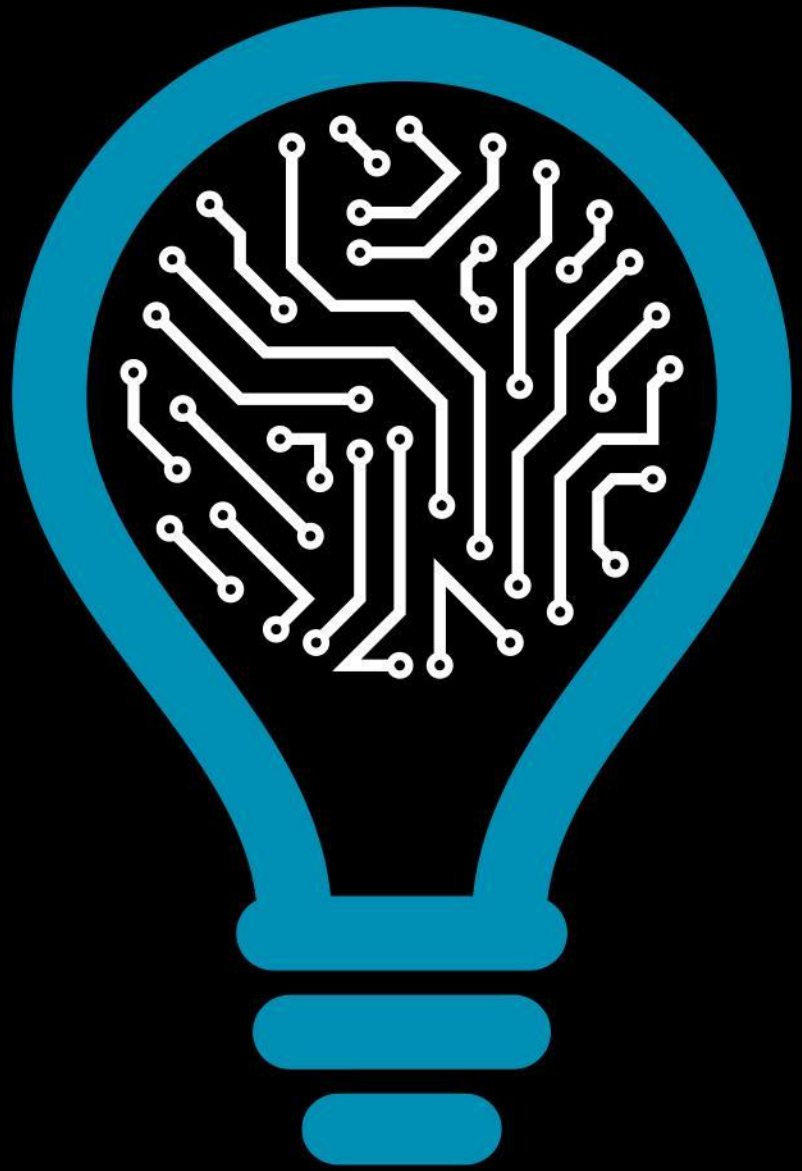
Edge AI - Lecture 2



TAIA - Advanced Topics on Artificial Intelligence

Tiago Filipe Sousa Gonçalves

tiago.f.goncalves@inesctec.pt | tiagofs@fe.up.pt

Acknowledgement: Leonardo Gomes Capozzi

leonardo.g.capozzi@inesctec.pt

INESCTEC

INSTITUTE FOR SYSTEMS AND COMPUTER ENGINEERING, TECHNOLOGY AND SCIENCE

INESCTEC

Outline

1. Back to the Future (of Computing)

2. Neuromorphic Computing: can we (literally) build inorganic brains?

3. Let it spike! A brief tutorial on spiking neural networks

4. Take-home messages and further readings

INESCTEC

1. Back to the Future (of Computing)

Foundations of Computing: Von Neumann Architecture

- Published by John von Neumann in 1945, **the Von Neumann architecture is based on the stored-program computer concept**, where instruction data and program data are stored in the same memory[1]

- This design is still used in most computers produced today and consists of:

  - **Control Unit (CU)**: controls the operation of the computer's ALU, memory and input/output devices, and provides the timing and control signals required by other computer components
  - **Arithmetic and Logic Unit (ALU)**: allows arithmetic and logic operations to be carried out
  - **Memory Unit**: consists of Random Access Memory (RAM), sometimes referred to as primary or main memory
  - **Registers**: are high speed storage areas in the CPU (all data must be stored in a register before it can be processed)
  - **Central Processing Unit (CPU)**: is the electronic circuit responsible for executing the instructions of a computer program (sometimes referred to as the microprocessor or processor) and contains the ALU, CU and a variety of registers
  - **Buses**: are the means by which data is transmitted from one part of a computer to another, connecting all major internal components to the CPU and memory
  - **Inputs/Outputs**: the input and output devices

Sources: [1] https://www.computerscience.gcse.guru/theory/von-neumann-architecture

Foundations of Computing: Von Neumann Architecture

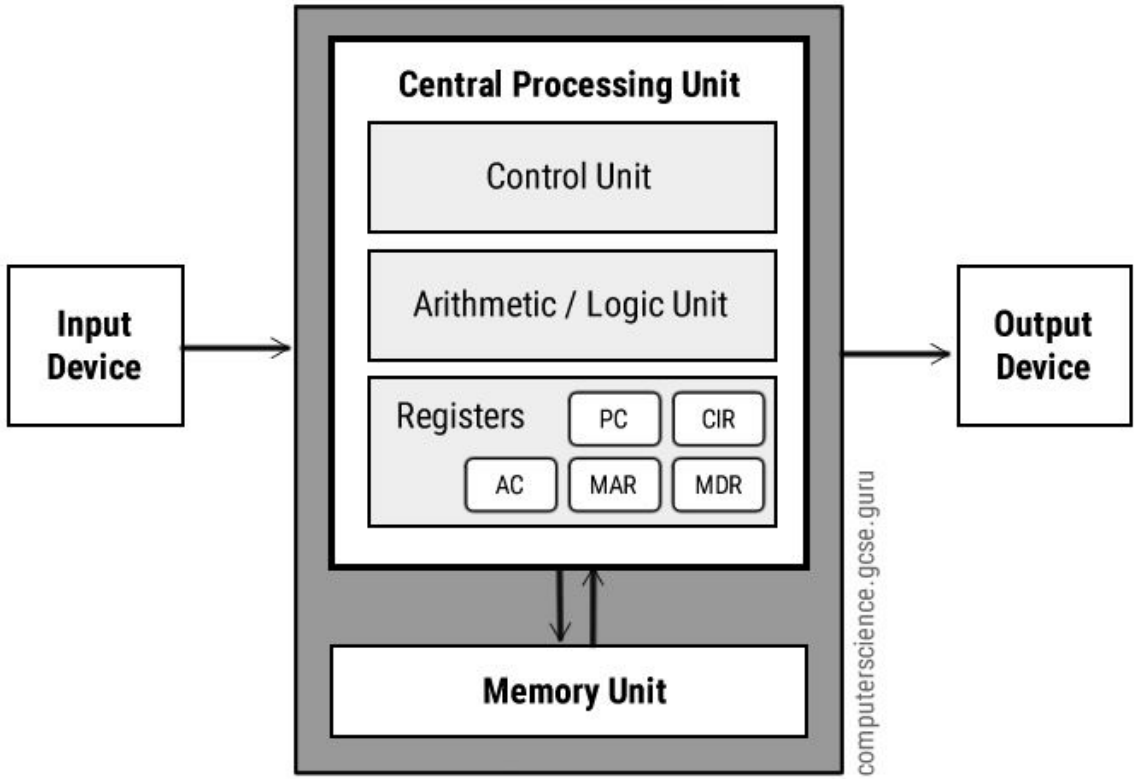


Figure - Von Neumann Architecture (Image from [1])

Sources: [1] https://www.computerscience.gcse.guru/theory/von-neumann-architecture

Changing the Paradigm: Is Moore's Law Dead?[1]

- **Moore's Law** is a term used to refer to the observation made by Gordon Moore in 1965 that the number of transistors in a dense integrated circuit doubles about every two years[1]

- Semiconductor process technology has always **increased in complexity**
  - In recent times, **complexity increases have been accelerating**
  - The evolution of semiconductor process technology is reaching **molecular limits**, and this is **slowing the exponential benefits of Moore's Law**

- **The pace of technological innovation is not slowing down!**

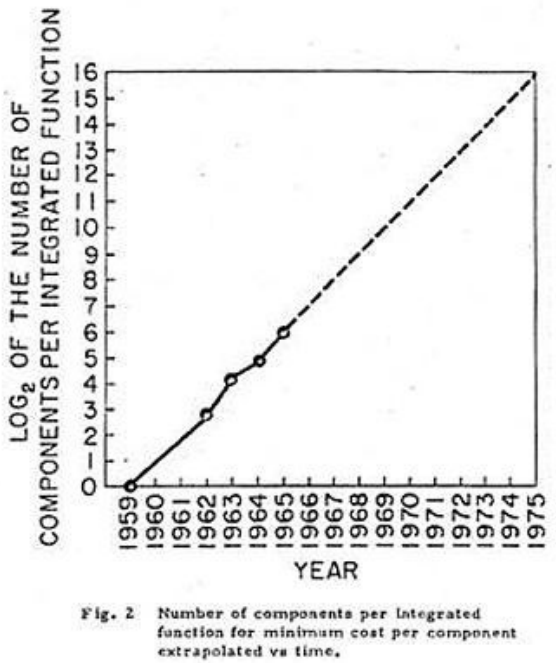


Figure - Moore's Law graph (Image from [2])

Sources: [1] https://www.synopsys.com/glossary/what-is-moores-law.html, [2] https://www.intel.com/pressroom/kits/events/moores_law_40th/index.htm

Changing the Paradigm: Other Ways of Computing

- General purpose computers **will probably keep** the Von Neumann architecture, however, there also other approaches (the **non-Von Neumann architectures**[1])
  - The Harvard architecture has two separate buses for instruction and data, hence, CPU can access instructions and read/write data at the same time
  - For instance, **X86 and ARM processors** use the **Modified Harvard Architecture** that has two separate caches (data and instruction)[2]

- On the other hand, there are other paradigms of computing that will eventually play an interesting role in our daily lives:
  - **Biochemical Computing**: aims at building non-silicon circuits and focus, instead, in using molecules to run programmed functions[3, 4]

  - **Quantum Computing**: leverages several concepts of quantum mechanics and physics to solve highly complex problems[5]

  - **Neuromorphic Computing**: uses new algorithmic approaches that emulate how the human brain interacts with the world to deliver capabilities closer to human cognition[6]

Sources: [1] https://www.encyclopedia.com/computing/dictionaries-thesauruses-pictures-and-press-releases/non-von-neumann-architecture, [2] https://www.geeksforgeeks.org/harvard-architecture/, [3] https://www.technologyreview.com/2000/05/01/236304/biological-computing/, [4] https://www.caltech.edu/about/news/computing-biochemical-circuits-made-easy-54206, [5] https://www.ibm.com/quantum-computing/what-is-quantum-computing/, [6] https://www.intel.com/content/www/us/en/research/neuromorphic-computing.html

Neuromorphic computing may be the key to Edge AI

- **The von Neumann Bottleneck**: von Neumann chips have to shuttle information back and forth between the memory and CPU, hence, **they waste time** (computations are held back by the speed of the bus between the compute and memory) **and energy**[1]

- Current trends of artificial intelligence (e.g., deep neural networks) require **huge processing power and energy consumption**[2]

- We know that the brain **is compact**, **does not consume huge amounts of energy** and has a **high degree of adaptability**[1]
  - Therefore, if we are able to **achieve computing paradigms that work just like the brain** (neuromorphic computing), then we may find the key to **effectively implement artificial intelligence at the edge**[3]

Sources: [1] https://www.zdnet.com/article/what-is-neuromorphic-computing-everything-you-need-to-know-about-how-it-will-change-the-future-of-computing/,
[2] https://towardsdatascience.com/the-future-of-artificial-intelligence-neuromorphic-computing-34bcc5cc35a,
[3] https://www.hpe.com/us/en/insights/articles/whats-this-neuromorphic-computing-youre-talking-about-2105.html

INESCTEC

2. Neuromorphic Computing: can we (literally) build inorganic brains?

The Neuron (or the basic working unit of the brain)[1]

- **Neurons are cells within the nervous system that transmit information** to other nerve cells, muscle, or gland cells

- **Most neurons have:**
  - **Cell body**: contains the nucleus and cytoplasm
  - **Axon**: extends from the cell body and often gives rise to many smaller branches before ending at nerve terminals (i.e., **output device**)
  - **Dendrites**: extend from the neuron cell body and receive messages from other neurons (i.e., **input device**)
  - **Synapses**: are the contact points where one neuron communicates with another

- **When neurons receive or send messages, they transmit electrical impulses along their axons**
  - Many axons are covered with a layered **myelin sheath**, which **accelerates the transmission** of electrical signals along the axon
  - This sheath is made by specialised cells called **glia**

Sources: [1] https://www.brainfacts.org/brain-anatomy-and-function/anatomy/2012/the-neuron

The Neuron (or the basic working unit of the brain)[1]

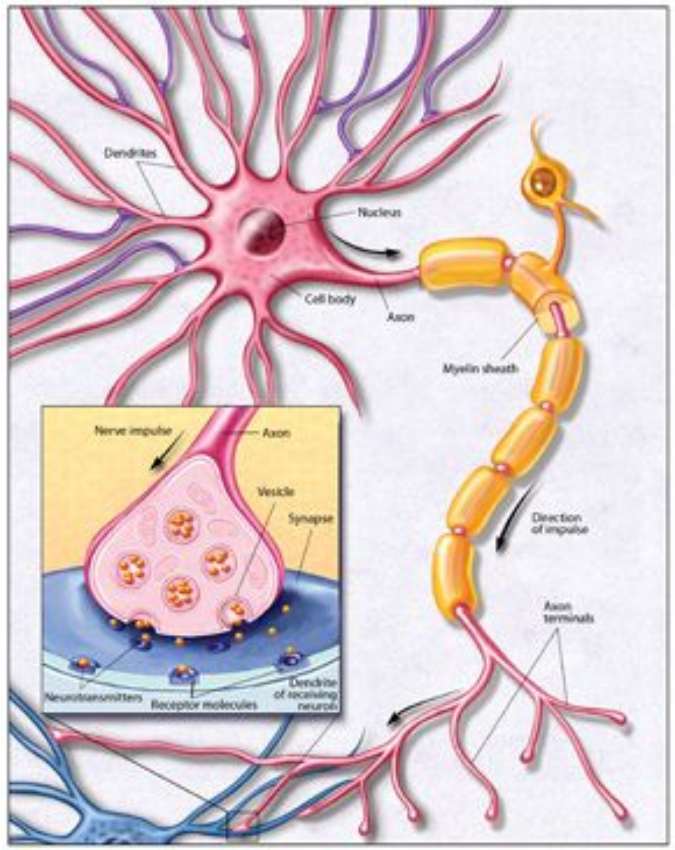


Figure - The neuron (Image from [1])

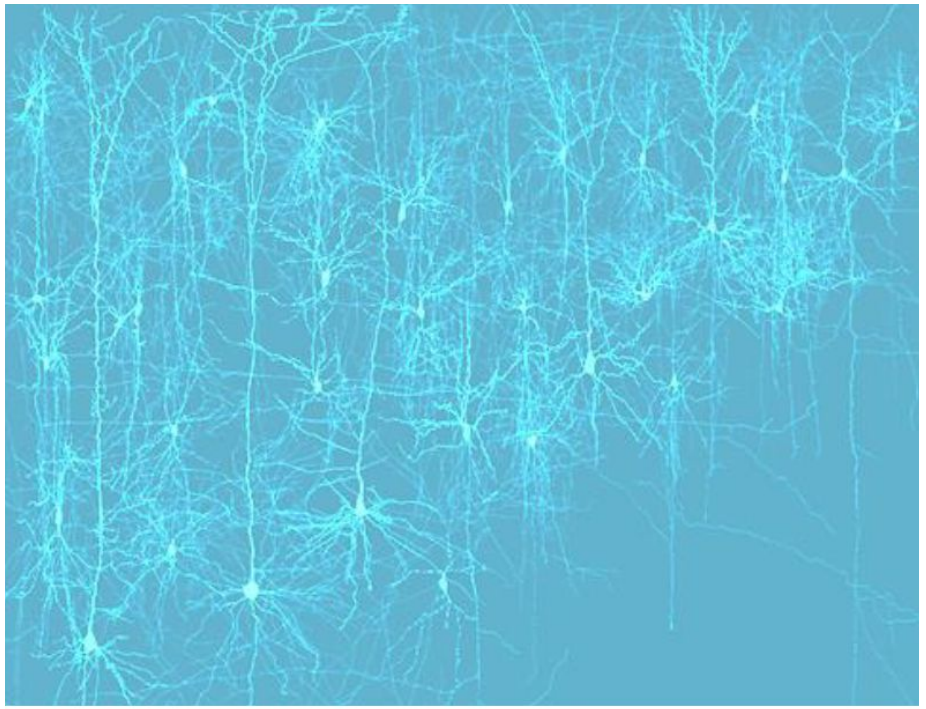


Figure - Cortical pyramidal neurons
(Image from Mafalda Sousa - i3S)

Sources: [1] https://www.brainfacts.org/brain-anatomy-and-function/anatomy/2012/the-neuron

Meet the Memristor (or the Inorganic Neuron)![1]

- **Memristor** (the junction of **“memory”** and **“resistor”**) is a type of passive circuit element capable of “remembering” its past states (i.e., electrical resistance value), even when no energy passes through it (i.e., **the non-volatility property**)
  - Memristors can also be used as **both memory and processing units**, thus being an answer to **the von Neumann Bottleneck**

- Memristors are **the elementary units of neuromorphic hardware** and allow us to achieve:
  - **High speed and high parallelism of calculations**, just like the brain, that it is highly parallelised (i.e., millions of calculations happening at the same time)

  - **Much lower energy consumption**

Sources: [1] https://towardsdatascience.com/the-future-of-artificial-intelligence-neuromorphic-computing-34bcc5cc35a

Meet the Memristor (or the Inorganic Neuron)!

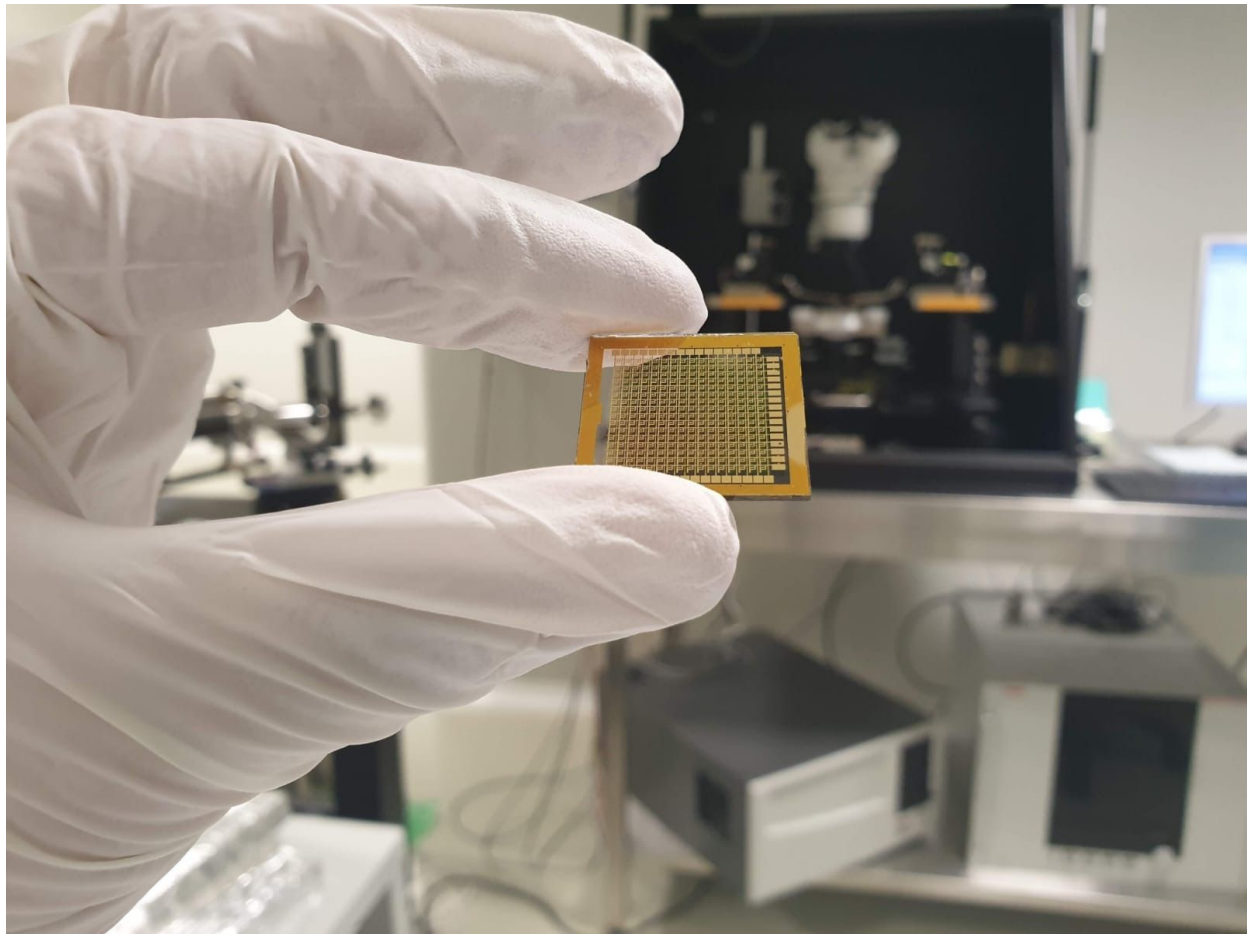


Figure - The memristor (Image from [1, 2])

Sources: [1] https://towardsdatascience.com/the-future-of-artificial-intelligence-neuromorphic-computing-34bcc5cc35a, [2] https://www.cyber-swarm.net/

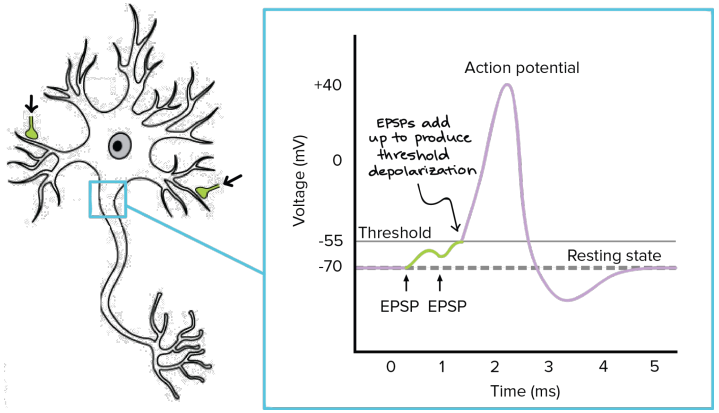
INESCTEC

3. Let it spike! A brief tutorial on spiking neural networks

Let's delve into the world of event-based processing...[1]

- We already know that **neuromorphic computing is inspired by the structure and functioning of the brain**
  - In neurons, the **information is processed via discrete action potentials (spikes) in time through synapses**

- **A spike is generated when the running sum of changes in the membrane potential crosses a threshold**
  - **The information about these external stimuli (or ongoing computations) is encoded in the rate of spike generation and on the temporal pattern of spike trains**



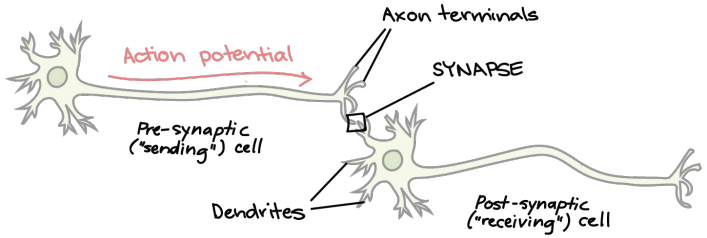


Figure - The synapse and action potential
(Images from [2])

Sources: [1] https://www.coursera.org/learn/computational-neuroscience, [2] https://www.khanacademy.org/science/biology/human-biology/neuron-nervous-system/a/the-synapse

And discover the realm of Spiking Neural Networks...

- **Spiking neural networks** can be seen as a more biologically realistic approach than artificial neural networks: the baseline architecture of a spiking neural network consists of **spiking neurons and interconnecting synapses** that are modeled by adjustable scalar weights

- One of the main concerns relies **on the proper encoding of the data** to be fed to the spiking neural network during the training phase:
  - Rate-based method
  - Temporal encoding
  - Population coding

- Several **dynamic models** have been proposed:
  - Spike response model (SRM)
  - Izhikevich neuron model
  - Leaky integrated-and-fire (LIF) neuron

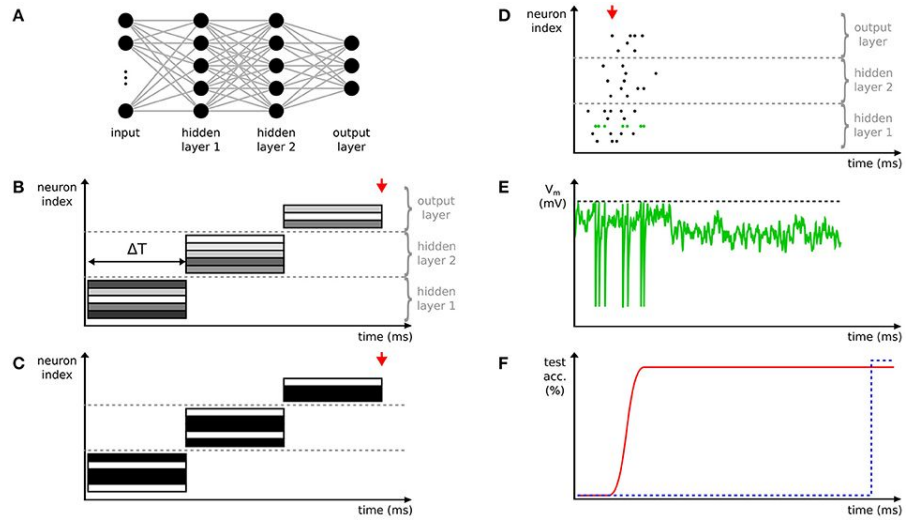


Figure - From deep artificial neural networks to spiking neural networks (Images from [1])

Sources: [1] Pfeiffer and Pfeil "Deep Learning With Spiking Neurons: Opportunities and Challenges"

And understand the realm of Spiking Neural Networks...

- In SNNs, the spike trains are propagated through **synaptic connections**:
  - **Excitatory**: increases the neuron's membrane potential upon receiving input
  - **Inhibitory**: decreases the neuron's membrane potential

- One of the paradigms in computational neuroscience relies on the concept of **spike-timing-dependent plasticity (STDP)**
  - **Long-term potentiation (LTP)**, if we know that the presynaptic neuron fires for a brief time before the postsynaptic neuron, the weight connecting them is strengthened
  - **Long-term depression (LTD)**, if the presynaptic neuron fires briefly after the postsynaptic neuron, then the causal relationship between the temporal events is nonsense and the weight connecting these two neurons is weakened

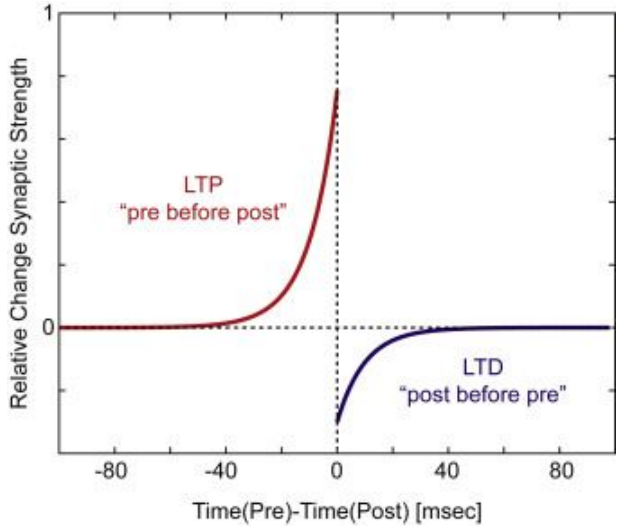


Figure - LTP and LTD (Images from [1])

$$\Delta w = \begin{cases} Ae^{-\frac{(|t_{pre}-t_{post}|)}{\tau}}, t_{pre} - t_{post} \leq 0, A > 0 \\ Be^{-\frac{(|t_{pre}-t_{post}|)}{\tau}}, t_{pre} - t_{post} > 0, B < 0 \end{cases}$$

Sources: [1] Flavio Fröhlich "Synaptic Plasticity"

Case Study: Spiking Neural Network Architecture[1]

- This architecture is composed of:
  - **An input layer**, that contains 28x28 or 32x32 neurons
  - **A processing layer**, composed of a variable, but equal, number of excitatory and inhibitory neurons
  - **The excitatory neurons are connected in a one-to-one fashion to inhibitory neurons**
  - **Each of the inhibitory neurons is connected to all excitatory ones, except for the one from which it receives a connection**

- **Images are modeled as a Poisson spike-train** and are fed into the excitatory layer

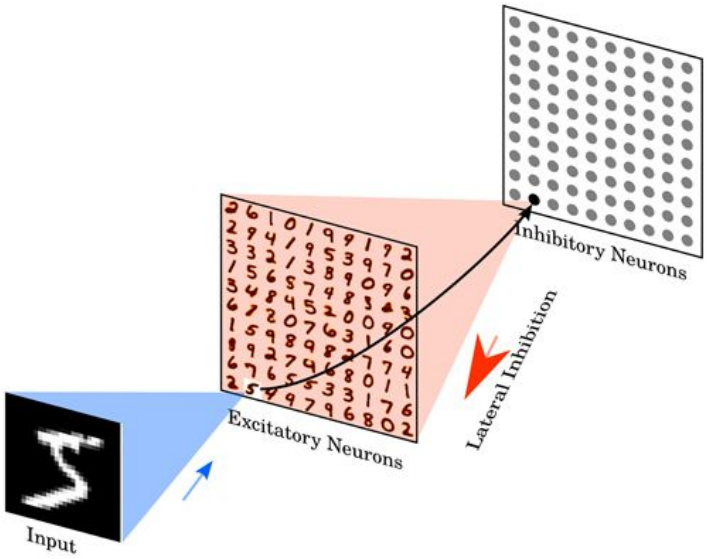


Figure - The spiking neural network architecture
(Image from [1])

Sources: [1] Diehl and Cook "Unsupervised learning of digit recognition using spike-timing-dependent plasticity"

Case Study: Spiking Neural Network Architecture[1]

- Each synapse keeps track of the **synaptic weight**, *w*, and the **presynaptic spike history**, x_{pre}
- Every time a presynaptic spike arrives at the synapse, the trace (x_{pre}) is increased by 1 and it **decays exponentially**
- **The higher the target value, the lower the synaptic weight will be**
  - This offset promotes that presynaptic neurons that rarely lead to the firing of the postsynaptic neuron will become more and more disconnected
- Each excitatory neuron's membrane has a value of Θ added to the threshold value v_{thr}, therefore the final threshold of an excitatory neuron becomes $v_{thr} + \Theta$
  - Θ is increased every time the neuron fires and decays exponentially
- Neurons that **fire very frequently** will have a **higher membrane threshold**, making them require more inputs to spike soon

$$\Delta w = \eta(x_{pre} - x_{tar})(w_{max} - w)^{\mu}$$

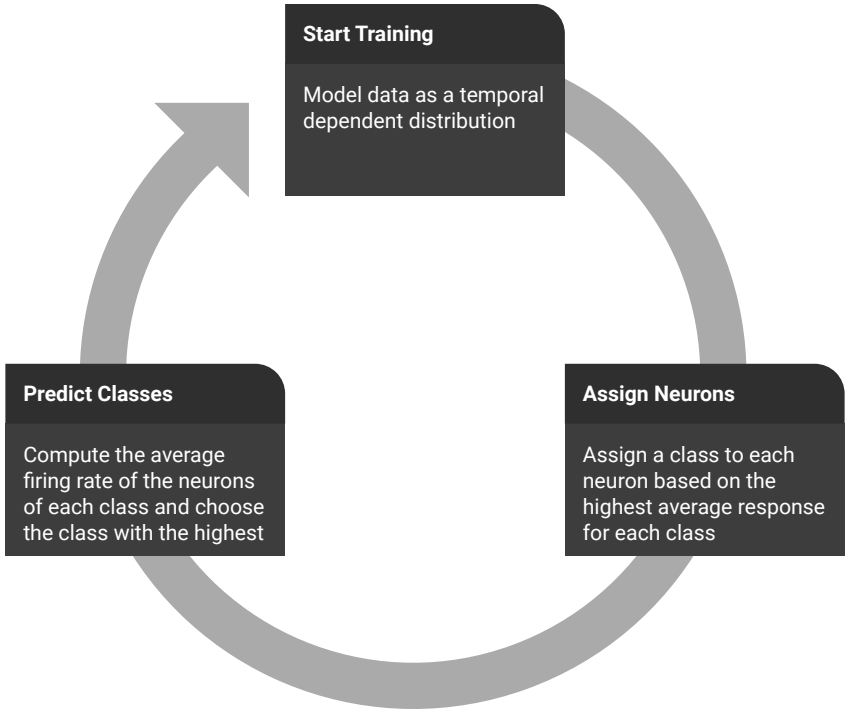
Sources: [1] Diehl and Cook "Unsupervised learning of digit recognition using spike-timing-dependent plasticity"

Case Study: Spiking Neural Network Architecture[1]

- **A spike is generated if the membrane crosses a given threshold** v_{thr}
- The membrane voltage is then reset to a reset potential v_{res}

$$\frac{dv_{mem}(t)}{dt} = \sum_i \sum_{s \in S_i} w_i \delta(t - s)$$

- **The complete training scheme**:



Sources: [1] Diehl and Cook "Unsupervised learning of digit recognition using spike-timing-dependent plasticity"

How can we code spiking neural networks?

- **There are several interesting frameworks and libraries to design and implement spiking neural networks:**
  - **BindsNET**[1]: written on top of the PyTorch framework, it is a spiking neural network simulation library that can leverage the spiking neural network development in CPU or GPU
  - **GENESIS**[2]: is a general purpose simulation platform that was developed to support the simulation of neural systems
  - **NEURON**[3]: is a simulator for models of neurons and networks of neuron
  - **Brian**[4]: is a free, open source simulator for spiking neural networks, written in the Python programming language and is available on almost all platforms
  - **NEST**[5]: is a simulator for spiking neural network models that focuses on the dynamics, size and structure of neural systems
  - **SpykeTorch**[6]: is a simulator of convolutional spiking neural networks with at most one spike per neuron
  - **SpikeNET**[7]: is a simulator for modeling large networks of asynchronously spiking neurons

Sources: [1] https://github.com/BindsNET/bindsnet, [2] https://github.com/genesis-sim/genesis-2.4, [3] https://github.com/neuronsimulator/nrn, [4] https://github.com/brian-team/brian2, [5] https://nest-simulator.org/, [6] https://cnrl.ut.ac.ir/SpykeTorch/doc/index.html, [7] Delorme et al. "SpikeNET: A simulator for modeling large networks of integrate and fire neurons"

4. Take-home messages and further readings

A (tentative) fair and accurate summary of this lecture

- **General purpose computers will keep using the von-Neumann architecture**, however, that are other computing paradigms that will eventually play an interesting role in our daily lives
  - Biochemical Computing
  - Quantum Computing
  - **Neuromorphic Computing**

- **Neuromorphic computing** may be the key to effectively implement artificial intelligence at the edge
  - **Memristors** are the **elementary units of neuromorphic hardware** and allow us to achieve **high speed and high parallelism of calculations**, and **much lower energy consumption**

- **Spiking neural networks can be seen as a more biologically realistic approach than artificial neural networks**
  - Besides, spiking neural networks **typically require fewer operations**, are more **energy-efficient**, and also more **hardware-efficient**, making them very appealing for the future

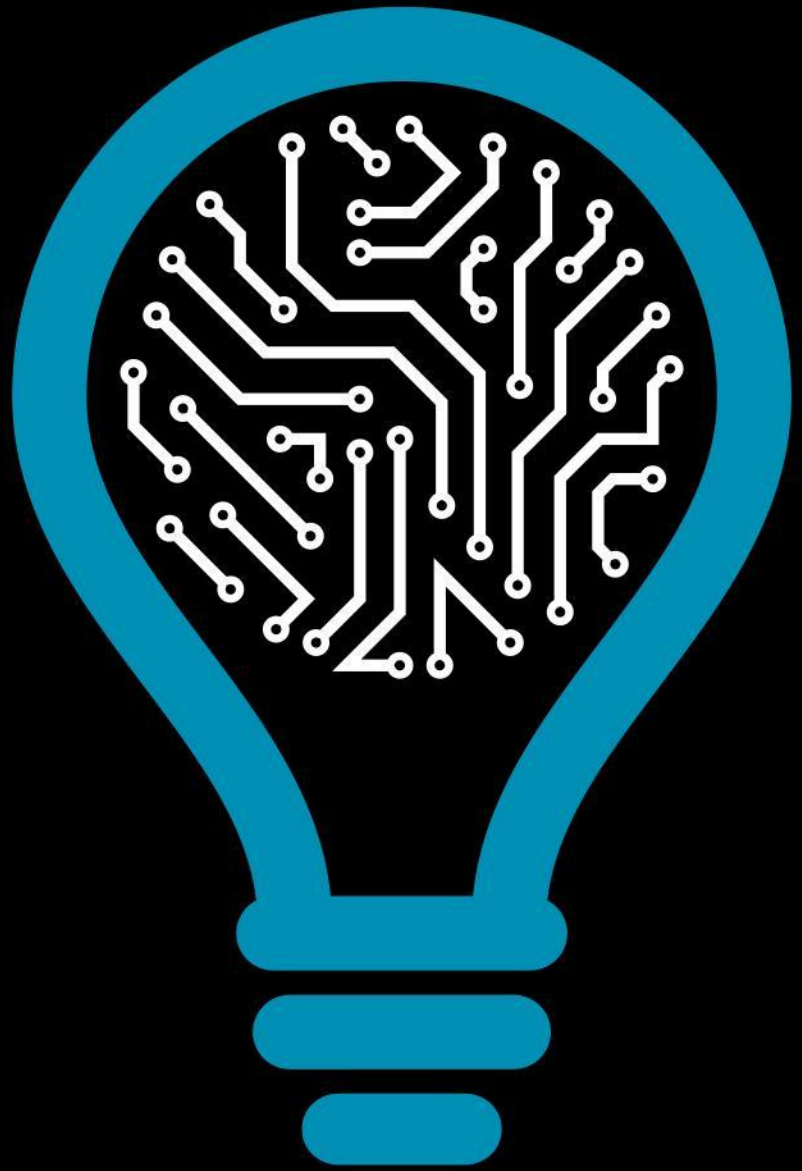
Further readings...

- Thubagere et al. “Compiler-aided systematic construction of large-scale DNA strand displacement circuits using unpurified components”
- https://cnvrg.io/spiking-neural-networks/
- Yang et al. “A memristor-based neural network circuit with synchronous weight adjustment”
- Kasabov et al. “Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition”
- Xu et al. “An optimal time interval of input spikes involved in synaptic adjustment of spike sequence learning”

Further readings...

- Walter et al. “Neuromorphic implementations of neurobiological learning algorithms for spiking neural networks”
- Demin et al. “Necessary conditions for STDP-based pattern recognition learning in a memristive spiking neural network”
- Hong et al. “Novel circuit designs of memristor synapse and neuron”
- Cho et al. “Efficient architecture for deep neural networks with heterogeneous sensitivity”
- Eom et al. “Deep-learned spike representations and sorting via an ensemble of auto-encoders”

Edge AI - Lecture 2



TAIA - Advanced Topics on Artificial Intelligence

Tiago Filipe Sousa Gonçalves

tiago.f.goncalves@inesctec.pt | tiagofs@fe.up.pt

Acknowledgement: Leonardo Gomes Capozzi

leonardo.g.capozzi@inesctec.pt

INESCTEC

INSTITUTE FOR SYSTEMS AND COMPUTER ENGINEERING, TECHNOLOGY AND SCIENCE