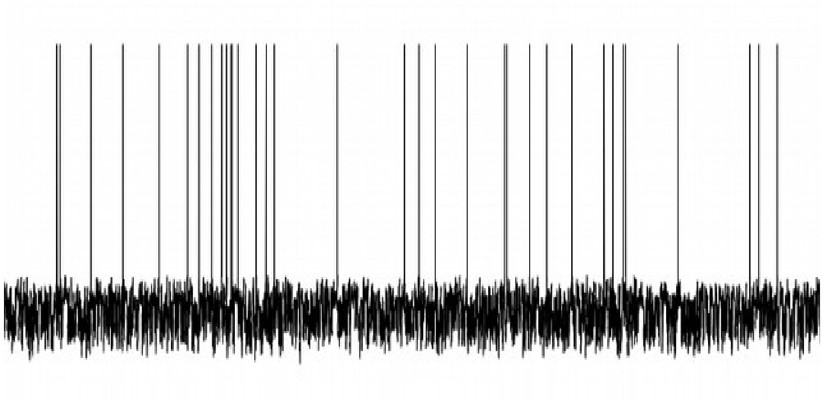
Computation in spiking neural networks
Opportunities and challenges

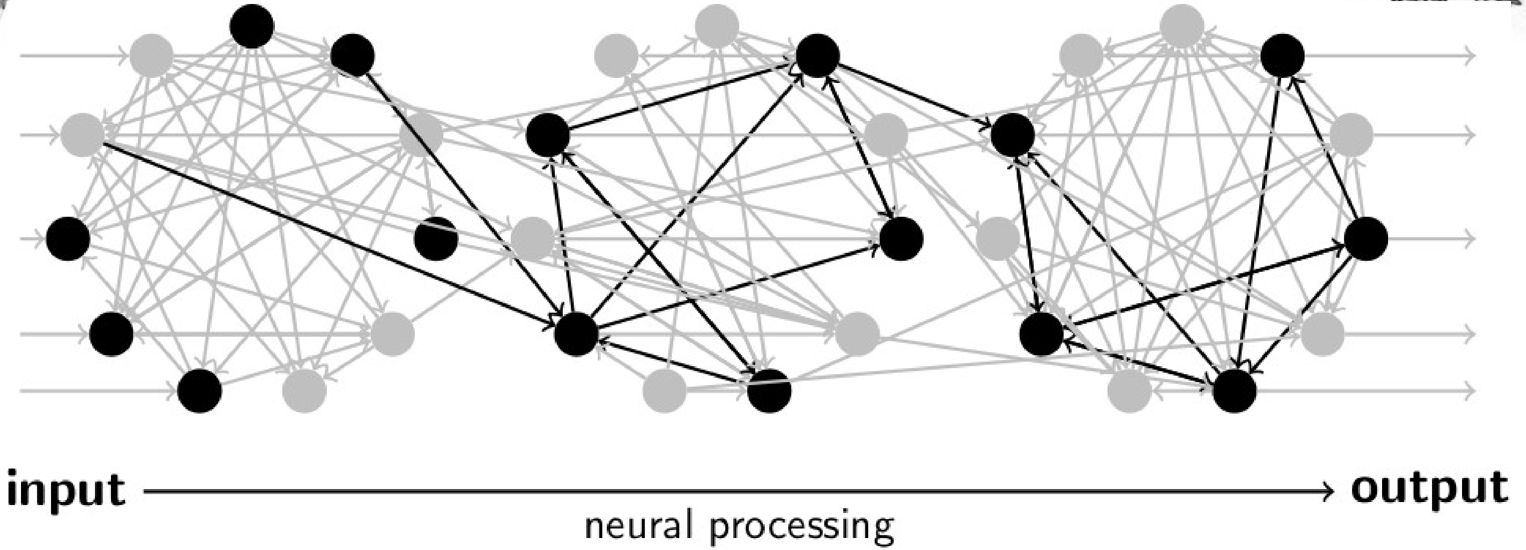
Friedemann Zenke
fzenke.net



Animals compute with spiking neural networks

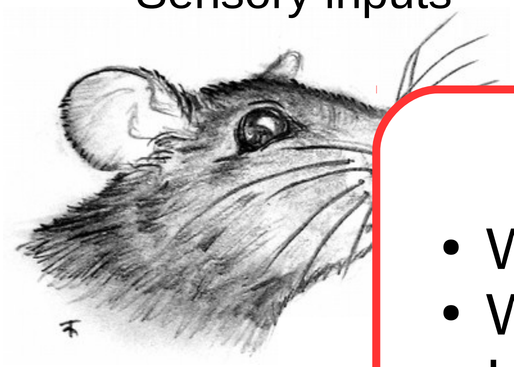
Sensory inputs

Behavior



Animals compute with spiking neural networks

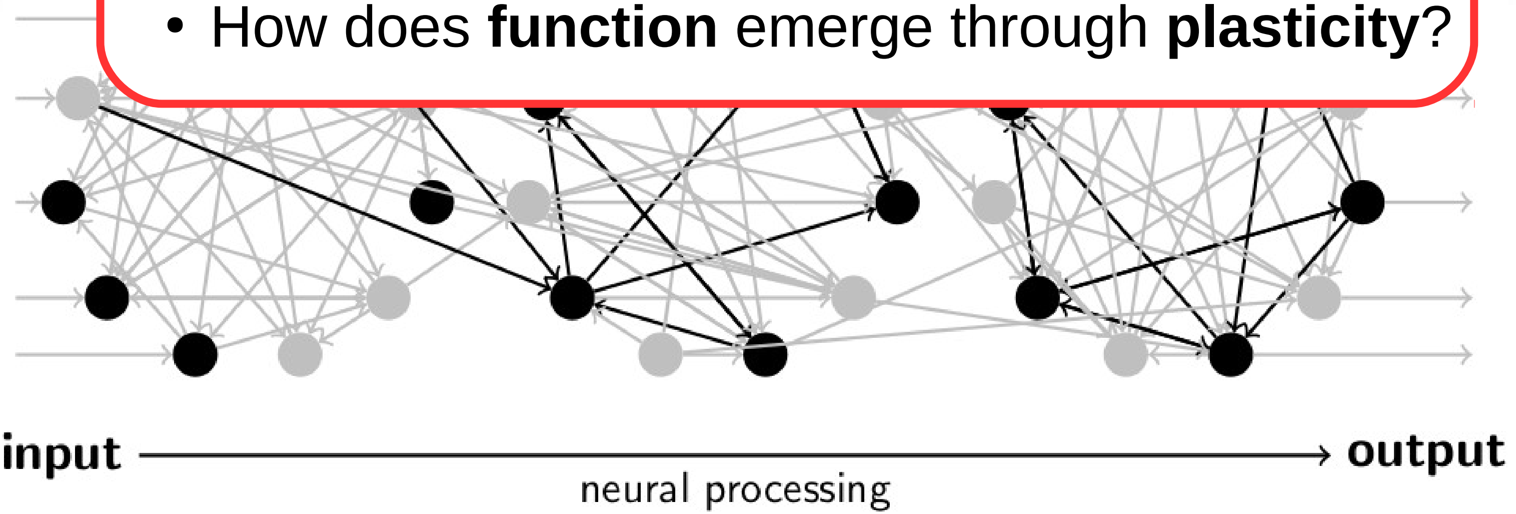
Sensory inputs

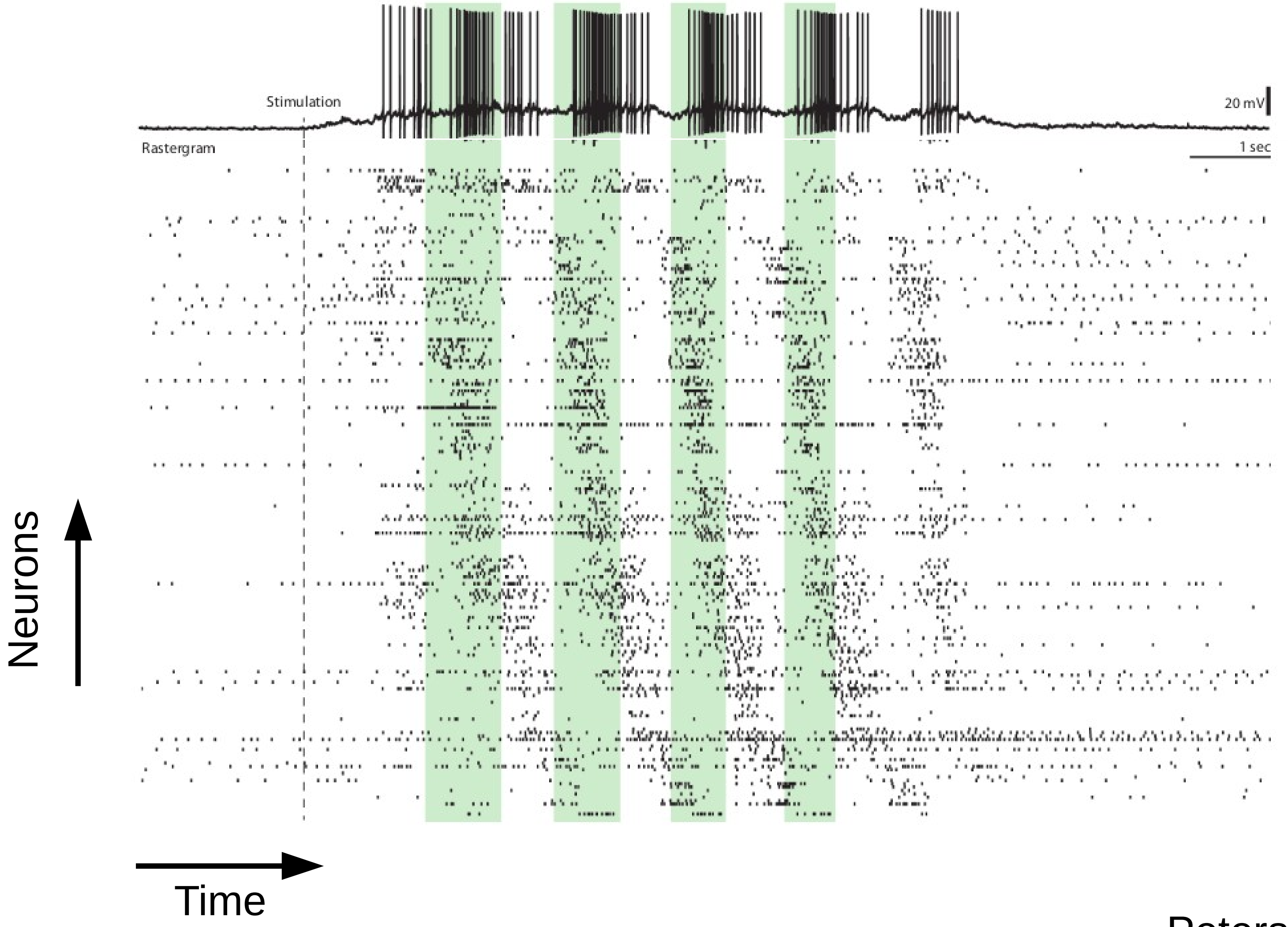


Behavior

Questions

- Which **structures** underlie network **functions**?
- What **functions** are implemented?
- How does **function** emerge through **plasticity**?





Petersen & Berg (2016)

“Why spikes?”

Google

Why spikes?

All | Shopping | Images | News | Videos | More

About 128,000,000 results (0.34 seconds)

The main benefit to wearing running **spikes** is that they
you traction even in wet or otherwise harsh conditions.
exist. Smaller, sharper **spikes**, such as metal needle **sp**
standard tracks for short distances.

[Do Spikes Make You Run Faster? | Livestrong.com](https://www.livestrong.com/article/509013-do-spikes-make-you-run-faster/)

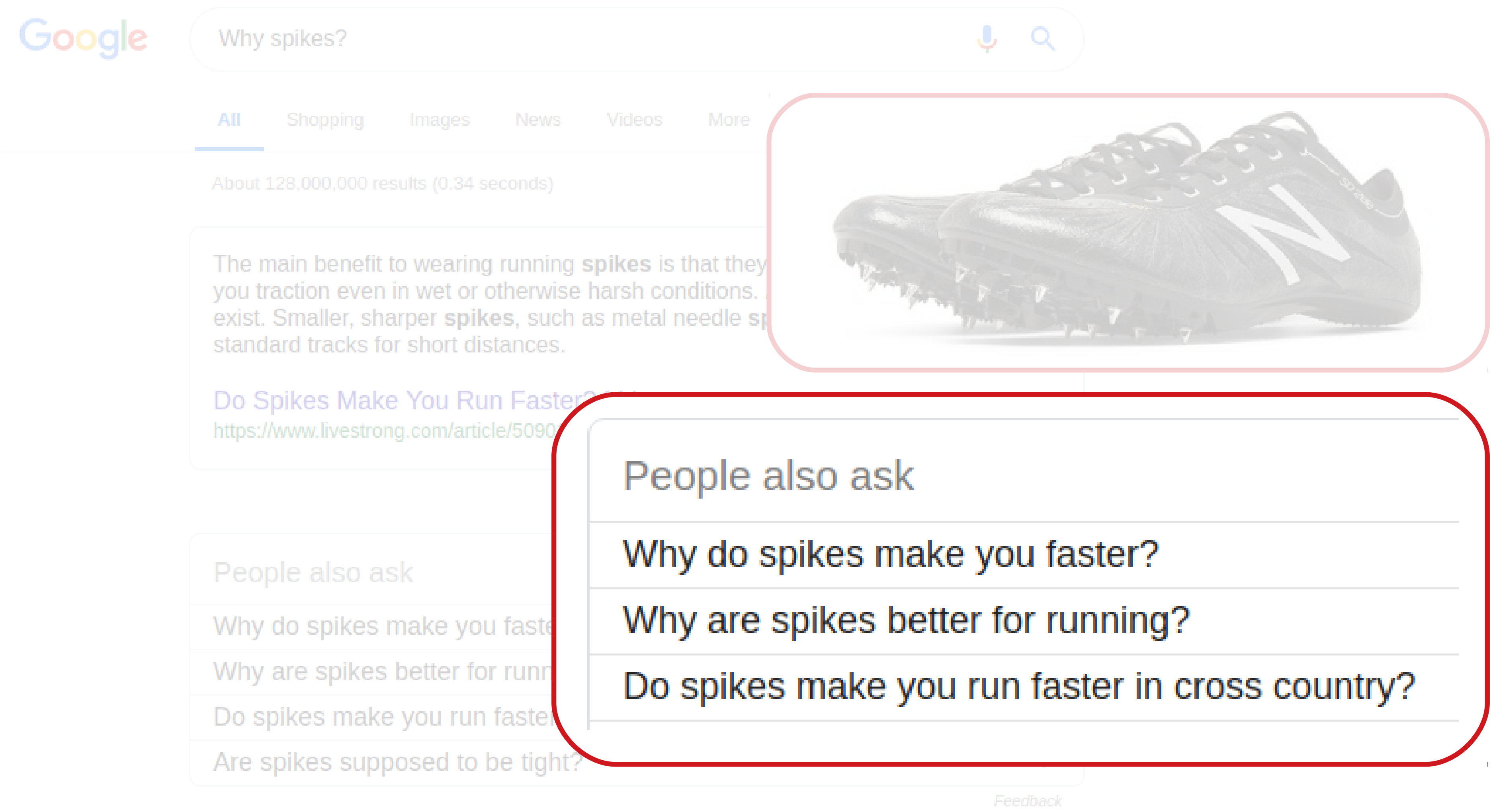
https://www.livestrong.com/article/509013-do-spikes-make-you-run-faster/

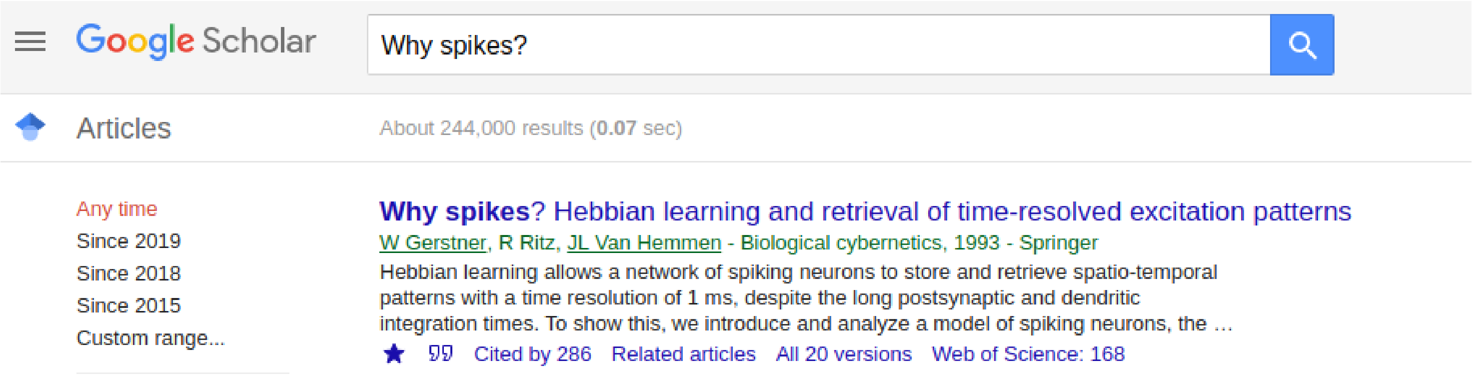
About this result   Feedback

People also ask

- Why do spikes make you faster?
- Why are spikes better for running?
- Do spikes make you run faster in cross country?
- Are spikes supposed to be tight?

Feedback


Google
Why spikes?
All
Shopping
Images
News
Videos
More
About 128,000,000 results (0.34 seconds)
The main benefit to wearing running spikes is that they
you traction even in wet or otherwise harsh conditions.
exist. Smaller, sharper spikes, such as metal needle sp
standard tracks for short distances.
Do Spikes Make You Run Faster?
https://www.livestrong.com/article/5090
People also ask
Why do spikes make you faster?
Why are spikes better for running?
Do spikes make you run faster in cross country?
Are spikes supposed to be tight?
Feedback



Why spikes? Hebbian learning and retrieval of $\boxed{\text{time-resolved}}$ excitation patterns

Wulfram Gerstner, Raphael Ritz, J. Leo van Hemmen

Physik-Department der TU München, James-Franck-Strasse, D-85747 Garching bei München, Germany

Received: 24 October 1992/Accepted in revised form: 21 April 1993

Gerstner, Ritz & van Hemmen (1993)

Opportunities

Time

Temporal processing

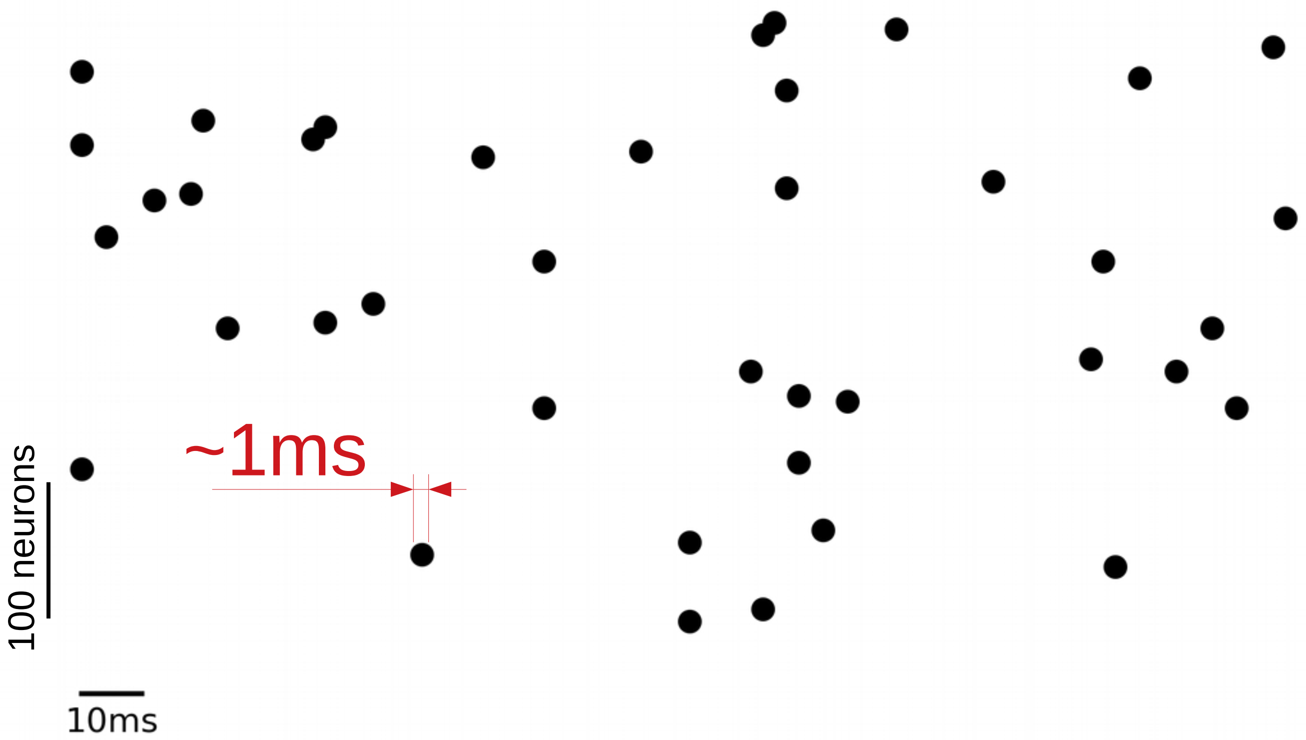
Processing speed

e.g.: visual processing < 150ms

Thorpe, Fize, and Marlot (1996)

Energy efficient

~20 watts per brain



Opportunities

Time

Temporal processing

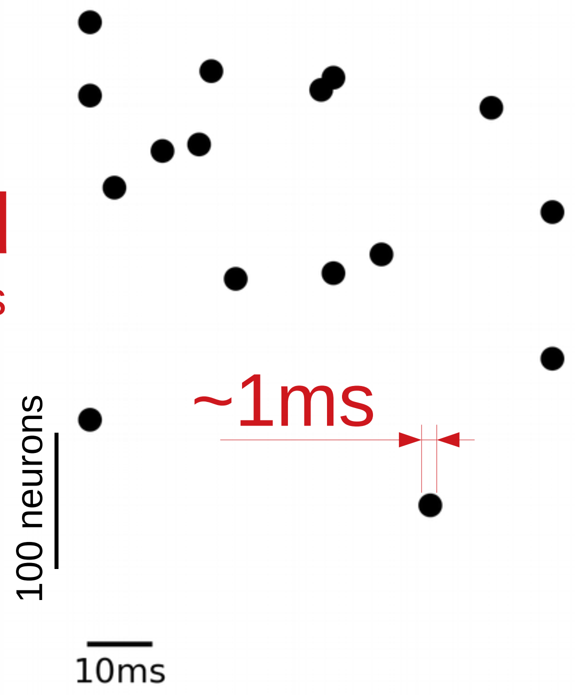
Processing speed

e.g.: visual processing < 150ms

Thorpe, Fize, and Marlot (1996)

Energy efficient

~20 watts per brain



Challenges (modeling)

Time

Input-output

Which function

Digital / binary

Challenges (modeling)

Time

Input-output

Which function

Digital / binary

Outline

- Train spiking networks* on spatiotemporal tasks
  (* multi-layer and recurrent)
- **Method:** Surrogate gradients
- **Application:** Classification
- Bio-plausible reductions

Challenges (modeling)

Time

Input-output

Which function

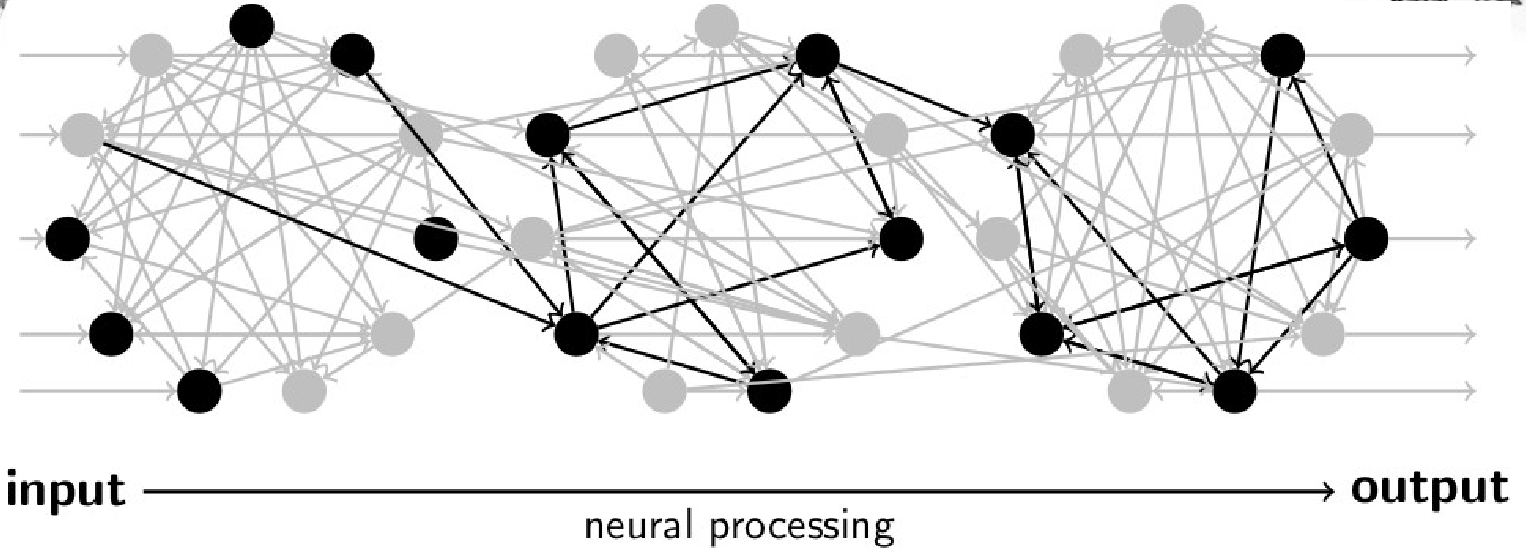
Digital / binary

How to (compute with) spikes?

Towards spiking network models which compute

Sensory inputs

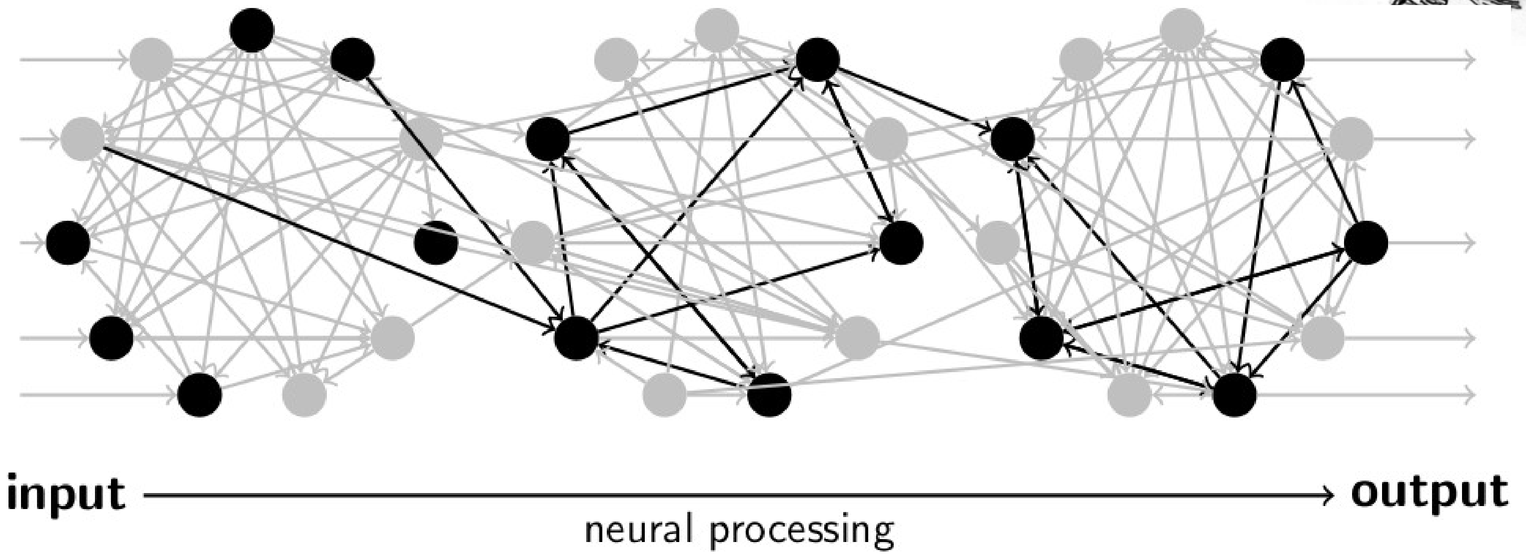
Behavior



Towards spiking network models which compute

Behavior

1) Input
(spatiotemporal)

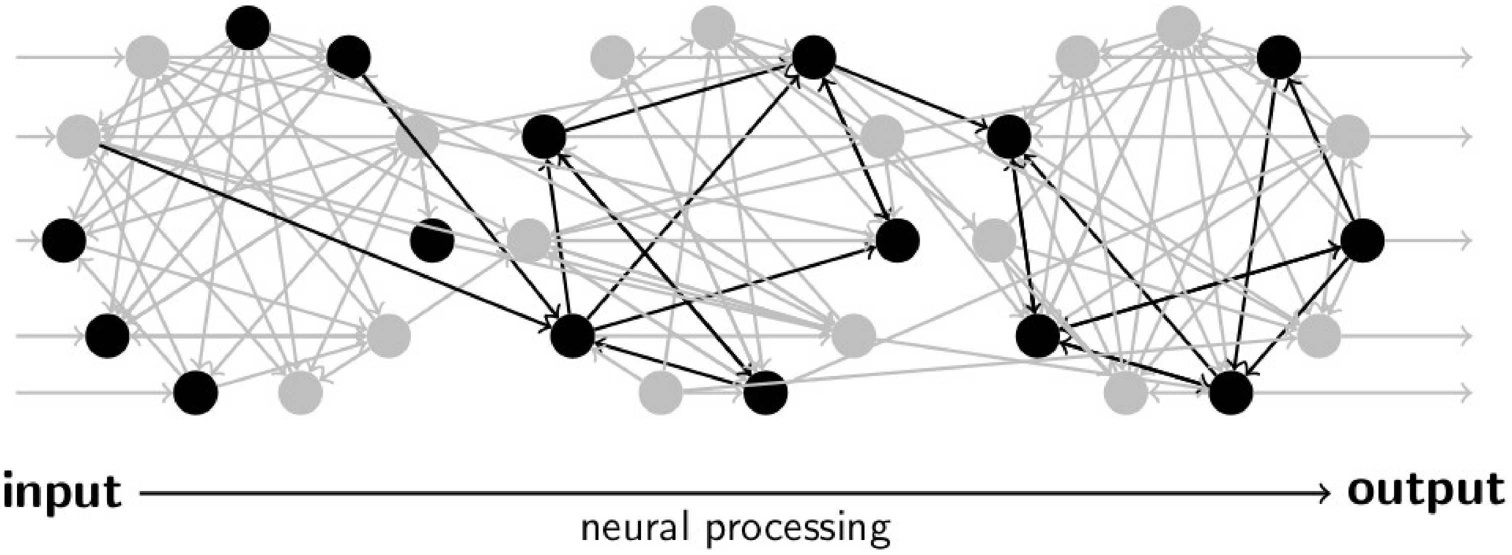


Towards spiking network models which compute

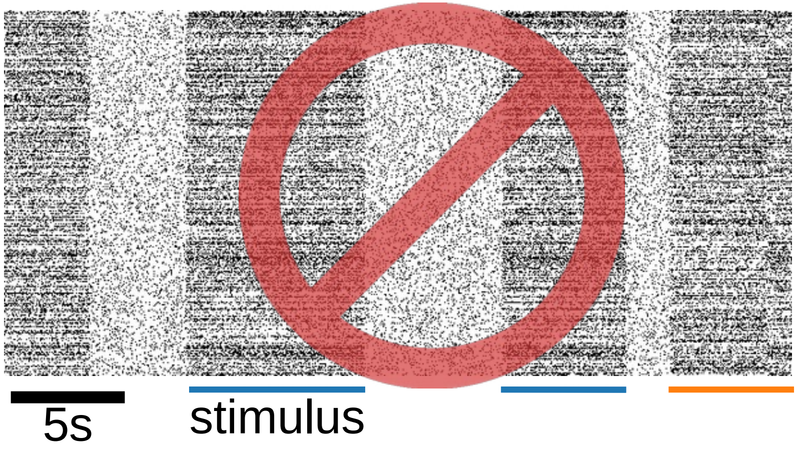
1) Input
(spatiotemporal)

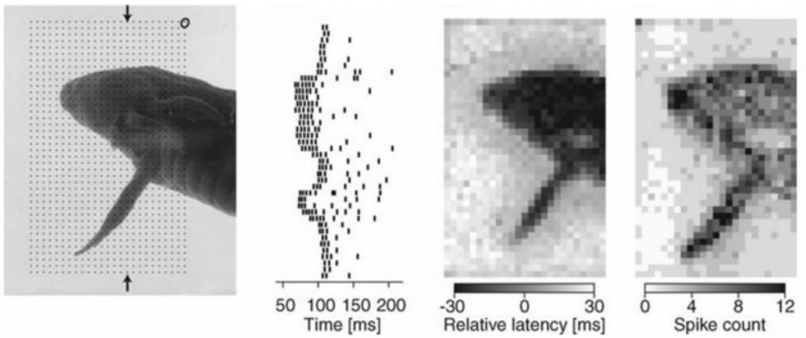
3) Adjust weights
(supervised learning)

2) Output
(classification)

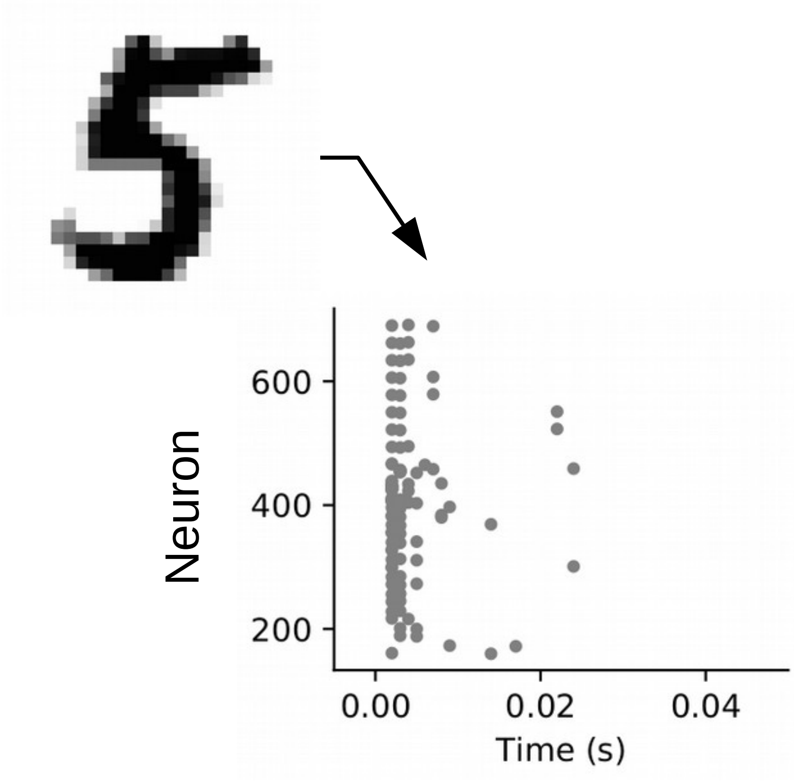


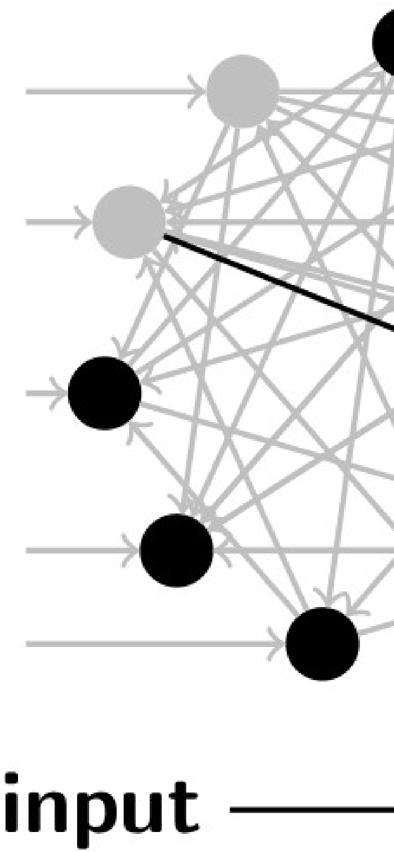
Input: Spatiotemporal spike patterns



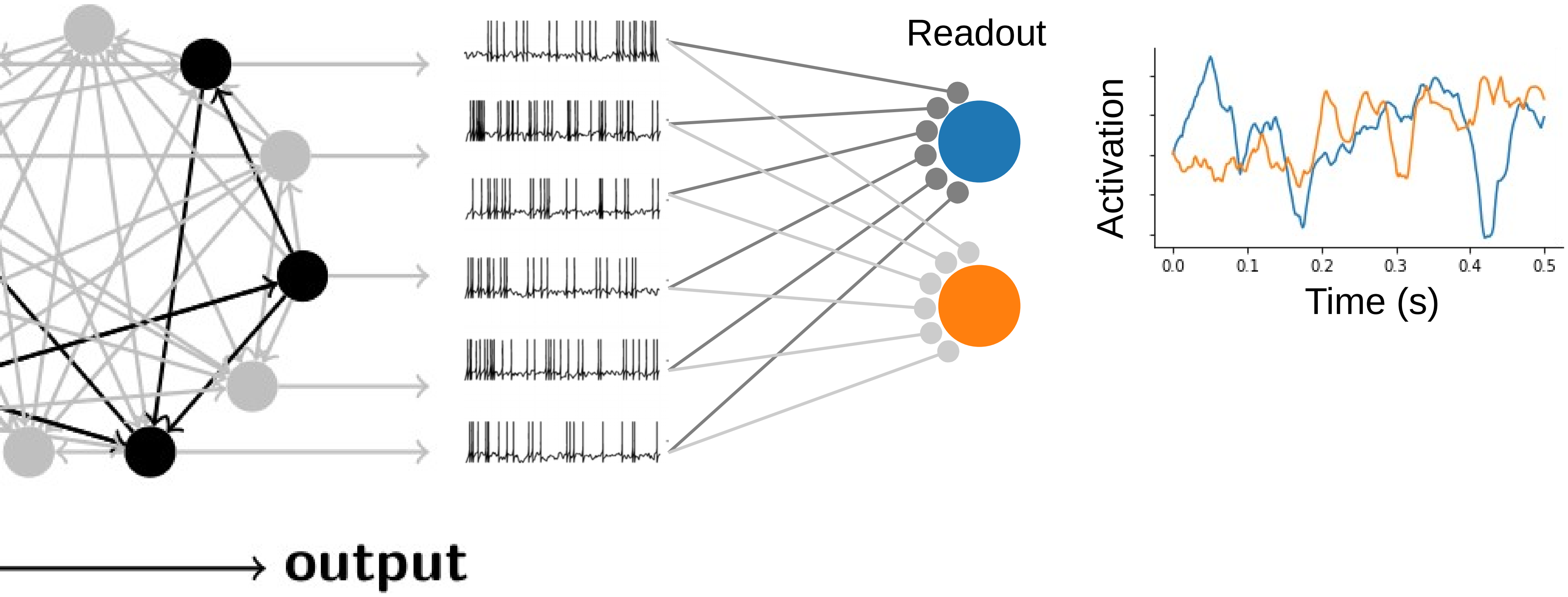


Gollisch & Meister (2008)

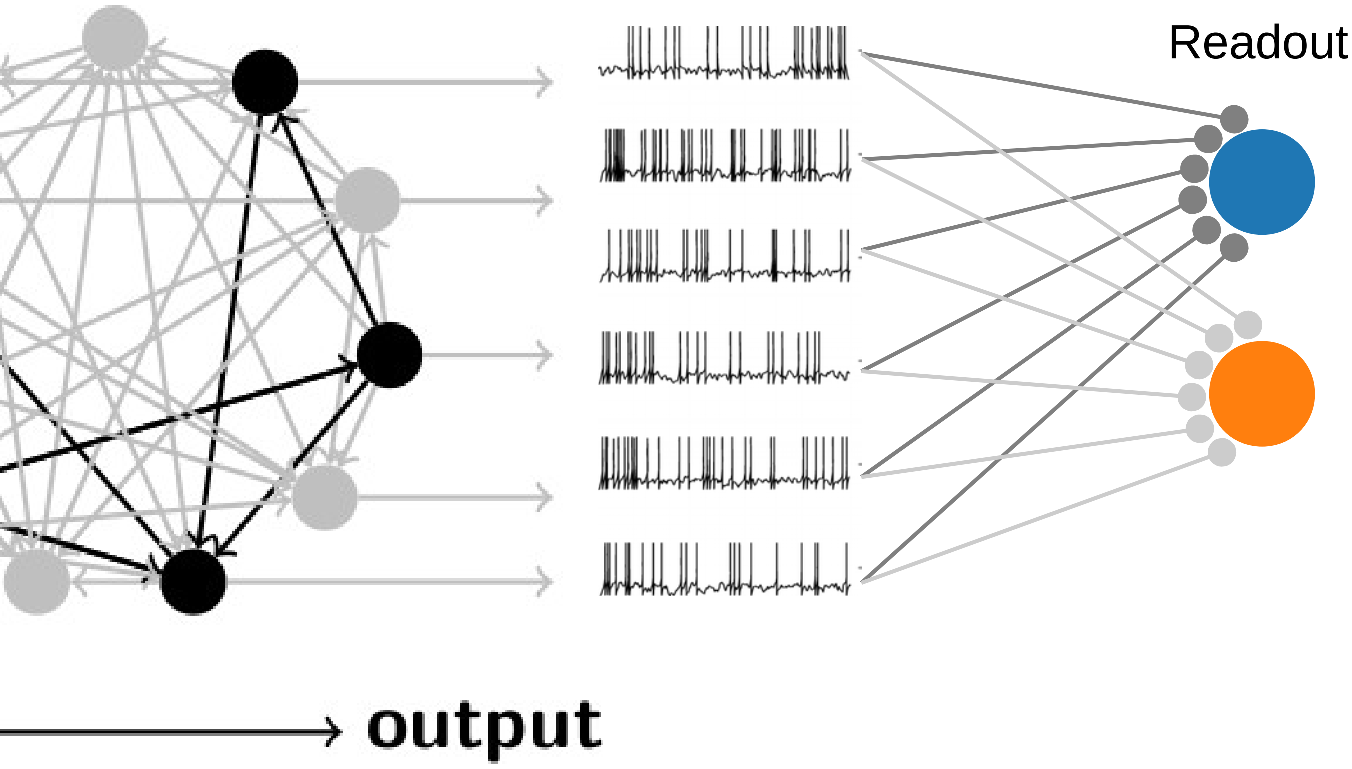


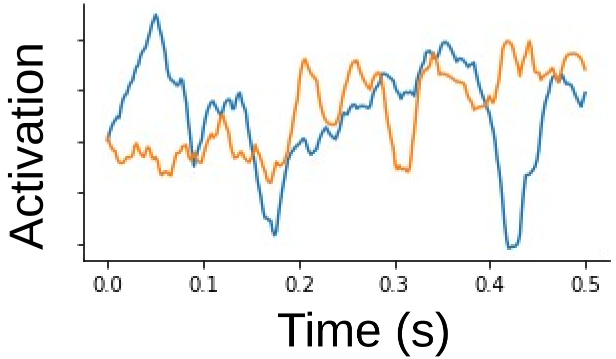


Output: Linear combination of filtered output spike trains



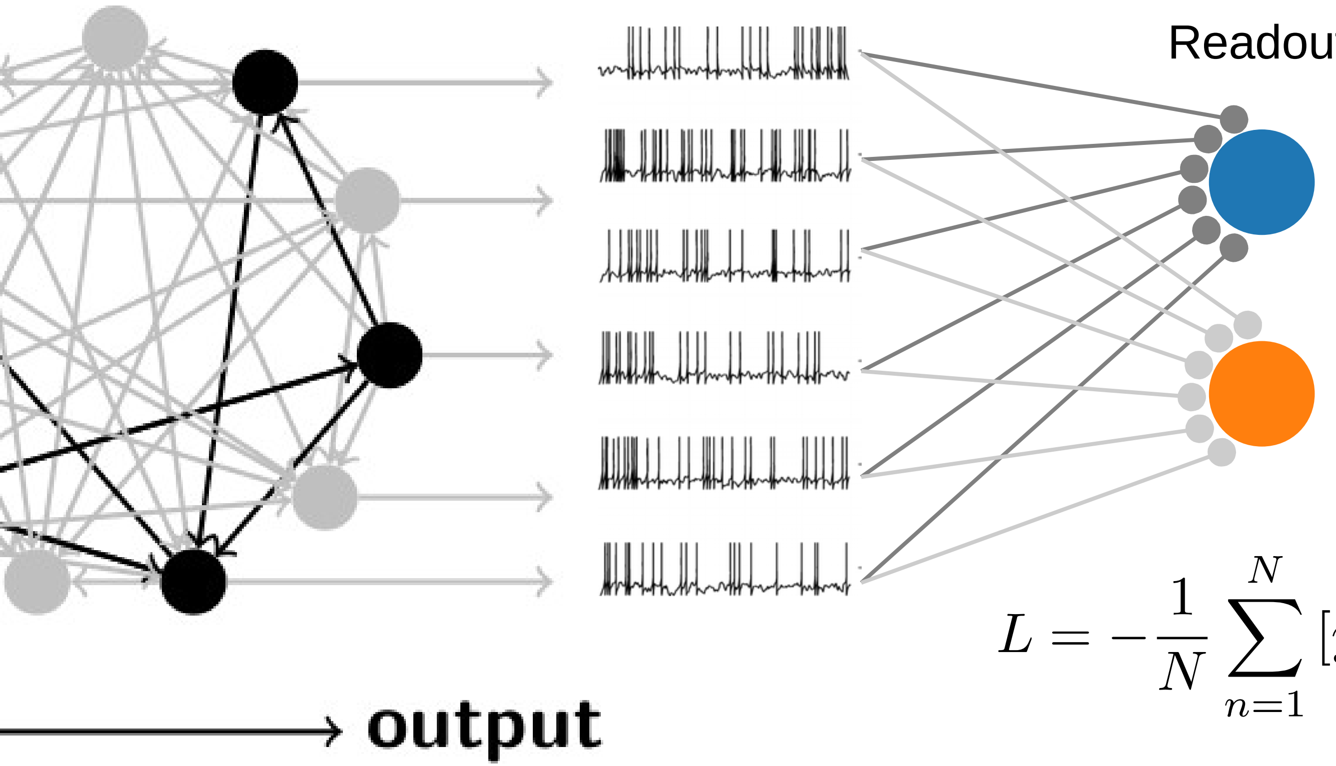
Output: Linear combination of filtered output spike trains

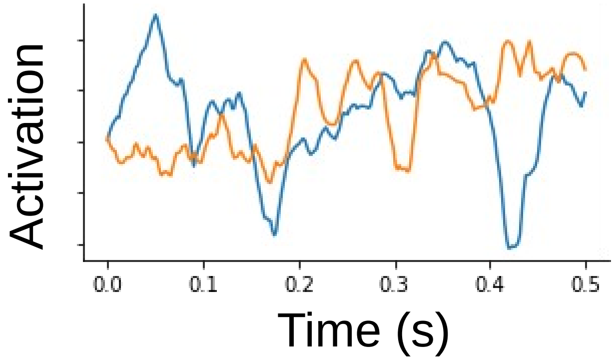




$$p_i = \operatorname*{softmax}_i \left(\max_t U_i(t) \right)$$

Output: Linear combination of filtered output spike trains





$$p_i = \operatorname*{softmax}_i \left(\boxed{\max_t U_i(t)} \right)$$

$$L = -\frac{1}{N} \sum_{n=1}^{N} \left[y_n \log p_n + (1 - y_n) \log(1 - p_n) \right]$$

Max over time idea from Tempotron

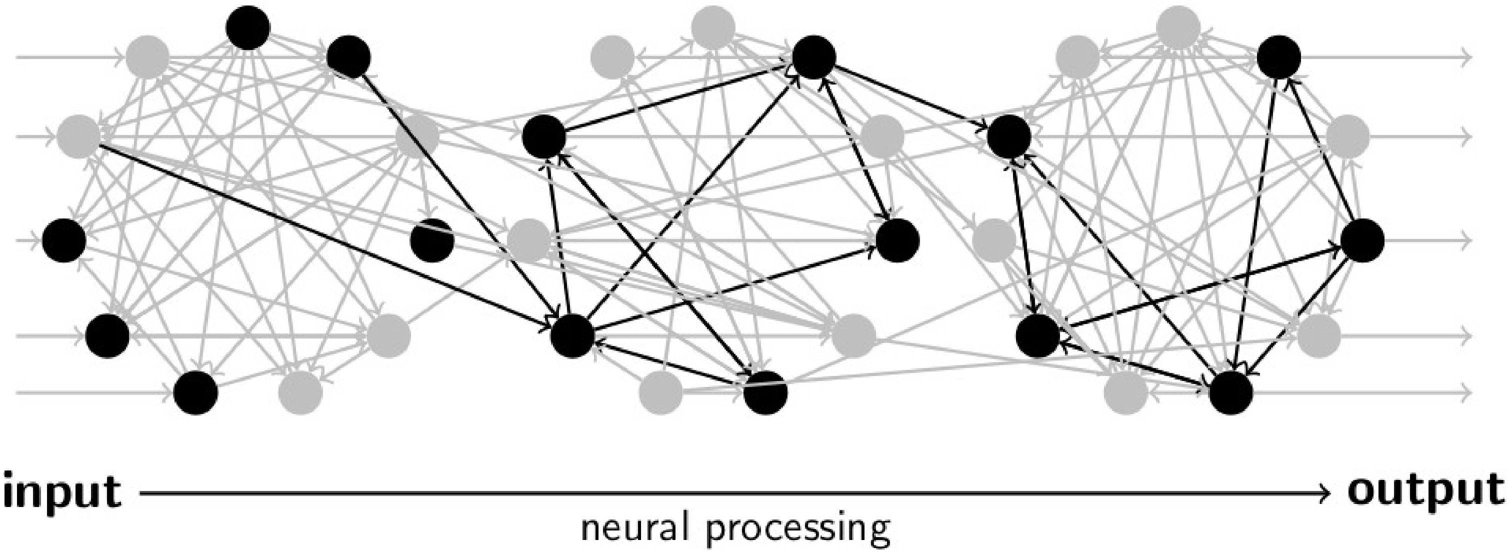
Gütig & Sompolinsky (2006); Gütig (2016)

Towards spiking network models which compute

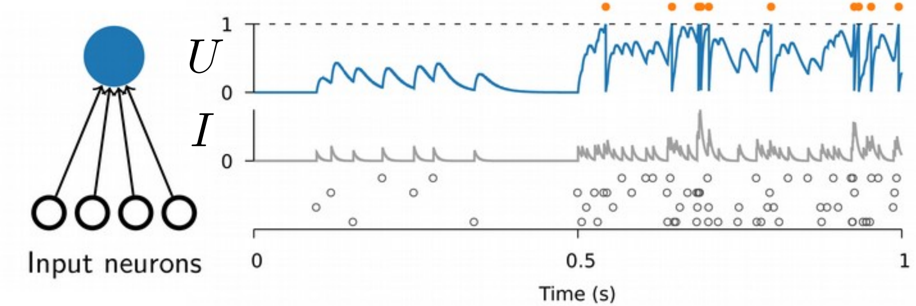
1) Input
(spatiotemporal)

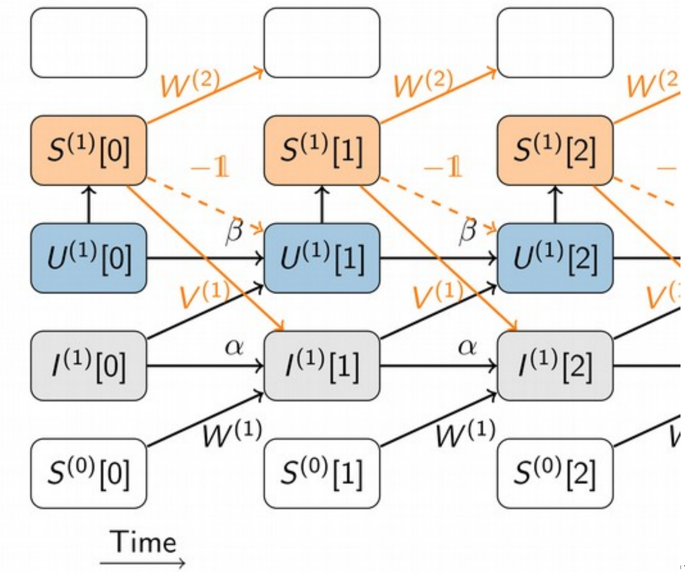
3) Adjust weights
(supervised learning)

2) Output
(classification)



Thinking of spiking neurons as RNNs





- Spiking neurons & networks are RNNs
- Known training procedures for networks **with hidden units**
- Several groups have realized this:
  - Esser, Merolla, Arthur, Cassidy, Appuswamy, Andreopoulos, Berg, McKinstry, Melano, Barch, et al. (2016)
  - Huh & Sejnowski (2018)
  - Shrestha & Orchard (2018)
  - Bellec, Salaj, Subramoney, Legenstein, and Maass (2018)
  - Neftci, Mostafa, & Zenke (2019). ArXiv

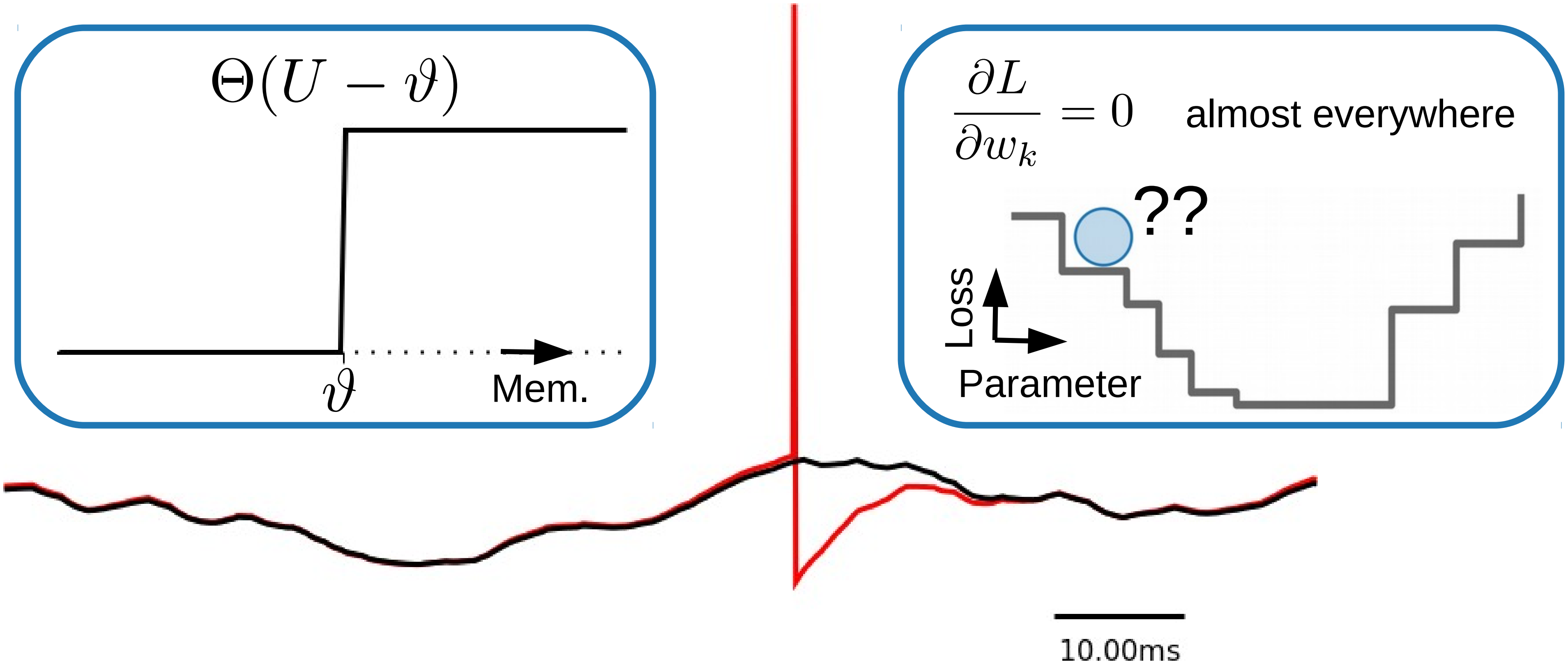
$$S_i^{(1)}[n] = \boxed{\Theta\left(U_i^{(1)}[n] - \vartheta\right)}$$

Problem

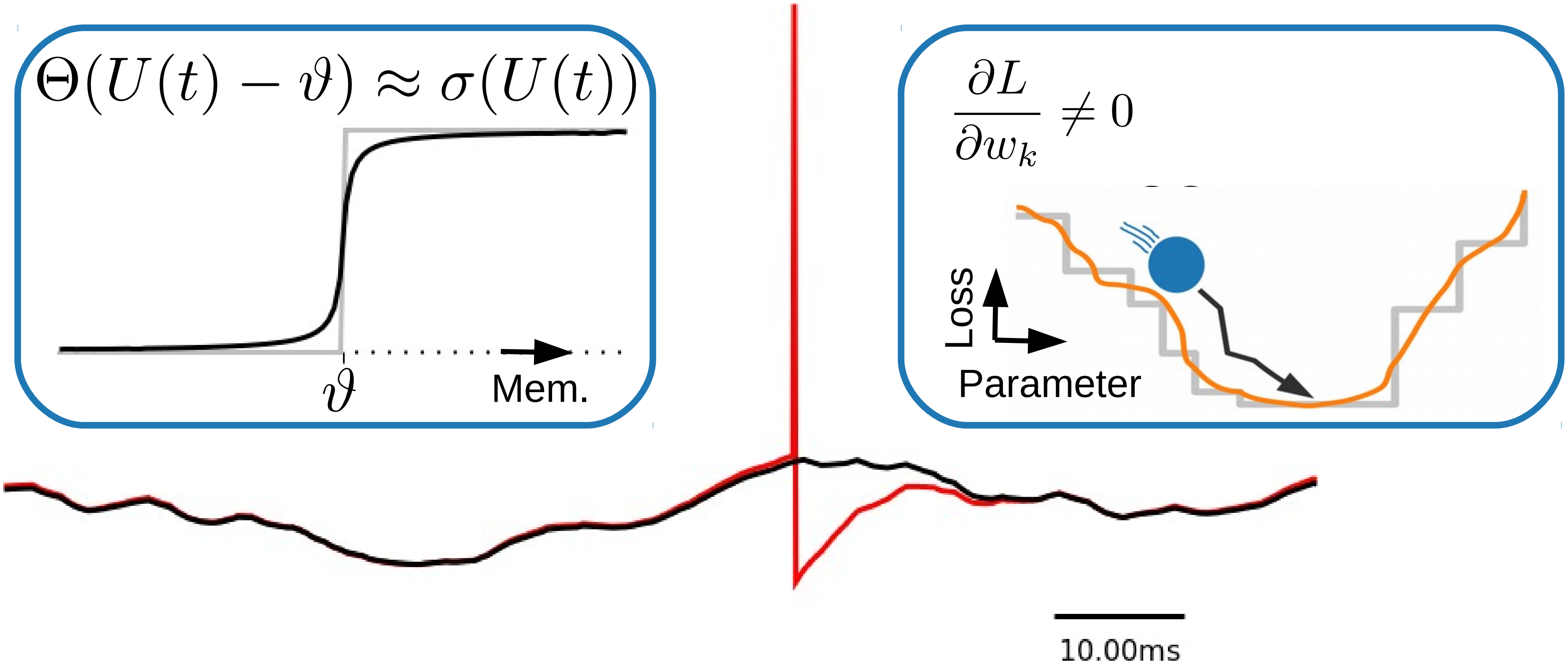
$$U_i^{(1)}[n+1] = \beta U_i^{(1)}[n] + I_i^{(1)}[n] - S_i[n]$$

$$I_i^{(1)}[n+1] = \underbrace{\alpha I_i^{(1)}[n]}_{\text{exp. current decay}} + \underbrace{\sum_j W_{ij} S_j^{(0)}[n]}_{\text{feed-forward input}}$$

Problem: The derivative of a spike is zero almost everywhere



What you want ...

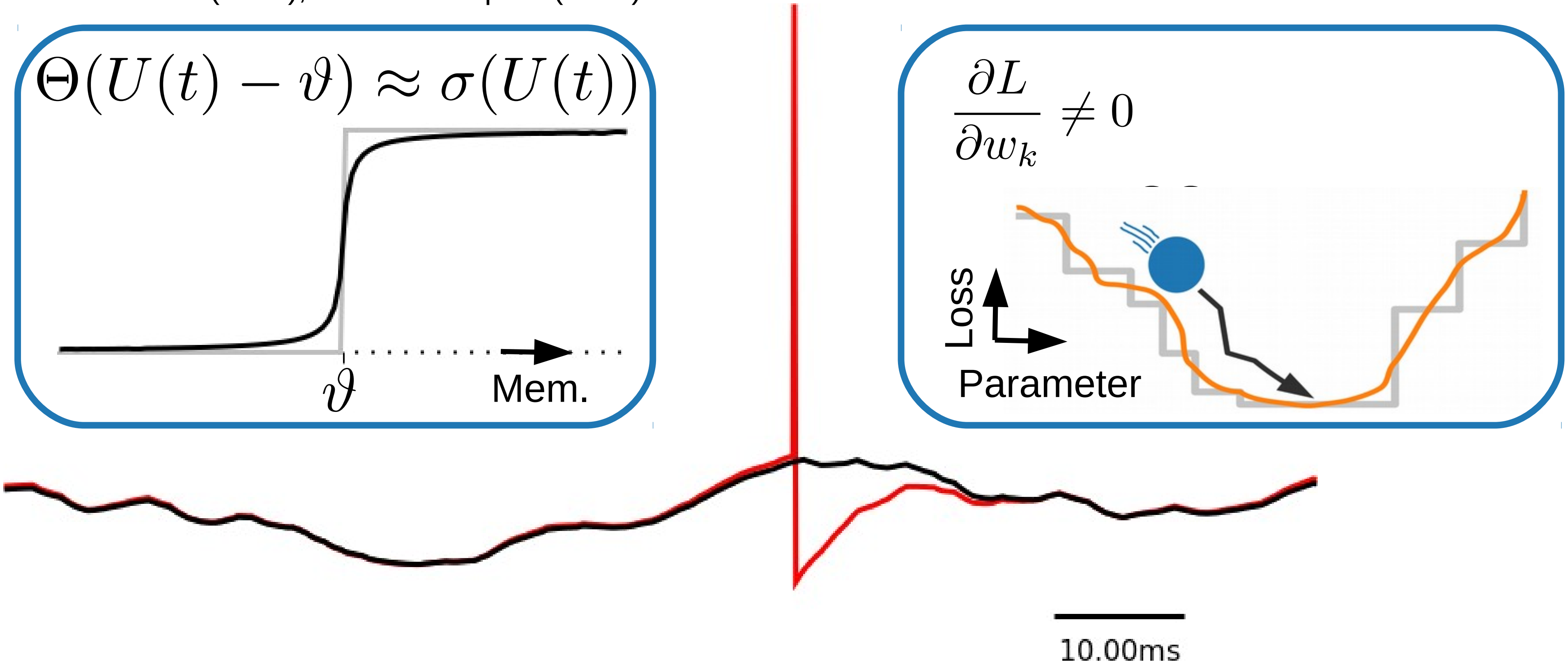


Option 1 (“classic”): Noise injection → well defined gradient in expectation values.
e.g.: Pfister, Toyoizumi, Barber & Gerstner (2006)
Gardner, Sporea & Grüning (2015)

Option 2: Make spikes differentiable.
Huh & Sejnowski (2018)

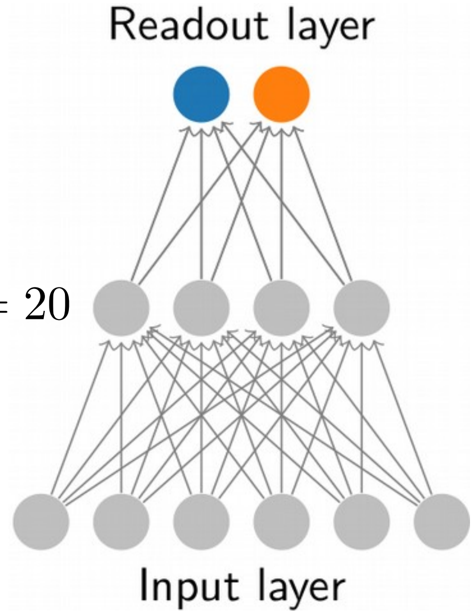
Option 3: “Know hidden layer targets”
Gilra & Gerstner (2017), Nicola & Clopath (2017)

Option 4: Use surrogate gradients.
Bellec, Salaj, Subramoney, Legenstein, and Maass (2018)
Shrestha & Orchard (2018), Zenke & Ganguli (2018), ...
In ML: “Straight-through estimators” Bengio et al. (2013)



Simulation experiments

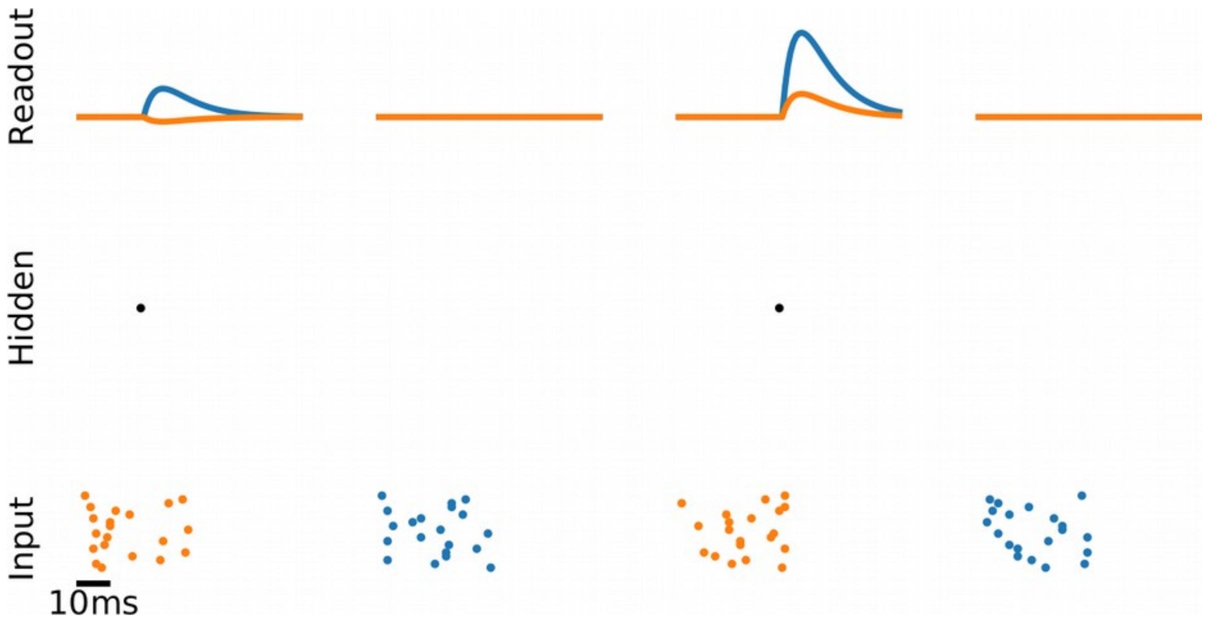
A synthetic spatiotemporal problem



$n = 20$

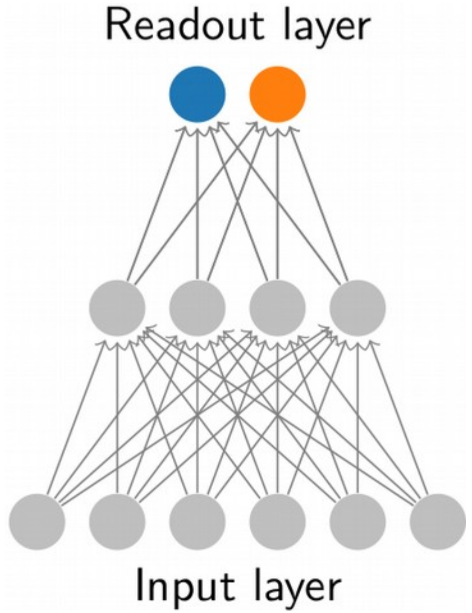
$n = 20$

Activity snapshots (randomly initialized network)

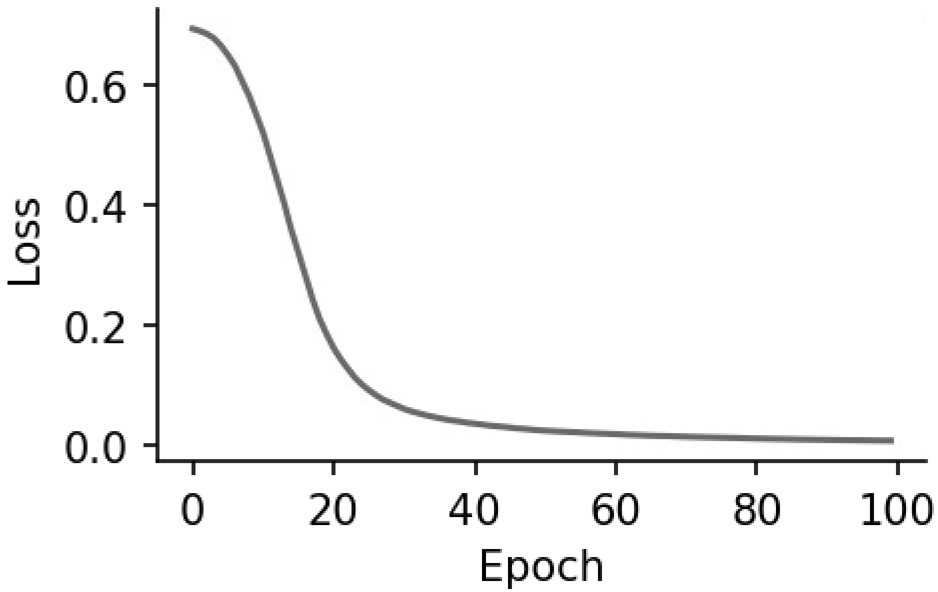


Dataset: 2000 samples from two smooth random manifolds

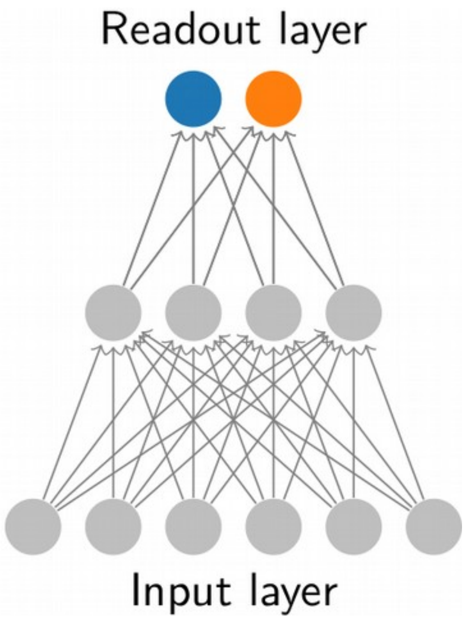
A synthetic spatiotemporal problem



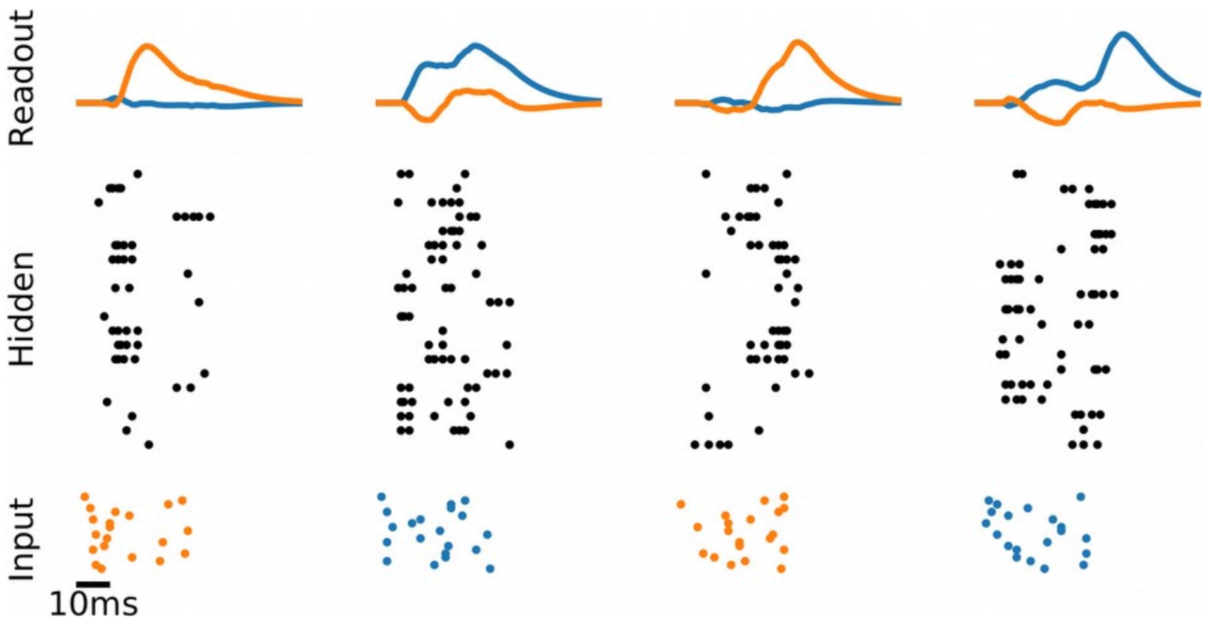
Evolution of loss during surrogate gradient descent



A synthetic spatiotemporal problem

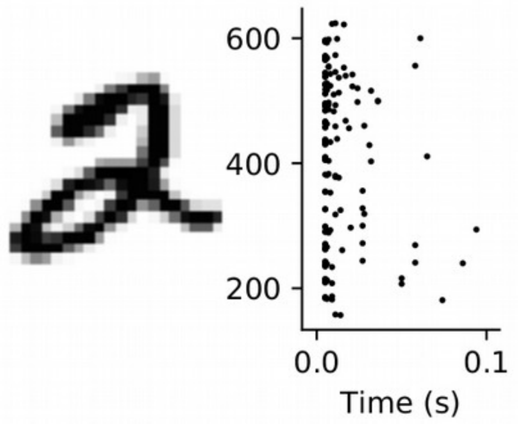


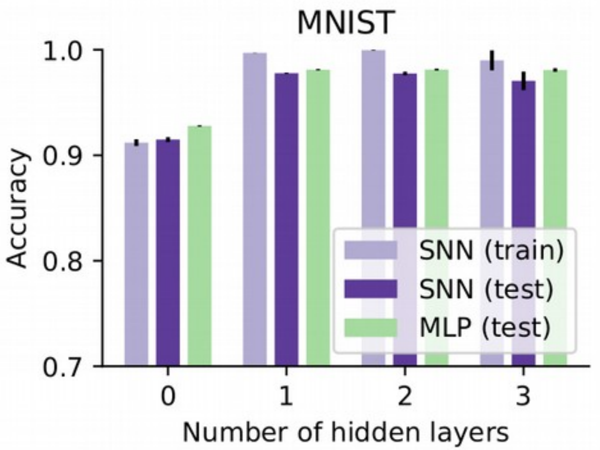
Activity snapshots (trained network)



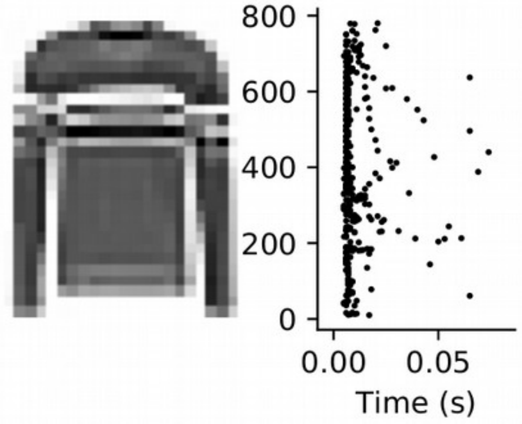
Benchmarks (preliminary)

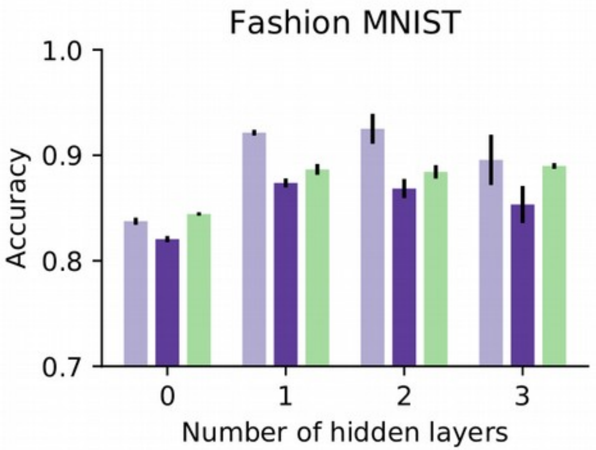
MNIST
LeCun, Cortes & Burges (1998)



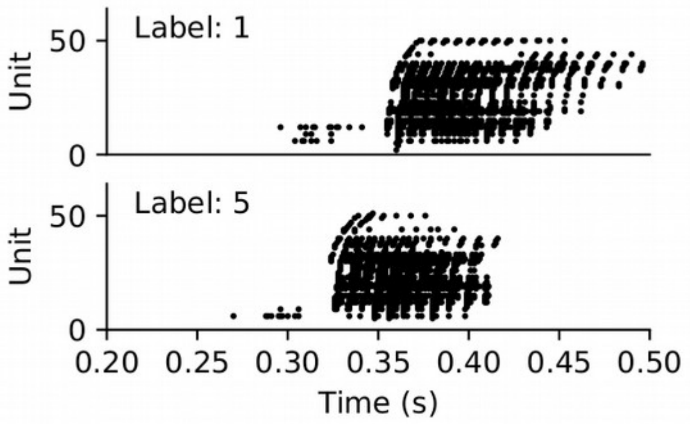


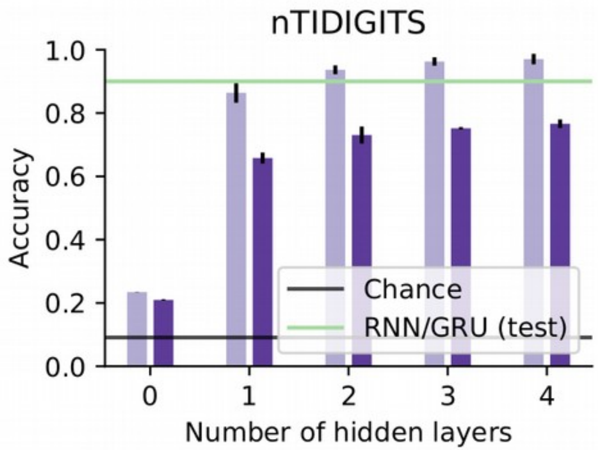
Fashion MNIST
Xiao, Rasul & Vollgraf (2017)





nTIDIGITS
Anumula, Neil, Delbruck & Liu (2018)

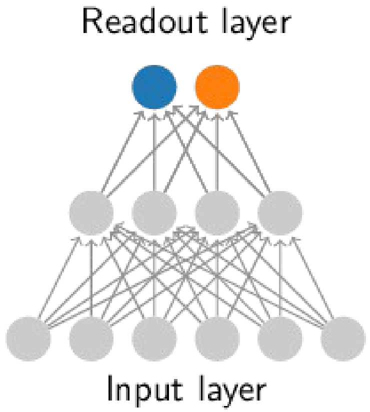


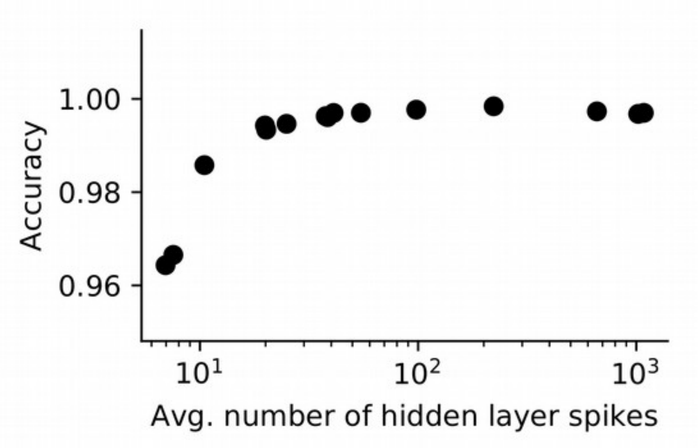


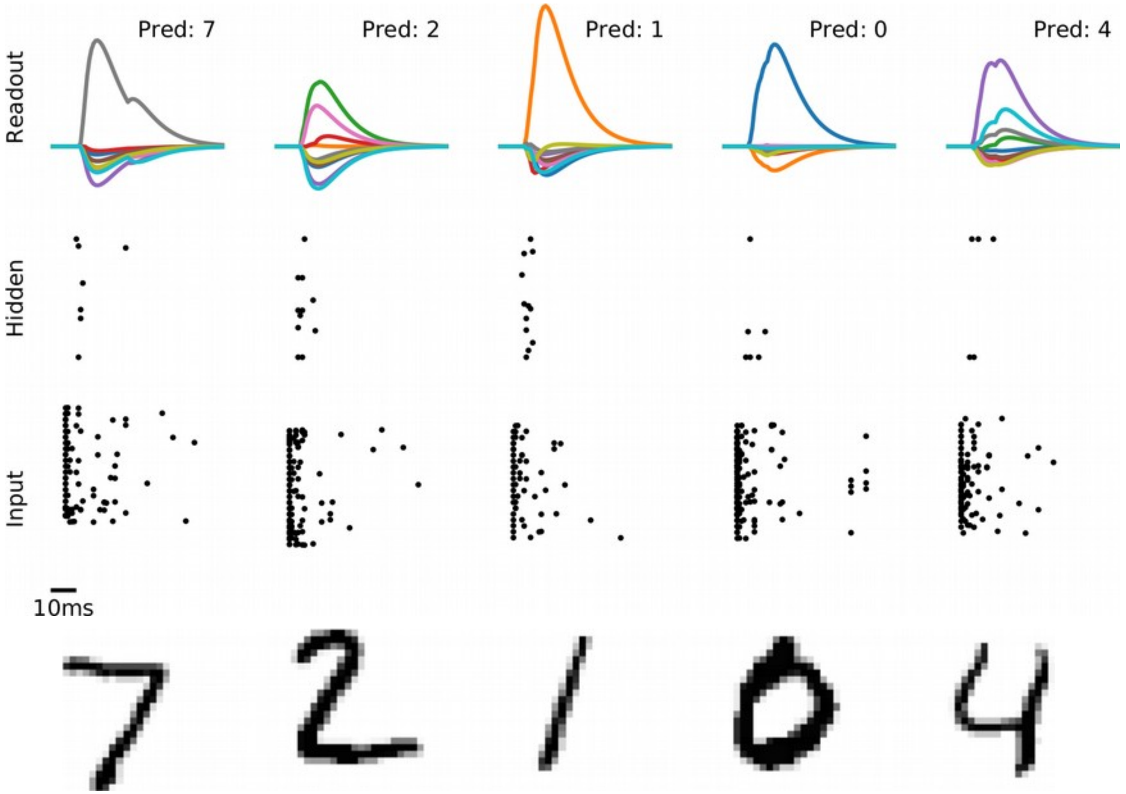
Do try this at home!

https://github.com/fzenke/spytorch

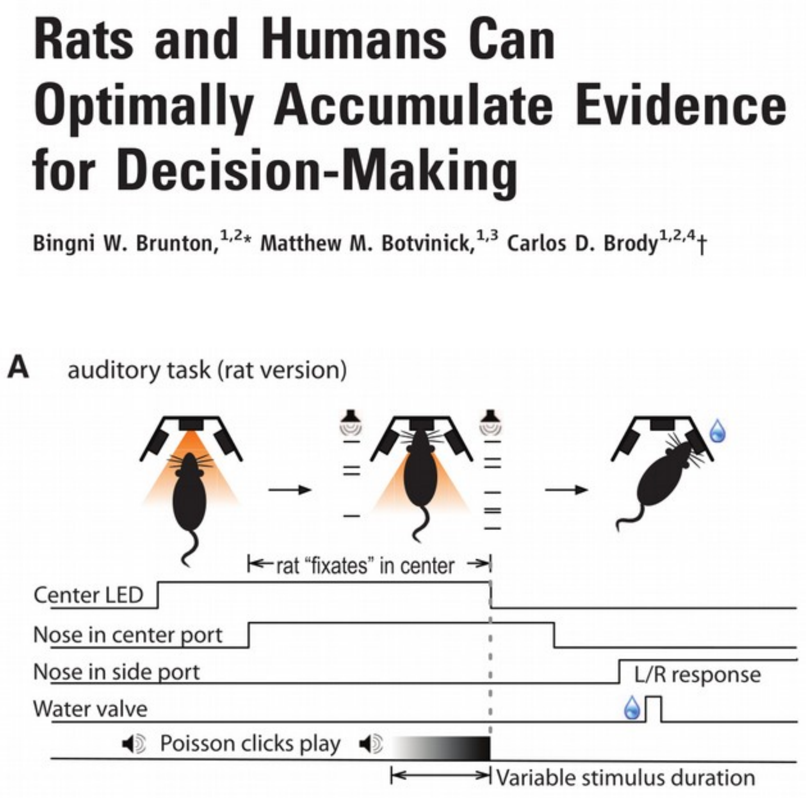
Lifetime sparseness does not hurt classification



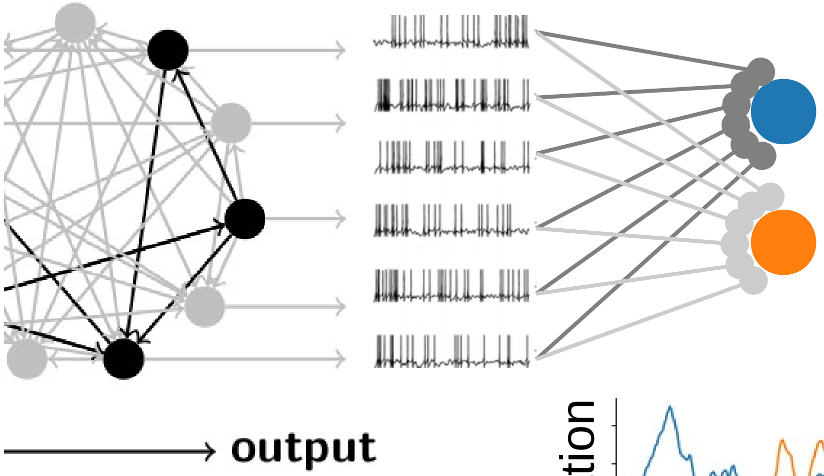


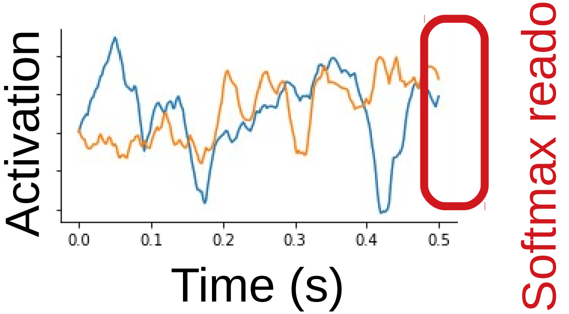


Back to bio-inspired: A DM problem



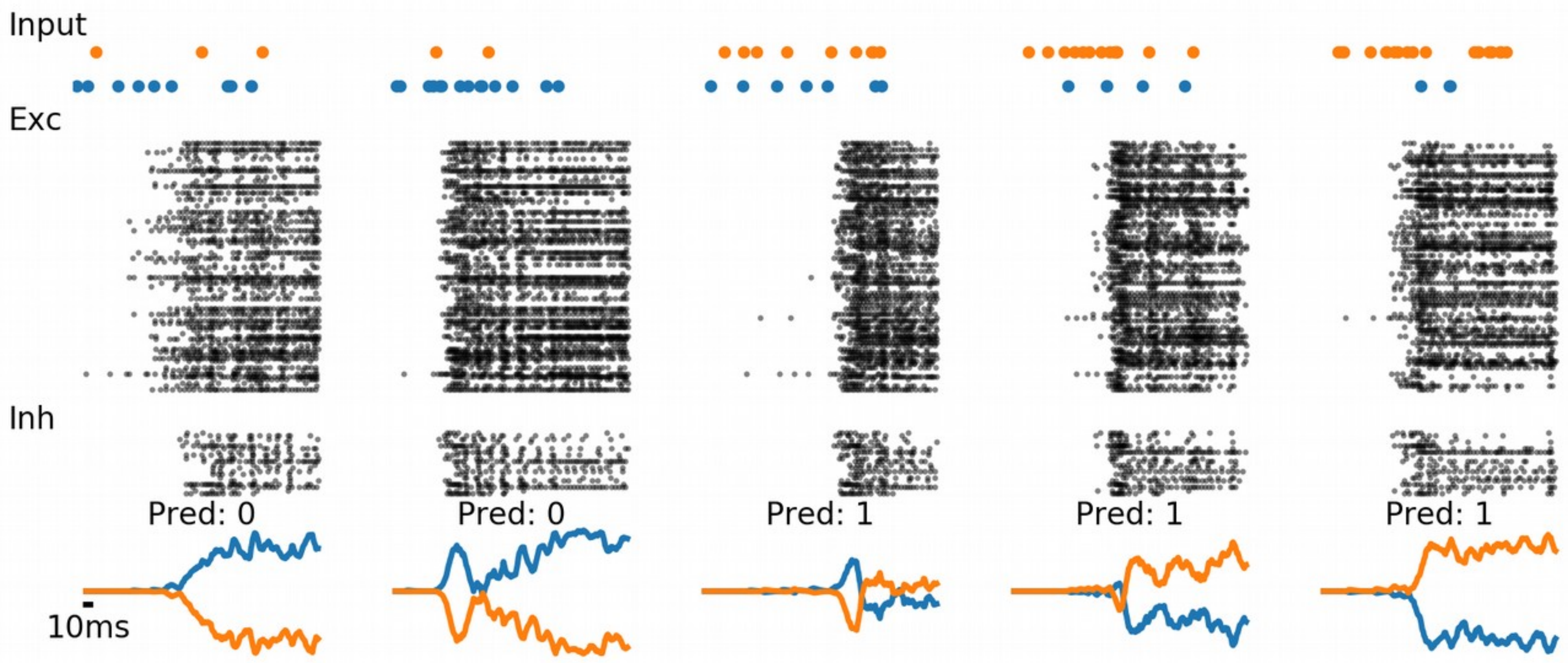
+ Short-term plasticity
Tsodyks & Markram (1997)





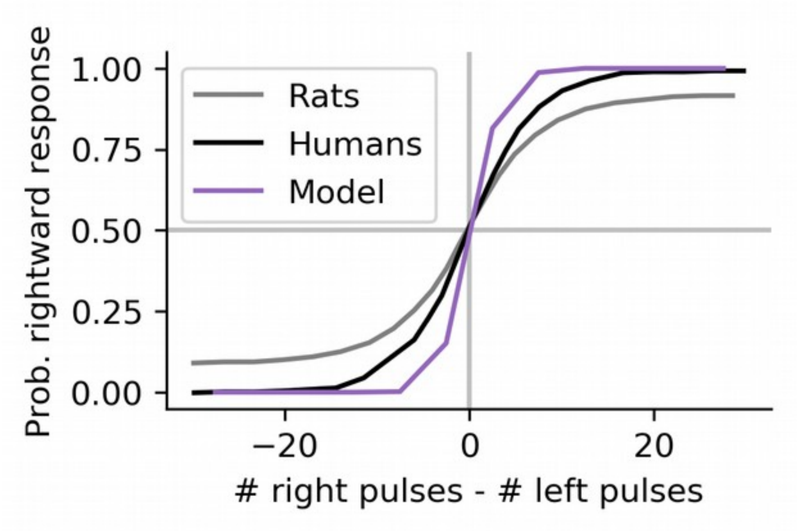
Brunton, Botvinick, and Brody (2013)

Activity snapshots for single decision making trials



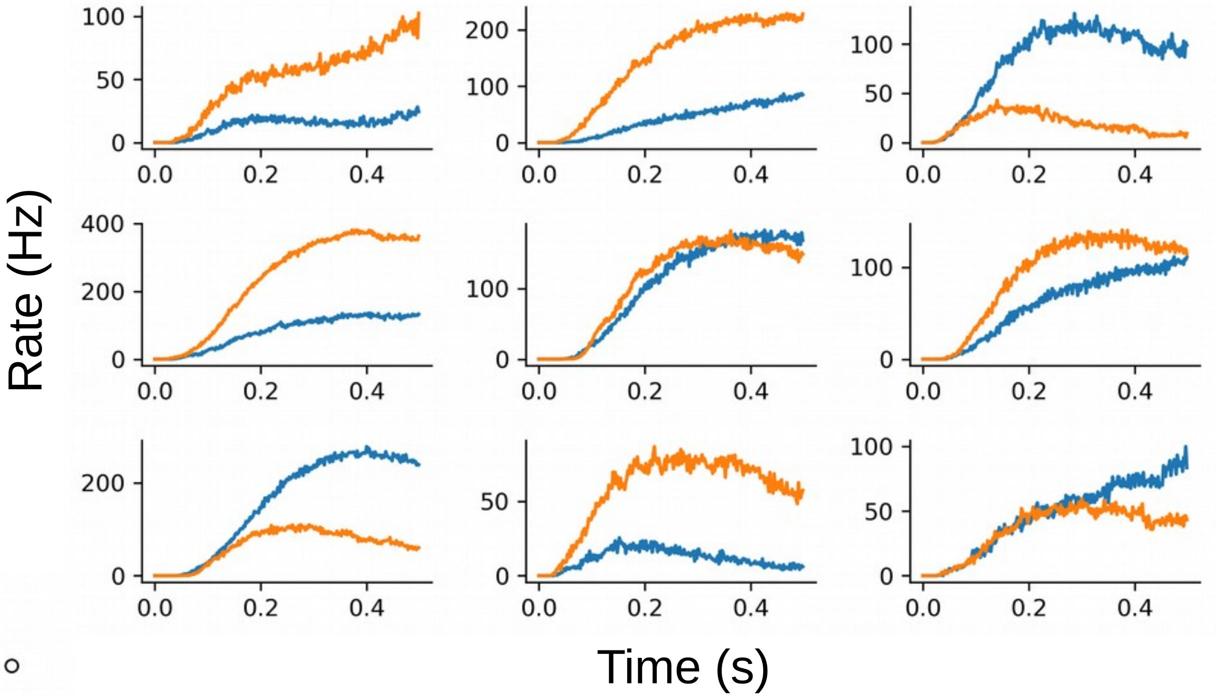
Network learns to use delay activity

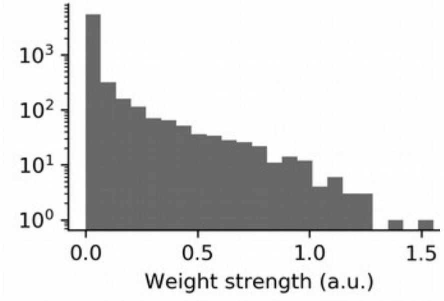
Spiking network solves the random clicks task

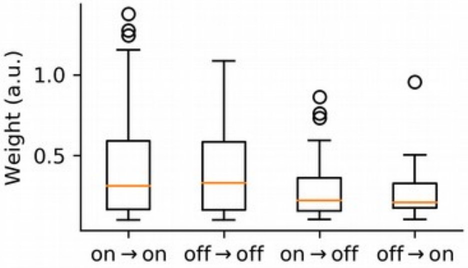


Data: Brunton, Botvinick, and Brody (2013)

Single cell PSTHs



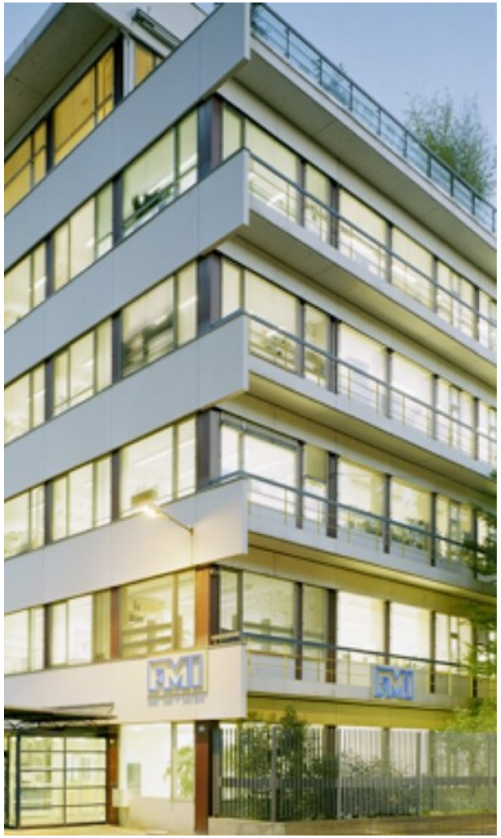




Summary

- A spiking neural network (SNN) is a binary RNN
- SNNs can be trained with surrogate gradients
- Trained SNNs
  - Can solve temporal problems
  - Promising benchmark results (room for improvement)
- Trained SNNs are a useful tool
  - If you fancy using them in hardware
  - Hypothesis generation
  - Control for bio plausible learning

Shameless plug



- Zenke Lab coming soon...
  - June 2019
  - At the FMI in Basel, Switzerland
  - www.zenkelab.org
- Positions available
  - PhD and post-doc level
  - PhD program deadlines:
    May 1st and Nov 16th
  - https://www.fmi.ch/training/PhD/

Thanks

Advisors

Surya Ganguli and the Gang

Tim Vogels and Group

Review/Tutorial : Neftci, Mostafa, & Zenke (2019). ArXiv

Emre Neftci, UC Irvine

Hesham Mostafa, UCSD

Funding: wellcome

Code & Tutorials: fzenke.net

Pencil sketches: kyadava.net