INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Attention gate between capsules in fully capsule-network speech recognition

Kyungmin Lee[1], *Hyeontaek Lim*[1], *Mun-Hwan Lee*[2], *Hong-Gee Kim*[2*]

[1]Samsung Research, Samsung Electronics, Seoul, South Korea
[2]Biomedical Knowledge Engineering Laboratory, Seoul National University, Seoul, South Korea
{k.m.lee, ht625.lim}@samsung.com, {munhwanlee, hgkim}@snu.ac.kr

Abstract

We present a novel capsule network-based speech recognition model that effectively utilizes the full context of past time capsules. The input capsule sequences are recurrently used by filtering unnecessary contextual information using multi-head attention, which uses previous time output vectors as keys and values, and current time output vectors as queries. We applied the attention gate to the sequential dynamic routing (SDR), an all-capsule speech recognition model. The proposed method attained higher accuracy than the existing SDR with two attention heads on all test sets of the TIMIT and *Wall Street Journal* (WSJ) corpora while maintaining the same algorithmic delay. For the WSJ corpus, 10.75% of a relative word error rate (WER) reduction was achieved when the required delay was set to 525 ms. In addition, the model showed a 1.76× reduction in delay while maintaining the WERs. The proposed method results in an increase of approximately 0.1% in the number of parameters.

Index Terms: speech recognition, capsule networks, sequential routing framework, multi-head attention

1. Introduction

Neural speech recognition (NSR) systems are systems composed of neural networks (NNs) that learn sequence-to-sequence mapping to convert variable-length speech signals into text [1, 2]. These systems are replacing conventional automatic speech recognition (ASR) systems because they require less handcrafted data, learn better representations, and are suitable for model compression [3]. An NSR model consists of two submodels, each of which can be pre-trained separately: a speech encoder for learning acoustic features and a decoder for learning linguistic information. NSR models can be classified as recurrent neural network-transducer (RNN-T) networks [1] or attention-based networks [2] according to how the two submodels are combined. Furthermore, a speech encoder can act as a type of NSR model in itself [1] when configured as a connectionist temporal classifier (CTC) network [4].

The selection of a speech encoder architecture is a crucial factor that determines the stream processing abilities of an NSR system. For instance, speech encoders built with either bidirectional long short-term memory (BLSTM) [1] or standard self-attention layers [5] contain a non-streamable architecture, as they require a full input sequence to encode a current frame. By adopting unidirectional long short-term memory (ULSTM) [1] and masked attention mechanisms [6, 7], including conformers [8], these sequential neural layers can encode input speech in an online processing manner by utilizing limited contexts surrounding current frames. However, a trade-off between recognition accuracy and the width of look-ahead contexts remains unavoidable [3, 9].

A capsule network (CapsNet) [10, 11] encodes a component of an object into a capsule: a group of neurons consisting of an activation scalar and instantiation vector. The former denotes the probability of the existence of the component, whereas the latter represents its multi-dimensional properties, such as scale and skew for images. A CapsNet trains the information transfer from lower to higher capsule levels via routing methods that employ unsupervised clustering in addition to gradient-based training procedures. With their remarkable abilities in encoding graphic features, CapsNets have received attention not only in visual tasks [12, 13], but have also been applied to time-series classification tasks [14, 15]. Recently, sequential routing framework (SRF) [16] that uses a capsule-only architecture and initializes routing coefficients with previous time output vectors was studied as an online speech recognition model with an algorithmic delay. SRF models designed to use CTC [4] as a loss function have yielded competitive accuracy in phoneme- and character-level speech recognition while maintaining their streaming abilities compared with both online and offline conventional CTC NNs. However, a dynamic routing (DR) [10] version of SRF requires a long algorithmic delay of almost 1 second to achieve competitive word error rates (WERs). This long delay is a crucial drawback that impedes the fast processing of speech commands and, consequently, the widespread adoption of CapsNets in ASR systems. The SRF models can be improved further by utilizing the complete context of past time capsules in addition to sequential routing.

In this paper, we present a novel multi-head attention (MHA)-based gate mechanism for CapsNet-based speech recognition models to efficiently encode sequential capsules. The idea behind the proposed method is that previous time capsules that have a stronger relationship with the current time capsule contain more valuable contextual information for encoding the current speech frame. The gate mechanism controls the information flow from previous to current time capsules by filtering information from previous time capsules that is less related to the current time capsule. Relationships between consecutive capsules are calculated using scaled dot-product operations with the consideration of an architectural characteristic of CapsNets where the input vectors have multi-dimensional representations for each dimension of a speech frame. Each capsule is transformed into an input for the gate mechanism. We applied the attention gate mechanism to the SRF models [16]. The additional number of parameters depends solely on the depth of the capsules and the number of attention heads, as the learnable parameters for the transformation are shared across capsules. Compared to existing SRF models, the proposed method not

* Corresponding author.

10.21437/Interspeech.2023-106

only achieved higher accuracy with the same algorithmic delay, but it also successfully reduced the number of look-ahead frames by almost half without degrading their accuracy.

To facilitate understanding of the proposed method, we first describe a CapsNet-based speech recognition system as a preliminary in Section 2. An explanation of the proposed gate mechanism follows in Section 3. The method is evaluated using the Linguistic Data Consortium (LDC) corpora in Section 4. We discuss the proposed method and its evaluation results in Section 5, and the paper is concluded in Section 6.

2. CapsNet-based speech recognition

2.1. Capsule network

CapsNets [10, 11] use a training mechanism called routing-by-agreement to learn to transfer information between capsule levels. Routing-by-agreement filters information based on a relationship intensity known as *agreements* between lower-level (C_l) and higher-level (C_{l+1}) capsules, where the range of l is either $[0, L]$ for capsule levels or $[1, L]$ for capsule layers.

DR [10] is a popular routing-by-agreement method that uses lengths of instantiation parameter vectors as activations [14, 17, 18]. To improve readability, we use u_i and o_j to denote the lower- and higher-level parameter vectors, respectively, where i and j are the capsule indices on each level. A relationship between u_i and o_j is represented by a prediction vector $\hat{u}_{j|i}$ that is calculated using a transformation matrix W_{ij} as follows:

$$\hat{u}_{j|i} = W_{ij} \times u_i. \tag{1}$$

A routing coefficient r_{ij} is zero-initialized prior to routing iterations. During each iteration, r_{ij} is first normalized to a coupling coefficient c_{ij} through a softmax function:

$$c_{ij} = \frac{\exp(r_{ij})}{\sum_{h=1}^{H_{l+1}} \exp(r_{ih})}, \tag{2}$$

where H_{l+1} represents the number of capsules on the $(l+1)$-th level. Subsequently, all $\hat{u}_{j|i}$ is scaled by corresponding c_{ij} and summed up for each j-th capsule; i.e., $s_j = \sum_i c_{ij}\hat{u}_{j|i}$. To ensure that the length of o_j falls within a valid range of probabilities, s_j is normalized using a squash function:

$$o_j = \frac{||s_j||^2}{1 + ||s_j||^2} \frac{s_j}{||s_j||}. \tag{3}$$

Finally, r_{ij} is updated using the newly computed o_j by a dot product between $\hat{u}_{j|i}$ and o_j. o_j is returned following the iteration.

2.2. Sequential routing framework

SRF [16] is an all-capsule speech recognition framework designed under the assumption that in continuous speech signals, the immediate prior time frame is similar to the current time frame. An SRF model is trained by computing an *agreement* between the past and current time capsules. Therefore, an SRF model can encode a speech frame in a non-iterative manner when the number of iterations Λ is set to 1. This is because routing between capsules iterates $t-1$ times to compute the t-th capsule C^t.

When it comes to the architecture of SRF, capsulation blocks consisting of convolutional layers convert an input speech sequence $x' \in \mathbb{R}^{T' \times F'}$ into a primary capsule group C_0. This capsule group consists of an activation group $A_0 \in \mathbb{R}^{T \times P_H}$ and a corresponding instantiation parameter group $U_0 \in \mathbb{R}^{T \times P_H \times P_D}$. In each layer l, C_l is sliced by the window size and encoded into C_{l+1}. ω_L and ω_R indicate the size of the window on the left (past) and right (future) sides, respectively. To encode the current time frame, $L \times \omega_R$ future capsules on C_0 are required. W_{ij} is shared across all window slices. In this research, we set the stride to 1 to ensure that the sequence length T is the same for both C_0 and C_L.

Sequential dynamic routing (SDR) [16] is a routing algorithm where SRF is applied to DR [10]. A key distinction between SDR and DR is that the first expectation step of SDR is performed by computing r_{ij} as a dot product of the previous time output vector o_j^{t-1} and the current time prediction vector $\hat{u}_{j|i}^t$. SDR models with balanced window settings ($\omega_R = \omega_L$) have demonstrated higher accuracy than those with unbalanced window settings.

3. Gated sequential routing

3.1. Applying attention gates to sequential routing

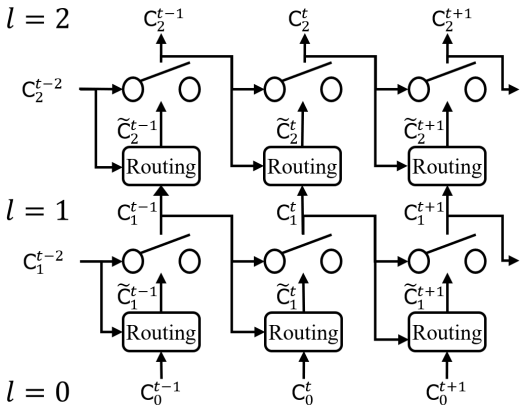


Figure 1: *A schematic of a gated sequential routing mechanism.*

The gate mechanism is applied to a sequential routing algorithm, as depicted in Figure 1. As in the original sequential routing algorithm, C_l^{t-1} and C_{l-1}^t are input into the l-th capsule layer to perform the first expectation step. In addition, C_l^{t-1} is combined with the current routing outputs using a gate mechanism to compute C_l^t. We implemented the gate mechanism using regular MHA to filter information from previous routing outputs. In this mechanism, C_l^{t-1} is used as a key and value, and a candidate current time capsule $\tilde{\mathsf{C}}_l^t$ is used as a query to assign a higher attention probability to C_l^{t-1} which has a stronger relationship with $\tilde{\mathsf{C}}_l^t$. To avoid redundant normalization, C_l^{t-1} is compared to $\tilde{\mathsf{C}}_l^t$ rather than C_l^t. Finally, C_l^t is calculated by normalizing the summation of an MHA output and $\tilde{\mathsf{C}}_l^t$ for A_l^t to have valid probabilities.

3.2. Gated sequential dynamic routing

The attention-based gate mechanism can be applied to SDR as explained in Algorithm 1. In addition to inputs for the original SDR algorithm, the number of attention heads $\mathcal{H}$ is input into a gated sequential dynamic routing (GSDR) as a new hyperparameter in line 1. The distinctive procedure of an attention gate mechanism is described in lines 9–21. As in line 9, the mechanism is applied at the last routing iteration. A concatenated attention head $\hat{h}$ is then declared as an empty vector in

Algorithm 1 Gated Sequential Dynamic Routing (GSDR) (line 7, expectation step; line 9, maximization step)

1: **procedure** GSDR($o^{t-1}, \hat{u}^t_{j|i}, \Lambda, l, \mathcal{H}$)
2: for all i on level l and j on level $(l+1)$: $r_{ij} \leftarrow 0$
3: for all j on level $(l+1)$: $o^t_j \leftarrow o^{t-1}_j$
4: **for** $\lambda = 1$ to Λ **do**
5: for all i on level l and j on level $(l+1)$:
6: $r_{ij} \leftarrow r_{ij} + \hat{u}^t_{j|i} \cdot o^t_j$
7: for all i on level l: $c_i \leftarrow \text{softmax}(r_i)$ ▷ (2)
8: for all j on level $(l+1)$: $s_j \leftarrow \sum_i c_{ij} \hat{u}^t_{j|i}$
9: **if** λ is Λ **then**
10: $\hat{h} \leftarrow ()$
11: **for** all j on level $(l+1)$ **do**
12: **for** $\eta = 1$ to $\mathcal{H}$ **do**
13: $k_\eta \leftarrow o^{t-1}_j \times W^k_\eta$
14: $v_\eta \leftarrow o^{t-1}_j \times W^v_\eta$
15: $q_\eta \leftarrow s_j \times W^q_\eta$
16: $h_\eta \leftarrow \text{softmax}(\frac{q_\eta \cdot k^\mathsf{T}_\eta}{\sqrt{|o^t_j|}}) v_\eta$ ▷ (2)
17: $\hat{h} \leftarrow (\hat{h}, h_\eta)$
18: **end for**
19: $s_j \leftarrow s_j + \hat{h} \times W_\mathcal{H}$
20: **end for**
21: **end if**
22: for all j on level $(l+1)$: $o^t_j \leftarrow \text{squash}(s_j)$ ▷ (3)
23: **end for**
24: return o^t
25: **end procedure**

line 10. For the η-th attention head, the three vectors with the same dimensions that are a key k_η, value v_η, and query q_η are computed in lines 13, 14, and 15, respectively. Accordingly, the three transformation matrices (i.e., W^k_η, W^v_η, and W^q_η) are in $\mathbb{R}^{|o^t| \times |k_\eta|}$. In line 16, attention energies are computed by a scaled dot-product operation between q_η and k_η and normalized into valid probabilities using (2). Then, v_η is multiplied by the attention probabilities. The outputs of attention heads are concatenated with $\hat{h}$ in line 17. Then s_j is updated by accumulating a projected $\hat{h}$ in line 19, and the gate mechanism is completed in line 21. The projection matrix $W_\mathcal{H}$ is in $\mathbb{R}^{\mathcal{H}|k_\eta| \times |o^t|}$.

4. Experiments

4.1. Experimental setup

The numbers of primary and intermediate capsules were set to 60 and 30, respectively, and the number of last capsules was set to the vocabulary size. We set $|k_\eta|$ to $|o^t|/\mathcal{H}$ in the GSDR models such that the number of additional parameters for the gate mechanism was the same regardless of $\mathcal{H}$. We set Λ to 1, and learnable variables were initialized with the *fan-ave* method [19] using a uniform distribution with a scaling factor of 1.0. As a regularization method, every capsule layer was followed by dropout layers [20] at a rate of 0.2. Capsulation blocks were constructed with two convolutional layers computed with 128 filters, followed by a linear projection layer consisting of two neurons per primary capsule. At the top of the capsulation block, another convolutional layer with filters twice the depth of the primary capsules expands the dimension of the projected vectors. Every layer in capsulation was activated with maxout [21], composed of two piecewise linear functions with a dropout rate of 0.2. The filter size of all convolutional layers was 3×3. The first two convolutional layers had a stride of 2, whereas the last convolutional layer had a stride of 1. We used an Adam [22] optimizer with the learning scheduler such that:

$$\text{Learning Rate} = \kappa \cdot \min(N_s^{-0.5}, N_s \times N_w^{-1.5}), \quad (4)$$

where κ is a scaling factor. N_s and N_w indicates the current step and the warming-up step, respectively. We set the beamwidth to 100 for CTC beam search decoding. All experiments were performed on NVIDIA™ RTX3090.

All speech corpora consist of 16-bit mono-channel read speech sampled at 16 kHz. The TIMIT corpus [23] consists of 6,300 utterances (training 4,620; test 1,680) recorded from 630 speakers. We used 3,696 utterances as our training set. Dialect utterances were tagged as "SA" and excluded. A total of 400 and 192 utterances from the test set were used as our validation and test sets. The dictionary consists of 61 phonemes and 2 special symbols that denote padding and a blank. We used 39 labels [24] mapped from the phoneme labels for phoneme error rate (PER) evaluations. For the *Wall Street Journal* (WSJ) corpus [25, 26], we used the *si284* data set containing 81 hours of training speech corpus (37,416 utterances) and used *dev-93* (503 utterances, 1.1 hours) and *eval-92* (333 utterances, 0.7 hours) for validation and evaluation, respectively. Transcription text consists of 32 labels, including letters of the English alphabet and 6 special symbols for padding, spacing, end-of-sentence (EOS), apostrophe, blank, and noise tagging. We extracted 40-dimensional filterbanks and their energies from the speech signals by setting hop and window size to 10 and 20 ms, respectively. The velocity and acceleration were appended by setting delta-window to 2, i.e., F' was set to 123. We normalized the speech features per speaker to have a zero mean and unit variance. The speech corpus pre-processing (i.e., data splitting and feature extraction) was performed using Kaldi[1] [27]. We used a version 2.5 of Tensorflow[2] [28] to implement and evaluate our models.

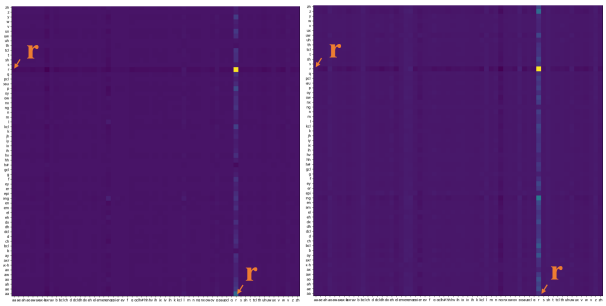
4.2. Results

In this section, we compare the proposed method with our implementation of the SDR models that exhibited the best accuracy in [16]. For evaluations on the TIMIT corpus [23], we set L to 7 and capsule depth to 8 for the models to have approximately 1.9 million parameters. This number of parameters increased by approximately 2,000 when the proposed gate mechanism was applied. Approximately 5,340 frames were batched for a training step. We first set $\kappa = 0.2$ and $N_w = 1200$ then reduced the κ to 0.05 at about a step 40,000 according to the CTC loss on *Valid*. Accordingly the learning rate was increased up to 0.0057 then decreased to 0.0002. We trained the models for a total of 41,800 steps on a graphics processing unit (GPU). Training the existing SDR and the GSDR models took 10 and 22 hours, respectively. By applying the gate mechanism to SDR, the average time required to decode *Valid* and *Test* has increased from 92 to 115 seconds and from 57 to 73 seconds, respectively. As observed from Table 1, PERs were increased by reducing ω_R from 1 to 0. In the unbalanced window settings, the GSDR models with the ω setting "2-0" (GSDR-ω20) have shown lower PERs for the test set compared with SDR-ω20 for every attention head setting. We have compared the first attention heads in the top capsule layers of GSDR-ω11-$\mathcal{H}$2 and GSDR-ω20-$\mathcal{H}$2 as depicted in Figure 2. The utterance

[1] https://github.com/kaldi-asr/kaldi.git
[2] https://github.com/tensorflow/tensorflow/tree/r2.5

Table 1: *Phoneme error rates (PERs) of sequential dynamic routing (SDR) models depending on window (ω_L, ω_R) settings, applications of the gate mechanism, and the number of attention heads on the TIMIT corpus [23] (LA: look-ahead)*

Model	Head	LA frame	Delay (ms)	PER(%) *Valid*	PER(%) *Test*
SDR-ω11	—	39	405	15.7	17.5
SDR-ω20	—	11	125	17.5	19.0
GSDR-ω11	1	39	405	16.1	18.3
GSDR-ω20	1	11	125	17.5	18.9
GSDR-ω11	2	39	405	15.8	17.4
GSDR-ω20	2	11	125	16.7	18.4
GSDR-ω11	4	39	405	16.3	17.8
GSDR-ω20	4	11	125	16.8	18.8



(a) *The 27th capsule group,* ω11 (b) *The 29th capsule group,* ω20

Figure 2: *The first attention maps of top layers in the GSDR models with different settings of window widths* ω*. The horizontal- and vertical-axis indicate corresponding output label indices (number of attention heads* $\mathcal{H} = 2$*).*

(ID: fdhc0-si1559) is 3.4 seconds long. A strong relationship was observed between the output capsule of "r" (994–1,054 ms) and all output capsules of GSDR-ω11-$\mathcal{H}$2 and GSDR-ω20-$\mathcal{H}$2 when the 27th (1080 ms) and 29th (1160 ms) input capsules, respectively, were encoded. In addition, we noted the strongest relationship between capsules representing the same label "r."

For the WSJ corpus [25, 26], we used a training batch consisting of approximately 27,980 frames. The total step was 83,200 and N_w was set to 15,000. κ was set to 0.1 first and decreased to 0.01 at the about 73,000th step according to the CTC loss on *dev-93*. Thus, the learning rate was up to 0.0008 then reduced to 3.5e-5. All SDR models in Table 2 contain approximately 21 million parameters, and their L and $\mathcal{H}$ were set to 10 and 2, respectively. The increase in parameters upon application of the gate mechanism was approximately 20,000. The SDR and GSDR models were trained on 2 GPUs during 2.6 and 5.4 days, respectively. The average decoding times for each test case increased by almost twice, from 243 to 457 seconds for *dev-93*, and from 171 to 363 seconds for *eval-92* when the gate mechanism was applied to SDR. GSDR-ω31 showed approximately 1% lower WERs compared with SDR-ω31 for both evaluation sets. GSDR-ω22 yielded the most accurate results in WERs. We also evaluated models with the window setting of ω11 to verify the recognition rates of another balanced window setting. GSDR-ω11 has shown similar WERs compared with SDR-ω22 at 1.76× lower delay, i.e., an algorithmic delay reduction from 925 to 525 ms.

Table 2: *Word error rates (WERs) of sequential dynamic routing (SDR) models depending on the depth of capsules, window (ω_L, ω_R) settings, and applications of the gate mechanism on the WSJ corpus [25, 26] (LA: look-ahead)*

Model	Depth	LA frame	Delay (ms)	WER(%) *dev-93*	WER(%) *eval-92*
SDR-ω22	20	91	925	21.3	16.9
SDR-ω31	20	51	525	23.3	17.9
SDR-ω11	26	51	525	24.1	18.6
GSDR-ω22	20	91	925	20.7	16.4
GSDR-ω31	20	51	525	22.2	17.6
GSDR-ω11	26	51	525	21.3	16.6

5. Discussion

The proposed method exhibits the highest accuracy when $\mathcal{H} = 2$ for the TIMIT corpus. This can be attributed to a decrease in both the variety of attention heads when $\mathcal{H} = 1$, and the dimensionality of vectors for each attention head when $\mathcal{H} = 4$. A potential future research direction for GSDR may involve investigating whether the increased delay in alignments within the attention map of GSDR-ω20-$\mathcal{H}$2 compared to GSDR-ω11-$\mathcal{H}$2 contributes to a degradation in recognition accuracy.

Although GSDR produced better performance than SDR with the same latency settings in all cases of WSJ, it required approximately twice the decoding time. Because this study was conducted to reduce the delay of SDR, the optimization of decoding speed for MHA calculations within each frame is beyond its scope.We think that GSDR can be further improved by reducing the computational burden. The recognition accuracy of GSDR falls far short of the state-of-the-art performances listed on the leader board[3]: PER 12.9% [29] for *Test* of TIMIT and WER 2.9% [30] for *eval-92* of WSJ. However, we still believe that our method yielded promising recognition accuracy as a singular structure requiring limited future inputs. Furthermore, a GSDR model can be seen as the acoustic component of an NSR model, such as the RNN-T model [1]. Therefore, there may be room to improve accuracy by integrating GSDR models within NSR models.

6. Conclusions

In this study, we have introduced a new method that incorporates a gated recurrent mechanism into the CapsNet architecture to fully utilize the past contextual information in the CapsNet-only ASR systems. The gate mechanism was implemented to control information flow between frame-wise adjacent capsules via MHA. By applying the mechanism to SDR, we could reduce algorithmic delay in character-level speech recognition by almost half while maintaining WERs on par with those of the unmodified SDR algorithm.

7. Acknowledgements

The proposed attention gate mechanism is an extension of Section 4.1.3 and 4.2.2 of "Toward streaming large vocabulary continuous speech recognition based on neural networks" [31] which is the Ph.D. dissertation of Dr. Kyungmin Lee (k.m.lee@samsung.com) under the supervision of Prof. Hong-Gee Kim (hgkim@snu.ac.kr).

[3]https://github.com/syhw/wer_are_we

8. References

[1] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.

[2] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[3] K. Kim*, K. Lee*, D. Gowda, J. Park, S. Kim, S. Jin, Y.-Y. Lee, J. Yeo, D. Kim, S. Jung, J. Lee, M. Han, and C. Kim, "Attention based on-device streaming speech recognition with large speech corpus," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 956–963.

[4] 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*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376.

[5] 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.

[6] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7829–7833.

[7] E. Tsunoo, Y. Kashiwagi, T. Kumakura, and S. Watanabe, "Transformer asr with contextual block processing," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 427–433.

[8] 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," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.

[9] B. Li, A. Gulati, J. Yu, T. N. Sainath, C.-C. Chiu, A. Narayanan, S.-Y. Chang, R. Pang, Y. He, J. Qin, W. Han, Q. Liang, Y. Zhang, T. Strohman, and Y. Wu, "A better and faster end-to-end model for streaming asr," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5634–5638.

[10] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, 2017, pp. 3856–3866.

[11] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[12] T. Hahn, M. Pyeon, and G. Kim, "Self-routing capsule networks," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, 2019, pp. 7656–7665.

[13] Y.-H. H. Tsai, N. Srivastava, H. Goh, and R. Salakhutdinov, "Capsules with inverted dot-product attention routing," in *International Conference on Learning Representations*, 2020.

[14] J. Bae and D. Kim, "End-to-end speech command recognition with capsule network," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*. ISCA, 2018, pp. 776–780.

[15] H. Jayasekara, V. Jayasundara, J. Rajasegaran, S. Jayasekara, S. Seneviratne, and R. Rodrigo, "Timecaps: Capturing time series data with capsule networks," *ArXiv*, vol. abs/1911.11800, 2019.

[16] K. Lee, H. Joe, H. Lim, K. Kim, S. Kim, C. W. Han, and H.-G. Kim, "Sequential routing framework: Fully capsule network-based speech recognition," *Computer Speech and Language*, vol. 70, p. 101228, 2021.

[17] R. LaLonde and U. Bagci, "Capsules for object segmentation," *CoRR*, vol. abs/1804.04241, 2018.

[18] M. Kwabena Patrick, A. Felix Adekoya, A. Abra Mighty, and B. Y. Edward, "Capsule networks – a survey," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 1, p. 1295–1310, jan 2022.

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, ser. JMLR Proceedings, vol. 9. JMLR.org, 2010, pp. 249–256.

[20] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[21] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio, "Maxout networks," *CoRR*, vol. abs/1302.4389, 2013.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[23] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "Darpa timit acoustic phonetic continuous speech corpus cdrom," 1993. [Online]. Available: https://catalog.ldc.upenn.edu/LDC93s1

[24] K. Lee and H. Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 11, pp. 1641–1648, 1989.

[25] J. S. Garofolo, D. Graff, D. Paul, and D. Pallett, "Csr-i (wsj0) complete," 1993. [Online]. Available: https://catalog.ldc.upenn.edu/LDC93S6A

[26] L. D. Consortium and N. M. I. Group, "Csr-ii (wsj1) complete," 1994. [Online]. Available: https://catalog.ldc.upenn.edu/LDC94S13A

[27] 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," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[28] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[29] S. Nayak, C. S. Kumar, and K. S. R. Murty, "Instantaneous frequency filter-bank features for low resource speech recognition using deep recurrent architectures," in *2021 National Conference on Communications (NCC)*, 2021, pp. 1–6.

[30] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "End-to-end speech recognition using lattice-free MMI," in *Proc. Interspeech 2018*, 2018, pp. 12–16.

[31] K. Lee, "Toward streaming large vocabulary continuous speech recognition based on neural networks," Ph.D. dissertation, Seoul National University, 2022.