Acquired Intelligence & Adaptive Behaviour

Artificial Neural Networks
Lab Session 2

Goals:

1. To explore the different dynamical regimes and understand how network structure effects dynamics .
2. To understand how the parameters effects the relationship between input and ouput.
3. To understand the main differences between feed-forward and recurrent neural networks.

Background: Recurrent neural networks are exhibit rich dynamics. The dynamics of the network depend on the choice of parameters. They can exhibit all types of dynamics from fixed point dynamics to oscillations and chaos, see lecture notes. Run the matlab code provided and explore how the dynamics depend on the parameters of the network.

Tasks:

1. Build a one node CTRNN in Matlab. Set all biases (theta) to zero.
   - Plot time versus y.
   - Whats kind of dynamics does it exhibit?
   - Whats happens when the weights are small?
   - What happens when it is large?
   - Does it ever oscillate (tell me if it does)?
   - How does the final position of the network depend on where you start y (initial condition) and how does it depend on weight strength?

2. Build a two node CTRNN network again set all biases (theta) to zero.

   - Find weights that make the network oscillate.
   - Find weights that show a *stable equilibrium* (i.e. the dynamics settles to fixed point)
   - Plot a phase plane representation of the oscillation, i.e., plot y1 versus y2
   - Whats the difference between the dynamics when the weights are small and when they are large?

3. Build a N node CTRNN (about 100 will do) node network. Agai set all biases (theta) to zero. Randomly set weights matrix with **randn**
   - Plot time versus y1...yN.
   - Plot y1 versus y2 or maybe y52 versus y68 (you can try many combinations)
   - What happens when all the weights are small?
   - What happen when all the weights are large?
   - How do biases effect this?

5. Describe the main differences between these kinds of networks and feed-forward networks discussed in Lecture 2.

4. (Advanced) Use the network from question (3) and examine how the final endpoint of the network depends on the strength of the weights. Try starting from different starting points (initial conditions y(:,1)) that are very close together. Look at this for large weights (randn*1) and for small weights (randn*0.001). Look at definitions of chaos online.