big data and cognitive computing

MDPI

Article

Mandarin Recognition Based on Self-Attention Mechanism with Deep Convolutional Neural Network (DCNN)-Gated Recurrent Unit (GRU)

Xun Chen †, **Chengqi Wang** †, **Chao Hu** * and **Qin Wang**

School of Information and Communication Engineering, Hainan University, Haikou 570228, China; chenxun@hainanu.edu.cn (X.C.); json1145623@163.com (C.W.); 22220854000059@hainanu.edu.cn (Q.W.)

* Correspondence: huchao@csu.edu.cn

† These authors contributed equally to this work.

Abstract: Speech recognition technology is an important branch in the field of artificial intelligence, aiming to transform human speech into computer-readable text information. However, speech recognition technology still faces many challenges, such as noise interference, and accent and speech rate differences. An aim of this paper is to explore a deep learning-based speech recognition method to improve the accuracy and robustness of speech recognition. Firstly, this paper introduces the basic principles of speech recognition and existing mainstream technologies, and then focuses on the deep learning-based speech recognition method. Through comparative experiments, it is found that the self-attention mechanism performs best in speech recognition tasks. In order to further improve speech recognition performance, this paper proposes a deep learning model based on the self-attention mechanism with DCNN-GRU. The model realizes the dynamic attention to an input speech by introducing the self-attention mechanism in a neural network model instead of an RNN and with a deep convolutional neural network, which improves the robustness and recognition accuracy of this model. This experiment uses 170 h of Chinese dataset AISHELL-1. Compared with the deep convolutional neural network, the deep learning model based on the self-attention mechanism with DCNN-GRU accomplishes a reduction of at least 6% in CER. Compared with a bidirectional gated recurrent neural network, the deep learning model based on the self-attention mechanism with DCNN-GRU accomplishes a reduction of 0.7% in CER. And finally, this experiment is performed on a test set analyzed the influencing factors affecting the CER. The experimental results show that this model exhibits good performance in various noise environments and accent conditions.

Keywords: self-attention mechanism; CTC; gated circulation units

Citation: Chen, X.; Wang, C.; Hu, C.; Wang, Q. Mandarin Recognition Based on Self-Attention Mechanism with Deep Convolutional Neural Network (DCNN)-Gated Recurrent Unit (GRU). *Big Data Cogn. Comput.* **2024**, *8*, 195. https://doi.org/10.3390/bdcc8120195

Academic Editors: Salvador García López and Tzung-Pei Hong

Received: 22 July 2024
Revised: 5 November 2024
Accepted: 16 December 2024
Published: 18 December 2024

1. Introduction

Since humans first started creating and using machines, they have held to an ideal: to develop machines that can understand human language [1,2], follow commands, and thus enable seamless human–machine communication. With the continuous advancement of science and technology, the emergence of speech recognition technology has brought this dream closer to reality [3,4]. Speech recognition technology allows machines to recognize and understand speech signals, converting them into corresponding text or actionable commands [5,6].

Speech recognition is an interdisciplinary field that is gradually evolving into a key technology for human–computer interaction in the realm of information technology [3]. The combination of speech recognition and speech synthesis technologies allows people to move away from traditional keyboard input, enabling control through voice commands [7–10]. As a result, the application of speech technology has become a competitive and rapidly growing high-tech industry [2,6].

Large-scale research in speech recognition began in the 1970s, with substantial progress made in recognizing individual words [1,6,11]. However, since the 1990s, progress in speech recognition research has slowed. Despite this, significant advances have been made in the application and commercialization of speech recognition technology [12–14]. Since 2009, breakthroughs in deep learning research [15,16], combined with the accumulation of vast amounts of speech data, have propelled speech recognition technology forward [4,17]. Deep learning leverages pre-trained multi-layer neural networks to significantly improve the accuracy of acoustic models [6,12,18].

The main contribution of this paper is the proposal of an innovative deep learning algorithm, the composite deep neural network (DCNN-GRU-Self-Attention), designed to improve the accuracy and efficiency of speech recognition. This algorithm combines gated recurrent units, deep convolutional neural networks [19,20], and a self-attention mechanism. Composite deep neural networks optimize the processing power of traditional acoustic models in several ways, particularly by efficiently processing complex speech sequence data, effectively capturing long-term dependencies, and enhancing multi-scale feature recognition. This network framework not only improves the model's ability to handle complex speech sequences and capture long-term dependencies but also significantly enhances the quality of speech recognition through the self-attention mechanism.

In summary, our research makes the following major contributions.

First, the DCNN-GRU-Self-Attention model proposed in this paper innovatively addresses the limitations of traditional deep neural networks in processing complex speech sequence data by combining gated recurrent units, deep convolutional neural networks [21], and the self-attention mechanism. This combination not only demonstrates excellent multi-scale analysis capabilities but also significantly improves the accuracy and quality of speech recognition through precise feature extraction and the efficient utilization of key information points [11,22].

Second, DCNN-GRU-Self-Attention optimizes the extraction of speech data features in a simulated speech recognition environment by incorporating a replicated Google SpecAugment data enhancement algorithm [15]. The adoption of this augmented feature algorithm effectively increases the diversity of the training data through operations such as time masking, frequency masking, and time warping, enabling the model to perform exceptionally well in processing the speech data in Aishell-1 and significantly improving the accuracy and efficiency of speech recognition.

Third, beyond basic speech feature extraction, our DCNN-GRU-Self-Attention model also incorporates the Connectionist Temporal Classification algorithm [23,24]. CTC offers the advantage of eliminating the need for aligned labels, handling variable-length inputs and outputs, and supporting end-to-end training, which simplifies data processing and annotation while further optimizing speech recognition performance and enhancing the accuracy and robustness of recognition. Finally, this study compares the traditional convolutional neural network model and bidirectional gated recurrent unit model using the Aishell-1 dataset, fully demonstrating the superior performance of the DCNN-GRU-Self-Attention model.

In this paper, we first introduce the research background and the limitations of existing methods, followed by a detailed description of our proposed DCNN-GRU-Self-Attention model and its innovations, particularly in combination with the Connectionist Temporal Classification algorithm. Next, we present the experimental setup and analyze the results to verify the superiority of our model by comparing it with deep convolutional neural networks and bidirectional gated recurrent unit neural networks. Finally, we summarize the research contributions and propose directions for future research.

2. Model Structure and Algorithms

This section proposes an innovative deep learning model, the DCNN-GRU-Self-Attention model, which combines the Connectionist Temporal Classification algorithm, the self-attention mechanism, and DCNN-GRU to improve the accuracy and robustness of speech recognition [25]. The components of the model and their respective roles are described in detail, and the superiority of the model is verified through experiments.

2.1. Connection Timing Classification Algorithm

Connectionist Temporal Classification is an unsupervised learning method for processing sequence data [23]. It is an end-to-end algorithm that maps an input sequence to an output sequence. A core idea of CTC is to treat mapping an input sequence to an output sequence as an alignment process, establishing a correspondence between the input and output sequences. However, in practice, input and output sequence lengths are often inconsistent, requiring specific processing by CTC.

In the CTC algorithm, an input sequence and an output sequence do not need to be in one-to-one correspondence [23]; a CTC algorithm introduces a BLANK character as a blank character to be added to a set of characters such that $L^* = L\bigcup\{B\}$.

Suppose an input sequence $X = \{x_1, x_2, \ldots\ldots x_T\}$ has length T, and a label sequence $L = \{\psi_1, \psi_2, \ldots\ldots\psi_N\}$ has length N and $N < T$. Insert a blank character B such that $L^* = L\bigcup\{B\} = \{\psi_1, \psi_2, \ldots\ldots\psi_N\}\bigcup\{B_1, B_2, \ldots\ldots B_M\}$. The length is N^* and $N^* = T$. Then, an infinite number of labeled sequential probabilistic paths can be obtained according to an arithmetic mechanism of the CTC algorithm [5].

CTC LOSS is defined as the sum of probabilities of all paths. Assuming T moments, outputs of each moment are independent from each other, and a probability between single paths is $P(\pi|X)$:

$$P(\pi|X) = \Pi_{t=1}^{T}\varphi, \forall\varphi\epsilon L^{*T} \tag{1}$$

L^{*T} is a character set, and φ is a probability of a single character predicted by a single path at moment t. Then, a sum of all path probabilities is $P(Z|X)$:

$$P(Z|X) = \sum_{\pi\epsilon B(Z)^{-1}} P(\pi|X) \tag{2}$$

B^{-1} is a mapping function of all paths of Z. Then, the CTC LOSS function is as follows:

$$Ln(s) = -Ln\prod_{(X,Z)\epsilon S} P(Z|X) \tag{3}$$

$$Ln(s) = -\Sigma_{(X,Z)\epsilon S} LnP(Z|X) \tag{4}$$

$$Ln(s) = -\Sigma_{(X,Z)\epsilon S} Ln\Sigma_{\pi\epsilon B(Z)^{-1}} P(\pi|X) \tag{5}$$

$$Ln(s) = -\Sigma_{(X,Z)\epsilon S} Ln\Sigma_{\pi\epsilon B(Z)^{-1}} \Pi_{t=1}\varphi, \forall\pi\epsilon L^* \tag{6}$$

In a decoding phase, prefix search decoding is used to find the best path probability, an optimal solution, that is, $P_{MAX}(\pi|X)$ π is a certain path:

$$P_{MAX}(\pi|X) = argmaxP(\pi|X), \forall\pi\epsilon L^* \tag{7}$$

2.2. Self-Attention Mechanism

The self-attention mechanism is a crucial technique in deep learning, particularly in the field of natural language processing [12,16]. It enables the model to focus on different parts of an input sequence to better understand the input and generate an appropriate output [10]. A fundamental idea behind the self-attention mechanism is that, for a given word, this model considers all words in its context and calculates the relevance scores between these words and the target words. Each part of an input sequence is then weighted according to these scores to produce a representation of words.

A key advantage of the self-attention mechanism is its ability to capture long-range dependencies in an input sequence. By weighting the entire input sequence, it can effectively capture dependencies over longer distances within a sentence or text. This capability enables the self-attention mechanism's superior performance when dealing with complex linguistic structures and long sequences.

There are several implementations of a self-attention mechanism, with the most common being Scaled Dot-Product Attention and Multi-Head Attention. Scaled Dot-Product Attention computes a dot product between an input sequence and a query vector, then obtains a weight distribution through the Softmax function [25]. Multi-Head Attention divides an input sequence into multiple sub-sequences, applies multiple independent self-attention mechanisms to each sub-sequence, and then stitches these results together to form the final output.

The structure of the self-attention mechanism is shown in Figure 1. Define an input sequence $X = \{x_1,x_2,\ldots\ldots x_T\}$, $x_i \epsilon R^{d_{model}}$ and define three weight matrices W_q, W_k, W_v such that query$=X_i$*W_q, key $=X_i$*W_k, Value $= X_i$*W_v, where $W_q \epsilon R^{d_{model}*d_k}$, $W_k \epsilon R^{d_{model}*d_k}$, and $W_v \epsilon R^{d_{model}*d_v}$. Then, we can obtain an attention score calculation formula:

$$Score(X_i, X_j) = softmax\{q_i * q_j\} \tag{8}$$

In the eighth formula, we normalize it to obtain the ninth formula:

$$Score(X_i, X_j) = softmax\{(q_i * q_j)/\sqrt{d_k}\} \tag{9}$$

d_k is a dimension of an embedding vector. The final output is out:

$$Out = Score(X_i, X_j) * V \tag{10}$$

$$Out = LayerNorm(Out + X_i) \tag{11}$$

Similarly, in the FeedForword layer, define 2 weight matrices F_q and F_w. Obtain $F_1 = F_q * X_i$, $F_2 = F_1 * F_w$; finally, the output is $out = Layer_norm(F_2 + X_i)$.

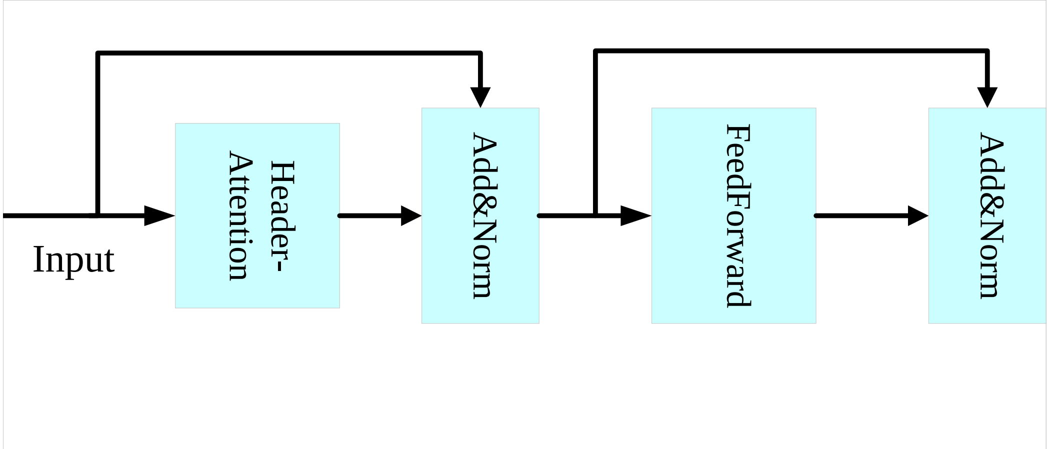

Figure 1. Structure diagram of the self-attention mechanism.

2.3. Self-Attention Mechanism with DCNN-GRU

As shown in Figure 2, the self-attention mechanism with DCNN-GRU model diagram consists of five parts: a convolutional neural network, a gated recurrent neural network, a self-attention mechanism layer, a fully connected layer, and a Softmax layer. Before the gated recurrent neural network, we use the convolutional neural network, which is primarily used to process the input data after they have been enhanced by the replicated Google SpecAugment data augmentation algorithm. The convolutional neural network layer employed is a two-dimensional convolutional layer, which serves to spatially localize input data through convolutional operations to extract local features.

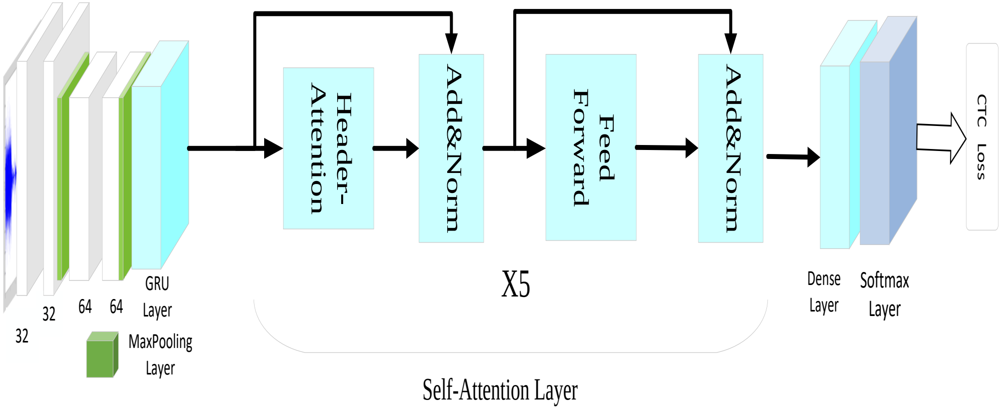

Figure 2. Self-attention mechanism with DCNN-GRU model diagram.

In this model structure, this convolutional neural network layer consists of two layers, each with 32 convolutional kernels, a BatchNormalization layer, and a MaxPooling2D layer. The third and fourth convolutional layers consist of one layer with 64 convolutional kernels, a BatchNormalization layer, and a MaxPooling2D layer. After this convolutional layer, there is a Reshape layer that transforms the input data into two dimensions, which are then fed into a gated recurrent network layer. The gated recurrent network layer has 128 output units.

The use of the gated recurrent neural network helps to better capture long-term dependencies in time series data, thereby improving this model's performance. This also addresses the vanishing gradient problem often encountered in recurrent neural networks, enabling this model to better handle long sequences of data.

Following this, there is a five-layer self-attention mechanism. The self-attention mechanism layer computes the correlations between positions in parallel, without relying on the sequential processing required by traditional recurrent neural networks. This allows this model to better handle long sequential data and capture global dependencies. It captures interdependent features over long distances in a sentence by calculating the weight of each word concerning all other words.

The last two layers are a fully connected layer and a Softmax layer. This fully connected layer connects all neurons of the previous layer to all neurons in the current layer, allowing the neural network to learn complex relationships and combinations of features. It maps the extracted "distributed feature representations" to a sample label space. This Softmax layer then extends the dimensionality of the output data from the fully connected layer to match the number of classes in the sample space and computes the probability of the samples of the output data using the Softmax function.

2.4. *Convolutional Neural Networks and Bidirectional Gated Recurrent Neural Networks*

Figure 3 represents the structure of the bidirectional gated recurrent neural network, which consists of 11 layers: 4 convolutional neural network layers, 2 bidirectional gated recurrent neural network layers, 3 fully connected neural network layers, and 1 Softmax layer. The first 2 convolutional neural network layers have 32 convolutional kernels, followed by 2 more convolutional neural network layers with 64 convolutional kernels.

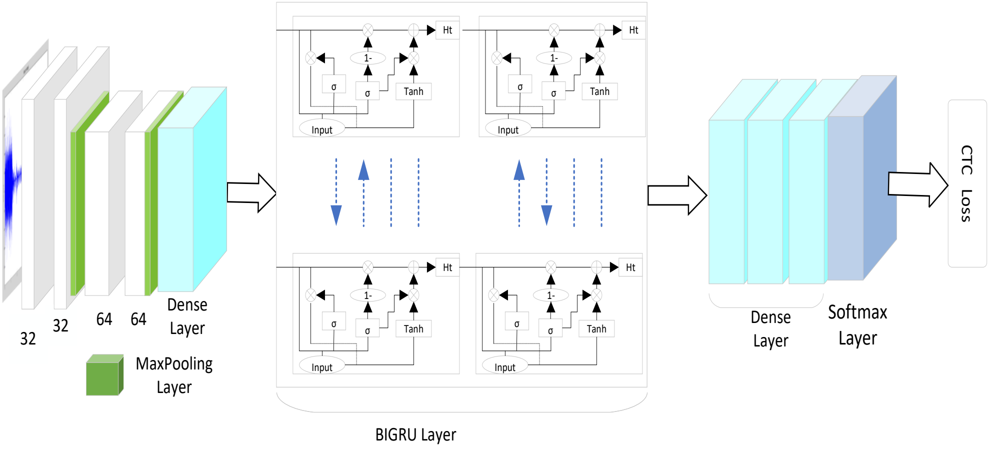

Figure 3. Deep convolutional neural network and two-way gated recurrent neural network.

After these convolutional layers, there is a fully connected neural network layer with 128 output units, followed by a 2-layer bidirectional gated recurrent neural network, a 3-layer fully connected neural network with 128 output units, and finally, a Softmax layer with 1,437 output units.

Figure 4 shows the structure of a deep convolutional neural network with 12 layers, including 10 convolutional neural network layers with the following number of convolutional kernels: 32, 32, 64, 64, 128, 128, 128, 128, 128, and 128. There is also a fully connected network layer with 128 output units, and a Softmax layer with output units corresponding to the number of characters in some sample data. The role of the Softmax layer is to calculate the probability of each character in the output units.

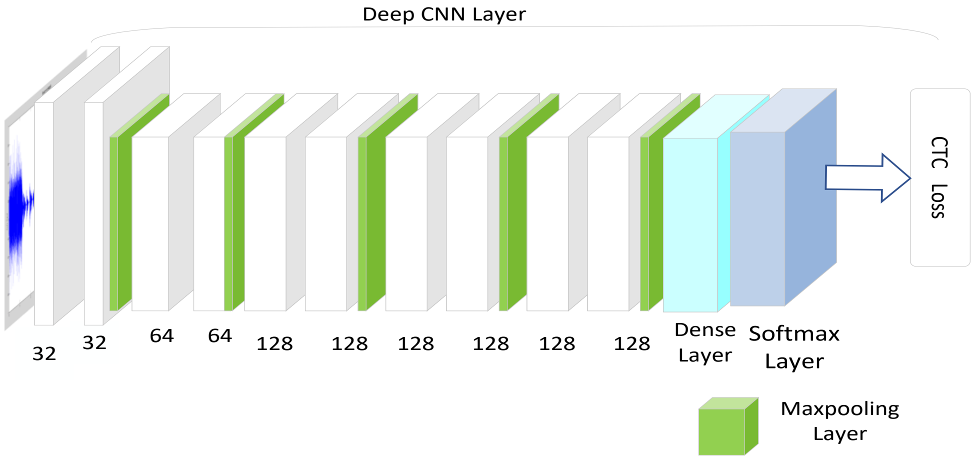

Figure 4. Deep convolutional neural network.

A bidirectional gated recurrent neural network combined with a deep convolutional neural network and a fully connected neural network is similar to a 12-layer convolutional neural network, except that the core components differ.

2.5. Chinese Pinyin

Chinese pinyin was chosen as the output for the experiment because pinyin and Chinese characters complement each other. Pinyin is a standardized phonetic system that helps people from different dialect areas communicate using a unified language. With pinyin, we can quickly pronounce Chinese characters, thus increasing reading speed. For beginners, pinyin can help them understand the meaning of Chinese characters faster. By learning pinyin, we can better memorize the pronunciation and spelling rules of Chinese characters, thus mastering them more effectively. In the model structure used in the experiment, the number of pinyin is 1437 while the number of Chinese characters is larger, indicating that many Chinese characters share the same pinyin. Choosing a smaller set of pinyin samples as the model output helps to reduce training time and improve recognition efficiency. For different model structures, it is critical to quickly recognize Chinese characters that share the same pinyin.

3. Related Work

In the field of neural networks, self-attention mechanisms and gated logic units are two key techniques that have been widely studied and applied. The self-attention mechanism allows models to dynamically adjust attention weights according to correlations at different locations in an input sequence, thus better capturing dependencies within the sequence. Convolutional neural networks process data through local connectivity, capturing local features in the input data. Gated logic units, on the other hand, control the flow of information through gating mechanisms, helping the network to better handle long-distance dependencies and the transfer of sequence information.

Researchers have begun to explore ways to combine self-attention mechanisms with DCNN-GRU to achieve better performance and generalization capabilities in the design of neural networks. The goal of this combination is to leverage the self-attention mechanism while further regulating the flow of information through DCNN-GRU, thereby enhancing the modeling and characterization capabilities of the model.

Research has shown that models combining self-attention mechanisms with DCNN-GRU have achieved success in various natural language processing tasks. For example, in machine translation tasks, models that integrate the self-attention mechanism with DCNN-GRU can better capture semantic relationships between source and target languages, improving translation quality and fluency. In text categorization and sentiment analysis tasks, this combined model can more accurately capture critical information and sentiment tendencies in the text, thereby enhancing categorization and sentiment analysis performance.

Although progress has been made, research on combining self-attention mechanisms with DCNN-GRU still faces challenges and unresolved issues. For instance, effectively designing the model structure, adjusting model hyperparameters, and addressing problems like gradient vanishing and gradient explosion require further exploration and research.

Therefore, future work will continue to explore the combination of self-attention mechanisms with DCNN-GRU and address related challenges to provide more effective and reliable solutions for applying neural networks in natural language processing and other fields. Through further research and experimentation, we aim to improve the performance and generalization ability of this model and advance the development and application of neural network technology in practical contexts.

4. Experimental Steps

4.1. Dataset

The AISHELL-1 dataset is a valuable resource in the field of speech recognition and speech processing, particularly for Mandarin Chinese. Developed by the Institute of Automation of the Chinese Academy of Sciences, it provides over 178 h of high-quality speech data, recorded by 400 participants from diverse backgrounds. Each participant contributed around 300 sentences, covering a wide array of topics, ensuring that this dataset is both diverse and generalizable.

Recordings of AISHELL-1 dataset were conducted in a quiet room environment, using three different devices to mimic various real-world recording conditions. This approach enhances its applicability to real-world scenarios, making it particularly useful for developing and testing Mandarin Chinese speech recognition systems.

4.2. Model Parameters

The convolutional neural networks used for all three sets of experimental model structures have convolutional kernels of 32, 32, 64, and 64. The step size of these networks is 1 × 1, and all pooling layers have a pool size of 1. The number of output units for the fully connected network is 1437, and the ReLU function is used as its activation function.

For the self-attention mechanism, three weight matrices are employed with three fully connected neural networks, and the number of output units is 128. The dropout is set to 0.2. The dimension of the hidden layer in this feed-forward layer is 128.

4.3. Training and Decoding

The experiments were performed using a replicated Google SpecAugment data enhancement algorithm on audio data, with a time window of 25 ms and various masking of the spectrograms—15% masking of a frequency range horizontally and 15% masking of a period vertically. Half of the extracted frequency feature data were used, as both sides are symmetric. The input data were three-dimensional to store more information about the data. All experiments were conducted using the TensorFlow 2 framework, with the NVIDIA AX5000 GPU. We used the ADAM optimizer with a learning rate ranging between 0.001 and 0.0000001 to train on this dataset. Additionally, we employed a dropout layer and a batch normalization layer to improve the performance of this model structure in experiments involving network models with self-attention mechanisms and DCNN-GRU. The batch normalization layer was used in deep convolutional neural networks and bidirectional gated recurrent neural networks to enhance the performance of the model structure.

The prefix search decoding algorithm is used to decode output data from three sets of neural networks. This algorithm is based on a prefix tree data structure, which leverages the properties of prefixes to enable quick search and decoding, making it suitable for processing large-scale sequential data. The prefix search decoding algorithm can accurately find the longest matching prefix sequence, thus ensuring accuracy in decoding. To obtain information on Chinese characters, the experiment also incorporates a 2-Gram model, which uses the 2-Gram algorithm primarily to convert pinyin into Chinese characters.

5. Analysis of Results

We conducted experiments to compare the efficiency and accuracy of three models on Chinese datasets. Figures 5 and 6 show the training times of the three models for each step and round of the training set. The bar graphs reveal that as the number of neural network layers increases, the training time for deep convolutional neural networks is significantly shorter than that of bidirectional gated recurrent neural networks. Notably, the training time for self-attention mechanisms with DCNN-GRU is shorter than that of Bi-GRUs.

The reason for this difference is that CNNs typically handle image data and may not require sequence-to-time-step conversion like RNNs do. Due to their structural characteristics, CNNs can speed up the training process. In contrast, Bi-GRUs may contain more parameters due to their bidirectional structure, resulting in longer training times. Self-attention mechanisms with DCNN-GRU networks, in general, are more complex than Bi-GRUs. The self-attention mechanism within these networks captures relationships between different positions in an input sequence, making them better at handling long-distance dependencies. As a result, the training time for self-attention mechanisms with DCNN-GRU networks is usually a bit shorter than that of Bi-GRUs.

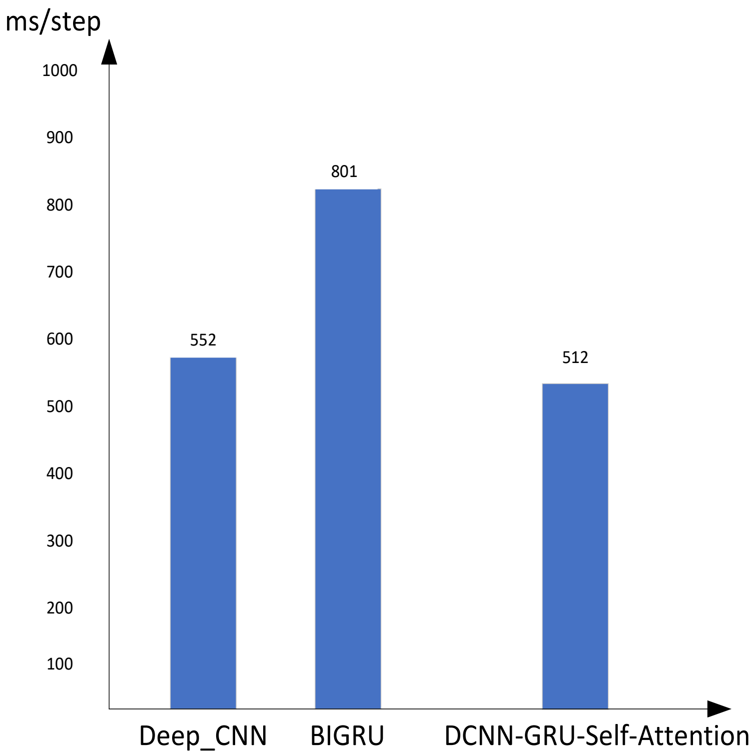

Figure 5. Training time per step. The figure shows the training time for each step of three models on the training set.

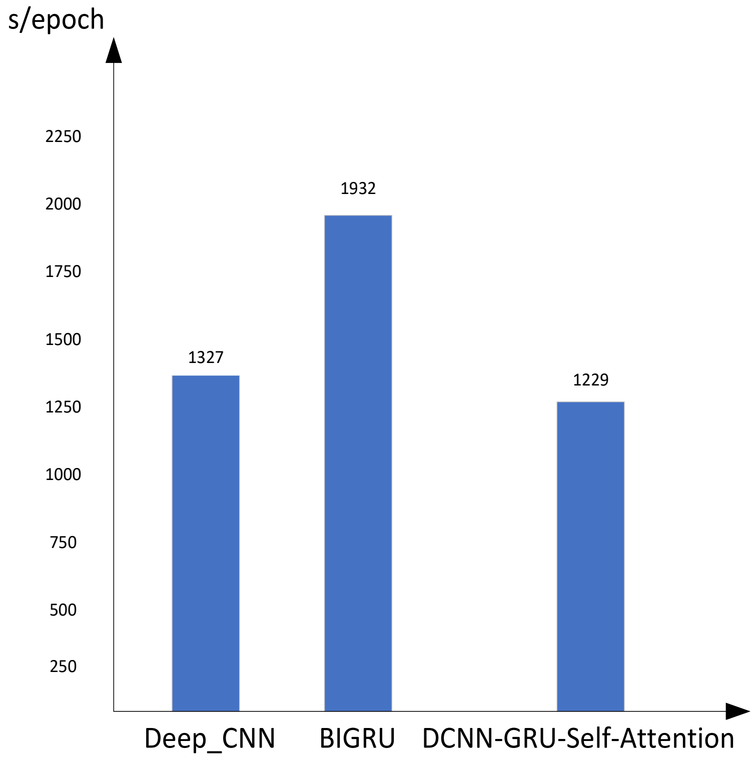

Figure 6. Training time per round. The figure shows the training time for each round of three models on the training set.

The time for Bi-GRUs is about 1.45 times that of DCNNs, both in terms of the time per step and time per round. The time for the self-attention mechanisms with DCNN-GRU networks is slightly shorter than that of Bi-GRUs, and about 0.92 times that of DCNNs, in both the time per step and time per round. This suggests that while DCNNs process data quickly, they may struggle with contextual information and are more susceptible to local minima, leading to an unstable training process.

Bidirectional gated recurrent neural networks are capable of capturing long-term dependencies in input sequences. Being able to process sequential data, they can automatically learn historical information and use it to predict future outcomes. However, they are prone to gradient vanishing or explosion problems, leading to longer training times and higher computational complexity. Self-attention mechanisms with DCNN-GRU networks, by capturing relationships between different positions in an input sequence, are better suited to handling long-distance dependencies.

5.1. Analysis on the Pinyin Sequence

This CER is obtained by calculating a substitution error rate, an insertion error rate, and a deletion error rate of the model on the test set. The former is a sum of the latter three calculations. Table 1 shows the values of three neural network models on CER, where it is evident that the substitution error rate is the most dominant part of CER and also varies the most between each model. This suggests that the similar pinyin part within the same neural network model could be a deficiency of the model. For the AISHELL-1 dataset, when speakers pronounce Chinese, some words share the same pinyin, but some pinyin has tones while others do not, which is a significant factor influencing CER.

Table 1. The table shows pinyin error rates on three neural network models, denoted by CER, where S stands for substitution error rate, I for insertion error rate, and D for deletion error rate.

Model	CER	S	I	D
BIGRU	15.815%	15.081%	0.407%	0.338%
Deep_CNN	20.694%	19.683%	0.366%	0.642%
DCNN-GRU-Self-Attention	15.114%	14.537%	0.228%	0.294%

For Table 2, the substitution error rate of deep convolutional neural network is 19.683% in CER and 33.002% in WER. This substitution error rate of the bidirectional gated recurrent neural network is 15.081% in CER and 27.85% in WER. This substitution error rate for the self-attentive mechanism with the DCNN-GRU neural network is 14.537% in CER and 27.621% in WER.

Table 2. The table shows the difference between the substitution error rate for the pinyin error rate and the substitution error rate for the word error rate on three neural network models, denoted by S (substitution error rate), CER (pinyin error rate), and WER (word error rate).

Model	S (CER)	S (WER)	Difference
BIGRU	15.081%	27.85%	12.769%
Deep_CNN	19.683%	33.002%	13.319%
DCNN-GRU-Self-Attention	14.537%	27.621%	13.084%

For Table 3, the deep convolutional neural network has an insertion error rate of 0.366% in CER and 0.295% in WER. This insertion error rate of the bidirectional gated recurrent neural network is 0.407% in CER and 0.297% in WER. The insertion error rate for the self-attentive mechanism with the DCNN-GRU neural network is 0.288% in CER and 0.202% in WER.

Table 3. The table shows the difference between the insertion error rate for the pinyin error rate and the insertion error rate of the word error rate on three neural network models, denoted by I (insertion error rate), CER (pinyin error rate), and WER (word error rate).

Model	I (CER)	I (WER)	Difference
BIGRU	0.407%	0.297%	0.11%
Deep_CNN	0.366%	0.295%	0.071%
DCNN-GRU-Self-Attention	0.288%	0.202%	0.086%

For Table 4, the deep convolutional neural network has a deletion error rate of 0.642% in CER and an insertion error rate of 0.431% in WER. This bidirectional gated recurrent neural network has a deletion error rate of 0.338% in CER and 0.23% in WER. The self-attentive mechanism with DCNN-GRU neural network has a deletion error rate of 0.294% in CER and 0.214% in WER.

Table 4. The table shows the difference between the deletion error rate of the pinyin error rate and the deletion error rate of the word error rate on three neural network models, denoted by D (deletion error rate), CER (pinyin error rate), and WER (word error rate).

Model	D (CER)	D (WER)	Difference
BIGRU	0.338%	0.230%	0.108%
Deep_CNN	0.642%	0.431%	0.211%
DCNN-GRU-Self-Attention	0.294%	0.214%	0.08%

For Table 5, this CER of the deep convolutional neural network is 20.694%, while the CER of the bidirectional gated recurrent neural network is 15.815%, which indicates that the accuracy of the pinyin sequence of the bidirectional gated recurrent neural network is higher than that of the deep convolutional neural network.

Table 5. The table shows the difference between pinyin error rates on a bidirectional gated recurrent neural network model and a deep convolutional neural network model, where CER stands for the pinyin error rate.

Index	BIGRU	Deep_CNN	Difference
CER	15.815%	20.694%	4.879%

For Table 6, this CER of the two-way gated recurrent neural network is 15.815%, while the CER of the self-attention mechanism with the DCNN-GRU neural network is 15.114%, which indicates that the accuracy of the pinyin sequence of the self-attention mechanism with the DCNN-GRU neural network is higher than that of this two-way gated recurrent neural network.

Table 6. The table shows the difference in the pinyin error rates, represented by CER, between a bidirectional gated recurrent neural network model and a self-attentive mechanism with the DCNN-GRU neural network model.

Index	BIGRU	DCNN-GRU-Self-Attention	Difference
CER	15.815%	15.114%	0.707%

For Table 7, this CER of the deep convolutional neural network is 20.694%, while the CER of the self-attention mechanism with the DCNN-GRU neural network is 15.114%, which indicates that the accuracy of the pinyin sequence of the self-attention mechanism with the DCNN-GRU neural network is higher than that of the deep convolutional neural network.

Table 7. The table shows the difference in pinyin error rates, represented by CER, between a deep convolutional neural network model and a self-attention mechanism with the DCNN-GRU neural network model.

Index	Deep_CNN	DCNN-GRU-Self-Attention	Difference
CER	20.694%	15.114%	5.58%

5.2. Analysis on the FLOPs

The DCNN-GRU-Self-Attention model boasts approximately 24.54 billion FLOPs, incorporating various deep learning architectures. In comparison, its overall computational expense is less than that of the DCNN model but slightly exceeds the BIGRU model. This underscores that integrating the self-attention mechanism has not drastically escalated the computational burden, while imparting greater expressive capability, thereby enhancing the model's capacity to capture temporal dynamics and feature correlations.

Conversely, the DCNN model, with approximately 47.69 billion FLOPs, incurs double the computational cost of the DCNN-GRU-Self-Attention model. This highlights the substantial computational demands of the DCNN model, particularly when dealing with numerous convolutional kernels or extensive feature maps, both of which significantly amplify the computational overhead. While the DCNN excels in feature extraction, it incurs a significant computational expenditure.

The BIGRU model, with approximately 24.22 billion FLOPs, aligns closely with the DCNN-GRU-Self-Attention model and falls below the DCNN model in terms of computational expense. This suggests that the GRU structure is less resource intensive in processing sequential data, making it well suited for constrained environments. Despite its modest computational cost, BIGRU remains effective in modeling long-term sequences, offering a commendable balance between computational efficiency and performance.

6. Conclusions

In this experiment, we introduced a neural network model based on the self-attention mechanism with DCNN-GRU applied to a Chinese dataset. The results, derived from 170 h of the AISHELL-1 Chinese dataset, show that our self-attention mechanism with DCNN-GRU model outperforms a bidirectional gated recurrent neural network model and a deep convolutional neural network model. The predictions of this model output are based on CER, highlighting the importance of capturing contextual information, especially in the context of Chinese speech recognition. These experiments confirm that the self-attention mechanism with DCNN-GRU model exhibits superior performance with the CTC algorithm in Chinese speech recognition. It is worthwhile to further study and explore deeper neural network models to achieve even better results in Chinese speech recognition.

Author Contributions: Conceptualization, C.W.; methodology, Q.W.; software, C.W.; data curation, C.W.; writing—original draft preparation, C.W.; supervision, X.C.; funding acquisition, X.C. and C.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by Hainan University under Grant (Project No: KYQD(ZR) 21014), the National Key Research and Development Program of China (Project No: 2021YFC3340800), the National Natural Science Foundation of China (Project No: 62177046), and the High-Performance Computing Center of Central South University (HPC).

Data Availability Statement: Dataset: https://opendatalab.com/OpenDataLab (accessed on 20 February 2024).

Acknowledgments: The author expresses sincere gratitude to Chen for his valuable support in accessing the High-Performance Computing Center of Central South University (HPC) resources. The author deeply appreciates his generous contribution.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Aguiar de Lima, T.; Da Costa-Abreu, M. A survey on automatic speech recognition systems for Portuguese language and its variations. *Comput. Speech Lang.* **2020**, *62*, 101055. [CrossRef]
2. Wei, D.; Hong, L. Chinese Speech Recognition Technology Based on Neural Network. *J. Sichuan Norm. Univ. (Nat. Sci. Ed.)* **2022**, *45*, 131–135.
3. Meng, J.; Zhang, J.; Zhao, H. Overview of the Speech Recognition Technology. In Proceedings of the 2012 Fourth International Conference on Computational and Information Sciences, Chongqing, China, 17–19 August 2012; pp. 199–202. [CrossRef]
4. Dargan, S.; Kumar, M.; Ayyagari, M.R.; Kumar, G. A Survey of Deep Learning and Its Ap-plications: A New Paradigm to Machine Learning. *Arch. Comput. Methods Eng.* **2020**, *27*, 1071–1092. [CrossRef]
5. Gaikwad, S.K.; Gawali, B.W.; Yannawar, P. A review on speech recognition technique. *Int. J. Comput. Appl.* **2010**, *10*, 16–24. [CrossRef]
6. Wang, D.; Wang, X.; Lv, S. An overview of end-to-end automatic speech recognition. *Symmetry* **2019**, *11*, 1018. [CrossRef]
7. Bird, J.J.; Wanner, E.; Ekárt, A.; Faria, D.R. Optimisation of phonetic aware speech recogni-tion through multi-objective evolutionary algorithms. *Expert Syst. Appl.* **2020**, *153*, 113402. [CrossRef]
8. Yu, L.F.; Liu, Q. Research and application of deep recurrent neural networks based voiceprint recognition. *J. Appl. Res. Comput.* **2019**, *36*, 153–158.
9. Goh, Y.H.; Lau, K.X.; Lee, Y.K. Audio visual speech recognition system using recurrent neural network. In Proceedings of the 2019 4th International Conference on Information Technology (InCIT), Bangkok, Thailand, 24–25 October 2019; pp. 38–43. [CrossRef]
10. Zhou, P.; Yang, W.; Chen, W.; Wang, Y.; Jia, J. Modality attention for end-to-end au-dio-visual speech recognition. In Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 6565–6569. [CrossRef]
11. Xie, X.; Liu, X.; Lee, T.; Hu, S.; Wang, L. BLHUC: Bayesian learning of hidden unit con-tributions for deep neural network speaker adaptation. In Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 5711–5715.
12. Li J. Recent advances in end-to-end automatic speech recognition. *Apsipa Trans. Signal Inf. Process.* **2022**, *11*, e8. [CrossRef]
13. Gonzalez-Dominguez, J.; Eustis, D.; Lopez-Moreno, I.; Senior, A.; Beaufays, F.; Moreno, P.J. A real-time end-to-end multilingual speech recognition architecture. *IEEE J. Sel. Top. Signal Process.* **2014**, *9*, 749–759. [CrossRef]
14. Hu, S.; Lam, M.W.; Xie, X.; Liu, S.; Yu, J.; Wu, X.; Liu, X.; Meng, H. Bayesian and Gaussian process neural networks for large vocabulary continuous speech recognition. In Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 6555–6559.
15. Joy, N.; Oglic, D.; Cvetkovic, Z.; Bell, P.; Renals, S. Deep scattering power spectrum features for robust speech recognition. In Proceedings of the International Speech Communication Association, Virtual, 25–29 October 2020; pp. 1673–1677.
16. Pham, N.Q.; Nguyen, T.S.; Niehues, J.; Müller, M.; Stüker, S.; Waibel, A. Very Deep Self-Attention Networks for End-to-End Speech Recognition. *arXiv* **2019**, arXiv:1904.13377.
17. Bai, K.; An, Q.; Liu, L.; Yi, Y. A training-efficient hybrid-structured deep neural network with reconfigurable memristive synapses. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2020**, *28*, 62–75. [CrossRef]
18. Shashidhar, R.; Patilkulkarni, S.; Puneeth, S.B. Combining audio and visual speech recog-nition using LSTM and deep convolutional neural network. *Int. J. Inf. Tecnol.* **2022**, *14*, 3425–3436. [CrossRef]
19. Oglic, D.; Cvetkovic, Z.; Bell, P.; Renals, S. A deep 2D convolutional network for wave-form-based speech recognition. In Proceedings of the International Speech Communication Association, Virtual, 25–29 October 2020; pp. 1654–1658.
20. Loweimi, E.; Bell, P.; Renals, S. On learning interpretable CNNs with parametric modulated Kernel-based filters. In Proceedings of the International Speech Communication Association, Graz, Austria, 15–19 September 2019; pp. 3480–3484.
21. Yakoub, M.S.; Selouani, S.A.; Zaidi, B.F.; Bouch, A. Improving dysarthric speech recogni-tion using empirical mode decomposition and convolutional neural networks. *EURASIP J. Audio Speech Music Process.* **2020**, *2020*, 1. [CrossRef]
22. Zaidi, B.F.; Selouani, S.A.; Boudraa, M.; Yakoub, M.S. Deep neural network architectures for dysarthric speech analysis and recognition. *Neural Comput. Appl.* **2021**, *33*, 9089–9108. [CrossRef]
23. Watanabe, S.; Hori, T.; Kim, S.; Hershey, J.R.; Hayashi, T. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE J. Sel. Top. Signal Process.* **2017**, *11*, 1240–1253. [CrossRef]
24. Martinez, B.; Ma, P.; Petridis, S.; Pantic, M. Lipreading using temporal convolutional net-works. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6319–6323.
25. Zhu, T.; Cheng, C. Joint CTC-Attention End-to-End Speech Recognition with a Triangle Recurrent Neural Network Encoder. *J. Shanghai Jiaotong Univ. (Sci.)* **2020**, *25*, 70–75. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.