Convolutional neural networks with radio-frequency spintronic nano-devices

NEUROMORPHIC
Computing and Engineering

PAPER

Convolutional neural networks with radio-frequency spintronic nano-devices

Nathan Leroux*, Arnaud De Riz, Dédalo Sanz-Hernández, Danijela Marković, Alice Mizrahi* and Julie Grollier

Unité Mixte de Physique CNRS/Thales, Université Paris-Saclay, 91767 Palaiseau, France
* Authors to whom any correspondence should be addressed.

E-mail: n.leroux@fz-juelich.de and alice.mizrahi@cnrs-thales.fr

Keywords: neuromorphic computing, spintronics, deep convolutional neural networks, radio-frequency, nano-devices

OPEN ACCESS

RECEIVED
9 November 2021

REVISED
13 April 2022

ACCEPTED FOR PUBLICATION
10 June 2022

PUBLISHED
1 July 2022

Abstract

Convolutional neural networks (LeCun and Bengio 1998 *The Handbook of Brain Theory and Neural Networks* 255–58; LeCun, Bengio and Hinton 2015 *Nature* **521** 436–44) are state-of-the-art and ubiquitous in modern signal processing and machine vision. Nowadays, hardware solutions based on emerging nanodevices are designed to reduce the power consumption of these networks. This is done either by using devices that implement convolutional filters and sequentially multiply consecutive subsets of the input, or by using different sets of devices to perform the different multiplications in parallel to avoid storing intermediate computational steps in memory. Spintronics devices are promising for information processing because of the various neural and synaptic functionalities they offer. However, due to their low OFF/ON ratio, performing all the multiplications required for convolutions in a single step with a crossbar array of spintronic memories would cause sneak-path currents. Here we present an architecture where synaptic communications are based on a resonance effect. These synaptic communications thus have a frequency selectivity that prevents crosstalk caused by sneak-path currents. We first demonstrate how a chain of spintronic resonators can function as synapses and make convolutions by sequentially rectifying radio-frequency signals encoding consecutive sets of inputs. We show that a parallel implementation is possible with multiple chains of spintronic resonators. We propose two different spatial arrangements for these chains. For each of them, we explain how to tune many artificial synapses simultaneously, exploiting the synaptic weight sharing specific to convolutions. We show how information can be transmitted between convolutional layers by using spintronic oscillators as artificial microwave neurons. Finally, we simulate a network of these radio-frequency resonators and spintronic oscillators to solve the MNIST handwritten digits dataset, and obtain results comparable to software convolutional neural networks. Since it can run convolutional neural networks fully in parallel in a single step with nano devices, the architecture proposed in this paper is promising for embedded applications requiring machine vision, such as autonomous driving.

1. Introduction

Convolutional neural networks have widely contributed to the success of artificial intelligence since LeNet-5 [3] outperformed other deep neural networks [2] for handwritten digits recognition. Due to their invariance to translations and local distortions, convolutional neural networks not only excel in image recognition, but also in signal processing [1], and they are at the core of artificial intelligence applications like generative adversarial networks [4]. For this reason, it is crucial that novel technologies, whose goal is to decrease the energy consumption of artificial intelligence, implement convolution neural networks efficiently [5–7]. Hardware developed to accelerate the multiply-and-accumulate operations of neural networks, such as crossbar arrays of memristors, initially focused on realizing fully-connected layers wiring all the input neurons to all the output neurons with independent synaptic weights [8, 9], an operation that they can achieve in a single step.

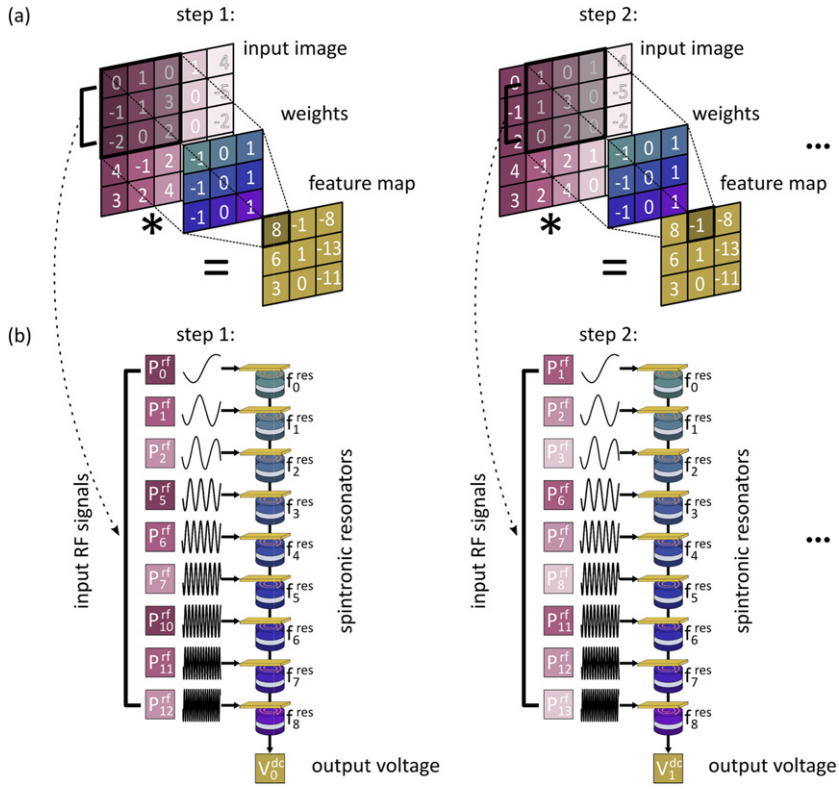

Figure 1. (a) Example of a 2D convolution with an input image of size 5 × 5 and a filter of size 3 × 3. Each element of the output feature map is the sum of the element-wise matrix multiplication (multiply-and-accumulate operation) between a subset of the input image and the weights of the kernel. The colors of the input image pixels indicate to which RF input signal they correspond in (b). (b) Schematic of the corresponding sequential convolution with RF signals and a chain of spintronic resonators. At each step the microwave powers of the input signals correspond to a mapping of a subset of the input image. The RF signals are injected to resonators through field-lines, represented by yellow stripes. Spintronic resonators are represented with colors corresponding to their matching weights, themselves represented in (a). At each step, the output voltage is a multiply-and-accumulate operation between the input microwave powers and the weights encoded into the resonance frequencies of the resonators.

Convolution layers, however, have a different operating principle. To perform a convolution on input data, multiply-and-accumulate operations with fixed synaptic weights (filters) are applied to consecutive subsets of the input. This process is often performed sequentially, as the filter is sled over neighboring inputs, as illustrated in figure 1(a). In addition, the convolution operation has to process several input channels, and different filters need to be applied in order to compute different output features. The whole process has therefore a strong sequential character that requires storing intermediate computation steps in memory, which, for convolutions, has a prohibitive cost in terms of energy consumption, speed, and area. Finding ways to eliminate this sequential nature and implement convolutional neural networks in a fully-parallel manner, so that they can process their inputs in a single step, is therefore of great interest.

A wide span of research investigates hardware neuromorphic implementations of convolutional layers using CMOS [10–12], memristor crossbar arrays [13–17] as well as optics and photonics [18–20]. A particular effort aims at unfolding each convolutional layer into a sparse matrix of synaptic weights and mapping it to a crossbar array of memories to process convolutions fully in parallel [10, 13, 14, 17]. Due to the sparsity of convolutional layers, these implementations can suffer from leakage currents from unused cells. A solution to mitigate these leakage currents is to use three-dimensional memristor circuits, in which the number of lithographic layers increases with the convolutional layer size [21]. We present here a 2D implementation of parallel convolutions that could lead to a simpler fabrication process. Spintronics, whose memory, multifunctionality and dynamics are attractive for information processing is another technology actively studied for neuromorphic computing. Spintronic devices present advantages for the integration since they are compatible with CMOS and the same technology can be used to implement both artificial neurons and synapses [22, 23]. Research shows that arrays of spintronic memories are promising for associative memories [24], spiking neural networks [25] and convolutional neural networks with time-domain computing [26]. However, implementing in parallel all the multiply-and-accumulate operations of a convolution requires extremely large crossbar arrays. Such a crossbar array of spintronic memories for parallel convolutions would particularly suffer from sneak-path currents [27] due to the small OFF/ON ratio of spintronic memories [28]. In addition, such crossbar would consume a

lot of energy because of the low resistance of spintronic memories, as discussed in [29]. A different approach is therefore required.

A way to overcome the sneak-path in spintronic devices arrays is to use spintronic devices that communicate through radio-frequencies such that the path of the information is not solely determined by electrical connectivity, but also constrained by selectivity in the frequency domain [30–32]. Frequency selectivity can be achieved by encoding the input information into radio-frequency signals of different frequencies, and using spintronic nano-resonators as artificial synapses that rectify frequency specific signals [33, 34]. Here we describe how we can perform several multiply-and-accumulate operations in parallel without the sneak-path currents arising in crossbar arrays of spintronic memories thanks to the topology of our architecture and the frequency selectivity of spintronic resonators. It has already been demonstrated that chains of spintronic resonators can natively implement multiply-and-accumulate operations on radio-frequency (RF) signals [30, 31]. This is promising to directly classify RF signals sensed from the environment with an antenna, as well as to establish communication between layers of neurons through RF signals rather than through wiring only. Furthermore, the RF multiply-and-accumulate operation has been demonstrated experimentally in a small system [34], and a simulated fully-connected perceptron has achieved classification as well as a software perceptron on an image benchmark of handwritten digits [33].

In the present paper, we adapt the concept of multiply-and-accumulate operation with spintronic nano-resonators to convolutional layers which are large and sparse. We show that chains of spintronic nano-resonators are suitable for the implementation of convolutional layers in deep neural networks based on RF communications between layers of neurons and synapses. First, we show how chains of spintronic resonators can implement convolutions on different sets of input RF signals presented sequentially as in figure 1(a), and we show that this method integrates selectivity in the frequency domain. Then we show that it is possible to achieve these convolutions in a single step, with different chains implementing different multiply-and-accumulate operations, thus enabling ultrafast computation. We present how the resonators can be spatially arranged as a matrix of weights of an unfolded convolution, and propose a spatial arrangement that does not suffer from the sparsity specific to this type of matrices. We explain how this architecture can operate convolutions to extract different features in parallel, and how to assemble the spintronic chains performing MAC operations on incoming RF signals with spintronic nano-oscillators that emulate neurons and emit RF signals [11, 35–42] in order to build deep neural networks. We also show that it is possible to train simultaneously all the spintronic resonators implementing the same filter coefficient by tuning them all at once. Finally, we simulate a full convolutional neural network made of these RF spintronic nano-devices and demonstrate an accuracy of 99.11% on the Mixed National Institute of Standards and Technology (MNIST) handwritten digits dataset, the same accuracy obtained for a software network with an equivalent architecture.

1.1. Radio-frequency multiplications for spintronic convolutions

In convolutional layers, the input image is convolved with multiple filters. The specificity of convolutions is their efficiency to extract spatial features, and each filter aims at extracting a different pattern present in the input image. To do so, each filter (which is often much smaller than the input image) slides over the input image, and at each position applies a multiply-and-accumulate operation to the corresponding image subset (see figure 1(a)). Then the outputs, also called feature maps, store the result of the corresponding multiply-and-accumulate matrix operations (the sum of the elements of an element-wise matrix multiplication between the filter and a subset of the input image). In this section, we show how to perform these different multiply-and-accumulate operations sequentially using RF encoded inputs and a single chain of spintronic resonators for each filter. A parallelized architecture is presented in the next section.

Figure 1(b) shows a chain of spintronic resonators performing multiply-and-accumulate operations of a convolution. First, the intensity values of the input image pixels are mapped to RF powers corresponding to the pixel values of the image. Then, at each step, the corresponding subset of the input image is injected into the chain of resonators. For instance, during the first step an RF signal is injected into the first diode (f_0^{res}) with a power P_0^{rf} and a frequency f_0^{rf} corresponding to the first pixel. During the second step, the power in the same diode is changed to P_1^{rf}, corresponding to the second pixel. Each RF signal is injected through an individual field-line to one of the spintronic resonators of the chain [43]. Each resonator has a resonance frequency close to the frequency of its input RF signal. The spintronic resonators are employed in a 'spin-diode' mode in which they filter and rectify the RF signals that they receive (see methods for more details) [27, 28]. Here, the rectification is caused by the mixing between the alternative current induced in the resonator by the current injected in the field-line (through capacitive or inductive effects), and the resistance oscillation of the resonator induced by magneto-resistive effects as the magnetization driven by the alternative field oscillates. Since the resonators of the chain are connected electrically, if we used the same RF frequency for all the inputs signals, each resonator could mix with signals induced by different field-lines and rectify them, hence causing crosstalk

issues. On the contrary, here we choose a different frequency for each input signal to ensure that each resonator only mixes with the RF signal transmitted by its field-line, and thus rectifies the proper input signal. The dc voltages produced by the resonators are summed because they are electrically connected in series. The total voltage of such a chain of resonators is a multiply-and-accumulate operation between the input microwave powers and synaptic weights that are encoded in the resonance frequencies of the resonators [33, 34]. This operation is described in detail in the methods section. In this implementation, the difference between the input RF frequency and the resonance frequency of the resonators implements the weights of the convolutional filter. Since in a convolution, each multiply-and-accumulate operation requires the same set of weights, these resonance frequencies are left unchanged between the different steps. Then, at each step of the convolution, the voltage of the chain encodes a different element of the output feature map.

This method is straightforward, but it has the defect of being sequential. Doing these operations one after the other is costly in memory because it requires to store all the elements of the output feature map between two convolutional layers, and it slows down computing since it requires approximatively as many steps as there are pixels in its input images, versus a single one for parallel convolutions [16]. In the next section we describe how to operate RF convolutions fully in parallel.

1.2. Fully-parallel architecture for radio-frequency convolutions

To perform all multiply-and-accumulate operations in parallel, we propose a novel architecture using multiple chains of spintronic resonators. This architecture is represented in figure 2(b). Unlike the sequential method, here we simultaneously send all the elements of the input image to the resonators. Each pixel is mapped to a different RF signal with power proportional to its value, which is simultaneously injected into several spintronic resonators, each belonging to a different chain. A single field-line corresponds to a row in figure 2(b). The different resonators of the same chain are arranged in a column and correspond to different coefficients of the filter. For instance, the resonators of the first column correspond to the coefficients $w_{1,1}$, $w_{1,2}$, $w_{2,1}$, and $w_{2,2}$ (in that order) of the filter, as it is represented in the left-hand side of figure 2(a). The resonators of the second column correspond to the coefficients $w_{1,0}$, $w_{1,1}$, $w_{1,2}$, $w_{2,0}$, $w_{2,1}$, and $w_{2,2}$ of the filter, as it is represented in the right-hand side of figure 2(a). The resonators of the fifth column correspond to all the coefficients of the filter, from the 1st to the 9th. The resonance frequencies of resonators in a row all match the corresponding input RF signal they need to rectify, but are not identical as they encode different synaptic weights: as the filter is sled, the same input gets multiplied by a different weight (see figures 1(a) and 2(a)). In figure 2(b) each spintronic resonator is represented with a color that corresponds to one of the synaptic weights, which themselves are represented in figure 2(a).

In memristor crossbar arrays, each memristor is placed at a crosspoint between input lines and output lines. Sneak-path currents between crosspoints are then possible [8, 21, 27, 44–46]. Here, since we inject the input signals in the spintronic resonators through field-lines electrically separated from the array, there are no current paths between the different chains of spintronic resonators. Moreover, the frequency selectivity of the spintronic resonators prevents them from being influenced by signals from neighboring field-lines.

We have to tune the resonance frequencies of the resonators to change the synaptic weights they implement in order to train the network. In figures 3(a) and (b), all resonators implementing the same filter coefficient are represented with the same color. In the crossbar architecture studied until now, all the resonators that encode the same synaptic weights are aligned in a diagonal (see figure 3(a)). Alternatively, as shown in figure 3(b), we can change the spatial arrangement of the spintronic resonators, to implement a compacter architecture in which the resonators implementing the same filter coefficient are aligned in a column. Since they are aligned, we can tune simultaneously the resonators coding for the same synaptic weight with a single write-line. Write-lines provide an electrical control of synaptic weights either by changing the state of memristors placed above each spintronic resonator as it was done in [47] or in [48], or by switching the magnetization of spintronic resonators between two states [49–54] such as in binary neural networks [55–59]. Independently of the control method, a physical implementation of a network with the proposed architecture can be trained with a number of field-lines that does not scale with the number of devices, but only with the number of synaptic weights per filter.

1.3. Multi-layer convolutional neural network implementation with chains of resonators and spintronic nano-oscillators

In the previous sections we presented convolutions with single channel images and only one filter per layer. In typical convolutional layers, the input image is convolved with multiple filters, that produce different feature maps. Each feature map becomes a different channel of the input image in the next layer, as shown in figure 4(b). Filters are 3D tensors whose depth is equal to the number of channels of their input image. The pixel values in each of the N_m different feature maps resulting from the convolution of an input image with N_c

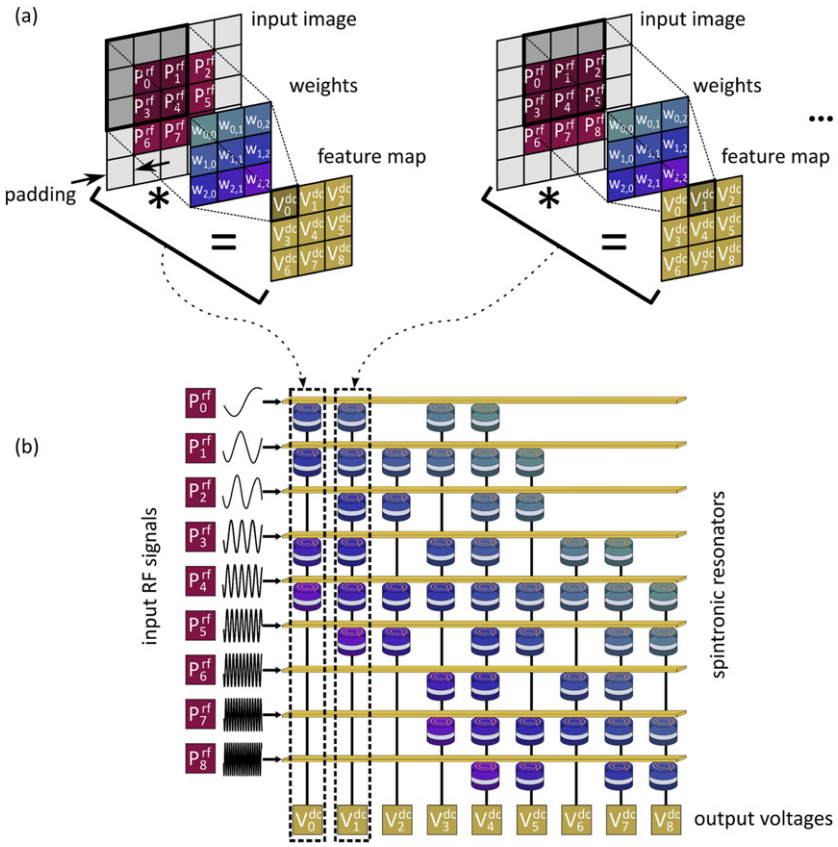

Figure 2. (a) Example of a 2D convolution with an image of size 3 × 3, a padding of 1 and a filter of size 3 × 3. Padding of 1 means that the image is padded at its outer edge with 1 layer of zeros. Each element of the output feature map is the sum of the element-wise matrix multiplication (multiply-and-accumulate operation) between a subset of the input image and the weights of the filter. (b) Schematic of the corresponding parallel convolution with RF signals and multiple chains of spintronic resonators. RF input signal powers are mapped to the image pixels. The RF signals are injected to resonators through field-lines, represented by yellow stripes. Spintronic resonators are represented with colors corresponding their matching weights, themselves represented in (a). Resonators encoding the same weights are aligned in diagonal. The voltage of each chain is a multiply-and-accumulate operation between a subset of the input microwave powers and synaptic weights encoded in the resonators' resonance frequencies.

channels with a filter of size $k \times k \times N_c$ are given by the formula:

$$z_{h,w,m} = \sum_{c=0}^{N_c-1} \sum_{j=0}^{k-1} \sum_{i=0}^{k-1} W_{i,j,c,m} x_{i+h,j+w,c} + b_m, \tag{1}$$

where W are the filter coefficients, x the input pixel values, b are the biases, m is the feature map index, h and w are the height and width positions of the pixel in the feature map, i and j are the vertical and horizontal coordinates of the pixel in the filter, and c is the input channel index. To make this operation fully parallel, additional resonators are employed as illustrated in figure 4(a). To implement different channels, additional sets of RF signals (in blue) are employed, which are sent to additional sets of spintronic resonators connected in series with the resonators rectifying the RF signals from the first channel (in red). Additional filters are implemented by adding new chains of resonators after the chains of resonators implementing the first filter, as in the right-hand side of figure 4(a). Similarly, multiple channels can be convolved with multiple filters with the compact architecture described in figure 3(b).

Building deep neural networks requires transferring information between different layers of neurons and synapses. We propose using spintronic oscillators as artificial neurons emitting RF signals as inputs to each convolutional layer. It was already demonstrated that these oscillators can be used as artificial neurons thanks to their nonlinear dynamics [35, 37, 38]. Here, we leverage the nonlinear transformation between the input direct current to the oscillators and the output microwave power they produce [35, 60, 61] (see methods). Using the fully parallel architecture, it is then possible to cascade the information between different convolutional layers: provided a voltage-to-current amplification, every direct voltage output of the convolution can be used to supply an artificial neuron of the next layer. Then the RF signals emitted by spintronic oscillators become the inputs of chains of spintronic resonators, these resonators rectify RF signals into direct voltage which supplies the next layer of spintronic oscillators, and so on. This architecture alternating between RF and DC signals

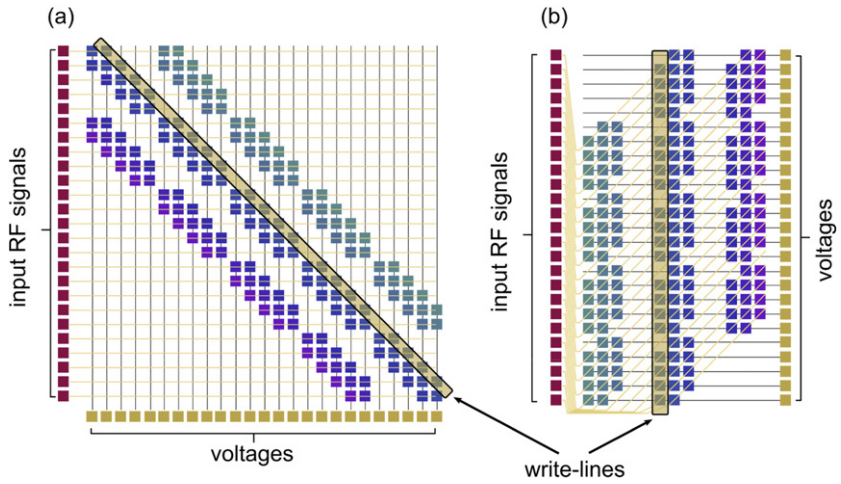

Figure 3. Schematics of a parallel convolution with an input image of size 5 × 5, a padding of 1 and a filter of size 3 × 3 with RF signals and chains of spintronic resonators implementing the synaptic weights. To simplify the figure, spintronic resonators are represented by colored squares in central panels and field-lines carrying RF signals are represented by yellow lines. Golden shade stripes represent write-lines that can simultaneously tune multiple spintronic resonators electrically. (a) Architecture similar to a traditional crossbar with input lines perpendicular to output lines. Spintronic resonators implementing the same synaptic weights are aligned in diagonals. (b) Compact architecture is a spatial rearrangement of synaptic components: unlike in traditional crossbars, output lines are parallel to input lines. Spintronic resonators implementing the same synaptic weights are aligned in columns.

is illustrated in figure 4(b). In practice, CMOS components are needed to convert voltages to currents and amplify them to match the threshold current of spintronic oscillators.

As in the two previous sections, a different frequency is assigned to each oscillator. Here again, because we use field-lines to transmit RF signals locally to specific resonators and because we rely on a resonant effect, each artificial synapse only multiplies its corresponding input. Thus, our implementation can perform convolutions in parallel using multiple resonator chains without sneak-path currents that can be problematic in memristor crossbar arrays [8, 21, 27, 44–46].

1.4. Handwritten digits classifications with a convolutional neural network implemented on spintronic nano-oscillators and resonators

In this section, we simulate a network with spintronic oscillators as neurons and spintronic resonators as synapses based on the proposed convolutional architecture. The goal is to prove that chains of resonators can calculate convolutional operations with high accuracy, that it is possible to tune the convolutional filter coefficients by tuning the resonance frequencies, and to demonstrate the capacity of spintronic oscillators to implement activation functions in such networks.

We benchmark our network on the standard MNIST dataset. It consists of 28 × 28 pixel images of handwritten digits. The topology of our network is shown in figure 5(a); 32 filters of 5 × 5 with stride 1 and padding 1 for the first convolutional layer, a max-pooling of size 2 × 2 and stride 2, a layer of spintronic oscillators as activation functions, 64 filters of 5 × 5 with stride 1 and padding 1 for the second convolutional layer, a second max pooling of size 2 × 2 and stride 2, a second layer of spintronic oscillators as activation functions, a fully connected layer of size 1600 × 10, and a softmax layer in the end. The physical analytical models used to simulate spintronic devices are detailed in the methods. It has already been proven experimentally in [27, 28] that fully-connected layers can be implemented by chains of spintronic resonators, applying multiply-and-accumulate operations on RF encoded inputs and the same operating principle is assumed here. Max-pooling layers and the softmax layer are assumed to be implemented by more classical technologies such as CMOS circuits.

To train the network, we use 60 000 images for training and 10 000 for testing. At each training iteration, a batch of 20 images is presented to the network, the output of the network is computed with the softmax layer, and the cost function is the cross-entropy loss [62]. We use the backpropagation algorithm [2] and Adam optimizer [63] of the software PyTorch to train the network. The learning rate is 10^{-4}.

In our implementation, the synaptic weights depend both on the frequency of the input and the resonance frequency of the resonators (see equation (7) in methods). The input frequency is kept fixed and the trained parameters are the resonance frequencies of the resonators. An additional constraint arises in the case of parallel convolutions due to weight sharing. Indeed, resonators corresponding to the same filter coefficient have to implement the same synaptic weight even if they receive input signals with different frequencies. In order to

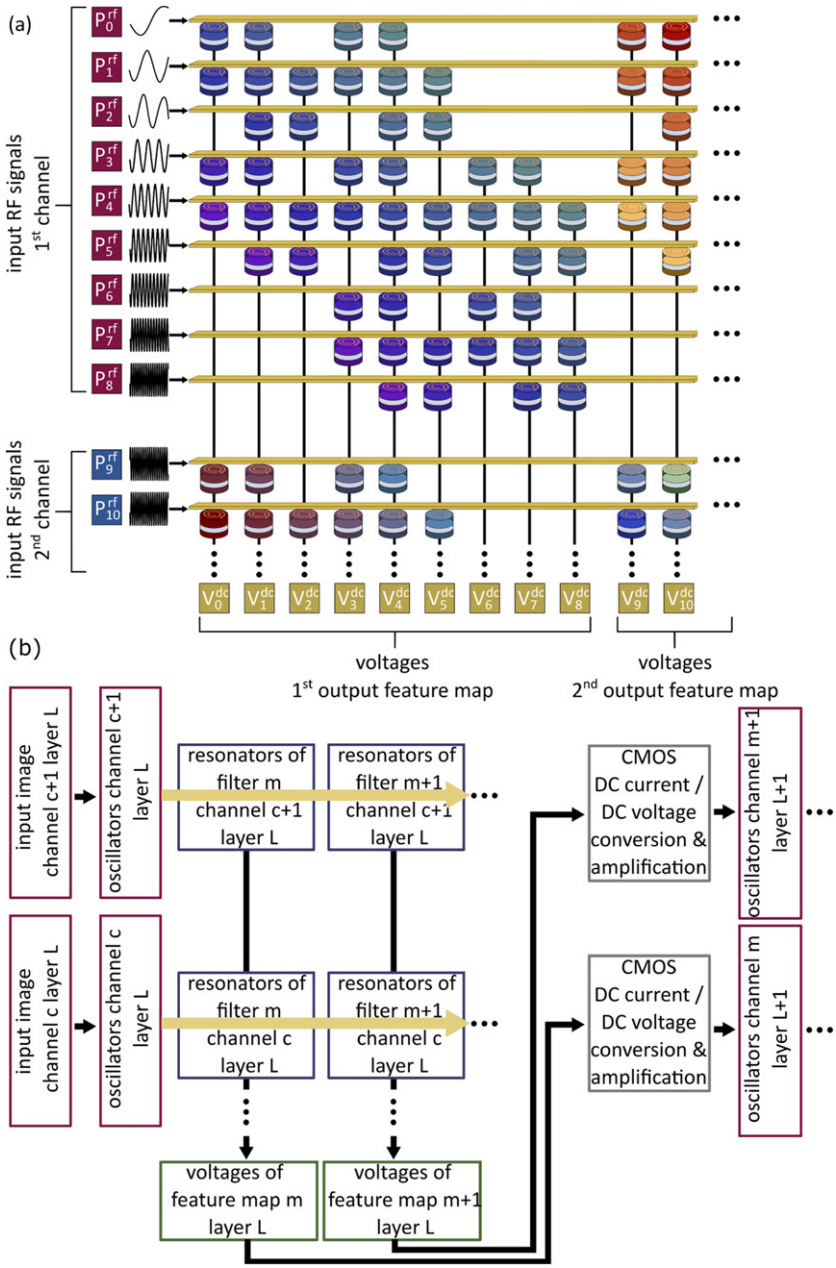

Figure 4. (a) Schematic of a parallel convolution with RF signals and chains of spintronic resonators implementing the synaptic weights similar to figure 2(b) but with different channels and features. First (second) input channel pixels are mapped to a first (second) set of RF signals represented in red (in blue). Chains of spintronic resonators corresponding to the outputs of the second feature map are parallel to the first set of chains corresponding to the outputs of the first feature map. (b) Diagram of a deep convolutional network with multiple channels and features with spintronic oscillators to emulate neurons and resonators to emulate synapses. Direct signal electrical connections are represented by black arrows while field-lines carrying RF signals are represented by yellow arrows.

ensure that this is the case, our training algorithm updates the resonance frequency of each resonator with a function that depends both on the frequency it receives, and a trainable parameter $\zeta_{i,j,c,m}$ learned through backpropagation that corresponds to its filter coefficient:

$$f^{\text{res}}_{i,j,c,h,w,m} \leftarrow f^{\text{RF}}_{h+i,w+j,c}\left(1-\zeta_{i,j,c,m}\right). \tag{2}$$

This expression indicates that when multiple spintronic resonators are tuned simultaneously with a single write-line in a hardware implementation, the resonance frequency update of each resonator should scale with its input signal frequency. In the methods section, we demonstrate that using this expression, chains of resonators voltages correspond to convolution outputs, described by equation (1). For the fully-connected layer, we simply set the resonance frequencies as trainable parameters learned through backpropagation, as it was done in [33].

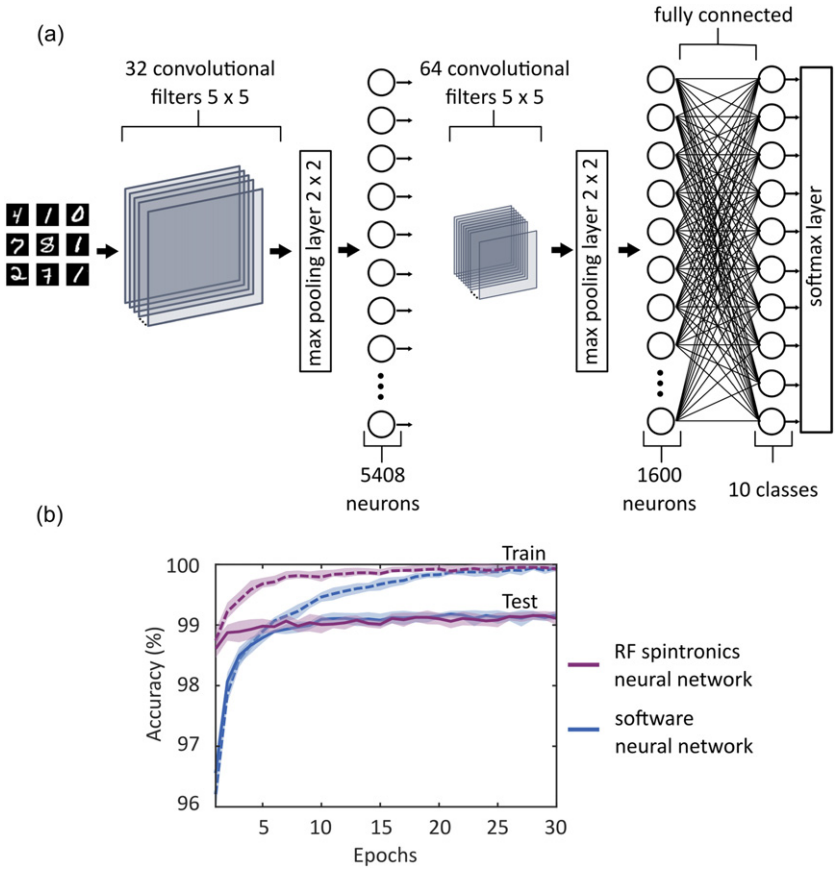

Figure 5. (a) Neural network architecture used to classify images of the MNIST dataset. (b) Accuracy of image recognition measured in percentage of successful classification, versus the number of epochs (the number of times we present the training dataset to the network to train it). The red (blue) solid line shows the mean accuracy for a simulated RF spintronic devices network (a software neural network) for ten repetitions on the test dataset. In green (black) solid line the mean accuracy for a simulated RF spintronic devices network (a software neural network) on ten repetitions on the training dataset. Shaded areas represent standard deviations.

To optimize the spacing between RF signal frequencies in the same neural layer in order to avoid crosstalk, we use a similar relation as in [33]:

$$f_{i+1}^{\mathrm{RF}} = f_i^{\mathrm{RF}} \left(\frac{1 + \frac{\Delta f_i^{\mathrm{RF}}}{f_i^{\mathrm{RF}}}}{1 - \frac{\Delta f_i^{\mathrm{RF}}}{f_i^{\mathrm{RF}}}} \right), \tag{3}$$

with Δf_i^{RF} the frequency linewidth of an oscillator. Using equation (3), the emission peaks of two oscillators are separated by their linewidth, which ensures that they do not interfere. The larger neural layer of the considered network (the input of the second convolutional layer) is made of 5408 spintronic oscillators. Tsunegi *et al* have shown that it is possible to make a spintronic oscillator with a quality factor of $\frac{f^{\mathrm{RF}}}{\Delta f^{\mathrm{RF}}} = 6400$ [64]. Using equation (3) with $\frac{\Delta f_i^{\mathrm{RF}}}{f_i^{\mathrm{RF}}} = \frac{1}{6400}$, we compute that we can arrange 5408 frequencies between 1 GHz and 5.4 GHz with spacings equivalent to the linewidths of the different oscillators. In conclusion, it is possible to arrange the frequencies of the oscillators in a reasonable range while avoiding crosstalk.

In experimental hardware implementations, signals should be amplified between a layer of spintronic oscillators and a layer of spintronic resonators. Therefore, in our simulations we introduce an amplification factor for each synaptic layer. Amplification factors are set as trainable parameters and are trained through backpropagation. Adjusting these parameters during training balances the fact that spintronic resonators can only be tuned within a finite range of synaptic weights.

We plot in figure 5(b) the learning curve of the network. We see that at the end of the training, our spintronic network (red solid line) has a mean accuracy of 99.11% on ten trials, with a standard deviation of 0.1%, as good as the 99.16% accuracy (with a standard deviation of 0.07%) we found with a software convolutional neural network with the same architecture (solid blue line). The difference in accuracy is smaller than the standard deviation, represented in shaded area in figure 5(b). In addition, the classification results are higher than with a multi-layer-perceptron [3], which indicates that the advantages of convolution are preserved with

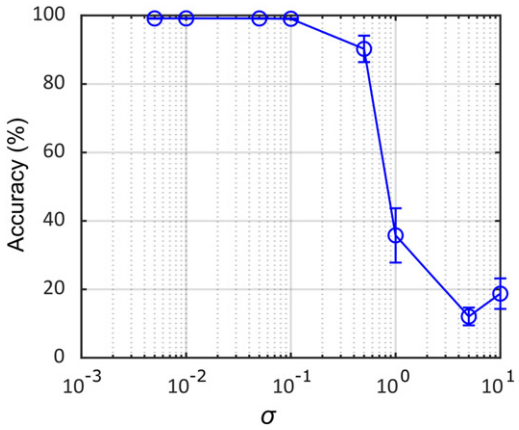

Figure 6. Accuracy of image recognition measured in percentage of successful classification on the MNIST test set, versus the standard deviation σ of variability of spintronic resonators resonance frequencies respect to the resonance bandwidth in log 10 scale. For instance, a variability of standard deviation $\sigma = 10^{-2}$ means that spintronic resonators have an initial standard deviation of 1% respect to the resonance bandwidth.

our RF spintronic network. These results show the feasibility of convolutions with RF signals and spintronic resonators.

In hardware implementation, spintronic resonators can have inter-devices variability. We simulate the variability of resonance frequency between different resonators implementing the same synaptic weight into our neural network. Before training, at initialization, instead of using equation (2), we set the resonance frequencies of the resonators of the network using

$$f^{\mathrm{res}}_{i,j,c,h,w,m} \leftarrow \left[f^{\mathrm{RF}}_{h+i,w+j,c} + \Gamma_{h+i,w+j,c}\mathrm{N}\left(0,\sigma\right)\right]\left(1-\zeta_{i,j,c,m}\right), \tag{4}$$

where devices variability is introduced by $\mathrm{N}\left(0,\sigma\right)$, a sample from the normal distribution of center 0 and standard deviation σ, and $\Gamma_{h+i,w+j,c}$ is the bandwith of resonance of the resonators [60]. The standard deviation σ quantifies how much the resonance frequencies are deviated from their ideal values with respect to their bandwidth. In figure 6, we plot the result of convolutional neural network simulations including the variability of resonance frequencies between resonators for different standard deviations σ. Since we tune simultaneously all the resonators implementing the same synaptic weight, we see in figure 6 that the variability between the resonance frequencies of these resonators degrades the performance of the network, but that it stays accurate when the resonance frequency deviation is below 10% of the resonance bandwidth ($\sigma = 0.1$), at which point the accuracy is 99.05%. These results show that if the resonance frequency deviation is below 10%, such a network can be trained even though by using single write-lines for many synapses, we have fewer degrees of freedom than if we had individual control over each synapse.

1.5. Benchmarking the performance of a convolutional neural network with chains of resonators and spintronic nano-oscillators

We consider that we can optimize the resonators and field-lines such that the sensitivity of rectification is 10^3 μV μW^{-1}, as demonstrated experimentally [65]. Then, if the RF power injected per resonator is 0.1 μW, the order of magnitude of the rectified voltage output by the resonators is 0.1 mV, which is sufficiently large to be amplified by CMOS dc amplifiers. The minimum power consumption per synapse is then 0.1 μW. The DC amplifiers must deliver a power sufficient to match the threshold current of spintronic oscillators. Projections show that spintronic oscillators can be downscaled to nano-pillars of 20 nm radius, and then have a threshold current density of 10^{10} A m^{-2} [66]. With a resistance area product $RA = 10^{-12}$ Ω m^2, as seen in the literature [61], and radius of 20 nm, the minimum power delivered by each DC amplifier to a spintronic oscillator is 0.1 μW, which means that the minimum power consumption of the system per neuron is 0.1 μW. In the convolutional neural network architecture we use to solve the MNIST dataset, there are 7792 neurons and 6.7 million synapses. We then estimate the total power consumption of the implementation with RF spintronic devices as $\sim$0.67 W. The computing speed is limited by the relaxation time of the oscillator with the smallest frequency. If the smallest frequency of the system is $f^{\mathrm{RF}}_{\mathrm{min}} = 1$ GHz and the magnetic damping is $\alpha = 0.01$, which is the magnetic damping of permalloy [67], then the relaxation time is $T_{\mathrm{min}} = \frac{1}{\alpha f^{\mathrm{RF}}_{\mathrm{min}}} = 100$ ns. Since the relaxation time of each resonator and oscillator is 100 ns and since all the devices operate in parallel, we estimate that the total latency to process one image is inferior to 1 μs. In comparison, a Nvidia Tegra TK1 GPU consumes 8 W during inference, with a latency of 650 μs per image for an accuracy of 98.8% on the MNIST

dataset [68]. We then estimate a factor 11.9 of power consumption reduction compared to a Nvidia Tegra TK1 GPU, and the extremely parallel architecture we propose leads to more two orders of magnitude of latency reduction, thus ensuring a drastic energy consumption reduction.

Since spintronic oscillators and resonators can be downscaled to 20 nm [66], we can consider unit cells of area 40×40 nm^2. The area of a traditional unfolded convolution implemented on a crossbar array scales with $N_{\text{in}}N_c \times N_{\text{out}}N_m$ [14, 21] where N_{in} is the number of pixels in the input images, N_c is the number of channels, N_{out} is the number of pixels in the output feature maps, and N_m is the number of output feature maps. The optimized architecture we propose in figure 3(b) mitigates the problem of unused space in unfolded convolution implementations on crossbar arrays [21], and scales only in $N_{\text{out}}N_m \times k^2N_c$, with k^2N_c the number of pixels in a convolutional filter. Scaling with the size of the convolutional filter instead of the size of the image is advantageous because the convolutional filter is often much smaller than the image. For instance, in the architecture we use to solve MNIST, in the first convolutional layer, $N_{\text{in}} = 784$, $N_c = 1$, $N_{\text{out}} = 676$, $N_m = 32$ and $k^2 = 25$. We can then compute that with a unit cell of 40×40 nm^2, a traditional crossbar array implementing this convolutional layer would have an area of 2.7×10^{-2} mm^2, whereas our optimized crossbar would have an area of 8.6×10^{-4} mm^2, hence two orders of magnitude less. Since the number of neurons is much smaller than the number of synapses (7792 neurons versus 6.7 million synapses in the example we use), and since CMOS amplifies are only used for neurons, we can consider the area footprint of CMOS amplifiers to be smaller than the area footprint of spintronic resonators.

2. Conclusion

Using chains of spintronic resonators as artificial synapses rectifying RF signals into dc voltages, and nano-oscillators as nonlinear activation functions converting direct currents to RF signals, our system cascades information between different neural layers and performs the different multiplications of convolutions fully in parallel. In this paper, we showed how we can use chains of resonators to rectify multiple RF signals to make convolutions. We described the concept through a sequential convolution with a single chain of resonators, and then we showed how it can be extended to operate convolutions in parallel with multiple chains. This parallelization provides a tremendous processing time reduction: a sequential convolution requires approximatively as many steps as there are pixels in its input images, versus a single one for parallel convolutions. We have highlighted that with the two different proposed architectures, a single write-line can tune simultaneously many resonators to adjust their synaptic weights. The number of resonators adjustable by the same write-line is equal to the number of positions the filters take in the convolution. This can reduce the complexity and time required to train such hardware neuromorphic architecture. Since we use a resonance effect and field-lines that locally inject RF signals to spintronic resonators, each synapse only multiplies its corresponding neural input, and our architecture does not suffer from sneak-path that are common in memristor crossbar arrays. We can thus use a large number of devices in parallel to perform convolutions in a single step. Using spintronic oscillators as RF emitters emulating neurons, we show that chains of resonators can rectify neural signals, and that these rectification voltages can supply another layer of spintronic oscillators, hence enabling direct connectivity between different neural layers. We have demonstrated the performance of such network on the MNIST dataset though physical simulations of these RF spintronic devices and obtained 99.11% accuracy. Spintronic oscillators and resonators are similar devices that require the same materials and are both compatible with CMOS technology. Moreover, as these components can be downscaled to 20 nm, our architecture can contain a very large number of devices on a small surface. The density of devices is critical in a neural network, both to reduce power dissipation and to allow parallelism. The computing parallelism also limits the inference latency, which is crucial for modern embedded applications such as autonomous driving. In conclusion, this work opens new avenues to very large spintronic neuromorphic networks able to efficiently perform convolutions.

3. Methods

For the numerical simulations of handwritten digit classification, the physical response of hardware-based convolutional layers, nonlinear layers and fully connected layers is simulated based on well-established physical models of spintronic devices as detailed below.

3.1. Spintronic oscillators simulations

The output of each spintronic oscillator is computed with the universal auto-oscillator theory [60] in the case of a spin-torque nano-oscillator the normalized magnetization power is

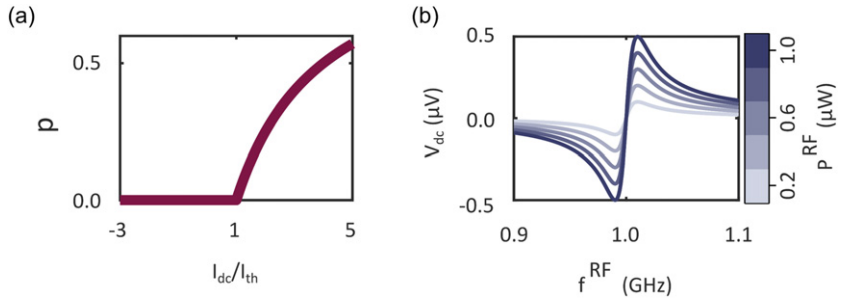

Figure 7. (a) Normalized magnetization oscillation power of a spin-torque nano-oscillator versus the ratio between direct current and threshold current. Plot realized using equation (5) with $Q = 2$. (b) Voltage rectified by spin-diode effect versus microwave frequency for different microwave powers $P^{RF} = 0.2$, 0.4, 0.6, 0.8 and 1 μW. Plot realized using equation (6) with $f^{res} = 1$ GHz, $\alpha = 0.01$, and A $= 1$ μV μW^{-1}.

$$p = \begin{cases} \dfrac{I_{dc}/I_{th} - 1}{I_{dc}/I_{th} + Q} & \text{if } I_{dc} > I_{th} \\ 0 \quad \text{if } I_{dc} \leqslant I_{th} \end{cases}, \tag{5}$$

with I_{dc} the direct current, and Q is the nonlinear damping coefficient. In this paper we chose $Q = 2$ according to experimental works [30, 36, 69]. This normalized magnetization power is simulated and plotted in figure 7(a). In the handwritten digits demonstration we chose a threshold current of 2 mA because it is coherent with experimental works [35, 47] and we clamp each direct current input to 8 mA because in practice, these nano-devices can be damaged above a particular current [61]. Since the threshold current density is typically around 10^{11} A m^{-2}, in the future the threshold current of these oscillators can be decreased to tens of μA with lateral size reduction to tens of nanometers. Moreover, projections show that size reduction could also reduce threshold current density down to 10^{10} A m^{-2} [66].

Most spintronic oscillators have a frequency dependence with input direct current [60, 70, 71]. We want to avoid this effect because, as we see in equation (7), the weights also depend on microwave frequencies, and we cannot allow the weights to depend on the input values. To avoid these frequency shifts, oscillators with compensated magnetic anisotropy can be used [69, 72].

3.2. Spintronic resonators simulations

Due to spin-diode effect, the voltage rectified by each spintronic resonator is:

$$V\left(P^{RF}, f^{RF}, f^{res}\right) = P^{RF} f^{RF} \frac{f^{RF} - f^{res}}{\alpha^2 f^{res\,2} + \left(f^{RF} - f^{res}\right)^2} \mathrm{A}. \tag{6}$$

This expression comes from the experimentally validated [65, 73] auto-oscillator model [60]. Here P^{RF} is the emitted power of an oscillator, f^{RF} its frequency, f^{res} is the resonance frequency of a spintronic resonator, α the magnetic damping of the material used for the spintronic resonators, and A a scaling factor. This voltages is simulated and plotted in figure 7(b). Writing equation (6) we neglect the frequency-symmetric Lorentzian component of the voltage and keep only the anti-symmetric Lorentzian component [73]. This approximation is useful because it lets the synaptic weight be either positive or negative, and is valid because in practice it is possible to tune the ratio between the symmetric and the anti-symmetric components by changing the orientation of a magnetic field [74, 75].

Then each rectification of a RF signal with a spintronic resonator is a multiplication operation on the microwave power P^{RF}, with a weight

$$W\left(f^{RF}, f^{res}\right) = f^{RF} \frac{f^{RF} - f^{res}}{\alpha^2 f^{res\,2} + \left(f^{RF} - f^{res}\right)^2} \mathrm{A}. \tag{7}$$

We see that because of the anti-Lorentzian shape of equation (7), we can tune the amplitude and the sign of the synaptic weight by tuning the resonance frequency f^{res}. In this paper we chose a magnetic damping coefficient $\alpha = 0.01$, which is the magnetic damping of permalloy [67], and an arbitrarily chosen amplification factor A $= 1$ μV μW^{-1}. For the neural network, the resonance frequencies are randomly initialized between 1 and 2 GHz for each layer.

3.3. Convolution specific multiply-and-accumulate operations with chains of spintronic resonators

In this subsection we demonstrate that with the correct choice of resonance frequencies, the voltages of our chains of spintronic resonators are convolution outputs like equation (1). From equation (6), we know that in a convolutional framework the voltage of each resonator inside each chain is

$$V_{i,j,c,h,w,m} = P^{\mathrm{RF}}_{h+i,w+j,c} f^{\mathrm{RF}}_{h+i,w+j,c} \frac{f^{\mathrm{RF}}_{h+i,w+j,c} - f^{\mathrm{res}}_{i,j,c,h,w,m}}{\alpha^2 {f^{\mathrm{res}}_{i,j,c,h,w,m}}^2 + \left(f^{\mathrm{RF}}_{h+i,w+j,c} - f^{\mathrm{res}}_{i,j,c,h,w,m}\right)^2} \mathrm{A}. \tag{8}$$

Replacing $f^{\mathrm{res}}_{i,j,c,h,w,m}$ in this equation by the expression of equation (2), we find

$$V_{i,j,c,h,w,m} = P^{\mathrm{RF}}_{h+i,w+j,c} \frac{\zeta_{i,j,c,m}}{\alpha^2 \left(1 - \zeta_{i,j,c,m}\right)^2 + \left(\zeta_{i,j,c,m}\right)^2} \mathrm{A}. \tag{9}$$

Since the total voltages of each chain of spintronic resonators is the sum of the voltages of each resonator of the chain, they are equal to

$$U_{h,w,m} = \sum_{c=0}^{N_C-1} \sum_{j=0}^{k-1} \sum_{i=0}^{\mathrm{k}-1} P^{\mathrm{RF}}_{h+i,w+j,c} \frac{\zeta_{i,j,c,m}}{\alpha^2 \left(1 - \zeta_{i,j,c,m}\right)^2 + \left(\zeta_{i,j,c,m}\right)^2} \mathrm{A}, \tag{10}$$

and we can identify this equation to equation (1), with $U_{h,w,m}$ equivalent to the outputs $z_{h,w,m}$, $P^{\mathrm{RF}}_{h+i,w+j,c}$ equivalent to the inputs $x_{i+h,j+w,c}$, and $\frac{\zeta_{i,j,c,m}}{\alpha^2\left(1-\zeta_{i,j,c,m}\right)^2+\left(\zeta_{i,j,c,m}\right)^2}\mathrm{A}$ equivalent to the filter coefficients $W_{i,j,c,m}$.

Acknowledgments

This project has received funding from the European Research Council ERC under Grant bioSPINspired 682955, from the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement RadioSpin No. 101017098 (project) and the French Ministry of Defense (DGA). The authors thank Erwann Martin and Jérémie Laydevant for their scientific support.

Data availability statement

The data that support the findings of this study will be openly available following an embargo at the following URL/DOI: https://github.com/NathanLerouxUMPhy/CNN-with-RF-spintronic-devices.git.

ORCID iDs

Nathan Leroux https://orcid.org/0000-0003-3672-0870
Dédalo Sanz-Hernández https://orcid.org/0000-0002-5552-8836
Danijela Marković https://orcid.org/0000-0001-7521-217X
Alice Mizrahi https://orcid.org/0000-0003-2043-049X
Julie Grollier https://orcid.org/0000-0003-4866-4490

References

[1] LeCun Y and Bengio Y 1998 Convolutional networks for images, speech, and time-series *The Handbook of Brain Theory and Neural Networks* (Cambridge, MA: MIT Press) pp 255–8
[2] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44
[3] LeCun Y, Bottou L, Bengio Y and Ha P 1998 Gradient-based learning applied to document recognition *Proc. IEEE* **86** 2278 https://ieeexplore.ieee.org/abstract/document/726791
[4] Karras T, Laine S and Aila T 2019 A style-based generator architecture for generative adversarial networks (arXiv:1812.04948 [Cs Stat])
[5] 2018 Big data needs a hardware revolution *Nature* **554** 145–6
[6] Marković D, Mizrahi A, Querlioz D and Grollier J 2020 Physics for neuromorphic computing *Nat. Rev. Phys.* **2** 499–510
[7] Christensen D V *et al* 2021 Roadmap on neuromorphic computing and engineering (arXiv:2105.05956 [Cond-Mat])
[8] Xia Q and Yang J J 2019 Memristive crossbar arrays for brain-inspired computing *Nat. Mater.* **18** 309
[9] Zhang W, Gao B, Tang J, Yao P, Yu S, Chang M-F, Yoo H-J, Qian H and Wu H 2020 Neuro-inspired computing chips *Nat. Electron.* **3** 371
[10] Esser S K *et al* 2016 Convolutional networks for fast, energy-efficient neuromorphic computing *Proc. Natl Acad. Sci. USA* **113** 11441

[11] Nikonov D E, Kurahashi P, Ayers J S, Li H, Kamgaing T, Dogiamis G C, Lee H-J, Fan Y and Young I A 2020 Convolution inference via synchronization of a coupled CMOS oscillator array *IEEE J. Explor. Solid-State Comput. Devices Circuits* **6** 170

[12] Xiang Y *et al* 2019 Hardware implementation of energy efficient deep learning neural network based on nanoscale flash computing array *Adv. Mater. Technol.* **4** 1800720

[13] Yakopcic C, Alom M Z and Taha T M 2016 Memristor crossbar deep network implementation based on a convolutional neural network *2016 Int. Joint Conf. on Neural Networks (IJCNN)* pp 963–70

[14] Yakopcic C, Alom M Z and Taha T M 2017 Extremely parallel memristor crossbar architecture for convolutional neural network implementation *2017 Int. Joint Conf. on Neural Networks (IJCNN)* pp 1696–703

[15] Yao P, Wu H, Gao B, Tang J, Zhang Q, Zhang W, Yang J J and Qian H 2020 Fully hardware-implemented memristor convolutional neural network *Nature* **577** 641–6

[16] Qin Y-F, Bao H, Wang F, Chen J, Li Y and Miao X-S 2020 Recent progress on memristive convolutional neural networks for edge intelligence *Adv. Intell. Syst.* **2** 2000114

[17] Gopalakrishnan R, Chua Y, Sun P, Sreejith Kumar A J and Basu A 2020 HFNet: a CNN architecture co-designed for neuromorphic hardware with a crossbar array of synapses *Front. Neurosci.* **14** 907

[18] Hamerly R, Bernstein L, Sludds A, Soljačić M and Englund D 2019 Large-scale optical neural networks based on photoelectric multiplication *Phys. Rev.* X **9** 021032

[19] Feldmann J *et al* 2021 Parallel convolutional processing using an integrated photonic tensor core *Nature* **589** 52–8

[20] Xu X *et al* 2021 11 TOPS photonic convolutional accelerator for optical neural networks *Nature* **589** 44–51

[21] Lin P *et al* 2020 Three-dimensional memristor circuits as complex neural networks *Nat. Electron.* **3** 225

[22] Grollier J, Querlioz D, Camsari K Y, Everschor-Sitte K, Fukami S and Stiles M D 2020 Neuromorphic spintronics *Nat. Electron.* **3** 360–70

[23] Kiraly B, Knol E J, van Weerdenburg W M J, Kappen H J and Khajetoorians A A 2021 An atomic Boltzmann machine capable of self-adaption *Nat. Nanotechnol.* **16** 414

[24] Jarollahi H, Onizawa N, Gripon V, Sakimura N, Sugibayashi T, Endoh T, Ohno H, Hanyu T and Gross W J 2014 A nonvolatile associative memory-based context-driven search engine using 90 nm CMOS/MTJ-hybrid logic-in-memory architecture *IEEE J. Emerg. Sel. Top. Circuits Syst.* **4** 460

[25] Kulkarni S R, Kadetotad D V, Yin S, Seo J-S and Rajendran B 2019 Neuromorphic hardware accelerator for SNN inference based on STT-RAM crossbar arrays *2019 26th IEEE Int. Conf. on Electronics, Circuits and Systems (ICECS)* pp 438–41

[26] Zhang Y *et al* 2021 Time-domain computing in memory using spintronics for energy-efficient convolutional neural network *IEEE Trans. Circuits Syst.* I **68** 1193

[27] Cassuto Y, Kvatinsky S and Yaakobi E 2013 Sneak-path constraints in memristor crossbar arrays *2013 IEEE Int. Symp. on Information Theory* pp 156–60

[28] Buhrman R 2009 Spin torque MRAM—challenges and prospects *2009 Device Research Conf.* p 33 https://ieeexplore.ieee.org/document/5354906

[29] Jung S *et al* 2022 A crossbar array of magnetoresistive memory devices for in-memory computing *Nature* **601** 211–6

[30] Marković D *et al* 2020 Detection of the microwave emission from a spin-torque oscillator by a spin diode *Phys. Rev. Appl.* **13** 044050

[31] Litvinenko A *et al* 2020 Ultrafast sweep-tuned spectrum analyzer with temporal resolution based on a spin-torque nano-oscillator *Nano Lett.* **20** 8

[32] Litvinenko A, Sethi P, Murapaka C, Jenkins A, Cros V, Bortolotti P, Ferreira R, Dieny B and Ebels U 2021 Analog and digital phase modulation and signal transmission with spin-torque nano-oscillators *Phys. Rev. Appl.* **16** 024048

[33] Leroux N, Marković D, Martin E, Petrisor T, Querlioz D, Mizrahi A and Grollier J 2021 Radio-frequency multiply-and-accumulate operations with spintronic synapses *Phys. Rev. Appl.* **15** 034067

[34] Leroux N *et al* 2021 Hardware realization of the multiply and accumulate operation on radio-frequency signals with magnetic tunnel junctions *Neuromorphic Comput. Eng.* **1** 011001

[35] Torrejon J *et al* 2017 Neuromorphic computing with nanoscale spintronic oscillators *Nature* **547** 428–31

[36] Romera M *et al* 2018 Vowel recognition with four coupled spin-torque nano-oscillators *Nature* **563** 230–4

[37] Marković D *et al* 2019 Reservoir computing with the frequency, phase, and amplitude of spin-torque nano-oscillators *Appl. Phys. Lett.* **114** 012409

[38] Tsunegi S, Taniguchi T, Nakajima K, Miwa S, Yakushiji K, Fukushima A, Yuasa S and Kubota H 2019 Physical reservoir computing based on spin torque oscillator with forced synchronization *Appl. Phys. Lett.* **114** 164101

[39] Zahedinejad M, Awad A A, Muralidhar S, Khymyn R, Fulara H, Mazraati H, Dvornik M and Åkerman J 2020 Two-dimensional mutually synchronized spin Hall nano-oscillator arrays for neuromorphic computing *Nat. Nanotechnol.* **15** 47–52

[40] Koo M, Pufall M R, Shim Y, Kos A B, Csaba G, Porod W, Rippard W H and Roy K 2020 Distance computation based on coupled spin-torque oscillators: application to image processing *Phys. Rev. Appl.* **14** 034001

[41] Nikonov D E, Csaba G, Porod W, Shibata T, Voils D, Hammerstrom D, Young I A and Bourianoff G I 2015 Coupled-oscillator associative memory array operation for pattern recognition *IEEE J. Explor. Solid-State Comput. Devices Circuits* **1** 85

[42] Watt S, Kostylev M, Ustinov A B and Kalinikos B A 2021 Implementing a magnonic reservoir computer model based on time-delay multiplexing *Phys. Rev. Appl.* **15** 064060

[43] Garcia M J *et al* 2021 Spin-torque dynamics for noise reduction in vortex-based sensors *Appl. Phys. Lett.* **118** 122401

[44] Kannan S, Rajendran J, Karri R and Sinanoglu O 2013 Sneak-path testing of crossbar-based nonvolatile random access memories *IEEE Trans. Nanotechnol.* **12** 413

[45] Cassuto Y, Kvatinsky S and Yaakobi E 2016 Information-theoretic sneak-path mitigation in memristor crossbar arrays *IEEE Trans. Inf. Theory* **62** 4801

[46] Joshi R and Acken J M 2021 Sneak path characterization in memristor crossbar circuits *Int. J. Electron.* **108** 1255

[47] Zahedinejad M, Fulara H, Khymyn R, Houshang A, Fukami S, Kanai S, Ohno H and Åkerman J 2020 Memristive control of mutual SHNO synchronization for neuromorphic computing (arXiv:2009.06594 [Phys.])

[48] Xu J-W, Chen Y, Vargas N M, Salev P, Lapa P N, Trastoy J, Grollier J, Schuller I K and Kent A D 2021 A quantum material spintronic resonator *Sci. Rep.* **11** 15082

[49] Jenkins A S *et al* 2014 Controlling the chirality and polarity of vortices in magnetic tunnel junctions *Appl. Phys. Lett.* **105** 172403

[50] Rivkin K and Ketterson J B 2006 Switching spin valves using Rf currents *Appl. Phys. Lett.* **88** 192515

[51] Wang W-G, Li M, Hageman S and Chien C L 2012 Electric-field-assisted switching in magnetic tunnel junctions *Nat. Mater.* **11** 64–8

[52] Cui Y-T, Sankey J C, Wang C, Thadani K V, Li Z-P, Buhrman R A and Ralph D C 2008 Resonant spin-transfer-driven switching of magnetic devices assisted by microwave current pulses *Phys. Rev.* B **77** 214440
[53] Sushruth M, Fried J P, Anane A, Xavier S, Deranlot C, Kostylev M, Cros V and Metaxas P J 2016 Electrical measurement of magnetic-field-impeded polarity switching of a ferromagnetic vortex core *Phys. Rev.* B **94** 100402
[54] Martins L, Jenkins A S, Alvarez L S E, Borme J, Böhnert T, Ventura J, Freitas P P and Ferreira R 2021 Non-volatile artificial synapse based on a vortex nano-oscillator *Sci. Rep.* **11** 16094
[55] Courbariaux M, Bengio Y and David J-P 2015 BinaryConnect: training deep neural networks with binary weights during propagations *Adv. Neural Inf. Process. Syst.* vol 28
[56] Hubara I, Courbariaux M, Soudry D, El-Yaniv R and Bengio Y 2016 Binarized neural networks *Adv. Neural Inf. Process. Syst.* vol 29 p 4107
[57] Bocquet M, Hirztlin T, Klein J-O, Nowak E, Vianello E, Portal J-M and Querlioz D 2018 In-memory and error-immune differential RRAM implementation of binarized deep neural networks *2018 IEEE Int. Electron Devices Meeting (IEDM)* pp 120.6.–4
[58] Laborieux A, Ernoult M, Hirtzlin T and Querlioz D 2021 Synaptic metaplasticity in binarized neural networks *Nat. Commun.* **12** 2549
[59] Laydevant J, Ernoult M, Querlioz D and Grollier J 2021 Training dynamical binary neural networks with equilibrium propagation (arXiv:2103.08953 [Cs])
[60] Slavin A and Tiberkevich V 2009 Nonlinear auto-oscillator theory of microwave generation by spin-polarized current *IEEE Trans. Magn.* **45** 1875–918
[61] Costa J D *et al* 2017 High power and low critical current density spin transfer torque nano-oscillators using MgO barriers with intermediate thickness *Sci. Rep.* **7** 7237
[62] Murphy K P 2012 *Machine Learning: A Probabilistic Perspective* (Cambridge, MA: MIT Press)
[63] Kingma D P and Ba J 2017 Adam: a method for stochastic optimization (arXiv:1412.6980 [Cs])
[64] Tsunegi S *et al* 2014 High emission power and *Q* factor in spin torque vortex oscillator consisting of FeB free layer *Appl. Phys. Express* **7** 063009
[65] Fang B *et al* 2016 Giant spin-torque diode sensitivity in the absence of bias magnetic field *Nat. Commun.* **7** 11259
[66] Chao X, Jamali M and Wang J-P 2017 Scaling effect of spin-torque nano-oscillators *AIP Adv.* **7** 056624
[67] Fuchs G D *et al* 2007 Spin-torque ferromagnetic resonance measurements of damping in nanomagnets *Appl. Phys. Lett.* **91** 062507
[68] Joseph V and Nagarajan C MADONNA: a framework for energy measurements and assistance in designing low power deep neural networks 7 (n.d.) http://cs.utah.edu/vinu/madonna.pdf
[69] Jiang S, Khymyn R, Chung S, Le T Q, Diez L H, Houshang A, Zahedinejad M, Ravelosona D and Åkerman J 2020 Reduced spin torque nano-oscillator linewidth using He^+ irradiation *Appl. Phys. Lett.* **116** 072403
[70] Dussaux A, Khvalkovskiy A V, Bortolotti P, Grollier J, Cros V and Fert A 2012 Field dependence of spin-transfer-induced vortex dynamics in the nonlinear regime *Phys. Rev.* B **86** 014402
[71] Zeng Z *et al* 2013 Ultralow-current-density and bias-field-free spin-transfer nano-oscillator *Sci. Rep.* **3** 1426
[72] Divinskiy B, Urazhdin S, Demokritov S O and Demidov V E 2019 Controlled nonlinear magnetic damping in spin-Hall nano-devices *Nat. Commun.* **10** 5211
[73] Tulapurkar A A, Suzuki Y, Fukushima A, Kubota H, Maehara H, Tsunekawa K, Djayaprawira D D, Watanabe N and Yuasa S 2005 Spin-torque diode effect in magnetic tunnel junctions *Nature* **438** 339–42
[74] Wang C, Cui Y-T, Sun J Z, Katine J A, Buhrman R A and Ralph D C 2009 Bias and angular dependence of spin-transfer torque in magnetic tunnel junctions *Phys. Rev.* B **79** 224416
[75] Wang C, Cui Y-T, Katine J A, Buhrman R A and Ralph D C 2011 Time-resolved measurement of spin-transfer-driven ferromagnetic resonance and spin torque in magnetic tunnel junctions *Nat. Phys.* **7** 496