NEUROTECH

Deliverable D2.2: First version of the NEUROTECH Roadmap and Benchmarks published.

- Project: NEUROTECH, Grant number 824103
- Report dissemination level: public
- Report target delivery date (updated version): project month 13 = November 2019

Summary

This deliverable was submitted with a delay compared to the date estimated in DoA because the date for the NEUROTECH Forum, during which contributions to the Roadmap from the community were collected fell in Month 13 (Nov. 4).

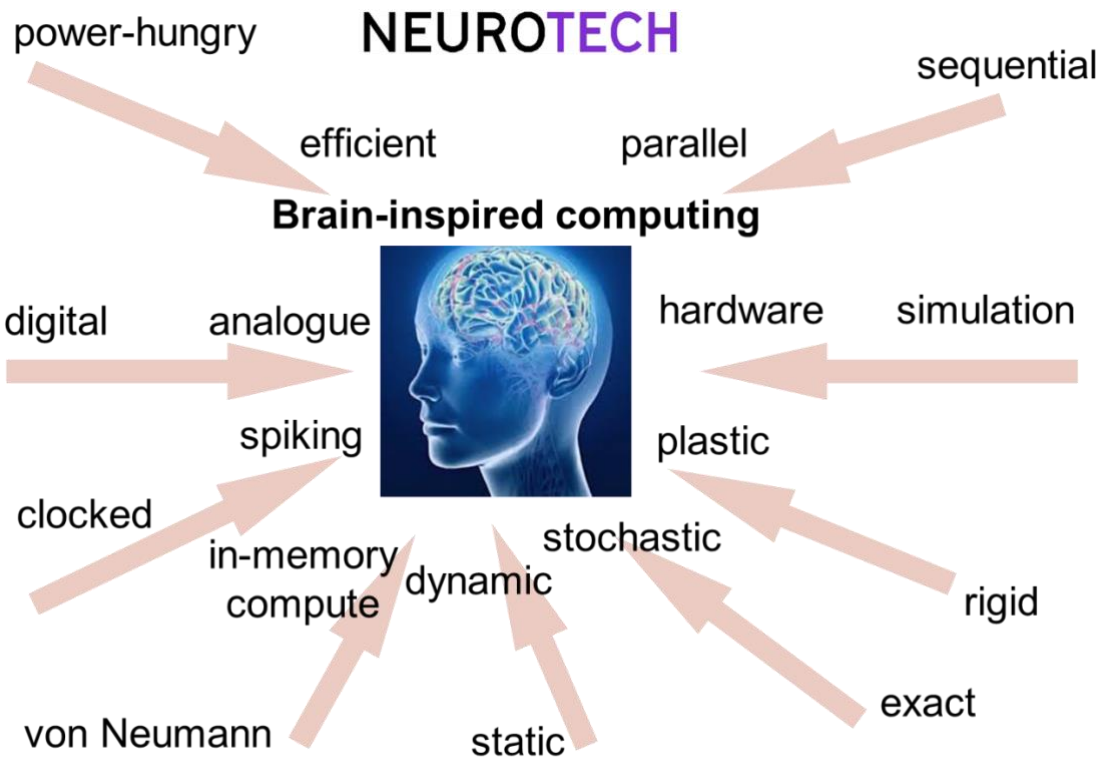
Introduction:

The goal of the Roadmap

This roadmap will be completed into two stages, where this is the first stage.
The aim of the first roadmap is to gather ideas and define directions. We identify key directions for neuromorphic computing, key applications, key technologies and main challenges. We present a qualitative state of the art and goals of the field.
The aim of the final roadmap is to complete the first by quantifying these goals and the state of the art, putting a timeline and proposing ideas to tackle the challenges.

Approach to develop a Roadmap

We used the NEUROTECH Forum to collect opinions from experts and the community on several questions related to the NCT Roadmap. This was done during a guided panel discussion (recording is available) and with an online questionnaire, which the Forum participants were invited to fill-in. The members of the Work Groups were also contacted to give their opinion.



Neuromorphic computing as a goal

Our first action for developing a Roadmap was to improve the definition of neuromorphic computing. Rather than having an inside/outside boundary, we see neuromorphic computing as a goal towards which different directions converge. These directions are schematized in Figure 1. They correspond to features of the brain as a computer, which we seek inspiration from. These directions structure the roadmap of neuromorphic computing as they are the guiding principles of the field.

Each of these directions represent a breakthrough from the current computing paradigm. In such, Neuromorphic computing represents an extremely ambitious multi-disciplinary effort. Each direction will require significant advances in computing theory, architecture and device physics.

Hardware vs. simulation

Taking inspiration from the brain for computing is already present in machine learning and artificial intelligence through artificial neural network algorithms. This abstract inspiration has already given rise to tremendous progress in image, video, audio and natural language processing, and to successful commercial applications. However, in order to unlock significant gains in terms of performance and efficiency, a more ambitious step needs to be taken: to build a new kind of computers, inspired from the brain at the hardware level. This is the goal of neuromorphic computing. We seek not just simulate artificial neural networks, but to actually build them.

Efficient vs. power-hungry

Application-wise, one key motivation for neuromorphic computing is to achieve higher power efficiency than existing solutions. Artificial neural networks, when run on conventional hardware, consume a lot of energy. State-of-the-art GPUs consume several hundreds of Watts, which limit the deployment of neural networks on embedded systems. Even supercomputers consuming a Mega Watt cannot emulate the whole human brain, which limits our ability to improve our understanding of the brain through such simulations. In comparison, the human brain only consumes 20 Watt. The energy efficiency of the brain is several hundreds of tera operations per second and per Watt, while existing solutions are limited to a few tera operations per second and per Watt. By building computers inspired from the brain at the hardware level, neuromorphic computing will bridge this energy efficiency gap.

Parallel vs. sequential

One of the most impressive features of the human brain is its massive parallelism. Although each neuron computes at the millisecond scale (much slower than CMOS transistors which function below the nanosecond), the brain can perform 100 Tera Operations per second, orders of magnitude more than artificial neural networks on conventional computers. Parallel computing is a much studied topic beyond the scope of neuromorphic computing. However, parallel computing in conventional computer architectures is quite limited. Approaching the parallelism of the brain will require drastic changes in computer architectures. Moreover, it will require low power components so that they can all function simultaneously. Indeed, in current processors, the whole chips cannot function simultaneously because of power budget.

In-memory computing vs. von Neumann architecture

Conventional computers rely on the von Neumann architecture, where memory and computing are physically separated. In consequence, a large part of the energy consumption and delays are due to the transfer of information between memory and computing parts, a phenomenon often referred to as “von Neumann bottleneck”. In neural network algorithms, this issue is critical because huge numbers of parameters need to be stored and frequently addressed. The brain is extremely different in this regard: memory and computing are completely intertwined. The neurons, which compute, are connected by synapses, which carry the memory. Neuromorphic computing aims at bringing memory and computing together to achieve “in-memory computing”.

In-memory computing is being made possible through the development of emerging nanoscale memory devices. Various classes of such memories exist and will be discussed in this roadmap. Their common assets are that they are non-volatile, fast and low energy, can be read and written electrically and can be monolithically integrated into CMOS chips.

Plastic vs. rigid

Learning in the brain is made possible by its plasticity. The connections between neurons – the synapses – are not rigid but plastic, which means they can be modified. Learning, both in the brain and in artificial neural networks

algorithms, corresponds to repetitive modification of the synapses until reaching a set of connections enabling the neural network to perform tasks accurately. In conventional computers, this is done by explicit modification of the memory banks storing the weights. Neuromorphic computing aims at building systems where weights are self-modified through local rules. Here again, the role of non-volatile memories intertwined with computing circuits is critical. Their dynamics makes it possible to implement bio-inspired learning rules. For instance, memristors can implement Spike Timing Dependent Plasticity, a bio-inspired rule for unsupervised learning.

Analogue vs. digital

Conventional computers rely on digital encoding: voltages in the processor at the steady state only take two values, which represent 0 and 1. Transient intermediary values do not represent anything. All numbers are coded in binary, as a string of 0 and 1. In the brain, this is not the case. The electrical potential at the membranes of neurons can take continuous values, and so can the synaptic weights. Reproducing such behavior with digital encoding takes large circuits. Thus, using directly an analogue encoding will improve efficiency. Neuromorphic computing aims at using components which intrinsic analogue behavior mimics the key functions of neurons and synapses. For neurons, this can be achieved by CMOS transistors used in an analogue regime and by emerging technologies such as spintronic nanodevices or photonics. For synapses, which also require non-volatility, emerging memories are a key enabler.

Dynamic vs. static

Conventional computers use the steady state of their circuits to encode information. On the contrary, the brain is a complex dynamic system. Biological neurons are non-linear oscillators that emit spikes of voltage. They are coupled to each other and capable of collective behavior such as synchronization. There are also some indications that the brain functions at the critical point between order and chaos. Neuromorphic computing aims at emulating such dynamic behavior in order to go beyond the possibilities of static neural networks, in particular regarding learning. Here again, it is key to have circuits and components which intrinsic analogue dynamics emulates neural functions. Coupled oscillators can be achieved with CMOS ring oscillators, spintronic devices, metal-oxide sandwiches, photonics devices etc.

Spiking vs. clocked

Conventional computers are run by a clock which sets the pace of all circuits. There is no such clock in the brain. Neurons emit and receive spikes in an asynchronous way. Neuromorphic aims at building computers built on these principles. By having activity only when necessary, energy consumption will be reduced.

Stochastic vs. exact

Conventional computers aim at very high precision, coding numbers in 64 bits floating point precision. In the brain, this is far from the case as biological environment is noisy and neurons and synapses exhibit variability and stochasticity. Resilience to such imprecision seems to be a key asset of neural

networks. There are even suggestions that the brain uses noise for computing. Relaxing the constraints on the exactitude of components and computing steps will decrease energy consumption. Obtaining accurate results with approximate computing components and steps is a goal of neuromorphic computing. This will be crucial to be able to use components in their analogue regime, where noise and variability are more significant.

Applications “pull”

Neuromorphic computing is both of scientific and practical interest. This illustrated by the fact that both academics and industrials (from large groups to start-ups) are active in the field.

By definition, neuromorphic computing should provide solutions for problems where the brain is particularly efficient. Neuromorphic computing does not aim at replacing general computing. Rather, neuromorphic computing will be used in specialized chips that work together will general purpose chips.

Here, we provide an overview of the most promising applications of neuromorphic computing. To select these applications, we have solicited the Work Groups – in particular the Industry group – as well as the Forum participants, both by email and during the panel discussion of the Forum. These answers complete the results of the internal discussions of the consortium.

Artificial Intelligence on the edge
Neuromorphic computing will provide systems capable of running state of the art artificial intelligence tasks – deep neural networks – while consuming little power and energy. This opens the way to the deployment of artificial intelligence on the edge and in embedded systems, where consumption and size are critical.
Key applications are:

- Detection (always-on sensor processing, very low latency and low power, ~10mW)
- Recognition (could be triggered by ultralow-power detection, power: ~0.1mW)
- Situation awareness (semantic map of the environment, needs to be stored and updated online)

Sensor processing
“Smart” sensors currently still rely heavily on computing centers where they send raw data to be processed and sent back. The ability for sensors to process information on site without data transfer would provide faster response as well as better security and privacy.
Neuromorphic computing could in particular be useful for the observation of sensory signals and decision to trigger further processing or an action made on the edge (bio-signal monitoring, fall detection, voice detection, etc.).
Neuromorphic computing will be a key enabler of an efficient and secure internet of things.

Health

Health is a field that is currently being transformed by neural networks, for instance for classifying tumor images into benign or malignant. Neuromorphic computing could bring further benefits, in particular for processing dynamical signals and time series. On example of application is ECG online evaluation.
The potential of neuromorphic computing for low power, small size chips performing artificial intelligence tasks can revolution biomedical sensors: implants could be capable of performing real time complex monitoring.
In health, the importance of data privacy is huge, making on-site processing of information even more critical.

Robotics

A natural application for neuromorphic computing is robotics. In particular, it could give rise to agile, compliant robots with Human-Robot Interaction (HRI) capabilities such as:

- Learning dynamical models
- Coordination of behavior
- Force control

Merging health with robotics is full of applications for neuromorphic computing. Smart pills capable of action in the body and prosthetics are two key examples.

Optimization

Artificial neural networks use learning to solve large optimization problems. This has many applications outside what is usually thought of as cognitive tasks. This includes:

- Complex systems with many parameters
- High performance computing
- Thermodynamic simulation (which involves massive matrix-multiplication tasks which could be accelerated similar to NCT)

Natural language processing

Neuromorphic computing has the potential to process natural language and perform tasks such as translation and interpretation. It will be able to process speech in real time, from the raw dynamical data, to the reasoning on the extracted meaning.

Personal assistants

Combining different applications of neuromorphic computing such as optimization and natural language processing will lead to more efficient personal assistants. These will be capable of time management and scheduling, but also of assistive robotics and care, in particular for elders.

Autonomous vehicles

Combining robotics, sensory processing, optimization, and potentially natural language processing, autonomous vehicles has a strong need for neuromorphic computing. Many large industrial groups are working on the topic. One critical limitation of the autonomous car is the power consumption and size of the computing systems it relies on (several kW and a large space in the truck).

Smart manufacturing
Industrial machines and processes can benefit greatly from neuromorphic computing. Optimization of a whole process or fabrication chain is one example. Robotics applications of neuromorphic computing will make fabrication more efficient. Neuromorphic computing can also provide solutions for anomaly detection in time series, automatization of controls and tests, design for manufacturing, defect detection and forecast, predictive maintenance of machines etc. These will make industry more sustainable.

Computational neuroscience
Neuromorphic chips will be privileged systems to simulate biological neural networks. Thus, they could contribute to understanding the brain. This would bring massive novel knowledge but also provide new treatment for neurological diseases. It might also bring some light on how to achieve general intelligence.

Technology: state of art and directions

The slow-down in the scaling of CMOS transistors (often referred to as the “end of Moore’s law”), combined with the fact that the requirements of neuromorphic computing completely differ from conventional computing systems, have called for the use of new technologies for building neuromorphic chip.

The involvement of novel technologies brings opportunities for neuromorphic computing, both in terms of functionalities (such as dynamical systems or memories) and efficiency (power consumption, size, speed etc.).
However, many of these technologies are not at the same maturity level as conventional digital CMOS transistors, which is a challenge for the development of neuromorphic chips, both for industrials and academics.

Here we review the major technologies used for neuromorphic computing. In this first draft of the roadmap, we have identified the major technologies and key points to evaluate the assets and drawbacks of these technologies.

We list here the major technologies, from the most mature to the most exploratory.

Digital CMOS

Analogue and mixed-signal CMOS

Phase-change memories

Resistive switching memories (filamentary, oxram)

Spintronics

Optics

For each technology, the final roadmap will provide the following information, from experts in each subfield:

1) Describe your technology in a few sentences
2) Give one or two recent major results involving your technology
3) Give the important numbers for your technology.
   - Size of a neuron
   - Size of a synapse
   - Endurance of devices
   - Retention?
   - Size of on chip network that could be built
   - For ‘read’, ‘write’, ‘MAC’ operations (and other relevant) please provide:
     - Energy of operation
     - Power consumption
     - Speed of operation
4) Give one to five key advantages of your technology, compared to others.
5) Give one to five challenges for your technology
6) How mature is your technology (widely fabricated, commercialized)?
7) Give one or two major advances expected in the next years
8) Give one or two specific applications where your technology would be useful

Challenges for neuromorphic computing

In order to unlock its potential and provide the applications described above, neuromorphic computing must overcome several challenges. Discussions within the consortium, the work groups and at the forum have allowed us to come up with a list of limitations and challenges that neuromorphic computing currently faces.

Neuromorphic computing is mostly a recent field of study. Although some work had started in the early days of computing, the recent progress both in artificial intelligence and in emerging technologies has brought a new boom in neuromorphic computing. This has opened the door to many subfields, technologies and research directions. This novelty of the field also implies a lack of maturity, which comes with challenges that can be classified into four main categories.

Lack of theoretical foundations

Neuromorphic computing in general
There are no clear theoretical foundations for neuromorphic computing. It is neither clear how exactly the brain works, nor which aspects of this working should be emulated by neuromorphic computing.

Dedicated research on these topics and collaboration with neuroscientists must be conducted. These must keep in mind how to translate theoretical findings in usable hardware.

Learning

Learning is a crucial element of computing systems inspired from the brain. In software artificial neural networks, it consumes huge amounts of data, time and energy. Neuromorphic computing aims at finding better approaches. However, these are still lacking clear solutions.

In particular, neuromorphic computing aims at developing:

- Training approaches using only local information
- Training approaches with low bit precision available on the hardware
- Better training & optimization of spiking neural networks
- Data and power efficient online learning
- Better understanding of unsupervised learning (and 3 factor rules which can implement all kinds of learning)

Relationship to novel substrates and architectures

Neuromorphic algorithms and architectures must be co-designed with their substrate. Theoretical foundations on how to achieve this are lacking.

New bio-inspired concepts should be selected and optimized for their compatibility with electronic implementation.

Computational models for non-Von Neumann architectures (beyond neural networks, non-linear oscillator networks, Ising machines, optimizers, etc...). must be developed.

The scalability concepts and laws (equivalent of Dennard scaling for neuromorphic computing) are lacking and would be useful.

Lack of technological maturity

Novel technologies themselves

Technologies beyond CMOS transistors in the digital regime suffer from low maturity. Some examples of such issues are: variability in analogue CMOS circuits, lack of endurance in memristive switching devices, difficulty to achieve analogue non-volatile memories.

To address this issue, the community should work both on material and device development and on novel computational paradigms that function in spite (or even thanks to) the issues faced by emerging technologies.

Accessibility

Neuromorphic systems and devices are hard to access. The community should work on making hardware available, packaged for use, reliable and affordable. The development of versatile neuromorphic building blocks to be integrated into larger systems is a possibility.

Lack of standardized tools and benchmarks

Lack of whole stack from Hardware to Software

Conventional computing has benefited from multi-decade development of the stack from hardware to software. This is not yet the case for neuromorphic

computing. The different layers of the stack are not independent or well-defined. Knowledge of the whole stack is important to develop neuromorphic systems. Working on the maturity of the stack would make it easier to address each issue and facilitate scaling up of systems to more complex networks and tasks.

Lack of tools for development
There are not yet standard tools for developing neuromorphic systems. For instance, having a tool comparable to TensorFlow for deep learning that could be used for spiking neural networks would be of great use.

Lack of benchmarks and targeted applications
Neuromorphic computing is not necessarily efficient for the same applications as conventional software neural networks. New standards applications and benchmarks are still lacking for neuromorphic computing. Corresponding datasets are also lacking.

Lack of solid community

A strong and well-identified community is critical for a scientific field of study, especially as new and growing field. In the case of the neuromorphic computing, this need is especially important but also complex to achieve. This is due to the heterogeneity and interdisciplinarity of the field. Neuromorphic computing brings together actors from computer science, neuroscience, physics, electronic engineering, material science and more. Academics, industrials and SME are involved. Such a diversity is a huge opportunity for the field both on the scientific and human sides. However, it requires special effort to make people from such different background communicate and collaborate.

While community networks and events can self-organize in more narrow and mature fields, this should not be expected for neuromorphic computing, where a conscious action is needed. The Neurotech consortium and the resulting events and actions are a first step. As a striking example, many forum participants confided that this was the first neuromorphic computing event they had the opportunity to attend. More educational materials are also required to keep the community up to date with developments that are not in their core expertise, as well as to involve new actors.

It is critical that such actions continue to be encouraged, both at the individual and institutional levels.

Needs for adoption by industry

Despise the importance and large span of applications for neuromorphic computing, a number of roadblocks need to be overcome in order to achieve adoption by industry.

Applications

The community need to find some “killer apps” that will demonstrate the potential of neuromorphic computing.

These demonstrations should highlight the fact the neuromorphic chips are competitive with existing solutions and in particular software based deep neural networks.

Finding these applications requires:

- Interactions between research actors and end users.
- Increasing performance (e.g., TOPS/mm2 or TOPS/W) by orders of magnitude for conventional neural networks (CNN, LSTM, FC, ...)
- Clear benchmarking of existing and proposed solutions, close to real applications.
- If neuromorphic computing cannot compete with software neural networks in general, finding areas where it can/
- Performing demos on niche tasks.

Maturity

Neuromorphic computing is still technologically immature. Steps to make it more usable will require:

- Increasing the technology readiness level of the beyond-CMOS technologies
- Improving our understanding of neuromorphic computing to avoid a black-box situation
- Definition and theorization of algorithms and computing/programming paradigms that use neuromorphic computing. For instance, spiking neural networks for performing engineering tasks.
- Improving the scalability of devices and architectures

Ease of use

In order to be adopted by industrials beyond pure research and development, neuromorphic computing should be easy to use. This requires:

- More tools and infrastructure for development and debugging
- Development of reliable compiler software stacks
- Design of user-friendly GUIs that can help end-users to write neuromorphic networks, such as spiking neural networks, that performs practical tasks.
- Training people to have knowledge of the whole stack (materials, devices, systems, algorithms, applications)

- Providing easier access to existing systems and platforms
- Developing user-friendly development kits and methods for easier training and programming
- Tacking the large amount of data needed for the training by developing systems that require less data and making more data available

- Catering to the development of communities to make skill transfer and collaboration easier.

Actors and roles

Research labs
RTOs
Industry (technology providers)
Industry (users)
Start-ups

- explore new ideas, novel technologies
- taking risks
- build proof-of-concept demos for new application