Assignment 8: Prof. Cisek

Joon Hwan Hong; no collaborations to declare

Note: No MATLAB code submitted as it was just console and basic manipulation as tasked in the assignment. Individual figures & the end state at Q16 is available for view >>individual_figures folder.

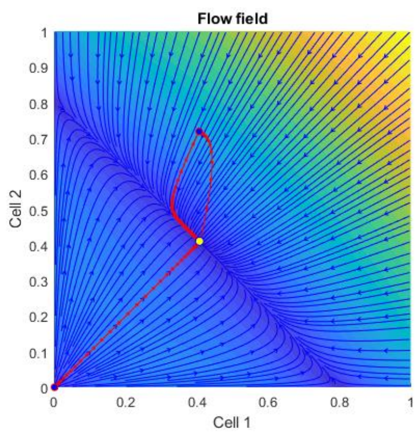
Exercise 1: Two independent cells

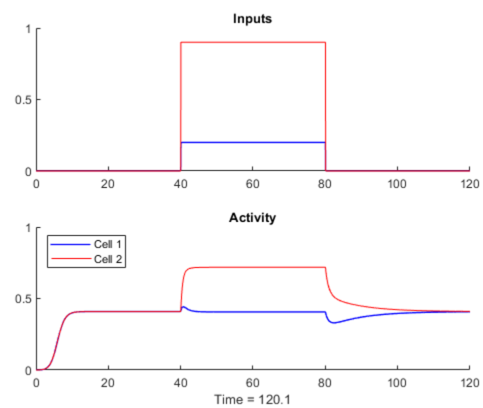
Q1: What is the value at which it equilibrates?

The blue line in figure 1 stops at an equilibrium value of 0.3333, or a 1/3 in the Y-axis.

Exercise 2: Two interacting cells

Q2: What caused this brief burst of activity?





Given the activity of each cell i equation (1) in the assignment, the change in activity of cell i is shaped by the excitation term and an inhibition term that is caused by an inhibitory edge from the other cell j vertex (if considering as a graph).

Thus, when both e1=.2 and e2=.9 were set, initially they both began to become excited (increase in activity). However, as both vertices have inhibitory edges acting upon the other, & cell 2 scaled at a higher magnitude, resulting in greater inhibition of cell 1, resulting in the “bump”/ brief burst of activity.

Q3: Note the brief dip in the activity of cell 1. What causes this?

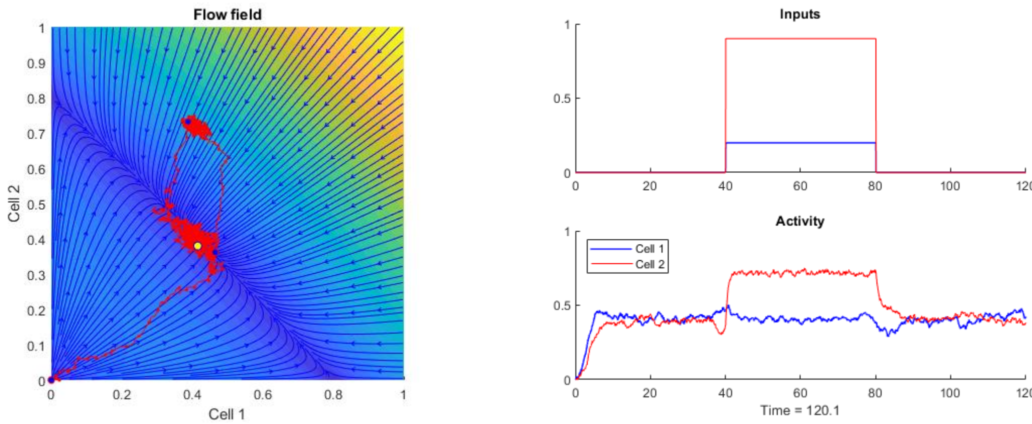
Again, referring to eq (1), a decrease occurs because the derivative definition results in a negative value when both e1 and e2 = 0, in other words, the loss of external input results in the inhibition term larger than excitatory term.

- just by estimating this is visible: (0.6)(**0**+0.4**0.9) – 0.4(0.2+0.7**0.9)

And eventually it recovers/converges to the expected value as the inhibition edge from cell 2 to cell 1 decreases in magnitude/value as cell 2 is decreasing as well, increasing once again and creating the brief dip in cell 1 activity.

Exercise 3: Noise in the system

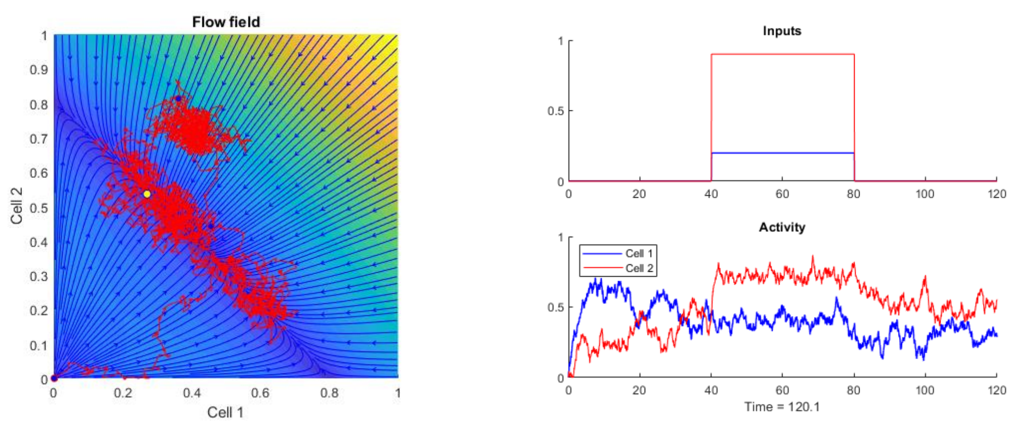
Q4: Describe what you see after each 40 seconds of simulation. Does the network still track the pattern of inputs, even with all of that noise? Or does it become completely chaotic and unrelated to its inputs



Figures above were generated with noise = 0.1. Now, the dot appears to generally follow the motions from exercise 2, albeit with Brownian motion/noise included. So overall, yes, the network still tracks the pattern of inputs even with the set noise level.

The general pattern is preserved when comparing to the no-noise system at all 40 second intervals. The activities rise when inputs increase, and return to basal levels when inputs are set back to 0s.

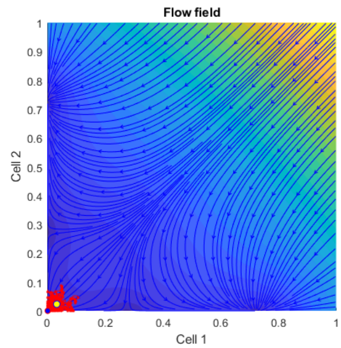
Q5: Describe the behaviour of the system



With a scalar noise value of 0.3, the behaviour of the system is with increased variance. With such high level of noise, it is difficult to determine in the input/activity plots where a “baseline” activity level would even be, and if the network recovered to it at all. Although it only follows a history of “favouring cell 2 then decreasing back to an extent” to a minute amount, it would not be appropriate to call this completely unrelated to inputs.

Exercise 4: Winner-take-all behaviour

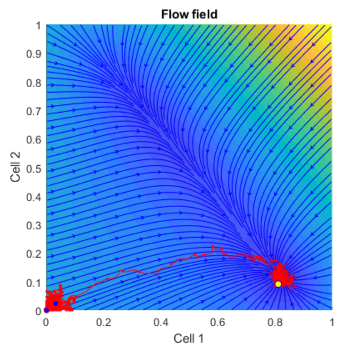
Q6: Describe what happens. What is the state of the system?



The system is unable to choose either cell as preference.

As the flow field indicates, until stimulus large enough to bring the yellow dot over ~0.3 of either cell for the vectors to point outwards away from the origin. Until then, all vectors point inward to the origin.

Q7: Describe what happened. What is the new state of the system? And now, turn off the inputs. Can you guess what will happen?

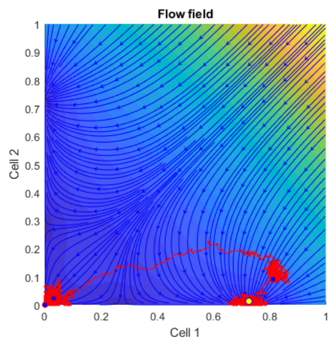


The system greatly favours cell 1 (chooses 1) with the given inputs of e1=.3 and e2=.1

Looking at the flow vectors, it appears that regardless of what position the state starts, it will eventually converge to the general pooled region.

Given the flow field when e1 and e2 are 0s in Q6 figure, I will hypothesize that the system will choose cell 1.

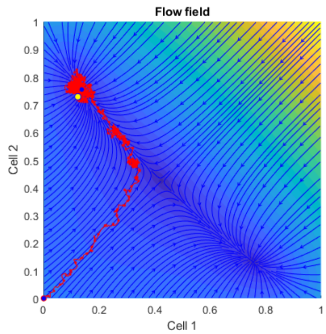
Q8: Describe what happened. What is the new state of the system? Is there a psychological phenomenon that this *reminds* you of?



The system chooses cell 1 as predicted. At the e1=0 e2=0 state, the vectors are set in a way that beyond ~0.3 for either cell, the system will favour one of the cells.

Once there is sufficient evidence in general (for either cell > ~0,3), if one cell is more favoured, the flow vectors will direct the state to choose either in a winner-takes all fashion once inputs are no longer present, making a decision after an input stimulus has ended.

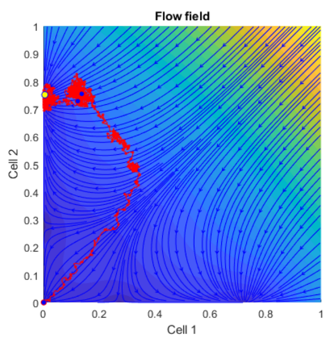
Q9: Describe what happened. What is the final state of the system, and why did this happen?



In this instance, the system choose cell 2. However, given that both inputs are the same. The decision-making process is essentially determined by noise; a stochastic decision.

The flow vectors will begin to favour which cell the noise addition happened to aid. Once enough noise contributed towards favouring one choice over the other, unless a strong stimulus is applied to favour the opposing decision, the flow vectors will reattract to cell 2's favour.

Q10: Turn off the inputs again. What happened?

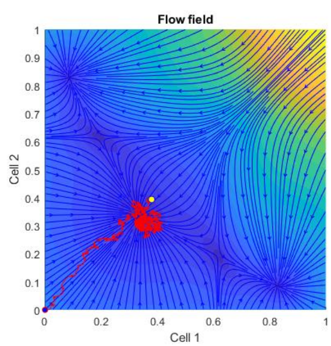


The system favours cell 2. This was expected.

The naïve state of e1=0 e2=0 was already known from previous questions Q6 & Q8 that once sufficient evidence (resulting in a state greater than ~0.3 for either) will lead to the decision of choosing that state.

Exercise 5: A stable attractor system

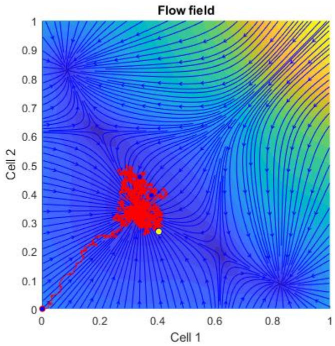
Q11: What did the network do? Did it make a decision?



The system is unable to make a decision. Three attractor states are formed (one each for favouring either cell, and the middle attractor state where the system is undecided).

Looking at the flow vectors, it appears that without a 'strong' stimulus, the system will be unable to escape the 'undecided attractor state'.

Q12: What happened? Did the network remember anything about its previous input pattern?

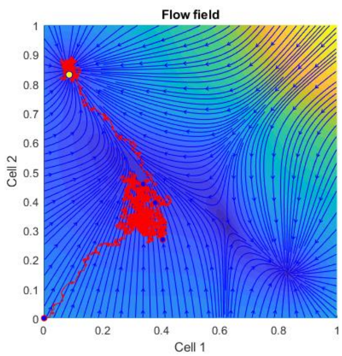


The system is still unable to escape the undecided attractor state, despite the small input increase initially for e2 to slightly favour cell 2.

Once the input stimulus is equalized to 0.1, the system remains in the undecided attractor state as predicted.

The system is unable to remember its previous input pattern because it was too weak; the system in the end just re-converged to the undecided attractor state.

Q13: What happens now? Can you explain it in terms of the flow field shown in figure 3?

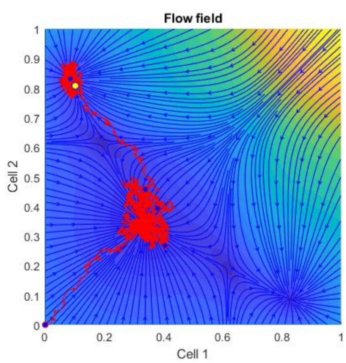


Given the e2=.2 input, the center undecided attractor no longer exists.

Given that the input e2 doubles e1, this resulted in a flow field where the central regions of low cell activity for both (which previously the undecided attractor state resided) now favouring towards cell 2.

Thus, the system follows the flow field to favour cell 2.

Q14: What happened? Did the network remember its previous decision?



The system remained as the equalized e1=e2=0.1 network has 3 attractor states, one of which favours cell 2, seen in Q11. And unlike Q12, the starting position of the system (yellow dot) was far way from the undecided attractor state, and was able to instead remain in a favouring cell 2 state.

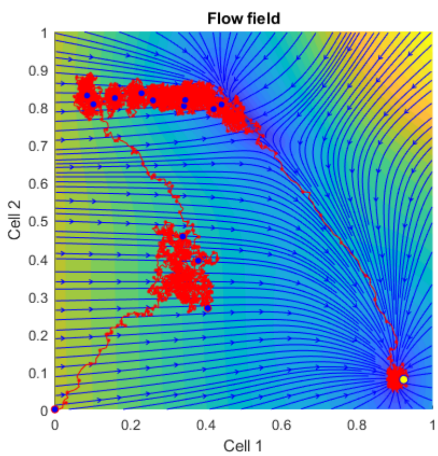
In other words, the network remembered its previous decision.

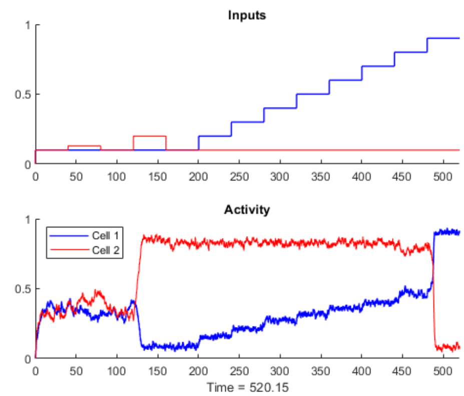
Q15: At what point did the system "change its mind" to option 1? Why was it so resistant to do that, even when the input to cell 1 was much stronger than the original signal that drove it to choose option 2?

The system was modified with the following logic until it favoured choice 1:

```
while(not e1)
      e1 ← e1 + 0.1
      run_network
```

The system changed its mind to option 1 at e1 = 0.9. The following is the result:





Only at such high value did it suddenly changed its preference, seen in the activity graph over time. It was stubborn due to the starting position of the network already heavily favouring cell 2.

Why this is the case is evident when looking at equation (1) in the assignment. Although the input of cell 1 was incrementally getting way larger than cell 2:

- For cell 2, the xi, and thus $f(x_i)$ sigmoidal function gave an incredibly large inhibition term when it started at already high values, deterring any large change in the system's position.
- While the cell 1's activity began at a low xi, resulting in an incredibly small $f(x_i)$ following the sigmoidal curve, resulting in ineffective excitation of itself & inhibition of cell 2, until e1=0.9, where it now began dominating

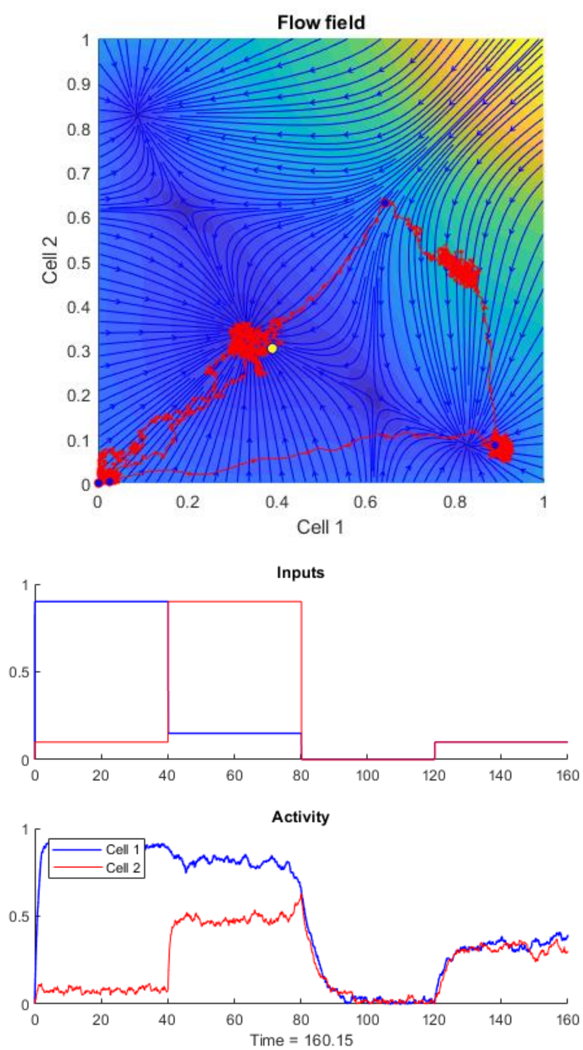
Q16: Can you think of a way to get the system back into an “undecided” state, simply by changing the inputs e1 and e2? Hint: You might need to change them more than once.

Essentially, the idea was to make it as close & ambiguous of a decision as possible (so the system was as “linear” in both cell 1 and cell 2 values as possible) then floor the entire system by setting both inputs e1=e2=0, then recover to the undecided state as the origin converging to the undecided state is already known from Q11 flow field.

Due to numerous testing, the system started with the following to recover to the Q15 state, hence a red line going from the origin to the Q15 state:

```
start_network
e1=.9;
e2=.1;
run_network
```

Then, the following was accomplished as planned, although it is important to note that a RNG seed was not selected, and thus the reproducibility is uncertain & based on RNG:



The following figures were generated by then running the following on the console. Again, no RNG seed selected to results will most likely vary, and given the extreme value changes I have done for the first step, very high chance the first value changes result in just flipping of attractor states.

```
e1=0.15;
e2=0.9;
run_network
% now it was on the (~0.6, ~0.6) region

e1=0.0;
e2=0.0;
run_network
% now it was on the origin region

e1=0.1;
e2=0.1;
run_network
% now it recovered to the undecided state.
```