

APPENDIX B

Command and Variable Summary

In this appendix we give a brief description of all of the commands available in the PDP programs. With each command we indicate the program or programs the command is relevant to and the number of the page where the command was described. Where there are different variants, we describe these and indicate where each variant was described. Since this summary includes all of the subcommands of the *set/* command, it also provides a summary of all of the variables that are accessible in the programs.

clear all [30]

Clears the display area.

ctest aa [171]

Completion test. Used to see how well an auto-associator can fill in cleared parts of a specified pattern on the pattern list. Prompts for pattern name or number and elements to clear for completion testing.

cycle iac, cs, bp, ia [30]

Runs *ncycles* of processing. In **bp**, applies only in *cascade* mode.

do all [30]

Executes the commands in a file a specified number of times. Prompts for the file name and an integer number of repetitions.

fcspec ia [223]

Forced-choice specification for testing letter perception in the interactive activation model. Prompts for a letter position (0-3), a correct alternative, and an incorrect alternative.

input cs, iac [30]

Allows user to specify external inputs to units by name or number. First asks whether old specifications should be cleared (indicate *y* or *n*) then prompts for unit names or numbers, followed by an external input value. The value specified is scaled by the parameter *estr*. To terminate input, enter *end* or press *return*.

log all [30]

Used to open or close a log file. Prompts for a file name or "-". If a file name is entered, the existing log file is closed if there is one, and a new one is opened with the specified name. If "-" is entered, any existing log file is closed.

newstart cs, ia [56]

Allows the program to be retested on the same problem, but with a new series of pseudorandom numbers. Chooses a new random seed for the random number generator then reinitializes the activations of units in the network.

pa, bp, aa, cl [104]

For these programs, *newstart* also reinitializes the weights in the network. In **pa** and **aa** they are reset to 0s, in **cl** they are reset to random initial values, and in **bp** they are reinitialized as specified in the *.net* file.

print ia [223]

Prints activations of all word and letter units and response strengths of all letter units on the screen in the interactive activation model.

ptrain pa, bp, aa, cl [104]

Permuted training. Executes *nepochs* of training, presenting each pattern or pattern pair that is on the pattern list once in each epoch. A new random order of presentation is used on each epoch.

quit all [31]

Quits the program. Prompts for confirmation. Enter *y* to quit, anything else to cancel the command.

reset iac [31]

Resets activations of units in the network to resting level.

cs, ia [56]

Allows the program to be retested on the same problem, with the exact same pseudorandom number sequence used to determine order of *rupdate* in *cs* and other random aspects of processing. Reinitializes the random number generator to the value of the *seed* variable, which will be the same as the value used when the process was last initialized unless *seed* has been set manually to a new value.

pa, bp, aa, cl [104]

For these programs, *reset* also reinitializes the weights in the network. This means that for *bp* and *cl*, the same random starting weights that were used on the previous run will be used again. *Note:* The starting weights used for Ex. 5.1 must be read in from the file *xor.wts* after the *reset* command is entered, using the *get/weights* command.

run all [31]

Used to pass a command to the command interpreter under which you are running the simulation program. The command to be run is terminated by *end* or an extra *return*. Thus, to see if a file exists in your directory named *foo.wts*, you could enter *run dir foo.wts end*.

strain pa, bp, aa, cl [104]

Sequential training. Executes *nepochs* of training, presenting each pattern or pattern pair on the pattern list once in each epoch. Patterns are presented in the same fixed order on each epoch, as specified in the *.pat* file.

tall pa, bp, aa, cl [104]

Tests all patterns on the pattern list. Patterns are presented for testing one at a time, and the program pauses after each pattern to allow you to examine the results. Learning is turned off while *tall* is in progress.

test iac, cs, bp, cl [31]

Presents a pattern from the pattern list for testing. Prompts for the name or number of the pattern.

pa [105]

Allows testing of a pattern from the pattern list, a distortion of such a pattern, or a specific pattern as entered from the keyboard, as indicated by user's response to the prompt. The pattern is tested against a target pattern which may either be one of the targets from the pattern list, a distortion of such a target, or a specific pattern entered from the keyboard. If the user chooses to enter a pattern from the keyboard, the pattern must be terminated by *end* or an extra *return*.

aa [171]

Allows testing of a pattern from the pattern list, a distortion of such a pattern, or a specific pattern entered from the keyboard; also allows retesting of the last pattern tested. This is useful for studying the effects of a single training experience on a specific pattern. A pattern entered from the keyboard must be terminated by *end* or an extra *return*.

trial ia [223]

Allows user to set up a processing trial consisting of a sequence of displays in the interactive activation model. Each display consists of an onset time followed by a string of four characters indicating the contents of the display. If the characters are letters, they specify the corresponding feature patterns in the Rumelhart-Siple font. Other special characters are used for other feature patterns, as described in Chapter 7.

disp/ all [31]

Header for various specific commands that allow users to display information on the screen or to adjust options associated with displaying information to the screen.

disp/ state all [31]

Display the current state of the network. Clears the screen and then redisplay the current states of all of the templates that should be visible, given the current value of the *dlevel* variable.

disp/ <template> all [31]

Display *<template>*, where *<template>* is the name of one of the templates defined in the *.tem* file.

disp/ opt/ all [31]

Header for various commands that allow use to alter different display characteristics.

disp/ opt/ lthresh ia [225]

Allows user to set the minimum activation required for letters to be entered on the display list in the **ia** program. Only letters in the display list are displayed when the screen is updated. The *lthresh* is ignored when an explicit display list is provided by the user.

disp/ opt/ standout all [31]

Allows the user to set the value of the *standout* option for displaying negative numbers. If *standout* is set to 0, minus signs are used to indicate negative numbers where possible. When *standout* is set to 1, as it is by default, reverse video is used for negative numbers.

disp/ opt/ wthresh ia [225]

Allows user to set the minimum activation required for words to be entered on the display list in the **ia** program. Only words in the display list are displayed when the screen is updated. The *wthresh* is ignored when an explicit display list is provided by the user.

disp/ opt/ <template> all [32]

Allows the user to set various options associated with the specified *<template>*, where *<template>* is the name of one of the templates defined in the *.tem* file.

exam/ all [32]

A synonym for *set/*; it allows the user to examine the current value of a variable and to set it to a new value if desired. The subcommands available under both *set/* and *exam/* are described below under *set/*.

get/ all [32]

Header for commands that allow the user to specify lists of values with a single command or to specify a file from which to read a complex specification such as the network specification.

*get/ annealing***cs** [56]

Used to specify an annealing schedule for *boltzmann* and *harmony* modes in the **cs** program. The user is prompted for an initial temperature, then for a sequence of time-temperature pairs. The program interpolates linearly between these milestones, and stops adjusting temperature at the last milestone. Follow the last time-temperature pair with *end* or type an extra *return*.

*get/ dlist***ia** [225]

Used to specify a display list specifying which letter and word units will be displayed when the screen is updated in the **ia** program. The user is prompted for words and letters to place on display list.

*get/ network***iac, cs, pa, bp** [32]

Used to tell the program to read a network specification file and set up the network of units according to its contents. Prompts for the name of the network specification file.

*get/ patterns***iac, cs, aa, cl** [32, 171]

Used to tell the program to read a list of patterns from a pattern specification file. Each pattern specification is a sequence of entries consisting of a pattern name (a string of characters beginning with a nondigit) followed by a sequence of *ninputs* floating-point numbers corresponding to the elements of the pattern. Entries in the pattern file are separated by white space (spaces, tabs, or newlines).

pa [105]

As above, except that in **pa**, the program expects the pattern specification file to contain pattern pairs. Each pair has a sequence of entries consisting of a pattern name, followed by *ninputs* floating-point numbers corresponding to the elements of the input pattern, followed by *noutputs* numbers corresponding to the elements of the target pattern.

bp [142]

As with **pa**, with special handling of negative input and target entries. A negative input entry specifies that the input element should be the previous activation of the unit indexed by the absolute value of the entry. A negative target entry specifies a "don't care" element; that is, that no target value is to be specified for the corresponding output unit.

get/ rpatterns aa [172]

Used to generate a list of random patterns of +1s and -1s for testing in the auto-associator. Prompts for a number specifying the number of patterns to construct, and then prompts for a number specifying the probability that each element will be +1. Patterns are named rN , where N is an integer from 0 to $npatterns - 1$.

get/ unames all but ia [33]

Allows the user to specify a list of names for the units in the network. Each name is a sequence of characters, and successive names are separated by spaces, tabs or newlines. Follow the last entry with *end* or type an extra *return*.

get/ weights all but ia [33]

Used to tell the program to read a file containing weights for all of the connections in the network. For networks that have bias terms or sigma terms (this applies to *harmony* mode in *cs* only), these are also read in from the same file. Prompts for the name of the weight file. See Appendix C for further information about the format of the weight file.

save/ all [33]

Header for commands that allow user to tell the program to save various kinds of things in a file.

save/ patterns aa [172]

Allows the user to save the patterns on the pattern list in a file, together with their names, so that they can be read in later using the *get/ patterns* command. Prompts for the name of the file.

save/ screen all [33]

Allows the user to save the current screen display in a file. Redisplays the screen exactly as it would look in the file, then prompts for the name of the file.

save/ weights all but ia [33]

Allows the user to save the current values of the weights (and bias or sigmas if any) in a file, so that they can be read in again later using the *get/ weights* command. The format of the file is described in Appendix C. The command prompts for the file name.

set/ all [33]

Allows the user to examine the current value of a variable and to set it to a new value if desired. The commands *set/* and *exam/* are synonyms. The following list indicates the subcommands available under *set/*; all of these are in fact the names of variables. Variables whose names are followed by "*" are installed when the network is initialized, and so will not be visible before the program is initialized (e.g., right after the program is started up and before the *get/ network* command has been executed). For vector variables, an index must be given after the variable name to specify which entry is to be accessed. For matrix variables, two indexes must be given. The first corresponds to the row of the matrix and the second to the column.

*set/ bias** **bp, cs, pa** [57]

Vector of bias parameters associated with the units in the network.

set/ dlevel all [35]

The global display level variable. When the display is updated, all templates with display levels less than or equal to the global display level are displayed.

set/ ecrit **aa, bp, pa** [106]

Criterion value for the total sum of squares *tss* used to determine when to stop training. Training stops when the *tss* falls below the value of *ecrit*.

set/ lflag **pa, bp, aa, cl** [106]

A flag variable indicating whether learning (connection strength adjustment) should take place or not. Learning occurs when *lflag* is not equal to 0.

set/ ncycles **aa, bp, cs, ia, iac** [35, 142]

The number of processing cycles executed when the *cycle* command is entered or when the *test* command is issued. In **bp**, applies only in *cascade* mode.

set/ nepochs **pa, bp, aa, cl** [106]

The number of epochs of training that are carried out when the *ptrain* or *stratrain* commands are issued.

set/ nupdates cs [57]

The number of processing unit random updates that are executed in each processing cycle in the *cs* program. Except in *harmony* mode, *nupdates* is generally set equal to *nunits* so that each unit is updated an average of one time per cycle.

set/ seed all [36, 57]

The current value of the seed that is used when the random number generator is reinitialized. Note that the *newstart* command assigns a new random value to *seed* before reinitializing the random number generator.

*set/ sigma** cs [57]

Vector of sigma values associated with units in *harmony* mode of the *cs* program.

set/ single all [36]

A flag variable that determines whether the program will pause or not after each processing step, where the size of the step is determined by the *stepsize* variable defined below. Pausing occurs if *single* is nonzero.

set/ slevel all [36]

The global save level variable. When the display is updated, all templates with display levels less than or equal to the global *dlevel* variable and less than the global *slevel* variable are logged in the log file, if one is open.

set/ stepsize iac [36], cs [57], pa [106]
bp [142], aa [172], cl [197], ia [226]

The size of the processing step between display updates and pauses. The recognized values are *update*, *cycle*, *ncycles*, *pattern*, *epoch*, and *nepochs*.

*set/ weight** all but ia [36]

Matrix of values for the weights or connection strengths in the network. The row index corresponds to the receiving unit, and the column index corresponds to the sending unit.

set/ config/ all [34]

Header for variables that are relevant to the network configuration. Generally these are specified in the *.net* file.

*set/ config/ bepsilon** **bp** [143]

Vector of modifiability parameters, associated with bias terms in **bp**. See note under *set/ config/ epsilon* below.

*set/ config/ epsilon** **bp** [143]

Matrix of modifiability parameters associated with weights in **bp**. Note that nonzero modifiability parameters are set to the new value of *lrate* when *lrate* is changed.

set/ config/ ninputs all but **ia** [36, 107]

The number of input units. In **iac**, **cs** and **aa**, this is used only when reading patterns from a file, and is generally set equal to *nunits*. In **pa**, **bp**, and **cl**, it is used in specifying the network configuration as well as the elements in each input pattern.

set/ config/ noutputs **pa, bp, cl** [106]

The number of output units.

set/ config/ nunits all but **ia** [36, 107]

The total number of units in the network.

*set/ config/ uname** **cs, iac** [36]

Vector of unit names. Installed when the *get/ unames* command is used to specify a set of unit names.

set/ env/ all [34]

Header for variables associated with the learning environment.

set/ env/ ipattern **iac, cs, aa, cl** [36]
pa [106]
bp [143]

Matrix of input pattern element values. The row index corresponds to the pattern number, the column index corresponds to the element number within the pattern.

set/ env/ maxpatterns all but **ia** [37]

The maximum number of patterns that can be read in. This is generally automatically incremented when necessary, but if a large number of patterns are to be read in reading occurs faster if *maxpatterns* is preset to a number greater than or equal to the number of patterns in the pattern file.

set/ env/ npatterns all but **ia** [37, 106]

The number of patterns in the pattern list.

set/ env/ pname all but **ia** [36, 106]

Vector of names associated with the patterns on the pattern list.

set/ env/ tpattern **pa** [106]
bp [143]

Matrix of target pattern values. The row index corresponds to the pattern number, and the column index corresponds to the element number within the pattern.

set/ mode/ all [34]

Header for variables associated with various mode switches. These are generally used to select among variants of the same general model.

set/ mode/ boltzmann **cs** [57]

When nonzero, causes the **cs** program to simulate a Boltzmann machine.

set/ mode/ bsb **aa** [172]

When nonzero, causes the **aa** program to simulate the brain-state-in-a-box model.

set/ mode/ cascade **bp** [143]

When nonzero, causes the **bp** program to accumulate net inputs gradually rather than in a single step.

set/ mode/ clamp **cs** [57]

Determines whether external inputs are treated as specifying values that the specified units should be clamped with, or as specifying inputs to the units which are scaled by *estr* and then added into each unit's net input.

set/ mode/ comprp **ia** [226]

Determines whether response probabilities are calculated in the **ia** program. If *comprp* is set to 2, response probabilities are computed for letter and word units. If set to 1, they are computed for letter units but not for words. If set to 0, no response probabilities are computed.

- set/ mode/ cs* **pa** [106]
- When nonzero, causes **pa** to set the activations of output units according to the continuous values returned by the logistic function.
- set/ mode/ follow* **bp** [143]
- When nonzero, causes the **bp** program to follow the direction of the gradient from one weight update to the next by computing the correlation of the weight error derivative vectors used in each successive weight update.
- set/ mode/ gb* **iac** [37]
- When nonzero, causes updating of activations to be done according to Grossberg's update rule in the **iac** program.
- set/ mode/ harmony* **cs** [58]
- When nonzero, causes **cs** to simulate Smolensky's harmonium model.
- set/ mode/ hebb* **aa** [107]
pa [173]
- When nonzero, causes **pa** and **aa** to use the Hebb rule rather than the delta rule in learning.
- set/ mode/ lgrain* **bp** [144]
- Determines the grain of learning in **bp**. When *lgrain* is set to *pattern*, weights are incremented after each pattern is processed. When *lgrain* is set to *epoch*, weight error derivatives are accumulated over an entire processing epoch and then the weights are incremented only at the end of the epoch.
- set/ mode/ linear* **aa** [107]
pa [173]
- When nonzero, causes **pa** and **aa** to use a linear activation rule instead of the default activation rules for these programs. See the relevant chapter for the exact update formulas used in each case.
- set/ mode/ lt* **pa** [107]
- When nonzero, causes output units in the **pa** program to act like linear threshold units.

set/ mode/ selfconnect **aa** [173]

When nonzero, allows the **aa** program to adjust the strength of the connection to each unit from itself. Otherwise these connections stay fixed at 0.

set/ param/ **all** [34]

Header for parameter variables. These generally adjust *model* parameters, such as the rate of learning, rate of activation, etc.

set/ param/ alpha **iac** [37]

Scales the strength of excitatory input to each unit.

set/ param/ alpha/ **ia** [226]

Scales the strength of the excitatory input from the specified level to the specified level.

set/ param/ beta/ **ia** [226]

Scales the strength of the decay at the specified level.

set/ param/ crate **bp** [144]

Determines the rate of growth of the net input to each unit in cascade mode.

set/ param/ decay **aa, iac** [37]

Decay rate parameter.

set/ param/ estr **cs, iac** [37]
aa [173]

Scales the strength of external input (input from outside the network).

set/ param/ estr/ **ia** [227]

For each letter position, scales the strength of input from feature to letter units.

set/ param/ fgain **ia** [227]

Scales resting activation levels of word units based on word frequency.

<i>set/ param/ fdprob/</i>	ia [227]
Sets the probability that features are detected in each display field specified by the <i>trial</i> command.	
<i>set/ param/ gamma</i>	iac [37]
Scales the strength of inhibitory input to each unit.	
<i>set/ param/ gamma/</i>	ia [227]
Scales the strength of inhibitory input from the specified level to the specified level.	
<i>set/ param/ istr</i>	cs [58] aa [173]
Scales the strength of internal input (input from other units in the network).	
<i>set/ param/ kappa</i>	cs [58]
Threshold parameter used in <i>harmony</i> mode.	
<i>set/ param/ lrate</i>	pa [107], bp [144] aa [173], cl [197]
The learning rate parameter.	
<i>set/ param/ max</i>	iac [37]
Maximum activation allowed for units.	
<i>set/ param/ max/</i>	ia [227]
Maximum activation allowed for units at the specified level.	
<i>set/ param/ min</i>	iac [37]
Minimum activation allowed for units.	
<i>set/ param/ min/</i>	ia [227]
Minimum activation allowed for units at the specified level.	
<i>set/ param/ momentum</i>	bp [144]
Fraction of previous weight increment incorporated in each new weight increment on each weight adjustment.	
<i>set/ param/ mu</i>	bp [144]
Fraction of the unit's prior activation that is added in when the	

activation of the unit is set on the basis of prior activations, as indicated by negative elements in the input pattern. See pages 156-157 for further explanation.

- set/ param/ noise* **pa** [107]
 Amount of random noise added to each input and target value during learning.
- set/ param/ orate* **ia** [227]
 Rate of integration of time-averaged activation for computing response strengths.
- set/ param/ oscale/* **ia** [227]
 Scale factor used in relating time-averaged activations to response strengths.
- set/ param/ pflip* **aa** [173]
 Probability of flipping the sign of each input pattern element during learning.
- set/ param/ rest* **iac** [37]
 Resting activation level of units.
- set/ param/ rest/* **ia** [227]
 Resting activation of units at the specified level.
- set/ param/ temp* **pa** [107]
 Temperature parameter used in scaling net inputs. Not subject to annealing in **pa**.
- set/ param/ thresh/* **ia** [227]
 Threshold that units on the specified level must exceed before they generate output to units on the specified receiving level.
- set/ param/ tmax* **bp** [144]
 Actual target activation used for target elements set to 1.0 in the pattern list. Actual target for elements set to 0 in the pattern list is $1 - tmax$.
- set/ param/ wrange* **bp** [144]
 Range of variability used when initializing random weights.

<i>set/ state/</i>	all [34]
Header for variables associated with the current state of the network.	
<i>set/ state/ activation*</i>	aa, cs, cl, iac [37] bp [144]
Vector of unit activations.	
<i>set/ state/ bed*</i>	bp [144]
Vector of bias error derivatives.	
<i>set/ state/ cpname</i>	all but ia [38, 107]
The name of the pattern being tested or just tested.	
<i>set/ state/ cuname</i>	cs [58]
The name of the unit just updated.	
<i>set/ state/ cycleno</i>	aa, bp, cs, ia, iac [37]
The current cycle number. (In bp , relevant only in <i>cascade</i> mode).	
<i>set/ state/ dbias*</i>	bp [144]
Vector of most recent delta biases, that is, of changes made to the bias values.	
<i>set/ state/ delta*</i>	bp [144]
Vector of delta terms associated with each unit.	
<i>set/ state/ dweight*</i>	bp [145]
Matrix of most recent delta weights, that is, of changes made to the weights. Indexes as with weights.	
<i>set/ state/ epochno</i>	pa, bp, aa, cl [107]
The current epoch number.	
<i>set/ state/ error*</i>	pa [107] bp [145] aa [173]
Vector or error values associated with units. These have a slightly different meaning in each of the three cases.	

<i>set/ state/ excitation*</i>	iac [37]
Vector of excitatory inputs to each unit.	
<i>set/ state/ extinput*</i>	cs, iac [38] aa [173]
Vector of external inputs to each unit from outside the network.	
<i>set/ state/ gcor</i>	bp [145]
Correlation of the weight error gradient calculated for the most recent weight update with the gradient calculated for the preceding update.	
<i>set/ state/ goodness</i>	cs [58]
Degree of constraint satisfaction due to the current state of the network.	
<i>set/ state/ inhibition*</i>	iac [38]
Vector of inhibitory inputs to each unit in the network.	
<i>set/ state/ input*</i>	pa [107]
Vector of activations of input units.	
<i>set/ state/ intinput*</i>	aa, cs [173]
Vector of internal inputs to each unit from other units in the network.	
<i>set/ state/ ndp</i>	pa [107] aa [173]
Normalized dot product of the current activation pattern with the target pattern.	
<i>set/ state/ netinput*</i>	cs, cl, iac [38] pa [107] bp [145]
Vector of net inputs to each unit, including external inputs, internal inputs, and bias terms if any.	
<i>set/ state/ nvl</i>	aa [108] pa [173]
Normalized vector length of the output activation vector.	

<i>set/ state/ output*</i>	pa [108]
Vector of activations of output units. First element indexes the first output unit.	
<i>set/ state/ patno</i>	iac, cs, cl, aa [38] pa, bp [108]
Number of the current pattern being tested.	
<i>set/ state/ prioract*</i>	aa [174]
Vector of previous activation values of units in the network.	
<i>set/ state/ pss</i>	aa, bp, pa [108]
Sum of squared errors for the pattern most recently tested.	
<i>set/ state/ target*</i>	pa [108] bp [145]
Vector of target values for output units.	
<i>set/ state/ temperature</i>	cs [58]
Current temperature used to scale net inputs. The parameter is adjusted during simulated annealing.	
<i>set/ state/ tss</i>	aa, bp, pa [108]
Total sum of squares over all patterns tested in the current or just-completed epoch.	
<i>set/ state/ unitno</i>	cs [58]
The unit number of the unit most recently updated.	
<i>set/ state/ updateno</i>	cs [58]
The update number within the present processing cycle; equivalent to the number of updates done since the beginning of the cycle.	
<i>set/ state/ vcor</i>	pa [108] aa [174]
The vector correlation of the pattern of activation with the external input (in aa) or the target (in pa).	
<i>set/ state/ wed*</i>	bp [145]
Matrix of weight error derivatives. Indexes as with weights.	