INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Continual Learning in Automatic Speech Recognition

Samik Sadhu[1], *Hynek Hermansky*[1,2]

[1]Center for Language and Speech Processing, Johns Hopkins University, USA
[2]Human Language Technology Center of Excellence, Johns Hopkins University, USA

samiksadhu@jhu.edu, hynek@jhu.edu

Abstract

We emulate continual learning observed in real life, where new training data, which represent new application domain, are used for gradual improvement of an Automatic Speech Recognizer(ASR) trained on old domains. The data on which the original classifier was trained is no longer required and we observe no loss of performance on the original domain. Further, on previously unseen domain, our technique appears to yield slight advantage over offline multi-condition training. The proposed learning technique is consistent with our previously studied *ad hoc* stream attention based multi-stream ASR.

Index Terms: Continual Learning, Lifelong Learning, Automatic Speech Recognition, Multi-stream ASR

1. Introduction

Most ASR systems use a neural network classifier trained on labeled acoustic data that estimates posterior probability of speech sounds from incoming acoustic signal. Typical learning requires training data from all application domains, which are expected to be encountered during the ASR deployment. When new application domains are encountered after deployment, ASRs typically generalize poorly. An instance of this can be seen in table 1. Both Wall Street Journal (WSJ) and Librispeech (sec. 5) consist of clean read speech data, however an ASR trained with one data does not generalize well to the other domain.

Table 1: *WER % on out-of-domain data sets. Both the acoustic and language models are trained on each individual data domain.*

Trained On	Tested On	
	WSJ	**Librispeech**
WSJ	13.2	43.2
Librispeech	28.5	21.0

An offline approach to overcome this problem is to accumulate the training data from new domains together with the original training data, and re-train the whole classifier using multi-task[1, 2], multi-condition[3, 4] or data-augmentation[5, 6] techniques. However, this is resource intensive and inconsistent with the natural online learning process observed in humans which leads to a gradual improvement in performance over old as well as future unknown data domains. Additionally, for data privacy as well as practical space constraints, it may not be possible to save all old training data for future re-training stages. In such a case, training data from the new domain could be used for *adaptation* of a part of the existing classifier, however, leading to some loss in performance on the old domains - a phenomenon known as *catastrophic forgetting*[7, 8, 9].

2. Continual Learning

Recent works on Continual Learning (CL) [10, 11, 12] aim at emulating the gradual learning observed in humans using machines by utilizing a continuous stream of data over a period of time to learn and share knowledge over different data domains. Training on new application domain should enhance the performance on all past domains (backward transfer) and on all future domains to come (forward transfer). To overcome catastrophic forgetting, some of the proposed CL models use network expansion techniques to add more parameters for new training data [13, 14, 15], while others use various regularization methods [16, 17] or memory replay techniques [18, 19] to prevent old knowledge from getting overwritten by new training data. In this paper, we propose a dynamically expanding model for continual learning.

3. Continual Learning in HMM-DNN ASR

In a HMM-DNN ASR[20], a Deep Neural Network (DNN) computes frame level likelihoods for an utterance of feature vectors $X = (x_1, x_2, \dots x_M)$, which are subsequently used by a Hidden Markov Model (HMM) to compute the likelihood of the utterance X. If W be all possible sequences of words, an ASR computes the best path sequence $W^\star$ as

$$W^\star = \arg\max_W p(X|W)P(W) \tag{1}$$

where $p(X|W)$ is obtained from the *acoustic model* consisting of the HMM-DNN combination and $P(W)$ is obtained from a n-gram *language model*. We use the notation $p(*)$ to denote likelihoods and $P(*)$ for probability. For data D, the DNN is trained to classify HMM-states c for every feature vector x. The posterior probabilities $p(c|x, D)$ are then divided by the prior probability of states $p(c|D)$ to obtain *pseudo* likelihoods $p(x|c, D)$ to be processed by the HMM.

The likelihood generated by the DNN is dependent on its training data D. Apart from x and c, we define the random variable $D \in \{1, 2, \dots N\}$ where N is a parameter denoting the number of training data domains. For brevity of notation, we denote the assignment $D = k$ as D_k. We would like to have a continual update of the likelihood estimation model $p(x|c, N)$ for $N \in \{1, 2, 3, \dots \infty\}$ without repeated retraining for every N.

The main idea is to factorize the likelihood model $p(x|c, N)$ into sub-models specific for each data domain D. We write $p(x|c, N)$ as the marginal over the joint distribution between x and D as

$$p(x|c, N) = \sum_{k=1}^{N} p(x, D_k|c) = \sum_{k=1}^{N} \frac{p(x, D_k, c)}{P(c)} \tag{2}$$

http://dx.doi.org/10.21437/Interspeech.2020-2962

The numerator term in eq. 2 can be broken down by chain rule as

$$p(x, D_k, c) = \quad P(c|D_k, x)p(x|D_k)P(D_k) \tag{3}$$

$P(c)$ can also be expressed as a marginal of the join distribution $P(c, D)$ as shown in eq. 4

$$P(c) = \sum_{j=1}^{N} P(c, D_j) \tag{4}$$

where $p(c, D_j)$ can be broken down by chain rule to obtain

$$P(c, D_j) = P(c|D_j)P(D_j) \tag{5}$$

Substituting eq. 3,4,5 into eq. 2, we get

$$p(x|c, N) = \sum_{k=1}^{N} \frac{P(c|D_k, x)p(x|D_k)P(D_k)}{\sum_{j=1}^{N} P(c|D_j)P(D_j)} \tag{6}$$

with the constraint $\sum_{k=1}^{N} P(D_k) = 1$.

This decomposition enables individual training of respective models on each domain. Only the acoustic state models $p(c|D_k, x)$, the probability of the data $P(x|D_k)$, the prior probability of classes $P(c|D_k)$ need to be retained for the subsequent likelihood computation. The training data for the particular domain are not required anymore.We propose the following continual model update algorithm.

Algorithm 1: Updating model $p(x|c, N)$

for *N=1* $\rightarrow \infty$ **do**
- *Train model* $P(c|D_N, x)$*; save model;*
- *Train model* $p(x|D_N)$*; save model;*
- *Save* $P(c|D_N)$*;*
- *Update likelihood model to* $p(x|c, N) = \sum_{k=1}^{N} \frac{P(c|D_k, x)p(x|D_k)P(D_k)}{\sum_{j=1}^{N} P(c|D_j)P(D_j)}$*;*

end

In the following subsections, we describe how to model each term in eq. 6 .

3.1. Computing $P(c|D_k, x)$

We generate HMM state labels by forced aligning data $D = k$ with an HMM-GMM ASR model[21]. This labelled data is then used to discriminatively train a neural network classifier to generate the posterior probability over the HMM states c given the input data x.

3.2. Computing $P(c|D_k)$

The prior probability of the HMM states for data $D = k$ is obtained by counting each state label from the forced alignment followed by normalization to obtain probability values in the range $[0, 1]$

3.3. Computing $p(x|D_k)$

The likelihood of x conditioned on training data $D = k$ is approximated by the variational lower bound from a Variational Autoencoder(VAE) [22]. The VAE is a generative neural network model with an inference network to map the data x to a distribution over the latent variable z with the assumption of the latent prior to be a centered multivariate isotropic Gaussian. Subsequently a generation network maps the latent variable z back to the feature vector x. A variational lower bound to the data likelihood $p(x|D_k)$ is maximized for training the neural network.

3.4. Computing $P(D_k)$

The priors reflect our understanding of the the relative occurrence of different data domains. As default, we start with assigning equal probabilities to all domains of the training data

$$p(D_k) = \frac{1}{N} \; \forall k \in \{1, 2, \ldots N\} \tag{7}$$

3.5. Relation to multi-stream ASR

Multi-stream systems[23, 24, 25] combine the outputs of different neural models using a stream attention module which determines the relative confidence of each stream. We show that $p(x|c, N)$ can be a interpreted as a multi-stream likelihood combination model where the performance of each stream D_k is evaluated independently and used as a weight in the linear fusion of likelihoods from the streams. $p(x, D_k, c)$ can be decomposed using chain rule as

$$p(x, D_k, c) = p(x|D_k, c)P(c|D_k)P(D_k) \tag{8}$$

Substituting eq. 8 into eq. 2, we get

$$p(x|c, N) = \sum_{k=1}^{N} p(x|D_k, c) \frac{P(c|D_k)}{P(c)} P(D_k) \tag{9}$$

$p(x|c, N)$ combines the likelihoods $p(x|D, c)$ using the fixed attention weights $p(D)$. This interpretation gives us the flexibility to replace $p(D)$ with a stream attention module like in a typical multi-stream scenario.

We would like to attend to the stream(s) or domain(s) that are the likely source(s) for the test utterance X. The average likelihood $p(X|D_k)$ over the utterance X obtained from the VAE determines how likely X is to have been obtained from domain D_k. We normalize these likelihood values from N VAEs by passing them through a softmax function as in eq. 10

$$\tilde{P}(D_k) = \frac{\exp(p(X|D_k))}{\sum_{v=1}^{N} \exp(p(X|D_v))} \tag{10}$$

Accordingly, the likelihood estimation eq. 6 can be modified to

$$p(x|c, N) = \sum_{k=1}^{N} \frac{P(c|D_k, x)p(x|D_k)\tilde{P}(D_k)}{\sum_{j=1}^{N} P(c|D_j)\tilde{P}(D_k)} \tag{11}$$

, where $\tilde{P}(D_k)$ is updated for every utterance X. The frame wise likelihoods generated by $p(x|c, N)$ can eventually be processed by the HMM and the best hypothesis word sequence $W^\star$ obtained as in eq. 1.

4. Performance Measures

4.1. ASR performance

ASR performance is determined by Word Error Rate (WER) - the percentage minimum edit distance between the true transcription of a test utterance and the best hypothesis from the ASR system.

4.2. CL performance

With N different training data obtained over a period of time, consider $W_{i,j}$ to be the WER of the proposed system on domain j after our CL system has been trained up to domain i. The CL performance measures are defined as follows

- **Average Error:**
  $\mathcal{A} = \frac{1}{N}\sum_{i=1}^{N} W_{N,i}$
- **Forward Transfer:**
  $\mathcal{F}[i] = R_i - W_{i-1,i}\ \forall i \in \{2, \dots N\}$
- **Backward Transfer:**
  $\mathcal{B}[i] = \frac{1}{i-1}\sum_{j=1}^{i-1}(W_{j,j} - W_{i,j})\ \forall i \in \{2, \dots N\}$

Here, $\mathcal{F}[i]$ determines the transfer of knowledge to domain i by all the domains observed before i and R_i denotes the WER on data i for a system without any training. Practically all ASR systems have a large vocabulary and thus R_i is assumed to be $= 100$. $\mathcal{B}[i]$ is the transfer of knowledge by domain i to all the domains before it. Both $\mathcal{F}[i]$ and $\mathcal{B}[i]$ are measured as reduction in WER.

5. Data sets

We test our system with a sequence of $N = 4$ transcribed speech data sets. The details are shown in table 2.

Table 2: *Data sets for CL evaluation*

Data set	**Training**	**Test**
WSJ (Data 1)	*train_si284* ($\sim$80 hrs) clean read speech	*test_eval92*(T_1)
Reverb (Data 2)	*tr_simu_8ch*($\sim$120 hrs) simluated reverberation	*et_simu_1ch*(T_2) *et_real_1ch*(M_2)
Librispeech (Data 3)	*train_clean_360*($\sim$360 hrs) clean read speech	*test_clean*(T_3) *test_other*(M_3)
Chime4 (Data 4)	*et05_multi_noisy*($\sim$110 hrs) simulated+ real noisy	*et05_simu_noisy*(T_4)

It is to be noted that the Chime4 data is the combined data from all channels. The test sets are divided into two groups, the matched test set for each data domain T_1, T_2, T_3, T_4 and additional test sets M_2, M_3 which have some inherent mismatches to all the data in table 2. The *test_other*(M_2) set in Librispeech consists of high WER speakers [26], while the Reverb test set *et_real_1ch*(M_3) consists of *real* reverberated speech.

6. System Details

We use 13 dimensional MFCC features with 4 frames of splicing on either side with *utterance wise* mean and variance normalization. We use a universal set of 38 phonemes, 3-state HMMs are defined for every context dependant phoneme and the number of context dependant HMM states are fixed at 3376 by state-tying. The HMM state alignments on the training data are obtained using forced alignment with HMM-GMM models trained with Kaldi [27] for each individual task. The language model is a tri-gram language model trained with Data 1 transcriptions (the language model is not continually learnt). The decoding graph for the CL system is obtained by composing H, C, L and G fsts for Data 1 generated with Kaldi. Our system also excludes all pre-processing steps (beamforming, denoising, de-reverberation etc.) as well as any speaker specific normalization, lattice re-scoring etc.

The neural network classifiers generating the HMM state posterior distributions are 4 layer Gated Recurrent Unit (GRU) models with 300 hidden units trained with PyTorch [28] using cross-entropy loss and the Adam optimization technique [29]. We use an initial learning rate of 0.001 and a learning rate reduction by 50% when the validation error is seen to be non-decreasing.

The VAE models have single encoder and decoder GRU layers with a hidden dimension of 300 and latent dimension of 100. They are trained with the same optimization strategy using PyTorch.

Stand-alone system (SS): Apart from the CL systems, we also train individual single condition HMM-DNN ASR systems on each individual data sets enlisted in table 2.

Multi-condition system (MS): MS is the offline HMM-DNN ASR model which is trained by pooling together all the training data enlisted in table 2. To compensate for the increased amount of training data, we add an extra GRU layer with 300 nodes for the MS neural network.

We do a comparative analysis of the continual learning system with the multi-condition and stand-alone systems in section 7.

7. Results

We evaluate the performance of the continually updated likelihood estimation models $p(x|c, N)$ for $N \in \{1, 2, 3, 4\}$ by computing the WER% on the different test sets in table 2 for every N. A comparison of the WER% results with equal prior and stream attention models can be seen in table 3. We observe that the stream attention strategy always outperforms equal prior in terms of average error $\mathcal{A}$ by virtue of its ability to attend to the most likely data domain. For the rest of this section, we will only be looking at the results with the stream attention based likelihood estimation model.

7.1. Knowledge transfer

A more detailed analysis of continual learning can be done by evaluating the forward and backward knowledge transfer measures defined in sec 4.2 (shown in table 4).

Table 4: *Knowledge transfer w.r.t. WER reduction*

	$\mathcal{F}[2]$	$\mathcal{F}[3]$	$\mathcal{F}[4]$	Avg
forward transfer	23.4	57.9	31.3	37.5
	$\mathcal{B}[2]$	$\mathcal{B}[3]$	$\mathcal{B}[4]$	Avg
backward transfer	-0.1	1.8	1.2	1.0

The last data domain (Chime4), shows the clear effect of forward transfer. The model shows a WER reduction of $\mathcal{F}[4] = 31.3$ below random on test set T_4 before the model has been updated with any noisy speech. Similar effects can be seen for the clean data domain Librispeech, which shows a positive WER reduction of $\mathcal{F}[3] = 57.9$ on test set T_3 below random. The backward transfer results in table 4 show that the system does not suffer from catastrophic forgetting i.e., performance on old data domains do not suffer because of new model updates. In fact, there is a positive backward transfer of knowledge to old data domains with an absolute improvement of $\mathcal{B}[4] = 1.2$ WER on test sets T_1, T_2 and T_3 after the system has been updated with all 4 data sets. A clear indication of knowledge retention can also be seen from the learning curve (WER% as a function of N) for the WSJ test set T_1. In

Table 3: *Comparison of continual learning results with uniform prior $P(D)$ and stream attention $\tilde{P}(D)$ in WER %*

	Test Sets									
	test_eval92(T_1)		*et_simu_1ch*(T_2)		*test_clean*(T_3)		*et05_simu_noisy*(T_4)		$\mathcal{A}$	
N	**Equal Prior** $P(D)$	**Stream Attention** $\tilde{P}(D)$	**Equal Prior** $P(D)$	**Stream Attention** $\tilde{P}(D)$	**Equal Prior** $P(D)$	**Stream Attention** $\tilde{P}(D)$	**Equal Prior** $P(D)$	**Stream Attention** $\tilde{P}(D)$	**Equal Prior** $P(D)$	**Stream Attention** $\tilde{P}(D)$
1	13.2		76.6		43.2		79.6		**53.2**	
2	14.1	13.3	31.9	30.4	43.2	42.1	76.3	76.4	41.4	**40.6**
3	11.6	11.8	31.4	28.1	31.3	30.2	68.6	68.7	35.7	**34.7**
4	11.7	11.3	32.5	28.5	32.2	30.4	49.7	46.0	31.5	**29.0**

spite of 3 model updates after WSJ data, the WER% on T_1 improves from 13.2 to 11.3, a 14% WER reduction. This property is an outcome of the dynamic network expansion of the proposed system. Knowledge learnt for one data set is preserved in network weights that remain unchanged for future domains.

7.2. Comparison of continuous learning with stand-alone and multi-condition systems

In table 5, we compare the ASR performance using the likelihood model $p(x|c, N = 4)$ with the individual stand-alone ASR models and a multi-condition ASR model trained with all 4 data sets. It is to be noted that the stand-alone systems are trained and tested on one particular domain and are blind to the existence of other domains.

The multi-condition performance is considered to be the *lower bound* to the CL system error [10]. This is because the multi-condition model is trained with all the data simultaneously and hence can share knowledge efficiently across tasks. This effect can be seen in the WER% improvement of this system over the stand-alone systems in table 5.

Continuous learning systems typically perform worse in comparison to the offline multi-condition model on image tasks [10]. However, for ASR tasks, we observed that the proposed system shows slight advantage with a 3% better WER for clean tasks T_1 and T_3 compared to the multi-condition system. Thus, multi-stream systems hold the potential to generalize better than multi-condition learning, a phenomenon that has been previously studied in similar *ensemble* systems in [30, 31]. On the other hand, in comparison to stand-alone models, the CL system effectively shares knowledge between tasks resulting in, on average, a 8% WER reduction.

Table 5: *Comparison of WER% of different models for $N = 4$*

	T_1	T_2	T_3	T_4	M_2	M_3	Avg
stand-alone	13.2	33.6	29.8	47.9	62.4	53.2	40.0
multi-condition	12.6	**26.1**	30.4	**42.5**	54.3	53.5	36.5
continuous	**11.3**	28.5	**30.4**	46.0	**52.7**	**52.9**	36.9

Additionally, for mismatched tasks M_2, M_3, the CL system shows better generalization capability in comparison to multi-condition learning with a 2% reduction in WER (see table 5). Hence, a continual learning approach gives some advantage for speaker mismatches (M_2) as well as domain mismatched settings (M_3) over multi-condition learning.

On average, multi-condition learning and the continual learning system show very similar performance.

7.3. Ablation Study

We look at the relative contribution of the numerator terms $p(x|D)$ and $\tilde{P}(D)$ in eq. 11 to the final performance of the CL system in table 6, where we eliminate the influence of a term by setting it to an uniform value of $\frac{1}{N}$.

Table 6: *Ablation study of CL model with $N = 4$ (results in WER %)*

	T_1	T_2	T_3	T_4
$p(x\|D_k) = \frac{1}{N}$, $\tilde{P}(D) = \frac{1}{N}$	11.7	32.6	32.3	49.8
$\tilde{P}(D) = \frac{1}{N}$	11.7	32.5	32.2	49.7
$p(x\|D_k) = \frac{1}{N}$	11.4	28.4	30.4	45.9

It can be seen from table 6 that with $p(x|D_k) = \frac{1}{N}$ the likelihood model performance is equivalent to that of the full CL model ("continuous" in table 5). Hence, the stream attention module $\tilde{p}(D)$ is the most dominant term in eq. 11 for likelihood computation.

8. Conclusions

In this paper, we moved beyond database specific ASR systems to a more natural continual learning paradigm for ASR. Our system generalizes well across both old and new data domains eliminating catastrophic forgetting which arises from typical adaptation techniques. Our continuous learning approach also achieves very similar performance to multi-condition training which requires repeated computations and storing all data sets indefinitely.

In contrast to regularization based CL systems or multi-condition training, the main disadvantage of our model is the linear increase in the number of model parameters with the number of available data domain - leading to an increase in overall inference time.

9. Acknowledgement

We would like to thank Sandeep Reddy Kothinti and Ruizhi Li for their valuable inputs. This work was supported by the Center of Excellence in Human Language Technologies, The Johns Hopkins University, and by a gift from Google Research.

10. References

[1] G. Pironkov, S. Dupont, and T. Dutoit, "Multi-task learning for speech recognition: an overview." in *ESANN*, 2016.

[2] Z. Chen, S. Watanabe, H. Erdogan, and J. R. Hershey, "Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[3] R. Lippmann, E. Martin, and D. Paul, "Multi-style training for robust isolated-word speech recognition," in *ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 12. IEEE, 1987, pp. 705–708.

[4] F. Ge, K. Li, B. Wu, S. M. Siniscalchi, Y. Yan, and C.-H. Lee, "Joint training of multi-channel-condition dereverberation and acoustic modeling of microphone array speech for robust distant speech recognition." in *Interspeech*, 2017, pp. 3847–3851.

[5] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Proc. Interspeech 2019*, pp. 2613–2617, 2019.

[6] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1469–1477, 2015.

[7] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[8] R. Ratcliff, "Connectionist models of recognition memory: constraints imposed by learning and forgetting functions." *Psychological review*, vol. 97, no. 2, p. 285, 1990.

[9] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[10] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, 2019.

[11] R. Aljundi, "Continual learning in neural networks," *arXiv preprint arXiv:1910.02718*, 2019.

[12] A. Gepperth and C. Karaoguz, "A bio-inspired incremental learning architecture for applied perceptual problems," *Cognitive Computation*, vol. 8, no. 5, pp. 924–934, 2016.

[13] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[14] J. Yoon, J. Lee, E. Yang, and S. J. Hwang, "Lifelong learning with dynamically expandable network," in *International Conference on Learning Representations*. International Conference on Learning Representations, 2018.

[15] R. Aljundi, J. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," *Proceedings CVPR 2017*, vol. 2017, pp. 3366–3375, 2017.

[16] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[17] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[18] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 6467–6476.

[19] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Advances in Neural Information Processing Systems*, 2017, pp. 2990–2999.

[20] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer Science & Business Media, 2012, vol. 247.

[21] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[22] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[23] H. Hermansky, "Multistream recognition of speech: Dealing with unknown unknowns," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1076–1088, 2013.

[24] X. Wang, R. Li, and H. Hermansky, "Stream attention for distributed multi-microphone speech recognition," *Proc. Interspeech 2018*, pp. 3033–3037, 2018.

[25] S. H. Mallidi and H. Hermansky, "Novel neural network based fusion for multistream asr," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5680–5684.

[26] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[27] P. Daniel, G. Arnab, B. Gilles, B. Lukas, G. Ondrej, G. Nagendra, H. Mirko, M. Petr, Q. Yanmin, S. Petr *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584, 2011.

[28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] N. Ueda and R. Nakano, "Generalization error of ensemble estimators," in *Proceedings of International Conference on Neural Networks (ICNN'96)*, vol. 1, 1996, pp. 90–95 vol.1.

[31] L. Deng and J. C. Platt, "Ensemble deep learning for speech recognition," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.