AUTOMATIC LOCALIZATION OF A LANGUAGE-INDEPENDENT SUB-NETWORK ON DEEP NEURAL NETWORKS TRAINED BY MULTI-LINGUAL SPEECH

Shigeki Matsuda, Xugang Lu, and Hideki Kashioka

Spoken Language Communication Laboratory, National Institute of Information and
Communications Technology, Kyoto, Japan
{shigeki.matsuda,xugang.lu,hideki.kashioka}@nict.go.jp

ABSTRACT

Deep neural networks (DNNs) have been successfully applied to automatic speech recognition (ASR). However, no study has investigated the possibility of building a language-independent sub-network DNN as the basis for further training of any new language using a simple plug-in of the sub-network. In this paper, we propose a novel technique to split a DNN into language-independent and -dependent sub-networks using multi-lingual speech training data. Our basic assumption is that, in a DNN for speech processing, language-independent feature processing is done in stages that are near to the input layer, while language-dependent processing is performed in stages that are near to the output layer. Based on this assumption, we propose a technique to simultaneously optimize multiple sub-networks in a DNN trained with multi-lingual speech data. The language-dependent and -independent processing boundaries in individual sub-networks are segmented automatically. We test our technique in phoneme classification experiments. The results demonstrate that a language-independent sub-network DNN extracted by our technique can be used as a universal network for speech processing of additional new languages.

Index Terms— Speech Recognition, Deep Neural Network, Restricted Boltzmann Machine

1. INTRODUCTION

Deep neural networks (DNNs) have been successfully applied to many applications, such as image processing and speech recognition [1, 2, 3]. The training of a DNN consists of two steps: One is greedy-layer wised unsupervised pre-training, while the other is supervised fine-tuning. In the pre-training, each layer is regarded as a restricted Boltzmann machine (RBM) and is trained based on a contrastive divergence (CD) algorithm [4]. The input to each layer is the output of the lower layer. After pre-training, all layers are stacked to make a DNN, and they are fine-tuned by using the conventional back propagation (BP) algorithm [5].

In recent years, DNN have been applied to automatic speech recognition (ASR). Speech recognition systems using DNN to calculate state output probabilities in a Hidden Markov Model (HMM) have achieved higher performances than Gaussian mixture model HMMs (GMM-HMMs) trained with discriminative training such as Boosted Maximum Mutual Information (BMMI) [6]. Many state-of-the-art performances in large vocabulary continuous speech recognition (LVCSR) tasks based on DNNs have been reported [7]. The basic advantage of using a DNN is that the state output probability distribution can be more accurately estimated than when using traditional GMM. In addition, by adding many hidden layers in training a DNN, a more robust generalization ability is achieved compared to using a shallow neural network, which has only one or a few hidden layers, as traditionally used for many years.

Although many successful applications of DNNs have been reported in ASR, no study has investigated the possibility of building a language-independent sub-network DNN that can be used as the basis for further training of any newly introduced language using a simple plug-in of the sub-network. This is a very important issue because if there were such a universal DNN, we could quickly build a DNN ASR for any additional language without training the DNN from the very beginning using a huge amount of training data. In this study, we investigate the possibility of finding such a language-independent sub-network in a DNN by using multi-lingual speech-training data.

Our assumption is based on the basic intuition that a low-level neural network performs basic acoustic feature detection that is common to all kinds of acoustic events, such as frequency and temporal transitions, while a high-level neural network detects specific features, such as vowels, stops, fricatives, or possibly language-dependent features. Based on this assumption, in a DNN for speech processing, language-independent feature processing is done in stages that are near to the input layer, while language-dependent processing is performed in stages that are near to the output layer. Under this assumption, we propose a novel technique to split a DNN into language-independent and dependent sub-networks using multi-lingual speech-training data. The proposed technique can simultaneously optimize multiple sub-networks in

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

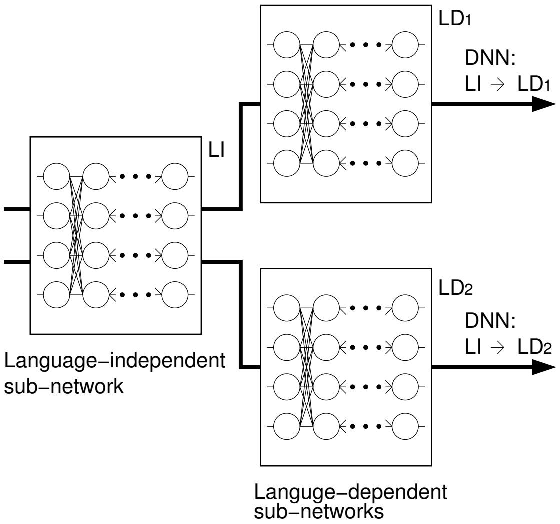
LD1
DNN:
LI → LD1
LI
LD2
DNN:
LI → LD2
Language−independent sub−network
Languge−dependent sub−networks

Fig. 1. *Connections between language-independent and language-dependent sub-networks*

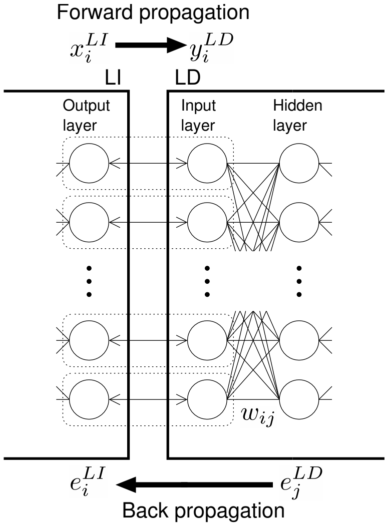

Fig. 2. *Forward and back propagation between sub-networks*

a DNN trained with multi-lingual speech data. The language-dependent and -independent processing boundaries in individual sub-networks are segmented automatically. We tested our technique by phoneme-classification experiments to confirm its effectiveness.

In bottle-neck feature based multi-lingual ASR, some researchers have proposed to adapt an MLP trained by multi-lingual speech to new language[8, 9]. Our work is different from their work since we try to expressly figure out a language-independent sub-network from deep networks.

The rest of this paper is organized as follows. Section 2 describes a technique for optimizing multiple sub-networks simultaneously and segmenting processing boundaries in individual sub-networks automatically. In Section 3, our proposed technique is evaluated by phoneme-classification experiments using Japanese, English, and Chinese speech data. Section 4 gives our conclusions, and discuses the promise of using a language-independent sub-network extracted by our proposed technique for additional new languages.

2. SIMULTANEOUS OPTIMIZATION OF MULTIPLE SUB-NETWORKS

In this section, we describe our proposed technique for optimizing DNNs consisting of multiple sub-networks. As described in the introduction, language dependent sub-network is located in upper layers to process language-specific information, while language independent sub-network in lower layers deals with basic acoustic feature detection which is common to all acoustic events. This idea is illustrated in Fig. 1 where a language-independent sub-network (LI) is connected to language-dependent sub-networks (LD$_n$). These connections between sub-networks represent multiple DNNs such as LI→LD$_1$ and LI→LD$_2$ as shown in Fig. 1.

In pre-training, the LI sub-network is pre-trained using multi-lingual speech data for initialization. The LD$_n$ sub-networks can be pre-trained in two types of ways for initialization. One way is to pre-train each LD$_n$ sub-network using the output calculated from LI sub-network for speech data of a language depending on the LD$_n$ sub-network. Another way is to pre-train the LD$_n$ sub-network using the output calculated from LI for multi-lingual speech data [10]. In this study, the later pre-training method is used for each of LI and LD$_n$ sub-networks.

In fine-tuning, a DNN including the common LI sub-network and one LD$_n$ sub-network is selected alternately. The selected DNN is trained using mini-batches consisting of several frames for parameter updating. All sub-networks are optimized simultaneously by estimating multiple DNNs (sharing the same LI sub-network) using a BP algorithm for all training languages. This estimation approach is similar to “embedded training” in conventional HMM parameter estimation for acoustic model where multiple phone models are updated simultaneously. In one DNN, how many layers are related to LI sub-network (and the left layers are related to LD sub-network) can be optimized in the training.

Neuron connections between sub-networks are shown in Fig.2. In the forward propagation phase, the input y_i^{LD} is set as the output x_i^{LI}.

$$x_i^{LI} = y_i^{LD} \tag{1}$$

where i is a index of neuron. The number of output neurons in LI sub-network should be the same as that in the input neurons of LD sub-network. In the back propagation phase, the error

Table 1. Phoneme classification rate of baseline system (%)

	Lang.		
# of layers	Jp	En	Ch
2	71.69	51.09	60.37
4	73.28	51.51	61.58
6	75.13	54.50	62.58

signal e_i^{LI} of neurons in the output layer of LI sub-network is calculated from e_j^{LD} of hidden neurons in the second layer of LD sub-network as shown by the following equation.

$$e_i^{LI} = x_i^{LI}(1 - x_i^{LI}) \sum_j w_{ij} e_j^{LD} \tag{2}$$

This technique can be applied to multiple DNNs, depending on the relationships among them. Processing boundaries between sub-networks are segmented in a relative way.

3. EXPERIMENTS

3.1. Experimental conditions

We tested our proposed technique on frame-level phoneme classification experiments using Japanese, English and Chinese speech data. Speech data collected from “VoiceTra” [11] were used for this evaluation. Our institute (NICT) released a network-based multilingual speech-translation system called “VoiceTra” as an application for smart phones at the end of July 2010. It is intended to be used in communication with foreign visitors in Japan and in conversations with local people on trips abroad. The acoustic parameters consisted of 12 MFCCs, log pow, 12 ΔMFCCs, Δ log pow, 12 $\Delta\Delta$MFCCs, and $\Delta\Delta$ log pow (a total of 39 dimensional acoustic features), extracted from frames of 20-ms length with 10-ms frame shifts. In all, 429 dimensional acoustic features consisting of 11 frames (including 5 preceding and succeeding frames) were used as input vectors of DNNs. The number of Japanese, English, and Chinese phonemes were 26, 39, and 30 respectively. The number of output neurons in DNNs is the same as the number of phonemes for each language. The training data included 40,000 utterances (about 25 hours) for each language. The number of evaluation data is 1,000. Each utterance has a terminal identification data (terminal ID), and any terminal ID in the evaluation data is not included in the training data. In pre-training, a fixed learning rate of 0.005 was used for estimating RBM parameters. Number of epochs was 100. In fine-tuning, the learning rate started at 0.001. If the error in development data increased, the learning rate was halved. The number of development data is 2,000. Any terminal ID in development data is not included in the training and evaluation data. RBMs and DNNs were trained with a mini-batch size of 128.

Table 2. Phoneme classification performance (%) of DNNs (LI$_{Jp,En}$→LD$_{En}$ and LI$_{Jp,En}$→LD$_{Jp}$)

# of layers		Lang.	
LI	LD	En	Ja
1	5	54.37	74.73
2	4	54.61	74.86
3	3	54.63	74.90
4	2	54.59	74.82
5	1	52.50	73.38

Table 3. Phoneme classification performance (%) of DNNs (LI$_{Jp,Ch}$→LD$_{Ch}$ and LI$_{Jp,Ch}$→LD$_{Jp}$)

# of layers		Lang.	
LI	LD	Ch	Ja
1	5	62.89	74.60
2	4	62.93	74.67
3	3	63.10	74.85
4	2	63.08	74.70
5	1	62.05	74.08

3.2. Baseline performance

We evaluated phoneme classification performances using conventional DNNs for each of the three languages as baseline systems. Total number of layers in DNNs were two, four, and six, excluding input layer. The number of hidden neurons was 512 in each layer. In pre-training, DNNs were initialized using Japanese, English and Chinese speech data. Table 3 shows the experimental results. In the table, “Jp”, “En”, “Ch” are Japanese, English and Chinese respectively. Experimental results show that phoneme classification performances were consistently improved by increasing the number of hidden layers.

3.3. Estimation of a language-independent sub-network

To estimate a language-independent sub-network that can be used for additional languages, two DNNs (LI$_{Jp,En}$→LD$_{Jp}$ and LI$_{Jp,En}$→LD$_{En}$) consisting of three sub-networks (LI$_{Jp,En}$, LD$_{Jp}$, and LD$_{En}$) were pre-trained and fine-tuned using Japanese and English speech data. We also performed same experiments on different language pairs of Japanese and Chinese. Sigmoid function was used as the output function of hidden and output neurons in the LI and hidden neurons in the LD. Softmax function was used for output neurons in the LD. The number of hidden neurons was 512 for each layer, the same as that in the baseline system. The number of layers in these DNNs was six.

Table 2 shows phoneme classification performance obtained using two DNNs (LI$_{Jp,En}$→LD$_{Jp}$ and LI$_{Jp,En}$→LD$_{En}$), and Table 3 shows this phoneme classification performance using other DNNs (LI$_{Jp,Ch}$→LD$_{Jp}$ and LI$_{Jp,Ch}$→LD$_{Ch}$).

Table 4. Phoneme classification performance (%) of DNNs (LI$_{Jp,En}$→LD$_{Ch}$ and LI$_{Jp,Ch}$→LD$_{En}$)

# of layers		Lang.	
LI	LD	Ch	En
1	5	62.44	53.00
2	4	63.50	54.74
3	3	63.57	55.66
4	2	62.63	55.34
5	1	59.29	51.18

As shown in Tables 2 and 3, the phoneme classification performance when using LD with more than two layers (using LI with less than four layers) was almost the same as the baseline performance. On the other hand, the phoneme classification performance of DNNs with only one-layer LD was degraded. This may be due to an insufficient number of parameters used for language-dependent processing in the LD. A combination of three-layers LI and three-layers LD archived slightly better performance compared with other combinations. We can deduce that the estimation accuracy of LI was improved by sharing parameters among multiple languages.

3.4. Evaluation for additional new language

We evaluated the phoneme classification performance of two DNNs (LI$_{Jp,En}$→LD$_{Ch}$ and LI$_{Jp,Ch}$→LD$_{En}$). In the pre-training phase, the LD$_{Ch}$ and LD$_{En}$ were initialized using the output of LI$_{Jp,En}$ and LI$_{Jp,Ch}$ using Japanese, English and Chinese speech data respectively. In the fine-tuning phase, only the parameters of the LD$_{Ch}$ and LD$_{En}$ were updated, while the parameters of LI$_{Jp,En}$ and LI$_{Jp,Ch}$ were fixed.

As shown in Table 4, even though only the parameters of language-dependent sub-networks (LD$_{Ch}$ and LD$_{En}$) were updated, the phoneme classification performances of these DNNs were similar to those of the baseline systems except for one-layer LD. From these experiments, we can confirm that a language-independent sub-network DNN estimated by our proposed technique can be used as a universal network for speech processing of additional new languages.

4. CONCLUSION

In this paper, we proposed a novel technique to split a DNN into language-independent and -dependent sub-networks using multi-lingual speech training data. Our technique optimizes simultaneously multiple sub-networks in a DNN trained with multi-lingual speech data. Experimental results demonstrate that a language-independent sub-network DNN extracted by our proposed technique can be used as a universal network for speech processing of other newly introduced languages.

In future work, we will evaluate our technique on LVCSR tasks and applying this technique for building multi-lingual ASR based on DNN. Moreover, it is difficult to train synapse weights that are near to the input layer when estimating an MLP with many hidden layers by the conventional BP algorithm. We will try to use more efficient training algorithm such as Hessian free estimation [12] to train LI sub-network precisely.

5. REFERENCES

[1] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, Vol. 2, No. 1, pp. 1–127, 2009.

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

[3] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic Modeling using Deep Belief Networks," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No. 1, pp. 14–22, 2012.

[4] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, Vol. 14, No. 8, pp. 1771–1800, 2002.

[5] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, Vol. 323, pp. 533–536, 1986.

[6] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature space discriminative training," In *Proc. ICASSP*, pp. 4057–4060, 2008.

[7] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No. 1, pp. 30–42, 2011.

[8] N. T. Vu, F. Metze, and T. Schultz, "Multilingual bottleneck features and its application for under-resourced languages," Proc. SLTU, 2012.

[9] S. Thomas, S. Ganapathy, and H. Hermansky, "Multilingual MLP features for low-resource LVCSR systems," In *Proc. ICASSP*, pp. 4269–4272, 2012.

[10] P. Swietojanski, A. Ghoshal, and A. Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," In *Proc. IEEE Workshop on Spoken Language Technology*, 2012.

[11] NICT Translation Apps `http://mastar.jp/translation/voicetra-en.html`

[12] J. Martens, "Deep learning via Hessian-free Optimization," In *Proc. ICML*, pp. 735–742, 2010.