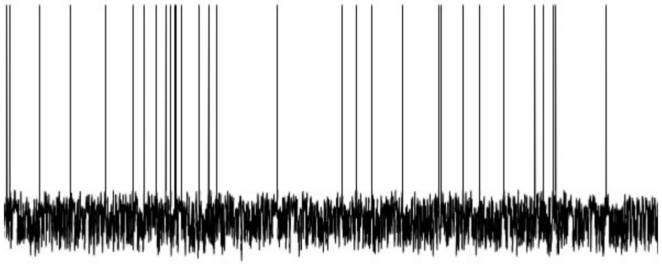
Building functional spiking neural networks using surrogate gradients

Friedemann Zenke

https://zenkelab.org

FMI

Friedrich Miescher Institute for Biomedical Research

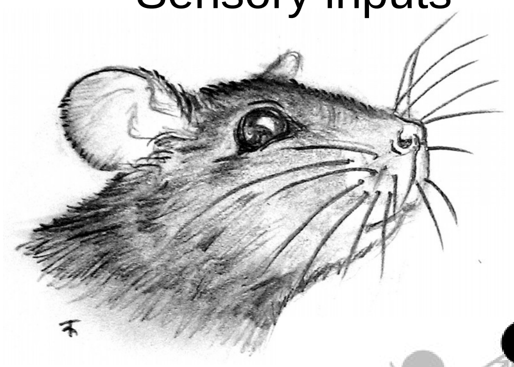


wellcome

UNIVERSITY OF OXFORD

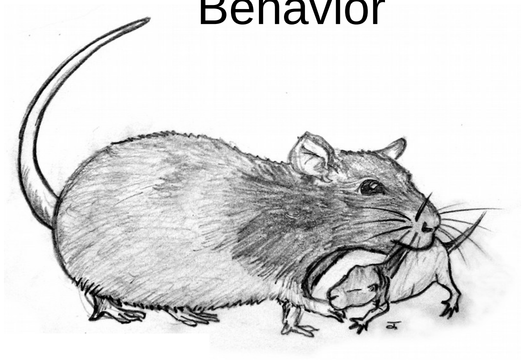
Animals process information using neural networks

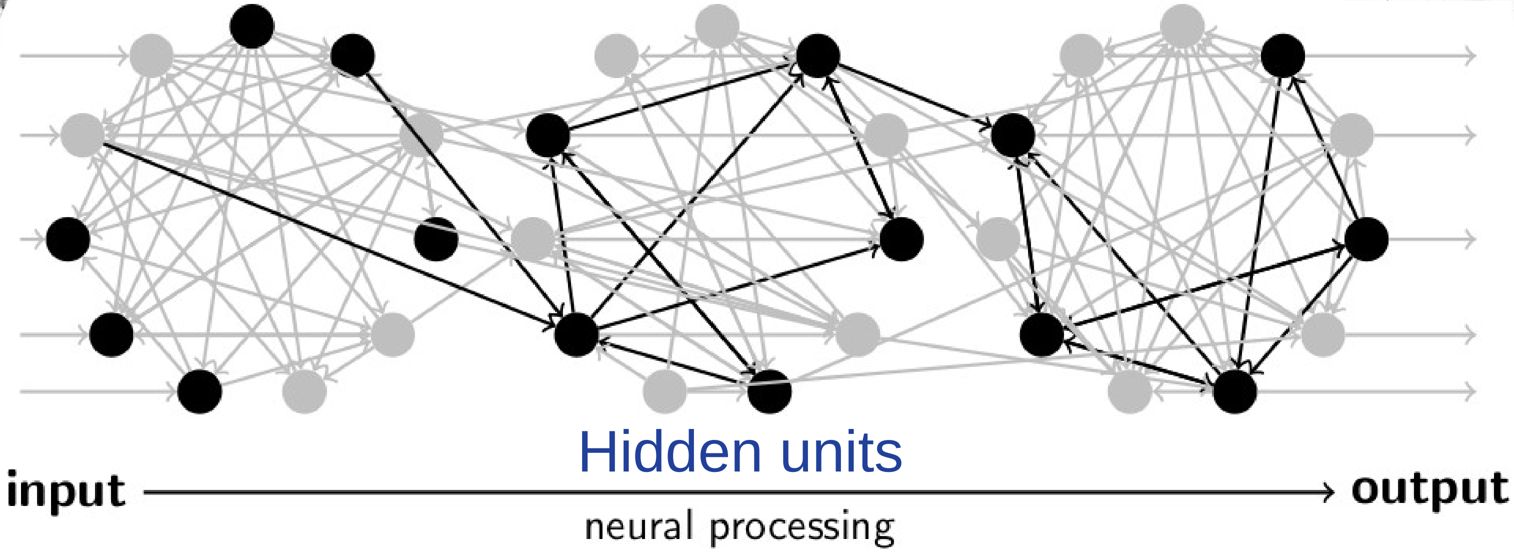
Sensory inputs



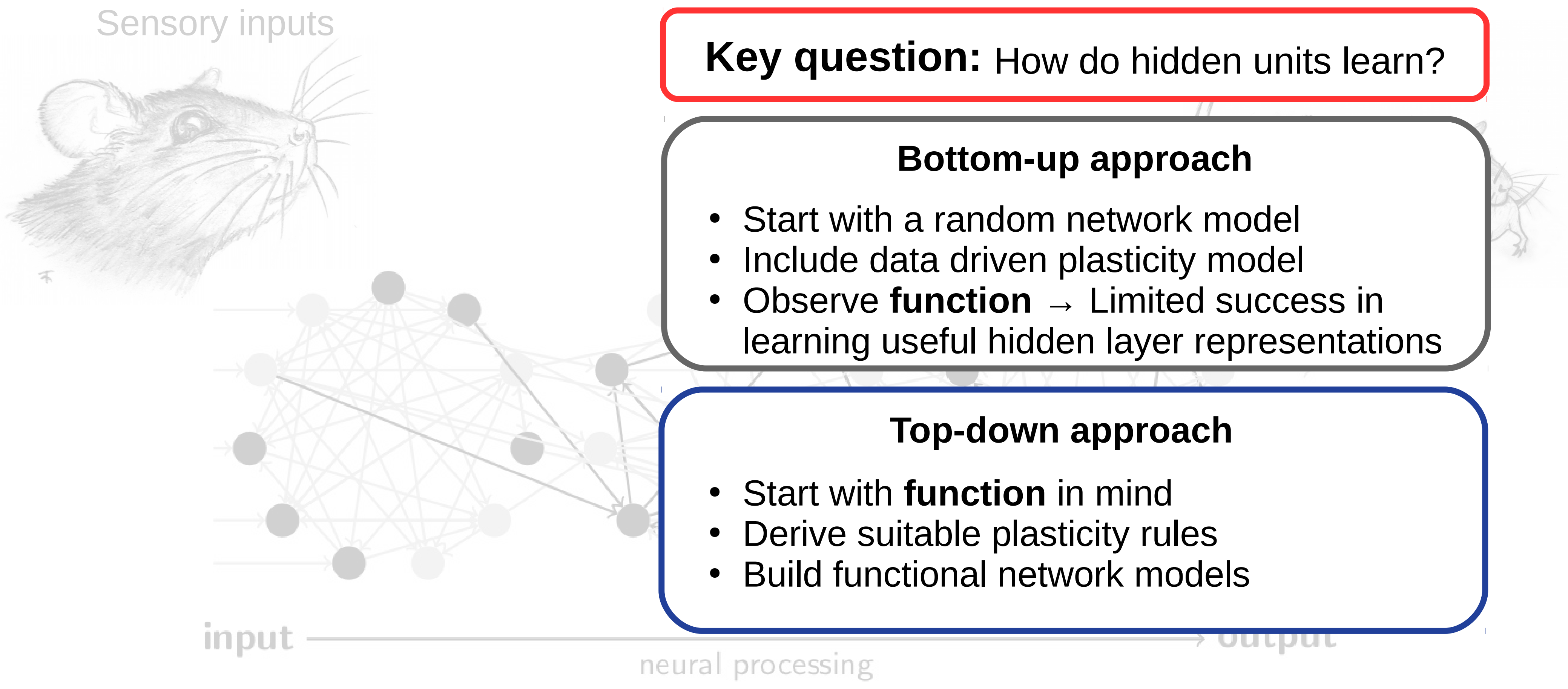
Function

Behavior





Animals process information using neural networks



Key question: How do hidden units learn?

Bottom-up approach

- Start with a random network model
- Include data driven plasticity model
- Observe **function** → Limited success in learning useful hidden layer representations

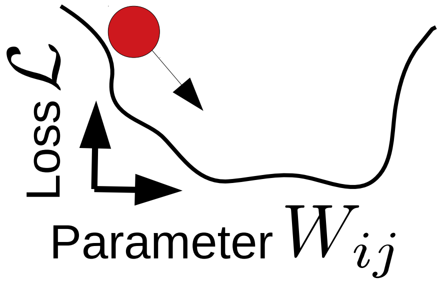
Top-down approach

- Start with **function** in mind
- Derive suitable plasticity rules
- Build functional network models

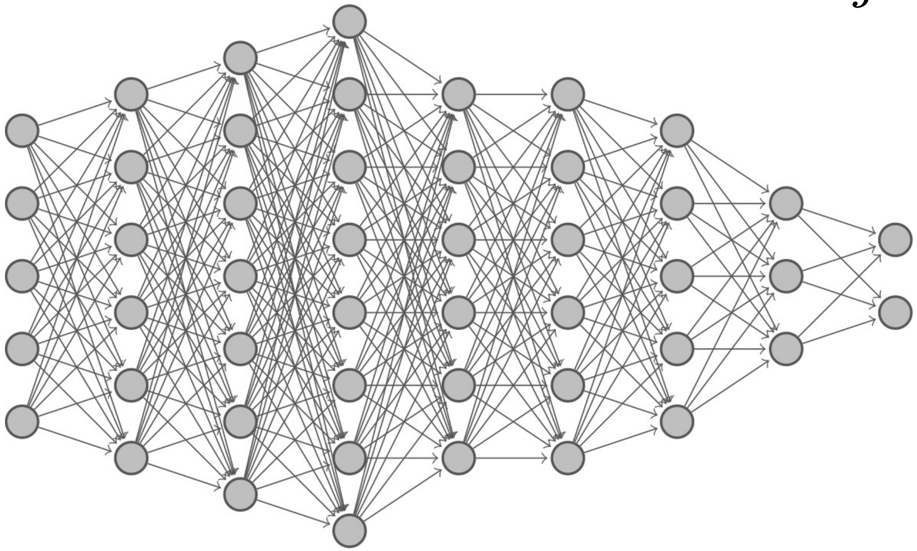
Deep learning provides a useful framework

3) Adjust weights
Gradient descent

$$\Delta W_{ij} \propto -\frac{\partial \mathcal{L}}{\partial W_{ij}}$$



1) Input data X

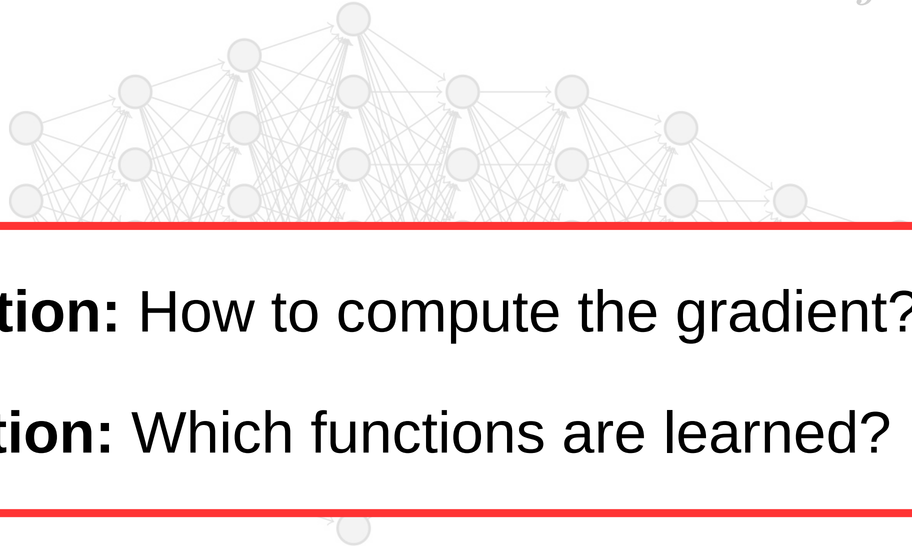


$\mathcal{L}$ **2) Loss function**

Deep neural networks implement functions
They ”learn”, but they don’t spike

3) Adjust weights
Gradient descent

$$\Delta W_{ij} \propto -\frac{\partial \mathcal{L}}{\partial W_{ij}}$$



2) Loss function

Algorithmic question: How to compute the gradient?

Conceptual question: Which functions are learned?

NATURE VOL. 337 12 JANUARY 1989 COMMENTARY 129

1989

The recent excitement about neural networks

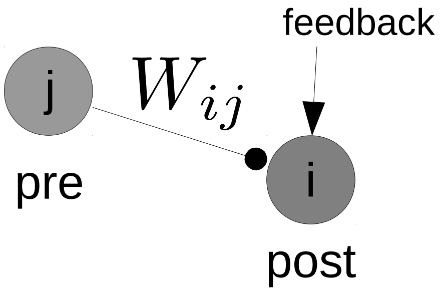
Francis Crick

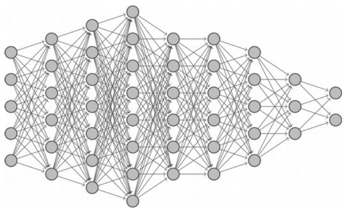
The remarkable properties of some recent computer algorithms for neural networks seemed to promise a fresh approach to understanding the computational properties of the brain. Unfortunately most of these neural nets are unrealistic in important respects.

“Unrealistic in important respects”

- Non-locality of learning rules
  (a.k.a. the weight transport problem)
- Graded activation functions vs spikes

The *more recent* excitement about (deep) neural networks





$$\Delta W_{ij} \propto (\text{pre}_j)\boxed{f(\text{post}_i)}\boxed{(\text{feedback}_i)}$$

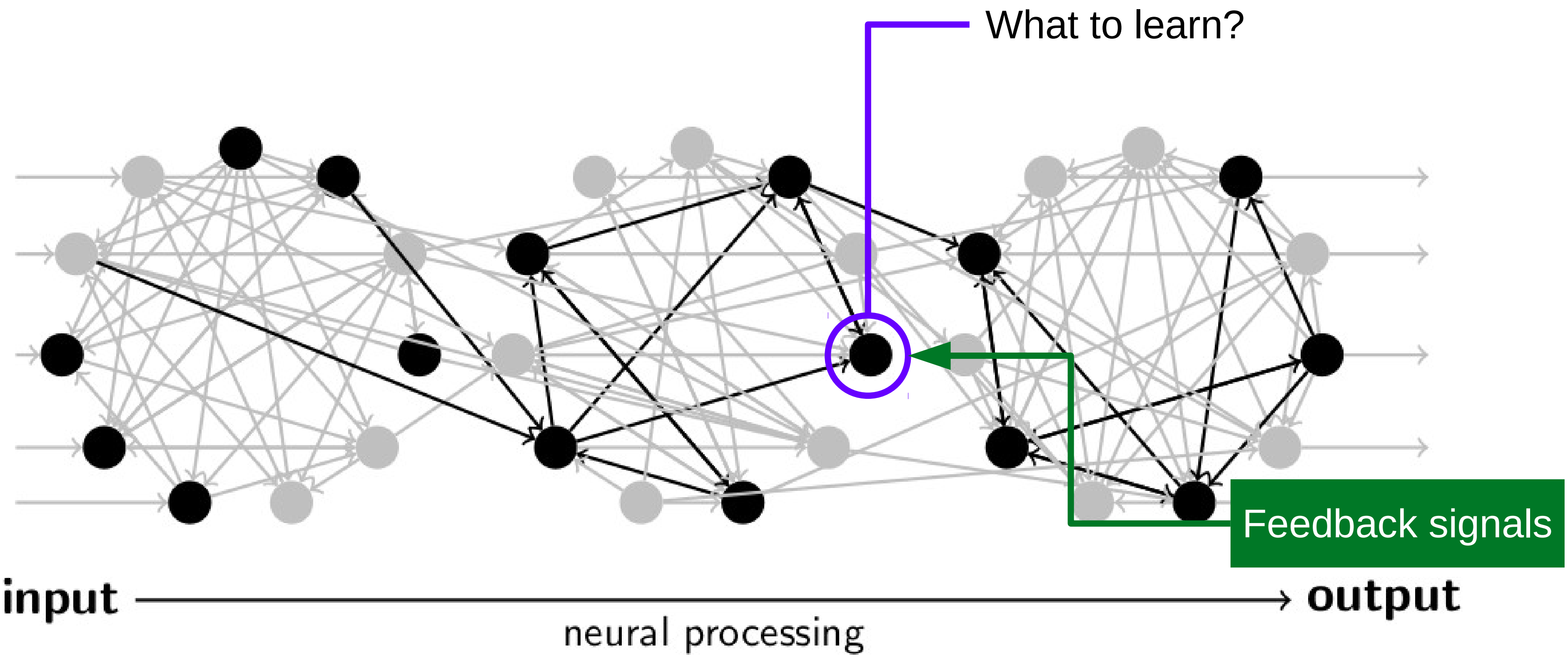
“Unrealistic in important respects”

- Non-locality of learning rules (a.k.a. the weight transport problem)
- Graded activation functions vs spikes

Plausible vector-valued feedback!

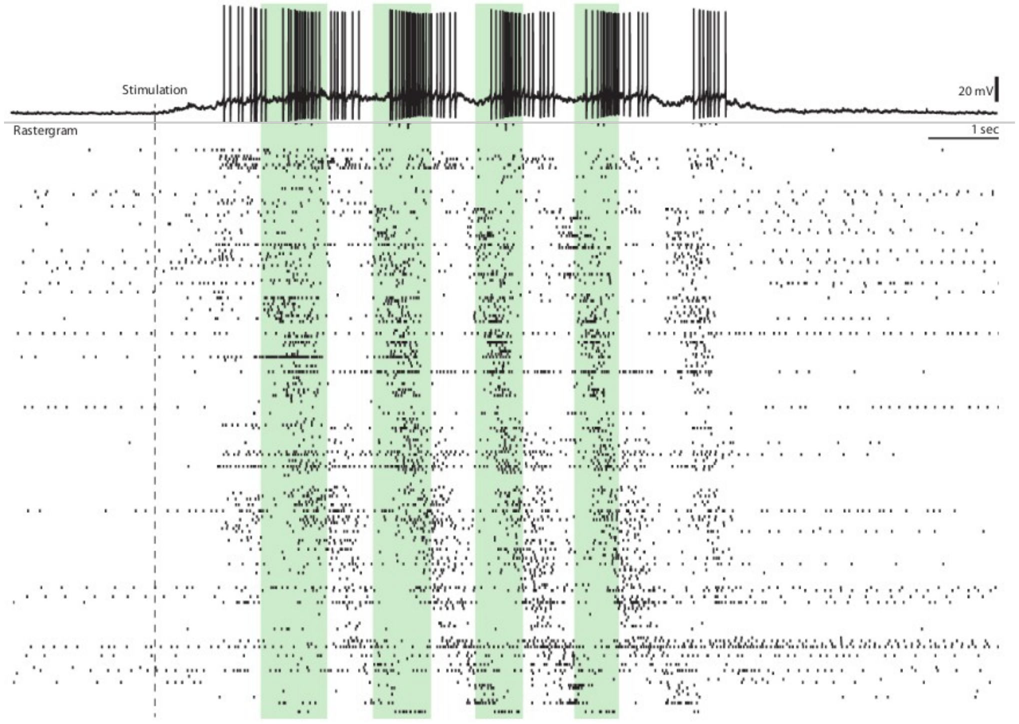
- Lillicrap et al. (2016)
- Nøkland (2016)
- Guerguiev et al. (2017)
- Scellier & Bengio (2017)
- Whittington & Bogacz (2017)
- Sacramento et al. (2018)
- Pozzi et al. (2018)

Spatial credit assignment



Neural networks use spikes to process temporal information

Neurons



1s Petersen & Berg (2016)

Time

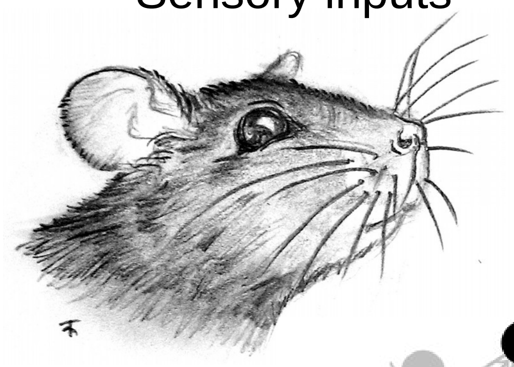
$$\Delta W_{ij} \propto (\mathrm{pre}_j(t)) f(\mathrm{post}_i(t))(\mathrm{feedback}_i(t))$$

Outline

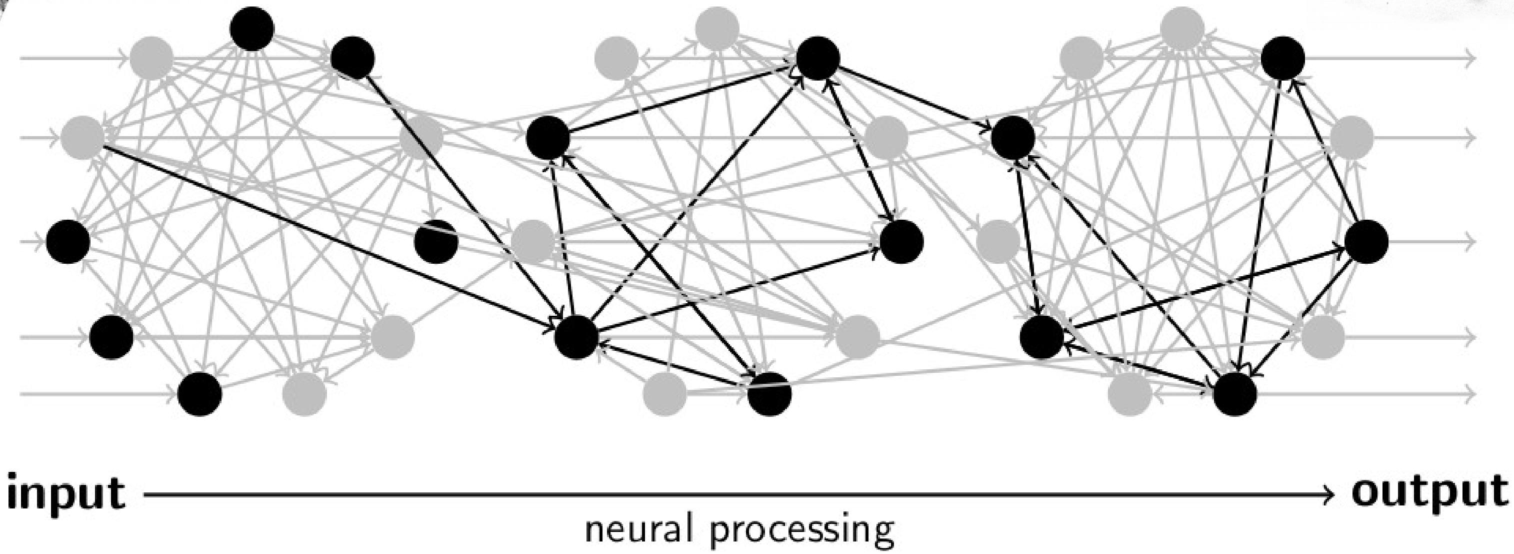
- **Aim:** Solve temporal tasks with spiking networks
- **Problem:** Spike → ill defined derivative
- **Solution:** Surrogate gradients
- **A look at:** Robustness, performance
  For a bio-plausible learning rule see Zenke & Ganguli (2018)

Towards functional neural network models

Sensory inputs



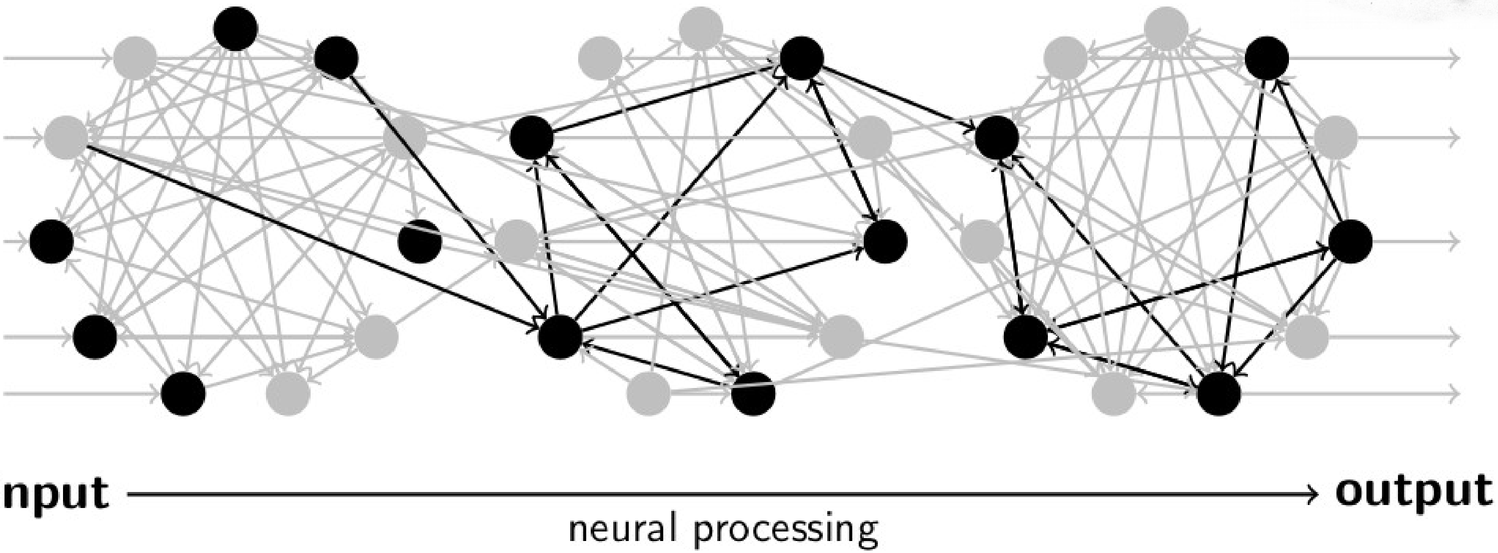
Behavior



Towards functional neural network models

Behavior

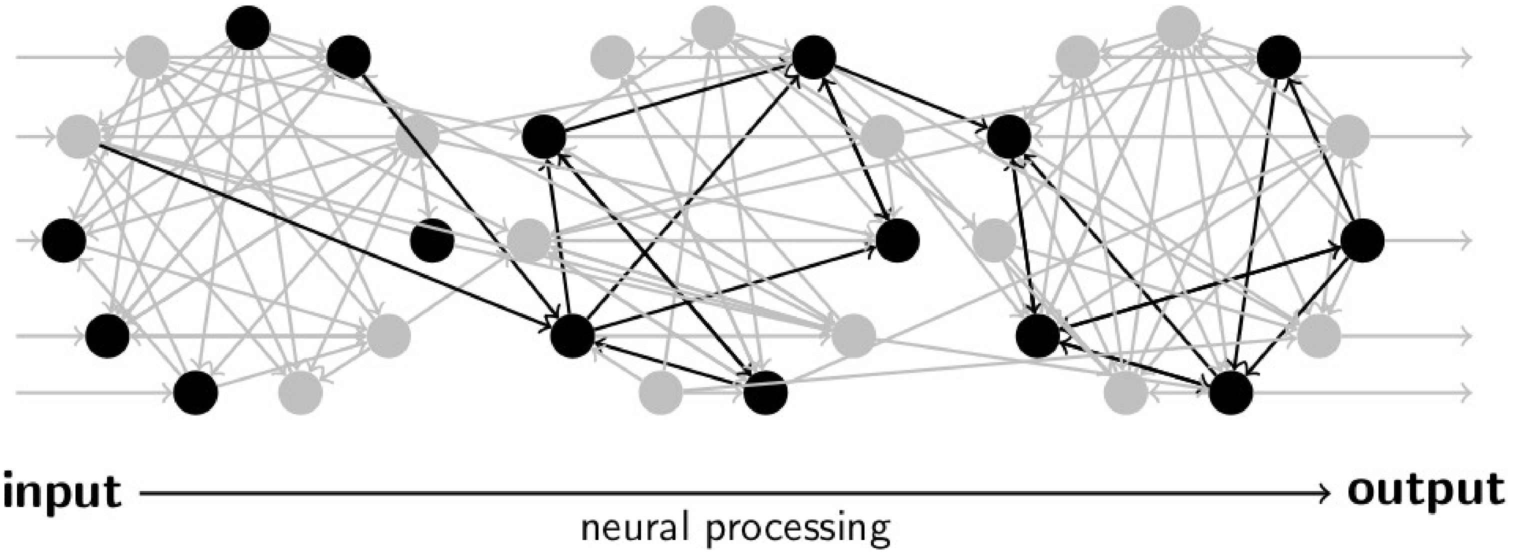
1) Input



Towards functional neural network models

1) Input

2) Output

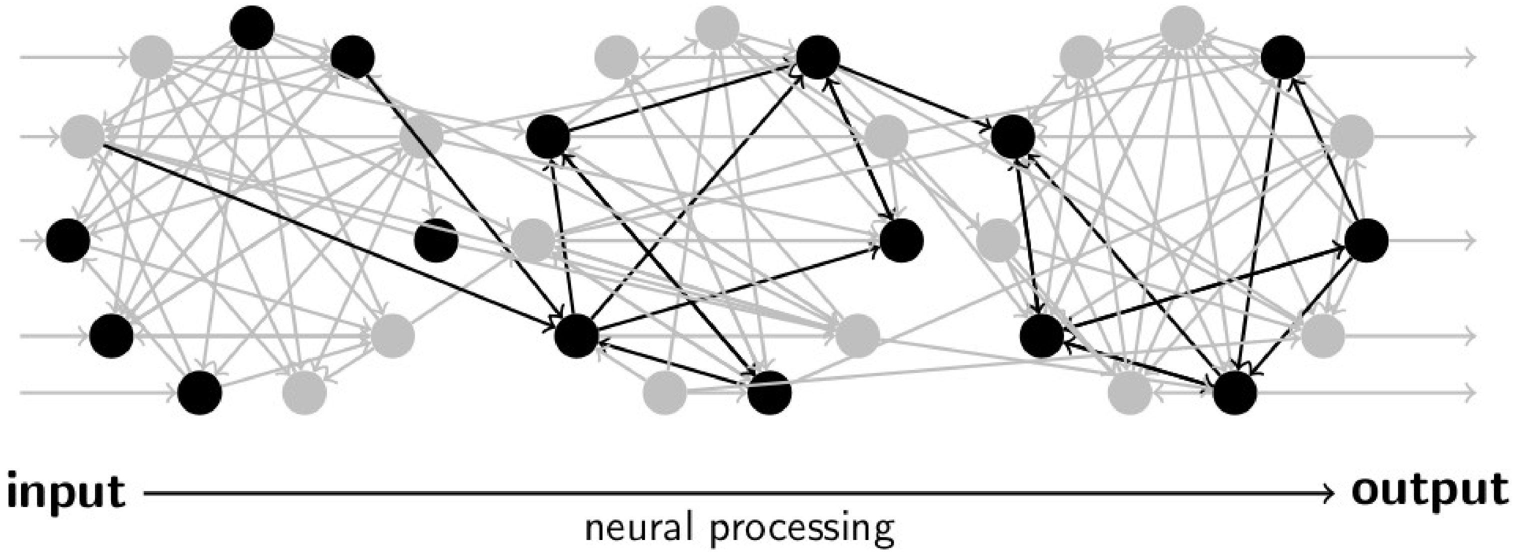


Towards functional neural network models

1) Input

3) Adjust weights

2) Output

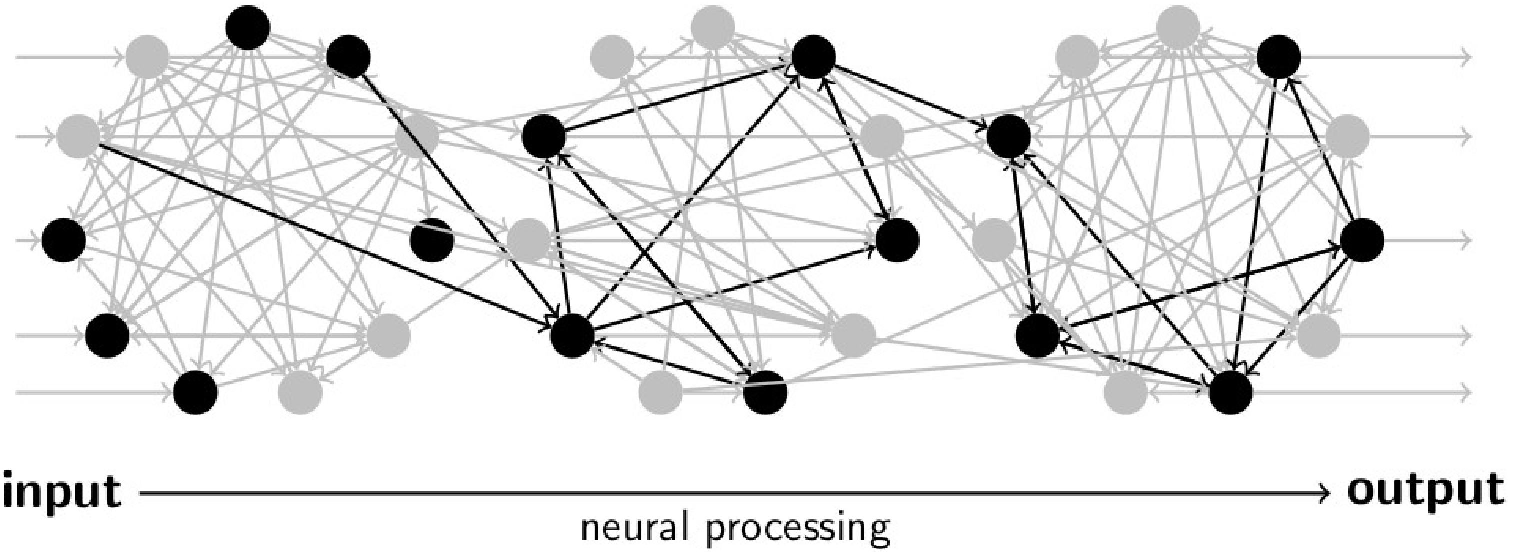


Towards functional neural network models

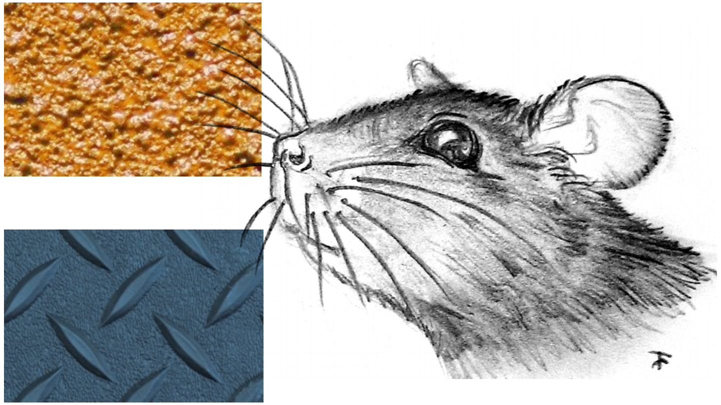
1) Input
(spatiotemporal)

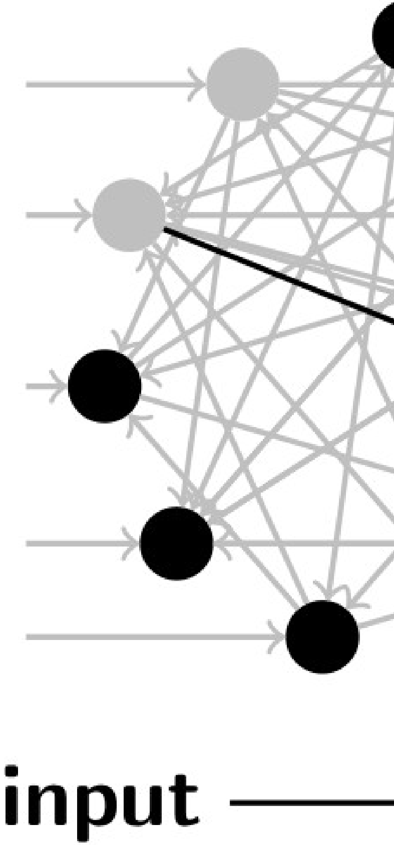
3) Adjust weights
(surrogate gradients)

2) Output
(classification)

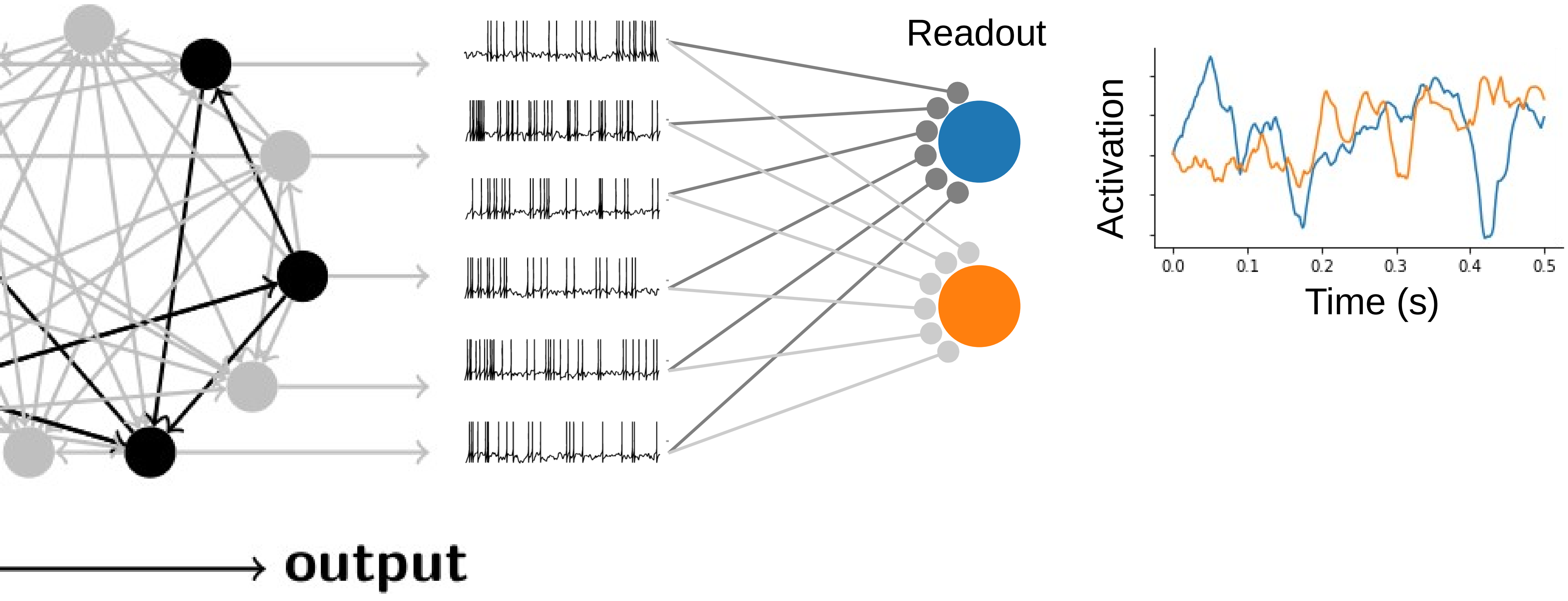


Input: Spatiotemporal spike patterns

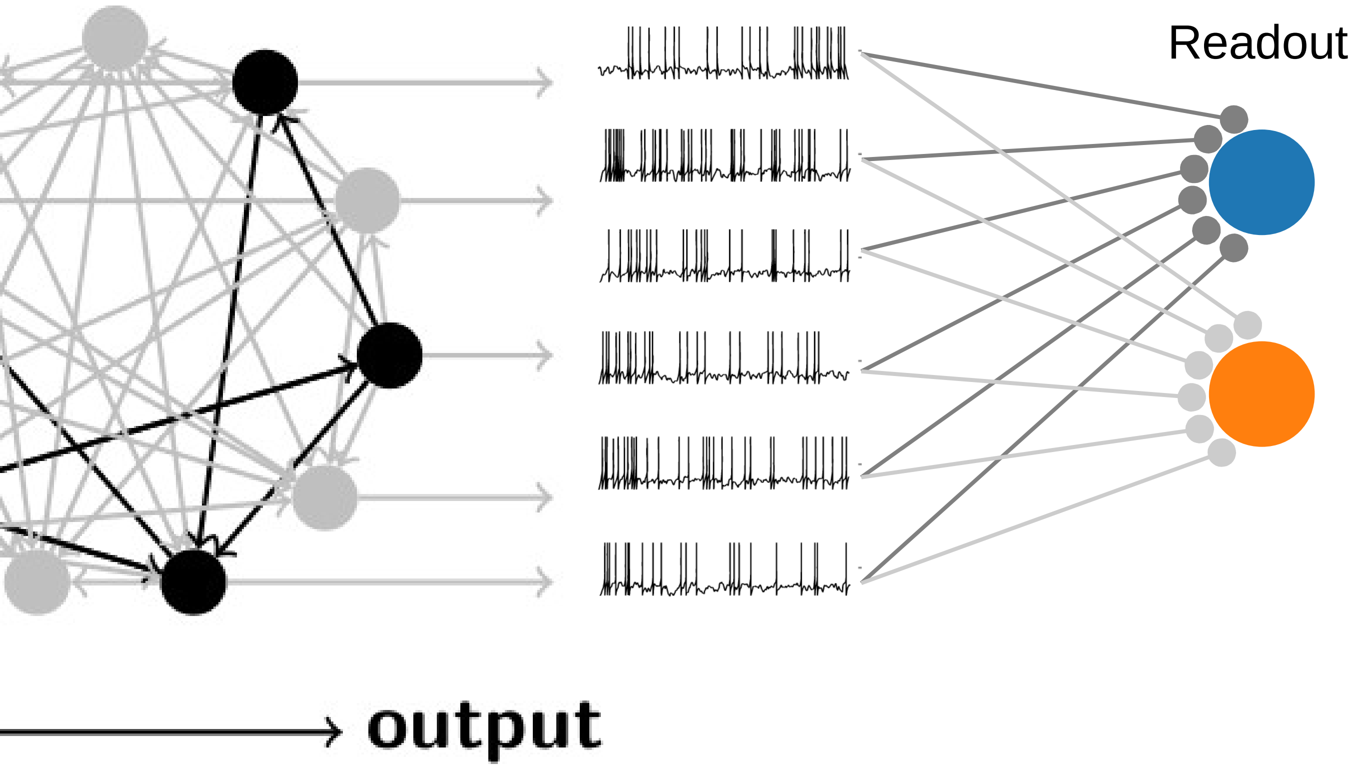


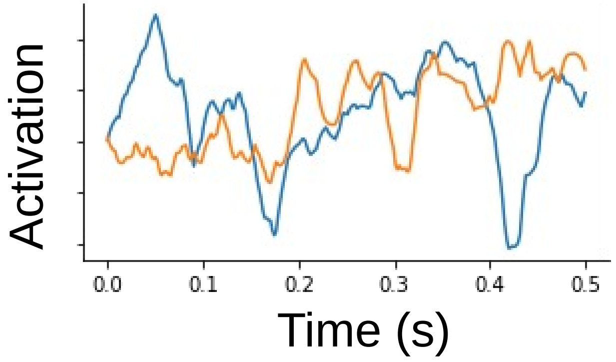


Output: Linear combination of filtered output spike trains



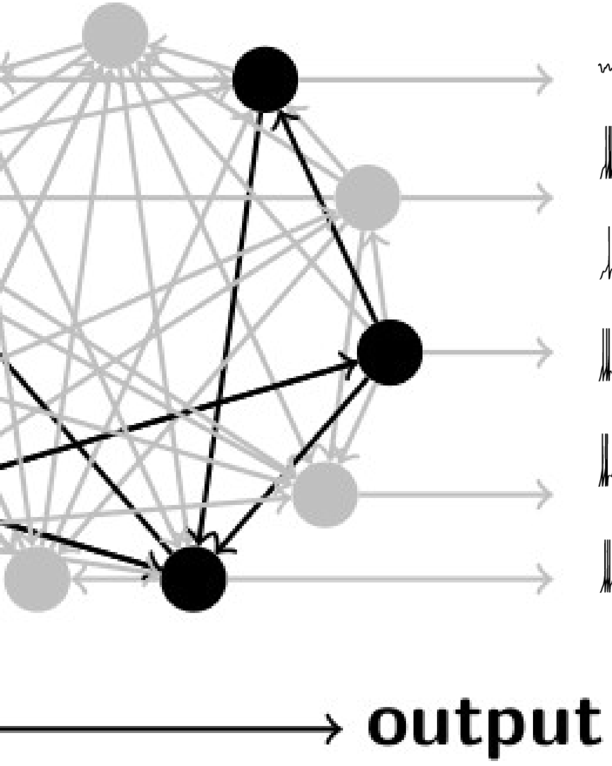
Output: Linear combination of filtered output spike trains

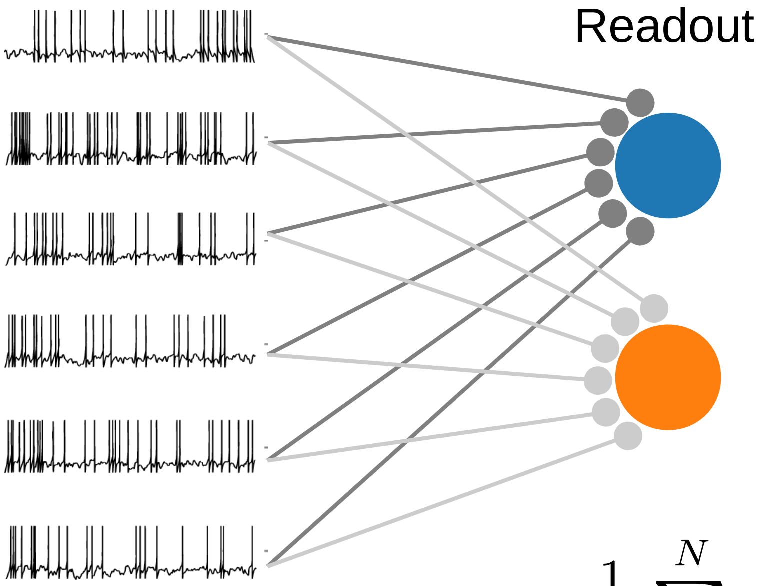


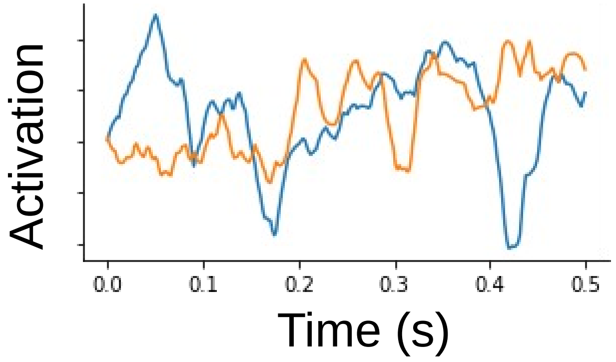


$$p_i = \operatorname{softmax}_i \left(\max_t U_i(t) \right)$$

Output: Linear combination of filtered output spike trains







$$p_i = \underset{i}{\mathrm{softmax}} \left(\boxed{\max_t U_i(t)} \right)$$

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

Max over time idea from Tempotron

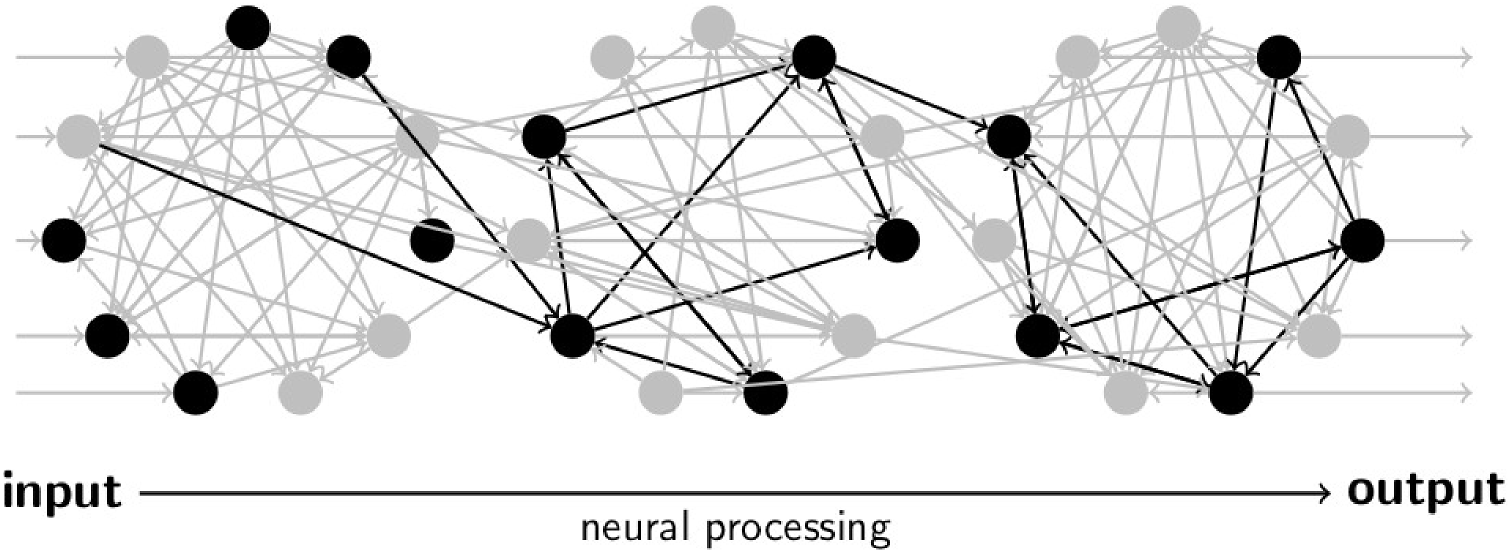
Gütig & Sompolinsky (2006); Gütig (2016)

Towards spiking network models which compute

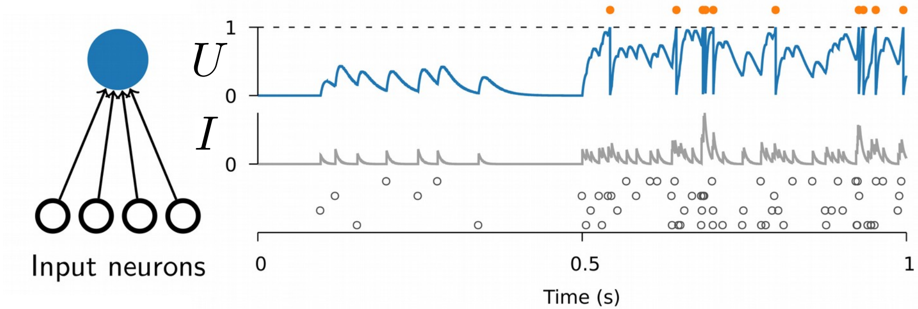
1) Input
(spatiotemporal)

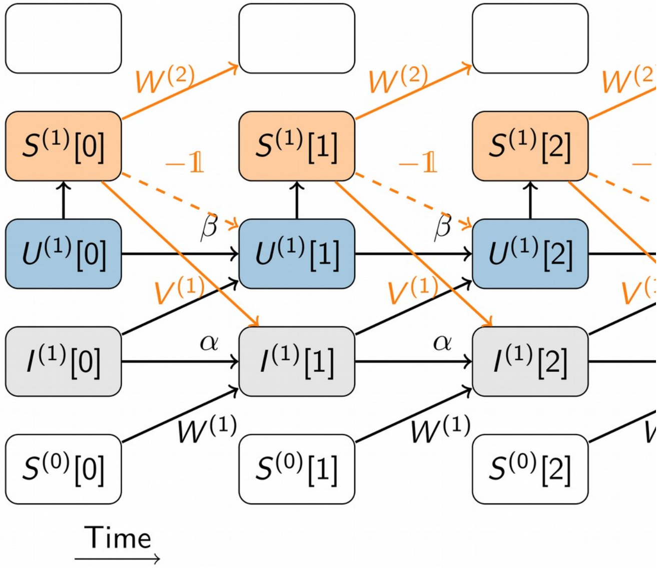
3) Adjust weights
(gradient descent)

2) Output
(classification)



Important insight: Spiking neural networks are binary RNNs with specific intrinsic recurrence





- Can be trained using BPTT or RTRL
- Several groups have realized this:
  - Esser, Merolla, Arthur, Cassidy, Appuswamy, Andreopoulos, Berg, McKinstry, Melano, Barch, et al. (2016)
  - Zenke & Ganguli (2018)
  - Huh & Sejnowski (2018)
  - Shrestha & Orchard (2018)
  - Bellec, Salaj, Subramoney, Legenstein, and Maass (2018)
  - Neftci, Mostafa, & Zenke (2019)

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

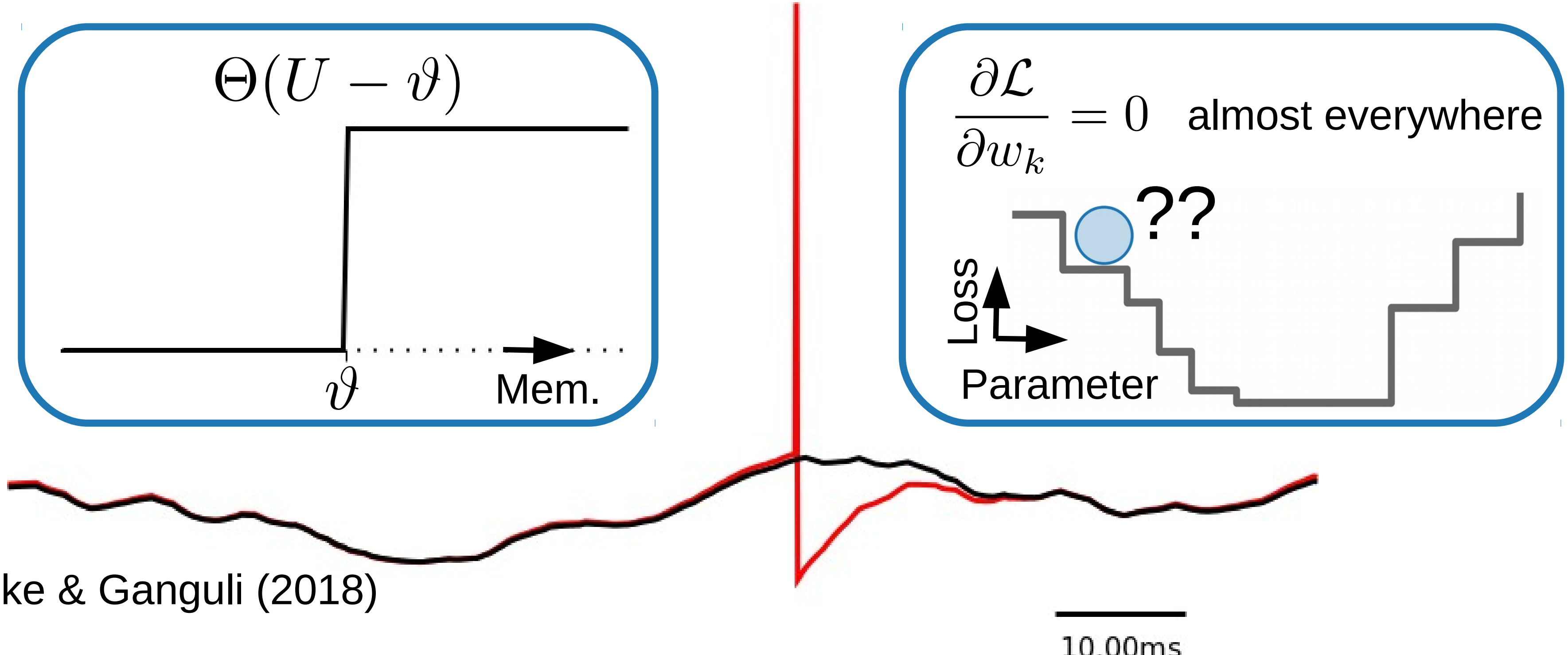
Problem

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

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

Neftci, Mostafa, & Zenke (in press)

Problem: The derivative of a spike train vanishes almost everywhere



Zenke & Ganguli (2018)

An **awesome problem** & a history of struggle

Today: Surrogate gradients.
Bohte (2011), Zenke & Ganguli (2018),
Shrestha & Orchard (2018),
Bellec, Salaj, Subramoney, Legenstein, and Maass (2018)
Neftci, Mostafa, & Zenke (2019)

In ML: “Straight-through estimators” Bengio et al. (2013)

Option 1: Noise injection. Pfister, Toyoizumi, Barber & Gerstner (2006)
Gardner, Sporea & Grüning (2015)

Option 2: Differentiate firing times.
Bohte, Kok, & Poutre (2002), Gütig & Sompolinski (2006), Gütig (2016), Mostafa (2018)

Option 3: Make spikes differentiable.
Huh & Sejnowski (2018)

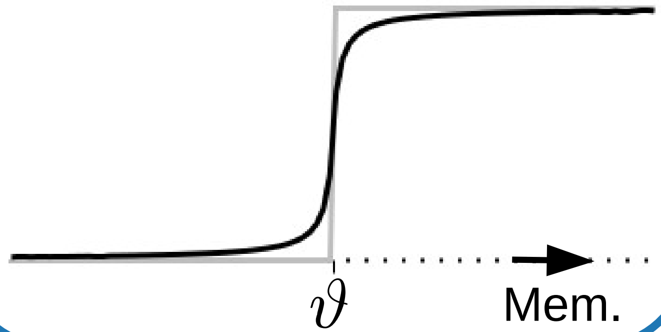
Option 4: Force hidden units “on target”.
Gilra & Gerstner (2017), Nicola & Clopath (2017)

Many more: e.g. firing-rate approaches
Hunsberger & Eliasmith (2015), Lee et al. (2016), ...

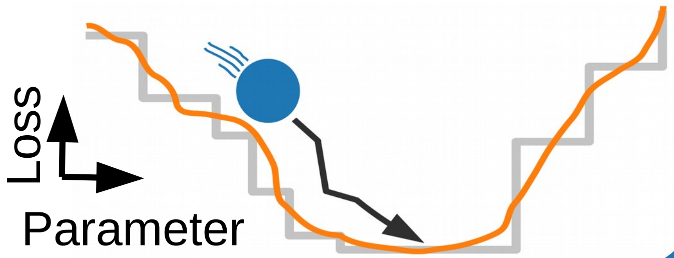
Today: Surrogate gradients.
Bohte (2011), Zenke & Ganguli (2018),
Shrestha & Orchard (2018),
Bellec, Salaj, Subramoney, Legenstein, and Maass (2018)
Neftci, Mostafa, & Zenke (2019)

In ML: “Straight-through estimators” Bengio et al. (2013)

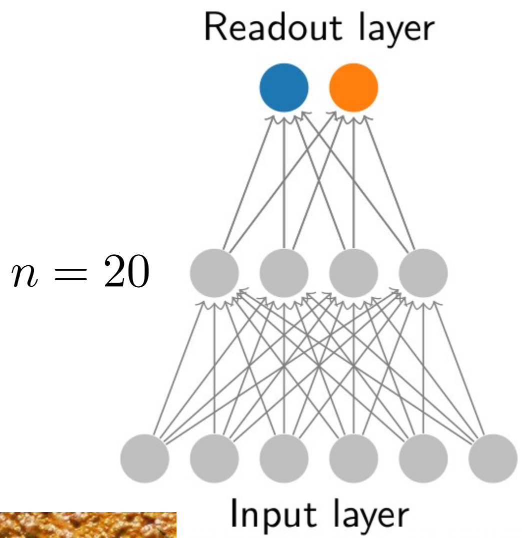
$$\Theta(U(t)-\vartheta) \approx \sigma(U(t))$$

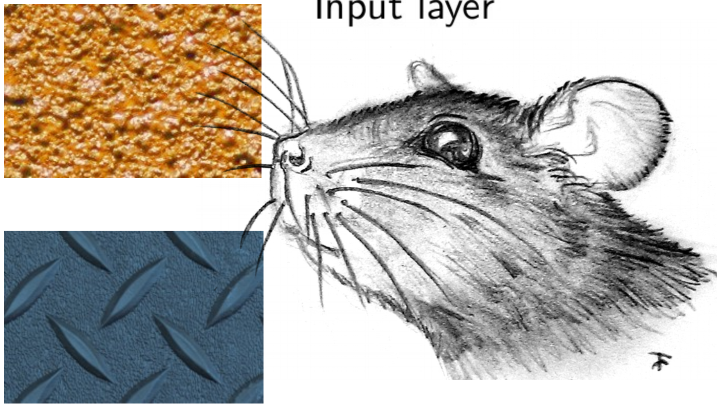


$$\frac{\partial \mathcal{L}}{\partial w_k} \neq 0$$

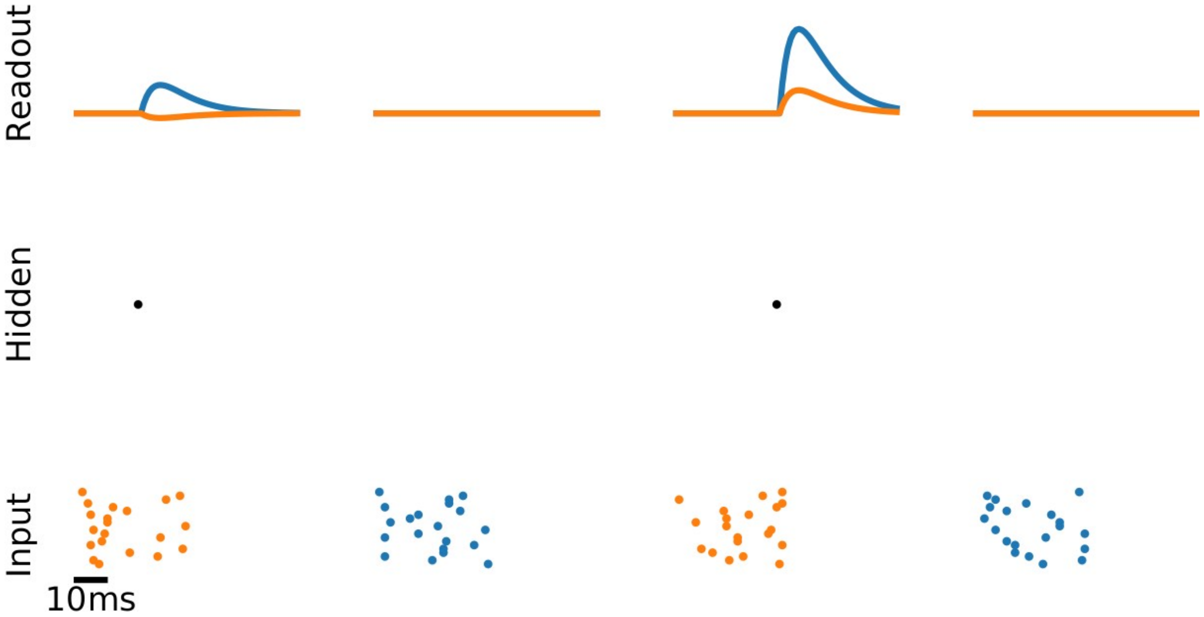


A two-class classification problem



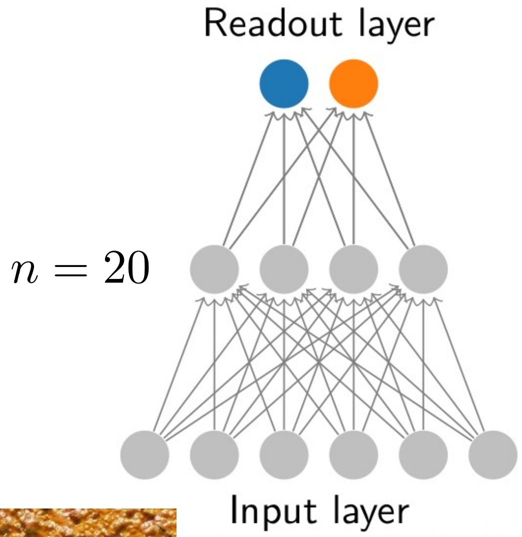


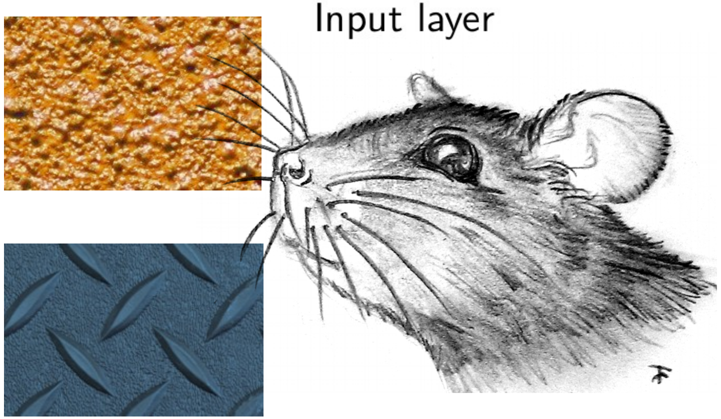
Activity snapshots (network has not learned anything)



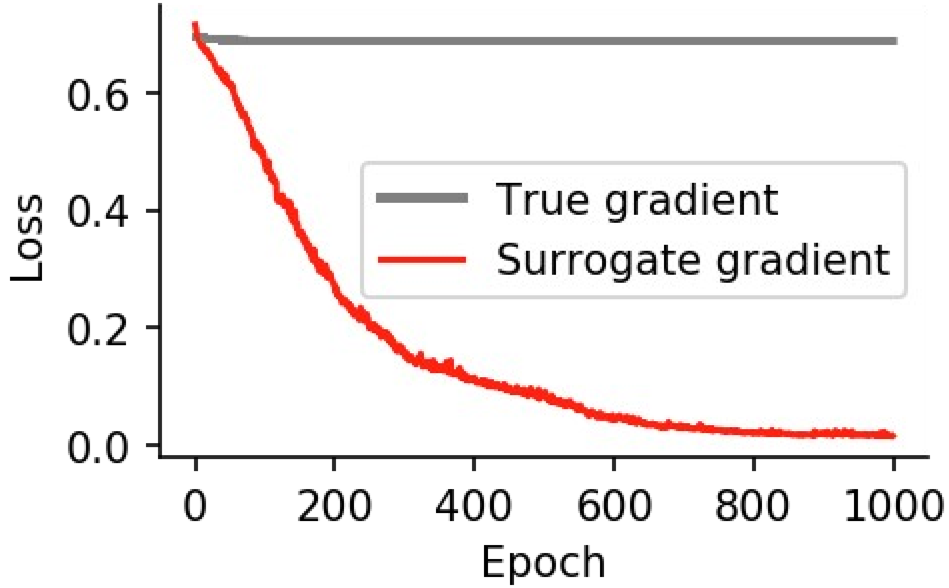
Synthetic data set: 2000 samples from two smooth random manifolds

A two-class classification problem

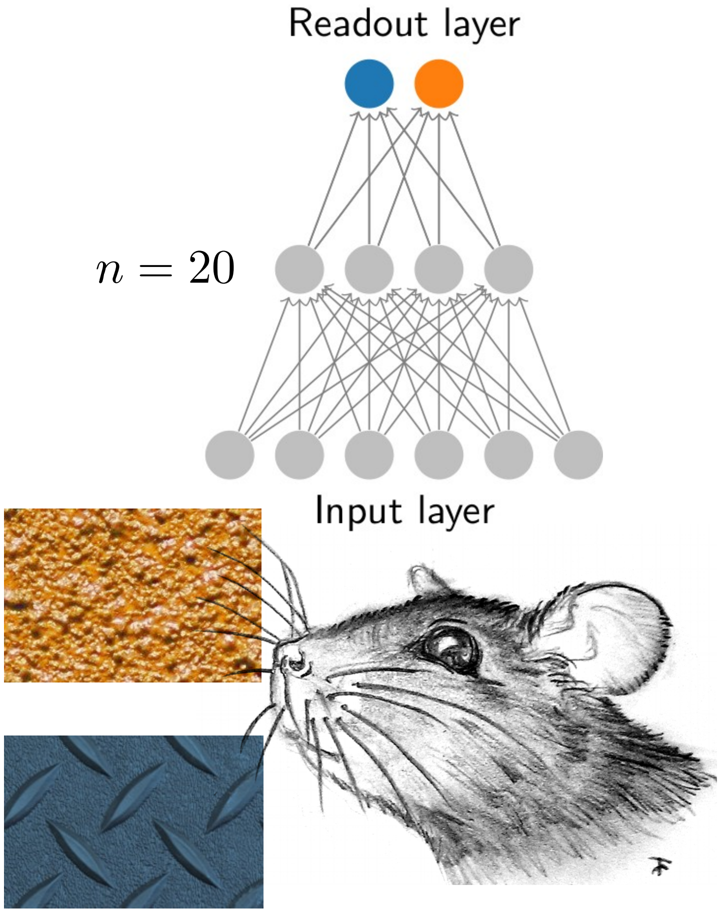




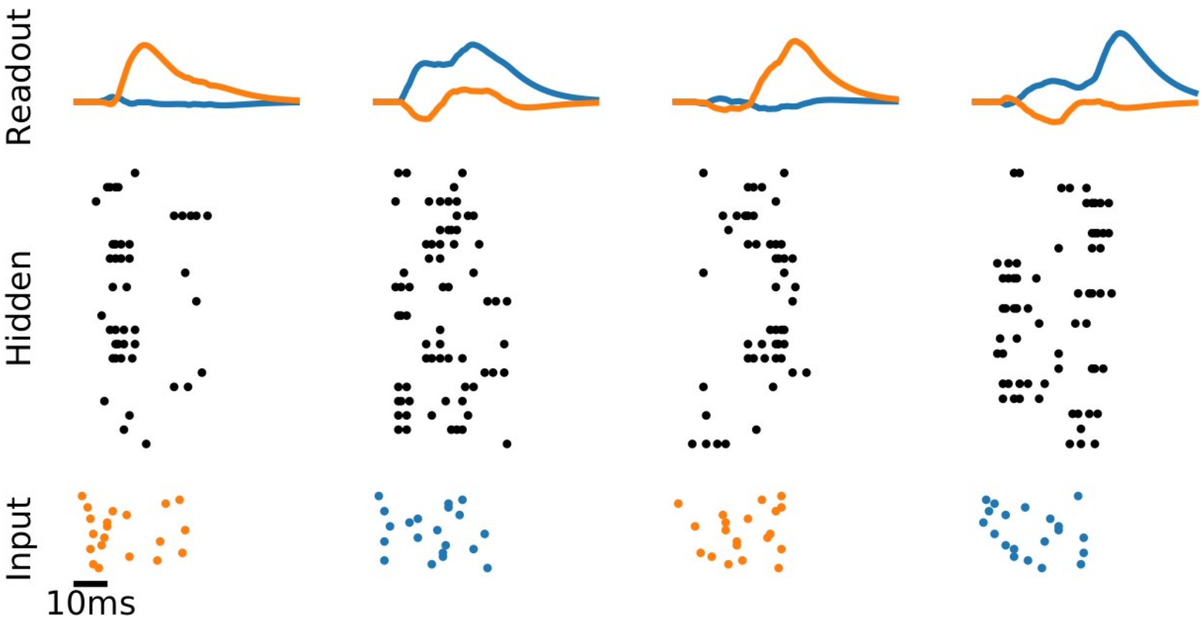
Evolution of loss during surrogate gradient descent



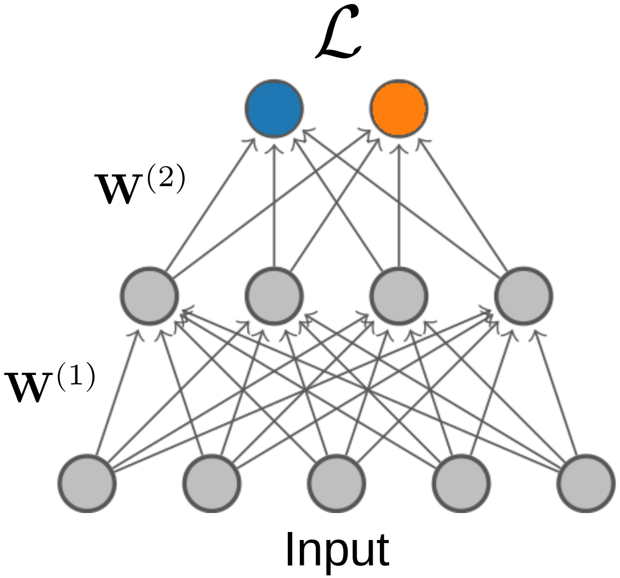
A two-class classification problem

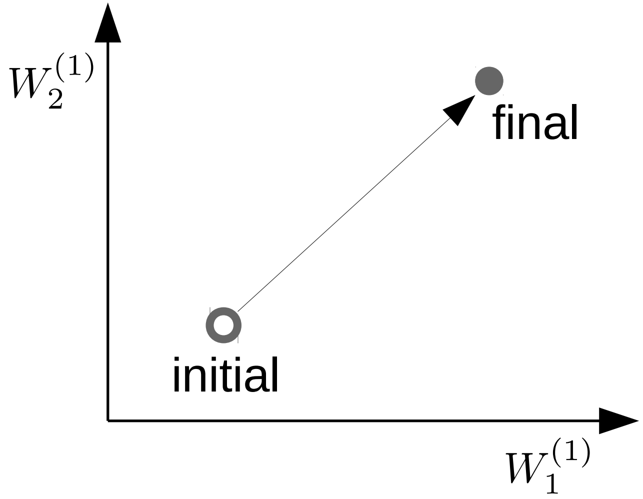


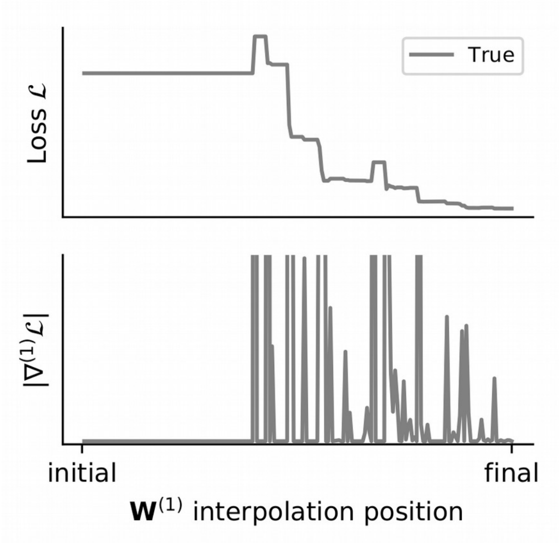
Activity snapshots (trained network)



The loss landscape of a spiking neural network

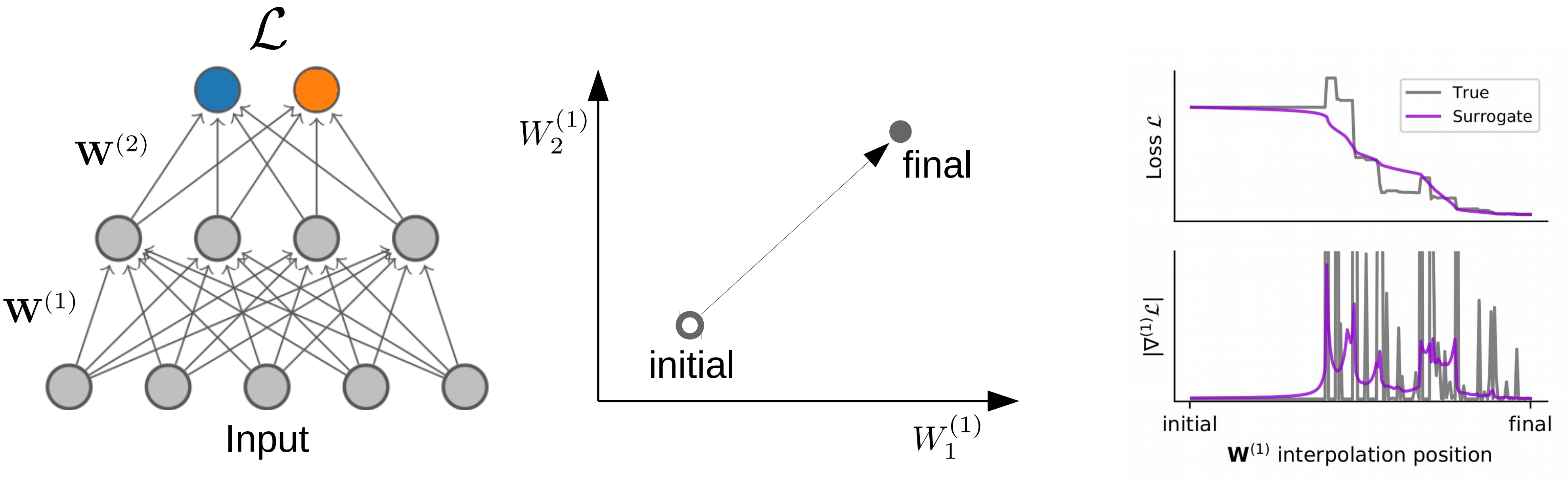






Neftci, Mostafa, & Zenke (in press)

The loss landscape of a spiking neural network



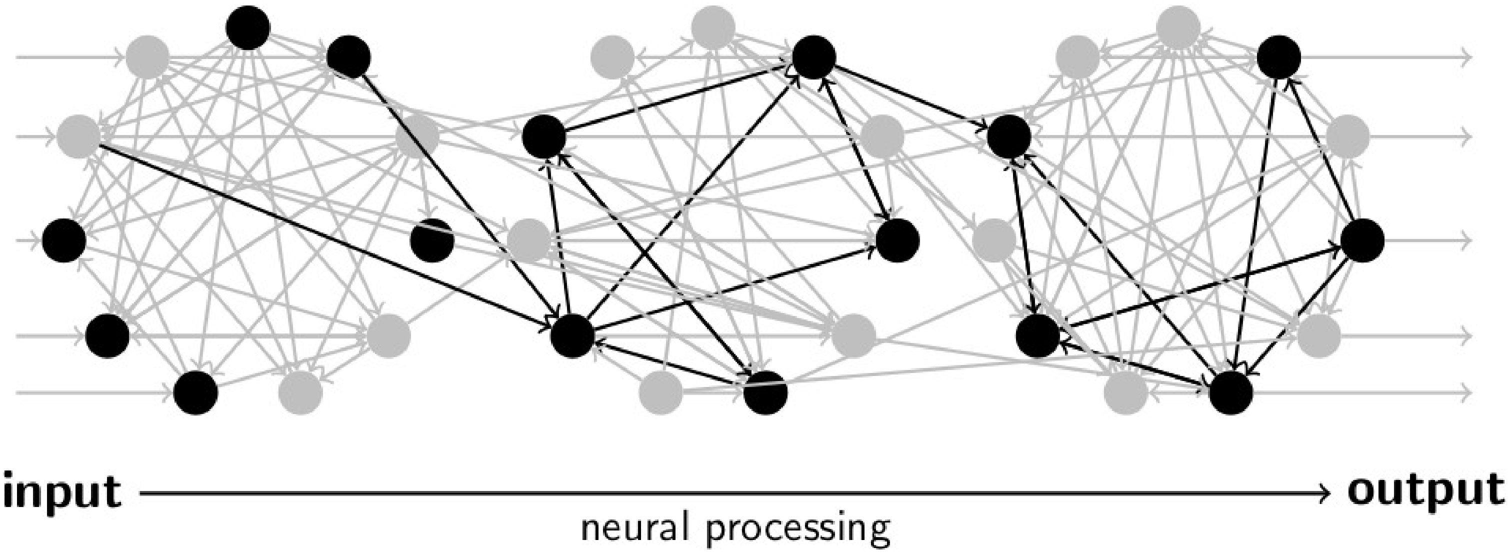
Neftci, Mostafa, & Zenke (in press)

Towards functional neural network models

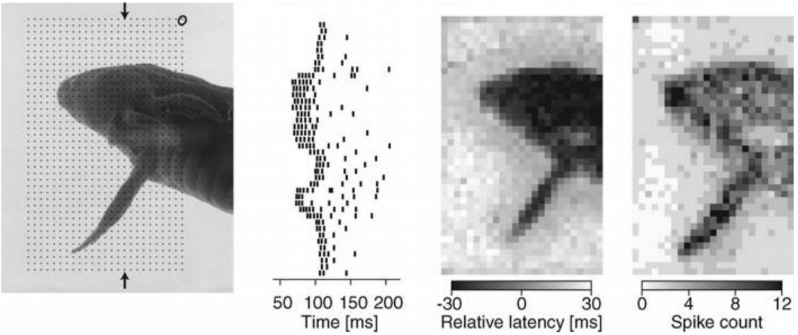
1) Input
(spatiotemporal)

3) Adjust weights
(surrogate gradients)

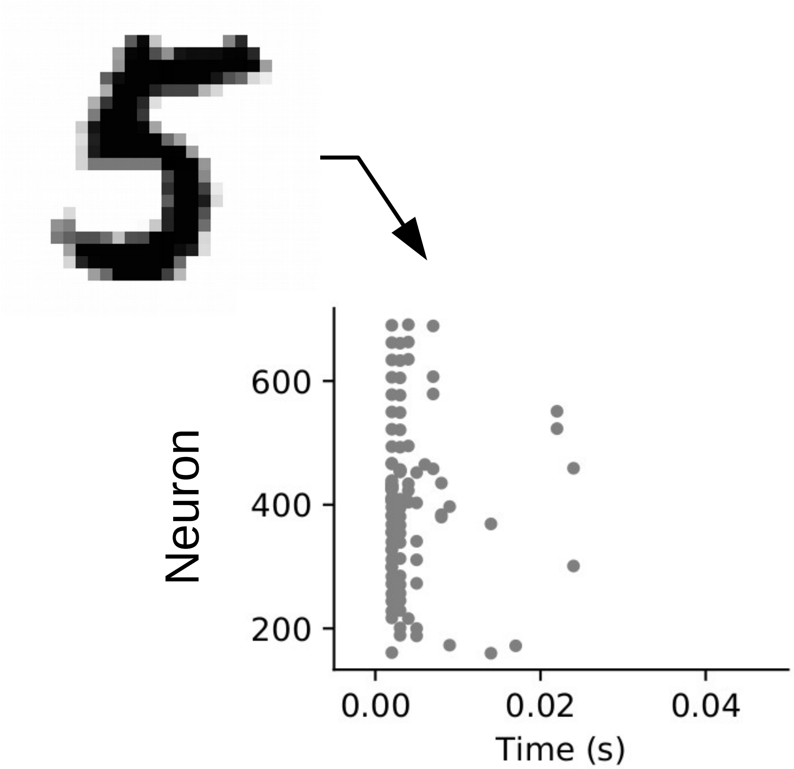
2) Output
(classification)

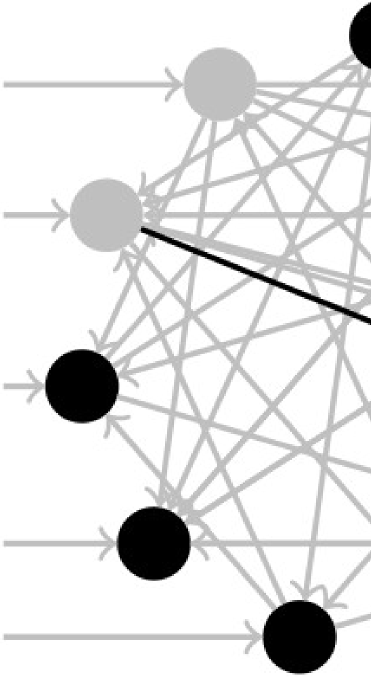


Input: Spatiotemporal spike patterns



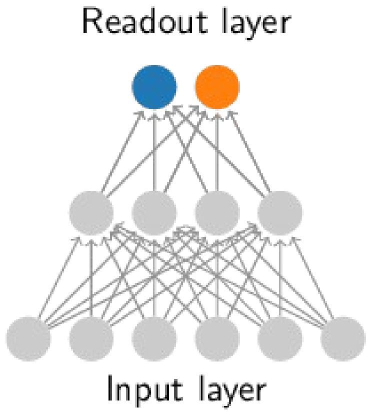
Gollisch & Meister (2008)

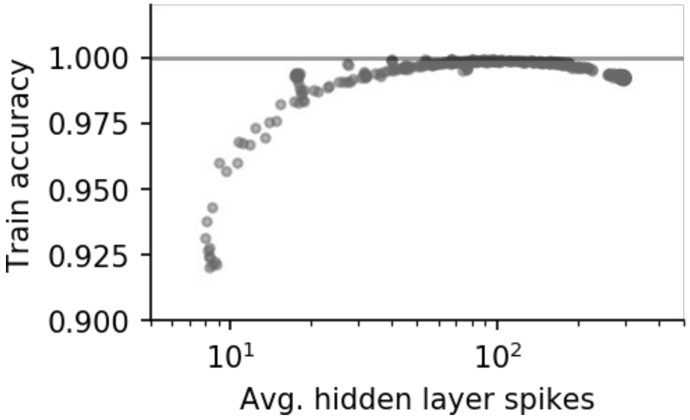


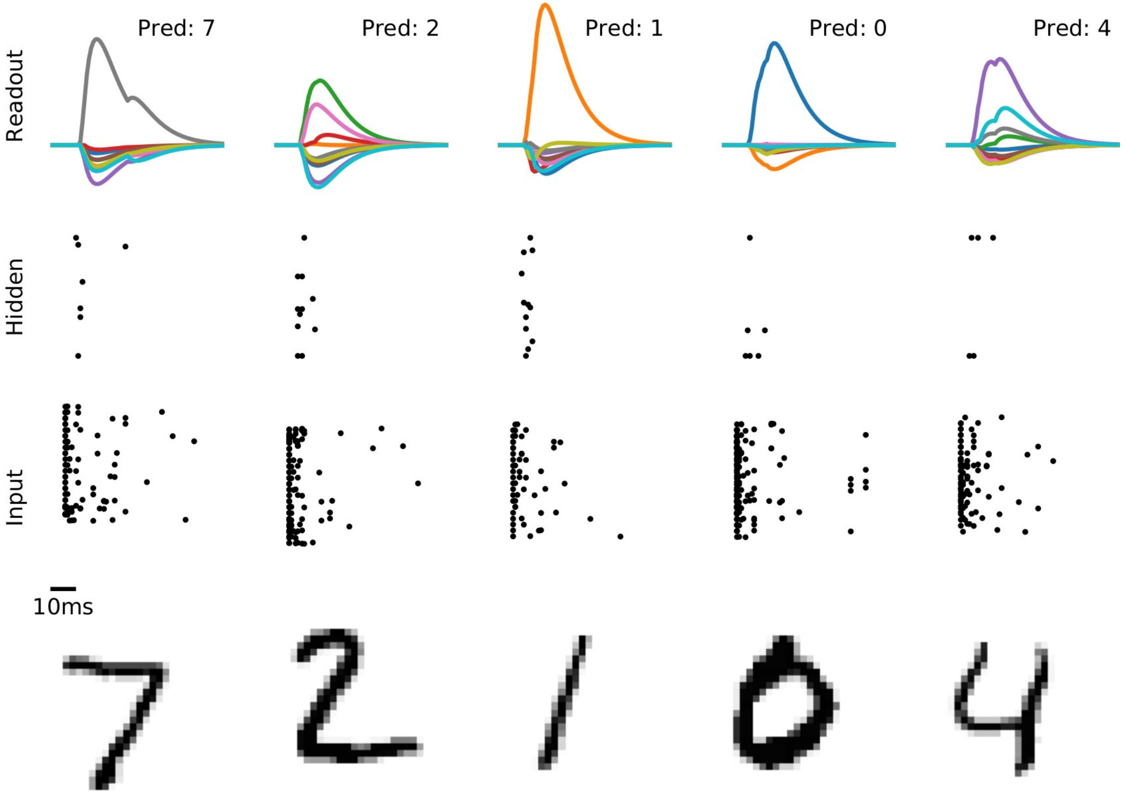


input

MNIST is solved with a handful of spikes

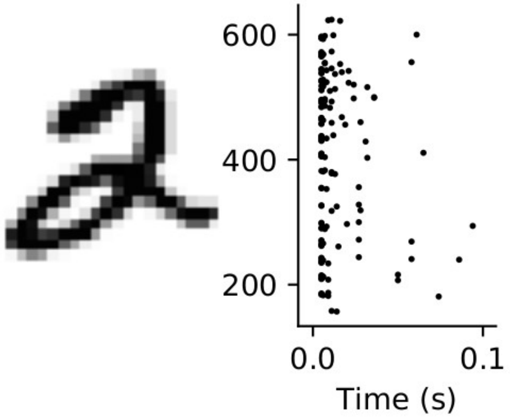


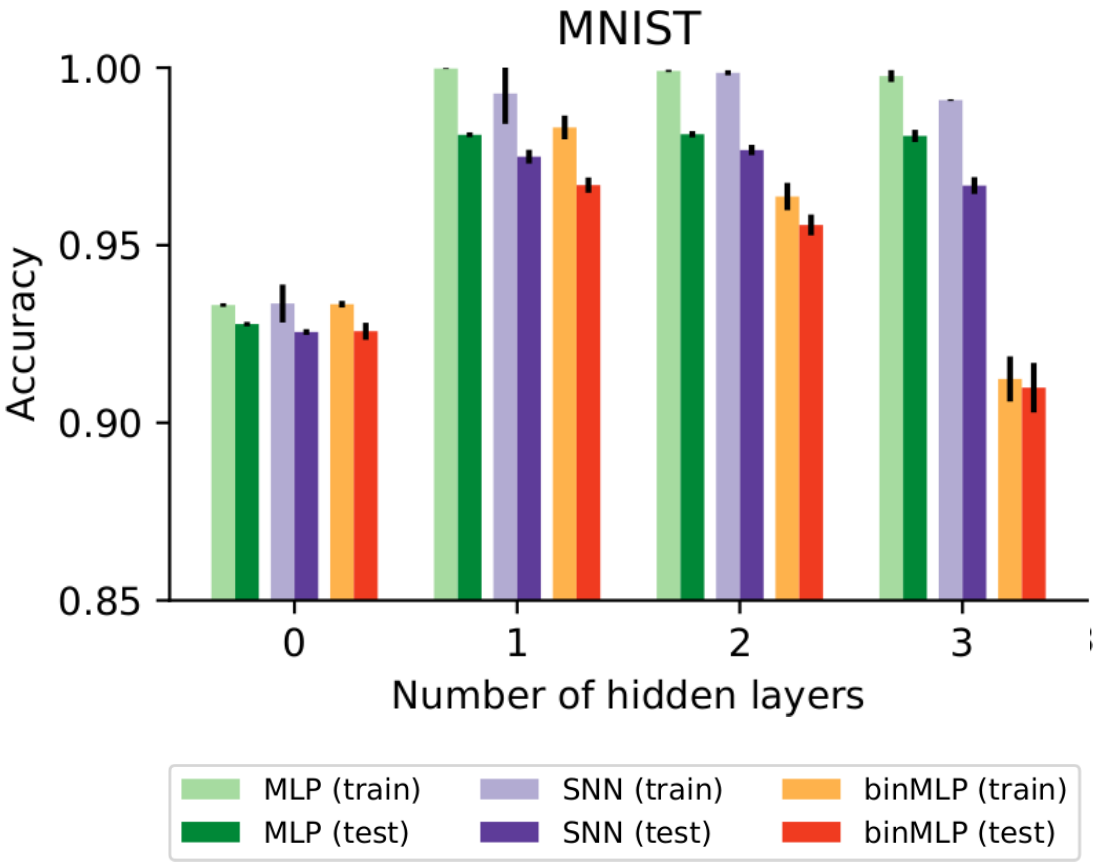




Benchmarks

MNIST
LeCun, Cortes & Burges (1998)

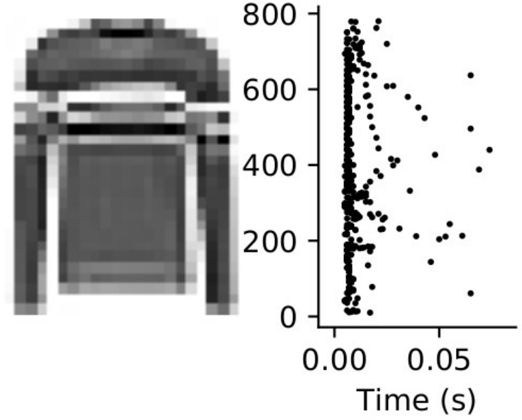


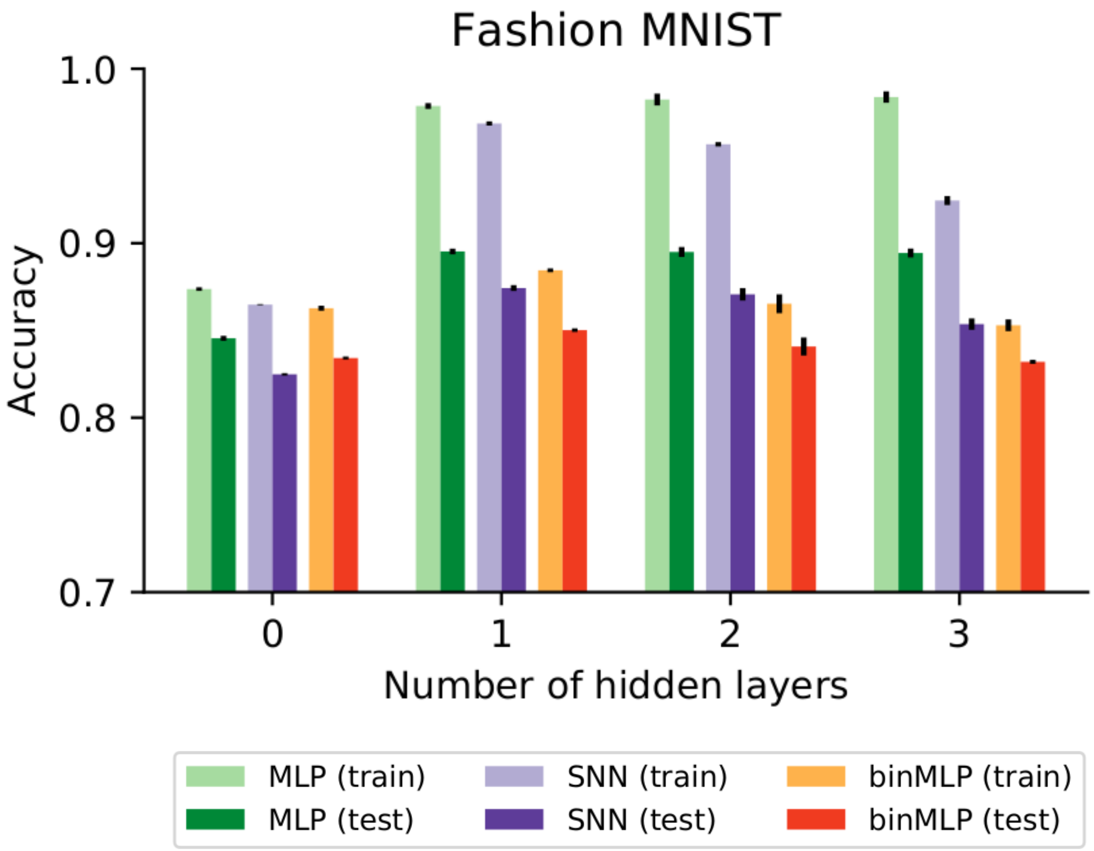


Zenke et al. (in prep.)

Benchmarks (2)

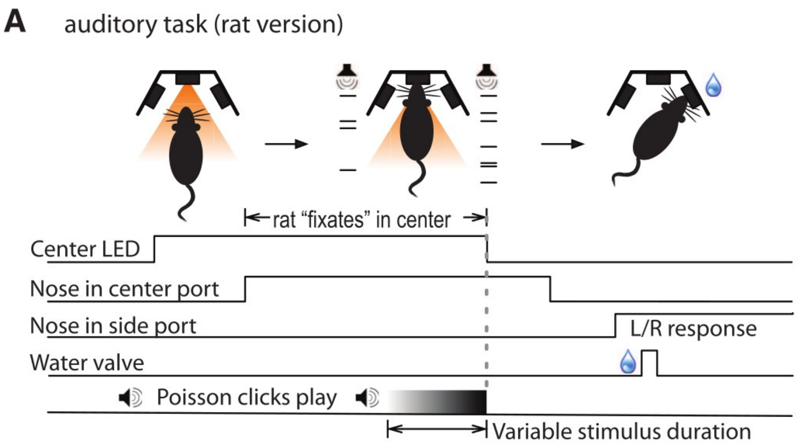
Fashion MNIST
Xiao, Rasul & Vollgraf (2017)





Zenke et al. (in prep.)

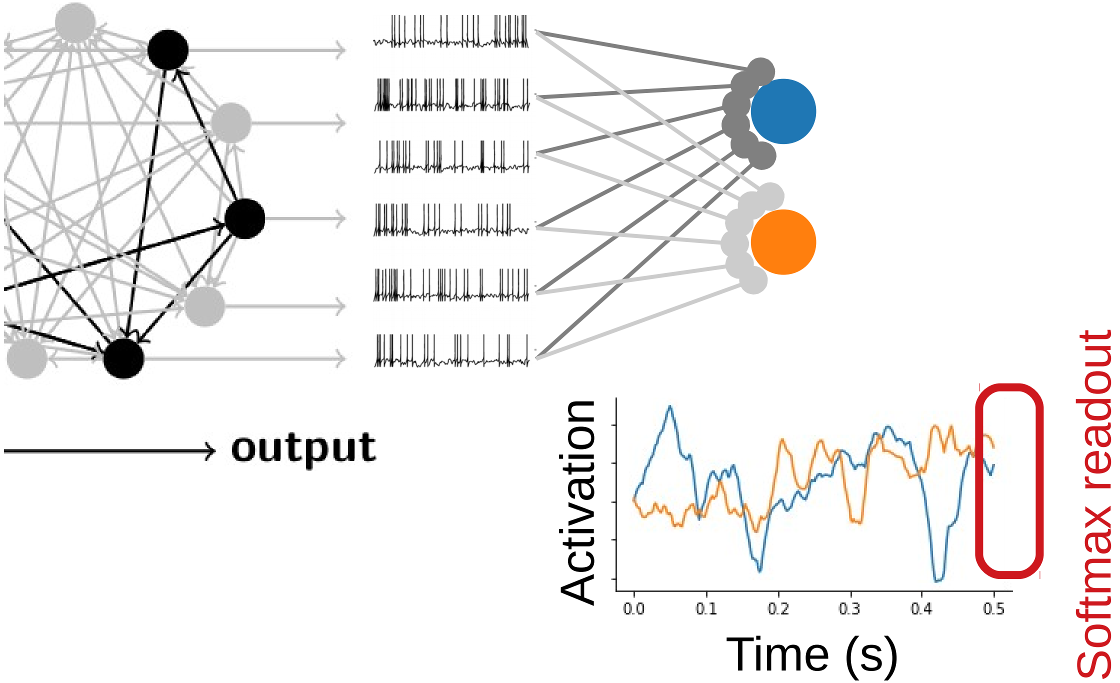
Back to bio-inspired: A DM problem



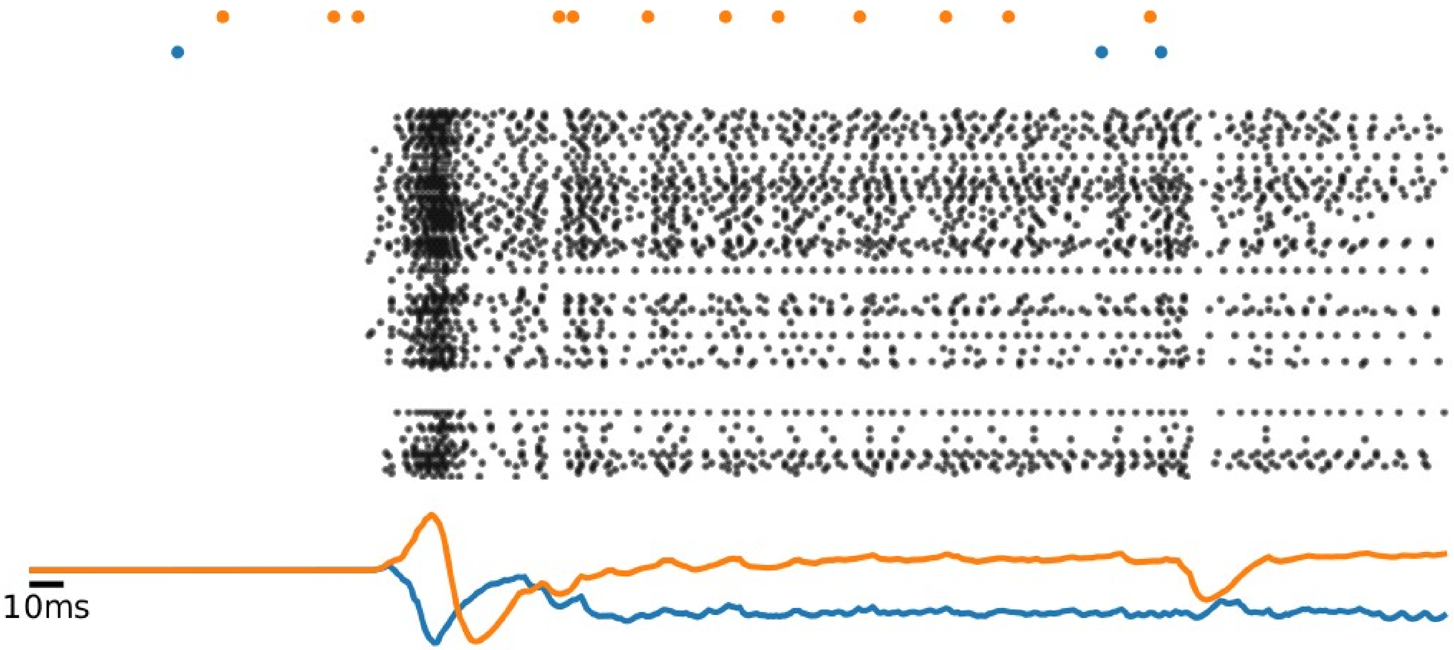
Brunton, Botvinick, and Brody (2013)

+ Short-term plasticity

Tsodyks & Markram (1997)



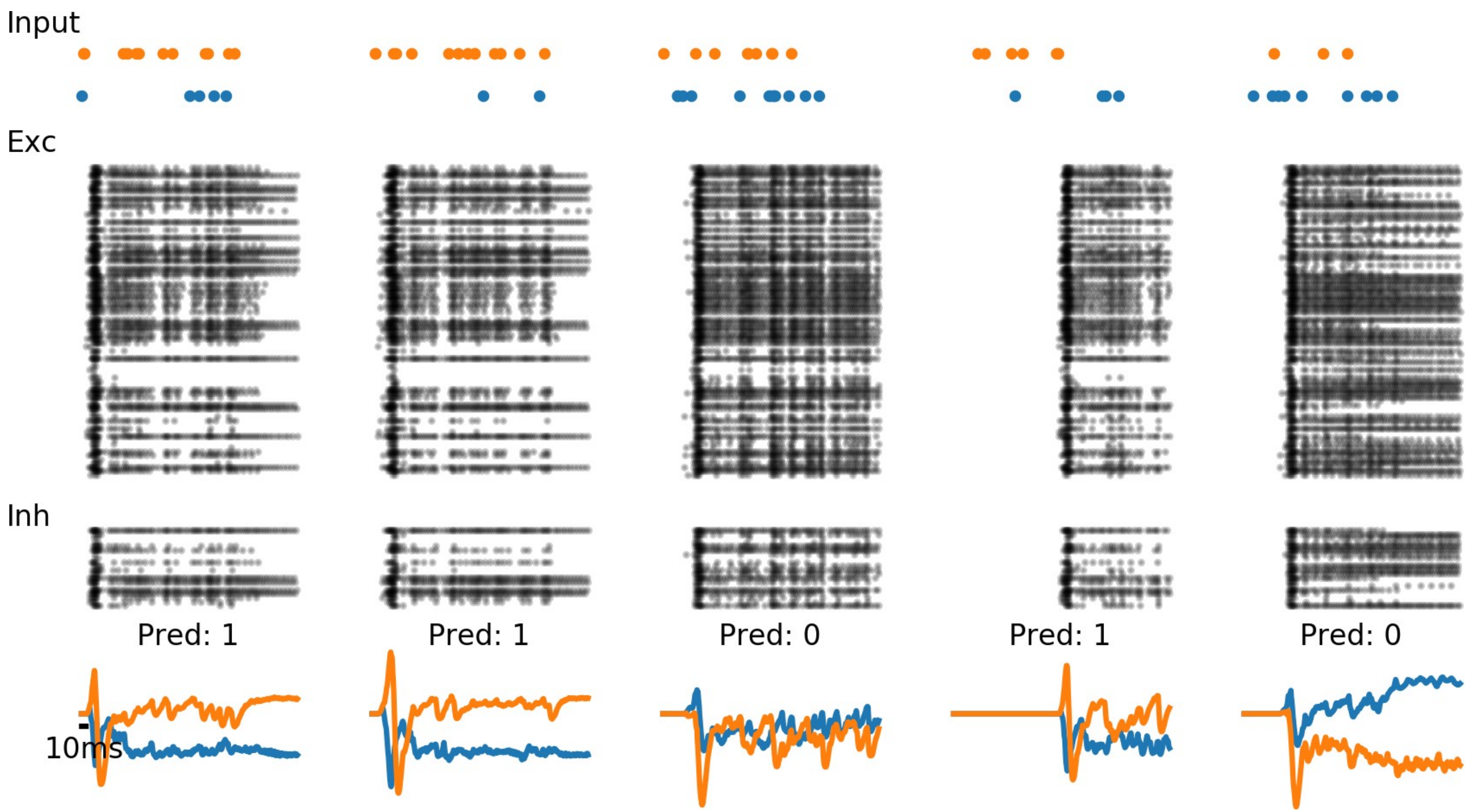
Activity snapshot for single decision making trials



Zenke et al. (in prep.)

Network learns to use delay activity

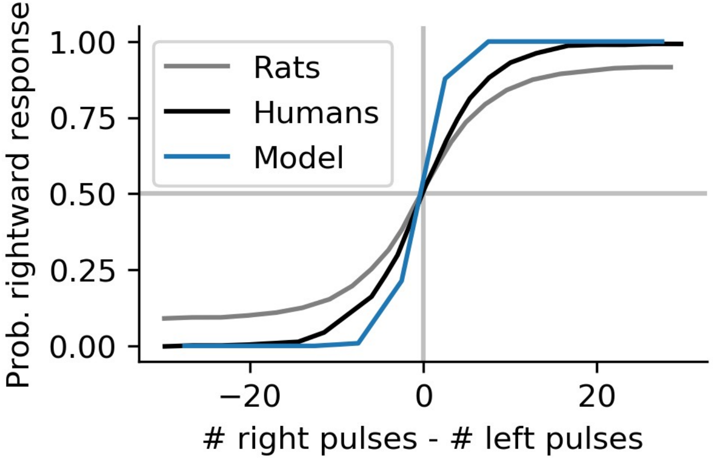
Activity snapshots for single decision making trials



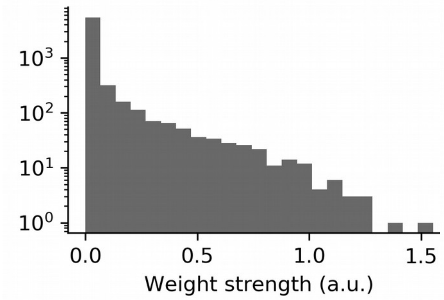
Zenke et al. (in prep.)

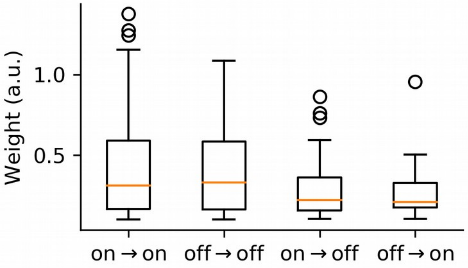
Network learns to use delay activity

Spiking network solves the random clicks task

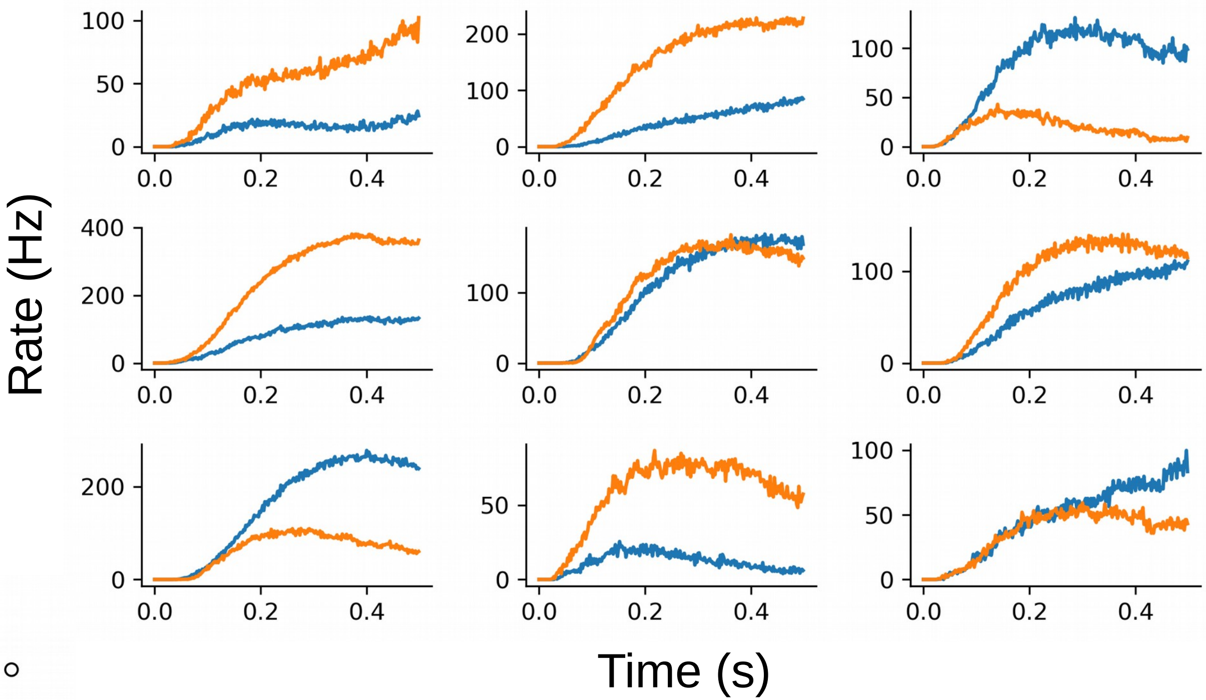


Data: Brunton, Botvinick, and Brody (2013)



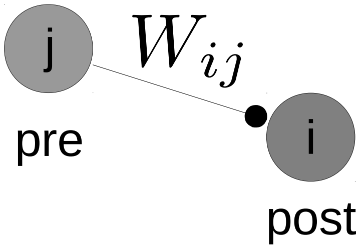


Single cell PSTHs

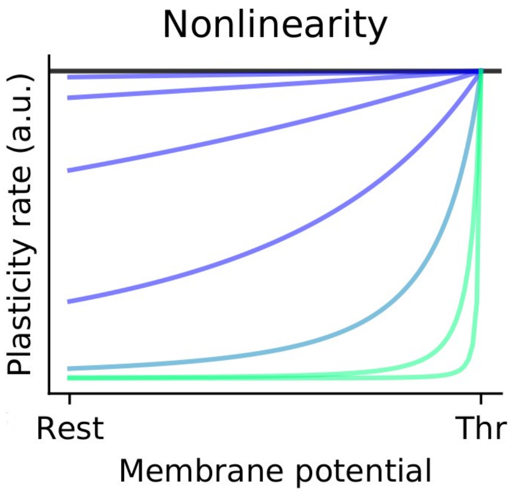


Zenke et al. (in prep.)

Surrogate gradient learning is robust to the choice of voltage nonlinearity

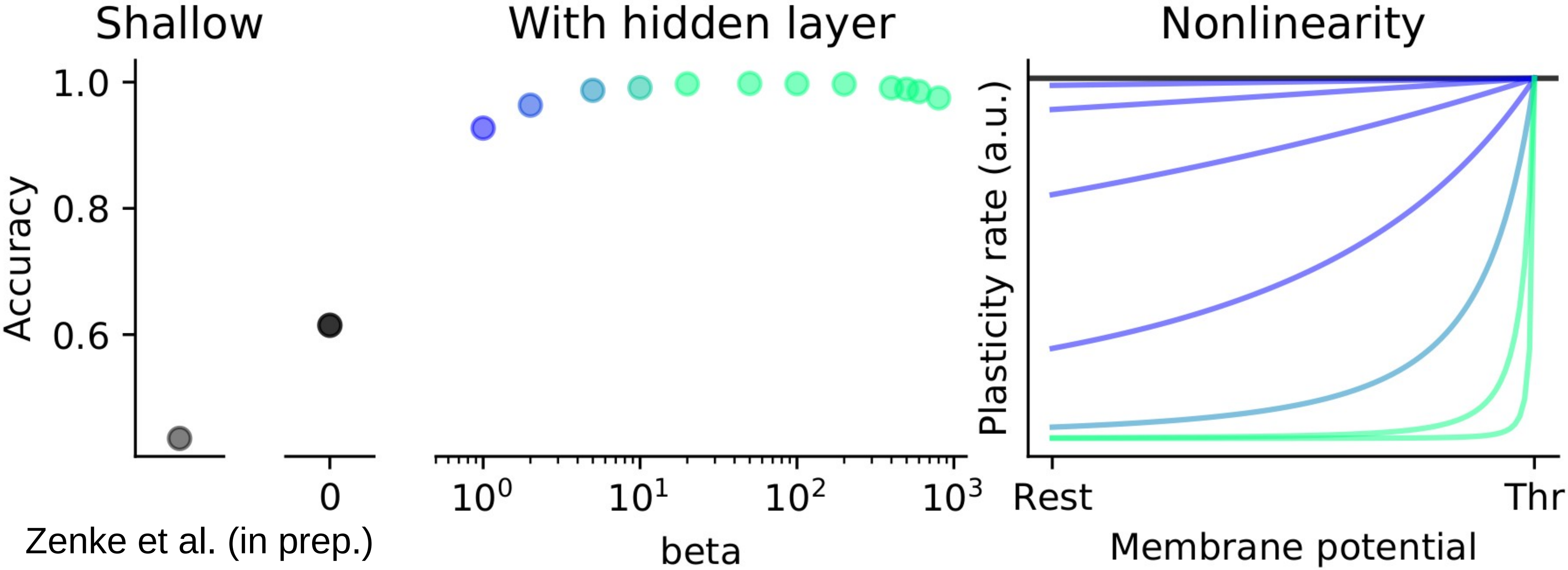


$$\Delta W_{ij} \propto (\mathrm{pre}_j)\boxed{f(U_i^{\mathrm{post}})}(\mathrm{feedback}_i)$$



Zenke et al. (in prep.)

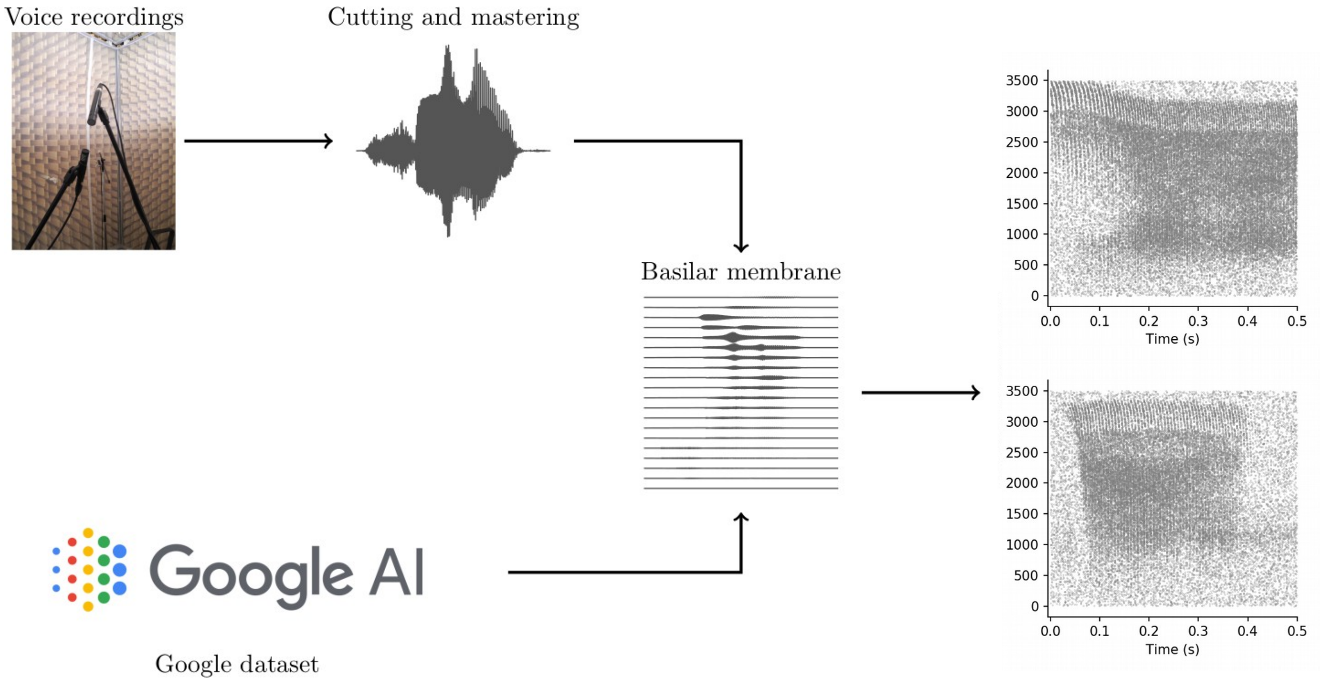
$$\Delta W_{ij} \propto (\mathrm{pre}_j)\boxed{f(\mathrm{post}_i)}(\mathrm{feedback}_i)$$



Zenke et al. (in prep.)

Benchmarks: The need for objective comparison of spiking networks

In collaboration with
Benjamin Cramer
Kirchhoff Institute of Physics
Uni Heidelberg



Spiking benchmark data sets
- Spoken digits & commands German/English
- More than 100k examples
- Spikes from cochlea model (3.5k channels)

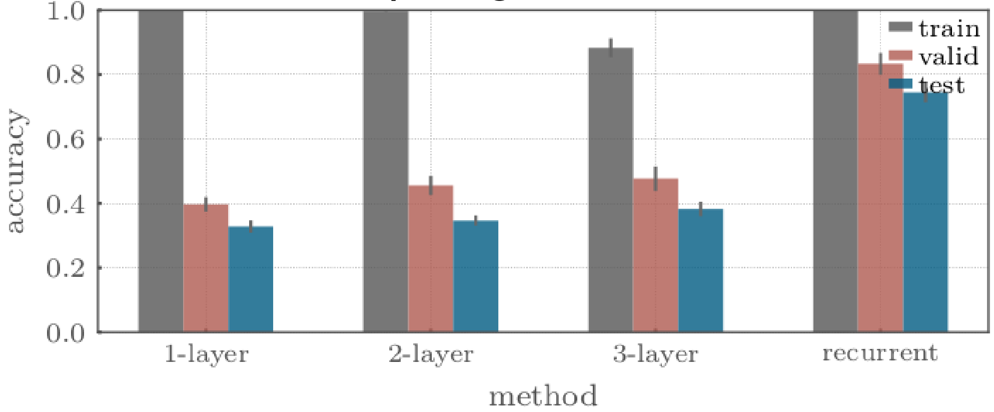
Cramer, Stradmann, Schemmel & Zenke (in prep.)

Benchmark results

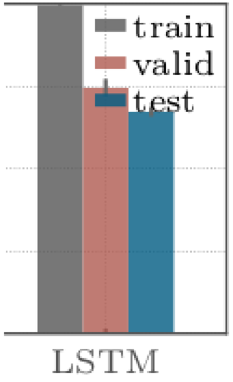
In collaboration with
Benjamin Cramer
Kirchhoff Institute of Physics
Uni Heidelberg

Preliminary

Spiking networks



LSTM



Cramer, Stradmann, Schemmel & Zenke (in prep.)

Summary & Outlook

- End-to-end training of spiking neural networks using surrogate gradients
- Learning is robust, but a nonlinear voltage-dependent learning rule is required
- What next …?
  - Study representation in functional spiking networks
  - Elucidate feedback channels
  - Study unsupervised cost functions (e.g. prediction)

Thanks

Post-doc advisors

Stanford University

Surya Ganguli and the Gang

UNIVERSITY OF OXFORD

Tim Vogels and Group

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

Emre Neftci, UC Irvine

Funding: wellcome

Code & Tutorials: fzenke.net

Artwork: K. Yadava (kyadava.net)