Biological Inspiration for Improving Computing and Learning in Spiking Neural Networks

Wolfgang Maass

Institute of Theoretical Computer Science
Graz University of Technology, Austria

Focus of our research in Graz: Getting attractive AI-performance into spike-based neuromorphic hardware

Options for that:

1. Train an ANN, and use its weights for an SNN on the chip

2. Train an SNN off-chip, and use its weights on the chip

3. Train spiking DNNs on the chip.

I will present biologically inspired methods for each of these three options.

We propose that this as a generally fruitful research strategy for neuromorphic hardware (NMH):

Combine the best of two worlds: ML/AI and brain science.

1. Train an ANN, and use its weights for the SNN on the chip

Question: If one needs to use offline training, why should one first train an ANN offline, rather than training an SNN offline?

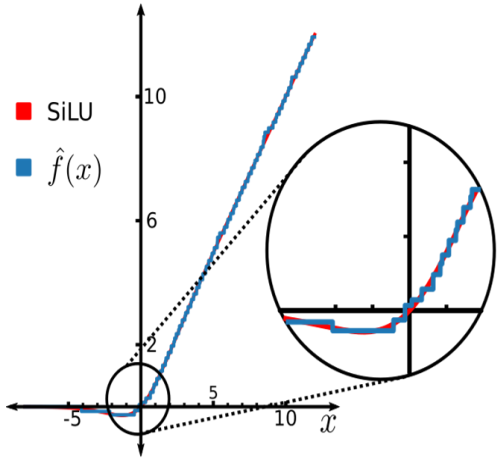
Answer: E.g. for image classification the industrial standard dataset ImageNet is so large (1.2 million training examples) and the best performing ANNs are so large (250 million neurons) that this training process can only be carried out by a few companies (such as Google or Facebook) which have the computing resources for that.

Fortunately, Google has made the weights of the best performing trained ANNs for ImageNet (**EfficientNet**) public. Hence we can use them also in SNNs. But the question is **how**?

Obstacles for common ANN2SNN conversion methods:

1. With a rate-coding conversion one cannot expect an energy advantage NMH.

2. The best performing ANNs (EfficientNet) use the SiLU activation function, which poses additional obstacles for rate coding, since it **outputs both positive and negative values.**



Methods for ANN-to-SNN conversions

Our **new method:**

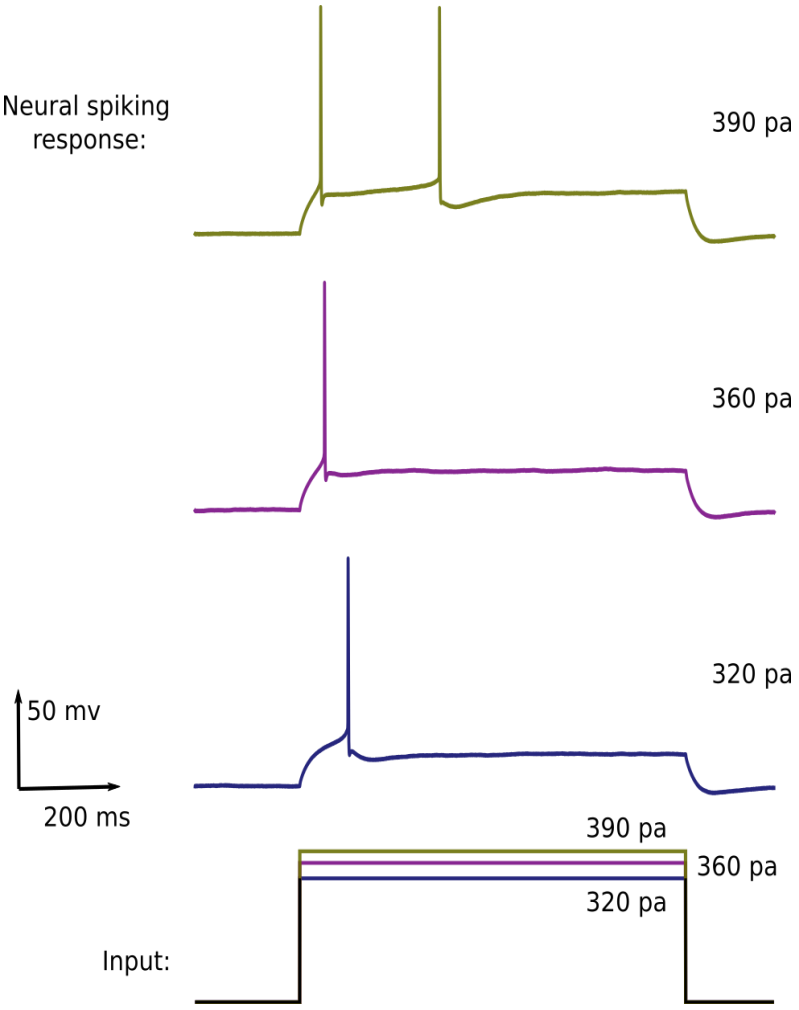
- Few spike (FS) coding.

Inspiration from biology:

Many neurons in the brain encode the amplitude of an input current by a **spike-pattern** with few spikes

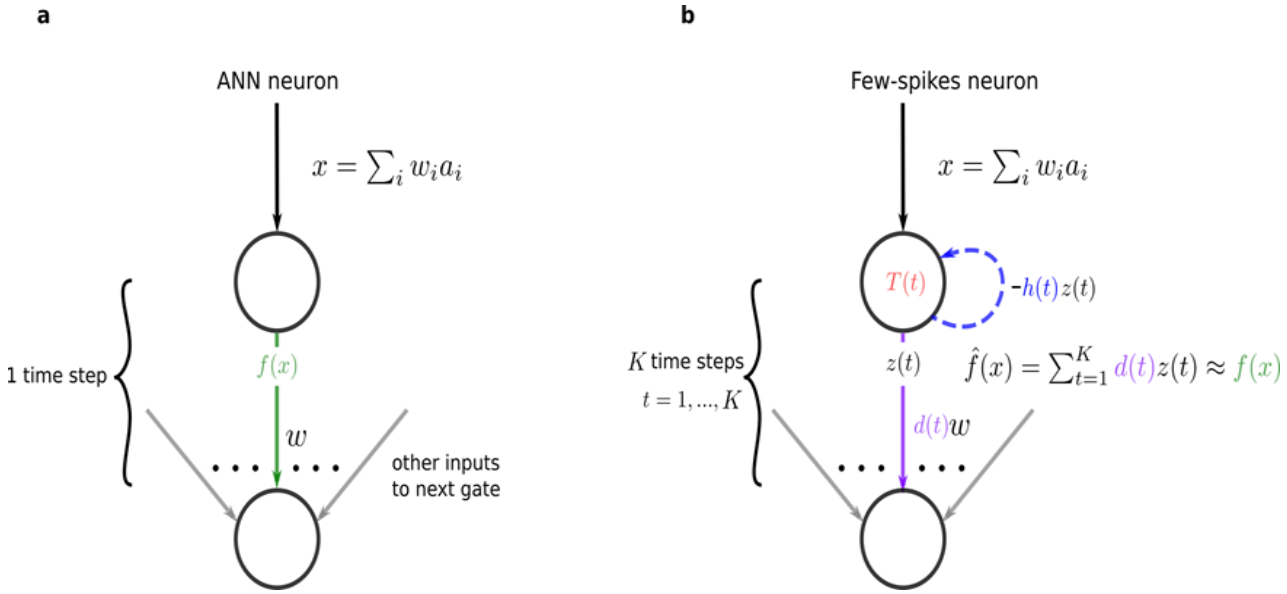
Inspiration from mathematics:

Binary coding



Source: Allen Cell Types Database (Layer 3 spiny neuron from the human middle temporal gyrus).

FS-coding requires a special type of spiking neuron: a FS-neuron

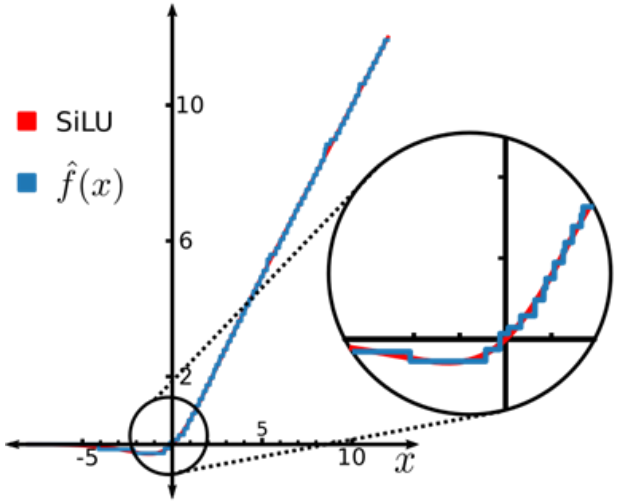


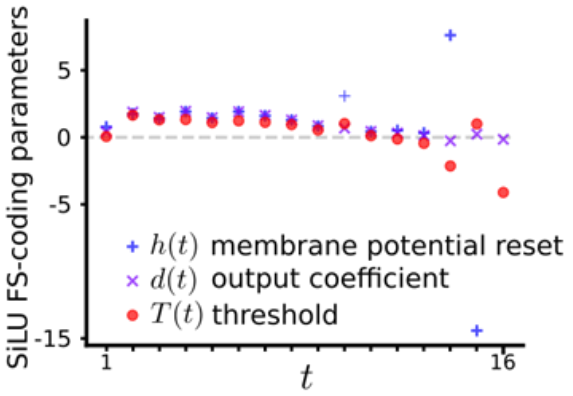
- To induce encoding of analog values x with few spikes within a short time window of K time steps (e.g., K = 16), we **endow the FS-neuron with an internal dynamics during these K time steps** (biological neurons actually also have such internal dynamics on a slower time scale).

- This internal dynamics is determined by parameters T(t), h(t), d(t) for t = 1, ..., K

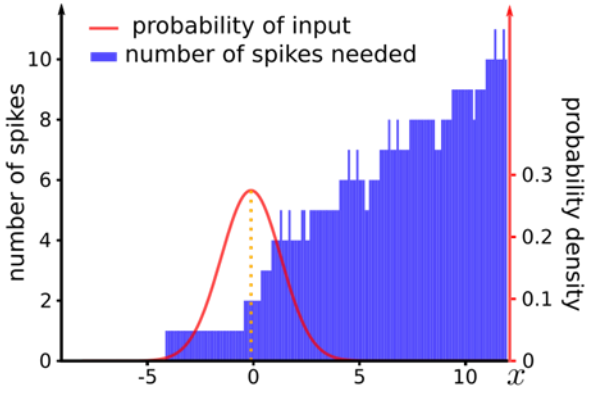
- These parameters are optimized to emulate specific ANN-neurons.

Example: Internal parameters of a FS-neuron that is optimized to emulate an ANN neuron with the SiLU activation function from EfficientNet





The FS-neuron that resulted from the parameter optimization uses the fewest spikes for the most frequently occurring values of x:



SNN performance on ImageNet that results from this new ANN2SNN conversion

baseball

seal

stork

ImageNet dataset:

- 1,281,167 training images
- 50,000 test images
- 1000 categories
  - among them for example 59 types of birds

Model	ANN accuracy	accuracy of the SNN produced by FS-conversion	# params	# layers	# neurons	# spikes
ImageNet2012						
EfficientNet-B7	85% (97.2 %)	83.57% (96.7%)	66M	218	259M	554.9M
ResNet50	75.22% (92.4%)	75.10% (92.36%)	26M	50	9.6M	14.045M

```
172 Great_Pyrenees
173 Chihuahua
174 tabby
175 marmoset
176 Labrador_retriever
177 Saint_Bernard
178 armadillo
179 Samoyed
180 bluetick
181 redbone
182 polecat
183 marmot
184 kelpie
185 gibbon
186 llama
187 miniature_pinscher
188 wood_rabbit
189 Italian_greyhound
```

Previous SNN record for ImageNet: 74.6% (Rueckauer et al., 2017)

Summary of section 1 of my talk

- FS conversion enables by far the best performance of SNNs on ImageNet.

- Since each layer of the SNN spends after FS-conversion just K (e.g., K = 16) time steps, the network can start to classify a new image every 2K time steps. Hence the resulting throughput is substantially higher than for rate-based ANN2SNN conversion.

- Discussions with several NMH designers lead to the conclusion that FS-neurons can be implemented at moderate cost in NMH. They will compete with new efforts to implement ANN neurons with ReLU activation functions directly in digital „NMH“. Not clear whether one can also implement the SiLU activation function efficiently directly in NMH.

- The option to encode information in spike patterns with few spikes seems to be under-researched (but used by nature)

- The paper on our approach appeared last week in print:

Christoph Stöckl, Wolfgang Maass. (2021). Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. Nature Machine Intelligence.

2. Train an SNN off-chip for on-chip inference

- Results in this direction were so far not encouraging for DNNs.

- But actually, one had focused on just one type of DNN: CNNs, where the problem is to get into a regime where the SNN achieves high accuracy with energy-efficient low firing rates.

Question: What is the situation for other DNNs that are important for AI?

I will focus here on **Relational Networks** (RelNets)., which have been developed in AI for reasoning about relations between items in a story, an image, a video, but can also be used for online reasoning about relations between items in simultaneously presented videos and stories.

For implementing RelNets in SNNs, we first have to emulate LSTM units in neuromorphic hardware (for encoding sentences).

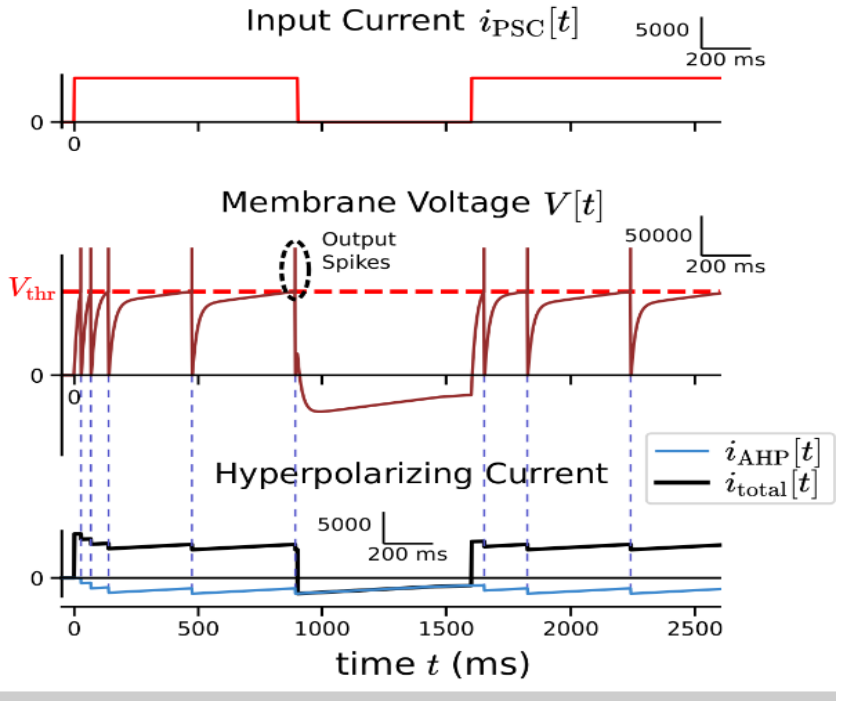
One can emulate LSTM units on Loihi via AHP-currents

It had already been shown in (Bellec et al., 2018) that LSTM units can be emulated by spiking neurons with spike frequency adaptation.

Spike-frequency adaptation can be implemented efficiently on Loihi in the same way as in the brain:

Via **spike-triggered**

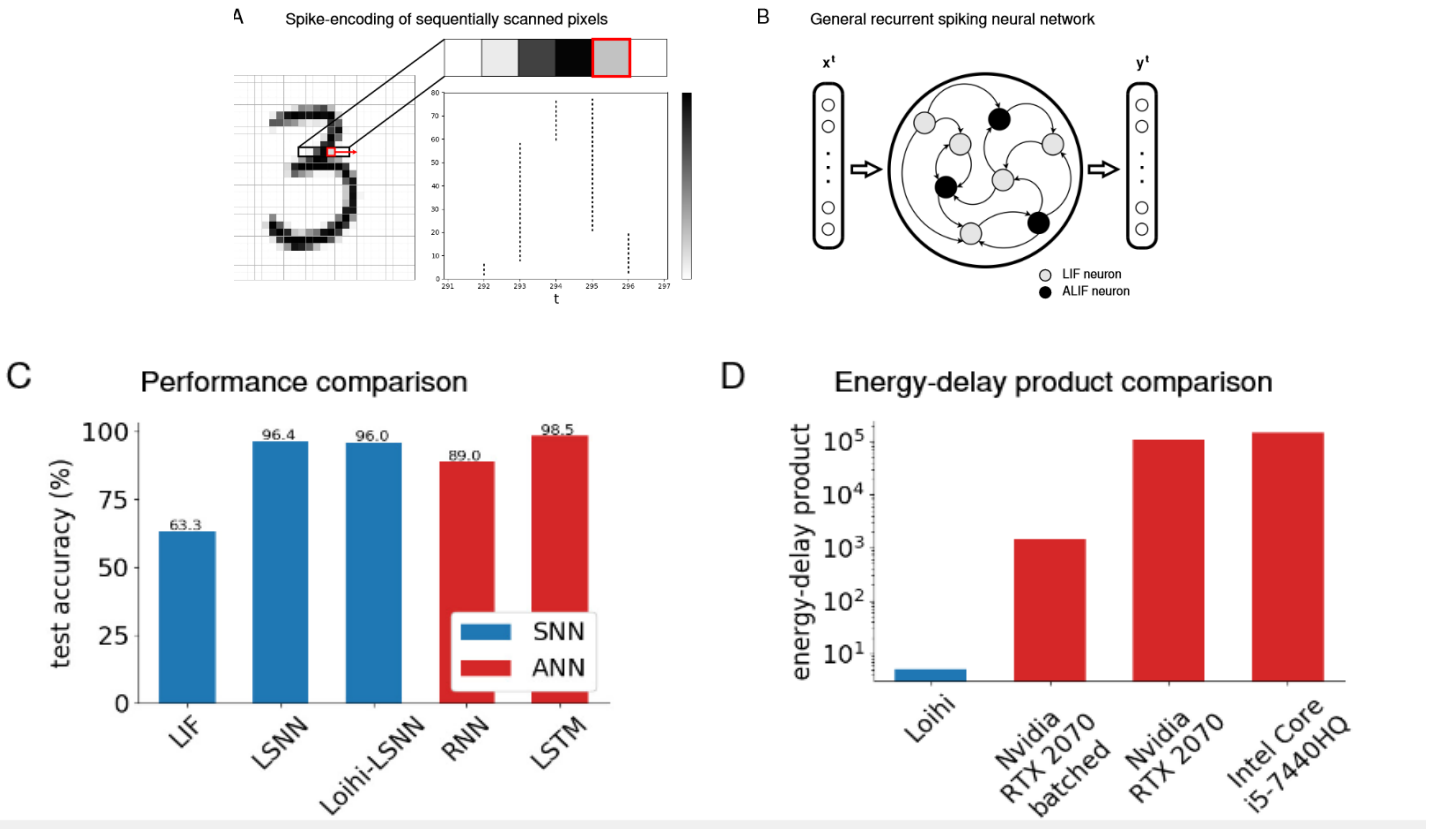
After-Hyper-Polarization (AHP) currents:



$$i_{\mathrm{AHP}}[t+\delta t] = \alpha_{AHP}\, i_{\mathrm{AHP}}[t] - (1-\alpha_{AHP})\beta\, z[t]$$

z[t] = 1 if the neuron fires at time t,
z[t] = 0 otherwise

First performance test:
Time series classification (sequential MNIST) on Loihi



We achieved via spiking neurons with AHP currents on Loihi **almost the same classification accuracy as LSTM networks**, but with an **Energy-Delay-Product that was by several orders of magnitude smaller.**

Back to RelNets, and the implementation challenge for large DNNs on Loihi

We developed a spike-based variation of the ANN-RelNet of

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., & Lillicrap, T. (2017). A simple neural network module for relational reasoning. arXiv preprint arXiv:1706.01427.

They had tested their RelNet on answering questions about

- relations between visual objects in an image (CLEVR dataset), and
- **relations between objects, persons, and actions in stories**, given as sequences of sentences in natural language (**bAbI tasks**).

We focused on the application to natural language, since that requires no CNN for preprocessing.

Examples for 6 bAbI tasks (Weston et al., 2016)

Task 5: Three Argument Relations

Mary gave the cake to Fred.
Fred gave the cake to Bill.
Jeff was given the milk by Bill.
Who gave the cake to Fred? A: Mary
Who did Fred give the cake to? A: Bill

Task 13: Compound Coreference

Daniel and Sandra journeyed to the office.
Then they went to the garden.
Sandra and John travelled to the kitchen.
After that they moved to the hallway.
Where is Daniel? A: garden

Task 7: Counting

Daniel picked up the football.
Daniel dropped the football.
Daniel got the milk.
Daniel took the apple.
How many objects is Daniel holding? A: two

Task 15: Basic Deduction

Sheep are afraid of wolves.
Cats are afraid of dogs.
Mice are afraid of cats.
Gertrude is a sheep.
What is Gertrude afraid of? A:wolves

Task 19: Path Finding

The kitchen is north of the hallway.
The bathroom is west of the bedroom.
The den is east of the hallway.
The office is south of the bedroom.
How do you go from den to kitchen? A: west, north
How do you go from office to bathroom? A: north, west

Task 20: Agent's Motivations

John is hungry.
John goes to the kitchen.
John grabbed the apple there.
Daniel is hungry.
Where does Daniel go? A:kitchen
Why did John go to the kitchen? A:hungry

A task is considered solved if there are at most 5% errors on new test stories from the task.

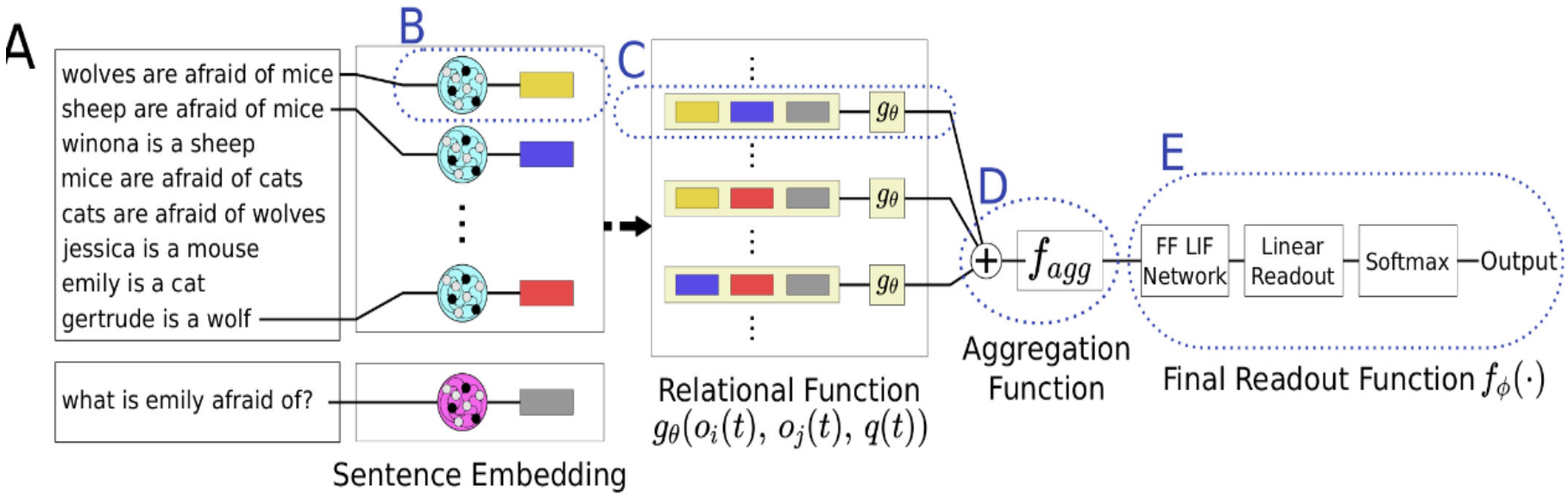
Structure of the spike-based RelNet

Example:

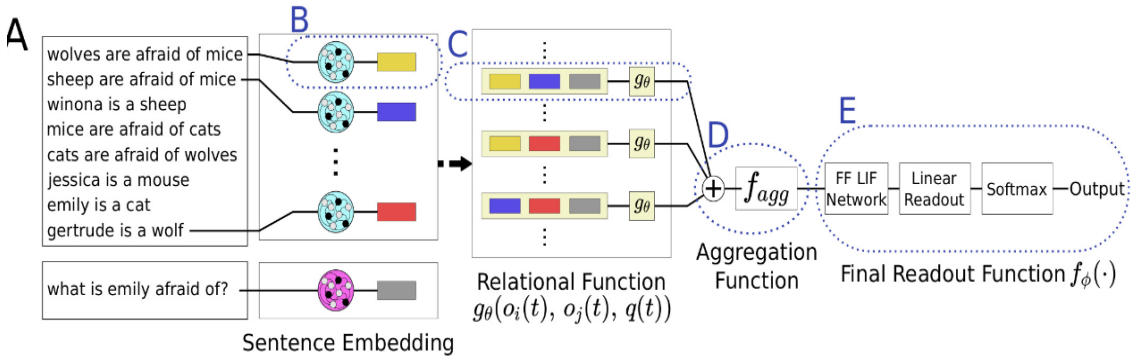
Application of the spiking RelNet to a Basic Deduction task:

Task 15: Basic Deduction

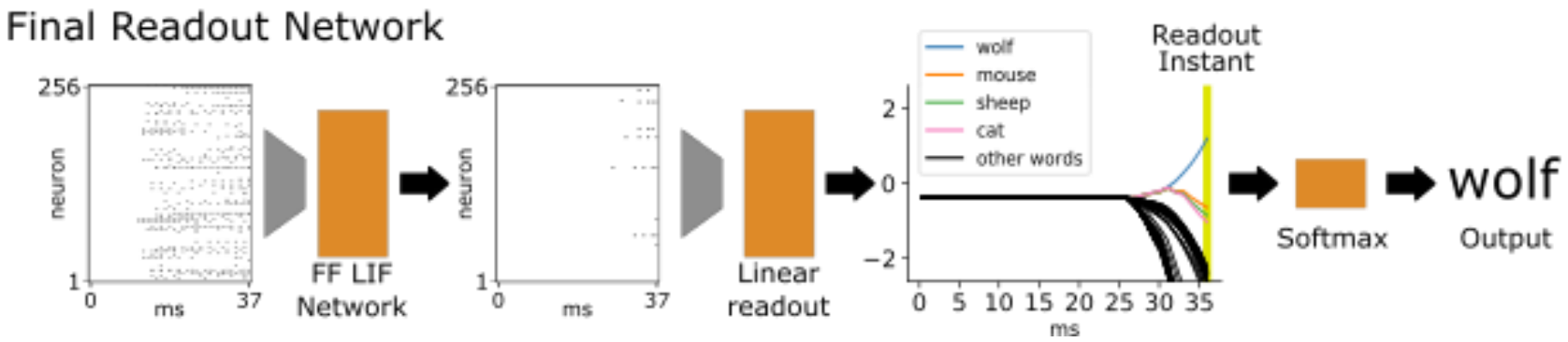
Sheep are afraid of wolves.
Cats are afraid of dogs.
Mice are afraid of cats.
Gertrude is a sheep.
What is Gertrude afraid of? A:wolves



Relational Reasoning on Loihi



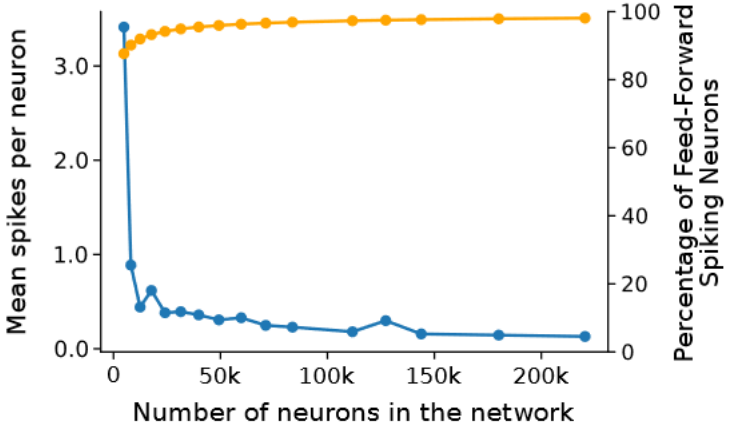
- More than 95% of RelNet consists of feedforward modules
- Hence we **needed to make activity in the feedforward modules extremely sparse** in order to be energy-efficient
- Note that the size of module C grows quadratically with the number of sentences in a story. This allowed us to measure energy consumption for several effective RelNet sizes
- An **important step for enforcing event based computing** was to insist that the SNN output is given at a single time step:

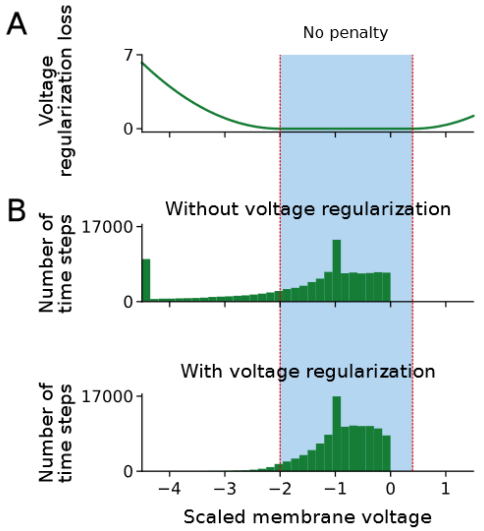


Bringing the spiking RelNet into an event based activity regime

- In fact, the activity became sparser for larger RelNets (their size grows roughly quadratically with the number of sentences in the story);
- Likely explanation: the number of „interesting relations“ that the network extracts does not grow equally fast
- We introduced in addition a generally useful tool for sparsening firing activity in gradient descent training:

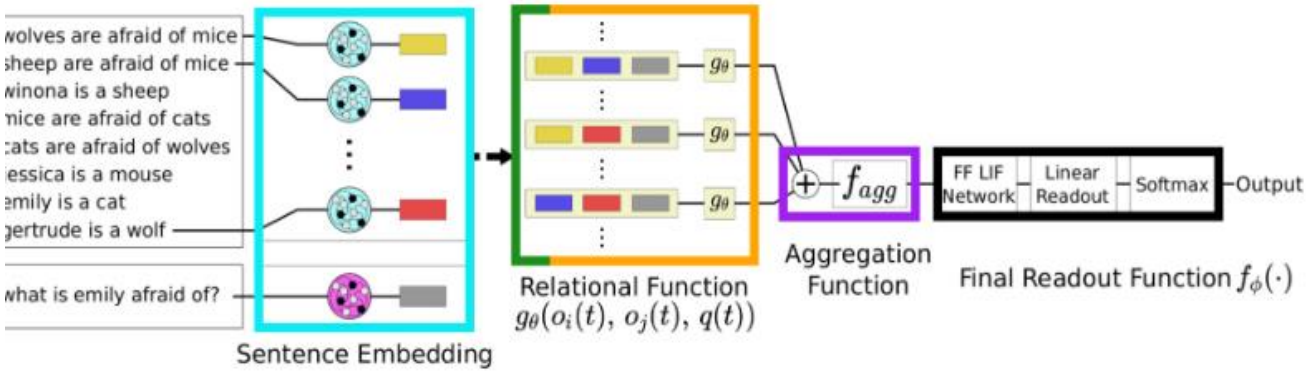
  Membrane voltage regularization
- **It allows us to use strong spike-rate regularization** without locking neurons into an ineffective strongly hyperpolarized state:



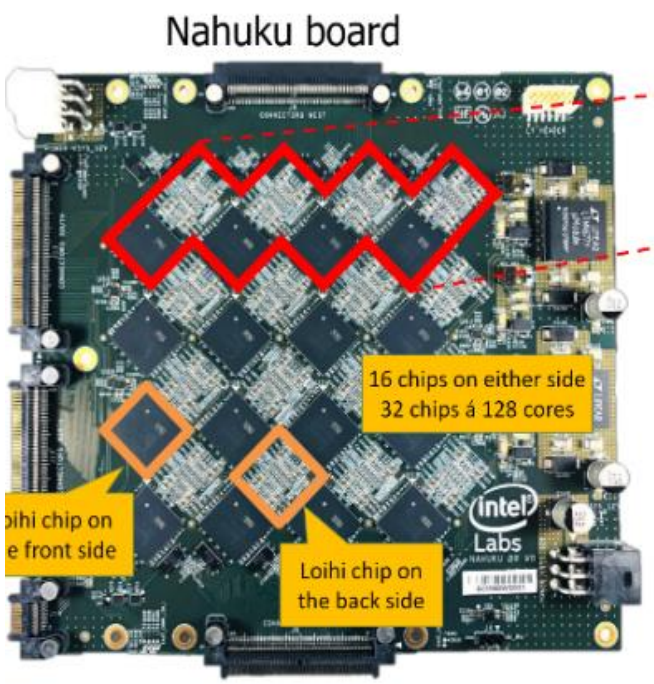


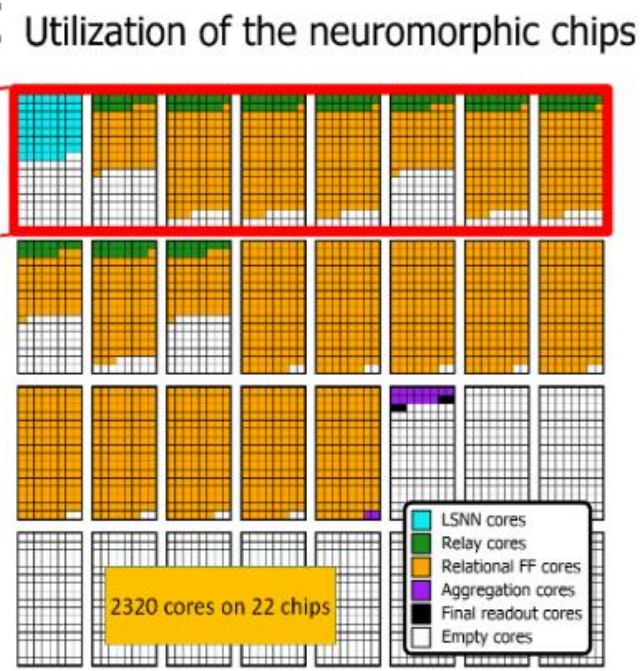
Implementation on 22 Loihi chips

A few tricks (and many additional cores) were needed to overcome constraints on fan-out and number of synapses of a core.



Additional relay cores (green) turned out to be useful for reducing inter-chip communication.





Energy consumption of ANN RelNets on GPUs relative to spiking RelNets on Loihi

	Relational reasoning				
	GPU				
# cores on Loihi	124	332	700	1552	2320
# sentences (RR)	2	6	10	16	20
Energy ratio	16.49x	11.92x	7.78x	5.32x	4.36x
Latency ratio	0.73x	0.56x	0.44x	0.33x	0.38x
EDP ratio	12.10x	6.73x	3.41x	1.73x	1.67x

All ratios shown against Loihi (=1).

RelNet can solve 17 of the 20 bAbI tasks. In addition Task 19 was excluded because it takes too much computation time on Loihi.

Results:

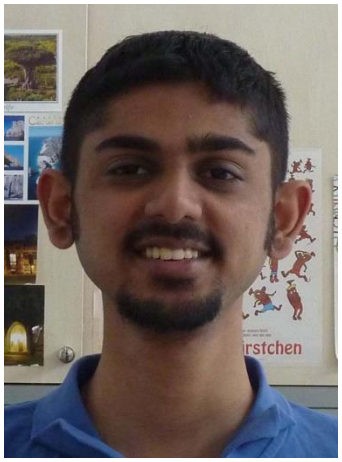
- Loihi needs for relational reasoning 4-12 time less energy than a GPU
- Loihi is somewhat slower than the GPU (apparently due to interchip communication and the larger number of computation steps needed on Loihi)
- Nevertheless, Loihi had for all problem sizes a lower EDP (6 times lower in the most frequently occurring range of stories with up to 6 sentences)

Summary of section 2 of my talk

- LSTM units can be efficiently emulated on Loihi with AHP-currents
- This makes emulations of LSTM networks on Loihi substantially more energy efficient than LSTM networks on GPUs
- RelNets use LSTM units in a small submodule, but still can be implemented more energy efficiently on Loihi
- One reason for that is intrinsic to this type of AI task: salient relations between items tend to be grow slowly with problem size (use attention for vision tasks?)
- Paper in preparation:

Arjun Rao, Philipp Plank, Andreas Wild, Wolfgang Maass. A long short-term memory for implementing AI in neuromorphic hardware

Andreas Wild



Arjun Rao

Philipp Plank

3. Train spiking DNNs on the chip

- E-prop can be implemented on SpiNNaker and Loihi2 for on-chip learning
- But software simulations suggest that e-prop learns slower than BPTT
- Hence we need variations of e-prop that enable fast learning.

One slide on *e-prop* *(„eligibility trace forward propagation“)*

Combines insight from **neuroscience** and **theory**:

- **From neuroscience:** Role of local **eligibility traces** and **top down learning signals** (third factors)

- **From theory:** Gradient descent for the network loss *E* can be written rigorously in the form:

$$\frac{dE}{dW_{ji}} = \sum_t L_j^t e_{ji}^t$$

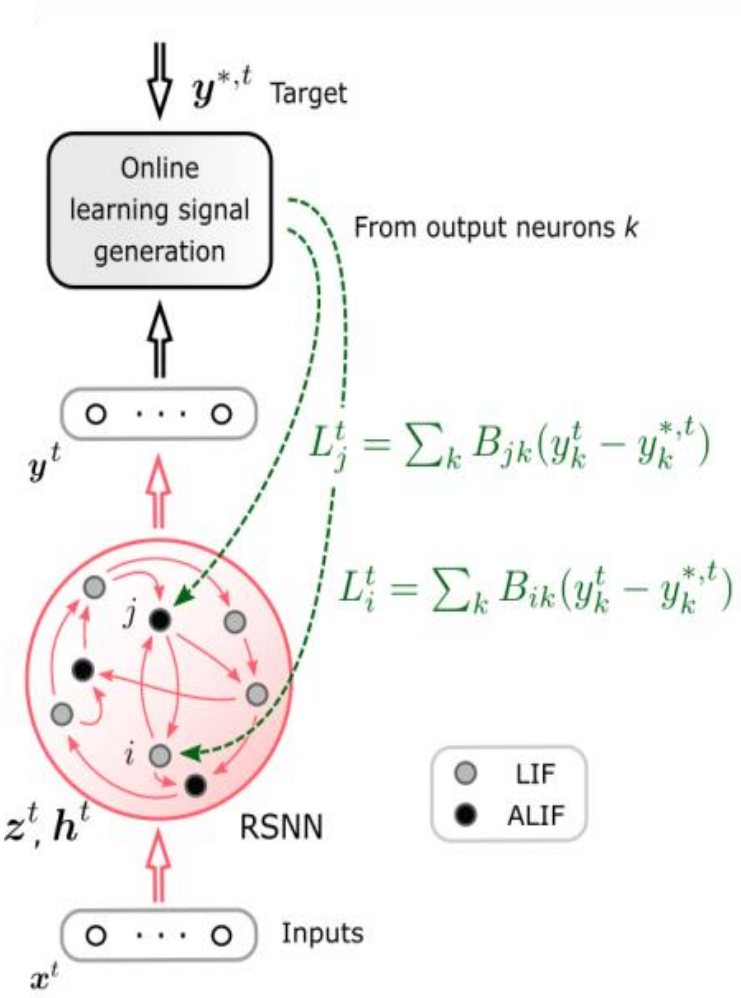
learning signal **eligibility trace**

This suggests the following **online** learning rule for the synapse from neuron i to j:

modify W_{ji} **at time** t **by** $-L_j^t \cdot e_{ji}^t$

Ideal learning signals L_j^t usually require knowledge of the future, and need to be replaced for online learning by online approximations

The **ideal learning signal** L_j^t for neuron j at time would be $\frac{dE}{dz_j^t}$, which also depends on losses **after time t** .



Simple online approximations of this ideal learning signals were explored in

G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein, and W. Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. Nature Communications, 2020.

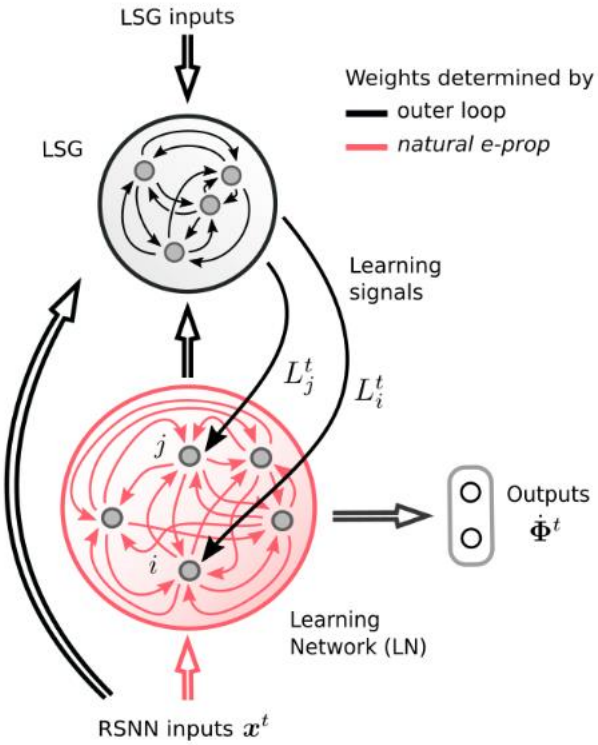
Biological inspiration:

Learning signals are generated in the brain by specialized brain structures, such as VTA

Hypothesis: The production of learning signals (such as DA) in the brain has been **optimized** by evolution in special brain structures, such as VTA, to support **fast learning of tasks that are important for survival**.

So lets do the same with our SNNs!

Rather than approximating **ideal learning signals based on gradients**, focus on generating learning signal that enable directly **fast learning** for more **limited ranges $\mathcal{F}$ of learning tasks.**

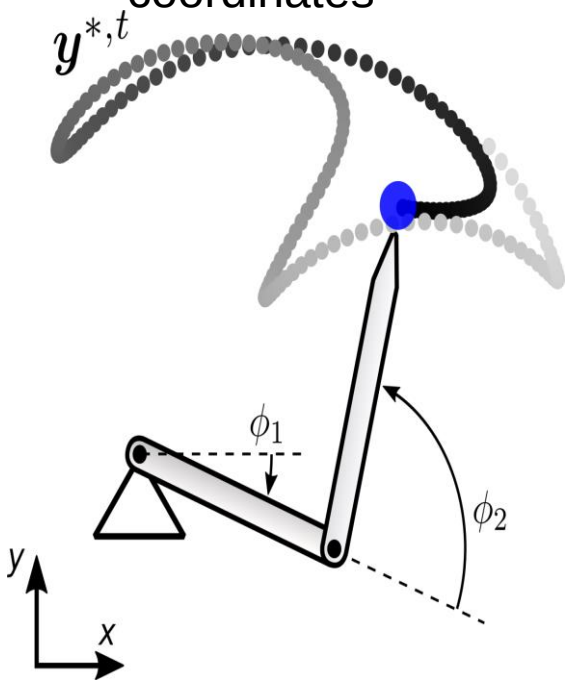


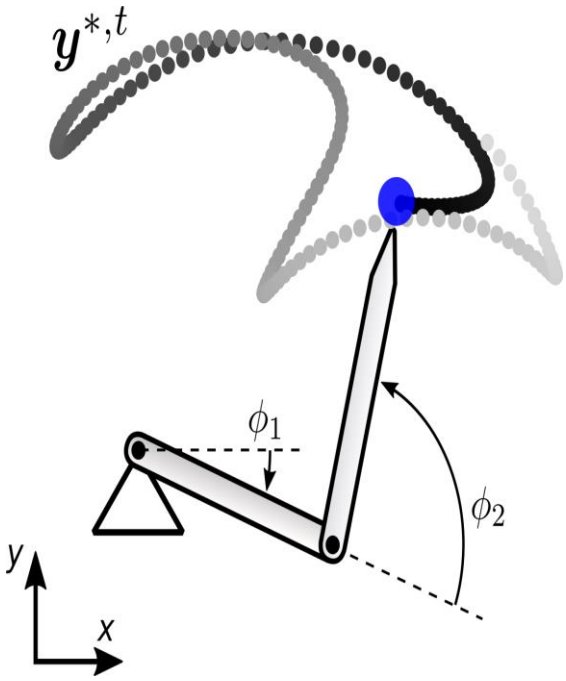
Example: Define the target range $\mathcal{F}$ of learning tasks as capability to reproduce any given arm movement

Movement demonstration in cartesian coordinates

Single weight update using learning signals provided by the error-module

Movement replicated using the two-joint arm





Result: One-shot learning of new arm movements

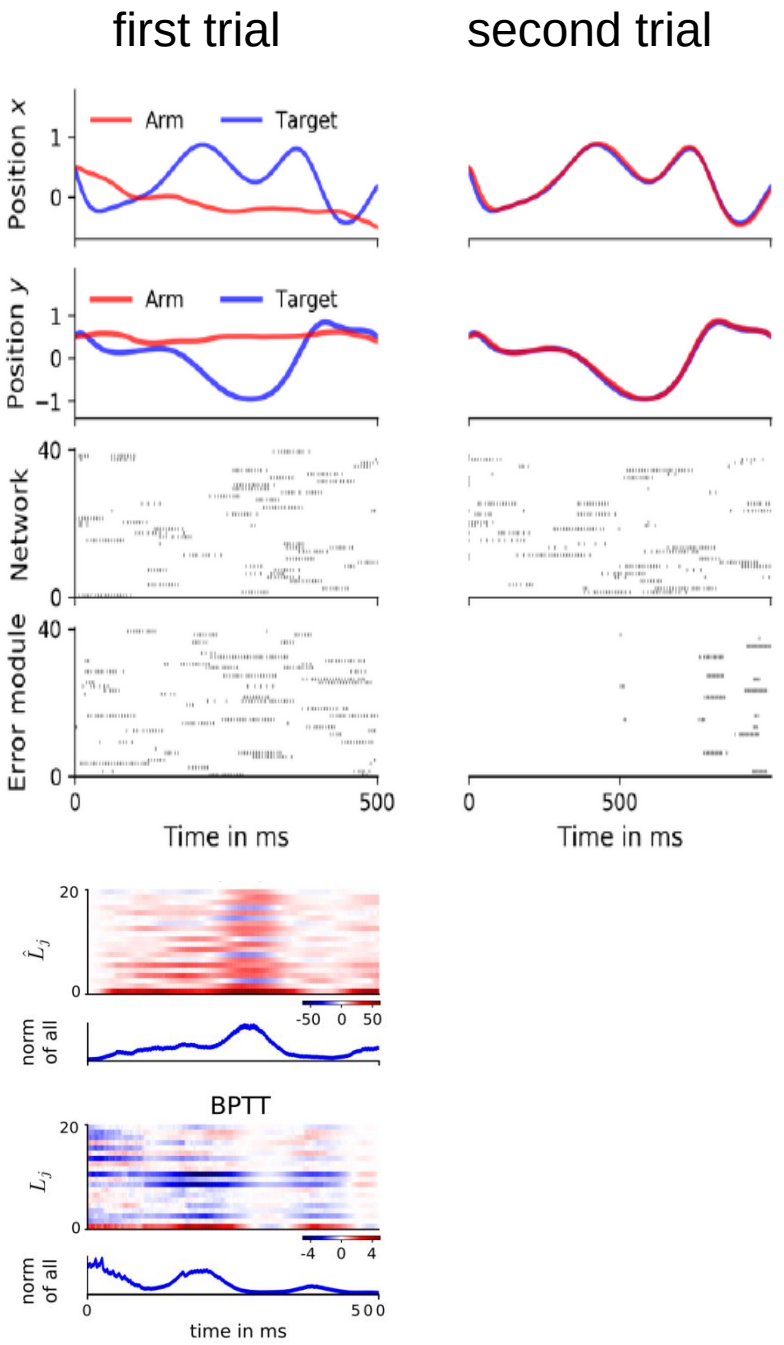
Arm movements

Firing activity in the RSNN

and in the error module (= another RSNN)

The learning signals that are emitted by the optimized error module (und used here by e-prop for one-shot learning)

are very different from the learning signals $\frac{dE}{dz_j^t}$ that BPTT proposes



Summary of part 3

- By optimizing the generation of learning signals via a special SNN (that can easily be implemented in NMH) **one can substantially speed up on-chip learning via e-prop for specific families of learning tasks.**

- Other applications that we explored: **One-shot learning of new (Omniglot) symbols**, and of **new spoken words**

First draft of a paper**:**

F. Scherr, C. Stoeckl, and W. Maass. One-shot learning with spiking neural networks. bioRxiv, 2020.

Summary of my talk

- I have demonstrated three biologically inspired methods for enhancing the performance and energy efficiency of spike-based AI tools
- We have shown that one can achieve in this way a classification performance of SNNs on ImageNet that is very close to the best CNN performance (using on average less than 2 spikes per neuron)
- We have also shown that in contrast to CNNs, large Relnets can be implemented efficiently in NMH
- Finally, fast and efficient variants of e-prop are on the way, enabling in some cases even one-shot learning by SNNs
- I view these as examples of a generally fruitful research strategy for NMH: To integrate the best of two worlds: ML/AI and brain science.