Argument information for pyramidal neuron simulation

LSS 19 February 2019.

Last update LSS 20 March 2019

public static void main(java.lang.String[] args)

throws java.io.IOException

***Parameters (in alphabetical order: )***

-alpha\_context followed by alpha value for contextual synapses: default 1000

-alpha\_driver followed by alpha value for driving synapses: default 1000

-alpha\_internal\_excitatory followed by alpha value for internal excitatory synapses: default 900

-alpha\_internal\_inhibitory followed by alpha value for internal inhibitory synapses: default 200

-apical\_gradient followed by apical gradient for apical dendrite: default 1

-apical\_multiplier followed by apical multiplier for apical dendrite: default 1

-axon\_threshold followed by axon threshold: default 1 (named pyr\_threshold)

-c input spike file name, get file name for external contextual spike inputs

-d followed by input spike file name, so get file name for external driving spike inputs

-debug followed by 1 (true) or 0 (false) to set debug on/off: default 0

-fileprefix followed by a string, to be prepended to file names: must be before other file names

-i\_refractory\_period followed by inhibitory neuron refractory period: default 0

-inhibitory\_threshold followed by inhibitory neuron threshold: default 1

-logisticGradientBasal followed by double defining logistic gradient for basal compartment

-logisticGradientTuft followed by double defining logistic gradient for apical tuft compartment

-logisticInterceptBasal followed by double defining logistic intercept for basal compartment

-logisticInterceptTuft followed by double defining logistic gradient for apical tuft compartment

-n followed by network specifier

-p\_refractory\_period followed by pyramidal neuron refractory period: default 0

-s followed by sampling rate (defaults to 10000)

-sout followed by spike output file name: will be csv, (neuron, time)

-t followed by end time (defaults to 5.0)

-t\_apical followed by time constant (tau) for basal dendrite: default 0.1

-t\_basal followed by time constant (tau) for basal dendrite: default 0.1

-t\_inhib followed by time constant (tau) for simple leaky compartment used in inhibitory neurons: default 0.2

-tf2\_k1 followed by K1 value to use when transferfunction==2 is selected: default 0.5

-tf2\_k2 followed by K2 value to use when transferfunction==2 is selected: default 1

-transferfunction followed by 1 (original) or 2 (Kay and Phillips 2011) apical dendrite and axon hillock function selector: default currently 1

-v followed by verbosity: controls amount of system.out data created: default 1

-wc followed by weight file for contextual inputs

-wd followed by weight file for driving inputs

-wi followed by weight and delay file for internal synapses

***Or, as used in RunSpikeSimulator.m:***

% default values for all the parameters

% more common ones

% fileprefix followed by a string, to be prepended to file names: must be before other file names

% for peseta

% fileprefix = '/Users/lss/Documents/workspace/PyramidalCells/Test\_nov2018/' ;

% for laptop

fileprefix = '/Users/lss/Documents/Research/neuronsimulation/PyramidalCells/Test\_Feb20\_2019/' ;

% c input spike file name, get file name for external contextual spike inputs

c = 'contextspikes.csv' ;

% d followed by input spike file name, so get file name for external driving spike inputs

d = 'drivingspikes.csv' ;

% n followed by network specifier

n = 'networkconfig.txt' ;

% sout followed by spike output file name: will be csv, (neuron, time)

sout = '' ;

% snumbersout followed by name of file to write number of spikes emitted

% to

snumbersout = '' ;

% wc followed by weight file for contextual inputs

% wc = '' ;

wc = 'contextweights.txt' ;

% wd followed by weight file for driving inputs

wd = 'drivingweights.txt' ;

% wd = '' ;

% wi followed by weight and delay file for internal synapses

wi = 'internalweights.txt' ;

% wi = '' ;

% t followed by end time (defaults to 5.0)

t = 5.0 ;

% s followed by sampling rate (defaults to 10000)

s = 10000 ;

% actual command to run opyramidal neuron simulation

commandtorun = 'java -jar pyramidal.jar ' ;

% less common ones

% alpha\_context followed by alpha value for contextual synapses: default 1000

alpha\_context=100 ; % 400

% alpha\_driver followed by alpha value for driving synapses: default 1000

alpha\_driver = 1000 ; % 400

% alpha\_internal\_excitatory followed by alpha value for internal excitatory synapses: default 900

alpha\_internal\_excitatory = 900 ;

% alpha\_internal\_inhibitory followed by alpha value for internal inhibitory synapses: default 200

alpha\_internal\_inhibitory = 200 ;

% apical\_gradient followed by apical gradient for apical dendrite: default 1

apical\_gradient = 1 ;

% apical\_multiplier followed by apical multiplier for apical dendrite: default 1

apical\_multiplier = 1 ;

% axon\_threshold followed by axon threshold: default 1 (named pyr\_threshold)

axon\_threshold = 1 ;

% i\_refractory\_period followed by inhibitory neuron refractory period: default 0

i\_refractory\_period = 0 ;

% inhibitory\_threshold followed by inhibitory neuron threshold: default 1

inhibitory\_threshold = 1 ;

% p\_refractory\_period followed by pyramidal neuron refractory period: default 0

% values for logistic functions

logisticGradientBasal = 1.0 ;

logisticGradientTuft = 1.0 ;

logisticInterceptBasal = 0 ;

logisticInterceptTuft = 0 ;

p\_refractory\_period = 0 ;

% t\_apical followed by time constant (tau) for apical dendrite: default 0.1

t\_apicaltuft = 0.1 ;

% t\_basal followed by time constant (tau) for basal dendrite: default 0.1

t\_basal = 0.1 ;

% t\_inhib followed by time constant (tau) for simple leaky compartment used in inhibitory neurons: default 0.2

t\_inhib = 0.2 ;

% transferfunction controls effect of apical tuft & axon hillock: 2 is Kay

% & Phillips 2011

transferfunction = 1 ;

% relevant only when transferfunction == 2

tf2\_k1 = 0.5 ;

tf2\_k2 = 2 ; % defaults from K&P 2011

% verbosity: v controls the amout of output generated

v = 1 ;

% debugging: debug controls whether the program outputs debug info (1) or

% not (0)

debug = 0 ;