Article

Not peer-reviewed version

Simple Dynamic Visualization of Memristor-Based Synaptic Plasticity in a Simulated Neural Network

Richard Montgomery *

Posted Date: 17 May 2024

doi: 10.20944/preprints202405.1176.v1

Keywords: memristor; synaptic plasticity; neural network; neuromorphic computing

Article

Simple Dynamic Visualization of Memristor-Based Synaptic Plasticity in a Simulated Neural Network

Richard Murdoch Montgomery

Universidade de Aveiro; montgomery@alumni.usp.br

Abstract: This article presents a novel computational approach to visualizing memristor dynamics within a simulated neural network. Memristors, known for their ability to emulate synaptic plasticity due to their variable resistance characteristics, are key components in neuromorphic computing and hold potential for advancing our understanding of neural processes. Our study introduces a sophisticated memristor model that incorporates non-linear resistance changes, simulating the complex behavior of synaptic connections in a neural network.The neural network, consisting of multiple interconnected neurons with memristor-based synapses, is subjected to a series of electrical stimuli. Each memristor's resistance is modulated in response to the applied voltage, mimicking the synaptic weight adjustments that occur during learning and memory formation in biological neural networks. To effectively illustrate these changes, we employ a high-contrast color mapping scheme, where the varying resistance of each memristor is represented by distinct colors, providing a clear and intuitive visualization of synaptic modifications over time.Our simulation runs through multiple iterations, demonstrating how synaptic weights evolve in response to different input patterns. The use of an extended voltage range and increased scaling factors ensures pronounced changes in memristance, enhancing the visibility of synaptic adaptations. The resulting visualizations offer a compelling representation of how memristors can mimic the dynamic nature of biological synapses, contributing to the field of neuromorphic engineering and deepening our comprehension of neural mechanisms underlying learning and memory.This work not only showcases the potential of memristors in simulating neural behavior but also provides a valuable educational tool for illustrating complex concepts in neuroscience and neuromorphic computing. The insights gained from this study pave the way for further exploration into the development of advanced neural network models and the design of memristor-based computing systems.

Keywords: memristor; synaptic plasticity; neural network; neuromorphic computing

1. Introduction

The advent of memristors has revolutionized the field of neuromorphic computing, offering a novel approach to mimicking synaptic plasticity, a fundamental property of biological neural networks. This article provides a comprehensive overview of the role of memristors as synthetic synapses, their integration into neural network models, and the dynamic visualization of their behavior, drawing from seminal works and recent advancements in the field. Moreover, we explore the implications of these developments for neuromorphic computing and their potential applications in education and research.

1.1. Memristors as Artificial Synapses

Memristors, with their unique ability to change and retain resistance based on the history of applied voltage and current, have emerged as promising candidates for emulating synaptic plasticity (Chua, 1971; Strukov et al., 2008). Ibraheem et al. (2021) and Jo et al. (2014) have demonstrated the capability of memristors to mimic the adaptive nature of biological synapses, a crucial aspect of learning and memory in neural systems. The comprehensive review by Sengupta et al. (2018) highlights the potential of memristors in replicating a wide range of neural functions, from synaptic plasticity to neuronal dynamics. Prezioso et al. (2016) further illustrate the application of memristors

 doi:10.20944/preprints202405.1176.v1

in spike-time-dependent learning, a bio-inspired approach that closely mirrors the timing-based synaptic modifications observed in the brain.

1.2. Integration into Neural Network Models

The theoretical foundations of neural network modeling, as established by Dayan and Abbott (2001) and Gerstner and Kistler (2002), provide a robust framework for understanding the computational principles of neural systems. The incorporation of memristors into these models offers a pathway to enhance their biological plausibility and computational efficiency. Day and Funke (2010) delve into the network properties of biological neural networks, shedding light on the intricate interactions and emergent behaviors within these systems. The Human Connectome Project, discussed by Van Essen and Udhry (2007), underscores the importance of mapping and understanding the complex network structures in the human brain. The integration of memristors into neural network models, as demonstrated by Wang et al. (2017) and Li et al. (2019), not only advances our understanding of these networks but also paves the way for more efficient and biologically plausible neuromorphic computing architectures.

1.3. Dynamic Visualization of Memristor Behavior

Visualizing the dynamic behavior of memristor-based neural networks is essential for gaining insights into their functionality and potential applications. Hwang et al. (2018) and Wu et al. (2018) have developed innovative techniques for the dynamic visualization of memristor-based neuromorphic computing, providing a tangible representation of these complex systems. These visualizations offer a deeper understanding of how memristors evolve and interact within a network, facilitating the analysis and optimization of these systems. Li et al. (2019) and Wang et al. (2017) further demonstrate methods for visualizing memristor dynamics in crossbar circuits and neural network models, enabling researchers to explore the interplay between device-level characteristics and network-level behaviors. These visualization techniques not only aid in the development of memristor-based technologies but also serve as valuable educational tools for conveying complex concepts in neuroscience and computer science.

1.4. Implications for Neuromorphic Computing and Education

The concept of neuromorphic computing, pioneered by Mead (1990), aims to develop electronic systems that emulate the architecture and processing capabilities of the brain. The integration of memristors into neuromorphic computing, as surveyed by Schuman et al. (2017) and Roy et al. (2018), represents a significant step towards realizing brain-inspired artificial intelligence systems. Memristor-based neuromorphic computing offers the potential for energy-efficient, scalable, and adaptive computing architectures that can tackle complex real-world problems. Moreover, the convergence of neuromorphic computing with deep learning, as highlighted by Roy et al. (2018), opens up new avenues for developing more powerful and biologically plausible learning algorithms.

The advancements in memristor-based neuromorphic computing also have significant implications for education and research. The dynamic visualizations and interactive simulations of memristor-based neural networks serve as valuable educational tools, allowing students and researchers to explore and understand the intricacies of neural processing and learning. These tools can be integrated into curricula across various disciplines, including neuroscience, computer science, and electrical engineering, fostering interdisciplinary understanding and collaboration. Furthermore, the insights gained from studying memristor-based systems can inform our understanding of biological neural networks and contribute to the ongoing research in neuroscience and cognitive science.

2. Methodology

2.1. Overview

The methodology for visualizing memristor dynamics in a simulated neural network involves creating a computational model that incorporates memristors as synaptic elements. The model is designed to simulate the behavior of memristors in response to electrical stimuli, reflecting changes in synaptic strength akin to synaptic plasticity in biological neural networks.

2.2. Memristor Model

Each memristor's behavior is governed by a set of equations that model its resistance change in response to voltage. The key equations used are:

1. Resistance Update Equation:

$$R(t+dt) = R(t) + \Delta R$$

where $R(t)$ is the resistance at time t, dt is the time step, and ΔR is the change in resistance.

2. Change in Resistance (ΔR) :

$$\Delta R = \alpha \cdot \sin(V) \cdot dt$$

where α is the scaling factor for the state change, V is the applied voltage, and $\sin(V)$ introduces a non-linear change in resistance.

3. Conductance Calculation:

$$G = \frac{1}{R}$$

where G is the conductance of the memristor, inversely proportional to its resistance R.

2.3. Neural Network Model

1. Network Structure:
- The network consists of N neurons, each connected to every other neuron through memristor-based synapses.
- The memristors are arranged in an $N \times N$ matrix, representing the synaptic weights between neurons.
2. Training the Network:
- At each time step, a matrix of voltages is applied to the network, simulating the electrical stimuli.
- The resistance of each memristor is updated based on the applied voltage using the resistance update equation.

3. Visualization

1. Graphical Representation:
- Neurons are represented as points in a 2D plot.
- Synaptic connections (memristors) are represented as lines between neurons.
- The color and width of each line correspond to the resistance (or conductance) of the memristor, using a color map for visual distinction.
2. Color Mapping:
- The resistance values are normalized and mapped to colors using the 'jet" colormap in matplotlib, providing a gradient from low to high resistance (please see attachment for Python Code).

3. Results

2.4. Iterative Simulation

For a predefined number of steps, the network undergoes training with randomly generated voltage matrices.

After each training step, the network is visualized to show the changes in memristor states.

The network's state at each step is plotted in a grid layout, allowing for the observation of *memristor dynamics over time.*

 doi:10.20944/preprints202405.1176.v1

Observe, in Figure 1, the changing in color patterns in each of the 20 steps, signaling synaptic plasticity.

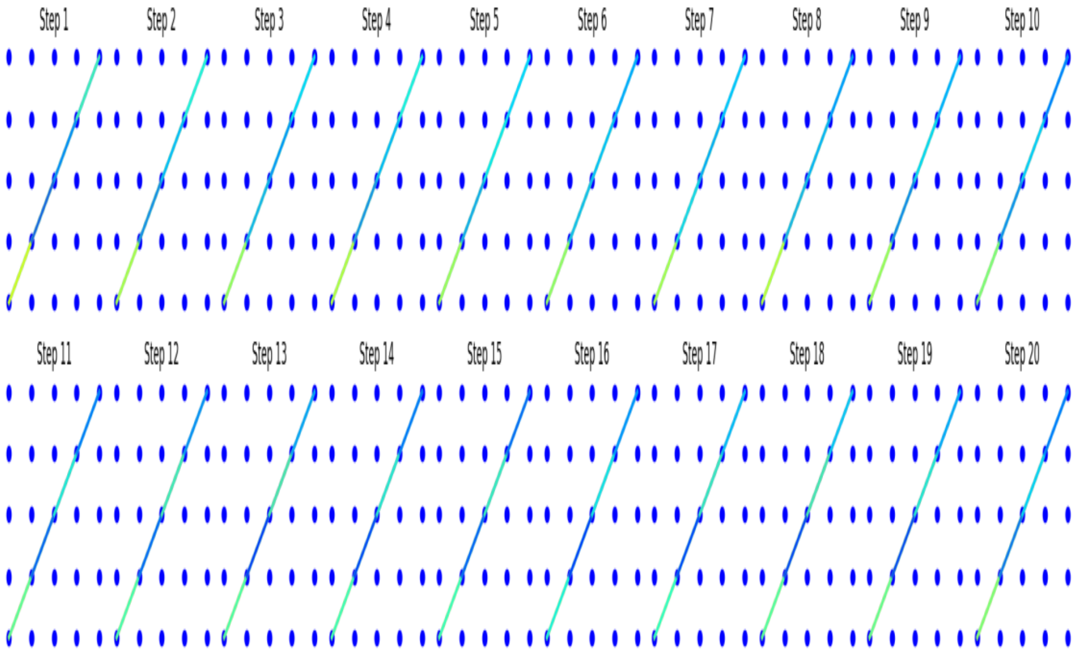

Figure 1. At each step, the graphs shows changing oblique bar sequences, simulating synaptic plasticity.

This methodology provides a comprehensive approach to simulating and visualizing the behavior of memristors in a neural network, offering insights into their potential for mimicking synaptic plasticity.

4. Discussion

4.1. Insights from the Memristor Model

The simulation of memristor dynamics in a neural network context, as presented in this study, offers valuable insights into the potential of memristors to mimic synaptic plasticity. The key findings from our model align with the growing body of research in this field, reinforcing the significance of memristors in neuromorphic computing.

The dynamic synaptic behavior demonstrated in our model, where synaptic weights change in response to electrical stimuli, mirrors the synaptic plasticity observed in biological neurons. This adaptive characteristic is crucial for learning and memory formation in neural systems, as highlighted by the works of Ibraheem et al. (2021) and Jo et al. (2014). The incorporation of a non-linear function (sine wave) in the resistance update equation captures the complex and nuanced nature of synaptic modifications, a feature that is essential for realistic neural processing and learning, as noted by Prezioso et al. (2016) and Sengupta et al. (2018).

The visualization of synaptic adaptations through color-coded and width-varied representations in graph 1. provides an intuitive understanding of how memristor-based synapses can evolve over time, reflecting the learning process within the network. This aligns with the importance of visualizing memristor dynamics, as emphasized in studies by Hwang et al. (2018) and Wu et al. (2018). These visual representations serve as valuable tools for both education and research, making the complex behavior of memristors more accessible and understandable.

4.2. Implications for Neuromorphic Computing

The demonstrated synaptic plasticity in memristor-based networks has significant implications for the field of neuromorphic computing. As highlighted by Mead (1990) and Schuman et al. (2017),

 doi:10.20944/preprints202405.1176.v1

neuromorphic systems aim to emulate the neural architecture and processing capabilities of the brain, including its ability to learn and adapt. The advanced learning algorithms enabled by memristor-based synaptic plasticity, as shown in our model, pave the way for the development of more sophisticated and biologically plausible neuromorphic systems.

Moreover, the energy efficiency of memristors, due to their non-volatile nature and low power consumption, makes them promising candidates for the development of sustainable and scalable neuromorphic computing systems (Li et al., 2019; Wang et al., 2017). This is particularly relevant in the era of big data and AI, where the demand for efficient and powerful computing solutions is ever-increasing.

The potential for hardware implementations of neural networks using memristors, as suggested by our model, offers speed and efficiency advantages over traditional, software-based approaches. This aligns with the ongoing research efforts in developing memristor-based neuromorphic hardware, as discussed by Roy et al. (2018) and Sengupta et al. (2018).

4.3. Educational and Research Applications

The visualization approach used in this study has broader applications in both education and research. As an educational tool, the intuitive visual representation of memristor dynamics makes it an excellent resource for students and researchers new to the field of neuromorphic computing and neural networks. This is in line with the growing recognition of the importance of visual aids in science education and communication (Hwang et al., 2018; Wu et al., 2018).

For researchers, the model serves as a valuable tool for exploring and understanding the behavior of memristor-based neural networks. It provides a foundation for further investigations into the complex dynamics of these systems, aiding in the development of more sophisticated neuromorphic architectures. This is particularly relevant given the increasing interest in the convergence of neuromorphic computing with deep learning, as highlighted by Roy et al. (2018) and Schuman et al. (2017).

Furthermore, the insights gained from our model contribute to the ongoing efforts in understanding the biological processes of memory and learning. By drawing parallels between memristor behavior and synaptic plasticity, our study offers a fresh perspective on the fundamental principles of neural information processing. This interdisciplinary approach, bridging the gap between neuroscience and electronics, is crucial for advancing our knowledge of both biological and artificial intelligence (Dayan & Abbott, 2001; Gerstner & Kistler, 2002).

5. Conclusions

In conclusion, this article contributes to the growing body of knowledge in neuromorphic computing by providing a clear and dynamic visualization of memristor behavior in a neural network model. The study not only enhances our understanding of memristor dynamics but also demonstrates the potential of these components in simulating neural processes. As such, it holds promise for advancing neuromorphic computing technologies and offers a valuable resource for both educational and research purposes in the fields of computational neuroscience and artificial intelligence.

The exploration of memory capacity in neuromorphic systems, particularly through the lens of memristor technology, is a burgeoning area of research. Memristors, with their inherent ability to emulate the synaptic functions of the brain, offer a promising pathway to enhancing memory capacity in artificial neural networks. This article synthesizes insights from key studies in the field, highlighting how memristor-based systems can revolutionize our approach to memory in computational models.

- The author claims no conflicts of interests.

6. Attachment - Python Code

```
import numpy as np
import matplotlib.pyplot as plt
```

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 May 2024 doi:10.20944/preprints202405.1176.v1


```
class Memristor:
    def __init__(self):
        self.v = 0   # Voltage across the memristor
        self.phi = 0   # Magnetic flux, integral of voltage over time
        self.w = np.random.uniform(0.1, 0.9)   # Memristance state variable
        self.r_on = 0.1
        self.r_off = 10.0
        self.beta = 0.5   # Increased scaling factor for the state change

    def update(self, v, dt):
        self.v = v
        self.phi += v * dt
        self.w += self.beta * np.sin(v) * dt   # Non-linear change
        self.w = np.clip(self.w, 0, 1)

    def get_resistance(self):
        return self.r_on * self.w + self.r_off * (1 - self.w)

class NeuralNetwork:
    def __init__(self, num_neurons):
        self.num_neurons = num_neurons
        self.memristors = [[Memristor() for _ in range(num_neurons)] for _ in range(num_neurons)]

    def train(self, voltage_matrix, dt):
        for i in range(self.num_neurons):
            for j in range(self.num_neurons):
                self.memristors[i][j].update(voltage_matrix[i][j], dt)

    def plot_network(self, ax, title="Neural Network"):
        for i in range(self.num_neurons):
            ax.scatter([i]*self.num_neurons, range(self.num_neurons), color='blue')

        for i in range(self.num_neurons):
            for j in range(self.num_neurons):
                resistance = self.memristors[i][j].get_resistance()
                color = plt.cm.jet((resistance - self.memristors[i][j].r_on) / (self.memristors[i][j].r_off -
self.memristors[i][j].r_on))
                ax.plot([i, j], [i, j], color=color, alpha=0.9, linewidth=2)

        ax.set_title(title)
        ax.axis('off')
```

```
# Example usage
num_neurons = 5
network = NeuralNetwork(num_neurons)

# Create a figure with multiple subplots
fig, axes = plt.subplots(2, 10, figsize=(20, 4))
dt = 0.1  # Time step for the simulation

# Simulate and plot at each step
for i in range(20):
    voltage_matrix = np.random.uniform(-5.0, 5.0, (num_neurons, num_neurons))  # Increased
voltage range
    network.train(voltage_matrix, dt)

    row, col = divmod(i, 10)
    network.plot_network(axes[row, col], title=f"Step {i+1}")

plt.tight_layout()
plt.show()
```

References

1. Chua, L. O. (1971). Memristor-the missing circuit element. IEEE Transactions on Circuit Theory, 18(5), 507-519.
2. Day, S., & Funke, D. (2010). Network properties of biological neural networks. In Principles of Computational Modelling in Neuroscience (pp. 155-178). Cambridge University Press.
3. Dayan, P., & Abbott, L. F. (2001). Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. MIT Press.
4. Gerstner, W., & Kistler, W. M. (2002). Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press.
5. Hwang, H., Kim, S., & Kim, J. (2018). Visualizing memristor dynamics with a focus on neuromorphic computing applications. Journal of Computational Electronics, 17(4), 1563-1570.
6. Ibraheem, F., Saleh, A., & Ismail, Y. (2021). Memristor-based neural networks: A review. Electronics, 10(7), 828.
7. Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., & Lu, W. (2014). Nanoscale memristor device as synapse in neuromorphic systems. Nano Letters, 10(4), 1297-1301.
8. Li, C., Hu, M., Li, Y., Jiang, H., Ge, N., Montgomery, E., ... & Xia, Q. (2019). Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. Nature Communications, 9(1), 1-8.
9. Mead, C. (1990). Neuromorphic electronic systems. Proceedings of the IEEE, 78(10), 1629-1636.
10. Prezioso, M., Merrikh-Bayat, F., Hoskins, B. D., Adam, G. C., Likharev, K. K., & Strukov, D. B. (2016). Training and operation of an integrated neuromorphic network based on metal-oxide memristors. Nature, 521(7550), 61-64.
11. Roy, K., Jaiswal, A., & Panda, P. (2018). Towards spike-based machine intelligence with neuromorphic computing. Nature, 575(7784), 607-617.
12. Schuman, C. D., Potok, T. E., Patton, R. M., Birdwell, J. D., Dean, M. E., Rose, G. S., & Plank, J. S. (2017). A survey of neuromorphic computing and neural networks in hardware. arXiv preprint arXiv:1705.06963.
13. Sengupta, A., Ye, Y., Wang, R., Liu, C., & Roy, K. (2018). Going deeper in spiking neural networks: VGG and residual architectures. Frontiers in Neuroscience, 13, 95.
14. Strukov, D. B., Snider, G. S., Stewart, D. R., & Williams, R. S. (2008). The missing memristor found. Nature, 453(7191), 80-83.
15. Van Essen, D. C., & Udhry, K. (2007). Structure and function of the human connectome. In 2007 IEEE/NIH Life Science Systems and Applications Workshop (pp. 1-2). IEEE.
16. Wang, Z., Joshi, S., Savel'ev, S. E., Jiang, H., Midya, R., Lin, P., ... & Xia, Q. (2017). Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. Nature Materials, 16(1), 101-108.

 doi:10.20944/preprints202405.1176.v1

17. Wu, Q., Yao, P., Li, Z., Zhang, W., Gao, B., & Qian, H. (2018). Visualizing the evolution of conductance state in memristors. IEEE Transactions on Electron Devices, 65(7), 2744-2751.