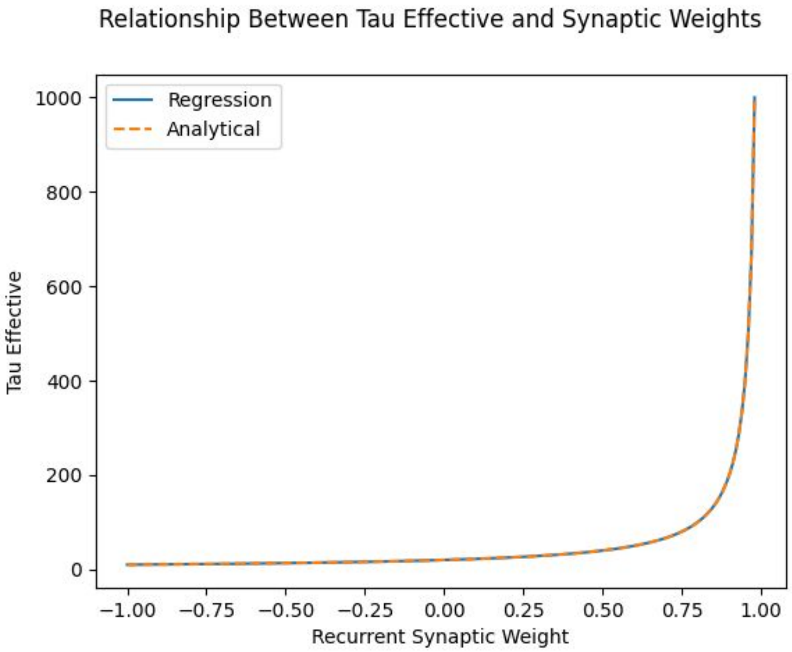
Goldman Lab Work

Alexandru Tapus

Week: 3/14/24-3/21/24

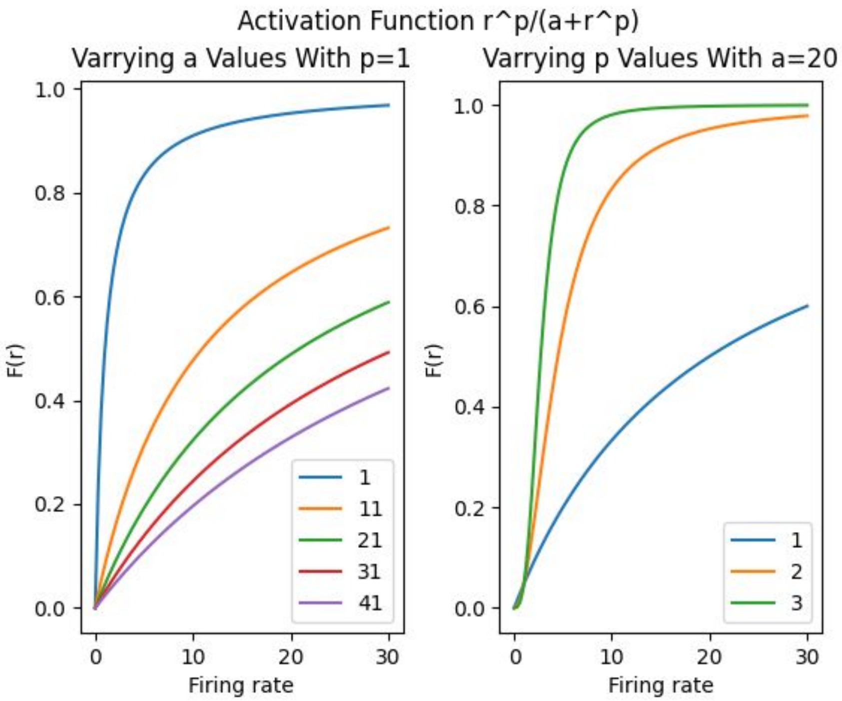
- Simulated a nonlinear autapse neuron
  - Graphed equation r^p/(a+r^p) for varying values of a and p
  - Found the fixed points of a neuron with this nonlinear firing rate
  - Graphed the firing rate over time for a rate model using a form of the graphed nonlinearity
- Predicted the effective time constant given the firing rate over time
  - Fed a linear rate model simulation into a linear regression model
    - X: Delta t from a time t0
    - Y: ln(r) at time t0 + deltaT
  - Repeated the aforementioned procedure for synaptic weights from -1 to .99

Effective Time Constant Over Synaptic Weights (Linear)



$$\tau_{eff} \cong \tau/(1-w)$$

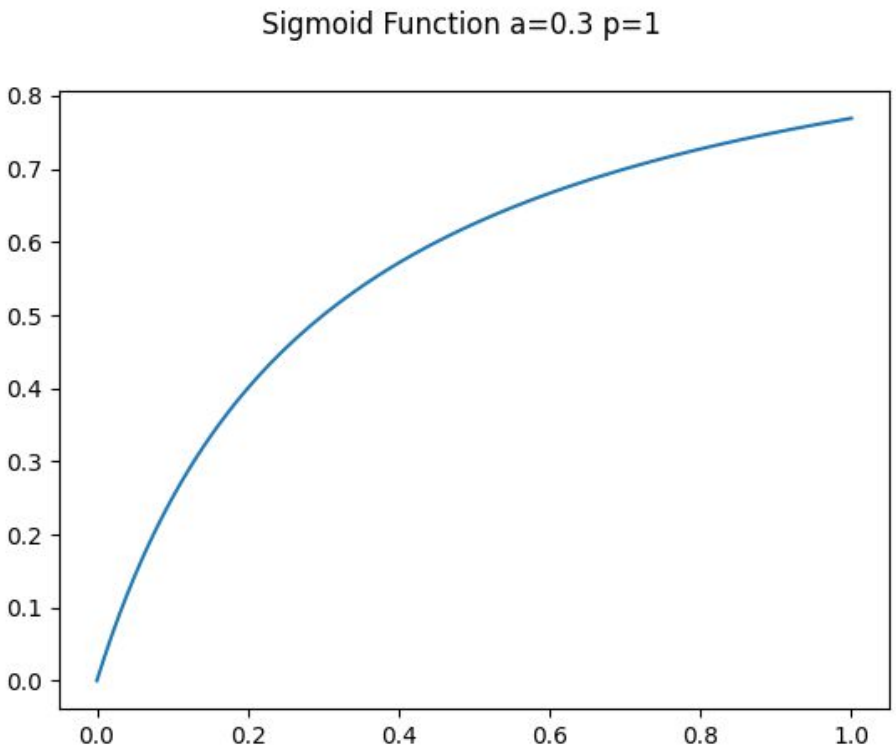
Graphs of Nonlinear Functions



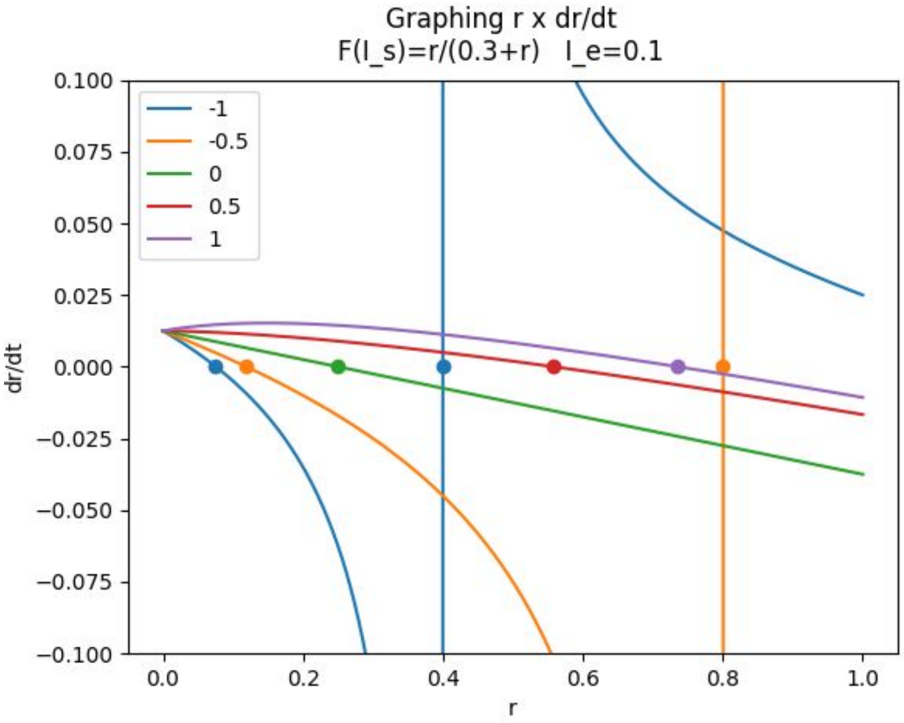
Trends in the functions

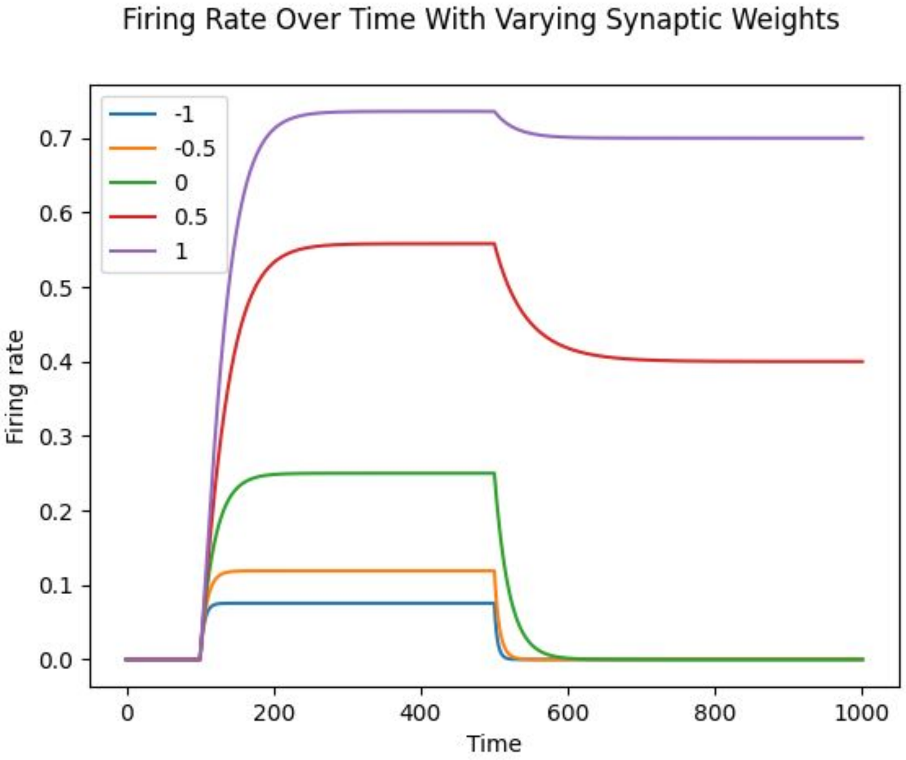
- Increasing a decreases slope
- Increasing p increases slope

Nonlinear Function Used



Fixed Points and Firing Rate Over Time (Incorrect)





Notes

Notes

- Used external input of arbitrary value of .1 spikes/sec
- Any large r will quickly overrun a=.3 and reach one
    - However, a larger a value would not change that $\lim_{r \to inf}(r/(a+r))$ always = 1, it would just approach 1 way slower

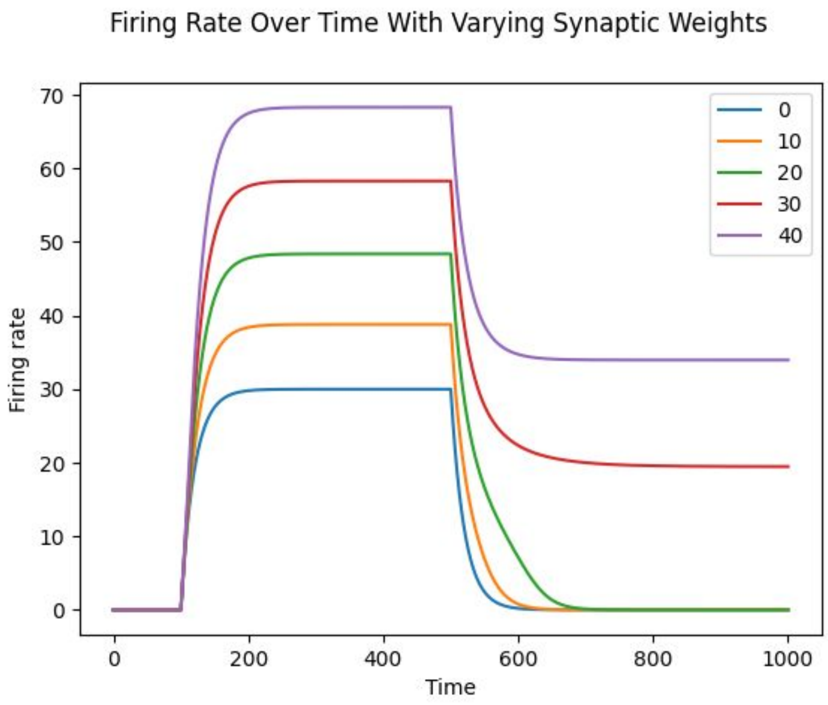
Week of 3/21/24-3/28/24

Finding Fixed Points of Nonlinear Autapse Neurons

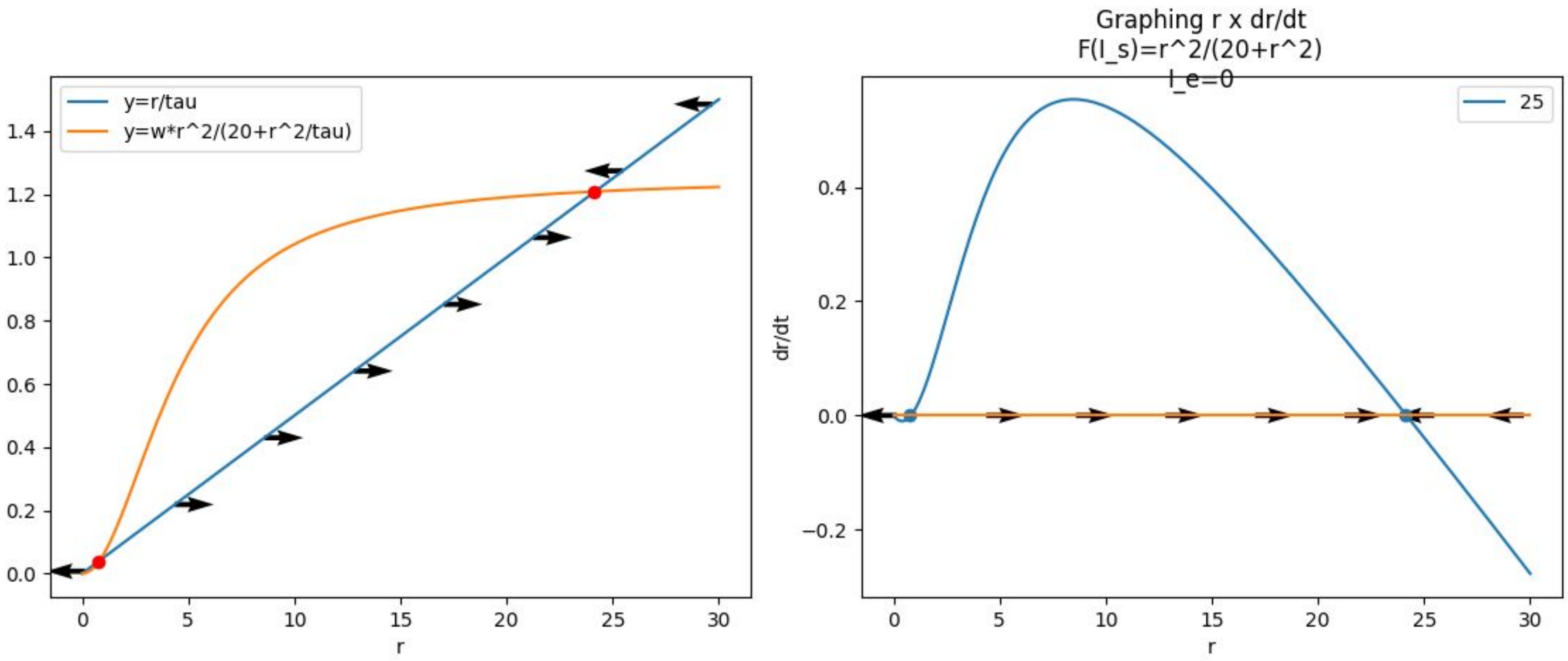
- Found the fixed points of the nonlinear autapse neurons
- Categorized them as stable or unstable
- Drew two representations for representing where the fixed points occur
  - Graph of drdt vs r
  - Graph of decay and input vs r (looking at where they intersect)

Redrew the correct firing rate curves for a nonlinear autapse neuron

Correct Nonlinear Autapse Neuron Firing Rate



Fixed Points With w=25 and F(r) = r^2/(20 + r^2)



Unstable Point: .800801

Stable Point: 24.1241

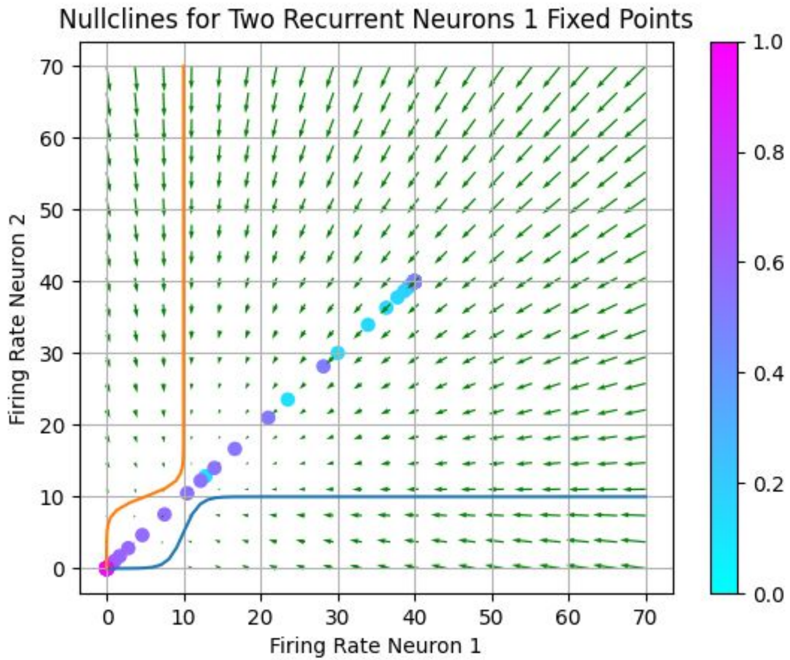
Week 4/8/24

Two Recurrently Connected Nonlinear Neurons

- Graphed the firing rates of the two neurons as a function of each other over time. (As well as on the actual r1 r2 graph to see the relationship between the nullclines and the change in firing rate)
- Drew the nullclines for several activation functions.
  - Attempted to have 1, 2, and 3 fixed points.
- Drew the vector field formed by the nullclines showing how the fixed points are approached.

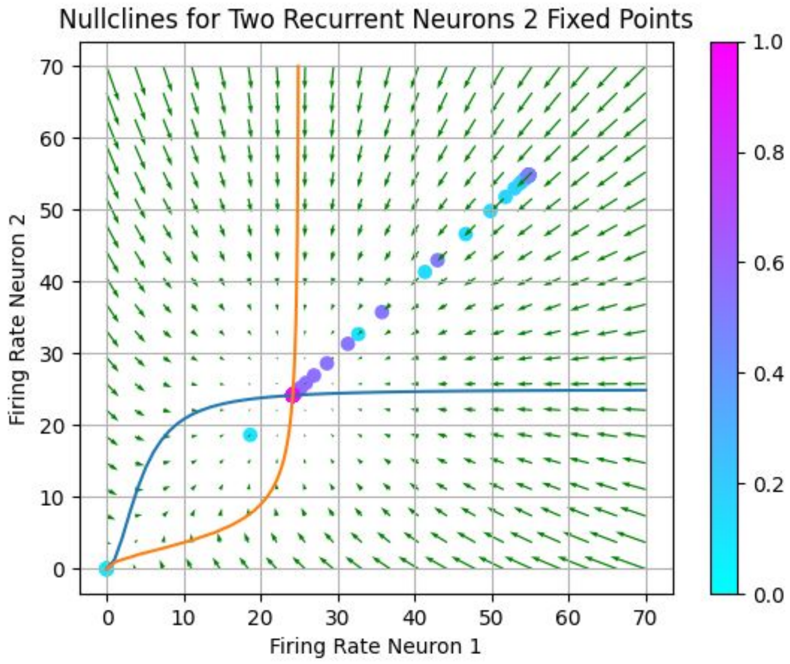
Two Recurrently Connected Neurons (Nonlinear)

1 Fixed Point



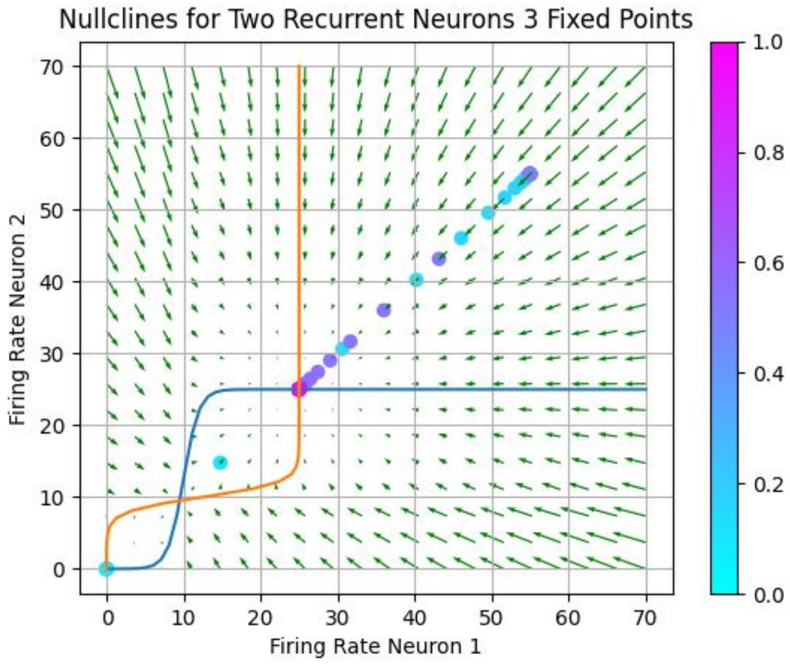
Two Recurrently Connected Neurons (Nonlinear)

2 Fixed Points



Two Recurrently Connected Neurons (Nonlinear)

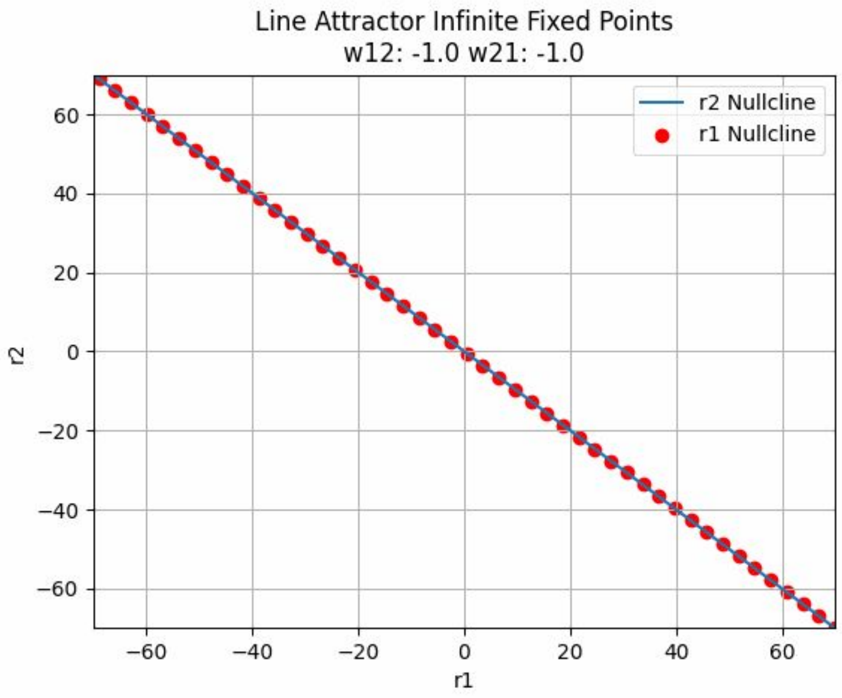
3 Fixed Points



Line Attractor Conditions

For two recurrently connected neurons.

- The two weight values have to be reciprocals of each other
- More line attractors can be made by allowing shifting with positive and negative external inputs.



To Do Next Week

Used the e sigmoid function on an autapse neuron.

Created an animation showing how the nullclines shift with the application of external inputs. (Next Week)

- Quick pulse current (on off)
- Box current
- Trapezoid shift for more gradual change

(Each nullcline is showing the conditions of no change for a given input) (Change the input and don’t always make it symmetrical)

Week 4/22 (Woods Hole Problem Set)

Calculating and proving the eigenvectors for a simple two neuron network

Calculating the amplification constant, effective time constant, and steady state values of the network

Running a simulation to validate the mathematical approach

Discover the weight values for which the neural system integrates the difference between the two neurons

Creating a winner-takes-all network

(If time, creating a model of perceptual rivalry by setting an r_{max} and slowly decreasing it until the other neuron takes over. R_{max} is reset when it gets close to 0 again)

Mathematical Analysis of the Network

Weight Matrix	Eigenvectors	Eigenvalues
$\begin{bmatrix} W_{self} & W_{other} \\ W_{other} & W_{self} \end{bmatrix}$	$[1,1]$	$W_{self} + W_{other}$
	$[1,-1]$	$W_{self} - W_{other}$

I_{common}:attenuated

$I_{different}$:amplified

For w_{self}=.2 w_{other}=-.7

- 1/(1–.5) = ⅔ 1/(1-.9) = 10

$\tau_{eff} = \tau/(1-\lambda)$

- 18/(⅔)= 27 18/10 = 1.8

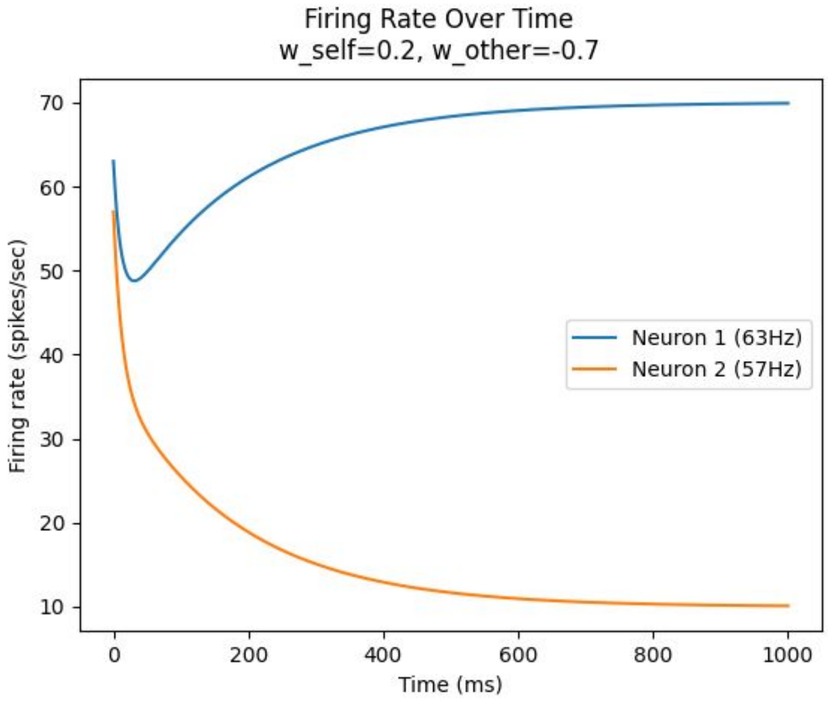
For current of (63Hz, 53Hz):

- $[b_1-b_2,b_1+b_2]$=[63,57] => b_1=60, b_2=-3

Calculating Steady State Firing Rate

- 60*⅔*[1,1] + -3*10*[1,-1] => [10,70]

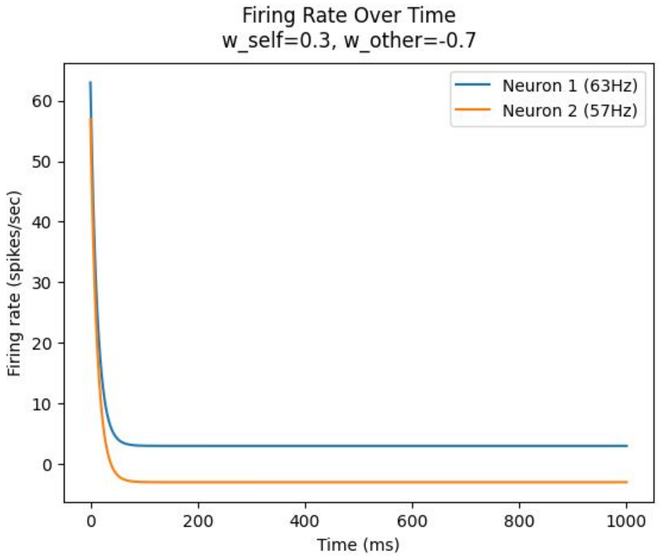
Simulation for Provided Weight Values



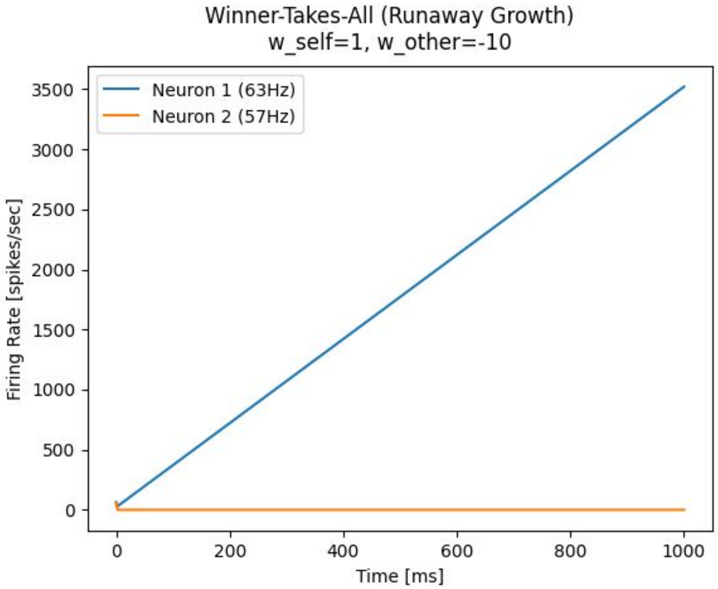
Integrating the Difference Between the Neurons

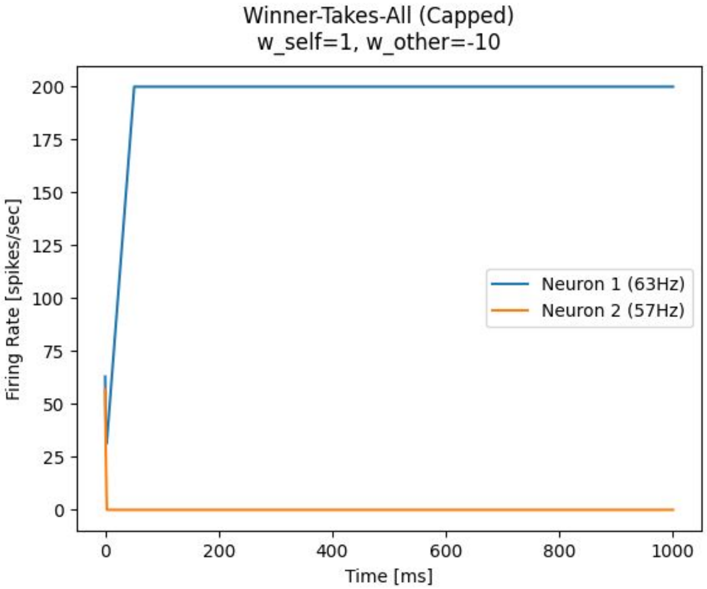
If the sum of the absolute values is 0, the difference is integrated

- (e.g. w_{self}=.3, w_{other}=-.7)
- No current added



Creating a Winner-Takes-All Algorithm





To Do Next Week

What is the condition that makes the winner neuron go to infinity? (Prove mathematically)

Make a 100 neuron network

Make an eigenvalue spectrum (Real and imaginary components of the network)

- Choose the weight matrix to be from a normal distribution N(M=?,SD=?)

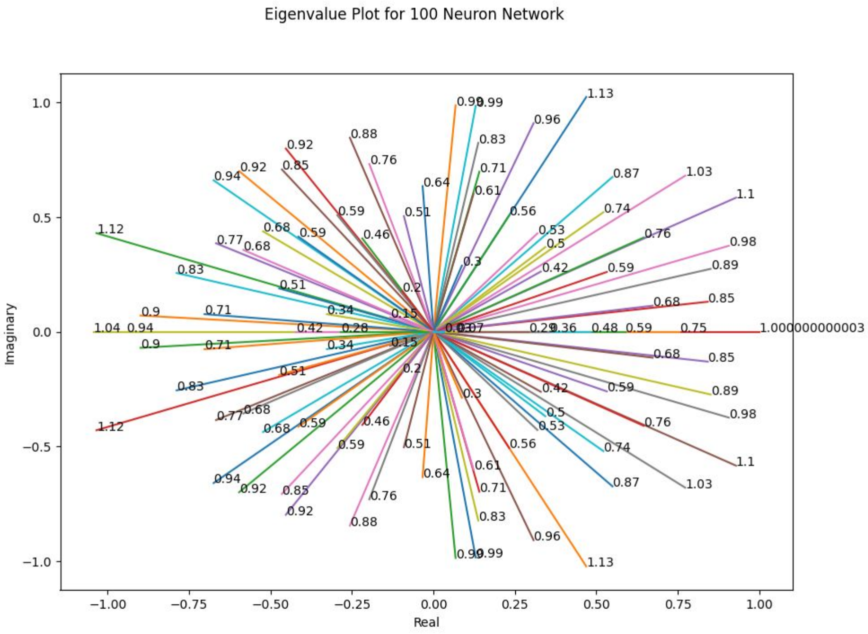
Be able to look at any two neurons relative to each other

Proof of Condition For Going To Infinity

The difference between the two neurons in a winner takes all network is accentuated by an eigenvalue of $w_{self} - w_{other}$.

- When w_{self} is small a small positive number and w_{other} is a large negative:
  - $w_{self} - w_{other} > 1$ and the model experiences exponential growth on that eigenvector.

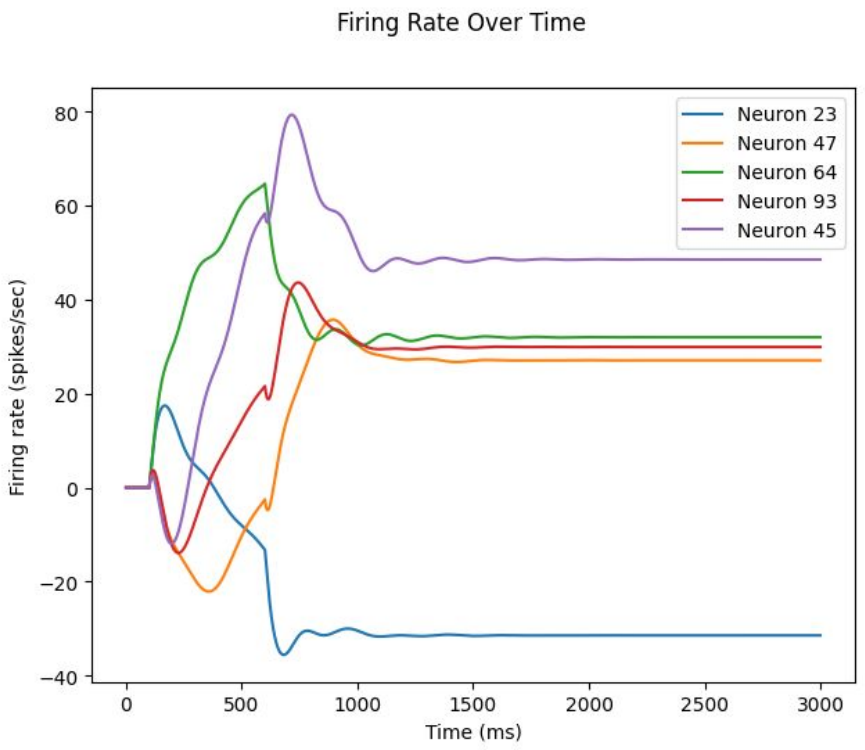
100 Neuron Network Eigenvalue Plot



Findings:

- Addition had no predictable effect on eigenvectors but increased the value of the real eigenvalues.
- Division by a constant divided all the eigenvalues by that same constant.
- Eigenvalue of 1 means that the network is an integrator.

Plot of Several Neurons in the Network Over Time



To Do

Look at the leading eigenvalue of the eigenvector.

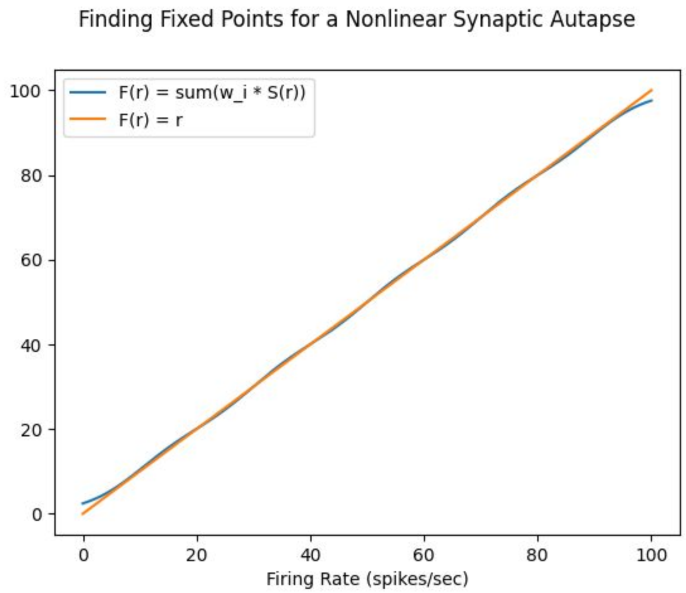
Add a pulse of input that corresponds to the eigenvector.

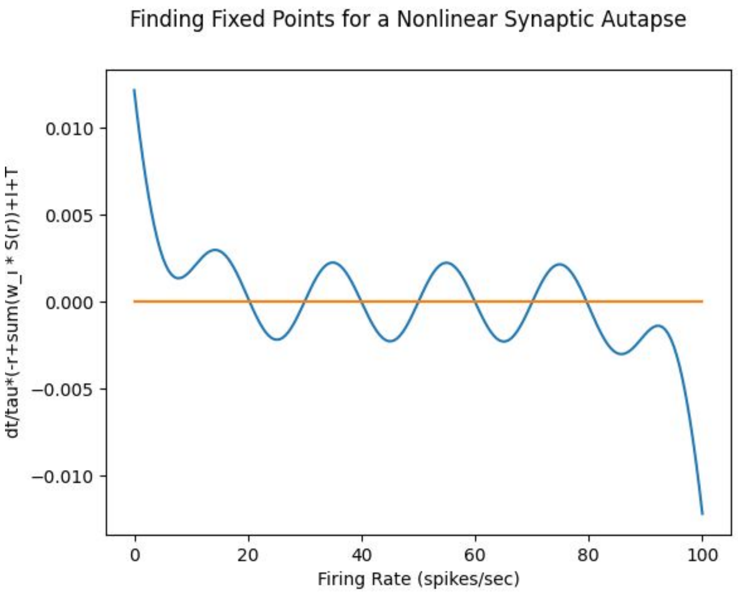
What does the leading eigenvalue represent (dict of eigenvalues and eigenvectors)

Create a matrix with eigenvalue 1+xi, 0+i

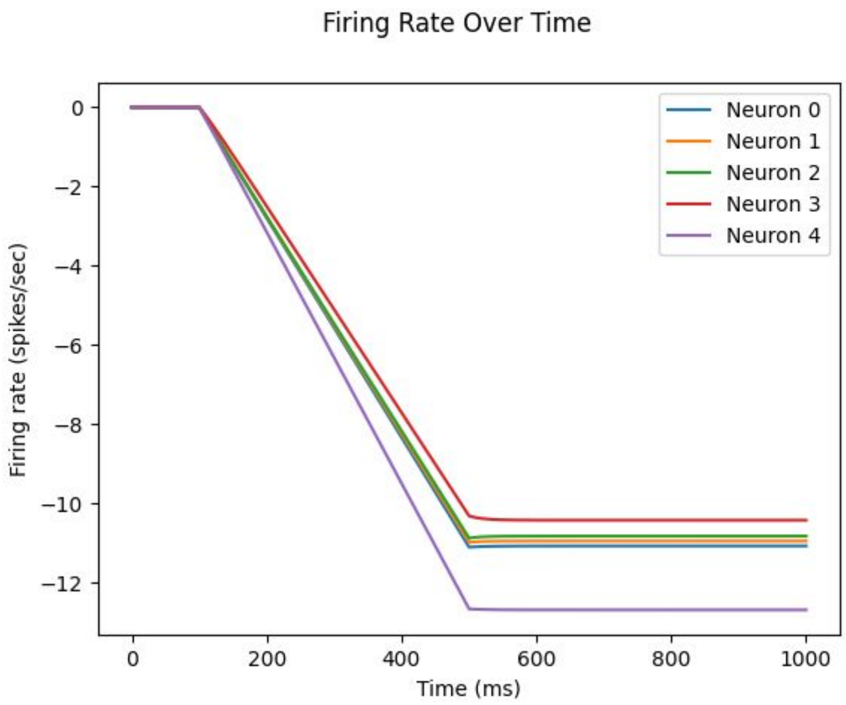
Creating a line attractor using a single synaptically connected neuron

Finding the Fixed Points of a Recurrent Synaptic Autapse





Applying Current Along a Real Eigenvector

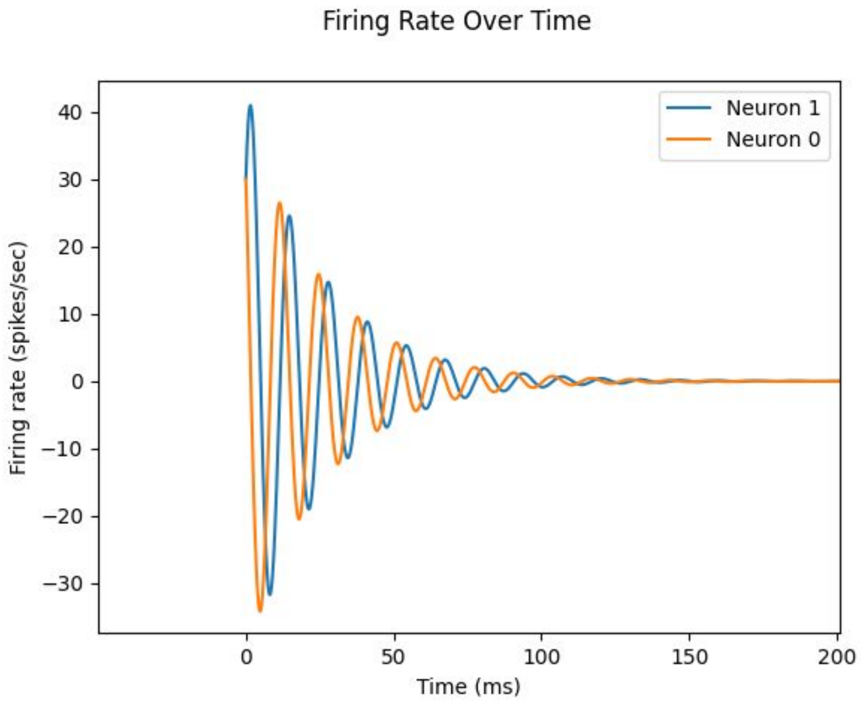


Week 5/13

Proved Euler’s Formula Using Taylor Series

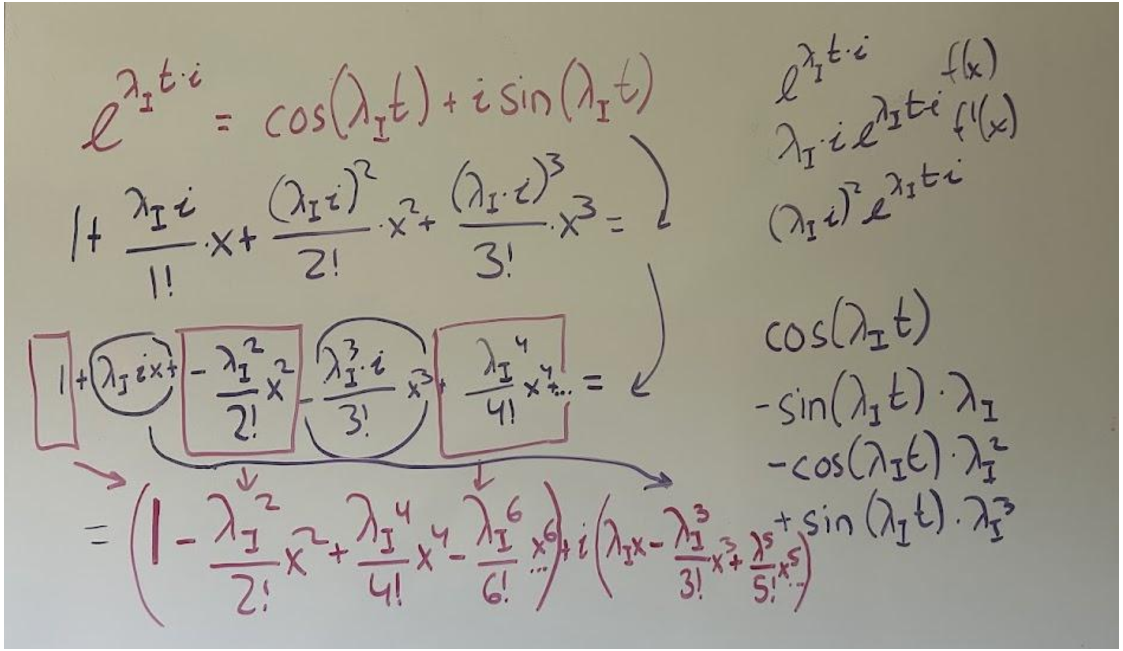
Observed complex eigenvector oscillations in a linear recurrent model (2 neurons)

Oscillations



Weight matrix = [[0,10][-9,0]]

Taylor Series and Imaginary Numbers



To Do

Nonlinear Recruitment Network

Github Project Files

https://github.com/AlexHackathon/GoldmanTapusWork