Projected Minimal Gated Recurrent Unit for Speech Recognition

RENJIAN FENG, (Member, IEEE), WEIJIE JIANG, NING YU, (Member, IEEE), YINFENG WU, (Member, IEEE), AND JIAXUAN YAN

School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing 100191, China

Corresponding author: Ning Yu (nyu@buaa.edu.cn)

This work was supported in part by the Defense Industrial Technology Development Program under Grant JSJL2018205A002, in part by the National Natural Science Foundation of China under Grant 61671039, and in part by the National Key Research and Development Program of China under Grant 2017YFF0204905.

ABSTRACT
Recurrent neural network (RNN) has the ability to learn long-term dependencies, which makes it suitable for acoustic modeling in speech recognition. In this paper, we revise RNN model used in acoustic modeling, namely, mGRUIP with Context module (mGRUIP-Ctx), and propose an advanced model which named Projected minimal Gated Recurrent Unit (PmGRU). The paper demonstrates two major contributions: firstly, in the case that adding context information to context module in mGRUIP-Ctx will bring about large amount of parameter, we propose to insert a smaller output projection layer after the mGRUIP-Ctx cell’s output to form the PmGRU, which is inspired by the idea of low-rank decomposition of matrix. The output projection layer has been proved to be able to save most of the effective information with the reduction of model parameters. Secondly, in the case that too much context information of previous layer introduced by context module will cause declining of model performance, we adjust the ratio of context information of the previous layer to the current layer by moving the position of batch normalization layer, and the final RNN model Normalization Projected minimal Gated Recurrent Unit (Norm-PmGRU) is generated. In the five automatic speech recognition (ASR) tasks, the Norm-PmGRU has been proved more effectively in the experiments compared with mGRUIP-Ctx, TDNN-OPGRU, TDNN-LSTMP and other RNN baseline acoustics models.

INDEX TERMS
Speech recognition, acoustic model, gated recurrent unit, batch normalization.

I. INTRODUCTION
In recent years, deep learning has been widely used in various fields, such as pattern recognition [1], [2], system control [3]–[5], and combination optimization [6], [7]. In the field of speech recognition, the hybrid model which relies on a combination of deep neural network (DNN) and Hidden Markov Model (HMM) still maintain state-of-the-art performance compared to End-to-End model [8]–[12]. DNN in DNN-HMM hybrid model, specifically feed-forward neural network (FFNN) and recurrent neural network (RNN), is used for acoustic modeling to predict phone-level target phonemes.

Time delay neural network (TDNN) [13]–[15] and feed-forward sequential memory network (FSMN) [16], [17] are two popular FFNN architectures which improve the performance of speech recognition by modeling the long temporal contexts. For contextual modeling in TDNN, all neurons at each layer receives input from a contextual window, which is composed of multiple outputs of the layer below. As for the acquisition of context information, FSMN is equipped with some memory blocks. These fixed-size memory blocks are obtained by FSMN using taped delay line structure to encode long-term context information.

Unlike FFNN only using the output of the previous layer as input, RNN uses the previous time step and the previous layer as input to determine the forward activation of the current time step. Thus, considering the ability to model a dynamic window of all sequence history instead of a fixed contextual window over the input sequence as FFNN does, RNN is more suitable for sequence modeling. However, as problems of vanishing gradient and exploding gradient highly happen in the vanilla RNN training process, Long Short-Term Memory (LSTM) [18], Gated Recurrent Unit (GRU) [19] and their Long Short-Term Memory with Projection (LSTMP) [20], Bidirectional Long Short-Term Memory with Projection (BLSTMP) [21], Output-Gate Projected Gated Recurrent Unit (OPGRU) [22], minimal Gated Recurrent Unit (mGRU, also named LiGRU) [23], [24], mGRUIP
with Context module (mGRUIP-Ctx) [25], are proposed to solve the above problems and well used in speech recognition field. In [21], the author compared the performance of TDNN-LSTMP and BLSTMP. In the LSTM, input gate, output gate, and forget gate are used to control the flow of information. While in the mGRU [23], [24], only the update gate are used. As a simplified version of GRU, the reset gate of mGRU is removed. It has shorter training time and higher accuracy than GRU. But mGRU does not have the ability to model the future context. In [25], authors insert an input linear projection layer into mGRU to compress the input vector and the hidden state vector, and adds context information which from the layer below into the projection layer to realize mGRUIP-Ctx. Due to the introduction of context information, mGRUIP-Ctx gets better performance than LSTM and GRU. The introduction of context information improves the performance of the model. Unfortunately, due to the large dimension of mGRUIP-Ctx output (in [25], the dimension of mGRUIP-Ctx output is 2560), when the length of context window is increased, the parameters in the model will increase significantly.

Low-rank factorization is a method of using matrix/tensor decomposition to estimate the effective information in matrix/tensor. If a weight matrix is low-rank, we can use the decomposition method to express the matrix as two small matrices, which will significantly reduce the number of parameters. This method is widely used in neural networks to reduce the number of parameters. In [16], [26], [27], it is used in FFNN such as full connection network, TDNN and FSMN. In [21], [28]–[30], it is used in RNN networks such as LSTM and GRU. These experiments keep the original network performance as much as possible while reducing the number of network parameters.

Batch normalization (BN) [31] is a optimization method that normalizes the variance and the mean of each layer’s pre-activation in each training mini-batch. It can be applied to different places in the RNN. In [32], the authors suggest that the best way to apply batch normalization in mGRUIP-Ctx as follows: For the candidate state, BN is used for input-to-hidden (ItoH) and hidden-to-hidden (HtoH) connections. For the update gate, BN is only used for ItoH. In [33], the authors suggest to apply BN to ItoH only for the candidate state. In [24], the authors apply BN to ItoH both for the candidate state and the update gate.

In this work, in order to increase the length of the context window without significantly increasing the parameters of the model, we propose a new architecture, which is named Projected minimal Gate Recurrent Unit (PmGRU). The proposed PmGRU is inspired by the decomposition of the low rank weight matrix in the previous work LSTM [21] and cFSMN [16], we insert a separate smaller linear projection layer named as output projection layer after mGRUIP-Ctx cell’s output. We believe that the linear projection layer can hold most of the valid information from the mGRUIP-Ctx cell’s output. To further improve the performance of the PmGRU, we apply BN to both ItoH and HtoH for the input projection vector to form the final model Norm-PmGRU.

We evaluated our proposed model on five ASR tasks. Result shows that Norm-PmGRU is more effective compared with mGRUIP-Ctx, TDNN-OPGRU, TDNN-LSTMP and other baseline models, which not only has the almost same accuracy of recognition or even better, but also uses the least number of parameters.

The rest of the paper is organized as follows. Section 2 presents the model architecture of mGRU and its variants, including mGRUIP-Ctx, the proposed PmGRU and Norm- PmGRU. Section 3 presents experimental setup. Section 4 presents experimental and analysis of Results. Finally, the conclusion is given in Section 5.

II. PROPOSED MODEL ARCHITECTURES
In this section, we will first make an introduction to the model structure of minimal Gated Recurrent Unit (mGRU) and mGRUIP with Context module (mGRUIP-Ctx). Then we will make each dimension have the mean of zero and the variance of 1.

\[ E[x^{(k)}] = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}} \]  

where \( x^{(k)} \) represents the \( k \)-th dimension of \( x \). The expectation \( E[x^{(k)}] \) and variance \( Var[x^{(k)}] \) are computed over the
training batch. In this article, the batch size is 128. That is, we will normalize each dimension of all 128 features.

**B. mGRUIP-CTX**

The mGRUIP with Context module (mGRUIP-Ctx) [25] [32] is obtained by inserting a linear input projection layer into mGRU. And a context module called temporal convolution is designed in input projection layer to effectively model the history and future context. The model is defined by the following equations:

\[
\begin{align*}
\text{1. } v'_{1l} &= W_{vl}^{l} \tilde{x}_{l}^{t} \\
\text{2. } v'_{2l} &= W_{vl}^{l} h'_{l-1} \\
\text{3. } h'_{l} &= v'_{1l} + v'_{2l} \\
\text{4. } z'_{l} &= \sigma \left( BN(W_{cv}^{l} v'_{1l}) + W_{cv}^{l} v'_{2l} \right) \\
\text{5. } \tilde{h}'_{l} &= \text{ReLU} \left( BN(W_{hv}^{l} h'_{l}) \right) \\
\text{6. } h'_{l} &= z'_{l} \odot h'_{l-1} + (1 - z'_{l}) \odot \tilde{h}'_{l}
\end{align*}
\]

where \(v'_{l}\) is a projection vector, it is calculated by adding the input vector \(\tilde{x}_{l}^{t}\) and the previous output state vector \(h'_{l-1}\) compressed into the lower dimensional space by weight matrices \(W_{vl}^{l}\) and \(W_{vl}^{l}\), respectively. \(v'_{1l}\) and \(v'_{2l}\) are the projected vector from \(\tilde{x}_{l}^{t}\) and \(h'_{l-1}\). \(z'_{l}\) is the update gate, of which the activation function is logistic sigmoid \(\sigma(\cdot)\). The output state vector \(h'_{l}\) is also the output vector of model.

\(\tilde{x}_{l}^{t}\) is the concatenation of the current input vector \(x_{l}^{t}\) and the output state vector of preceding layer of both the future frames and history contexts:

\[
\tilde{x}_{l}^{t} = \left[ h_{l-1}^{t-1} \times K_1; \ldots; h_{l-1}^{t-1} \times i; x_{l}^{t}; h_{l+2}^{t-1} \times j; \ldots; h_{l+2}^{t-1} \times K_2 \right]
\]

In particular, \(x_{l}^{t}\) is the input vector of layer \(l\), and \(h_{l+1}^{t-1}\) is the output state vector of layer \(l+1\) on the \((t + s \times i)\)th frame. \(S_1(S_1 \geq 1)\) and \(S_2(S_2 \geq 1)\) are the step size for the history and future frames, respectively. \(K_1(K_1 \geq 1)\) is the order of history information and \(K_2(K_2 \geq 1)\) is the order of future information. \(i(1 \leq i \leq K_1)\) and \(j(1 \leq j \leq K_2)\) are the loop index.

The mGRUIP-Ctx architecture is shown in Figure 1. In this diagram, each line carries an entire vector, from the output of one node to the inputs of others. Lines merging denotes concatenation, and lines forking means that the contents are being copied and the copies are going to different destinations. The circles represent pointwise operations, like vector element-wise product, and the solid boxes represent the network layers that need to be learned.

**C. PMGRU**

The introduction of context information always leads to an improvement in model performance. We assume that the number of hidden neurons of the model is \(\text{dim}(\text{cell})\), and the dimension of the input project layer of the model is \(\text{dim}(ip)\). The expression \(v'_{l}\) and \(\tilde{x}_{l}^{t}\) show that if the number of contexts contained in the context module is \(n\), the size of context module for mGRUIP-Ctx can be calculated as follow:

\[
N_{mGRUIP-Ctx} = n \times \text{dim}(\text{cell}) \times \text{dim}(ip)
\]

In order to reduce the increment of the model size which is caused by the increasing of contexts, an output projection layer is added to mGRUIP-Ctx cell’s output. The new model is called Projected minimal Gated Recurrent Unit (PmGRU). It reduces the final output to a lower dimension. Assuming that the dimension of the output project layer of the model is \(\text{dim}(op)\), the size of context module for PmGRU can be calculated as follow:

\[
N_{PmGRU} = n \times \text{dim}(ip) \times \text{dim}(op) + \text{dim}(\text{cell}) \times \text{dim}(op)
\]

Assuming \(\text{dim}(op)\) equal with \(\text{dim}(ip)\), the difference between \(N_{mGRUIP-Ctx}\) and \(N_{PmGRU}\) is:

\[
N = \text{dim}(ip) \times ((n - 1) \times \text{dim}(cell) - n \times \text{dim}(ip))
\]

Compared to the expression of mGRUIP-Ctx, PmGRU adds an equation:

\[
y'_{l} = W_{y} h'_{l}
\]

And the equation (12) now becomes:

\[
\tilde{x}_{l}^{t} = \left[ y'_{l-1}^{t-1} \times K_1; \ldots; y'_{l-s+1}^{t-1} \times i; y'_{l-s+2}^{t-1} \times j; \ldots; y'_{l+s-1}^{t-1} \times K_2 \right]
\]

\(y'_{l}\) is the output of \(h'_{l}\) through output projection layer. \(\tilde{x}_{l}^{t}\) is the concatenation of the output vector \(y'_{l-1}^{t-1}\) of preceding layer.

Figure 2 is an expansion diagram of PmGRU along the time axis. The superscript of letters represents the number of layers where PmGRU units are located, and the subscript of letters represents the time point where PmGRU units are located. The dashed arrow in the graph represents context module, which extracts context information from the previous layer and adds it to the affine transformation of \(h'_{l-1}\) to get the input projection vector \(v'_{l}\).

**D. APPLYING BATCH NORMALIZATION IN PMGRU**

In the architecture of PmGRU, its context information comes from two parts. The first part is the history context information from this layer (the previous hidden state vector \(h'_{l-1}\)), it’s called HtoH (hidden-to-hidden connection). The second part of the context information is generated by the context module, which is similar to TDNN. It’s called ItoH.
we can find that the input projection vector $v$ in input projection vector. From equation (6) (7) (8) (12), occurring between the context information from HtoH and decline. We consider that it is because of the ratio imbalance length reaches a certain number, the model’s accuracy will decline. We consider that it is because of the ratio imbalance occurring between the context information from HtoH and ItoH in input projection vector. From equation (6) (7) (8) (12), we can find that the input projection vector $v_l$ is obtained by adding the lower layer information $v_{l1}$ introduced from ItoH and the current layer information $v_{l2}$ introduced from HtoH. Since the $v_{l1}$ is obtained by concatenating the input values of multiple time points in the lower layer, the values of each dimension of $v_{l1}$ will be much larger than those of the corresponding dimension of $v_{l2}$.

In this case, if we want to adjust the ratio of $v_{l1}$ and $v_{l2}$ in $v_l$, the general idea is to standardize the interior of vectors $v_{l1}$ and $v_{l2}$ respectively. However, considering that each dimension of a vector represents different feature, it is not appropriate to standardize the interior of the vector directly. We think of using batch normalization method to standardize each dimension of the vector in a batch.

There are three ways to set the BN layer.

1) Only apply BN to HtoH:

$$v'_l = v''_l + BN(v''_l)$$  \hspace{1cm} (15)

2) Only apply BN to ItoH:

$$v'_l = BN(v'_l) + v''_l$$ \hspace{1cm} (16)

3) Apply BN to both HtoH and ItoH:

$$v'_l = BN(v'_l) + BN(v''_l)$$ \hspace{1cm} (17)

The rest of the equations in Norm-PmGRU are shown below:

$$v''_l = Wv'_l x_l$$ \hspace{1cm} (18)

$$v'_{2l} = Wv'_{2l} h'_{t-1}$$ \hspace{1cm} (19)

$$z'_l = \sigma (Wz'_l)$$ \hspace{1cm} (20)

$$\widetilde{h}'_l = ReLU \left( BN \left( W_hv'_l \right) \right)$$ \hspace{1cm} (21)

$$h'_l = z'_l \odot h'_{l-1} + (1 - z'_l) \odot \widetilde{h}'_l$$ \hspace{1cm} (22)

$$\tilde{y}_l = W_y h'_l$$ \hspace{1cm} (23)

$$\tilde{y}'_l = \left[ y'_{l-s_1 \times 2} ; \cdots ; y'_{l-s_1 \times i} ; y'_{l-s_1 \times (i+2)} ; \cdots ; y'_{l-s_1 \times K_2} \right]$$ \hspace{1cm} (24)

The input projection vector $v'_l$ will be used to update $z'_l$ and $\tilde{h}'_l$ after affine transformation. When updating $\tilde{h}'_l$, a BN layer is added before the activation function $ReLU()$. No special operations are performed when $z'_l$ is updated.

III. EXPERIMENTAL PREPARATION

This section will introduce five ASR tasks and their experimental setups used in this work. All our experiments were performed on the Kaldi toolkit [34].

A. EXPERIMENTAL TASK

In order to compare the differences between the results of each model, we conducted experiments on four Chinese Mandarin and one English tasks.

1) ST-CMDS TASK

The ST Chinese Mandarin Corpus contains 86 hours of speech. We divide the data sets into training set and test set, the ratio of which is 14:1. The test set contains 74359 characters.

2) AISHELL-1 TASK

The AISHELL-1 Mandarin speech task [35] contains 178 hours of speech, and is divided into training(85%), development(10%) and testing(5%) sets. The development and test sets include 205341 and 104765 characters, respectively.

VOLUME 8, 2020 215195
3) AIDATATANG TASK
The AIDATATANG open Mandarin speech task contains 200 hours of acoustic data, which is mostly mobile recorded data. The database is divided into training set, development set, and testing set in a ratio of 7:1:2. The development and test sets include 234524 and 468933 characters, respectively.

4) AISHELL-2 TASK
AISHELL-2 is a 1000-hour Mandarin Chinese speech corpus. The performance of this task is evaluated on two public test sets: AISHELL2-2018A-EVAL-dev and AISHELL2-2018A-EVAL-test, including 24802 and 49532 characters, respectively.

5) LIBRISPEECH TASK
LibriSpeech is a 1000-hour English Speech Corpus. We select two datasets with high label accuracy, dev_clean and test_clean as test sets which contain 54402 and 52576 words respectively.

B. GMM-HMM EXPERIMENTAL SETUP
GMM-HMM model was used to generate alignment between corpus and it transcription. A monophone GMM-HMM model was firstly trained using 13-dimensional Mel Frequency Cepstrum Coefficient (MFCC) plus 3-dimensional pitch features [36]. After using decision tree to bind monophones, a small triphone model and a larger triphone model were then consecutively trained using 48-dimensional delta features [35]. Finally, Maximum likelihood linear transform (MLLT) and speaker adaptive training (SAT) were applied in training Speaker Dependent GMM-HMM model. But for AISHELL-2, we stopped GMM-HMM training at the speaker independent stage. Because from our experience, on thousands of hours of corpus, the performance of the final system primarily depends on the generalization ability from the DNN model. It is not worth spending too much time at GMM-HMM stage.

C. DNN EXPERIMENTAL SETUP
After getting the alignment of corpus and the word transcription, we used the alignment to train DNN model. All the DNN models in this paper are trained according to the following settings. In order to make the DNN-based acoustic model more robust to the tempo and volume variances of the testing data, we used the speed perturbation technique [37] to expand the training set to three times of the original training set, in which the speed perturbation factor is 0.9, 1.0 and 1.1. The 40-dimensional MFCC, 3-dimensional pitch and 100-dimensional i-Vector features are used in the training of DNN-based acoustic models [35]. The input feature at time t spliced from frame t − 2 to frame t + 2. All models had a delay tolerance of 50ms during training. Add the fixed delay of 20ms in the previous step, so all models have a delay of 70ms. And we used the subsampling technique to reduce the output frame rate of the model from 100Hz to 33Hz [38].

The training criterion adopted in this paper is the lattice-free maximum mutual information (LF-MMI) criterion [39]. The objective function of LF-MMI is given by:

\[
F_{LF-MMI} = \frac{1}{M} \sum_{m=1}^{M} \log \sum_{S_m} S_m p(O_m | S_m) p(W_m)
\]

where \(O_m\) is the speech corpus and \(W_m\) is the corresponding reference transcription. \(S_m\) is a hypothesis sequence from the denominator graph. \(S_m\) and \(S_m\) are state sequences corresponding to \(W_m\) and \(W_m\) respectively.

IV. EXPERIMENTAL AND ANALYSIS OF RESULTS
In this section, we evaluate the effectiveness of our two proposed models on ASR tasks. The first model is the Projected minimal Gated Recurrent Unit (PmGRU), and the second one is the Normalization Projected minimal Gated Recurrent Unit (Norm-PmGRU). All the models in this section are trained after GMM-HMM step following the process of DNN introduced above.

A. PMGRU EXPERIMENTS
We evaluated the PmGRU on AISHELL-1 task and compared its performance with the mGRUIP with Context module (mGRUIP-Ctx). When setting up the PmGRU architecture, we refer to the parameter settings of the mGRUIP-Ctx [32]. The architecture of the neural network is shown in Figure 3. The parameters of PmGRU architecture are set as follows:

1) The PmGRU architecture has five PmGRU hidden layers, the first one of which does not contain context module.

2) Considering that the dimension of input projection layer in mGRUIP-Ctx is 1/10 of the previous normal layer dimension, we set the dimension of the output projection layer of PmGRU to 1/10 of the dimension of the original output layer. Specifically, the dimension of memory cells in PmGRU structure is 2560, so the dimension of PmGRU output projection layer is set as 256. But for the last PmGRU hidden layer, since a 512-dimensional bottleneck layer is applied in training Speaker Dependent GMM-HMM model. The dimension of memory cells in PmGRU structure is 2560, so the dimension of PmGRU output layer is set as 256.

The performance of PmGRU with different context modules is compared. The step size \(s_1\) and \(s_2\) corresponding to context order \(K_1\) and \(K_2\) as well as the final results are shown in Table 1. The format of context order and step size in the table is \(K_1 \times s_1\); \(K_2 \times s_2\). The “Total” item in the table is obtained by weighted averaging of the “Dev” and “Test”
The weight of the two items is the number of characters in the corresponding data set.

Firstly, comparing the mGRUIP-Ctx and PmGRU_A, when the context module is the same, the character error rate (CER) of two models is close, but PmGRU_A provides 20% relative size reduction compared to mGRUIP-Ctx. This shows that the output projection layer can reduce the size of the model while retaining valid information. Compared with mGRUIP-Ctx, the best-performing model PmGRU_C provides 3.2% relative CER reduction whereas consuming 16% less size budget.

Secondly, from the comparison of the four PmGRU models, we can find that within a certain range, as the length of the context increases, the accuracy of the model also increases. However, after the context length exceeds a certain limit, the performance of the model decreases.

### B. NORM-PMGRU EXPERIMENTS

For the situation mentioned above that longer context length of context module (ItoH) will result in reducing of the model’s accuracy, we think that it is caused by ratio imbalance between the context information from HtoH and ItoH in input projection vector \( v_t \). It can assume that too much context information introduced by ItoH will hide the context information of HtoH. Therefore, we believe that adding batch normalization layer can adjust ratio of context information from HtoH and ItoH in input projection vector \( v_t \). Several groups of comparative experiments are made on AISHELL-1, and the experiments setup and results are shown in Table 2.

### TABLE 2. Comparison of BN methods with PmGRU on AISHELL-1 task.

| Model          | BN on | CER(%) |
|----------------|-------|--------|
|                | ItoH  | HtoH   | Dev | Test | Total |
| Norm-PmGRU_A   | Y     | N      | 5.86 | 6.92 | 6.21  |
| Norm-PmGRU_B   | Y     | Y      | 5.57 | 6.61 | 5.92  |

The parameter settings of Norm-PmGRU_A and Norm-PmGRU_B in Table 2 are the same as those of PmGRU_C and PmGRU_D in Table 1, respectively. From the comparison of the Norm-PmGRU_A model in Table 2, we can find that the model of applying batch normalization to ItoH and HtoH has the best performance. The values of HtoH and ItoH in a batch are standardized respectively, and their effects on the input projection vector are similar. Comparing the results in Table 2 and Table 1, we can find that the Norm-PmGRU_A has a relative 1.7% CER reduction compared to the PmGRU_C. By comparing Norm-PmGRU_A and Norm-PmGRU_B, we find the performance of Norm-PmGRU_B is not significantly improved compared with that of Norm-PmGRU_A, but the latency of Norm-PmGRU_B is increased by 20 ms. So all the Norm-PmGRU we used in the next step refers to Norm-PmGRU_A.

In order to study the effects of these two improvements on the model, the ablation experiments are conducted. The experimental results are shown in Table 3. Y in the table indicates that the item exists in the model, and N indicates that the item does not exist in the model. From the results, we find
TABLE 3. The ablation experiment of PmGRU on AISHELL-1 task.

| Model       | Output Proj | BN | CER(%)  |
|-------------|-------------|----|---------|
|             |             |    | Dev     | Test  | Total |
| PmGRU_A     | Y           | N  | 5.87    | 6.88  | 6.21  |
| Norm-mGRU1P-Ctx | N           | Y  | 5.64    | 6.67  | 5.99  |
| Norm-PmGRU_A | Y           | Y  | 5.57    | 6.61  | 5.92  |

TABLE 4. Configurations of baseline models.

| Model    | Architecture | Latency (ms) |
|----------|--------------|--------------|
| BLSTM    | [L_f, L_a], [L_f, L_a] | 2030         |
| TDNN-LSTM | T T T L_f T T L_f T T L_f | 210          |
| TDNN-OPGRU | T T O T T O T O | 210          |
| TDNN-mGRU | m T m T m T m T m B | 290          |

that both using BN (for HtoH and ItoH) and adding output projection layer can improve the performance of the model. However, compared with adding output projection layer, BN can improve the model more obviously.

The impact of BN on the model performance is further analyzed. Figure 4 is the result of decoding a segment of audio (the number of BAC009S0764W0121 selected in AISHELL-1 TEST set) with the PmGRU_C and the three Norm-PmGRU_A models of different BN methods. We retain the input projection vector \( v_t \) of the fifth PmGRU layer, the context information vector \( v_{1t} \) from the lower layer (ItoH) and the context information vector \( v_{2t} \) from current layer (HtoH) of each decoded frame. In Figure 4, the horizontal axis represents the time, and the vertical axis represents the average value of each dimension in vector. The blue circle represents input projection vector \( v_t \), the green pentagram represents \( v_{1t} \) (ItoH), and the red triangle represents \( v_{2t} \) (HtoH).

From Figure 4, we can find that the average value of input projection vector in PmGRU is mainly affected by the average value of context information vector from ItoH. When we only use BN for ItoH or HtoH, the input projection vector will be more affected by another non-BN connection. Only the input projection vector formed by BN for ItoH and HtoH respectively can take into account the influence of both the information from current layer and the information from the layer below.

C. COMPARISON BETWEEN NORM-PMGRU AND BASELINE MODELS

The comprehensive evaluation of the performance of Norm-PmGRU on ASR tasks is conducted. The details of baseline models in the experiments are shown in Table 4. The network architecture of Time Delay Neural Network-Long Short-Term Memory with Projection(TDNN- LSTMP), Bidirectional Long Short-Term Memory with Projection (BLSTMP), and Time Delay Neural Network-Output gate Projected Gated Recurrent Unit(TDNN-OPGRU) are described in [20] [21] [22], respectively. As [24] does not give the details of the Time Delay Neural Network-minimal Gated Recurrent Unit (TDNN-mGRU), when we set up the architecture of TDNN-mGRU, the parameters of TDNN are set as following the context module parameters of mGRU1P-Ctx [32], and the cell dimension is consistent with the setting of LSTMP in TDNN-LSTMP [20]. In Table 4, \( L_f \) is forward LSTMP layer, \( L_b \) is backward LSTMP layer, \( T \) is TDNN layer, \( m \) is mGRU layer, \( O \) is OPGRU layer, \( B \) is bottleneck layer.
1) COMPARISON BETWEEN NORM-PMGRU AND LSTM’S VARIANTS

In this section, we compare the performance of Norm-PmGRU with TDNN-LSTM and BLSTM in different ASR tasks. All results are shown in Table 5.

LSTM is formed by inserting LSTM a recurrent projection layer and a non-recurrent projection layer after the cell output units. These two projection layers make the LSTM architecture use parameters more efficiently than the standard LSTM architecture. BLSTM is a combination of forward LSTM and backward LSTM. This structure provides complete past and future context information for each point in the output layer sequence. When comparing Norm-PmGRU with BLSTM, we can find that Norm-PmGRU performed better than BLSTM on ST-CMDS, AIDATATANG, AISHELL-1 and LibriSpeech. And the Norm-PmGRU reduces the parameters by about 45% and has less latency compared with BLSTM. Because BLSTM has a latency of 2030 ms, it is not suitable for online speech recognition.

The Norm-PmGRU performs better than TDNN-LSTM on four Chinese ASR tasks, not only has a lower CER, but also requires smaller size. Specifically, as 45% fewer parameters than TDNN-LSTM, Norm-PmGRU achieves a relative CER reduction of 1.56% over TDNN-LSTM on ST-CMDS task, 6.18% on AISHELL-1 task, 5.40% on AIDATATANG task, and 1.68% on AISHELL-2 task. At the same time, Norm-PmGRU also brings about 3.5 times the decoding speed which is commonly measured by real time factor (RTF). The minor inadequacy is that Norm-PmGRU introduces a latency 80 ms longer than TDNN-LSTM.

2) COMPARISON BETWEEN NORM-PMGRU AND GRU’S VARIANTS

In this section, we compare the performance of Norm-PmGRU with TDNN-mGRU, TDNN-OPGRU and mGRU in different ASR tasks. All results are shown in Table 5.

The OPGRU develops from GRU by projecting its memory cell into recurrent part and non-recurrent part, and replacing its reset gate with an output gate. Recurrent part in projected memory cell is used for recurrence of OPGRU, and the whole projected memory cell is the output feeding into next layer, where the concept is similar to LSTMP. Comparing Norm-PmGRU with TDNN-OPGRU, we can see that the Norm-PmGRU has better performance than TDNN-OPGRU in all the five tasks. Specifically, even that model size of Norm-PmGRU is 39% smaller than TDNN-OPGRU, its CER(WER) is 3.65%, 7.36%, 8.51%, 4.15% and 4.32% relatively lower than TDNN-OPGRU on Task ST-CMDS, AISHELL-1, AIDATATANG, AISHELL-2, LibriSpeech, respectively.

mGRU is a simplified version of GRU by removing out the reset gate. The mGRU-Ctx and Norm-PmGRU are built on the basis of mGRU. By adding TDNN layer to introducing context information, TDNN-mGRU still performs not better than mGRU-Ctx and Norm-PmGRU. mGRU-Ctx is formed by adding an input projection layer
to the mGRU and then introducing context module, while Norm-PmGRU is obtained by adding an output projection layer to the mGRUIP-Ctx and changing position of the batch normalization layer. Norm-PmGRU performs better than mGRUIP-Ctx in the four Chinese tasks. Compared with mGRUIP-Ctx, the CER of Norm-PmGRU relative reduces by 5.95%, 4.52%, 4.83% and 2.43% at task ST-CMDS, AISHELL-1, AIDATATANG and AISHELL-2 respectively, which we believe is the result of adding more context information to Norm-PmGRU. At the same time, Norm-PmGRU’s architecture uses parameters more efficiently due to the existence of output projection layer. More context information has been successfully added to the Norm-PmGRU, while the parameter used is decreased by about 16% than that in mGRUIP-Ctx.

V. CONCLUSION

In this paper, we revised mGRUIP-Ctx for speech recognition acoustic model. The proposed Norm-PmGRU architecture is an improved version of the mGRUIP-Ctx, in which an output projection layer is added after mGRUIP-Ctx’s output. After that, batch normalization is used to further improve model performance as well as to adjust the ratio of context information of the previous layer to the current layer.

Experiments on five different ASR tasks have shown the effectiveness of the proposed model. Compared to baseline models, our model not only has the almost same accuracy of recognition or even better, but also uses the least number of parameters. Specially, on task ST-CMDS, AISHELL-1, AIDATATANG and AISHELL-2, the character error rate (CER) of Norm-PmGRU relatively reduces by 5.95%, 4.52%, 4.83% and 2.43% than mGRUIP-Ctx. On task LibriSpeech, Norm-PmGRU has the almost same word error rate (WER) with mGRUIP-Ctx. In the meantime, the parameter used in Norm-PmGRU is decreased by about 16% than that in mGRUIP-Ctx.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their industrious work and insightful comments.

REFERENCES

[1] Y. Sun, Y. Chen, X. Wang, and X. Tang, “Deep learning face representation by joint identification-verification,” in Proc. Adv. Neural Inf. Process. Syst., vol. 3, 2014, pp. 1988–1996.
[2] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal Process. Mag., vol. 29, no. 6, pp. 82–97, Nov. 2012.
[3] L. Liu, Y.-J. Liu, and S. Tong, “Neural networks-based adaptive finite-time fault-tolerant control for a class of strict-feedback switched nonlinear systems,” IEEE Trans. Cybern., vol. 49, no. 7, pp. 2536–2545, Jul. 2019.
[4] H. Liang, L. Zhang, Y. Sun, and T. Huang, “Containment control of semi-Markovian multiagent systems with switching topologies,” IEEE Trans. Syst., Man, Cybern., Syst., early access, Oct. 28, 2019, doi: 10.1109/TSMC.2019.2946248.
[5] Q. Zhou, W. Wang, H. Liang, M. Basin, and B. Wang, “Observer-based event-triggered fuzzy adaptive bipartite containment control of multi-agent systems with input quantization,” IEEE Trans. Fuzzy Syst., early access, Nov. 15, 2019, doi: 10.1109/TFUZZ.2019.2953575.
[6] T.-J. Lee, B.-M. Jang, and D.-I. Dan Cho, “A non-iterative pose-graph optimization algorithm for fast 2D SLAM,” in Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO), Dec. 2014, pp. 1596–1601.
[7] M. Zorzi, A. Zanella, A. Testolin, M. De Filippo De Grazia, and M. Zorzi, “Cognition-based networks: A new perspective on network optimization using learning and distributed intelligence,” IEEE Access, vol. 3, pp. 1512–1530, 2015.
[8] A. D. Amodei, S. Ananthanarayan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, and J. Chen, “Deep speech 2: End-to-end speech recognition in English and mandarin,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 173–182.
[9] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Mar. 2016, pp. 4960–4964.
[10] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bachchiani, “State-of-the-art speech recognition with sequence-to-sequence models,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Apr. 2018, pp. 4774–4778.
[11] G. Shuang, C. Weng, G. Chen, S. Gu, M. Luo, D. Yu, and L. Xie, “Investigating End-to-end speech recognition for mandarin-english code-switching,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2019, pp. 6056–6060.
[12] A. B. Nassif, I. Shahin, I. Attilli, M. Azzeh, and K. Shaalan, “Speech recognition using deep neural networks: A systematic review,” IEEE Access, vol. 7, pp. 19143–19165, 2019.
[13] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, “Phoneme recognition using time-delay neural networks,” IEEE Trans. Acoust., Speech Signal Process., vol. 37, no. 3, pp. 328–339, Mar. 1989.
[14] V. Peddinti, D. Povey, and S. Khudanpur, “A time delay neural network architecture for efficient modeling of long temporal contexts,” in Proc. 16th Annu. Conf. Int. Speech Commun. Assoc., Jan. 2015, pp. 3214–3218.
[15] V. Peddinti, G. Shen, D. Povey, and S. Khudanpur, “Reverberation robust acoustic modeling using i-vectors with time delay neural networks,” in Proc. 16th Annu. Conf. Int. Speech Commun. Assoc., Jan. 2015, pp. 2440–2444.
[16] S. Zhang, H. Jiang, S. Xiong, S. Wei, and L.-R. Dai, “Compact feed-forward sequential memory networks for large vocabulary continuous speech recognition,” in Proc. Interspeech, Sep. 2016, pp. 3389–3393.
[17] S. Zhang, M. Lei, Z. Yan, and L. Dai, “Deep-FSMN for large vocabulary continuous speech recognition,” in Proc. Int. Conf. Acoust., Speech Signal Process. (ICASSP), Apr. 2018, pp. 5869–5873.
[18] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
[19] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN Encoder–Decoder for statistical machine translation,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1724–1734.
[20] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” in Proc. Ann. Conf. Int. Speech Commun. Assoc., 2014, pp. 338–342.
[21] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, “Low latency acoustic modeling using temporal convolution and LSTMs,” IEEE Signal Process. Mag., vol. 25, no. 3, pp. 373–385, Mar. 2018.
D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, “Semi-orthogonal low-rank matrix factorization for deep neural networks,” in Proc. Annu. Conf. Int. Speech Commun. Assoc., Sep. 2018, pp. 3743–3747.

T. Mori, A. Tjandra, S. Sakti, and S. Nakamura, “Compressing end-to-end ASR networks by tensor-train decomposition,” in Proc. Annu. Conf. Int. Speech Commun. Assoc., Sep. 2018, pp. 806–810.

A. Tjandra, S. Sakti, and S. Nakamura, “Compressing recurrent neural network with tensor train,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), May 2017, pp. 4451–4458.

Z. Lu, V. Sindhwani, and T. N. Sainath, “Learning compact recurrent neural networks,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Mar. 2016, pp. 5960–5964.

S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. 32nd Int. Conf. Mach. Learn., 2015, pp. 448–456.

J. Li, Y. Shan, X. Wang, and Y. Li, “Improving gated recurrent unit based acoustic modeling with batch normalization and enlarged context,” in Proc. 11th Int. Symp. Chin. Spoken Lang. Process. (ISCSLP), Nov. 2018, pp. 126–130.

C. Laurent, G. Pereyra, P. Brakel, Y. Zhang, and Y. Bengio, “Batch normalized recurrent neural networks,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Mar. 2016, pp. 2657–2661.

D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The Kaldi speech recognition toolkit,” presented at the IEEE Workshop Autom. Speech Recognit. Understand., Big Island, HI, USA, Dec. 11–15, 2011. [Online]. Available: https://infoscience.epfl.ch/record/192584

H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, “AISHELL-1: An open-source mandarin speech corpus and a speech recognition baseline,” in Proc. 20th Conf. Oriental Chapter Int. Coordinating Committee Speech Databases Speech I/O Syst. Assessment (O-COCOSDA), Nov. 2017, pp. 1–5.

P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, “A pitch extraction algorithm tuned for automatic speech recognition,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2014, pp. 2494–2498.

R. Feng (Member, IEEE) received the Ph.D. degree in electronics science and technology from X’ian Jiaotong University, X’ian, China, in 2004. From 2005 to 2006, he was a Postdoctoral Research Associate with Osaka University, Osaka, Japan. From July 2004 to December 2006, he was an Assistant Professor with the School of automation, University of Posts and Telecommunications, Beijing, China. Since January 2007, he has been an Associate Professor with the School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing. His current research interests include speech recognition, signal processing, and wireless sensor networks.

YINFENG WU (Member, IEEE) received the B.S. and M.S. degrees from the Guilin University of Electronic Technology, Guilin, China, in 2002 and 2005, respectively, and the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2008. From June 2012 to September 2013, he was a Visiting Scholar with Cornell University, Ithaca, NY, USA. He is currently an Assistant Professor with Beihang University. His research interests include speech recognition, digital signal processing, multi-source information fusion, wireless sensor networks, and intelligent instrumentation.

JIAJUN HU received the B.S. degree from the Hefei University of Technology, Hefei, China, in 2020. She is currently pursuing the M.A.Eng. degree with the School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing. Her current research interests include speech recognition, digital signal processing, and deep learning.