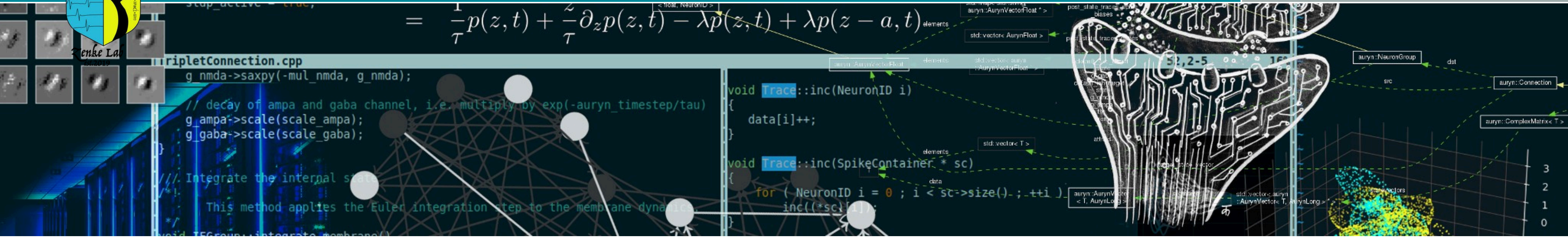


Finding the needle in the haystack

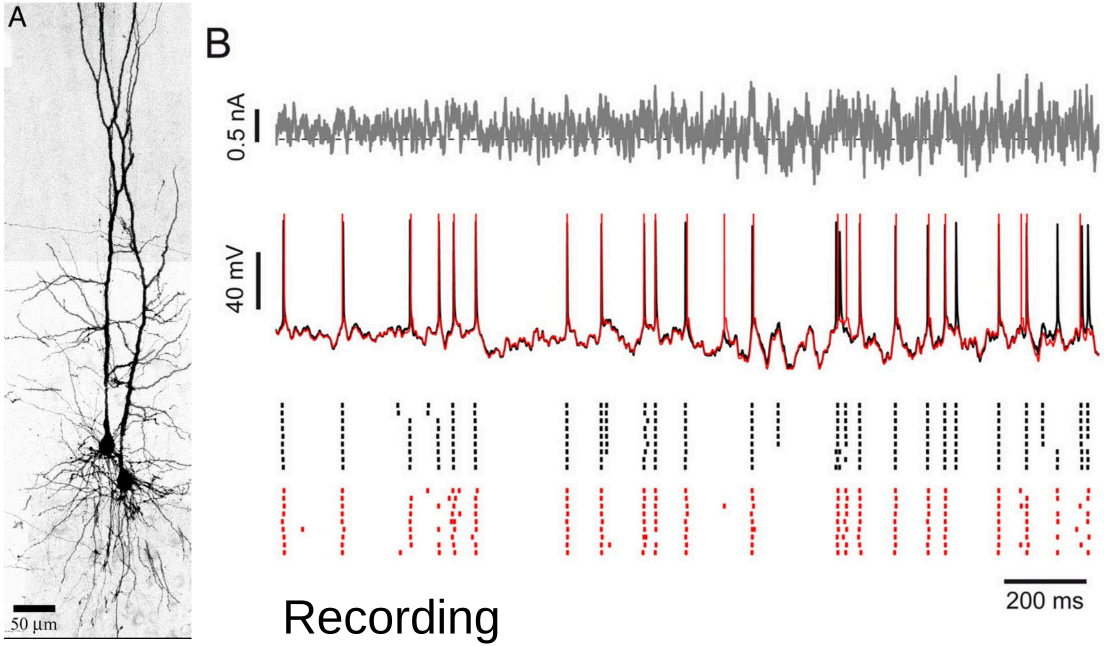
Functional circuit and network models for neuroscience

Friedemann Zenke

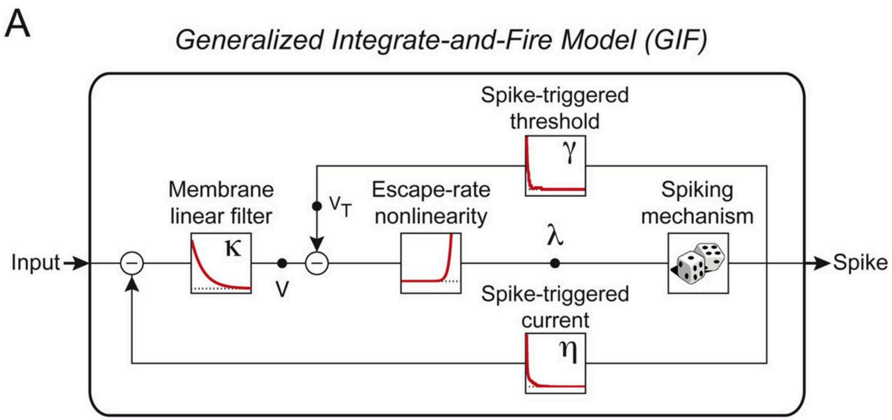
Computational Neuroscience @ FMI

www.zenkelab.org

Single neuron models are pretty good



Recording
Model



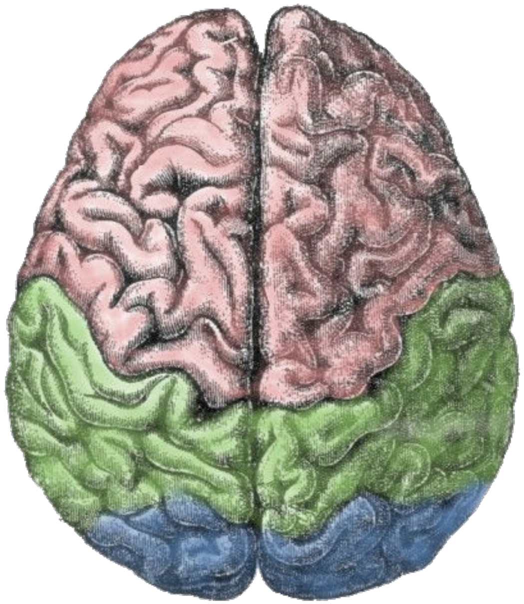
Pozzorini, C., Mensi, S., Hagens, O., Naud, R., Koch, C., and Gerstner, W. (2015)

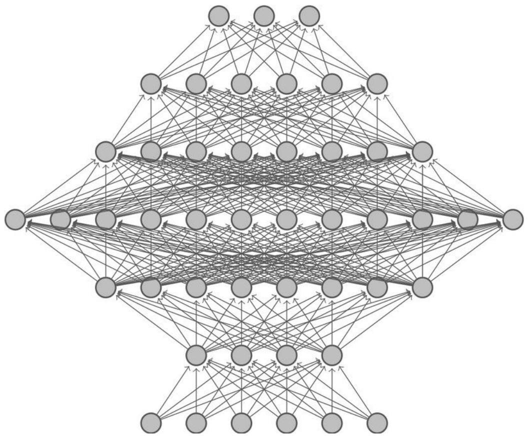
FMI
Friedrich Miescher Institute
for Biomedical Research

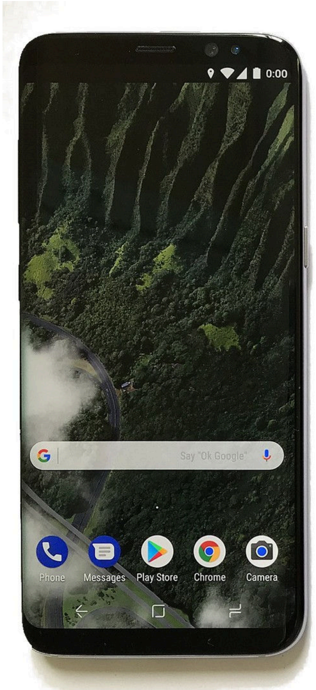
Yet we lack a similarly comprehensive understanding of neural network dynamics

Biological neural networks

Artificial neural networks







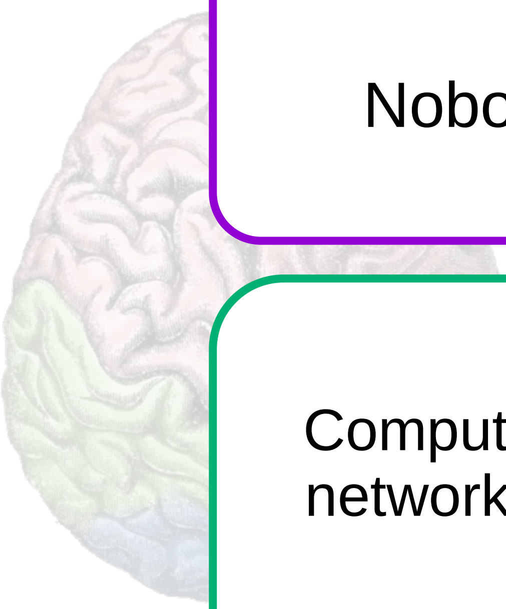
Source: https://en.wikipedia.org/wiki/Human_brain

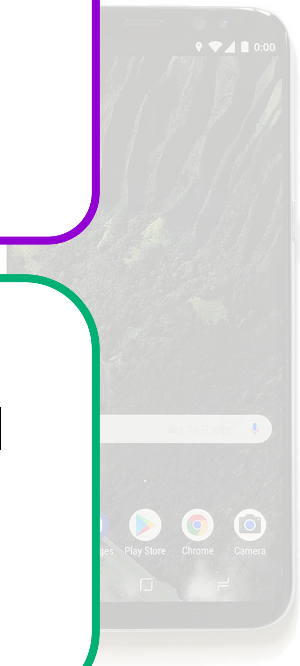
Source: https://simple.wikipedia.org/wiki/Smartphone

Yet we lack a similarly comprehensive understanding of neural network dynamics

Biological neural networks

Artificial neural networks





Issue

Nobody *really* understands how neural networks work.

Conundrum

Computer scientists who engineer deep neural networks do not understand how they operate.

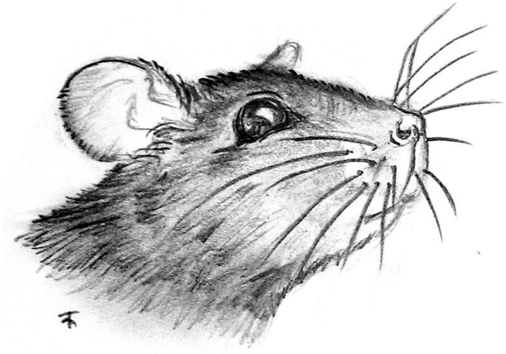
→ **optimization algorithms**

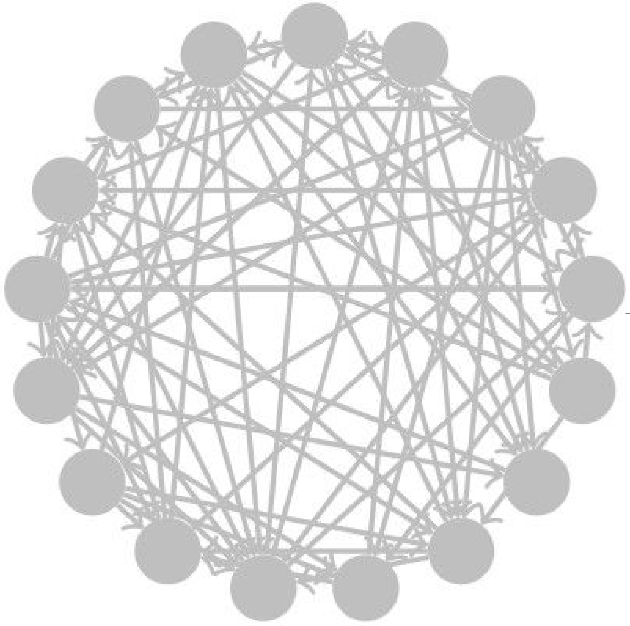
Source: https://en.wikipedi

dia.org/wiki/Smartphone

The space of all possible networks is ginormous

Sensory inputs





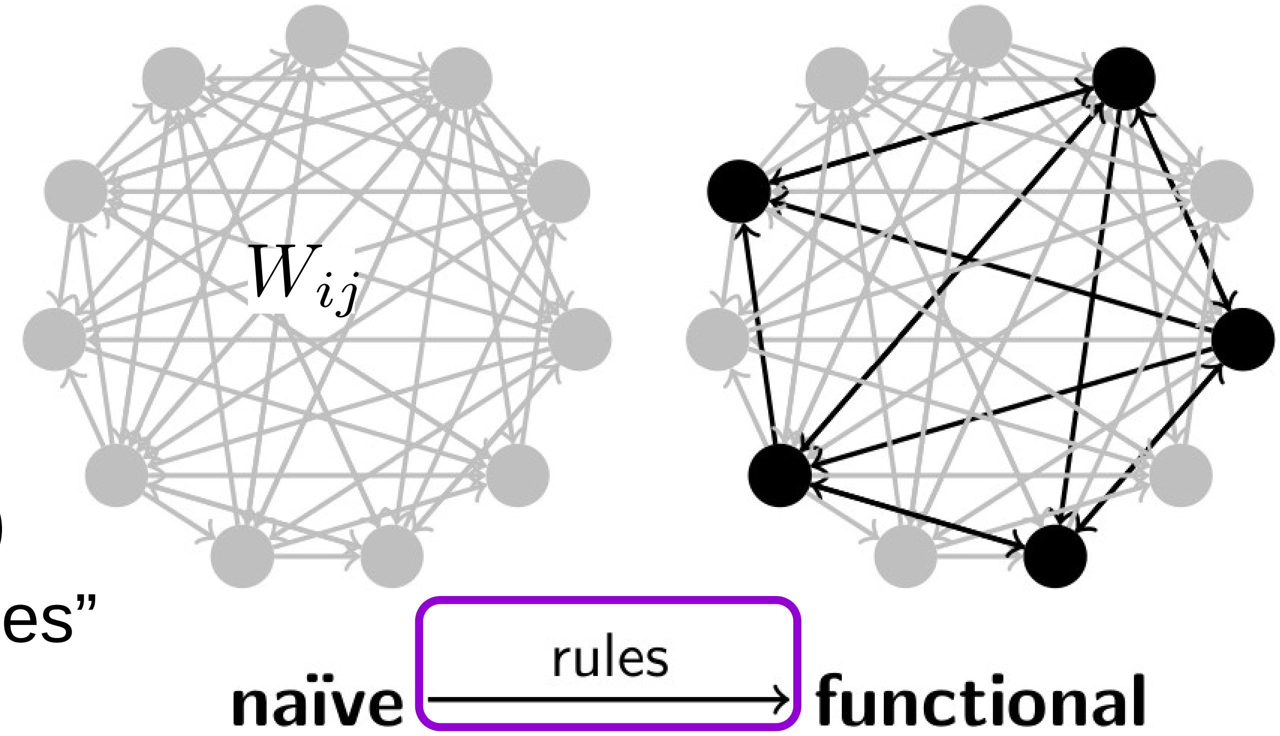
$n = 17$

Behavior

Possible configurations

$$2^{(17^2)} \approx 9.9 \times 10^{86}$$

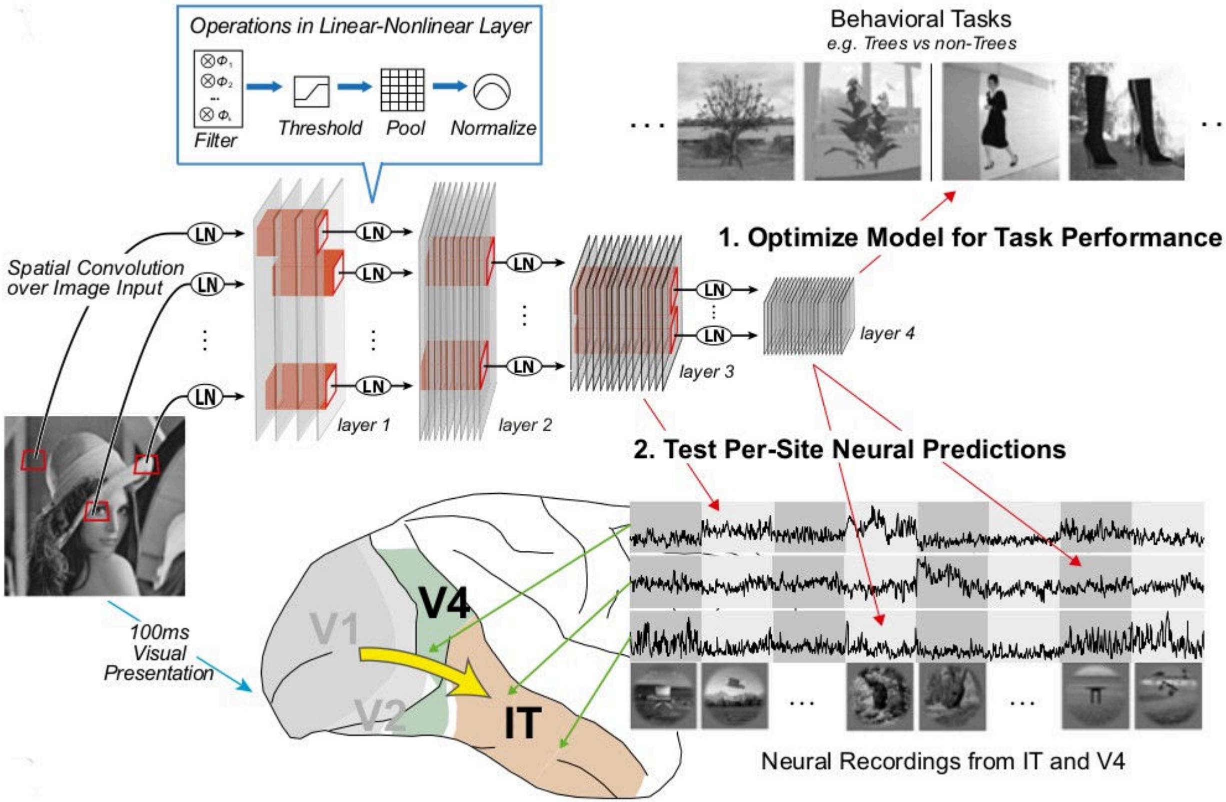
Brains (and modelers) **need to find functional configurations** in this ginormous search space



$$\Delta W_{ij}(t)$$

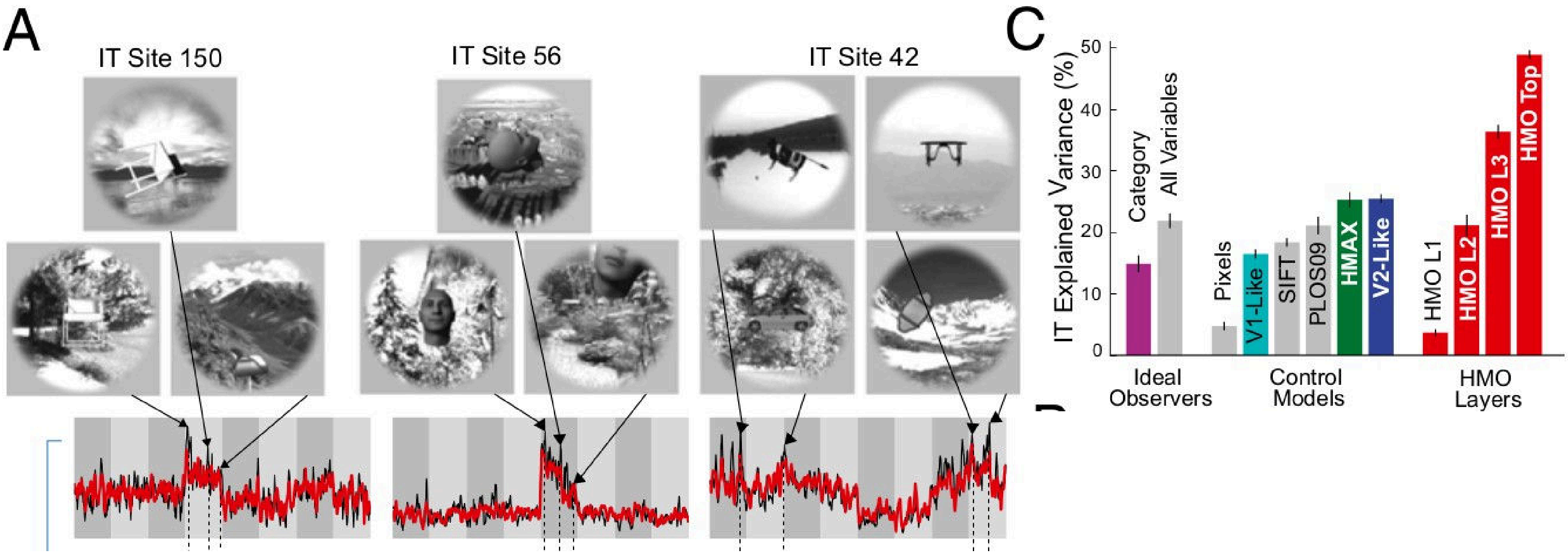
“learning rules”

End-to-end optimization of computational tasks yields predictive models for neuronal activity



Yamins, D.L.K., Hong, H., Cadieu, C.F., Solomon, E.A., Seibert, D., and DiCarlo, J.J. (2014)

End-to-end optimization of computational tasks yields predictive models for neuronal activity



Yamins, D.L.K., Hong, H., Cadieu, C.F., Solomon, E.A., Seibert, D., and DiCarlo, J.J. (2014)

Deep networks are biologically implausible in many respects

Friedrich Miescher Institute for Biomedical Research

- Convolutions, weight sharing, …
- No neuron types (no inhibitory neurons)
- Largely ignore developmental plasticity
- Supervised learning from labeled data
- **Graded activation functions & no spikes**
- …

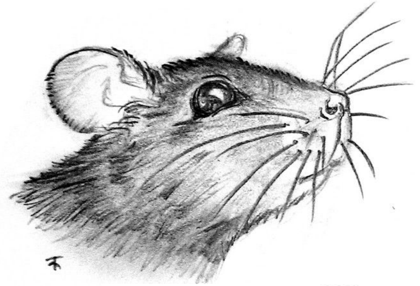
Core questions for today

Friedrich Miescher Institute
for Biomedical Research

- Can we use end-to-end learning …
  - … to build plausible network models?
  - … as a framework to understand neural plasticity?

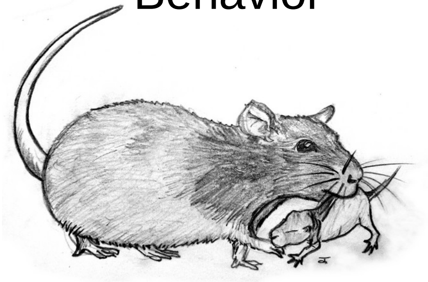
Toward task-optimized plausible networks

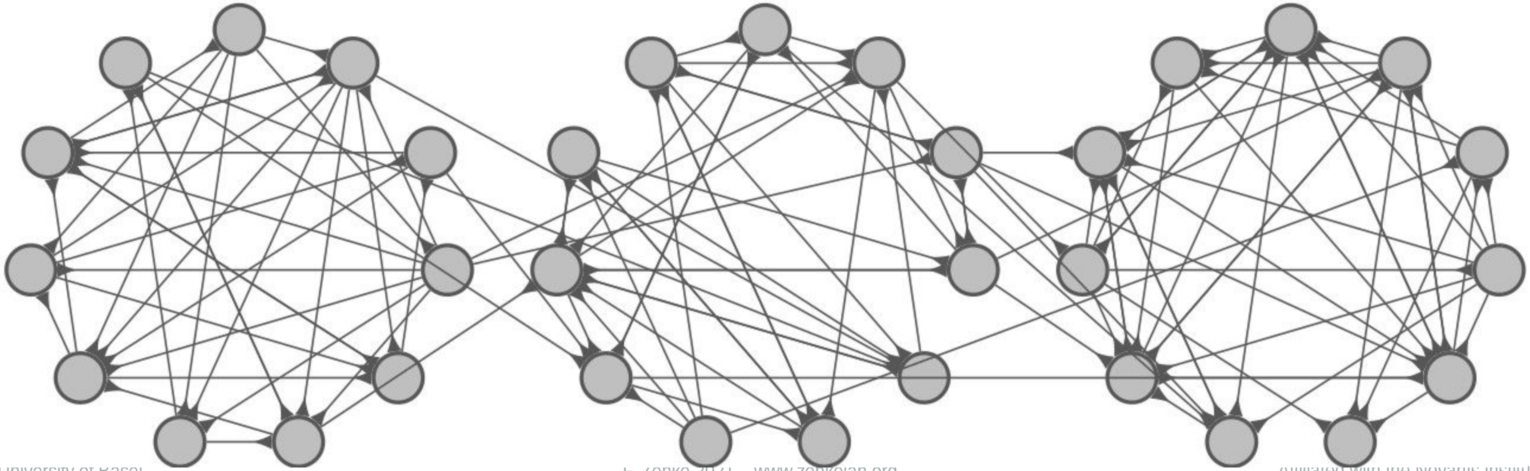
Sensory inputs



Sophisticated function →

Behavior



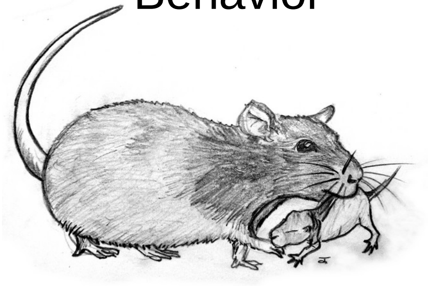


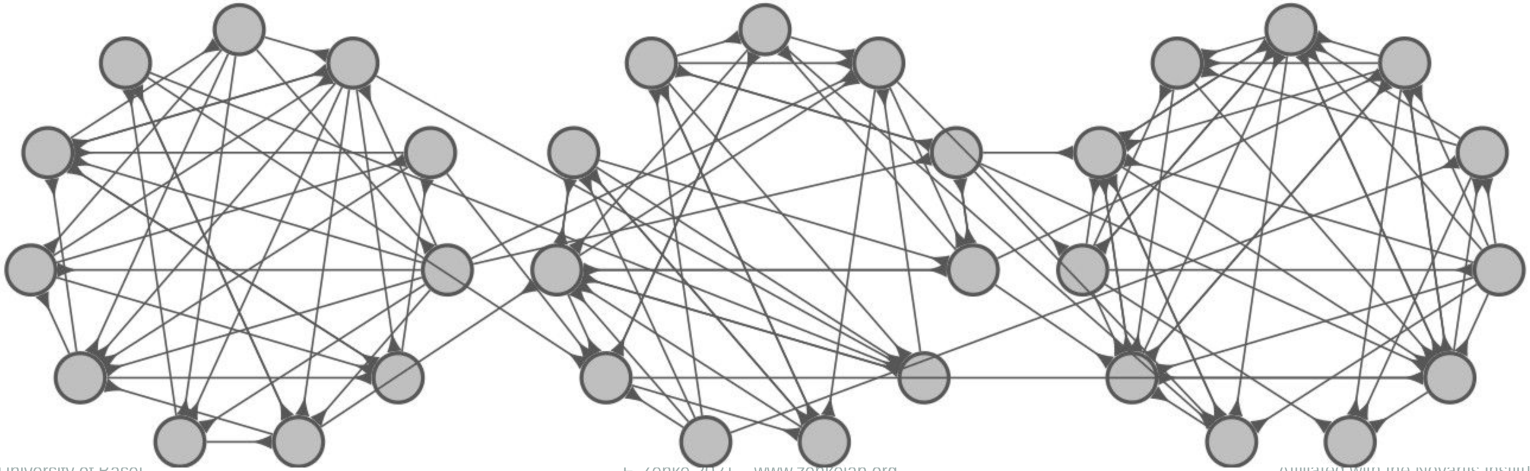
Toward task-optimized plausible networks

1) Input

Sophisticated function →

Behavior

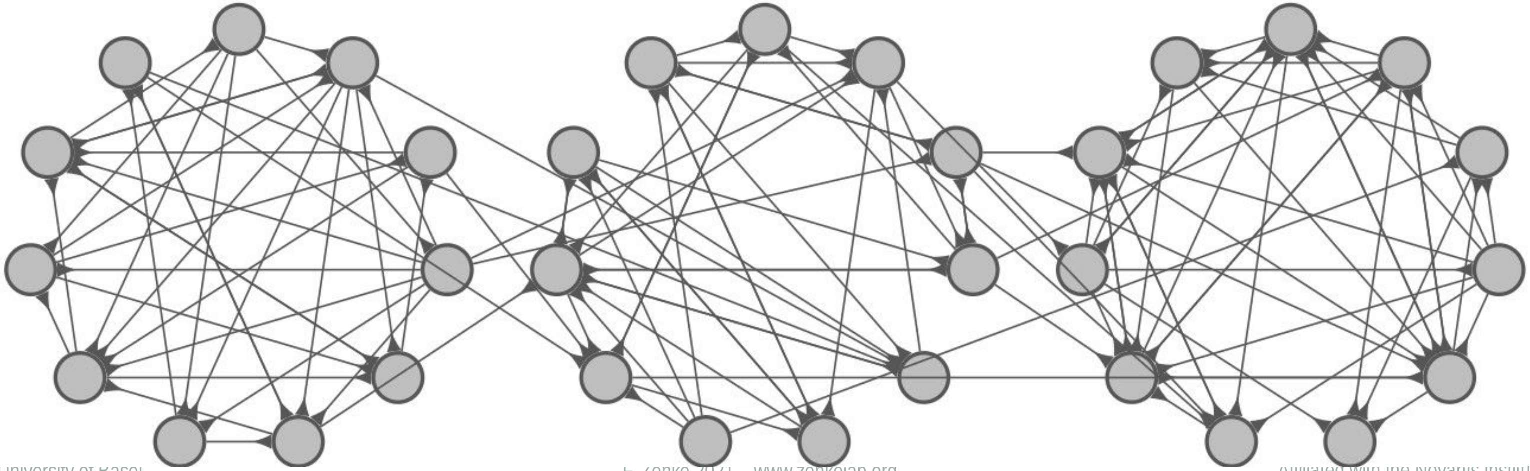




Toward task-optimized plausible networks

Friedrich Miescher Institute for Biomedical Research

1) Input → Sophisticated function → 2) Desired output



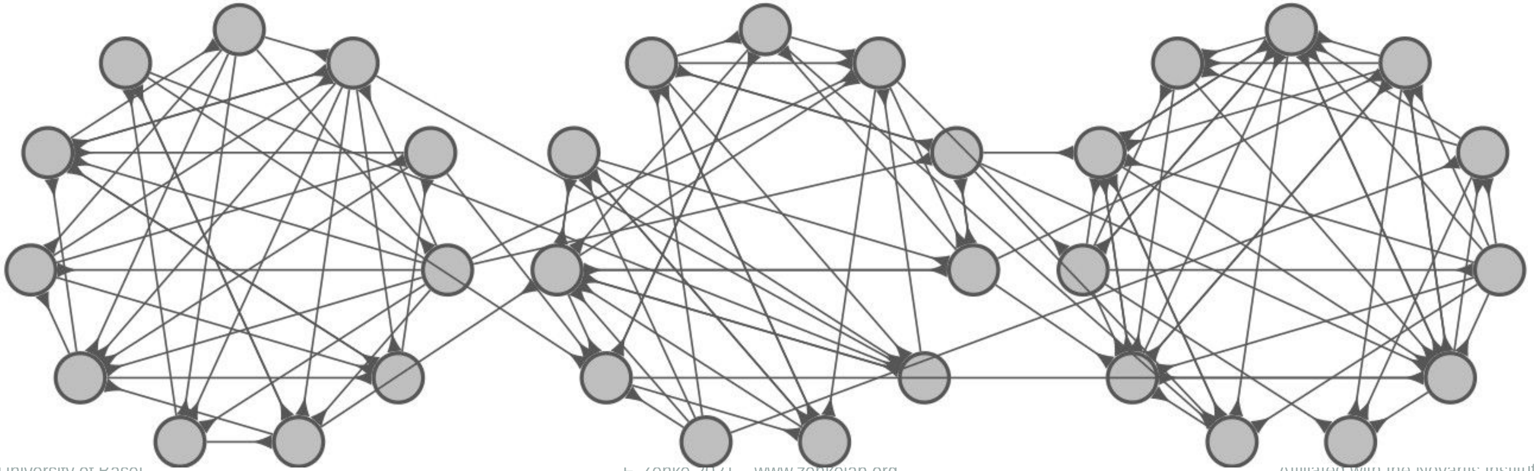
Toward task-optimized plausible networks

Friedrich Miescher Institute for Biomedical Research

1) Input — Sophisticated function → **2) Desired output**

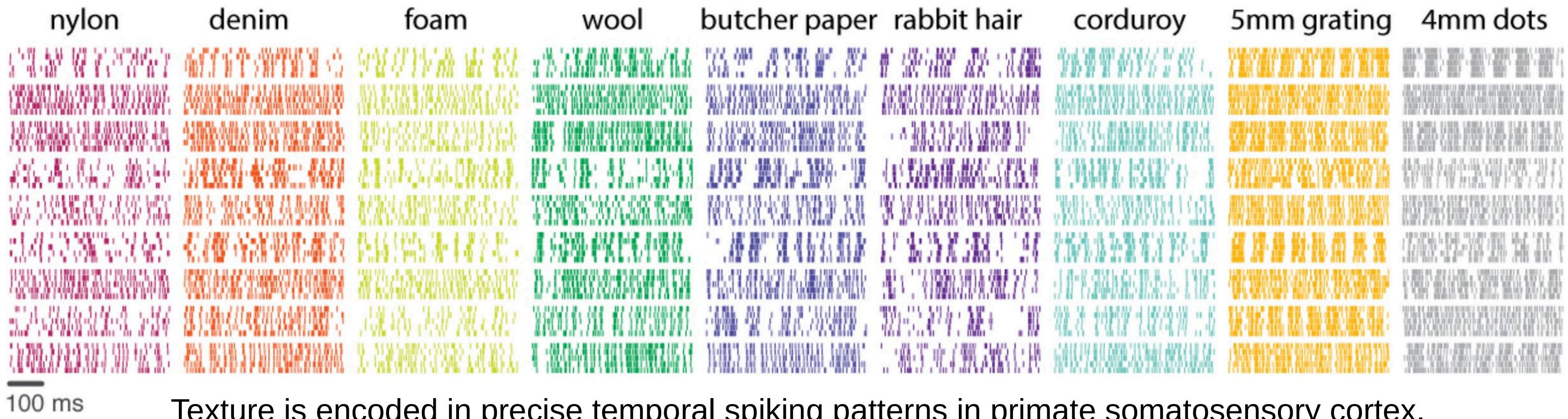
3) Find connectivity

- Gradient-based “plasticity rules”
- Plausible approximations

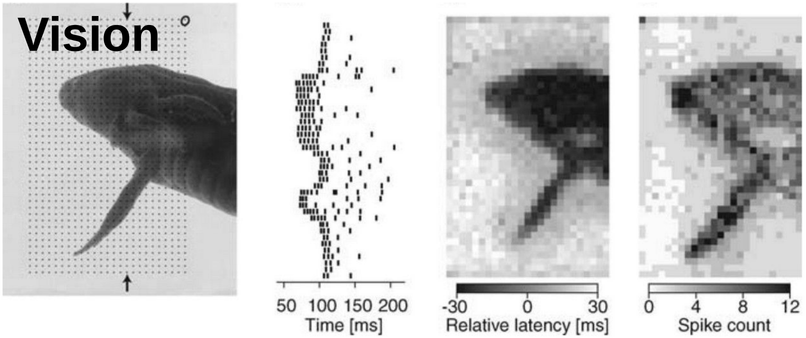


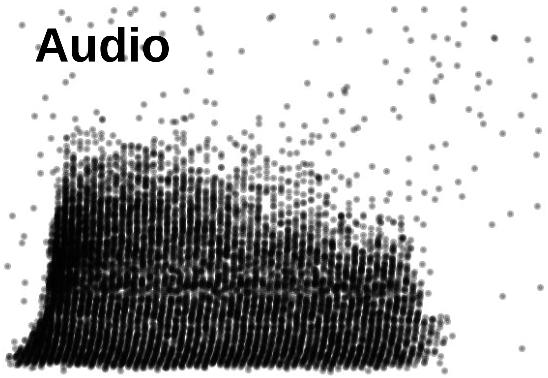
Input: Spatiotemporal spike patterns

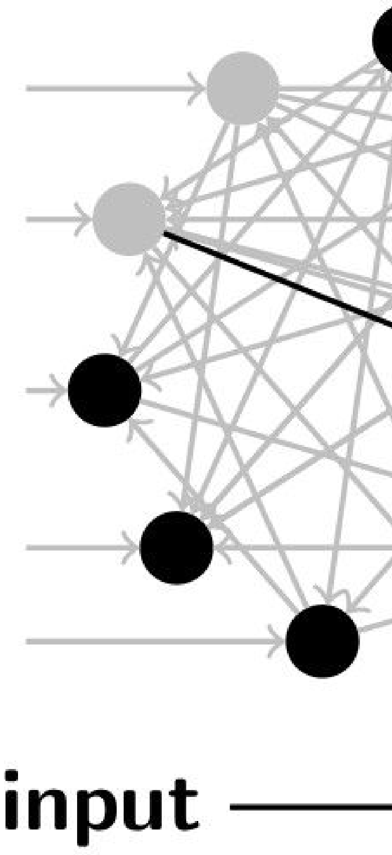
FMI
Friedrich Miescher Institute
for Biomedical Research



Touch Texture is encoded in precise temporal spiking patterns in primate somatosensory cortex.
Long, K.H., Lieber, J.D., and Bensmaia, S.J. (2021). BioRxiv 2021.04.11.439354.



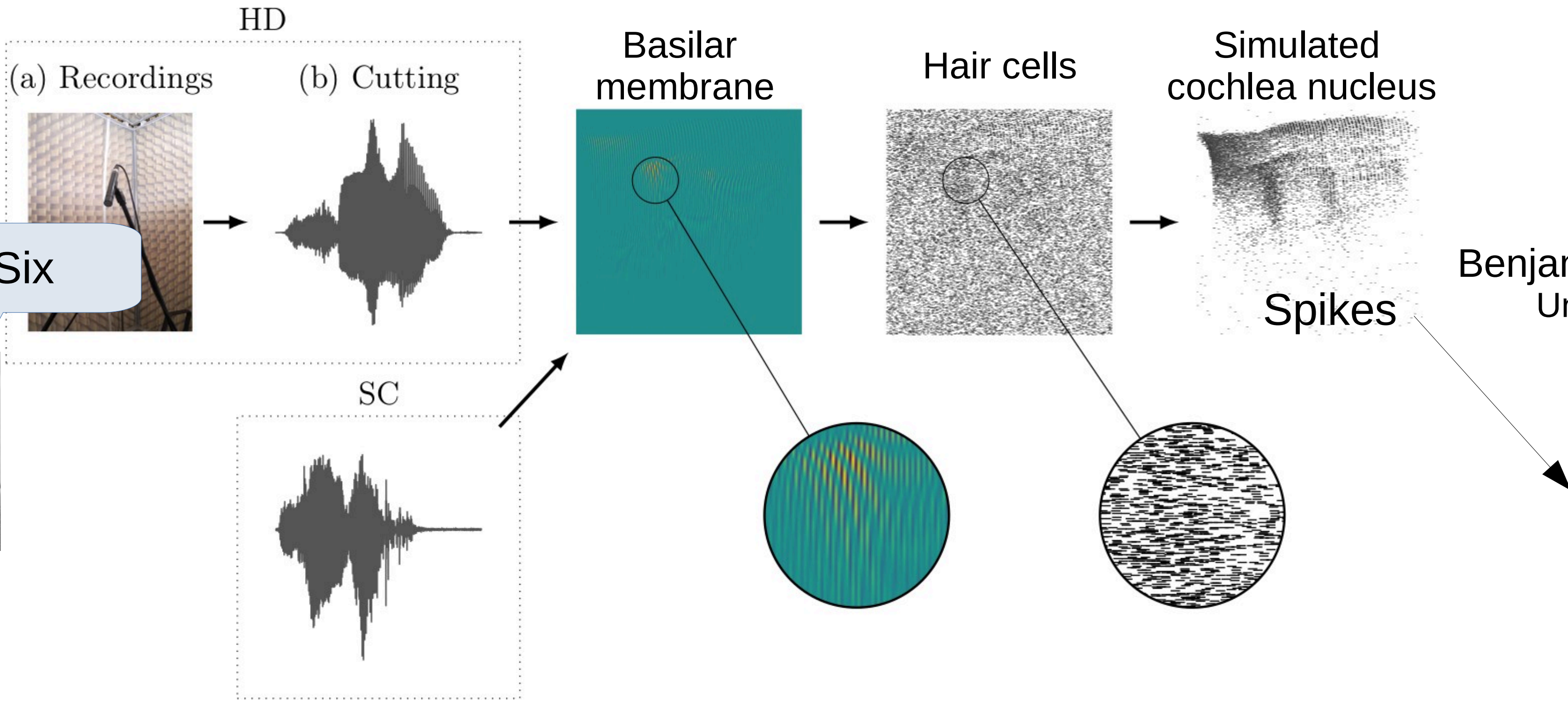




Gollisch & Meister (2008)

Cramer et al. (2020)

Towards realistic input abstractions

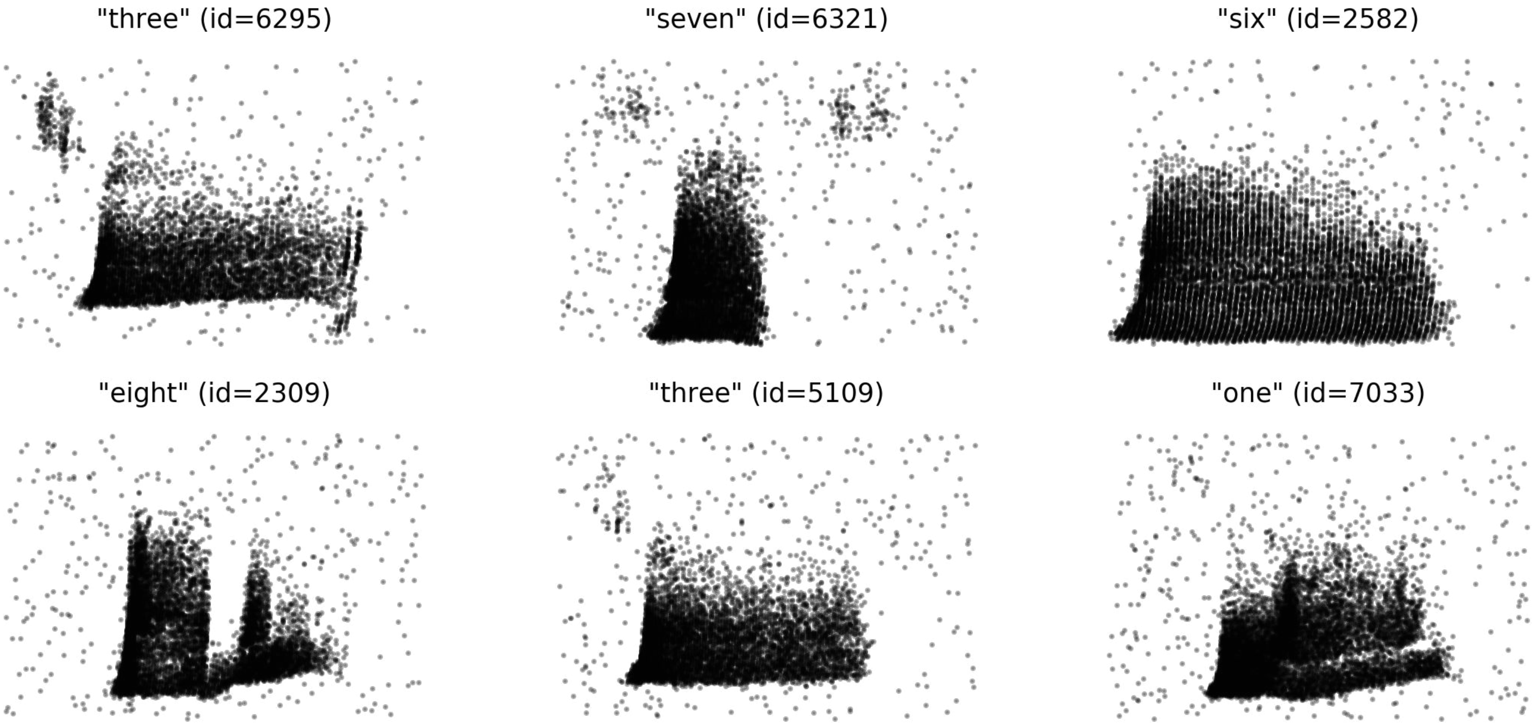


Benjamin Cramer
Uni Heidelberg

Dataset & code: www.compneuro.net
Cramer, B., Stradmann, Y., Schemmel, J., and Zenke, F. (2020)

Towards realistic input abstractions

Friedrich Miescher Institute
for Biomedical Research



Benjamin Cramer
Uni Heidelberg

Datasets

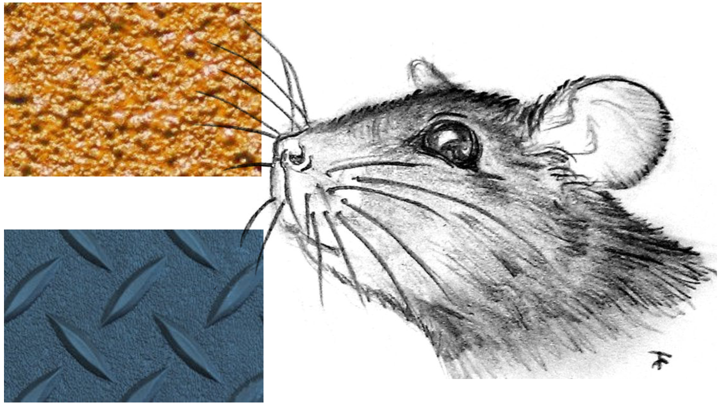
- Heidelberg Digits (~10k, 20 classes)
- Speech commands (~100k, 35 classes)

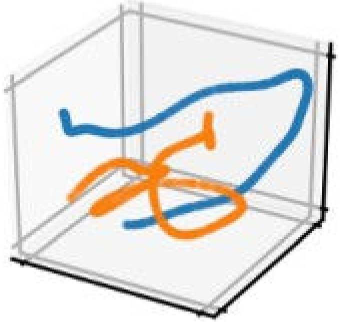
Dataset & code: www.compneuro.net
Cramer, B., Stradmann, Y., Schemmel, J., and Zenke, F. (2020)

Input: Spatiotemporal spike trains

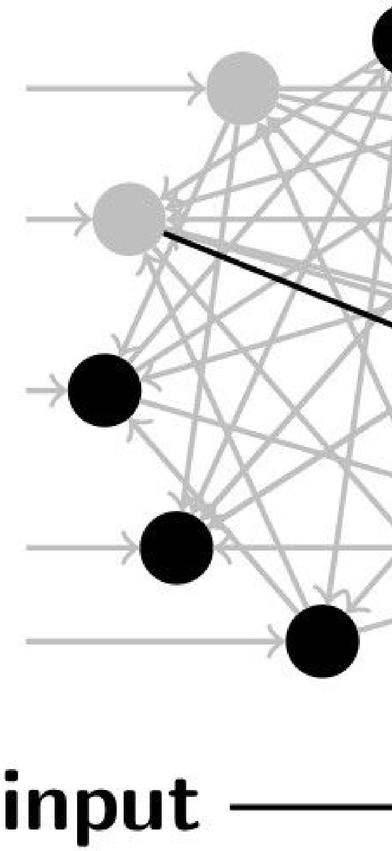
Example for this talk

Binary discrimination task

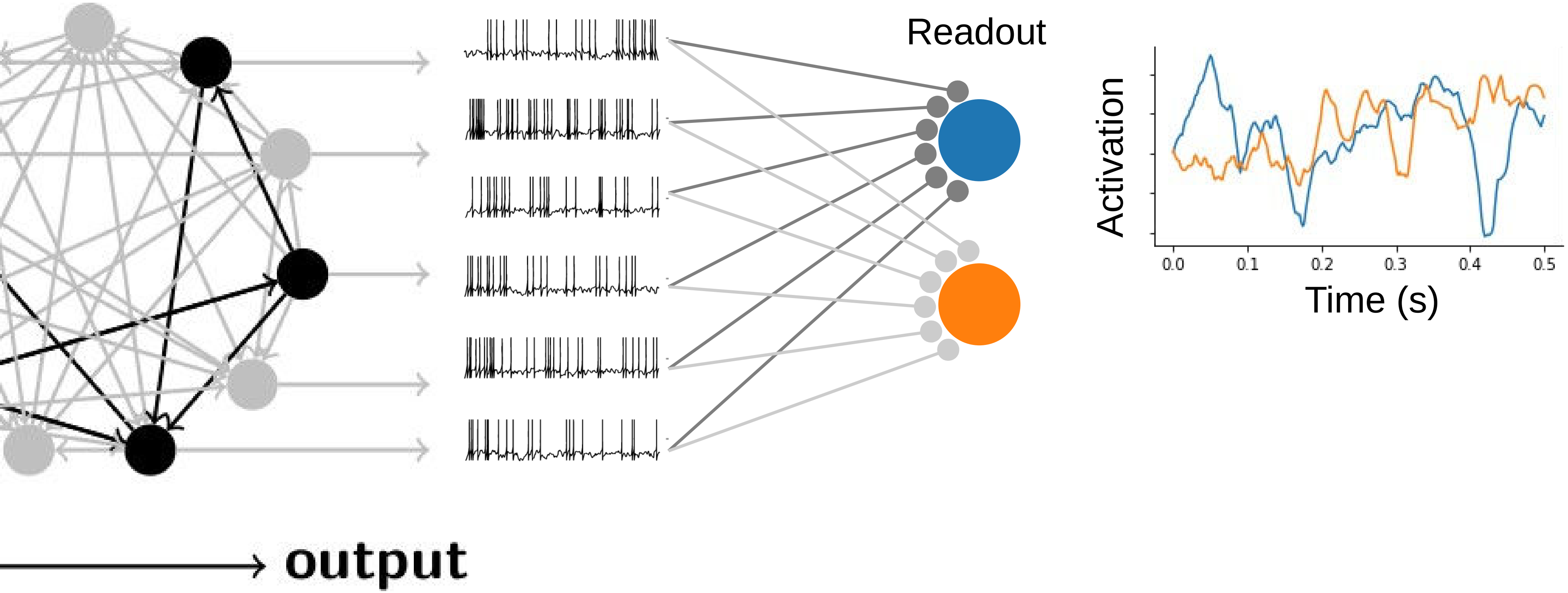




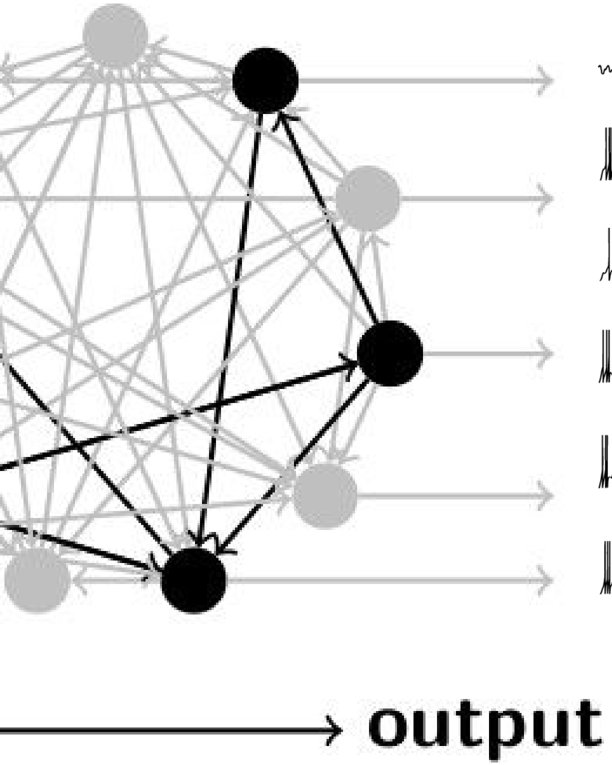
Synthetic inputs from two smooth random manifolds.

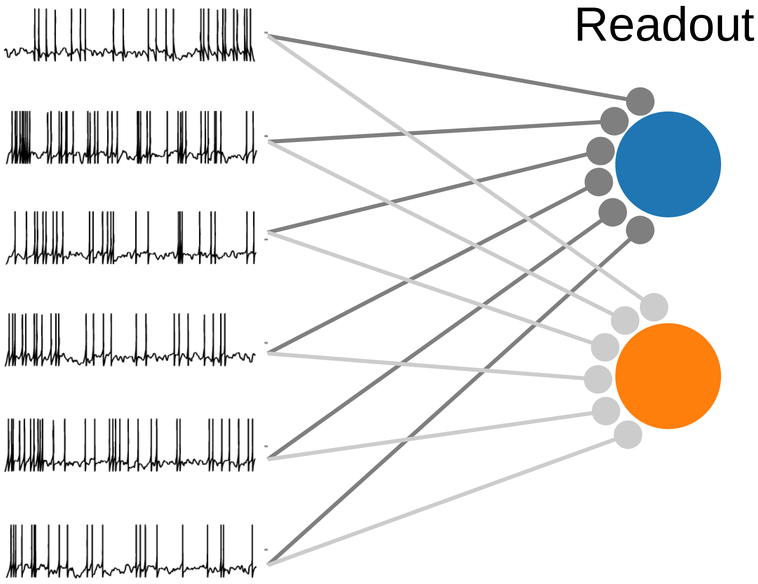


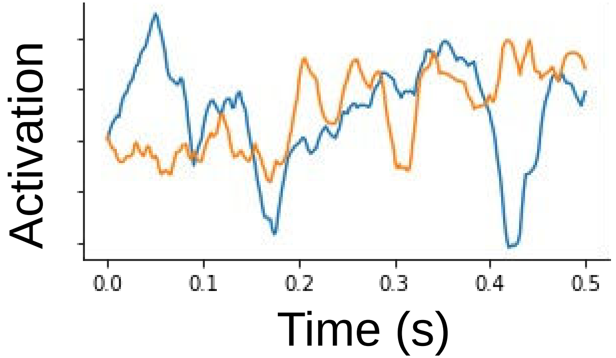
Output: Linear combination of filtered output spike trains



Output: Linear combination of filtered output spike trains

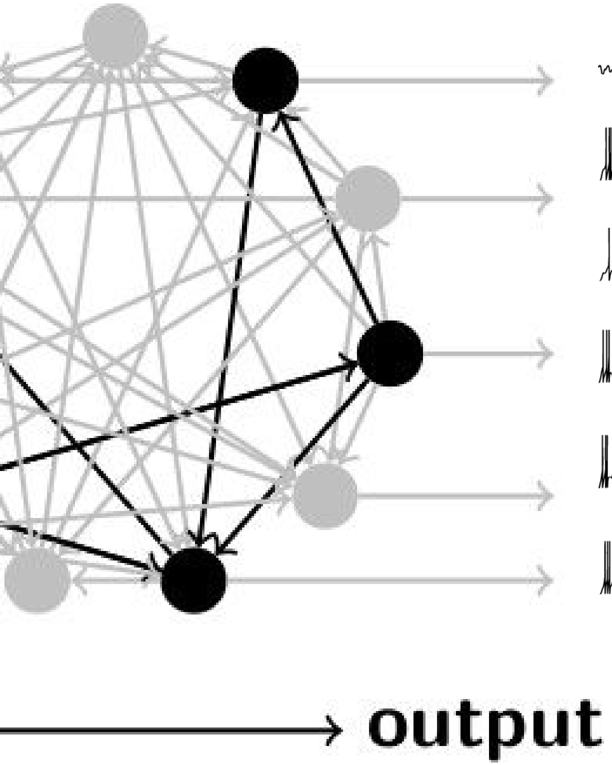


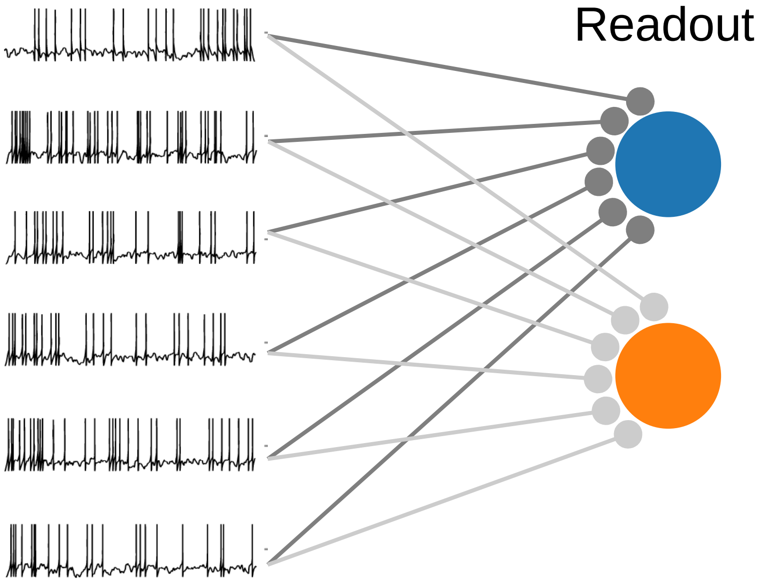


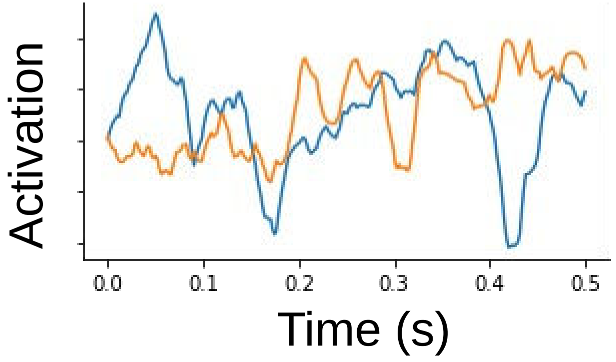


$$p_i = \underset{i}{\mathrm{softmax}} \left(\max_t U_i(t) \right)$$

Output: Linear combination of filtered output spike trains





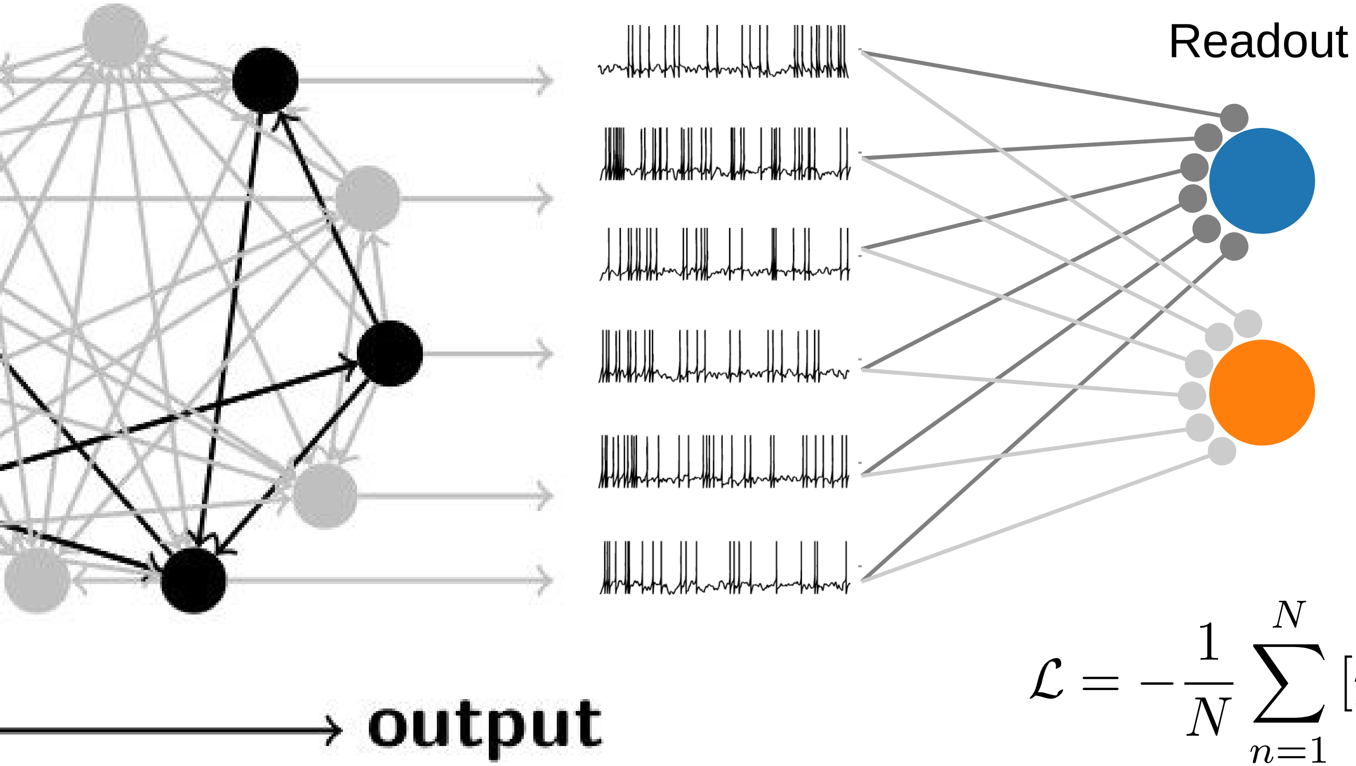


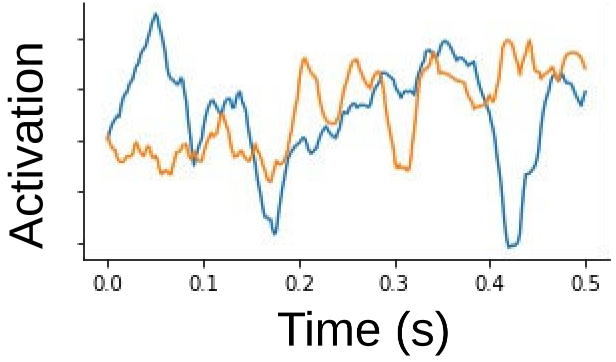
$$p_i = \operatorname*{softmax}_i \left(\boxed{\max_t U_i(t)} \right)$$

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

Max over time idea from Tempotron
Gütig & Sompolinsky (2006); Gütig (2016)

Output: Linear combination of filtered output spike trains





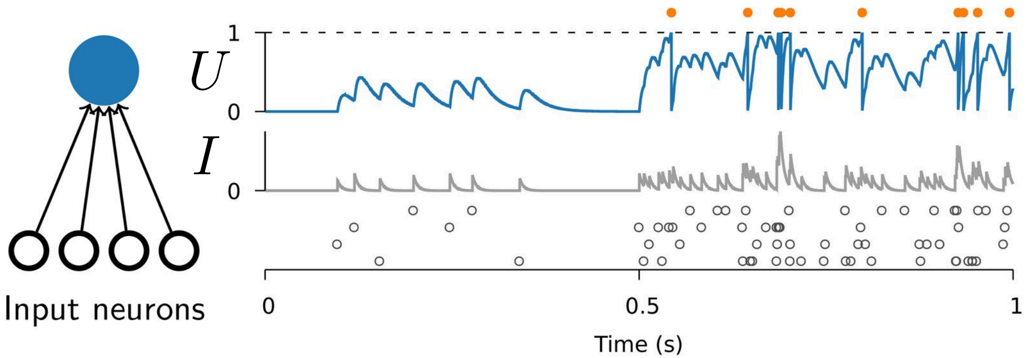
$$p_i = \underset{i}{\mathrm{softmax}}\left(\boxed{\max_t U_i(t)}\right)$$

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

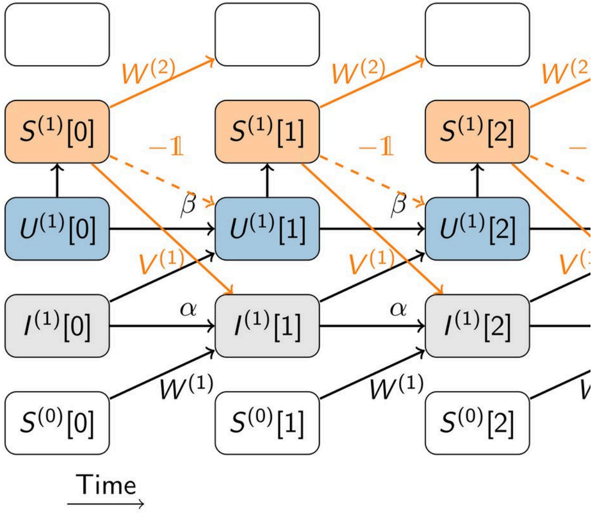
Max over time idea from Tempotron
Gütig & Sompolinsky (2006); Gütig (2016)

Training spiking networks end-to-end

Friedrich Miescher Institute for Biomedical Research



- Spiking neurons & networks are RNNs
- Known training procedures for networks **with hidden units**
  - Backpropagation-through time (BPTT)
  - Real-time recurrent learning (RTRL)



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

Problem

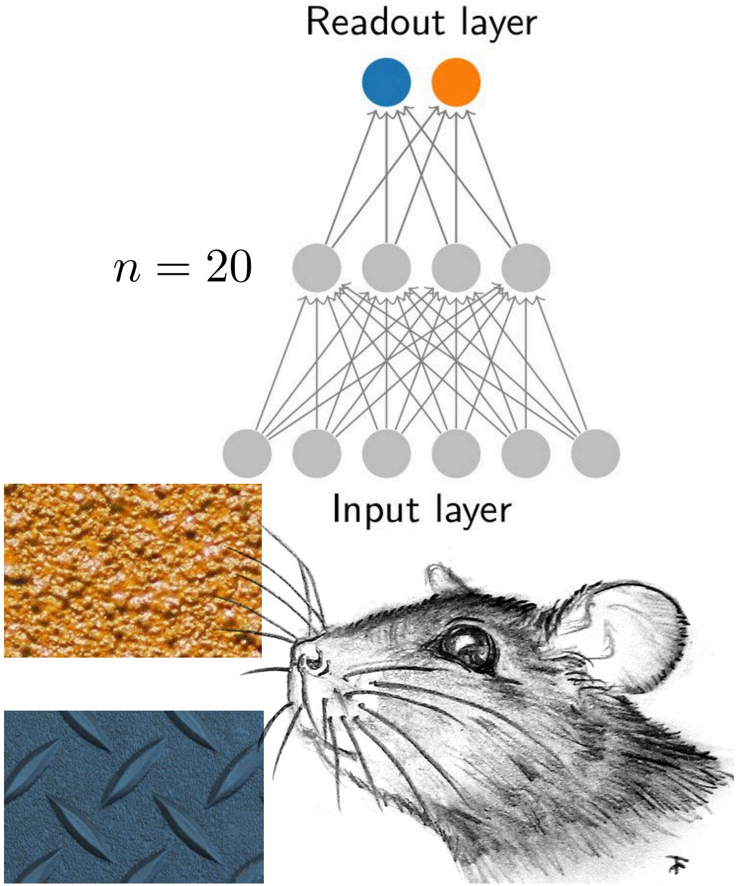
$$U_i^{(1)}[n+1] = \beta U_i^{(1)}[n] + I_i^{(1)}[n] - S_i[n]$$

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

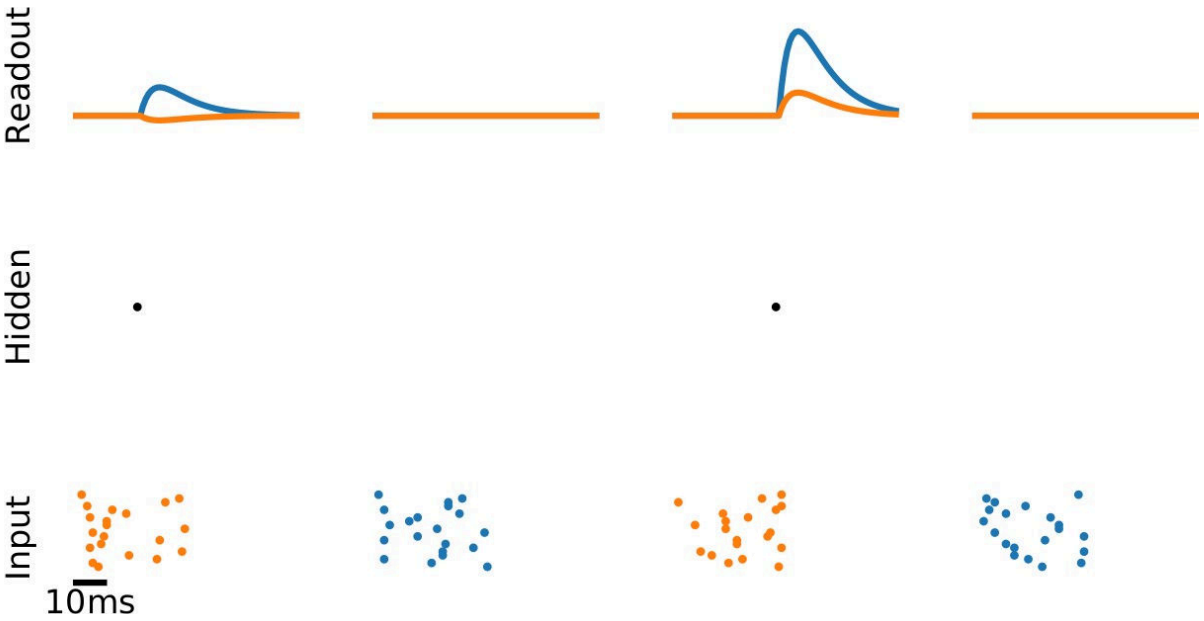
Forward Euler integration

Neftci, Mostafa, & Zenke (2019)

Example: A two-fold classification problem



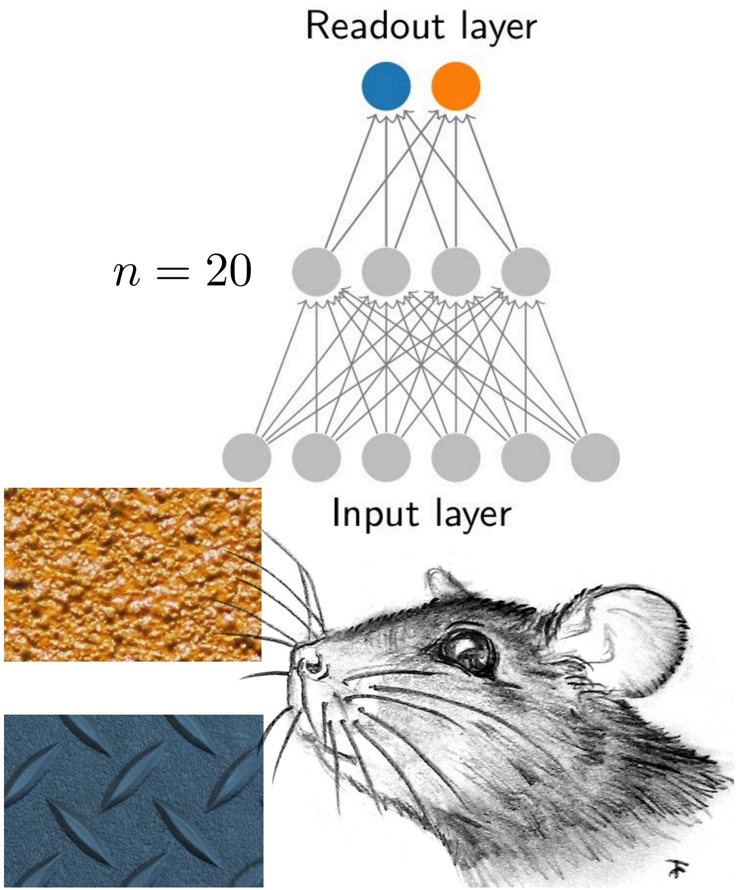
Activity snapshots (network has not learned anything)



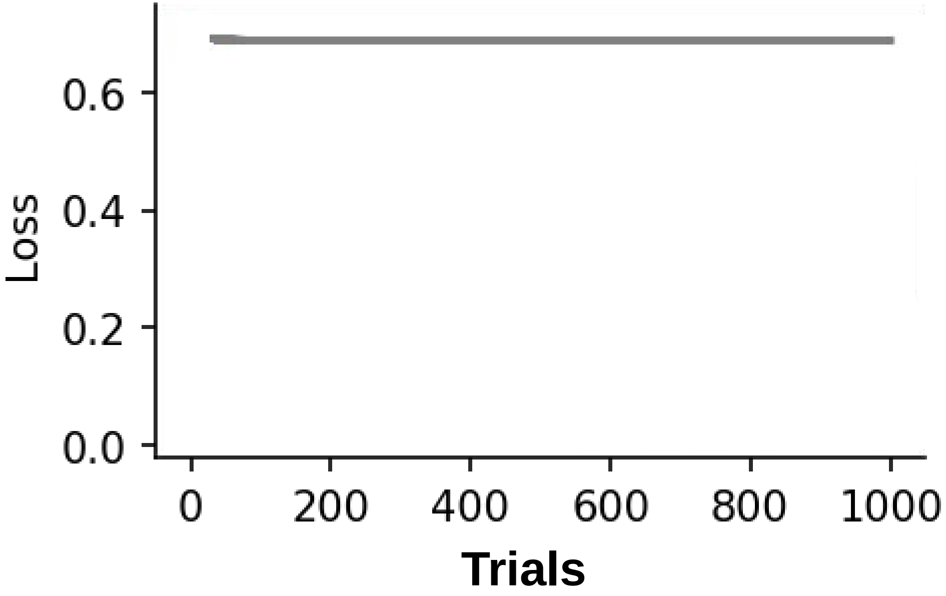
Synthetic data set: 2000 samples from two smooth random manifolds

Example: A two-fold classification problem

$n = 20$

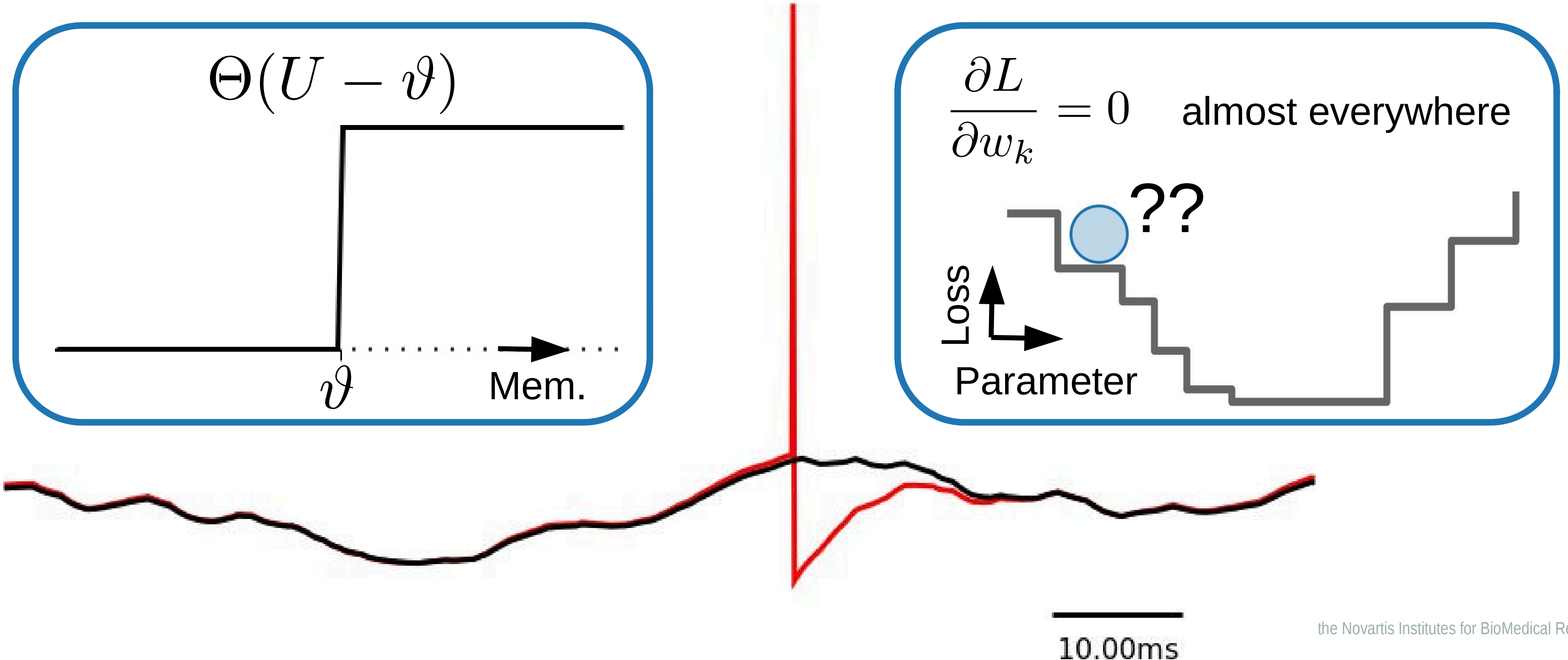


Evolution of loss during gradient descent



Problem: The derivative of a spike is zero almost everywhere

Friedrich Miescher Institute
for Biomedical Research



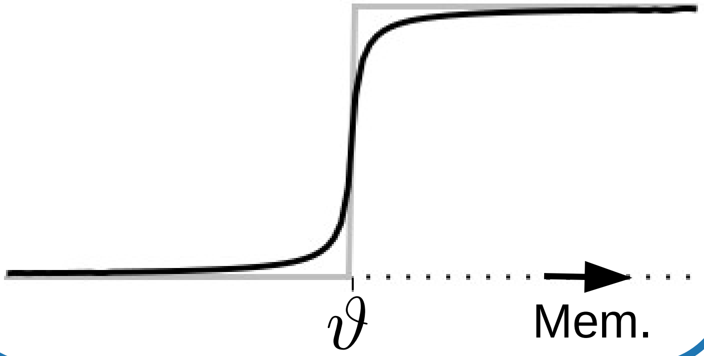
Option 1 ("classic"): Noise injection → gradient in expectation values.
e.g.: Pfister, Toyoizumi, Barber & Gerstner (2006), Gardner, Sporea & Grüning (2015)

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

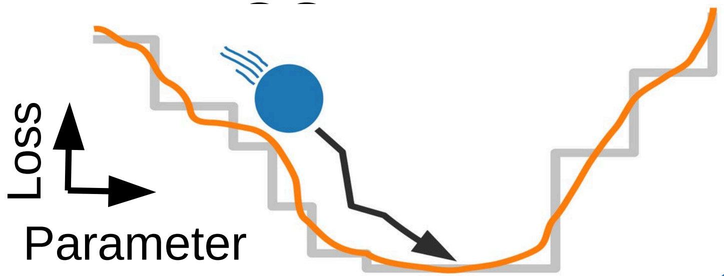
Option 3: "Know hidden layer targets"
Gilra & Gerstner (2017), Nicola & Clopath (2017)

Option 4: Use surrogate gradients. Bohte (2011), Bellec, Salaj, Subramoney, Legenstein, and Maass (2018) Shrestha & Orchard (2018), Zenke & Ganguli (2018), ...
In ML: "Straight-through estimators" Bengio et al. (2013)

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

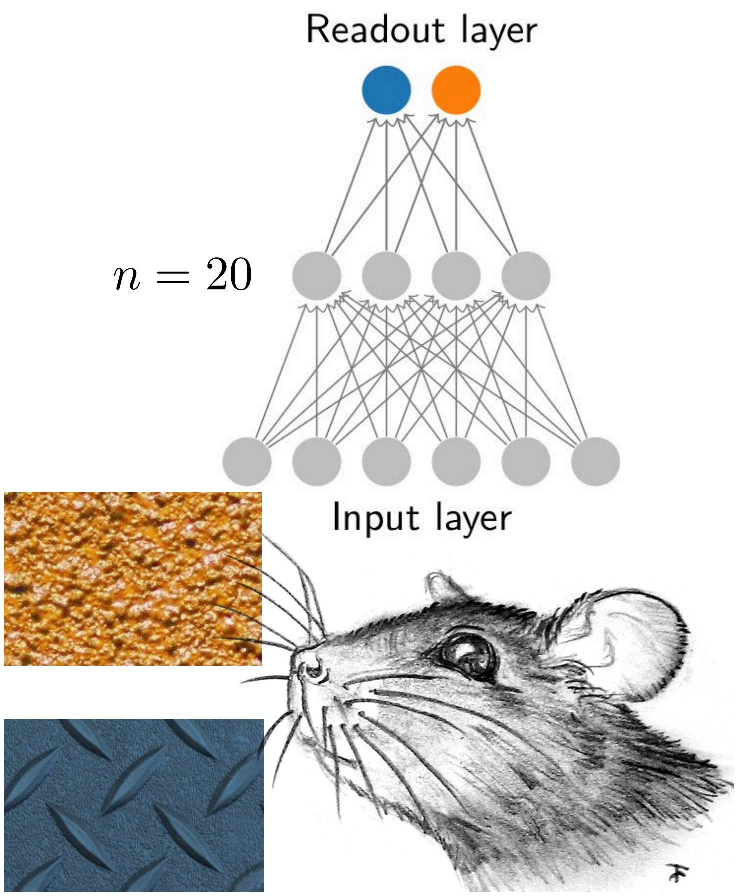


$$\frac{\partial L}{\partial w_k} \neq 0$$

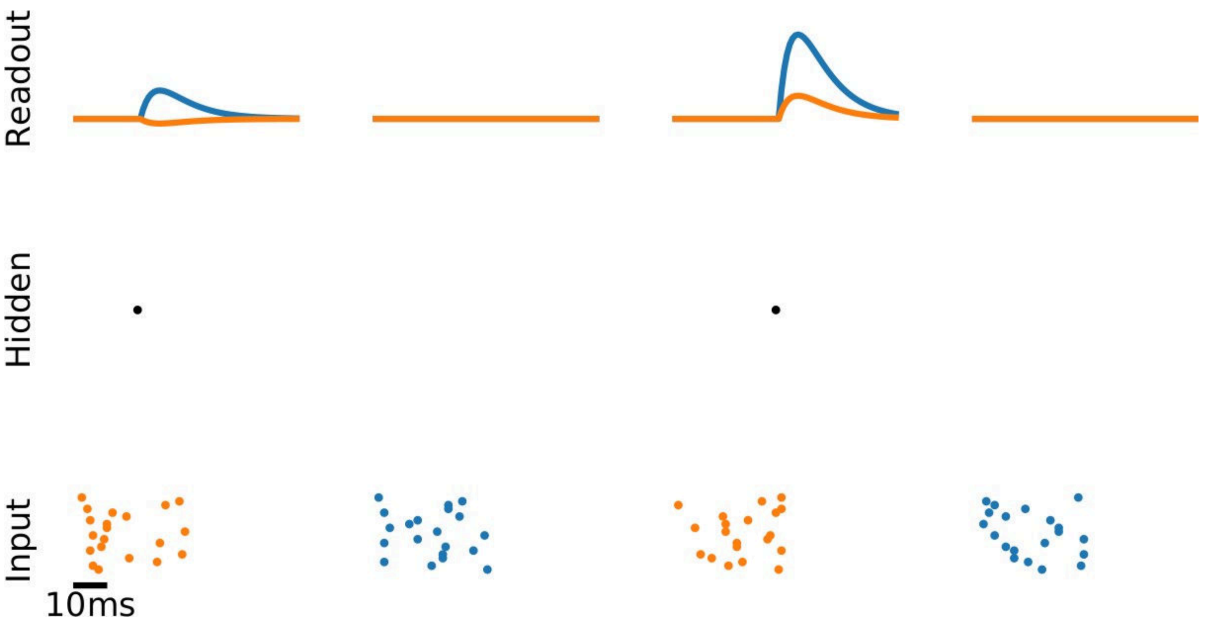


Example: A two-fold classification problem

Friedrich Miescher Institute
for Biomedical Research

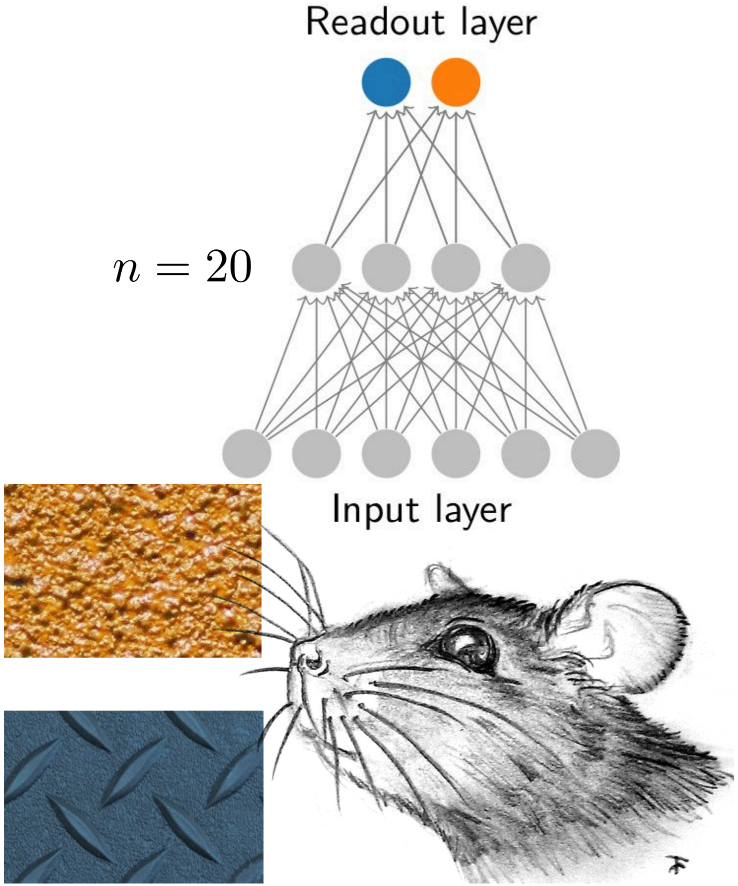


Activity snapshots (network has not learned anything)

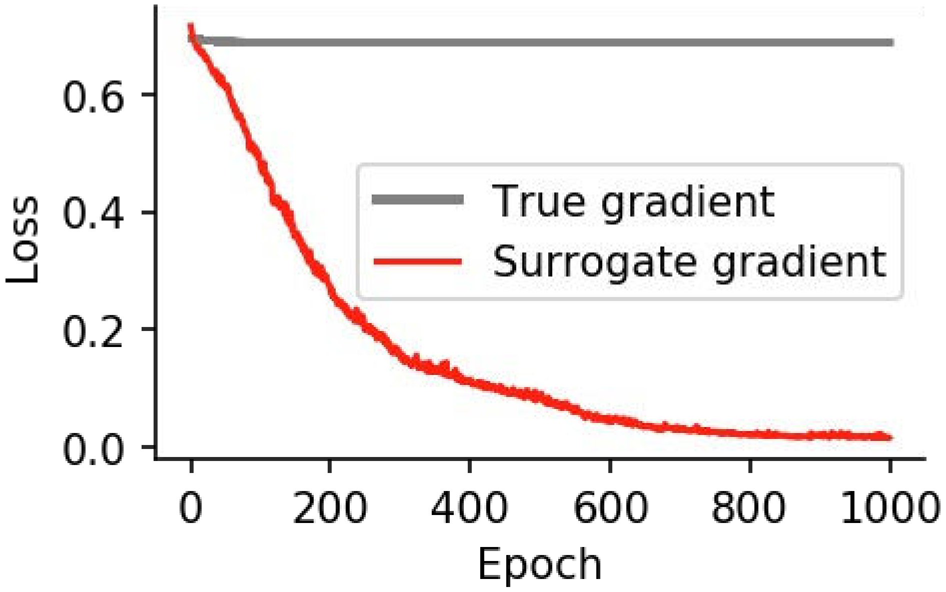


Synthetic data set: 2000 samples from two smooth random manifolds

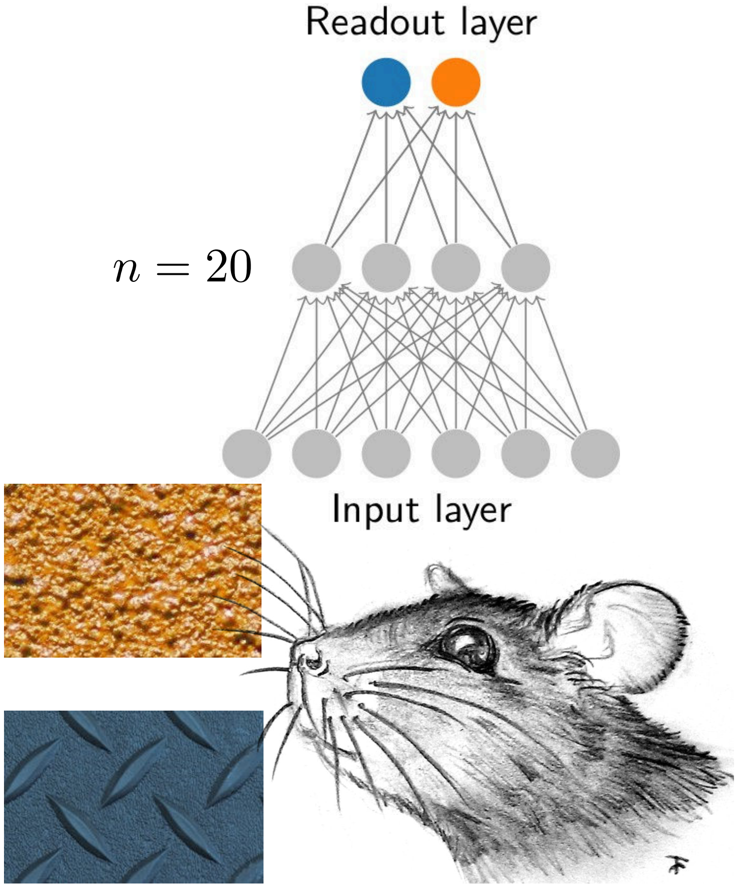
Example: A two-fold classification problem



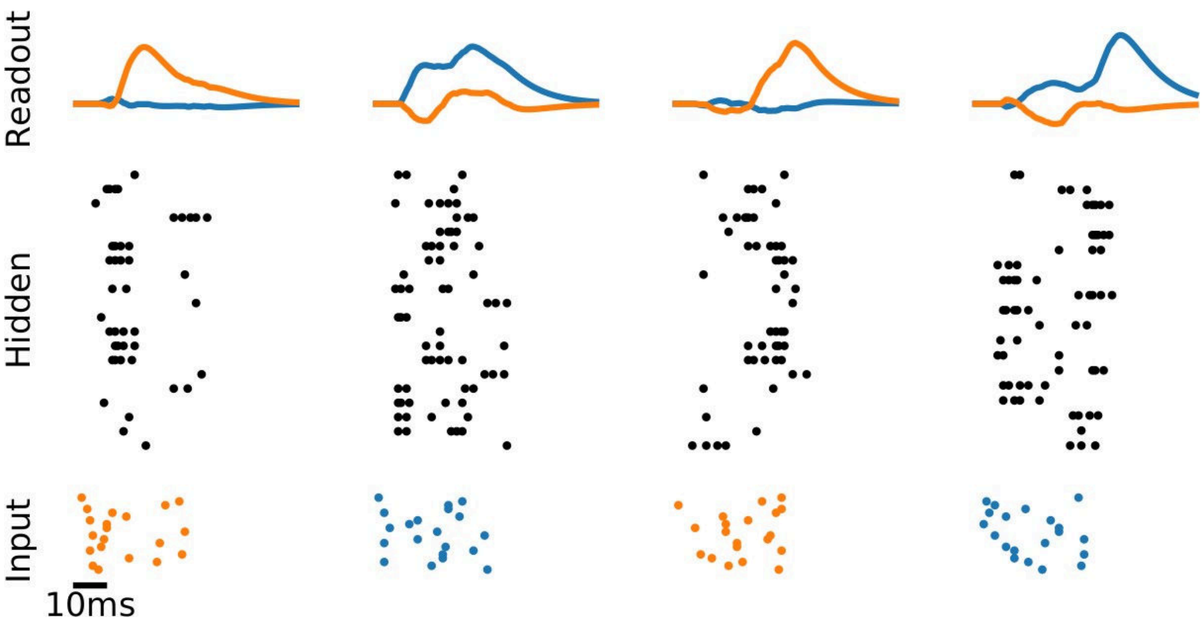
Evolution of loss during surrogate gradient descent



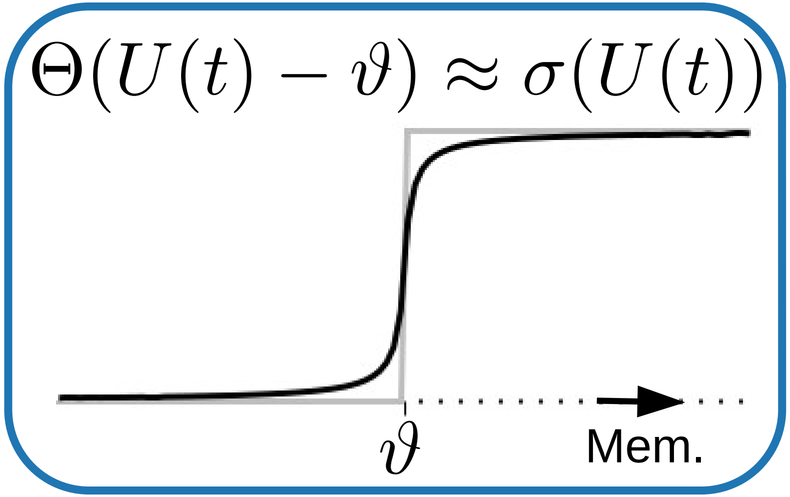
Example: A two-fold classification problem

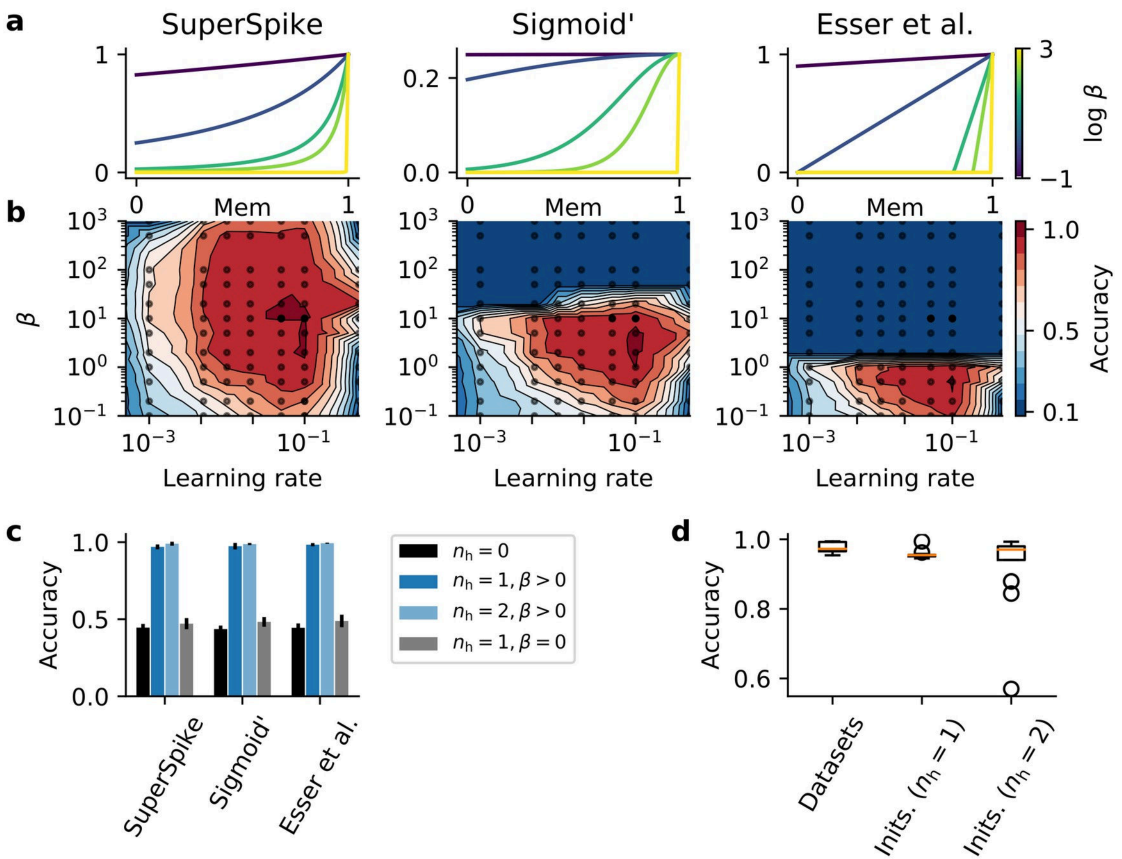


Activity snapshots (trained network)



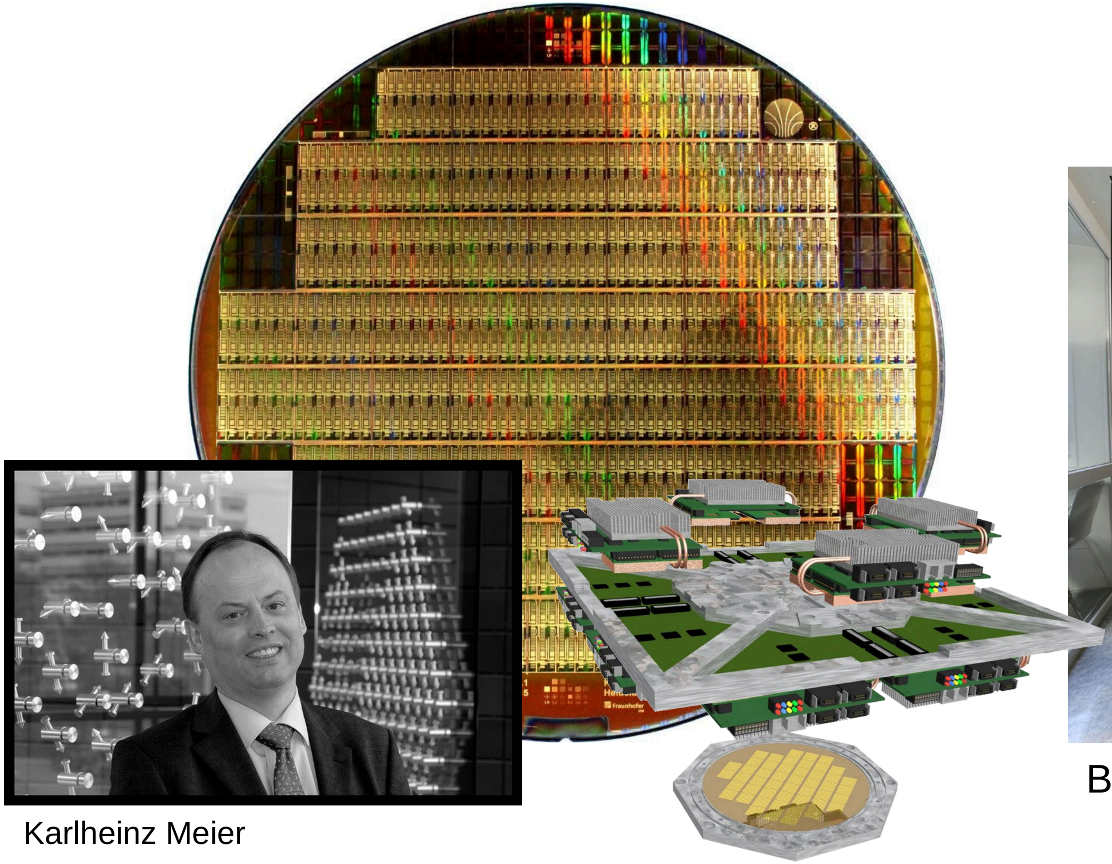
Surrogate gradient learning is remarkably robust to different choices of surrogates





Zenke and Vogels (2021)

Can surrogate gradients breathe life into neuromorphic systems?

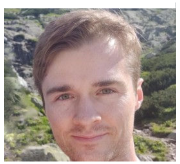


Karlheinz Meier

BrainScales Neuromorphic System

Johannes Schemmel
Uni Heidelberg

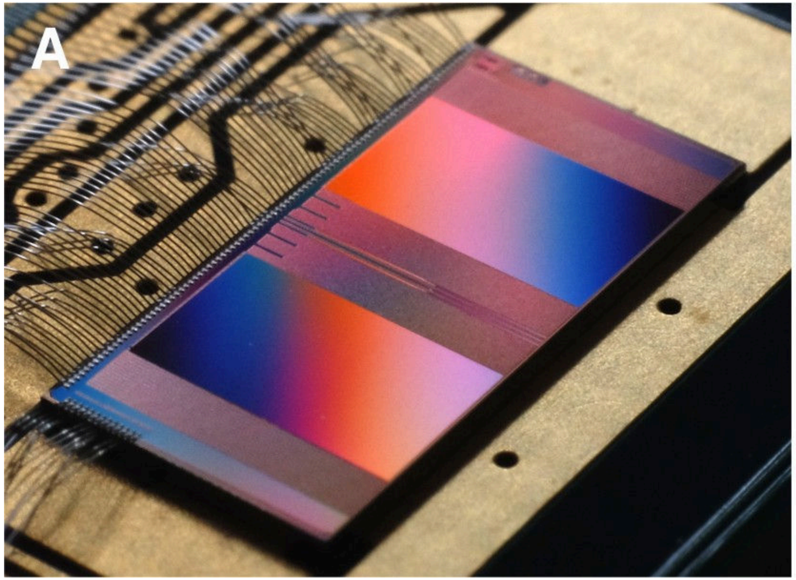
Benjamin Cramer

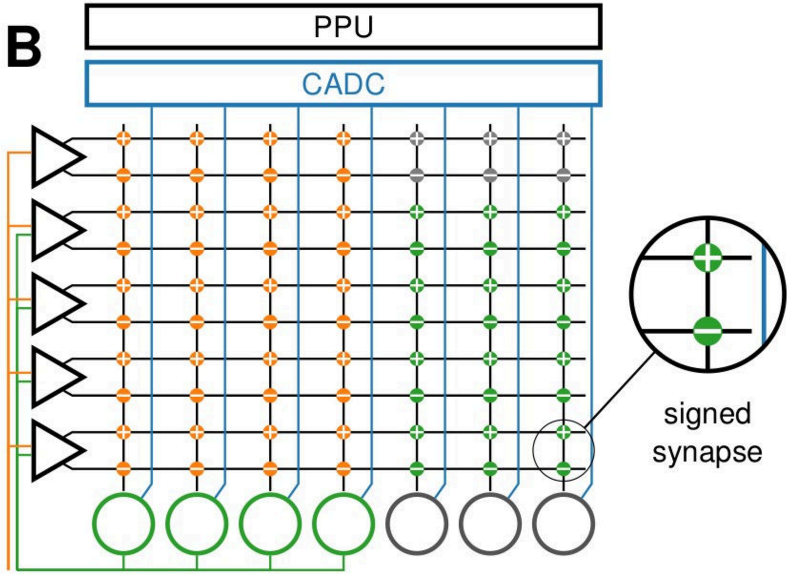


Sebastian Billaudelle

Functional spiking neural networks trained on accelerated **analog neuromorphic hardware**

Friedrich Miescher Institute
for Biomedical Research





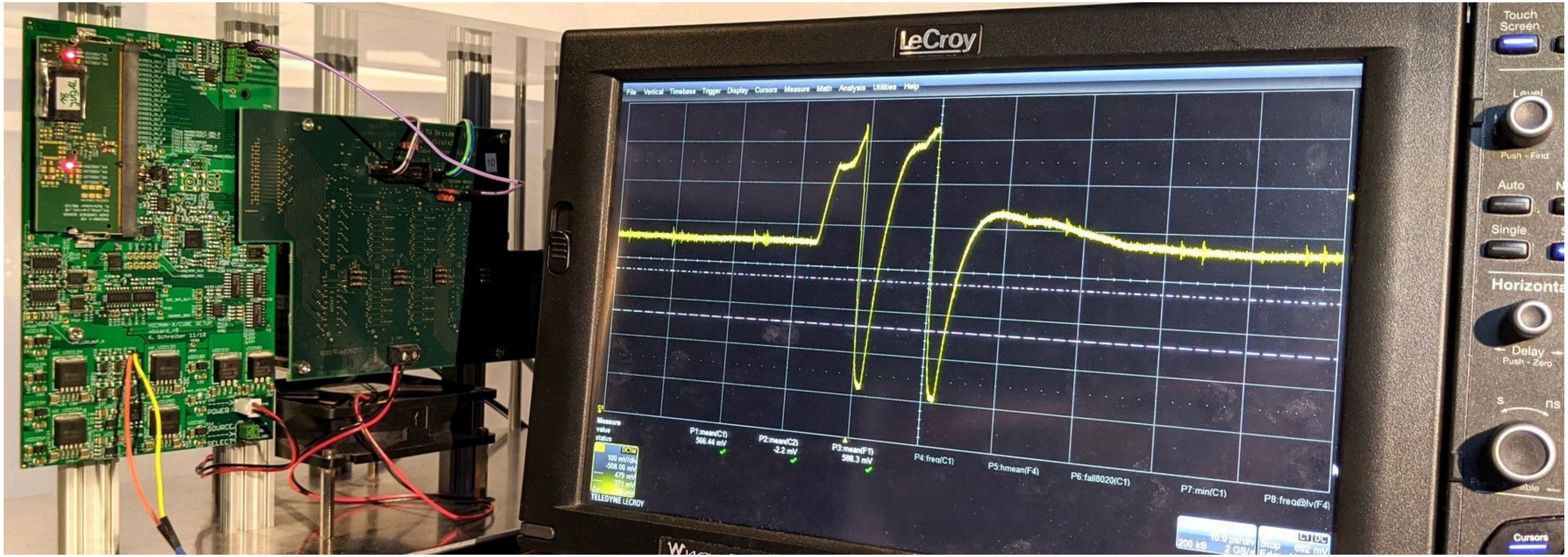
Benjamin
Cramer

Sebastian
Billaudelle

Cramer, B., Billaudelle, S., Kanya, S., Leibfried, A., Grübl, A., Karasenko, V., Pehle, C., Schreiber, K., Stradmann, Y., Weis, J., Schemmel, J., & Zenke, F. (submitted).

Functional spiking neural networks trained on accelerated **analog neuromorphic hardware**

FMI
Friedrich Miescher Institute
for Biomedical Research



Benjamin Cramer

Sebastian Billaudelle

Cramer, B., Billaudelle, S., Kanya, S., Leibfried, A., Grübl, A., Karasenko, V., Pehle, C., Schreiber, K., Stradmann, Y., Weis, J., Schemmel, J., & Zenke, F. (submitted). https://arxiv.org/abs/2006.07239

Surrogate gradients require voltage traces

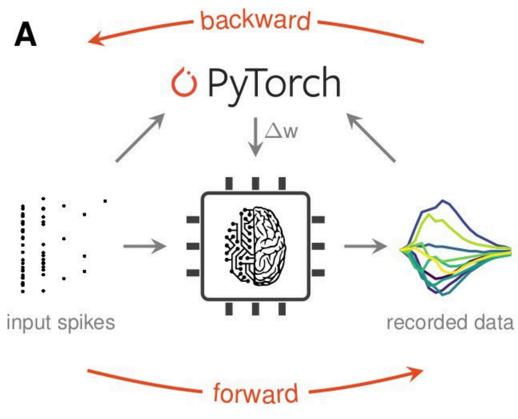
→ In-the-loop training

Friedrich Miescher Institute for Biomedical Research

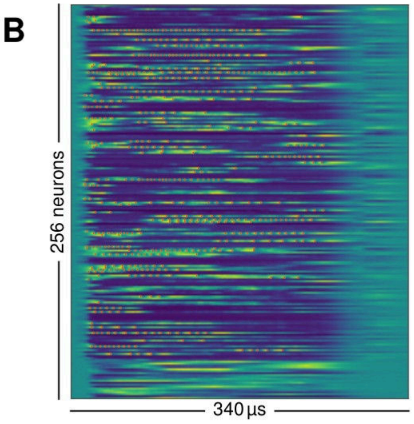
1) Measure on-chip analog voltage traces

2) Inject voltage into computational graph

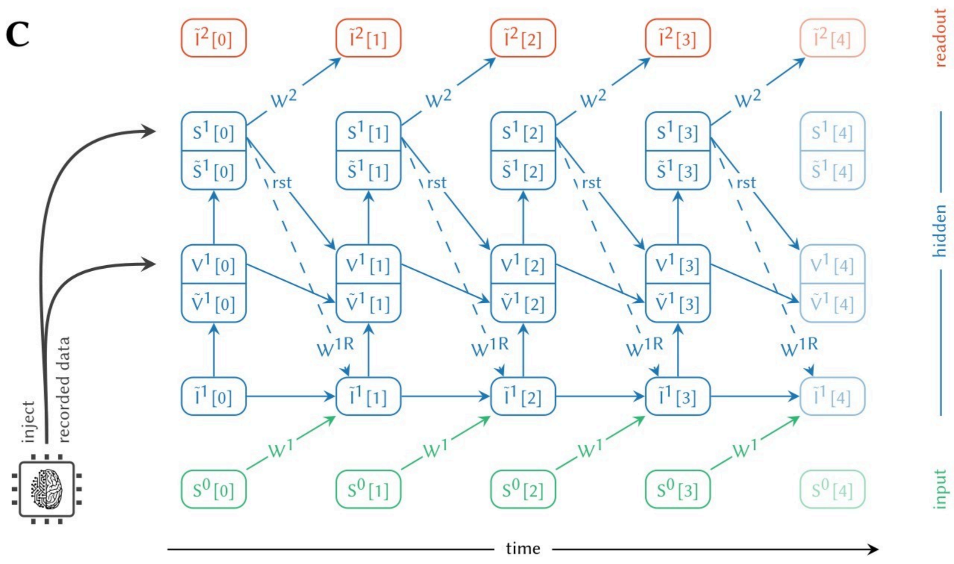
A



B



C

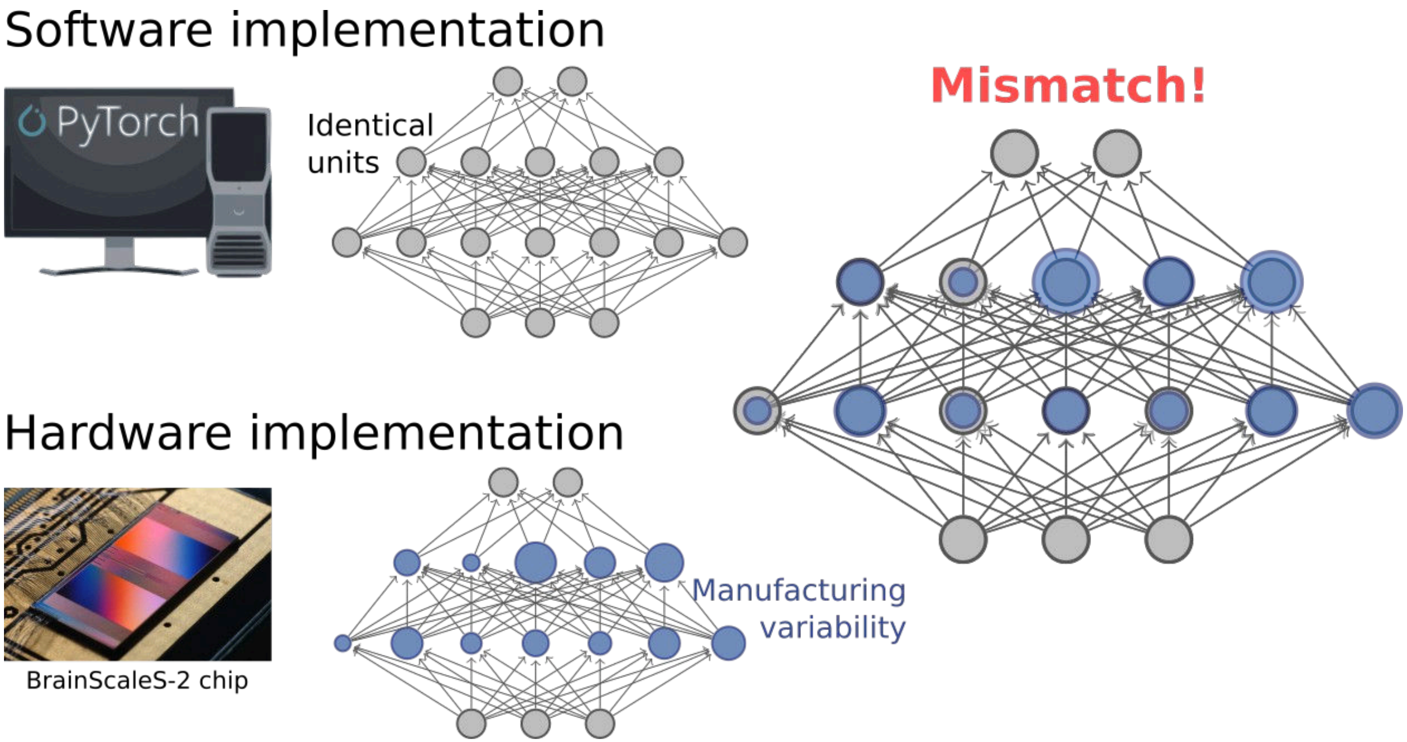


3) Compute surrogate gradients → update weights

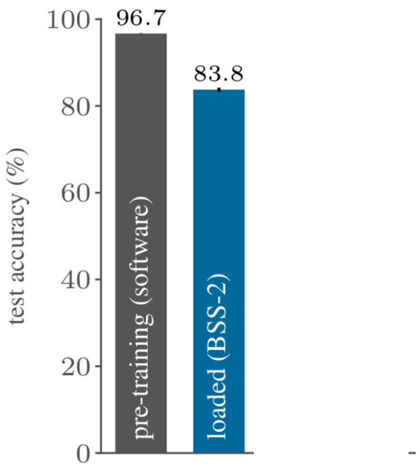
Cramer, B., Billaudelle, S., Kanya, S., Leibfried, A., Grübl, A., Karasenko, V., Pehle, C., Schreiber, K., Stradmann, Y., Weis, J., Schemmel, J., & Zenke, F. (submitted). https://arxiv.org/abs/2006.07239

Surrogate gradient learning **automatically compensates for device mismatch**

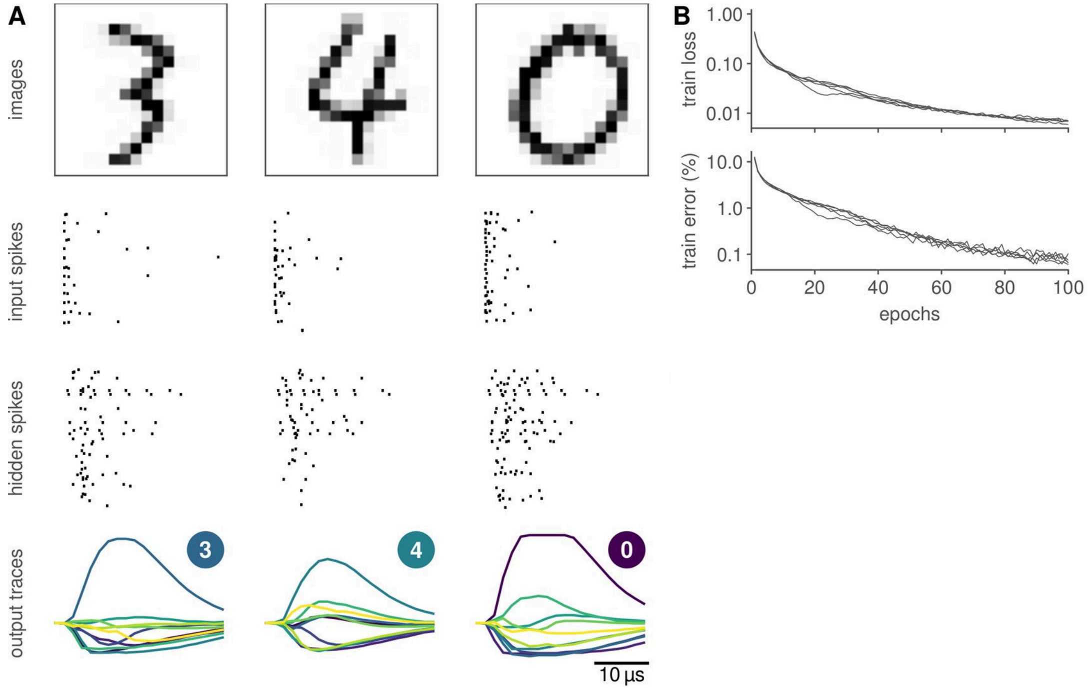
Friedrich Miescher Institute
for Biomedical Research



16x16 MNIST



Functional spiking neural networks trained on accelerated **analog neuromorphic hardware**

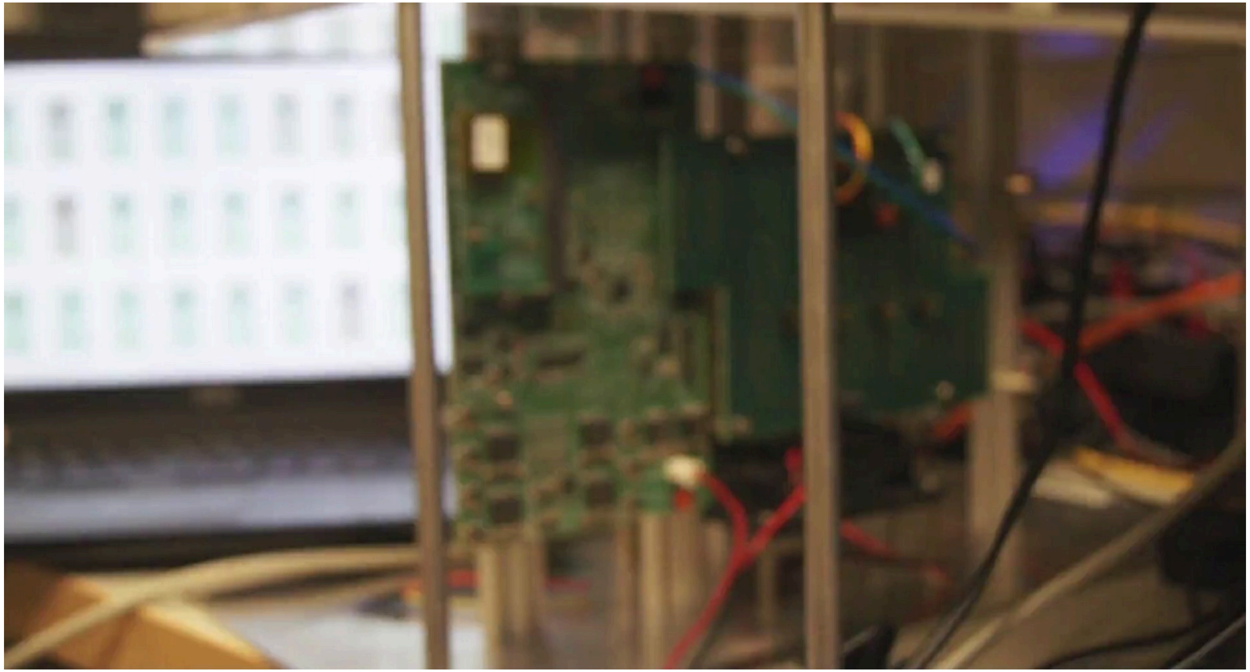


80k
nages/sec
200mW

Benjamin
Cramer

Sebastian
Billaudelle

Cramer, B., Billaudelle, S., Kanya, S., Leibfried, A., Grübl, A., Karasenko, V., Pehle, C., Schreiber, K., Stradmann, Y., Weis, J., Schemmel, J., & Zenke, F. (submitted).

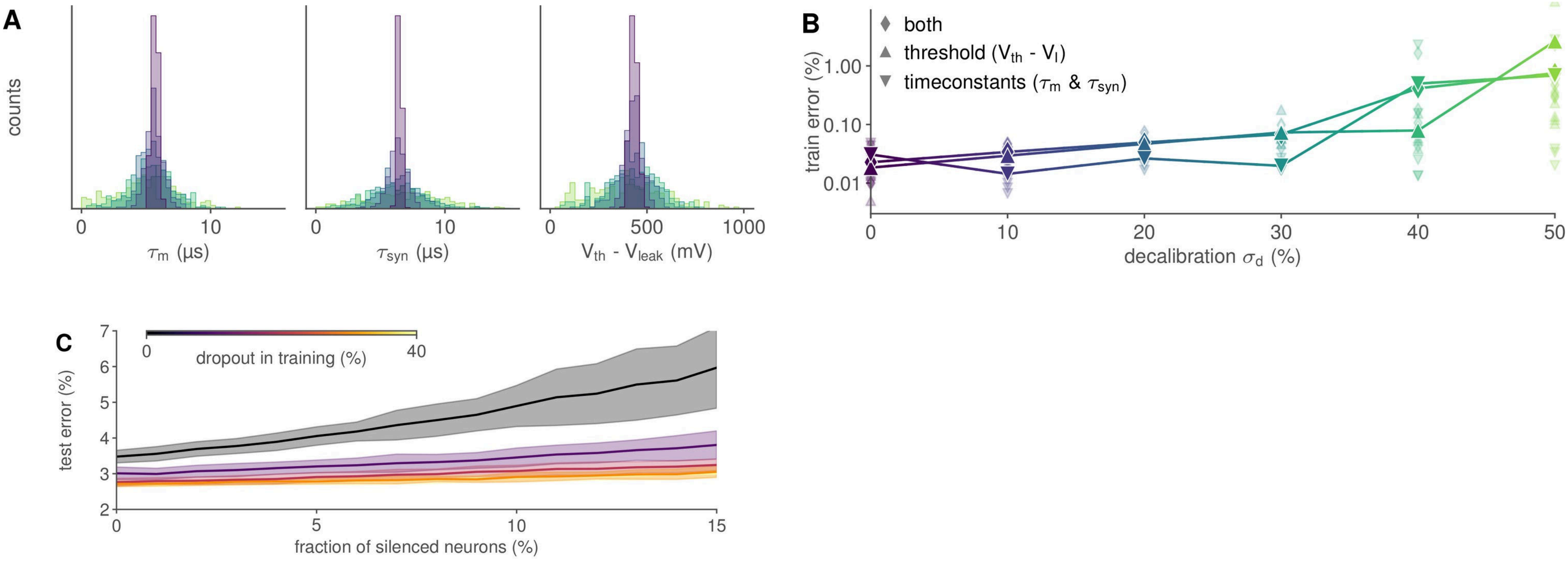


Benjamin Cramer

Sebastian Billaudelle

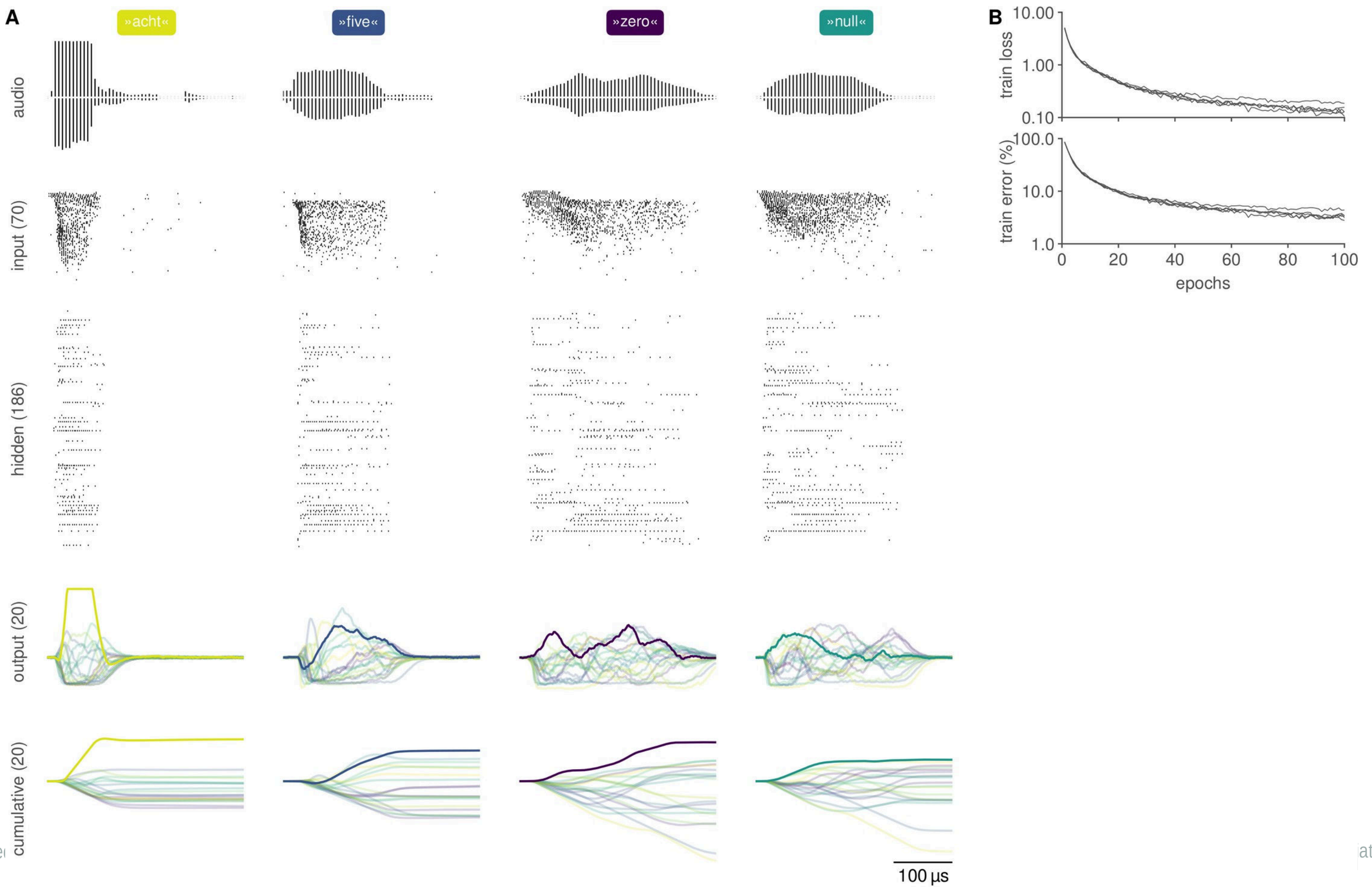
Surrogate gradient learning is self-calibrating on **analog neuromorphic hardware**

Friedrich Miescher Institute for Biomedical Research



Cramer, B., Billaudelle, S., Kanya, S., Leibfried, A., Grübl, A., Karasenko, V., Pehle, C., Schreiber, K., Stradmann, Y., Weis, J., Schemmel, J., & Zenke, F. (submitted).

Heidelberg spiking digits: Speech recognition



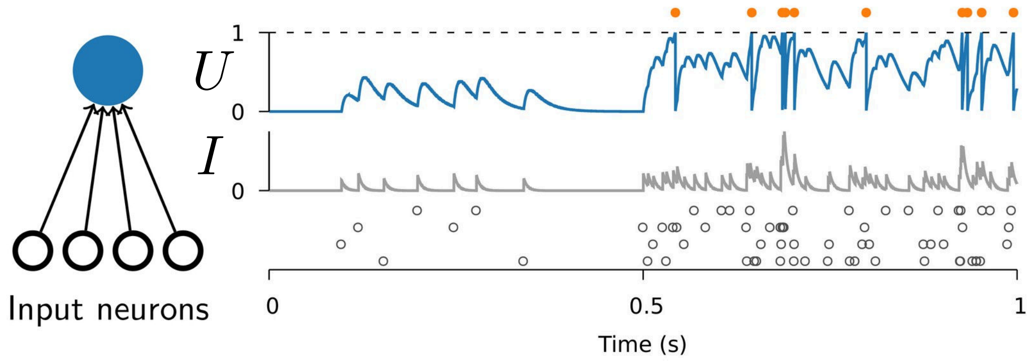
Benjamin Cramer

Sebastian Billaudelle

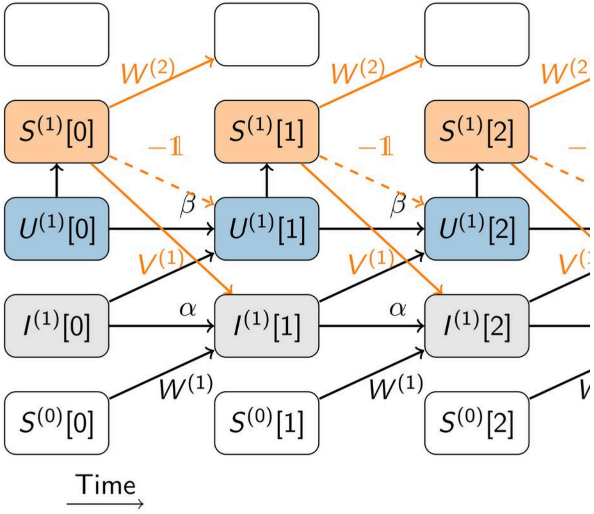
Interim summary

- Surrogate gradients
  - Effectively bringing end-to-end optimization to spiking networks
  - Remarkably robust
  - Can self-calibrate and breathe life into neuromorphic hardware
- What about biologically plausible learning?
  - Biological solutions to the spatial credit assignment problem
    Lillicrap et al. (2014, 2016), Guergiuev et al. (2016), Baldi et al. (2016), Samadi et al. (2017), Payeur et al. (2021)
  - Biological solutions to temporal credit assignment (**Today**)
    (i.e., not back-propagation through time)

Training spiking networks end-to-end



- Spiking neurons & networks are RNNs
- Known training procedures for networks with hidden units
  - Backpropagation-through time (BPTT)
  - **Real-time recurrent learning (RTRL)**



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

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

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

Explicit recurrence

Implicit recurrence

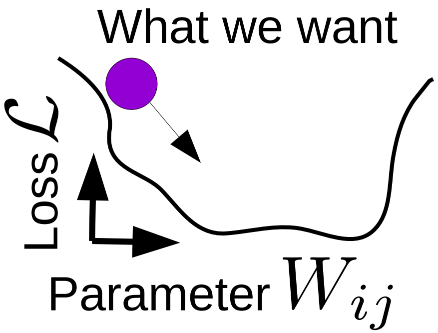
Neftci, Mostafa, & Zenke (2019)

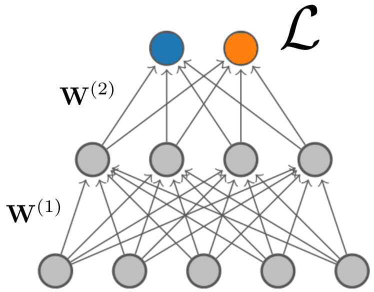
RTRL with spiking neurons

$$I_i^{(l)}[n+1] = \alpha I_i^{(l)}[n] + \sum_j W_{ij}^{(l)} S_j^{(l-1)}[n] + \sum_j V_{ij}^{(l)} S_j^{(l)}[n]$$

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

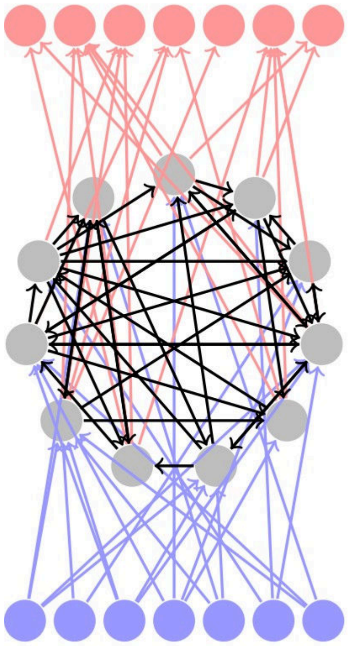
$$S_i^{(l)}[n] = \Theta(U_i^{(l)}[n] - \vartheta)$$





Zenke, F., and Neftci, E.O. (2021)

RTRL with spiking neurons



$$\frac{\partial S_k^{(l)}[n]}{\partial P} = \Theta'(U_k^{(l)}[n] - \vartheta)\left[\frac{\partial U_k^{(l)}[n]}{\partial P}\right]$$

$$\frac{\partial U_k^{(l)}[n+1]}{\partial P} = \beta\frac{\partial U_k^{(l)}[n]}{\partial P} + \frac{\partial I_k^{(l)}[n]}{\partial P} - \frac{\partial S_k^{(l)}[n]}{\partial P}$$

Zenke, F., and Neftci, E.O. (2021)

RTRL with spiking neurons

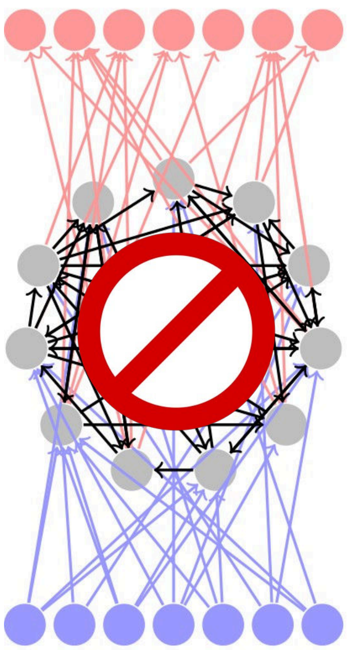
$$\frac{\partial I_k^{(l)}[n+1]}{\partial W_{ij}^{(a)}} = \alpha \frac{\partial I_k^{(l)}[n]}{\partial W_{ij}^{(a)}} + \delta_{ik} S_j^{(l-1)}[n] + \sum_p V_{kp} \frac{\partial}{\partial W_{ij}^{(a)}} S_p^{(l)}[n]$$

$$\frac{\partial I_k^{(l)}[n+1]}{\partial V_{ij}^{(a)}} = \alpha \frac{\partial I_k^{(l)}[n]}{\partial V_{ij}^{(a)}} + \delta_{ik} S_j^{(l)}[n] + \sum_p V_{kp} \frac{\partial}{\partial V_{ij}^{(a)}} S_p^{(l)}[n]$$

Zenke, F., and Neftci, E.O. (2021)

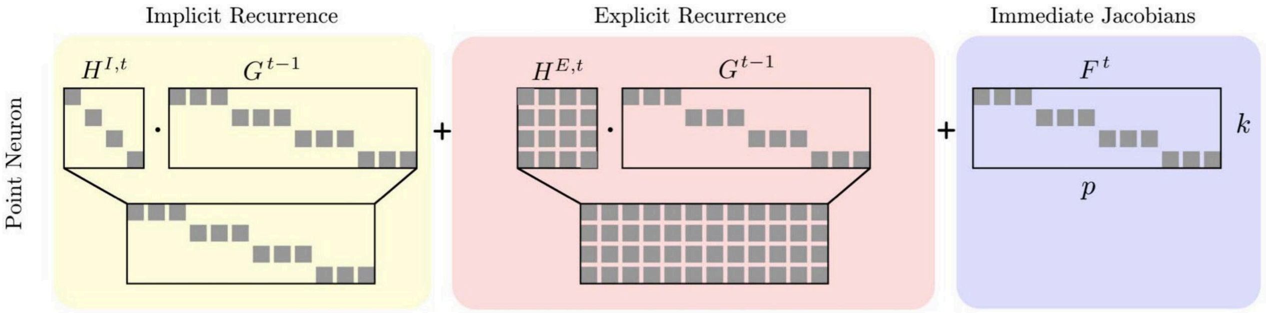
Ignoring **explicit recurrence** in gradient computation makes **learning rules local***

(* we still need to worry about spatial credit assignment though)



$$\frac{\partial I_k^{(l)}[n+1]}{\partial W_{ij}^{(l)}} = \alpha \frac{\partial I_k^{(l)}[n]}{\partial W_{ij}} + \sum_m \delta_{ik}\delta_{jm} S_m^{(l-1)}[n]$$

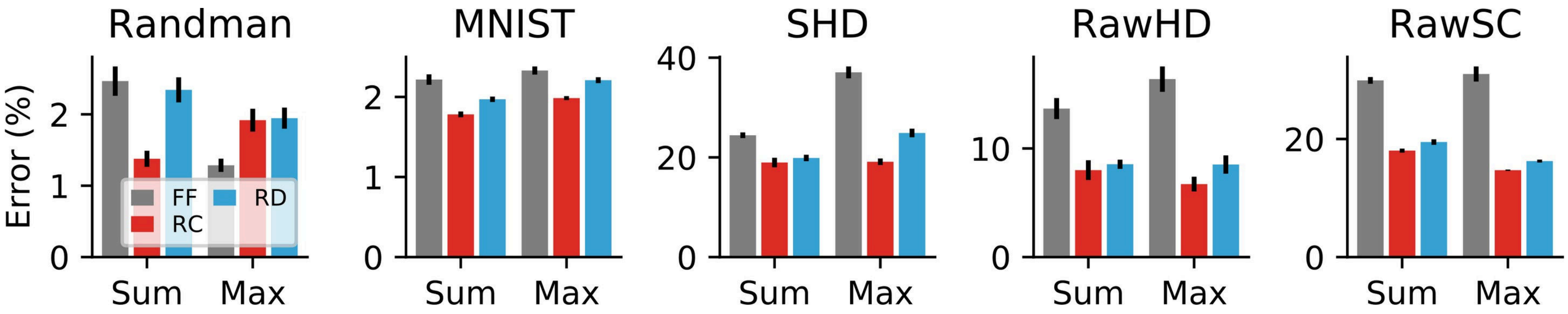
$$= \alpha \frac{\partial I_k^{(l)}[n]}{\partial W_{ij}^{(l)}} + \delta_{ik} S_j^{(l-1)}[n]$$



Zenke, F., and Neftci, E.O. (2021)

Approximate learning rules take advantage of recurrence and do almost as well as full BPTT

Friedrich Miescher Institute for Biomedical Research

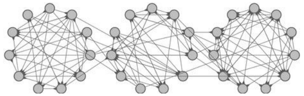


Speech processing problems

No recurrent connections

Recurrent (full BPTT)

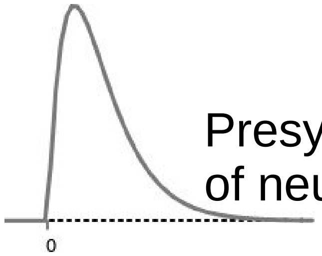
Recurrent (ignoring explicit recurrence in gradient)



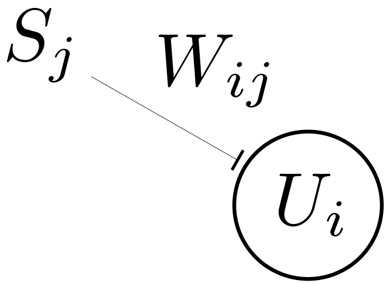
Zenke, F., and Neftci, E.O. (2021)

Friedrich Miescher Institute
for Biomedical Research

Real neurons could actually do this ...



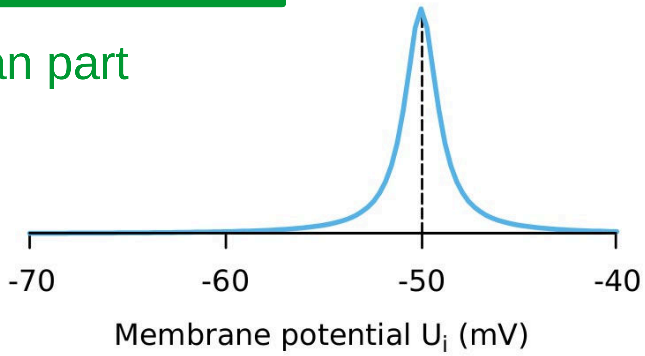
Presynaptic filter – e.g., presence of neurotransmitter at the PSD



$$\Delta W_{ij} \propto \underbrace{e_i(t)}_{\text{feedback}} \; \alpha * \underbrace{(\epsilon * S_j(t)}_{\text{pre}} \underbrace{\boxed{\sigma'}(U_i))}_{\text{post}}$$

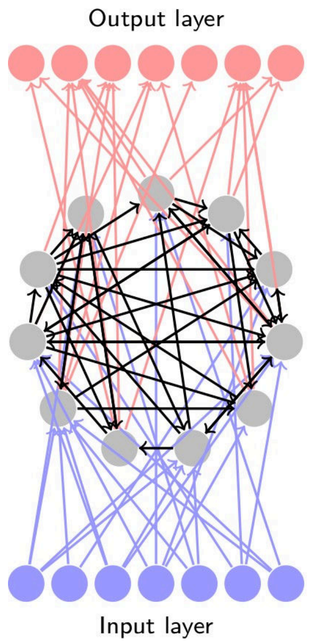
Third factor — Hebbian part

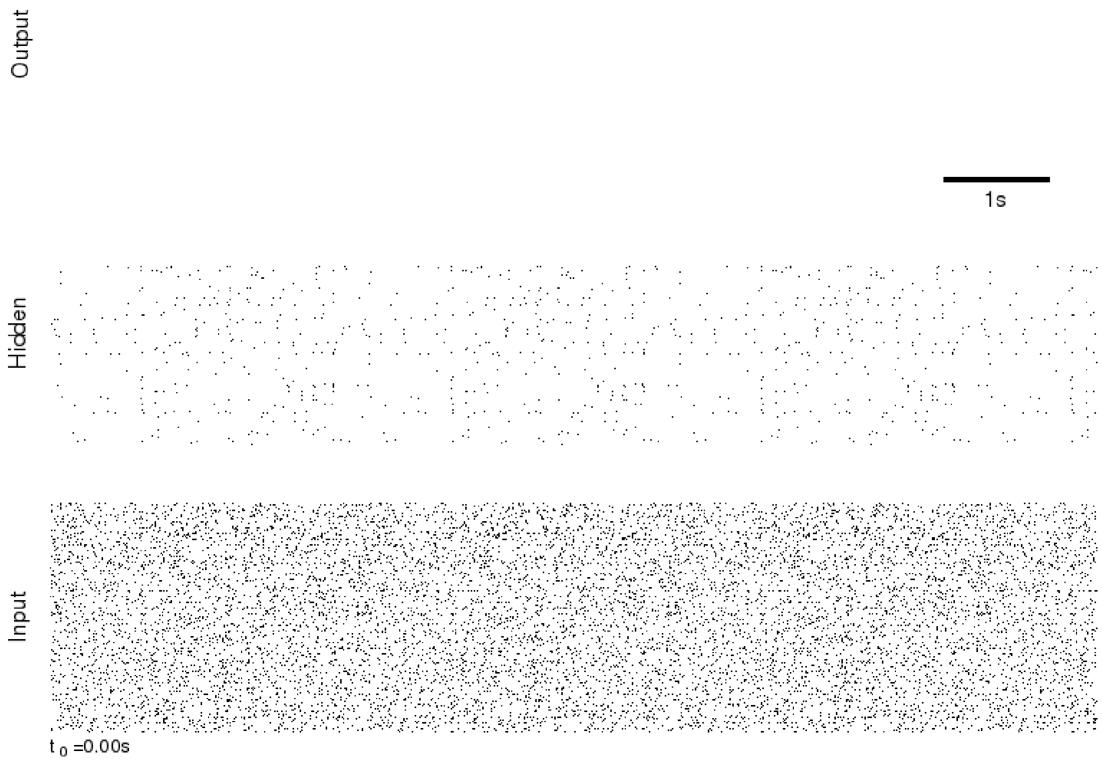
- Hebbian part
- Voltage-based nonlinearity
- Eligibility trace (“Ca transient”)
- Third factor required for spatial credit assignment
  Not sure yet, but could be neuromodulators or inhibitory microcircuits



Zenke & Ganguli (2018)

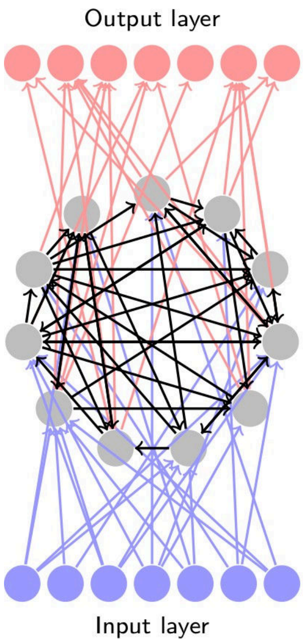
Network activity during online learning

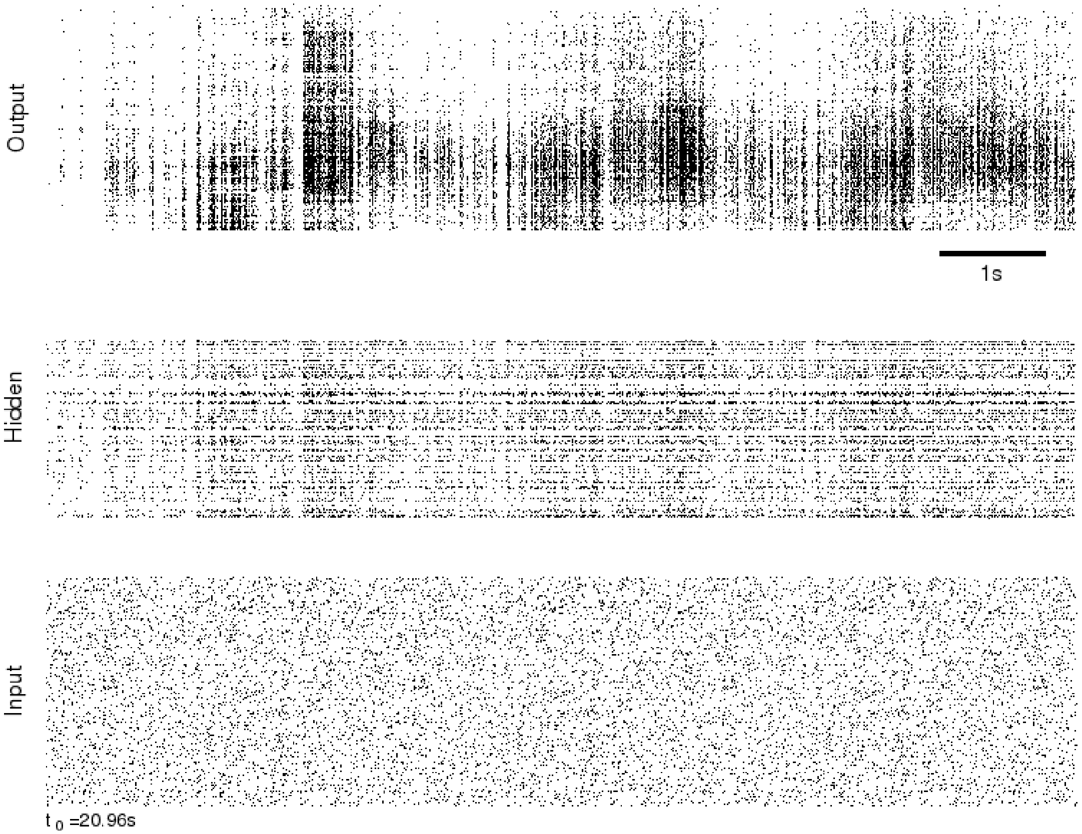




Zenke & Ganguli (2018)

Network activity during online learning

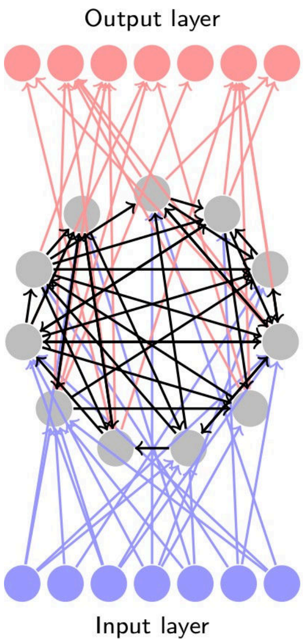


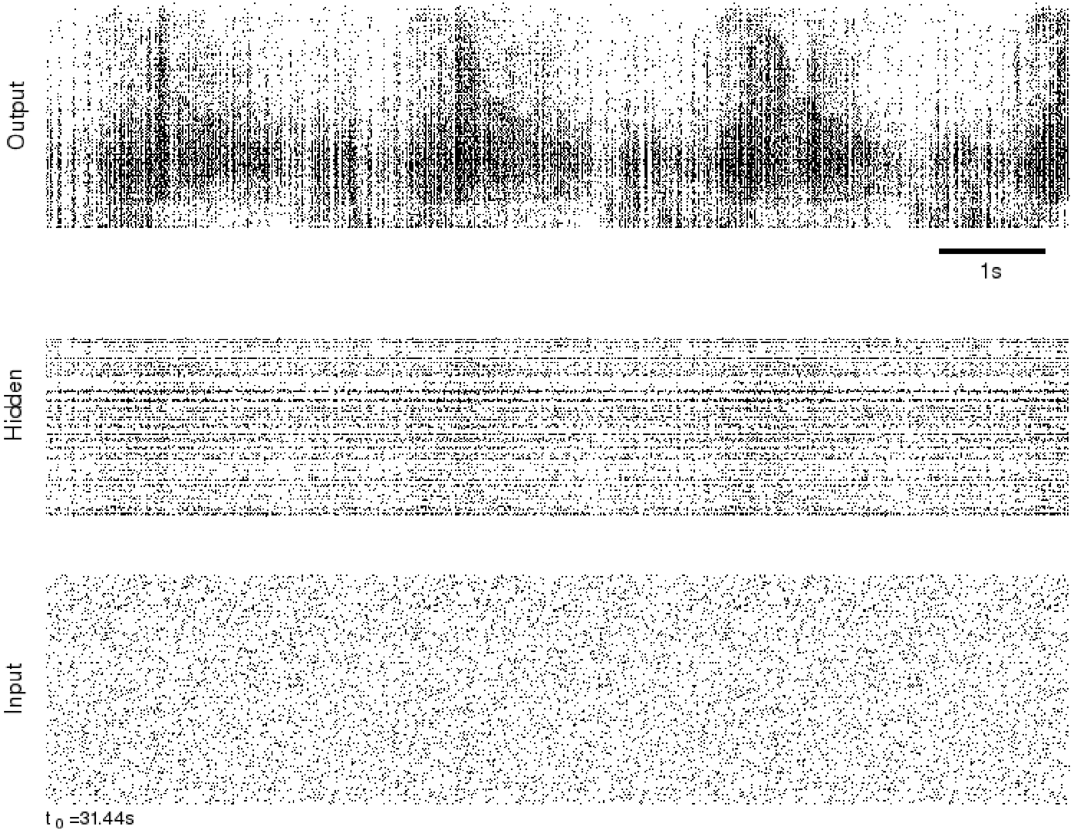


Zenke & Ganguli (2018)

Network activity during online learning

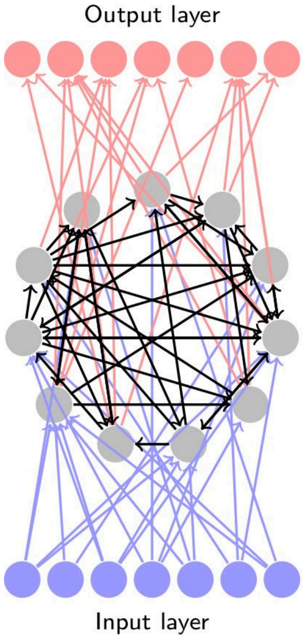
Friedrich Miescher Institute
for Biomedical Research

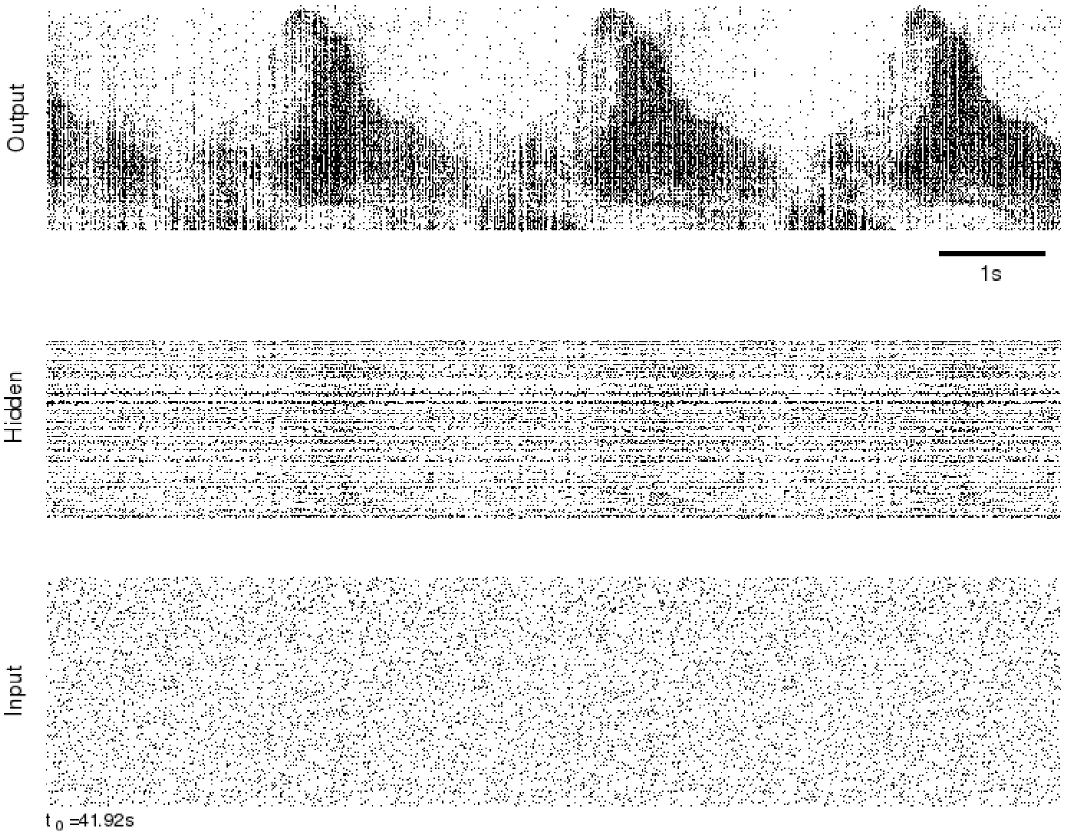




Zenke & Ganguli (2018)

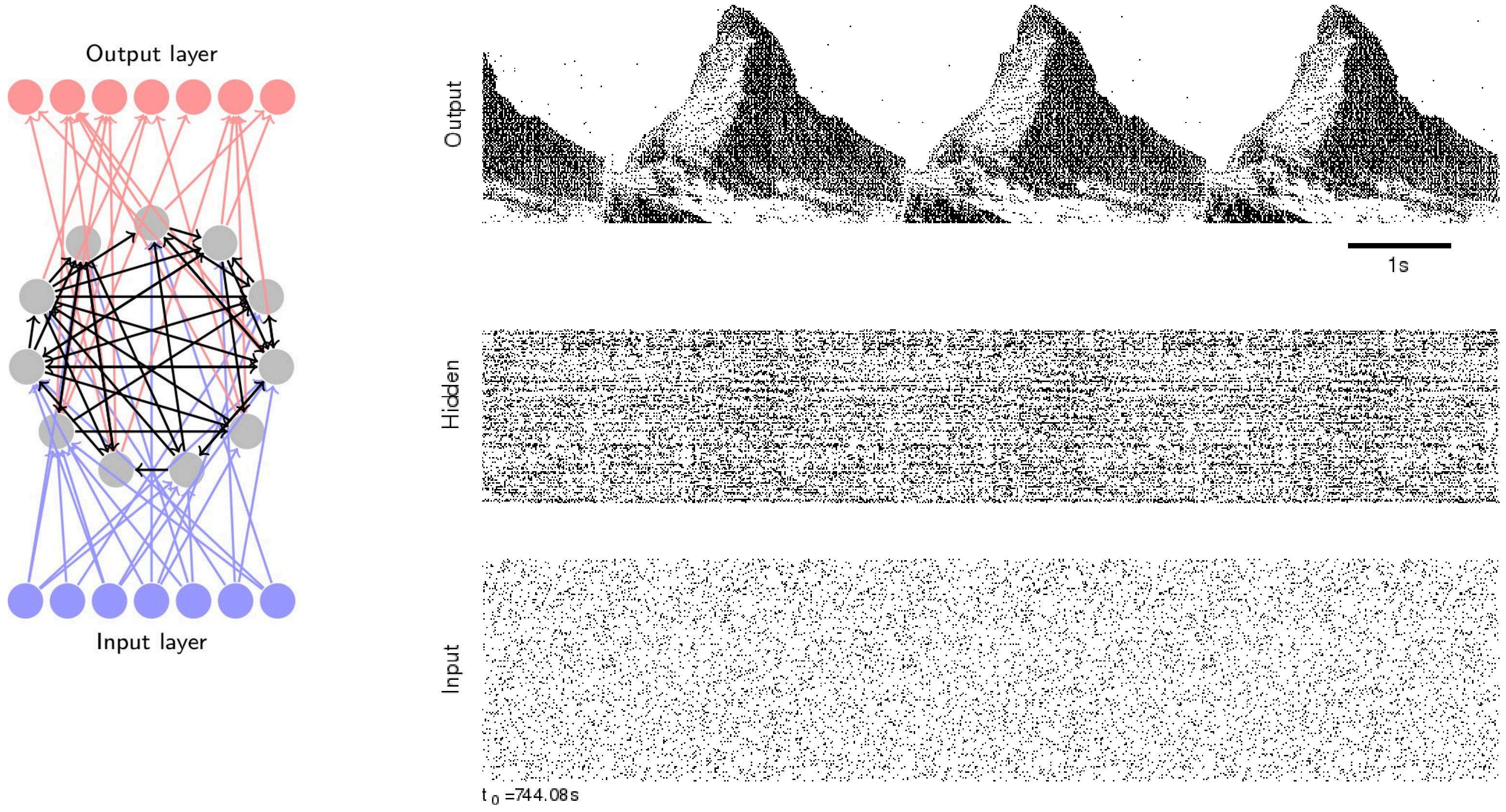
Network activity during online learning





Zenke & Ganguli (2018)

Network activity during online learning



Zenke & Ganguli (2018)

Summary

- Surrogate gradients are an effective tool for building functional spiking neural networks …
- … and studying plasticity mechanisms from a functionally motivated angle
- Current and future work
  - Functional networks with anatomical constraints
  - Unsupervised and self-supervised learning
  - Model validation through quantitative comparison to in-vivo data

Please try this at home

- https://github.com/fzenke/spytorch

Thanks!

Postdoc advisors


Surya


Tim

Collaborators

Emre Neftci
UC Irvine

Johannes Schemmel
Uni Heidelberg

Nicol Harper
Uni Oxford

Students


Kris


Manu


Manvi


Julian


Julia

External


Benjamin


Sebastian


Luke

With Karlheinz Meier † & Johannes Schemmel at University of Heidelberg

With Nicol Harper at University of Oxford

FMI International PhD program

NOVARTIS

Affiliated with the Novartis Institutes for BioMedical Research

Artwork: kyadava.net