OPEN ACCESS
Global Journal of Research in Engineering & Computer Sciences
ISSN: 2583-2727 (Online)
Volume 04 | Issue 03 | May-June | 2024
Journal homepage: https://gjrpublication.com/gjrecs/
Review Article

From Artificial Neural Networks to Spiking Neural Networks: A Comprehensive Review

*Prof. Deeksha[1] and Dr. Padmanayana[2]

[1]Research Scholar, Srinivas University, Mangalore, India
[2]Professor and Head, Department of Computer Science and Design, Srinivas Institute of Technology, India

DOI: 10.5281/zenodo.11618186 **Submission Date:** 28 April 2024 | **Published Date:** 12 June 2024

***Corresponding author:** Prof. Deeksha
Research Scholar, Srinivas University, Mangalore, India

Abstract
Artificial neural networks (ANNs) have demonstrated encouraging results in many applications, but when compared to biological neural networks (BNNs), they still fall short in many aspects Spiking neural networks (SNNs) bridge the gap between ANNs and BNNs by leveraging biologically realistic neurons. Spiking neural networks are of great interest in machine learning because they can achieve high-performance computing with low power consumption. This article provides a comparative analysis of ANN and SNN.

Keywords: Artificial neural networks, spiking neural networks, activation functions, Moore's law.

1. INTRODUCTION
Moore's Law is a principle that states that the number of transistors on a silicon chip will double approximately every eighteen months. However, with the advancement in technology, transistors and other components have now reached a nanoscale level. Further reducing the size of these components would lead to excessive heat or heat dissipation, rendering the circuit inoperable. As a result, the size of transistors and other components cannot be reduced any further. Most hardware today is based on von Neumann architecture. The von Neumann architecture separates the memory and the computations. The computations are executed in the form of programs, which are sequences of machine instructions. Instructions are performed by a processor. A processor instruction usually has several arguments that it takes from processor registers (small but very fast memory cells located in the processor). At that, the instructions and most of the data are stored in the memory separately from the processor. The processor and the memory are connected by a data bus by which the processor receives instructions and data from the memory. The first bottleneck of this architecture is the limited throughput of the data bus between the memory and the processor. During the execution of a program, the data is loaded mainly by the transfer of processing data from/to random Access Memory (RAM). Moreover, the maximum throughput of the data bus is much less than the speed at which the processor can process data. Another important limitation is the big difference in the speed of RAM and processor registers. This can cause latency and processor downtime while it fetches data from the memory. This phenomenon is known as the von Neumann bottleneck[1].To surpass this physical limit, researchers have been exploring two fields - quantum computing and neuromorphic computing. Quantum computing utilizes quantum states to perform calculations and is highly powerful, making it useful in some areas due to its speed and ability to solve large computational problems. Neuromorphic computing, on the other hand, mimics biological architectures in computing systems and is efficient and flexible, allowing it to quickly adapt to changes in events.

Neuromorphic Computing, a concept pioneered in the late 1980s, is receiving a lot of attention lately due to its promise of reducing the computational energy, latency, as well as learning complexity in artificial neural networks[2].ANNs are computational models inspired by the human brain, designed to process information through interconnected neurons. ANNs learn from examples and adjust synaptic connections, similar to biological systems, for tasks like pattern recognition and data classification [3].

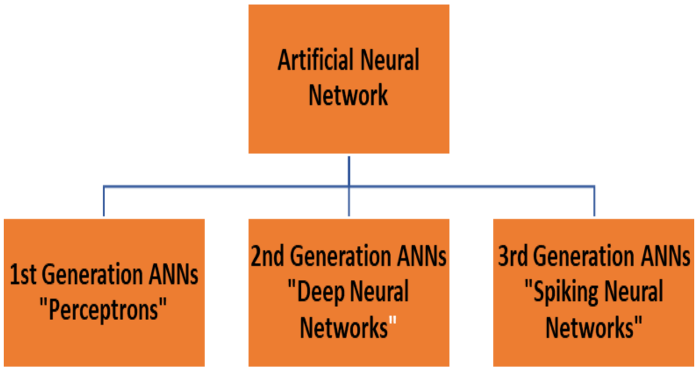

Fig 1: Three generations of ANN

2. CLASSIFICATION OF ANN

ANNs are massively parallel computing systems made up of a vast number of interconnected, basic processors.The ANNs can be separated into three generations based on their computational units and performance (Fig1).

The first generation of Artificial Neural Networks (ANNs) began in 1943 with the work of McCulloch and Pitts. They developed a computational model for neural networks where each neuron is called a "perceptron". Their model was later improved upon with extra hidden layers (Multi-Layer Perceptron) for better accuracy, called MADALINE, by Widrow and his students in the 1960s. However, the first-generation ANNs were far from biological models and were simply giving digital outputs. They were decision trees based on if-else conditions.[4]

The second generation of ANNs built upon the previous generation by applying functions into the decision trees of the first-generation models. The functions work between each visible and hidden layer of the perceptron and create the structure called "deep neural networks". Therefore, second-generation models are closer to biological neural networks. The functions of the second-generation models are still an active area of research, and the existing models are in great demand from markets and science. Most of the current developments in Artificial Intelligence (AI) are based on these second-generation models, and they have proven their accuracy in cognitive processes. Some of the second-generation neural network models are

2.1 FEED FORWARD NEURAL NETWORK

The most common type of ANN is the Feedforward Artificial Neural Network (FFANN). It is used for prediction and regression problems. Sometimes, an FFANN is referred to as a multi-layer perceptron (MLP). There is an input layer of neurons which accepts input data, and one or multiple layers of hidden neurons, and an output layer of neurons giving the output data. Each neuron in the previous layer is connected to the neurons in the following layer[5].

2.2 CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network is a class of ANN used for image analysis such as pattern recognition and object identification using convolution operation between matrices. Such a network has four types of layers starting from the first: convolutional layer, optional non-linearity layer, pooling layer, and fully connected layer[5]

2.3 RADIAL BASIS FUNCTIONAL NEURAL NETWORK

An input vector, an output layer with one node for each category, an RBF neuron layer, and an RBF neuron layer make up a Radial Basis Function Network. The input is compared to samples from the training set, where a prototype neuron is stored for each neuron, as part of the classification process.

2.4 RECURRENT NEURAL NETWORK

A Recurrent Neural Network differs from a FFANN by having at least one feedback connection, thus having at least one feedback loop. It is used for prediction problems and pattern recognition. As stated in [5]it provides feedback from its output to other neuron inputs. RNN can have a hidden layer of neurons or the feedback can be fed back to the same neuron, therefore creating a self-feedback loop.

2.5 LSTM: LONGSHORT-TERM MEMORY

Long Short-Term Memory (LSTM), is a novel recurrent network architecture designed to address the vanishing gradient problem in recurrent backpropagation[6]. A memory cell is introduced by LSTM networks. They can process data with gaps in memory. When employing RNNs, one issue that may be considered is the time delay. However, when we have a large amount of pertinent data and wish to extract significant information from it, we should apply LSTMs if our RNN fails.

Artificial neural networks simplify complex processes through weighted sums and activation functions. An artificial neuron has multiple inputs, and each input has a specific weight assigned to it to represent its importance. The inputs are added up and then passed through an activation function to produce an output. This process enables artificial neurons to transform inputs into outputs. Artificial neural networks, by imitating biological neural networks, can accomplish remarkable feats of learning and adaptation. They can handle data that is noisy, incomplete, or ambiguous and can generalize well to new scenarios. Additionally, they can learn from their errors and enhance their performance with time. This makes them extremely dynamic and powerful tools for machine learning and artificial intelligence.

3. BIOLOGICAL NEURON VS ARTIFICIAL NEURON: A COMPARISON

An artificial neuron can be crudely compared to a biological neuron in terms of its connections and functioning. In this analogy, the connections between nodes represent the axons and dendrites, the connection weights represent the synapses, and the threshold approximates the activity in thesoma. Both the biological neuron and the artificial neural network learn by gradually adjusting the magnitudes of the weights or synapses strengths [7].

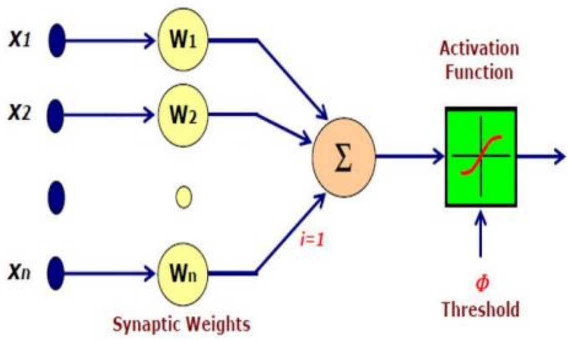

Fig 2: (ANN) model based on the biological neural system

3.1 BASIC ELEMENTS OF ANN

A neuron consists of three basic components –weights, thresholds, and a single activation function. An Artificial neural network (ANN) model based on the biological neural system is shown in Figure 2. In ANN the inputs are passed through the weighted summation function ∑ followed by the non-linearity function, the non-linearity function used is sigmoid, Tanh, ReLu.

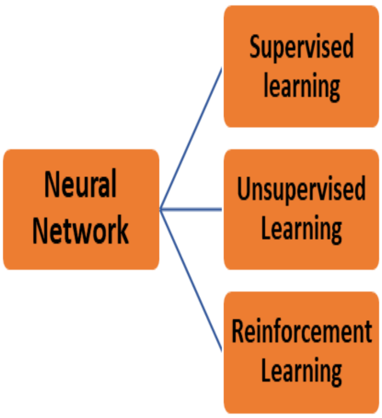

Fig 3: Types of learning in ANN

In Figure 3, a brief classification of different learning algorithms is presented. Training refers to the process of teaching the neural network to adjust its weight and bias. On the other hand, learning is the internal process that occurs during training where the artificial neural system updates or adapts its weights and biases. The process of training an artificial neural network involves both training and learning. Once a network has been configured for a specific application, it is ready for training. To begin this process, the initial weights are determined randomly. then the training, or learning, starts. There are two techniques to training supervised and unsupervised. Supervised training is a means for supplying the required output to the network, either by manually "grading" the network's performance or by feeding the desired outputs into the inputs. Unsupervised training is where the network must make sense of the inputs without external help.

Different Training /Learning procedures available in ANN are

1. Supervised learning
2. Unsupervised learning
3. Reinforced learning

3.2 TYPES OF LEARNING IN NEURAL NETWORKS

3.2.1 Supervised Learning

This kind of learning is carried out under a supervision of a teacher as the name would imply. This process of learning is interdependent. The input vector is fed to the network during ANN training under supervised learning, and the network outputs an output vector. A comparison is made with this output vector, with the intended output vector in mind. If there is a discrepancy between the intended output vector and the actual output, an error signal is produced. The weights are changed based on this error signal until the intended and actual outputs coincide.

3.2.2 Unsupervised Learning

This type of learning, as the name suggests, operates independently without the guidance of a teacher. During the training of an Artificial Neural Network (ANN) using unsupervised learning, input vectors of similar types are combined to form clusters. Upon application of a new input pattern, the neural network provides an output response indicating the class to which the input pattern belongs. No feedback from the environment is given regarding the desired output or its correctness. Therefore, in this learning paradigm, the network itself must discover patterns, features, and relationships within the input data.

3.2.3 Reinforcement Learning

This kind of learning is used to enhance the network over some important information, as the name implies.

This learning process is similar to supervised learning; however, we might have very little information. During the training of the network under reinforcement learning, the network receives some feedback from the environment. This makes it somewhat similar to supervised learning. However, the feedback obtained here is evaluative, not instructive, which means there is no teacher as in supervised learning. After receiving the feedback, the network performs adjustments of the weights to get better critical information in the future.[8]

3.3 ACTIVATION FUNCTIONS

A mathematical formula known as an activation function establishes the result of each neural network component, such as a neuron or perceptron. It receives information from every neuron and converts it into an output, which is often in the range of -1 to +1. It could be described as the additional weight or effort placed on top of the input in order to get a precise result. Additionally, we can use ANN activation functions on the input to get the desired result precisely

3.3.1 Threshold Function

Threshold function is commonly referred to as a Heaviside function. In neural computation, such a neuron is referred to as the McCulloch–Pitts model, in recognition of the pioneering work done by McCulloch and Pitts (1943). In this model, the output of a neuron takes on the value of 1 if the induced local field of that neuron is nonnegative, and 0 otherwise. This statement describes the all-or-none property of the McCulloch–Pitts model[8]

3.3.2 Sigmoid Function

The sigmoid function, whose graph is “S”-shaped, is by far the most common form of activation function used in the construction of neural networks. It is defined as a strictly increasing function that exhibits a graceful balance between linear and nonlinear behavior[8].

3.3.3 The TanH function

Is zero-centered making it easier to model inputs that are strongly negative strongly positive or neutral.

3.3.4 The ReLu function

The ReLu function is highly computationally efficient but is not able to process inputs that approach zero or negative.

3.3.5 The Leaky ReLu

The leaky Relu function has a small positive slope in its negative area, enabling it to process zero or negative values.

3.3.6 The Parametric ReLu function
Parametric ReLu function allows the negative slope to be learned, performing backpropagation to learn the most effective slope for zero and negative input values.

3.3.7 SoftMax
SoftMax is a special activation function used for output neurons. It normalizes outputs for each class between 0 and 1, and returns the probability that the input belongs to a specific class.

3.3.8 Swish
It is a new activation function discovered by Google researchers. It performs better than ReLu with a similar level of computational efficiency.

4. Neural Network Applications
[9]discusses various aspects of pattern recognition, including the definition of pattern classes, feature extraction, classifier design, and performance evaluation. It highlights the challenges in recognizing complex patterns with arbitrary orientation, location, and scale. Artificial Neural Networks. ANNs are presented as a powerful tool for pattern recognition due to their ability to learn complex nonlinear input-output relationships, adapt to data, and provide efficient learning algorithms.

[10]discusses the application of artificial neural networks (ANNs) to clinical medicine, highlighting their ability to identify complex patterns in data that may not be apparent through traditional analyses. NNs have been applied to various clinical diagnoses, including appendicitis and biopsy examinations, with the potential to improve diagnostic accuracy and patient outcomes.

As discussed in [11]ANNs are used for modeling solar steam generating plants, estimating heating loads, and predicting energy consumption in passive solar buildings. energy systems problems1. ANNs can handle noisy and incomplete data, non-linear problems, and can perform predictions at high speed. ANNs have been applied in various energy-related fields such as solar energy, heating loads of buildings, and power systems for tasks like system modeling, performance prediction, and optimization.

4.1 SPIKING NEURAL NETWORK
Spiking Neural Networks (SNNs) belong to the third generation of neural network models, which aim to increase the level of realism in neural simulations. In contrast to traditional neural networks, SNNs incorporate the concept of time into their operating model. Instead of firing at each propagation cycle, neurons in SNNs only fire when a membrane electrical charge reaches a specific value. This unique feature of SNNs allows for a more accurate representation of biological neural networks, making them particularly useful in fields such as neuroscience and artificial intelligence.[12]

In SNN, several models are commonly used including the Hodgkin-Huxley model (HH model), integrate and fire model (IF model), leaky integrate and fire model (LIF model), Izhikevich model, and spike response model (SRM).

4.1.1 Hodgkin–Huxley (HH) Model
Hodgkin and Huxley conducted experiments on the giant axon of a squid and found that the generation of an action potential involves two types of ion channels, namely the K+ channel and the Na+ channel. Although the Hodgkin-Huxley model is biologically accurate, it has some limitations as it only describes the channels and flow of ions in the neuron when generating spikes. Additionally, it requires large computational resources and is not feasible for large-scale simulations.[13]

4.1.2 Leaky Integrate and Fire (LIF) Model
The integrate-and-fire (IF) model is a type of model used to study the behavior of neurons. In this model, the input current is integrated over time until the membrane potential reaches a threshold, without considering the biological ion channel behavior. To reflect the diffusion of ions that occurs through the membrane when some equilibrium is not reached in the cell, the leaky integrate-and-fire (LIF) model introduces a "leak" term to the IF model. LIF is a simplified version of the process of action potentials that only focuses on the changes of membrane potential at a macroscopic level. Its computational procedure is easier, and it is widely used. However, LIF cannot stimulate more neuronal behaviors except for leakage, accumulation, and threshold excitation.[14]

4.1.3 Izhikevich model

The Izhikevich model is a computational model that aims to strike a balance between biological accuracy and computational efficiency. This model has the ability to simulate more than 20 different neuronal behaviors and can achieve this in just 1 millisecond using only 13 floating-point operations. This is significantly less than what is required by the HH model. It is crucial for a neuron model to find the right balance between bio-mimicry and computational cost in order to be integrated on a large scale. Therefore, finding an appropriate neuron model is one of the key research problems of the present time.[14]

4.2 Learning in SNN

Synaptic plasticity refers to the ability of synaptic connections to change their strength, which is thought to be the basic mechanism underlying learning and memory in biological neural networks Various forms of synaptic plasticity co-exist. They differ mainly on a time scale: some processes, e.g. pulse paired facilitation, decay on the order of about 10–100 ms; other processes, such as long-term potentiation (LTP) or long-term depression (LTD), persist for hours, days, or longer In this section we discuss various models of learning for spiking neural networks that explore spike-timing based synaptic plasticity.

4.2.1 STDP

Spike timing-dependent plasticity (STDP) is an unsupervised learning technique rooted in biology, following a Hebbian rule: when two interconnected neurons fire simultaneously, the synaptic strength between them should increase. However, in STDP, if a presynaptic spike occurs before a postsynaptic spike, the synapse's weight can either strengthen or weaken.

Conversely, if a postsynaptic spike precedes a presynaptic one, the synapse's weight may decrease or increase. When the weight increases in the first scenario and decreases in the second, it's termed Hebbian STDP. Conversely, if the opposite occurs, it's labeled as anti-Hebbian STDP.[15]

4.2.2 Backpropagation

Backpropagation techniques, unlike STDP, are not concerned with biological plausibility. Instead, they concentrate on understanding intricate spatiotemporal relationships between spikes. This has the significant disadvantage of requiring lengthy training cycles because networks must perform forward passes regularly, which takes a long time on typical computer hardware even with parallelization. This is due to the necessity of requiring the network over numerous time steps to create one spike response for each input[15]

4.2.3 ANN–SNN Conversion

The process of transitioning from deep learning models to Spiking Neural Networks (SNNs) often results in superior performance on prominent supervised-learning benchmark datasets like MNIST, CIFAR10, and ImageNet. This conversion method involves training an Artificial Neural Network (ANN) using backpropagation, then adjusting the weights and parameters of spiking neurons to convert it into an SNN. The objective is to maintain the same input-output mapping as the original ANN. These conversion approaches are typically classified into two categories: ordinary conversion and constrain-then-train conversion. The primary distinction lies in how the ANN is treated during training. In ordinary conversion techniques, the ANN is trained once and can subsequently be converted into SNNs with various parameters (such as neuron types, time constants, reset voltages, etc.). Conversely, in constrain-then-train conversion methods, the ANN is subject to constraints during the training process itself.[15]

Spiking Neural Networks (SNNs) offer several advantages over regular Artificial Neural Networks (ANNs), which are typically composed of artificial neurons that use continuous-valued activations and backpropagation for training.

5. APPLICATIONS OF SNN

[16] explores the scalability of neuromorphic computing for computer vision tasks, aiming to replicate non-neuromorphic performance while reducing power consumption.

[17] discusses SNNs as a bridge between artificial neural networks (ANNs) and biological neural networks (BNNs), highlighting their potential for high-performance computing with low power consumption. It presents a comparative analysis of different biological spiking neuron models for classifying handwritten digits from the MNIST dataset, with the Leaky Integrate-and-Fire neuron model achieving the best results. Four neuron models are compared in terms of performance, with a focus on the Leaky Integrate-and-Fire model due to its simplicity and computational efficiency.

[18] discusses a neuromorphic accelerator using a spin-orbit-torque magnetic random-access memory (SOT-MRAM) crossbar array interfaced with spiking neurons and peripheral circuits. It compares SOT-MRAM with other

nonvolatile memory devices like phase-change memory (PCM), resistive random-access memory (RRAM), and spin-transfer torque MRAM (STT-MRAM), highlighting SOT-MRAM's advantages in switching speed, energy consumption, and throughput. The document explores the benefits of the proposed design for large-scale neuromorphic accelerators using a device-circuit algorithm framework for standard MNIST image classification.

5.1 Advantages Of SNN

- Temporal processing: SNNs are well-suited for tasks that require precise timing and temporal processing. They can represent and process information with precise spike timings, which is important in tasks like event-based vision, speech recognition, and robotics.
- Energy efficiency: SNNs are often more energy-efficient compared to ANNs, particularly in applications where sparsity and event-driven processing are beneficial. This is because SNNs tend to produce spikes only when necessary, minimizing overall computational load and energy consumption.
- Sparse activations: SNNs are inherently sparse, meaning only a small percentage of neurons are active at any given time. This sparsity can help reduce computational and memory requirements, making them suitable for efficient hardware implementations.
- Neuromorphic hardware: SNNs are well-matched with neuromorphic hardware, which is designed to emulate the brain's structure and function. These specialized hardware platforms can exploit the event-driven nature of SNNs for real-time and low-power applications.
- Robustness to noise and variations: SNNs are often more robust to noise and variations in input data. They can effectively handle noisy or incomplete sensory inputs, making them suitable for applications in noisy environments or when dealing with sensor data that may have uncertainties.
- Biological plausibility: SNNs are biologically inspired and aim to mimic the behavior of real neurons more closely than traditional ANNs. This can be advantageous for research in computational neuroscience and neurobiology, as well as for creating brain-inspired AI models.
- Spike-timing-dependent plasticity (STDP): SNNs can take advantage of STDP, a learning rule that strengthens or weakens synaptic connections based on the relative timing of pre-and post-synaptic spikes. This mechanism is thought to be important in biological learning and memory, and it can be implemented in SNNs to enable unsupervised learning.
- Event-driven computation: SNNs perform computations in an event-driven manner, reacting to changes in input without the need for constant processing. This property is beneficial in applications where responsiveness and low latency are essential.

6. CONCLUSION

In this article SNNs are compared to traditional Artificial Neural Networks (ANNs), highlighting that SNNs could be more energy-efficient on neuromorphic hardware, but such hardware is not widely available. Spiking neural networks have high potential in the context of emulating the human brain. With the most realistic model of biological neurons nowadays, even neuroscientists and biologists have big expectations due to the investigation of brain diseases. These neuron models initialize new opportunities to develop better learning behavior resulting in more intelligent systems. With the continuously increasing performance of computer systems spiking neural networks can be simulated on a grand scale, building large networks with a more complex structure achieving higher artificial intelligence. Even though spiking neural networks are a stupendous realistic model of natural neural networks, they are transient due to major advances in neuroscience. In the future, there will be a newer generation of neural network models, with more realistic behavior representing the human brain in a more detailed way.

REFERENCES

1. L. Zhang, J. Woo, K. Joao Ventura, J. Lottier Molin, C. Dias, and D. Ivanov, “Neuromorphic artificial intelligence systems.”
2. N. Rathi et al., “Exploring Neuromorphic Computing Based on Spiking Neural Networks: Algorithms to Hardware,” ACM Comput Surv, vol. 55, no. 12, Mar. 2023, doi: 10.1145/3571155.
3. S. B. Maind and P. Wankar, “International Journal on Recent and Innovation Trends in Computing and Communication Research Paper on Basic of Artificial Neural Network”, [Online].
4. S. B. Maind and P. Wankar, “International Journal on Recent and Innovation Trends in Computing and Communication Research Paper on Basic of Artificial Neural Network”, [Online]. Available: http://www.ijritcc.org
5. N. Assani, P. Matic, N. Kastelan, and I. R. Cavka, “A Review of Artificial Neural Networks Applications in Maritime Industry,” IEEE Access, vol. 11, pp. 139823–139848, 2023, doi: 10.1109/ACCESS.2023.3341690.
6. “2604”. Long Short Term Memory, Neural computation 9(8):1735-1780,1997

7. I. A. Basheer and M. Hajmeer, “Artificial neural networks: fundamentals, computing, design, and application,” 2000. [Online]. Available: www.elsevier.com/locate/jmicmeth
8. S. S. Haykin and S. S. Haykin, Neural networks and learning machines. Prentice Hall/Pearson, 2009.
9. D. Bhattacharyya, A. April, J. Kumar Basu, and T.-H. Kim, “Use of Artificial Neural Network in Pattern Recognition 2 2 2 Use of Artificial Neural Network in Pattern Recognition,” 2020. [Online]. Available: https://www.researchgate.net/publication/228566377
10. “Application of artificial neural networks to clinical medicine 1135.”
11. S. A. Kalogirou, “Applications of arti®cial neural-networks for energy systems.” [Online]. Available: www.elsevier.com/locate/apenergy
12. E. Rusakov, “Spiking Neural Networks.”
13. K. Yamazaki, V. K. Vo-Ho, D. Bulsara, and N. Le, “Spiking Neural Networks and Their Applications: A Review,” Brain Sciences, vol. 12, no. 7. MDPI, Jul. 01, 2022. doi: 10.3390/brainsci12070863.
14. J. Wang, “A Review of Spiking Neural Networks,” SHS Web of Conferences, vol. 144, p. 03004 2022, doi: 10.7544/issn1000-1239.2019.20190240
15. P. Pietrzak, S. Szczęsny, D. Huderek, and Ł. Przyborowski, “Overview of Spiking Neural Network Learning Approaches and Their Computational Complexities,” Sensors, vol. 23, no. 6. MDPI, Mar. 01, 2023. doi: 10.3390/s23063037
16. K. Patel, E. Hunsberger, S. Batir, and C. Eliasmith, “A Spiking Neural Network for Image Segmentation.”
17. S. Yadav, S. Chaudhary, and R. Kumar, “Comparative Analysis of Biological Spiking Neuron Models for Classification Task,” in 2023 14th International Conference on Computing Communication and Networking Technologies, ICCCNT 2023, Institute of Electrical and Electronics Engineers Inc., 2023
18. Verma, A. Nisar, S. Dhull, and B. K. Kaushik, “Neuromorphic Accelerator for Spiking Neural Network Using SOT-MRAM Crossbar Array,” IEEE Trans Electron Devices, vol. 70, no. 11, pp. 6012–6020, Nov.2023, doi:10.1109/TED.2023.3317357

CITATION
Deeksha, & Padmanayana. (2024). From Artificial Neural Networks to Spiking Neural Networks: A Comprehensive Review. In Global Journal of Research in Engineering & Computer Sciences (Vol. 4, Number 3, pp. 52–59). https://doi.org/10.5281/zenodo.11618186