Parameters for HyPerCol

```
    randomSeed                      seed the column's random number
generator.
    dt                              time increment per timestep
    nx                              number of horizontal pixels in the
column.
    ny                              number of horizontal pixels in the
column.
    numSteps                        number of timesteps (can be overridden
on command line)
    outputPath                      path for directory into which output
files will be written.
                                        The directory must exist; PetaVision
won't create it.
                                        (can be overridden on command line)
```

Parameters for layer types

```
HyPerLayer (can't be instantiated; only derived classes can be)
    restart                         Boolean; whether to load *_last.pvp
files at initialization
    nxScale                         scale factor between layer size and
column size:
    nyScale                         layer size = (n{x,y}Scale) * (column's
n{x,y})
    nf                              Number of features.  For images, use
nf=3 for RGB
    marginWidth                     Width of boundary in pixels
    writeStep                       How often to write updates
    mirrorBCflag                    Whether to use mirror boundary
conditions at edges
    spikingFlag                     Whether there is spiking
    writeNonspikingActivity         Whether to write activity at each step
in the nonspiking case
    Vrest                           Sets initial value of clayer->V
ANNLayer
    VMax
    VMin
    VThreshold
GeislerLayer
GenerativeLayer
    relaxation                      GenerativeLayer rule: V(new) = V(old) +
relaxation*dV/dt
LogLatWTAGenLayer
PoolingANNLayerType
    bias                            The update rule is the pointwise product
```

```
of exc*inh*(bias*exc+(1-bias)*inh)
                                        exc is the excitatory channel; inh
is the inhibitory.
TrainingLayer
    traininglabelsPath              The path to the file containing the list
of training labels

    displayPeriod                   Display period for each training label
    distToData                      Delay between top of hierarchy and this
layer
PtwiseProductLayer
HMaxSimple
Image
    imagePath (string)              The path to the file containing the list
of images

    writeImages                     Whether to write images (not used by
Image; used by Movie and possibly other derived classes)
Bars
    initialPosition                 Initial position.  It will be rounded
toward zero and then moduloed the size in the appropriate dimension
    initialOrientation              Zero value or "false" = vertical;
nonzero or "true" = horizontal
    pSwitch                         Probability of switching from horiz to
vert or vice versa
    pMove                           Probability of changing position without
changing orientation
    barWidth                        Width of the bar
    barStep                         Step size
    randomWalk                      If changing position, 50/50 chance of
moving each direction
    moveForward                     If changing position, increase x or y
    moveBackward                    If changing position, decrease x or y
    randomJump                      If changing position, jump randomly
CreateMovies
    displayPeriod
    foregroundval
    backgroundval
    isgray
    rotateangle
    centerx
    centery
    period
    linewidth
    vx
    vy
    vr
    isshiftx
    isshifty
    isrotate
ImageCreator
```

```
Movie
    imagelistPath (string)          The path to the file containing the list
of images

    displayPeriod                   Length of time each image in the movie
is presented.
    offsetX                         Horizontal offset into image
    offsetY                         Vertical offset into image
    jitterFlag                      Whether to use jitter
    stepSize                        If jittering, amount to move
    recurrenceProb                  If jittering, prob. that offset returns
to bias position
    persistenceProb                 If jittering, prob. of no jitter when
updateImage is called
    biasChangeTime                  If jittering, how often to update bias
position
    writePosition                   If jittering, whether to write bias and
offset to image-pos.txt
    randomMovie                     Whether to call randomFrame()
    randomMovieProb                 If randomMovie is set, prob. that a
given pixel is 1.
LGN
LIF
    Vexc
    Vinh
    VinhB
    tau
    tauE
    tauI
    tauIB
    VthRest
    tauVth
    deltaVth
    noiseAmpE
    noiseAmpI
    noiseAmpIB
    noiseFreqE
    noiseFreqI
    noiseFreqIB
Retina
    poissonEdgeProb
    poissonBlankProb
    noiseOnFreq
    noiseOffFreq
    beginStim
    endStim
    burstFreq
    burstDuration
```

Parameters for connection types

```
HyPerConn
    preLayerName (string)           Connection's presynaptic layer's name
    postLayerName (string)          Connection's postsynaptic layer's name
    channelCode                     Connection's channel code (0=Exc, 1=Inh, 
2=InhB)

    nxp                             Number of horizontal pixels in each 
patch
    nyp                             Number of vertical pixels in each patch
    nfp                             Number of features in each patch
    writeStep                       How often to write updates for 
intermediate timesteps
                                    If negative, don't write intermediate 
timesteps.
                                    Setting to zero (or any value <= 
column's dt) will write every timestep.
    delay                           Delay between presynaptic activity and 
effect on postsynaptic layer
    rmax                            maximum connection distance; doesn't 
appear to be used
    wMax
    wMin
    // If the connection is created with a filename, the weights are 
initialized from the filename,
    // and initFromLastFlag, randomFlag, smartWeights, cocircWeights are all 
ignored.
    // Otherwise, the flags above are checked in the order listed below; 
once one flag is
    // true, all flags afterward are ignored.  If none of the flags are 
true, gauss2DCalcWeights
    // is called.
    // Regardless of the initialization method used, if normalize is true, 
the initialization
    // routine calls normalizeWeights.
    initFromLastFlag                Boolean: whether to initialize weights 
with data from previous run
    randomFlag                      Boolean: whether to initialize with 
random weights
    uniformWeights                  Boolean: if initializing with random 
weights, whether to use uniform distribution
    wMinInit                        If uniformWeights is true, the minimum 
value for the initial weights
    wMaxInit                        If uniformWeights is true, the maximum 
value for the initial weights
    gaussianWeights                 Boolean: if initializing with random 
weights, whether to use Gaussian distribution
    wGaussMean                      If gaussianWeights is true, the mean 
value for the initial weights
```

```
    wGaussStdev                     If gaussianWeights is true, the standard deviation for the initial weights
    smartWeights                    Boolean: whether to initialize weights with initializeSmartWeights
    cocircWeights                   Boolean: whether to initialize weights with initializeCocircWeights
    noPre                           If cocircWeights is true, # of pre-synaptic features used in cocircCalcWeights
    noPost                          If cocircWeights is true, # of post-synaptic features used in cocircCalcWeights
    sigmaCocirc                     If cocircWeights is true,
    sigmaKurve                      If cocircWeights is true,
    sigmaChord                      If cocircWeights is true,
    deltaThetaMax                   If cocircWeights is true,
    cocircSelf                      If cocircWeights is true,
    deltaRadiusCurvature            If cocircWeights is true,
    no                              If gauss2DCalcWeights is called, number of orientations (not used)
    strength                        Used if gauss2DCalcWeighs is used (also used if normalize is true)
    aspect                          If gauss2DCalcWeights is used, aspect ratio of the Gaussian
    sigma                           If gauss2DCalcWeights is used, standard deviation
    rMax                            If gauss2DCalcWeights is used, cut-off for argument of the exponential
    numFlanks                       If gauss2DCalcWeights is used, number of Gaussian profiles
    flankShift                      If gauss2DCalcWeights is used, shift of each Gaussian profile
    rotate                          If gauss2DCalcWeights is used, rotation of Gaussian profiles
    normalize                       Boolean: whether to call normalizeWeights during initialization
    strength                        Used if normalize is true (also used if gauss2DCalcWeights is called)
    normalize_max                   If normalize is used,
    normalize_zero_offset           If normalize is used,
    normalize_cutoff                If normalize is used,
    // End of initializeWeights-related parameters.
AvgConn
    maxRate
ConvolveConn
KernelConn
    symmetrizeWeights               Boolean: whether to symmetrize weights
CocircConn
GaborConn
    lambda
    invert
GeislerConn
GenerativeConn
```

```
    weightUpdatePeriod              How many timesteps between updating the
weights
    relaxation                      Coefficient on derivative used in
gradient descent Wnew=Wold-relaxation*(dF/dW)
    activityThreshold               Activities with absolute value below
this value are zeroed out
    nonnegConstraintFlag            Values of weight constrained to be
nonnegative
    normalizeMethod                 The method for normalizing.
                                    0 do not normalize
                                    1 HyPerConn::normalizeWeights
                                    2 Sum of squares across patches
normalized
    normalizeConstant               Value of sum of squares when
normalizeMethod=2
FeedbackConn
    originalConnName (string)       Name of connection being transposed

    // pre and post layer names inferred from feedforward name
    // many parameters inferred from feedforward connection
TransposeConn
    originalConnName (string)       Name of connection being transposed

    // pre and post layer names need to be specified (don't confuse with
TransposeConn)
    // many parameters inferred from original connection
PoolingGenConn
    secondaryPreLayerName (string)  Name of secondary pre-synaptic layer
    postSecondaryLayer (string)     Name of secondary post-synaptic layer
LateralConn
IdentConn
    // nxp not used (must be 1)
    // nyp not used (must be 1)
    // nfp not used (must be number of features in pre&post-synaptic layers,
which must be equal)
KernelCopyConn
    originalConnName (string)       Name of original connection

    // nxp, nyp, nfp inferred from original connection
PoolConn
RuleConn
STDPConn
    stdpFlag (boolean)
    ampLTP                          If stdpFlag is set,
    ampLTD                          If stdpFlag is set,
    tauLTP                          If stdpFlag is set,
    tauLTD                          If stdpFlag is set,
    dWMax                           If stdpFlag is set,
    localWmaxFlag                   If stdpFlag is set,
SubunitConn
```

Parameters for ColProbes

```
ColProbe
    colProbeTypeCode
    colProbeSerialNo
GenColProbe
```

Parameters for ConnectionProbes

```
ConnectionProbe
```

Parameters for LayerProbes

```
LayerProbe (can't be instantiated; only derived classes can be)
    layerProbeTypeCode
    layerProbeSerialNo
    targetLayer
L2NormProbe
    coeff               The coefficient that multiplies
\|targetLayer\|_{L2}/2
                            This term is not used by the L2NormProbe class
                            directly, but is read by pv.cpp if the probe is
                            inserted into a GenColProbe.

SparsityTermProbe
    coeff               The coefficient that multiplies the Olshausen-Field
                            term log(1+x^2)
                            This term is not used by the L2NormProbe class
                            directly, but is read by pv.cpp if the probe is
                            inserted into a GenColProbe.
LogLatWTAProbe
    coeff               The coefficient that multiplies the
                            lateral-competition term log(1+a'*Lslash*a)
                            This term is not used by the L2NormProbe class
                            directly, but is read by pv.cpp if the probe is
                            inserted into a GenColProbe.
VProbe
```

Codes for channels

- 0 Excitatory channel channelExc
- 1 Inhibitory channel channelInh
- 2 Inhibitory channel B channelInhB