TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät für Elektro- und Informationstechnik, Professur für Grundlagen der Elektrotechnik und Elektronik

Winter semester 2024/25

Neural Networks and Memristive Hardware Accelerators

Prof. Dr. phil.nat.habil. Ronald Tetzlaff
Dr.-Ing. Richard Schroedter
Dipl.-Ing. Steffen Seitz
Dr.-Ing. Carsten Knoll

Lecture „Neural Networks and Memristive Hardware Accelerators“
Prof. Dr. phil. nat. habil. Ronald Tetzlaff

Winter semester 2024/25

Lecture 1: Introduction and semester projects

- Dr.-Ing. Richard Schroedter
- Dipl.-Ing. Steffen Seitz

Introduction and semester projects: Topics

- Lecture organization
- Introduction to neural networks
  - Machine Learning motivation, e.g. visual system
  - Why If-then rules cannot describe everything? → NN solves complex problems
- Introduction to memristive computing
  - End of Moore‘s Law
  - 4th circuit element
  - Memristive crossbar arrays performing Matrix-Vector-Multiplication
  - Scaling of memory capacity
  - Memristor examples
- Presentation of semester project topics

Introduction and semester projects : Study goals

- You understand the abilities and limitations of machine learning and you can give some recent examples.
- You can explain why hardware accelerators are required for the execution of neural networks.
- You can give examples how memristive neuromorphic devices can emulate the brain’s neural networks.
- You know how to perform your semester project and your group found a topic.

Lecturers

— **Prof. Ronald Tetzlaff** (Head of the course)

Chair of Fundamentals of Electrical Engineering;
wide research area of chaotic systems, memcomputing and artificial intelligence

— **Dr. Richard Schroedter** (Memristive Computing)

Post-Doc researching on bio-inspired memristive in-memory computing and neural network accelerators

— **Dipl.-Ing. Steffen Seitz** (Machine Learning)

PhD student working on neural networks for fault detection and explainable decisions

— **Dr.-Ing. Carsten Knoll** (Python Course)

Post-Doc researching on nonlinear control, machine learning, semantic technology

Lecture Schedule, Python Course, Semester project, Exams

Lecture Tuesdays 9:20am in TOE/317	Python Fridays 9:20am online	Semester project homework
1 – Introduction	Python Course 1	
2 – Machine Learning	Python Course 2	Project
3 – Deep Neural Networks	Python Course 3	Project
4 – Convolutional NN	Python Course 4	Project
5 – PyTorch tutorial	Python Course 5	Project
6 – Sequential NN	Python Course 6	Project
7 – Neural Networks Hardware	Python Course 7	Project
8 – Memristor circuit theory	Python Course 8	Project
9 – SPICE tutorial	Python Course 9	Project
10 – Memristor physics	Python Course 10	Project
11 – Crossbar and logic circuits	Python Course 11	Project
12 –Mapping and performance	Python Course 12	Project
13 – Memristive neurons	Python Course 13	Project
14 – Spiking Neural Networks	**Semester**	Project
Exams	**Examination Period**	Presentation

12 Lectures with tutorials on PyTorch and SPICE

Bring your laptop to the tutorials

Python Course → OPAL link

Parallel to lecture to learn basic python given by Dr. Carsten Knoll (online)

Semester Project (1/5 of module grade)

Group project of **4** students on different topics, form groups, **choose** favorite topic, **selection** done in 2nd lecture, then **contact** your supervisor

→ Submit your **implementation code** before the exam to supervisor

→ **Presentation** (10 min)

Examinations (4/5 of module grade)

Written exam (180 min) on all topics of the lecture

Timetable winter semester 2024/25

Week	Date	Time	Room	Lecture/Tutorial/Presentation
42	15.10.2024	9:20-10:50	TOE/317	Lecture 1: Introduction
43	22.10.2024	9:20-10:50	TOE/317	Lecture 2: Machine Learning
44	29.10.2024	9:20-10:50	TOE/317	Lecture 3: Deep Neural Networks
45	05.11.2024	9:20-10:50	TOE/317	Lecture 4: Convolutional Neural Networks
46	12.11.2024	9:20-10:50	TOE/317	Lecture 5: PyTorch tutorial
47	19.11.2024	9:20-10:50	TOE/317	Lecture 6: Sequential Neural Networks
48	26.11.2024	9:20-10:50	TOE/317	Lecture 7: Neural network hardware accelerators
49	03.12.2024	9:20-10:50	TOE/317	Lecture 8: Memristors circuit theory
50	10.12.2024	9:20-10:50	TOE/317	Lecture 9: SPICE tutorial
51	17.12.2024	9:20-10:50	TOE/317	Lecture 10: Memristor physics and modeling
52	24.12.2024			holiday
1	31.12.2024			holiday
2	07.01.2025	9:20-10:50	TOE/317	Lecture 11: Memristive crossbars and logic circuits
3	14.01.2025	9:20-10:50	TOE/317	Lecture 12: Mapping and performance
4	21.01.2025	9:20-10:50	TOE/317	Lecture 13: Memristive neurons
5	28.01.2025	9:20-10:50	TOE/317	Lecture 14: Spiking Neural Networks
6	04.02.2025	9:20-10:50	TOE/317	* Semester Project Presentation
7-9	Written exam			

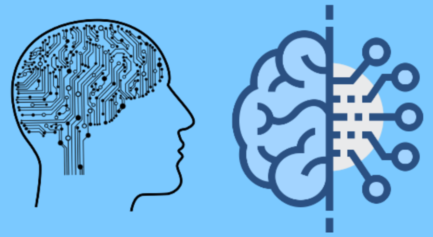
Date	Time	Room	Python Course	
18.10.2024	9:20-10:50	Online	Introduction	mandatory
25.10.2024	9:20-10:50	Online	Files and Functions	
01.11.2024	9:20-10:50	Online	Numerical computation	
08.11.2024	9:20-10:50	Online	2D Visualization	
15.11.2024	9:20-10:50	Online	Object orientation	project specific
22.11.2024	9:20-10:50	Online	Data processing, analysis	
29.11.2024	9:20-10:50	Online	Advanced programming	
06.12.2024	9:20-10:50	Online	Performance optimization	
13.12.2024	9:20-10:50	Online	Symbolic computation	
20.12.2024	9:20-10:50	Online	3D Visualization	
27.12.2024	9:20-10:50	Online	holiday	
03.01.2025	9:20-10:50	Online	holiday	
10.01.2025	9:20-10:50	Online	GUI part 1	
17.01.2025	9:20-10:50	Online	GUI part 2	
24.01.2025	9:20-10:50	Online	Comm. with ext. hardware	

Introduction to neural networks

Neural Networks in the context of Machine Learning

Artificial Intelligence

Technique that enables computers to mimic human behavior

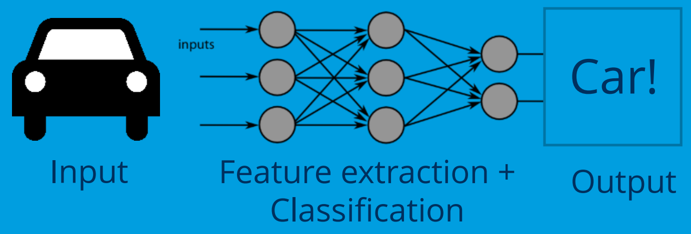


Machine Learning

Ability to learn without explicitly being programmed

Deep Learning (Neural networks)

Extract patterns from without expert assistance



From human intelligence to artificial intelligence

What is **Intelligence**?

- Work definition: “A system is called intelligent when it can solve problems independently and efficiently.”
- **Artificial intelligence (AI)** = intelligent behavior in machines

In general, AI can be separated into:

Weak AI	**Strong AI**
Goal: Solve a complex task that was **only solvable** by humans.	Goal: Build a machine that works **just like** a human or even better.

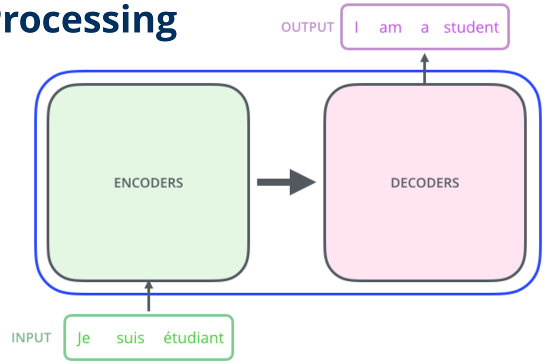
Mainzer, Klaus. *Artificial Intelligence - When Do Machines Take Over?* Springer Berlin Heidelberg, 2020

AI in the Media…

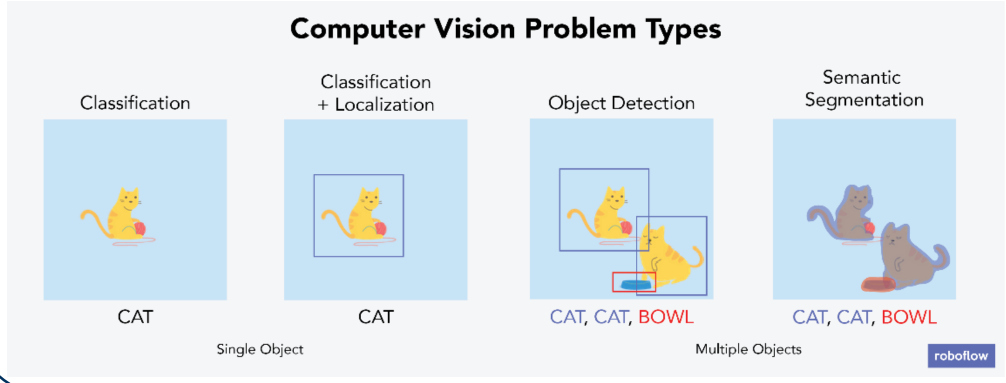
Thats
horrible…

Machine Learning - (some) Fields of Application

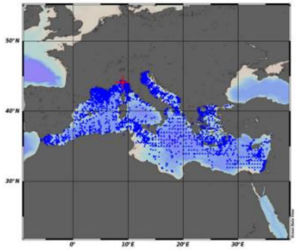
Natural Language Processing

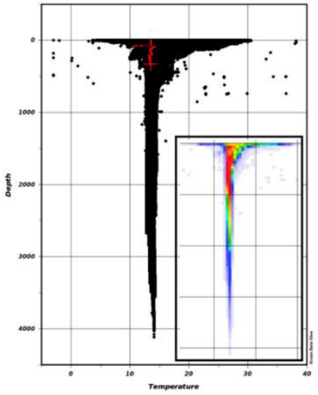


Computer Vision



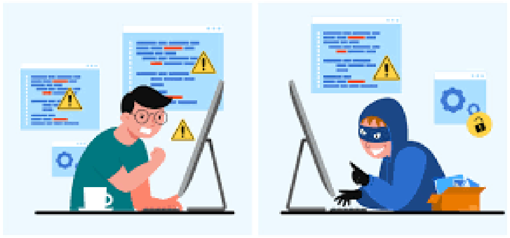
Anomaly Detection

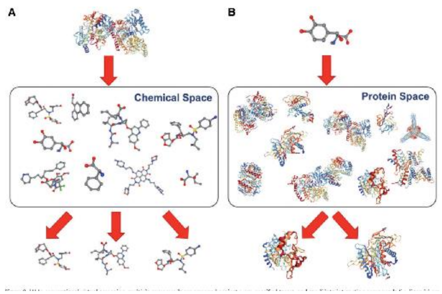




Mieruch et al. *Frontiers in Marine Science*, 8, 2021

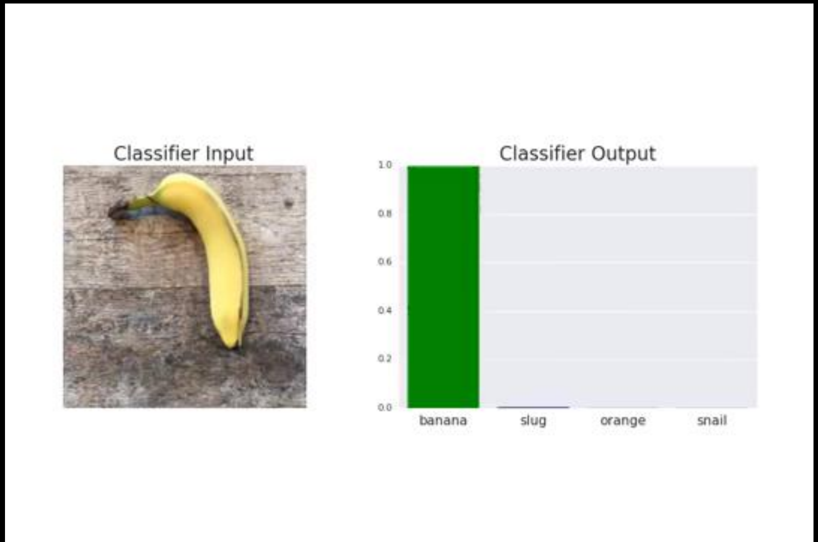
Fraud Detection, Drug Development....





Weak AI
Applications

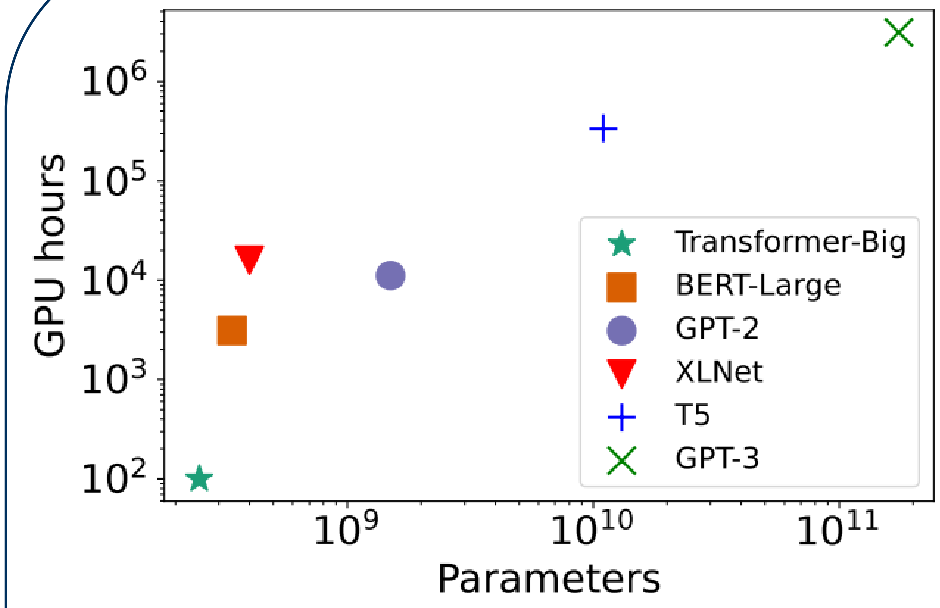
There are controversial aspects of modern AI!



https://www.youtube.com/watch?v=cQ54GDm1eL0

https://www.youtube.com/watch?v=Khuj4ASldmU

Neural Networks require potent (and costly) hardware!



LightSeq2: Accelerated Training for Transformer-based Models on GPUs

Xiaohui Wang[1], Yang Wei[1], Ying Xiong[1], Guyue Huang[2], Xian Qian[1], Yufei Ding[2], Mingxuan Wang[1], Lei Li[2*]

[Wang et al., SC22, 2022]

Tesla's "Dojo" Supercomputer used to compute self-driving car software uses **>7600 graphic cards**

But why is it possible to **run** these applications on a **PI** or even a **mobil** device ? → Soon ☺

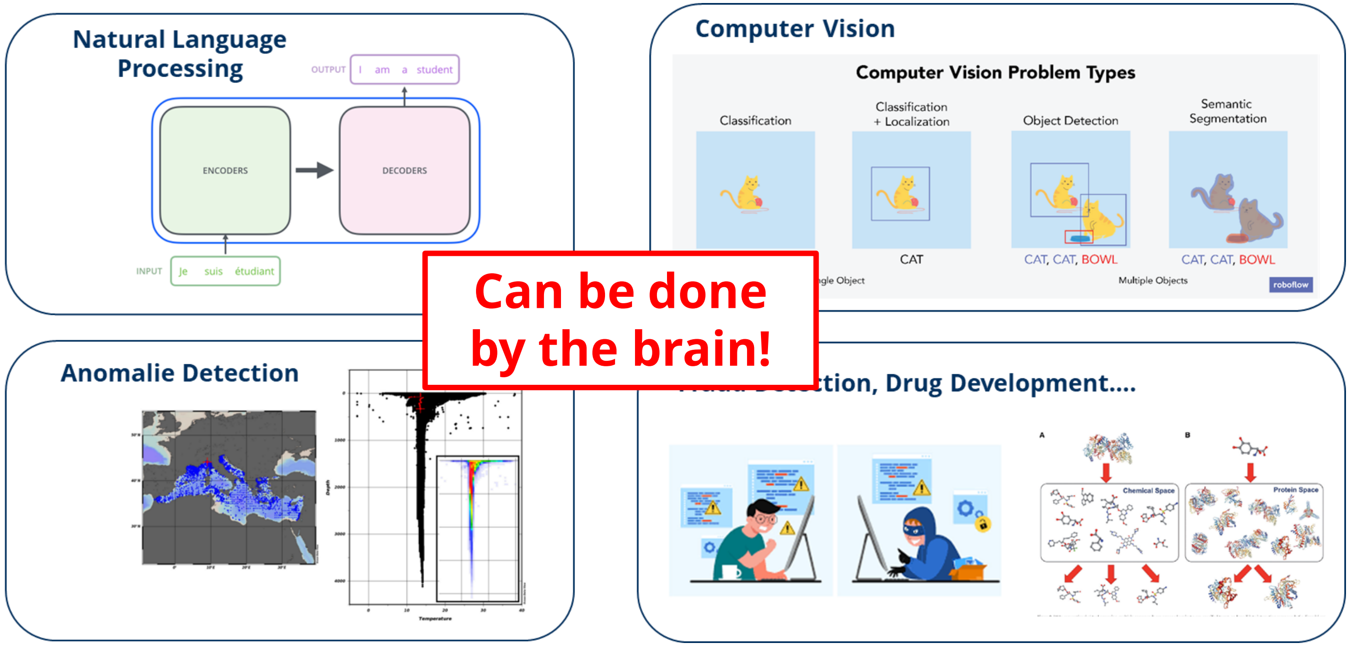
Brain Inspired Computing

Can we find more **suitable computing hardware**? → The human brain is extremely potent and **efficient**

It only needs **15-20W** to operate while performing these extremely difficult tasks!

Develop computer architectures that work similar to the brain!

→ **Neuromorphic- & Memcomputing**

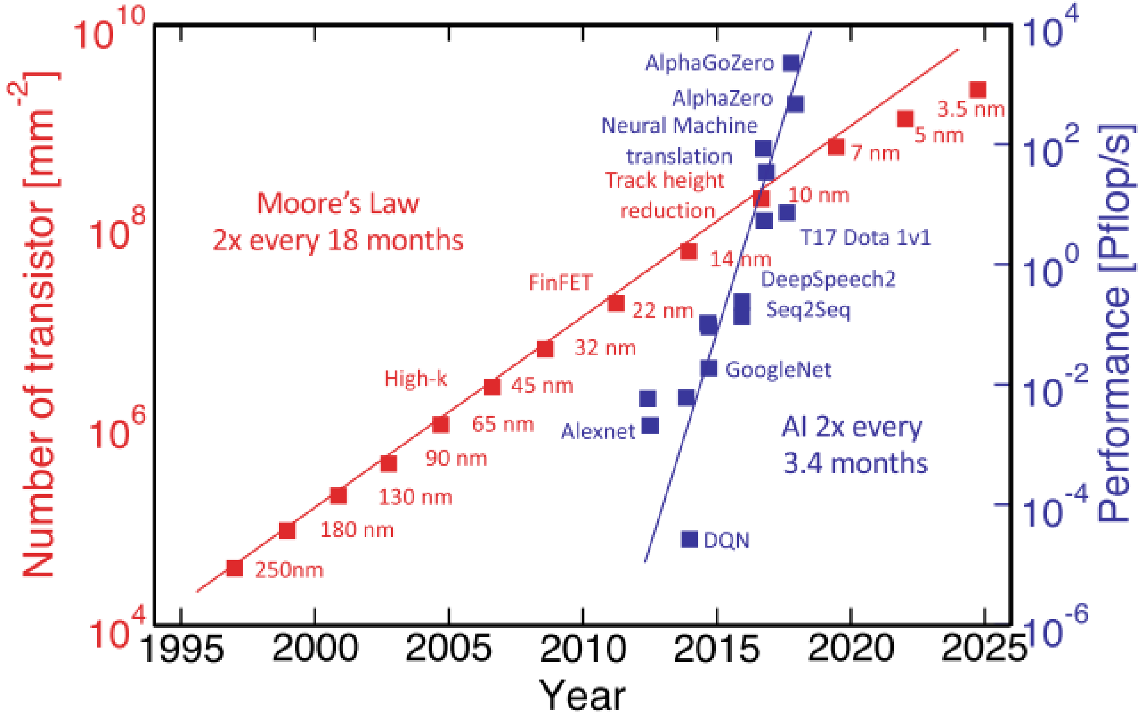


Introduction to memristive computing

Computing performance for AI

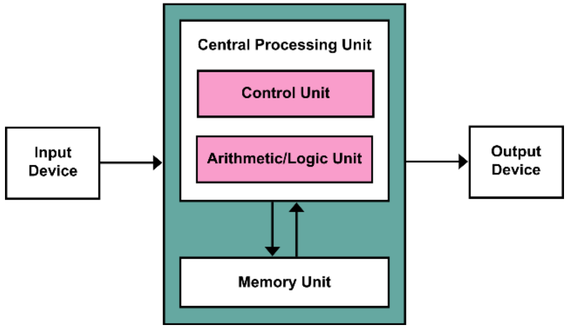
- Moore's Law (red): no. of transistors per **doubles every 18 months**
  - Area down scaling by factor 0.7 (area reduces by 50%)
  - End of Dennard scaling, due to physical limits (leakages, threshold voltage)
    → No energy saving through miniaturization
- Required AI performance: **doubles every 3.4 month**

Pflops/s = Peta (10^{15}) floating point operations



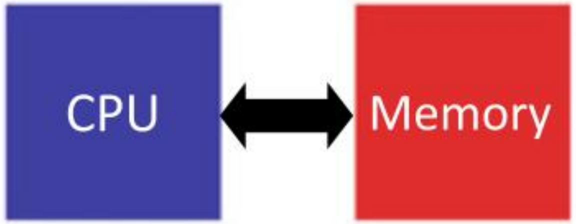
Von-Neumann bottleneck

— Von-Neumann bottleneck: large data transfer between separated **computing** (CPU or GPU) and **memory** (RAM or hard disc)



von Neumann (1945)

— Energy dissipation of transistors reached **Landauer limit** (thermal fluctuation)
→ cannot be overcome with shrinking modern Complementary-Metal-Oxide-Semiconductor (**CMOS**) technologies

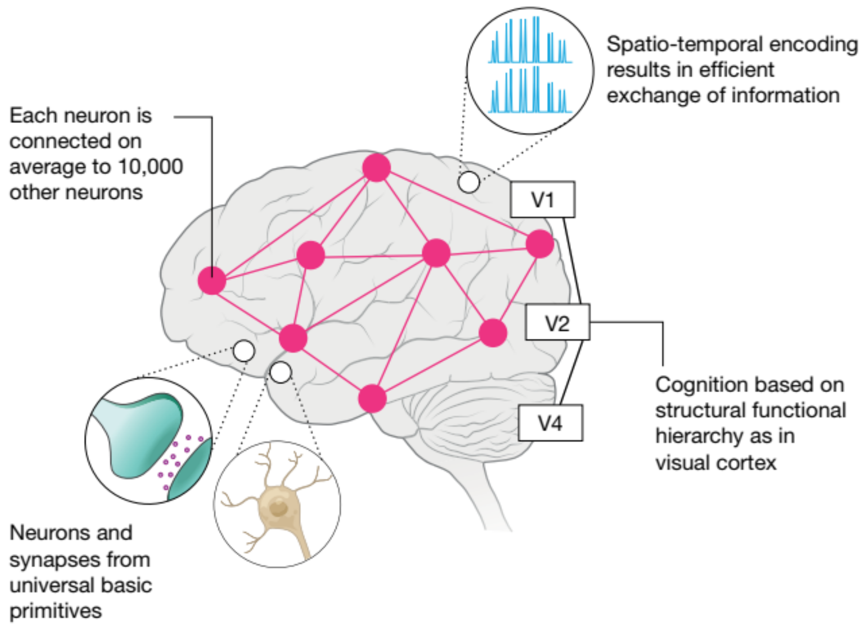


von Neumann bottleneck

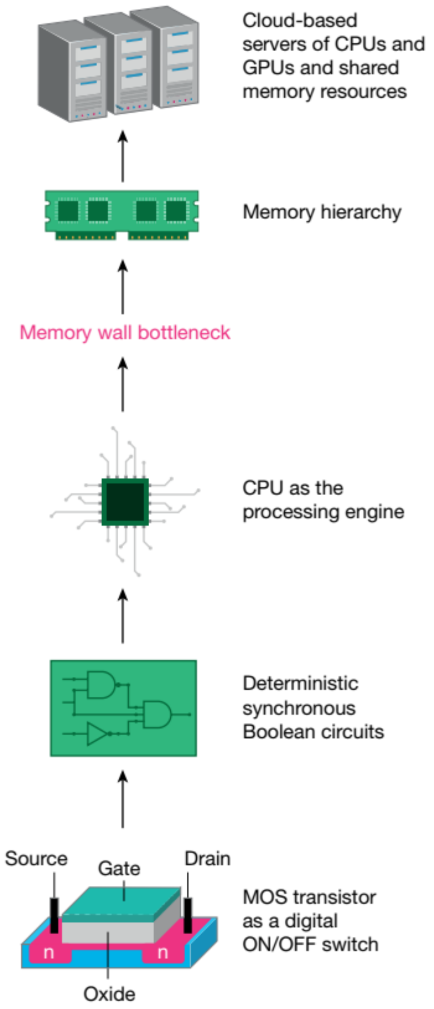
→ **Solution „Computing-in-memory“ (CIM)**

In-memory computing

Human Intelligence versus **Systems with Artificial Intelligence**



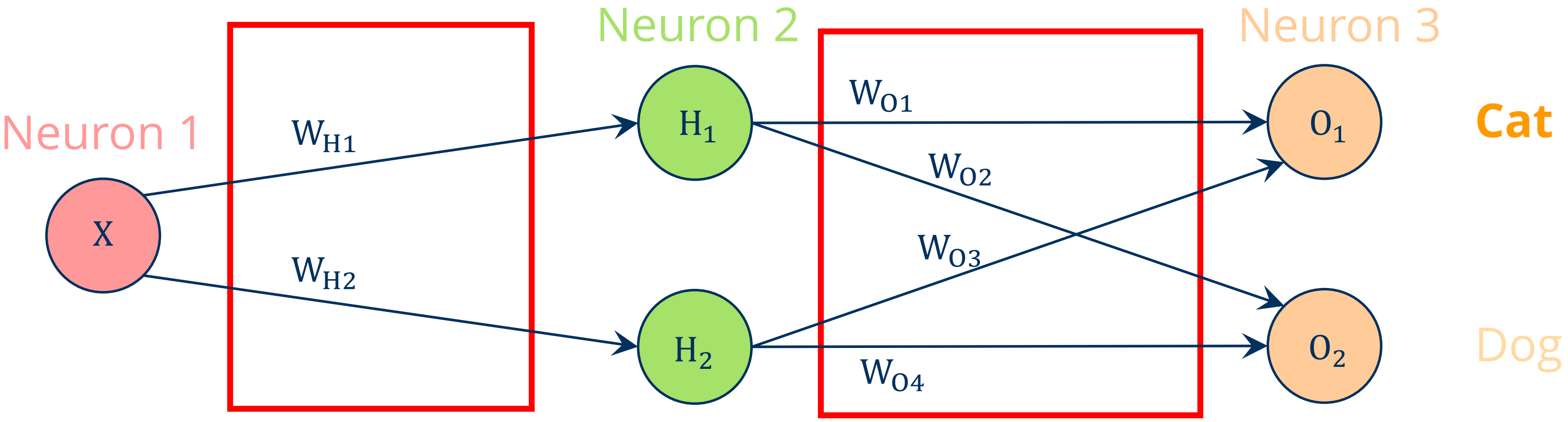
	Human	AI systems
Structure	Network of ~86 billion of neurons; 3D interconnected through ~60 trillions of synapses	Nanometer silicon technology, von-Neumann bottleneck; only 2D connected
Abilities	Simultaneous recognition, reasoning, control and movement, creativity	ChatGPT, image/ speech recognition, gaming (Go, Chess) etc.
Energy-consump-tion	~20 W	Standard PC performing only recognition among 1000 different kinds of objects expends about 250 W
Operation function-ality	Spike-based spatio-temporal processing; event-driven, sparse and inherently stochastic	Deterministic Boolean (digital) computation

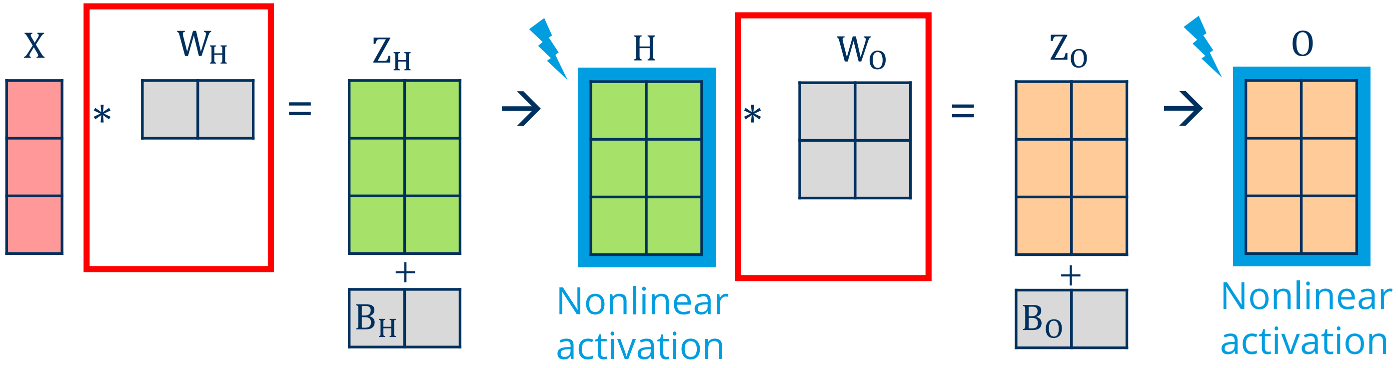


Roy et al. *Nature*, 575 (7784) 2019

How to accelerate Neural Network computation?

Neuron → Multiplication with Weights W → Neuron fires → and so on…





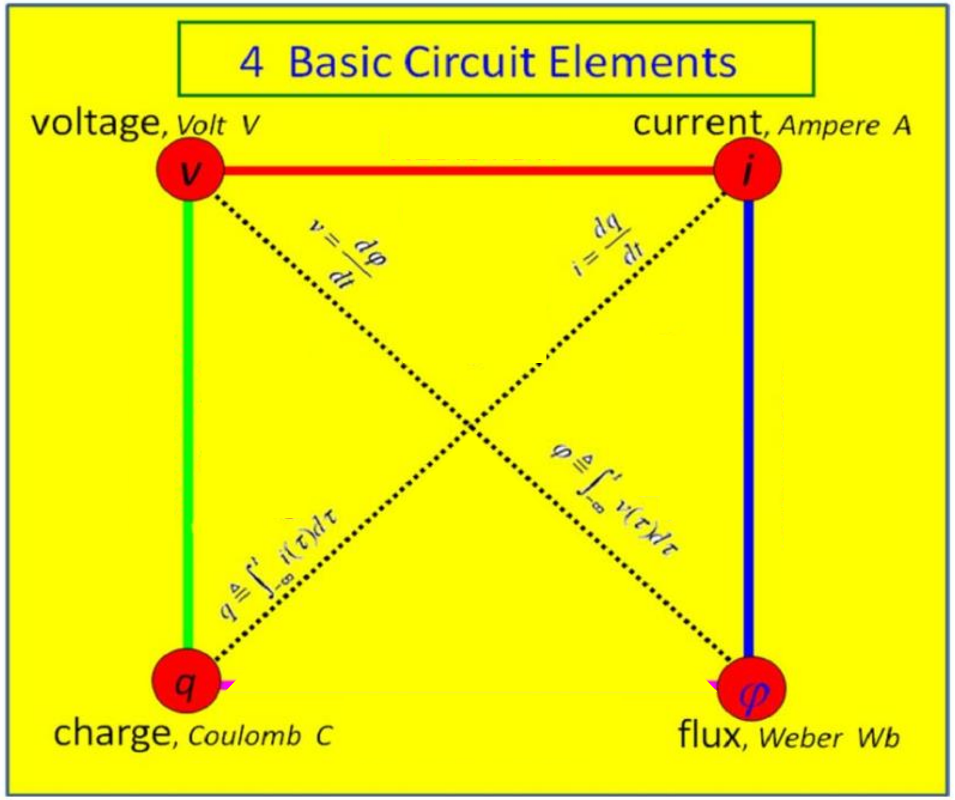
Neural Networks perform massively Matrix-Vector-Multiplications.

Memory Resistor = Memristor

Theory proposed by Leon Chua in 1971 → A resistor that stores non-volatile information.

Which basic circuit elements exist?

1. Charge / Voltage: Capacitor C = $\frac{\mathrm{d}q}{\mathrm{d}v}$
2. Current / Magn. Flux: Inductor L = $\frac{\mathrm{d}\varphi_{magn}}{\mathrm{d}i}$
3. Voltage / Current: Resistor R = $\frac{\mathrm{d}v}{\mathrm{d}i}$
4. Charge / Flux: Memristor M = $\frac{\mathrm{d}\varphi}{\mathrm{d}q}$ with $\varphi \neq \varphi_{magn}$

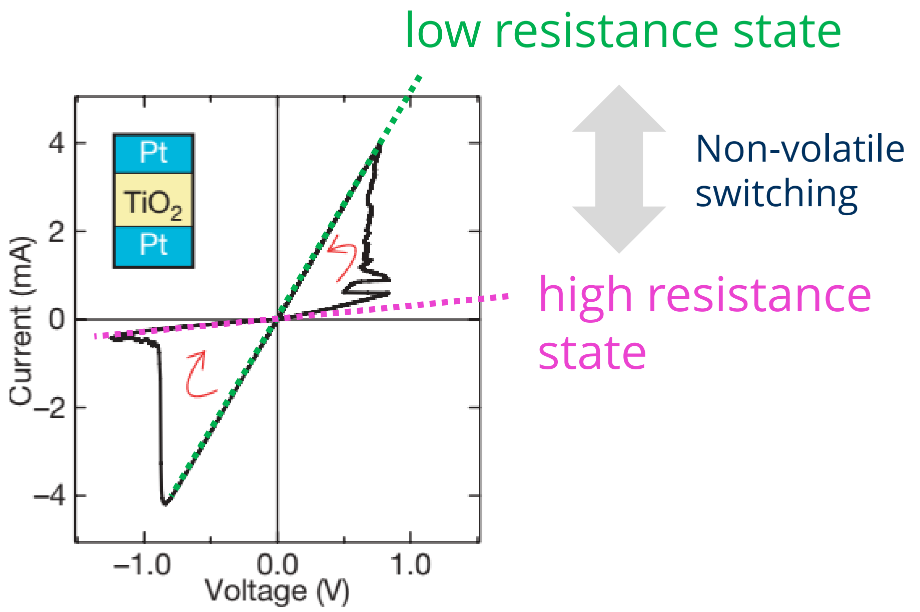


Memory by time integrals: $\varphi = \int v \,\mathrm{d}t$
$q = \int i \,\mathrm{d}t$

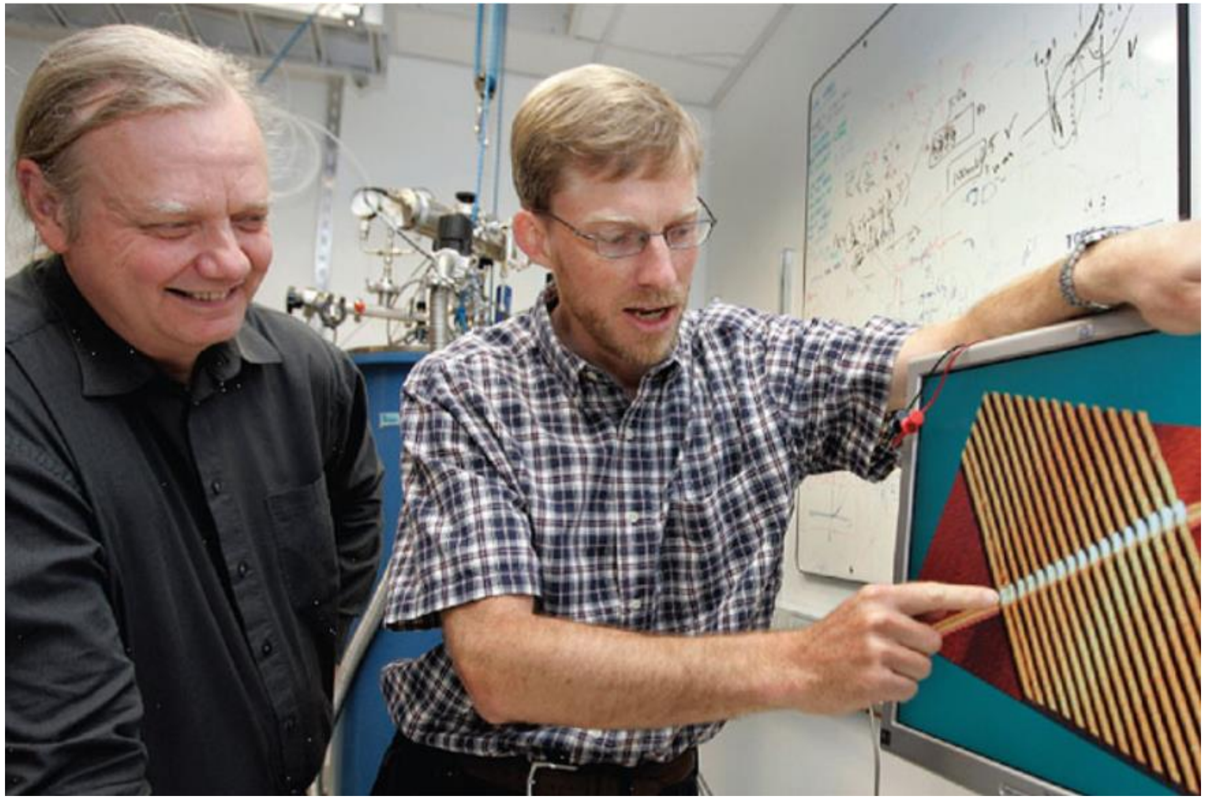
Memristive behavior experimentally shown

→ 2007 HP Labs show the memristor behavior using 5 nm titanium oxide layer (TiO_2).

Oxygen ions are moving within the device switching from high resistance to low resistance.



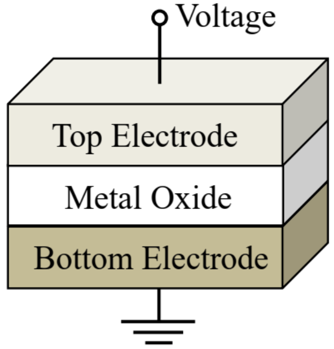
Strukov et al. "The Missing Memristor Found." *Nature*, vol. 453, no. 7191, 7191, 2008, pp. 80–83



R. Stanley Williams and Duncan Stewart at HP Labs

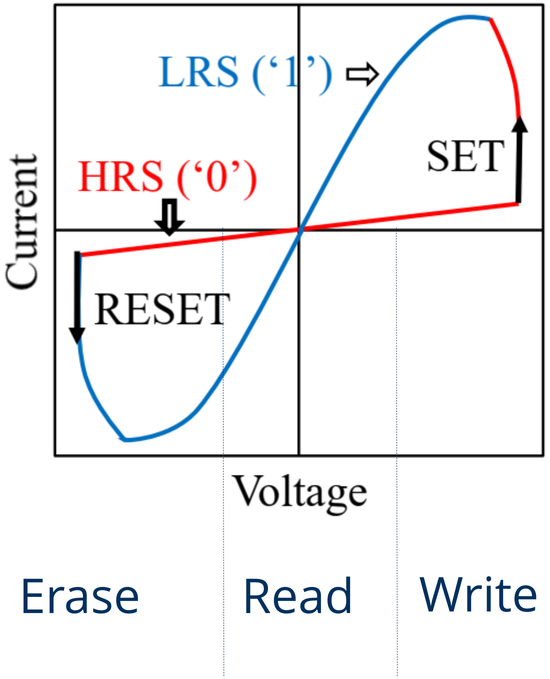
Crossbar-Array performing Matrix-Vector-Multplications

Memristor cell

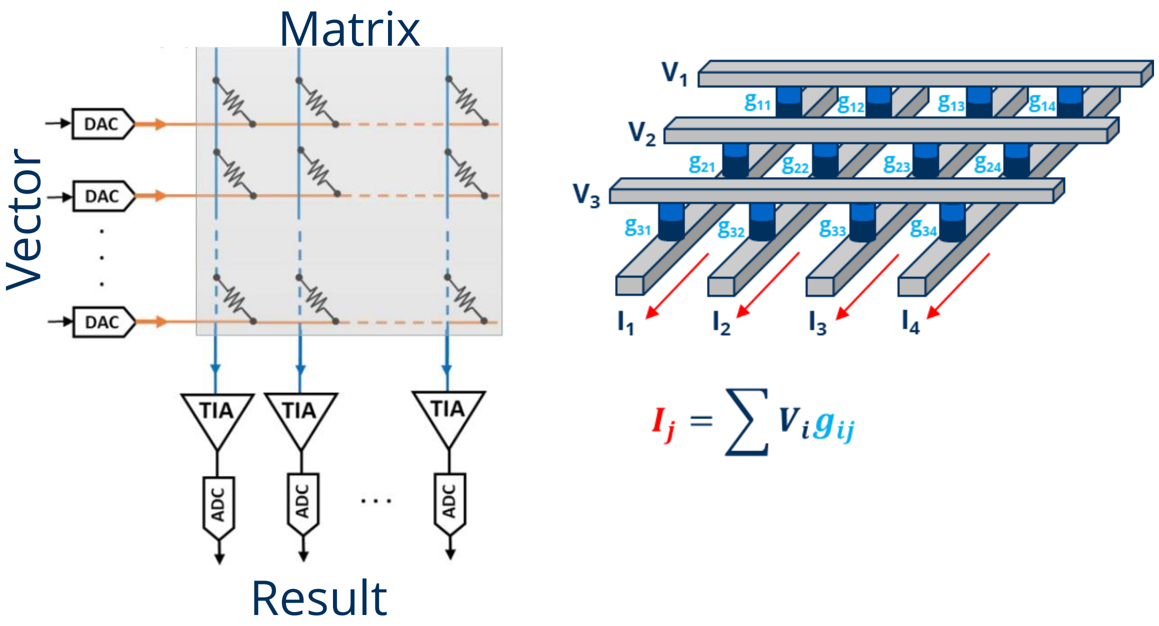


Metal-Oxide-Metal stack in nanometer stale = resistive random access memory (**ReRAM**)

I-V Relation



Crossbar Array



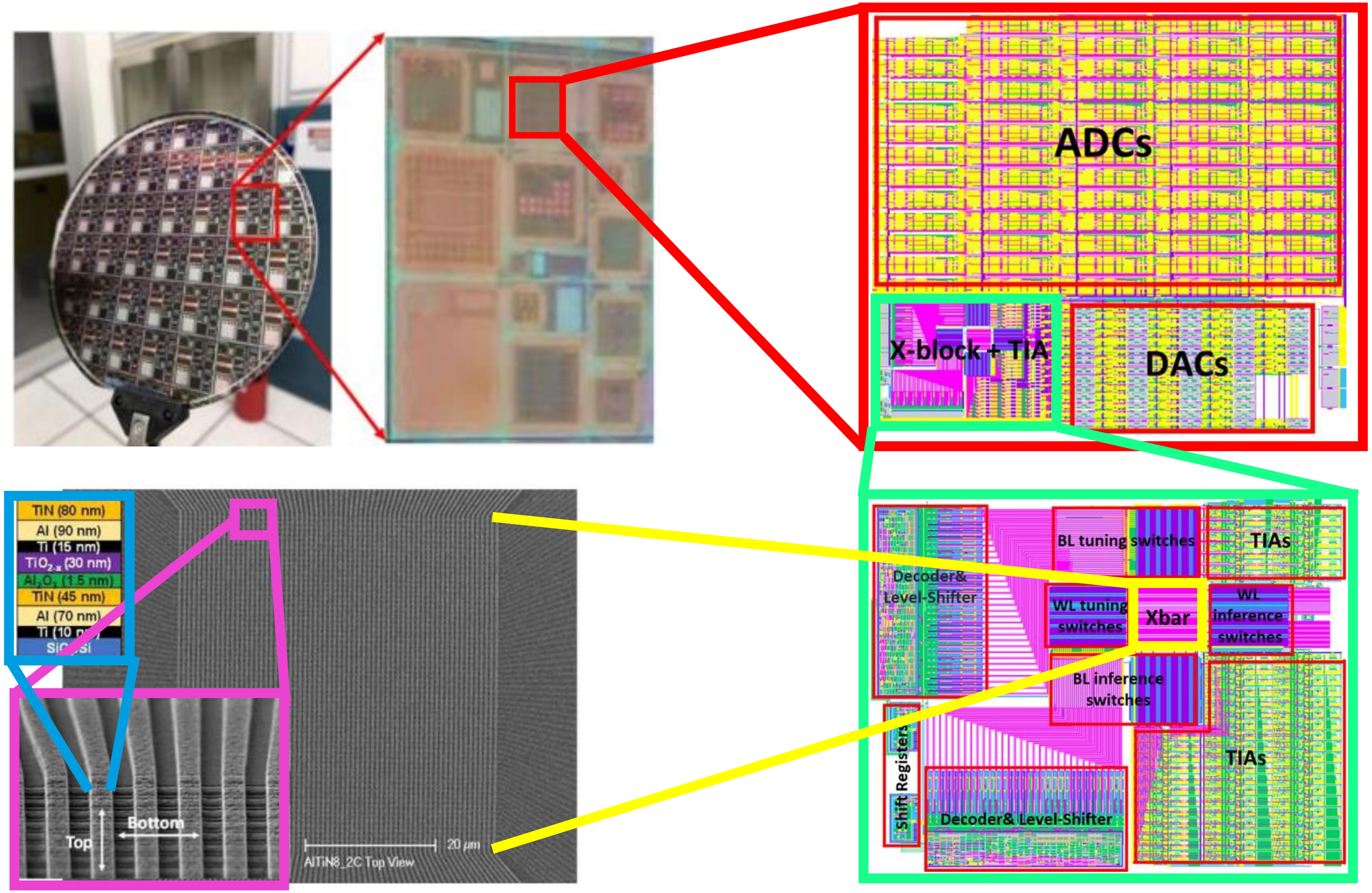
$$I_j = \sum v_i g_{ij}$$

Memristor stores weights in the conductances
→ Analog Matrix-Vector-Multiplication in one step!

Video: How memristive hardware accelerates neural networks

Future Computers Will Be Radically Different https://www.youtube.com/watch?v=GVsUOuSjvcg

Memristive crossbar prototypes → Let‘s scale down ☺

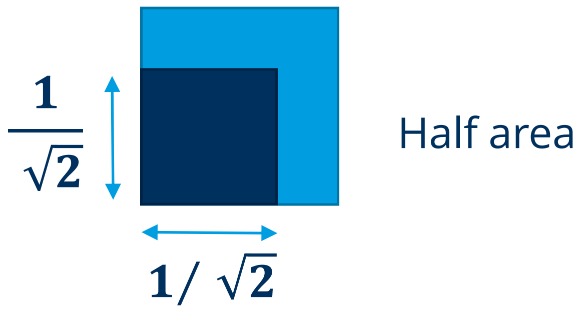


Mahmoodi, et al. *IEEE (IEDM)*, 2019

Fahimi. UC Santa Barbara, Disseration, 2021

How to reach large scale memory capacity?

1) Two-dimensional cell down-scale

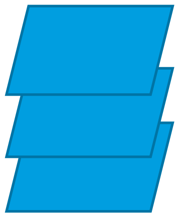


→ Process challenge **in fabrication**

2) Multiple logical bits per cell

→ Endurance, retention challenge **on device level**

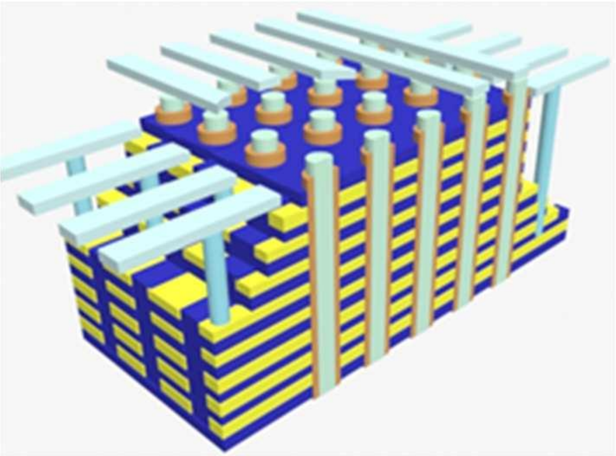
3) 3D stacked layers



→ 3D integration challenge **in design**

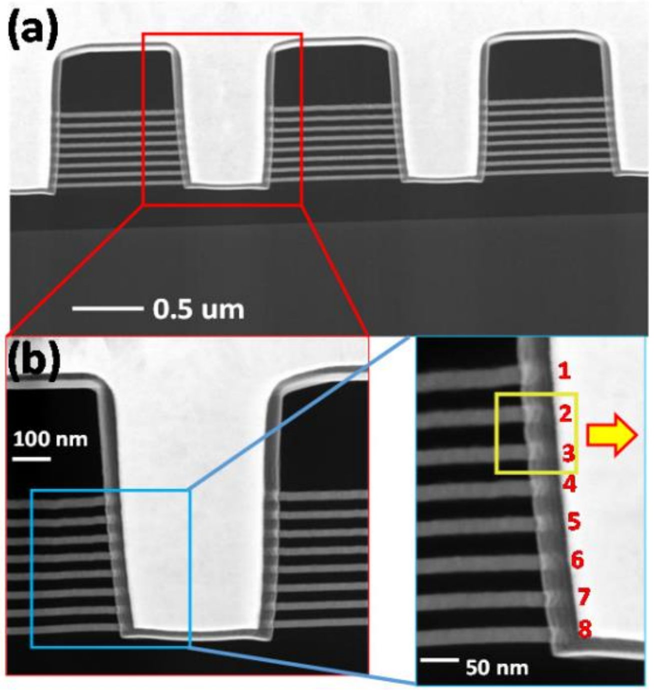
3D integration example

- 8 layers of vertical RRAM with HfO_2/TiOx cells
- Forming free, self-selective cells with high endurance
- Area switching mechanism (resistance scales with area)

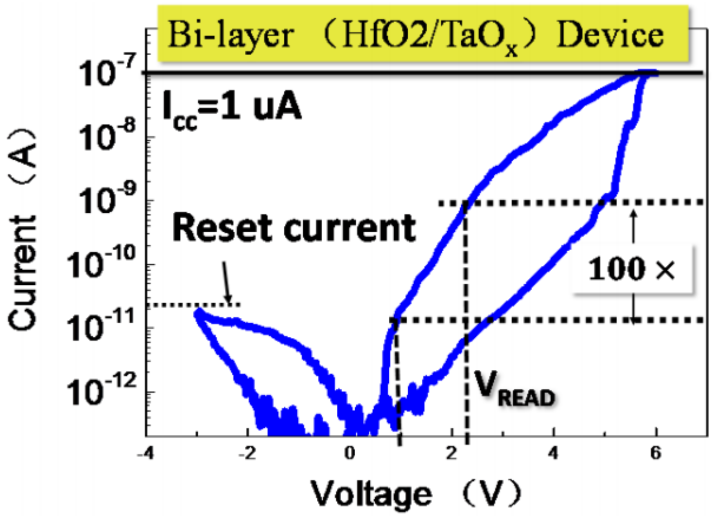


metrics

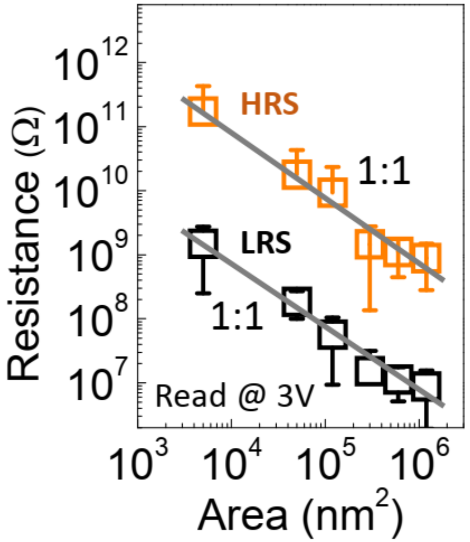
	3D VRRAM (This work)
Non-volatile	yes
Cell size (F^2)	4/n (BiCS)
R/W Latency	300 ns /100ns
Endurance	10^7
Program energy	<1 μA
High voltage required	4~9V
Data retention	3 years
Scalability	5 nm



IV characteristic



area scaling



Luo et al. *IEDM*, 2017

Motivation for this lecture

For emulating the brain's neural networks with neuromorphic chips we need understanding of

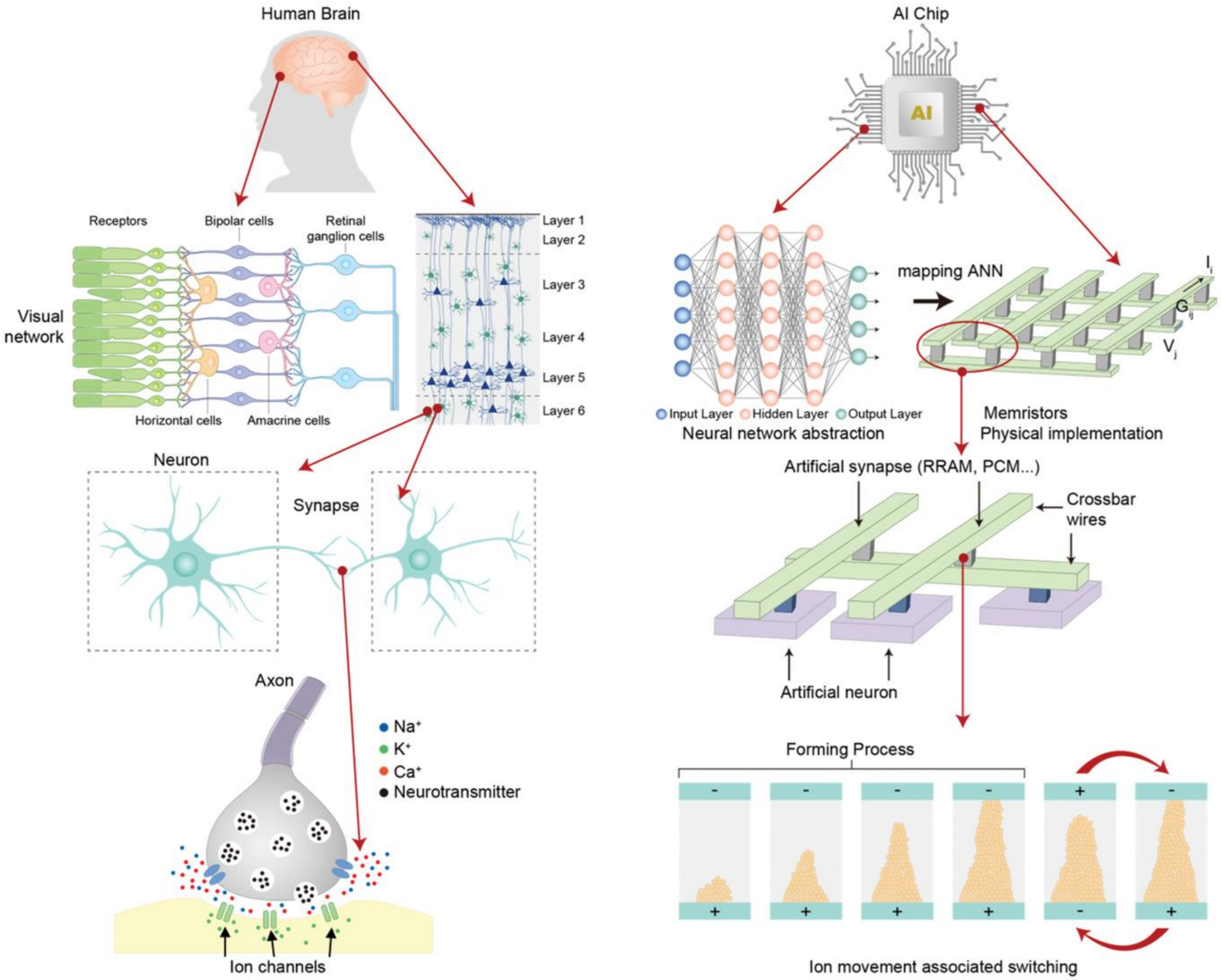
— Neural networks architectures (Artifical NN, Spiking NN)

— Memristive device physics and modeling

— Crossbars and neurons

— System architecture and mapping

→ Basics on neural networks and memristors are covered in this course.



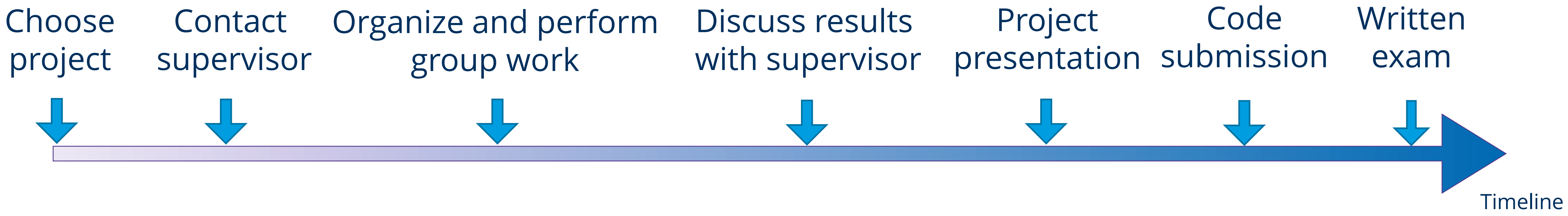
Tang et al, *Adv. Mat.* 2019

Semester project

Semester project

Your task will be a completion of a research project in a group work:

- **Analyze** the dataset and perform **literature search**
- Identify the **main task** for your **research question**
- Organize the **group work** and **plan tasks** to achieve the main goal
- **Conduct** and properly document the research; in case of problems contact your supervisor
- Present your **results** to your colleagues in **10 minutes** (max. 10 slides) on 04.02.2025 at 9:20 in TOE/317
- Write a **README file** and send the **commented code** to your supervisor

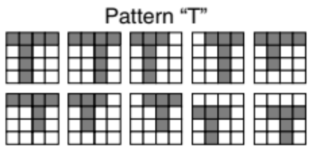


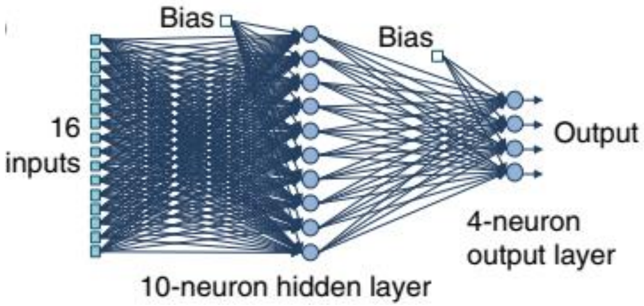
Choose one of 12 semester project topics

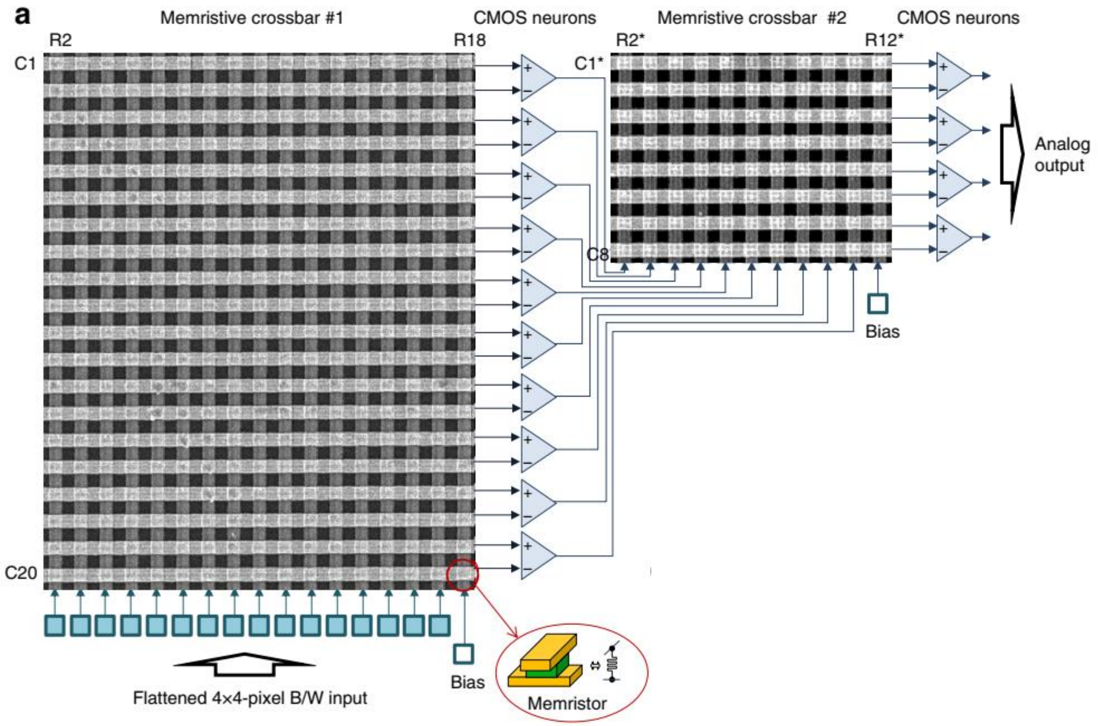
- **Pick your favorite** projects in **groups of 4 students**
- **Sign up** in OPAL with your group → Group A to Group T
- **Selection** will be done in at the end of **lecture 2**
- **Contact** your **supervisor** for a meeting (one email per group)

Project 1: Simulation of a neural network with memristor crossbar using MemTorch

- **Topic**: Pattern recognition of four letters *A,T,V,X* using small 1R memristor crossbars and software neurons
- **Task**: Implement pattern recognition with a neural network in PyTorch and map the weights to the memristor crossbar using MemTorch library
- **Challenge**: Correct mapping of weights and software neurons and handle the memristor non-idealities
- **Supervisor**: Dr. Richard Schroedter, richard.schroedter@tu-dresden.de



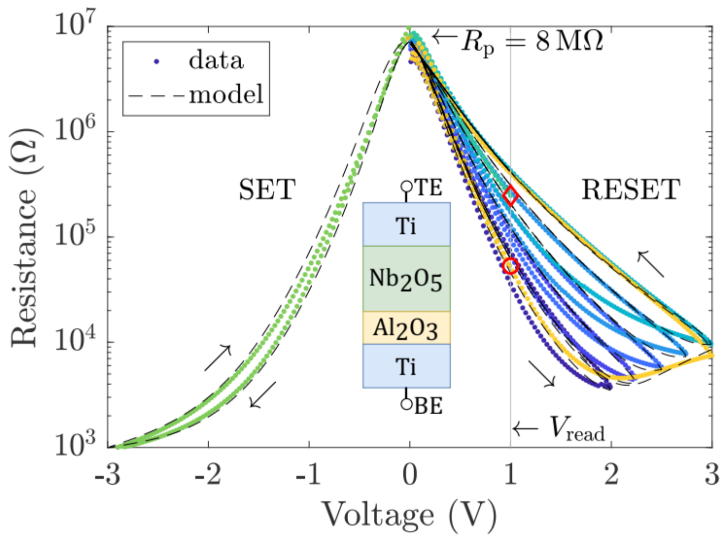


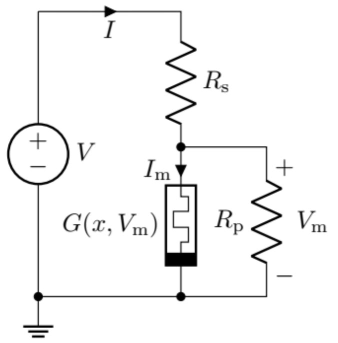


Bayat et al. *Nature Communications*, vol. 9, no. 1, 1, June 2018, p. 2331. https://doi.org/10.1038/s41467-018-04482-4.

Project 2: Fitting compact memristor model to measurement data

- **Topic**: Compact model development by fitting the I-V and pulse measurement data set to a given set of memristor equations
- **Task**: Implement the compact memristor model given as SPICE code and fit optimization for given measurement data using Python
- **Challenge**: Solving the compact memristor ODE and run the optimization including both, I-V and pulse measurement data
- **Supervisor**: Dr. Richard Schroedter, richard.schroedter@tu-dresden.de



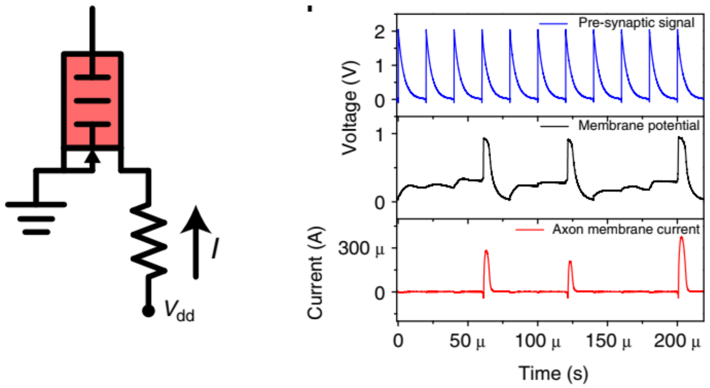


Schroedter, et al. "SPICE Compact Model for an Analog Switching Niobium Oxide Memristor." *IEEE MOCAST*, 2022, https://doi.org/10.1109/MOCAST54814.2022.9837726.

Project 3: Neuro-transistor simulation with memristor-based memcapacitor

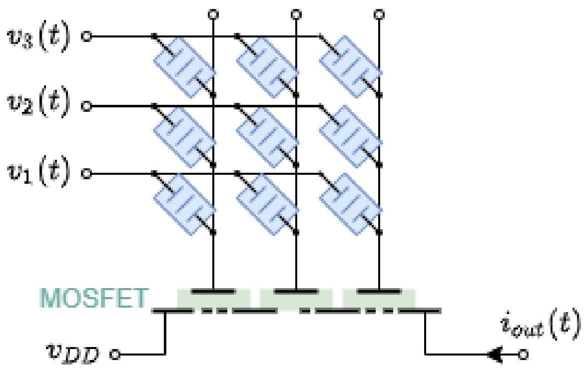
- **Topic**: Realization of the neuron functionality with a pseudo-memcapacitive transistor using LTSPICE
- **Task**: Implement the memristor-based memcapacitor at the gate of a transistor in LTSPICE and find a pulse input setting which realizes the neuron firing after integration through the transistor. Extend to a multi-channel neurotransistor.
- **Challenge**: Analyze the parameter range of neuron firing functionality for different pulse schemes and capacitances
- **Supervisor**: Dr. Richard Schroedter, richard.schroedter@tu-dresden.de

Neurotransistor

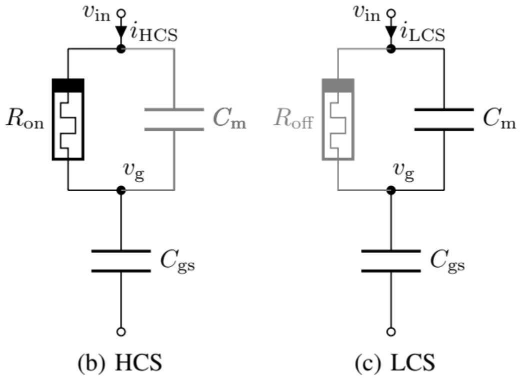


Wang et al. "Capacitive Neural Network with Neuro-Transistors." *Nature Communications*, vol. 9, no. 1, 1, 2018, https://doi.org/10.1038/s41467-018-05677-5.

Multi-channel neurotransistor

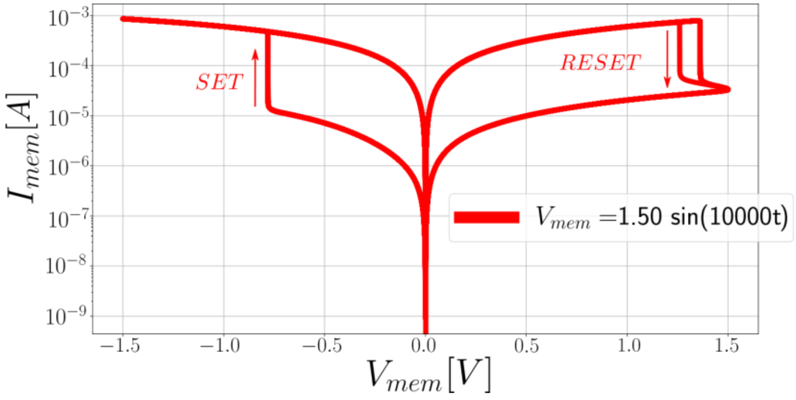


memristor-based memcapacitor:

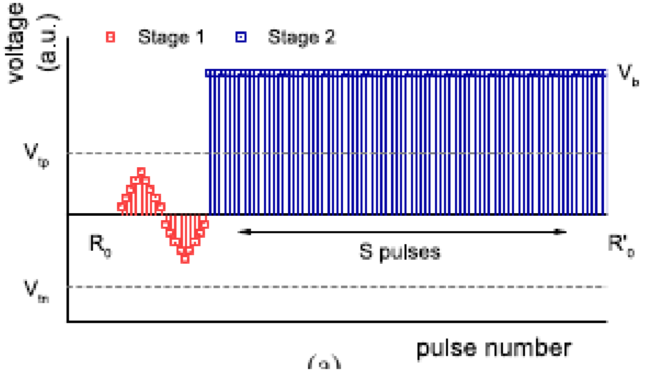


Project 4: Algorithms of characterization routines of memristors

- **Topic**: Memristive device characterization
- **Task**: Writing characterization sequence in Python for device-characterization.
- **Challenge**: In neural network applications, memristors require high precision programmability to guarantee uniform and accurate performance across a large number of networks.

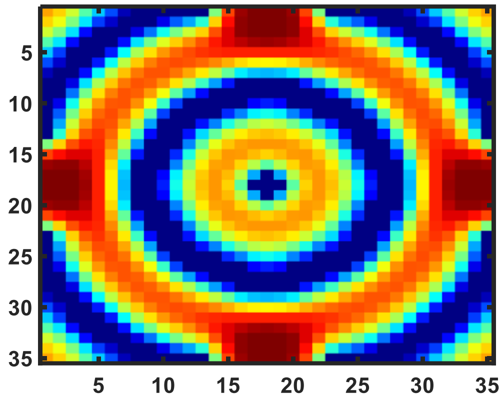


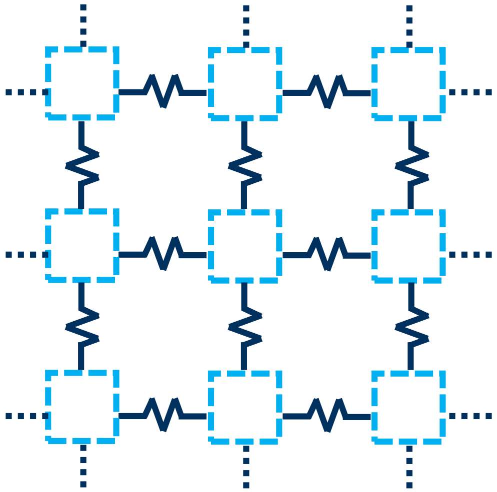
- **Supervisor**: Dr. Dimitris Prousalis, dimitrios.prousalis@tu-dresden.de

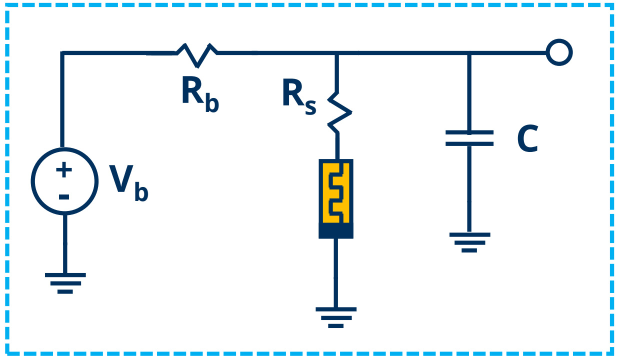


Project 5: Pattern Formation with Locally Active Memristors

- **Topic**: Pattern Formation in Reaction-Diffusion Memristor Cellular Nonlinear Networks (RD-MCNNs)
- **Task**: Simulation of RD-MCNNs employing locally active memristors and investigation of the impact of model parameters on the emerging patterns.
- **Challenge**: Network simulations on different platforms (e.g. Matlab, C/C++, Python, LTSpice etc.) and comparison of simulation speed, investigation of the impact of the device characteristics on the emerging patterns, design of a classification algorithm of the emerging patterns.
- **Supervisor**: Dr. Ahmet Samil Demirkol
ahmet_samil.demirkol@tu-dresden.de



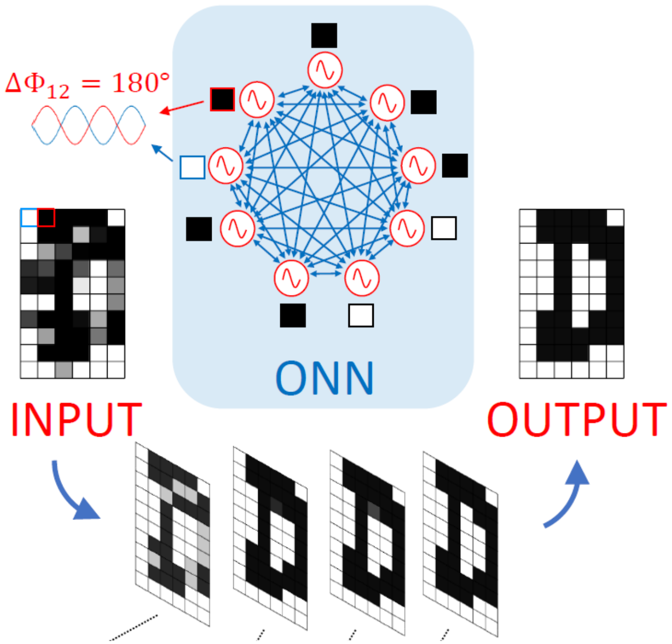


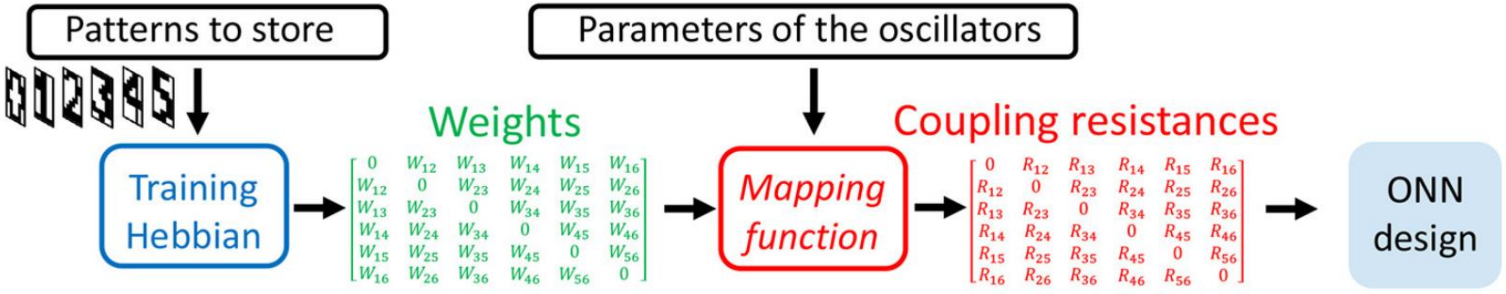


A. S. Demirkol *et. al.*, Pattern formation dynamics in a Memristor Cellular Nonlinear Network structure with a numerically stable VO2 memristor model, *Japanese Journal of Appl. Phys.*, **61,** SM0807, 2022, **doi:** 10.35848/1347-4065/ac8489

Project 6: Pattern Recognition with MONNs

- **Topic**: Digit Recognition with Memristor Oscillatory Neural Networks (MONNs)
- **Task**: Design and simulation of MONNs employing locally active memristor based neurons for letter/digit recognition.
- **Challenge**: Implementation of the MONN code, proper training of the network, mapping weights to coupling resistances, detection of the phase of the oscillators.
- **Supervisor**: Dr. Ahmet Samil Demirkol
ahmet_samil.demirkol@tu-dresden.de

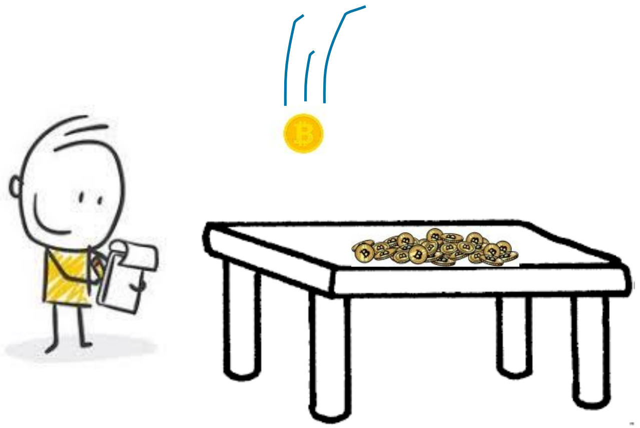




C. Delacour *et al*., "Oscillatory Neural Networks for Edge AI Computing," *ISVLSI*, Tampa, FL, USA, 2021, pp. 326-331, doi: [10.1109/ISVLSI51109.2021.00066](https://doi.org/10.1109/ISVLSI51109.2021.00066).

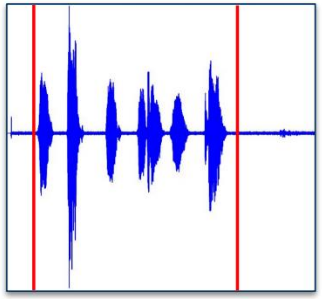
Project 7: Coin detection using PyTorch

- **Topic**: Recognize the coin based on raw sound data
- **Task**: Implement a Time-Series Classification model to detect different Euro coins and visualize similarities between them
- **Challenge**: You have to create the label vector based on the file name. You need to grab intermediate layer outputs and teach t-SNE to yourself, which is a state of the art clustering method.
- **Supervisor**: Dipl-Ing. Steffen Seitz, steffen.seitz@tu-dresden.de

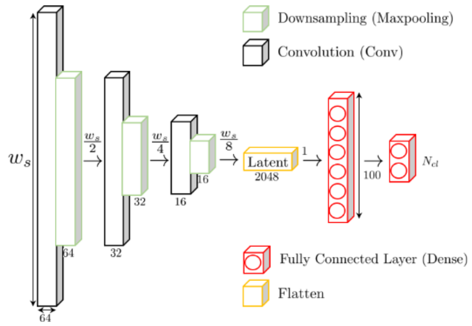


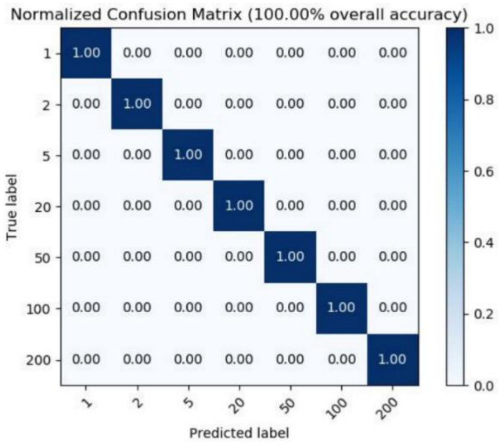
Euro Coins

Ultrasound **signal**



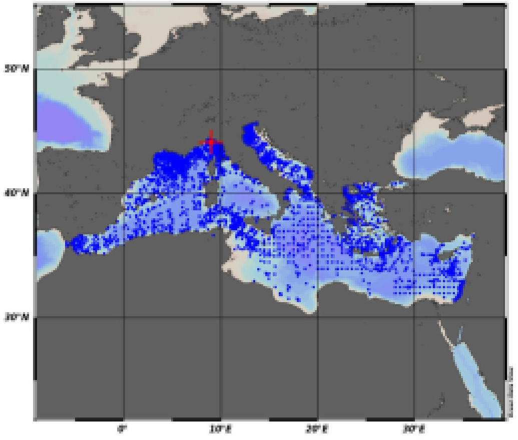
NN **classifier**

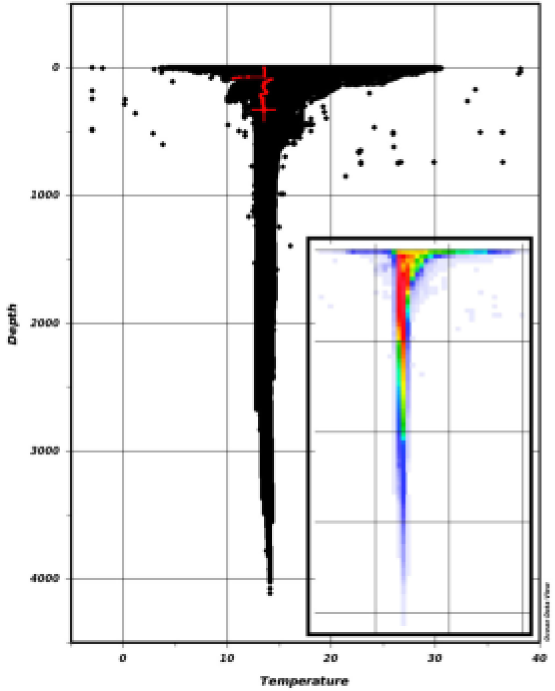




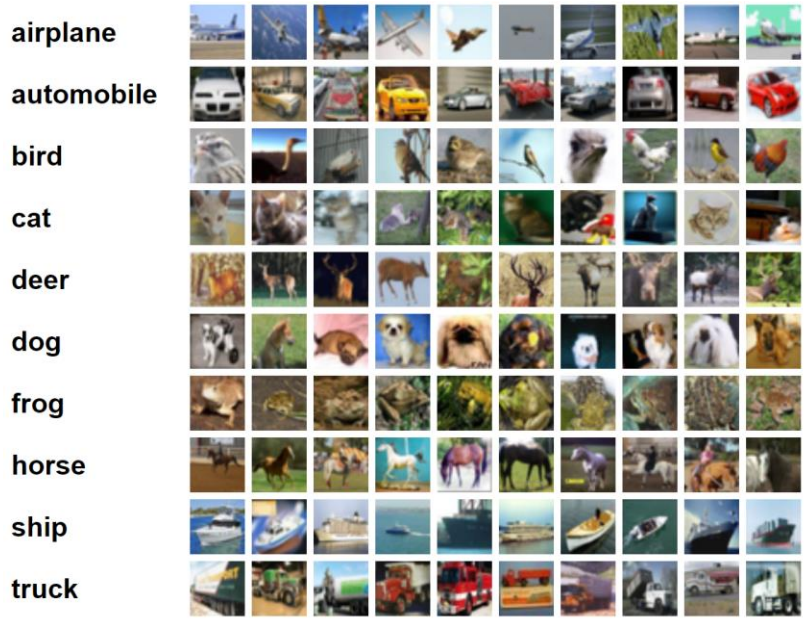
Project 8: AI-based system for ocean data quality control

- **Topic**: Build an AI-based system for supporting quality control of ocean temperature measurements
- **Task**: Build a classifier that detects/flags suspect gradient temperature errors in a depth profile from a climate model related measurement
- **Challenge**: You need to find out if you can exploit any neural network function to inform your network. Additionally you will need to compare multiple models/network functions and decide which features are worth using in your model
- **Supervisor**: Dipl-Ing. Steffen Seitz, steffen.seitz@tu-dresden.de

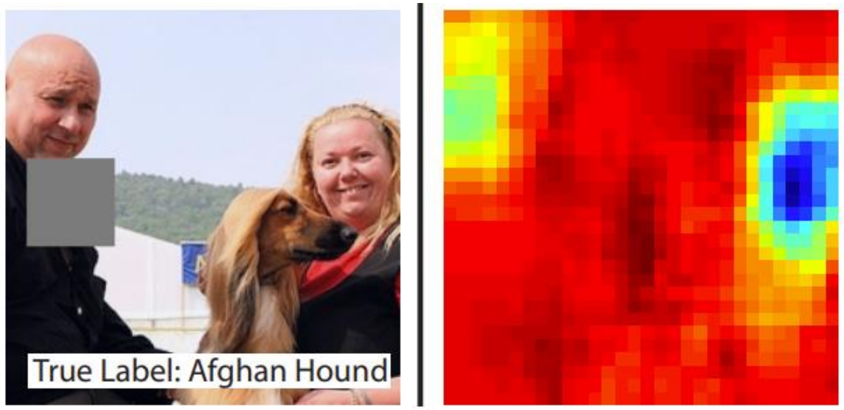




Project 9: Multi-Class Image Localization by Occlusion

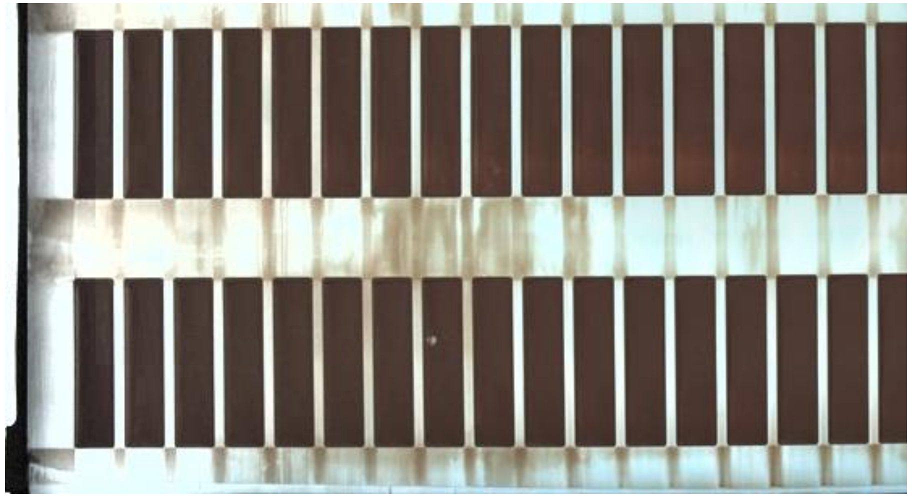


- **Topic**: Build a Multi-Class Object Localization algorithm that is able to track the location of the class in the image.
- **Task**: Implement a MLP and CNN based approach to classify the 10 CIFAR categories and localize the classes by occluding parts of the image and classify for the object in the remaining picture!
- **Challenge**: You will need to track your network training time as well as the layer parameters and FLOPs. Additionally you will need to learn about transfer learning and how to train pre-trained models. You have to utilize a GPU.
- **Supervisor**: Dipl-Ing. Steffen Seitz, steffen.seitz@tu-dresden.de

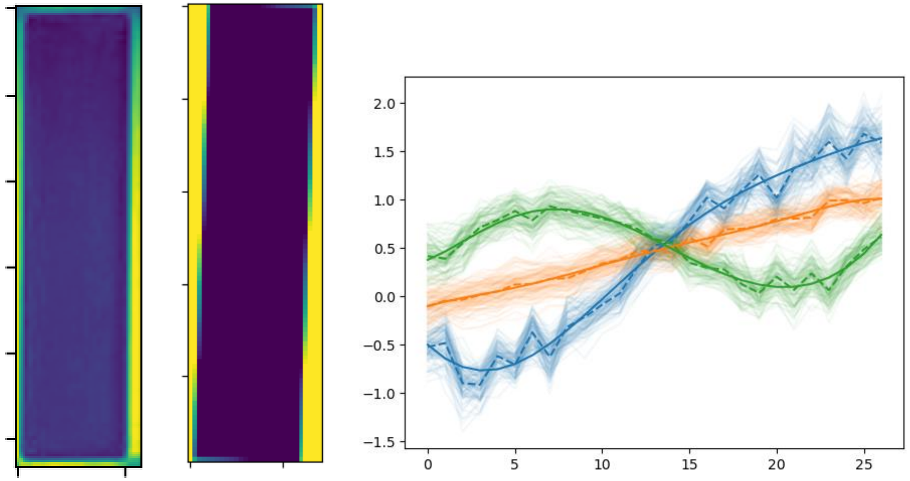


Based on Zeiler and Fergus (2013) - https://arxiv.org/pdf/1311.2901.pdf

Project 10: Precise angle identification of rotated rectangles

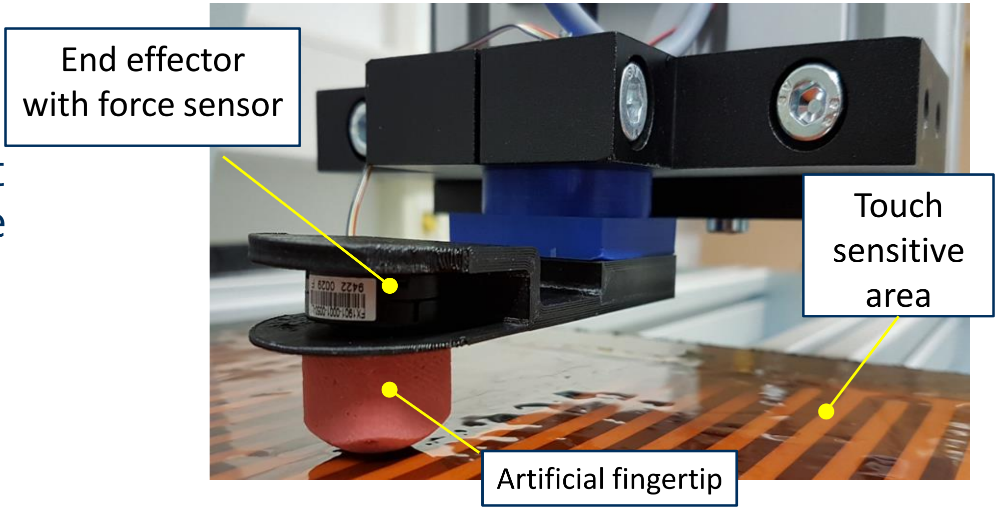


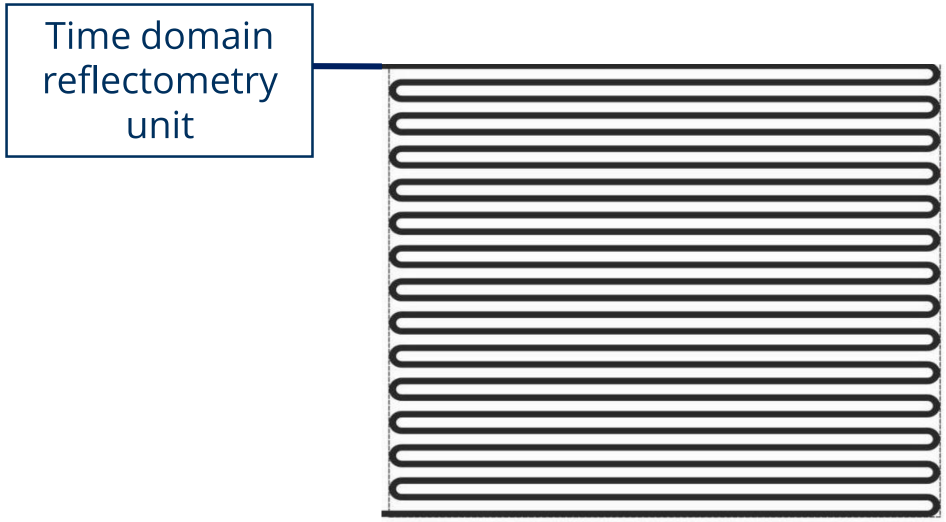
- **Topic**: Build a NN-based algorithm that is able to identify the rotation angle of a rectangle
- **Tasks**: Create an artificial dataset. Find a suitable architecture (e. g. combination of CNN and MLP) which solves the angle identification task. Train and test with your own dataset. Study the influence of noise and generalization capabilities.
- **Challenge**: CNNs are mostly used for classification (discrete output). Here, we have a regression task (continuous output).
- **Supervisor**: Dr. Carsten Knoll
- carsten.knoll@tu-dresden.de



Project 11: Force and Location Detection of Touch Sensor using PyTorch

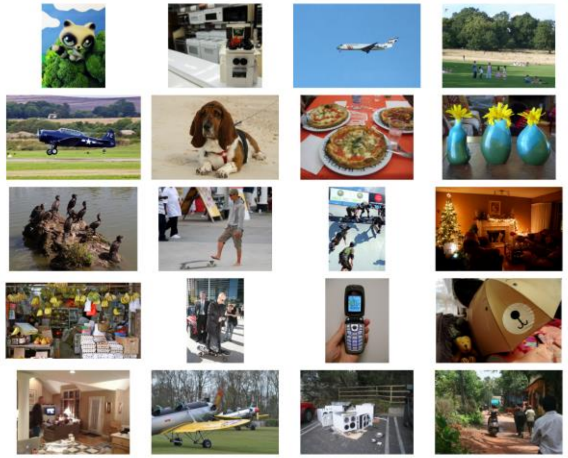
- **Topic:** Touch force and touch location classification
- **Task**: Develop a classification algorithm to distinguish different touch force levels and the touch locations along a folded single line using a time domain reflectometry sensor. The data set contains time series from 200 locations and 6 different forces collected with a robot arm.
- **Challenge**: The folded design of the sensor, along with overlaps in touch forces across its wire lines, make difficult to precise impedance measurements. This, combined with the sensor natural noise, increases the challenge of detecting touch force, differentiating touch intensities, locating their exact positions and resolving overlaps in the presence of such noise.
- **Supervisor**: M.Sc. Adnan Haidar, adnan.haidar@tu-dresden.de



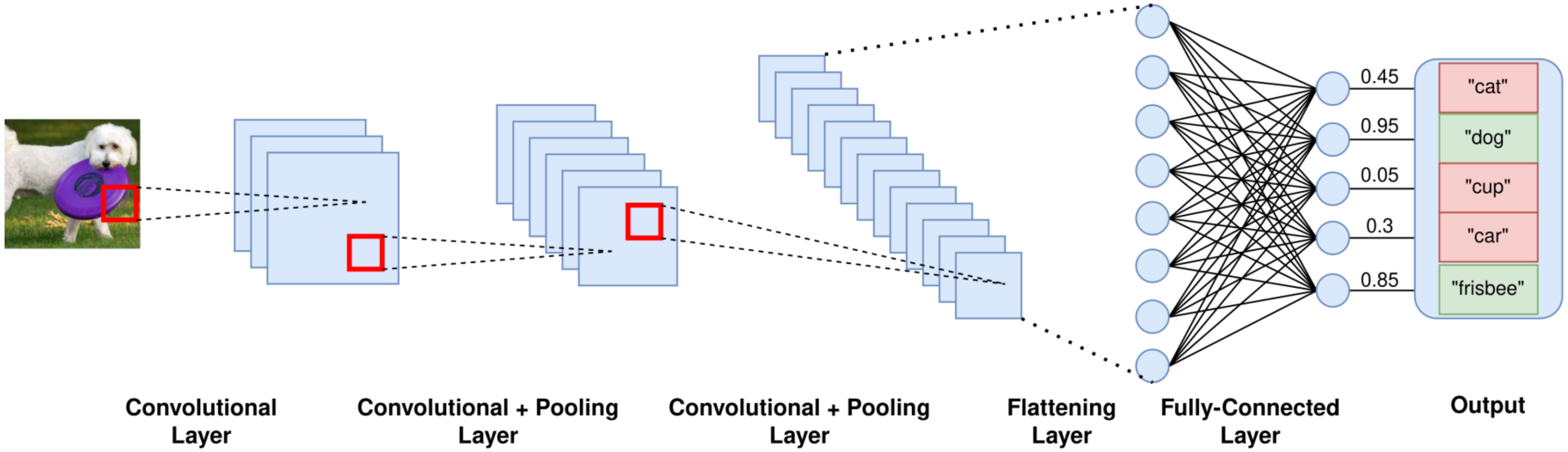


sensitive area

Project 12: Multi-Label Image Classification on the COCO Dataset



- **Topic**: Multi-Label Image Classification on the COCO2017 Dataset
- **Task**: Perform dataset analysis, implement data preprocessing, build, train and evaluate different neural networks in a multi-label classification task using PyTorch.
- **Challenge**: You will need to understand the dataset's structure and deal with the problem of class inbalance. You will learn about transfer learning. You will have to use a GPU for training/inference.
- **Supervisor**: Dipl.-Ing. Kilian Göller, kilian.goeller@tu-dresden.de



Literature

Artificial Intelligence:

— Mainzer, Klaus. *Artificial Intelligence - When Do Machines Take Over?* Springer Berlin Heidelberg, 2020. https://doi.org/10.1007/978-3-662-59717-0

Emerging Neuromorphic Devices:

— Tang et al. “Bridging Biological and Artificial Neural Networks with Emerging Neuromorphic Devices: Fundamentals, Progress, and Challenges.” *Advanced Materials*, vol. 31, no. 49, 2019, p. 1902761. https://doi.org/10.1002/adma.201902761

Other related courses at TU Dresden

— *Machine Learning* and *Computer vision* (Prof. Bjoern Andres)

— *Memory Technology 2* (Prof. Thomas Mikolajick), innovative memory technologies like FeRAM, MRAM, PCM, RRAM

Introduction Lecture: Summary

- You can classify Artificial Intelligence, Machine Learning and Neural networks.
- You are able to describe the challenges of today's hardware to train and compute neural networks.
- You know the Von-Neumann bottleneck between processing and storage unit and you can explain why compute-in-memory is one solution to overcome it.
- You understand that the memristor is a memory resistor which can switch non-volatile between different resistances, called low/high resistance states.
- You can explain how memristive crossbars are applied to accelerate matrix-vector-multiplications for inference.

Next lecture: 2. Machine Learning (Steffen Seitz)