Interspeech 2022
18-22 September 2022, Incheon, Korea

Improving Generalization of Deep Neural Network Acoustic Models with Length Perturbation and N-best Based Label Smoothing

Xiaodong Cui[1], *George Saon*[1], *Tohru Nagano*[2], *Masayuki Suzuki*[2], *Takashi Fukuda*[2], *Brian Kingsbury*[1], *Gakuto Kurata*[2]

[1]IBM Research AI, IBM T. J. Watson Research Center, USA
[2]IBM Research Tokyo, Japan

{cuix,gsaon,bedk}@us.ibm.com, {tohru3,szuk,fukuda1,gakuto}@jp.ibm.com

Abstract

We introduce two techniques, length perturbation and n-best based label smoothing, to improve generalization of deep neural network (DNN) acoustic models for automatic speech recognition (ASR). Length perturbation is a data augmentation algorithm that randomly drops and inserts frames of an utterance to alter the length of the speech feature sequence. N-best based label smoothing randomly injects noise to ground truth labels during training in order to avoid overfitting, where the noisy labels are generated from n-best hypotheses. We evaluate these two techniques extensively on the 300-hour Switchboard (SWB300) dataset and an in-house 500-hour Japanese (JPN500) dataset using recurrent neural network transducer (RNNT) acoustic models for ASR. We show that both techniques improve the generalization of RNNT models individually and they can also be complementary. In particular, they yield good improvements over a strong SWB300 baseline and give state-of-art performance on SWB300 using RNNT models.

Index Terms: length perturbation, label smoothing, n-best hypotheses, deep neural networks, automatic speech recognition

1. Introduction

Generalization is a fundamental problem in machine learning. In ASR, acoustic models with deep neural network (DNN) architectures may suffer from overfitting due to their huge number of parameters. To make DNN models generalize well, techniques such as model regularization (e.g. ℓ_1-norm or ℓ_2-norm regularization [1] and dropout [2]) and data augmentation [3][4][5][6] have been broadly used in training. In this paper we introduce two techniques to improve generalization of DNN acoustic models in ASR. One is data augmentation based on length perturbation which randomly drops and inserts frames of an utterance to alter the length of the speech feature sequence. The other is label smoothing based on n-best hypotheses.

Length perturbation compresses and stretches the feature sequence of an utterance in the training to provide a perturbed variant of it having a different length. This is conducted by both frame skipping and frame insertion. Frame skipping and frame insertion as separate approaches have been used in the speech community for different purposes. In most cases, frames are skipped in ASR systems, in a fixed or dynamic manner, for reduced processing time in training or decoding [7][8][9][10]. In [11], SpliceOut is proposed to treat frame skipping as a time masking approach to improve generalization of DNN models in various speech recognition and audio classification tasks. In [10], it is observed that DropFrame, despite being aimed at reducing training time, may also help to improve performance of end-to-end models. Analogously, frame insertion or time stretching is a common perturbation technique for speech and audio signals in the other direction [12][13][14]. From the length perturbation perspective, the current application of either frame skipping or frame insertion tends to perturb the length of an utterance biased towards one direction. The length perturbation approach investigated in this paper consists of perturbation both ways.

Label smoothing, first introduced in [15], aims to improve generalization in machine learning by avoiding overconfidence over labels. Although researchers are still trying to gain insights into the working mechanism of label smoothing [16][17], it has been shown to be helpful in a broad variety of machine learning tasks [18][19][20][21]. In its conventional setting, label smoothing is accomplished by smoothing a one-hot label vector with a uniform distribution across all class labels under the cross-entropy loss function. Since ASR is essentially a sequence to sequence mapping problem and the acoustic model of interest in this work is recurrent neural network transducers (RNNTs)[22][23] estimated under the maximum likelihood loss function, we approach label smoothing from a sequence perspective. We choose n-best hypotheses as competing "classes" and use them as "noisy" labels in the training with probability.

ASR experiments are carried out on the 300-hour Switchboard (SWB300) dataset [24][25] which consists of narrowband speech and an in-house 500-hour Japanese (JPN500) dataset which consists of broadband speech to evaluate the two proposed techniques with various configurations. The acoustic models are RNNTs [8][26][27]. We show that both techniques can improve word error rates (WERs) separately on the two datasets. Moreover, the two techniques can be complementary. In particular, when combining length perturbation and n-best label smoothing, we obtain state-of-the-art WERs on the SWB300 dataset using RNNTs.

The remainder of the paper is organized as follows. Section 2 and Section 3 are devoted to the length perturbation and n-best label smoothing, respectively. Experimental results on SWB300 and JPN500 are reported in Section 4. Section 5 concludes the paper with a summary.

2. Length Perturbation

Implementation of the proposed length perturbation is given in Algorithm 1. The length of an utterance is perturbed first by skipping frames and then followed by inserting frames, both with a probability. To skip frames, one randomly samples r_s percentage of frames from the utterance to operate on. For each sampled frame x, t_s consecutive frames are dropped starting from x where t_s is an integer upper bounded by a hyper-parameter T_s. Analogously, to insert frames, one randomly samples r_p percentage of frames from the utterance to operate on. For each

10.21437/Interspeech.2022-821

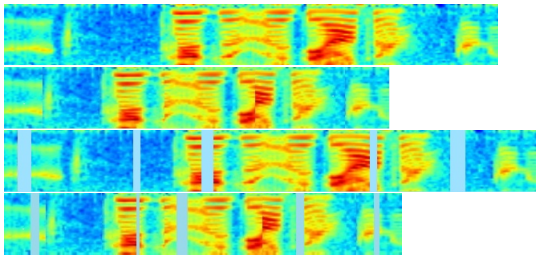

Figure 1: *Illustration of length perturbation. (a) is the logMel spectrum of an utterance. (b) is the logMel spectrum after frame skipping. (c) is the logMel spectrum after frame insertion. (d) is the logMel spectrum after frame skipping followed by frame insertion.*

sampled frame y, t_p consecutive blank frames (zero vectors) are inserted after y where t_p is an integer upper bounded by a hyper-parameter T_p. This is illustrated in Fig.1 where the logMel spectrum of an utterance is demonstrated in Fig.1a. Fig.1b and Fig.1c are its perturbed versions by frame skipping and insertion, respectively. Fig.1d is the overall effect of perturbation after both frame skipping and insertion. All three perturbations are carried out with certain probabilities according to Algorithm 1.

Length perturbation can help in scenarios where there is a mismatch in the length of utterances between training and test conditions. In addition, by randomly dropping frames, it can also perturb the "memory" of a sequence model such as long short-term memory (LSTM) [28] network to avoid simply memorizing the history of the feature sequence in the training. Therefore it can improve the generalization. Furthermore, the applied frame insertion can be viewed as simulating a SpecAug [6] mechanism for an utterance of longer length (Fig.1c). It encourages the system to fill in those inserted frames during training, which is also similar in spirit to the filling in frames (FIF) idea in voice conversion [29].

Algorithm 1 Length Perturbation of Input Utterances

$L \leftarrow$ Total number of utterances
$p_s \leftarrow$ Probability of perturbation by dropping frames;
$r_s \leftarrow$ Percentage of frames to be perturbed in an utterance;
$T_s \leftarrow$ Maximum frames to drop;
$p_p \leftarrow$ Probability of perturbation by inserting frames;
$r_p \leftarrow$ Percentage of frames to be perturbed in an utterance;
$T_p \leftarrow$ Maximum frames to insert;

for $1 \leftarrow 1, \cdots, L$ **do**
  Perturb this utterance by dropping frames with prob. p_s:
    DropFrames(r_s, T_s)
  Perturb this utterance by inserting frames with prob. p_p:
    InsertFrames(r_p, T_p)

function DROPFRAMES(r_s, T_s)
  Sample r_s percentage of frames from the utterance
  for each sampled frame x **do**
    Randomly generate an integer $t_s \in [1, T_s]$
    Drop t_s consecutive frames starting from x

function INSERTFRAMES(r_p, T_p)
  Sample r_p percentage of frames from the utterance
  for each sampled frame y **do**
    Randomly generate an integer $t_p \in [1, T_p]$
    Insert t_p consecutive blank frames after y

this is one this is one of the most highly taxed areas in the country
01 this is one this is one the most highly taxed areas in the country
02 this is one this is one the most highly tax areas in the country
03 this is one this is one the most highly taxed areas and country
04 this one this is one the most highly taxed areas and the country
05 this is one this is one the most highly tax areas and country

Figure 2: *An example of a ground truth label (top line in bold) and its 5 best hypotheses.*

3. N-best Based Label Smoothing

Label smoothing introduces a small amount of noise to ground truth labels to avoid training with over-confidence to help generalization. For a classification problem with cross-entropy loss, labels are typically provided as one-hot vectors. Suppose y is a class label for a sample x and there are K classes in total. Label smoothing smoothes the label with a uniform distribution over K classes weighted by ϵ as shown in Eq.1 where $\mathbf{1}$ is an all-ones vector

$$\tilde{y} = (1-\epsilon) \cdot y + \epsilon \cdot \tfrac{1}{K} \cdot \mathbf{1} \tag{1}$$

Suppose p is the ground truth (one-hot) distribution, q is the distribution to be learned and u is the uniform distribution. Label smoothing amounts to imposing a regularization term $\sum_{i=1}^{n} H_i(u,q)$ to the original cross-entropy term $\sum_{i=1}^{n} H_i(p,q)$ as shown in Eq.2 where $i = 1, \cdots, n$ and n is the total number of samples.

$$\mathcal{L} = (1-\epsilon) \sum_{i=1}^{n} H_i(p,q) + \epsilon \sum_{i=1}^{n} H_i(u,q). \tag{2}$$

Extending the label smoothing setting in Eq.1 to RNNT training under the maximum likelihood loss is not straightforward as the softmax output of RNNT reflects local decisions while the learning is focused on the whole sequence. From a sequence classification perspective, each sequence may represent a class and all sequences over the output space $\mathcal{Y}$ form a countably infinite set of classes in that sequence space. In this light, we may smooth out the ground truth label sequence with a small number of competing label sequences, which motivates us to investigate the label smoothing strategy based on n-best hypotheses.

Let γ be a random variable uniformly distributed in $[0, 1]$ and a constant $\epsilon \in [0, 1]$. Suppose $\boldsymbol{y}$ is a ground truth label sequence and $\Omega_{\boldsymbol{y}|\boldsymbol{x}} = \{\hat{\boldsymbol{y}}_1, \cdots, \hat{\boldsymbol{y}}_K\}$ is the n-best set that consists of K n-best hypotheses of $\boldsymbol{y}$ given $\boldsymbol{x}$. Select uniformly at random from K hypotheses a hypothesis $\hat{\boldsymbol{y}}_i$, $i \in \{1, \cdots, K\}$, and replace the ground truth label with it with probability ϵ. This is given in Eq.3 where $\mathbb{I}(\cdot)$ is the indicator function:

$$\tilde{\boldsymbol{y}} = \mathbb{I}(\gamma \leq 1-\epsilon) \cdot \boldsymbol{y} + \mathbb{I}(\gamma > 1-\epsilon) \cdot \hat{\boldsymbol{y}}_i. \tag{3}$$

Fig.2 shows an example of a ground truth label sequence and its n-best hypotheses that are used to smooth the ground truth label. The n-best hypotheses are generated by the baseline RNNT models.

4. Experiments

4.1. SWB300

The RNNT acoustic model for SWB300 has 6 bi-directional LSTM layers in the transcription network with 1,280 cells in each layer (640 cells in each direction). The prediction network is a single-layer uni-directional LSTM with 768 cells. The outputs of the transcription network and the prediction network are projected down to a 256-dimensional latent space where

they are combined by element-wise multiplication in the joint network. After a hyperbolic tangent nonlinearity followed by a linear transform, it connects to a softmax layer consisting of 46 output units which correspond to 45 characters and the null symbol. The acoustic features are 40-dimensional logMel features extracted every 10ms and their first and second order derivatives. The logMel features are after conversation based mean and variance normalization. Every two adjacent frames are concatenated and appended by a 100-dimensional i-vector [30] as speaker embedding. Therefore the input to the transcription network is 340 in dimensionality.

The training data goes through three steps of data augmentation. First, it is augmented by speed and tempo perturbation [4]. This is conducted offline and produces additional 4 replicas of the original training data, which gives rise to about 1,500 hours of training data in total. After the speed and tempo perturbation, mix-up sequence noise is injected where an utterance is artificially corrupted by adding a randomly selected down-scaled training utterance from the training set [5]. After that, SpecAugment is applied where the logMel spectrum of a training utterance is randomly masked in blocks in both the time and frequency domains [6]. Dropout [2] is also used in the LSTM layers with a dropout rate of 0.25 and the embedding layer with a dropout rate of 0.05. In addition, DropConnect [31] is applied with a rate of 0.25, which randomly zeros out elements of the LSTM hidden-to-hidden transition matrices. A Connectionist Temporal Classification (CTC) [32] model is used to initialize the transcription network.

Optimizer `AdamW` is used for the training. The learning rate starts at 0.0001 in the first epoch and then linearly scales up to 0.001 in the first 10 epochs. It holds for another 6 epochs before being annealed by $\frac{1}{\sqrt{2}}$ every epoch after the 17$^{\text{th}}$ epoch. The training ends after 30 epochs. The batch size is 64 utterances. An alignment-length synchronous decoder [33] is used for inference. We measure the WERs with and without an external LM. When decoding with an external LM, density ratio LM fusion [34] is used. The external LM is trained on a target domain corpus (Fisher and Switchboard) and the source LM is trained only on the training transcripts. The length perturbation is applied before mix-up and SpecAug, all of which are carried out on the fly in the data loader. We evaluate various hyper-parameter configurations for length perturbation (p_s,p_p,r_s,T_s,r_p,T_p) and n-best label smoothing (K and ϵ) on Hub5 2000, Hub5 2001 and RT03 test sets. The data preparation pipeline follows the Kaldi [35] s5c recipe.

Table 1 breaks down the performance of length perturbation on frame insertion (2$^{\text{nd}}$-5$^{\text{th}}$ rows), frame skipping (6$^{\text{th}}$-9$^{\text{th}}$ rows) and perturbation both ways (10$^{\text{th}}$-12$^{\text{th}}$ rows), respectively, on the Hub5 2000 test set. The length perturbation is applied in the first 25 epochs and lifted afterwards. From the table, one can observe that both frame insertion and skipping alone can improve WERs. The best WER is obtained when perturbing both ways (avg. 10.7% without using external LM and 9.4% when using external LM.)

Table 2 shows the performance of n-best label smoothing under various ϵ, which controls the probability to replace the ground truth label with a noisy label, and K, which is the total number of n-best hypotheses considered. Similarly, the label smoothing is applied in the first 25 epochs and lifted afterwards. The best performance (avg. 10.9% without using external LM and 9.5% when using external LM) is achieved when $\epsilon = 0.1$ and $K = 20$.

Experimental results on combining the two techniques are reported in Table 3 where label smoothing is applied up to 15 epochs and length perturbation is applied between 16 to 30 epochs. After 30 epochs both techniques are lifted and the training continues for another 5 epochs with the learning rates boosted by 2 times. The model used for decoding is after 35 epochs. It shows that the techniques can be complementary. By combining the two techniques we can get avg. WER 10.7% without using the external LM and 9.2% with the external LM. This is by far the state-of-the-art single-model result on the Hub5 2000 test set using RNNT, to the best of our knowledge.

Table 1: *Length perturbation using various hyper-parameter configurations on SWB300.*

	w/o LM			w/ LM		
	swb	ch	avg	swb	ch	avg
baseline	7.4	15.0	11.2	6.1	13.5	9.8
$p_p=0.6, r_p=0.05, T_p=5$	7.1	15.1	11.1	6.0	13.3	9.7
$p_p=0.6, r_p=0.1, T_p=3$	**7.2**	**14.9**	**11.1**	**6.1**	**13.1**	**9.6**
$p_p=0.6, r_p=0.1, T_p=7$	7.1	14.9	11.0	6.1	13.3	9.7
$p_p=0.7, r_p=0.1, T_p=5$	7.3	14.7	11.0	6.1	13.3	9.7
$p_s=0.7, r_s=0.1, T_s=5$	7.0	14.8	10.9	6.0	13.4	9.7
$p_s=0.7, r_s=0.1, T_s=7$	6.9	14.5	10.7	5.9	13.0	9.5
$p_s=0.8, r_s=0.1, T_s=7$	**6.8**	**14.8**	**10.8**	**5.8**	**13.0**	**9.4**
$p_s=0.7, r_s=0.1, T_s=9$	6.9	14.8	10.9	6.0	13.1	9.6
$p_s=p_p=0.6, r_s=r_p=0.1$ $T_s=7, T_p=3$	7.1	14.7	10.9	6.1	13.2	9.7
$p_s=p_p=0.7, r_s=r_p=0.1$ $T_s=7, T_p=3$	**6.9**	**14.4**	**10.7**	**5.9**	**12.8**	**9.4**
$p_s=p_p=0.8, r_s=r_p=0.1$ $T_s=7, T_p=3$	7.0	14.5	10.8	5.9	13.0	9.5

Table 2: *N-best based label smoothing using various configurations on SWB300.*

	w/o LM			w/ LM		
	swb	ch	avg	swb	ch	avg
baseline	7.4	15.0	11.2	6.1	13.5	9.8
$\epsilon=0.1, K=20$	**7.1**	**14.7**	**10.9**	**6.0**	**13.0**	**9.5**
$\epsilon=0.2, K=20$	7.2	14.5	10.9	6.1	13.1	9.6
$\epsilon=0.1, K=30$	7.3	14.6	11.0	6.1	13.1	9.6
$\epsilon=0.2, K=30$	7.1	14.9	11.0	6.0	13.0	9.5

Table 3: *Combination of length perturbation and n-best label smoothing on SWB300.*

	w/o LM			w/ LM		
	swb	ch	avg	swb	ch	avg
baseline	7.4	15.0	11.2	6.1	13.5	9.8
$\epsilon=0.1, K=20, p_s=p_p=0.5$ $r_s=r_p=0.1, T_s=5, T_p=5$	6.9	15.0	11.0	5.8	13.0	9.4
$\epsilon=0.1, K=20, p_s=p_p=0.6$ $r_s=r_p=0.1, T_s=5, T_p=5$	7.0	14.4	10.7	5.9	12.7	9.3
$\epsilon=0.1, K=20, p_s=p_p=0.5$ $r_s=r_p=0.1, T_s=7, T_p=3$	**6.9**	**14.5**	**10.7**	**5.9**	**12.5**	**9.2**
$\epsilon=0.1, K=20, p_s=p_p=0.5$ $r_s=r_p=0.1, T_s=7, T_p=3$	6.8	14.6	10.7	5.9	12.7	9.3

Table 4 reports the WERs using label smoothing (nbestls), length perturbation (lenpb) and their combination on Hub5 2000, Hub5 2001 and RT03. The external LM is used in the decoding. For comparison, we present the single-model result reported in [8] as one baseline (1$^{\text{st}}$ row) and the baseline used in this work (2$^{\text{nd}}$ row). The difference is the learning rate schedule and the number of epochs. In [8], the maximum learning rate is set to 5e-4 and the OneCycleLR policy [36] is used for 20 epochs. The current baseline gives slightly better performance. The models that give the best performance on Hub5 2000 in Tables 1, 2 and

3, respectively, are used to evaluate on Hub5 2001 and RT03. As can be seen, although the hyper-parameters of label smoothing and length perturbation are optimized on Hub5 2000, the models generalize well on Hub5 2001 and RT03.

Table 4: *WERs of length perturbation and n-best label smoothing on Hub5 2000, Hub5 2001 and RT03 test sets.*

	Hub5'00		Hub5'01			RT'03	
	swb	ch	swb	s2p3	s2p4	swb	fsh
baseline[8]	6.3	13.1	7.1	9.4	13.6	15.4	9.5
baseline	6.1	13.5	6.7	9.6	13.4	15.7	9.0
lenpb	5.9	12.8	6.5	9.1	13.0	15.2	8.8
nbestls	6.0	13.0	6.6	9.0	12.7	14.8	8.8
nbestls+lenpb	5.9	12.5	6.6	8.7	12.8	14.0	8.5

4.2. JPN500

The RNNT acoustic model for JPN500 has 6 bi-directional LSTM layers in the transcription network with 1,280 cells in each layer (640 cells in each direction). The prediction network is a single-layer uni-directional LSTM with 1,024 cells. The outputs of the transcription network and the prediction network are projected down to a 256-dimensional latent space where they are combined by element-wise multiplication in the joint network. After a hyperbolic tangent nonlinearity followed by a linear transform, it connects to a softmax layer consisting of 3547 output units which correspond to Japanese characters and the null symbol. The acoustic features are 40-dimensional log-Mel features extracted every 10ms and theirs first and second order derivatives. The logMel features are after utterance based mean normalization. Every four adjacent frames are concatenated. This more aggressive frame skipping is to reduce the length mismatch between the feature sequence and character label sequence. The input to the transcription network is 480 in dimensionality.

There is no data augmentation in the training. But dropout is used in the LSTM layers with a dropout rate of 0.25 and the embedding layer with a dropout rate of 0.05. Optimizer `AdamW` is used for the training. The learning rate starts at 0.0001 in the first epoch and then linearly scales up to 0.001 in the first 10 epochs. It holds for another 6 epochs before being annealed by $\frac{1}{\sqrt{2}}$ every epoch after the 17^{th} epoch. The model is obtained after 30 epochs. The batch size is 256 utterances. The same alignment-length synchronous decoder is used for inference. No external LM is used in decoding. Both length perturbation and label smoothing are applied in the first 25 epochs and lifted afterwards, respectively. We evaluate various hyper-parameter configurations for length perturbation (p_s,p_p,r_s,T_s,r_p,T_p) and n-best label smoothing (K and ϵ) on 13 real-world test sets from a broad variety of domains and report the average CERs across these test sets.

Table 5 and Table 6 show the respective performance of length perturbation and n-best based label smoothing on JPN500 using various hyper-parameter settings. Following a similar trend in SWB300, both techniques help to consistently improve the CERs over the baseline. The length perturbation can reduce the CER from 19.4% in the baseline to 18.5% and the label smoothing can reduce to 18.6%. One interesting observation is that since JPN500 has a more aggressive downsampling strategy (4-frame skipping) in input feature processing compared to that of SWB300 (2-frame skipping) the optimal length perturbation setting for JPN500 tends to favor frame insertion more than frame skipping over SWB300 ($T_s=3, T_p=5$ for JPN500 vs. $T_s=7, T_p=3$ for SWB300). Table 7 shows that the two techniques can also be complementary. In the experiments, the label smoothing is applied up to 15 epochs and length perturbation is applied between 16 to 25 epochs. After 25 epochs both techniques are lifted and the training continues for another 5 epochs. The model used for decoding is after 30 epochs. The combination of the two techniques can achieve a CER of 18.4%, which amounts to 1% absolute improvement over the 19.4% baseline averaging across 13 test sets.

Table 5: *Length perturbation using various hyper-parameter configurations on JPN500.*

	CER
baseline	19.4
$p_p=0.6, r_p=0.1, T_p=3$	19.0
$p_p=0.6, r_p=0.1, T_p=5$	19.2
$p_p=0.7, r_p=0.1, T_p=5$	**18.7**
$p_s=0.5, r_s=0.1, T_s=3$	**18.5**
$p_s=0.6, r_s=0.1, T_s=3$	18.7
$p_s=0.5, r_s=0.1, T_s=5$	18.7
$p_s=p_p=0.6, r_s=r_p=0.1, T_s=3, T_p=3$	19.9
$p_s=p_p=0.6, r_s=r_p=0.1, T_s=3, T_p=5$	**18.6**
$p_s=p_p=0.6, r_s=r_p=0.2, T_s=3, T_p=5$	**18.6**

Table 6: *N-best label smoothing using various configurations on JPN500.*

	CER
baseline	19.4
$\epsilon=0.2, K=10$	19.0
$\epsilon=0.2, K=20$	19.3
$\epsilon=0.2, K=30$	**18.6**
$\epsilon=0.3, K=30$	18.9

Table 7: *Combination of n-best label smoothing and length perturbation on JPN500.*

	CER
baseline	19.4
$\epsilon=0.2, K=30, p_s=p_p=0.5, r_s=r_p=0.1, T_s=3, T_p=5$	18.5
$\epsilon=0.2, K=30, p_s=p_p=0.4, r_s=r_p=0.1, T_s=3, T_p=5$	**18.4**
$\epsilon=0.2, K=30, p_s=p_p=0.3, r_s=r_p=0.1, T_s=3, T_p=5$	18.7

4.3. Discussion

Since the length perturbation perturbs an utterance both ways, it has SpliceOut [11] and DropFrame [10] as a special case of one-way perturbation, which corresponds to the results in Table 1 and Table 5 where only T_s is used ($T_p = 0$). The length perturbation can be used by itself as shown in the JPN500 case or used together with other data augmentation techniques as shown in the SWB300 case. The implementation of the length perturbation may have numerous variations such as the order of frame skipping and insertion or whether frame skipping and insertion should take place in different utterances. This will be further investigated in the future work.

5. Summary

In this paper we introduce length perturbation and n-best based label smoothing to improve the generalization of DNN acoustic modeling. We evaluate the two techniques extensively on both SWB300 and JPN500 datasets and show that both techniques can improve accuracy over strong baselines with RNNT acoustic models. The techniques can be complementary. By combining the two techniques, we have obtained state-of-art single-model results on SWB300 using RNNT.

6. References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[2] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[3] 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.

[4] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Interspeech*, 2015, pp. 3586–3589.

[5] G. Saon, Z. Tuske, K. Audhkhasi, and B. Kingsbury, "Sequence noise injected training for end-to-end speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6261–6265.

[6] 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," in *Interspeech*, 2019, pp. 2613–2617.

[7] Y. Miao, J. Li, Y. Wang, S.-X. Zhang, and Y. Gong, "Simplifying long short-term memory acoustic models for fast training and decoding," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 2284–2288.

[8] G. Saon, Z. Tueske, D. Bolanos, and B. Kingsbury, "Advancing RNN transducer technology for speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[9] I. Song, J. Chung, T. Kim, and Y. Bengio, "Dynamic frame skipping for fast speech recognition in recurrent neural network based acoustic models," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4984–4988.

[10] V. Sunder, S. Thomas, H.-K. Kuo, J. Ganhotra, B. Kingsbury, and E. Fosler-Lussier, "Towards end-to-end integration of dialog history for improved spoken language understanding," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[11] A. Jain, P. R. Samala, D. Mittal, and P. Jyothi, "SpliceOut: a simple and efficient audio augmentation method," *arXiv preprint arXiv:2110.00046*, 2021.

[12] L. Prananta, B. M. Halpern, S. Feng, and O. Scharenborg, "The effectiveness of time stretching for enhancing dysarthric speech for improved dysarthric speech recognition," *arXiv preprint arXiv:2201.04908*, 2022.

[13] T.-S. Nguyen, S. Stuker, J. Niehues, and A. Waibel, "Improving sequence-to-sequence speech recognition training with on-the-fly data augmentation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7689–7693.

[14] R. Mignot and G. Peeters, "An analysis of the effect of data augmentation methods: experiments for a musical genre classification task," *Transactions of the International Society for Music Information Retrieval*, vol. 2, no. 1, pp. 97–110, 2019.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 2818–2826.

[16] R. Muller, S. Kornblith, and G. Hinton, "When does label smoothing help," in *Advances in Neural Information Processing Systems (NeuIPS)*, 2019, pp. 4694–4703.

[17] Y. Xu, Y. Xu, Q. Qian, H. Li, and R. Jin, "Towards understanding label smoothing," *arXiv preprint arXiv:2006.11653*, 2020.

[18] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2018, pp. 8697–8710.

[19] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence model," in *Interspeech*, 2017, pp. 523–527.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhit, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 6000–6010.

[21] A. Zeyer, K. Irie, R. Schluter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *Interspeech*, 2018, pp. 7–11.

[22] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

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

[24] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: telephone speech corpus for research and development," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1992, pp. 517–520.

[25] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: a resource for the next generation of speech-to-text," in *Proceedings of ICLRE*, 2004, pp. 69–71.

[26] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving RNN transducer modeling for end-to-end speech recognition," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019.

[27] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, S. Yuan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S. y. Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6381–6385.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "MaskCycleGAN-VC: learning non-parallel voice conversion with filling in frames," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5919–5923.

[30] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using I-vectors," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2013, pp. 55–59.

[31] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, "Regularization of neural networks using DropConnect," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2013, pp. 1058–1066.

[32] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2006, pp. 369–376.

[33] G. Saon, Z. Tuske, and K. Audhkhasi, "Alignment-length synchronous decoding for RNN transducer," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7804–7808.

[34] E. McDermott, H. Sak, and E. Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 434–441.

[35] 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 *Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011.

[36] L. N. Smith and N. Topin, "Super-convergence: very fast training of neural networks using large learning rates," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 2019.