POLYN
TECHNOLOGY
NEUROMORPHIC ANALOG SIGNAL PROCESSING

Neuromorphic Analog Implementation for Tiny AI Applications

White Paper

TABLE OF CONTENTS

INTRODUCTION

The recent shift from cloud to edge computing has changed the sensor node paradigm, from a simple Digital Signal Processor (DSP) near the sensor, to the use of deep neural networks at the sensor level. Of the approaches invented in this field during the last five years, all are targeted at energy optimization and computing latency reduction.

A major advantage of a neural network is its ability to do things in parallel. With traditional computers, the processing is sequential; when one task is completed, the next begins. The best way to implement parallel computation is neuromorphic computing, a method of computer engineering in which elements mimic the systems in the human brain and the nervous system. The term refers to the design of both hardware and software computing elements. The human brain processes information in parallel, which is many orders of magnitude more energy-efficient than any digital processor.

To achieve the advantages of parallel computation, it is necessary to move away from the algorithm-based consecutive computation of digital computers, and find new computing architectures that can perform tasks as efficiently as the human brain.

STATE OF THE ART OF NEURAL NETWORK IMPLEMENTATIONS

An algorithm is always a step-by-step execution within a Von Neumann and further architectures (even if there are many processor cores, each one processes the information sequentially). Hence, the fundamental difference between a neural network and an algorithm is parallel computation. The neural network model supposes parallel data distribution on all neurons of one layer, and further, instant data propagation to another layer.

An error in the algorithm code always leads to a system error. Errors in one or several connections in a neural network do not lead to failure, and all connections are interdependent, work in parallel, and are duplicated on multiple levels.

DIGITAL NEURAL NETWORK IMPLEMENTATIONS

A digital neural network is a model with simulated neurons, using standard step-by-step consecutive math operations in the digital processor core. Digital implementation is not well suited to neural network processing, since it cannot process massively parallel data, which is optimal for neural network computation. Digital neural networks can be implemented on traditional processors, but with much less efficiency.

There has been significant progress in digital neural network implementation during the past 20 years. It has mainly focused on parallel processing developments resulting in a certain level of parallelism of GPU and TPU solutions. However, the energy consumption problem has not been resolved.

The first hardware targeted at efficient neural networks was the Graphic Processor Unit (GPU), which allowed highly parallel computation. Soon after that, specialized Tensor Processor Units (TPUs) were developed to better process highly parallel neural network-based algorithms. Since TPU was deliberately developed for neural network inference, over time the number of processing units was increased, in comparison with the GPU, and the number of instructions per core was reduced. Despite all improvements, there are disadvantages to using a GPU and TPUs; namely, the intensive exchange with memory. This leads to high power consumption and high latency of inferences. Increasing the number of processing units adds to the power budget as well.

For a better understanding of neural network processing efficiency, we can look at the human brain, where all neurons in each layer work in parallel, as in an analog scheme. The human brain, with its 80 billion neurons, consumes just 20 watts of power, in comparison to the many kilowatts required for cloud-based neural networks. An analog neuromorphic design is required for efficient neural network implementation.

WHY THE ANALOG NEUROMORPHIC MODEL IS BETTER THAN DIGITAL

Neuromorphic analog computing mimics the brain's function and efficiency by building artificial neural systems that implement «neurons» and «synapses» to transfer electrical signals via an analog circuit design. This circuit is the breakthrough technology solution to the Von Neumann bottleneck problem. Analog neuromorphic ICs are intrinsically parallel, and better adapted for neural network operations.

An analog neuromorphic model of a single neuron with a fixed resistor as weights representation is better than digital model, because *N* multibit MAC operations in analog essentially require *N+1* resistors, while in digital, they normally require *N·(10~40)* transistors per bit, or *N·(80~320)* transistors for 8-bit precision. Therefore, the analog neuromorphic model is fundamentally superior in terms of single-neuron performance. Any current disadvantages of analog circuits are not fundamental, and are only related to some engineering challenges.

The most important advantage of digital solutions is the possibility of reusing a single computational block (i.e. an ALU or a hardware multiplier) many times while feeding it new weights and data.

Though this approach dramatically reduces the chip area, it creates a memory-related bottleneck in power, as the data transfer becomes more energy-consuming than the computations themselves.

Current efforts are being undertaken to make analog neural network processors mostly utilize the in-memory computing approach.

IN-MEMORY APPROACH

In-memory computing addresses the memory problem, but because it puts all the weights on a chip in crossbars, it suffers from a large chip size.

Another problem of in-memory computing is poor area utilization. On one hand, the memory array cannot be too small (in this case, the memory cells overhead will dominate the area, which is not efficient). On the other hand, the memory crossbar is an all-to-all connection, but the possible number of inputs for each neuron is limited by noise. So, if the array is too big, most memory cells are useless, which creates a significant area overhead. In practice, memory utilization isn't higher than 40-50%, even in cases when the network is optimized for the hardware, and such hardware optimization significantly constrains the network architecture (typical arrays are limited to 256-512 cells in width and height, while reasonably small networks may easily have 2,000-4,000 neurons in a single layer).

Another limitation of in-memory computing is limited precision, a measure of quality. The SRAM is 1 bit per cell. Existing Flash memory is up to 4-5 bits per cell.

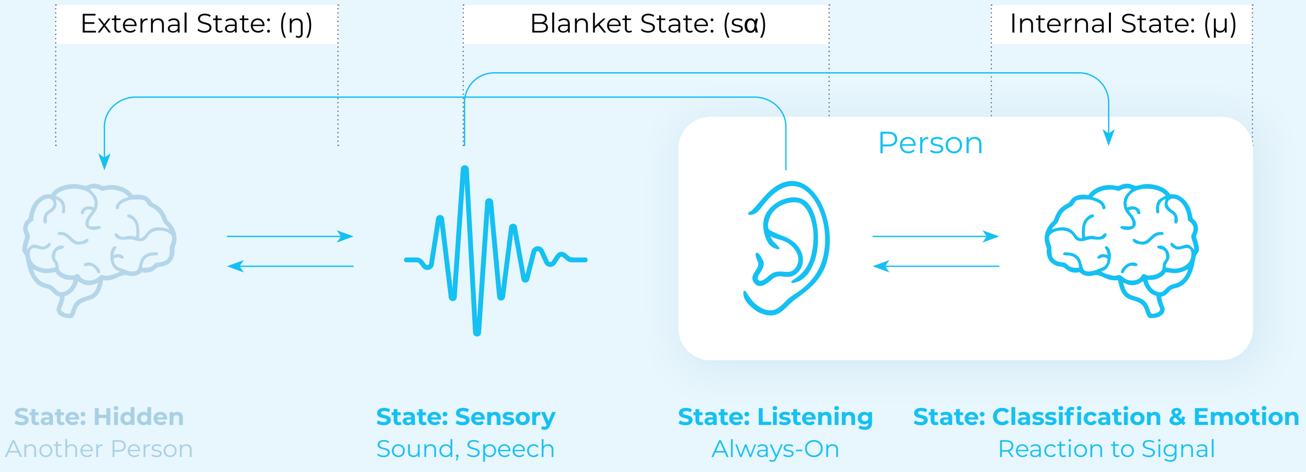
Other types of in-memory options (MRAM, PCM, ReRAM, FeRAM) rate low in TRL (technology readiness level) and don't promise multibit production-level solutions in the next few years.

NEUROSCIENCE AND AI DESIGN

Karl Friston, a leading scientist in the area of neurophysiology and AI, invented the Free Energy Theory, which proposes that brain connections minimize entropy by means of making representations that predict sensory signals. The model of the environment is built on the basis of sensory information and its interpretation. More information input results in a more complex model of the environment. The Friston concept does not limit itself to brain operation and is valid for any AI system.

According to neurobiological research, the retina, visual nerve, and some areas of the neocortex are fixed at a very early age, and do not change for the entire human life. The same is true for the auditory system (see below), which is the sensory system for hearing. It includes both the sensory organs (the ears) and the auditory parts of the sensory system. The active state (always-on) located in our ear detects and extracts sounds, which the brain interprets.

Voice perception process:



The picture demonstrates human perception of the voice according to Karl Friston's theory: there are blanket states of the peripheral cortex (the human retina and ear), reflecting direct perception of information, which are always on, and transfer information to the brain hemispheres for classification. *Blanket* states are formed after birth and in early childhood, and do not change during a human's life; they have a fixed neuron connection structure. The output coming from those *blanket* states in AI terminology is called «embeddings».

Embeddings are representations containing densely packed information about sensory input formed by a neural network or biological nervous system. Embeddings are formed in hidden layers of a neural network, and contain the most significant information about input data. Embeddings are used as input data for further efficient processing, classification and interpretation.

THE NASP (NEUROMORPHIC ANALOG SIGNAL PROCESSING) CONCEPT

NASP technology perceives raw data signals to add «intelligence» to various sensors. The architecture contains artificial neurons (the nodes performing computations) and axons (the connections with weights between the nodes) implemented using circuitry elements: neurons are implemented with operational amplifiers, and axons using thin-film resistors.

The NASP chip design embodies the approach of a sparse neural network with only the necessary connections between neurons required for inference. In contrast to in-memory designs, where each neuron is connected to each neighboring neuron, the NASP approach simplifies the chip layout. This design especially suits Convolutional Neural Networks (CNN), where connections are very sparse, as well as RNN, Transformers, and Autoencoders.

The NASP converts the already trained and optimized neural network math model into the chip structure, offering area utilization close to 100% (and it can be exactly 100% in practice), and 8 bits per weight with current technology. This approach yields a faster time to market, lower technical risks, and better performance. Furthermore, NASP proposes a **hybrid core approach,** similar to that described in the latest neuroscientific works on human brain data processing.

THE NASP HYBRID SOLUTION APPROACH

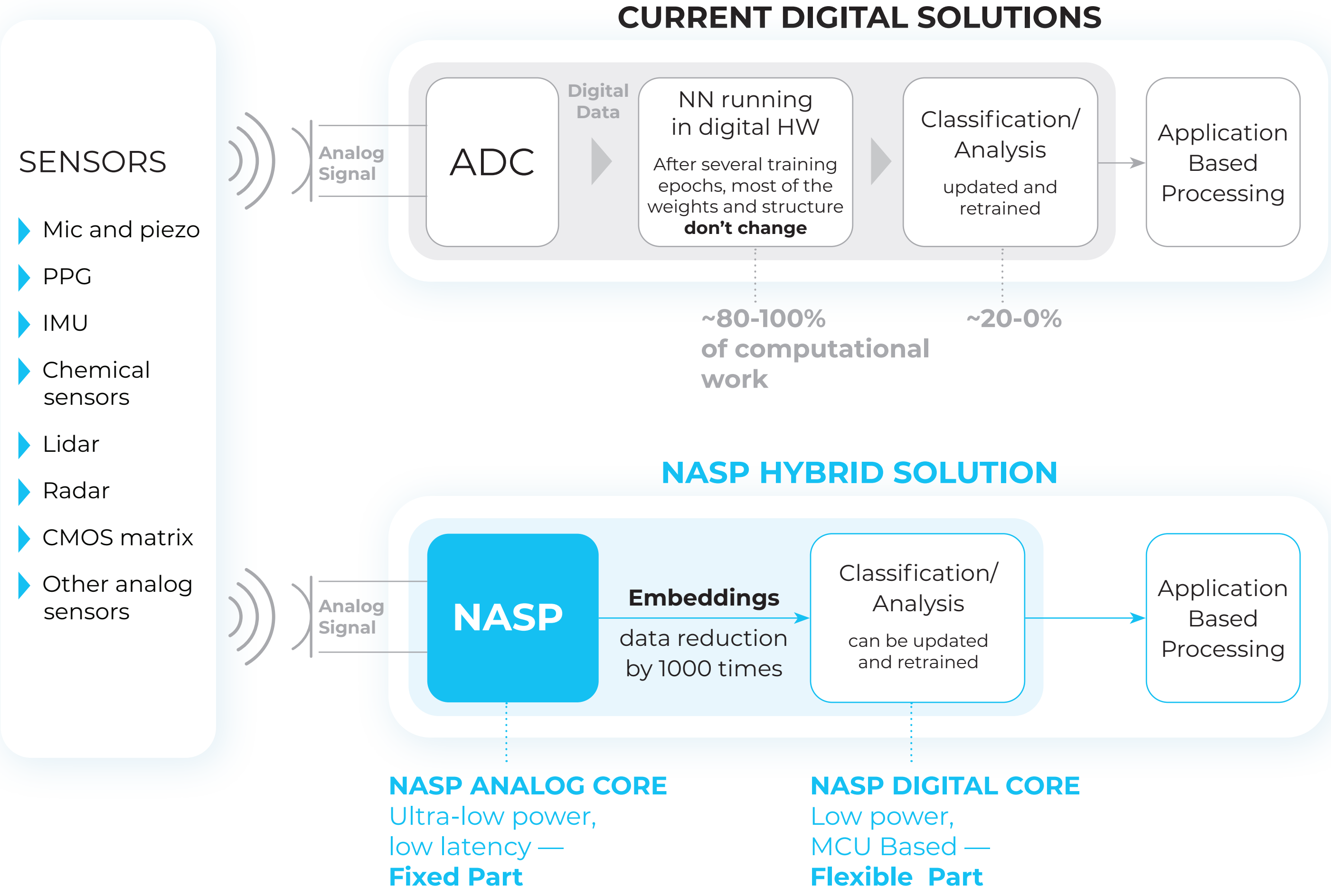
NASP technology combines the fixed weights method, which implies complete separation of inference and training, with a fixed chip structure, similar to the human visual nerve and retina, and a flexible function (can differ depending on application) responsible for further classification of the received embeddings.

There is a well-known phenomenon of Machine Learning (ML): after several hundred training cycles (also known as epochs), the deep convolutional neural network maintains fixed weights and the structure of the first 80-90% of the layers, and in the following cycles, only the few last layers responsible for classification continue to change weights.

This property is also used in the Transfer Learning Technique. This fact is the key to a hybrid concept, where a combination of fixed neural networks is responsible for pattern detection, combined with a flexible algorithm (which could be any type, including an additional flexible neural network) responsible for the pattern interpretation.

The fundamental parts of the **NASP Hybrid Core** concept:

- A fixed Neuromorphic Analog Core — ultra-low power and low latency, generating embeddings.
- A fully flexible digital core for final classification.



A neural network implementation with Transfer Learning technique in a digital standard node vs the NASP hybrid solution that uses the analog circuit.

The neural network (NN) running on digital processors (CPU, GPU, and TPU) allocates resources in the following way:

1. Raw data pre-processing consumes about 80% of the computational resources
2. Classification and decision making consume much fewer resources.

The NASP solution uses the principle of Transfer Learning where the most layers of a neural network responsible for raw data preprocessing (1) remain unchanged after a certain number of training epochs (Fixed Analog Core), and only last few layers (2) are updated while receiving new data and retraining (Flexible Digital Core).

NEURAL NETWORK PRUNING

Another important element of the NASP solution is effective pruning, in order to minimize the trained network, which is to be converted into chip production files. Since the neural core runs with weights, fixed after training of the neural network, we can effectively prune the initial neural network before converting it into an internal math model in a digital format. Pruning can reduce the neural network 2 to 50 times, depending on its structure. And since the final chip is built according to the prepared architecture and structure, pruning drastically reduces the final chip size and power consumption.

NASP PERFORMANCE

Below is a comparison of energy per inference metric for a NASP chip vs some digital solutions.

	NASP	RASPBERRY PI3 B+	SNAPDRAGON-710	JETSON TX1
MobileNet V.2 (joule/inference)	2,5 x10^{-3}	3,25	0,72	0,2

Raspberry PI3 B+ has one processor, Snapdragon has a processor with a GPU accelerator, and Nvidia Jetson has a 256 Core GPU. For NASP, we have used a chip simulation (D-MVP) model and data from the NASP Test Chip.

The comparison demonstrates orders of magnitude more energy efficiency of NASP, which is due to the analog neuromorphic nature of the IC. Also, it is important that the NASP chip size completely fits the requirements of the neural network (Mobile Net V2), and there is absolutely no overhead in NASP. It is especially important for small neural networks, where the NASP efficiency advantage becomes much greater.

SELECTED USE CASES

HUMAN ACTIVITY RECOGNITION

The embedding extraction and processing approach in Human Activity Recognition is based on 3-axis accelerometer signals. A trained autoencoder neural network encodes various types of human activities, and then decodes them without any loss of accuracy. After generating such an array of patterns (embeddings), an analyzing neural network recognizes the human activities encoded in the patterns.

The system consists of the encoder function, implemented in fixed neurons in analog, and the classifier, implemented in a digital processor. The fixed analog part of the NASP chip takes approximately 90% of the whole workload, and the activity recognition interpretation takes around 10%, resulting in a low load on the digital subsystem.

An important feature of the use of such embeddings generated by the encoder neural network is that, if a human practiced some new physical activity (which was not trained previously during the neural network training stage), the unique descriptor would be formed anyway, differentiating this activity from other classes of embeddings. Thus, it would be a totally new compact size pattern, dedicated to that activity. In other words, it is still possible to introduce new classes not included in the fixed trained network. Thus, it significantly broadens the application of NASP-based products.

PREDICTIVE MAINTENANCE (PDM)

There is a huge data flow generated by vibrational sensors that measure machinery, tracks, railway cars, wind turbines, and oil and gas pumps, to be transferred wirelessly to analytic equipment. This data flow shortens the battery life of operating sensor nodes. A NASP solution reduces the data flow from vibration sensors 1000 times, using the same encoder-decoder approach, and transmitting through LoRa (or another low power technology) only embeddings extracted from the initial data. It is worth noting that the autoencoder systems and embedding will create new classes, describing new signals of vibration sensors, even if they were not trained to recognize these types of signal patterns.

Thus, by applying an encoder neural network, in this case, rigidly built-in NASP, we can obtain the whole range of different vibration signals from various vibration sensors, which can be then analyzed by a digital system to recognize machine malfunctions. The use of the embeddings reduces data sent to the cloud, solving the fundamental problem of low bandwidth required by IoT systems.

CONCLUSION

The Neuromorphic Analog Signal Processing (NASP) technology, described here, provides the optimal answer to the power consumption and computing latency challenges by dividing the solution into fixed and flexible parts. The fixed part is analog; and only a relatively small output, based on embeddings, is sent to flexible digital processing for analytics. The use of embeddings is specific to NASP. It allows combining advantages of the fixed weights part of the NASP chip, and flexible weights in a digital co-processor. The use cases, presented here, help to evaluate the possibilities of the NASP technology for IoT sensory systems.