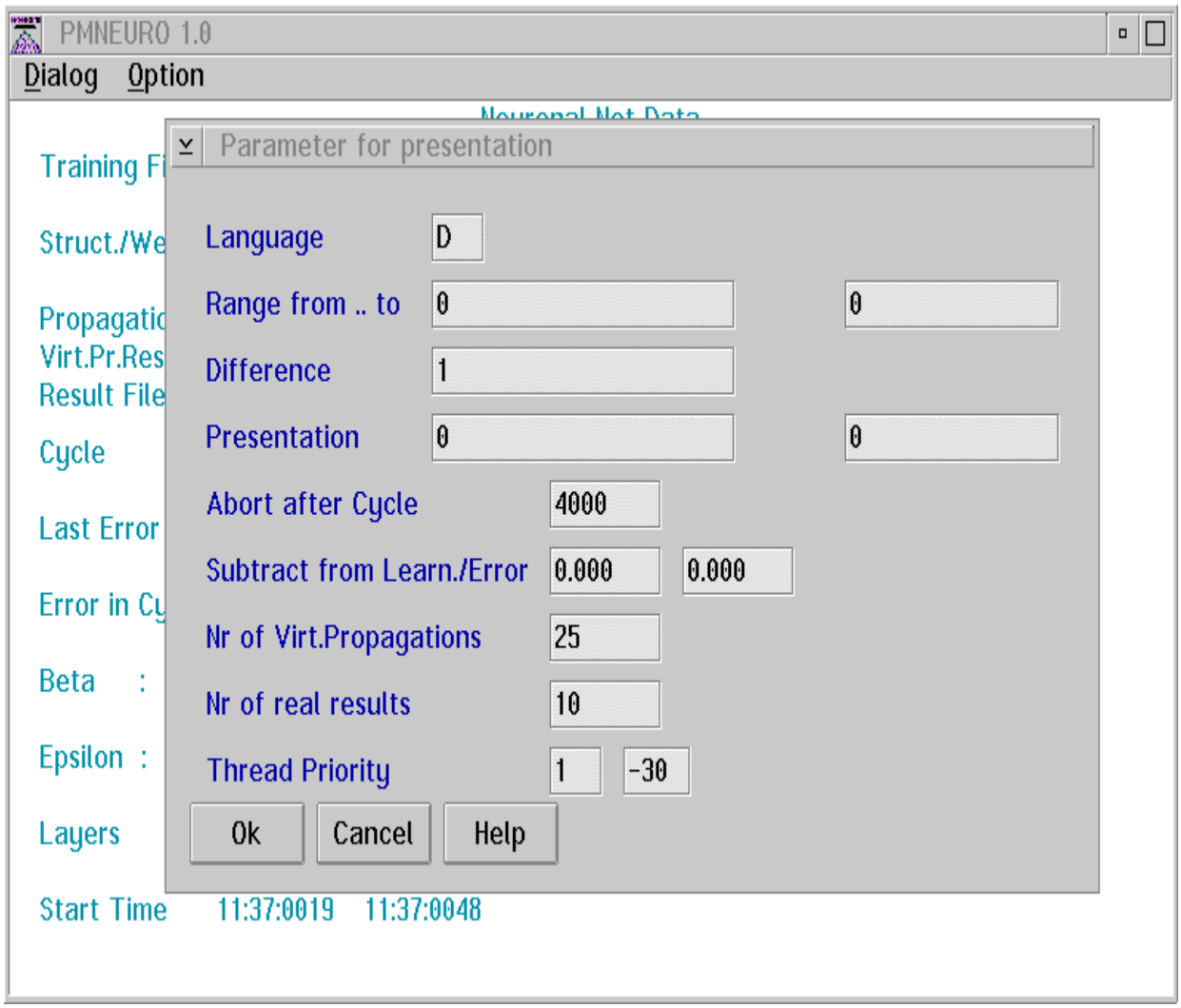
related to the example at the beginning of the text :

Propagate data 0.03 0.04 0.05 0.06 (original prop. data file)
after 1 prop 0.04 0.05 0.06 0.0678 as result (now the new prop. data file)
after 2 prop. 0.05 0.06 0.0678 0.0765 as result (")



During training and after end of training the values are shown in the program window

Do you know 'BIAS' value ? It's the 49+1 input neuron in the first layer therefore 3 50 28 1 !!



The 'Option-->Default' window.

Language	D = german, empty = english
Range from .. to	range of real results, they are shown on PMNEURO chart window
Difference	between two real results
Presentation	of the real results in a rectangle
Abort	you can train forever, or stop after this value
Nr of real results	under 'Option-->Compare' you can load a data file with real results, the values will be shown in the presentation rectangel
Thread	there are four (0,1,2,3) and (-31 --- +31) priority levels; but be warned, using or testing other values then (1,-30) can slow down other programs or the whole system !

Examples :
Creating new network :
(do not type in the commas)
1. Choose Dialog-->New
type in mb0607r2.dat , 49 , 3 , 49 , 1 , 8 push OK button
Parameter Learn./ Error
type in 0.8 , 0.5 , 49 push OK button
Number neurons per layer
type in 28 push OK button
Training results
push OK button

Save trained network :

2. Choose Dialog-->Save
type in mb0607r2.wei push OK button

Propagate after training and saving :

4. Choose Dialog-->Propagate
type in/select mb0607r2.dot push OK button

File for results
type in mb0607r2.rst , 1 push OK button

New features : make a new data set from propagation data and propagation result :

Dialog-->Virtuell

Make a new structure/weight network :

Dialog-->Virt.Stru

after Result window push OK button

Make a new propagation :

Dialog-->Virt.Prop ==> creates a new result file, named : virt1a.erg

The last three points can be done in a loop, using the value in
'Option-->Default-->Nr of virt propagations' and Dialog-->Virt.Cycle
default is 25 , this results in 25 new files virt1a.erg thru virt25a.erg

Important : there must be at first a normal propagation trial.

Showing propagation results and real results in the chart window :
Option-->Compare
type in/select mb0607r2.rea push OK button
Option-->Default
type in Range 1, 100
type in Presentation 5 , 20 push OK button