WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 5, Number 2, Year 2025
DOI: https://doi.org/10.62051/ijcsit.v5n2.05
Journal homepage: https://wepub.org/index.php/IJCSIT/index

An Improved Method for Tsetlin Machine Based on Multi-level Feedback and Dynamic Parameter

Yi Shen *, Zhenkun Hu

School of Information and Engineering, Chongqing Jiaotong University, Chongqing ,400074, China
*Corresponding Author: Yi Shen

ABSTRACT

Addressing the issues of slow convergence and insufficient stability in the training process of Tsetlin Machine (TM), this paper proposes an innovative approach that integrates a multi-level feedback mechanism with dynamic parameter adjustment. By introducing a four-level feedback mechanism, this method accelerates the reinforcement of high-contribution clauses and the correction of significant errors through fine-grained feedback intensity allocation. Additionally, a dynamic parameter adjustment strategy for s is proposed, which optimizes state transition probabilities across different training stages using periodic steps d, thereby enhancing the model's exploration-exploitation balance capabilities. Experimental results demonstrate that this method improves training speed by approximately two times on the MNIST dataset, with an accuracy enhancement of 0.5%. These findings confirm that the proposed method significantly enhances model convergence efficiency and generalization performance through graded feedback intensity and adaptive parameter mechanisms, offering new insights for the deployment of interpretable AI in resource-constrained scenarios.

KEYWORDS

Tsetlin Machine; Learning Automata; Machine Learning; Artificial Intelligence.

1. INTRODUCTION

In the field of machine learning, Tsetlin Machine (TM), as a logic-based finite automaton model, has garnered significant attention due to its high interpretability and concise structure.TM performs classification or regression tasks through the combination of logical clauses, enabling an intuitive explanation of the model's decision-making process [1]. However, despite demonstrating outstanding performance in various applications, the training process of TM still faces pressing issues, primarily manifested in slow convergence and difficulty in achieving stable convergence [2]. These problems stem largely from limitations in the design of the clause feedback adjustment process and parameter control within TM. Therefore, this paper proposes a method that combines a multi-level feedback mechanism with dynamic parameter adjustment for s, aiming to enhance the training efficiency and classification accuracy of TM.

The training of traditional TM relies on a feedback mechanism to adjust the weights and configurations of clauses. Specifically, TM reward correctly classified clauses through feedback I and penalize incorrectly classified clauses through feedback II [3]. While this binary feedback approach is effective to some degree, it has limitations in effectively distinguishing different levels of correctness and error, as well as the individual clauses' contributions to the overall error, within a short period. This results in difficulties for the model in quickly finding the optimal combination of clauses when faced with complex or high-dimensional data, thereby affecting both the overall convergence speed and the final classification performance. To address this issue, this paper introduces a multi-level feedback mechanism, refining the original two feedback processes into multiple grades, including strong positive feedback, weak positive feedback, weak negative feedback, and strong negative feedback. By allocating different intensities of feedback to clauses based on voting results, this fine-grained feedback mechanism can more precisely adjust clause weights, enabling the model to more effectively correct severe errors and solidify high-confidence correct classifications, thereby significantly improving classification accuracy and accelerating convergence speed.

During the training process of TM, the parameter s controls the probability of clause state adjustments. In traditional methods, the parameter s is typically determined before the start of training and remains constant throughout the entire training process [4]. This fixed parameter setting lacks flexibility and cannot be dynamically adjusted according to the training progress, limiting the model's performance and optimization capabilities at different training stages. To overcome this limitation, this paper proposes a strategy for dynamically adjusting the parameter s. By adjusting the step size d at different stages and exploring different values of s within a specified range, combined with the learned patterns from various s values, we aim to increase overall accuracy and training speed, stabilize existing clause configurations, and enhance the model's convergence precision.

2. TSETLIN MACHINE

TM is a novel, automaton-based machine learning model that essentially simplifies complex neural network structures into a series of straightforward state transition rules [1] . These rules, grounded in TM, emulate the manner of human logical reasoning. TM uses Tsetlin Automaton (TA) for nonlinear decision strategies, which allows for understanding and interpretability of the problem without sacrificing performance. It enhances the efficiency of Tiny Machine Learing(TinyML) tasks in embedded systems, particularly showcasing real-time and highly interpretable computational advantages on low-power, resource-constrained devices [5]. In terms of real-time performance, TM improves computational efficiency by reducing parameter requirements and opting logic-based processing methods. In real-time interaction fields, such as speech recognition and gesture recognition, TM surpasses traditional neural networks by circumventing the intricacies of weight and gradient adjustments, demonstrating faster convergence and enabling low-power computation [2]. More importantly, the decision-making process of TM is completely transparent, and the logic of each step is traceable and explainable, which makes it possible to deploy machine learning models in sensitive applications.

During the data processing and logical operation stage, the booleanized data is fed into the clause computation module. This module meticulously processes and analyzes the input features based on a boolean logic operation mechanism [6]. In the training process, the output result of each clause undergoes processing by the summation and threshold judgment module, which generates the expected output class through a voting mechanism. This expected output is then compared with the true class label [7]. The result of this comparison triggers two feedback mechanisms: Feedback I (indicating correct classification) and Feedback II (indicating incorrect classification). These two feedback signals will guide TA within the clauses to adjust their internal state values, gradually steering the clauses towards more accurate classification logic during the training process, thereby enhancing the overall detection performance of the system.During the inference stage, the system determines the final classification category based on the voting results of various clauses. Specifically, among multiple candidate categories, the category with the highest number of votes is considered as the system's classification result for the input data.

TM also incorporates hyperparameters s and T to control the frequency of feedback occurrences and the probability of TA state transitions, respectively. A higher value of T encourages the algorithm to explore more novel or uncommon clause combinations, whereas a lower value of T tends to reinforce known effective clause combinations [8]. During the feedback process, the hyperparameter s influences the probability of TA state updates, determining whether a TA changes its state value upon receiving feedback. For both types of feedback, the probabilities of TA rewarding, penalizing, and maintaining its original state are governed by s. A higher value of s implies that upon receiving feedback, TA is less likely to adjust its behavior, leading to a slower but potentially more stable learning process. Conversely, a lower value of s tends to favor the exploration of new rules, accelerating the learning rate but potentially increasing volatility during the learning process.

3. EXPERIMENTS AND RESULTS

3.1. Experimental Setup

In this experiment, the number of clauses was set to 500, the parameter s was set to 10, T was set to 30. Based on the voting statistics depicted in Figure 1 during the training feedback process, the occurrence threshold for multi-level feedback is set at 30 votes.

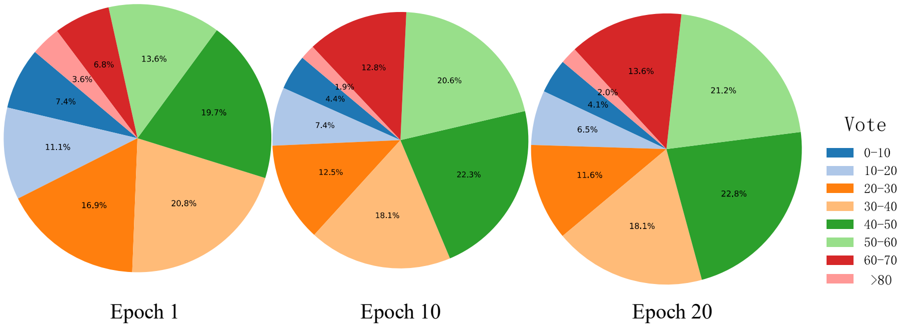

Figure 1. Voting Counts During Training Process

The dynamic parameter, which controls the exploration range of s, adjusts the step size d dynamically [9]. When model performance fails to improve for three consecutive cycles, the step size d is increased to broaden the search scope. Conversely, when performance improves, the step size d is reset to focus on local optimization.

The candidate values for s are given by Equation (1), where $rand()$ represents a random number within the range [0,1] and d = 0.5. The introduction of a random perturbation term disrupts the symmetry of a fixed step size, enhancing the diversity and randomness of the search. Additionally, if the current candidate value results in decreased performance, it reverts back to the historically optimal value of s to avoid getting trapped in a local optimum.

$$s_{(k)} = \begin{cases} s_{\text{best}} + d * rand() & (k=1) \\ s_{\text{best}} & (k=2), \\ s_{\text{best}} - d * rand() & (k=3) \end{cases} \tag{1}$$

3.2. Multi-level Feedback

The essence of the multi-level feedback mechanism lies in refining the intensity of feedback, enabling the model to provide varying degrees of reward or punishment based on the correctness or degree of error in classification results. This mechanism significantly accelerates the training speed of the model, allowing it to achieve higher accuracy in a shorter period.

Figure 2 illustrates the changes in training accuracy over time for both the original and multi-level feedback TM. At the beginning of training, the accuracy of the multi-level feedback TM rapidly increases, particularly within the first 50 training epochs. Specifically, the multi-level feedback TM achieves 98% accuracy within 50 epochs, which is equivalent to the accuracy of the original TM at

200 epochs. Furthermore, the multi-level feedback TM reaches 98.6% accuracy at 200 epochs, comparable to the accuracy of the original TM at 300 epochs. This demonstrates that the multi-level feedback mechanism quickly stabilizes high-contributing clauses through strong positive feedback and promptly eliminates significant errors through strong negative feedback, reducing redundant adjustments and thereby significantly shortening convergence time and accelerating the model's convergence speed. The training speed of the multi-level feedback TM is approximately twice as fast as that of the original TM.

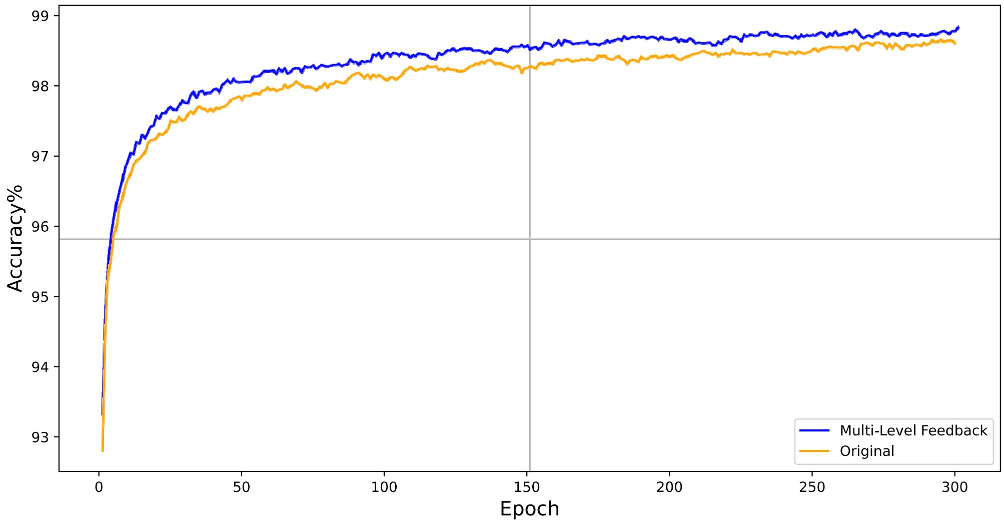

Figure 2. Comparison of Training Accuracy between Original and Multi-Level Feedback TM

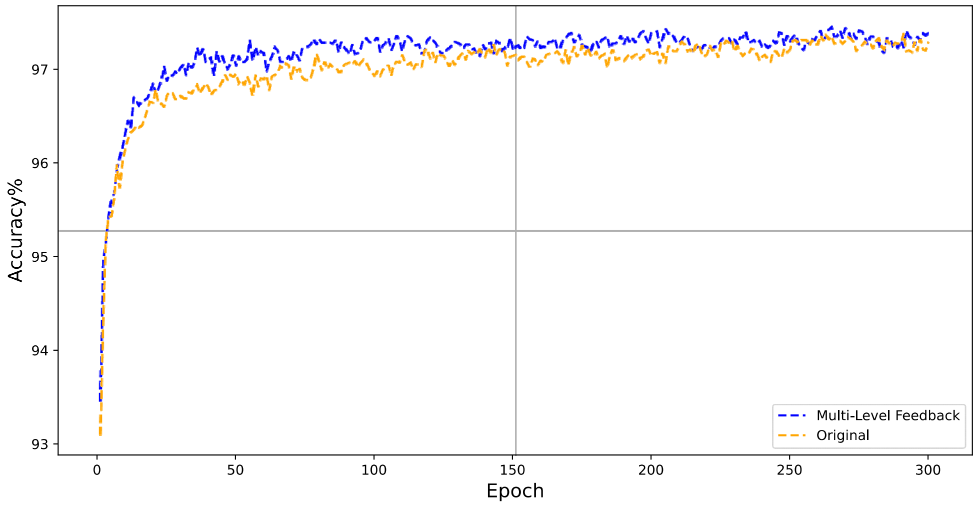

Figure 3. Comparison of Test Accuracy between Original and Multi-Level Feedback TM

Similarly, Figure 3 shows that the accuracy of the multi-level feedback TM on the test set increases more rapidly than that of the original TM; however, the final test accuracy of the multi-level feedback TM is not significantly higher than that of the original TM. This suggests that while the multi-level feedback mechanism accelerates convergence speed, it does not significantly improve the model's generalization ability compared to the significant increase in training accuracy. Essentially, the multi-level feedback mechanism enables the model to achieve higher accuracy in a shorter period by accelerating training speed, but this improvement is primarily observed on the training set. Since the multi-level feedback mechanism primarily enhances model performance by accelerating training speed, its impact on the model's performance on unseen data is limited.

3.3. Dynamic Parameter

The core of the dynamic parameter mechanism lies in the flexibility and adaptability of its phased learning approach. By introducing different parameter s at various stages, the model is able to learn different logical combinations at each stage and progressively accumulate these combinations, thereby exploring more possibilities during the training process. This mechanism significantly impacts both the training speed and ultimate performance of the model.

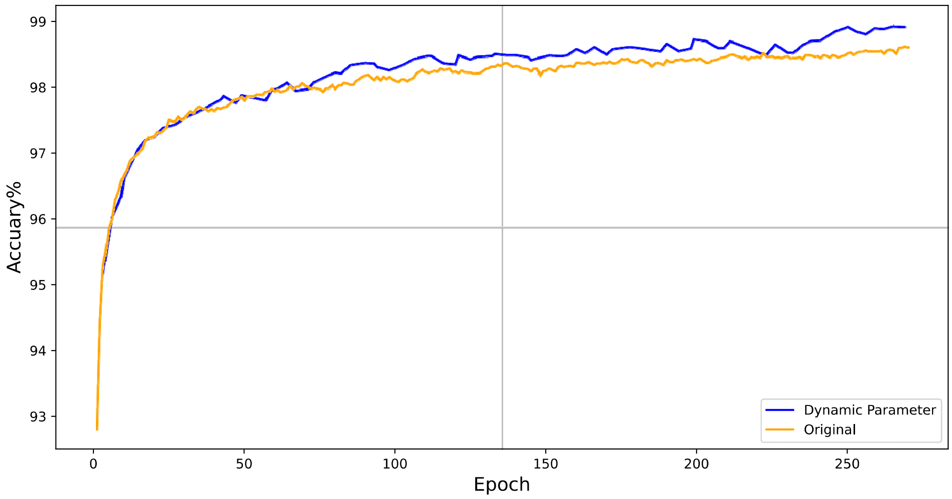

Figure 4. Comparison of Training Accuracy Between Original and Dynamic Parameter TM

As observed in Figure 4, during the initial phase of training, the dynamic parameter Transition TM is in an exploratory stage where it experiments with various logical combinations and accumulates learning experiences. Consequently, during this initial phase, the dynamic parameter TM keeps pace with the original TM in terms of training progress, and the rate of accuracy improvement for both is similar. As the training process advances, the dynamic parameter TM leverages the logical combination knowledge accumulated during the early stages to initiate optimization and adjustment strategies. This transition directly leads to a significant leap in the accuracy of the dynamic parameter TM. Specifically, the accuracy achieved by the dynamic parameter TM after just 100 training epochs is comparable to that of the original TM after 150 training epochs, demonstrating a shortened training cycle.

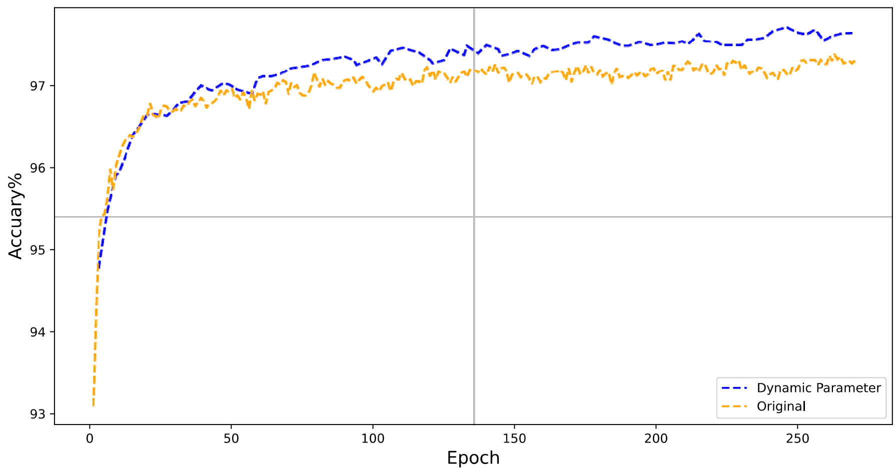

Figure 5. Comparison of Test Accuracy Between Original and Dynamic Parameter TM

Figure 5 reveals a similar trend in the performance of the dynamic parameter TM on the test set compared to the training phase. During the initial training period, the test accuracy of the dynamic parameter TM is close to that of the original TM. However, as training progresses, the test accuracy of the dynamic parameter TM rapidly improves in the mid-to-late stages, ultimately surpassing the original TM by 0.5%. This phenomenon indicates that the dynamic parameter TM not only possesses a significant advantage in training speed but also exhibits superior generalization ability on the test set. Specifically, the dynamic parameter TM, through a more nuanced feedback mechanism, can more effectively adjust the weights of clauses, achieving higher accuracy on the test set. This improvement is primarily attributed to the optimization of logical combinations during the training process by the dynamic parameter mechanism, enabling the model to better capture data characteristics and thus perform better on unseen data.

3.4. Multi-Level Feedback and Dynamic Parameters

The essence of both multi-level feedback and dynamic parameter s is to optimize model performance by adjusting the state of the TA. Multi-level feedback refines the feedback intensity, enabling the model to provide varying degrees of rewards or penalties based on the correctness or error degree of classification results, thereby accelerating changes in the TA and enhancing the model's training speed and classification accuracy. Dynamic parameter s, on the other hand, dynamically adjusts the value of s during the training process, altering the probability of state adjustments in the TA, thus optimizing the exploration-exploitation balance of the model to achieve higher accuracy.

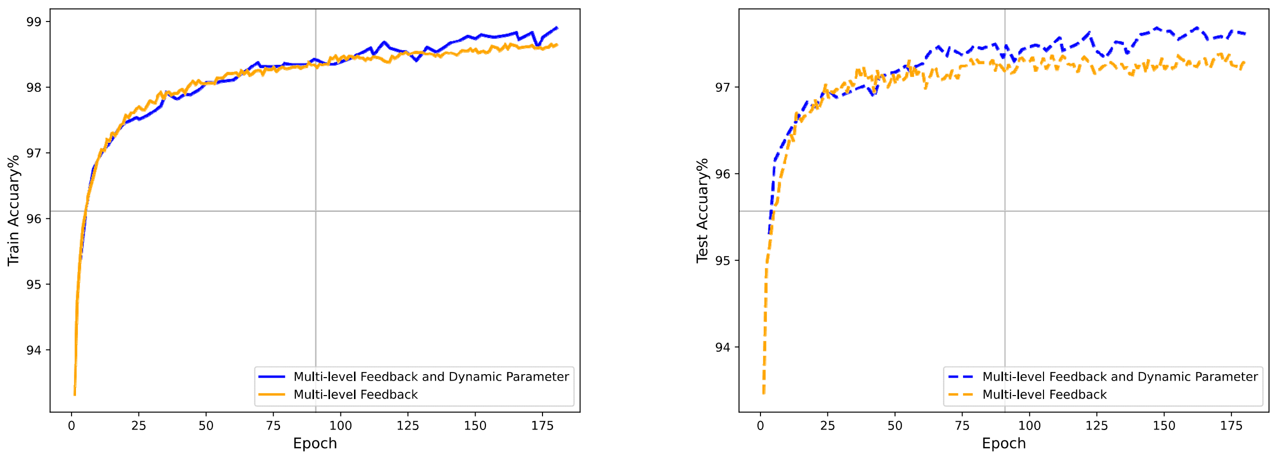

Figure 6. Comparison of Test Accuracy Between Multi-Level Feedback and Dynamic Parameter TM and Multi-Level Feedback TM

Figure 6 illustrates that TM combining multi-level feedback with dynamic parameter exhibits training speeds comparable to the multi-level feedback-only TM. However, in the mid-to-late stages, due to the expansion of the parameter s range, the accuracy increases. This suggests that the introduction of dynamic parameter *s* further optimizes model performance in the mid-to-late stages of training, enabling the model to attain higher accuracy.

Figure 7 reveals that the combined model outpaces the dynamic parameter-only model in training speed, yet the ultimate accuracy improvement is not significant. This indicates that while the multi-level feedback mechanism indeed accelerates changes in the TA during training, thereby enhancing training speed, the ultimate accuracy enhancement is primarily attributed to the optimizing effect of dynamic parameter *s*. This result further validates the complementarity of multi-level feedback and dynamic parameter s in model optimization. Multi-level feedback speeds up changes in the TA during training, rapidly boosting accuracy, while dynamic parameter s leverages varying *s* values to facilitate better changes in the TA, achieving higher accuracy.

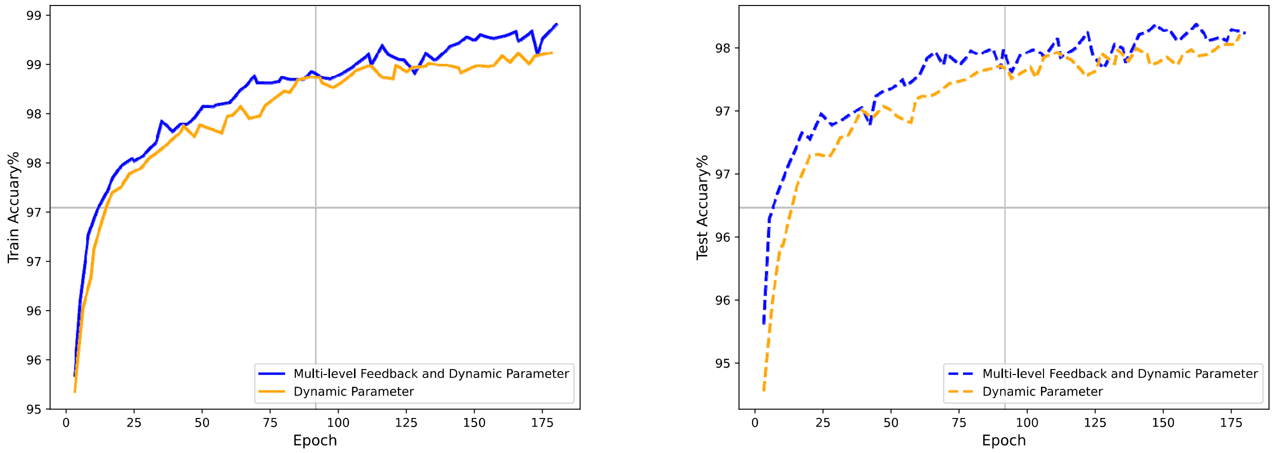

Figure 7. Comparison of Test Accuracy Between Multi-Level Feedback and Dynamic Parameter TM and Dynamic Parameter TM

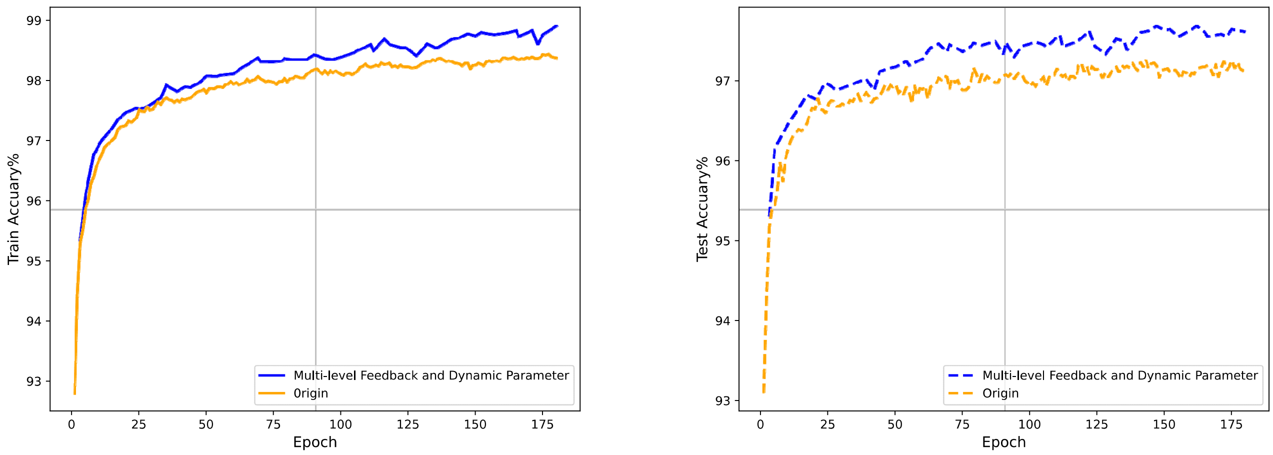

Figure 8. Comparison of Test Accuracy Between Multi-Level Feedback and Dynamic Parameter TM and Original Model

Table 1. Comparison of Accuracy Rates Among Different Clauses

Clause	500 (Original)	500	1000	1500
Accuracy	97.21%	97.71%	98.27%	98.38%

Figure 8 and Table 1 demonstrate that compared to the original TM, TM combining multi-level feedback with dynamic parameter s exhibits a doubled training speed in both training and testing phases. Additionally, the final accuracy increases by 0.5%, from 97.21% to 97.71%. This underscores that the combination of multi-level feedback and dynamic parameter s not only accelerates model training but also significantly enhances model classification accuracy. In summary, the dynamic parameter mechanism significantly enhances both the training speed and classification accuracy of the TM. During training, the dynamic parameter TM achieves faster convergence and higher training accuracy through initial exploration and subsequent optimization. During testing, the dynamic parameter TM not only maintains the advantages gained during training but also demonstrates better generalization ability on the test set, ultimately achieving higher test accuracy.

4. CONCLUSION

This paper primarily explores the application of the multi-level feedback mechanism and dynamic parameter s in TM and their impact on model performance. The multi-level feedback mechanism enhances the model's ability to provide varying degrees of rewards or penalties based on the correctness or degree of error in classification results by refining the feedback intensity. This enables the model to learn effective logical combinations more quickly, thereby improving both the training speed and classification accuracy of the model.The dynamic parameter s optimizes the balance between exploration and exploitation in the model by dynamically adjusting the value of s during the training process, which changes the state adjustment probability of TA. This adjustment allows the model to more effectively explore the parameter space during the initial stages of training to find optimal clause combinations. Meanwhile, in the mid-to-late stages of training, the model can converge more rapidly to achieve higher accuracy.The combination of the multi-level feedback mechanism and dynamic parameter s significantly enhances the training speed and classification accuracy of the TM. The multi-level feedback mechanism accelerates the changes in TAs, enabling the model to improve accuracy more rapidly. The dynamic parameter s optimizes the balance between exploration and exploitation, enabling the model to achieve higher accuracy. The combination of these two methods demonstrates significant advantages across multiple datasets, not only accelerating training speed but also enhancing the model's generalization capabilities, resulting in excellent performance on various datasets.

REFERENCES

[1] O.-C. Granmo, "The tsetlin machine–a game theoretic bandit driven approach to optimal pattern recognition with propositional logic," arXiv preprint arXiv:1804.01508, 2018.

[2] J. Lei, A. Wheeldon, R. Shafik, A. Yakovlev and O. -C. Granmo, "From Arithmetic to Logic based AI: A Comparative Analysis of Neural Networks and Tsetlin Machine," 2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2020, pp. 1-4.

[3] Phoulady A, Granmo O C, Gorji S R, et al. The weighted tsetlin machine: compressed representations with weighted clauses[J]. arxiv preprint arxiv:1911.12607, 2019.

[4] Tarasyuk O, Rahman T, Shafik R, et al. Systematic Search for Optimal Hyper-parameters of the Tsetlin Machine on MNIST Dataset[C]//2023 International Symposium on the Tsetlin Machine (ISTM). IEEE, 2023: 1-8.

[5] R. Shafik, A. Wheeldon, and A. Yakovlev, "Explainability and depend ability analysis of learning automata based ai hardware," in 2020 IEEE 26th International Symposium on On-Line Testing and Robust System Design (IOLTS). IEEE, 2020, pp. 1–4.

[6] J. Lei, T. Rahman, R. Shafik, A. Wheeldon, A. Yakovlev, O.-C. Granmo, F. Kawsar, and A. Mathur, "Low-power audio keyword spotting using tsetlin machines," Journal of Low Power Electronics and Applications, vol. 11, no. 2, p. 18, 2021.

[7] R. Saha, O.-C. Granmo, and M. Goodwin, "Mining interpretable rules for sentiment and semantic relation analysis using tsetlin machines," in International Conference on Innovative Techniques and Applications of Artificial Intelligence. Springer, 2020, pp. 67–78.

[8] K. D. Abeyrathna, O.-C. Granmo, and M. Goodwin, "Extending the tsetlin machine with integer-weighted clauses for increased interpretability," IEEE Access, vol. 9, pp. 8233–8248, 2021.

[9] Olsen A R. A Scalable Architecture for Parallel Execution of the Tsetlin Machine[D]. Universitetet i Agder; University of Agder, 2019.