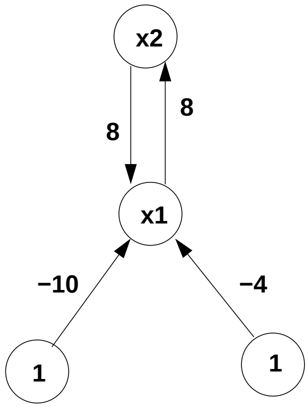
Exercises for "Vertiefung Neuronale Netze"
SS 2018 Sheet 12

Due on: 13.7.2018

Task 12.1, First steps: In these exercises we will explore the behaviour of small recurrent networks. In the experiments we will be using two closely cooperating building blocks: `graph_edit` and `net`. `graph_edit` can be used to generate and connect neurons as well as show their activity by clicking with your mouse. The simulation dynamics are seperatly implemented in the building block `net` (it also contains a call to a method in `graph_edit`, to show the recalculated activities).

Load the circuit **SmallNet** (in the directory `/vol/ni/share/lehre/neuroII/circuits/`) and get acustomed to the GUI: Create the following mini network consisting of 4 neurons:



Tips: With they keys Add-Neuon, Del-Neuron, Mov-Neuron, you can add, delete and move neurons. Added neurons wil be initialized with the activity value set by the *activity* slider. The keys Add-Connection and Del-Conntection work simililarly. The direction of a connection is defined at its creation by "Zugrichtung" (forward and backwards connections can be differentiated without arrows, as they are drawn shifted depending on their direction using the right-hand traffic rules.)

You can alter the activity and connection strength of a generated network later on by selecting a neuron/a connection (Key Set-Neuron/ Set-Connection). Note that the selected neuron/connection is highlighted in red/yellow. The sliders represent the activity/weighting and changes are instantaneously introduced into the network [1] (ATTENTION: this is also valid for adding new connections: the last added connection is selected automatically and is consequently affected be changing the slider!).

The key Rnd-Init initializes the activity of all neurons to a random value in the interval from 0 to 1. Zero sets the value close to zero (The Fermi function can not reach 0 itself). Pure inputneurons are not affected, as they count as external "sources", with constant values.

The key Clear **deletes the whole network**.

The keys Step/Run execute a single/iterated steps in the network dynamics. The sleep parameter allows to decrease the speed for easier observation. The last parameter β has the same meaning as in Task 1. This means the activities are determined again by the rule

$$x_r(t+1) = f(\beta \sum_{r'} w_{rr'} x_{r'}(t))$$

("synchron dynamics") .

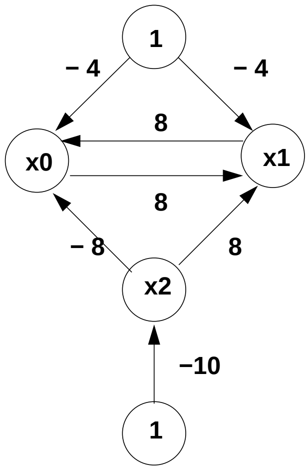
Task 12.2, Hysteresis: We want to analyze the network generated in the previous task (for $\beta = 1$) under a varying input of the bottom left neuron. You can change the input either by changing the activity of the input neuron or by changing the weight going to the first middle neuron. The following manual assumes the second method, as it is more comfortable to set negative as well as positive input values.

a) Set the activity of the inner neuron to (almost) zero using the Zero key. Now watch the behaviour of the middle neurons if the input is varying from -10 to +10 (The easiest way is to activate the key "Run"). For which input value are the middle neurons switched on/off? Explain the arising of the observed "hysteresis interval"! Can you find input values which make the network oscilating?

[1] as the slider can only take integer pixelvalues, you can only approximately set the following values. The experiments are designed robustly enough to not being harmed by this constraint.

b) Stop the network (Set "Run" to "aus" (off)) and configure the input value to a value from the previously found "hysteresis interval". Set the middle neuron to one and let the network run. What are you observing?

Task 12.3, Oscilator dynamics and hysteresis: Research the behaviour of the following bigger network (Again vary the input by changing the weight of the lowest neuron)

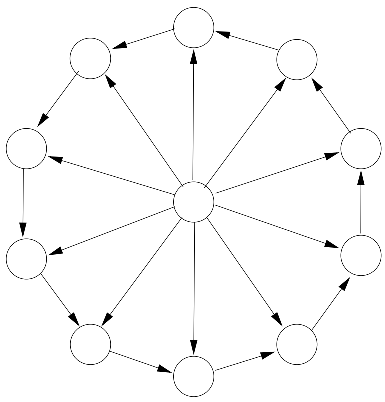


Task 12.4, Ringoszillator:

Create a ring structured network (with e.g. 10 neurons), which are all connected with weights of equal strength into the same direction. Starting from this circuitry we want to construct a oscillator network, with a circling activation pattern ("Ringoszillator").

a) Initialize all neurons to small values (key Zero) and set one neuron to the value 1. What do you expect will happen if you perform steps with these initial condition? Test your expectation and explain the result. How is the parameter β helpful for reaching the desired "wandering behaviour"?

b) Extend the network with an additional neuron, inhibiting all ring neurons with equally strong inhibitory connections (e.g. weight strength -10):



Select the new neuron and go into the "Run" mode. Have a look on how different inhibition (adjust it by changing the activity of the inhibitory neuron) affects the ringneurons (Tip: using RndInit you can try different initial values fast).

c) Now add additional "shortcuts" (inhibitory or excitatory) between ring neurons and analyze how these connections affect the generatable osciallation patterns.

d) You can perform other experiments by changing the update rule for the neuron activity implemented in the building block `net`. A simple experiment is e.g. the switching the synchronus dynamics used so far into an asynchronus dynamics, such that the activity changes in every neuron is directly written into the network (without buffering in the array `gpfAct0`). In other experiments you might want to use other activation functions (Caution, match also the one in the routine `inv_fermi()`) and the scaling of the screen view `graph_edit:prog:show`.