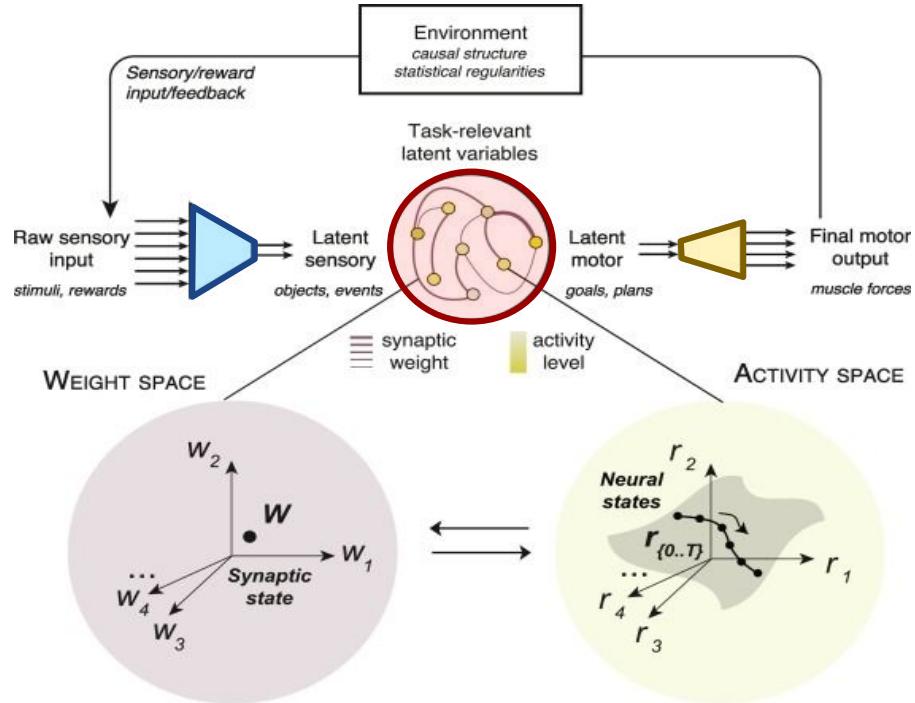
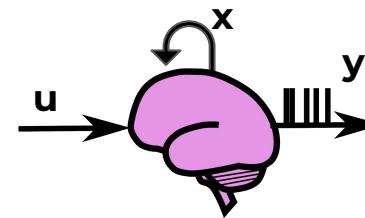


Recurrent Neural Networks:

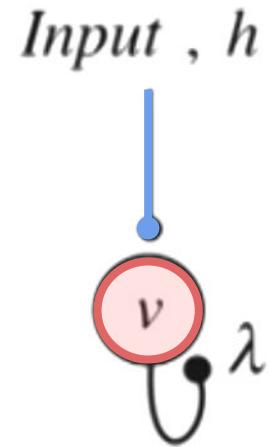
A model organism
for studying
computation
over time
in neuroscience



Simplest recurrent network: a neuron with an self-synapse

$$\tau_n \frac{dv}{dt} = -v + h + \lambda v$$

$$\tau_n \frac{dv}{dt} = -\underbrace{(1-\lambda)v}_{\text{red bracket}} + h \quad \underbrace{\lambda v}_{\text{blue bracket}}$$



- We examined three cases:

$$\lambda < 1$$

$$\lambda = 1$$

$$\lambda > 1$$

RNNs as a description of firing rate dynamics in a network of neurons

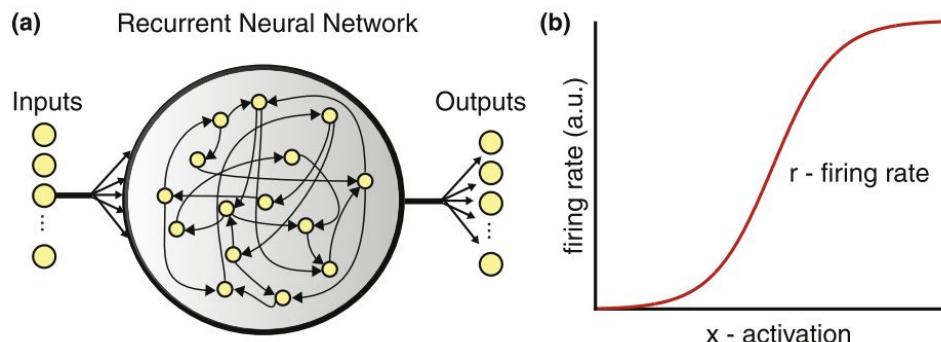
$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$$
$$\tau \dot{\mathbf{x}}(t) = -\mathbf{x}(t) + \mathbf{Jr}(t) + \mathbf{Bu}(t) + b$$

*describes the
network
connections*

The diagram illustrates the mathematical connection between two equations describing the dynamics of a recurrent neural network. The top equation, $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$, represents the state-space dynamics. The bottom equation, $\tau \dot{\mathbf{x}}(t) = -\mathbf{x}(t) + \mathbf{Jr}(t) + \mathbf{Bu}(t) + b$, represents the firing rate dynamics. Red curved arrows highlight the correspondence between the \mathbf{Ax} term in the top equation and the $-\mathbf{x}(t)$ term in the bottom equation, and between the \mathbf{Bu} term in the top equation and the $\mathbf{Bu}(t)$ term in the bottom equation. A purple arrow points upwards from the bottom equation towards the text "describes the network connections", which is centered between the two equations and describes the meaning of the matrix \mathbf{J} .

RNNs as a description of firing rate dynamics in a network of neurons

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$$
$$\tau \dot{\mathbf{x}}(t) = -\mathbf{x}(t) + \mathbf{Jr}(t) + \mathbf{Bu}(t) + b$$



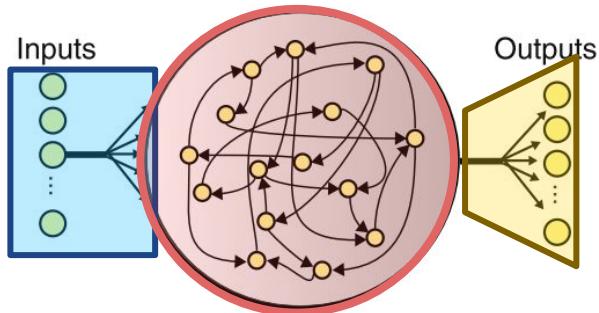
RNNs as a description of firing rate dynamics in a network of neurons

$$\dot{\mathbf{x}} = \underbrace{\mathbf{Ax}}_{\text{dynamics}} + \underbrace{\mathbf{Bu}}_{\text{inputs}}$$

$$\mathbf{y} = \underbrace{\mathbf{Cx}}_{\text{outputs}}$$

$$\tau \dot{\mathbf{x}}(t) = -\mathbf{x}(t) + \mathbf{Jr}(t) + \mathbf{Bu}(t) + b$$
$$\mathbf{z}(t) = \underbrace{\mathbf{Wr}(t)}_{\text{outputs}}$$

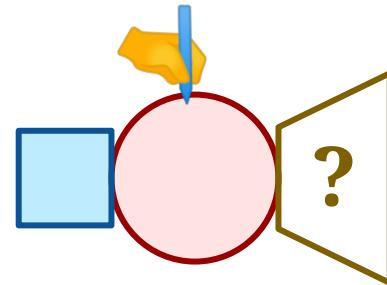
(a) Recurrent Neural Network



For much of the history of neuroscience..

Network models have been
constructed “**by hand**

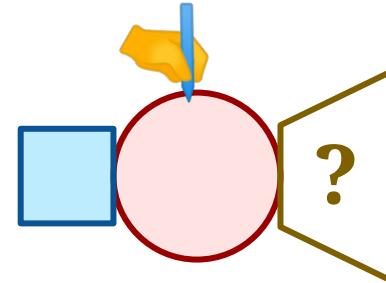
Often with parameters drawn from
distributions



For much of the history of neuroscience..

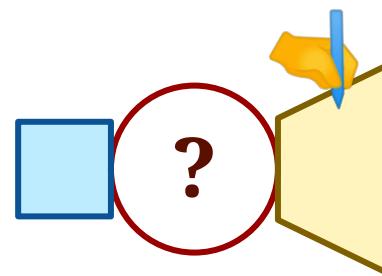
Network models have been
constructed “**by hand**” [\[Wang2018\]](#)

Often with parameters drawn from
distributions



Key Twist: Recent work has focused on optimizing
RNNs (not for structural realism) but instead to
accomplish some task

- e.g. integration of inputs
- Often some degree of “hand-tuning” & hyper parameter adjustment happens after



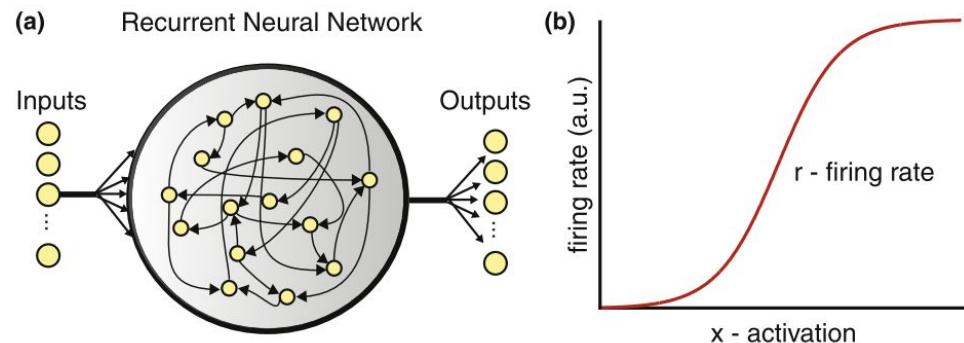
Why Study RNNs?

Have brain-like features

- Feedback
- Nonlinearity
- Distributed computing

Can perform complex tasks

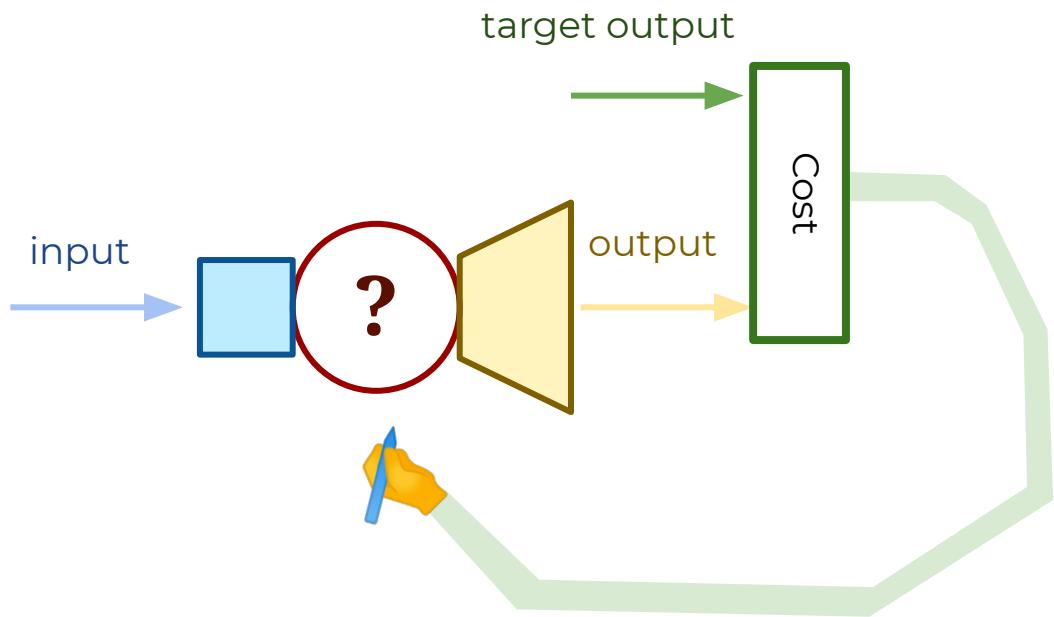
Can be analyzed in detail



Why Study (this specific form of) RNNs?

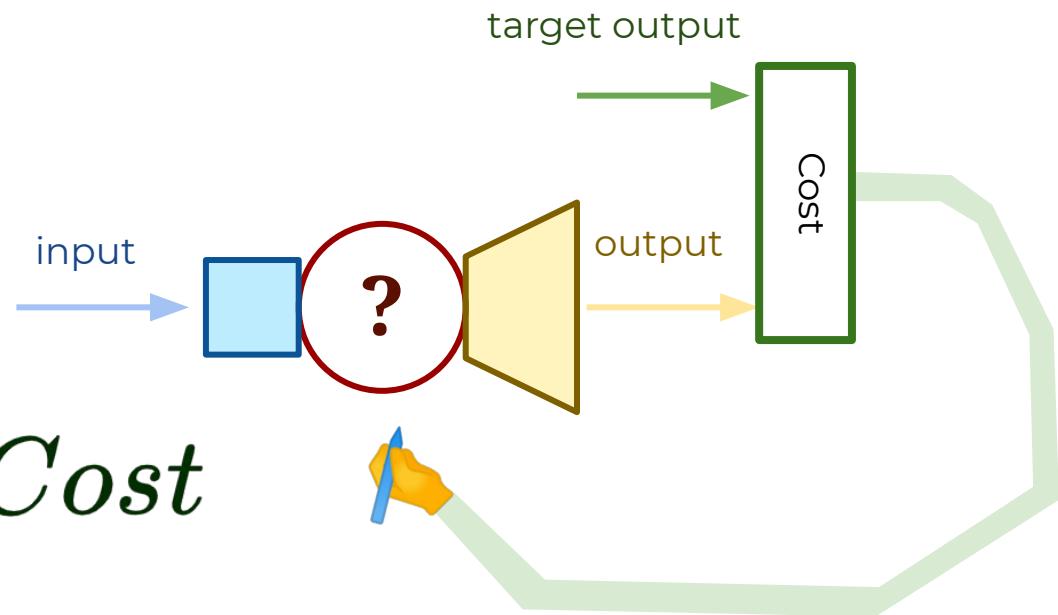
En route to a Theory of Cognition as Dynamics		
Where are we?	What's the problem?	How do we solve it?
Linear RNNs $\dot{x} = -x + Jx$ Two patterns, stable one → TFP, 0 	Activity that does not decay is unstable, i.e., blows up 	Nonlinearities $\phi(x) = \tanh(x)$
Nonlinear RNNs, random J $\dot{x} = -x + J\phi(x)$ Tame instability + get rich dynamics 	Dynamics are chaotic, can't reliably get the same pattern twice 	External inputs to control chaos $h = I \cos(\omega t + \theta)$
Driven nonlinear RNNs, random J $\dot{x} = -x + J\phi(x) + h$ Input-driven activity on rich ongoing background Without chaotic sensitivity to initial conditions, reliable responses 	Brain does more than reconcile subtle inputs and ongoing activity	Train nonlinear RNNs to get a specialized J: $J_{\text{rand}} \rightarrow J_{\text{trained}}$

Training an ANN

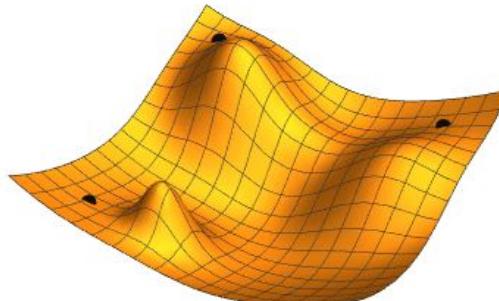


*Adjust weights in
“error minimizing”
directions
A.k.a.
Gradient Descent*

Training an ANN

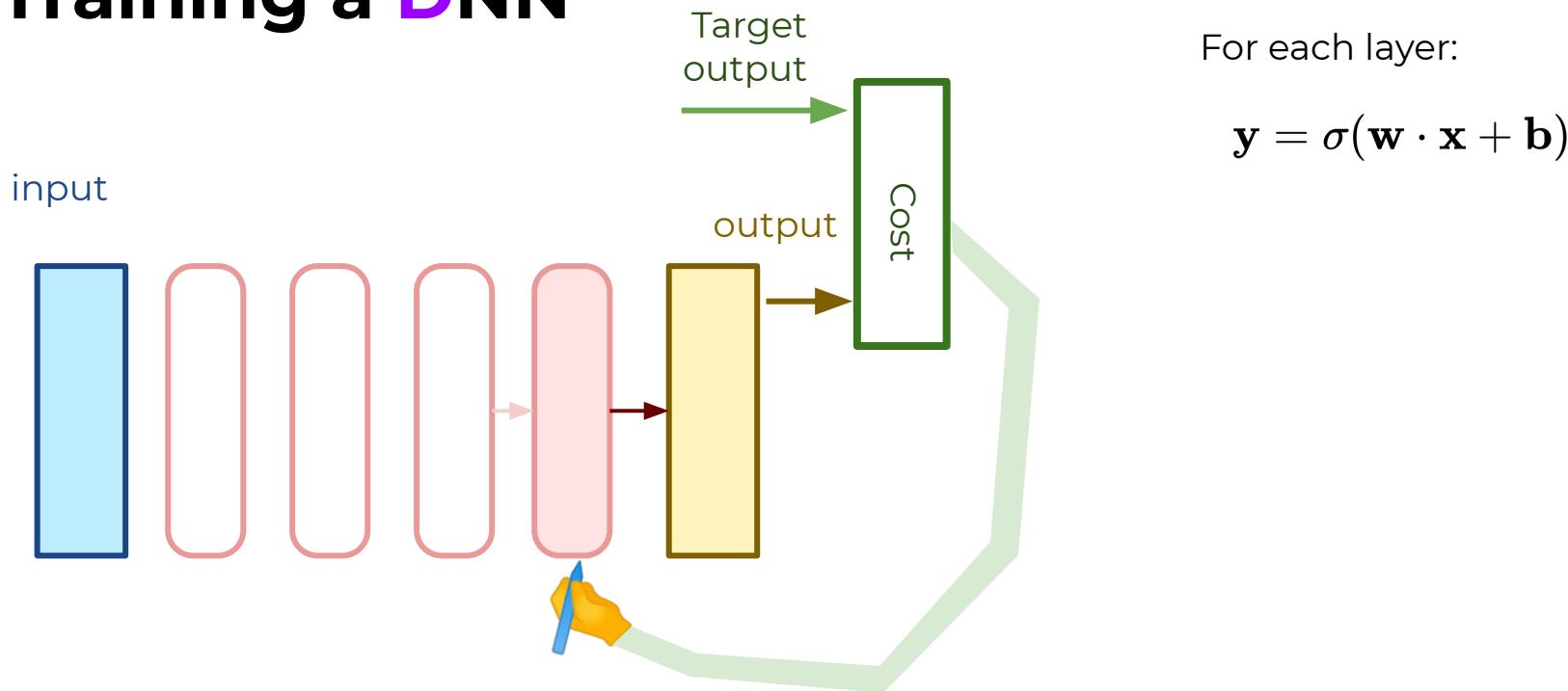


$$w \leftarrow w - \eta \nabla_w Cost$$



*Adjust weights in
“error minimizing”
directions
A.k.a.
Gradient Descent*

Training a DNN

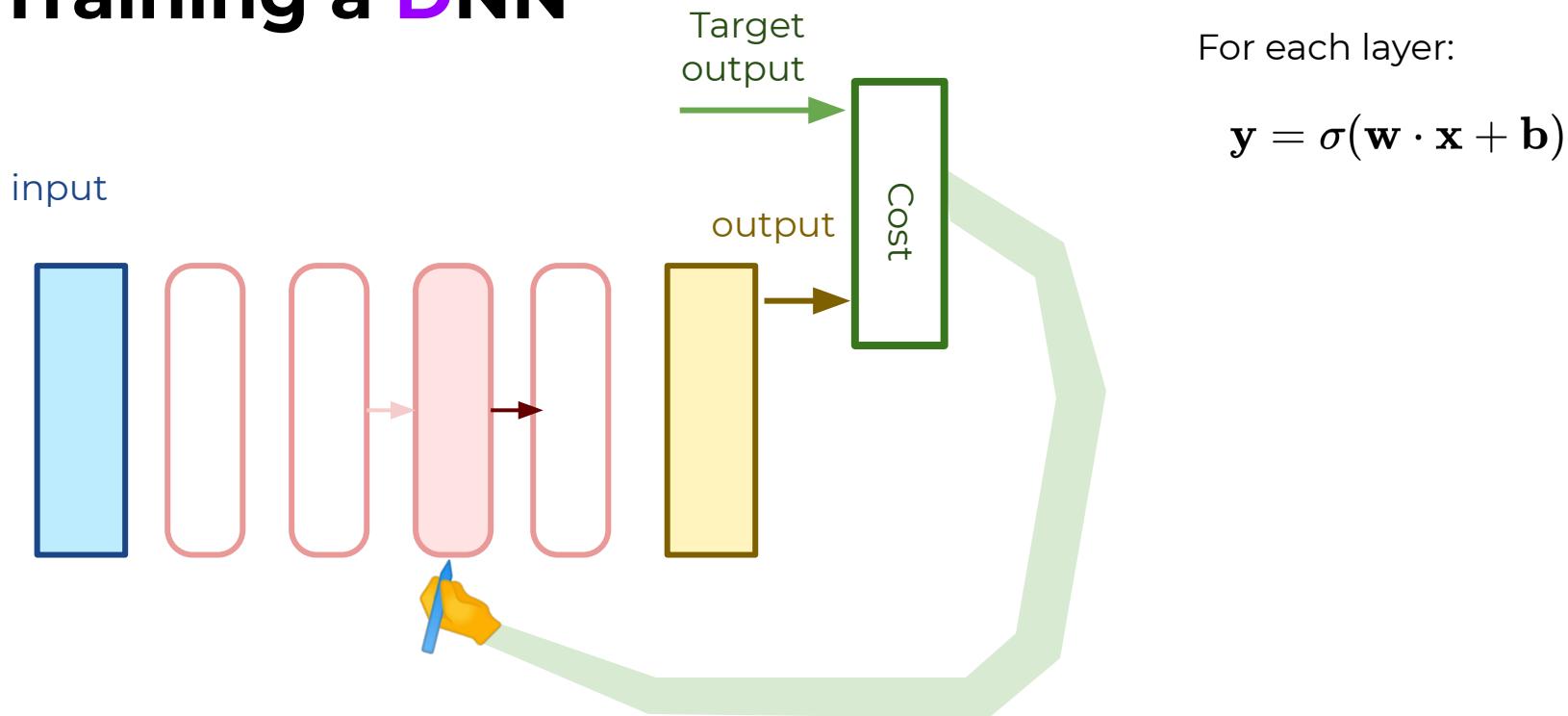


For each layer:

$$\mathbf{y} = \sigma(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$$

Backpropagation (through layers)

Training a DNN

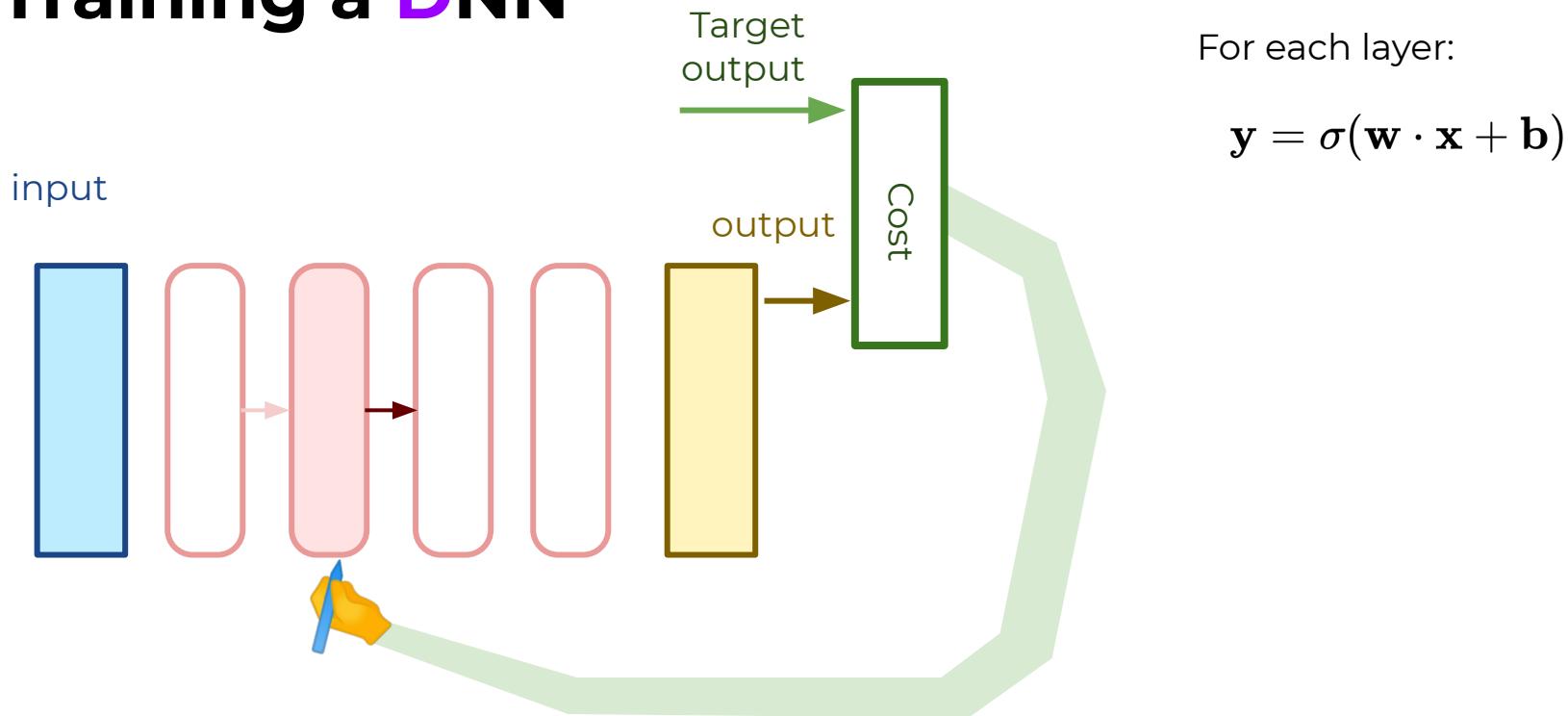


For each layer:

$$\mathbf{y} = \sigma(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$$

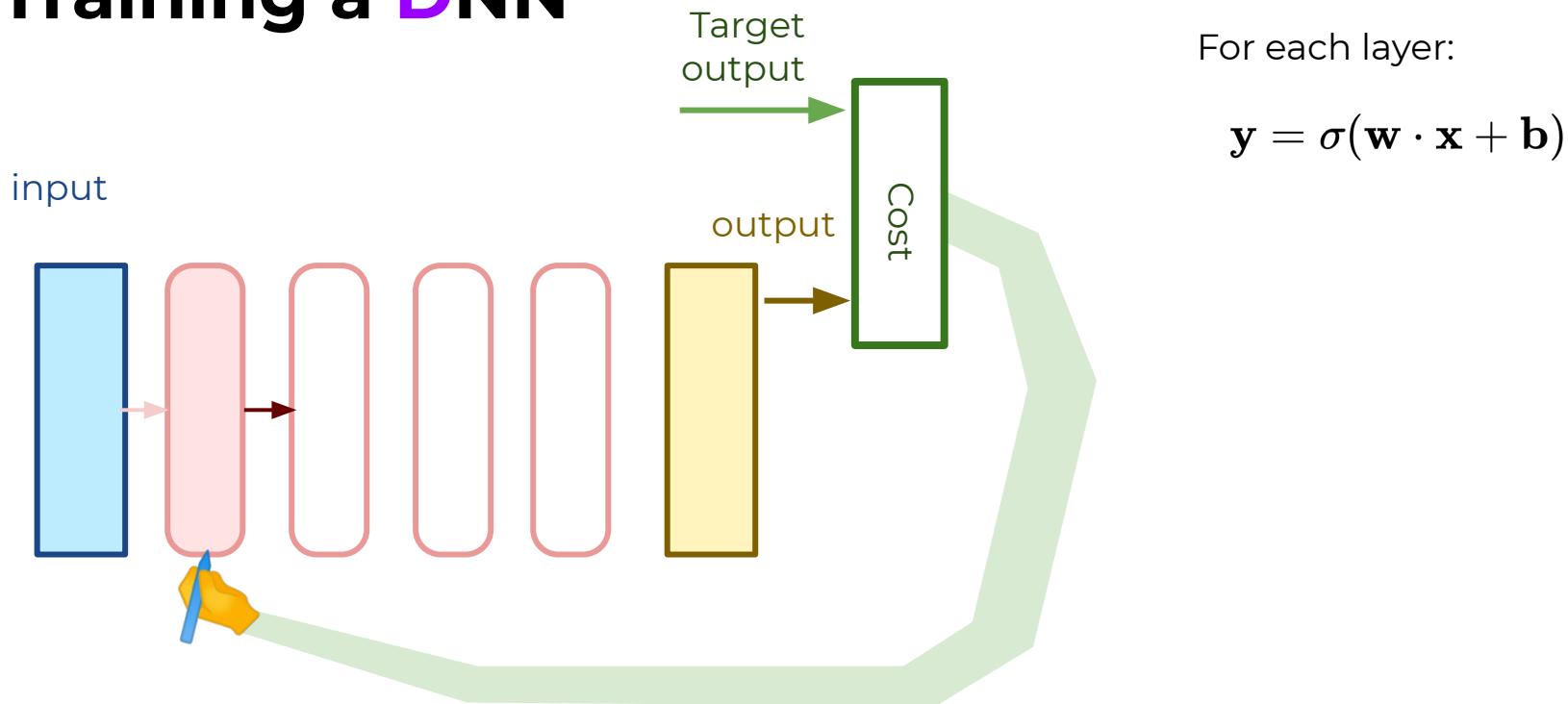
Backpropagation (through layers)

Training a DNN



Backpropagation (through layers)

Training a DNN

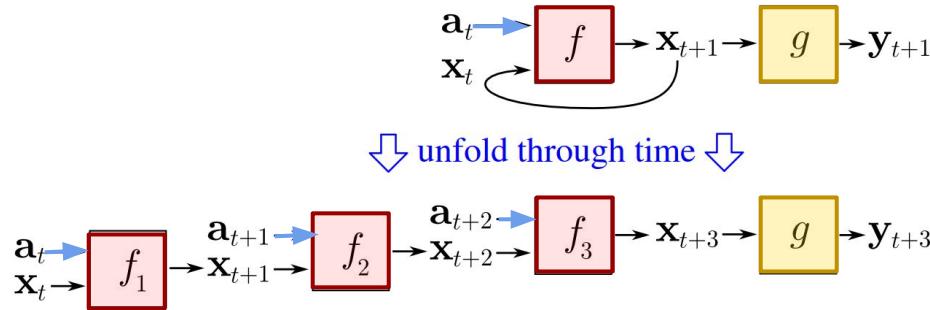


Backpropagation (through layers)

Training an RNN

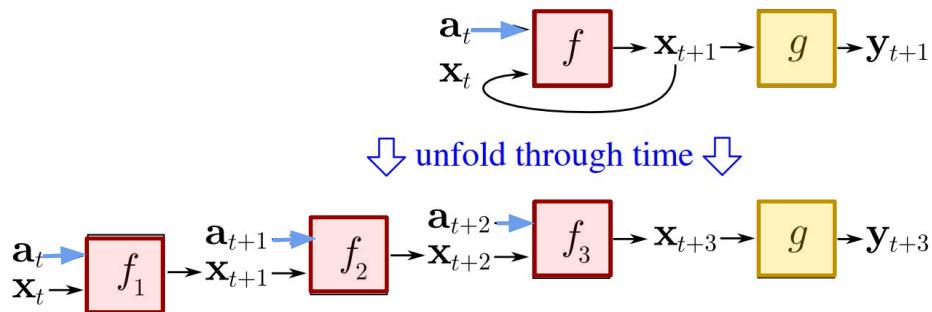
How to deal with each sample depending on the previous sample?

- “Unroll” network over time
- Then treat this like a DeepNN

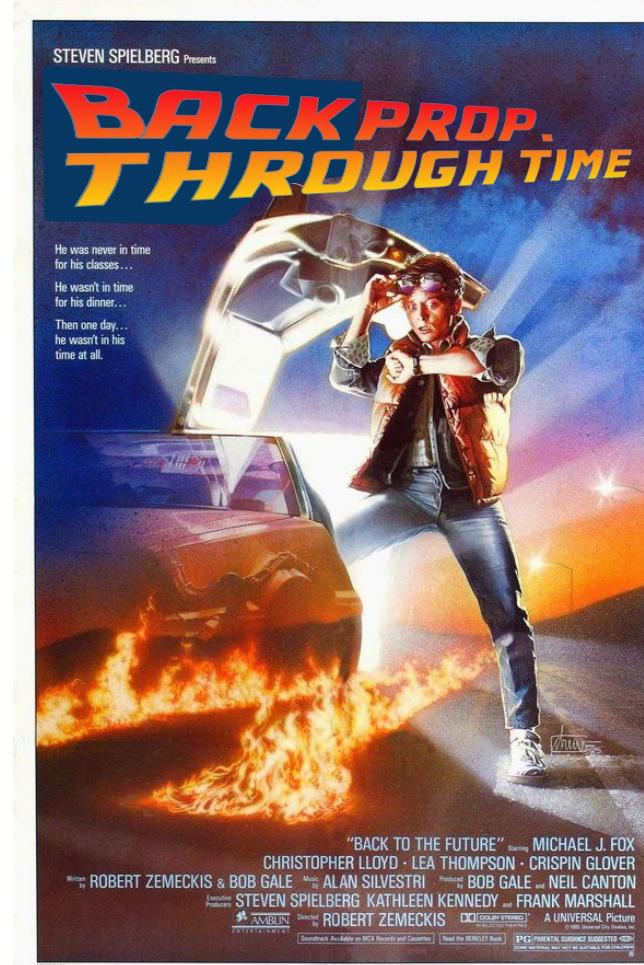


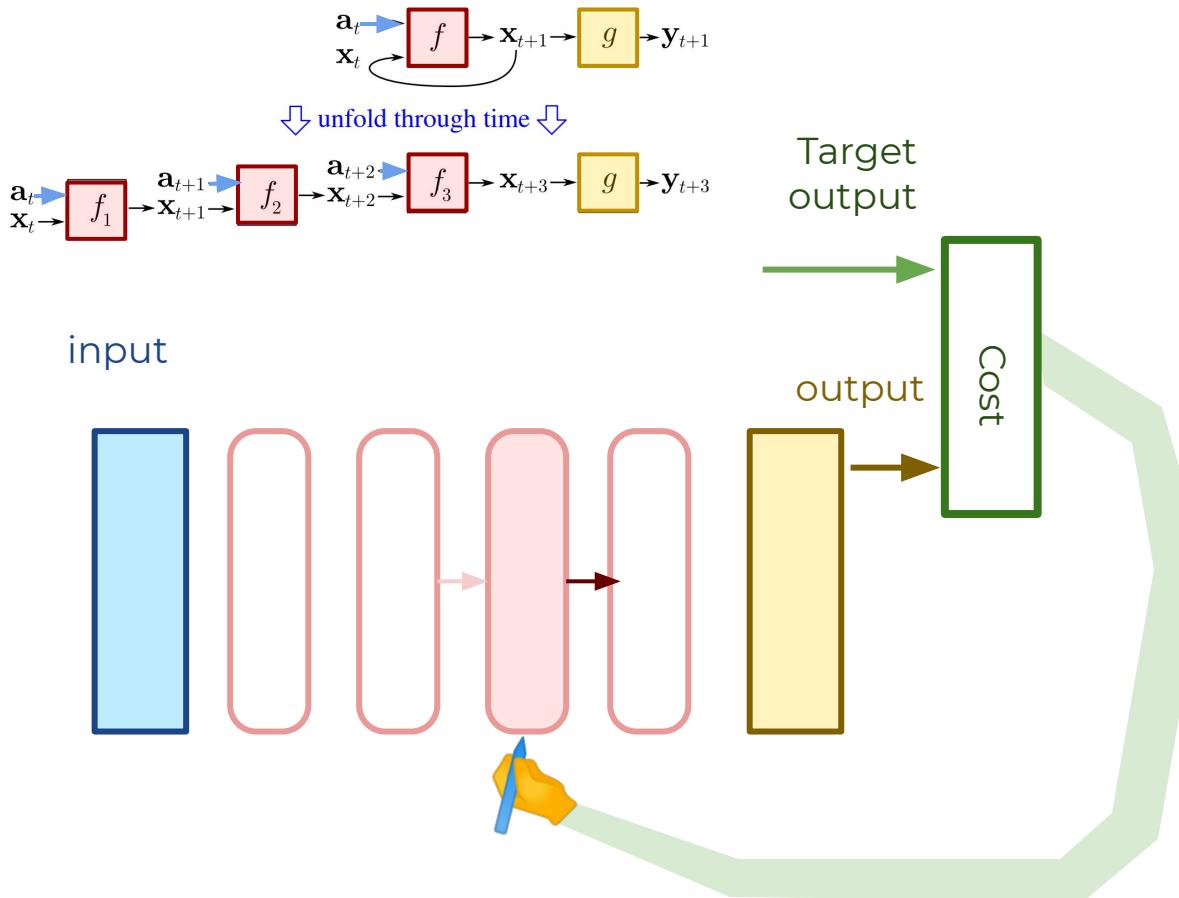
Training an RNN

Backpropagation (**through time**)

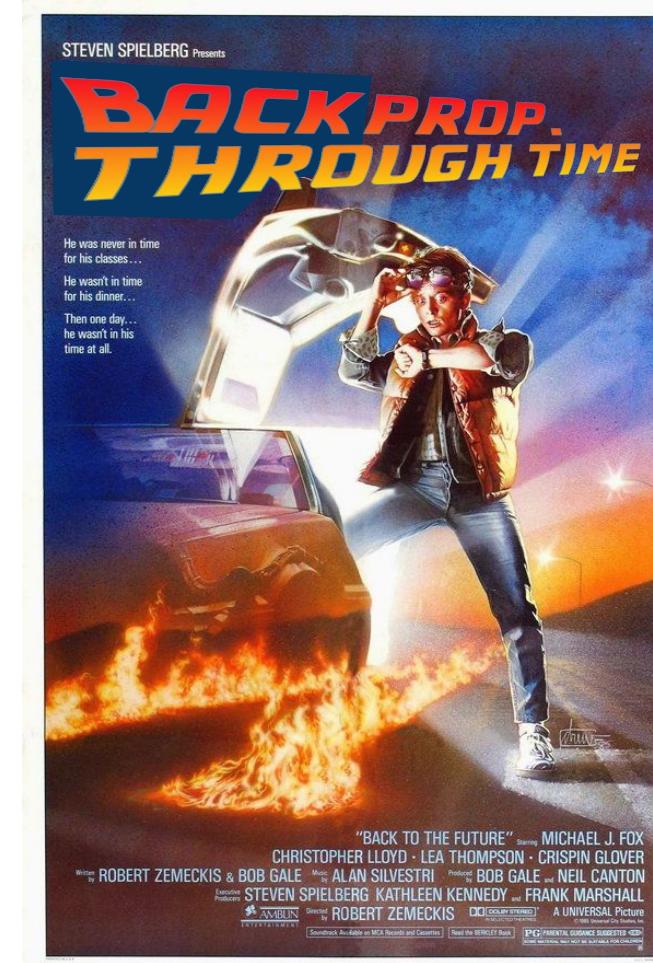


[“Backpropagation through time” Wikipedia](#)



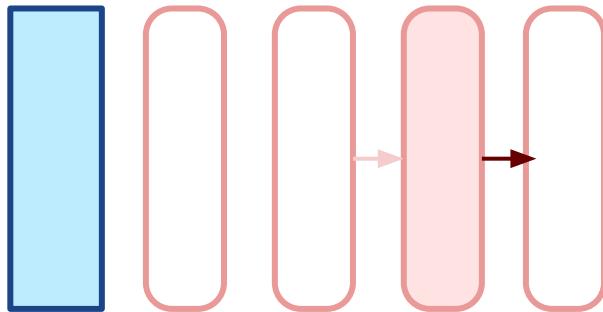


Backpropagation (through time)



Training an RNN

input



Target
output



output

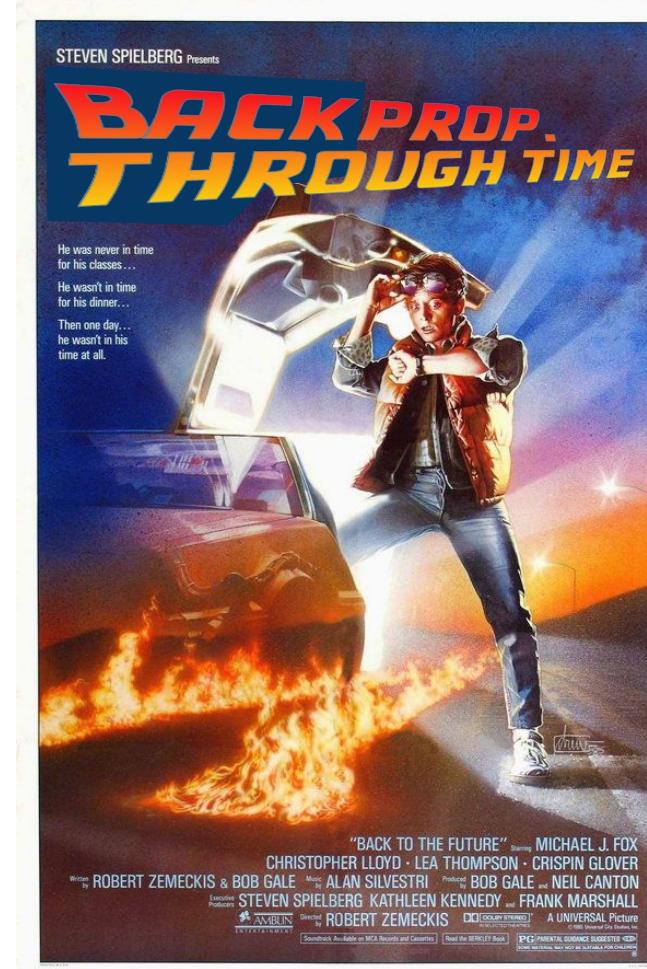
Cost

Backpropagation (through time)

See also:

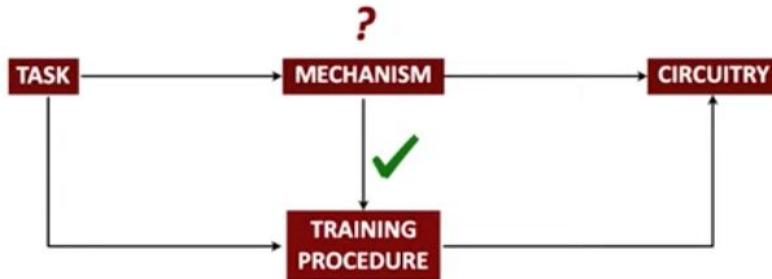
[“What is backpropagation really doing?” 3Blue1Brown](#)

[“Backpropagation through time: what it does and how to do it” \(1990\) Werbos](#)



Training an RNN: Further Reading

Approaches for building/training neural network models in neuroscience



Attractor models, integrator networks, etc.

Wang, Deneve, Eliasmith, Fiete, Goldman, Hopfield, Seung, Sompolinsky labs, and others

FORCE, Back-propagation, Back-propagation through time (BPTT), etc.

Buonomano, Rajan, Jaeger, Maass, Sussillo, Yang, Murray, Abbott, Kao, Shenoy labs, others

Neural networks as hypothesis generators OR

Network models trained to match Neural and/or Behavioral Data directly

Goldman, Rajan, Sussillo, Mante, Yang, Yamins, DiCarlo, Wang labs, and others

See also:

AutoLFADS -

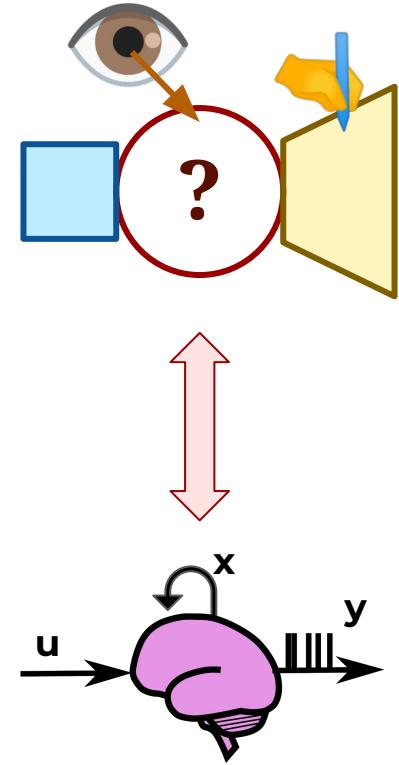
"A large-scale neural network training framework for generalized estimation of single-trial population dynamics"

(2021) Keshtkaran

RNNs for hypothesis generation

By interrogating the structure of a task-optimized RNN we can generate hypotheses :

- Suggest (novel? unexpected?) mechanisms for neural circuits accomplishing tasks
- Suggest new experiments & analysis
- Become a testing ground for analysis



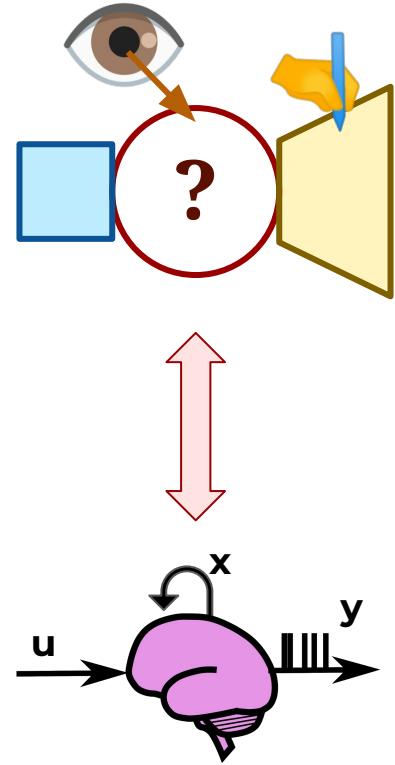
See

["Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks."](#) (2013) Sussillo, Barak
For how to use fixed points, dynamical systems analysis to make sense of what's going on inside an RNN

How to use RNNs to interpret how it accomplishes tasks

Interrogate & interpret through the lens of dynamical systems

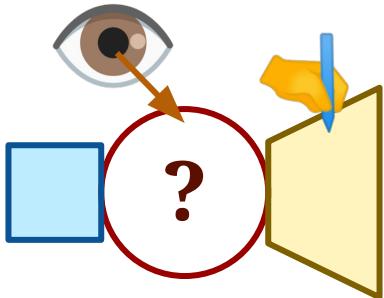
- Find fixed points
- Locally linearize
- Categorize behavior (by eigenvalues!)
- Fixed points form a “dynamical skeleton”



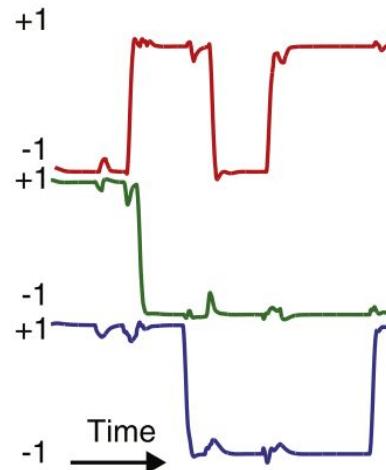
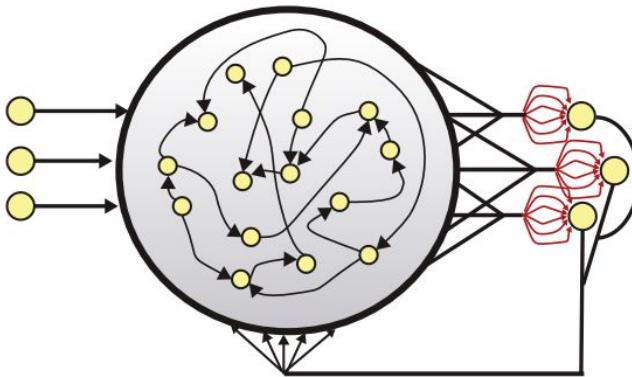
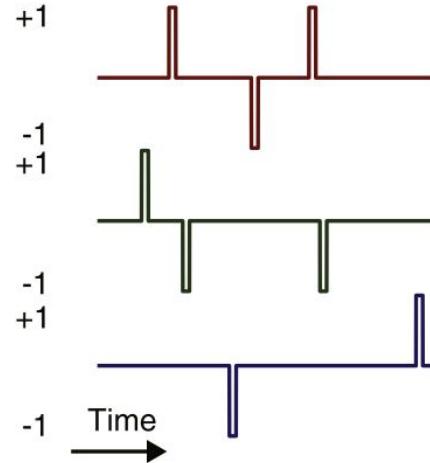
See

[“Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks.”](#) (2013) Sussillo, Barak
For how to use fixed points, dynamical systems analysis to make sense of what's going on inside an RNN

Interrogate & interpret through the lens of dynamical systems



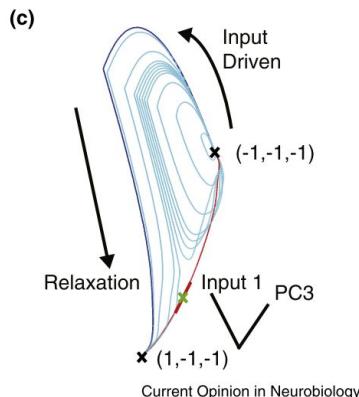
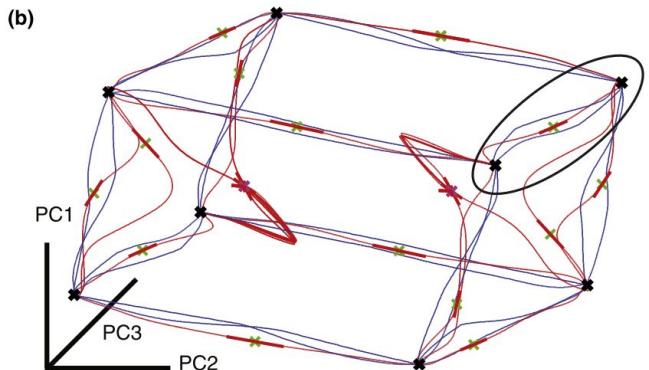
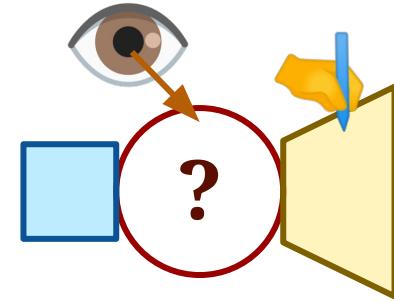
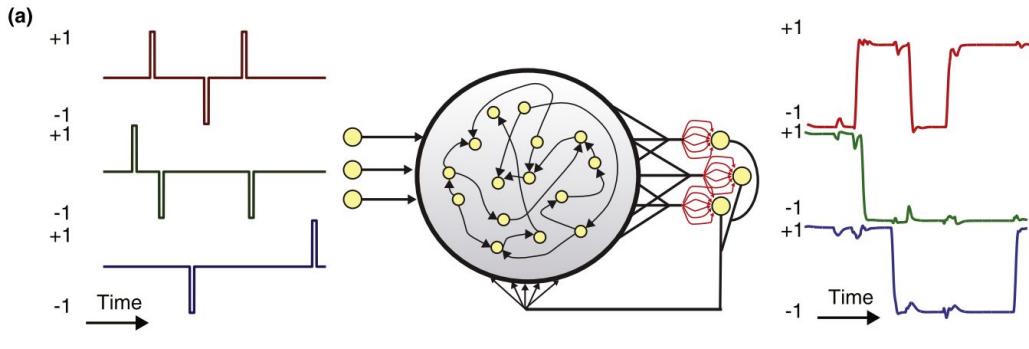
(a)



See

["Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks."](#) (2013) Sussillo, Barak
For how to use fixed points, dynamical systems analysis to make sense of what's going on inside an RNN

Interrogate & interpret through the lens of dynamical systems



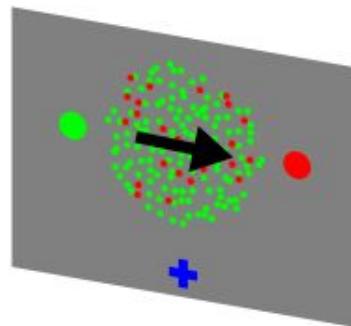
Current Opinion in Neurobiology

See

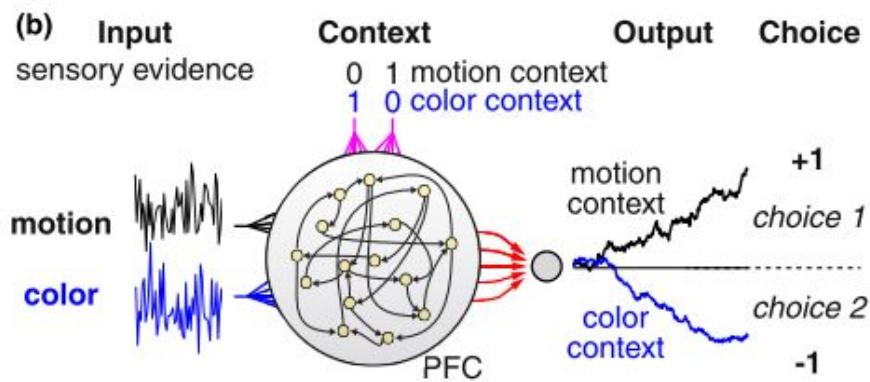
["Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks."](#) (2013) Sussillo, Barak
For how to use fixed points, dynamical systems analysis to make sense of what's going on inside an RNN

Revisiting context-dependent evidence accumulation

(a)

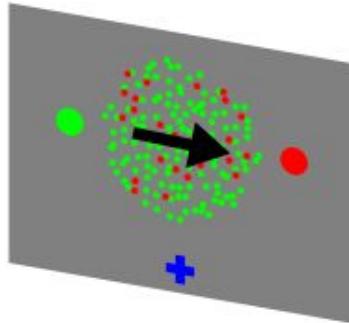


fixation point = cue
[motion context
+ color context]

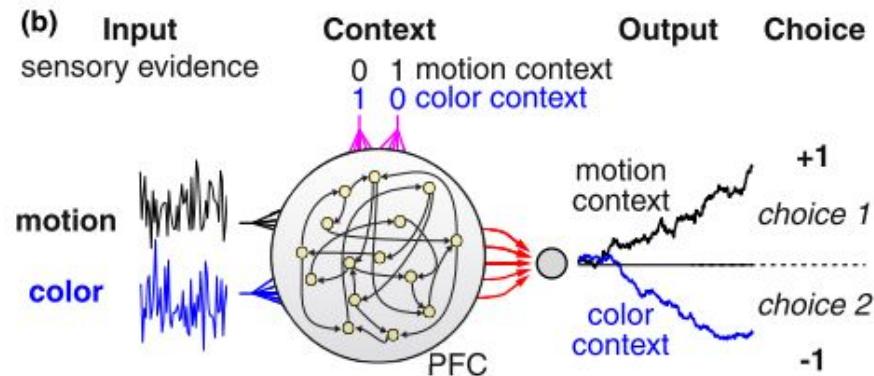


Revisiting context-dependent evidence accumulation

(a)



fixation point = cue
[
■ motion context
+ color context

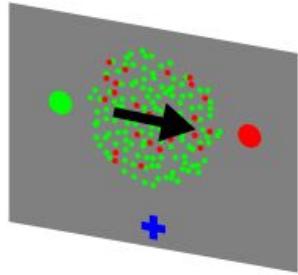


Prevailing theory: Top-down attention “gates” representation of features in cortex

Let's ask RNNs what they do!

RNN selects which features to accumulate

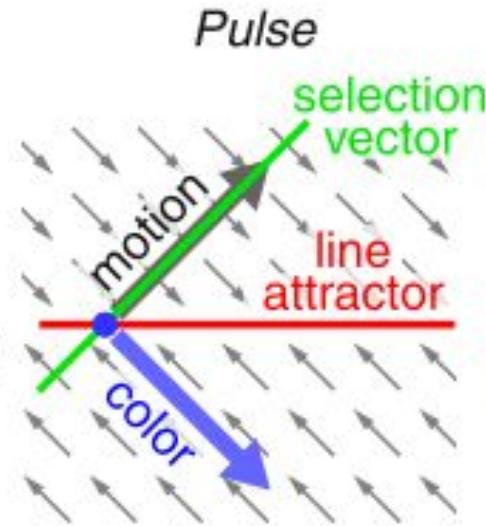
(a)



fixation
point = cue

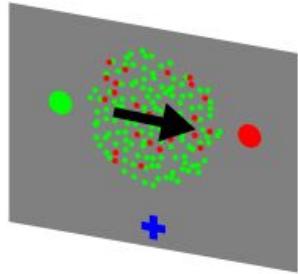
- motion
context
- color
context

**motion
context**



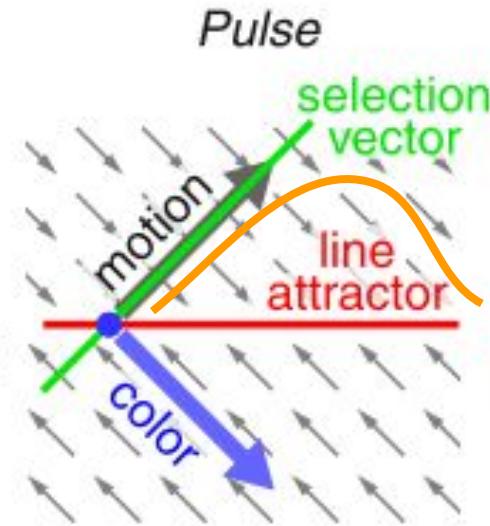
RNN selects which features to accumulate

(a)



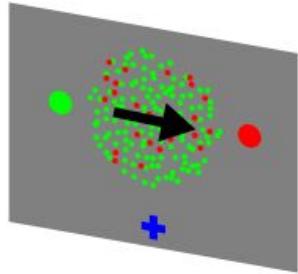
fixation point = cue
[motion context
color context]

motion context



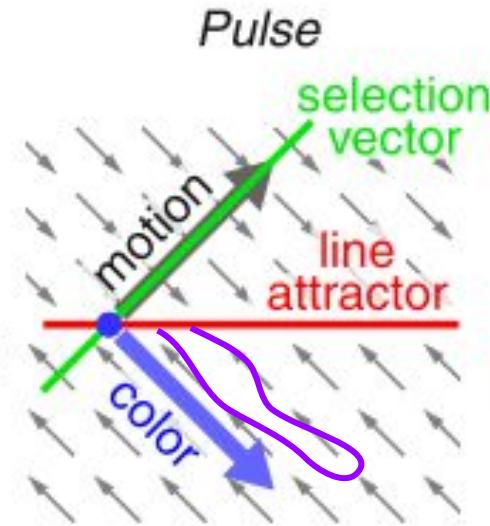
RNN selects which features to accumulate

(a)



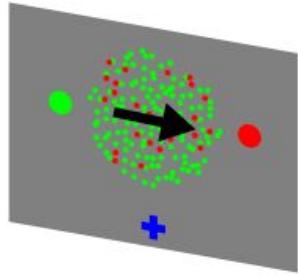
fixation point = cue
[motion context
color context]

motion context

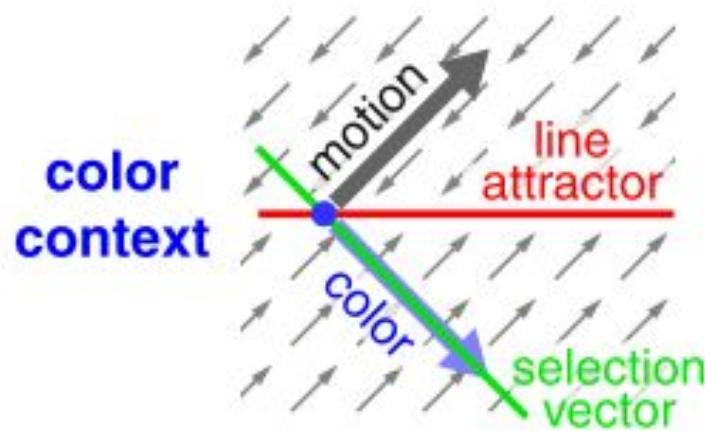


RNN selects which features to accumulate

(a)

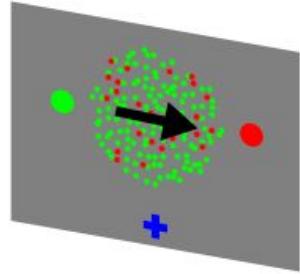


fixation point = cue
[motion context
+ color context]



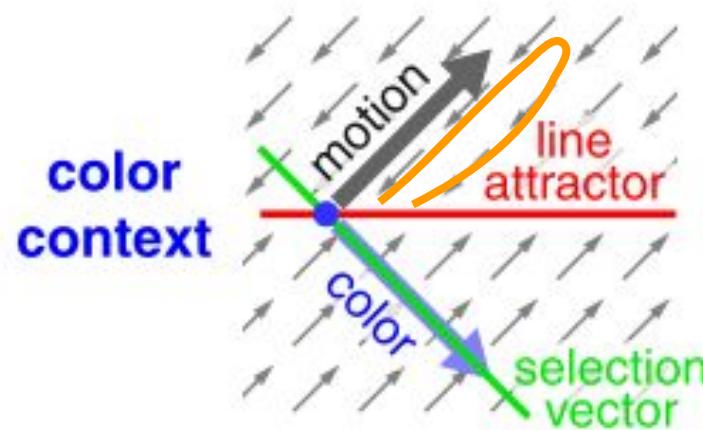
RNN selects which features to accumulate

(a)



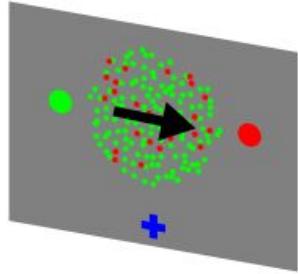
fixation
point = cue

- motion context
- ✚ color context



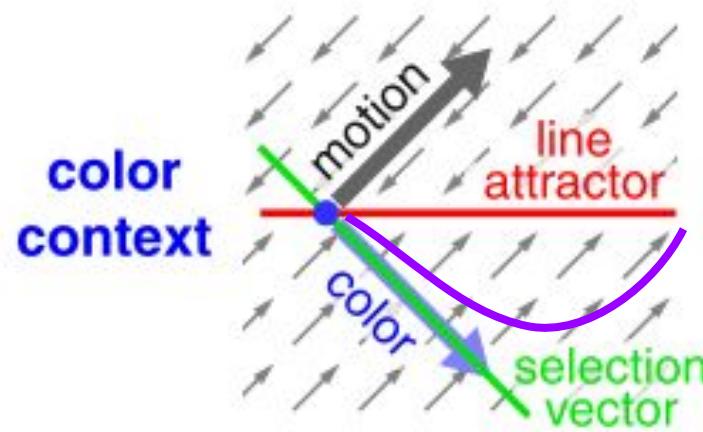
RNN selects which features to accumulate

(a)



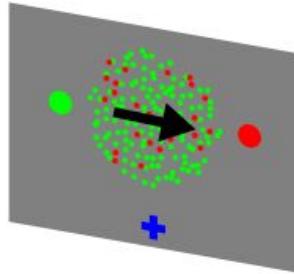
fixation
point = cue

- motion
context
- color
context

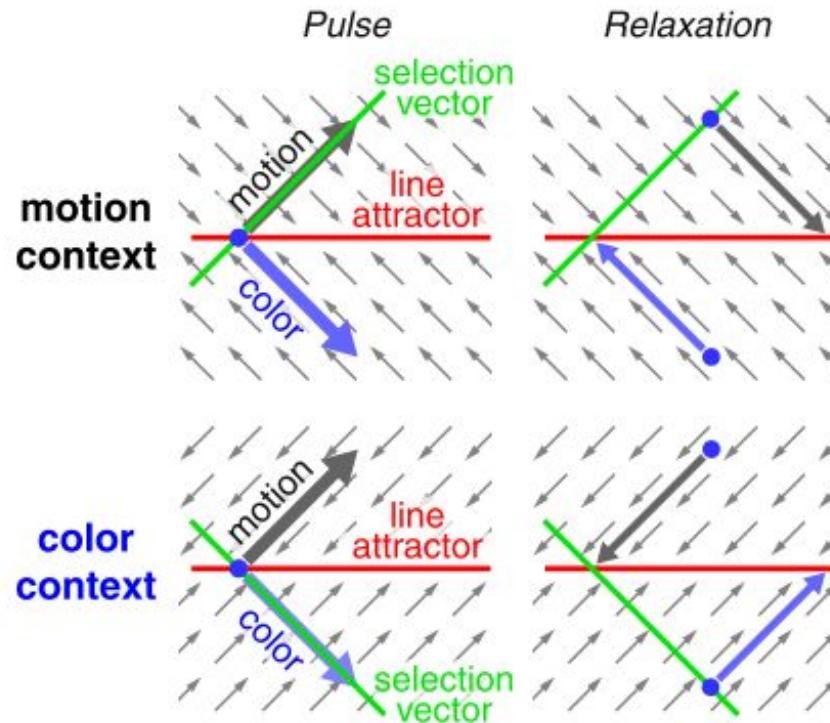


RNN selects which features to accumulate

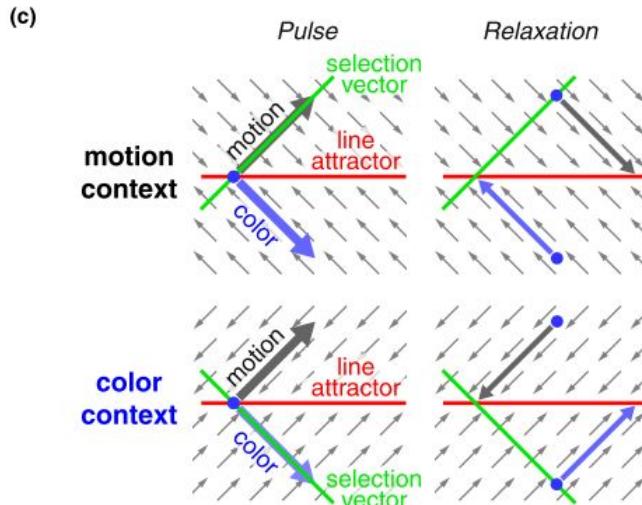
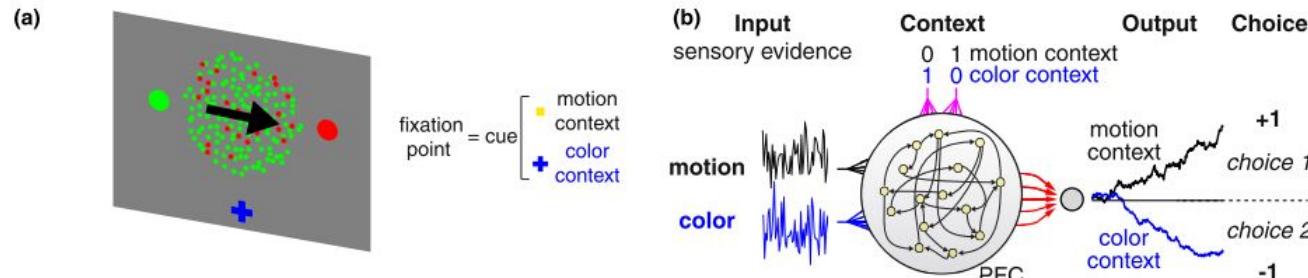
(a)



fixation point = cue
[motion context
color context]



RNN selects which features to accumulate

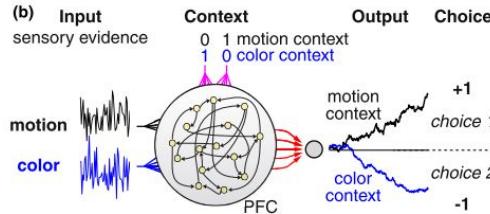
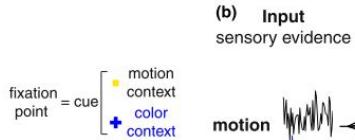
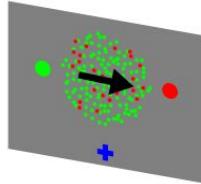


RNN's solution involves nonlinear "non-normal" dynamics

- BOTH motion and color are represented in dynamical state space (contrary to gating theory)
- Evidence for each is accumulated with a line attractor
- Line attractors are ORTHOGONAL to each other
- **Input is selectively routed** to one or the other using a context-dependent selection vector

Revisiting context-dependent evidence accumulation

(a)



RNN's solution involves nonlinear "non-normal" dynamics

- BOTH motion and color are represented in dynamical state space (contrary to gating theory)
- Evidence for each is accumulated with a line attractor
- Line attractors are ORTHOGONAL to each other
- **Input is selectively routed** to one or the other using a context-dependent selection vector

Non-normal network dynamics:

["Structure and dynamical behavior of non-normal networks"](#) (2018) Asllani et al.

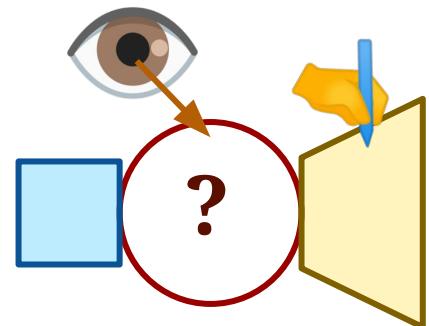
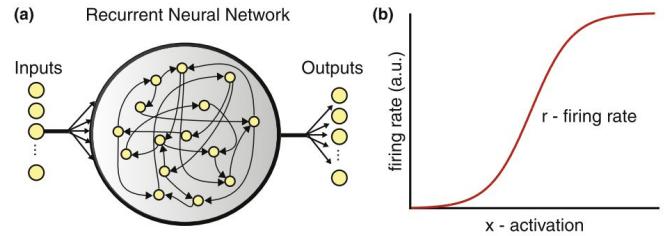
["Efficient communication over complex dynamical networks: The role of matrix non-normality"](#) (2020) Baggio et al.

["Talk: Just-in-time computations by non-normal attractor dynamics in the prefrontal cortex"](#) (2020) Stroud

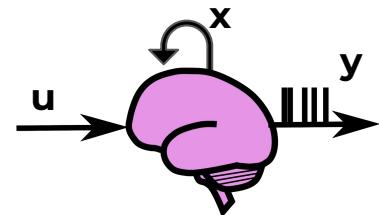
["Balanced Amplification: A New Mechanism of Selective Amplification of Neural Activity Patterns"](#) (2009) Murphy & Miller

Intermediate Summary

- RNNs combine dynamics with function-approximation capability of ANNs
- By training them to accomplish complex tasks we can **generate hypothesis** for how
- The structure of RNNs can be interpreted through the **lens of dynamical systems**

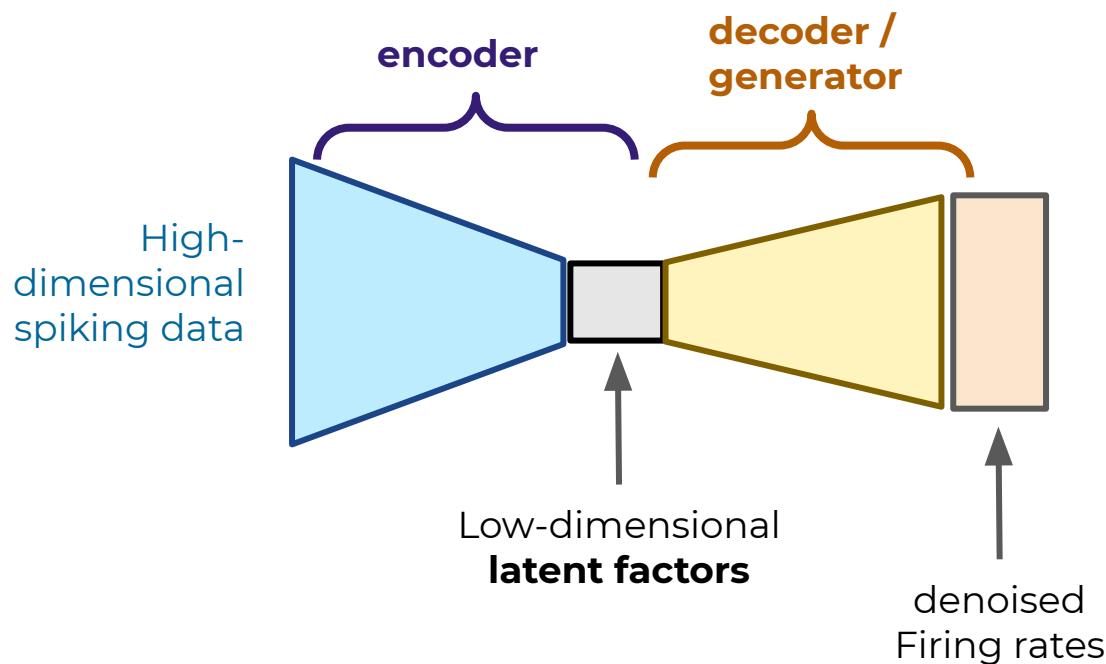


$$\tau \dot{\mathbf{x}}(t) = -\mathbf{x}(t) + \mathbf{J}\mathbf{r}(t) + \mathbf{B}\mathbf{u}(t) + b$$



Latent Factor Analysis via Dynamical Systems (LFADS)

An autoencoder...



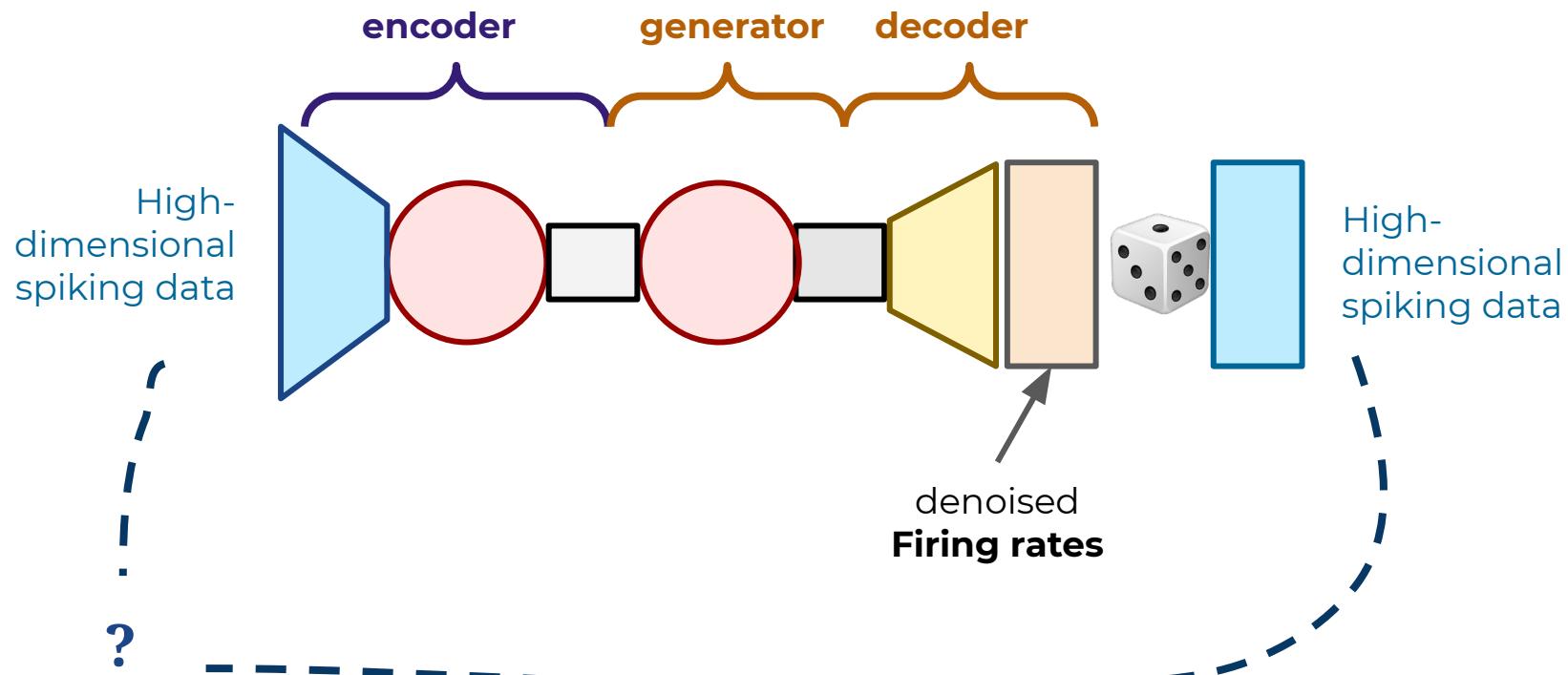
Denoise (single trial) neural data

By reconstructing it from low-dimensional factors

Learn something about the* underlying mechanism along the way

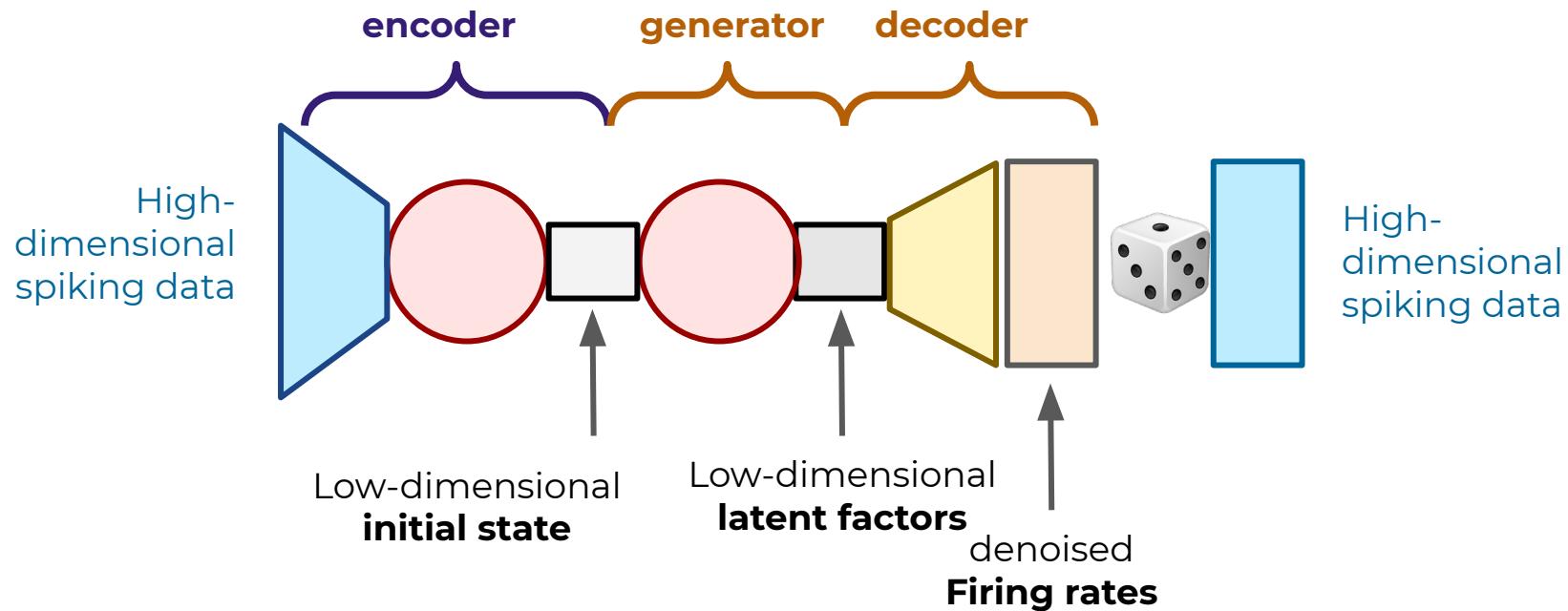
Latent Factor Analysis via Dynamical Systems (LFADS)

An autoencoder + dynamics!



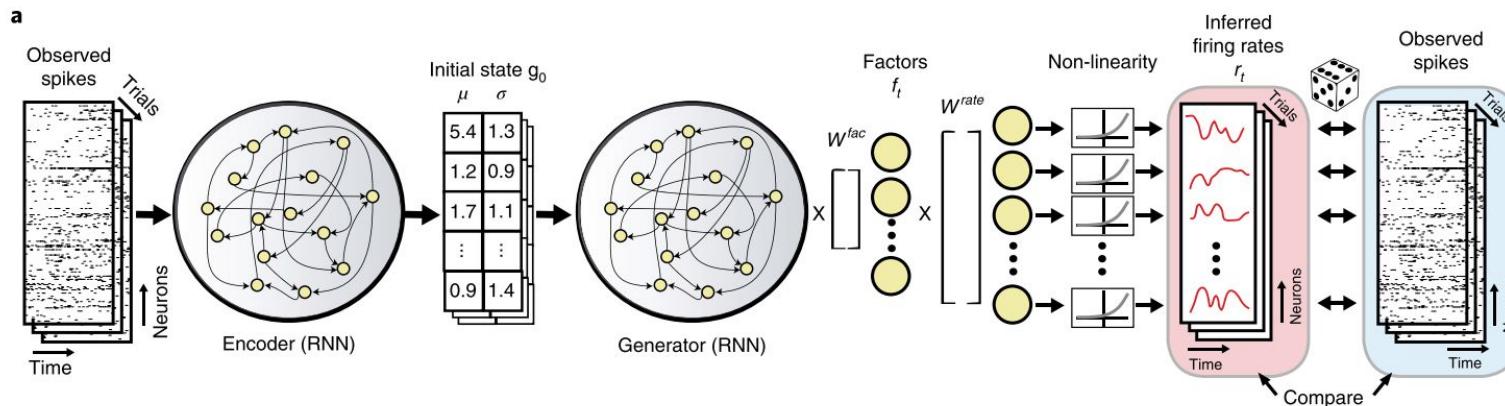
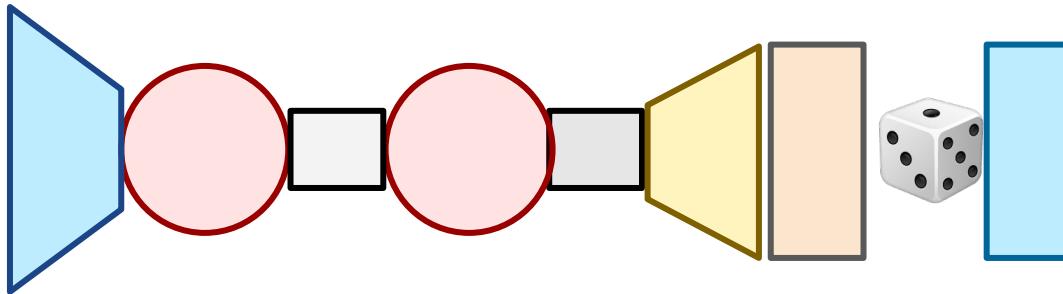
Latent Factor Analysis via Dynamical Systems (LFADS)

An autoencoder + dynamics!



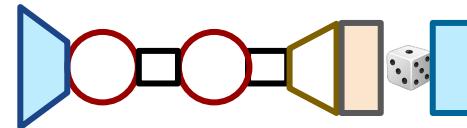
Latent Factor Analysis via Dynamical Systems (LFADS)

An autoencoder + dynamics!

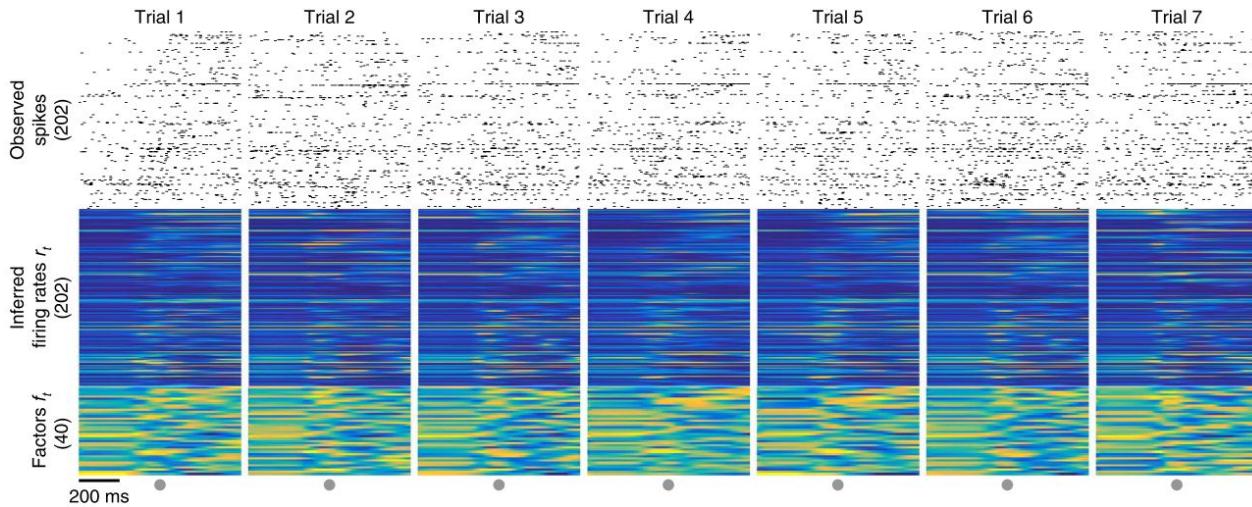
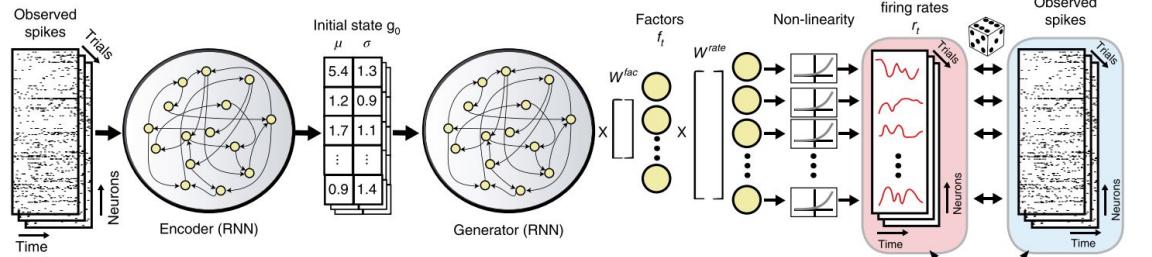


[“Inferring single-trial neural population dynamics using sequential auto-encoders” \(2018\) Pandarinath et al.](#)

LFADS Generates predictions of single-trial firing rates & spiking

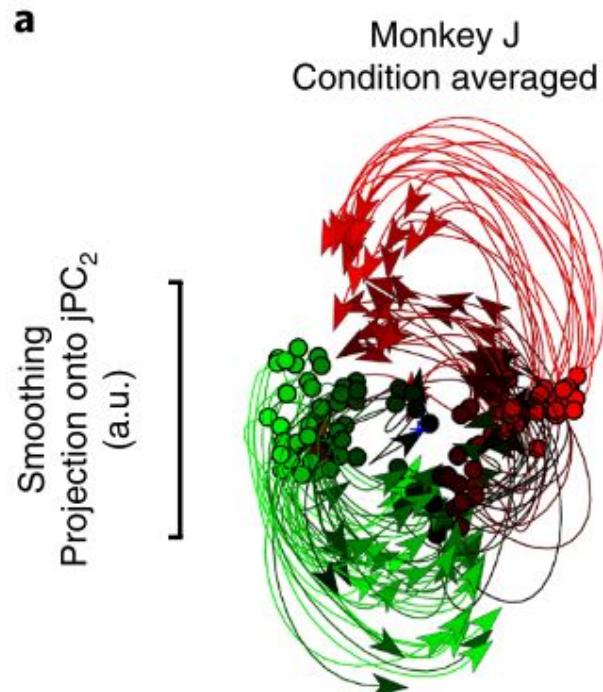
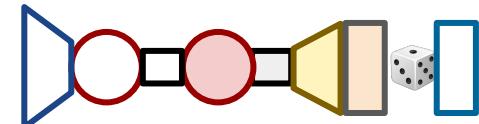


a

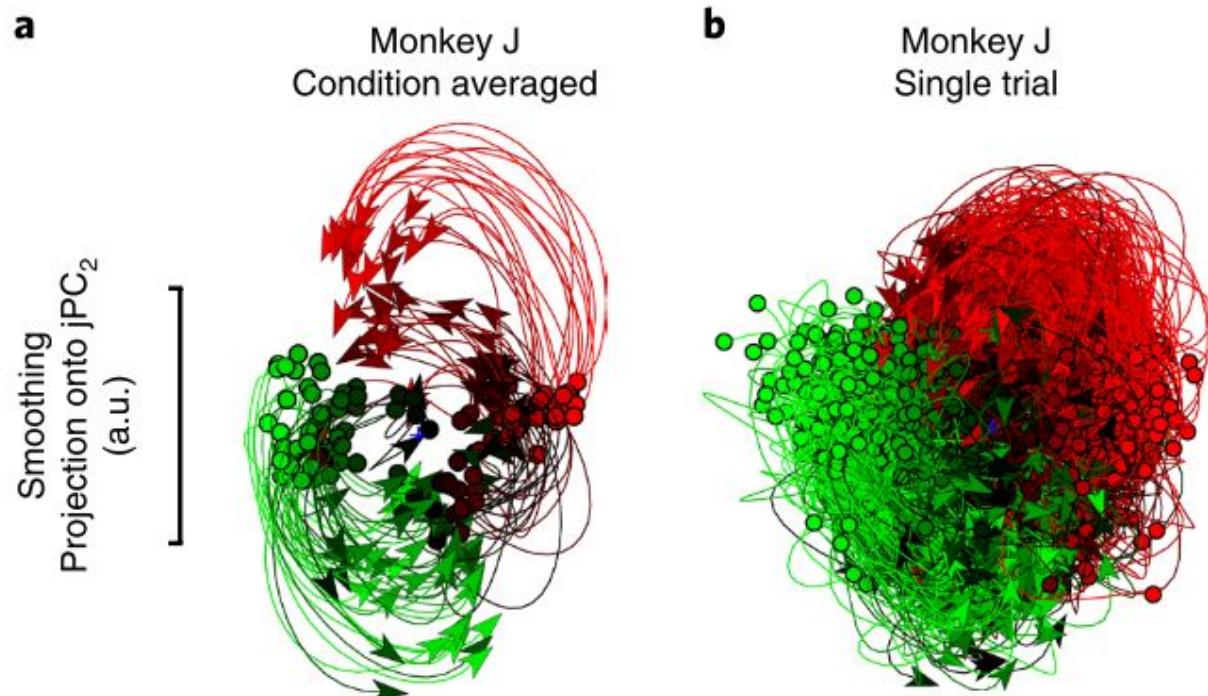
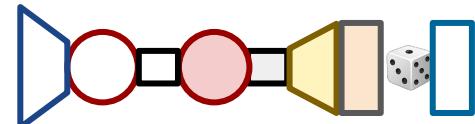


["Inferring single-trial neural population dynamics using sequential auto-encoders"](#) (2018) Pandarinath et al.

Spike-smoothing is insufficient to uncover rotation trajectories on single trials

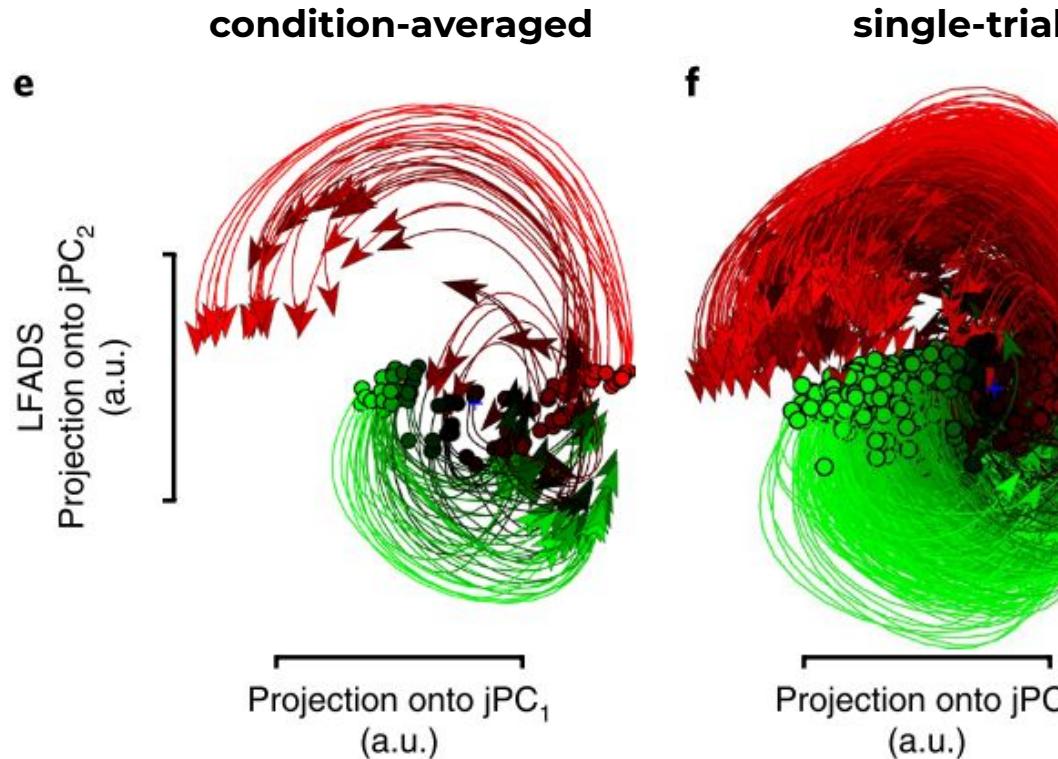
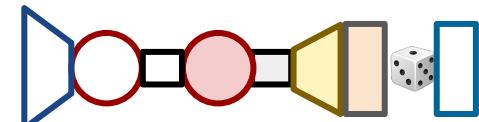


Spike-smoothing is insufficient to uncover rotation trajectories on single trials



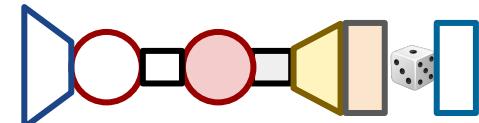
[“Inferring single-trial neural population dynamics using sequential auto-encoders” \(2018\) Pandarinath et al.](#)

But LFADS uses optimized latent structure to reveal **rotations on single-trials**

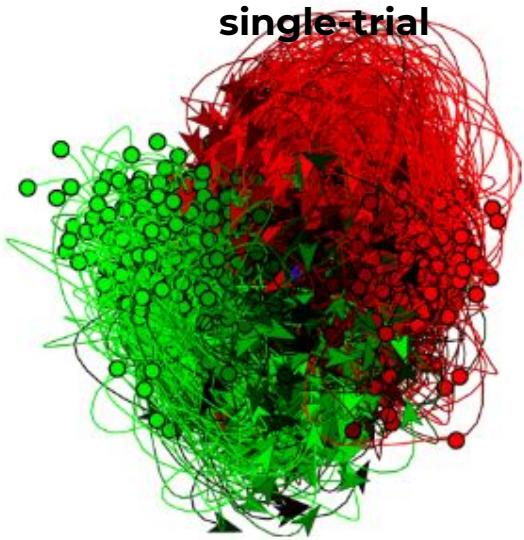


This is a critical tool for understanding computation through dynamics!

But LFADS uses optimized latent structure to reveal **rotations on single-trials**

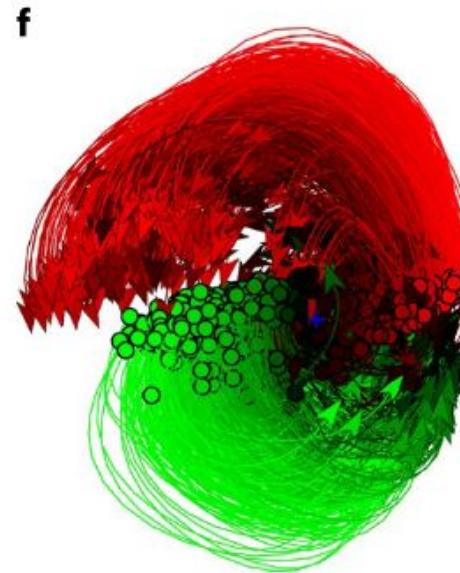


Smoothing
single-trial



*This is a critical
tool for
understanding
computation
through
dynamics!*

LFADS single-trial

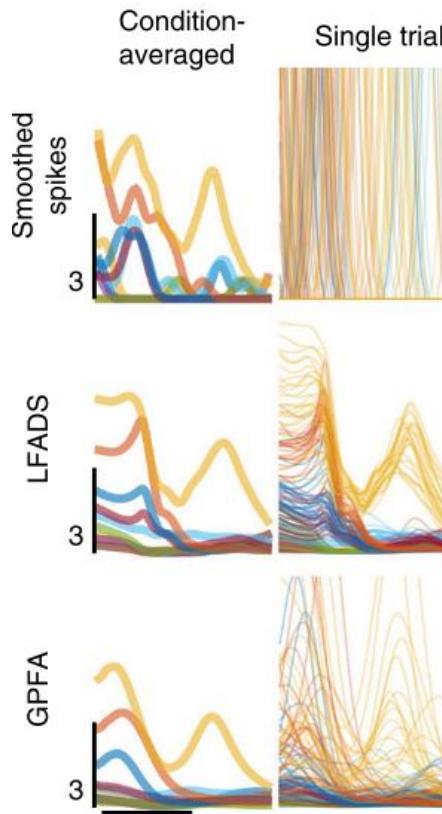


But LFADS uses
optimized latent
structure to reveal
single-trial trajectories

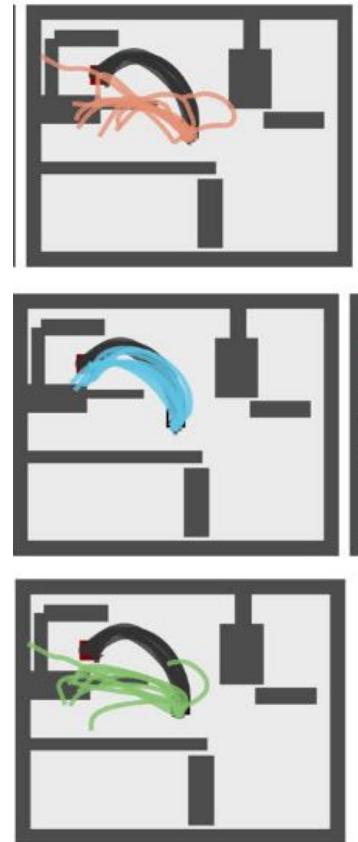
*This is a critical
tool for
understanding
computation
through
dynamics!*

"Inferring single-trial neural population dynamics using
sequential auto-encoders" (2018) Pandarinath et al.

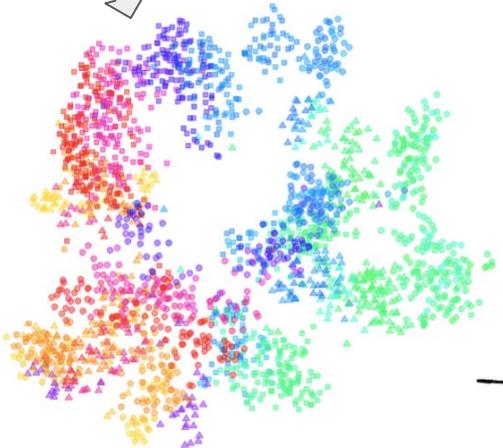
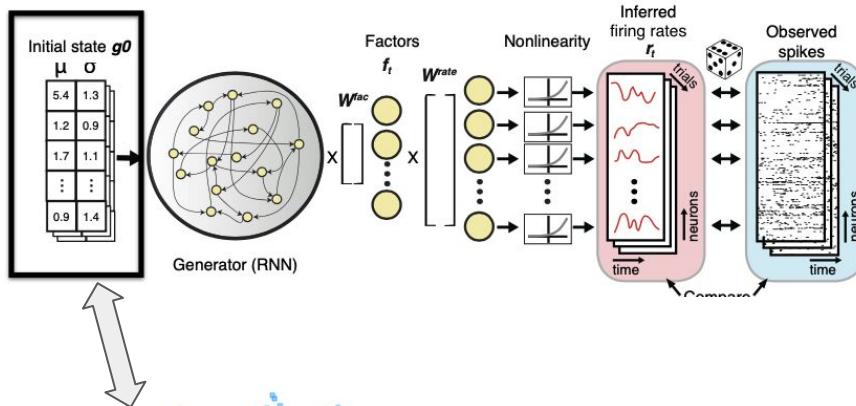
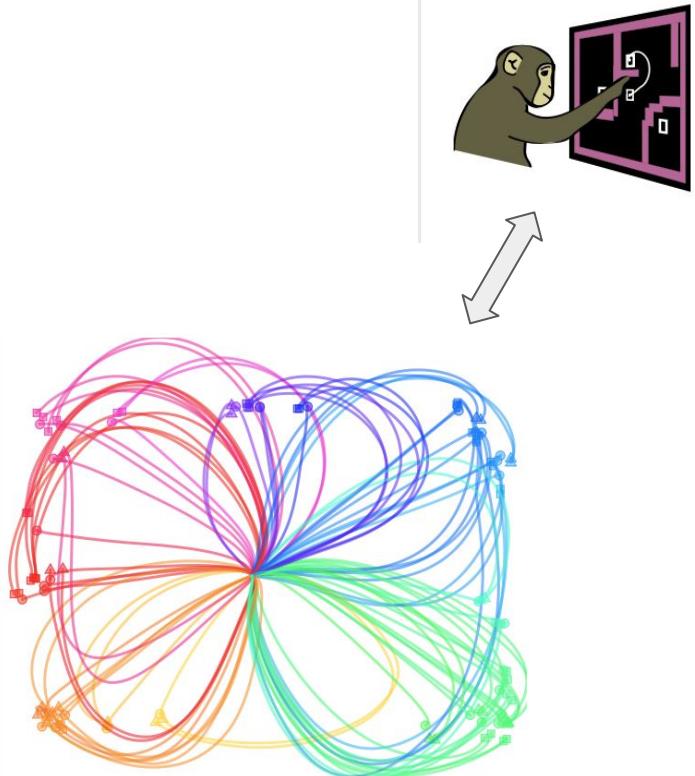
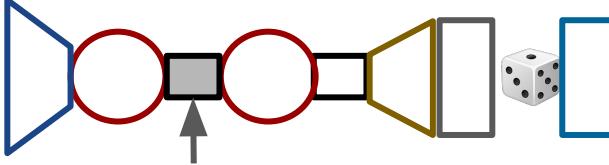
Firing rates



Reach kinematics



Inferred initial state corresponds with task / behavioral variables



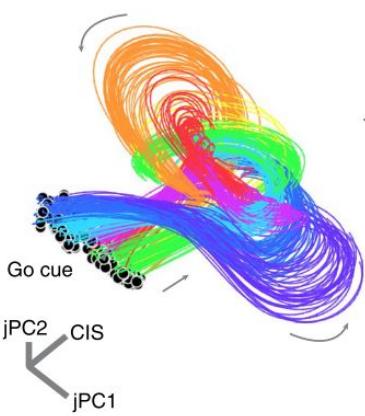
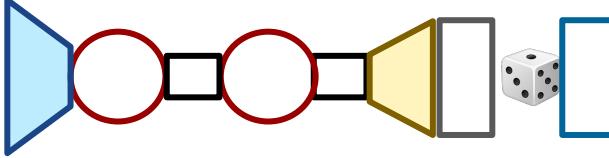
tSNE of inferred initial conditions

Color = target location

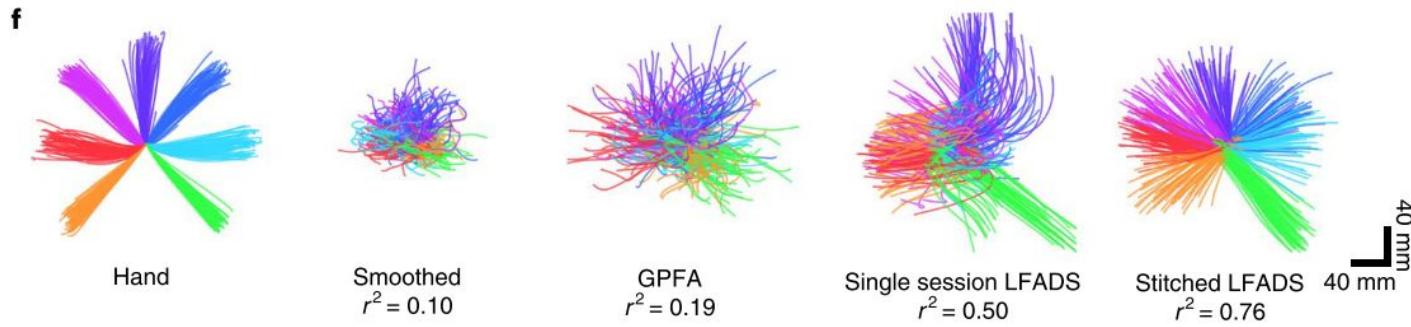
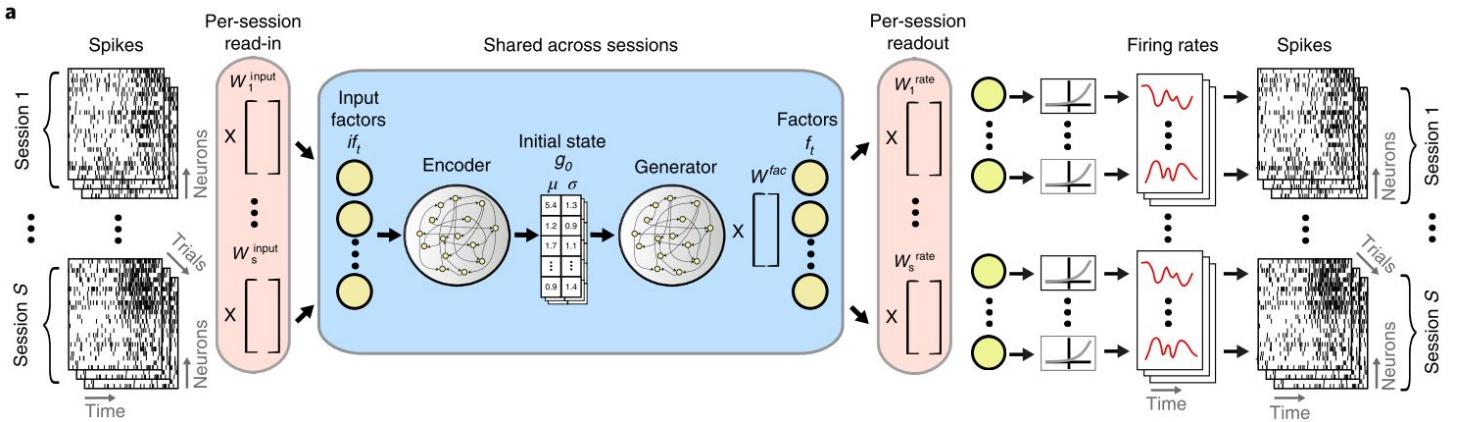
"Inferring single-trial neural population dynamics using sequential auto-encoders" (2018)
Pandarinath et al.

Bonus:

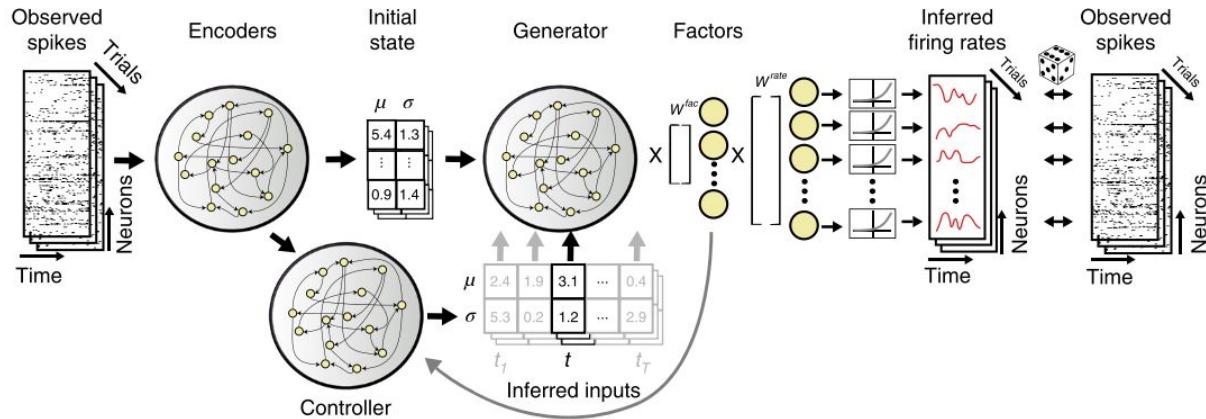
LFADS allows “stitching” together of neural data across sessions



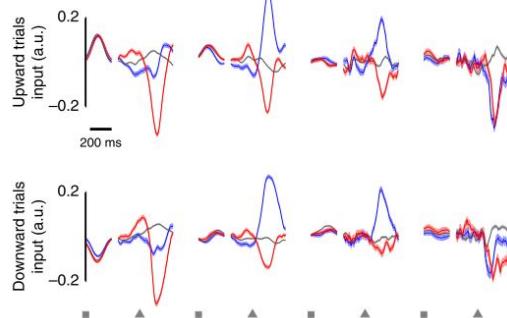
“Inferring single-trial neural population dynamics using sequential auto-encoders” (2018)
Pandarinath et al.



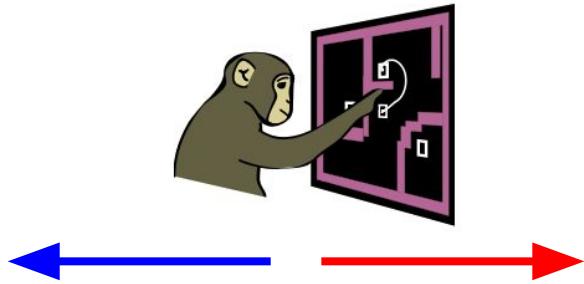
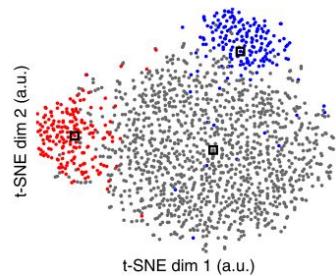
Bonus: LFADS infers latent perturbations



d Unperturbed Perturb right Perturb left
Input dim 1 dim 2 dim 3 dim 4

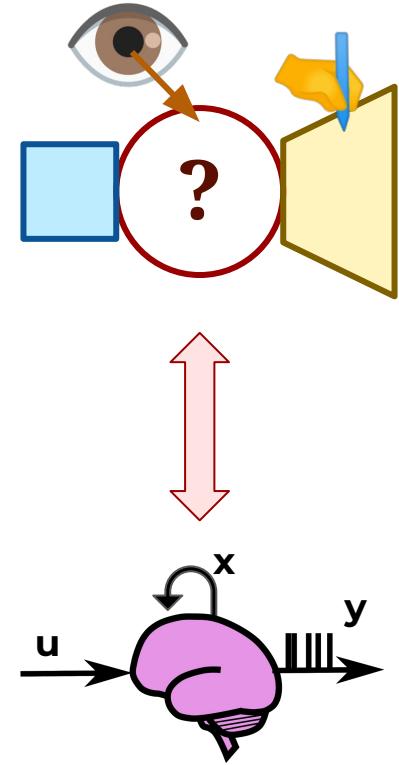


e



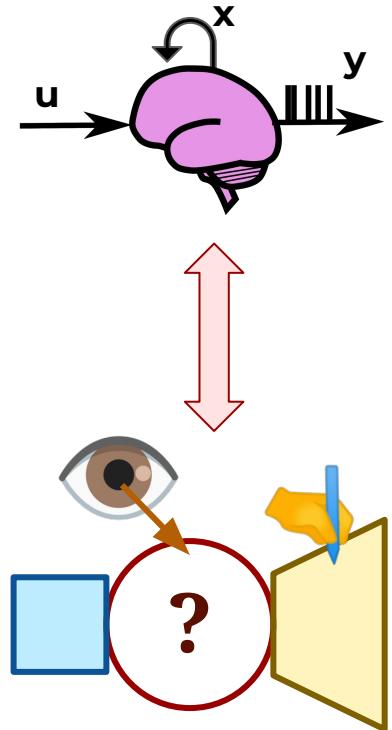
Takeaway Points

- RNNs combine **dynamics with function-approximation** capability of ANNs
- By training them to accomplish complex tasks we can **generate hypothesis** for how
- The structure of RNNs can be interpreted through the **lens of dynamical systems**
- **If outputs look similar**, some evidence for the RNN's hypothesis
 - Regardless, gives you a space to **perturb and evaluate alternate hypotheses**
- **Estimation of single-trial dynamics unlocks unprecedented access to test hypotheses**



Recurrent Neural Networks:

A model organism
for studying
computation
over time
in neuroscience

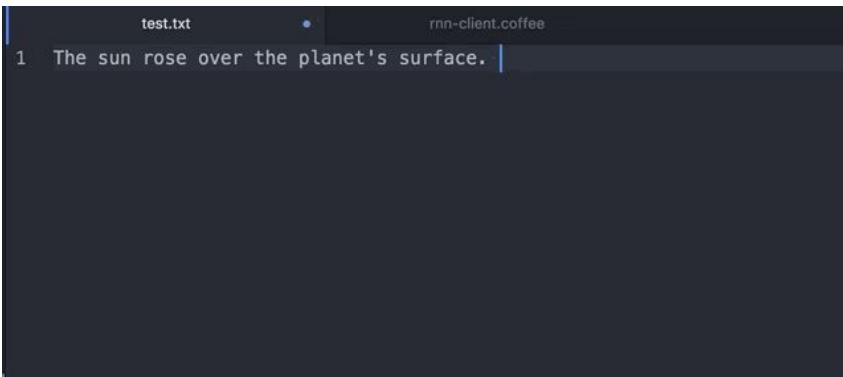


Additional Resources

Diverse application of RNNs for ML

Text generation (recipes, dungeons & dragons spells, “the onion” headlines)

Story “autocomplete” [\[demo\]](#)



Feeling writer's block? Hit **<Tab>** and an artificial neural network running in your browser will finish your sentence as if it were written by **Shakespeare**, the **US Supreme Court**, or **Tupac Shakur**.

Expressive piano performance [\[video\]](#)



Doodle alongside Sketch-RNN

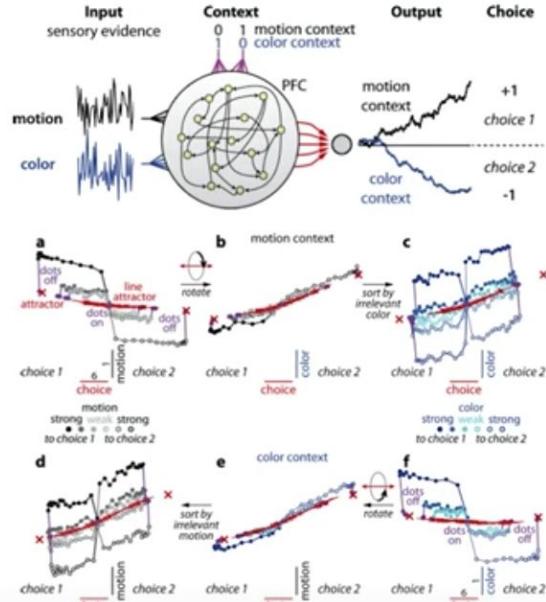
handwriting synthesis

- Never gonna give you up
- Never gonna let you down

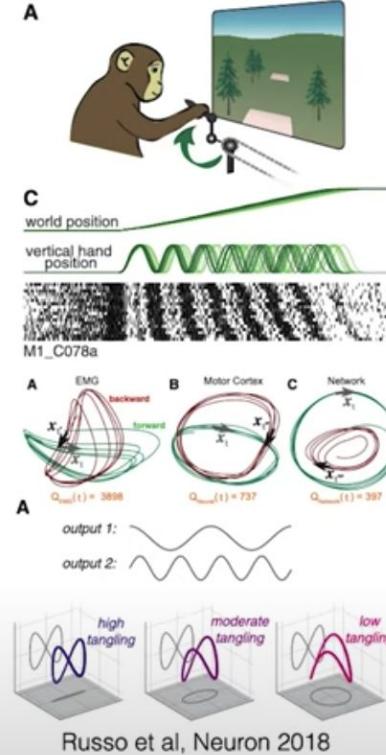
Never gonna run around and desert you

A few example RNNs from neuroscience, among many many others

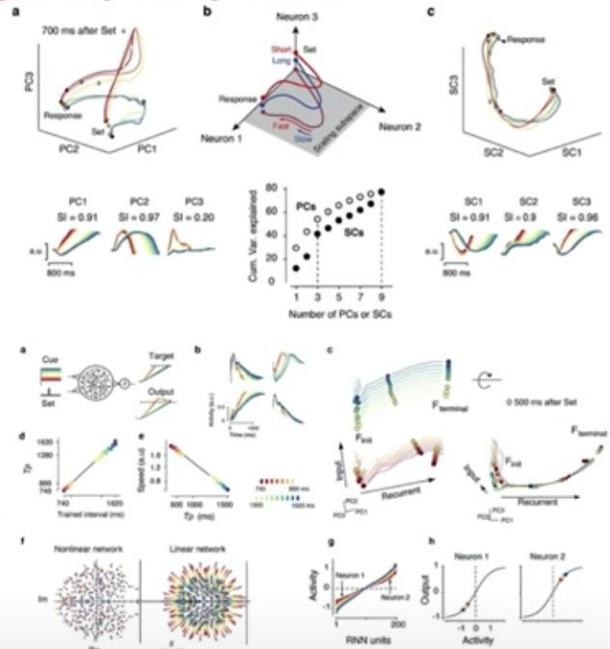
“Artificial Neural Networks for Neuroscientists: A Primer”
 (2020) Yang & Wang



Mante, Sussillo, et al, Nature, 2013



Russo et al, Neuron 2018



Wang*, Narain* et al, Nat Neuro 2018

Critical reading: Barak, 2017; Vogels, Rajan & Abbott, 2005; Yang & Wang Primer, Neuron 2020

“Context-dependent computation by recurrent dynamics in prefrontal cortex”
 (2013) Mante, Sussillo et al.

“Motor cortex embeds muscle-like commands in an untangled population response” (2018)
 Russo et al.

“Flexible timing by temporal scaling of cortical responses” (2017) Wang et al.

Applications of RNNs to understand the brain

"Task representations in neural networks trained to perform many cognitive tasks" (2019) Yang et al.

"Flexible Sensorimotor Computations through Rapid Configuration of Cortical Dynamics" (2018) Remington et al.

"Context-dependent computation by recurrent dynamics in prefrontal cortex" (2013) Mante, Sussillo et al.

"The authors provide a detailed example of how the mechanisms of a dynamical computation can be discovered in an RNN, providing a powerful hypothesis for the underlying mechanism in a complex cortical circuit."

"A neural network that finds a naturalistic solution for the production of muscle activity" (2015) Sussillo et al.

"Hierarchical recurrent state space models reveal discrete and continuous dynamics of neural activity in C. elegans" (2019) Linderman et al.

"Inferring single-trial neural population dynamics using sequential auto-encoders" (2018) Pandarinath et al.

"Emergence of grid-like representations by training recurrent neural networks to perform spatial localization" (2018) Cueva & Wei

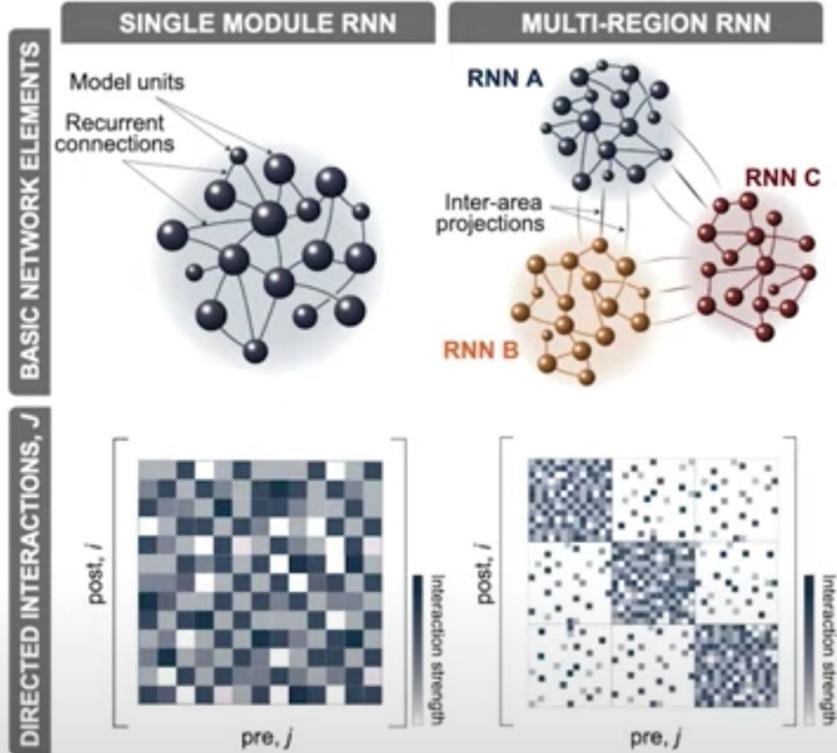
"Task-Driven Convolutional Recurrent Models of the Visual System" (2018) Nayebi

"Predictive coding arises in RNNs due to energy constraints" (2021) Ali et al.

"Motor cortex activity across movement speeds is predicted by network-level strategies for generating muscle activity" (2021) Saxena et al.

"Motor cortex embeds muscle-like commands in an untangled population response" (2018) Russo et al.

1. From single module- to multi-region RNNs



21 Rajan & Abbott, with Sompolinsky, 2010a-c, 09, 07, 06, 05

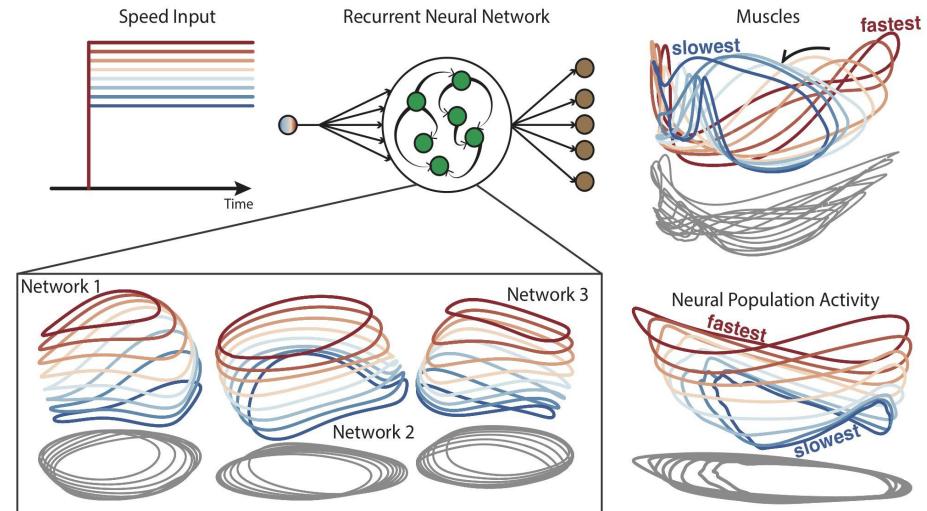
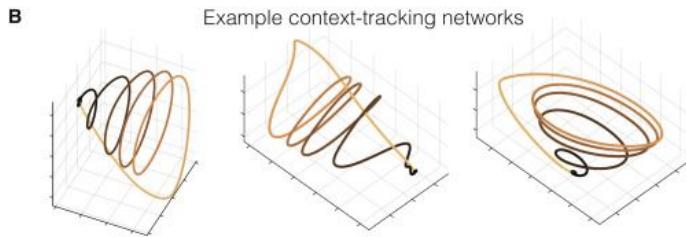
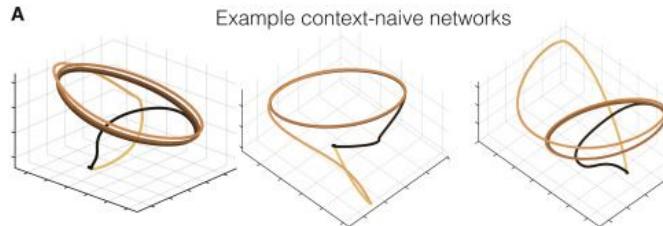
["Rethinking brain-wide interactions through multi-region 'network of networks' models"](#) (2020) Perich & Rajan

["Inferring brain-wide interactions using data-constrained recurrent neural network models"](#) (2020) Perich et al.

["COSYNE 2021 - Tutorial on RNN \(Part 2\)"](#) by Kanaka Rajan

["Stimulus-dependent suppression of chaos in recurrent neural networks"](#) (2010) Rajan, Abbott, Sompolinsky

RNNs for generating & testing hypotheses



"Neural trajectories in the supplementary motor area and primary motor cortex exhibit distinct geometries, compatible with different classes of computation" (2019)
Russo et al.

"Motor cortex activity across movement speeds is predicted by network-level strategies for generating muscle activity"
(2021) Saxena et al.

A mostly complete chart of
Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

- Backfed Input Cell
- Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- △ Different Memory Cell
- Kernel
- Convolution or Pool

Perceptron (P)



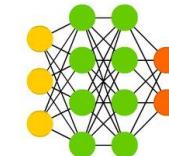
Feed Forward (FF)



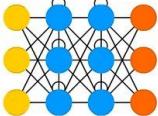
Radial Basis Network (RBF)



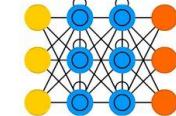
Deep Feed Forward (DFF)



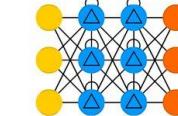
Recurrent Neural Network (RNN)



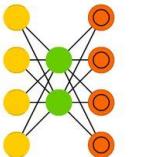
Long / Short Term Memory (LSTM)



Gated Recurrent Unit (GRU)



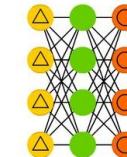
Auto Encoder (AE)



Variational AE (VAE)



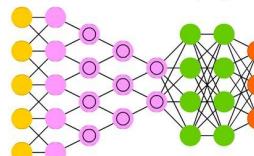
Denoising AE (DAE)



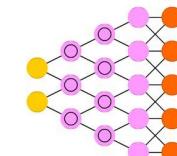
Sparse AE (SAE)



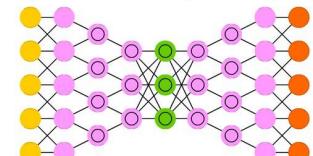
Deep Convolutional Network (DCN)



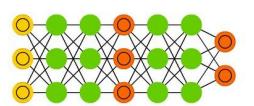
Deconvolutional Network (DN)



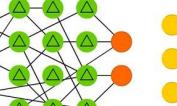
Deep Convolutional Inverse Graphics Network (DCIGN)



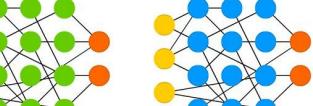
Generative Adversarial Network (GAN)



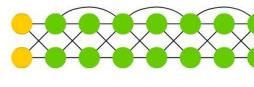
Liquid State Machine (LSM) Extreme Learning Machine (ELM)



Echo State Network (ESN)



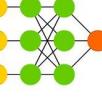
Deep Residual Network (DRN)



Kohonen Network (KN)



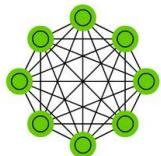
Support Vector Machine (SVM)



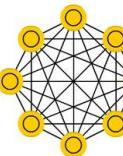
Neural Turing Machine (NTM)



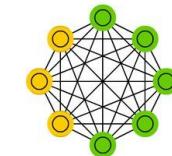
Markov Chain (MC)



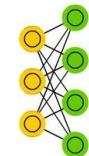
Hopfield Network (HN)



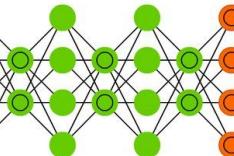
Boltzmann Machine (BM)



Restricted BM (RBM)



Deep Belief Network (DBN)



High Level Reviews of RNNs in Neuro

“Deep Reinforcement Learning and Its Neuroscientific Implications” (2020) Botvinick et al.

“A deep learning framework for neuroscience” (2019) Richards et al.

“Opportunities and obstacles for deep learning in biology and medicine” (2017) Ching et al.

“Recurrent neural networks as versatile tools of neuroscience research” (2017) Barak

“Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing” (2015) Kriegeskorte

“Artificial Neural Networks for Neuroscientists: A Primer” (2020) Yang & Wang

Interpretation of RNNs in Neuro

“COSYNE 2021 - Tutorial on RNN (Part 1)” by Kanaka Rajan

What can we learn about the brain from Recurrent Neural Networks?

Lecture 1: Foundational elements of recurrent neural network (RNN) models



1. Building blocks of neural network models
2. Recurrent part of RNNs
3. Linear networks and non-linearity
4. Dynamical regimes

Lecture 2: Applications of RNNs in the field of neuroscience



1. Training RNNs to ‘do something’
2. Reverse-engineering RNNs
3. Where ^ is useful for us neuroscientists
4. Looking to the future

En route to a Theory of Cognition as Dynamics

Where are we?

Linear RNNs

$$\dot{x} = -x + Jx$$

Two patterns, stable one → TFP, 0



Nonlinear RNNs, random J

$$\dot{x} = -x + J\phi(x)$$

Tame instability + get rich dynamics



Driven nonlinear RNNs, random J

$$\dot{x} = -x + J\phi(x) + h$$

Input-driven activity on rich ongoing background

Without chaotic sensitivity to initial conditions, reliable responses

What's the problem?

Activity that does not decay is unstable, i.e., blows up



Dynamics are chaotic, can't reliably get the same pattern twice



How do we solve it?

Nonlinearities $\phi(x) = \tanh(x)$

External inputs to control chaos
 $h = I \cos(\omega t + \theta)$

Train nonlinear RNNs to get a specialized J: $J_{\text{rand}} \rightarrow J_{\text{trained}}$



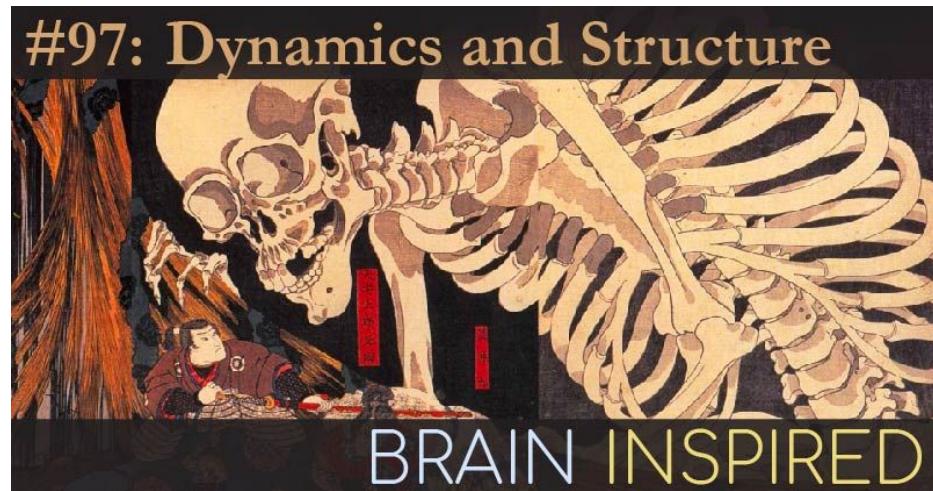
Interpretation of RNNs in Neuro

"Learning interpretable continuous-time models of latent stochastic dynamical systems" (2019)
Duncker et al.

"Interrogating theoretical models of neural computation with deep inference" (2019) Bittner

"Opening the Black Box - using Dynamics to understand RNNs (to understand the brain)" (2013)
Sussillo, Barak

"Dynamics & Structure" Sussillo & Barak
@ Brain Inspired Podcast



Tutorials / How-To for an RNN

“The Unreasonable Effectiveness of Recurrent Neural Networks” -

Andrej Karpathy

- Easy to read
- Lots of fun examples
- Has code for RNNs for text generation

“Practicum: RNN and LSTM architectures” - Alfredo Canziani & Yann

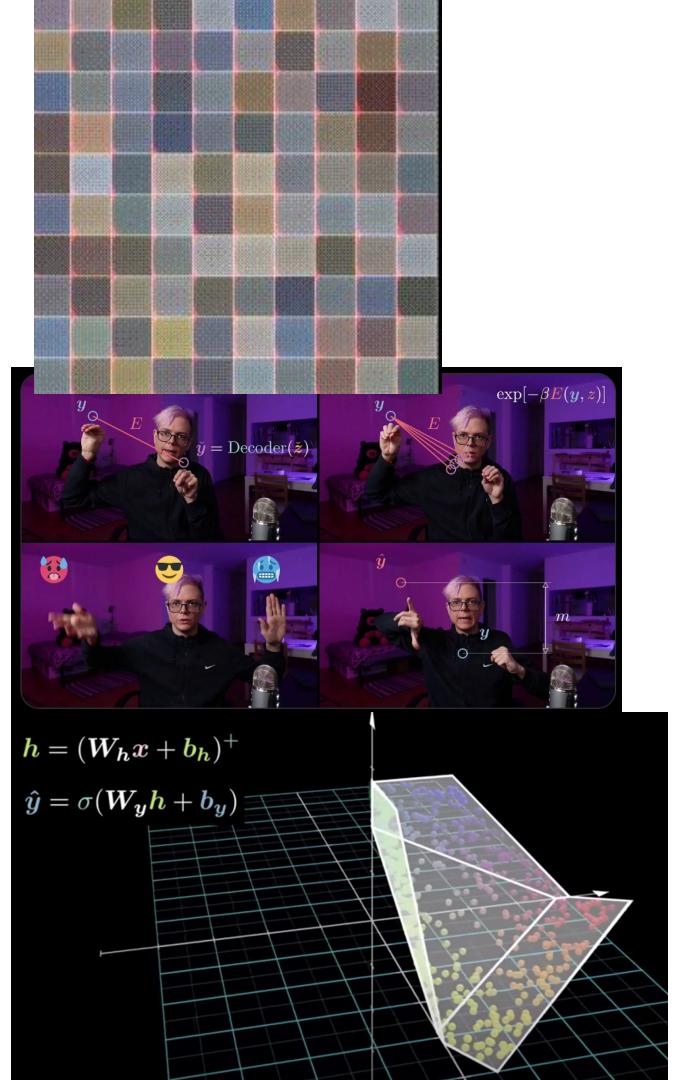
LeCunn

- Focus on visual explanation
- Has associated code notebooks

“The Neural Network, A Visual Introduction” vcubingx

“Deep Learning” 3Blue1Brown

“What is backpropagation really doing?” 3Blue1Brown

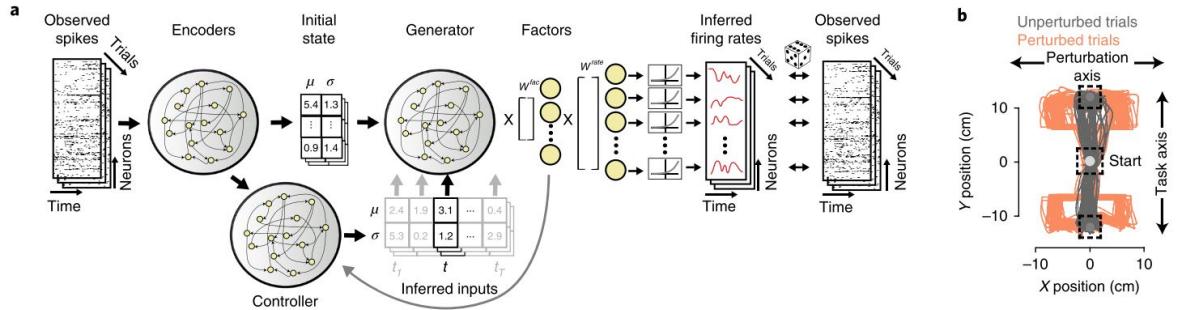


Input-driven dynamics in RNNs

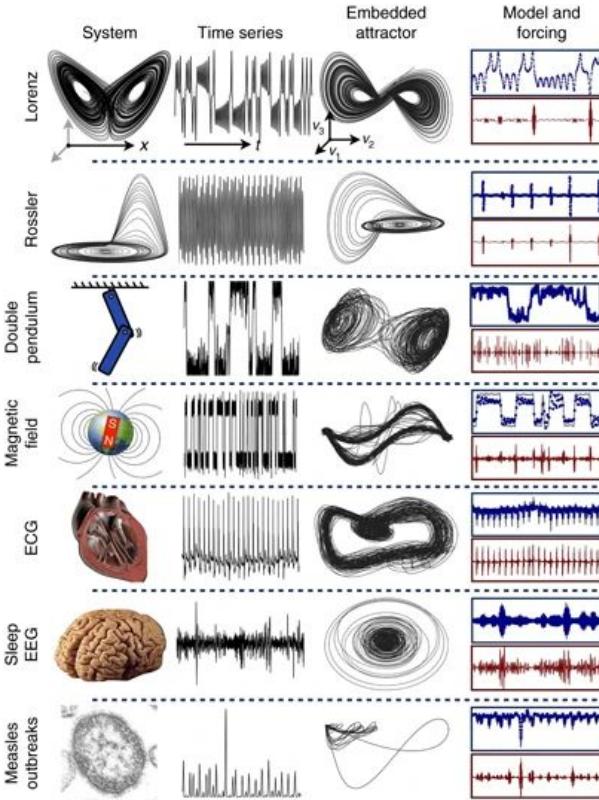
"Chaos as an intermittently forced linear system" (2017)

Brunton et al.

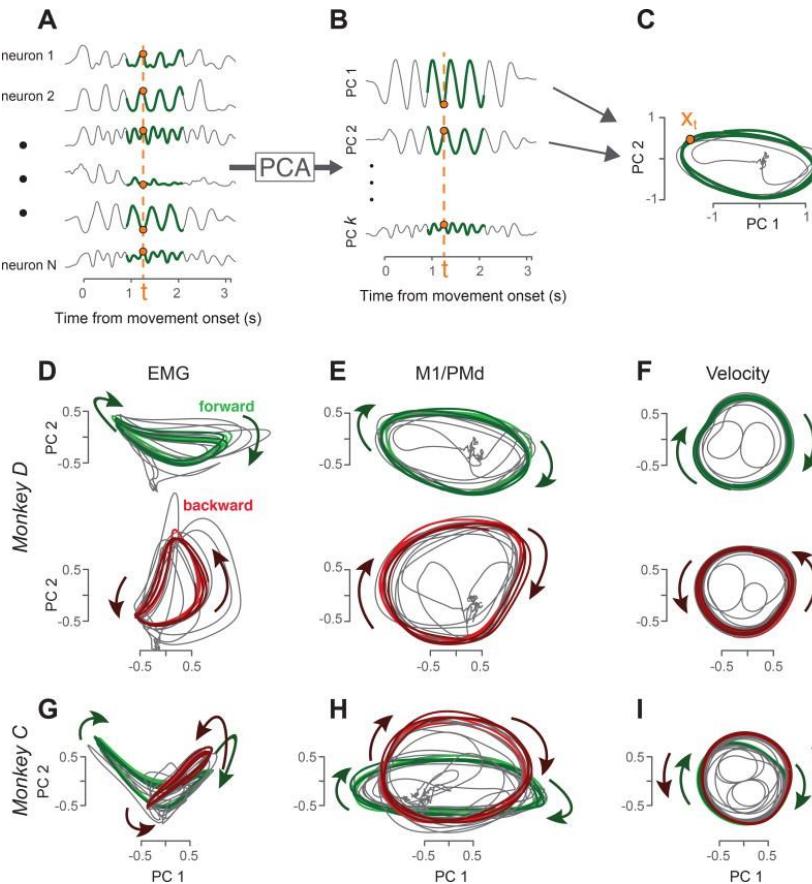
"Echo State Property Linked to an Input: Exploring a Fundamental Characteristic of Recurrent Neural Networks"
(2013) Manjunath & Jaeger



LFADS uncovers the presence, identity, and timing of unexpected perturbations in the cursor jump task



Tangling Metric



$$Q(t) = \max_{t'} \frac{\|\dot{x}_t - \dot{x}_{t'}\|^2}{\|x_t - x_{t'}\|^2 + \epsilon}$$

“where x_t is the neural state at time t : i.e., a vector containing the neural responses at that time, \dot{x}_t is the temporal derivative of the neural state, $\|\cdot\|$ is the Euclidean norm, and ϵ is a small constant that prevents division by zero (*methods*). $Q(t)$ becomes high if there exists a state at a different time, t' , that is similar but associated with a dissimilar derivative. For example, tangling is high if two trajectories pass through similar points but in different directions”

“Motor cortex embeds muscle-like commands in an untangled population response” (2018) Russo et al.