(27)

Adversarial Multi-task Learning of Speaker-invariant Deep Neural Network Acoustic Models for Speech Recognition

L. Tóth [1] and G. Gosztolya [2]
[1] University of Szeged, Institute of Informatics, Árpád tér 2, H-6720 Szeged, Hungary
[2] MTA-SZTE Research Group on Artificial Intelligence, Tisza Lajos krt. 103, H-6720 Szeged, Hungary
E-mails: {tothl, ggabor}@u-szeged.hu

Summary: The performance of speech recognition systems has greatly improved with the introduction of Deep Neural Network (DNN) acoustic models. However, making these systems robust against all possible kinds of environmental conditions is still an important research topic. The adversarial multi-task DNN training method was proposed recently, and it has already been successfully applied to increase the domain and noise robustness of DNN acoustic models. Here, we evaluate the efficiency of this training method in increasing the speaker-invariance of a speech recognition system that is based on a convolutional neural network (CNN). Moreover, we propose a solution to handle those cases where speaker labels are not available for the training dataset. In the supervised case we report relative error rate reductions of 3-4 %. With the unsupervised method the improvements are somewhat smaller, but consistent across all tested parameter values.

Keywords: Speech recognition, Deep neural networks, Adversarial training, Multi-task training, Speaker-invariant.

1. Introduction

The introduction of deep learning in speech recognition has significantly reduced the error rate of speech recognition systems [1]. However, improving the robustness of these recognizers to various environmental factors is still in the focus of research [2], as the performance of current systems may drop drastically in different background noise, in reverberant environments, or simply with different speaker accents, just to name but a few possible adversarial conditions.

The sensitivity to these environmental factors can partly be explained by the fact that neural networks are inclined to overfit the actual training data, which reduces their generalization ability. Regularization methods are frequently applied to tackle this overfitting phenomenon. For example, it is frequently observed that presenting multiple tasks to the network at the same time – known as multi-task training [3] – also has a regularization effect. That is, having to solve two (or more) similar, but slightly different tasks at the same time forces the network to find a more general and more robust inner representation. Multi-task training has been successfully applied in several speech recognition studies [4, 5].

While multi-task training seeks to minimize the error of both tasks, there is a newer variant of the method known as *adversarial* multi-task training. Here, the error of the secondary task is *maximized*. With this modification, we expect the network to find an inner representation that is invariant with respect to the secondary task [6]. In speech technology, adversarial multi-task training has mostly been applied to increase the noise-robustness of DNN-based acoustic models [7, 8], as sensitivity to the background noise of the actual application domain is perhaps the most common adversarial factor. But we can also find examples where it is used to make the system less sensitive to other factors like accented speech [9].

The performance of speech recognizers may also vary significantly among speakers. In this paper, we seek to apply the adversarial multi-task training method to alleviate this issue. Our starting point will be the recent study of Meng et al. [10]. In contrast with their study, here we work with convolutional neural nets instead of fully connected DNNs. As the convolutional structure already makes the network less sensitive to speaker variance, it is not clear whether adversarial training can reduce this sensitivity any further. A second difference is that here we use the TIMIT database, which contains shorter samples from significantly more speakers than the corpus used in [10], so the task is presumably more difficult.

The approach of Meng et al. assumes that speaker-level annotation is available for the training data, permitting supervised training. However, most of the datasets available for training speech recognizers do not contain any information about the speakers. Thus, we also describe an experiment where we apply a clustering method which assigns the files to automatically designed speaker clusters, and the CNN is trained using these cluster labels. We will refer to this method as the unsupervised version of the approach.

2. Adversarial Multi-task Training

In multi-task training we train the neural network to solve multiple (in this case two) tasks in parallel, based on the same set of input features [6]. The two tasks should be related, but slightly different. Multi-task training requires a special network architecture where the network has separate output layers dedicated to the two tasks, and the uppermost

hidden layers are also task-specific. However, there is only one, shared input layer, and the lowermost hidden layers are also shared between the two tasks. The multi-task DNN architecture is illustrated in Fig. 1.

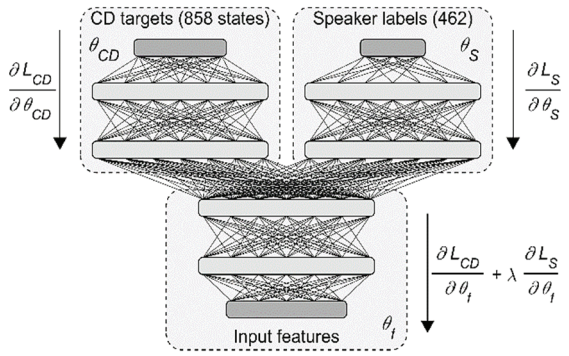

Fig. 1. Architecture of the (adversarial) multi-task deep neural network.

In our case, one of the output layers is trained to recognize the context-dependent (CD) states of the Hidden Markov Model (HMM) speech recognizer. The other, secondary output layer is trained to identify the speaker label of the actual training sentence. During multi-task training, we minimize the error functions (L_{CD} and L_S) of the two output layers simultaneously. Thus, during the backpropagation training process we have to sum the error values of the two branches when they reach the shared layers. This means that $\lambda = 1$ in the formula of Fig. 1. The fact that the lower layers are shared between the two tasks forces the network to find a hidden representation that is useful for minimizing both error functions.

In the case of *adversarial* multi-task training, the goal is to minimize the error of the main task, and *maximize* the error of the secondary task at the same time. Ganin et al. proposed the following solution for this [6]: we will keep minimizing the error of the task-specific layers. However, when the error backpropagation reaches the shared layers, the sign of the error for the secondary task is flipped, which is technically realized by using a negative λ value. This modification practically turns minimization into maximization with respect to the shared layers. This way, the network will seek a shared hidden representation that is optimal for solving the first task, but contains no useful information for solving the secondary task. As in our case the secondary task is speaker identification, the optimal hidden representation would be totally speaker-invariant, and the classification error rate of the secondary branch would be 100 %.

In his original paper, Shinohara suggested introducing the adversarial secondary task only gradually by slowly increasing the (absolute) value of λ [7]. That is, in the kth training iteration the value of λ would be set to

$$\lambda_k = \min(\frac{k}{c},1) \cdot \lambda \,, \qquad (1)$$

which means in practice that λ attains its final value after c training epochs. He proposed setting c to 10, but we also experimented with the value of 7, as we observed that during the backpropagation process the halving of the learning rate typically starts after 6-7 training epochs.

Another meta-parameter of the model is the number of layers in the network, and their division between the shared and the task-specific parts. As in our previous studies (see e.g. [11]) we obtained the best results with 4-5 hidden layers, here we worked with a network depth of 4 hidden layers. As regards the depth where the two branches should join, it seems reasonable that having more shared layers is better when the two tasks are quite similar, while quite different tasks would require more task-specific layers. However, the optimal structure can only be found experimentally. For example, in an earlier paper where the tasks were relatively different, we found an early division to be optimal [12]. For the actual speech plus speaker recognition task, Meng et al. applied a network with 2 shared and 2-5 task-specific layers [10]. In the pilot tests we obtained the best results with 3 joint and 1-1 task-specific layers, so we present detailed results only with this network architecture.

3. Experimental Set-up

The neural network model we applied here contained convolutional neurons in its lowest layer, which performs convolution along the frequency axis (for more details, see our earlier study [11]). The shared part consisted of three hidden layers, while the task-specific parts contained one hidden layer for both tasks. All fully connected hidden layers contained 2000 rectified linear (ReLU) neurons. The main output layer consisted of 858 softmax neurons, corresponding to the context-dependent states of the hidden Markov model speech recognizer. The output layer for the secondary task contained 462 softmax neurons, and it was trained to discriminate the speakers of the training dataset. The training was performed using the standard backpropagation algorithm with a mini-batch size of 1000 training vectors. The learning rate was fixed at a value of 0.001 until the training error kept decreasing on the validation set. Afterwards, the learning rate was halved in each training epoch. The cost function applied was the standard cross-entropy error rate for both output layers, measured at the level of the training vectors (acoustic data frames). As the task of the CNN is framewise classification, in some cases we will directly report the frame error rates obtained. The speech recognition system applies the standard HMM/DNN hybrid scheme to convert the frame-level probability estimates into a sequence of phones [11]. To evaluate the accuracy of the whole speech recognizer, we will report the phone error rates attained.

As the training database we used the TIMIT English speech corpus, which contains speech samples from 462 speakers in the training subset. The core test

set we used here contains samples from 24 speakers who are separate from the training set. As the development set, we randomly held out the samples of 44 speakers from the train set, which roughly corresponds to 10 % of the training material.

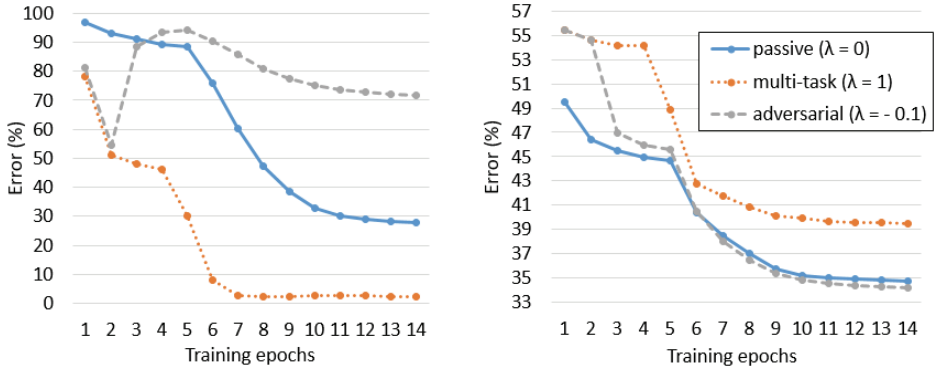

Fig. 2. The error curve for the secondary task on the train set (left), and the error curve for the main task on the development set (right) as a function of the number of training iterations.

4. Results and Discussion

Fig. 2 shows a typical example of how the (frame-level) classification error rate of the CNN changes during the training epochs. On the left we plotted the error rate of the secondary (speaker classification) task on the train set. We note that the secondary output will not be used by recognizer, so we plotted it only to gain an insight into what happens during adversarial multi-task training. The figure on the right shows the error for the main task – in this case on the development set, as this is the value that we seek to minimize. When $\lambda = 0$, the secondary branch of the network is allowed to learn, based on the hidden representation formed in the uppermost shared hidden layer, but it cannot modify this representation. Hence, we called this case the 'passive learning' scenario, and the result obtained in this configuration for the main task will serve as our baseline. We observe that in this case the speaker classifier branch can achieve an error rate below 30 % on the train set, while the error curve for the main output stops just below 35 %. By setting $\lambda = 1$, we get a classic multi-task model. In this case the training error rate of the secondary branch goes down to 3 %, but the price is that the error rate of the main task increases to 39 %. In the last experiment, we let the system run in multi-task mode for 2 iterations (to aid visualization), but then we turned on adversarial training by setting λ to -0.1. As the result, the corresponding error curve on the left quickly raises to the 70-90 % range, and remains there until the end of training. However, adversarial training has a beneficial effect on the main task, as the corresponding error curve goes slightly below that of the baseline (passive) model on the right.

Having found that adversarial training can outperform the standard multi-task training approach, we looked for the optimal parameter values. Table 1 shows how the parameters λ and c influence the error rates of the speech recognizer. As regards the main task, we report phone recognition error rates (PhER) from the full system on the development and test sets, while for the secondary task we report only frame-level error rates (FrER) of the neural network on the train set, as this output is not used by the recognizer. To reduce the effect of random initialization, we report the average scores of repeating the training 3 times. The results indicate that the error rate of the secondary task consistently increased as λ increased, just as expected. Moreover, compared to the baseline, the recognition error rate also consistently improved for all λ values both on the development and test sets, reaching the optimum at $\lambda = -0.15$ and $c = 10$. The relative error rate reduction lay between 3 % to 4 % on average, and it was 4.2 % in the best case.

Table 1. The frame error rates of the CNN (secondary task, train set) and the phone error rates of the speech recognizer (main task, development and test sets).

Parameters		FrER (train set, secondary task)	PhER (dev. set, main task)	PhER (test set, main task)
λ	c			
0 (baseline)		36 %	16.6 %	18.8 %
-0.03	7	57 %	16.3 %	18.3 %
-0.06	7	73 %	16.2 %	18.4 %
-0.10	7	82 %	16.1 %	18.3 %
-0.10	10	79 %	**16.0 %**	**18.0 %**
-0.15	10	85 %	16.2 %	18.1 %
-0.20	10	90 %	16.3 %	18.2 %

Table 2. The error rates of the speech recognizer as a function of the number of clusters.

No. of Clusters	Parameters		Phone Error Rate (%)	
	c	λ	dev. set	test set
–	baseline		16.6 %	18.8 %
50	7	-0.10	16.3 %	18.6 %
100	10	-0.15	16.0 %	18.0 %
150	10	-0.10	16.0 %	18.3 %
200	10	-0.10	16.2 %	18.4 %
250	10	-0.10	16.0 %	18.3 %

5. Unsupervised Case

The TIMIT corpus is an old-fashioned database in the sense that its content was carefully planned, and it also contains a detailed description of the speakers. Nowadays, we use much larger databases and we prefer to record these under realistic application conditions. Unfortunately, speaker annotation is not available for most of the newer databases. In these cases, we cannot directly apply the adversarial training method, as the missing speaker labels must be replaced by some other training targets. A possible solution is to use automatically determined labels. Here, we utilized a refined version of the unsupervised speaker clustering algorithm called bottom-up Hierarchical Agglomerative Clustering with a Generalized Likelihood Ratio (GLR) [13-15] for this purpose. This algorithm arranges the files into clusters, based on the similarity of the speakers' voices. The only assumption of the algorithm is that each file contains the voice of only one speaker, which is satisfied in the case of TIMIT, where each file contains a single sentence. The only meta-parameter of the algorithm is the number of clusters. We set this parameter to 50, 150, 150, 200 and 250, and then we repeated the DNN training experiment, but this time using the speaker clusters as training targets. Table 2 shows the results we obtained, for each cluster size reporting the score of just the best performing meta-parameters. As we can see, the improvements and the best result on the development set were comparable to those for the supervised method. On the test set, the decrease in the error rate was the same in the best case, though slightly less on average. However, the improvement was consistently present for all cluster sizes used.

6. Conclusions

Here, we evaluated the adversarial multi-task training method proposed by Meng et al. within the framework of CNNs. Moreover, we found a way to make the method work when speaker annotation is not available. We found that the adversarial training method is beneficial for CNNs as well, and that our unsupervised training approach can attain error rate reductions that are comparable to those of the original method.

Acknowledgements

We acknowledge the support of the Ministry of Human Capacities, Hungary, grant 20391-3/2018/FEKUSTRAT. László Tóth was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and the UNKP-18-4 New Excellence Program of the Hungarian Ministry of Human Capacities. The Titan X graphics card used in this research was donated by the Nvidia Corporation.

References

[1]. D. Yu, L. Deng, Automatic Speech Recognition – A Deep Learning Approach, *Springer*, 2015.

[2]. J. Li, L. Deng, Y. Gong, R. Haeb-Umbach, An overview of noise-robust automatic speech recognition, *IEEE/ACM Trans. on Audio, Speech and Language Processing*, Vol. 22, Issue 4, 2014, pp. 745-777.

[3]. R. Caruana, Multitask learning, *Journal of Machine Learning Research*, Vol. 28, 1997, pp. 41-75.

[4]. M. Seltzer, J. Droppo, Multi-task learning in deep neural networks for improved phone recognition, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'13)*, 2013, pp. 6965-6969.

[5]. P. Bell, S. Renals, Regularization of deep neural networks with context-independent multi-task training, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'15)*, 2015, pp. 4290-4294.

[6]. Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, *Journal of Machine Learning Research*, Vol. 17, 2016, pp. 1-35.

[7]. Y. Shinohara, Adversarial multi-task learning of deep neural networks for robust speech recognition, in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'16)*, 2016, pp. 2369-2372.

[8]. P. Denisov, N. Vu, F. Font, Unsupervised domain adaptation by adversarial learning for robust speech recognition, in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'16)*, 2016, pp. 2369-2372.

[9]. S. Sun, C. Yeh, M. Huang, M. Ostendorf, L. Xie, Domain-adversarial training for accented speech recognition, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'18)*, 2018, pp. 4854-4858.

[10]. Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gong, B. Juang, Speaker-invariant training via adversarial learning, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'18),* 2018, pp. 5969-5973.

[11]. L. Tóth, Phone recognition with hierarchical convolutional deep maxout networks, *EURASIP*

Journal on Audio, Speech and Music Processing, 2015, 25.

[12]. L. Tóth, T. Grósz, A. Markó, T. Csapó, Multi-task learning of speech recognition and speech synthesis parameters for ultrasound-based silent speech interfaces, in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'18)*, 2018, pp. 3172-3176.

[13]. K. J. Han, S. Kim, S. S. Narayanan, Strategies to improve robustness of Agglomerative Hierarchical Clustering under data source variation for speaker diarization, *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 16, Issue 8, 2008, pp. 1590-1601.

[14]. W. Wang, P. Lu, Y. Yan, An improved speaker clustering, *Acta Acustica*, Vol. 33, Issue 1, 2008, pp. 9-14.

[15]. H. Kaya, A. Karpov, A. Salah, Fisher vectors with cascaded normalization for paralinguistic analysis, in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'15)*, 2015, pp. 909-913.