Analog artificial neural network

Filip Paulů

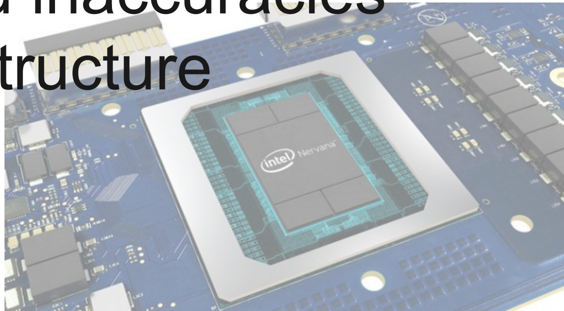
Czech Technical University in Prague

Digital (classic) vs. Analog

- software
- algorithm
- flexible structure
- von Neuman bottleneck
- processor clock

```
import sys
from pySageDecoder import PySageDecoder, PySageDecodable
import tensorflow_wrapper as tf_wp


if __name__ == '__main__':
    if len(sys.argv) < 2:
        sys.exit("expected arguments: bbn-decoder-args tf_model_file")

    tf_model_file = sys.argv.pop()
    tf_model_op = tf_wp.load_model(tf_model_file)
    decoder = PySageDecoder()
    decoder.setup(sys.argv)

    while not decoder.done():
        features = decoder.get_features()
        log_posterior = tf_wp.get_log_posterior(features, tf_model_op)
        decodable = decoder.get_matrix_decodable(log_posterior)
        decoder.decode(decodable)
        decoder.next()

    decoder.finalize()
```

- hardware
- electric circuit
- technology
- faster
- low-power
- noise and inaccuracies
- defined structure

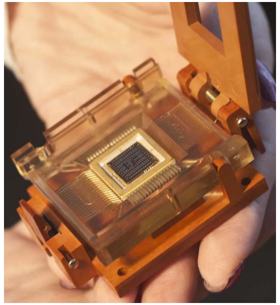


History

~1985 - 256 synapsys

~1995 - ~2017 - pause

~2014 - new start with memristor



Only forward propagation

FPGA VLSI ? Memristor

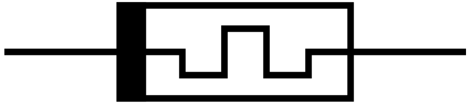
1980 ~1995 ~2010 2018

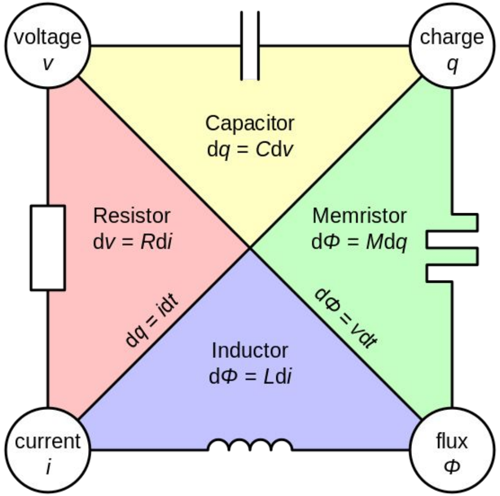
Today

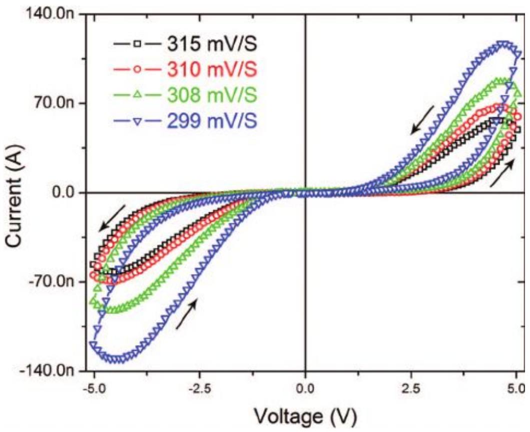
- GPU
- Analog accelerator
  - forward propagation
    - as USB devices
    - mobile devices
- 2018 simulation of analog structure with supervised learning

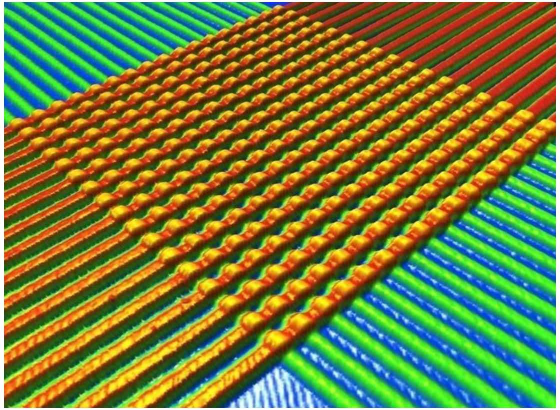
Memristor

“memory-resistor”

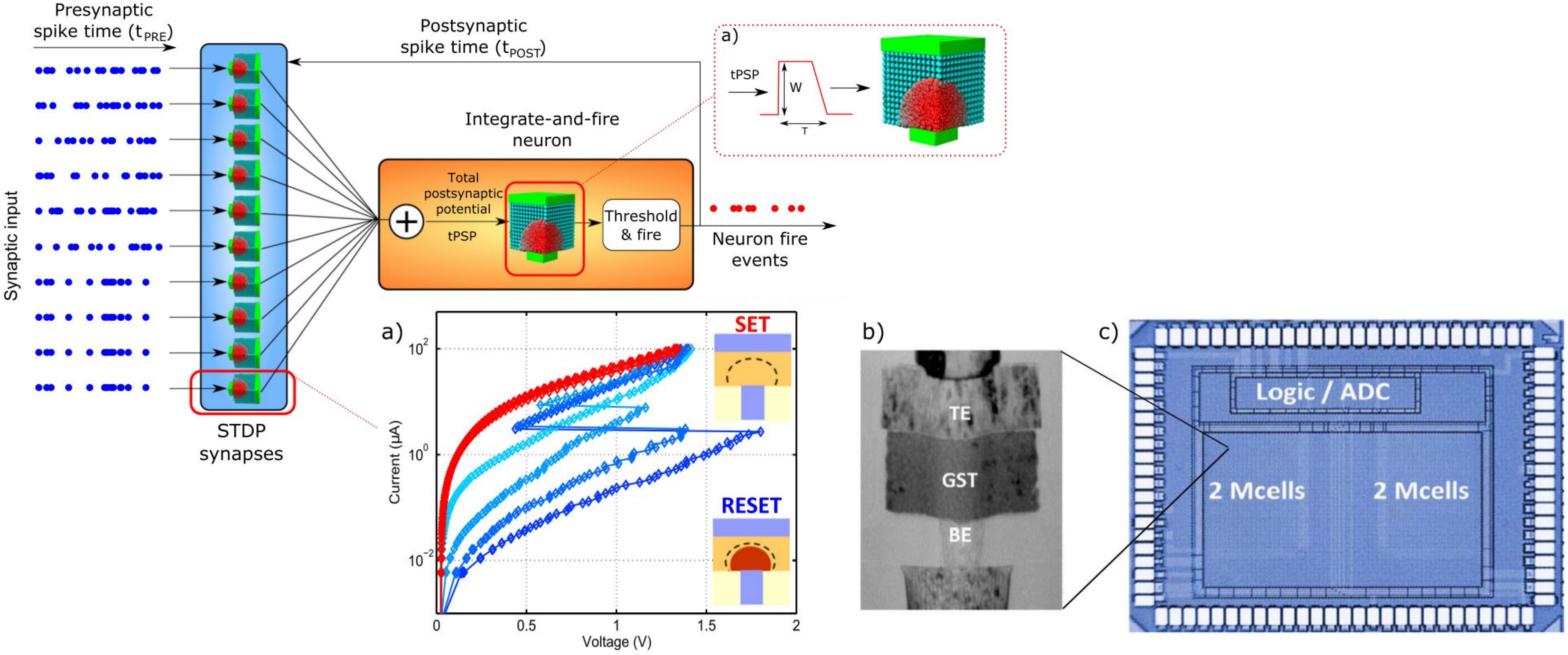








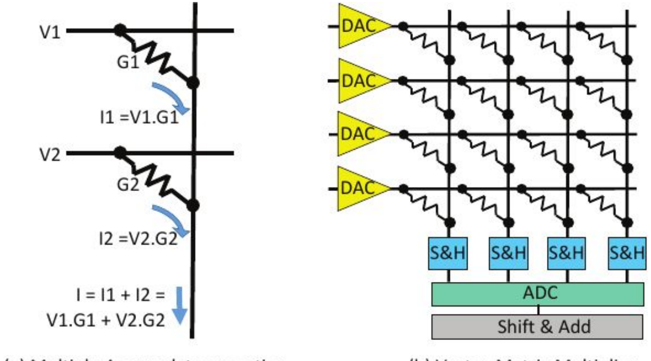
All-memristive neuromorphic computing



ISAAC: CNN accelerator

- 128 x 128 crossbar arrays
- 450× speedup
- 150× lower energy

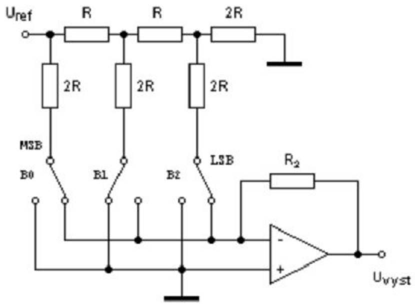
than an NVIDIA K20M GPU



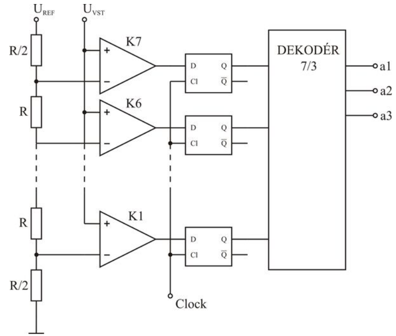
(a) Multiply-Accumulate operation

(b) Vector-Matrix Multiplier

DAC



ADC

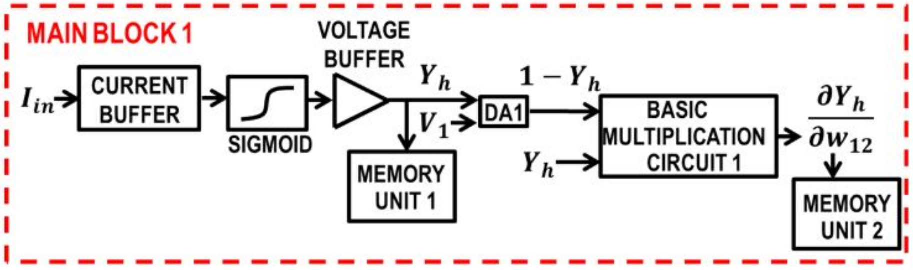


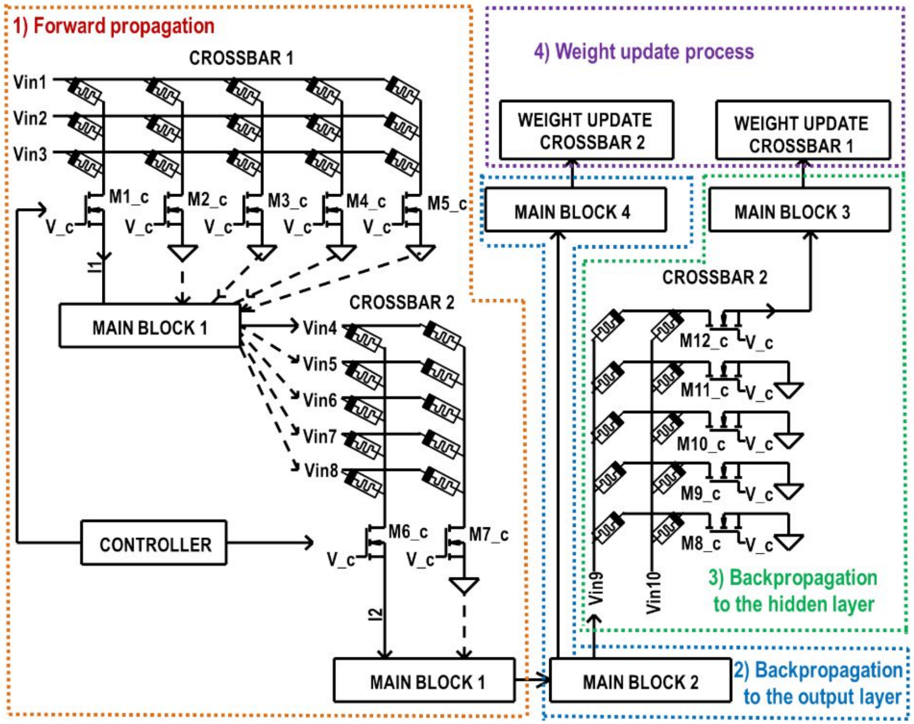
Learning

- Off-chip learning
- Chip-in-the-loop learning
- On-chip learning

Backpropagation (2018)

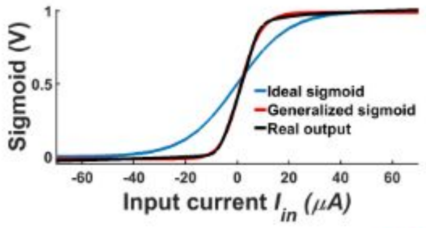
- circuit design
- SPICE
- memristive crossbar
- 180nm CMOS

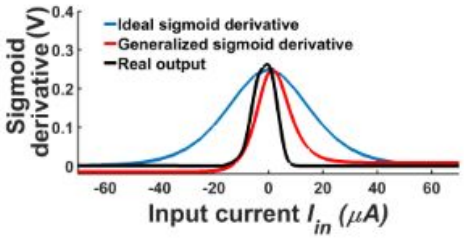


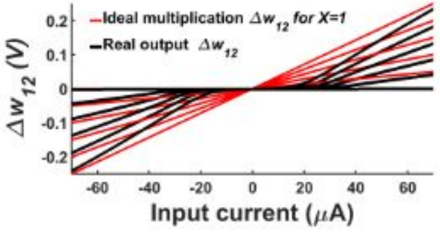


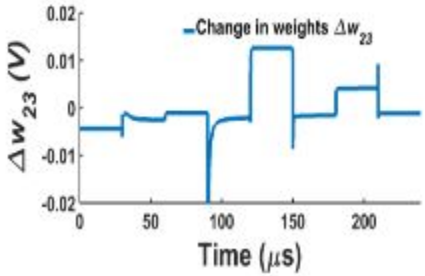
Backpropagation

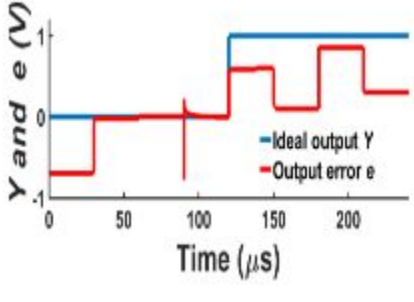
- not ideal components
- hard to simulation





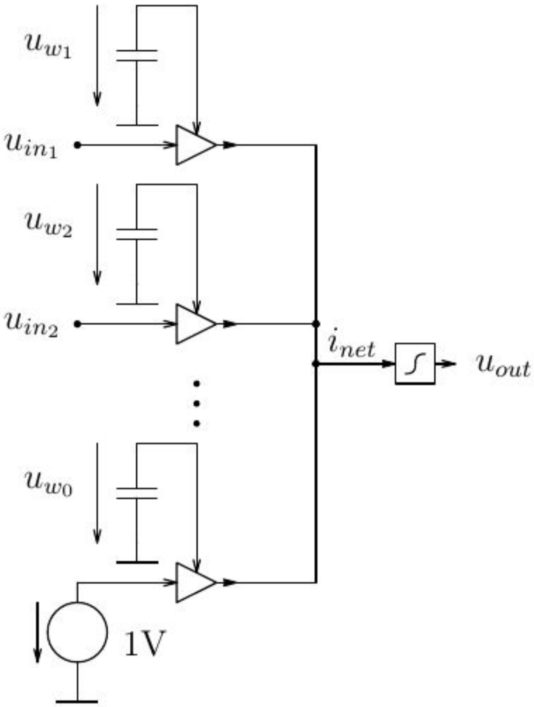


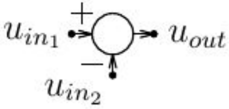




My research

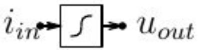
- on-chip design
- supervised learning
- all-analog



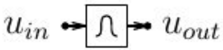


subtractor

multiplier



activation f.



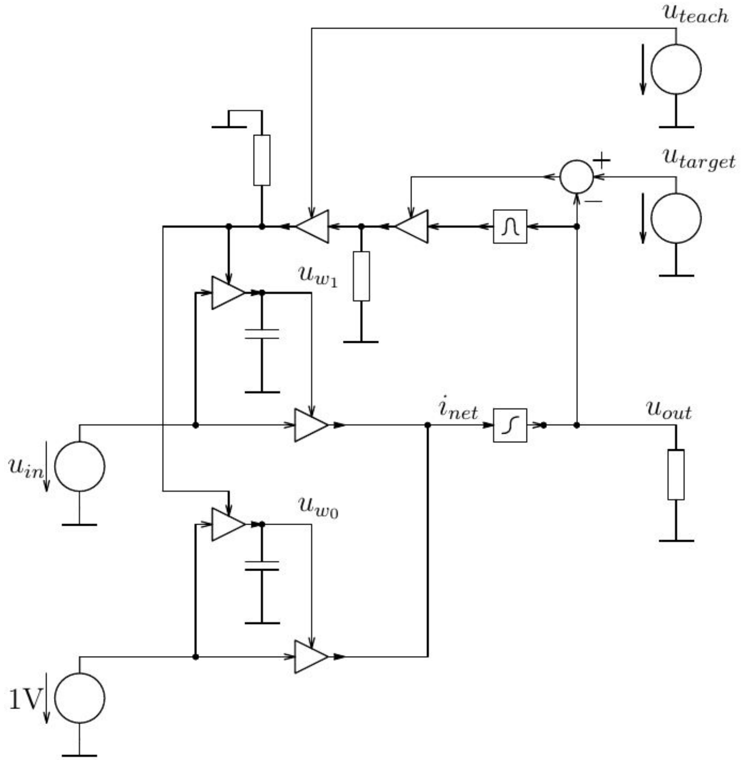
derivation of activation f.

Backpropagation in perceptron

$$u_{w_{t+1}} = u_{w_t} - \eta \frac{\partial E_{total}}{\partial u_{w_t}}$$

$$\frac{\partial E_{total}}{\partial u_w} = \frac{\partial E_{total}}{\partial u_{out}} \frac{\partial u_{out}}{\partial u_{net}} \frac{\partial u_{net}}{\partial u_w}$$

$$\frac{\partial E_{total}}{\partial u_w} = (u_{out} - u_{teacher}) \frac{\partial u_{out}}{\partial u_{net}} u_{in}$$

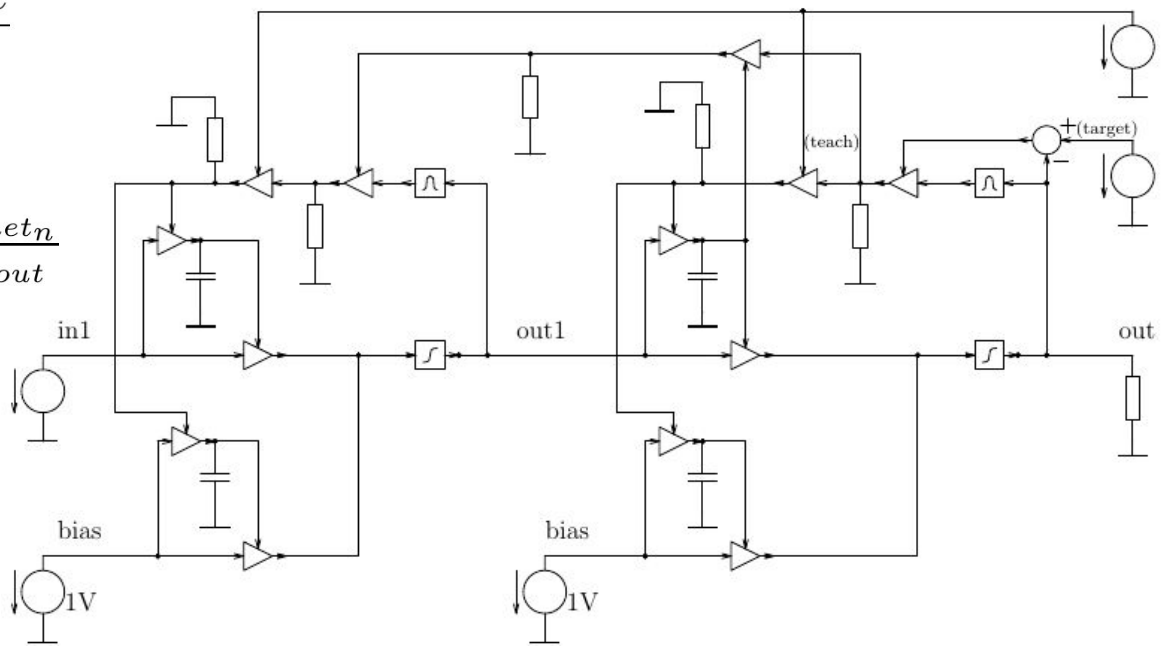


Backpropagation between layers

$$\frac{\partial E_{total}}{\partial u_w} = \frac{\partial E_{total}}{\partial u_{out}} \frac{\partial u_{out}}{\partial u_{net}} \frac{\partial u_{net}}{\partial u_w}$$

$$\frac{\partial E_{total}}{\partial u_{out}} = \sum_n \frac{\partial E_{total}}{\partial u_{out_n}} \frac{\partial u_{out_n}}{\partial u_{net_n}} \frac{\partial u_{net_n}}{\partial u_{out}}$$

$$\frac{\partial u_{net_n}}{\partial u_{out}} = u_{w_n}$$

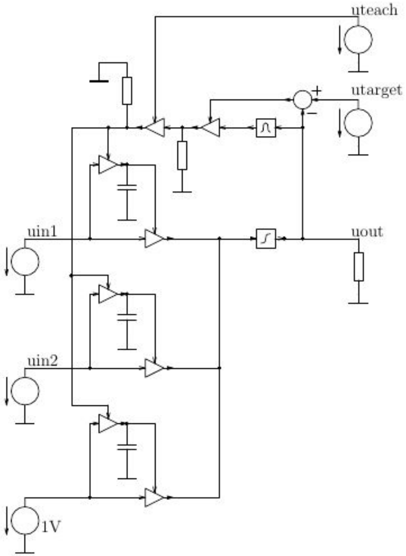


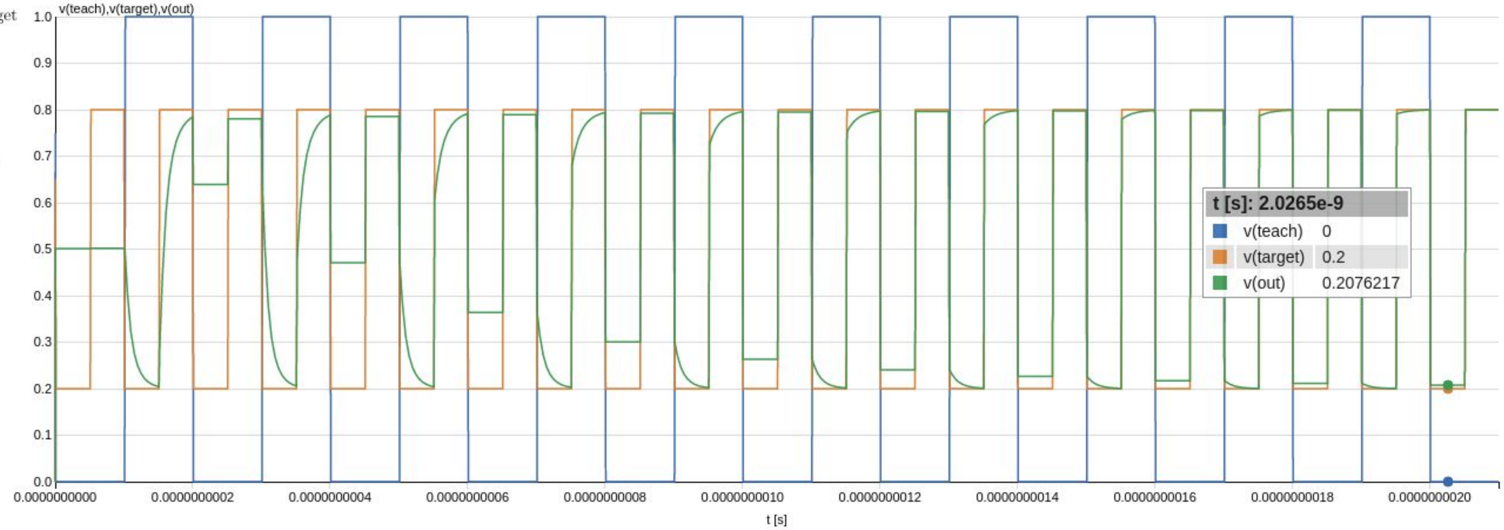
Simulation results

- SPICE

Tab. 1: Dataset č.1

u_{in_1} [V]	u_{in_2} [V]	u_{target} [V]
0.1	0.9	0.2
0.7	0.3	0.8





References

- Krestinskaya, O., & James, A. P. (2018). Analog Backpropagation Learning Circuits for Memristive Crossbar Neural Networks, 0–4.
- Narayanan, P., Fumarola, A., Sanches, L. L., Hosokawa, K., Lewis, S. C., Shelby, R. M., & Burr, G. W. (2017). Toward on-chip acceleration of the backpropagation algorithm using nonvolatile memory. IBM Journal of Research and Development, 61(4), 11:1-11:11.
- Forssell, M. (2014). Hardware Implementation of Artificial Neural Networks Mats Forssell, 1–4.
- Pantazi, A., Woźniak, S., Tuma, T., & Eleftheriou, E. (2016). All-memristive neuromorphic computing with level-tuned neurons. Nanotechnology, 27(35).
- Ali Shafiee, Anirban Nag, Naveen Muralimanohar, Rajeev Balasubramonian, John Paul Strachan, Miao Hu, R. Stanley Williams, and Vivek Srikumar. 2016. ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars. SIGARCH Comput. Archit. News 44, 3 (June 2016), 14-26.

Thank you for your attention

Filip Paulů