POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Learning in Analog Spiking Neural Network with Floating Gate Synapses in Standard CMOS Technology

Laurea Magistrale in Electronic Engineering - Ingegneria Elettronica

Author: Giovanni Camisa

Advisor: Prof. Giorgio Ferrari

Co-advisor:

Academic year: 2021-2022

1. Introduction

The rise in general usage of electronics is closely tied to an increase in the power and miniaturization of electronic components. This trend has proven to be true for more than fifty years However, the high number of devices per unit surface area combined with the high operating frequency required for the top processors leads to intolerable power dissipation, which is even worse considering the high operating frequency required for the top processors in the market today. Furthermore, current systems are almost all based on the Von Neumann architecture, in which the computing unit and the memory unit are physically separated. The main problem lies in the different performance evolution of the CPU and memory units, with the former achieving far more speed than the latter. To solve these issues new technologies able to process the data directly where they are stored have emerged during the last two decades. In this context, bio-inspired computing is of increasing interest. It imitates the biological process happening in the brain and replicates the neuron-synapse interaction for locally computing and storing the information. More specifically, Spiking Neural Networks (SNNs), i.e. a neural network based on the processing of spikes, have become the most promising method to solve machine learning-based problems due to their biological and hardware plausibility and reduced complexity compared to Artificial Neural Networks (ANNs), i.e., a neural network based on the processing of numbers. In particular, the SNNs are the best candidate for real-time processing near the sensors, i.e., edge computing, because they can be implemented on extremely power-efficient dedicated hardware. However, the hardware implementable training algorithms for SNNs are still too immature to compete with ANN performance on real-world applications. After performing a comprehensive analysis of the learning techniques, this thesis work proposes an online training for SNNs compatible with a CMOS implementation based on floating-gate synapses. The simulations and the analysis have been based on a neuromorphic chip in standard CMOS technology designed and implemented during previous thesis works [2] [4] [3].

2. Description of the CMOS spiking neural network

The neuromorphic chip was realized in order to gather together the bio-plausible feature with a low power implementation. The chip (Fig. 1) was entirely made with CMOS technology and is composed by three main circuits: the neuron circuit, the synapses circuit and the STDP circuit.

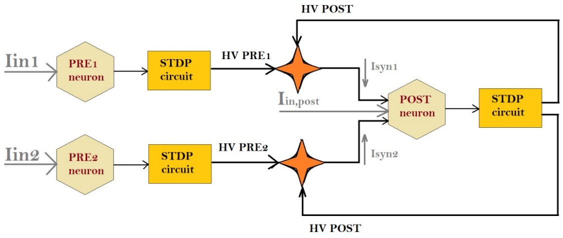

Figure 1: Block scheme of the neuromorphic chip. The neurons have been arranged in order to have two presynaptic neurons and one postsynaptic.

2.1. Neuron circuit

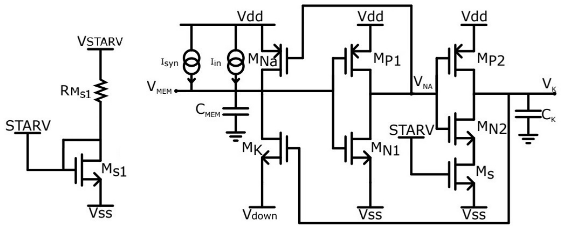

Figure 2: Circuital scheme of the neuron and the starver bias.

There are three neuron circuits on the chip: two predisposed as presynaptic neurons and one as postsynaptic. Both the presynaptic neurons are linked to the postsynaptic by a synapse.
The neuron circuit (Fig. 2) is able to emulate the main states of a biological neuron: the resting state, the depolarization state, the repolarization state and the hyperpolarization state. When C_{mem} charges due to the input current and crosses the threshold of the inverter composed by M_{P1} and M_{N1}, a positive feedback is triggered, which leads to the complete closing of M_{Na} and $V_{mem} = V_{dd}$. At the same time the commutation of the first inverter (M_{P1} and M_{N1}) makes the second inverter commute (M_{P2} and M_{N2}), activating a negative feedback. This latter feedback is delayed with respect to the positive by C_K and forces the discharge of C_{mem} for a period of time determined by the bias of the current generator M_S. In order to consume as less power as possible the circuit was powered between $V_{dd} = 0.4V$ and $V_{ss} = 0V$. V_{down} is about 0V and controlled with an external trimmer for testing.
This neuron can be catalogued as a Leaky Integrated and Fire (LIF) neuron model.

2.2. The synapse

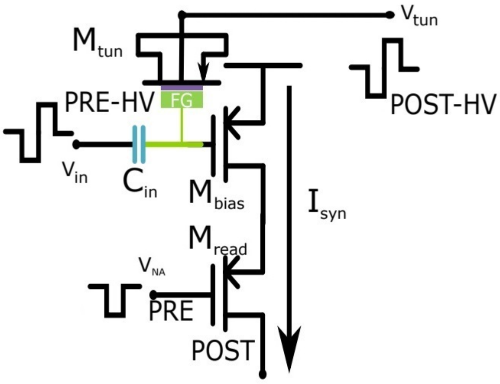

Figure 3: Circuital scheme of the floating gate synapse composed by three PMOS transistors.

The synapse has the role of storing the strength of the connection between two neurons. A spike of the pre-synaptic neuron injects a current into the capacitance C_{mem} of the postsynaptic neuron based on the charge stored in the floating gate. The synapse is composed by three pMOS transistors (Fig. 3): two of $3.3V$ technology (M_{tun} and M_{bias}) and one of $1.8V$ technology (M_{read}).
When a presynaptic neuron spikes a negative pulse (the V_{Na} signal from the presynaptic neuron, Fig. 2) is applied at the gate of M_{read}. The pulse has a width of $1ms$ and turns on M_{read} generating a current controlled by the amount of charge stored in the floating node FG given by C_{in} and the gates of M_{tun} and M_{bis}. The V_{in} and V_{tun} are high voltages (up to 5V) controlled by the presynaptic and postsynaptic neurons, respectively. These voltages can activate a tunneling current in the gate oxide of M_{tun} for injecting or ejecting charge from the floating node following a proper learning rule. PMOS transistors have been specifically employed for the floating gate to assure that each tunneling transistor would have its own bulk to prevent charge sharing and cross talk between synapses.

2.3. The STDP circuit

The STDP (Spike Timing Dependent Plasticity) circuit is in charge of generating high voltage signals that activates the synapse tunneling. Each spike produces a voltage that starts from 0V, goes down to -4.5V for a selectable time (from a few ms up to hundreds of ms), then goes up to +4.5V for a second independent selectable time, and finally comes back to 0V.

3. Experimental results

To properly bias and test the chip, a dedicated PCB has been built [4] with a total of twelve BNCs and twelve trimmers. Different experiments have been carried through to validate the performance of this implementation with its relative advantages and problems.

3.1. STDP characteristic

The previously chosen method to train the network is the pair based Spike Timing Dependence Plasticity. This training method belongs to the unsupervised paradigm and can perform competitive learning and a Winner-Take-All approach. To understand if it was possible to implement a bigger network with this kind of circuit and training, the effective STDP characteristic has been performed. The trimmer voltages have been specifically chosen to ensure biologically plausible timings. The presynaptic neuron 1 has been forced to trigger twice, while the postsynaptic neuron one time. The first spike of the presynaptic neuron 1 was synchronized with different delays with the one of the postsynaptic neuron to force a change of the synaptic weight. The second spike of the presynaptic neuron was used to trigger the synapse and measure its weight change. The resulting STDP characteristic (Fig 4 **a**) respect to the simulated one (Fig 4 **b**) is shifted in time by $\approx 20ms$. This effect, probably due to the high variability of the technology, makes the algorithm treat causal spike pairs, up to $\Delta t \approx 20ms$, as anti-causal. With this characteristic it would be impossible to perform competitive learning. In fact, the pair STDP learning performs a WTA learning that increases the synaptic weights of the presynaptic neurons that contribute most at the spiking of the postsynaptic neurons (i.e. causal relationship). The measured STDP would address as anti-causal the presynaptic neurons that contributes most to the spiking of the postsynaptic neurons ($0ms < \Delta T < 20ms$) and thus would penalize those synapses and support more the synapses linking presynaptic neurons that contribute less at the spiking of the postsynaptic neurons ($\Delta t < 40ms$).

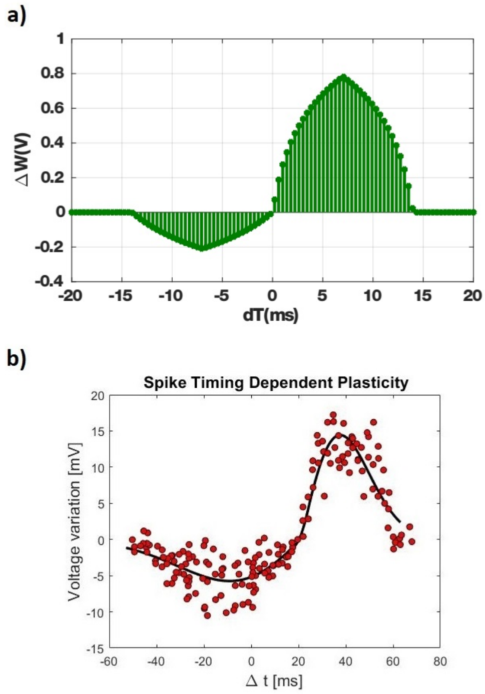

Figure 4: **a**) Expected STDP simulated in [2]. ΔW is the floating gate voltage. **b**) STDP characteristic acquired from measurements. The Δt range is from $-50ms$ to $68ms$ with 3 measurements performed after each $2ms$. The black line represents the interpolated curve.

3.2. Power consumption estimation

The major advantage of the neuromorphic architecture of the chip is the high power efficiency. To estimate the power dissipation of the chip, another board has been assembled with the only difference of having the $0.4V$ and the $0.8V$ power supplies directly connected to a semiconductor parameter analyzer in charge of providing the required voltage and measure expected low currents. The measurements have been performed by configuring the trimmers and applying a constant bias to both the presynaptic neuron 2 and the postsynaptic neuron in order to make them spike at $\approx 91Hz$, i.e. 1 spike each $\approx 11.1ms$.
The resulting measurements estimate the real power consumption of a single neuron can be

estimated as $Power = 20pW$ and the real energy consumption for a spike $E_{spike} = 20pW \cdot 11.1ms/spike = 222fJ$. Note that this is an overestimation, because it includes also the power dissipation of the input current mirrors used to inject a current into the neurons. They require $\approx 40pA$ with a power dissipation of $\approx 16pW$ each. Although overestimated, these results are still impressive since a biological neuron consumes $100pJ$ per action potential and $10fJ$ per synaptic transmission [5]. Compared to a standard digital processor, we should consider that simple access to a 32-bit DRAM consumes about 0.64 nJ [6], three orders of magnitude more than a single neuron.

4. A new learning algorithm

In order to still manage to use this incredibly low power dissipation technology another training algorithm has been thought. A supervised learning was preferred because of the purpose of performing real time classification.

4.1. The algorithm

The choice of the algorithm is the result of a long study of the state of the art spiking neural networks training combined with the need of having effective classification that could be obtained in reasonable time. The decreed algorithm was originally developed by A. Renner, A. Sornborger et al. [1]. It is a mathematical optimization supervised training belonging to the family of surrogate gradient learnings. Since the algorithm in [1] was thought for a SNN without memory, it has been decided to adapt the algorithm to our model by imposing the need of multiple spikes from a single neuron to identify the backwards path that lead to the right classification and changing its pseudoderivative.
If we consider a four layer SNN implemented with a LIF neuron model:

$$o = f(W_3 f(W_2 f(W_1 x)))$$

$$f(x) = H(x - V_{th})$$

$$H(x) = \begin{cases} 0 & \text{if} \quad x < 0 \\ 1 & \text{if} \quad x \geq 0 \end{cases}$$

Where x is a binary input array, o the binary output array, W_1, W_2 and W_3 are the three synaptic weight matrices, $f(\cdot)$ is the spiking activation function and $V_{th} = 200mV$.

The algorithm in question can be expressed as:

$$d_3^{th} = (o - t) \circ f'(W_3 h_2^{th})$$

$$d_2^{th} = sgn(W_3^T d_3^{th}) \circ f'(W_2 h_1^{th})$$
$$d_1^{th} = sgn(W_2^T d_2^{th}) \circ f'(W_1 x^{th})$$
$$\frac{\partial L}{\partial W_l} = d_l^{th} (a_{l-1}^{th})^T$$
$$W_l^{new} = W_l^{old} - \eta \frac{\partial L}{\partial W_l} \qquad l = 1, 2, 3$$

Where t is the target output, d_l^{th} are the surrogate backward propagated local gradients, which represent the amount by which the loss function L changes when the activity of a neuron changes in a certain amount of time. $\circ$ represent the elementwise product between vectors, $sgn(x)$ is the sign function, and a_l^{th} denotes if the activation of the layer l i.e. $f(W_l a_{l-1})$ with $a_0 = x$, $a_1 = h_1 = f(W_1 x)$, $a_2 = h_2$, $a_3 = o$ happens more than an N_{th} amount of times when the same data is presented to the network for a given time. The only hyperparameter of the algorithm is the learning rate η. Finally, $f'(\cdot)$ is the pseudo derivative of the activation function, which is the derivative of the following activation function:

$$f_{surrogate}(x) = \frac{1}{V_{th}} tanh(\frac{x}{V_{th}} - V_{th})$$

Whose derivative can be calculated as:

$$f'(x) = 1 - tanh^2(\frac{x}{V_{th}} - V_{th})$$

4.2. Binary HAR dataset

The dataset used to evaluate the performance of the training algorithm (HAR dataset) consist of approximately 1.600.000 raw data samples coming from a 3-axis accelerometer installed on a phone. The data represent the instantaneous accelerations measured by the accelerometer and have been labelled into two different classes: still and walk. The dataset scope is to be able to identify if a person is walking or staying still only by using the output of the accelerometers and a neural network.
100.000 random data have been selected from the dataframe and encoded in order to be compatible with a SNN. In a similar way to how ADC works, the accelerometer data was quantized using a limited number of bits. However, differently from an ADC the acquire value would

not be converted into a binary number, but instead there would be a neuron representative for each LSB. For each input value, only the neuron associated with the nearer LSB produces a spike. The encoded data have been divided into a training set (80%) and a test set (20%) to train and test the network.

4.3. SNN simulations

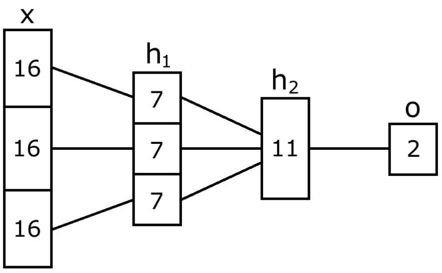

Figure 5: 48-21-11-2 network feedforward connections. The neuron clusters connected by straight lines are fully connected.

A 48-21-11-2 has been simulated and trained with the HAR test set encoded with the LSB encoding. The input layer is not fully connected to the first hidden layer, but the input neurons corresponding to each axis (16 neurons for 4bit coding) are only connected to 7 neurons of the hidden layers (Fig 6). The synaptic weight initialization has been sampled from a random distribution still centered in 0 and with $1.1V_{th}$ variance. The order of the input data was randomly chosen and the same input was presented to the network for $220ms$, i.e. the time equivalent of 11 input neuron spikes. During this period, the spiking activities of the output neurons were monitored and each $20ms$ the error and equivalent weight updates would be calculated ($output = 1$ if there was a spike, 0 otherwise). The N_{th} chosen to trigger the backpropagation is 3 (spikes). Furthermore, it was chosen a batch of 1000 data (1/8 of the training set). The weight updates would be summed and normalized by $N_{batch} = 1000$ and then performed:

$$W_l^{new} = W_l^{old} + \frac{\eta}{N_{batch}} \sum_{k=1}^{N_{batch}} \left(\frac{\partial L}{\partial W_l} \right)_k$$

The network has proven able to fit the data and achieve convergence in less than 10 training cycles (Fig. 6)

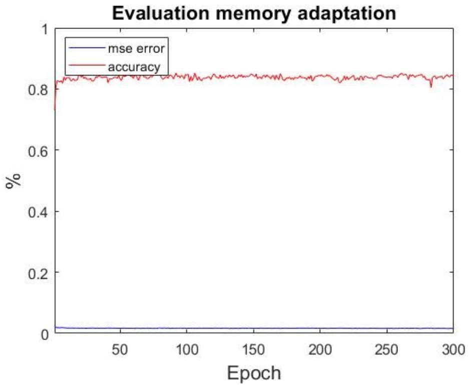

Figure 6: Evaluation of 48-21-11-2 network trained the algorithm adapted to force the use of memory. The network almost instantly converges to the minima.

4.4. Shallow ANN and SNN comparison

4.4.1 Accuracy comparison

To properly evaluate the performance of the algorithm to train the SNN, multiple networks have been trained with the same sizes and the same learning rate. The final accuracy on training set and test set achieved after 300 training epochs have been compared with the final accuracy achieved by a standard Shallow ANN.
The Shallow ANN is a fully connected network trained with the HAR dataset, simple enough to be implemented on a microcontroller. It has three layers (3-3-2): the first two uses the ReLu as activation function, while the last one uses the logistic function. It has been trained with the Adam optimizer on a binary crossentropy loss function with a batch size of 1000 data for 500 epochs. This process has been repeated for 500 times with the original HAR dataset and 500 with a 4bit quantized version of it (similar to the LSB encoding) to evaluate the effect of the quantization of the dataset. The resulting accuracy variability obtained on the test sets (Fig 7 **a** and **b**) shows that there is not much difference in the distributions, since they are both centered around 78% accuracy and have the same variability. The only visible effect is that the probability of achieving accuracy greater than 83% for a network trained with the quantized dataset is lower than the one of a network trained with the original dataset. The maximum accuracy achieved with the original dataset is 0.838%, while with the quantized dataset it is 84.2%.

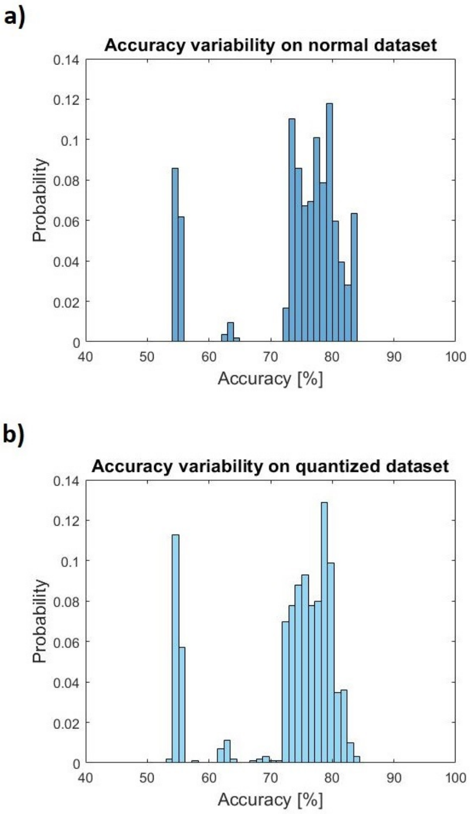

Figure 7: **a**) Probability distribution of the accuracy achieved by the Shallow ANN after being trained 500 times with the original HAR dataset. Accuracy calculated on the test set. **b**) Probability distribution of the accuracy achieved by the Shallow ANN after being trained 500 times with the quantized HAR dataset. Accuracy calculated on the test set.

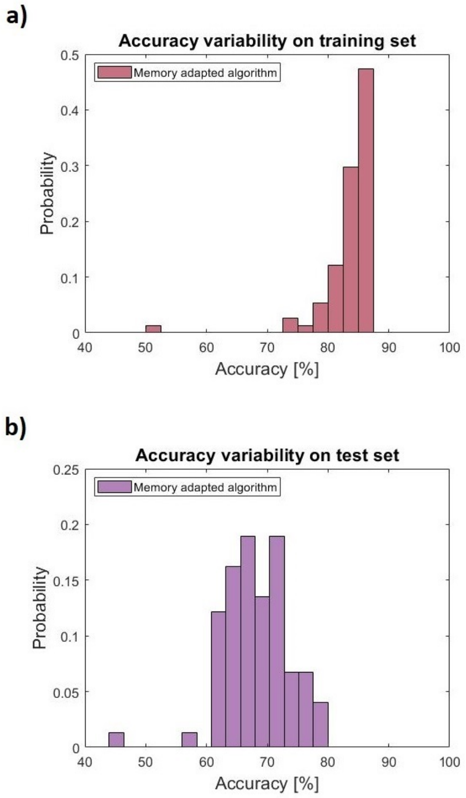

Figure 8: **a**) Probability distribution of the accuracy achieved by the SNN after being trained 70 times with the proposed algorithm. Accuracy calculated on the training set. **d**) Probability distribution of the accuracy achieved by the SNN after being trained 70 times with the proposed algorithm. Accuracy calculated on the test set.

The SNNs have been trained 70 times. The variability distributions of both test and training set (Fig 8 **a**, **b**) show that the algorithm can achieve and higher accuracy overall on the training set comparable with the higher accuracy achieved by the the Shallow ANN. Furthermore, the highest accuracy achieved by the SNN on the test set is higher than the mean accuracy achieved by the Shallow SNN. The maximum accuracy achieved by the SNN on the training set is 86.2%, while on the test set is 79.1%.

4.4.2 Power comparison

From the simulation results and the measurements, it has been possible to estimate the power dissipation of a SNN implemented with the fully CMOS-compatible neurons and synapses discussed in section 2 and trained with the previous algorithm. The simulations show that a 48-21-11-2 SNN trained with the algorithm spikes on an average mean of 58 times to catalogue an input data, with a minimum of 46 and a maximum of 75 spikes. Considering the worst case scenario of 75 spikes occurring simultaneously during inference, the total power dissipated by the network to catalogue an input data would be of $P_{neurons} \approx 75{\cdot}20pW = 1.5nW$ (high overestimation, considering that not all the neuron spikes at the same time and considering the power overestimation in section 3.2) and an energy consumption of $E_{neurons} = 75 \cdot 222fJ = 16.65pJ$ (high overestimation, see section 3.2 for power evaluation). It has not been possible to measure the exact power dissipation of a synapse, however it was estimated its value by comparing the data obtained form the measurements with the data obtained from the technology simulations [2]. The overestimated power dissipation of the synapse is $P_{synapse} \approx 50pW/spike$ and an energy consumption of $E_{synapse} \approx 22,2fJ/spike$. If it is considered that in the worst simulated scenario of 75 total spikes the input neurons spiked 33 times, the hidden layer 1 neurons spiked 20 times, the hidden layer 2 neurons spiked 17 times and the output neurons spiked 5 times, it can be estimated a total dissipation of $P_{synapses} = (33 \cdot 7 + 20 \cdot 11 + 17 \cdot 2)50pW = 24,3nW$ (overestimation, considering that not all the neuron spikes at the same time) and $E_{synapses} = (33 \cdot 7 + 20 \cdot 11 + 17 \cdot 2)22fJ = 10,67pJ$. This would lead to the estimation of a total dissipation of $P_{inference} = 1.5nW + 24,3nW \approx 26nW$ and $E_{inference} = 16,65pJ + 10,67pJ = 27.32pJ$. Since the start of art commercial solution (LSM6DSOX by ST) based on a dedicated machine learning core, consumes $P_{micro} = 1.8V \cdot 4uA$ for the same classification task, it can be concluded that the simulated SNN implemented in standard CMOS technology would dissipate 2 orders of magnitude less. Furthermore, this SNN, differently from a microcontroller, would consume energy only during inference, since without spikes the power consumption of the network is practically negligible.

5. Conclusions

The aim of this thesis work was to develop a suitable supervised online-learning that could be implemented on an analog spiking neural network with floating gate synapses in Standard CMOS technology and able to achieve state of the art performance. After understanding the limitations of the previously designed neuromorphic chip of a CMOS-based spiking neural network a new suitable training algorithm solution for this technology have been provided. The simulation shows that the algorithm can achieve similar performance to ANNs. Furthermore, the estimated power and energy dissipation of the new network would be in the nW range and thus orders of magnitude less than an implementation based on a standard digital processor.

6. Acknowledgements

I would like to acknowledge professor Giorgio Ferrari for his patience, dedication and engagement while being my advisor for this thesis. Without his guidance and support, all of this would have been impossible. Thank you professor.

References

[1] A. Zlotnik L. Tao A. Renner, F. Sheldon and A. Sornborger. The backpropagation algorithm implemented on spiking neuromorphic hardware. *arXiv*, 2021.

[2] Michele Mastella. Analog spiking neural network with floating gate synapses in standard cmos technology. Master's thesis, Politecnico di Milano, 2019.

[3] Cristina Polidori. Analog spiking neural network with floating gate synapses in standard cmos technology. Master's thesis, Politecnico di Milano, 2021.

[4] Elisabetta Polidori. Analog spiking neural network with floating gate synapses in standard cmos technology. Master's thesis, Politecnico di Milano, 2021.

[5] Anders Sandberg. Energetics of the brain and ai. *arXiv*, 2016.

[6] T. Yang V. Sze, Y. Chen and J. S. Emer. How to evaluate deep neural network processors: Tops/w (alone) considered harmful. *IEEE Solid-State Circuits Mag.*, 12(3):28–41, 2020.