Evaluating Spiking Neural Network Models: A Comparative Performance Analysis

Sanaullah*
Bielefeld University of Applied Science, Germany

Shamini Koravuna
University of Bielefeld, Germany

Ulrich Rückert
University of Bielefeld, Germany

Thorsten Jungeblut
Bielefeld University of Applied Science, Germany

1 INTRODUCTION:

Spiking Neural Networks (SNNs) are inspired by the functioning of biological neurons in the brain and use discrete time steps to simulate the behavior of neurons through the generation of spikes [1]. This allows SNNs to integrate input from interconnected neurons, and when the input reaches a certain threshold, it generates a spike that is transmitted to other neurons. To build an SNN, a mathematical model of the spiking behavior of neurons is required, and several models are available, including the leaky integrate-and-fire (LIF) model [2], the Adaptive Exponential (AdEx) [3], and the Nonlinear Integrate-and-Fire (NLIF) models [4]. These models and others like them are used to simulate the behavior of neurons in SNNs and improve the accuracy of the network's outputs. Overall, SNNs and their models are powerful tools for modeling neural processing and have potential applications in a wide range of fields. Although SNNs and their simulators have the potential to offer many benefits [5, 6], but their implementation poses certain challenges. One of these challenges is determining the most suitable SNN model, especially when it comes to classification, which requires high accuracy and low-performance loss.

In order to address this issue, a study was conducted to compare the performance, behavior, and spikes generation methodology of different SNN models using the same number of inputs and neurons. The challenge of determining the most suitable model was addressed by comparing the performance of different models, and the results were analyzed to determine the most effective one. Overall, this study sheds light on the challenges and potential benefits of SNNs and their models. It highlights the importance of comparing different models to determine the most suitable and provides valuable insights for researchers and practitioners working in this area.

2 RESULTS AND DISCUSSION:

SNN models are typically built using mathematical equations that describe the behavior of spiking neurons. These equations take into account various factors such as the input current, membrane potential, and membrane time constant, to simulate the behavior of biological neurons. Each SNN model has its own set of equations that determine its behavior. In this study, we investigate three different SNN models, namely LIF, NLIF, and AdEX. Each of these models is implemented using the update method, which takes an input current and a time step and returns whether a spike has occurred or not.

Types of SNN Models:

For each model, we initialize the instance variables and generate random weights for each neuron. One of the simplest SNN models is the LIF model, which is described by Eq. 1:

$$\tau_m \frac{dV}{dt} = -V(t) + I(t) \tag{1}$$

where τ_m is the time constant, $V(t)$ is the membrane potential of the neuron at time t, and $I(t)$ is the input current at time t. The membrane potential is updated based on the Eq. 2:

$$V(t) \leftarrow V(t) + \frac{-V(t) + I(t)}{\tau_m} \cdot dt \tag{2}$$

If membrane potential reaches the threshold potential V_{th}, the neuron fires a spike and the membrane potential is reset to the resting potential V_{reset}, which is described in Eq. 3:

$$\text{if } V(t) \geq V_{\text{th}}, \text{ then } V(t) \leftarrow V_{\text{reset}} \tag{3}$$

A variation of the LIF model is the Non-Linear Integrate-and-Fire (NLIF) model, which takes into account the non-linear relationship between the membrane potential and the input current. The NLIF model is described by Eq. 4:

$$\frac{dV}{dt} = \frac{-V + I}{\tau} \tag{4}$$

where V is the membrane potential, I is the input current, τ is the membrane time constant, and $\frac{dV}{dt}$ represents the change in the membrane potential over time. The membrane potential is updated based on the Eq. 5:

$$V(t) \leftarrow \begin{cases} V_{\text{reset}} + \alpha \cdot (V(t) - V_{\text{th}}) \ \& \ \text{if } V(t) \geq V_{\text{th}} \\ V(t) \cdot \beta \ \& \ \text{otherwise} \end{cases} \tag{5}$$

where V_{reset} is the resting potential, α and β are scaling factors, and V_{th} is the threshold potential. The membrane potential is multiplied by β if it is below the threshold, which models the leakage of current from the neuron over time. If the membrane potential reaches the threshold potential, the neuron fires a spike and the membrane potential is reset

to the resting potential plus a scaled depolarization of the membrane potential (as described by Eq. 6):

$$\text{if } V(t) \geq V_{\text{th}}, \text{ then } V(t) \leftarrow V_{\text{reset}} + \alpha \cdot (V(t) - V_{\text{th}}) \tag{6}$$

Another SNN model is the Adaptive Exponential (AdEX) model, which captures the dynamic behavior of spiking neurons more accurately than the LIF or NLIF models. The AdEX model is described by Eq. 7:

$$\frac{dV}{dt} = \frac{-V + \tau_m I - V_{rheo} + \Delta_T \exp\left(\frac{V - V_{spike}}{\Delta_T}\right)}{\tau_m} \tag{7}$$

where V is the membrane potential of the neuron, I is the input current, τ_m is the membrane time constant, V_{rheo} is the rheobase potential, Δ_T is the slope factor, and V_{spike} is the threshold potential at which the neuron fires an action potential. If V exceeds V_{spike}, a spike is generated and V is reset to V_{reset}.

The dataset for this study was generated using the following approach; Let $n_{samples} = 1000$, $x_1 \sim \mathcal{N}(0,1)$, $x_2 \sim \mathcal{N}(3,1)$, $X = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$, $y = \begin{bmatrix} 0_{n_{samples}} & 1_{n_{samples}} \end{bmatrix}$, where $0_{n_{samples}}$ and $1_{n_{samples}}$ are the vectors of length $n_{samples}$ filled with zeros and ones, respectively. To shuffle the dataset, let $indices = \begin{bmatrix} 0 & 1 & \cdots & 2n_{samples} - 1 \end{bmatrix}$, and apply a random permutation to $indices$. Then, let X and y be the arrays obtained by indexing X and y with the shuffled $indices$.

where $n = 1000$ is the number of samples, $x_{1,i} \sim \mathcal{N}(0,1)$ and $x_{2,i} \sim \mathcal{N}(3,1)$ are the features of the i-th sample for $i \in 1, \ldots, n$, and y is a vector of length $2n$ where the first n elements are 0 and the last n elements are 1. The dataset is then shuffled using the indices $indices = [0, 1, \ldots, 2n-1]$, and X and y are updated accordingly.

Comparison

In this study, we explored the performance of different SNN models by simulating their behavior using the equations that define them. Each of these models is implemented using the update method, which takes an input current and a time step and returns whether a spike has occurred or not. We then evaluated the performance of each model by calculating its classification accuracy and performance loss. The proposed approach also randomly initializes weights for each model and visualizes the spiking activity of the neurons over time. Figure 1 shows the classification accuracy and performance loss of different SNN models. LIF model had an accuracy of 71.65%, while NLIF had an accuracy of 67.05%. AdEX model had the highest accuracy of 90.65%. The performance loss of LIF model was -6.86% relative to NLIF, -26.52% relative to AdEX with LIF, and -35.20% relative to AdEX with NLIF. Furthermore, The performance of each two compared models is measured in terms of their accuracy. We used 1000 samples as inputs and 1000 neurons for each model execution. However, the values of other parameters varied for each model.

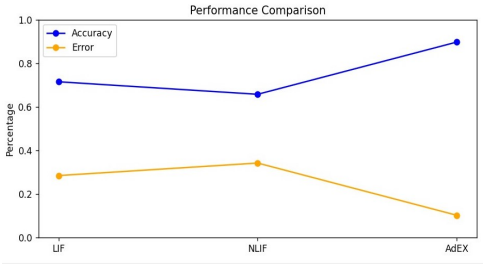

Figure 1: Performance of different SNN models on a classification based on 1000 inputs using 1000 neurons

These results provide insights into the suitability of different SNN models for classification tasks and can aid in selecting the appropriate model for a given task. The detailed results can be viewed in [7].

3 CONCLUSION

In conclusion, a study was conducted to compare different SNN models using the same inputs and neurons, providing insights for researchers and practitioners in this area. The study evaluated model performance by measuring classification accuracy and performance loss and visualizing spiking activity. In the future, further research could explore additional SNN models and compare their performance using various benchmarks to determine the most suitable model for specific applications, potentially saving time and resources. Overall, this study highlights the potential benefits of SNNs and their models and offers valuable insights for future research and development.

ACKNOWLEDGEMENTS

This research was supported by the research training group "Dataninja" (Trustworthy AI for Seamless Problem Solving: Next Generation Intelligence Joins Robust Data Analysis) funded by the German federal state of North Rhine-Westphalia and the project SAIL. SAIL is funded by the Ministry of Culture and Science of the State of North Rhine-Westphalia under grant no NW21-059B.

REFERENCES

[1] Samanwoy Ghosh-Dastidar and Hojjat Adeli. Spiking neural networks. 2009.
[2] Doron Tal and Eric L Schwartz. Computing with the leaky integrate-and-fire neuron: logarithmic computation and multiplication. 1997.
[3] Wulfram Gerstner and Romain Brette. Adaptive exponential integrate-and-fire model. *Scholarpedia*, 2009.
[4] Renaud Jolivet, Timothy J Lewis, and Wulfram Gerstner. Generalized integrate-and-fire models of neuronal activity approximate spike trains of a detailed model to a high degree of accuracy. 2004.
[5] Sanaullah, Shamini Koravuna, Ulrich Rückert, and Thorsten Jungeblut. SNNs model analyzing and visualizing experimentation using ravsim. Springer International Publishing, 2022.
[6] Marcel Stimberg, Romain Brette, and Dan FM Goodman. Brian 2, an intuitive and efficient neural simulator. 2019.
[7] Github: Availability of detailed results. https://github.com/Rao-Sanaullah/Evaluating-Spiking-Neural-Network-Models-A-Comparative-Performance-Analysis. Accessed: April 2023.