Maxnet is a competitive network (a recurrent neural net with mutual inhibition) that is a means of identifying the strongest signal among a collection of neuron outputs. In the Maxnet design, the neurons are initialized to some starting state (presumably due to a transient stimulus at the inputs), and the Maxnet outputs all decay towards zero, due to the influence of lateral inhibition between neurons. If the lateral inhibition influence is too strong, all outputs will be suppressed to zero. For smaller values of inhibition, the network will converge to a stable solution in which one and only one neuron is firing. This neuron will correspond to the neuron with the largest initial condition.

The accompanying Matlab code consists of maxnet.m and activation_fnc.m. The number of neurons in the network is specified by "nnodes". The strength of lateral inhibition is specified by "eps" (which should be a negative number). The desired activation function should accept a net input value (a scalar) and return an output that is 0 if the input is negative but is identical to the input if the input is nonnegative.

Fill in the body of "activation_fnc.m" and complete populating the matrix W of weights. (Note: weight W(i,j) corresponds to the influence of neuron "j" on neuron "i").

Maxnet.m has a "pause" that requires you to hit "enter" to step through iterations of the network feedback. You must hit ctl-C to get out of the infinite loop (or set an alternative termination condition in the main loop).

Once your program is running correctly, experiment with: initial conditions, number of neurons, and inhibition strength. Report on your observations. Can you make any generalizations regarding suitable values for mutual inhibition?