Computational neurodynamics

Exercise Sheet 2 (Unassessed)
Connecting populations and Braitenberg vehicles

All the files for these exercises can be found online at

https://github.com/pmediano/ComputationalNeurodynamics

Question 1.

a) Start up Python and run `Run2L`. The code that simulates the dynamics of the Izhikevich neurons is in `IzNetwork`. Inspect the code in these two files and make sure you understand how it works. In particular, note how the network is constructed in layers, and how the code connects layers together. (The use of layers here, to represent distinct neural populations, is for software engineering purposes only, and doesn't relate to the layered character of mammalian cortex.)

b) Create a new class `QIFNetwork` that works with quadratic integrate-and-fire neurons instead of Izhikevich neurons. Experiment with `Run2L` and notice the differences in behaviour between both neuron models. You can use `QIFNeuronDemo` as a starting point.

To construct a heterogeneous population, introduce some random variation in the v_r and v_c parameters, along the lines of the variation in c found in the original `Connect2L`.

Question 2.

a) Run `RobotRun4L` and observe a Braitenberg vehicle in action. Inspect the code and make sure you understand how it works. Experiment with the settings and explore how the neural network inside affects the behaviour of the robot (number of neurons, synaptic connections, neuron model parameters, etc).

b) Alter the network inside the robot so that it avoids objects rather than approaching them.

c) (for enthusiasts only) In which cases does the vehicle fail most often? Think about the cases where there is one object at each side of the robot. Using the ideas in Topic 6 (Competition) modify the neural net inside the vehicle to deal with these cases more effectively.