A Neuromorphic Architecture From Single Transistor Neurons With Organic Bistable Devices For Weights

Robert A. Nawrocki, Sean E. Shaheen, Richard M. Voyles

Abstract— Artificial Intelligence (AI) has made tremendous progress since it was first postulated in the 1950s. However, AI systems are primarily emulated on serial machine hardware that result in high power consumption, especially when compared to their biological counterparts. Recent interest in neuromorphic architectures aims to more directly emulate biological information processing to achieve substantially lower power consumption for appropriate information processing tasks. We propose a novel way of realizing a neuromorphic architecture, termed Synthetic Neural Network (SNN), that is modeled after conventional artificial neural networks and incorporates organic bistable devices as circuit elements that resemble the basic operation of a binary synapse. Via computer simulation we demonstrate how a single synthetic neuron, created with only a single transistor, a single-bistable-device-per-input, and two resistors, exhibits a behavior of an artificial neuron and approximates the sigmoidal activation function. We also show that, by increasing the number of bistable devices per input, a single neuron can be trained to behave like a Boolean logic AND or OR gate. To validate the efficacy of our design, we show two simulations where SNN is used as a pattern classifier of complicated, non-linear relationships based on real-world problems. In the first example, our SNN is shown to perform the trained task of directional propulsion due to water hammer effect with an average error of about 7.2%. The second task, a robotic wall following, resulted in SNN error of approximately 9.6%. Our simulations and analysis are based on the performance of organic electronic elements created in our laboratory.

I. INTRODUCTION

THE term Artificial Intelligence (AI) was coined by John McCarthy circa 1956 and is usually described as “the study and design of intelligent agents” where the system perceives its environment and takes appropriate action that maximize its chances of success [1]. Today, about six decades later, AI can be found in automobiles [2], computer games [3], medical diagnosis [4], and even email filtering [5]. However, the forecast of human-like machines with human-line intelligence has largely disappointed. DARPA’s Grand Challenge has shamelessly exposed the limitations of today’s state-of-the-art AI [6]. Even though IBM’s Watson, a supercomputer system aimed at understanding the meaning and context of human language, in a quiz show Jeopardy! beat two of its most successful contestants, it used 90 of IBM POWER 750 servers with a total of 2880 Power7 processor cores and 16 Terabytes of RAM. It also required a team of IBM’s programmers to explicitly code its knowledgebase and search patterns [7].

Many researchers believe that the limitation of AI stems from the fact that it is emulated on serial machines, built according to the architecture that dates back to Alan Turing [8]. This often results in AI that is explicitly coded for a specific task, such as face recognition [9] or industrial power system control [10], or simply relies on brute force strategy [11]. Today’s approaches to emulate biological information processing have to run on supercomputers and consume megawatts of power [12]. In comparison, a fruit fly uses µW of power for such various and complex tasks as flight control, food search, and avoidance of predators. Furthermore, today’s AI is slow when compared to its biologic counterpart; catching a baseball in mid-flight requires massive information processing parallelism as visual information about the constantly-changing trajectory has to be correlated to inverse kinematics needed to accurately position the arm. Such a task would require a supercomputer the size of a large apartment [13].

A. Biological Networks

In a biological brain, as compared to Turing-based machines, there is no macroscopic separation of information and information processing. A biological neuron is responsible for processing, transmitting, and storing information. A single neuron can receive information from as many as 10,000 neurons. It, then, can be connected to as many as few million other neurons [14]. This massive parallelism and modular design is believed to be behind such spectacular energy efficiency, high-speed of operation as well as generic intelligence and robustness against damage [15,16].

B. Artificial Neural Networks

Neuromorphic architectures aim to mimic information processing the way it is achieved in biological systems. The immediate benefit of the biology-inspired architecture seems to be reduced power consumption, inherent fault-tolerant architecture, generic AI application and the foundation for true hardware-based AI.

Artificial Neural Networks (ANN), being the earliest and most ubiquitous example of neuromorphic architecture, are mathematical constructs that attempt to capture the essential components and functionalities of biological neural networks. They are most often used to model systems where
the non-linear relationship is not explicitly known and difficult or impossible to determine analytically. ANNs have been employed in applications such as image processing [17], and hand-writing recognition [18] where the system is expected to learn over time. However, their real-time performance can be limited, especially if the system is expected to perform on-line learning, due to the fact that they are emulated on serial machines.

C. Literature Review

A number of research groups have demonstrated possible realizations of hardware neural networks. Deshmukh et al. [19] have proposed realizing the functionality of a neuron with a conventional CMOS technology. They constructed a discrete neuron with binary connection weights, based on logic gates and flip flops, capable of simple pattern recognition. Similar to their design, our proposal, in the most basic form, can also use binary connection weights. However, such weights are a limitation as such a system is not suitable for many applications. As such, our design allows, with a relative ease, to increase the granularity of connection weights.

Gupta et al. [20] proposed a design aiming at realizing the sigmoidal activation function, commonly used with neural networks. Their realization featured fifteen transistors as well as four current sources capable of producing a family of outputs with various saturation levels of the sigmoidal curve. Similar to this proposal, our circuit can also obtain a family of activation functions that can approximate a sigmoidal curve. However, our design is only based on a single transistor as compared to fifteen transistors and four current sources proposed by Gupta et al. Additionally, it goes activation function and offers the functionality of an entire neuron.

A number of companies, including Intel, IBM, AMD, Hitachi, and Siemens, offer commercial chips based on CMOS technology, that, to a various extent (degree of precision, including or excluding learning ability, various network architectures), realize functionality of neural networks [21]. At present our design does not incorporate any learning algorithms. Pre-computed connection weights, via external software training such the Matlab ANN ToolBox, have to be manually imported. Additionally, there is a limit to the precision of both the activation function and the connection weights. The accuracy of activation function is explicitly related to the physical characteristics of a transistor used. The precision of connection weights, however, can be arbitrarily extended.

A device made of molecules and nanoparticles, termed nanoparticle organic memory field-effect transistor, or NOMFET [22], was shown to exhibit a facilitating (excitatory) and depressing (inhibitory) behavior of a biological spiking synapse. By applying positive or negative Gate voltage, while holding Source and Drain contacts grounded, facilitating or depressing behavior can be programmed respectively. This device makes use of the charge storage capability of nanoparticles embedded in an insulating material that is sandwiched between source and drain of a transistor. Applying voltage repeatedly results in steadily increasing or decreasing voltage response of the NOMFET due to individual charges being continuously and cumulatively retained by Gold nanoparticles.

Jo et al. [23] have suggested the possibility of employing a silicon-based memristor to realize the time-dependent characteristics of a biological spiking synapse. Their design utilizes metal-oxide semiconductor neurons and silicon-based memristors (short for memory resistor) as a synapses for synaptic functions such as spike, timing dependent plasticity. In this design, memristors are arranged into a cross-bar with pre- and post-synaptic neurons being located on adjacent sides of the cross-bar. This allows for every post-synaptic neuron to receive information from every pre-synaptic neuron. The use of memristive synapses results in neural behavior akin to spiking, biological neurons.

The strength of a design that exhibits spiking behavior is that it does not require an external learning circuit. The value (charge) of a synapse is directly related to the time-dependent input signal; the synapse learns continuously. In biology this concept is known as a synaptic plasticity. Lack of stimuli, however, results in synapse de-learning or forgetting information stored. The design that we propose, based not on the memristive synapse operating in its linear region but rather on the saturation property of organic bistable devices (discussed later in text). Once synaptic weights are set, during the learning stage, the synapse will retain this information indefinitely (the limitation being physical degradation of the device). This property makes it much more analogous to conventional ANNs and easier to employ in standard engineering applications.

II. Architecture

In a biological neuron the synapse is responsible for converting the chemical or electrical input signal and passing it onto the soma, or neuron body, for further processing [14]. Its purpose is to provide a neuron with an input that is proportional to the importance of the signal from that specific neuron: An input from an “important” neuron is passed much more easily (the synapse is more sensitive to the signal and will pass a signal of a minimal strength) to the neuron body than an input from an “un-important” neuron (the synapse is less sensitive to the signal and the signal will have to be very strong in order for the synapse to respond to it). It is generally accepted that the information is stored in the synaptic strength. Learning (neuro- or synaptic plasticity) is accomplished by modifying, either increasing or decreasing, the strength (sensitivity) of the synapse.

In an artificial neuron the functionality of a synapse is realized via a connection weight. Analogously to a biological synapse, connection weight from an “important” neuron is significantly greater than a connection weight from an “un-important” neuron. Learning in an ANN is accomplished by modifying the numerical value of the aforementioned connection weight.

The function of a soma, in both biological and artificial neurons, is to sum all the input signals and produce an
output signal when a predefined threshold value has been exceeded by all these summed inputs. Whereas biological neurons encode the information by modifying the frequency of firing, in artificial neural networks it is the amplitude of the signal that changes.

A. Background

Expanding on the idea originally proposed by Likharev [24] of realizing a nanoscale hybrid semiconductor/nanodevice integrated circuit, termed CMOL (CMOS/nanowire/molecular-nanodevice), we proposed [25]-[26] an analogous, single-transistor architecture that uses a bistable device to realize the multiplicative property of a synapse.

B. Operation of a Synthetic Neuron

Equation 1 represents the algorithm commonly used to compute the output of an artificial neuron. In general, it performs three functions: it multiplies all the inputs \( x_i \) by their corresponding connection weights \( w_i \), it sums all of these products \( \sum x_i w_i \), and it produces the output based upon the utilized activation function, denoted by \( \phi \) in Equation 1 [27]. Conceptually, the multiplication is accomplished by the artificial synapse, while the summation and activation are performed by the artificial soma. Hence every representation of a neuron must be able to perform those three functions, namely multiplication, summation, and activation.

\[
f(x) = \phi(\sum x_i w_i)
\]

Equation 1. A function that describes the operation of an artificial neuron.

More generally, many neural representations also include bias value commonly added to the product of input values and connection weights. However, this value, for the purposes of simplicity, can be treated as equal to zero.

C. Bistable Device as a Synapse for Connection Weight

In the most basic form a bistable device can be thought of as a memory device represented by two resistors; it can be in one of two possible states, with its resistance corresponding to being either in the ON (low resistance) or OFF (high resistance) state. The operation of this device can be summarized as follows [28],[29]. It is in the OFF state (high resistance) until the input voltage is increased past a threshold voltage \( V_{ON} \). It stays in the ON state (low resistance) until the voltage is reduced below a threshold voltage \( V_{OFF} \).

With the output of a bistable device (current) equal to a product of input (voltage) and connection weight (conductance), and the fact that its value can be electrically modified (analogous to modifying the value of connection weight during the neural network training), a bistable device seems a natural choice to realize the functionality of artificial or synthetic binary synapse.

It should be mentioned that both biological and artificial synapses can be either excitatory (analogous to positive connection weights) or inhibitory (negative connection weights). The use of a bistable device only allows for positive connection weights.

Figure 1 portrays the IV characteristics of an organic bistable created in our laboratory, with the appropriate ON and OFF currents and their respective \( R_{ON} \) and \( R_{OFF} \) marked. The near-linear ON and OFF resistances \( (R_{ON} \text{ and } R_{OFF} \text{ respectively}) \) are clearly visible.

The organic bistable device, shown in Figure 1 was constructed as a layered device. An active layer, responsible for resistive switching, is sandwiched between top and bottom electrodes. The switching mechanism is explained via the effect known as charge trapping [30]. The active layer is constructed from an insulating material, poly(methylmethacrylate), doped with highly conductive ZnO QDs. When the electric field is first applied, the active layer is first highly resistive (OFF state). However, increasing the electric field past a threshold value, results in some of the electrons being trapped by the ZnO QDs. Subsequently, the resistance across the active layer is significantly lower due to the aforementioned trapped electrons (ON state).

D. Single Transistor Circuit as a Soma for Summation and Activation Function

In both biological neurons and artificial neurons, the soma performs two functions: i) summation of the inputs (from dendrites in the biological case) and ii) mapping of the summed-input signal to the output signal, usually along a sigmoidal function. Also, an important property of a neuron is that there are minimum and maximum values where the output saturates; for an unbounded input there is a bounded output.

We have developed a simple, single-transistor circuit, shown in Figure 2A, which produces a suitable approximation to a neural soma. Given the weighted inputs described above (in a form of a value, or a state, of the organic bistable device), the single-transistor circuit sums the inputs and produces a nonlinearly scaled output that approximates a sigmoid. To emulate sigmoidal behavior, we used a single characteristic \( V_{SD} \) (Source-Drain voltage) curve by choosing \( V_G \) (Gate voltage) based on utilized \( R_{ON} \) and \( R_{OFF} \) resistances. Additionally, our design allows
different neural behaviors (for different types of neurons) to be achieved by employing different gate voltages. For an extended range of output values, \( V_G \) could be made variable.

Figure 2A demonstrates the circuit proposed to realize the functionality of an artificial neuron. However, as already mentioned in *Bistable Device as a Synapse for Connection Weight*, with operating voltage below a threshold voltage, a bistable device can be approximated with two resistors (corresponding to either ON, or low resistance, or OFF, or high resistance, states). Therefore Figure 2B illustrates a circuit used for the purposes of analysis.

The analysis of the circuit can be presented as follows. The input voltage, \( V_{in1} \) or \( V_{in2} \), is appropriately reduced based upon the state of the synapse (ON or OFF, corresponding to \( R_{on1} \) or \( R_{on2} \)). Subsequently, the synaptic resistances form a voltage divider with the \( R_{base} \). This is also where the input currents are being summed up by the artificial soma. This summed current is being used as the input to the somatic transistor, with the output voltage (OFETs characteristic curve) being determined by the Drain voltage. Because the \( R_{out} \) is used as a voltage divider, its value should be kept sufficiently large to allow for the maximum output of the neuron.

**III. SINGLE NEURON SIMULATION**

The simulation and analysis presented in this and the following sections are based on the electrical characteristics of the elements, organic bistable devices and organic field effect transistors, developed in our laboratory.

### A. Single Synthetic Neuron Behavior

As can be seen in Figure 2A, for a binary connection weight, a single neuron consists of a single bistable device per input, a single transistor, and two resistors. A single bistable device, being in either an ON or OFF state, corresponding to either low \( (R_{on}) \) or high \( (R_{off}) \) resistance, is used as a binary connection weight. Figure 3, depicting output voltage plotted against input voltage, reveals that the activation function of the synthetic neuron looks remarkably similar to a sigmoidal activation function. It demonstrates that only a single transistor with one bistable device per input is sufficient to obtain a sigmoidal activation function commonly associated with analog artificial neural networks. Equation 2 shows the formula used to calculate the error:

\[
err = \frac{|p| - |q|}{|p|}
\]

Equation 2. Equation used to calculate the error.

where \( p \) denotes the value of ANN trained in Matlab, and \( q \) is the SNN value obtained from the simulation.

**Figure 2.** Schematics of a single neuron with two inputs (organic bistable devices, or OBD, analogous to synapses in biological neurons, are used as connection weights in artificial neurons), denoted by \( V_{in1} \) and \( V_{in2} \), and one output, marked \( V_{out} \), from an organic field-effect transistor (OFET) used for summation of memristive currents and to provide activation function (OFET along with two resistors forms a synthetic soma used to provide a bounded output based on received input). (A) The actual circuit proposed, with bistable devices used to function as artificial synapses. (B) A bistable device is replaced with two resistors, to simplify the process of circuit analysis.

**Figure 3.** (A) Comparison of activation functions of a conventional neural network unit (MATLAB’s *tansig* was modified to increase the upper and lower limits as well as its slope: \( y = \frac{1.33*1/2}{(1 + \exp(-0.8*%)) - 1}) \) with the synthetic neuron shown in Figure 2 with binary connection weights for output of a neuron (measured at a drain of an OFET) for input between -5V and 5V. The difference between these two functions, calculated according to Equation 2, was calculated to be approximately 5%. (B) The actual circuit used to obtain the Activation Function in (A), with values specified by electrical characteristics of elements obtained in our laboratory.
The symmetrical behavior of the activation function is realized because OFETs have a truly symmetrical design (source and drain are made of the same material and their assignment is purely arbitrary) which results in a remarkably symmetrical IV characteristics.

B. Single Synthetic Neuron as a Logic Gate

For some tasks binary connection weights are sufficient for proper operation [24],[31]. However, other tasks may require a finer granularity of the connection weights. Because bistable devices are formed at a cross-point of two wires, their physical size can be minimal. Therefore, increasing the smoothness of connection weights can be easily accomplished by increasing the total number of bistable devices assigned to an individual synapse. Three such devices, with only a single one being in an ON state, would result in a synapse with 4 possible values. However, setting a subset of these devices to an ON state would result in \(2^n\) possible values, with \(n\) being the number of bistable devices employed per connection weight. Making such devices with variable ON and OFF resistances (accomplished by varying their size) would result in more linear distribution of quantized connection weight values.

From the pattern recognition or classification perspective, a single artificial neuron can be used to perform a binary or two-dimensional (two-class) classification [24]. Table 1 demonstrates an example where a single synthetic neuron, with four bistable devices per input (16 quantized connection weight levels), was trained to behave like a logic gate. Two Boolean logic gates, AND and OR, were obtained. The ability of a single neuron to perform a binary or two-dimensional classification can be easily extended to a multi-dimensional classification capability of a network of neurons.

<table>
<thead>
<tr>
<th>(\text{AND gate} )</th>
<th>(\text{OR gate} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN1</td>
<td>IN2</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>5 0</td>
<td>0 3</td>
</tr>
<tr>
<td>0 5</td>
<td>0 3</td>
</tr>
<tr>
<td>5 5</td>
<td>2 2</td>
</tr>
</tbody>
</table>

IV. SIMULATION OF NETWORK OF SYNTHETIC NEURONS

To substantiate the efficacy of this approach, we carried out simulations showing that a network of synthetic neurons is able to correctly classify complex non-linear relationships. Next, we present two such examples. We use the term SNN to refer to simulations based on the synthetic neuron in Figure 2 and ANN to refer to conventional neural network simulations using Matlab (Neural Network Toolbox version 4.0.6), trained with backpropagation algorithm.

A. Synthetic Neural Network for Prediction of Water Hammer Directionality

Our group is involved in a project aiming at utilizing the effect of water hammer (a liquid traveling through a pipe will exert a forward, jerky motion on the pipe, due to a sudden closure of a valve) in aiding the locomotion of tethered robot. Previously, we have demonstrated that a neural network can be used to predict the direction of the water hammer impulsive force from a collection of discrete bend sensors placed along the pipe’s length [26]. From this prior dataset relating the shape of the pipe to the angle of propulsion we have retrained a neural network in Matlab. We then manually exported those weights to an SNN simulation (the network consisted of 4 input neurons, 4 hidden neurons, and one output neuron). Each synthetic neuron had inputs composed of three bistable devices which quantized the input weights to eight levels (weight quantization error = 7.9%) used for quantized connection weights. Figure 4 shows an example of a shape used for the shape-extraction experiment. Figure 5 demonstrates 20 distinct propulsion directions (given in radians) associated with 20 distinct shapes, with outputs from ANN and SNN. Shape 0 (direction along a straight, or non-bent hose) is a reference vector (angle equal to 1.57 [rad]); shapes above or below the horizontal axis indicate shapes that resulted in angle of propulsion either to the left or to the right of the reference shape. It can be seen that our SNN is not only able to correctly classify two distinct patterns (left or right propulsion), but is remarkably accurate (average error = 8.8%, calculated according to Equation 2) in relation to the ability to correctly identify the angle.
B. Synthetic Neural Network for Wall Following Robot

To further investigate the effectiveness of the SNN to map complex input-output relationships, we trained it to perform a task of a robotic wall following. We obtained input data from 9 sonar sensors mounted on a mobile NOMAD SCOUT2 robot, with the output being motor commands sent to two individual motors. With this data set, a conventional ANN was trained in Matlab to an error of less than one percent. The data set that the network was trained on consisted of 100 data points. However, the testing set consisted only of a subset of the training data set, with a total of 20 examples (seen in Figure 6). The neural network contained 9 input cells, 15 hidden cells, and two output cells. The connection weights were then manually exported to SNN. Figure 6 demonstrates the comparison of the output of ANN trained in MATLAB to the normalized SNN output, with Figure 6A showing the right wheel output, while Figure 6B depicts the output of the left wheel. The average error, computed using Equation 2, of the right wheel output was about 8.8% while the average error of the left wheel output was about 11.6%, resulting in an average output error of 10.2%.

![Figure 6. Comparison of the ANN and SNN outputs for a robotic wall following task: the average SNN error being approximately 9.6%. (A) The output of the right wheel, with SNN accuracy of 8.6%. (B) The output of the left wheel, with an average the SNN error of about 10.5%.

VI. IMPLEMENTATION

So far we have only simulated the operation of a synthetic neuron via a simulation based on electrical properties of individual devices created in our laboratory. We have yet to demonstrate the operation of a tangible synthetic neuron. Because the design incorporates three separate electrical elements, namely transistors, resistors, and organic bistable devices, manufacturing such device presents considerable challenges. There are two possible ways that this could be accomplished, each one with its advantages and disadvantages.

The first possibility for creating a tangible neuron, or a network of such neurons, is creating the entire design on a single chip. The most obvious benefit of this solution is reduced size of the entire network of neurons. However, the tradeoff lies in increased complexity of the manufacturing process. Individual elements, such as the OFET of resistor, are made from chemical elements that would have to be either selectively deposited or removed from the chip.

An approach alternative the first solution, is to manufacture individual electrical elements on separate chips and then connect them all together, for instance via wire bonding technique, according to the proposed architecture. This would be a simpler method to the single-chip approach, as it would not require the use of expensive and intensive manufacturing processes of dissimilar materials on a single dye. However, from utilitarian perspective, this would result in a significantly larger device.

VII. CONCLUSION

We have demonstrated, via computer simulation, that a simplified hardware neuron can be implemented with only three distinct electrical elements; two organic bistable devices (for two-input neuron), one transistor, and two resistors. This neuron will possess the necessary functionality of an artificial neuron, namely synaptic multiplication, somatic summation and activation function. We have also demonstrated that a single neuron, with a connection weight quantized to sixteen levels, can successfully function as either AND or OR logic gate. Additionally, we have shown that the SNN based on synthetic neurons can be trained to perform, with a great deal of accuracy, a real-world task, namely the ability to
distinguish the directed propulsion of a hose due to water hammer effect. Furthermore, we also have demonstrated that the SNN can be used for another complex, non-linear classification of a wall following routine. The analysis of our SNN presented here was based on the electrical characteristics of devices, namely organic bistable devices and organic field effect transistors, created in our laboratory.

At present the network does not possess built-in learning capability and the connection weights have to be imported manually. However, addition of an external circuit used for the purposes of training will allow for the network to be fully self-contained without the need of external training and weight import.

ACKNOWLEDGMENT

We would also like to acknowledge the help of Xiaoting Yang, Rachelle Cobb, Yanzhe Cui, and Brian A. Bailey as well as our colleagues at the Collaborative Mechatronics Laboratory and Department of Physics and Astronomy at University of Denver.

REFERENCES