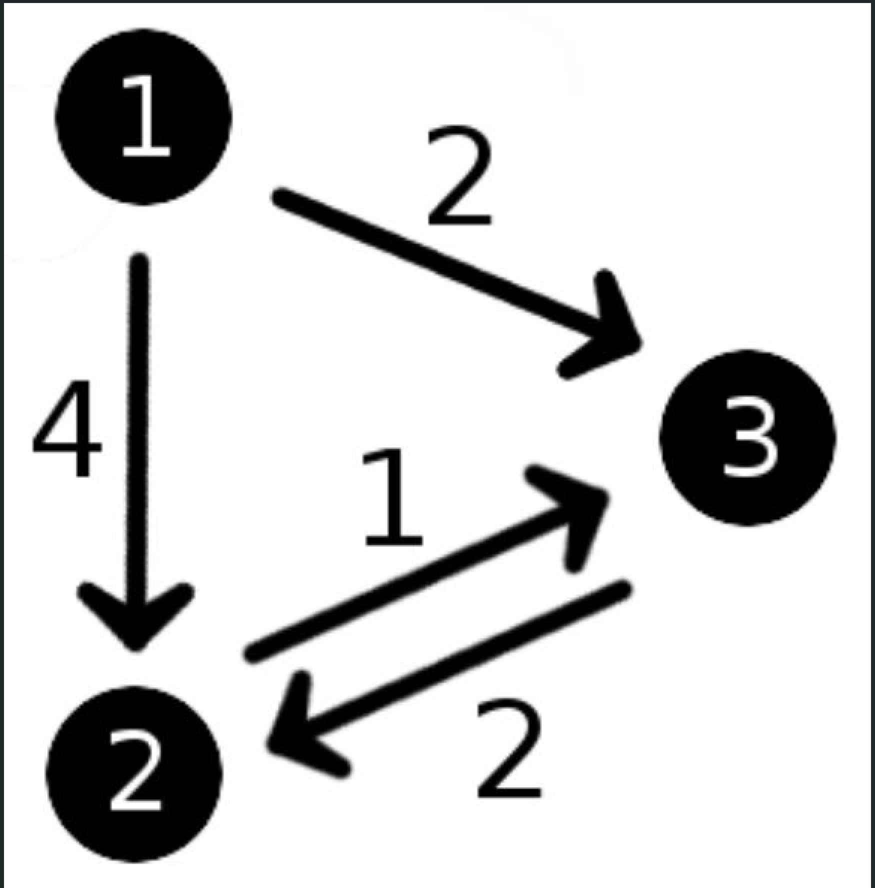
Spatiotemporal Patterns of Neural Activity

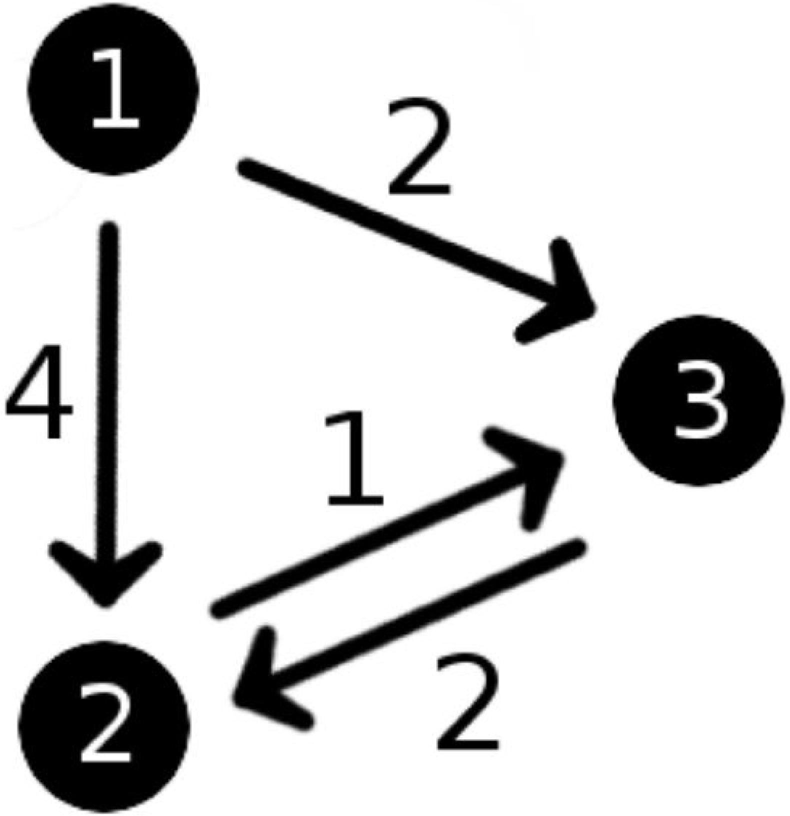
Also Known As - Synfire Braids, Synfire Chains, Polychrony

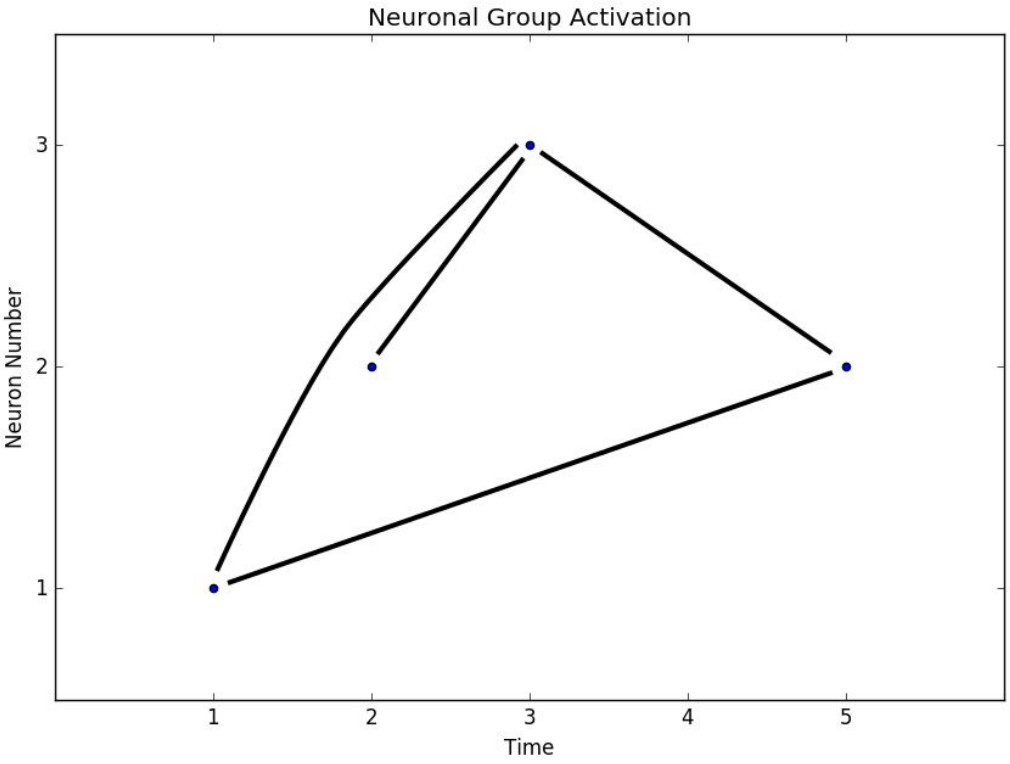
Polychrony - Concepts

Take a recursive network with particular activation rules

- If a subset of neurons are activated at particular times, the resulting cascade of neural activity is deterministic
- This activation sequence is what we call the spatiotemporal pattern of neural activity, or the polychronous group - polygroup for short



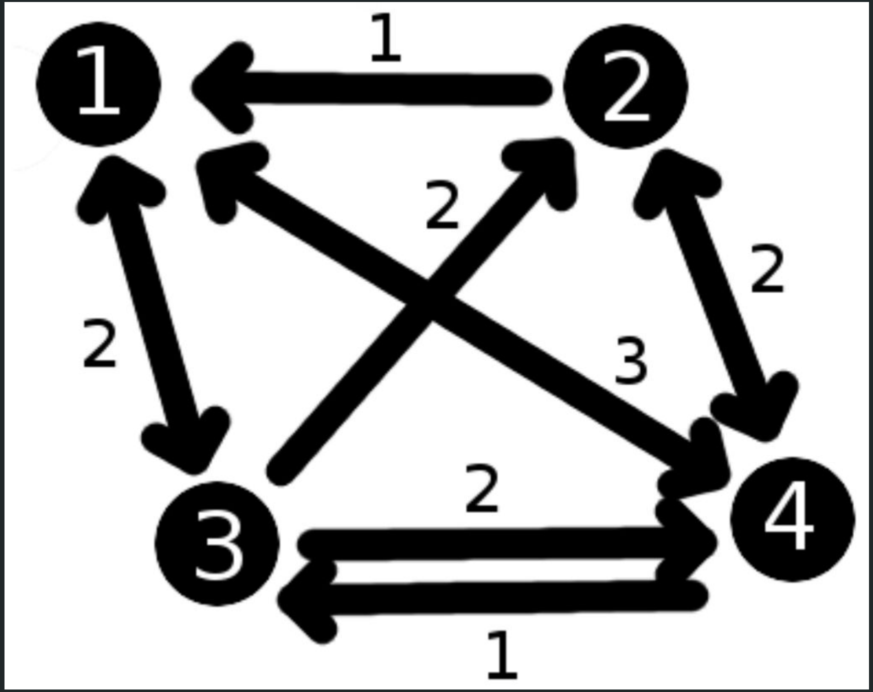


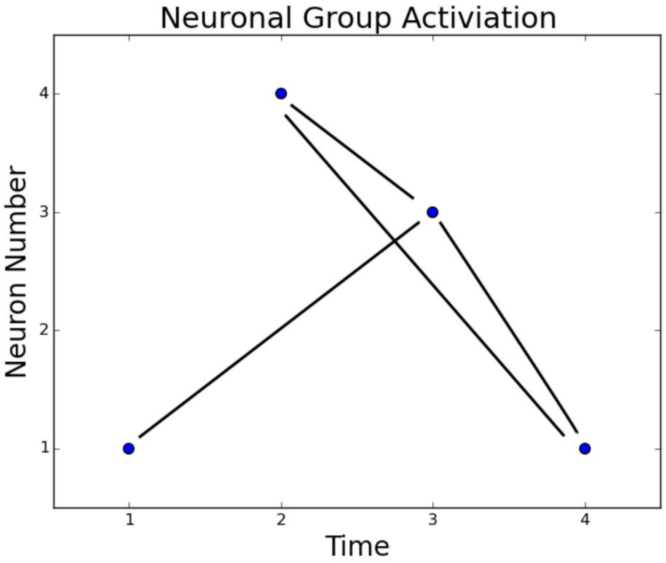


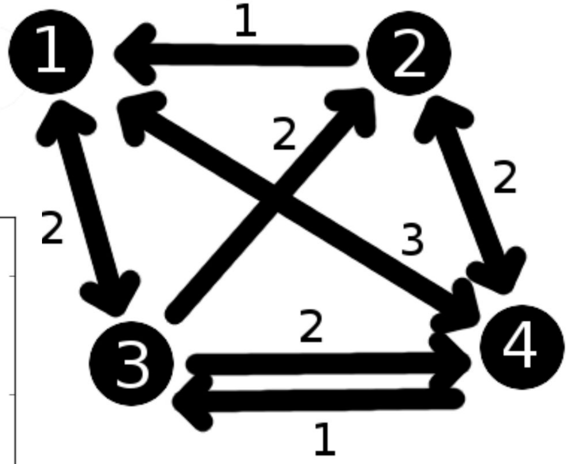
Activating neuron 1 at time 1 and neuron 2 at time 2 creates a polygroup

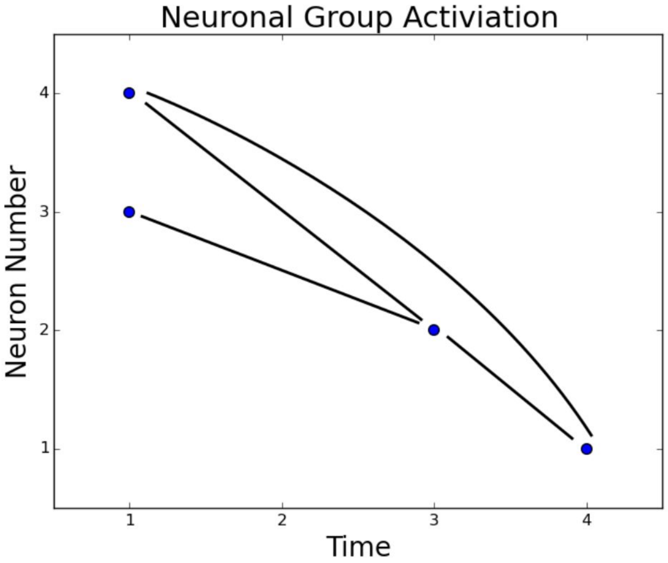
Take a slightly more complicated network

- The number of polygroups increases....
- How can we find them?
- What is the longest one?
- What is the shortest one?
- What are the properties governing these answers?
- Can we construct a network with these properties such that particular polygroups are formed?





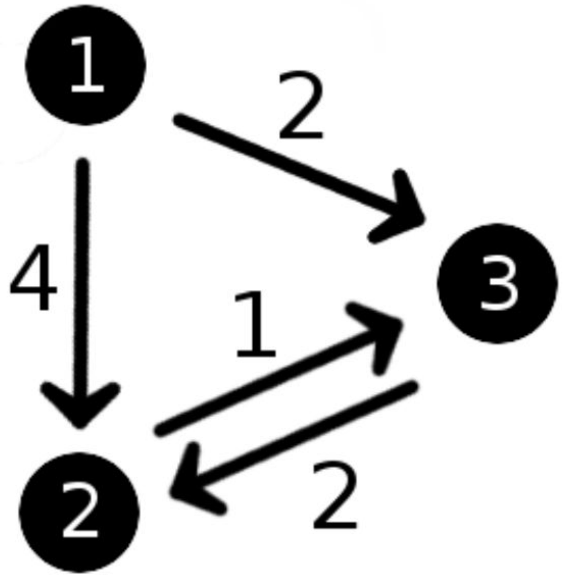




This time there are at least two polygroups. How can we find them all?

Polychrony - Enter a Pseudo Algebra

Networks and Neurons



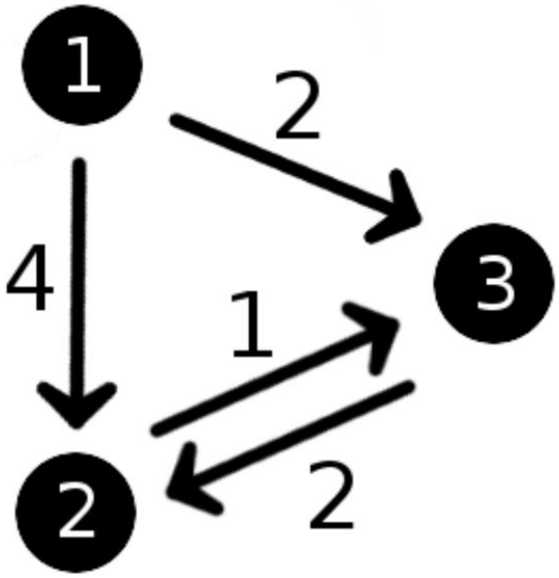
$$N=\begin{bmatrix} 0 & 0 & 0 \\ 4 & 0 & 2 \\ 2 & 1 & 0 \end{bmatrix}$$

$$\vec{n}_1=\begin{pmatrix} 0 \\ 4 \\ 2 \end{pmatrix}$$

$$\vec{n}_2=\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\vec{n}_3=\begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}$$

Activating Neurons

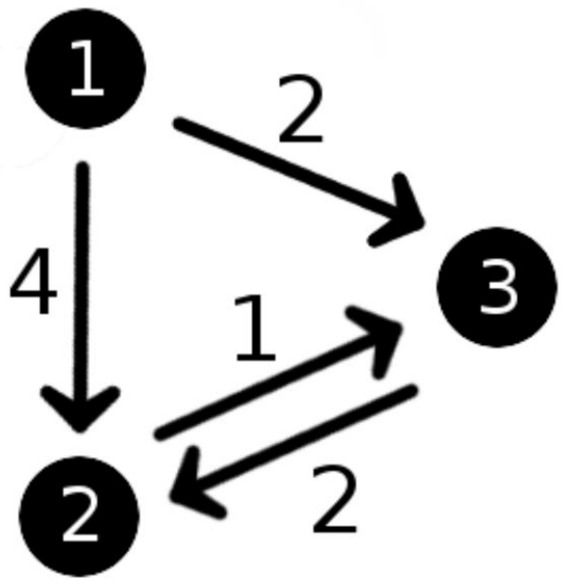


$$\vec{n}_1 = \begin{pmatrix} 0 \\ 4 \\ 2 \end{pmatrix} \qquad \vec{n}_1(1) = \begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\vec{n}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \qquad \vec{n}_2(2) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\vec{n}_3 = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} \qquad \vec{n}_3(1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ * & 0 & 0 \end{pmatrix}$$

Adding Activations

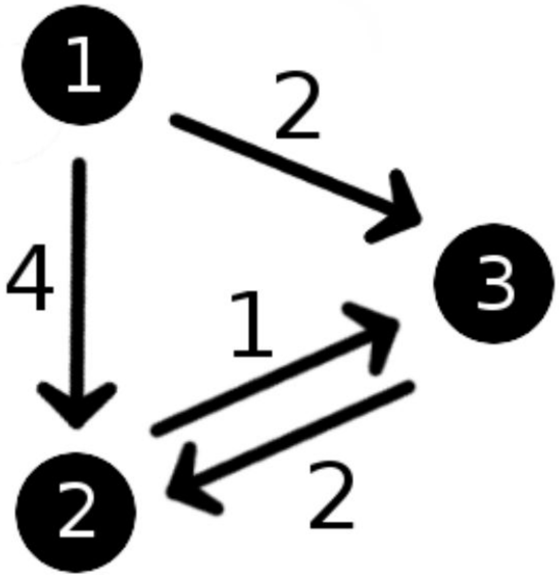


$$\vec{n}_1 = \begin{pmatrix} 0 \\ 4 \\ 2 \end{pmatrix} \qquad \vec{n}_1(1) = \begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\vec{n}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \qquad \vec{n}_2(2) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\vec{n}_1(1) + \vec{n}_2(2) = \begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \end{pmatrix}$$

The Cascade



$$\vec{n}_1(1)+\vec{n}_2(2)=\begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \end{pmatrix}=\begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 1 \\ 0 & 0 & * & 0 & 0 \end{pmatrix}+\vec{n}_3(3)$$

$$\begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 1 \\ 0 & 0 & * & 0 & 0 \end{pmatrix}+\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & * & 0 & 0 \end{pmatrix}=\begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 2 \\ 0 & 0 & * & 0 & 0 \end{pmatrix}$$

The Cascade

$$\begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 2 \\ 0 & 0 & * & 0 & 0 \end{pmatrix} = \begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & * \\ 0 & 0 & * & 0 & 0 \end{pmatrix} + \vec{n}_2(5)$$

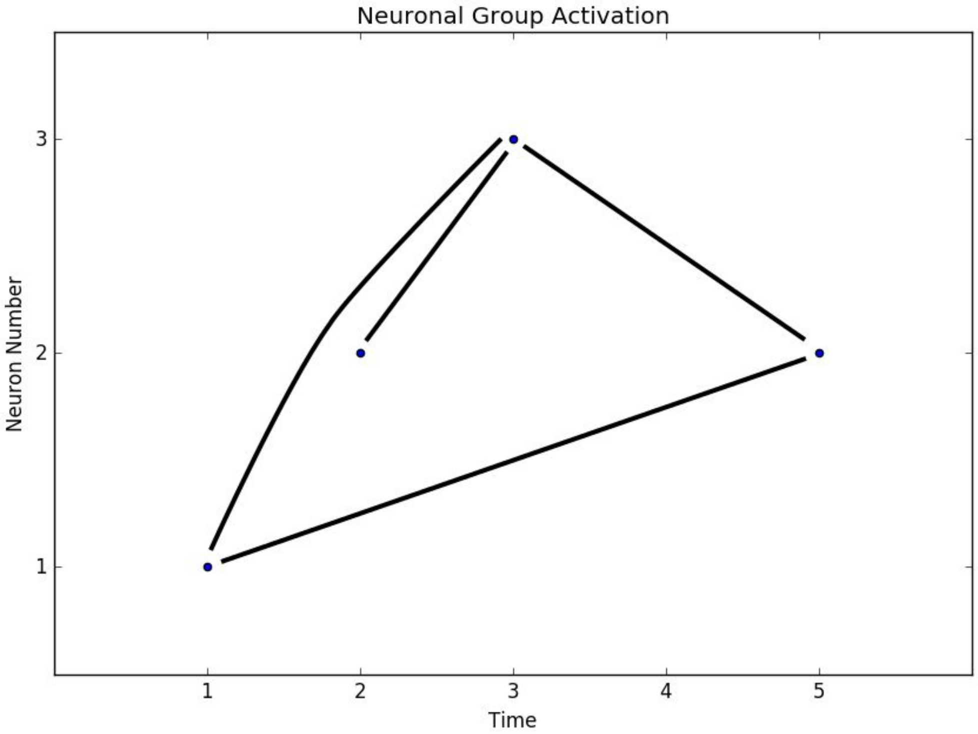
$$\begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & * \\ 0 & 0 & * & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & * & 0 \\ 0 & 0 & * & 0 & 0 & 1 \end{pmatrix}$$

$$\vec{n}_1(1) + \vec{n}_2(2) = \begin{pmatrix} * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & * & 0 \\ 0 & 0 & * & 0 & 0 & 1 \end{pmatrix}$$

The Cascade

$$\vec{n}_1(1)+\vec{n}_2(2)=\begin{pmatrix} * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & * & 0 \\ 0 & 0 & * & 0 & 0 & 1 \end{pmatrix}$$

$$\vec{n}_1(1)+\vec{n}_2(2)=\{(1,1),(2,2),(3,3),(2,5)\}$$



Thoughts on Activations and Cascades

An algebraic means of constructing the raster plots

Can construct all polygroups (needs proof)

Moderately interesting, but entirely cumbersome

Need a more succinct syntax

Activation Decompositions

Decomposing Activations

$$\vec{n}_1 = \begin{pmatrix} 0 \\ 4 \\ 2 \end{pmatrix} \quad \vec{n}_1(1) = \begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\vec{n}_1(1) = \begin{pmatrix} 0 \\ 4 \\ 2 \end{pmatrix}(1) = \begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} = *_1(1) + \vec{b}_1(0) + \vec{b}_2(4+1) + \vec{b}_3(2+1)$$

where...

$$*_1(1) = \begin{pmatrix} * \\ 0 \\ 0 \end{pmatrix} \quad \vec{b}_1(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \vec{b}_2(5) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \vec{b}_3(3) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Decomposing Activations

In other words we can represent the activation of neuron 1 succinctly as

$$\vec{n}_1(1) = *_1(1) + \vec{b}_2(5) + \vec{b}_3(3)$$

where the asterisk term is only for bookkeeping and can usually be omitted

This is much better than dealing with those cludgy activation matrices...

$$\vec{n}_1(1) = \begin{pmatrix} 0 \\ 4 \\ 2 \end{pmatrix}(1) = \begin{pmatrix} * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Decomposing Activations

$$\vec{n}_1 = \begin{pmatrix} 0 \\ 4 \\ 2 \end{pmatrix} \quad \vec{n}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \vec{n}_3 = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}$$

Adding activations can now be compactly represented with these “basis neurons”

$$\vec{n}_1(1) = *_1(1) + \vec{b}_1(0) + \vec{b}_2(5) + \vec{b}_3(3)$$

$$\vec{n}_2(2) = *_2(2) + \vec{b}_1(0) + \vec{b}_2(0) + \vec{b}_3(3)$$

$$\vec{n}_1(1) + \vec{n}_2(2) = *_1(1) + *_2(2) + \vec{b}_2(5) + \vec{b}_3(3) + \vec{b}_3(3)$$

This notation leads to the following convenient property for cascades

$$\vec{b}_i(t) + \vec{b}_i(t) = *_i(t) + \vec{n}_i(t)$$

Decomposing Activations

$$\vec{n}_1=\begin{pmatrix}0\\4\\2\end{pmatrix} \quad \vec{n}_2=\begin{pmatrix}0\\0\\1\end{pmatrix} \quad \vec{n}_3=\begin{pmatrix}0\\2\\0\end{pmatrix}$$

Applying the cascade to our equation yields

$$\vec{n}_1(1)+\vec{n}_2(2)=*_1(1)+*_2(2)+\vec{b}_2(5)+\vec{b}_3(3)+\vec{b}_3(3)$$

$$\vec{n}_1(1)+\vec{n}_2(2)=*_1(1)+*_2(2)+\vec{b}_2(5)+*_3(3)+\vec{n}_3(3)$$

$$\vec{n}_1(1)+\vec{n}_2(2)=*_1(1)+*_2(2)+*_3(3)+\vec{b}_2(5)+\vec{b}_1(0)+\vec{b}_2(5)+\vec{b}_3(0)$$

$$\vec{n}_1(1)+\vec{n}_2(2)=*_1(1)+*_2(2)+*_3(3)+\vec{b}_2(5)+\vec{b}_2(5)$$

$$\vec{n}_1(1)+\vec{n}_2(2)=*_1(1)+*_2(2)+*_3(3)+*_2(5)+\vec{n}_2(5)$$

Decomposing Activations

$$\vec{n}_1=\begin{pmatrix}0\\4\\2\end{pmatrix} \quad \vec{n}_2=\begin{pmatrix}0\\0\\1\end{pmatrix} \quad \vec{n}_3=\begin{pmatrix}0\\2\\0\end{pmatrix}$$

Applying the cascade to our equation yields

$$\vec{n}_1(1)+\vec{n}_2(2)=*_1(1)+*_2(2)+*_3(3)+*_2(5)+\vec{n}_2(5)$$

$$\vec{n}_1(1)+\vec{n}_2(2)=*_1(1)+*_2(2)+*_3(3)+*_2(5)+\vec{b}_3(6)$$

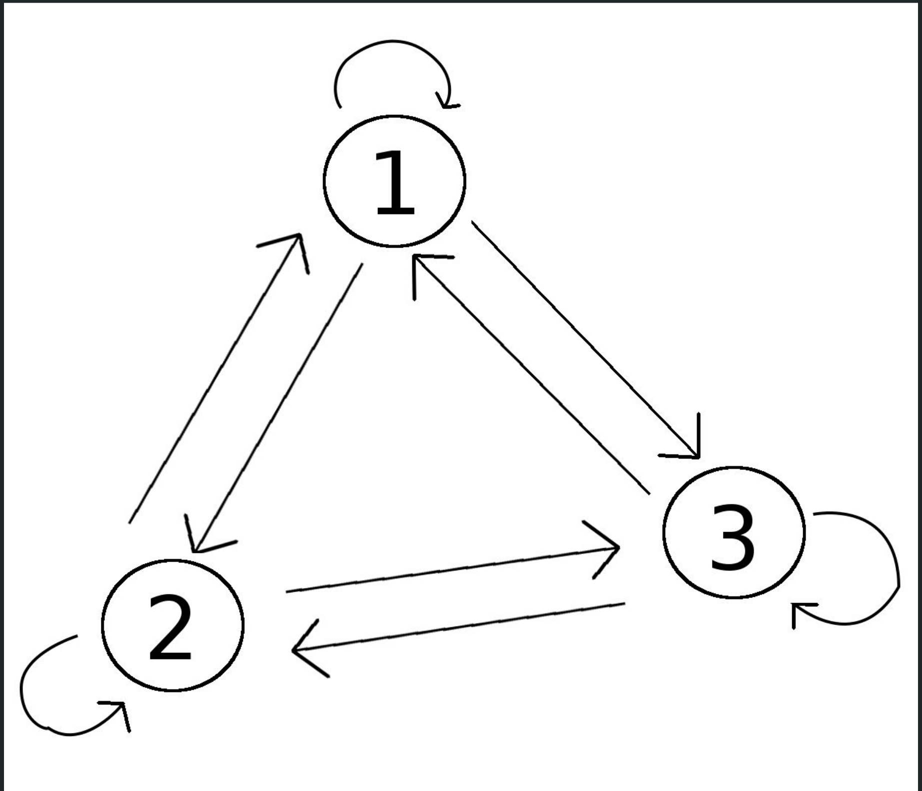
Which once again is another representation of our polygroup

$$\vec{n}_1(1)+\vec{n}_2(2)=\{(1,1),(2,2),(3,3),(2,5)\}$$

Generalization

Arbitrary Network

What can the above techniques allow us to do with an arbitrary network?



Define the Arbitrary Network

$$N=\begin{bmatrix} n_{11} & n_{12} & n_{13} \\ n_{21} & n_{22} & n_{23} \\ n_{31} & n_{32} & n_{33} \end{bmatrix}$$

$$\vec{n}_1=\begin{pmatrix} n_{11} \\ n_{21} \\ n_{31} \end{pmatrix} \qquad \vec{n}_2=\begin{pmatrix} n_{12} \\ n_{22} \\ n_{32} \end{pmatrix} \qquad \vec{n}_3=\begin{pmatrix} n_{13} \\ n_{23} \\ n_{33} \end{pmatrix}$$

Encoding

What can we do with this arbitrary network? Let us encode information into it...

Assume each neuron's number is the value assigned to a sequence when it is activated

Let us try to encode the following sequence

$$S=\{3,1,1,2\}$$

Encoding

$$S=\{3,1,1,2\}$$

This means we need neuron 3 to fire first, followed by neuron 1 twice, finished by neuron 2...

Can we design a network with this property?

Because neuron 3 and 1 are the first to fire, we choose them to be the neurons which activate the polygroup

Encoding

$$S = \{3, 1, 1, 2\}$$

We describe this mathematically as follows

$$\vec{n}_3(t_1) + \vec{n}_1(t_2) = *_3(t_1) + *_1(t_2) + \vec{n}_1(t_3) + \vec{n}_2(t_4)$$

where...

$$t_1 < t_2 < t_3 < t_4$$

We are looking for a particular network, N, and a set of activation times, t1 through t4 such that the above expression is true

Intuitively, we can see that there will be infinitely many solutions. Can we find one of them? Yes, we can construct it using the basis neurons.

Encoding

$$S=\{3,1,1,2\}$$

Let's expand the left hand side and see what happens

$$\vec{n}_3(t_1)+\vec{n}_1(t_2)=*_3(t_1)+*_1(t_2)+\vec{b}_1(t_1+n_{13})+\vec{b}_2(t_1+n_{23})+\vec{b}_3(t_1+n_{33})+\vec{b}_1(t_2+n_{11})+\vec{b}_2(t_2+n_{21})+\vec{b}_3(t_2+n_{31})$$

Neuron 1 is the next which must be activated, which implies...

$$t_1+n_{13}=t_2+n_{11}=t_3$$

The above would create the following cascade

$$\vec{b}_1(t_1+n_{13})+\vec{b}_1(t_2+n_{11})=*_1(t_3)+\vec{n}_1(t_3)$$

Encoding

$$S=\{3,1,1,2\}$$

Putting the new cascade into the expression yields the following RHS

$$*_3(t_1)+*_1(t_2)+*_1(t_3)+\vec{b}_2(t_1+n_{23})+\vec{b}_3(t_1+n_{33})+\vec{b}_2(t_2+n_{21})+\vec{b}_3(t_2+n_{31})+\vec{b}_1(t_3+n_{11})+\vec{b}_2(t_3+n_{21})+\vec{b}_3(t_3+n_{31})$$

Neuron 2 is the next which must be activated, which implies...

$t_1+n_{23}=t_2+n_{21}=t_4$ OR $t_1+n_{23}=t_3+n_{21}=t_4$ OR $t_2+n_{21}=t_3+n_{21}=t_4$

Notice the right most expression is rubish. This is because it is requiring that the activation of neuron 1, at two different times, somehow converge onto neuron 2.

For lack of any other constraints or insight, we choose the leftmost expression

Encoding

$$S=\{3,1,1,2\}$$

We also require that the rest of the activity does not create any more activations!

This complicates things slightly, but we can add these constraints as well

$$*_3(t_1)+*_1(t_2)+*_1(t_3)+\vec{b}_2(t_1+n_{23})+\vec{b}_3(t_1+n_{33})+\vec{b}_2(t_2+n_{21})+\vec{b}_3(t_2+n_{31})+\vec{b}_1(t_3+n_{11})+\vec{b}_2(t_3+n_{21})+\vec{b}_3(t_3+n_{31})$$

$$t_1+n_{33}-t_2-n_{31}>0$$

$$t_1+n_{33}-t_3-n_{31}>0$$

Encoding

$$S=\{3,1,1,2\}$$

Putting all of this together gives produces a system of equations

$$t_2+n_{23}-t_2-n_{21}=0$$

$$t_1+n_{13}-t_2-n_{11}=0$$

$$t_1+n_{33}-t_2-n_{31}>0$$

$$t_1+n_{33}-t_3-n_{31}>0$$

$$t_1 < t_2 < t_3 < t_4$$

These leaves us with many questions and comments

Encoding

What are the properties of this system such that a solution exists, or does not exist?

What are the limits of encoding? Is there a maximum length that can be encoded given a particular number of neurons?

Would a different “Arbitrary Network” provide different encoding options?

Given the large degrees of freedom, what more assumptions can be made? Can we determine the activation times a priori and construct the network accordingly?

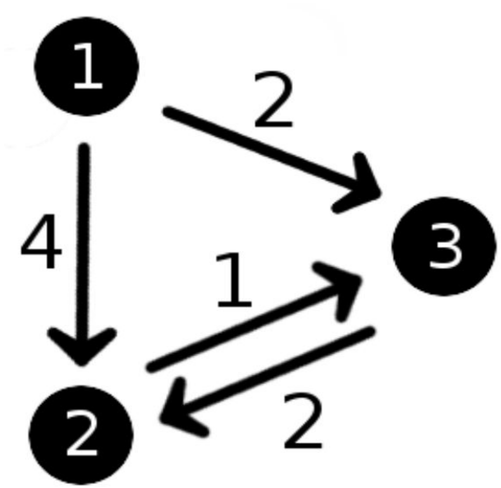
What if we had more information about the delays in connectivity?

Determining Paths and Delays

The Tree of Matrices

Let us reexamine our initial network...

Can we determine the delays required to get from any one neuron to another?



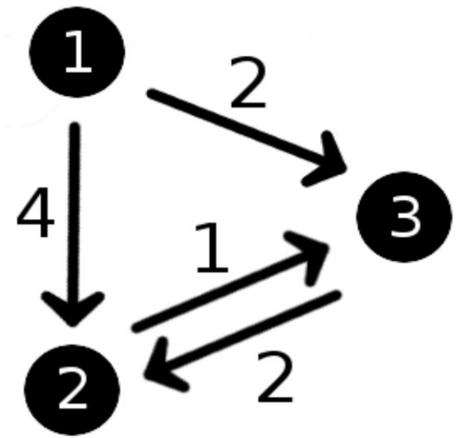
$$N = \begin{bmatrix} 0 & 0 & 0 \\ 4 & 0 & 2 \\ 2 & 1 & 0 \end{bmatrix}$$

The Tree of Matrices

Yes. Let's make up another goofy math operation. Matrix multiplication...

$$\begin{bmatrix} 0 & 0 & 0 \\ 4 & 0 & 2 \\ 2 & 1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 \\ 4 & 0 & 2 \\ 2 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 4 \\ 0 & 5 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}$$

The Tree of Matrices



$$\begin{bmatrix} 0 & 0 & 0 \\ 4 & 0 & 2 \\ 2 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 4 \\ 0 & 5 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 5 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 4 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$