Development of Complex Curricula for Molecular Bionics and Infobionics Programs within a consortial* framework**

Consortium leader

PETER PAZMANY CATHOLIC UNIVERSITY

Consortium members

SEMMELWEIS UNIVERSITY, DIALOG CAMPUS PUBLISHER

The Project has been realised with the support of the European Union and has been co-financed by the European Social Fund ***

**Molekuláris bionika és Infobionika Szakok tananyagának komplex fejlesztése konzorciumi keretben

***A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

MODELLING NEURONS AND NETWORKS

(Idegsejtek és neuronhálózatok modellezése)

Lecture 11
Emergent simulation environment exercises, part 1

(1. Emergent gyakorlat)

Szabolcs Káli

Overview

This lesson is the first part of the Emergent simulation environment exercises. You will learn about:

- The Emergent simulation environment
- Modeling inhibition in cortical networks
- Roles of feedforward and feedback inhibition
- Effects of learning on inhibitory networks
- Bidirectional excitation
- ...

The Emergent simulation environment

- Full-featured neural network simulator
- Simple drag-and drop interface
- Scripting language: C Super-script (CSS): syntax is between C and C++, used to automate tasks and extend functionality of the software
- Networks can be inspected in 3D
- Supports a 3D physics engine for Newtonian physics simulations
- Descendant of PDP and PDP++, in development for decades
- Has been used to simulate complex cognitive phenomena such as
  - Visual object recognition and spatial attention
  - Long-term and short-term memory
  - Language (word representation, phonology, semantics)
  - Frontal lobe executive functions

Installing

Homepage: http://grey.colorado.edu/emergent/index.php

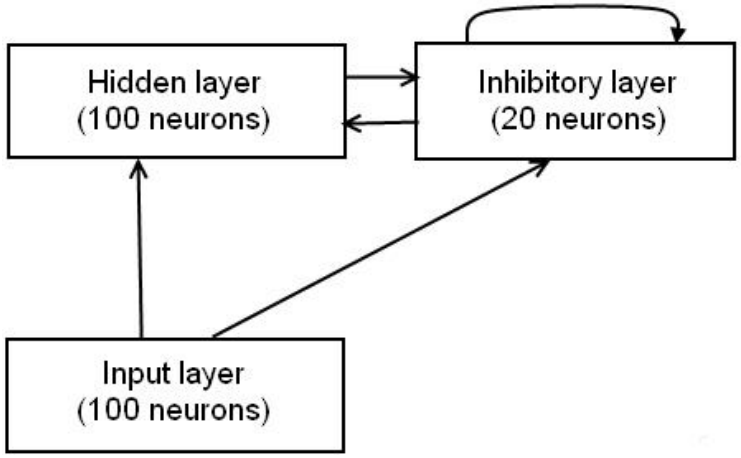
- Available for Linux, Windows, OSX
- Follow instructions on webpage for installing

Note: there is a windows bug in the current version (5.1.0): If you get a CSS_NUMBER syntax error on line 000001 when loading a project (and emergent freezes), and you are using some non-English input formats (like Hungarian), do the following:
1.Go to the control panel, then regional and language options.
2.Click customize in the Regional options tab
3.Set the decimal symbol to a dot („.” character) in the Numbers tab.
4.Click Ok twice to close the regional options and the custom options tab, then restart emergent.

Modeling inhibition in cortical networks

Download the simulation file (inhib.proj) from http://grey.colorado.edu/CompCogNeuro/index.php/CECN1_Inhibition, and open it (File – Open). This exercise will follow the instructions written at this website.

The network structure of the model is the following:



Neurons in the model are passive point neurons, characterized by a membrane potential and a continuous activation variable which takes values between 0 and 1. They receive conductance-based inputs, which determine their membrane potential, and the activation is then a sigmoidal function of the membrane potential.

Modeling inhibition in cortical networks

First examine the weights of the network:

- Press the *r.wt* button in the right *FFExcite* tab, this will allow you to display input weights to the neurons.
- Click on neurons in the hidden or inhibitory group to display their weights. Most of the weights are random, except those from inhibitory units, which are set to 0.5.

Then press the *act* button in the *FFExcite* tab to display the average activation levels for the neurons.
Next run the simulation by switching to the Control panel tab in the middle, pressing *init*, then *run*.

The plotted window displays that the average inhibition is 20 percent (red line, 100 percent=all inhibitory neurons at maximum activation) and the average excitation is around 10 percent (black line).

Emergent simulation environment

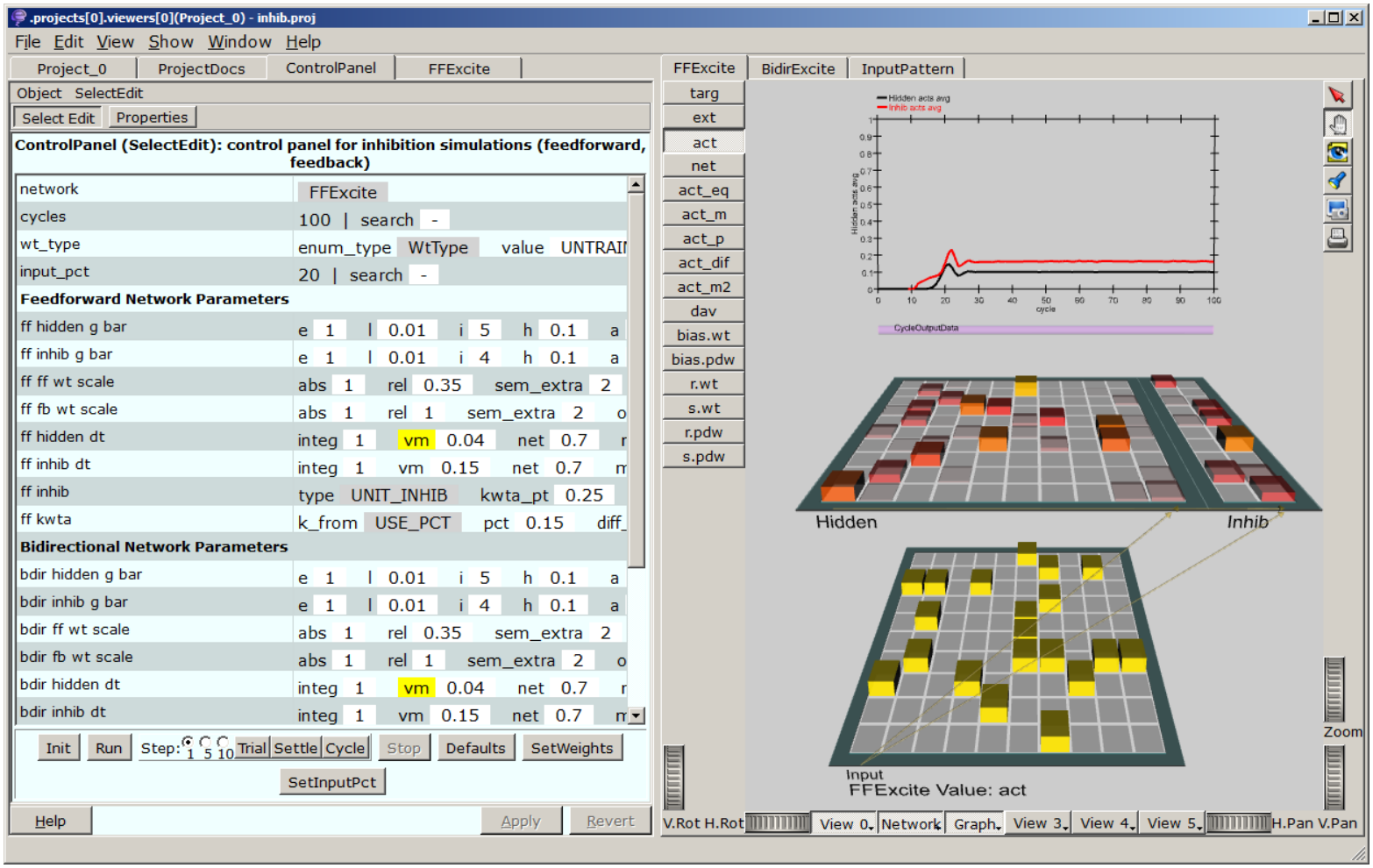


Figure: the inhibition simulation after running with default parameters

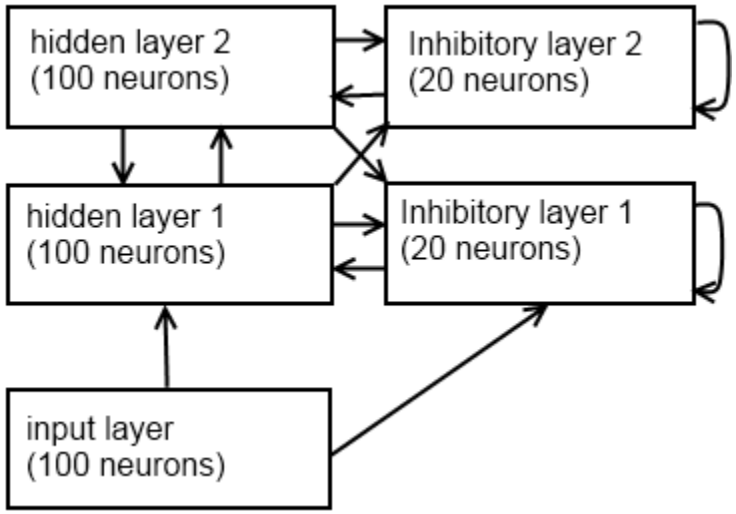
Exercises for the inhibitory network simulation

Load default parameters by pressing Defaults after each exercise.

1. Try increasing and decreasing the inhibitory conductance on excitatory neurons (*ff_hidden_g_bar.i* in the control panel tab) and observe the effect on average activity. You can try the same manipulation with inhibitory neurons (*ff_inhib_g_bar.i*). Press enter or Apply to apply changes. What are the effects of changing *ff_hidden_g_bar.i*? What are the effects of changing *ff_inhib_g_bar.i*?
2. Eliminate the feedforward excitatory inputs to the inhibitory neurons by setting ff_ff_wt_scale.rel to 0. Then reset the simulation and set the feedback weights from the hidden layer to the inhibitory neurons to 0 by setting ff_fb_wt_scale.rel to 0. What is the role of the feedforward and feedback inputs in controlling activity levels?
3. What is the relevance of the faster time scale of inhibition (try setting ff_inhib_dt.vm to 0.04)? What happens if both updates are fast (*ff_hidden_dt.vm*: 0.1, *ff_inhib_dt.vm*: 0.2)?

Exercises: Bidirectional excitation

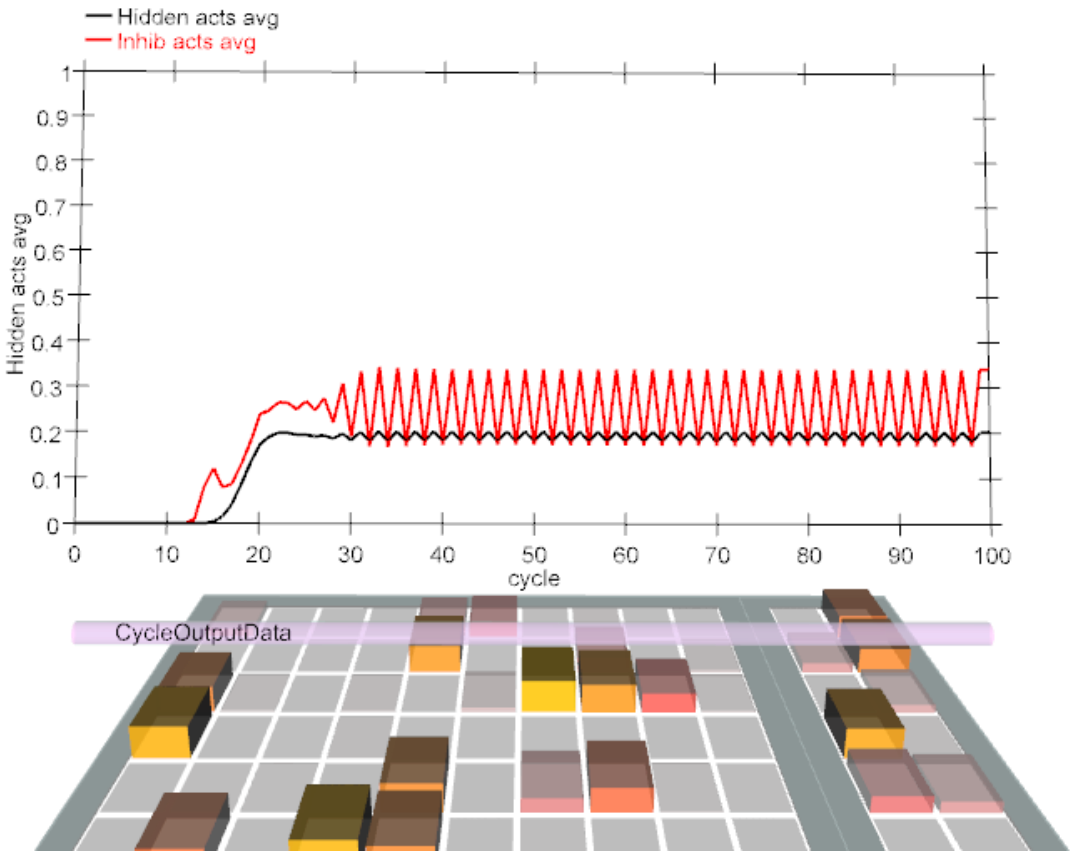
We now add a second hidden layer to the model network:



Exercises: Bidirectional excitation

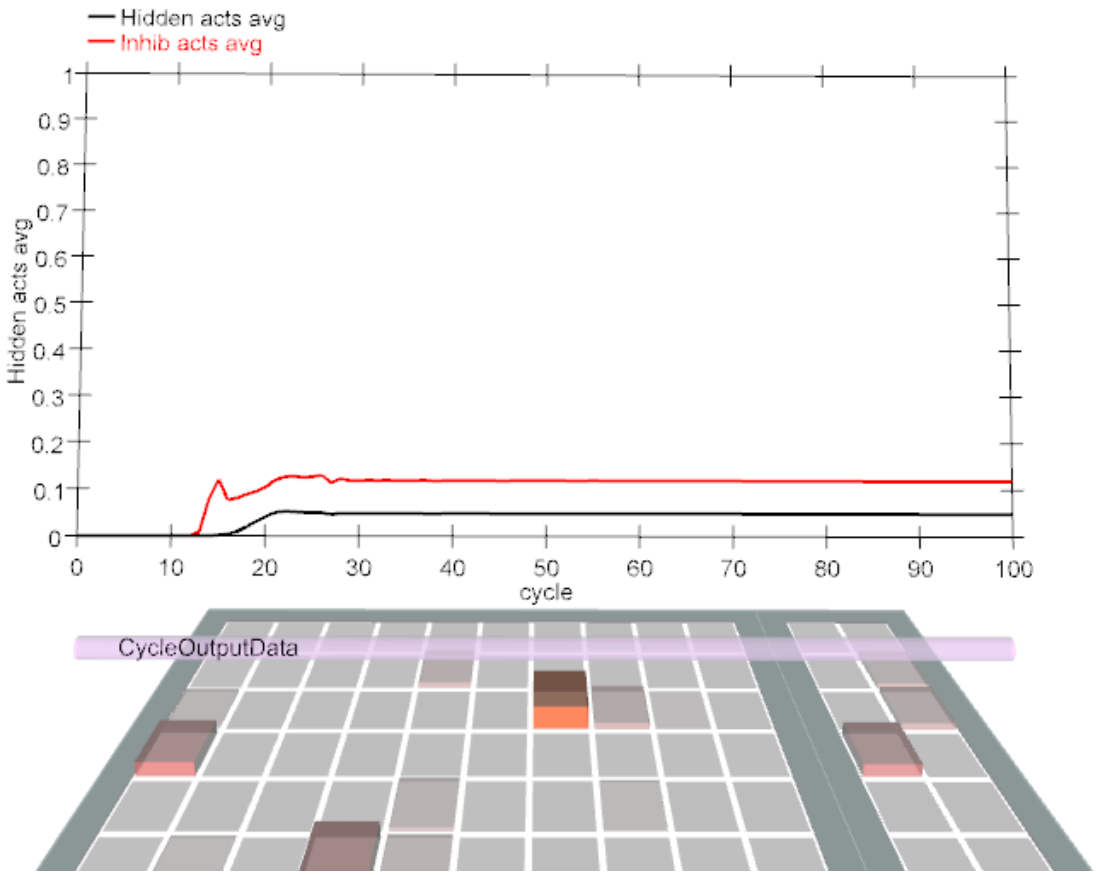
4. Run once to get a baseline. Then set *wt_type* to TRAINED and press the setWeights button to simulate the effects of learning, and Run the simulation again. Do you see any change in the activity level? How would you restore the same activity level? (Weights in the trained setting have increased variance and mean.)
5. Once again, do a default run for comparison, and then change the network to BidirExcite in the control panel (press Apply). Look at the weights and then Run this network. Now decrease *bd_hidden_g_bar i* to 3, and rerun the simulation. What happens?
6. Set *bd_hidden_g_bar_i* back to 5, and do a comparison run. Then change the number of active input neurons (*input_pct*) from 20 to 15, hit Apply, SetInputPct and then Run. Also try increasing the number of active inputs.

Solutions for the inhibitory network simulation



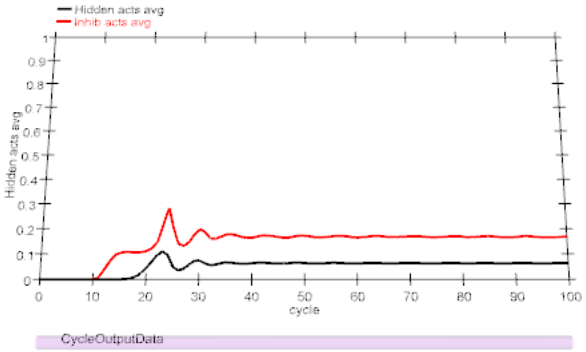
Exercise 1: If the maximal inhibitory conductance on excitatory neurons is decreased to 3, the hidden layer activity increases, causing oscillations in the inhibitory neurons.

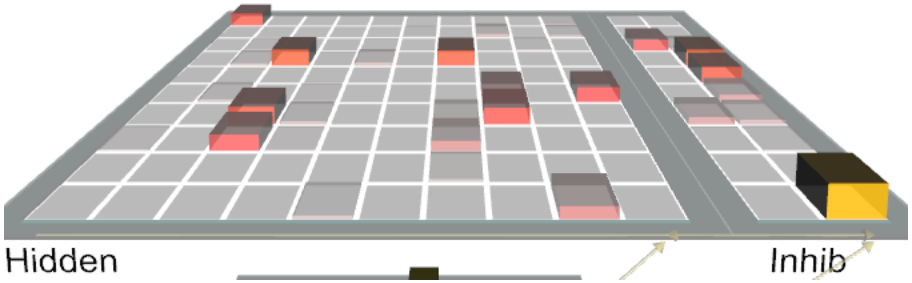
Solutions for the inhibitory network simulation



Exercise 1: If the maximal inhibitory conductance on excitatory neurons is increased to 5, all activity decreases.

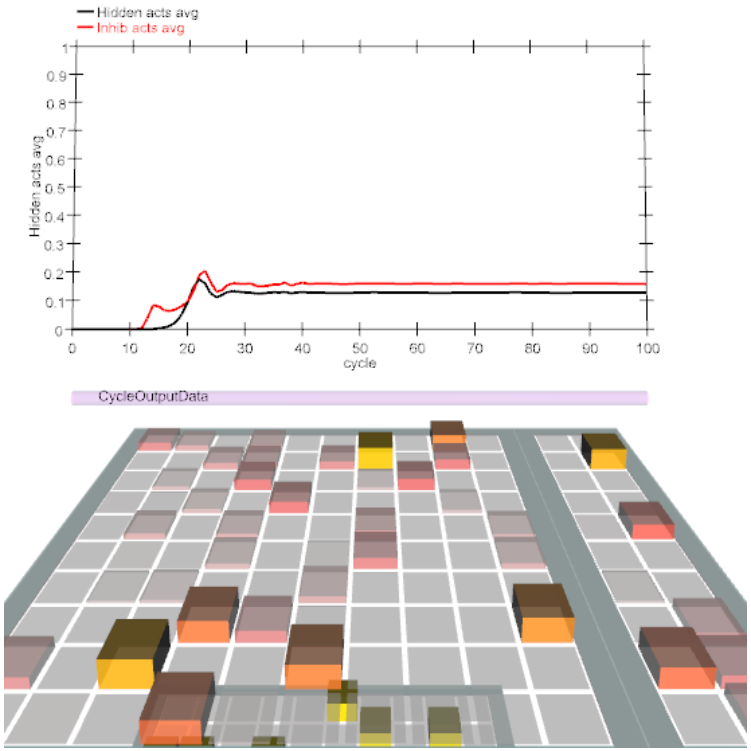
Solutions for the inhibitory network simulation





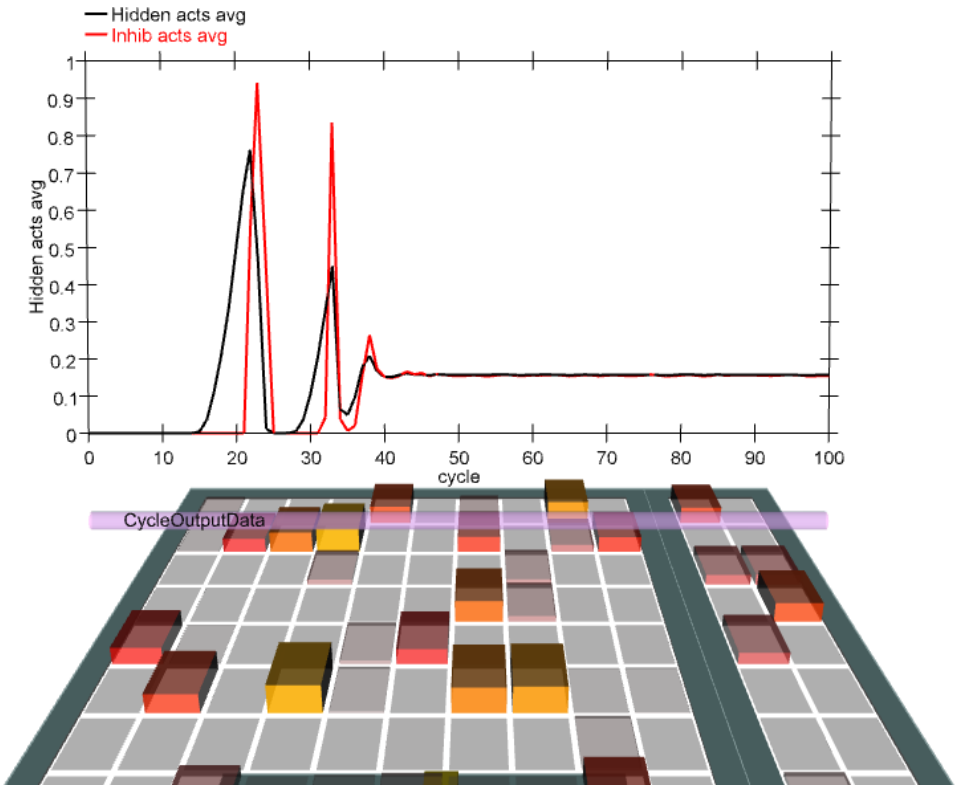
Exercise 1: *ff_inhib_g_bar.i* decreased to 3. The inhibitory activity stays the same, because decreasing the strength of inhibition on the inhibitory neurons causes their activity to rise, which in turn increases the effect of inhibition. The activity of excitatory neurons decreases, and the net input to inhibitory neurons is mostly unchanged.

Solutions for the inhibitory network simulation



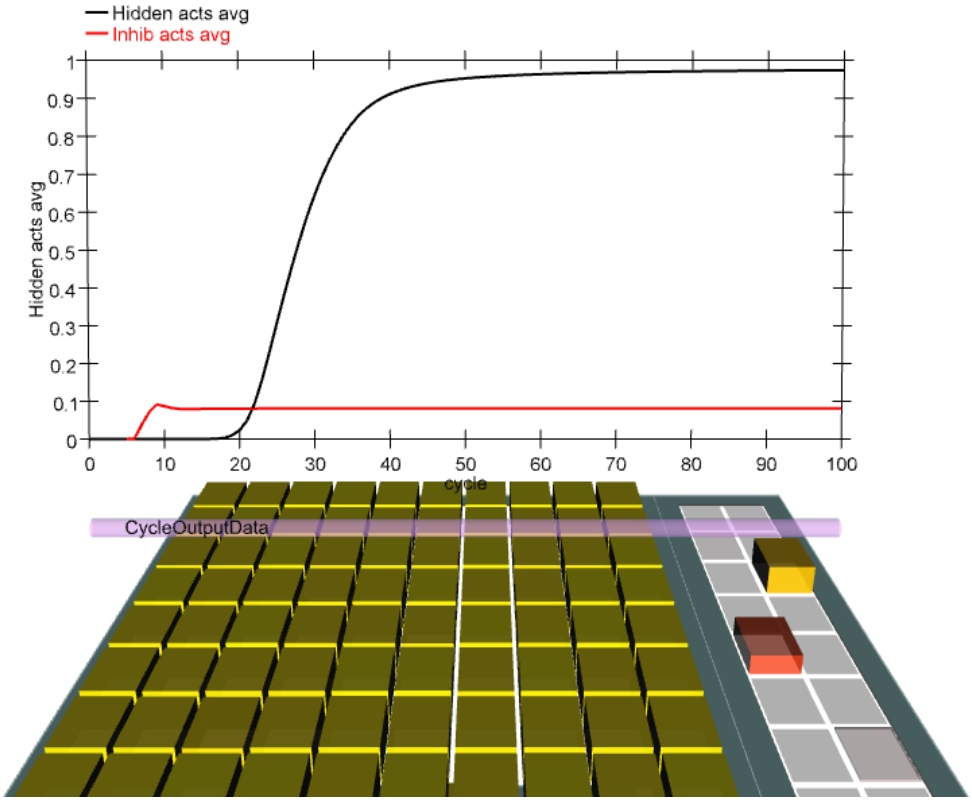
Exercise 1: *ff_inhib_g_bar.i* increased to 5. The inhibitory activity stays about the same, while excitatory activation levels increase. Actual activity levels are determined by the fixed points of recurrent dynamics. Mutual inhibition between inhibitory neurons changes the steepness of the net inhibitory feedback as activity levels vary.

Solutions for the inhibitory network simulation



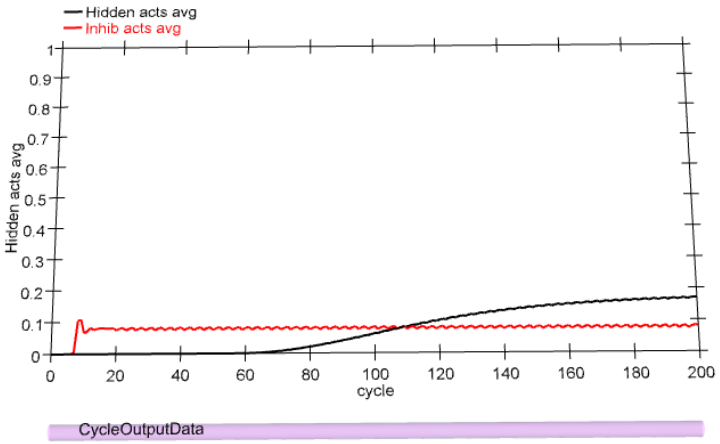
Exercise 2: when the feedforward weights from the input to the inhibitory neurons are set to 0, the network will stay in a steady-state after initial oscillations. The system is oscillating because inhibition follows the activity of the hidden layer.

Solutions for the inhibitory network simulation



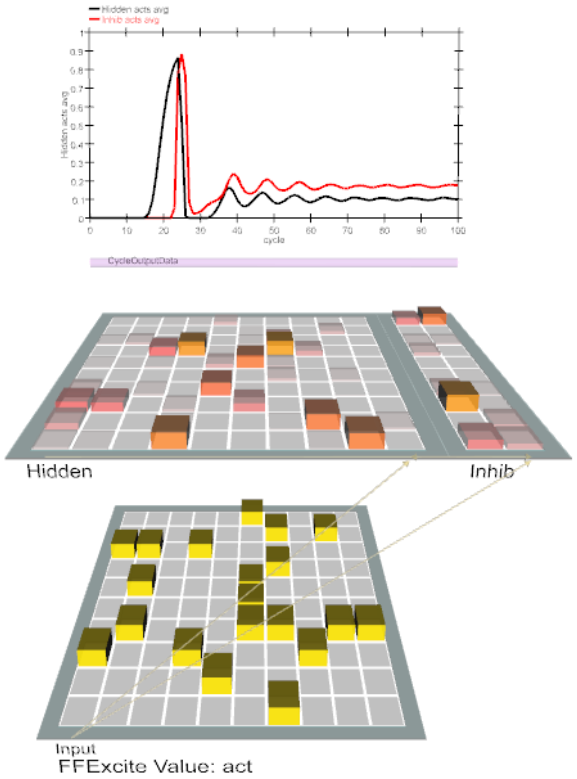
Exercise 2: when the feedback weights from the hidden layer to the inhibitory neurons (*ff_fb_wt_scale*) are set to 0, the hidden layer neurons will be in a maximal activation state after a while, because feedforward inhibition is not strong enough to control activity.

Solutions for the inhibitory network simulation



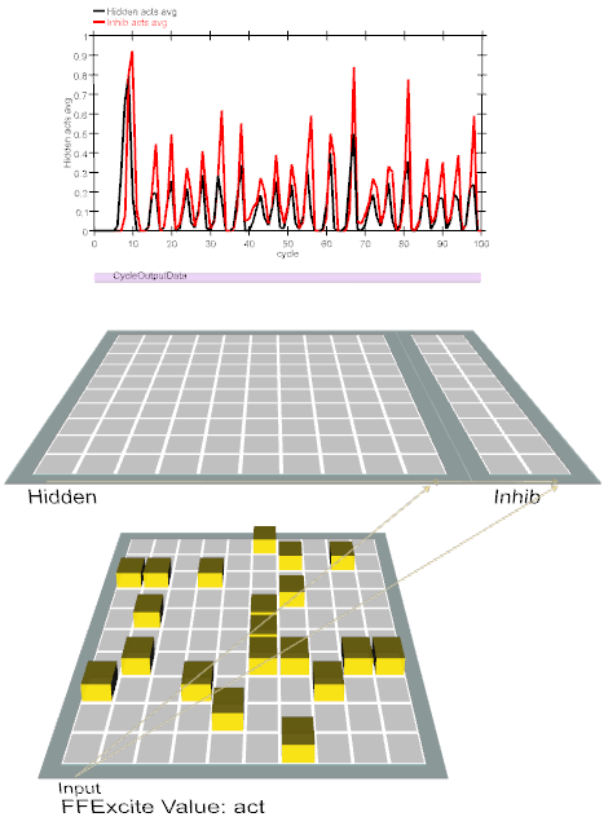
Exercise 2: The feedback weights from the hidden layer to the inhibitory neurons (*ff_fb_wt_scale*) are set to 0, and the excitatory conductance in the hidden layer (*ff_hidder_gbar.e*) is reduced to 5. Because of the decreased activity in the hidden layer the system can stay in a stable state.

Solutions for the inhibitory network simulation



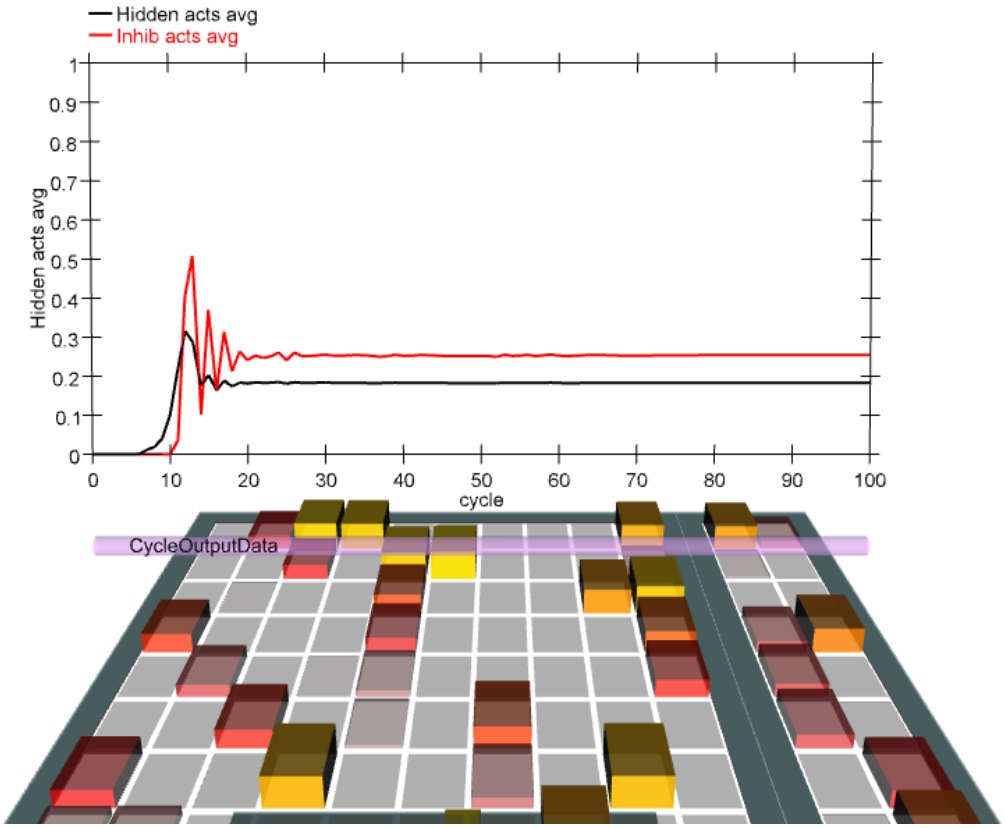
Exercise 3: when the excitatory neurons are updated faster, the system oscillates. With default parameters, the faster time constant enables the inhibitory layer to adapt faster to changes in the input.

Solutions for the inhibitory network simulation



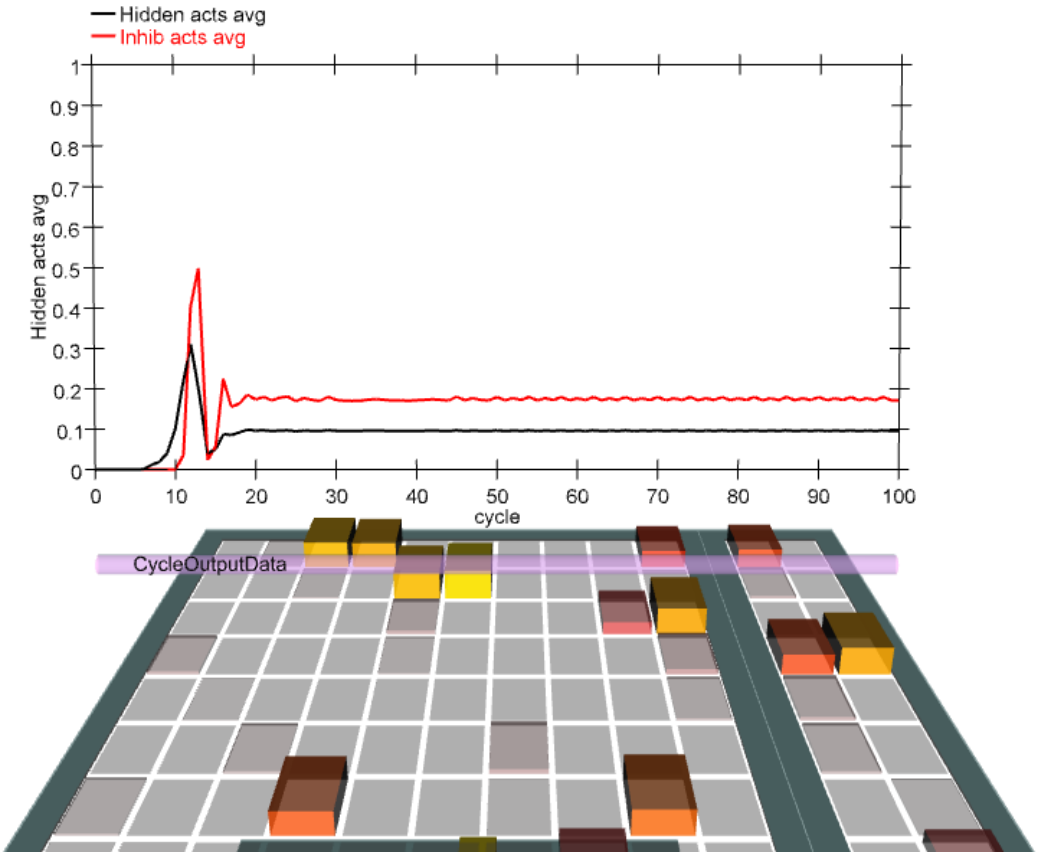
Exercise 3: If both time constants are fast, the system oscillates between states: The input causes hidden layer activity, which increases inhibitory activity after a short while. This inhibits the hidden layer, thus both hidden and inhibitory activity drop.

Solutions for the inhibitory network simulation



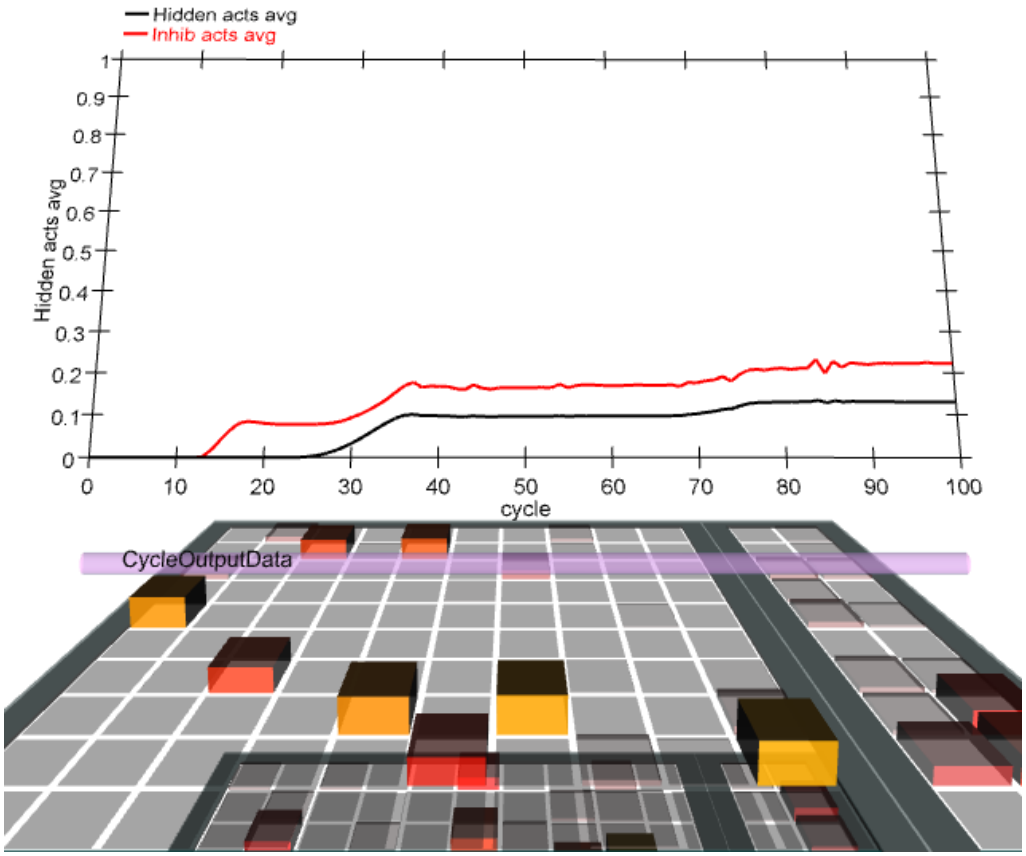
Exercise 4: With trained weights the average hidden layer activity is much higher.

Solutions for the inhibitory network simulation



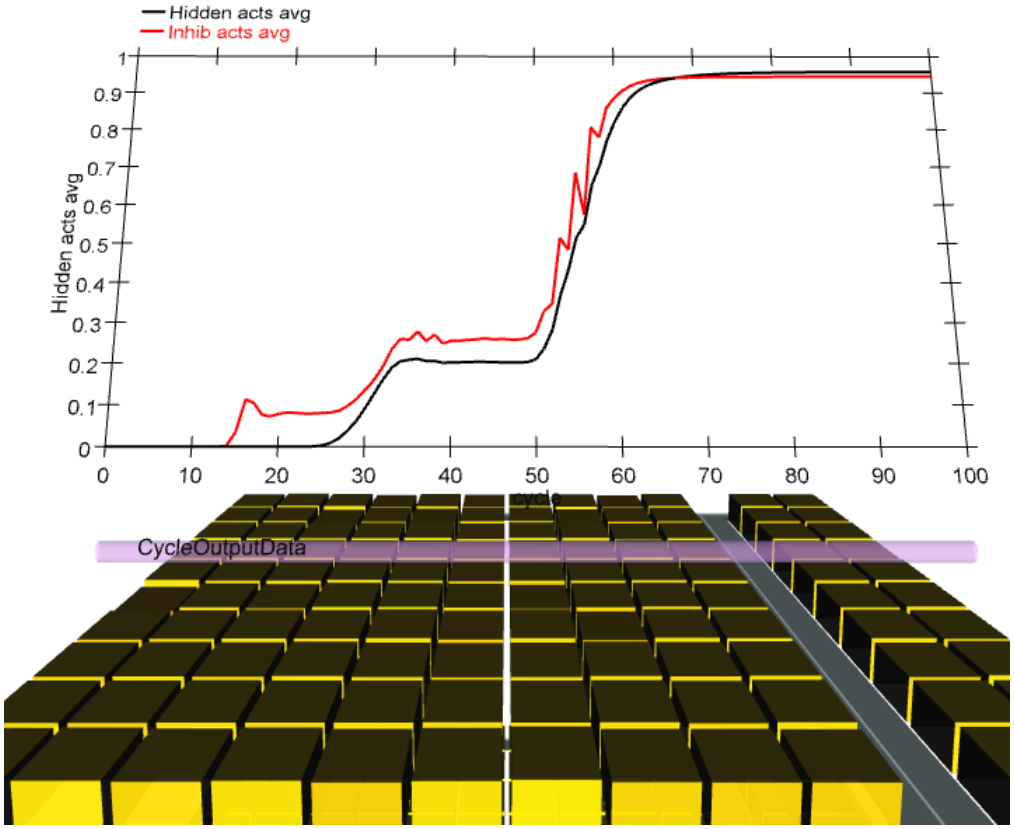
Exercise 4: If the inhibitory current into the excitatory neurons is higher (*ff hidden gbar i* is set to 8 from 5), inhibition counters the elevated activity levels.

Solutions for the inhibitory network simulation



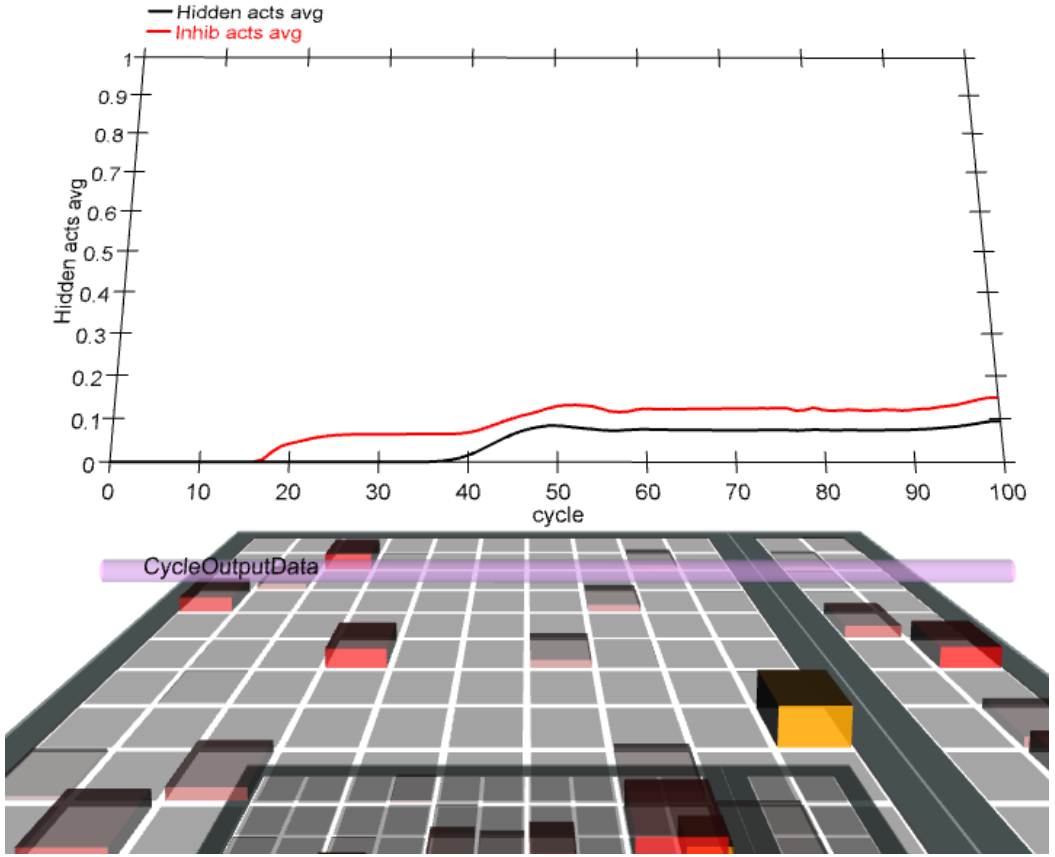
Exercise 5: Baseline activity of the BidirExcite simulation. The two steps in the network activity correspond to the activation of the two hidden layers.

Solutions for the inhibitory network simulation



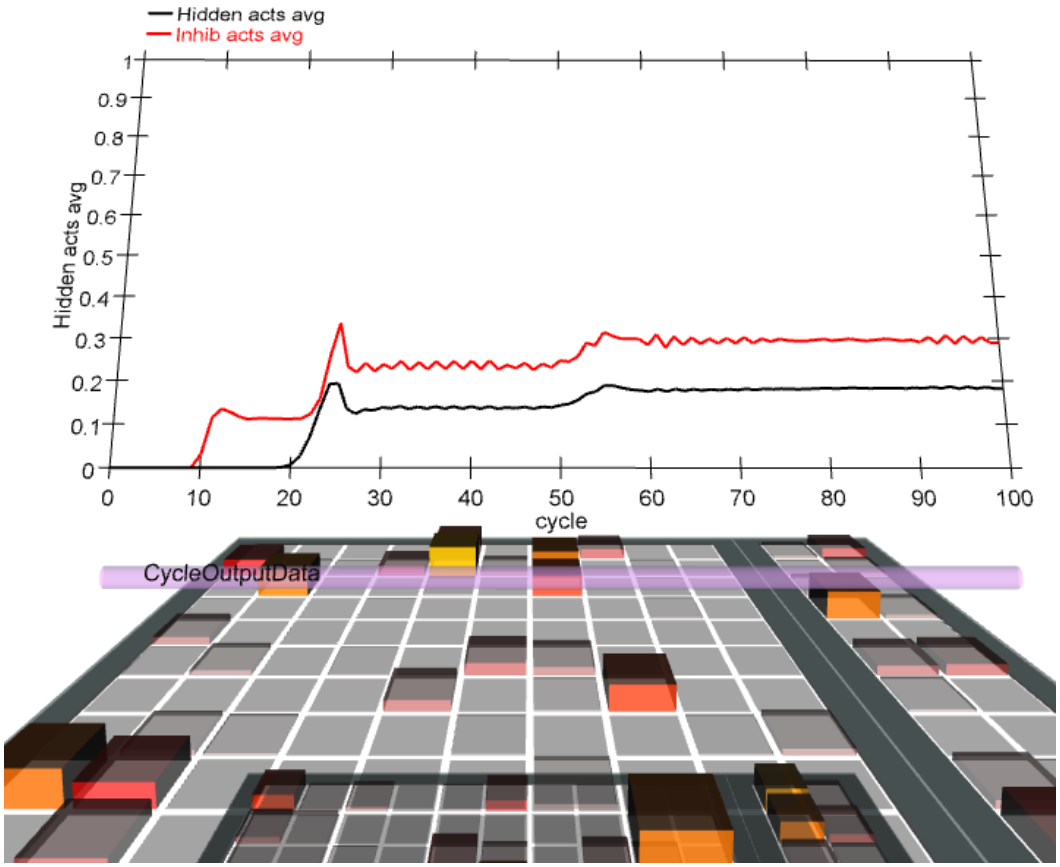
Exercise 5: When the inhibition on the excitatory neurons is reduced and the secondary layer becomes active the network becomes overactivated.

Solutions for the inhibitory network simulation



Exercise 6: With 15 active inputs the network activity slightly decreases

Solutions for the inhibitory network simulation



Exercise 6: With 30 active inputs network activity increases in a controlled manner, showing that the network is robust and functions under different levels of input.

Summary

- In this lesson we learned about:

- The basics of the Emergent simulation environment
- We used Emergent to analyze inhibition in cortical networks:
- The inhibitory-excitatory dynamics of the system can result in behaviors where the state of the network has multiple lengthy trajectories in the phase space.
- We observed that when the feedforward weights from the input to the inhibitory neurons are disabled the network exhibits initial oscillations before settling into a steady-state because the inhibition follows the activity of the hidden layer.
- When the feedback weights from the hidden layer to the inhibitory neurons are set to 0, the hidden layer neurons will be in a maximal activation state after a while, because inhibitory activity does not follow hidden layer activity properly.
- We added another layer of neurons to the simulation to analyze more complex network activity.