Speech Recognition Classification Leveraging a Spiking Neural Network Training Algorithm Based on Tempotron

Erbin Ma [#], Wenxin Ge [*, #], Chongxuan Xiao

College of Sciences, Northeastern University, Shenyang, China, 110000

* Corresponding Author Email: master.gwx@gmail.com

[#]These authors contributed equally.

Abstract. This paper examines the potential of Tempotron-based spiking neural network (SNN) training algorithms for speech recognition classification. Initially, the speech signal dataset was processed, with the signal decomposed into different frequency bands through Mel filter banks and the variations in speech intensity encoded using latency encoding techniques. These encoded features were then used as inputs, generating a feature map through the self-organizing map (SOM) unsupervised learning algorithm. This map reflected the similarity and topological structure between different features, providing a more abundant and abstract feature representation. Subsequently, for the task of speech signal classification, the paper established a two-layer SNN based on event-driven mechanisms, employing the improved LIF neuron model for information transmission. The network was then trained using the optimized Tempotron learning algorithm. Finally, the feature spike sequences extracted by the SOM were used as the input for the SNN. Five different weight update rules were employed in the Tempotron learning algorithm, and the resulting accuracy was recorded. The results of the experiment indicated that the combination of the Adam optimization algorithm with the Tempotron learning algorithm achieved a classification accuracy of 98.53%. This evidence demonstrates the efficacy of this paper in optimizing the Tempotron learning algorithm, which will inform future research aimed at enhancing classification results.

Keywords: MFCC, SOM, Improved LIF Model, Two-layer SNN, Adam-Tempotron Learning Algorithm.

1. Introduction

Speech signal processing and recognition have long been pivotal research directions within the field of artificial intelligence [1]. The human brain has been perceived as the optimal exemplification of intelligent entities known till date [2]. As the comprehension of deep learning deepens, paralleled by a heightened understanding of the functional principles of biological neural systems, several researchers have transposed findings from neuroscience into the application of SNNs [3]. Grounded in their study of action potentials in squids, biologists Hodgkin and Huxley put forth the first spiking neuron model - the Hodgkin-Huxley (HH) model [4]. Progressing further, in 2002, Gerstner proposed the Leaky Integrate-and-Fire (LIF) model based on the Hodgkin-Huxley framework. The emergence of the LIF model significantly contributed to understanding the dynamical properties of spiking neurons and found wide applications in subsequent research. However, in comparison to biological neurons, the LIF model was still overly simplistic. This necessitated the development of more complex neuron models to better emulate the operation mechanisms of biological neurons [5], which would significantly impact the design and construction of SNNs. In 2015, Diehl introduced a SNN with a fully connected layer structure, and thereby developed more intricate models for training on large-scale datasets [6].

In terms of learning algorithms, the training of SNNs can be categorized into supervised [7] and unsupervised forms. Unsupervised learning algorithms are primarily the backpropagation algorithms for Spike-Timing-Dependent Plasticity (STDP) [8], while the Tempotron learning algorithm is a classic example of supervised learning methods, proposed by Gütig [9]. In 2009, Masquelier and Thorpe applied STDP to multilayer SNNs [10]. In 2017, Tavanaei optimized this work [11]. In 2018, Wu Jibin utilized Tempotron learning algorithms of a single-layer network architecture on the

TIDIGITS dataset [12], implementing gradient descent methods to ensure that the maximum membrane voltage of the target spiking neurons exceeds the threshold, while that of the non-target spiking neurons persistently remains under the threshold. Nonetheless, the drawback of the Tempotron learning algorithm lies in the maximum one-time spike of output layer neurons, making it unsuitable for use in multi-layer networks. In 2019, Shrestha proposed a new universal backpropagation mechanism, which overcame the non-differentiability issue of spike functions, updated existing learning methods, and attained the latest performance of SNNs on the TIDIGITS dataset [13].

Recent research indicates that associating input spatio-temporal spike patterns with desired spike sequences to establish causal relationships enables the extension of Tempotron algorithms to multi-layer networks [14]. In 2018, Shrestha further investigated the novel backpropagation algorithm, markedly improving the accuracy of multi-layer SNNs [15]. Meanwhile, Zhang Tielin and Xu Bo delved into the multi-level optimization methods for SNNs [16] and Fang, W. [17] proposed "SEW" and "ResNet" to ameliorate degradation issues in deep SNNs. Moreover, in 2021, Cong Shi further developed a hardware-friendly multi-layer SNN method based on previous research [18], providing invaluable insights for the study of multi-layer Tempotron algorithms.

The aim of this paper is to explore the combination of SNNs and SOM for handling and recognizing speech signals to complete classification tasks. The focus of this study is on improving the supervised Tempotron learning algorithm to enhance the accuracy of speech recognition tasks. At the same time, this paper delved into and optimized different synaptic weight update rules to find the best strategy for speech signal classification. Additionally, based on the TIDIGITS dataset, this paper explored other deep learning algorithms which also employed the same dataset. Algorithms demonstrating higher classification accuracy were chosen for a thorough comparison and analysis against the algorithm presented within this paper.

At present, the application of SNNs is primarily concentrated in image classification fields, with relatively less research undertaken in domains like semantic segmentation, speech recognition, and natural language processing [19]. Nevertheless, several researchers have begun investigating and predicting the potential applications of SNNs in syllable-based speech recognition tasks [20]. This paper, through improving the training algorithms for SNNs, holds promise in further promoting the development of speech signal processing and speech recognition sectors, offering superior performance and outcomes for relevant technological applications.

2. SOM-SNN Algorithm

2.1. Algorithm Framework

This research is based on the optimized Tempotron learning algorithm and constructs a two-layer SNN for speech recognition and classification. The approach generally comprises three steps:

1. Preprocessing and Feature Extraction of Speech Signals: The speech signals are transformed into dynamic features through the process of MFCC feature extraction. 2. Generating Feature Mapping Using the SOM Algorithm: The dynamic features are encoded and dimension-reduced through the SOM algorithm to obtain low-dimensional features. 3. Constructing an SNN for Speech Recognition and Classification: A two-layer SNN is constructed, the LIF neuron model is enhanced, and five distinct weight update methodologies are employed to optimize the Tempotron learning algorithms. Ultimately, the Adam-Tempotron learning algorithm, exhibiting the most optimal performance, is developed. The framework of the specific algorithm implementation is shown in the Figure 1 below:

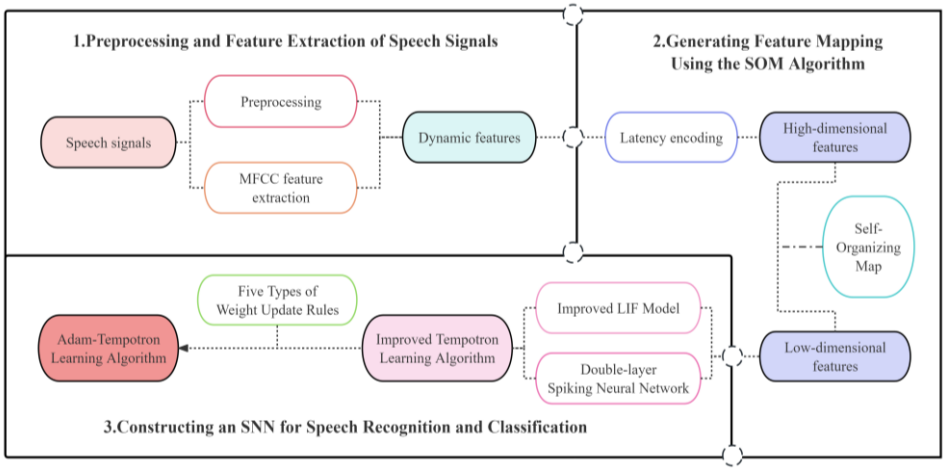

Figure 1. SOM-SNN Algorithm Framework

2.2. Theory of Speech Processing

2.2.1 MFCC Feature Extraction

The fundamental principle of Mel Frequency Cepstral Coefficients (MFCC) [21] feature extraction is to transform the speech signal into the Mel scale frequency domain and extract the inverted spectral parameters in accordance with the characteristics of the human auditory system. The fundamental principle of feature extraction encompasses the following steps, as illustrated in Figure 2.

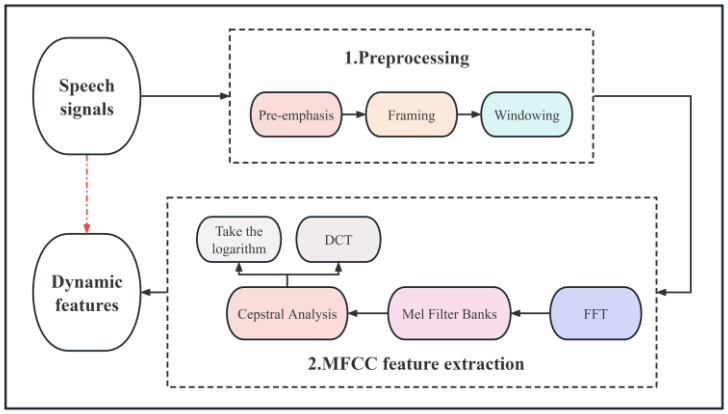

Figure 2. MFCC Feature Extraction Steps

The process primarily involves two steps: ①Preprocessing, which includes: emphasis on the high-frequency portion of the speech via pre-emphasis; signal segmentation into short-time windows via framing and windowing. ②MFCC feature extraction, which includes: performing Fast Fourier Transform to obtain spectral information; utilizing Mel filters on the spectrogram for filtering, converting it into the Mel scale and extracting energy; taking the logarithm of energy in the Mel filter bank, performing Discrete Cosine Transform to acquire the final dynamic features.

2.2.2 Latency Encoding

Time encoding conveys information through the arrival times of different spikes, while latency encoding is based on time encoding and uses precise timing of spikes and latency for encoding.

In latency encoding, the features of the speech signal can be mapped onto the release latencies of the spikes. Different speech features such as pitch, tone, etc. can be represented through spike encoding using different latency times.

2.2.3 Self-Organizing Map (SOM)

The SOM [22] is a classic unsupervised learning neural network algorithm. SOM enables the clustering and visualization of high-dimensional data by mapping it into a low-dimensional

topological structure (typically a two-dimensional grid), thereby simulating the information processing mechanism observed in the human cerebral cortex. As illustrated in Figure 3, the network structure is as follows:

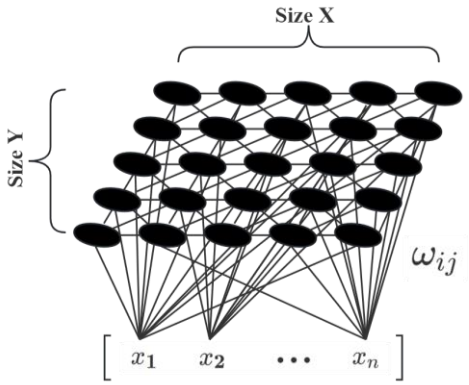

Figure 3. A SOM network structure that maps the n-dimensional space of input objects to an output plane with topological properties.

Competitive learning among neurons plays a pivotal role in the training process of SOM. Each neuron engages in a competitive process to identify itself as the optimal matching unit for the input data. Upon being selected as the optimal matching unit, the neuron adjusts its weight vector to more closely align with the input data, and this adjustment affects the weight vectors of the neuron's neighboring neurons to some extent. Gradually adjusting the connection weights between neurons enables the network to establish an ordered mapping of the frequently activated neurons, thereby enhancing the capture of the input data's features and facilitating the generation of a more ordered representation.

2.3. Theory of Speech Classification

2.3.1 SNN Introduction

This is a bio-inspired neural network model that simulates the spike transfer between neurons in the biological nervous system. Different from traditional neural networks, SNN [23] conveys information by simulating output spike sequences of neurons, as shown in Figure 4.

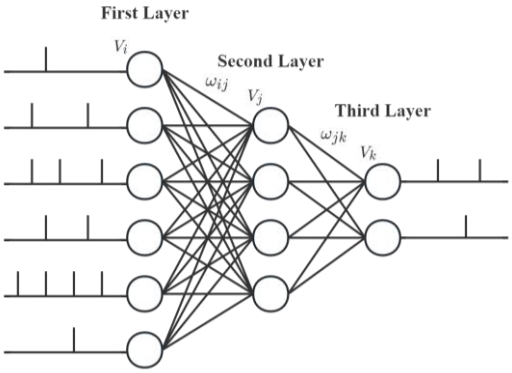

Figure 4. SNN Model

Where, V represents a neuron's identifier; ω represents the synaptic weights between interconnected neurons across different layers.

Distinct from traditional neural networks, SNN substitutes binary spike signals for floating-point numbers to perform computations, thus discretizing the input signals into 0 and 1. Moreover, in SNN, the passage of information is determined by event-driven factors. A single neuron remains in a silent state when there are no inputs. Upon receiving inputs, the information is accumulated within a specific time window. An event is triggered only when the membrane voltage reaches the neuronal threshold, after which the neuron's state is reset.

2.3.2 Improved LIF Model

This model overlooks the complex dynamics of neuronal membrane potential and the details of ion channels. It's a simplified neuron model with the following basic principles:

Membrane Potential Evolution: The membrane potential of a neuron satisfies an ordinary differential equation, describing the evolution of membrane potential over time, as depicted below:

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

Where, $V(t)$ represents the membrane voltage of a neuron, which is the weighted sum of all spike inputs from synaptically connected preceding neurons; τ_m denotes the neuron's membrane potential integration time constant; V_{rest} indicates the resting potential; I_t represents the input current, accounting for the sum of synaptic inputs, calculated as follows:

$$I(t) = \sum \omega \sum_{t=1}^{T} S_i(t) \tag{2}$$

Where, ω represents the synaptic weight of a certain neuron at moment t; $\sum_{t=1}^{T} S_i(t)$ accounts for the input spike sequence of a certain neuron at moment t, calculated as follows:

$$S_i(t) = H\left(V(t) - \theta\right) \tag{3}$$

Where, $H(x)$ symbolizes the Heaviside function; θ denotes a given neuron's threshold.

1. Spike Generation: A threshold potential (V_{th}) is introduced. When the membrane potential exceeds this threshold, the model generates a spike output and resets the membrane potential to the resting potential (V_{rest}). The expression is as follows:

$$If \quad V(t) \geq V_{th}, \quad V(t) = V_{rest} \tag{4}$$

2.3.3 Tempotron Learning Algorithm

This is a bio-plausible learning algorithm used for binary classification of neurons. Figure 5 demonstrates the Tempotron rule: the spiking neuron integrates incoming spikes into the membrane potential and when the membrane potential exceeds a certain threshold, it generates an output spike [9].

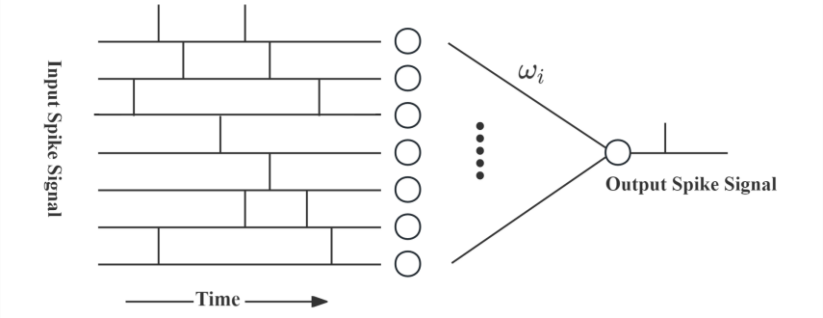

Figure 5. Demonstration of the Tempotron Rule

In this algorithm, each input sample is classified into one of two categories, represented as Y and N. The classification decision is made based on whether or not the Tempotron neuron generates a spike - it should spike in the Y category mode, and remain at rest in the N category mode. When misclassification occurs, the Tempotron learning algorithm adjusts synaptic weights to gradually optimize the decision accuracy of the neuron.

2.3.4 Adam Weight Update Rule

Adam [24] is an adaptive moment estimation algorithm, combining the advantages of SGDM (Stochastic Gradient Descent with Momentum) and RMSprop [25]. It adjusts the learning rate for each parameter adaptively by computing the first-order and second-order moment estimates of the gradient at a certain time step, thereby adapting to the variations of different parameters. The formula for calculating the gradient at time step t is as follows:

$$g_t = \nabla f(\omega_t, V_t) \tag{5}$$

Where, V_t represents a certain neuron at time t ; ω_t represents the synaptic weight corresponding to this neuron at time t. Next, the first-order moment estimate (m_t) and the second-order moment estimate (v_t) of the updated gradient are calculated, with the formulas are as follows [24]:

$$\begin{aligned} m_t &= \lambda_1 m_{t-1} + (1-\lambda_1) g_t \\ v_t &= \lambda_2 v_{t-1} + (1-\lambda_2) g_t^2 \end{aligned} \tag{6}$$

Where, $\lambda_1, \lambda_2 \in [0,1)$ denote the exponential decay rates for the moment estimates. Then, correct the bias of the first-order and second-order moments, according to the following formula:

$$\begin{aligned} m_t &= \frac{m_t}{1-\lambda_1^t} \\ \hat{v}_t &= \frac{v_t}{1-\lambda_2^t} \end{aligned} \tag{7}$$

Finally, update the synaptic weights, as demonstrated by the following formula:

$$\omega_{t+1} = \omega_t - \alpha \frac{m_t}{\sqrt{\hat{v}_t} + \varepsilon} \tag{8}$$

Where, α represents the global learning rate; ε is an extremely small constant that can ensure numerical stability.

3. Experiment Content

3.1. Model Framework Method

This research is based on an optimized Tempotron learning algorithm, constructing a two-layer SNN for speech recognition. The detailed three-step process can be seen in Figure 1, and Figure 6 illustrates the algorithm implementation framework as follows:

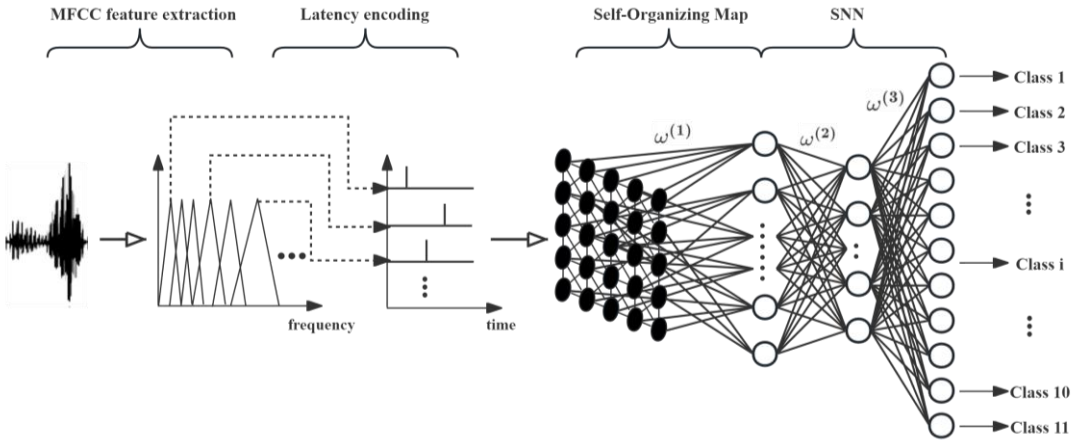

Figure 6. Algorithm Implementation Framework: Firstly, the speech signal is decomposed using Mel filters in order to extract MFCC features. These features are then applied to a latency encoding of the intensity changes in each frequency band. Finally, the SOM algorithm is used to generate a feature map, which provides a rich and abstract representation of the features. Finally, an event-driven, two-layer SNN is constructed, which utilizes the feature sequence obtained from the SOM as an input in order to achieve the objective of speech recognition and classification.

3.2. Preprocessing and Feature Extraction of Speech Signals

In order to convert the speech signals into a digital form, it is first necessary to pre-process the input speech signals. As speech signals are a non-stationary process, they continuously change over time on a global scale. As a result, digital signal processing techniques which are typically used for stationary signals cannot be directly applied to the entire speech signal. However, over a shorter range of time (approximately between 10 to 30 milliseconds), the characteristics of speech signals are relatively stable and can be considered stationary.

Therefore, this paper initially framed and applied a window function to the complete speech signal. In order to maintain continuity between frames, frame shifting was required, which involved setting a certain overlap between adjacent frames. The framing operation is achieved by weighting with a moving window of finite length. In this experiment, the Hamming window was selected as the window function to obtain the windowed speech signal. Subsequently, in order to convert the one-dimensional time-domain speech signal into a frequency-domain signal for processing, a FFT is implemented for each frame. This operation transforms the one-dimensional speech signal into a two-dimensional spectrogram, providing information in the time domain, frequency domain, and energy, among others.

Next, the paper performed the calculation of the MFCC, the formula for which is as follows:

$$Mel(f) = 2595 * \lg(1 + \frac{f}{700}) \tag{9}$$

Where, f represents the actual frequency, unit: Hz; $Mel(f)$ indicates the Mel scale frequency, unit: Mel.

This process also includes the Discrete Cosine Transform of the spectrum, as well as taking the logarithm of the obtained coefficients. This operation helps to convert the characteristics of the audio signal to a set of more representative coefficients for subsequent pattern recognition and speech recognition tasks.

Subsequently, the paper conducted a normalization operation, which aided in consolidating the data between 0 and 1. This ensured that each feature contributes equally to the result, thereby improving the accuracy of the classifier model.

Finally, this paper applied latency encoding techniques to encode variations in speech intensity. For the changes in speech intensity within each frequency band, the paper selected suitable latencies

and phase differences to transform the amplitude change information into a form characterized by phase difference. This process involved reversing the operation of phase difference and latency information to reconstruct the signal.

3.3. Generating Feature Mapping Using the SOM Algorithm

This paper applied the encoded data to the SOM unsupervised neural network learning algorithm to further obtain dynamic speech features.

It is known that the core of the SOM clustering algorithm is a two-dimensional neural network, composed of multiple neurons. Each neuron has a weight vector, which signifies the features of the data point it represents. During the training process, each data point competes with surrounding neurons and gets mapped to the nearest neuron, thereby forming different clusters.

The training of the SOM clustering algorithm is divided into the initialization phase and the iteration phase. During the initialization phase, the weight vector of each neuron needs to be randomly initialized to represent different regions in data space. During the iteration phase, the weight vector of the neurons will be continuously adjusted to better represent the features of the data points. Each time during the iteration, a data point is chosen at random, and the neuron closest to it is found. The weight vector of that neuron is then adjusted to be closer to the features of that data point. Additionally, the weight vectors of surrounding neurons are also adjusted to more closely resemble the features of that data point. Through multiple iterations, the weight vectors of the neurons will gradually converge, forming diverse clusters.

This paper initially tried using the K-means clustering algorithm, but the results were not satisfactory. This is because K-means needs to determine the number of classes first and only updates relevant parameters when the most similar class is found, making it susceptible to interference from noise data. Considering that the dataset includes pronunciation of the digits 0-9 and some pronunciations without explicit meaning, the paper eventually chose to use the SOM algorithm. The SOM algorithm exhibits superior performance in terms of accuracy and topology by maintaining the weight topography of neighbor nodes. The parameters featuring more distinct classification characteristics generated by the SOM algorithm are used as input vectors for training the neural network.

In the classification recognition task, feature information of speech signals can be obtained through the SOM algorithm. As illustrated by the relevant research [12] and through several adjustments of the output neurons of the SOM network, the paper finally decided to set the output of the SOM network to 576 neurons since this number of output neurons results in the best classification effects for the speech recognition task.

3.4. Constructing an SNN for Speech Recognition and Classification

Before implementing speech recognition classification, it should be noted that the construction of a speech recognition SNN requires the appropriate selection of network simulation strategies, information encoding methods, information transmission methods (i.e., spiking neuron models), and matching learning algorithms derived from the features of speech signals and the characteristics of the SNN.

Through previous work, the paper had extracted the features of the speech signals and performed latency encoding to convert them into spike sequences. Next, it was necessary to select an appropriate neuronal model and learning algorithm for speech recognition classification. Inspired by related research [12], the paper ultimately selected the LIF model and the Tempotron learning algorithm. However, improvements needed to be made in order to successfully construct an SNN.

For this experiment, this paper used an improved version of the LIF model [26], which drives LIF neurons through exponentially decaying synaptic currents, the equation is as follows:

$$V(t) = \sum \omega \sum K(\Delta t) + V_{rest} \quad \forall t \in [0, T]$$
$$K(\Delta t) = V_0 \left(\exp(\frac{-\Delta t}{\tau_m}) - \exp(\frac{-\Delta t}{\tau_s}) \right) \tag{10}$$
$$\Delta t = t - t_i > 0$$

Where, $K(\Delta t)$ represents the postsynaptic membrane potential kernel function with values ranging from 0-1; τ_s denotes the decay time constant of the synaptic current; T symbolizes the time length of the coding time window; t and t_i respectively represent the moment when a neuron reaches a certain potential value within the time domain and the moment when the spike is released.

As for the Tempotron learning rule, even though it was initially designed for binary classification tasks, it could also adopt a one-versus-all strategy to handle multi-class classification tasks [12]. Specific to this experiment, an output neuron needed to be trained for each unique class. During training, for a neuron representing class i, the paper marked all training samples with label i as Y (positive samples), while other samples were labeled as N (negative samples). During testing, the paper detected the membrane potentials of all output neurons and determined the sample classification according to observed activation signals. If no output neurons generated effective activation signals, the paper marked the test samples using the output neuron with the highest membrane potential. Conversely, if any neuron generated an effective activation signal, the paper marked the samples based on the neuron that first sends an activation signal.

Specifically, the Tempotron learning algorithm optimizes the synaptic weight by minimizing the error between the target value and the actual output membrane potential. The learning rule is as follows:

$$\Delta \omega_i = \begin{cases} \lambda \sum K \Delta t_{\max}, & if\ Y \\ -\lambda \sum K \Delta t_{\max}, & if\ N \\ 0, & otherwise \end{cases} \tag{11}$$

Where, $\Delta t_{\max}$ represents the duration when a certain neuron attains maximum potential value after firing a spike within the time domain; λ denotes the learning rate, signifying the maximum change value of the synaptic weight.

After completing the tasks mentioned above, the SNN could then be officially constructed to complete the speech recognition classification tasks. Moving forward, the paper aimed to classify the speech signals by building an event-driven, two-layer SNN. After obtaining SOM's output neurons, it was necessary to use these as inputs for the SNN, designing it so that 11 neurons represent eleven speech categories, and initially stipulating the number of neurons in the intermediate layer as 200. Thereafter, in order to find a suitable number of neurons in the two-layer SNN for achieving better classification effects, the paper randomly selected a weight update strategy and a set of base parameters, and repeatedly adjusted the number of neurons in the intermediate layer. Ultimately, it was determined that setting it to approximately 400 leads to the highest classification precision.

Subsequently, the paper used improved LIF neurons in the network to simulate the process of information transmission in the human brain. An optimized Tempotron learning algorithm was applied for training. This algorithm simulates how neurons adjust synaptic weights upon receiving spike inputs to achieve pattern classification. The paper coupled this with the Adam update rule. The Tempotron rule updates the neuron responses and adjusts the weights, while the first and second moment estimations in the Adam algorithm carry out gradient updates and dynamic learning rate adjustments. This results in a more efficient optimization of the weights and accelerates model convergence.

In addition, this paper expanded the scope of the study and applied four other different synaptic weight update rules to optimize the Tempotron learning algorithm. These included standard Stochastic Gradient Descent (SGD), Stochastic Gradient Descent with Momentum (SGDM), Mini-Batch Gradient Descent (MBGD), and the RMSProp update rule [27].

Once all preparations were completed, this paper used the designed network to take the feature spike sequences extracted by SOM as model inputs. Then, the paper carried out a series of experiments using the Tempotron learning algorithm under different weight update rules and recorded their respective classification accuracy. During the research process, the paper first adjusted parameters such as threshold potential V_{th} and the decay time constant τ_m of synaptic current through deep analysis of the causes of changes in the classification accuracy, in pursuit of the best parameter configuration. Then, upon fixing the optimum parameters, the paper assessed the classification performance of five weight update rules across eleven different speech categories, seeking to identify the best weight update rule. This process aimed to further enhance the performance and generalization capability of the algorithm, achieving better speech recognition tasks.

Through systematic experiments and parameter adjustments, the paper strived to explore and sought the optimal model configuration, and ensured the effectiveness and reliability of the algorithm in speech classification tasks, providing strong support for further research and application.

4. Results

4.1. Dataset

A speech dataset with a high degree of popularity was selected for this study: the TIDIGITS dataset [28]. This dataset contains WAV files categorized by gender and speaker, with the speech signal sampled at 20 kHz.

This dataset is regarded as a commonly used benchmark for speech recognition algorithms, comprising digit speech from 111 male and 114 female speakers. The dataset was used for SOM training, with all 12,373 contiguous digit pronunciations, and for SNN training and testing, with isolated digit pronunciations. A random selection of 3,500 digit pronunciations was used for training, with the remaining 1,450 used for testing. The isolated digits comprise the digits "zero" to "nine" and the letter "oh", which correspond to the categories to be trained and tested in the speech recognition task, respectively. These categories are numbered "1" to "11".

4.2. Relevant Parameters

After identifying the datasets and algorithms used in the experiment, it was necessary to determine the optimal values of relevant parameters, as well as the best choices for weight updating rules, through multiple experiments. However, due to the randomness in the division of both training and test sets, as well as the relevant random steps involved in the training process, each training and testing data object varies, thereby influencing the final accuracy rate.

To mitigate these impacts of randomness, this paper adopted the mean value of the results from six separate runs under the same experimental conditions and processes as the final outcome of the model. That is to say, each accuracy value presented in this paper represents the mean result obtained from six repeated runs of the program.

It should be noted that: With each run of this experiment, the classification accuracy for the 11 different classes of numerical speech, as well as the mean classification accuracy across these 11 classes of speech, is obtained. The calculation formula for the accuracy of each class of speech recognition ($Accuracy(i)$) is as follows:

$$Accuracy(i) = \frac{num_i(r)}{num_i(t)} \tag{12}$$

Where, $num_i(r)$ denotes the count of correctly classified instances for a certain class i of numeric speech in the test set; $num_i(t)$ denotes the total count of instances for the certain class i of numeric speech in the test set.

Moreover, within this experiment, the mean accuracy rate of speech recognition and classification across the 11 classes is regarded as the accuracy rate of the corresponding algorithm run.

In the classification recognition tasks, numerous experiments were conducted in terms of parameter setting, revealing the two parameters having the most significant impact on the model: threshold potential V_{th} and neuron membrane potential integration time constant τ_m. By setting other reasonable parameters and iterating through multiple experiments, this paper obtained graphs of classification accuracy under different parameters based on five weight update rules, as shown in Figure 7 and Figure 8.

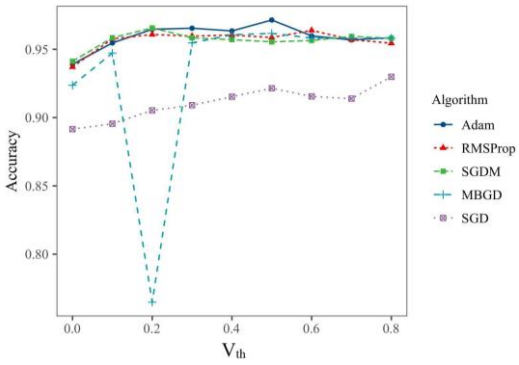

Figure 7. Graph of accuracy changes for the five weight update rules at different V_{th}

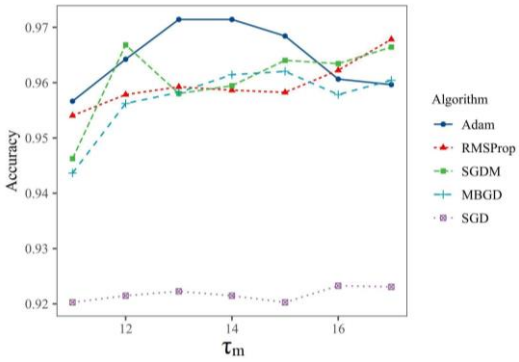

Figure 8. Graph of accuracy changes for the five weight update rules at different τ_m

The figures indicate that the classification accuracy of the algorithm based on the five weight updating rules is comparable to the optimal value when the threshold potential V_{th} is 0.5 and the membrane potential integration time constant τ_m is 14ms. Consequently, the subsequent experiments will be conducted under these parameters.

4.3. Weight Update Rules

Figure 9 intuitively displays the recognition accuracy of 11 different speech categories corresponding to the five weight update rules obtained under optimal parameter conditions, as well as the comparison of recognition accuracy stability under different weight update rules.

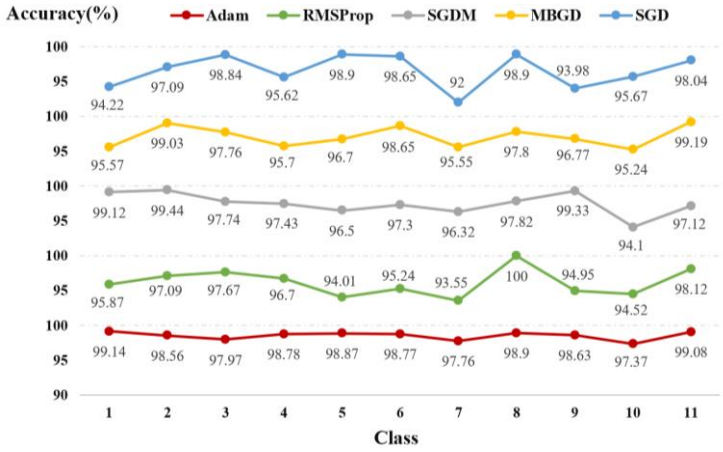

Figure 9. Comparison graph of recognition accuracy and stability under different weight update rules

The graphs demonstrate that the Adam weight updating method, when employed, performs most consistently on the test data in each category, exhibiting excellent performance. In particular, the algorithm based on Adam weight updating achieves a classification accuracy of 98.53%, indicating excellent classification ability. This indicates that the first-order and second-order moment estimation of the Adam algorithm plays a pivotal role in the weight update process for the salient weight adjustment of neurons. The combination of Tempotron's rule and Adam's algorithm results in highly stable and accurate results in pattern classification tasks.

Based on these results, this paper could conclude that the Adam weight update method is an effective strategy. It can quickly and accurately adjust synaptic weights when neurons receive spiking inputs, thereby achieving high efficiency and accuracy in pattern classification.

4.4. Relevant Algorithms

In this experiment, the TIDIGITS dataset was employed for the purpose of conducting other speech classification tasks. The algorithms utilized, the network model structure, and the comparison of classification accuracy results are presented in Table 1.

Table 1: Detail comparison of different algorithms under the TIDIGITS dataset

Algorithm	Architecture	Accuracy
①Tavanaei et al.[11]	Spiking CNN and HMM	96.00%
②SOM-SNN[12]	MFCC-SOM-SNN	97.60%
③SLAYER[15]	MFCC-SOM,484-500-500-11	99.09%±0.13%
④Adam-Tempotron	MFCC-SOM-SNN,576-400-11	98.53%

Algorithm ① employs unsupervised learning to extract discriminative features from speech signals, thereby achieving feature discovery through an architecture comprising spiking convolution/pooling layers and fully connected spiking layers. Furthermore, a classification accuracy of 96% was achieved through the use of probabilistic LIF neurons and spike timing correlation plasticity learning rules. Algorithm ④, which was employed in this experiment, employs different strategies and techniques in dealing with the speech recognition classification task in comparison to Algorithm ①. It achieves a higher accuracy rate (98.53%) through well-designed feature extraction and optimized learning algorithms. Taken together, Algorithm ④ shows more outstanding performance in terms of performance metrics, combining different levels of optimization and feature processing techniques to improve the effectiveness and accuracy of speech recognition classification.

In comparison to Algorithm ②, Algorithm ④ employs a comparable approach to pre-processing and feature extraction of speech signals. However, in terms of the neural network structure, the single-layer SNN is expanded into a two-layer structure, and the Adam weight update rule is combined with the Tempotron learning algorithm, thus enabling more effective capture and representation of

complex patterns and features in the data. This structural improvement enhances the performance of Algorithm ② in the speech recognition task, resulting in a higher classification accuracy (98.53%), which demonstrates a clear advantage compared to Algorithm ② (97.60%).

Algorithm ③ adopts the SRM spike neuron model and the direct training algorithm, and constructs a three-layer network structure of SNN. Although Algorithm ④ has a slightly lower accuracy rate compared to Algorithm ③, the SNN constructed by Algorithm ④ performs better in terms of convergence speed. Here, this paper compared the two networks by calculating their compute cost (C_{cost}), with the equations as follows [29]:

$$C_{cost} = C_{add} \tag{13}$$

Where, C_{add} represents the computation cost of the additions in the network, which is directly proportional to the total number of spike events.

Therefore, compared to Algorithm ③, Algorithm ④ involves lower computation cost and faster convergence speed, making it suitable for use in applications that require high real-time performance [29].

In conclusion, the Adam-Tempotron algorithm, which combines the Adam algorithm with the Tempotron learning algorithm, achieved a classification accuracy of 98.53% on the TIDIGITS dataset. This result demonstrates that the optimization of the Tempotron learning algorithm in this study is reasonable and helps to improve future research work to achieve more superior classification results.

This experiment demonstrated the potential for improving Tempotron, a supervised learning algorithm based on synaptic weight adjustment, and provided valuable experience and guidance for future SNN model design and optimization. This paper not only enriches the theoretical paradigm of SNN learning algorithms but also verifies their practical application in speech recognition through experimental verification.

5. Conclusions

This paper employs a SNN based on spiking neurons, combined with the optimized Tempotron learning algorithm, to achieve higher accuracy in speech recognition classification tasks. By using Mel filter banks to decompose signals, latency encoding technology to handle changes in speech intensity, and combining the SOM unsupervised learning algorithm to produce feature mapping, the algorithm in this paper realizes rich and abstract feature representation. Building on the research of Tempotron algorithm by scholars at home and abroad, a two-layer SNN network is established using the improved LIF neuron model to simulate information transmission, and a variety of weight update rules are used to optimize the Tempotron learning algorithm for training, effectively improving the classification accuracy.

However, in addition to the content discussed in this paper, there are many questions that need further discussion:

1. The Tempotron algorithm learns by optimizing the difference between the membrane potential's target and maximum values through gradient descent. However, it can emit at most one spike, limiting its expressiveness. It might be worth considering improving this algorithm by incorporating multiple sets of spikes, enabling it to directly tackle multi-classification problems.
2. Although SNN has unique advantages, it also has some limitations, such as high training complexity and high hardware requirements. Therefore, using it in combination with other types of neural network models can play to their respective strengths while overcoming their limitations. Also, improving the speech data feature extraction algorithm by reducing the feature dimension can enhance algorithm execution efficiency.
3. The speech data used consist of simple words, and there is little background noise in the data. It would be beneficial to further enhance the robustness and noise resistance of the algorithm, allowing it to be applied to more speech scenarios and data sets. This could widen the application

scenarios of the SNN, further tapping into its potential and promoting its implementation in practical applications.

References

[1] Pietrzak P, Szczęsny S, Huderek D, et al. Overview of Spiking Neural Network Learning Approaches and Their Computational Complexities [J]. Sensors (Basel, Switzerland), 2023, 23(6).

[2] Cox, D. D., Dean, T. Neural networks and neuroscience-inspired computer vision [J]. Current biology: CB, 2014, 24(18): R921-r929.

[3] Cao Y, Chen Y, Khosla D. Spiking Deep Convolutional Neural Networks for Energy-Efficient Object Recognition [J]. International Journal of Computer Vision, 2015, 113(1): 54-66.

[4] Hodgkin A L, Huxley A F. A quantitative description of membrane current and its application to conduction and excitation in nerve [J]. The Journal of physiology, 1952, 117(4): 500-544.

[5] Wang Q, Zhang T, Han M, et al. Complex dynamic neurons improved spiking transformer network for efficient automatic speech recognition [Z]. Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence. AAAI Press. 2023: Article 12.10.1609/ aaai. v37i1.25081

[6] Diehl P U, Cook M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity [J]. Frontiers in computational neuroscience, 2015, 9: 99.

[7] Knudsen E I. Supervised learning in the brain [J]. The Journal of neuroscience: the official journal of the Society for Neuroscience, 1994, 14(7): 3985-3997.

[8] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors [J]. Nature, 1986, 323(6088): 533-536.

[9] Gütig R, Sompolinsky H. The tempotron: a neuron that learns spike timing-based decisions [J]. Nature neuroscience, 2006, 9(3): 420-428.

[10] Masquelier T, Guyonneau R, Thorpe S J. Competitive STDP-Based Spike Pattern Learning [J]. Neural computation, 2009, 21(5): 1259-1276.

[11] Tavanaei A, Maida A. Bio-inspired Multi-layer Spiking Neural Network Extracts Discriminative Features from Speech Signals; proceedings of the Neural Information Processing, Cham, F 2017//, 2017 [C]. Springer International Publishing.

[12] Wu J, Chua Y, Li H. A Biologically Plausible Speech Recognition Framework Based on Spiking Neural Networks; proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), F 8-13 July 2018, 2018 [C].

[13] Shrestha A, Fang H, Wu Q, et al. Approximating Backpropagation for a Biologically Plausible Local Learning Rule in Spiking Neural Networks [Z]. Proceedings of the International Conference on Neuromorphic Systems. Knoxville, TN, USA; Association for Computing Machinery. 2019: Article 10.10.1145/3354265.3354275

[14] Yu Q, Tang H, Hu J, et al. Temporal Learning in Multilayer Spiking Neural Networks Through Construction of Causal Connections [M]//YU Q, TANG H, HU J, et al. Neuromorphic Cognitive Systems: A Learning and Memory Centered Approach. Cham; Springer International Publishing. 2017: 115-129.

[15] Shrestha S B, Orchard G. SLAYER: Spike Layer Error Reassignment in Time [J]. Adv Neur In, 2018, 31.

[16] Zhang T L, Xu B. Research Advances and Perspectives on Spiking Neural Networks [J]. Jisuanji Xuebao/Chinese Journal of Computers, 2021, 44: 1767-1785.

[17] Fang W, Yu Z F, Chen Y Q, et al. Deep Residual Learning in Spiking Neural Networks [J]. Advances in Neural Information Processing Systems 34 (Neurips 2021), 2021, 34: 21056-21069.

[18] Shi C, Wang T, He J, et al. DeepTempo: A Hardware-Friendly Direct Feedback Alignment Multi-Layer Tempotron Learning Rule for Deep Spiking Neural Networks [J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2021, 68(5): 1581-1585.

[19] Morabito F C, Kozma R, Alippi C, et al. 1 - Advances in AI, neural networks, and brain computing: An introduction [M]//KOZMA R, ALIPPI C, CHOE Y, et al. Artificial Intelligence in the Age of Neural Networks and Brain Computing (Second Edition). Academic Press. 2024: 1-8.

[20] Ramírez-Mendoza A M E, Yu W, Li X. A New Spike Membership Function for the Recognition and Processing of Spatiotemporal Spike Patterns: Syllable-Based Speech Recognition Application [J]. Mathematics, 2023, 11(11): 2525.

[21] Darch J, Milner B, Vaseghi S. Analysis and prediction of acoustic speech features from mel-frequency cepstral coefficients in distributed speech recognition architectures [J]. The Journal of the Acoustical Society of America, 2008, 124(6): 3989-4000.

[22] Naskath J, Sivakamasundari G, Begum A A S. A Study on Different Deep Learning Algorithms Used in Deep Neural Nets: MLP SOM and DBN [J]. Wireless personal communications, 2023, 128(4): 2913-2936.

[23] Tavanaei A, Ghodrati M, Kheradpisheh S R, et al. Deep learning in spiking neural networks [J]. Neural Networks, 2019, 111: 47-63.

[24] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization [J]. arXiv e-prints, 2014: arXiv:1412.6980.

[25] Zou F, Shen L, Jie Z, et al. A Sufficient Condition for Convergences of Adam and RMSProp; proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), F 15-20 June 2019, 2019 [C].

[26] Lee J H, Delbruck T, Pfeiffer M. Training Deep Spiking Neural Networks Using Backpropagation [J]. Frontiers in neuroscience, 2016, 10: 508.

[27] Nguyen L C, Nguyen-Xuan H. Deep learning for computational structural optimization [J]. ISA transactions, 2020, 103: 177-191.

[28] R. Gary Leonard, and George Doddington. TIDIGITS LDC93S10. Web Download. Philadelphia: Linguistic Data Consortium, 1993.

[29] Deng L, Wu Y, Hu X, et al. Rethinking the performance comparison between SNNS and ANNS [J]. Neural networks: the official journal of the International Neural Network Society, 2020, 121: 294-307.