Improving Regression in Spiking Neural Networks for Oceanographic Data Analysis

Alissa J Kane
Computer and Information Science Department
University of Massachusetts Dartmouth
North Dartmouth, MA 02747
akane6@umassd.edu

Yuchou Chang
Computer and Information Science Department
University of Massachusetts Dartmouth
North Dartmouth, MA 02747
ychang1@umassd.edu

***Abstract*—Spiking neural networks (SNNs), as the third-generation neural networks, can work under an energy efficient mode. SNNs are different from the second-generation neural networks which consume a lot of energy and power. SNNs are suitable for oceanographic data analysis on the edge devices underwater since the devices have constrained power supply and limited communication bandwidth in underwater environments. Although SNNs have been widely used in classification tasks, SNN-based regression tasks are studied less because SNNs are generally considered to process discrete and sequential spikes. The existing regression model based on the membrane potential of Leaky Integrate-and-Fire (LIF) neuron uses constant settings and this mechanism may not be adaptive and capable of analyzing oceanographic data which are complicated and dynamic. In this paper, we proposed three novel regression models of Adaptive Threshold Adjustment, Heterogeneous Neurons, and Nonlinear Integration to improve the existing LIF-based model. Experimental results on real oceanographic data indicate that the proposed regression models outperform the existing model through qualitative and quantitative analysis. Those SNN regression models could be implemented on edge devices within underwater environments in the future.**

Keywords— Spiking Neural Networks (SNNs), Leaky Integrate-and-Fire (LIF), Regression, Oceanographic Data Analysis, Energy Efficiency, Edge Computing.

I. Introduction

Spiking Neural Networks (SNNs) are recognized as the third-generation neural networks [6]. SNNs provide a significant energy efficiency advantage over the second-generation neural networks [7]. This energy-saving property makes SNNs suitable for edge devices. SNNs have low power consumption, so they can be used for applications in power-constrained environments. These networks simulate the brain's neural information processing and use asynchronous spikes. The SNN approach not only reduces energy usage but also enhances computational efficiency because SNNs generally work under the event-driven mechanism. Therefore, SNNs present a promising solution for deploying intelligent functionalities on edge devices, where power efficiency is a major concern.

Oceanographic data analysis often occurs underwater, where communication bandwidth is limited. Data cannot be transferred to cloud computing for further processing and analysis. Underwater edge devices are typically battery-powered, which can be used to handle oceanographic data analysis tasks. Given their constrained energy resources, SNNs may provide a feasible solution. Based on the low energy consumption of edge devices, SNNs can efficiently perform data regression analysis in underwater environments. By employing SNNs, edge devices can process and analyze oceanographic data with less power usage than that on regular artificial neural network models. This approach ensures prolonged operation and reliable data analysis, even with limited bandwidth and energy in underwater environments. Therefore, SNNs present a feasible method for oceanographic data regression analysis and optimize performance in power-limited underwater environments.

SNNs process discrete spikes, so this is a challenge for regression tasks. To address this, the Leaky Integrate-and-Fire (LIF) neuron's membrane potential is often used. By training the membrane potential, regression task aims to have it follow a given trajectory over time. This approach [2] involves training a LIF neuron to ensure its membrane potential accurately tracks the input training samples. LIF-based regression achieves precise alignment between the neuron's potential and the desired regression trajectory. This regression method uses the unique properties LIF neurons of SNNs and enables regression tasks in spiking neural networks.

Existing LIF-based regression tasks have limitations, resulting in suboptimal regression accuracy. Inspired by concepts like Adaptive Threshold Adjustment, Heterogeneous Neurons, and Nonlinear Integration, which closely resemble mechanisms in the human brain, we aim to improve the existing regression method in SNNs. Motivated by these brain-inspired ideas, this paper presents enhancements to the traditional LIF-based regression approach. We apply these improved methods to oceanographic data regression analysis. Our goal is to achieve higher accuracy and efficiency in regression tasks. So, we can achieve the advanced SNN regression techniques for better performance on energy-constrained devices within underwater environments. In this paper, the first section is the introduction. The second part gives the related work. The proposed method is introduced in the third section. The fourth section presents the experimental results. Conclusion is given in the last part.

II. Related Works

SNNs are often used for classification tasks [8] due to their discrete spike-based information flow, which naturally aligns with the requirements of classification. The inherent temporal coding capabilities of SNNs make them particularly suitable for tasks involving temporal patterns, such as speech recognition [9] and event detection [10]. However, regression tasks require precise continuous value predictions. They are challenging to achieve with the spike-based nature of SNNs. This precision requirement has led to limited research and application of SNNs in regression.

Initial attempts [11] to apply SNNs for regression tasks have shown promise in environments where power and computational efficiency are critical. The work [2] by Henkes et al. explores nonlinear regression with SNNs and demonstrates SNN potential for handling complex regression tasks. A regression framework was proposed based on the Leaky Integrate and Fire (LIF). SNN-based regression is very energy efficient while keeping the precision and generalizability. A study [12] has focused on the application of LIF neurons for regression tasks. The membrane potential dynamics of LIF neurons are utilized to encode and process continuous-valued data. This approach has shown promise in various scenarios, including underwater environmental data analysis and autonomous systems. The current LIF-based regression [2] still has some drawbacks to achieve optimal regression performance. We proposed three new regression models to improve the existing method.

III. Proposed Methods

A. Data Cleaning

Prior to implementing the models, the data was cleaned and transformed. The data, retrieved from The California Cooperative Oceanic Fisheries Investigations [13], contained many columns describing various aspects of the ocean, including depth, temperature, salinity, and oxygen levels. For each implementation, two of these attributes (columns) were extracted for analysis. After narrowing down the target variables, any null values were dropped. For this paper, 'depth-oxygen' and 'depth-salinity' will be the focused columns, where depth will be used to predict oxygen and salinity using SNN regression. The data was then normalized before being exported for the regression task.

To begin work on the models, the data was imported using the Pandas 'read_csv' function. A sample of 500 points was extracted, and the data was then split into training, with 400 points, and testing, with 100 points. Both sets of data were then converted using the 'SNNDataset' class, each with 100 time steps. This class designated column 0 as the 'feature' and column 1 as the 'label'. We proposed three novel models with the goal of improving the original SNN with LIF regression [2]: Adaptive Threshold Adjustment, Heterogeneous Neurons, and Nonlinear Integration. Each of the models builds upon the original SNN with LIF model [1-2], but with various new improvements.

B. Adaptive Threshold Adjustment

The Adaptive Threshold Adjustment model implements neurons that can adjust their firing threshold for a given input spike, rather than remaining fixed. With this mechanism, the network can optimize its response based on the intensity and structure of the input to improve its training process. This updated model should yield better results as the neurons can adapt their threshold and recognize patterns amongst complex data. Unlike the traditional SNN regression with constant LIF potential, the new adaptive threshold adjustment makes this threshold a variable. Fig. 1 demonstrates the difference between constant and adaptive threshold mechanisms. The membrane potential exhibits periodic peaks at times T1 to T6. The constant threshold represented by the black dashed line remains unchanged over time. In contrast, the adaptive threshold shown by the green dashed line adjusts based on the membrane potential's behavior. It lowers after a spike and gradually increases until the next spike. This adaptive mechanism allows the neuron to dynamically alter its firing threshold. This potentially improves the network's ability to handle varying input patterns and enhances its overall computational efficiency.

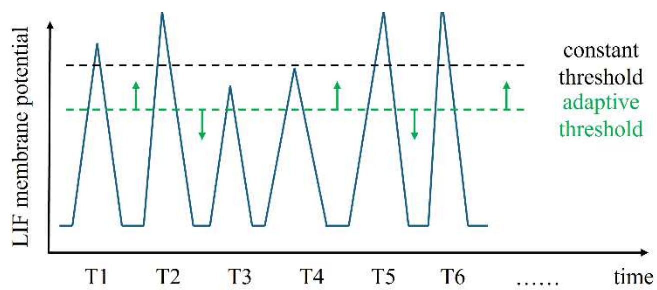

Fig. 1. Membrane potential dynamics of an LIF neuron over time. It illustrates constant (black dashed line) and adaptive (green dashed line) thresholds. The adaptive threshold adjusts in response to spikes.

C. Heterogeneous Neurons

In the Heterogeneous Neurons model, numerous neurons are used to simulate the billions of neurons found in the human brain. Unlike the original LIF regression model, this model allows for variability across the neurons. Variables including decay rate and threshold are randomly initialized for each neuron before the model is trained. With each neuron having its own characteristics, the model is better built for a broad range of data.

Incorporating various decay rates across neurons brings several benefits to the Heterogeneous Neurons model. Decay rates determine how quickly a neuron's membrane potential returns to its resting state after receiving an input spike. By introducing variability in these decay rates, the model can simulate a more diverse set of neuronal behaviors. It is close to the biological variability seen in the human brain. This diversity enables the network to capture a wider range of temporal dynamics and patterns within the input data. It enhances the network's ability to generalize across different tasks and datasets. Different decay rates also allow the network to balance sensitivity and stability. Some neurons may respond quickly to new information, while others retain information for longer periods. This approach simulates the natural distribution of fast- and slow-responding neurons in biological systems. So, the mechanism can improve the model's robustness and accuracy. Using various decay rates enhances the network's adaptability and resilience and makes it better suited for complex and dynamic oceanographic data.

D. Nonlinear Integration

The Nonlinear Integration mechanism implements a nonlinear activation function called Rectified Linear Unit. Unlike the original SNN with LIF model, which sums signals linearly, we proposed the model uses the ReLU function, seen in (1) to process input data [3].

$$ReLU(x) = \max(0, x), \qquad (1)$$

where x represents the input data. The use of a nonlinear activation function allows a more flexible response amongst the neurons. This leads to a more refined output and better accuracy. This mechanism can allow the model to recognize complex relationships in the data that the original SNN with LIF may miss. Compared to linear models, ReLU can more effectively capture complicated patterns and features within the data and improve the regression model's accuracy and output quality. When dealing with complex datasets, nonlinear activation functions enable the model to better adapt to data diversity and nonlinear characteristics and enhance overall performance.

E. Performance Evaluation

Each of the three proposed models was run through the same training and evaluation code. The models were trained at a learning rate of 0.001 for 200 iterations using Mean Square Error loss. They were then evaluated with L1 loss, L2 loss (Mean Square Error), and Relative Error. L1 loss is calculated using the Mean Absolute Error, seen in Equation (2). L2 loss as shown in Equation (3) is calculated similarly, using Mean Squared Error [4].

$$L1 = \frac{1}{n}\sum_{i=1}^{n}\left|y_{actual} - y_{predicted}\right|, \qquad (2)$$

$$L2 = \frac{1}{n}\sum_{i=1}^{n}\left(y_{actual} - y_{predicted}\right)^2, \qquad (3)$$

where y_{actual} represents the actual values in the dataset, $y_{predicted}$ denotes the predicted values from the proposed methods or the traditional LIF-based regression model [2], and n is the total number of testing data.

Relative error was calculated manually utilizing the 'torch.linalg.norm' function. This calculates the vector norm of the difference between the target value and the model output (predicted) value [5]. That norm is then divided by the vector norm of the target value to achieve the relative error. The formula for relative error can be seen in Equation (4),

$$Relative\ Error = \frac{\left|y_{actual} - y_{predicted}\right|}{y_{actual}}, \qquad (4)$$

where, for each iteration, the corresponding error value was appended to a list. The mean value of each list was returned as the overall error value for that model. These three-performance metrics will be used in analyzing the accuracy of each model.

To highlight the difference between the proposed model and the original, the 'Improvement' formula was used. This calculated the percentage to which each of the models improved or worsened compared to the original SNN with LIF model. The formula used can be seen in Equation (5).

$$Improv. = \left(\frac{original - new}{original}\right) \cdot 100, \qquad (5)$$

where $original$ represents a performance metric value by the original method shown in reference [2] and new denotes a performance metric value by each of three proposed methods. The improvement percentage can show each new method's potential for enhancing the original method.

III. Experimental Results

The first dataset to be tested was 'depth-oxygen'. Based on the depth of the water, oxygen levels can be predicted. The visual output of the various models can be seen in Fig. 2. Target curve with orange color indicates the ground-truth data, and the Output curve with blue color represents the predicted values by each of regression models. The visual representation of the models' output indicates better accuracy in the proposed regression models. The SNN with LIF is the traditional method presented in reference [2]. The other three new regression models of Adaptive Threshold Adjustment, Heterogeneous Neurons, and Nonlinear Integration Mechanism have better curve fitting performance than the SNN with LIF.

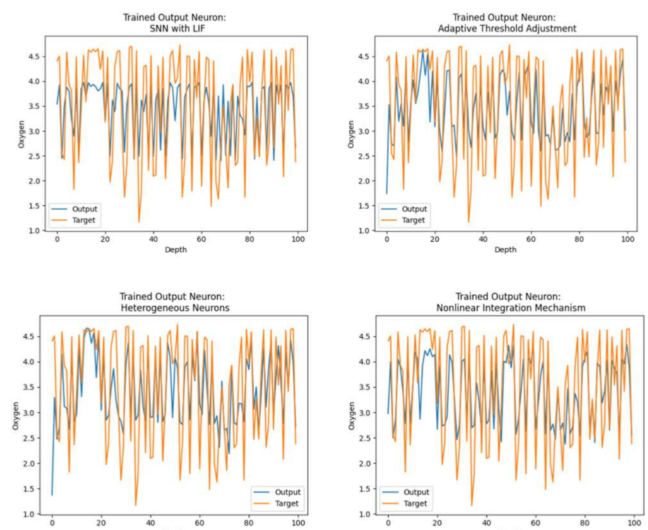

Fig. 2. Four Model Outputs for Depth & Oxygen Data

Numerical performance metrics were also calculated to provide a quantitative observation at these models' accuracy. The results of the previously described performance metrics of each model can be viewed in Table 1.

TABLE 1. MODEL RESULTS FOR DEPTH-OXYGEN

Results for Depth & Oxygen

Model	L1-loss	L2-loss	Relative Error
SNN with LIF	1.41779	0.572722	0.226047
Adaptive Threshold Adjustment	1.14941	0.528879	0.211923
Heterogeneous Neurons	0.857147	0.547101	0.205856
Nonlinear Integration Mechanism	1.16397	0.42953	0.192769

Equation (5) was also used to calculate how much each model improved when compared to the original SNN with LIF mode. These results can be seen in Table 2. Looking at the results of Table 2, specifically at L1 loss, the Adaptive Threshold Adjustment model improved in accuracy by 18.9295% when compared to SNN with LIF. The Heterogeneous Neurons model saw the greatest improvement for this metric at 39.5436%. Nonlinear Integration also improved in accuracy by 17.9024%. This means that the absolute value of the difference between each target and predicted point decreased by at least 17% for each model. This decrease demonstrates an improvement as the model is predicting values closer to the target data. Similar patterns can be viewed in the subsequent metrics. Adaptive Threshold Adjustment improved in L2 loss by 7.65527%. The Heterogeneous Neurons model improved by 4.47365% while the Nonlinear Integration model improved by 25.002%. This means that the squared difference between each target and predicted point decreased by as much as 25%. Like L1 loss, this signifies an improvement in the proposed models as their output more closely predicts the target. Finally, Adaptive Threshold Adjustment also improved in Relative Error by 6.24842%. Heterogeneous Neurons improved in this metric as well by 8.93246%. The Nonlinear Integration Mechanism improved by 14.7219% in Relative Error. This means that the absolute value of the difference between each target and predicted point, divided by the target, decreased by at least 6%. A decrease in relative error, like the loss values, means that the model is more closely predicting the target values. These values illustrate an improvement in all tested performance metrics for the proposed models when compared to SNN with LIF.

TABLE 2. PERCENT IMPROVEMENT FROM SNN WITH LIF FOR DEPTH-OXYGEN

Percent of Improvement from SNN with LIF

Model	L1 Loss	L2 Loss	Relative Error
Adaptive Threshold Adjustment	18.9295	7.65527	6.24842
Heterogeneous Neurons	39.5436	4.47365	8.93246
Nonlinear Integration Mechanism	17.9024	25.002	14.7219

The models were also tested on the ‘depth-salinity’ dataset to verify these performance patterns. The visual output of the three proposed models on this data, seen in Fig. 3, suggests that all of them performed better than that of the traditional SNN with LIF regression model [2].

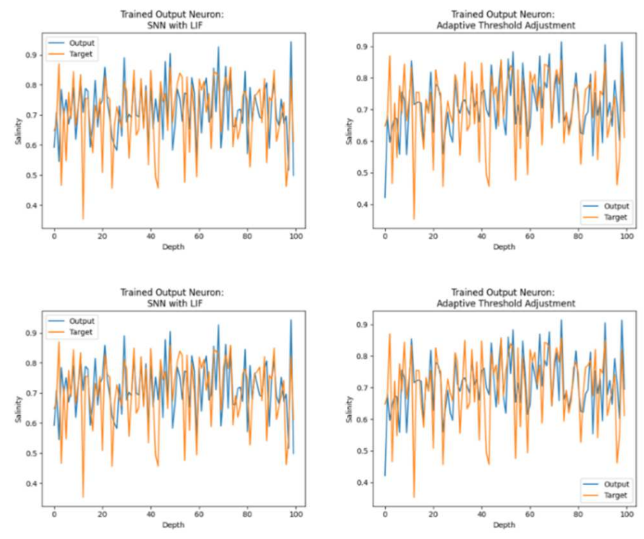

Fig. 3. Four Model Outputs for Depth & Salinity Data

A visual representation of the tested performance metrics, seen in Fig. 4, provides a better comparison between the accuracy of these models.

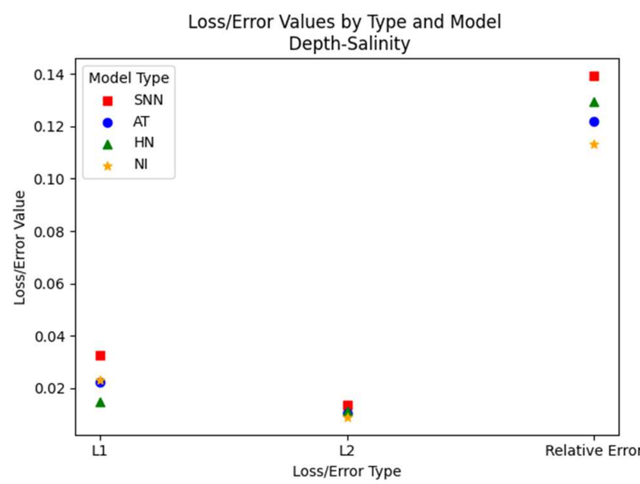

Fig. 4. Model Loss & Error Values for Depth & Salinity. Note. SNN refers to SNN with LIF, AT refers to Adaptive Threshold Adjustment, HN refers to Heterogeneous Neurons, and NI refers to Nonlinear Integration.

Fig. 4 shows that while the loss and error values appear similar across all the models, SNN with LIF [2] had the highest loss/error value of them all for each metric. To provide a better quantitative analysis of these results, Equation (5) was employed to determine to what extent the proposed models outperformed the original. The data from these calculations can be seen in Table 3.

TABLE 3. PERCENT IMPROVEMENT FROM SNN WITH LIF FOR DEPTH-SALINITY

Percent of Improvement from SNN with LIF

Model	L1 Loss	L2 Loss	Relative Error
Adaptive Threshold Adjustment	31.7136	23.6223	12.717
Heterogeneous Neurons	54.3297	18.6309	7.31843
Nonlinear Integration Mechanism	28.8932	35.4578	18.8351

When compared to SNN with LIF, Adaptive Threshold Adjustment improved in L1 loss by 31.7136%. The Heterogeneous Neurons model also saw a substantial improvement of 54.3297%. Nonlinear Integration Mechanism also improved by a value of 28.8932%. Regarding L2 loss, Adaptive Threshold Adjustment improved by 23.6223%. The Heterogeneous Neurons model also saw an improvement of 18.6309%. The Nonlinear Integration model also improved in this metric by 35.4578%. Finally, Adaptive Threshold Adjustment improved in Relative Error by 12.717%. Heterogeneous Neurons also improved in this metric at 7.31843%. The Nonlinear Integration model also improved in Relative Error by 18.8351%. These positive improvement rates highlight to what extent the loss and error values decreased compared to SNN with LIF, demonstrating that the proposed models better predicted the target data.

IV. CONCLUSIONS

The results obtained show that all three proposed models outperformed the SNN with LIF model for every performance metric for both the 'depth-oxygen' and 'depth-salinity' datasets. Furthermore, Equation (5) highlighted that some of the models achieved vast improvements when compared to the original model. For example, the Heterogeneous Neurons model achieved over a 50% improvement in L1 loss for the 'depth-salinity' dataset. With the smallest overall improvement value being approximately 4%, these results show that the proposed models were able to improve in accuracy for each metric tested. Overall, the results suggest that the proposed models of Adaptive Threshold Adjustment, Heterogeneous Neurons, and Nonlinear Integration Mechanism were able to improve the regression task for oceanographic data analysis compared to the conventional SNN with LIF model.

ACKNOWLEDGMENT

The project is supported by UMass Dartmouth's Marine Undersea Technology (MUST) research program with project S31320000059156.

REFERENCES

[1] J. K. Eshraghian, M.Ward, E, Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu. "Training spiking neural networks using lessons from deep learning". Proceedings of the IEEE, 111(9) September 2023.

[2] A. Henkes, J. K. Eshraghian, and H. Wessels. "Spiking neural networks for nonlinear regression", arXiv preprint arXiv:2210.03515, October 2022.

[3] PyTorch Contributors. "ReLU". PyTorch. https://pytorch.org/docs/stable/generated/torch.nn.ReLU.html (accessed Jul. 11, 2024)

[4] A. Shekhar. "What are L1 and L2 loss functions?". Outcome School. https://outcomeschool.com/blog/l1-and-l2-loss-functions (accessed Jul. 11, 2024)

[5] PyTorch Contributors. "torch.linalg.norm". PyTorch. https://pytorch.org/docs/stable/generated/torch.linalg.norm.html (accessed Jul. 11, 2024)

[6] W. Maass, "Networks of spiking neurons: The third generation of neural network models," Neural Networks, vol. 10, no. 9, pp. 1659-1671, 1997.

[7] S. Kundu, G. Datta, M. Pedram, and P. A. Beerel, "Spike-thrift: towards energy-efficient deep spiking neural networks by limiting spiking activity via attention-guided compression," IEEE Winter Conference on Applications of Computer Vision (WACV), 2021.

[8] M. Yao, G. Zhao, H. Zhang, Y. Hu, L. Deng, Y. Tian, B. Xu, and G. Li, "Attention spiking neural networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 8, pp. 9393 - 9410, 2023.

[9] J. Wu, E. Yilmaz, M. Zhang, H. Li, K. C. Tan, "Deep spiking neural networks for large vocabulary automatic speech recognition," Frontiers in Neuroscience, vol. 14, 2020.

[10] A. Roy, M. Nagaraj, C. M. Liyanagedera, and K. Roy, "Real-time event-based speed detection using spiking neural networks," IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2023.

[11] E. Guerrero, F. M. Quintana, and M. P. Guerrero-Lebrero, "Event-based regression with spiking networks," Lecture Notes in Computer Science (LNCS), vol. 14135, pp. 617–628, 2023.

[12] M. D. Florio, A. Kahana, and G. E. Karniadakis, "Analysis of biologically plausible neuron models for regression with spiking neural networks," , arxiv, 2023. DOI:10.48550/arXiv.2401.00369.

[13] California Cooperative Oceanic Fisheries Investigations (CalCOFI). CalCOFI dataset: https://www.kaggle.com/sohier/calcofi/download.