INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

SpeechMoE: Scaling to Large Acoustic Models with Dynamic Routing Mixture of Experts

Zhao You[*1], *Shulin Feng*[*1], *Dan Su*[1], *Dong Yu*[2]

[1]Tencent AI Lab, Shenzhen, China
[2]Tencent AI Lab, Bellevue, WA, USA

{dennisyou, shulinfeng, dansu, dyu}@tencent.com

Abstract

Recently, Mixture of Experts (MoE) based Transformer has shown promising results in many domains. This is largely due to the following advantages of this architecture: firstly, MoE based Transformer can increase model capacity without computational cost increasing both at training and inference time. Besides, MoE based Transformer is a dynamic network which can adapt to the varying complexity of input instances in real-world applications. In this work, we explore the MoE based model for speech recognition, named SpeechMoE. To further control the sparsity of router activation and improve the diversity of gate values, we propose a sparsity $L1$ loss and a mean importance loss respectively. In addition, a new router architecture is used in SpeechMoE which can simultaneously utilize the information from a shared embedding network and the hierarchical representation of different MoE layers. Experimental results show that SpeechMoE can achieve lower character error rate (CER) with comparable computation cost than traditional static networks, providing 7.0%∼ 23.0% relative CER improvements on four evaluation datasets.

Index Terms: mixture of experts, dynamic routing, acoustic model, speech recognition

1. Introduction

Owing to powerful representation, Deep Neural Networks (DNN) have gained great success in speech recognition [1, 2]. Various types of neural network architectures have been employed in ASR systems, such as convolutional neural networks (CNNs) [3, 4], long short-term memory (LSTM) [5], gated recurrent unit[6], time-delayed neural network [7], feedforward sequential memory networks (FSMN) [8], etc. Recently, more powerful deep models such as Transformer [9], Emformer [10] and Conformer [11] have proved their efficacy to further improve the speech recognition performance.

Increasing model and training data size has been shown an effective way to improve the system performance, which is especially demonstrated in the field of language modeling [12, 13]. Recently, deep mixture of experts (MoE) based approaches [14, 15] have been intensively investigated and applied in different tasks such as language modeling [16, 17] and image classification [18, 19, 20, 21]. The benefits mainly come from two aspects: First, MoE is an effective way to increase model capacity. Second, with introduction of the sparsely-gated mixture-of-experts layer [22], an attractive property of MoE models is the sparsely dynamic routing, which enables us to satisfy training and inference efficiency by having a sub-network activated on a per-example basis.

In real-world applications, speech recognition systems need to be robust with different input conditions such as speakers, recording channels and acoustic environments. Larger models are appealing while the increase of training and inference cost can not be afforded. The major problem is that the computational cost of a static model is fixed and can not be adaptive to the varying complexity of input instances. Therefore, developing mixture of expert models for speech recognition with dynamic routing mechanism is a promising exploration.

In this study, we explore mixture of experts approach for speech recognition. We propose a novel dynamic routing mixture of experts architecture, similar to [17], which comprises of a set of experts and a router network. The router takes output of the previous layer as input and routes it to the best determined expert network. We find that the balance loss proposed in [17] achieves balanced routing but the sparsity of router activation can not always be guaranteed. Here, we propose a sparsity $L1$ loss to encourage the router activation to be sparse for each example. Besides, we use a mean importance loss to further improve the balance of expert utilization. Furthermore, a shared embedding network is used in our architecture to improve the route decisions, whose output will be combined with the output of previous layers as the input of routers.

The rest of the paper is organized as follows. Section 2 reviews the related works of MoE and Section 3 represents our proposed method SpeechMoE. The experimental setup is described in Section 4 and the experimental results are reported in Section 5. Finally, we conclude this paper in Section 6.

2. Related works

In this section, we mainly describe two different architectures of MoE.

2.1. DeepMoE

The DeepMoE architecture proposed in [20] can achieve lower computational cost and higher prediction accuracy than standard convolutional networks. The architecture designs a sparse gating network which can dynamically select and re-weight the channels in each layer of the base convolutional network. Fig.1(a) shows the detailed architecture of DeepMoE. The DeepMoE consists of a base convolutinal network, a shared embedding network and a multi-headed sparse gating network. The gating network transforms the output of the shared embedding network into sparse mixture weights:

$$g^l(e) = f(W_g^l \cdot e) \tag{1}$$

where $g^l(e)$ is the sparse mixture weights of l-th convolutional layer, e is the output of the shared embedding network, and f is the activation operation(i.e., ReLU). Then, the output of l-th

[*]Equal contribution.

http://dx.doi.org/10.21437/Interspeech.2021-478

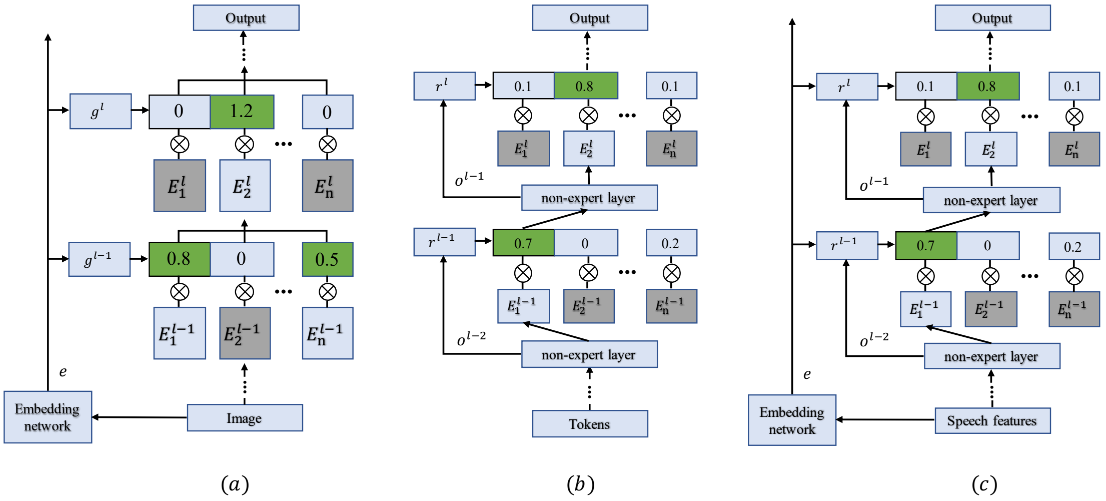

Figure 1: *(a), (b) and (c) represent the architecture of DeepMoE, Switch Transformer and SpeechMoE respectively. Similar to Switch Transformer, only one expert with the largest router probability in each MoE layer is used in the SpeechMoE, which is different from DeepMoE. Besides, the SpeechMoE utilizes a shared embedding and output of the previous layer as the input of each router.*

convolutional layer can be formulated as:

$$y^l = \sum_{i=1}^{n} g_i^l E_i^l \tag{2}$$

where n is the input channels number of l-th convolutional layer and E_i^l is the i-th channel of l-th convolutional layer, treated as the i-th expert in l-th layer.

The loss function for training DeepMoE is defined as:

$$L(x; y) = L_c(x; y) + \alpha L_g(x; y) + \beta L_e(x; y) \tag{3}$$

where x and y are the input image feature and target label, respectively. L_c is the classification loss, L_g is the L1 regularization term which controls sparsity of the gating network and L_e is the additional classification loss which encourages the diversity of shared embedding network.

2.2. Switch Transformer

Fedus et al. proposed the Switch Transformer [17] for language modeling, which further reduces computation and communication costs by simplifying the MoE routing algorithm. The architecture of Switch Transformer is described in Fig.1(b), where experts refer to feed-forward networks and the non-expert layers refer to the self-attention layers. Each MoE layer consists of n experts and a router layer. It takes output for the previous layer as input and routes it to the top-1 expert with the largest router probability. Let W_r^l and o^{l-1} be the router weights of the l-th layer an the output of the previous layer, then the router probability can be defined as follows:

$$r^l = W_r^l \cdot o^{l-1} \tag{4}$$

$$p_i^l = \frac{exp^{r_i^l}}{\sum_{j=1}^{n} exp^{r_j^l}} \tag{5}$$

Then, the selected expert's output is also gated by router probability to get output of the MoE layer,

$$y^l = p_i^l E_i^l \tag{6}$$

Since only one expert is active in each layer, the Switch Transformer can keep the computational cost constant while scaling to a very large model. To encourage a balance load across experts, the balancing loss [17] is added into the loss function and defined as:

$$L_b = n \cdot \sum_{i=1}^{n} s_i \cdot P_i \tag{7}$$

where s_i is the fraction of samples dispatched to expert i, P_i is the fraction of router probability allocated for expert i.

3. SpeechMoE

3.1. Model architecture

Fig.1(c) shows an overview of the architecture of our proposed SpeechMoE. For speech recognition, its input is speech features (e.g. fbanks) and the input frames will be dispatched to experts in each layer. Similar to the Switch Transformer, SpeechMoE only selects one expert in each layer to reduce the computational cost. Compared with Switch Transformer and DeepMoE, the SpeechMoE concatenates the shared embedding with output of the previous layer as the input of routers, which can be defined as:

$$r^l = W_r^l \cdot Concat(e; o^{l-1}) \tag{8}$$

This router mechanism comes from two considerations: (1) All gating values in DeepMoE are controlled by the shared embedding, which may decay to similar gating results in each layer. Utilizing the hierarchical representation from output of each layer may lead to diverse routing results for SpeechMoE. (2) The shared embedding relative to the goal task may be helpful to get a better routing strategy, providing a high-level distinctive representation and making the experts specialized to process distinct input frames.

3.2. Training objective

3.2.1. sparsity $L1$ loss

In our study, we find that the router probability distribution tends to be uniform when we only use the balancing loss proposed in [17], resulting in a bad performance. In order to encourage the sparsity of router activation, we propose a sparsity $L1$ loss, defined as follows:

$$L_s = \frac{1}{m}\sum_{i=1}^{m} \parallel \hat{f}_i \parallel_1 \tag{9}$$

where $\hat{f}_i = \frac{f_i}{\|f_i\|_2}$, stands for the unit normalized router probability distribution of sample i, and m is the number of samples in this mini-batch. Due to the unit normalization, minimizing the $L1$ norm will force the distribution close to space axes and attain sparsity.

3.2.2. Mean importance loss

We have also observed that model isn't balanced enough when increasing the number of experts. To solve this problem, we use a modified importance loss[22] to replace the balancing loss, defined as follows:

$$Imp = \frac{1}{m}\sum_{i=1}^{m} p_i \tag{10}$$

$$\bar{L}_m = n\sum_{j=1}^{n} {Imp_j}^2 \tag{11}$$

The mean importance is defined as the mean activation of experts on batch of samples and the loss is defined as the squared sum of mean importance of each expert. It's clear that when mean importance of each expert is averaged $\frac{1}{n}$, the loss reaches the minimum. Compared with the balancing loss in which s_i is not differentiable, the mean importance loss is more smooth, leading to a more balanced routing strategy.

3.2.3. Loss function

Given the input x and the target y, the full loss function of our method is defined as

$$L(x; y) = L_c(x; y) + \alpha L_s(x) + \beta \bar{L}_m(x) + \gamma L_e(x; y) \tag{12}$$

Among these items, L_c is the CTC loss [23] for speech recognition, L_s and $\bar{L}_m$ are the mentioned sparsity $L1$ loss and mean importance loss, used to encourage sparsity and diversity of the SpeechMoE model. Similar to [20], we introduce an additional embedding loss L_e, which is also the CTC loss. It shares the same goal with our SpeechMoE model and provides reliable embeddings for the routers. α, β, and γ are the scale for L_s, $\bar{L}_m$ and L_e respectively.

4. Experimental Setup

4.1. Training setup

The speech features used in all the experiments are 40-dimensional log-Mel filterbank features appended with the first-order and the second-order derivatives. Log-mel filterbank features are computed with a 25ms window and shifted every 10ms. We stack 8 consecutive frames and subsample the input frames with 3. A global mean and variance normalization is applied for each frame. All the experiments are based on the CTC learning framework. We use the context-independent-syllable-based acoustic modeling method [24] for CTC learning. The target labels of CTC learning are defined to include 1394 Mandarin syllables, 39 English phones, and a blank. Character error rate results are measured on the test sets and the floating point operations (FLOPs) for a one-second example is used to evaluate the inference computational cost. We use a pruned, first pass, 5-gram language model. All the systems use a vocabulary that consists of millions of words. Decoding is performed with a beam search algorithm by using the weighted finite-state transducers (WFSTs).

4.2. Datasets

Our training corpus is mixed data sets collected from several different application domains, all in Mandarin. In order to improve system robustness, a set of simulated room impulse responses (RIRs) are created with different rectangular room sizes, speaker positions, and microphone positions, as proposed in [25]. Totally, It comes to a 10k hours training corpus.

To evaluate the performance of our proposed method, we report performance on 3 types of test sets which consist of hand-transcribed anonymized utterances extracted from reading speech (1001 utterances), conversation speech (1665 utterances) and spontaneous speech (2952 utterances). We refer them as Read, Chat, and Spon respectively. In addition, to provide a public benchmark, we also use AISHELL-2 development set (2500 utterances) recorded by high fidelity microphone as the test set.

4.3. Acoustic Model

Our acoustic models consist of four components: MoE layer, sequential memory layer [26], self-attention layer [27] and the output softmax layer. Each MoE layer includes a router and a set of experts which is a feed forward network with one hidden layer of size 1024 activated by ReLU and an projection layer of size 512. For the sequential memory layer, the look-back order and look-ahead order of each memory block is 5 and 1 respectively, and the strides are 2 and 1 respectively. For the self-attention layer, we set the model dimension $d = 512$ and the number of heads $h = 8$. For every layer excluding the output softmax layer, the residual connection is applied.

The backbone of our model consists of 30 MoE layers, 30 sequential memory layers and 3 self-attention layers. Each MoE layer is followed by one sequential memory layer, and a self-attention layer is inserted after each 10 consecutive MoE and sequential memory layers. In our experiments, we vary the number of experts of MoE layers to be 2, 4 and 8, which are marked as MoE-2e, MoE-4e and MoE-8e respectively. The shared embedding network is a static model without MoE layers but a similar structure to the backbone.

In our study, we built two baseline systems for evaluating the performance of our proposed method:

- Baseline 1 (B1): The static model without MoE layers but a similar structure to the backbone of SpeechMoE models, which can also be treated as MoE-1e. Since the proposed method uses an extra embedding network, B1 model is designed to have 60 layers to be FLOP-matched with our MoE models.

- Baseline 2 (B2): The model with 4 experts in each MoE layer, which does not have the shared embedding network and is trained with only the auxiliary balancing loss proposed in Switch Transformer.

For all experiments on MoE models, we set the hyper-parameters $\alpha = 0.1$, $\beta = 0.1$ and $\gamma = 0.01$.

Table 1: *Results of adding sparseness $L1$ loss.*

Model	Params	FLOPs	Test set			
			Read	Chat	Spon	AISHELL
B1	71M	2.3B	2.0	22.92	24.95	4.52
B2	134M	2.3B	1.81	22.49	24.90	4.50
MoE-$L1$	134M	2.3B	**1.69**	**22.47**	**24.70**	**4.25**

Table 2: *Results of augmenting shared embedding network and utilizing mean importance loss.*

Model	Params	FLOPs	Test set			
			Read	Chat	Spon	AISHELL
MoE-$L1$	134M	2.3B	1.69	22.47	24.70	4.25
+emb	170M	2.3B	**1.63**	**22.15**	**24.15**	**4.16**
+imp loss	170M	2.3B	**1.58**	**21.57**	**23.31**	**4.00**

5. Experimental Results

5.1. Adding sparsity $L1$ loss

In this section, we investigate the performance of adding the sparsity $L1$ loss in training. We have trained two baseline systems for this evaluation. The first baseline system(B1) is the static model trained based on L_c loss and the other one(B2) is trained based on L_c and L_b loss mentioned above. Our result of adding sparsity $L1$ loss relative to B2 is marked as MoE-$L1$.

As shown in Table 1, B2 performs a little better than B1 with more parameters and comparable computational cost. It is as expected that the MoE-$L1$ which uses both balancing loss and sparsity $L1$ loss achieves the best performance compared with two baseline systems. This indicates that the additional sparsity $L1$ loss brings about more sparsity to router probability distribution. The routers become more distinctive and specialized for varying input frames so that the model get a better performance.

5.2. Augmenting shared embedding network

In this section, we evaluate the performance of the new router architecture which concatenates the shared embedding with output of the previous layer as the input of the router. As can be observed in Table 2, the proposed router architecture achieves lower character error rate comparing with MoE-$L1$ model.

It is worthy to note that only using output of previous layer as input does not work very well, which contradict with the method used in [17]. A reasonable explanation is that for language modeling, the word input as high-level representation already has good distinction, while for speech recognition the spectrum input is low-level feature which can not provide enough distinction information for routers, so the shared embedding network which converts low-level features to high-level embedding, is necessary to help router attain better selecting effect.

5.3. Utilizing mean importance loss

The last line of Table 2 presents the effects of the mean importance loss in place of the balancing loss. We observe that the proposed loss can further achieve lower character error rate than MoE-$L1$ model with embedding network on the four test sets. Since the mean importance loss encourages all experts to have equal importance, it will help the routers dispatch input frames to experts in a balanced way, avoiding the situation that some experts get no samples for training. Thus, the experts will be more diverse and result in a better performance.

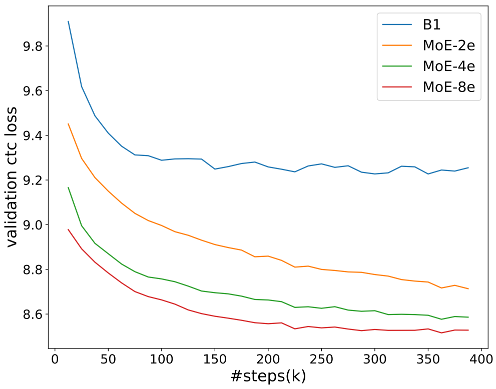

Figure 2: *Validation CTC loss for increasing expert number*

Table 3: *Results of increasing the number of experts.*

Model	Params	FLOPs	Test set			
			Read	Chat	Spon	AISHELL
B1	71M	2.3B	2.0	22.92	24.95	4.52
MoE-2e	105M	2.3B	1.62	21.82	23.52	4.08
MoE-4e	170M	2.3B	1.58	21.57	23.31	4.00
MoE-8e	297M	2.3B	**1.54** **(-23.0%)**	**21.31** **(-7.0%)**	**22.97** **(-7.9%)**	**3.98** **(-11.9%)**

5.4. Increasing the number of experts

In this section, we investigate the effect of increasing the number of experts. Table 3 shows the performance comparison on different number of experts with SpeechMoE. Line 2 presents the results of the baseline system (B1). The following three lines present results of 3 different number of experts which are marked as MoE-2e, MoE-4e and MoE-8e respectively. The results clearly show that performance get better as the number of experts increases. Specifically, MoE-8e achieves up to 23.0% relative CER improvement over the baseline model on the Read test set, and the gain is between 7.0%∼11.9% for other more realistic test sets.

Fig.2 shows the validation CTC loss of MoE with different number of experts and the baseline model. As shown, the MoE-8e model produces the lowest CTC loss compared with both the baseline model and the other SpeechMoE models. Moreover, we observe that having more experts speeds up training. This suggests that increasing the number of expert leads to more powerful models.

6. Conclusions and future work

In this paper, we explore a mixture of experts approach for speech recognition. We propose a novel dynamic routing acoustic model architecture, the router module is enhanced by combining the previous layer's output and embedding from an isolated embedding network. We also improve the training loss that can both achieve better sparsity and balancing among different experts. Thorough experiments are conducted on training with different loss and varied number of experts. Future work includes both extending training data scale and number of experts, increasing by one or two orders of magnitudes, and exploring the proposed SpeechMoE model with other end-to-end training framework such as transformer transducers.

7. References

[1] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," in *IEEE Transactions on audio, speech, and language processing*, vol. 20. IEEE, 2012, p. 30–42.

[2] D. Yu and J. Li, "Recent progresses in deep learning based acoustic models," in *IEEE/CAA Journal of Automatica Sinica*, vol. 4. IEEE, 2017, p. 396–409.

[3] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 8614–8618.

[4] Y. Qian and P. C. Woodland, "Very deep convolutional neural networks for robust speech recognition," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 481–488.

[5] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 2013, pp. 273–278.

[6] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light gated recurrent units for speech recognition," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 92–102, 2018.

[7] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[8] S. Zhang, M. Lei, Z. Yan, and L. Dai, "Deep-fsmn for large vocabulary continuous speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5869–5873.

[9] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5884–5888.

[10] Y. Shi, Y. Wang, C. Wu, C.-F. Yeh, J. Chan, F. Zhang, D. Le, and M. Seltzer, "Emformer: Efficient Memory Transformer Based Acoustic Model For Low Latency Streaming Speech Recognition," *arXiv e-prints*, p. arXiv:2010.10759, Oct. 2020.

[11] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," 2020.

[12] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, "Megatron-lm: Training multi-billion parameter language models using model parallelism," *arXiv preprint arXiv:1909.08053*, 2019.

[13] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[14] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[15] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural computation*, vol. 6, no. 2, pp. 181–214, 1994.

[16] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," *arXiv preprint arXiv:2006.16668*, 2020.

[17] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *arXiv preprint arXiv:2101.03961*, 2021.

[18] S. Gross, M. Ranzato, and A. Szlam, "Hard mixtures of experts for large scale weakly supervised vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6865–6873.

[19] K. Ahmed, M. H. Baig, and L. Torresani, "Network of experts for large-scale image categorization," in *European Conference on Computer Vision*. Springer, 2016, pp. 516–532.

[20] X. Wang, F. Yu, L. Dunlap, Y.-A. Ma, R. Wang, A. Mirhoseini, T. Darrell, and J. E. Gonzalez, "Deep mixture of experts via shallow embedding," in *Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 552–562.

[21] S. Cai, Y. Shu, and W. Wang, "Dynamic routing networks," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3588–3597.

[22] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[23] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[24] Z. Qu, P. Haghani, E. Weinstein, and P. Moreno, "Syllable-based acoustic modeling with ctc-smbr-lstm," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 173–177.

[25] I. Himawan, P. Motlicek, D. Imseng, B. Potard, N. Kim, and J. Lee, "Learning feature mapping using deep neural network bottleneck features for distant large vocabulary speech recognition," in *International Conference on Acoustics, Speech and Signal Processing*, 2015.

[26] S. Zhang, M. Lei, Z. Yan, and L. Dai, "Deep-fsmn for large vocabulary continuous speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5869–5873.

[27] Z. You, D. Su, J. Chen, C. Weng, and D. Yu, "Dfsmn-san with persistent memory model for automatic speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7704–7708.