Computational Neuroscience Meeting
Florence, Italy

July 5, 2025

Training spiking neural networks to generate experimentally recorded neural activities

Christopher Kim

Howard University
Washington, DC, USA

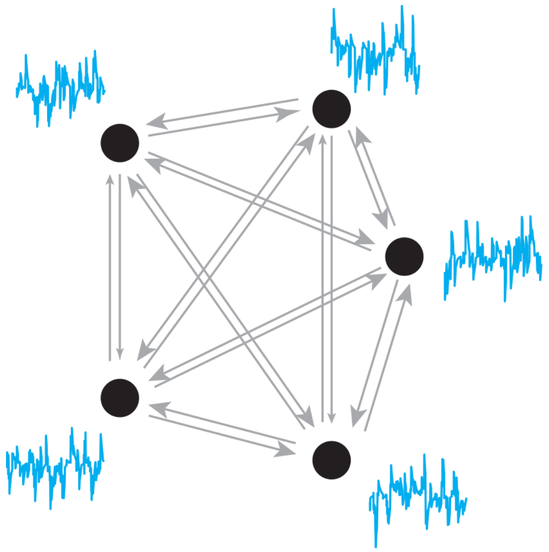
Remy Yovanno

NIMH/NIH
Bethesda, MD, USA

Tutorial objectives

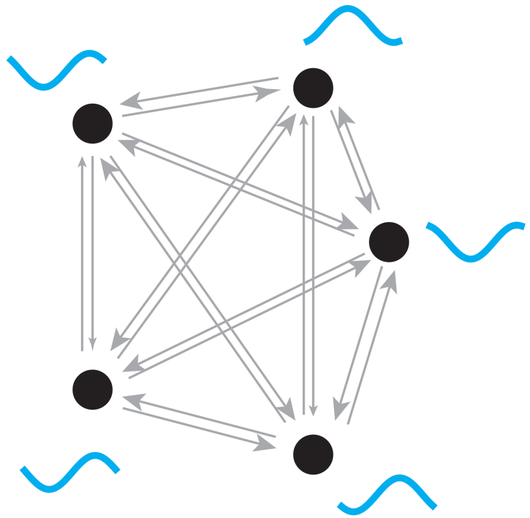
1. Train a recurrent neural network to generate target activity patterns

Activity of neurons
before training



Neurons generate
random activity patterns

Activity of neurons
after training



Neurons generate
target activity patterns
(e.g. sine waves)

Tutorial objectives

2. Apply the training scheme (**R**ecursive **L**east **S**quares) to different network models

abstract

realistic

1. Rate-based network
   - Recurrent network dynamics without spiking activity
   - Introduce the RLS algorithm

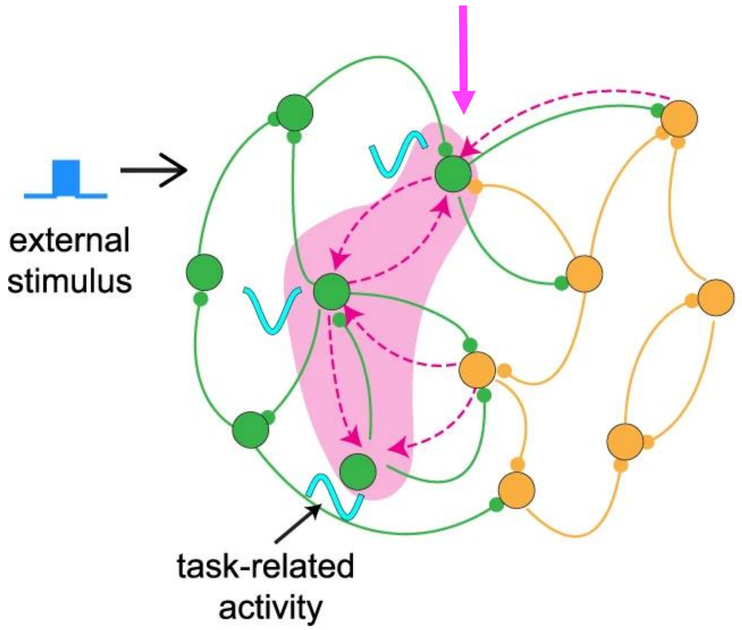
2. Spiking neural network (generic)
   - Recurrent network dynamics with spiking neurons
   - Adopt RLS to spiking neural networks

3. Spiking neural network operating in the balanced regime
   - Consider strongly recurrent excitatory-inhibitory spiking neural network
   - Train a subset of neurons to generate experimentally recorded neural activity

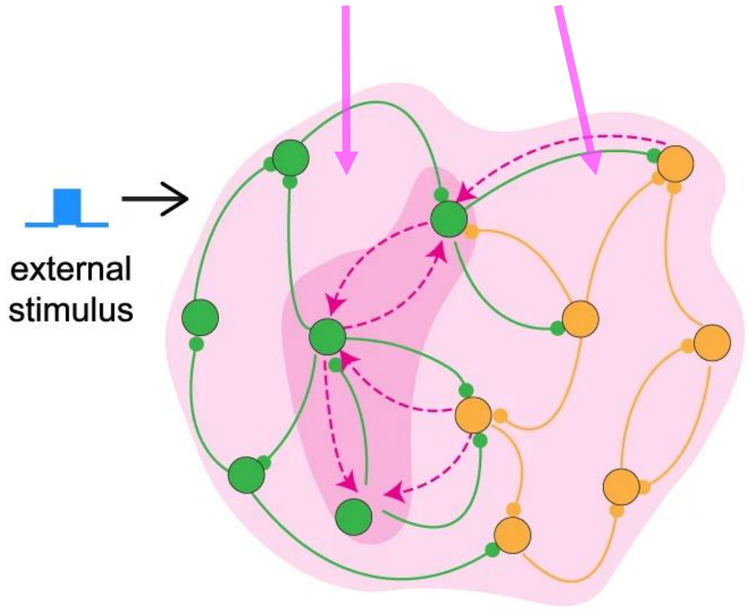
Tutorial objectives

3. Analyze the activity of trained / untrained neurons in a trained network

A subset of neurons are
trained to generate target activity



Examine what happens to
neurons that are not trained



Tutorial objectives

What can we learn from trained networks?

Analyze the differential equations of trained networks to characterize the neural dynamics (e.g., attractors, separatrix)

- Finkelstein, Fontolan et al (2021). Attractor dynamics gate cortical information flow during decision-making

Analyze the trained recurrent connectivity structure to understand how the network generates the neural activity

- Rajan et al (2016). Recurrent network models of sequence generation and memory
- Andalman et al (2019). Neuronal dynamics regulating brain and behavioral state transitions

Related studies:
Abstract latent model that characterizes the dynamics of neural data

- Vinograd et al (2024). Causal evidence of a line attractor encoding an affective state
- Nair et al (2023). An approximate line attractor in the hypothalamus encodes an aggressive state

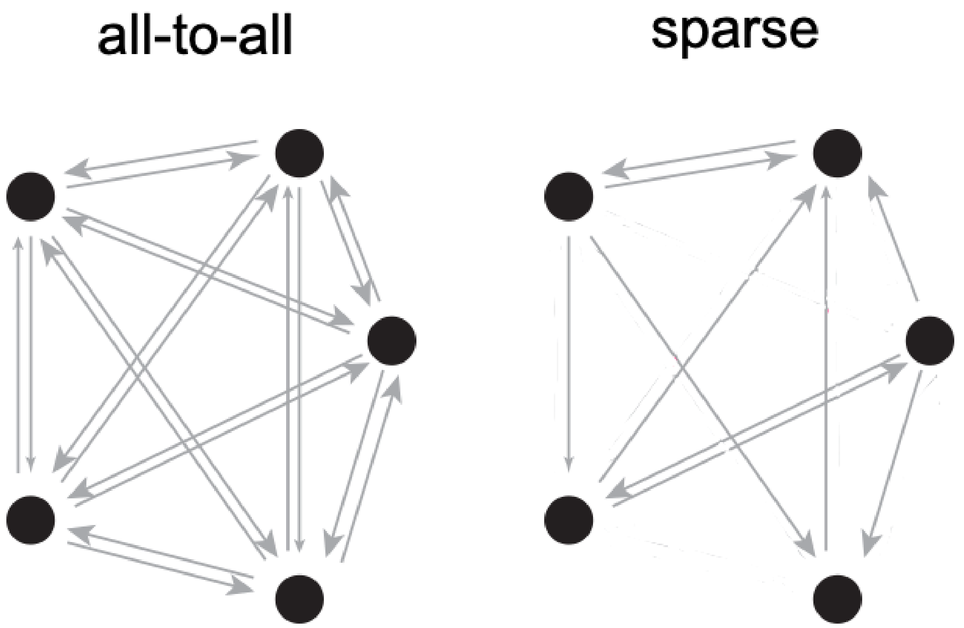
Connectome-constrained neural network models

- Cowley et al (2024). Mapping model units to visual neurons reveals population code for social behaviour
- Lappalainen et al (2024). Connectome-constrained networks predict neural activity across the fly visual system

Part 1. Rate-based model

Initial network

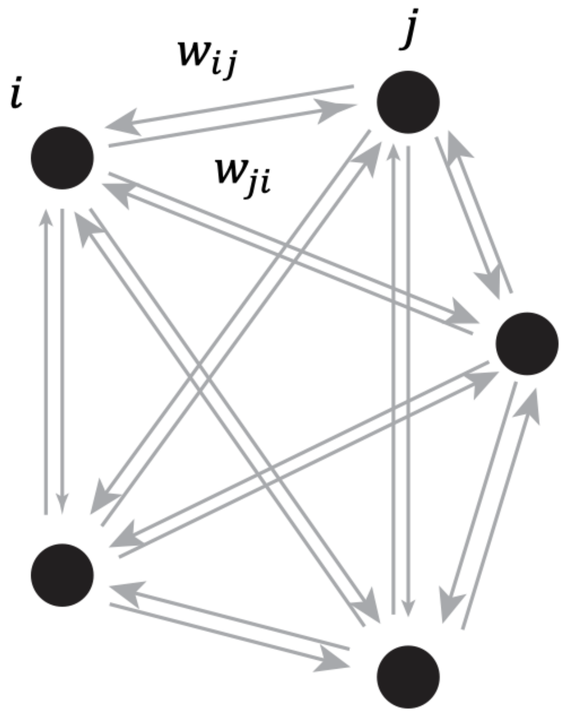
Define connectivity



Initial network

Define connectivity

Initialize weights



large w_{ij} = strong coupling

small w_{ij} = weak coupling

Initial weights: $w_{ij} \sim \mathcal{N}(0, \sigma^2)$

mean = 0, std = σ

Initial network

Define connectivity

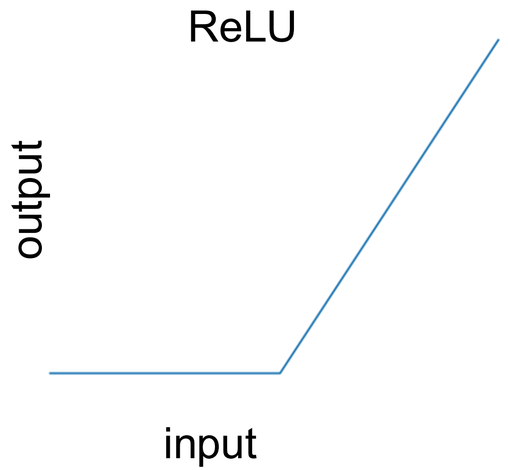
Initialize weights

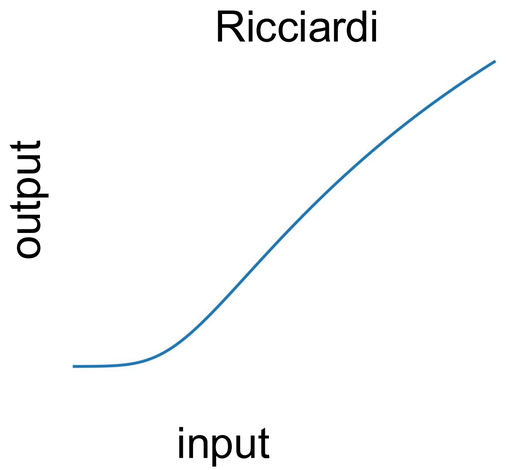
Single neuron model

Rate model

Define activation function

How does a neuron’s input relate to its output?





Easy to implement

Agrees with *in vivo* neural measurements

Captures nonlinearity required for suppression

LaFosse et al. (2024)

Sanzeni et al. (2020)

Initial network

Define connectivity

Initialize weights

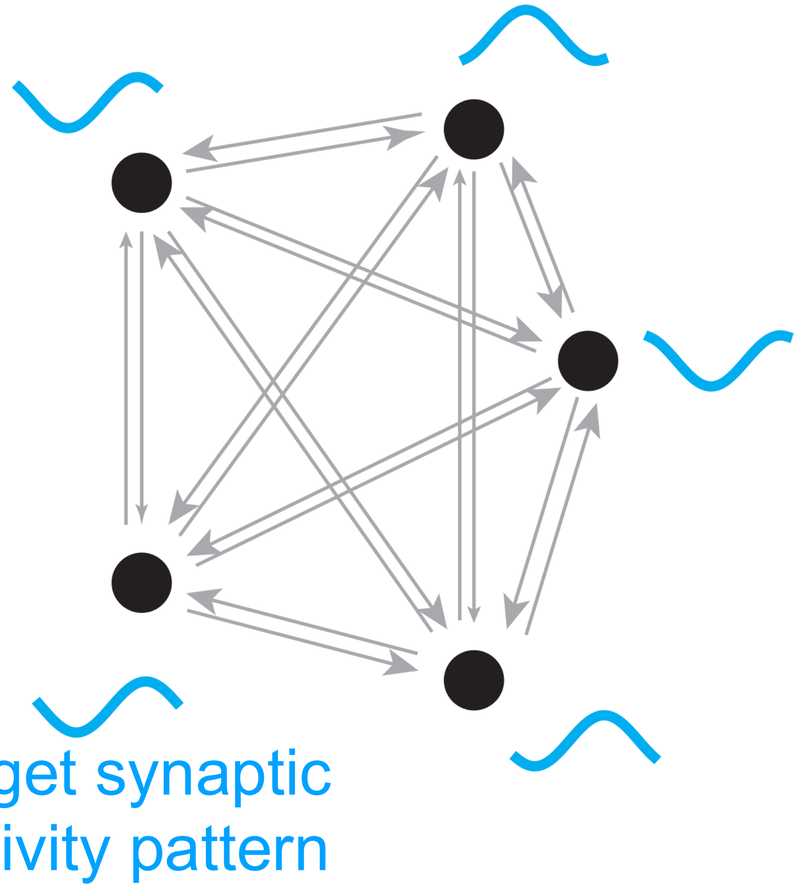
Single neuron model

Rate model

Define activation function

Target activity patterns

Generate target rates



Initial network

Define connectivity

Initialize weights

Single neuron model

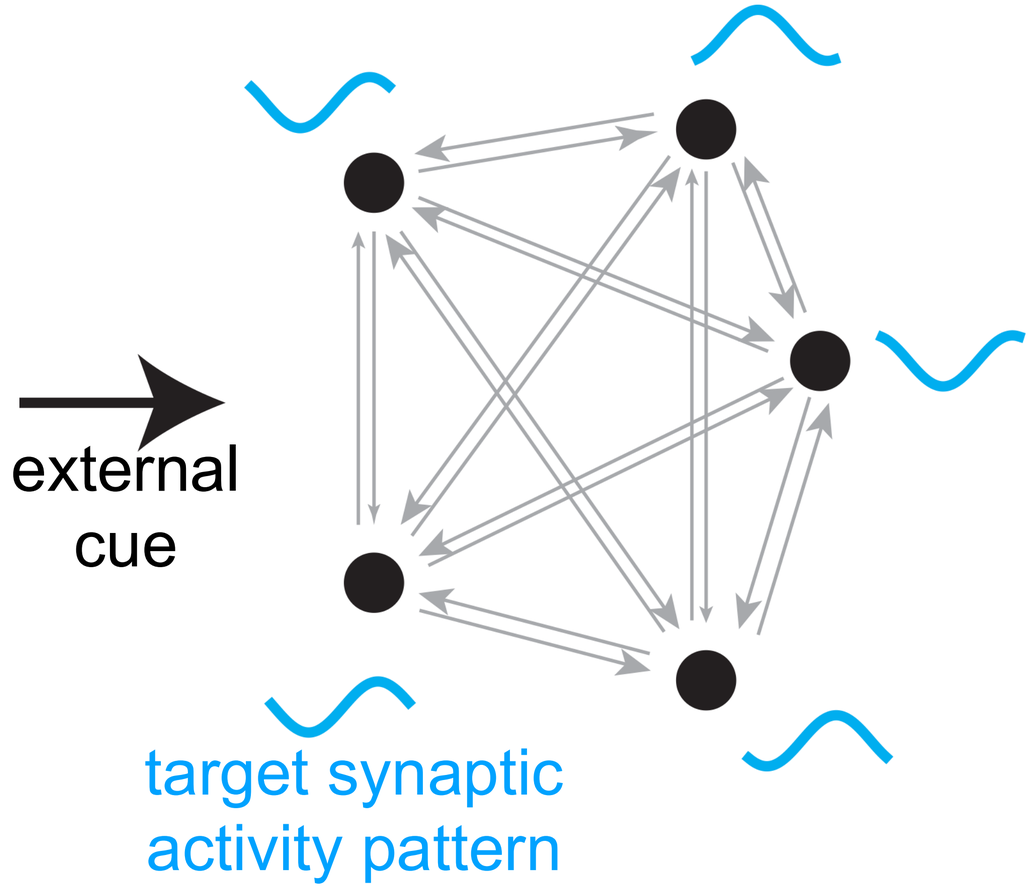
Rate model

Define activation function

Target activity patterns

Generate target rates

Define external stimulus



Initial network

Define connectivity

Initialize weights

Single neuron model

Rate model

Define activation function

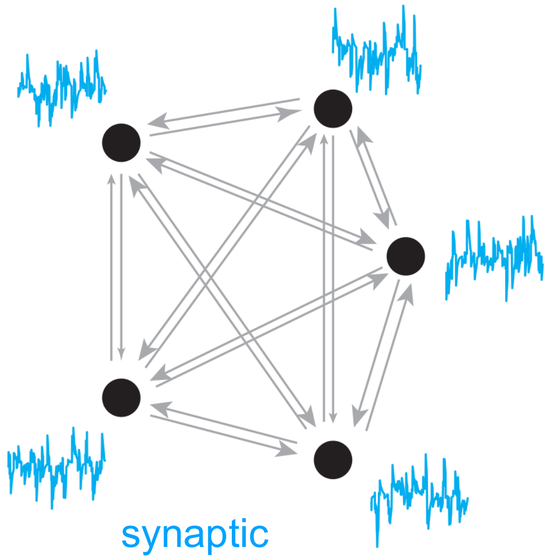
Target activity patterns

Generate target rates

Define external stimulus

Train RNN activity

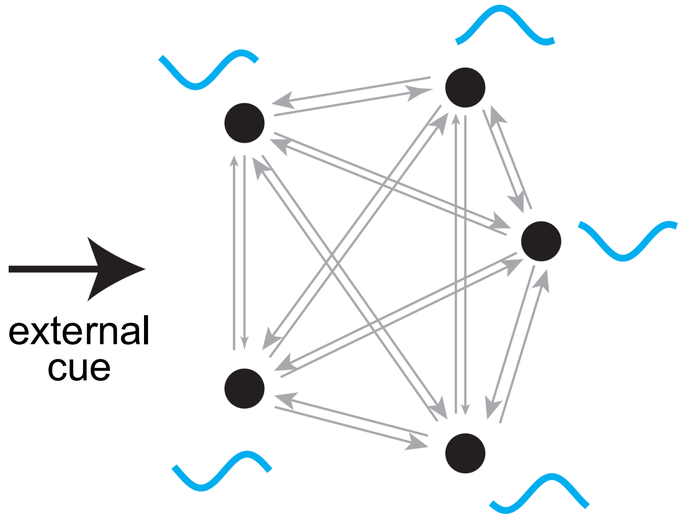
Before learning



synaptic activity (u)

After learning

external cue



target synaptic activity pattern

Initial network

Define connectivity

Initialize weights

Single neuron model

Rate model

Define activation function

Target activity patterns

Generate target rates

Define external stimulus

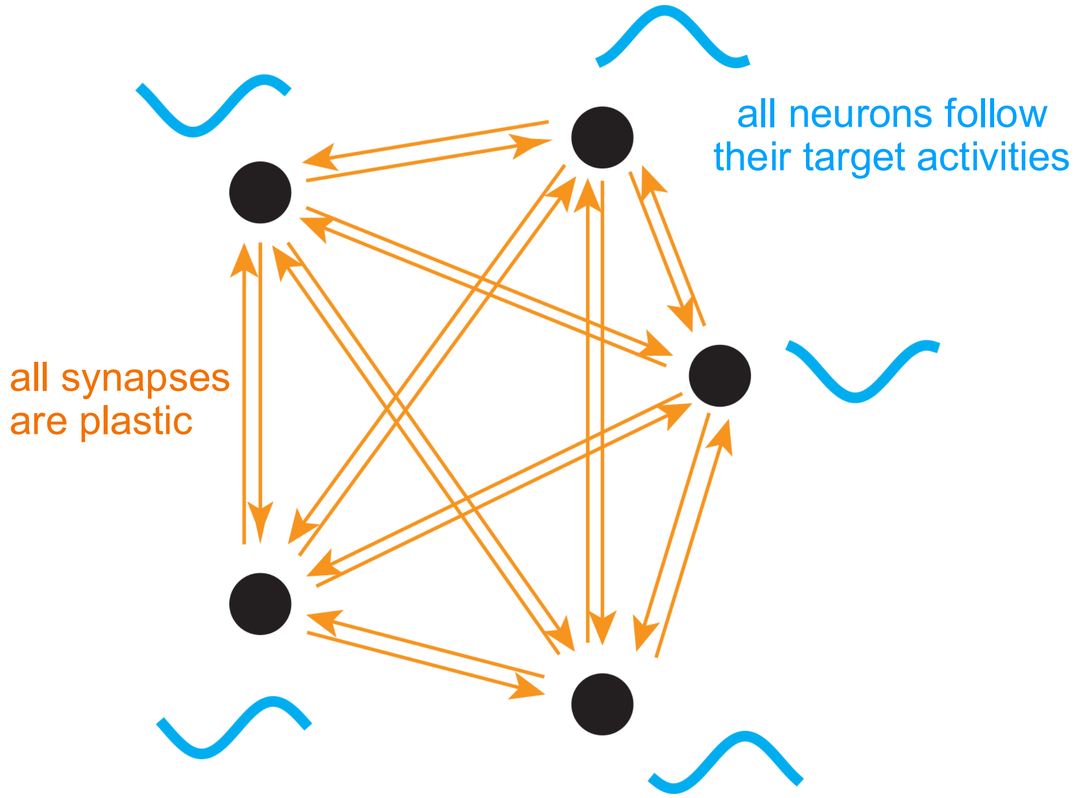
Train RNN activity

Select neurons to train

Initialize plastic connections

All the synapses are plastic

All the neurons are trained



Initial network

Define connectivity

Initialize weights

Single neuron model

Rate model

Define activation function

Target activity patterns

Generate target rates

Define external stimulus

Train RNN activity

Select neurons to train

Initialize plastic connections

Rate model

Simulate network dynamics

Network dynamics of a rate-based RNN

Numerically integrate the rate equation below:

activation function

$$\tau \frac{dr}{dt} = -r + \phi(w_{rec} \cdot r + I_{ext})$$

time constant

input

recurrent + external

Initial network

Define connectivity

Initialize weights

Single neuron model

Rate model

Define activation function

Target activity patterns

Generate target rates

Define external stimulus

Train RNN activity

Rate model

Select neurons to train

Initialize plastic connections

Simulate network dynamics

Update plastic weights

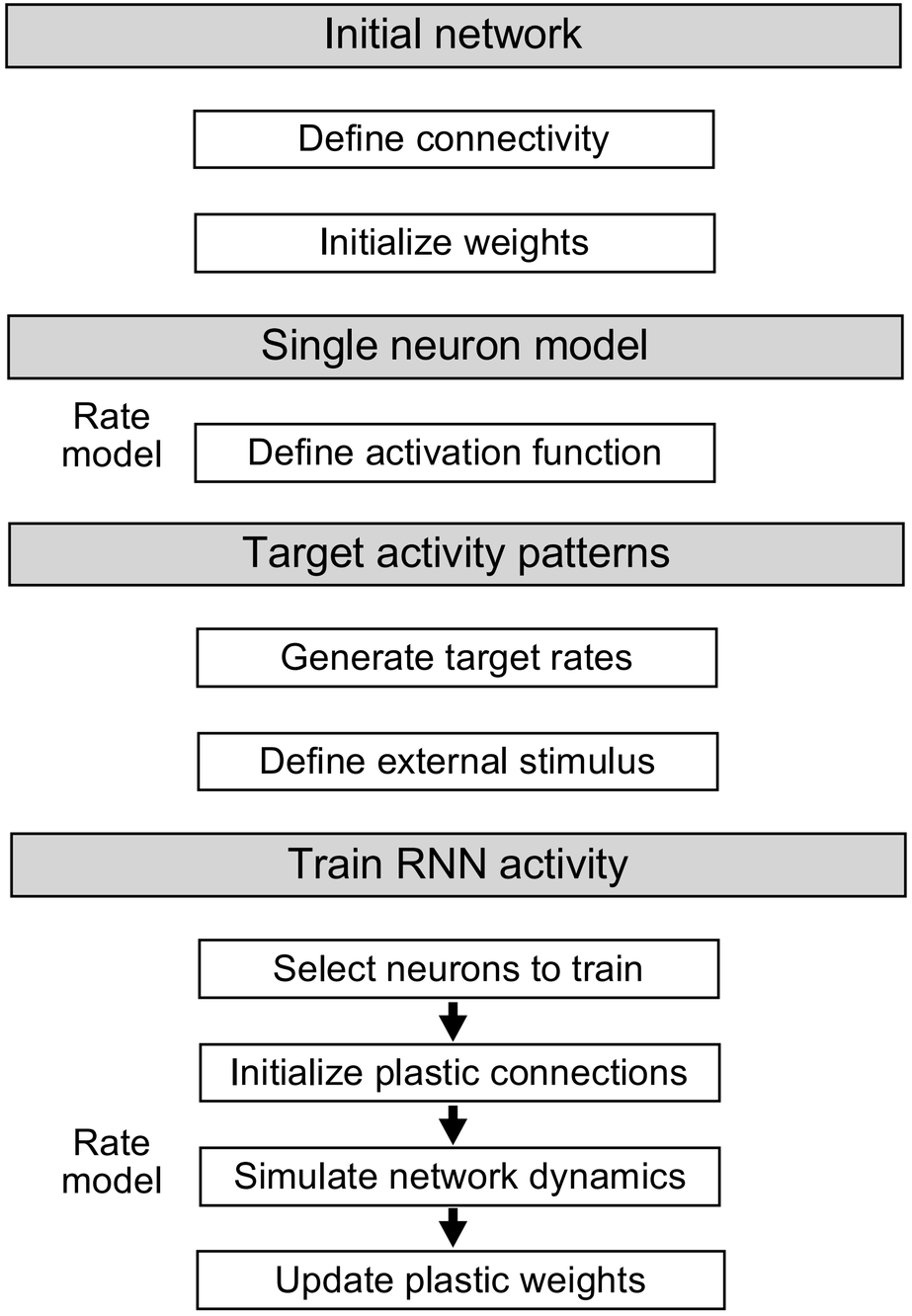
target activity pattern

network activity

We'd like to train the network to learn the target activity.

Here, we are training the recurrent input to each cell $w_{rec}r$.

This is accomplished by changing the recurrent weights w_{rec}



Weights are updated by applying a learning rule

backpropagation through time (BPTT)	**recursive least squares (RLS)**
Applies weight updates after processing entire timeseries at once. Uses the chain rule to compute the gradient of the cost function and update weights accordingly.	Updates weights at each learning timestep. Recursively computes the inverse of the correlation matrix of inputs, P, which is used to update the weights directly.
Pros: Can be used with deep network architectures Cons: Uses more memory for longer target timecourses.	Pros: Converges quickly and handles long timecourses efficiently. Cons: Difficult to apply to deep network architectures and LSTMs

Both process sequential data and work well to train RNNs.

Initial network

Define connectivity

Initialize weights

Single neuron model

Rate model

Define activation function

Target activity patterns

Generate target rates

Define external stimulus

Train RNN activity

Rate model

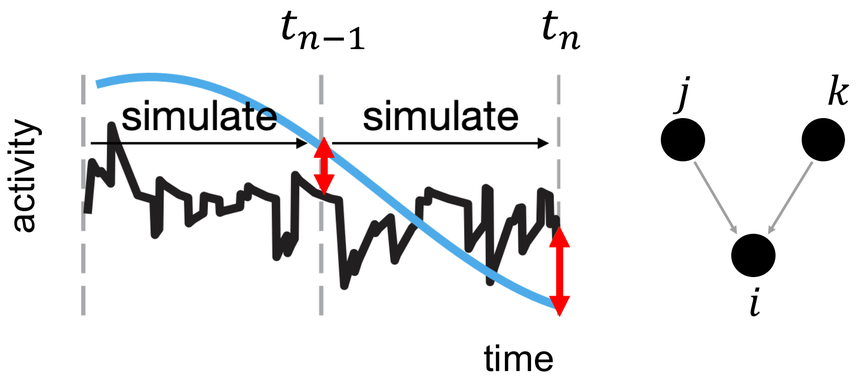
Select neurons to train

Initialize plastic connections

Simulate network dynamics

Update plastic weights

Overview of recursive least squares (RLS)



inverse correlation matrix P_i

$$\begin{bmatrix} P_{jj} & P_{kj} \\ P_{jk} & P_{kk} \end{bmatrix}$$

1. Simulate the network for time Δt

2. Update the inverse correlation matrix of inputs P, given the current network state.

3. Compute the **error** between the **target** and **network** activity at t_{n-1} and t_n.

4. Use the error and P to find the optimal update of the recurrent weights to minimize the error through the cost function.

Initial network

Define connectivity

Initialize weights

Single neuron model

Rate model

Define activation function

Target activity patterns

Generate target rates

Define external stimulus

Train RNN activity

Rate model

Select neurons to train

Initialize plastic connections

Simulate network dynamics

Update plastic weights

Minimize the following cost function by optimizing $\mathbf{w}$:

$$C = \frac{1}{2}\sum_{t=1}^{T}[u_t - f_t]^2 + \frac{\lambda}{2}\parallel \mathbf{w} \parallel^2$$

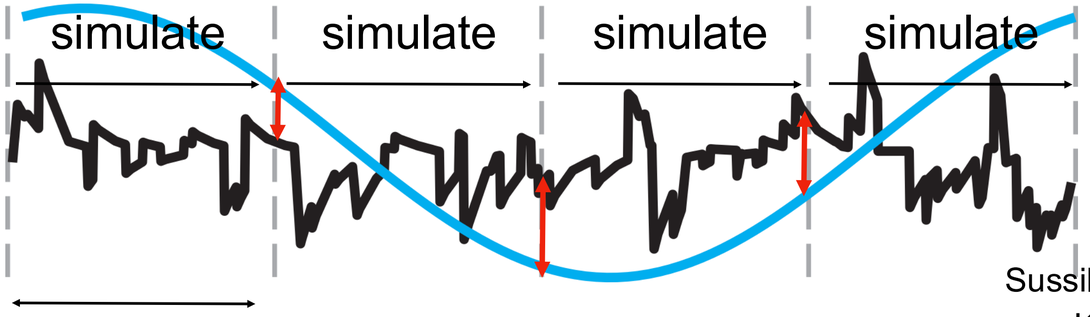
synaptic drive *u* should follow the target pattern *f*

Express u in terms of $\mathbf{w}$ and $\mathbf{r}$.

$$C = \frac{1}{2}\sum_{t=1}^{T}[\mathbf{w}\cdot\mathbf{r}_t - f_t]^2 + \frac{\lambda}{2}\parallel \mathbf{w} \parallel^2$$

Derive RLS (recursive least squares) algorithm that optimizes $\mathbf{w}$ every Δt (=20ms) to reduce C.

update $\mathbf{w}_{t-1}$ update $\mathbf{w}_t$ update $\mathbf{w}_{t+1}$



Sussillo and Abbott (2009)
Kim and Chow (2018)
Nicola and Clopath (2018)

Initial network

Define connectivity

Initialize weights

Single neuron model

Rate model

Define activation function

Target activity patterns

Generate target rates

Define external stimulus

Train RNN activity

Rate model

Select neurons to train

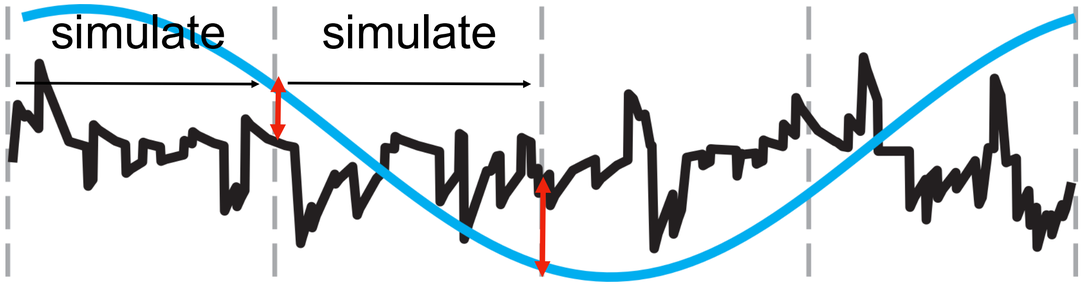
Initialize plastic connections

Simulate network dynamics

Update plastic weights

update $\mathbf{w}_{n-1}$

update $\mathbf{w}_n$



At time step n-1

$$C(\mathbf{w}_{n-1}) = \frac{1}{2}\sum_{t=1}^{n-1}(\mathbf{w}_{n-1}\cdot\mathbf{r}_t - f_t)^2 + \frac{\lambda}{2}\parallel \mathbf{w}_{n-1} \parallel^2$$

$$\nabla_{\mathbf{w}_{n-1}} C(\mathbf{w}_{n-1}) = \sum_{t=1}^{n-1}(\mathbf{w}_{n-1}\cdot\mathbf{r}_t - f_t)\mathbf{r}_t + \lambda\mathbf{w}_{n-1} = \mathbf{0}$$

Sussillo and Abbott (2009)
Kim and Chow (2018)
Nicola and Clopath (2018)

Initial network

Define connectivity

Initialize weights

Single neuron model

Rate model

Define activation function

Target activity patterns

Generate target rates

Define external stimulus

Train RNN activity

Rate model

Select neurons to train

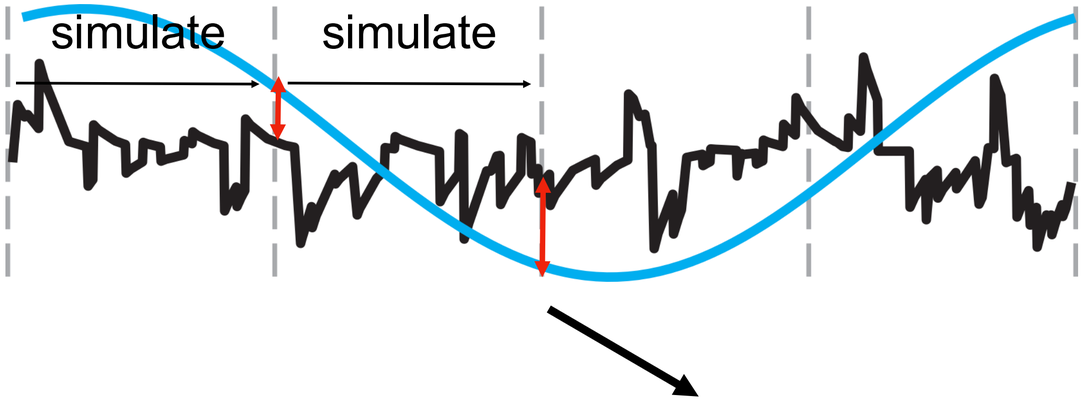
Initialize plastic connections

Simulate network dynamics

Update plastic weights

update $\mathbf{w}_{n-1}$

update $\mathbf{w}_n$



$$C(\mathbf{w}_n) = \sum_{t=1}^{n} (\mathbf{w}_n \cdot \mathbf{r}_t - f_t)^2 + \lambda \parallel \mathbf{w}_n \parallel^2$$

$$\nabla_{\mathbf{w}_n} C(\mathbf{w}_n) = \sum_{t=1}^{n} (\mathbf{w}_n \cdot \mathbf{r}_t - f_t)\mathbf{r}_t + \lambda \mathbf{w}_n = \mathbf{0}$$

Sussillo and Abbott (2009)
Kim and Chow (2018)
Nicola and Clopath (2018)

Initial network

Define connectivity

Initialize weights

Single neuron model

Rate model

Define activation function

Target activity patterns

Generate target rates

Define external stimulus

Train RNN activity

Rate model

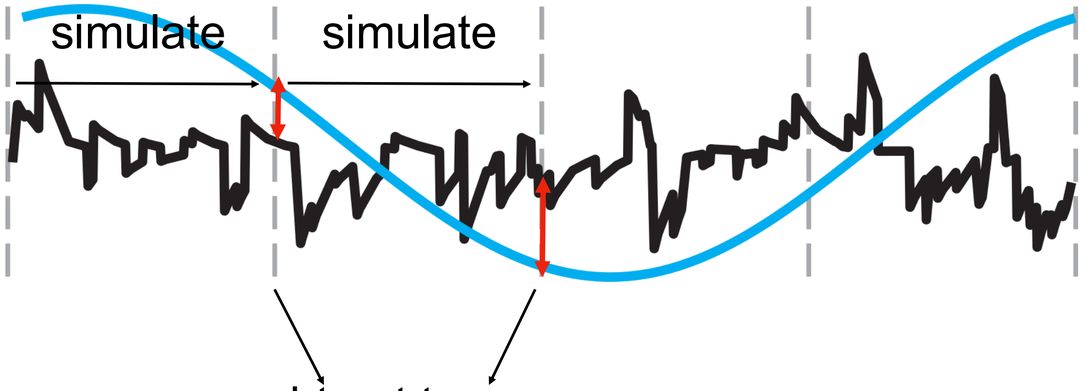
Select neurons to train

Initialize plastic connections

Simulate network dynamics

Update plastic weights

update $\mathbf{w}_{n-1}$ update $\mathbf{w}_n$



subtract two eqn
to get an update rule

$$\mathbf{w}_n = \mathbf{w}_{n-1} - e_n P_n \mathbf{r}_n$$

$$e_n = \mathbf{w}_{n-1} \cdot \mathbf{r}_n - f_n$$

$$P_n = \left(\sum_{t=1}^{n} \mathbf{r}_t \mathbf{r}_t' + \lambda I \right)^{-1}$$

Update P_n recursively:

$$P_n = P_{n-1} - \frac{P_{n-1} \mathbf{r}_n \mathbf{r}_n' P_{n-1}}{1 + r_n' P_{n-1} r_n}$$

Sussillo and Abbott (2009)
Kim and Chow (2018)
Nicola and Clopath (2018)

Initial network

Define connectivity

Initialize weights

Single neuron model

Rate model

Define activation function

Target activity patterns

Generate target rates

Define external stimulus

Train RNN activity

Rate model

Select neurons to train

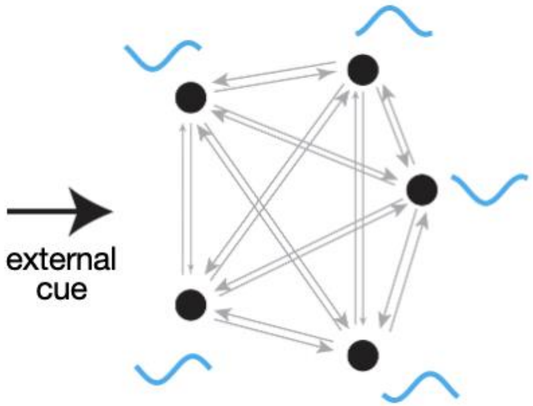
Initialize plastic connections

Simulate network dynamics

Update plastic weights

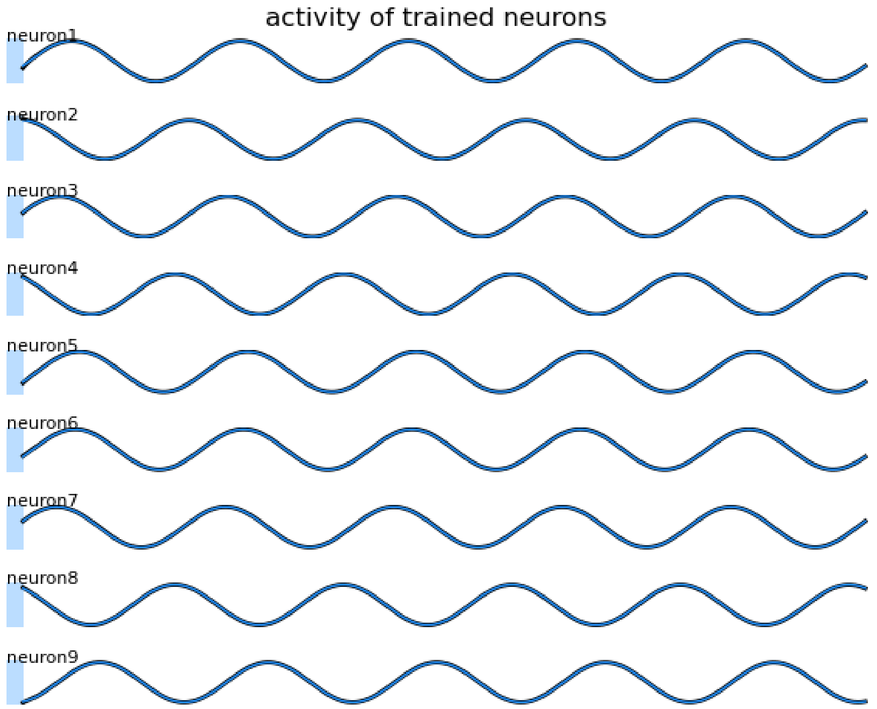
Assess learning error

assessing activity in the trained neurons



target activity pattern

network activity



Part 2. Spiking neural networks

Generic network

Initial network

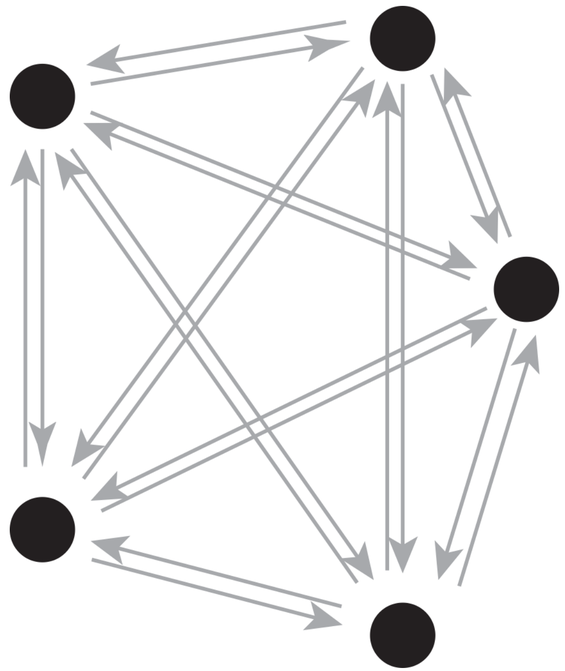
Define connectivity

Initialize weights

Connectivity: all-to-all

Initial weights: $w_{ij} \sim \mathcal{N}(0, \sigma^2)$

mean = 0, std = σ



Initial network

Define connectivity

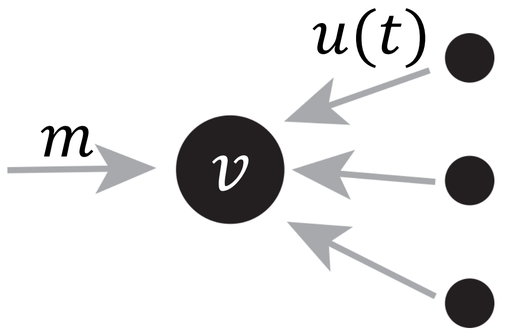
Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Leaky integrate-and-fire neuron



v: voltage at cell body

m: constant input

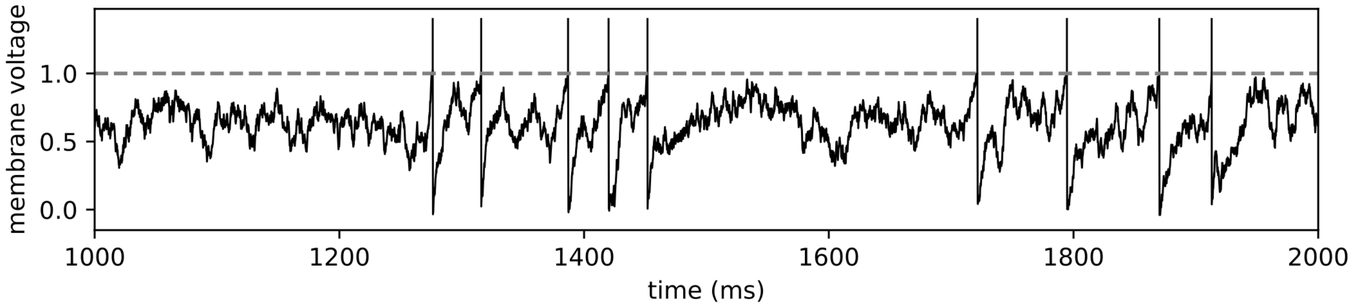
u: inputs from other neurons

$$\tau_m \frac{dv}{dt} = -v + m + u$$

spike threshold: If $v(t) = v_{thresh}$

spike: $v(t) = \delta(t)$

reset: $v(t) = v_{reset}$



Initial network

Define connectivity

Initialize weights

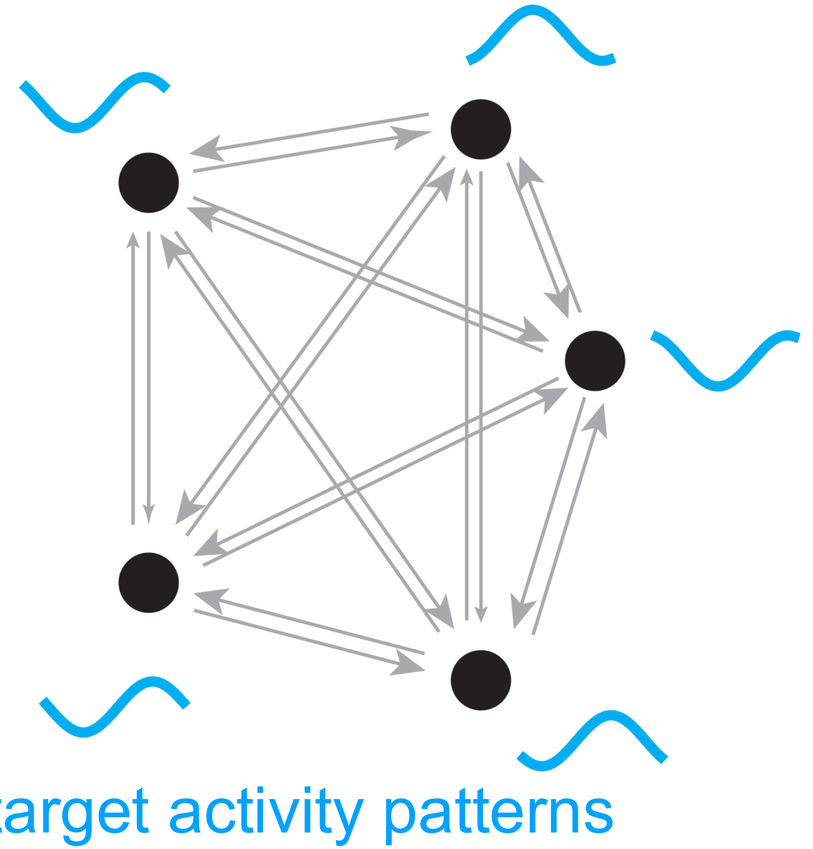
Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity



The synaptic inputs (u) to neurons
will be trained to follow the target patterns

Initial network

Define connectivity

Initialize weights

Single neuron model

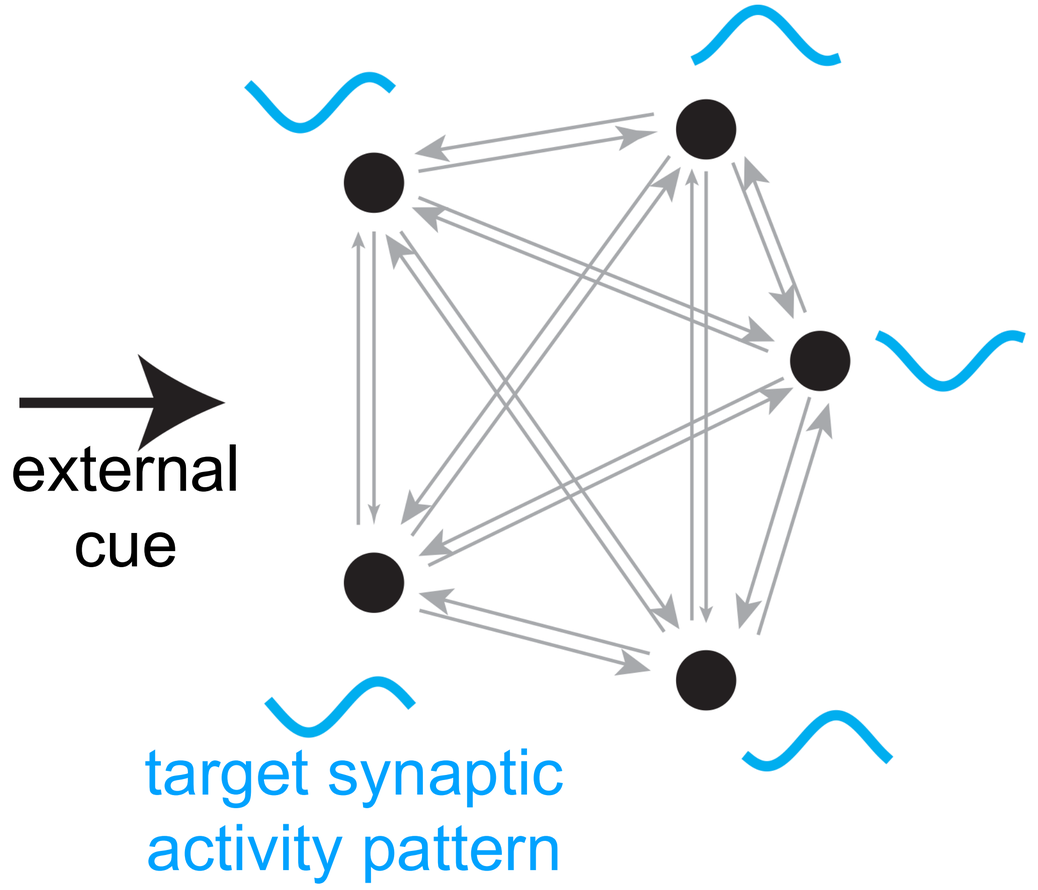
Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus



Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

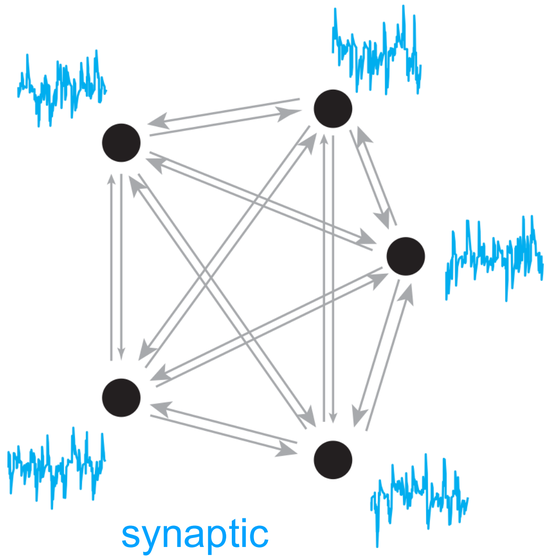
Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

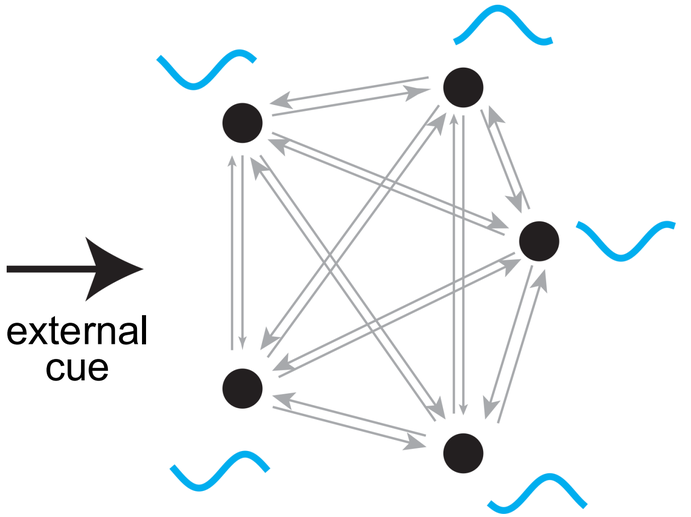
Before learning



synaptic activity (u)

After learning

external cue



target synaptic activity pattern

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

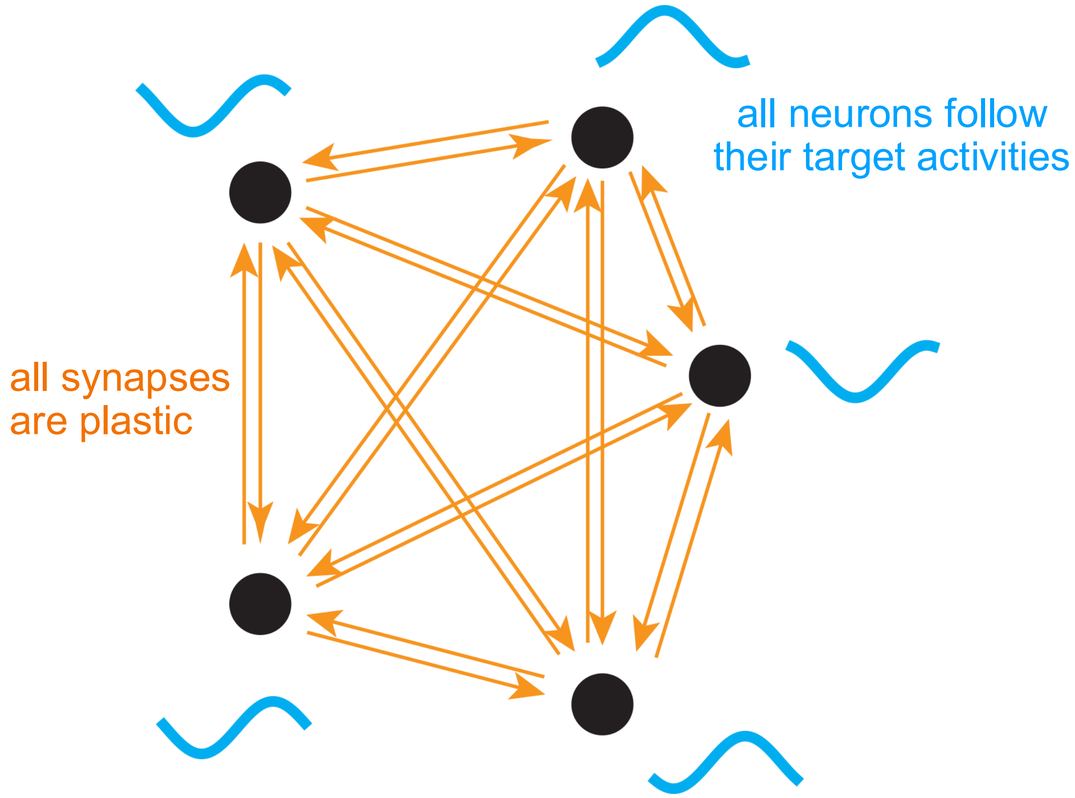
Train RNN activity

Select neurons to train

Initialize plastic connections

All the synapses are plastic

All the neurons are trained



Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

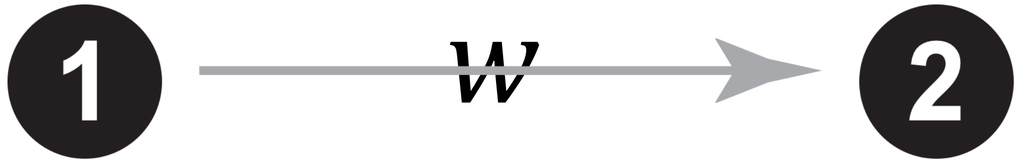
Select neurons to train

Initialize plastic connections

Spiking model

Simulate network dynamics

Propagating spikes to another neuron



$$\tau_m \frac{dv_1}{dt} = -v_1 + m_1$$

neuron 1 voltage

$$\tau_m \frac{dv_2}{dt} = -v_2 + m_2 + u_2$$

neuron 2 voltage

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

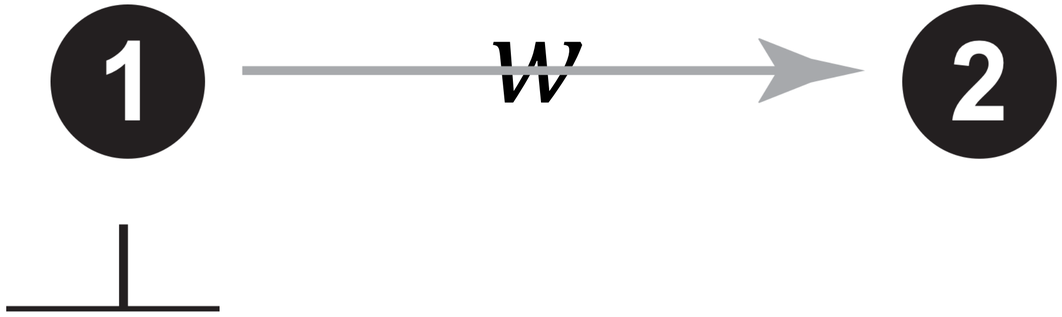
Select neurons to train

Initialize plastic connections

Spiking model

Simulate network dynamics

Propagating spikes to another neuron



$\delta_1(t)$

spike

cross spike threshold

$$\tau_m \frac{dv_1}{dt} = -v_1 + m_1$$

neuron 1 voltage

$$\tau_m \frac{dv_2}{dt} = -v_2 + m_2 + u_2$$

neuron 2 voltage

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

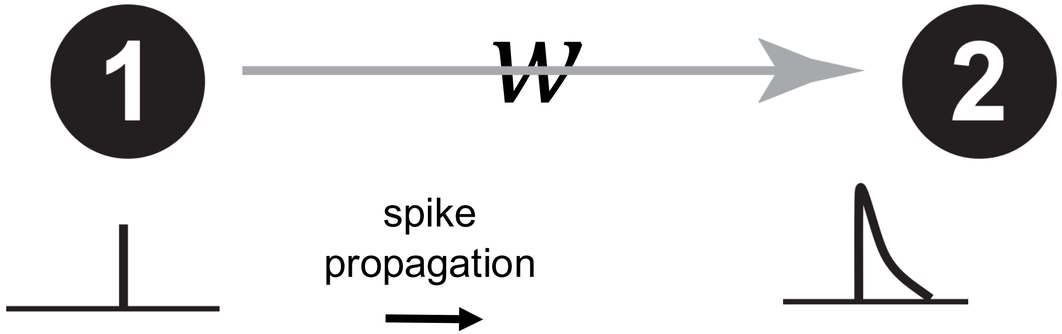
Select neurons to train

Initialize plastic connections

Spiking model

Simulate network dynamics

Propagating spikes to another neuron



$\delta_1(t)$

spike

$$\tau_s \frac{du_2}{dt} = -u_2 + w\delta_1(t)$$

synaptic filtering

cross spike threshold

$$\tau_m \frac{dv_1}{dt} = -v_1 + m_1$$

neuron 1 voltage

$$\tau_m \frac{dv_2}{dt} = -v_2 + m_2 + u_2$$

neuron 2 voltage

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

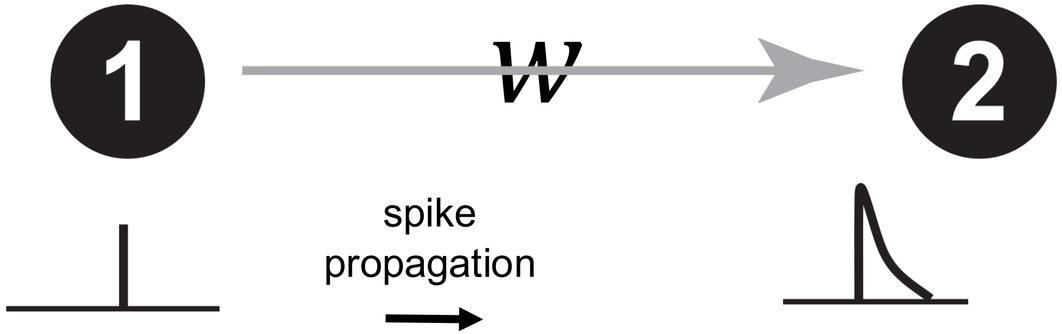
Select neurons to train

Initialize plastic connections

Spiking model

Simulate network dynamics

Propagating spikes to another neuron



$\delta_1(t)$

spike

$$\tau_s \frac{du_2}{dt} = -u_2 + w\delta_1(t)$$

synaptic filtering

cross spike threshold

integrate synaptic input

$$\tau_m \frac{dv_1}{dt} = -v_1 + m_1$$

neuron 1 voltage

$$\tau_m \frac{dv_2}{dt} = -v_2 + m_2 + u_2$$

neuron 2 voltage

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

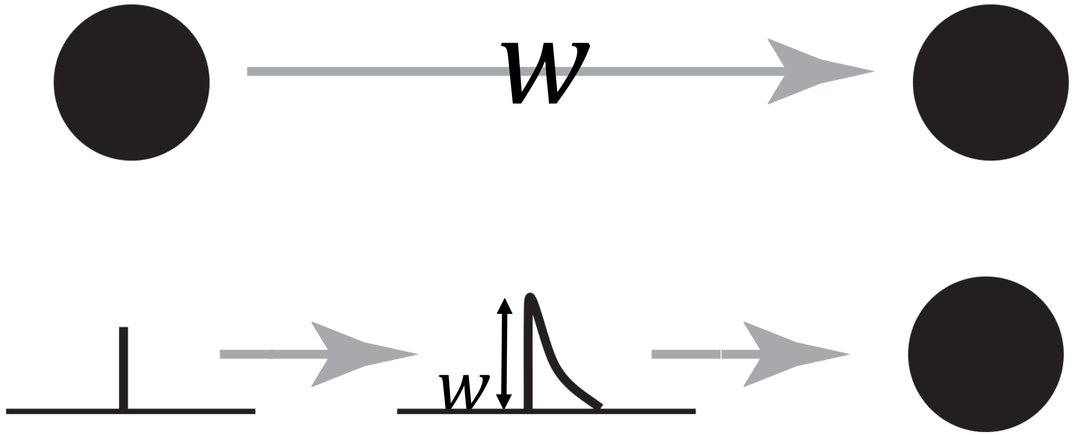
Select neurons to train

Initialize plastic connections

Spiking model

Simulate network dynamics

Propagating spikes to another neuron



$\delta(t)$

spike

$\tau_s \frac{du}{dt} = -u + w\delta(t)$

synaptic filtering

$\tau_m \frac{dv}{dt} = -v + m + u$

post-synaptic neuron voltage

Kim and Chow (2018)

Nicola and Clopath (2018)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

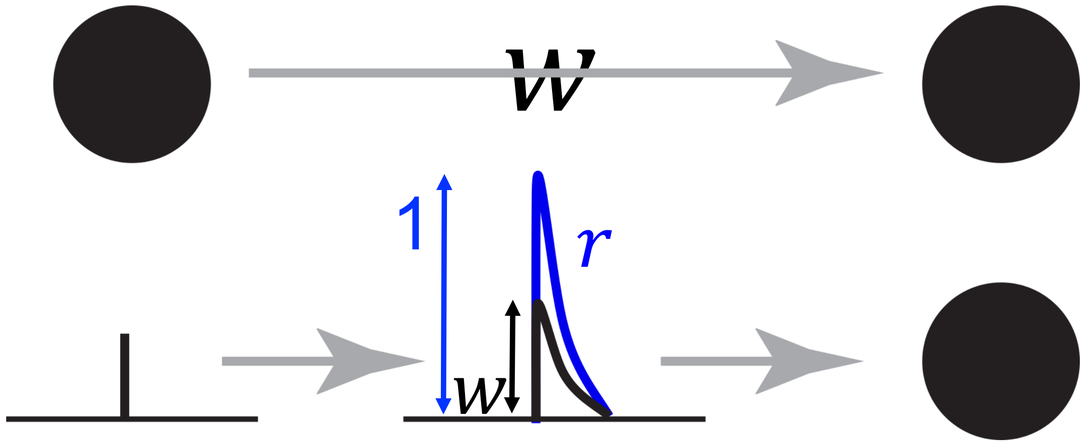
Select neurons to train

Initialize plastic connections

Spiking model

Simulate network dynamics

Propagating spikes to another neuron



$\delta(t)$

spike

$$\tau_s \frac{dr}{dt} = -r + \delta(t)$$

$$u = wr$$

This expression of u will be used for network training

$$\tau_m \frac{dv}{dt} = -v + m + u$$

post-synaptic neuron voltage

Kim and Chow (2018)

Nicola and Clopath (2018)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

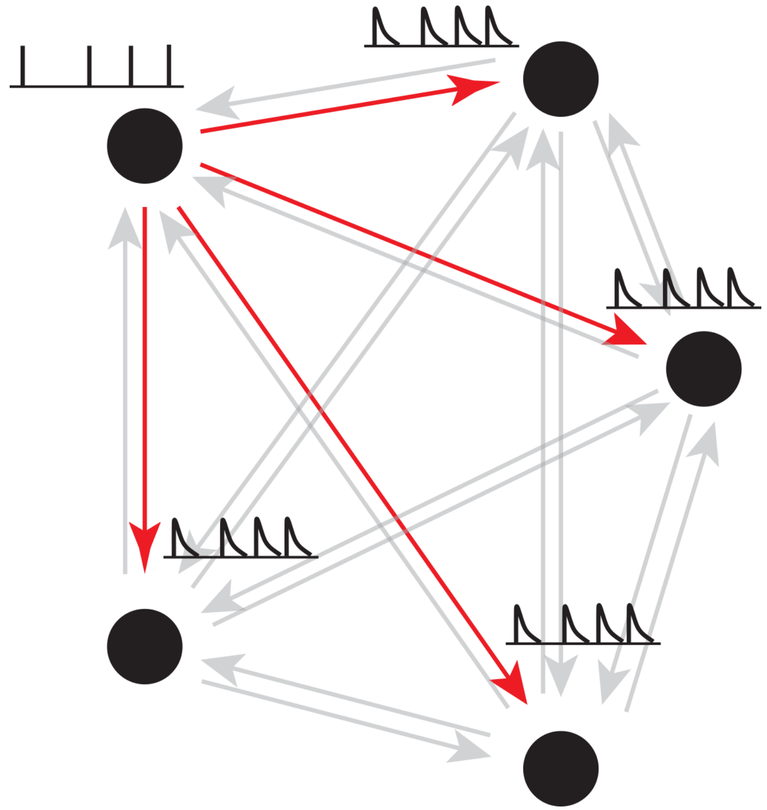
Select neurons to train

Initialize plastic connections

Spiking model

Simulate network dynamics

Propagation of spikes in a network



Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

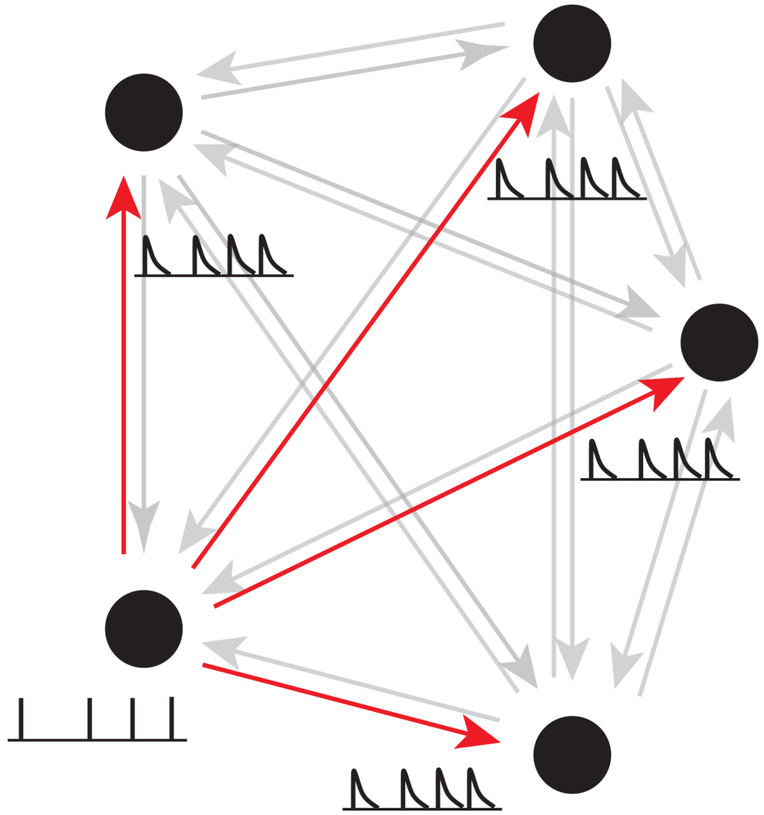
Select neurons to train

Initialize plastic connections

Spiking model

Simulate network dynamics

Propagation of spikes in a network



Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

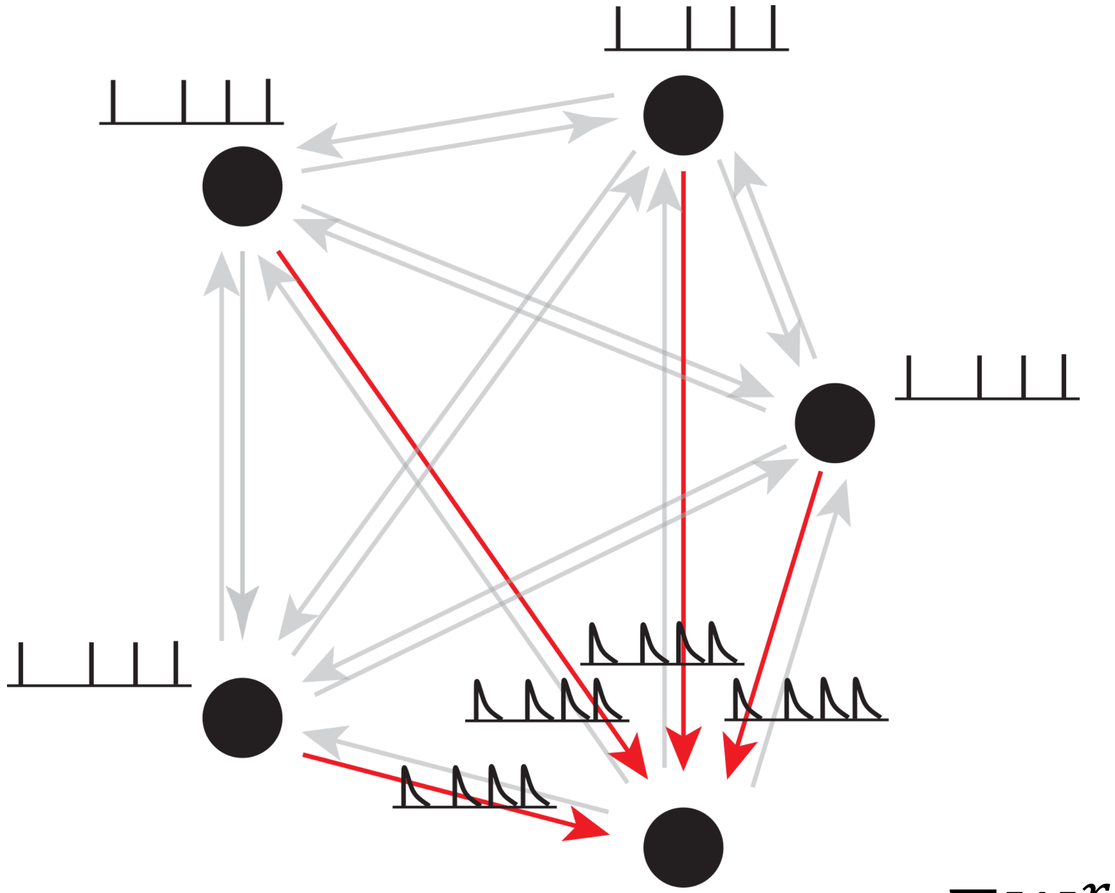
Select neurons to train

Initialize plastic connections

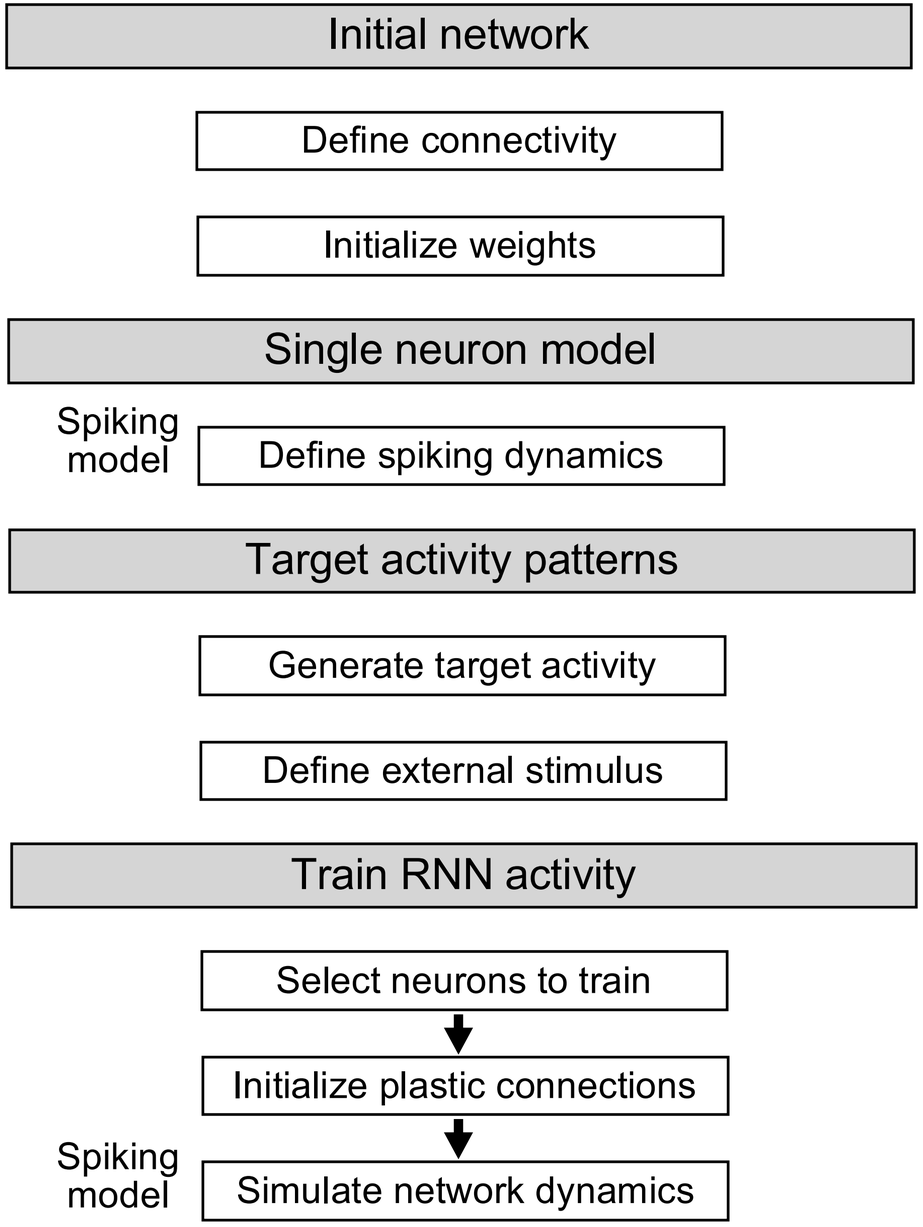
Spiking model

Simulate network dynamics

Propagation of spikes in a network



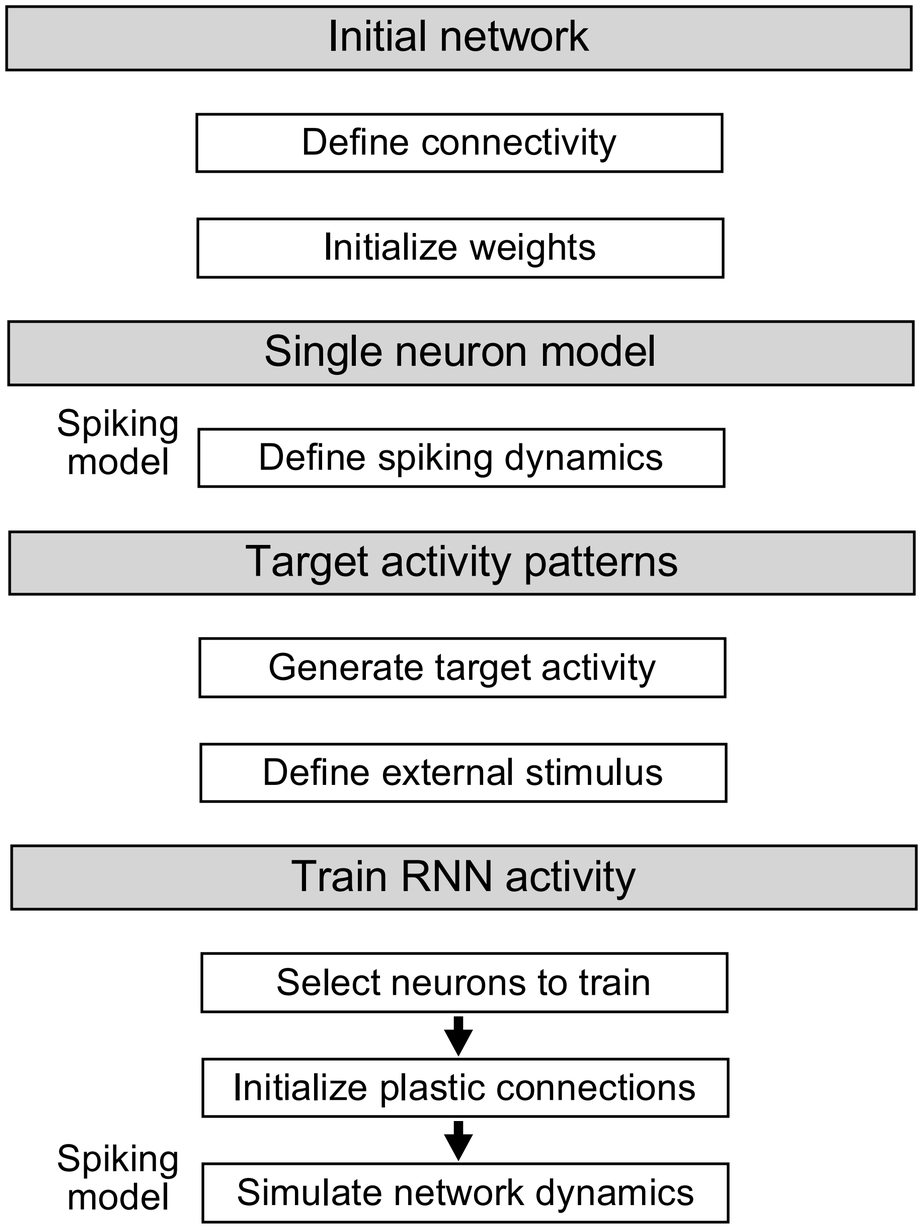
$$u_i = \sum_j W_{ij}^{rec} r_j$$



Dynamics of a network of N spiking neurons

$$\tau_m \dot{v}_i = -v_i + m_i + u_i$$

$$\tau_s \dot{u}_i = -u_i + \sum_j W_{ij}^{rec} \sum_{t_j^k < t} \delta(t - t_j^k)$$



Dynamics of a network of N spiking neurons

$$\tau_m \dot{v}_i = -v_i + m_i + u_i$$

$$\tau_s \dot{u}_i = -u_i + \sum_j W_{ij}^{rec} \sum_{t_j^k < t} \delta(t - t_j^k)$$

$$\tau_m \dot{v}_i = -v_i + m_i + u_i$$

$$\tau_s \dot{r}_j = -r_j + \sum_{t_j^k < t} \delta(t - t_j^k)$$

$$u_i = \sum_j W_{ij}^{rec} r_j$$

Kim and Chow (2018)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target rates

Define external stimulus

Train RNN activity

Spiking model

Select neurons to train

Initialize plastic connections

Simulate network dynamics

Update plastic weights

Minimize the following cost function by optimizing $\mathbf{w}$:

$$C = \frac{1}{2}\sum_{t=1}^{T}[u_t - f_t]^2 + \frac{\lambda}{2}\parallel \mathbf{w} \parallel^2$$

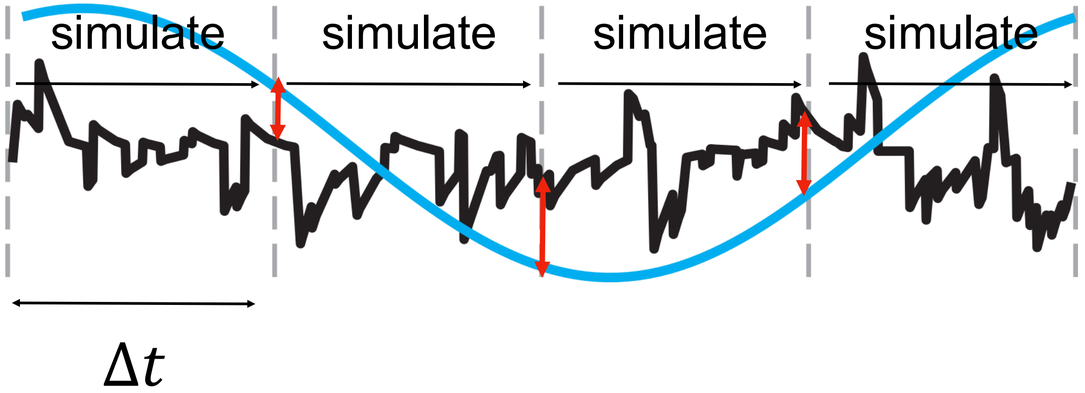
synaptic drive *u* should follow the target pattern *f*

Express u in terms of $\mathbf{w}$ and $\mathbf{r}$.

$$C = \frac{1}{2}\sum_{t=1}^{T}[\mathbf{w}\cdot\mathbf{r}_t - f_t]^2 + \frac{\lambda}{2}\parallel \mathbf{w} \parallel^2$$

Derive RLS (recursive least squares) algorithm that optimizes $\mathbf{w}$ every Δt (=20ms) to reduce C.

update $\mathbf{w}_{t-1}$ update $\mathbf{w}_t$ update $\mathbf{w}_{t+1}$



Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

Spiking model

Select neurons to train

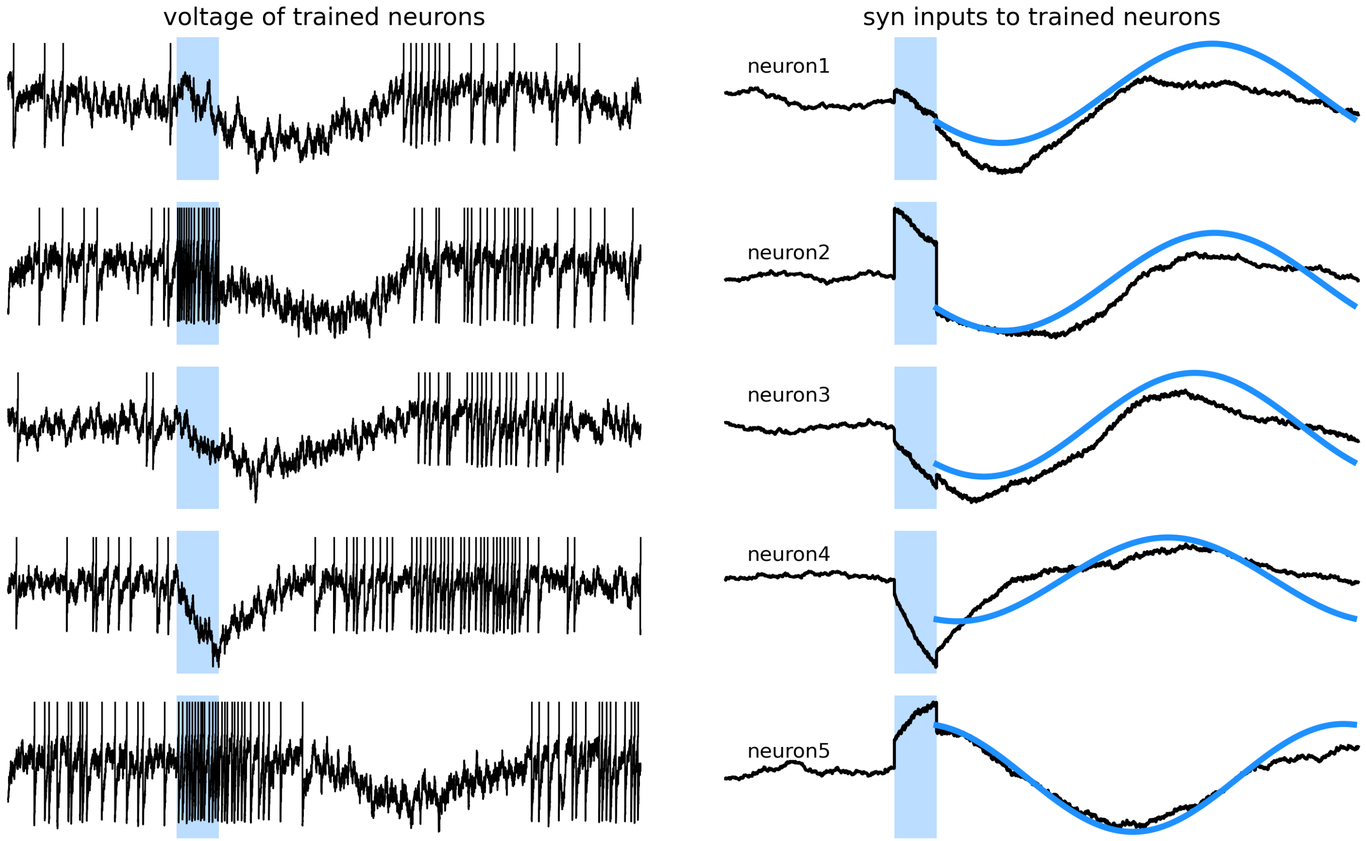
Initialize plastic connections

Simulate network dynamics

Update plastic weights

Assess learning error

Activity of trained neurons



Part 3. Spiking neural networks

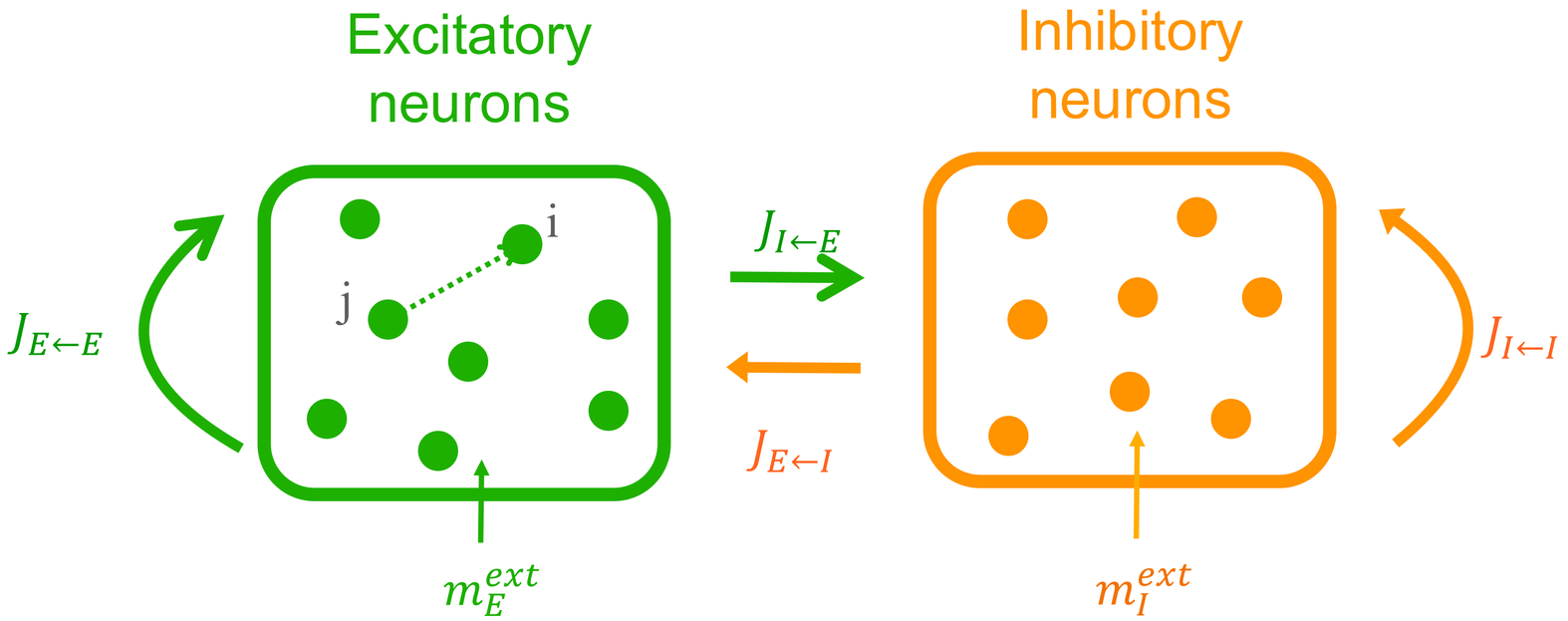
Exc-Inh balanced network

Initial network

Define connectivity

Initialize weights

Exc-Inh network with strong synaptic coupling



Num of neurons $N \approx 10^3$

Num of connections per neuron $\mathbf{K} \approx 10^2$

Sparse random connections $Pr(J_{\alpha\beta}^{ij} \neq 0) = K/N \ll 1$

Strong weights $J_{\alpha\beta}^{ij} = \frac{\bar{J}_{\alpha\beta}}{\sqrt{\mathbf{K}}}$

Strong inputs $m_\alpha^{ext} = \bar{\mu}_\alpha \sqrt{\mathbf{K}}$

Brunel (2000)

Van Vreeswijk and Sompolinsky (1996)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

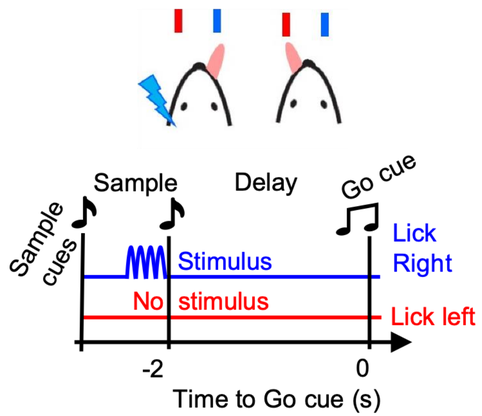
Define spiking dynamics

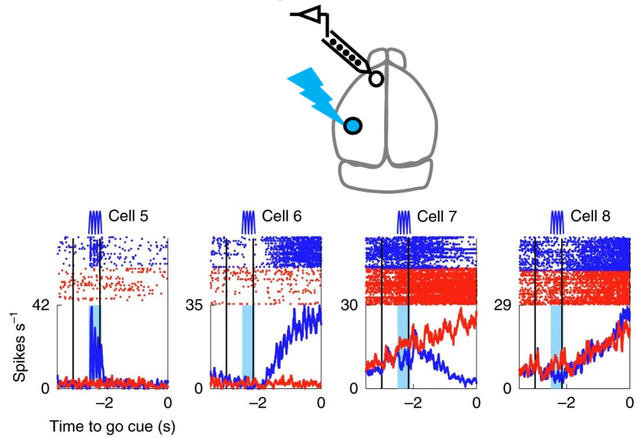
Target activity patterns

Generate target activity

Experimental setup

Finkelstein, Fontolan et al (2021)





Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

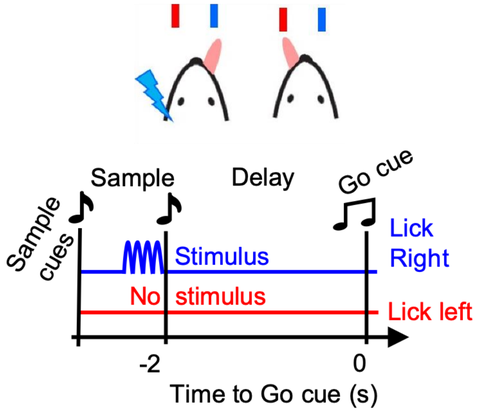
Define spiking dynamics

Target activity patterns

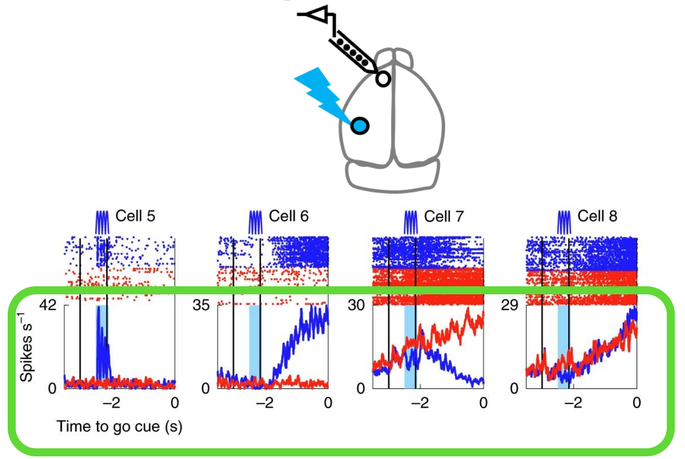
Generate target activity

Experimental setup

Finkelstein, Fontolan et al (2021)



Finkelstein, Fontolan et al (2021)



Target rate patterns (r):
trial-averaged spike rates of motor neurons

Use the activation function of LIF neuron

$$u = f^{-1}(r)$$

Target synaptic activity patterns (u)

Kim, Finkelstein et al (2023)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

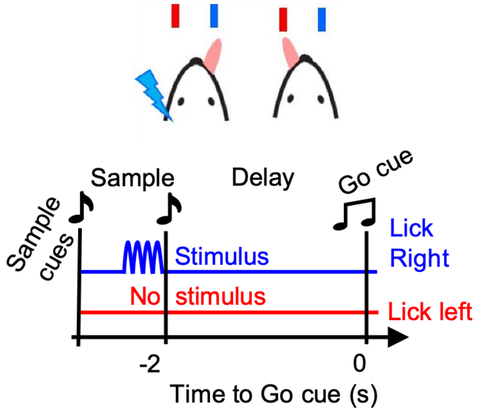
Generate target activity

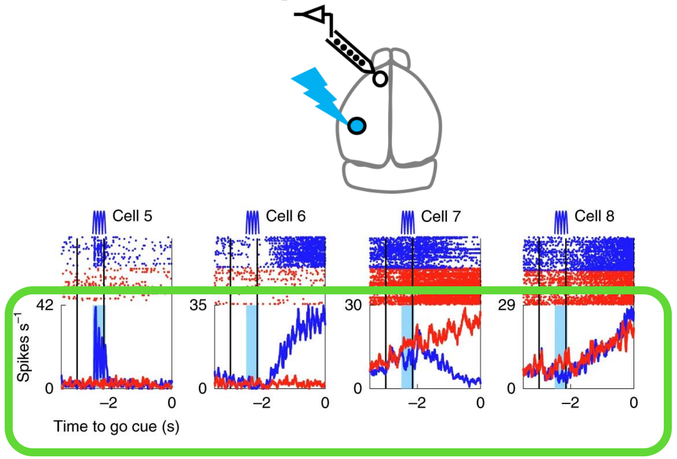
Define external stimulus

Train RNN activity

Experimental setup

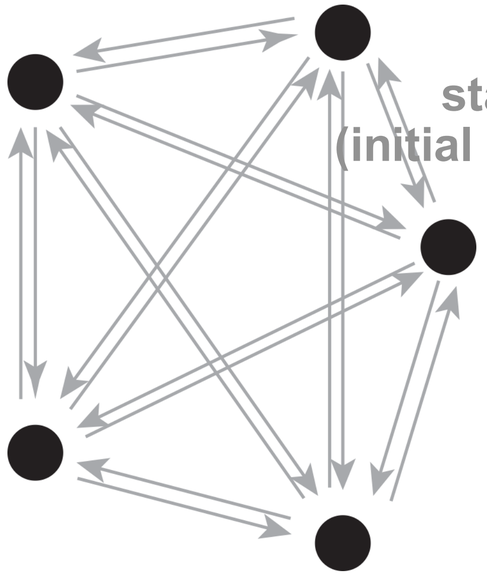
Finkelstein, Fontolan et al (2021)





Finkelstein, Fontolan et al (2021)

Set up the (static) initial balanced network



static synapses (initial balanced network)

Kim, Finkelstein et al (2023)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

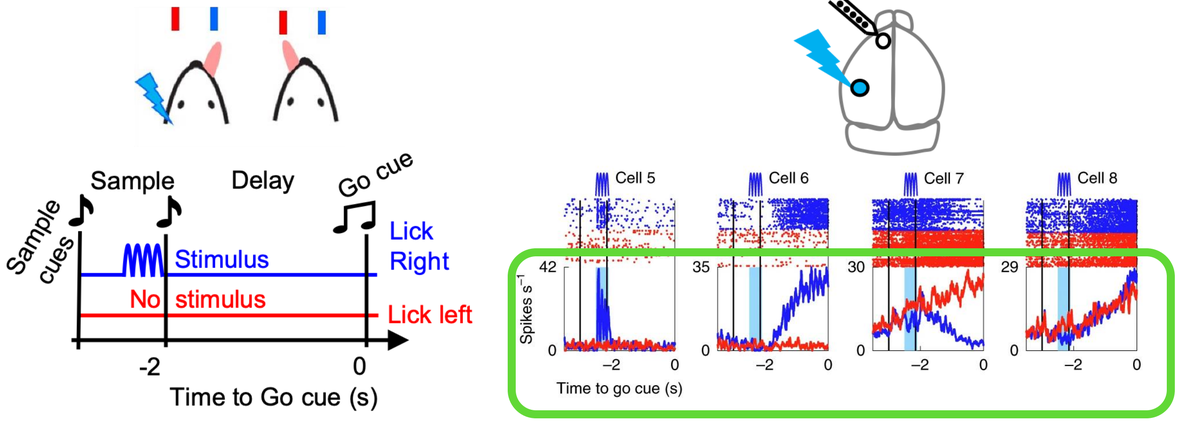
Define external stimulus

Train RNN activity

Select neurons to train

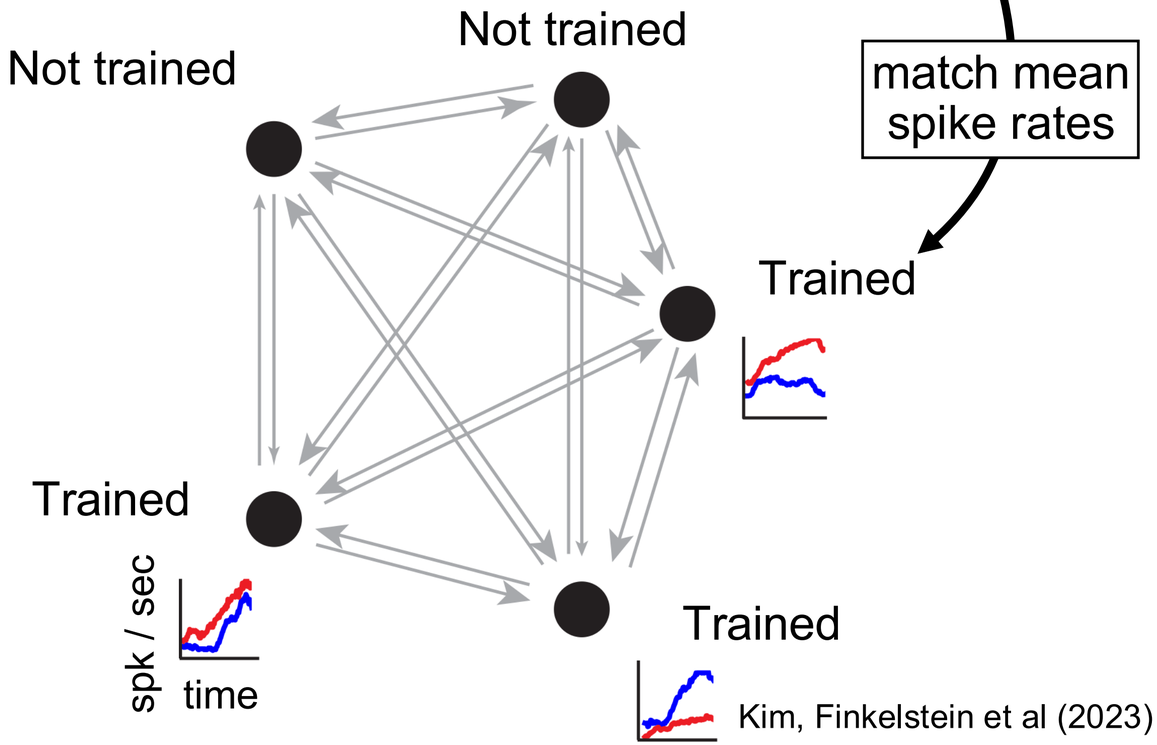
Experimental setup

Finkelstein, Fontolan et al (2021)



Select a subset of neurons to train

match mean spike rates



Kim, Finkelstein et al (2023)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

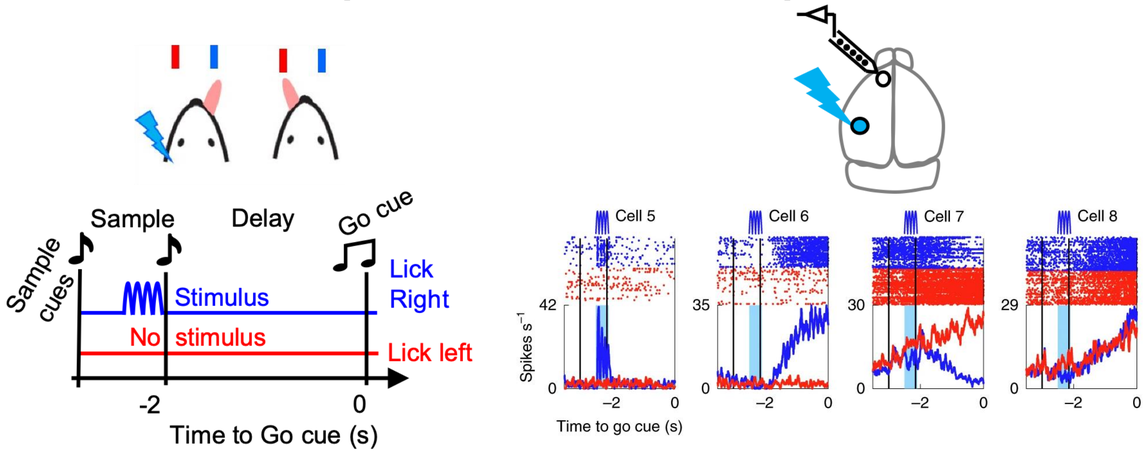
Train RNN activity

Select neurons to train

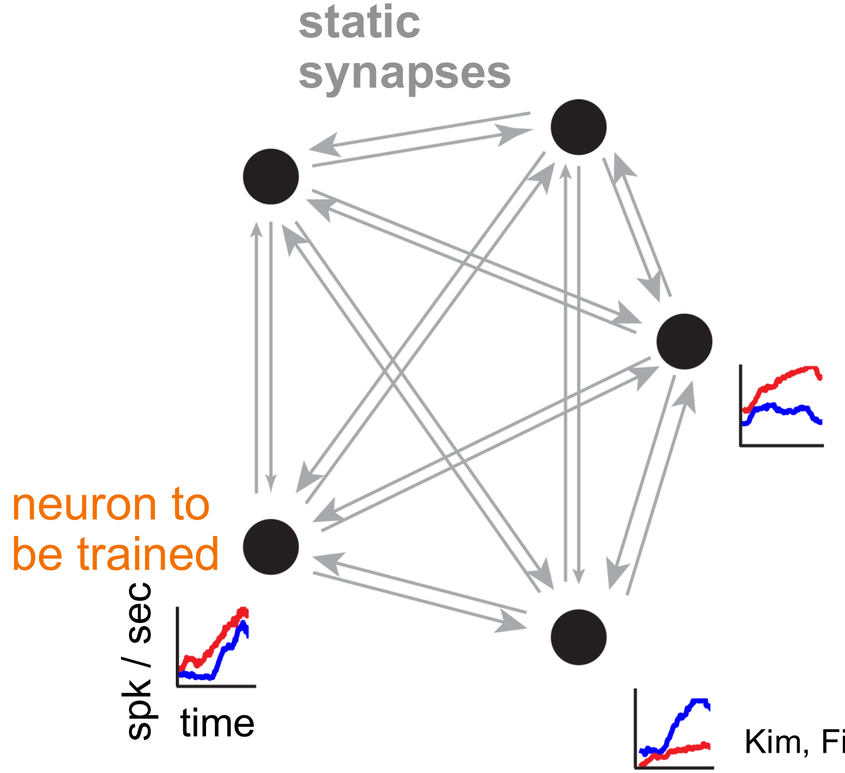
Initialize plastic connections

Experimental setup

Finkelstein, Fontolan et al (2021)



Plastic synapses are sparse



Kim, Finkelstein et al (2023)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

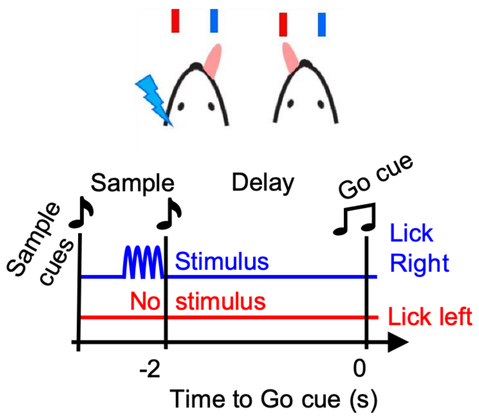
Train RNN activity

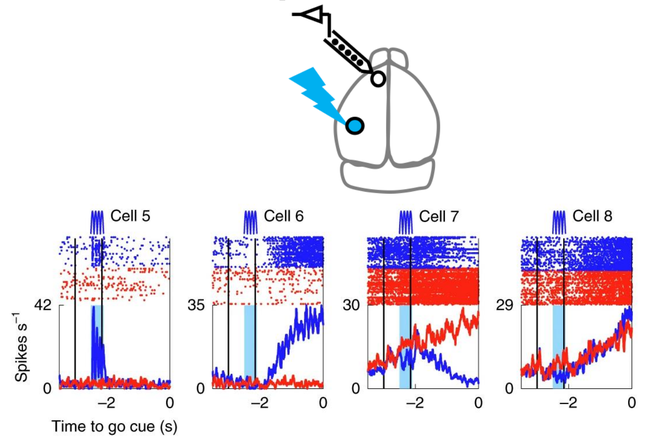
Select neurons to train

Initialize plastic connections

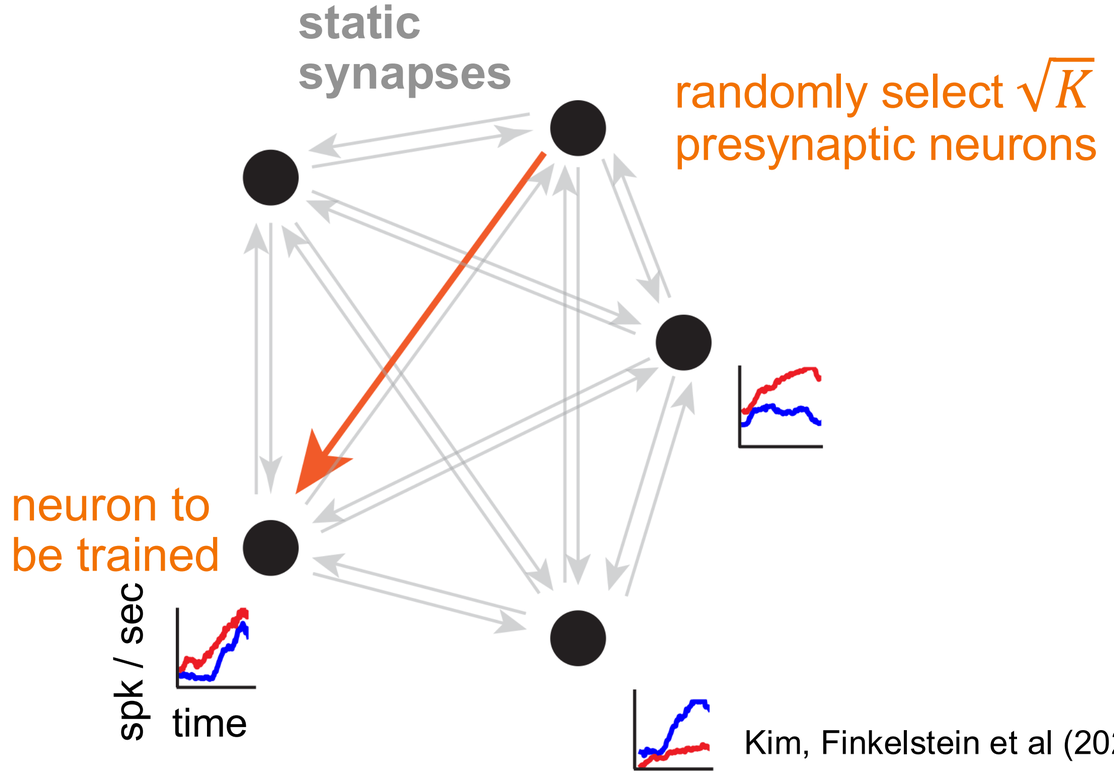
Experimental setup

Finkelstein, Fontolan et al (2021)





Plastic synapses are sparse



Kim, Finkelstein et al (2023)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

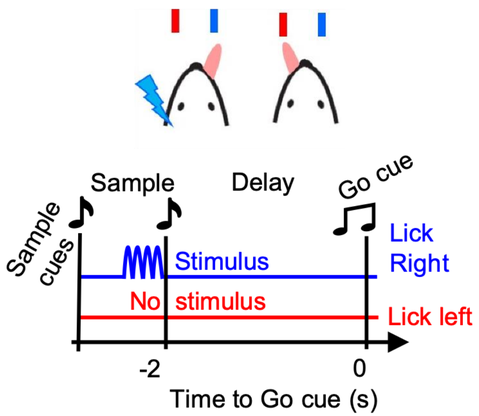
Train RNN activity

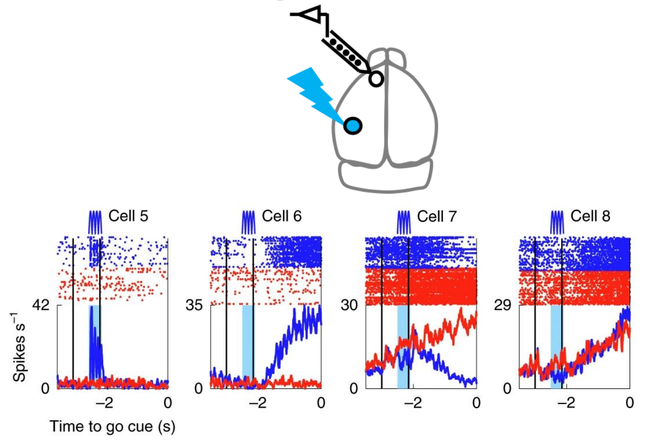
Select neurons to train

Initialize plastic connections

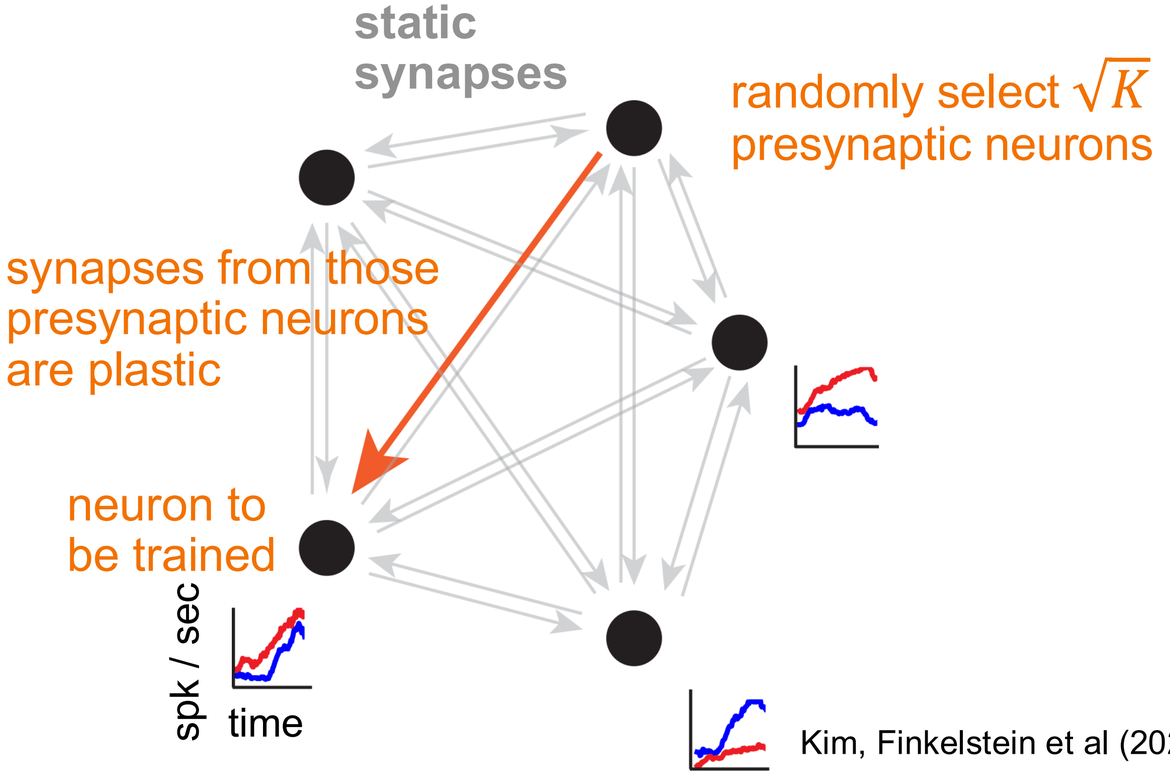
Experimental setup

Finkelstein, Fontolan et al (2021)





Plastic synapses are sparse



Kim, Finkelstein et al (2023)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

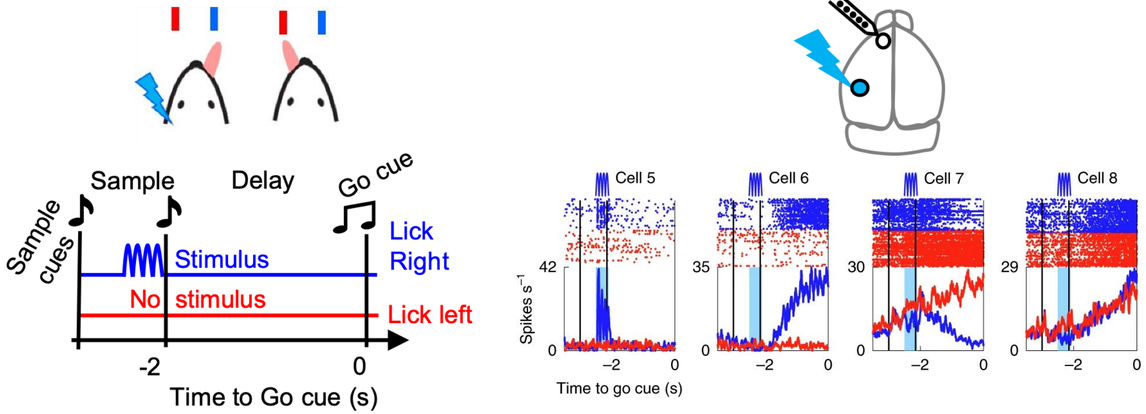
Train RNN activity

Select neurons to train

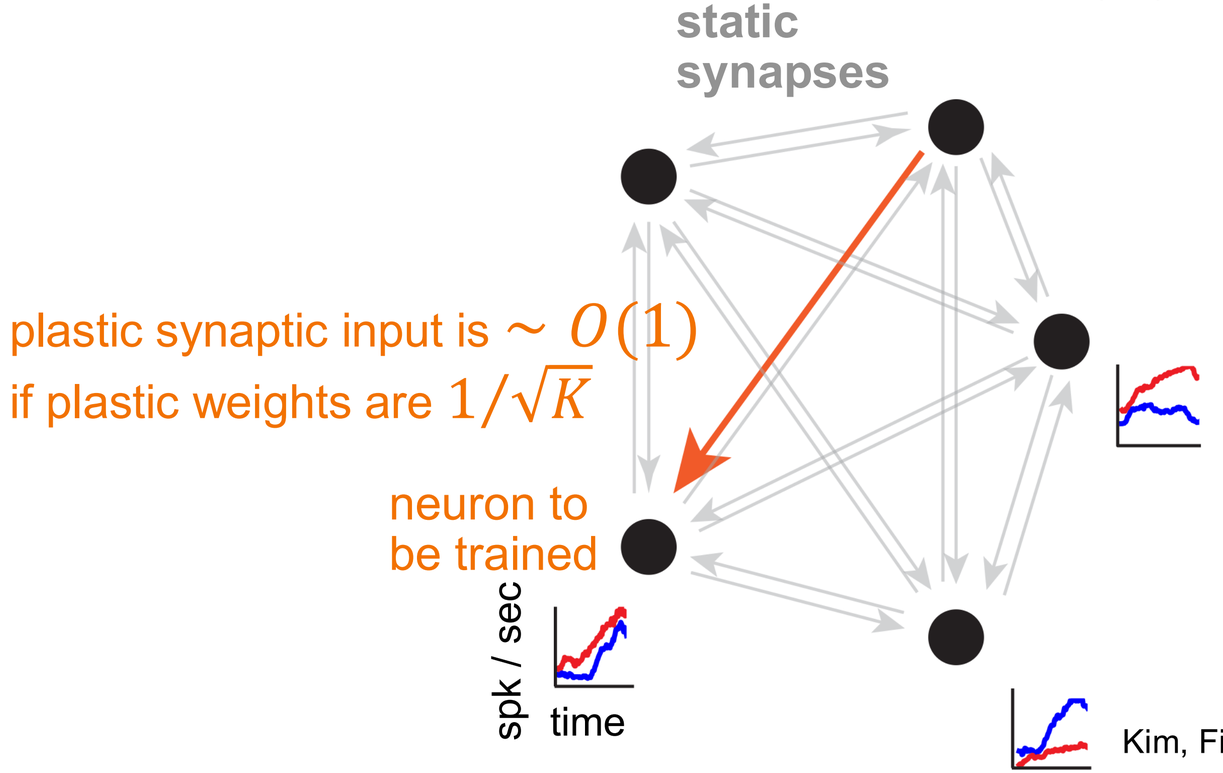
Initialize plastic connections

Experimental setup

Finkelstein, Fontolan et al (2021)



Plastic inputs are $\sim O(1)$



Kim, Finkelstein et al (2023)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

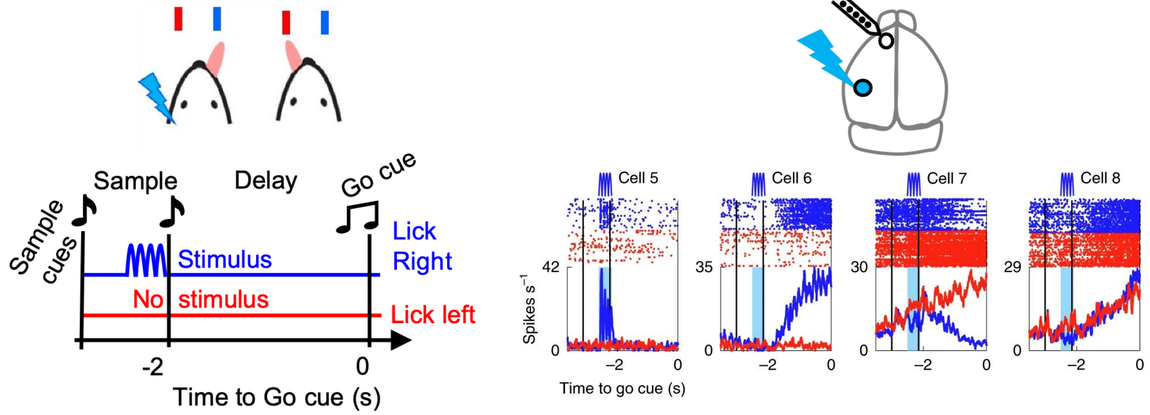
Train RNN activity

Select neurons to train

Initialize plastic connections

Experimental setup

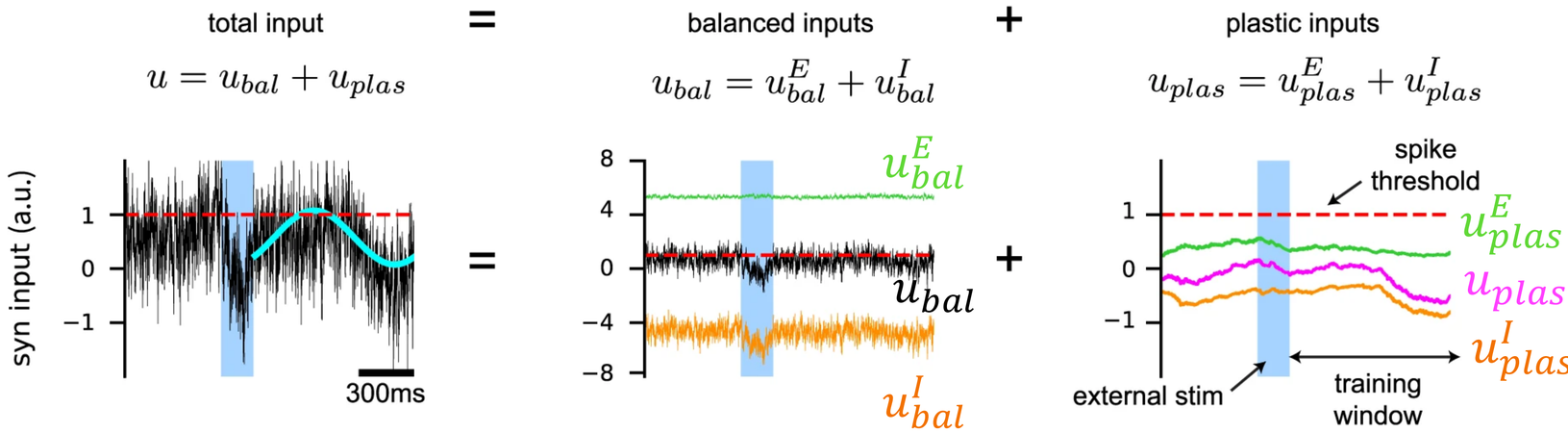
Finkelstein, Fontolan et al (2021)



Plastic inputs are $\sim O(1)$

B

Synaptic inputs to a trained neuron



Kim, Finkelstein et al (2023)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

Spiking model

Select neurons to train

Initialize plastic connections

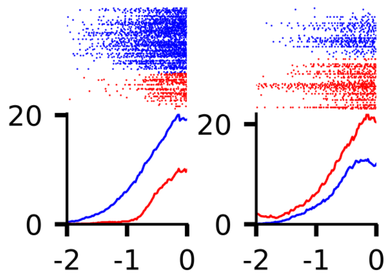
Simulate network dynamics

Update plastic weights

Assess learning error

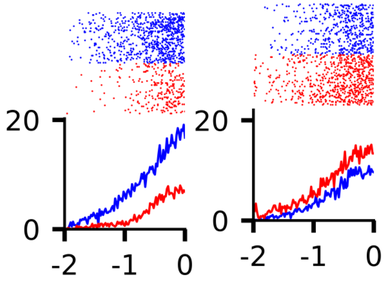
Target activity patterns

Motor neurons
(Pyramidal cells)

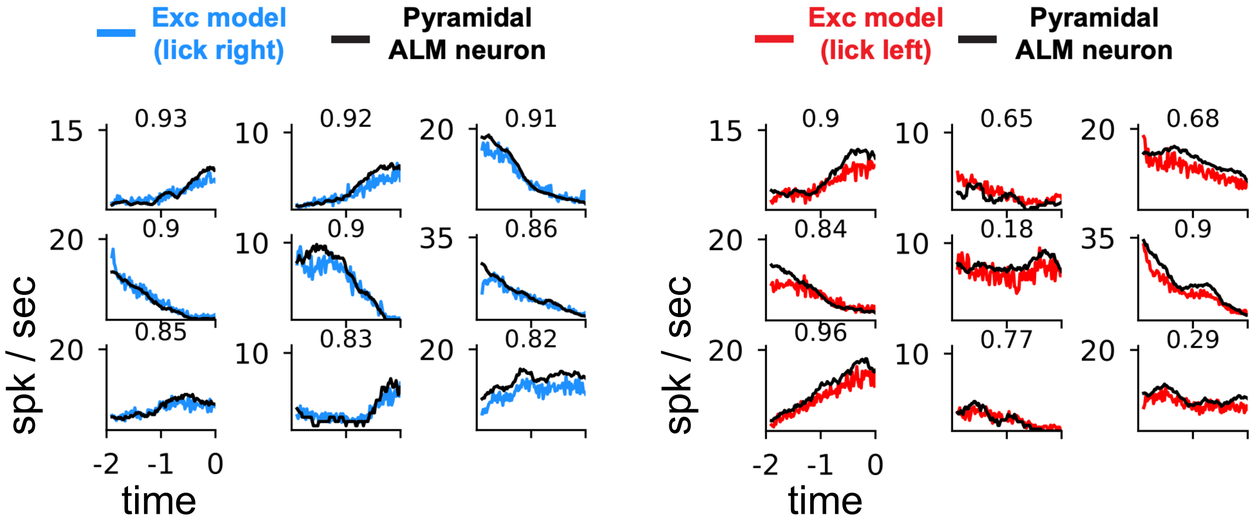


Trained activity patterns

Spiking neuron models
(excitatory)



More examples of trained neurons



Kim, Finkelstein et al (2023)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

Spiking model

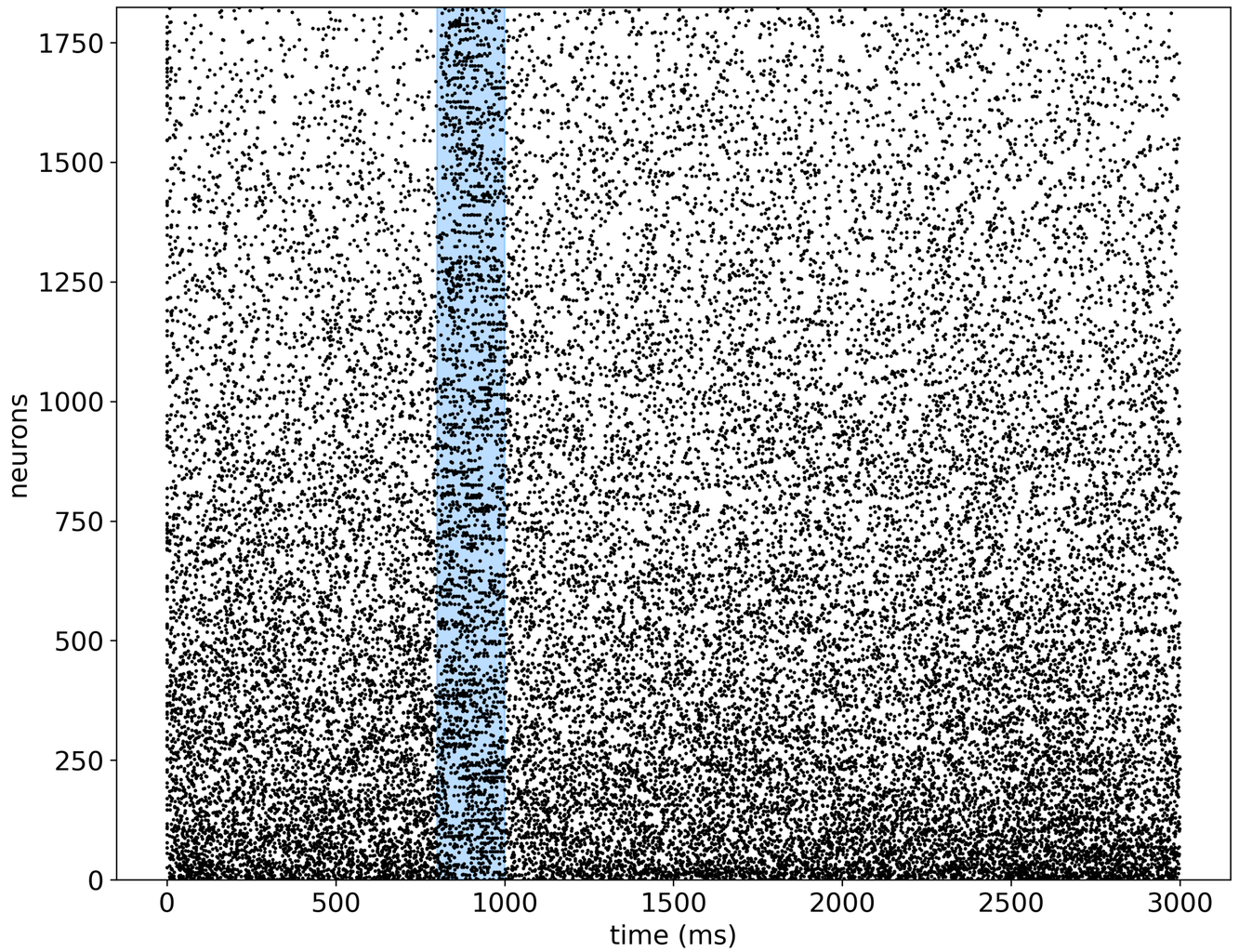
Select neurons to train

Initialize plastic connections

Simulate network dynamics

Update plastic weights

Assess learning error



Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

Spiking model

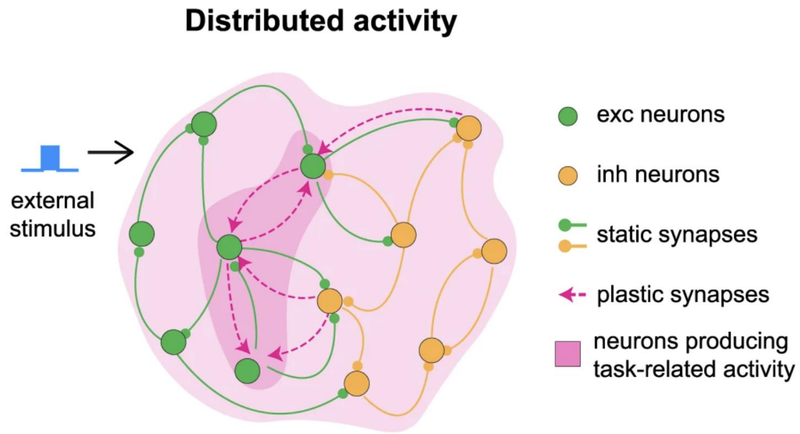
Select neurons to train

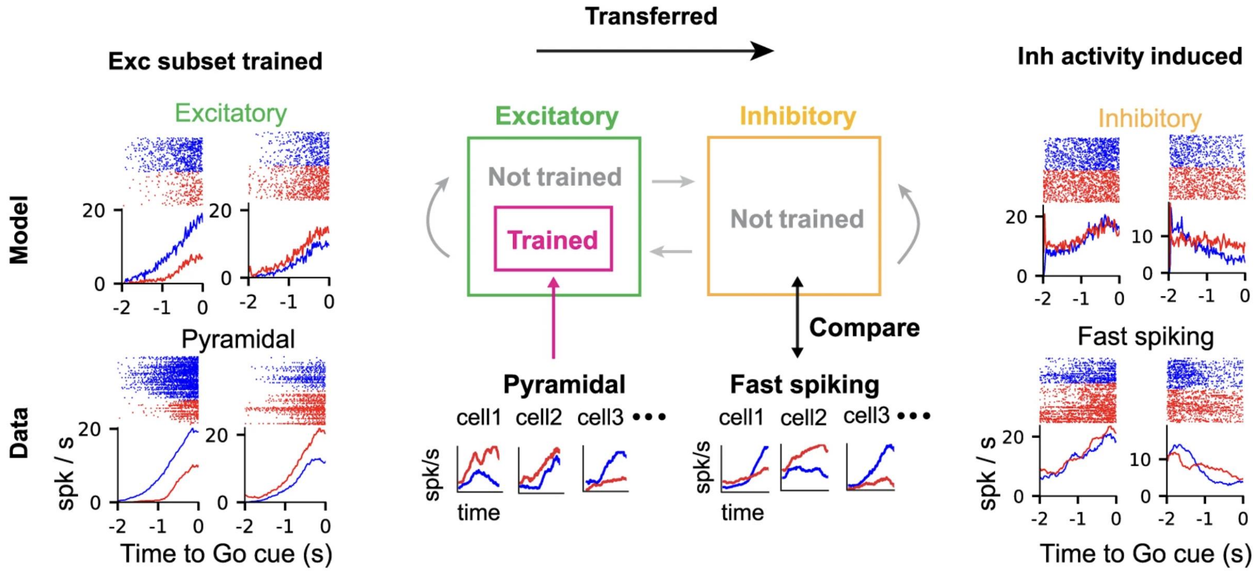
Initialize plastic connections

Simulate network dynamics

Update plastic weights

Assess learning error





When neural activity is perturbed through learning, the trained activity propagates through the balanced network, producing patterns of activity in untrained neurons.

Kim, Finkelstein et al (2023)

Initial network

Define connectivity

Initialize weights

Single neuron model

Spiking model

Define spiking dynamics

Target activity patterns

Generate target activity

Define external stimulus

Train RNN activity

Spiking model

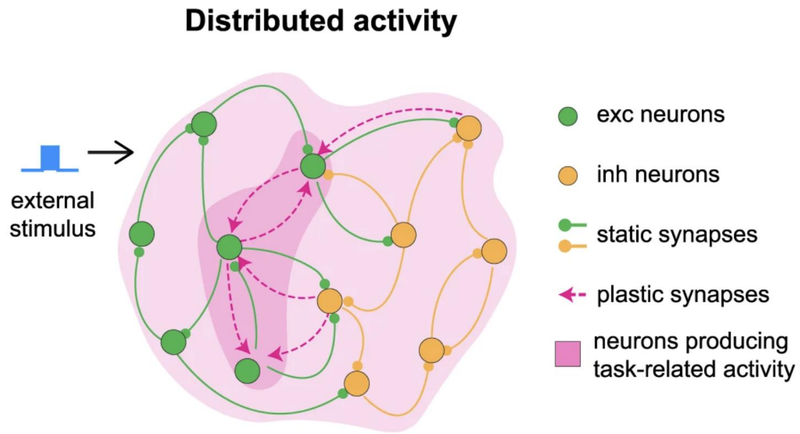
Select neurons to train

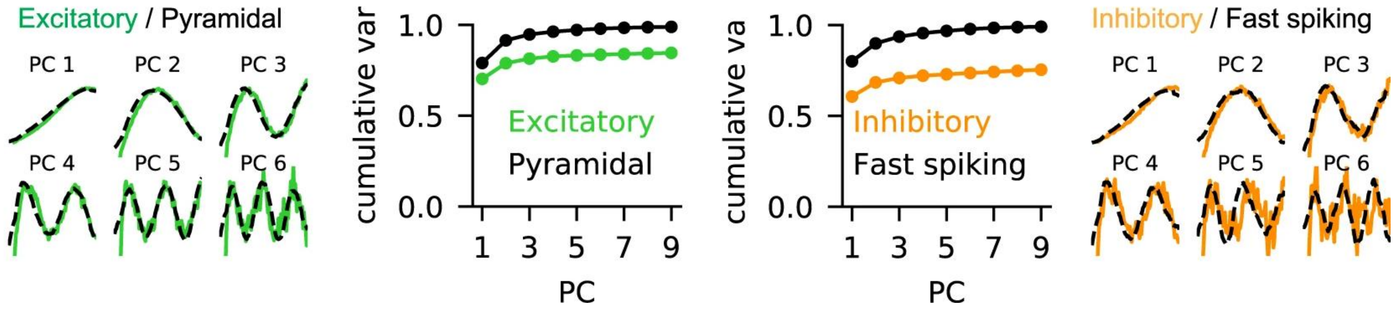
Initialize plastic connections

Simulate network dynamics

Update plastic weights

Assess learning error





This propagation of activity is seen at the population level for the trained and untrained populations.

The activity of the untrained inhibitory neurons in the model match recorded neural activity for fast spiking neurons in ALM, highlighting the role of E/I balance in spreading learned activity.

Kim, Finkelstein et al (2023)

References

Sompolinsky, H., Crisanti, A., & Sommers, H. J. (1988). Chaos in random neural networks. *Physical review letters*, *61*(3), 259.

Sussillo, D., & Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, *63*(4), 544-557.

LaFosse, P. K., Zhou, Z., O'Rawe, J. F., Friedman, N. G., Scott, V. M., Deng, Y., & Histed, M. H. (2024). Cellular-resolution optogenetics reveals attenuation-by-suppression in visual cortical neurons. *Proceedings of the National Academy of Sciences*, *121*(45), e2318837121.

Sanzeni, A., Histed, M. H., & Brunel, N. (2020). Response nonlinearities in networks of spiking neurons. *PLoS computational biology*, *16*(9), e1008165.

Nicola, W., & Clopath, C. (2017). Supervised learning in spiking neural networks with FORCE training. *Nature communications*, *8*(1), 2208.

Kim, C. M., & Chow, C. C. (2018). Learning recurrent dynamics in spiking networks. *Elife*, *7*, e37124.

Kim, C. M., Finkelstein, A., Chow, C. C., Svoboda, K., & Darshan, R. (2023). Distributing task-related neural activity across a cortical network through task-independent connections. *Nature Communications*, *14*(1), 2851.

Finkelstein, A., Fontolan, L., Economo, M. N., Li, N., Romani, S., & Svoboda, K. (2021). Attractor dynamics gate cortical information flow during decision-making. *Nature neuroscience*, *24*(6), 843-850.

Van Vreeswijk, C., & Sompolinsky, H. (1996). Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, *274*(5293), 1724-1726.

Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of computational neuroscience*, *8*, 183-208.