1
Fig. 3.9 Synaptic plasticity: Spike-timing dependent plasticity (STDP) and its parameters from Bienenstock et al. [70] (55% Increase in Spontaneous Firing Rate (ITP)).

Table 3.1 Categories of different design options for hardware implementation of neuro-inspired computing. Representative prototypes are shown.

<table>
<thead>
<tr>
<th>Digital Representation</th>
<th>Off-the-shelf technologies</th>
<th>CMOS ASIC</th>
<th>Emerging resistive synaptic devices</th>
</tr>
</thead>
</table>

Fig. 3.1 A resolution shift of the computing paradigm from the computation-centric (von Neumann architecture) to the data-centric (neuro-inspired architecture).

Fig. 3.2 (a) Hybrid I-V characteristics of a MTF device. (b) Circuit configuration of a neuromorphic system with MTF devices and RRAM synaptic weight. (c) SPICE simulation waveform of the oscillation neuron.

Table 3.1 Categories of different design options for hardware implementation of neuro-inspired computing. Representative prototypes are shown.

<table>
<thead>
<tr>
<th>Digital Representation</th>
<th>Off-the-shelf technologies</th>
<th>CMOS ASIC</th>
<th>Emerging resistive synaptic devices</th>
</tr>
</thead>
</table>

Fig. 5. A qualitative comparison of neuron models in terms of biological inspiration and complexity of the neuron model.
Ionic Channels in Hodgkin-Huxley Model
A Memristive System (Chua and Kang, Proc. of the IEEE, 1976)

- Parallel conductance model of the membrane
- Hodgkin-Huxley equations

\[
\frac{dV}{dt} = \frac{1}{C_m} \left( I_{\text{ion}} - g_{\text{Na}}(V - E_{\text{Na}}) - g_{\text{K}}(V - E_{\text{K}}) - g_{\text{Cl}}(V - E_{\text{Cl}}) \right)
\]

Fig. 7. A breakdown of various models in memristive implementations grouped by circuit type and used to reflect the number of memristor papers.

Fig. 8. Different network topologies that might be desired for neuromorphic systems. Determining the level of connectivity is crucial for a neuromorphic implementation and thus finding the appropriate hardware that can accommodate the level of connectivity is a considerable challenge.

Fig. 9. An overview of on-chip training/learning algorithms. The size of the box corresponds to the number of papers in that category.

Fig. 10. An overview of hardware implementations in neuromorphic computing. These implementations are relatively basic hardware implementations and do not contain the more control devices as discussed in Section V.B.

Fig. 9.1 Representative resistive NVM arrays with peripheral circuits are shown. (a) ITIR array with row-by-row operation. (b) Pseudo-crossover array implemented by setting BL in ITIR memory array. (c) True crossover array without selector transistors for array-level parallelism.
Fig. 11. Device-level components and their relative popularity in neuromorphic systems. The size of the boxes corresponds to the number of works referenced that have included those components.

Fig. 12. Main research questions and their answers on a fabric substrate.

Fig. 13. Functional Circuitry on Commercial Fabric via Textile-Compatible Nanoscale Film Coating Process for Fibertronics.

Fig. 14. Examples from different image data sets: MNIST [2618], CIFAR10 [2619], and SVHN [2620]) to which neuromorphic systems have been applied for classification purposes.

Fig. 100% real fabric functional circuit with a missing 4th electronic component of memristor beyond resistor, capacitor and inductor.
Linear in Weight Update: The linearity in weight update refers to the linearity of the curve between the device conductance and the number of identical programming pulses. Ideally, this should be a linear relationship for the direct mapping of the weights in the algorithms to the conductance in the devices. However, the relative synaptic devices generally have the nonlinearity in weight update (see Fig. 1.3). The trajectory of the long-term potentiation (LTP) process that increases the conductance differs from that of the long-term-depression (LTD) process that decreases the conductance. The weight tends to saturate at the end of LTP or LTD processes. This nonlinearity is undesired because the change of the weight (ΔW) depends on the current weight (W), or in other words, the weight update has a history dependence. Recent results have shown that this nonlinearity has caused the learning accuracy loss in the neural networks [41, 42].
Programming Energy Consumption: The estimated energy consumption per synaptic event is around $1 \times 10^{-12}$ J in biological synapses. Most RRAM/CHiPAM devices show a programming energy around $100 \text{ fJ} - 1 \text{ pJ}$, while most PCM devices may have even higher programming energy $10 - 100 \text{ pJ}$. The fundamental challenge is that it is much more difficult (thus paying more energy) to move the ions/defects in solid-state devices than moving calcium ions in the liquid environment in biological synapses. A back-of-the-envelope calculation is given as follows. In biological synapses, the spike voltage is $\approx 100 \text{ mV}$, the ionic current $\approx 1 \text{ nA}$, and the spike period $\approx 1 \text{ ms}$; therefore, the energy is about $10 \text{ fJ}$. In resistive synaptic devices, the typical programming voltage is $\approx 1 \text{ V}$, and the programming current is typically $\approx 1 \text{ mA}$, although the programming speed can be accelerated less than the real time to be $\ll \mu \text{s}$, still the energy is on the order of $\text{pJ}$. Further device engineering is thus needed to reduce the energy consumption.

Fig. 1.3 The measured nonlinearity in the weight update reported from the literature: (a) TaOx/TaOx device [99], (b) PCMO device [36], and (c) Au@Si device [35].

Requirement of NVM. The number of endurance is much application dependent, relying on how many weight updates are required in the training processes. For a relatively simple task (i.e., the MNIST handwritten digit recognition [43]), 60,000 training images with 50 training epochs (no repeated) give a maximum weight update possibility to be $3 \times 10^8$ updates. Actually not every synapse is updated in the training; thus, an endurance $\approx 10^5$ is sufficient for training MNIST datasets [19]. However, considering more challenging tasks (i.e., ImageNet challenge [44]), much more endurance may be required.

Uniformity and Variability. Poor uniformity or significant variability in emerging NVMs is a major barrier for digital-memory applications. In contrast, the neural networks promise robustness against device variations. The device variations could partially be tolerated by two mechanisms: the massive (thus maybe redundant) connections between neuron nodes by synaptic arrays and the iterative weight update process during the training. The degree of variations that can be tolerated at the system level strongly depends on the network architecture and the accuracy required by the target application. The device-algorithms co-simulations have shown the reasonable robustness against device variations in different neural networks [42, 45].

Fig. 11.1 (a) Traditional processor-co-processor architecture with shared memory; (b) PRIME architecture using 3D integration technology; (c) PRIME design.

\[
\begin{align*}
&\frac{y_i}{y_i} + \frac{y_j}{y_j} + \frac{y_k}{y_k} + \frac{y_l}{y_l} \\
&\frac{z_i}{z_i} + \frac{z_j}{z_j} + \frac{z_k}{z_k} + \frac{z_l}{z_l} \\
&\frac{x_i}{x_i} + \frac{x_j}{x_j} + \frac{x_k}{x_k} + \frac{x_l}{x_l}
\end{align*}
\]
**Forward Propagation**

\[ x_i = \text{ff}(\sum w_{ij} x_j) \]

- **Fig. 11.28** Forward propagation operation in a deep neural network. The multiply-accumulate operation occurs on the column array. Neuron circuitry must handle the nonlinear squashing function (Adapted from Fernandez et al. [19]).

**Reverse Propagation**

\[ \delta_i = f'(x_i) \sum \delta_j w_{ij} \]

**Fig. 11.22** Reverse propagation operation in a deep neural network. Multiply-accumulate operation on \( \delta \) occurs on the column array. Neuron circuitry must handle generation and multiplication of the derivative of the squashing function (Adapted from Fernandez et al. [19]).

**Fig. 11.18** Schematic showing memristor-compatible [1] weight update rule for analog bidirectional NVMs. Weight increases (decreases) can be implemented either as a SET operation on \( G^+ \) or a RESET operation on \( G^- \) devices. Asymmetry in the partial SET and RESET operation is compensated by applying a different learning rate parameter (reset factor) that modulates the number of pulses fed from the neurons into the array (Adapted from Fernandez et al. [19]).

**Fig. 3.6** Similar on/off ratio of 10 bits in the 64-kb NROM synaptic array.

**Fig. 6.4** (a) The state of the neuron matrix for the IC, which is simulated with one epoch of training, specifies the direction of the state updates for each device in a stimulus cycle. (b) Whether in a binary or eight-valued IC, the (a) hardware results in a binary result. The voltage shown is an unweighted and unpowered state for the first and second states, respectively. The waveforms show the timing characteristics. Gray arrows show the steps implemented in hardware, while all remaining steps are unselected in software.

**Programming by \( \frac{1}{2} \)-scheme**

- \( \frac{1}{2} \)
- \( \frac{1}{4} \)
Figure 4. Circuit diagram for a cross-matrix architecture with the following elements and functions:

- **Crossbar Array**: A 2D array of neural cells connected in parallel.
- **Neural Cells**: Each cell is responsible for computing a weighted sum of inputs.
- **Weighted Sum**: The output of each neural cell is a weighted sum of its inputs.
- **Synaptic Weighting**: The weight of each connection is indicated by the thickness of the bars connecting the cells.

Figures 5-10 illustrate the performance and behavior of the cross-matrix architecture under various conditions.

Figures 11-15 show the results of experiments with different input patterns and network configurations, demonstrating the robustness and adaptability of the architecture.

---

**B.T. Murphy**

**Yield Model**

\[ Y = Y_0 e^{-2A} > P_{\text{defects}} \]

- **Y**: Yield
- **Y_0**: Initial yield
- **A**: Area
- **P_{\text{defects}}**: Defect density

**Yield**: The yield of a chip area containing the N transfer circuit.