DYNAMIC FRAME SKIPPING FOR FAST SPEECH RECOGNITION IN RECURRENT NEURAL NETWORK BASED ACOUSTIC MODELS

Inchul Song[1], *Junyoung Chung*[2*], *Taesup Kim*[2], *Yoshua Bengio*[2†]

[1]Samsung Advanced Institute of Technology, Republic of Korea
[2]MILA, Université de Montréal, Canada
inchul2.song@samsung.com, {junyoung.chung,taesup.kim,yoshua.bengio}@umontreal.ca

ABSTRACT

A recurrent neural network is a powerful tool for modeling sequential data such as text and speech. While recurrent neural networks have achieved record-breaking results in speech recognition, one remaining challenge is their slow processing speed. The main cause comes from the nature of recurrent neural networks that read only one frame at each time step. Therefore, reducing the number of reads is an effective approach to reducing processing time. In this paper, we propose a novel recurrent neural network architecture called Skip-RNN, which dynamically skips speech frames that are less important. The Skip-RNN consists of an acoustic model network and skip-policy network that are jointly trained to classify speech frames and determine how many frames to skip. We evaluate our proposed approach on the Wall Street Journal corpus and show that it can accelerate acoustic model computation by up to 2.4 times without any noticeable degradation in transcription accuracy.

Index Terms— recurrent neural networks, neural acoustic models, dynamic frame skipping, policy gradient methods

1. INTRODUCTION

Deep neural network (DNN) acoustic models have been shown to perform very well for large vocabulary continuous speech recognition (LVCSR) [1]. As an alternative to DNNs, recurrent neural networks (RNNs) are more suited for modeling sequential data such as speech signals. In particular, acoustic models based on Long Short-Term Memory (LSTM) units [2] have been shown to outperform DNNs and conventional RNN based acoustic models in LVCSR [3, 4].

Despite their excellent performance, one drawback of RNN-based models is their slow processing speed. It is prevalent in many recurrent models that process a sequence of speech frames one frame at a time, thereby introducing sequential dependencies which increase the processing time in proportion to the length of an utterance. One effective approach to accelerating the processing speed of neural acoustic models is to skip speech frames that are not relevant in achieving high accuracy. This technique is called *frame skipping* and was first proposed in [5] for DNN acoustic models. Instead of making a prediction for each speech frame, acoustic model computation is done at a lower frame rate and the predicted labels for the non-skipped frames are copied to the skipped frames. Frame skipping was also applied to RNNs in [6]. We refer to these approaches using a fixed skip rate as *static frame skipping* in the sense that speech frames are skipped at a predefined interval without considering information variability in speech signals.

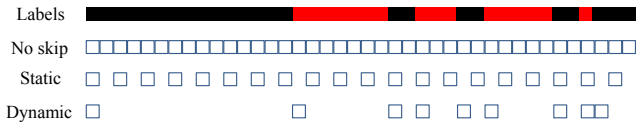

Fig. 1. Static vs. dynamic frame skipping. The first row shows label (HMM state) repetitions in a training example. Whenever a label changes, the color changes either from black to red or red to black. The boxes indicate the non-skipped speech frames in different frame skipping strategies.

Although static frame skipping can reduce processing time without significant degradation in transcription accuracy [6], fixed skipping intervals fail to consider the variable durations of phonemes. This motivates us to consider the more adaptive strategy proposed in this paper, which we call *dynamic frame skipping*. For example, Fig. 1 shows the durations of the labels (i.e., HMM states) extracted from a training example. An acoustic model using static frame skipping reads the speech frames at every pre-defined interval, regardless of whether a label changes or not. On the other hand, an acoustic model using dynamic frame skipping can adaptively skip the speech frames by detecting label changes. Therefore, dynamic frame skipping can avoid reading redundant frames, and the processing time can be further reduced. The concept of dynamically skipping time steps for RNNs has been applied in the NLP domain [7], where the authors propose a task specific architecture designed for NLP problems such as sentiment analysis and automatic Q&A. However, the

*Currently at DeepMind.
†CIFAR Fellow.

architecture is not readily applicable to acoustic modeling.

In this paper, we propose a novel recurrent neural network architecture called Skip-RNN, which can be used to skip speech frames dynamically based on information variability in the input utterance. The Skip-RNN consists of an acoustic model network and skip-policy network, which share some of the internal representations of speech signals. The whole network is jointly trained to classify speech frames and determine how many frames to skip. The skip-policy network, whose objective function is non-differentiable, is trained by a policy gradient method. We evaluate the proposed architecture on the Wall Street Journal corpus by training LSTM-based acoustic models. We show that our dynamic frame skipping approach can accelerate acoustic model computation by up to 2.4 times, while maintaining almost the same transcription accuracy. We also demonstrate through visualization that the trained model is able to predict the durations of HMM states and make skip decisions accordingly.

2. RELATED WORK

Frame skipping is grounded on the idea of exchanging the access to information with speed gain by not wasting the computational effort to process that information, thereby trading off between accuracy and speed. There have been sophisticated variable-rate processing schemes that are mainly developed for GMM/HMM systems [8, 9]. Frame skipping is first applied to DNN acoustic models in [5]. Instead of emitting the prediction at every speech frame, an acoustic model runs at a lower frame rate. In [6], frame skipping is applied to RNN acoustic models. The authors find that, when frame skipping is applied to an RNN acoustic model only at test time, the performance degrades severely because frame skipping breaks the original temporal dependency modeled during training. To remedy the mismatch between training and testing times, an RNN acoustic model is trained also with frame skipping by splitting each training utterance into multiple shorter utterances. A similar idea is also considered in [10] for conventional RNN acoustic models and [11] for CTC-based ones where acoustic model computation happens at a reduced frame rate.

Our method is motivated by the work of [7] in which the authors propose a recurrent neural network architecture that can selectively process words in a sentence. Although the architecture is similar to ours, it is designed to solve NLP problems and not directly applicable to acoustic modeling.

3. MODEL

In this paper, we follow the standard approach used in hybrid systems [12], where frame-level HMM states given by a GMM-HMM system through forced alignment are provided as targets.

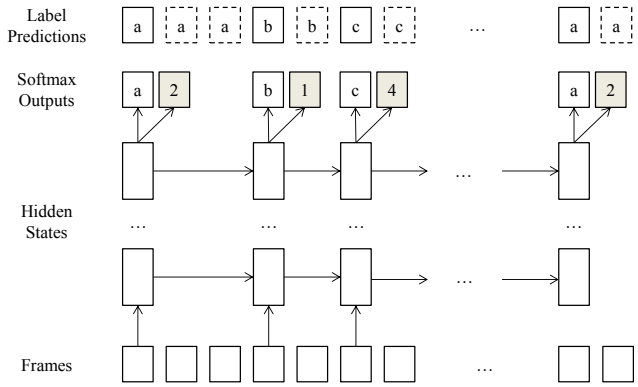

Fig. 2. An example showing how the Skip-RNN processes a training example. The alphabets indicate the labels (HMM states). For brevity, instead of showing the output of the label softmax layer, the label with the highest probability is shown. The numbers in the shaded boxes indicate the skip actions sampled from the outputs of the skip softmax layer. The dashed boxes indicate that the labels in them are copied from the non-skipped frames.

The Skip-RNN is a deep LSTM-RNN with two output layers on top: the label and skip softmax layers. The label softmax layer has an output dimension that corresponds to the total number of possible HMM states. The skip softmax layer matches its output dimension to the maximum allowable number M of frames to skip. The first dimension of the skip softmax layer corresponds to the probability of not skipping any frame, the second the probability of skipping one frame, and the n-th the probability of skipping $n-1$ frames.

From now on, we describe how the Skip-RNN processes frames from a training example. Given a training example $x_{1:T}$, it reads the first frame x_1 and computes its hidden states. Then the last hidden state is used to compute the label softmax distribution that determines the distribution over the HMM states and the skip softmax distribution that determines the distribution over the skip actions. Let s_1 be a skip action drawn from the skip softmax distribution to decide how many frames to skip, then the next frame to read becomes $x_{1+(s_1+1)}$, and the output of the label softmax layer is copied to the skipped frames. This process continues until the last frame x_T is reached. Figure 2 illustrates how the Skip-RNN processes a training example.

The Skip-RNN contains a set of parameters θ_l that are used to predict the HMM states and another set of parameters θ_s that are used to predict how many frames to skip. The optimization for θ_l can be done by minimizing the negative log-likelihood using stochastic gradient descent, as done in typical acoustic model training. On the other hand, finding θ_s can be handled as a reinforcement learning problem by using the standard policy gradient method [13].

3.1. Reward Function

Let $s_{1:N}$ be a sequence of skip actions during the training with a training example $x_{1:T}$, and h_i be the last hidden state that is used to determine the i-th skip action s_i. The i-th skip action s_i is obtained by sampling from the multinomial distribution $p(s_i|h_i;\theta_s)$ given by the skip softmax layer. A reward is given to each skip action under the current skip policy π_{θ_s}.

The Skip-RNN attempts to skip as many frames as possible without loss of accuracy. To this end, the Skip-RNN is designed to skip all the frames except the first one within the duration of an HMM state. Let M be the maximum allowable number of frames to skip, and $y_{1:T}$ be the label sequence of the example $x_{1:T}$. Suppose that the Skip-RNN is about to make the i-th skip decision while processing the j-th frame with the corresponding label y_j. Let the number of frames the label y_j is repeated starting from the j-th frame be $D(y_j)$. Under the constraint that the maximum allowable number of frames to skip is M, the target number of frames to skip for the i-th skip decision is $s_i^* = \min(D(y_j), M)$. We give the reward, r_i, to the i-th skip action as follows:

$$r_i = -|s_i^* - s_i|. \tag{1}$$

That is, we penalize the network in proportion to how much mistake it makes. This is because otherwise the network would make large skips to get penalized less often. Note also that we do not give a posive reward to the correct skip to prevent the network from accumulating rewards by making small skips over short-lasting HMM states.

3.2. Training with Policy Gradients

The policy gradient method maximizes the following expected cumulative future rewards, $J(\theta_s) = \mathbb{E}_{\pi_{\theta_s}}\left[\sum_{i=1}^N \gamma^{i-1} r_i\right]$, whose gradient is:

$$\nabla_{\theta_s} J(\theta_s) = \mathbb{E}_{\pi_{\theta_s}}\left[\sum_{i=1}^N \nabla_{\theta_s} \log \pi_{\theta_s}(s_i|h_i) R_i\right], \tag{2}$$

where $R_i = \sum_{k=i}^N \gamma^{k-i} r_k$ is the cumulative future rewards for the current action[1], and $\gamma \in [0,1]$ is a discount factor. Eq. 2 can be approximated by sampling action trajectories from the current policy π_{θ_s} and collecting the corresponding rewards[2].

[1] Instead of γ^{i-1}, we use γ^{k-i} that has less bias.

[2] We tried estimating the parameters θ_s by supervised learning, using target skips s_i^* as the targets for the skip softmax layer. However, the resulting network performed worse than the one trained by policy gradients in terms of processing speed. We conjecture that this is because supervised learning is concerned only with maximizing immediate rewards, without considering future rewards, i.e., it is similar to the case when $\gamma = 0$ in Eq. 2. But in general, as noted in [14], acting to maximize immediate rewards may actually reduce the sum of the rewards.

The variance of the estimated gradients can be high. We thus adopt the variance reduction strategy of [15]. We subtract from R_i the output of a linear baseline $b(h_i)$ that depends on the last hidden state h_i. This does not change the value of the expectation in Eq. 2 while reducing the variance. The parametrized baseline is trained to minimize the squared loss between R_i and $b(h_i)$. We also regularize the network by adding the entropy of the policy $H(\pi_{\theta_s}(\cdot|h_i))$ to the objective function, as suggested in [16], to encourage more exploration and prevent premature convergence.

4. EXPERIMENTS

4.1. Settings

We evaluate the proposed method on a benchmark dataset for large vocabulary automatic speech recognition: the Wall Street Journal (WSJ) corpus [17]. The WSJ corpus primarily consists of read speech with texts drawn from a machine-readable corpus of Wall Street Journal news text. We follow the standard Kaldi recipe s5 [18] to prepare the speech data. A baseline GMM-HMM system is trained on the 81 hours training set (train_si284) by Kaldi recipe tri4b, which consists of LDA preprocessing of data with MLLT and SAT for adaptation. We generate a forced alignment for each utterance to obtain frame-level targets. There are 3436 triphone states in total. We use the dataset test_dev93 as the development set and test_eval92 as the test set. Each frame in the acoustic signal is represented by 40 log Mel-filterbank outputs (plus energy), together with their first and second derivatives. Each utterance is then represented as a sequence of frames where the size of each frame is 123.

All neural acoustic models in the experiments have four unidirectional LSTM hidden layers with 512 LSTM cells in each layer without peephole connections. For the baseline acoustic model, we use the static frame skipping (SFS) method proposed in [6]. Each training utterance is split into K shorter sub-utterances, where i-th sub-utterance is formed by extracting and concatenating the frames at time $i, i+K, i+2K, \ldots$ $(1 \le i \le K)$. This increases the number of training utterances by K times. At test time, the first sub-utterance for each test utterance is always used.

The Adam optimizer [19] is used to train all models with the initial learning rate set to 0.001. We block the gradients from flowing through the last hidden state when updating the parameters θ_s with policy gradients in order to stabilize the training process. We apply gradient-norm clipping with a threshold of 1.0. The mini-batch size is set to 16. The discount factor γ is set to 0.99 in all experiments. During inference, we use greedy evaluation by selecting the most probable skip action given by the skip softmax output.

We measure only the forward propagation time spent on numerical operations as in [5, 6]. Optimization and implementation techniques in auto-differentiation frameworks such

Method	Dev WER (%)	Frame Usage (%)	Time (sec)	Test WER (%)	Time (sec)
SFS ($K = 1$)	8.36	100	848	5.87	551
SFS ($K = 2$)	8.61	50	446	6.04	286
SFS ($K = 3$)	8.91	33	310	5.95	200
SFS ($K = 4$)	9.21	25	242	6.52	257
SFS ($K = 5$)	10.37	20	199	6.98	128
Skip-RNN ($M = 4$)	8.63	40	384	6.18	250
Skip-RNN ($M = 6$)	8.50	37	352	5.71	235
Skip-RNN ($M = 8$)	8.56	36	350	6.40	230

Table 1. Comparison between static frame skipping and our proposed method on the WSJ corpus.

as TensorFlow [20] can heavily affect computation speed. In order to make a proper comparison, those factors are not considered. The reported times are measured by running each model one pass on the whole dev or test set. All models are implemented in TensorFlow.

4.2. Results

Table 1 shows the word error rates (WERs), frame usage, and times for acoustic model computation of the SFS and Skip-RNN on the dev and test sets. In the SFS, the WER increases rapidly as more frames are skipped. For example, while the SFS with $K = 5$ uses only 20% of the frames on the dev set, its WER degrades by 2.01% absolute from 8.36% to 10.37%. On the other hand, the Skip-RNN with $M = 6$ uses 37% of the frames and still performs reasonably well on the dev set with only 0.14% increase in WER.

The effect of varying the maximum allowable skip M of the Skip-RNN is also shown in Table 1. As M increases, the Skip-RNN can make larger skips, which results in reduction of frame usage. However, the WER on the dev set does not degrade significantly. This is because the Skip-RNN learns to use larger skips only for long-lasting HMM states, e.g., those representing silences. Note that the Skip-RNN with $M = 6$ performs the best. When M is small, the Skip-RNN processes frames more frequently and thus needs to model the temporal dependencies between the same HMM states as well as different ones. As M increases, the Skip-RNN only needs to focus on modeling the temporal dependencies between different HMM states, which is an easier task. This results in a reduced WER on the dev set. However, when M becomes too large, it is more likely that the Skip-RNN makes excessive skips and thus makes mistakes. Although this does not lead to much increase in WER on the dev set, the WER on the test set degrades due to overfitting. Surprisingly, when $M = 6$, the Skip-RNN even outperforms the standard LSTM acoustic model on the test set.

In Fig. 3 (a) and (b), we visualize how the SFS with $K = 3$ and Skip-RNN with $M = 8$ process examples from the dev set. As shown in Fig. 3 (a), the SFS repeatedly processes speech frames regardless of the durations of HMM states. This not only slows down processing speed, but also makes the SFS waste its modeling power to capture trivial temporal patterns between the same HMM states. Note that the SFS also does not consider label boundaries, thus is more likely to miss some of the label predictions altogether. When K is large, this results in a large decrease in terms of accuracy. On the other hand, Fig. 3 (b) shows that the Skip-RNN adapts the frame skip rate dynamically, making small skips for short-lasting HMM states and jumping large for long-lasting ones. Utterances typically have long silences at the beginning and end. The Skip-RNN exploits this fact and makes the maximum allowable skips over those HMM states. This leads to great reduction in acoustic model computation while maintaining accuracy.

(a) The SFS with $K = 3$

(b) The Skip-RNN with $M = 8$

Fig. 3. Skip actions taken by the Skip-RNN in comparison to the SFS for examples from the dev dataset. In each figure, the first row shows label changes and the second row indicates non-skipped frames.

5. CONCLUSION

In this paper, we consider a dynamic frame skipping strategy to accelerate the processing speed of RNN-based neural acoustic models. We propose a novel RNN architecture called Skip-RNN. The Skip-RNN is a deep LSTM-RNN with two sub-networks that are jointly trained to learn how to classify speech frames and how many frames to skip. We show through experiments on the Wall Street Journal corpus that it can accelerate acoustic model computation by 2.4 times without any noticeable degradation in transcription accuracy.

6. ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the following agencies for research funding and computing support: NSERC, Samsung, Calcul Quebec, Compute Canada, the Canada Research Chairs and CIFAR.

7. REFERENCES

[1] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition," *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005*, pp. 753–753, 2005.

[4] Hasim Sak, Andrew W Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling.," in *Interspeech*, 2014, pp. 338–342.

[5] Vincent Vanhoucke, Matthieu Devin, and Georg Heigold, "Multiframe deep neural networks for acoustic modeling," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7582–7585.

[6] Yajie Miao, Jinyu Li, Yongqiang Wang, Shi-Xiong Zhang, and Yifan Gong, "Simplifying long short-term memory acoustic models for fast training and decoding," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2284–2288.

[7] Adams Wei Yu, Hongrae Lee, and Quoc V Le, "Learning to skim text," *arXiv preprint arXiv:1704.06877*, 2017.

[8] KM Ponting and SM Peeling, "The use of variable frame rate analysis in speech recognition," *Computer Speech & Language*, vol. 5, no. 2, pp. 169–179, 1991.

[9] Qifeng Zhu and Abeer Alwan, "On the use of variable frame rate analysis in speech recognition," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*. IEEE, 2000, vol. 3, pp. 1783–1786.

[10] Golan Pundak and Tara N Sainath, "Lower frame rate neural network acoustic models.," in *Interspeech*, 2016, pp. 22–26.

[11] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," *arXiv preprint arXiv:1507.06947*, 2015.

[12] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 273–278.

[13] Ronald J Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[14] Richard S Sutton and Andrew G Barto, "Reinforcement learning: An introduction. 1998," *A Bradford Book*, 1998.

[15] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al., "Recurrent models of visual attention," in *Advances in neural information processing systems*, 2014, pp. 2204–2212.

[16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

[17] Douglas B Paul and Janet M Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[18] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[19] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: A system for large-scale machine learning.," in *OSDI*, 2016, vol. 16, pp. 265–283.