INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Adversarial Multi-task Learning of Deep Neural Networks for Robust Speech Recognition

Yusuke Shinohara

Corporate Research and Development Center, Toshiba Corporation
1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582, Japan
yusuke.shinohara@toshiba.co.jp

Abstract

A method of learning deep neural networks (DNNs) for noise robust speech recognition is proposed. It is widely known that representations (activations) of well-trained DNNs are highly invariant to noise, especially in higher layers, and such invariance leads to the noise robustness of DNNs. However, little is known about how to enhance such invariance of representations, which is a key for improving robustness. In this paper, we propose adversarial multi-task learning of DNNs for explicitly enhancing the invariance of representations. Specifically, a primary task of senone classification and a secondary task of domain (noise condition) classification are jointly solved. What is different from the standard multi-task learning is that the representation is learned adversarially to the secondary task, so that representation with low domain-classification accuracy is induced. As a result, senone-discriminative and domain-invariant representation is obtained, which leads to an improved robustness of DNNs. Experimental results on a noise-corrupted Wall Street Journal data set show the effectiveness of the proposed method.

Index Terms: speech recognition, noise robustness, deep neural networks, adversarial multi-task learning

1. Introduction

Robustness against noise, reverberation, channel, speaker and other factors of variation is a long-standing research topic in speech recognition [1][2], and is a key ingredient for the successful deployment of the technology to a wide range of applications. A speech recognition system is said to be robust if its performance does not degrade significantly under variation of speech signals caused by these factors. We focus on noise robustness in this paper, as noise is a primary example of such factors, but our proposed method can be applied to other factors of variation as well.

Deep neural network (DNN) acoustic models [3] trained with speech data recorded in various noise conditions have been shown to be remarkably robust against noise [4]. It is widely known that representations (activations of units in hidden layers) of well-trained DNNs are highly invariant against noise, especially in higher layers, and such invariance of representations achieves the noise robustness of DNNs [5][6]. It has been indicated that improving the invariance of representations, for instance by feature engineering [7][8], further improves the robustness.

However, although it is of primal importance, little is known about how to enhance invariance of representations, especially in the context of robust speech recognition. If we can attain representations with improved invariance, it is expected that speech recognizers with improved noise robustness is realized.

In this paper, we propose adversarial multi-task learning of DNNs for explicitly enhancing invariance of representations. Specifically, a primary task of senone (tied states of context-dependent phones) classification and a secondary task of domain (noise condition) classification are jointly solved in a multi-task learning framework, where input layers to extract the representation are shared among the tasks. What is different from the standard multi-task learning is that the representation is learned adversarially to the secondary task, i.e. representation with low domain-classification accuracy is preferred, so that domain-dependent information is purged from the representation. As a result, senone-discriminative and domain-invariant representation is obtained, which leads to improved noise robustness of the DNN. Experimental results on a noise-corrupted Wall Street Journal dataset show the validity of the proposed method.

2. Related work

Multi-task learning of DNNs for improving speech recognition accuracy is emerging recently [9][10][11]. For instance, Bell and Renals [10] jointly optimized a primary task of senone classification and a secondary task of monophone classification. By solving the primary task jointly with the secondary task, the accuracy of the primary task was improved, and thus the speech recognition accuracy was improved. However, what was used by Bell and Renals is the standard (not adversarial) multi-task learning. There has been no work, to the best of our knowledge, that used adversarial multi-task learning for speech recognition.

Adversarial learning of DNNs is one of the hottest topics in deep learning recently [12][13][14]. The method is attracting much attention because of its surprising ability to generate realistic images from the learned network. Following the success in generative tasks, Ganin et al. [15][16] proposed to use adversarial framework for learning domain-invariant representations, evaluated their algorithm in unsupervised domain adaptation tasks, and achieved state-of-the-art performance. This method and other related methods have never been applied to supervised learning tasks, except for a recent preliminary study in [17][18], where they have tested their method on a face recognition task using 190 training images with five domains (illumination conditions). However, study with datasets of practical sizes has never been reported before.

http://dx.doi.org/10.21437/Interspeech.2016-879

3. Method

3.1. Single-task learning

In this subsection, we briefly review the conventional training method, i.e. single task learning of DNNs.

When we train a DNN for robust speech recognition, we usually use a multi-condition dataset, which is a collection of speech data recorded in various noise conditions, such as car noise at 10 dB (in signal-to-noise ratio) and station noise at 20 dB. Let $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ denote the training dataset, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the input vector, e.g. Mel-frequency filterbank coefficients, $y_i \in \{1, \ldots, C_y\}$ is the senone class, respectively of the i-th data point (i.e. frame), d is the dimensionality of the input vector, C_y is the number of senone classes, and N is the number of data points. The DNN is trained to minimize the cross-entropy loss function,

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_i \log P(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}), \tag{1}$$

where $P(y|\boldsymbol{x};\boldsymbol{\theta})$ is calculated with a parametric classifier with parameters $\boldsymbol{\theta} \in \mathbb{R}^M$, i.e. a DNN with a set of tunable weights and biases, where M is the number of parameters. Stochastic gradient descent (SGD) is commonly used to optimize the parameters so as to minimize the loss function. Specifically, for each mini-batch (a small set of data points), the cross-entropy loss function is defined, its gradient w.r.t. the parameters is calculated via back-propagation, and the parameters are updated by a small step towards the gradient direction as

$$\boldsymbol{\theta} \longleftarrow \boldsymbol{\theta} - \epsilon \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}, \tag{2}$$

where $\epsilon \in \mathbb{R}$ is the learning rate. This update procedure is repeated until convergence.

3.2. Adversarial multi-task learning

In this subsection, we introduce adversarial multi-task learning of DNNs. Note that the algorithm was originally proposed by Ganin et al. [15, 16] for unsupervised domain adaptation, but its application to supervised learning tasks has never been examined before.

In adversarial multi-task learning of DNNs, we use a multi-condition dataset in a similar manner as single-task learning, but this time an additional class label is given for each data point. Namely, the training dataset is denoted by $\{\boldsymbol{x}_i, y_i, z_i\}_{i=1}^N$, where $z_i \in \{1, \ldots, C_z\}$ denotes the domain class (noise condition) of the i-th data point, and C_z denotes the number of domain classes. In our experiment, we have used 17 noise conditions for training ($C_z = 17$), such as car noise at 10 dB and exhibition-booth noise at 20 dB.

Figure 1 depicts the overall architecture of the adversarial multi-task DNN [1]. This multi-task DNN simultaneously executes senone classification and domain classification. It consists of three sub-networks, namely two output sub-networks, one for the primary task of senone classification and the other for the secondary task of domain classification, and an input sub-network shared among the tasks. The shared input sub-network

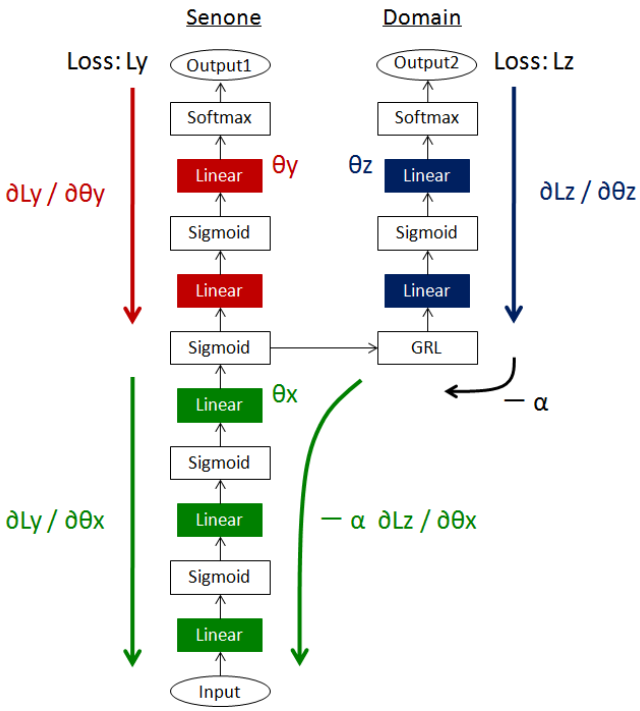

Figure 1: *An example of the adversarial multi-task deep neural network. See text for the gradient reversal layer (GRL) and other details. The architecture was originally proposed in [15][16] for unsupervised domain adaptation. Best viewed in color.*

acts as a feature extractor to convert an input vector to its representation. Each output sub-network acts as a classifier to calculate posterior probabilities of classes given the representation. Other details of the Figure are described in the following.

Different from the standard multi-task learning, in which the representation (input sub-network) is trained so as to maximize the classification accuracies of the primary and secondary tasks, in adversarial multi-task learning, the representation is learned adversarially to the secondary task (and friendly to the primary task), so that domain-dependent information is purged from the representation as it is irrelevant or nuisance for the primary classification task.

Let the parameters of the DNN consist of three parts, $\boldsymbol{\theta} = \{\boldsymbol{\theta}_x, \boldsymbol{\theta}_y, \boldsymbol{\theta}_z\}$, and $\boldsymbol{\theta}_x$, $\boldsymbol{\theta}_y$, and $\boldsymbol{\theta}_z$ denote the parameters of the input and output sub-networks, respectively. The cross-entropy loss functions for the primary and secondary tasks are defined as

$$\mathcal{L}_y(\boldsymbol{\theta}_x, \boldsymbol{\theta}_y) = -\sum_i \log P(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_x, \boldsymbol{\theta}_y), \tag{3}$$

$$\mathcal{L}_z(\boldsymbol{\theta}_x, \boldsymbol{\theta}_z) = -\sum_i \log P(z_i|\boldsymbol{x}_i;\boldsymbol{\theta}_x, \boldsymbol{\theta}_z), \tag{4}$$

The parameters are updated as

$$\boldsymbol{\theta}_y \longleftarrow \boldsymbol{\theta}_y - \epsilon \frac{\partial \mathcal{L}_y}{\partial \boldsymbol{\theta}_y}, \tag{5}$$

$$\boldsymbol{\theta}_z \longleftarrow \boldsymbol{\theta}_z - \epsilon \frac{\partial \mathcal{L}_z}{\partial \boldsymbol{\theta}_z}, \tag{6}$$

$$\boldsymbol{\theta}_x \longleftarrow \boldsymbol{\theta}_x - \epsilon \left(\frac{\partial \mathcal{L}_y}{\partial \boldsymbol{\theta}_x} - \alpha \frac{\partial \mathcal{L}_z}{\partial \boldsymbol{\theta}_x} \right), \tag{7}$$

[1] Each box represents a processing component that converts a vector (or a matrix if a minibatch is used) to another vector (or a matrix). For instance, the Linear component linearly transforms a vector, and the Sigmoid component applies sigmoid transform to each element of a vector.

where $\alpha \in \mathbb{R}$ is a positive scalar parameter to control the strength of regularization. In our experiment, α was gradually changed from 0 to α_{max}, as was done in [15][16]. Specifically, for the k-th epoch, the parameter was set as

$$\alpha_k = \min\left(\frac{k}{10}, 1\right) \cdot \alpha_{max}. \tag{8}$$

To implement this idea, the gradient reversal layer (GRL) proposed by Ganin [15, 16] is used. In the forward propagation phase, the GRL acts as an identity map, i.e. simply passes the input to the output as it is. Let $\boldsymbol{\xi}^{in}$ and $\boldsymbol{\xi}^{out}$ denote input and output vectors of the layer, respectively. Then the forward step is simply,

$$\boldsymbol{\xi}^{out} \longleftarrow \boldsymbol{\xi}^{in}. \tag{9}$$

In the backward propagation phase, the GRL reverses the gradient (multiplies $-\alpha$). The backward step is,

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}^{in}} \longleftarrow -\alpha \frac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}^{out}}, \tag{10}$$

Thus the input sub-network is trained adversarially to the secondary task.

After the training is finished, the output sub-network for the secondary task is removed, and the resulting DNN is used for testing.

4. Experiments

The conventional method (single-task learning) and the proposed method (adversarial multi-task learning) of training DNNs for noise robust speech recognition were comparatively evaluated on a noise-corrupted Wall Street Journal (WSJ) dataset.

4.1. Setup

We have made a noise corrupted version of the WSJ dataset by artificially adding different types of noise at different SNR levels. Specifically, for each of the 37,318 utterances of the original WSJ training set, we have randomly picked one from four noise types (car 2000cc, exhibition booth, station, and crossing) and one from five SNR levels (5, 10, 15, 20, and 100 dB), and added the selected noise at the selected SNR to the utterance to create a noise corrupted training set. Also, for each of the 333 utterances of the original WSJ evaluation set, we have randomly picked one from the same five SNR levels, and added noise at the selected SNR level to the utterance; we have repeated this procedure for each of the 12 noise types, four of which are known (used for training) and the other eight are unknown, to create 12 evaluation sets.

We have used MFCC $+\Delta + \Delta\Delta$ features (39 dimensions) concatenated over 11 frames to form the input vector of 429 dimensions. A Gaussian mixture model (GMM) based acoustic model with 3398 senones was trained with the boosted MMI criterion [19], and used for forced-alignment to generate alignment, which was then used for the training of single-task (ST) and adversarial multi-task (AMT) DNNs.

The ST-DNN has the input layer with 429 units, four hidden layers with 1024 units, and the output layer with 3398 units. The AMT-DNN has the same set of layers, and additionally one hidden layer with 512 units and output layer with 17 units for the secondary task, as depicted in Figure 1. Parameter α_{max} was set to 0.1.

The GMM acoustic model was trained with Kaldi [20], then the DNNs were trained with an open-source deep-learning framework called Chainer [21], and finally the resulting DNNs were evaluated with Kaldi.

Table 1: *Word error rates (%) on the evaluation set (333 utterances) corrupted with 12 different noises. DNNs trained with single-task learning (ST) and adversarial multi-task learning (AMT) were compared. Relative error rate reduction (%) is also shown.*

	noise	ST	AMT	RERR
k	car 2000cc	5.83	5.56	4.63
k	exhib. booth	6.80	6.66	2.06
k	station	7.89	7.76	1.65
k	crossing	6.96	6.65	4.45
unk	car 1500cc	5.58	5.46	2.15
unk	exhib. aisle	7.71	6.93	10.12
unk	factory	12.17	12.92	-6.16
unk	highway	9.73	9.52	2.16
unk	crowd	6.72	6.40	4.76
unk	server room	8.54	7.76	9.13
unk	air cond.	6.96	6.98	-0.29
unk	elev. hall	9.23	9.60	-4.01
-	average	7.84	7.68	2.04

4.2. Result

Table 1 shows the experimental results. Word error rates (WERs) for the 12 evaluation sets are shown. The first column shows the noise category, either 'k' (known) or 'unk' (unknown). The second column shows the noise type. The third and fourth columns show WERs for the ST and AMT DNNs, and the final column shows the relative error rate reduction (RERR). The AMT-DNN had better WER than ST-DNN for nine of the 12 evaluation sets. On average, ST-DNN had WER of 7.84%, and AMT-DNN had WER of 7.68%, which translates to an overall RERR of 2.04%. Some noise types had large improvements and others did not. We have tried to figure out why this happened, but so far we did not find any clear explanation. Further study is needed to reveal how adversarial multi-task learning works in different noise conditions.

5. Discussion

In this paper, we have presented adversarial multi-task learning of DNNs for noise robust speech recognition. The adversarial multi-task DNN consists of three sub-networks, namely, the shared input sub-network for extracting the representation, the primary output sub-network for senone classification, and the secondary output sub-network for domain (noise condition) classification. Unlike the standard multi-task learning where the representation (input sub-network) is trained to maximize the primary and secondary classification accuracies, the representation is trained adversarially to the secondary task, so that the representation become senone-discriminative and domain-invariant. The improved domain-invariance leads to the improved noise robustness of the DNN.

Although a straightforward optimization procedure is used in this work, adversarial learning is known to be difficult to get its best performance. More dedicated algorithm and tuning may further improve the performance. Application of adversarial multi-task learning is not limited to noise robustness; the same framework can be used for improving robustness against other factors of variation.

6. References

[1] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[2] T. Yoshioka, A. Sehr, M. Delcroix, K. Kinoshita, R. Maas, T. Nakatani, and W. Kellermann, "Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 114–126, 2012.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "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.

[4] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013.

[5] D. Yu, M. L. Seltzer, J. Li, J. Huang, and F. Seide, "Feature learning in deep neural networks - Studies on speech recognition tasks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

[6] M. Seltzer, "Robustness is dead! Long live robustness!" in *REVERB Workshop*, May 2014.

[7] A. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012.

[8] R. Zhao, J. Li, and Y. Gong, "Variable-activation and variable-input deep neural network for robust speech recognition," in *Processing of the IEEE Spoken Language Technology Workshop (SLT)*, December 2014.

[9] M. L. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[10] P. Bell and S. Renals, "Regularization of context-dependent deep neural networks with context-independent multi-task training," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015.

[11] Z. Chen, S. Watanabe, H. Erdogan, and J. Hershey, "Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks," in *Proceedings of the INTERSPEECH*, 2015.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[13] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[14] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[15] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

[16] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[17] Y. Li, K. Swersky, and R. S. Zemel, "Learning unbiased features," in *Proceedings of the NIPS Workshop on Transfer and Multitask Learning*, 2014.

[18] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, "The variational fair autoencoder," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[19] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008.

[20] 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 *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.

[21] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.