Neuromorphic Architectures for Spiking Deep Neural Networks

Giacomo Indiveri, Federico Corradi, and Ning Qiao
Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland
e-mail: giacomo@ini.uzh.ch

Introduction: **DNN!**s (**DNN!**s) have recently shown state-of-art performance in multiple benchmarks tasks such as computer vision,machine translation,and speech recognition [**LeCun˙etal15**]. They are composed of many layers of neurons coupled among each other via weighted connections (see Fig. **??**). Currently, most applications solved by **DNN!**s are running on conventional computing systems which are not ideally suited to implement such massively parallel architectures. Here we propose a new set of neuromorphic devices and systems that can implement spiking **DNN!**s efficiently, using low-latency, low-power, and compact circuits. The neuromorphic devices comprise mixed-signal analog/digital circuits. The analog circuits implement compact and faithful models of biological synapses and spiking neurons, and carry out the neural computation operations of the network. The digital circuits include both asynchronous designs, used to transmit the spike-events among neurons and across devices, and standard digital logic elements, used to program the network topology and the routing schemes for implementing a wide variety of different **DNN!** architectures. We demonstrate a convolutional network application example implemented using exclusively neuromorphic devices, from the visual input stages to the output classifier ones. By construction, the architecture carries out robust computation which is mismatch-tolerant: by using multiple (mismatched) silicon neurons in the learning process as weak classifiers we implement a well known machine learning ensemble method which has been shown to be provide substantial gains in accuracy and recognition rates [**Breiman96**].

The experimental setup: The experimental setup used to demonstrate the spiking convolutional network is depicted in Fig. **??**. It comprises a silicon retina **DVS!** (**DVS!**) device, a full custom multi-chip board, and the **ROLLS!** (**ROLLS!**) device used as classification layer. The **DVS!** retina used to provide the visual input signals to the convolutional network is currently commercially available [**iniLabs**].

The multi-chip board comprises 9 "cxQuad" multi-neuron chips (see Fig. **??**). The cxQuad chip is implemented using a novel scalable and reconfigurable multi-core hierarchical routing scheme. It utilizes the **AER!** (**AER!**) to route spikes among neurons both within a core and across cores (also across chip boundaries) with asynchronous digital logic, and implements neural and synaptic dynamics using log-domain pulse integrator **DPI!** (**DPI!**) filters [**Qiao˙etal15**] and adaptive-exponential **IF!** (**IF!**) neuron circuits [**Qiao˙etal15**]. Each chip has 1k neurons and 64k synapses distributed among four cores. Specifically, each core has 16×16 units, which contain one neuron and one synapse block. The synapse block comprises a linear integrator circuit which integrates input events from 64 12-bit programmable **CAM!**s (**CAM!**s) cells. Output events generated by the neurons can be routed to the same core, via a Level-1 router, to other cores on the same chip, via a Level-2 router, or to cores on different chips, via a Level-3 router. The memory used by the routers to store post synaptic destination addresses is implemented using 4k 12-bit SRAMs blocks distributed among the Level-1, 2, and 3 router circuits. Thanks to the scalable architecture and to the on-chip programmable routers, the routing of all 9k neurons on the board can be easily configured to implement a wide range of connections schemes, without requiring external mapping, memory, or computing support.

The classification layer is implemented by the **ROLLS!** chip, which comprises 256 neurons and 133,120 synapses (see Fig. **??**). The synapse circuits are of three different types: linear time-multiplexed, **STP!** (**STP!**) synapses, and **LTP!** (**LTP!**) synapses. The **STP!** synapses have analog circuits that can reproduce short-term adaptation dynamics and digital circuits that can set and change the programmable weights. The **LTP!** synapses contain analog learning circuits and digital state-holding logic. The learning circuits implement a stochastic binary plasticity **STDP!** (**STDP!**) model [**Qiao˙etal15**] which slowly drives the weights to one of two stable states, depending on the analog synapse's weight value, compared to a threshold bias (see [**Qiao˙etal15**] for a through description and characterization of these circuits).

Results: We implemented the **DNN!** of Fig. **??** by programming the weights of the convolution layer to extract oriented edges, and teaching the classification layer to recognize eight different visual symbols. Figure **??** shows the spiking neuron activities for the different layers in the network. The output of the convolutional network input, convolution and pooling layers (see second, third and fourth row of Fig. **??**) evidences the differences in the spiking activities as a function of the different input symbols presented. The classifier layer training is supervised: it consists in driving the neurons belonging to the true class ensemble with a high-frequency teacher signal, and inhibiting all other neurons, during the presentation of the true class stimulus. This is evident in the last row of Fig. **??**, where different ensembles of neurons are sequentially activated by the teacher signal as different symbols are presented. The plastic bi-stable synapses at the beginning of the experiment are all set to their low state (see fifth row in Fig. **??**). During training, the synapses stimulated by the relevant features from the convolution layer and belonging to the neurons that are driven by the teacher signal will tend to potentiate, while all other synapses will tend to depress. At the end of the training procedure, during the testing phase, the high clustered activity in the classification layer in response to the corresponding visual symbol presentation demonstrates successful classification (see bottom row, right plot in Fig. **??**).

Figure 1: Deep Neural Network architecture.

Figure 2: Experimental setup: a silicon retina vision sensor converts visual stimuli displayed on the screen into streams of spike Address-Events (AEs). The retina AEs are mapped onto the convolutional layer cxQuad board via a direct cable connection. The spiking output of the convolutional network is then mapped onto the classification layer **ROLLS!** device, which is configured to use ensembles of neurons for classifying the features extracted by the convolution layer.

Figure 3: cxQuad chip, fabricated using a 180 nm 1P6M CMOS process. It occupies an area of 43.79 mm^2 and comprises 1k neurons and 64k*12-bit **CAM!** programmable synapses subdivided among 4 cores. In addition it integrates 4k*12-bit SRAMs, 3-level hierarchical routers, two temperature compensated bias generator circuits, and one input predecoder block.

Figure 4: ROLLS chip, fabricated using a 180 nm 1P6M CMOS process. It occupies an area of 51.4 mm^2 and comprises 64 k STP programmable synapses, 64 k LTP learning synapse, 256 shared synapses, and 256 analog neurons. Furthermore, it comprises digital **AER!** input/output blocks, an on-chip temperature compensated bias generator, and analog current/voltage to spike-frequency converters.

Figure 5: Convolution and learning performance. The top row shows the eight **visual stimuli** used to train the classification layer. The **Convolutional Network** row shows activations of the input, convolution, and pooling layers implemented by the cxQuad board, for the different visual stimuli. White dots represent spiking activity of neurons in the corresponding layer. The input layer of the convolutional network is composed of 32×32 neurons; the convolution layer is composed of four 16×16 features maps; the pooling layer is composed of a single 16×16 array of neurons. The **Learning Synaptic Matrix** row shows the state of the ROLLS LTP bi-stable synapses before and after training. Black dots represents synapses in the low state, while white dots represents synapses in high state. Horizontal green lines divide the groups of ensembles trained separately for the eight different input stimuli. The **Classification Layer** row shows the spiking activity during the training phase (left), and the test phase (right). Each dot in the plot represents a spike. The horizontal axis represents time, and the vertical one the neuron address. Regions highlighted by the boxes in the right plot evidence how the ensemble of neurons stimulated by the stimulus it was trained to recognize produces a higher activity, compared to all other neurons in the same vertical region, therefore demonstrating correct classification performance.