MULTILINGUAL ACOUSTIC MODELS USING DISTRIBUTED DEEP NEURAL NETWORKS

G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, J. Dean

Google Inc., USA

ABSTRACT

Today's speech recognition technology is mature enough to be useful for many practical applications. In this context, it is of paramount importance to train accurate acoustic models for many languages within given resource constraints such as data, processing power, and time. Multilingual training has the potential to solve the data issue and close the performance gap between resource-rich and resource-scarce languages. Neural networks lend themselves naturally to parameter sharing across languages, and distributed implementations have made it feasible to train large networks. In this paper, we present experimental results for cross- and multi-lingual network training of eleven Romance languages on 10k hours of data in total. The average relative gains over the monolingual baselines are 4%/2% (data-scarce/data-rich languages) for cross- and 7%/2% for multi-lingual training. However, the additional gain from jointly training the languages on all data comes at an increased training time of roughly four weeks, compared to two weeks (monolingual) and one week (crosslingual).

Index Terms— Speech recognition, parameter sharing, deep neural networks, multilingual training, distributed neural networks

1. INTRODUCTION

Speech recognition has advanced remarkably over the last decade and is used in a growing number of applications and services such as Google Voice Search [1]. As a consequence, there is often the need to efficiently train accurate acoustic models for a large number of languages.

Traditionally, languages (and dialects) are considered independently, and a separate acoustic model is trained for each language from scratch. Recently, excellent results have been achieved using Deep Neural Networks (DNNs) for acoustic modeling [2, 3, 4, 5, 6]. Potential limitations of this monolingual approach include the training cost, in particular for DNNs [2, 3, 4, 5], and the limited data for many languages. This usually results in considerable differences in quality between resource-rich and resource-scarce languages, for example, because only small models with low complexity can be estimated for the latter. In general, data scarcity is not only a problem of costly data collection but also an unavoidable bottleneck for languages with low traffic and new launches where it is difficult to find large amounts of representative data.

Multitask learning is an attractive alternative to the traditional, single-task approach described above. In this approach, multiple tasks are learned in parallel and use a shared representation [7]. An architecture for multilingual speech recognition is shown in Fig. 1. In this example, the feature extraction is shared whereas there is a separate classifier for each language. Multitask learning may be beneficial for several reasons. For example, [8, 9] show that learning based on knowledge transfer generalizes better. According to [7, 10], the mechanisms that help multitask learning for DNNs include representation bias (local optima supported by all tasks are preferred), overfitting prevention (more reliable feature estimation, in particular for resource-scarce tasks), eavesdropping (learning some features may be easier in a parallel task), data amplification (the extra information of different noise added to the same feature may help), and attribute selection (may select better features).

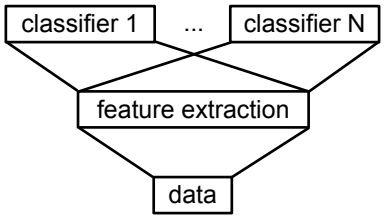

Fig. 1. Example for a multilingual architecture for speech recognition with a language-independent feature extraction and language-specific classifiers on top of it.

The contributions of this paper include an experimental evaluation of the multilingual approach based on DNNs with a shared feature extraction, as summarized in Fig. 2 (b). The multilingual DNN includes eleven Romance languages with different types of data, including supervised/unsupervised and read/spontaneous data, and different amounts of training data per language dialect, ranging from 100 hours to 1500 hours and amounting to roughly 10k hours of data in total. For the joint training of the languages, the implementation for distributed DNNs introduced in [11, 12] is used.

Note that "resource-scarce" is relative to the model size. For the network size used in this paper, a few hundred hours of data implies data scarcity. In particular, our main goal is to train more accurate networks for the given data - rather than to bootstrap a language with little or no data from another language by using model adaptation [13, 14], a tandem approach [15, 16, 17, 18, 19], a phone mapping [20], unsupervised pre-training [21], initialization with an existing neural network [22, 19, 23], or building a language-universal acoustic model based on a shared phone set [24]. Also, multilingual recognition including language identification is beyond the scope of this paper [20]. The multilingual approach in [25] is similar to our approach in that the phones of the different languages are kept separate and parts of the model are shared across the languages (the states of the subspace Gaussian mixture models in case of [25]). An overview on cross- and multi-lingual speech recognition can be found in [26].

The remainder of the paper is organized as follows. Section 2 describes the multilingual DNN used in this paper in more detail and compares it to existing variants. Section 3 summarizes the distributed DNN implementation from [11, 12]. The experimental evaluation is given in Section 4. The paper concludes with Section 5.

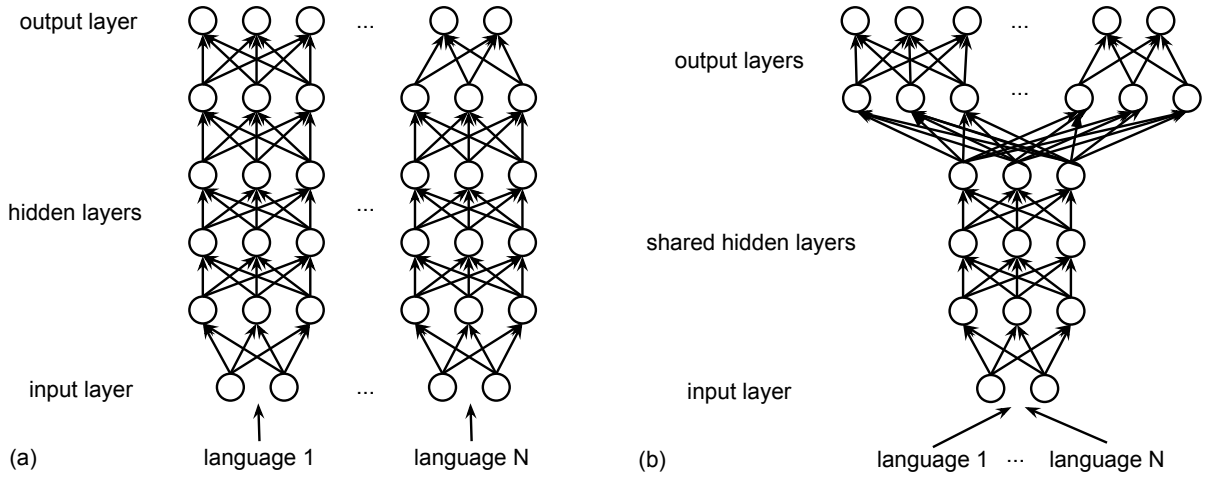

Fig. 2. (a) Monolingual neural networks and (b) the corresponding multilingual neural network. Comparing with Fig. 1, the (shared) hidden layers correspond with the feature extraction and the output layer(s) correspond with the classifier.

2. CROSS- AND MULTI-LINGUAL APPROACHES

This section describes different approaches to learning based on knowledge transfer. Two examples for crosslingual training are feature learning (Section 2.1) and transfer learning (Section 2.2). Multilingual training is an instance of multitask learning (Section 2.3), which is the main topic of this work.

2.1. Feature Learning

Feature learning has been used in speech recognition for discriminative features or tandem features [15, 16, 17]. In this approach, features based on neural networks, for example, are trained using data from one [17] or multiple [18] languages. On top of these features, a comparably lightweight classifier (for example, Gaussian mixture models or a neural network with only a couple of layers) is trained for another language, keeping the features fixed (see Fig. 1). For the model given in Fig. 2 (a), feature learning uses the weights of the bottom three hidden layers of a source network in the target networks and keeps these weights fixed during training of the target networks. This approach is efficient and can learn sophisticated features, if there is sufficient data for the source language, while avoiding overfitting for resource-scarce languages. The drawback of this approach may be that the features and the classifier are not jointly optimized.

2.2. Transfer Learning

Transfer learning [27] refers to the situation where the network of a target language is initialized with an existing source network, for example, by pre-training or fine-tuning [21, 22, 19, 23], and is not trained from scratch. In Fig. 2 (a), this means that (some of) the weights from a monolingual network are used to initialize the corresponding weights of another monolingual network. Unsupervised pre-training [28] is similar but is expected to be less effective than transfer learning because it does not use crosslingual data and the training criterion does not directly optimize discrimination. Transfer training may be a good choice for fast bootstrapping of neural networks. This approach is limited because knowledge transfer is only possible from the source to the target task and only biases the initialization. Overfitting remains an issue, unless the hypothesis space is reduced by using regularization (similar to model adaptation [13, 14]) or keeping the bottom hidden layers fixed during training (see [15] and feature learning in Section 2.1).

2.3. Multitask Learning

In this approach, multiple tasks are learned in parallel and use a shared representation [7, 25]. In this paper, we use an architecture based on DNNs with a shared feature extraction and language-specific classifiers, see Fig. 1 and Fig. 2 (b). In particular, the feature extraction and the classifiers are jointly optimized on the shared data for the different languages. As a side effect, it is expected that multitask learning is less sensitive to the optimal tuning of the network size. Feature learning (Section 2.1) and transfer learning (Section 2.2) can be considered (efficient) approximations of this implementation of multitask learning. These "approximations," however, only cover the mechanisms of representation bias and overfitting prevention. Full multitask learning is required for the other mechanisms of eavesdropping, data amplification, and attribute selection (see Section 1), which come at the expense of a training algorithm with a high complexity and that is hard to parallelize.

3. DISTRIBUTED DEEP NEURAL NETWORKS

Stochastic gradient descent (SGD) [29] is the most commonly used optimization procedure for training DNNs [2, 3, 4, 5]. Unfortunately, the traditional formulation of SGD is inherently sequential, making scaling networks and data sets difficult. For this reason, GPU-based implementations for DNN training have become the standard, in order to reduce the per-step training times [2, 3, 4, 5]. Instead of GPUs, we use the software framework DistBelief proposed in [11, 12] that supports distributed computation on multiple CPUs in DNNs. DistBelief is briefly summarized in this section.

DistBelief includes two complementary types of parallelism: distributed optimization over multiple model instances, and distributed computation within each model instance. For each model instance, the framework distributes computation across several machines, automatically parallelizing computation within each machine using all available cores, and managing communication, synchronization and data transfer between machines.[1] To distribute

[1]The performance benefits of distributing a deep network across multiple machines depends on the connectivity structure and computational needs of the model, see [10] for benchmarks.

optimization across many such model instances, DistBelief uses a variant of asynchronous stochastic gradient descent, Downpour SGD.

The basic approach to Downpour SGD is as follows. We divide the training data into a number of subsets and run a copy of the model on each of these subsets. Models periodically update their copies of the model parameters by requesting fresh values from the parameter server. The models send updates to a centralized parameter server, which keeps the current state of all parameters for the model, sharded across many machines (see Fig. 3). This approach

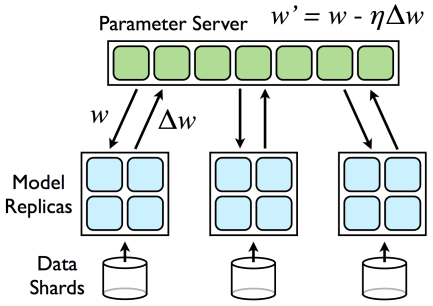

Fig. 3. Downpour SGD. Model replicas asynchronously fetch parameters w and push gradients w to the parameter server.

is asynchronous in two distinct aspects: the model replicas run independently of each other, and the parameter server shards also run independently of one another. In the simplest implementation, before processing each mini-batch, a model replica asks the parameter server service for an updated copy of its model parameters. Because DistBelief models are themselves partitioned across multiple machines, each machine needs to communicate with just the subset of parameter server shards that hold the model parameters relevant to its partition. After receiving an updated copy of its parameters, the DistBelief model replica processes a mini-batch of data to compute a parameter gradient, and sends the gradient to the parameter server, which then applies the gradient to the current value of the model parameters. Downpour SGD is more robust to machines failures than standard (synchronous) SGD because if one machine in a model replica fails, the other model replicas continue processing their training data and updating the model parameters via the parameter servers. On the other hand, the multiple layers of asynchronous processing in Downpour SGD introduce a great deal of additional stochasticity in the optimization procedure. A model replica is almost certainly computing its gradients based on a set of parameters that are slightly out of date because some other model replica will likely have updated the parameters on the parameter server in the meantime. The Adagrad [30] adaptive learning rate procedure uses a separate, adaptive learning rate for each parameter rather than a single, fixed learning rate on the parameter server (η in Fig. 3). The use of Adagrad for Downpour SGD increases the maximum number of model replicas that can productively work simultaneously and has virtually eliminated stability concerns in training DNNs. Adagrad usually requires "warmstarting" the model training with one or a few model replicas before activating the other replicas.

DistBelief lends itself naturally to multitask learning as it allows for training with a large number of (partially overlapping) model replicas. Each task processes its data on its own set of machines. The computed gradients are sent to the parameter server, which updates the parameters in a consistent way and sends the updated parameters back.

4. EXPERIMENTAL RESULTS

This section provides experimental results for multilingual training for a set of Romance languages.

4.1. Data & Setup

The experiments are performed on the Romance languages including Catalan, different Spanish dialects, French, Italian, two Portuguese dialects, Romanian, and Basque (which is not a Romance language but a language isolate surrounded by Romance languages), see Table 1. The type of training data varies between the languages and is a realistic mix of read/spontaneous speech and supervised/unsupervised data coming from different sources. The amount of available training data for each language is shown in Table 1. The test sets consist of data from Voice Search, Voice IME such as dictation or read test data, including five hours/25000 words or more per language. The data for ca-ES, eu-ES, pt-PT, and ro-RO are read speech whereas the remaining 95% of the data are spontaneous speech.

The setup for DNN training and the hybrid decoding is based on the setup in [31]. The DNN is bootstrapped from a standard HMM-based system using discriminately trained Gaussian Mixture Models (GMMs) without speaker adaptation. The number of Gaussian densities depends on the amount of data, ranging from 33000 to 250000. The input for the DNN is eleven contiguous frames of 40-dimensional log-filterbank features. The DNN consists of four hidden layers each with 2560 nodes and logistic activation, and an output layer with softmax activation representing the context-dependent states from the baseline GMM model, see Fig. 2. The number of context-dependent states differs from language to language and depends on the available data, see Table 1. Unsupervised pre-training [28] is not used as it has not helped in control experiments. A standard trigram language model is used for decoding. For multilingual training, the bottom three hidden layers are shared. The optimization of the DNNs is done in the distributed implementation described in Section 3 using a variant of SGD. The stopping criterion is early stopping (the frame accuracy on held-out data degraded slightly after a while in almost all training runs) or when there is no change in frame accuracy on the held-out data for a relatively long time (on the order of a week).

4.2. Detailed Analysis for pt-PT

Next, a detailed analysis of the results is given for pt-PT, which is considered a resource-scarce language for which the effects of cross- and multi-lingual training are expected to be most significant.

Fig. 4 shows the word error rate over the training time for the different initialization and training scenarios. The baseline is the DNN trained from scratch with in-language data only (column 'from scratch' in Fig. 4). Transfer learning, i.e., initializing the pt-PT DNN with a DNN of another language gives an absolute gain of 1% in word error rate ('from en-US, all layers'). Here, we used an American English DNN (en-US) trained on 3000 hours of data for roughly two weeks to guarantee consistency of the results throughout the paper. DNNs of other languages, for example, pt-BR, were also tested and give similar results. Keeping the bottom hidden layers in the DNN fixed during training (feature learning), helps to avoid overfitting ('from en-US, last 3 layers' etc.). Compared to training from scratch, feature and transfer learning not only give better word error rates but also tend to converge faster.

The plot suggests that only training the last two layers (the output layer and the last hidden layer) is a good operating point in terms of word error rate, overfitting, and convergence speed. The multilingual DNN shown in Fig. 2 (b) was chosen based on this observation,

Table 1. Word error rates (%) for cross- and multi-lingual training. The format 'X / -Y%' for DNN means word error rate/relative gain where the relative gain is given for monolingual training over GMM and for cross-/multi- over mono-lingual training.

					Monolingual		Crosslingual		Multilingual
					GMM	DNN			
						from scratch	from en-US		
	Language	Region	Train data (hours)	#States		all layers	top 2 layers	all layers	
eu-ES	Basque	Spain	80	1600	18.9	16.3 / -14%	16.2 / -1%		15.4 / -6%
ca-ES	Catalan	Spain	100	3300	25.7	22.1 / -14%			19.9 / -10%
pt-PT	Portuguese	Portugal	100	2900	25.7	21.8 / -15%	20.9 / -4%	21.0 / -4%	20.5 / -6%
ro-RO	Romanian	Romania	220	5700	16.9	12.9 / -24%	12.1 / -6%		11.7 / -9%
es-AR	Spanish	Argentina	270	3500	48.4	40.2 / -17%	38.8 / -3%		37.7 / -6%
es-419	Spanish	Latin Amer. & Carib.	920	5500	49.5	39.7 / -20%			38.4 / -3%
fr-FR	French	France	1140	6200	33.7	30.7 / -9%			30.2 / -2%
pt-BR	Portuguese	Brazil	1450	4700	36.4	31.1 / -15%	31.1 / -0%	30.7 / -1%	30.7 / -1%
it-IT	Italian	Italy	1460	5100	19.4	16.0 / -18%		15.6 / -3%	15.6 / -3%
es-ES	Spanish	Spain	1490	4900	31.2	25.8 / -17%		25.4 / -2%	25.1 / -3%
es-MX	Spanish	Mexico	1490	3700	49.8	37.3 / -25%		36.8 / -1%	36.5 / -2%

i.e., the bottom three hidden layers are shared while the top hidden layer and the output layer are language-specific.

Multilingual training gives another absolute gain of 0.5% in word error rate over transfer learning, without overfitting. From this, we conclude that for resource-scarce languages, the joint optimization of the feature extraction and the classifiers on all data helps in addition to the better initialization. In terms of the mechanisms for multitask learning (Section 1), this implies that beside representation bias and overfitting prevention (which are also active for feature and transfer learning), eavesdropping, data amplification or attribute selection are relevant mechanisms as well. On the downside, the training time is significantly higher than for crosslingual training: the number of epochs needed for convergence is roughly the same but the time spent per epoch is significantly larger because the other ten languages simultaneously update the shared hidden layers. To be fair, the training times for the resource-rich languages are much larger, say, up to one week for crosslingual and up to two weeks for monolingual training. More recent, internal experiments suggest that the training time of multilingual DNN can be reduced considerably by a better initialization scheme (for example, train the DNNs for each language separately with feature learning first).

4.3. Multilingual Network for Romance Languages

Table 1 summarizes the word error rates for all eleven Romance languages. This table includes results for the GMM baseline and different variants of DNNs for comparison. First, the DNNs ('DNN, from scratch') outperform the GMMs ('GMM'). Across the different languages and conditions, the DNNs reduce the word error rate by 10%-25% relatively. Second, cross- and multi-lingual training ('DNN, from en-US' and 'DNN, multilingual') consistently yield better word error rates than monolingual training ('DNN, from scratch'). For the resource-scarce languages, overfitting is a severe issue for mono- and cross-lingual training. For this reason, only the last two layers are updated in crosslingual training in this case ('last 2 layers' vs. 'all layers'). In general, early stopping is essential for mono- and cross-lingual training to obtain competitive error rates. Finally, multilingual training ('DNN, multilingual') often gives an additional gain over crosslingual training ('DNN, from en-US'), in particular for the resource-scarce languages with only a few hundred hours of data.

In summary, the conclusions from Section 4.2 for the resource-scarce language pt-PT carry over to all Romance languages, although the effect of multitask learning tends to be the less pronounced the more training data there is.

5. SUMMARY

We presented an empirical comparison of mono-, cross-, and multilingual acoustic model training using deep neural networks. Experiments were performed for eleven Romance languages with a total amount of 10k hours of data. This large-scale experiment is enabled by a highly distributed software framework for deep neural networks. Our results for crosslingual training support the findings by other groups at smaller scale that crosslingual training outperforms monolingual training (up to 6% relative gain in word error rate). Furthermore, it was shown that multilingual training can give an additional gain on top of crosslingual training, which tends to be larger the smaller the amount of data is (up to 5% relative gain). This suggests that joint optimization of the languages on the shared data can be beneficial, in addition to using discriminative features trained on a single language and improved model initialization, although at the cost of considerably increased training times. In the future, we will try to train larger multilingual networks to better exploit the available data.

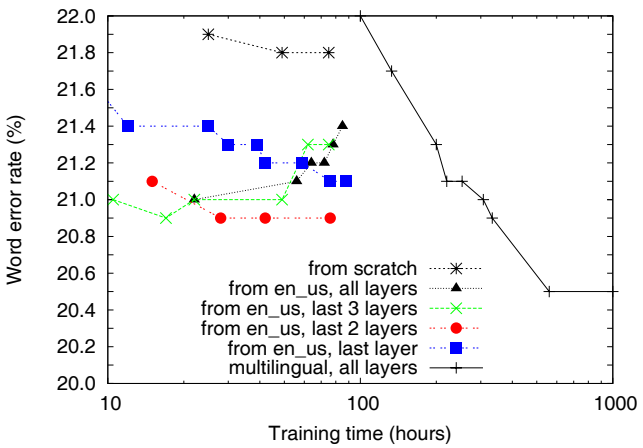

Fig. 4. Progress of word error rate with training time for pt-PT for different initialization and training scenarios.

6. REFERENCES

[1] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope, ""Your word is my command": Google search by voice: A case study," in *Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics*, chapter 4, pp. 61–90. Springer, 2010.

[2] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *INTERSPEECH*, 2011, pp. 437–440.

[3] G.E. Dahl, D. Yu, and L. Deng, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," in *ICASSP*, 2011.

[4] 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, Special Issue on Deep Learning for Speech and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[5] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[6] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *ICASSP*, 2013.

[7] K. Kovač, "Multitask learning for Bayesian neural networks," M.S. thesis, University of Toronto, 2005.

[8] S. Thrun, "Is learning the n-th thing any easier than learning the first?," in *Advances in Neural Information Processing Systems*, 1996, vol. 8, pp. 640–646.

[9] J. Baxter, "A model of inductive bias learning," *Journal of Artificial Intelligence Research*, vol. 12, pp. 149–198, 2000.

[10] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, pp. 41–75, 1997.

[11] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng, "Building high-level features using large scale unsupervised learning," in *International Conference on Machine Learning*, 2012, pp. 81–88.

[12] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, "Large scale distributed deep networks," in *Advances in Neural Information Processing Systems*, 2012, to appear.

[13] F. Diehl, *Multilingual and Crosslingual Acoustic Modelling for Automatic Speech Recognition*, Ph.D. thesis, Universitat Politècnica de Catalunya, 2007.

[14] J. Zheng and A. Stolcke, "fMPE-MAP: Improved discriminative adaptation for modeling new domains," in *INTERSPEECH*, 2007, pp. 1573–1576.

[15] A. Stolcke, F. Grezl, M. Hwang, X. Lei, N. Morgan, and D. Vergyri, "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," in *ICASSP*, 2006, pp. 321–324.

[16] S. Thomas, S. Ganapathy, and H. Hermansky, "Cross-lingual and multi-stream posterior features for low-resource LVCSR systems," in *INTERSPEECH*, 2010.

[17] C. Plahl, R. Schlüter, and H. Ney, "Cross-lingual portability of Chinese and English neural network features for French and German LVCSR," in *ASRU*, 2011, pp. 371–376.

[18] Y. Qian and J. Liu, "Cross-lingual and ensemble MLPs strategies for low-resource speech recognition," in *INTERSPEECH*, 2012.

[19] S. Thomas, S. Ganapathy, and H. Hermansky, "Multilingual MLP features for low-resource LVCSR systems," in *INTERSPEECH*, 2012.

[20] T. Schultz and A. Waibel, "Multilingual and crosslingual speech recognition," in *DARPA Workshop on Broadcast News Transcription and Understanding*, 1998, pp. 259–262.

[21] P. Swietojnski, A. Ghoshal, and S. Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *Workshop on Spoken Language Technology*, 2012.

[22] N. T. Vu, W. Breiter, F. Metze, and T. Schultz, "An investigation on initialization schemes for multilayer perceptron training using multilingual data and their effect on ASR performance," in *INTERSPEECH*, 2012.

[23] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *ICASSP*, 2013.

[24] H. Lin, L. Deng, D. Yu, Y. Gong, A. Acero, and C.-H. Lee, "A study on multilingual acoustic modeling for large vocabulary ASR," in *ICASSP*, 2009, pp. 4333–4336.

[25] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, D. Povey, A. Rastrow, R.C. Rose, and S. Thomas, "Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models," in *ICASSP*, 2010, pp. 4334 – 4337.

[26] H. Bourlard, J. Dines, M. Magimai-Doss, P. Garner, D. Imseng, P. Motlicek, H. Liang, L. Saheer, and F. Valente, "Current trends in multilingual speech processing," *Sadhana*, vol. 36, no. 5, pp. 885–915, 2011.

[27] D. Cireşan, U. Meier, and J. Schmidhuber, "Transfer learning for Latin and Chinese characters with deep neural networks," in *International Joint Conference on Neural Networks*, 2012, pp. 1–6.

[28] G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[29] L. Bottou, "Stochastic gradient learning in neural networks," in *Neuro-Nîmes*, 1991.

[30] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[31] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *INTERSPEECH*, 2012.