INVESTIGATIONS ON SEQUENCE TRAINING OF NEURAL NETWORKS

Simon Wiesler[1], *Pavel Golik*[1], *Ralf Schlüter*[1], *Hermann Ney*[1,2]

[1]Human Language Technology and Pattern Recognition,
Computer Science Department, RWTH Aachen University, Aachen, Germany
[2]LIMSI CNRS, Spoken Language Processing Group, Paris, France

ABSTRACT

In this paper we present an investigation of sequence-discriminative training of deep neural networks for automatic speech recognition. We evaluate different sequence-discriminative training criteria (MMI and MPE) and optimization algorithms (including SGD and Rprop) using the RASR toolkit. Further, we compare the training of the whole network with that of the output layer only. Technical details necessary for a robust training are studied, since there is no consensus yet on the ultimate training recipe. The investigation extends our previous work on training linear bottleneck networks from scratch showing the consistently positive effect of sequence training.

Index Terms— deep neural networks, speech recognition, sequence training, optimization

1. INTRODUCTION

Hybrid deep neural network (DNN) hidden Markov models (HMMs) have become the dominant approach for automatic speech recognition (ASR) [1, 2, 3]. In the conventional hybrid HMM framework [4], neural networks (NNs) are trained on frame-level, usually according to the cross-entropy (CE) criterion. The advantages of such a *frame-discriminative* training are its efficiency and its robustness. Mostly, the training objective is optimized with stochastic gradient descent (SGD) using a GPU-based implementation.

However, frame-discriminative training considers only the emission model. The other knowledge sources of the speech recognition system – the HMM transition model, the pronunciation lexicon, and the language model (LM) – are not taken into account. In contrast, the typical discriminative training criteria used for Gaussian mixture models (GMMs), e.g. the maximum mutual information (MMI) [5, 6] and minimum phone error (MPE) [7] criterion, are defined on sequence-level and include all knowledge sources. In a number of recent works [8, 9, 10, 11, 12], it has been shown that hybrid DNN-HMMs can be improved consistently by training them according to these criteria well-known from GMMs. This type of training is known as *sequence-discriminative training* or short *sequence training*.

DNN sequence training has turned out to be a complex technique and it is not possible to draw clear conclusions from previous work about the best training setup. In particular, some authors found MPE or the closely related minimum Bayes risk (MBR) criterion to perform better than MMI [11], while others found the converse to be true [10, 13]. In addition, several authors observed problems with the stability of sequence training and proposed different modifications of the training criteria [10, 11].

Another debated issue is the choice of the optimization algorithm. Most groups use SGD on utterance-level [10, 11, 12]. Kingsbury et al. [9] found Hessian-free (HF) [14], a second-order batch algorithm, which has been specifically designed for neural network training, to yield improvements over SGD. So far, there has been no comparison of SGD with other batch algorithms.

In this paper, we present experiments on an English broadcast conversations recognition task with our our implementation of sequence training in the RASR toolkit [15]. We compare different different training criteria and optimization algorithms. Our best result is achieved with the batch algorithm Rprop [16]. As a batch algorithm, Rprop can be parallelized straight-forwardly. In contrast to HF, it is simple to implement and has only very few tuning parameters. Furthermore, we extend our previous work on neural networks with a linear bottleneck structure and show that a slight improvement with bottleneck network remains after sequence training.

2. SEQUENCE-DISCRIMINATIVE TRAINING

In the following, we define the hybrid DNN-HMM and the two most common sequence-discriminative training criteria. For notational convenience, we assume there is only a single utterance with a feature sequence $\mathbf{x} = (x_1, \dots, x_T)$ and correct word sequence $\mathbf{w} = (w_1, \dots, w_N)$. The Viterbi alignment of the utterance is denoted by $\mathbf{s} = (s_1, \dots, s_T)$. $\mathcal{L}$ is a (word) lattice representing the most likely word sequences of the utterance. We write $\mathcal{L}(\mathbf{v})$ for the sub-lattice of $\mathcal{L}$ consistent with a word sequence $\mathbf{v}$.

2.1. Hybrid DNN-HMMs

In hybrid DNN-HMM systems, the DNN is used as a model for the posterior probability $p(s|x)$ of an HMM state s given an acoustic observation x. For brevity, we interpret the DNN as a log-linear model with a parameterized feature extractor $\phi_W(x)$ that is not further specified:

$$p_\theta(s|x) = \frac{1}{Z(x)} \exp(\lambda_s^T \phi_W(x) + \alpha_s) \ . \tag{1}$$

Here, $\Lambda = (\lambda_1; \dots; \lambda_S)$ is the weight matrix, $\alpha = (\alpha_1, \dots, \alpha_S)$ is the bias vector, W are the parameters of the feature extractor, and $\theta = (\Lambda, \alpha, W)$ is the tuple with all parameters of the DNN. The factor $Z(x)$ is the normalization constant:

$$Z(x) = \sum_{\bar{s}} \exp(\lambda_{\bar{s}}^T \phi_W(x) + \alpha_{\bar{s}}) \ . \tag{2}$$

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreements no. N287658 (EU-Bridge) and no. 287755 (transLectures). H. Ney was partially supported by a senior chair award from DIGITEO, a French research cluster in Ile-de-France.

The frame-level posterior can be transformed to a quantity which can be used as an emission score in an HMM speech recognizer:

$$p(x|s) = \frac{p_\theta(s|x)p(x)}{p(s)} = C\exp(\lambda_s^T \phi_W(x) + \alpha_s - \ln p(s)) \ . \quad (3)$$

Here, $C > 0$ is a constant which is independent of s and can therefore be discarded in recognition. Dividing by the prior probability is equivalent to adjusting the bias parameter of the DNN.

2.2. Cross-entropy

The CE criterion is the most common frame-discriminative training criterion:

$$\mathcal{F}^{(\mathrm{CE})}(\theta) = -\sum_t \log p_\theta(s_t|x_t) \ . \quad (4)$$

Note that the CE criterion is well-defined because the output of the network is normalized. The normalization is performed on frame-level, thus all states "compete" against each other with equal weight. In recognition, the normalization constant is not required, see Equation (3).

2.3. Maximum mutual information

Sequence-discriminative MMI directly optimizes the posterior of the whole training utterance, thereby taking into account all knowledge sources of the speech recognition system:

$$\mathcal{F}^{(\mathrm{MMI})}(\theta) = -\log p_\theta(\mathbf{w}|\mathbf{x}) \ . \quad (5)$$

The posterior probability of an utterance is of the form

$$p_\theta(\mathbf{w}|\mathbf{x}) = \frac{p(\mathbf{w})q_\theta(\mathbf{x}|\mathbf{w})^\beta}{\sum_{\mathbf{v}\in\mathcal{L}} p(\mathbf{v})q_\theta(\mathbf{x}|\mathbf{v})^\beta} \quad (6)$$

with

$$q_\theta(\mathbf{x}|\mathbf{v}) = \sum_{s_1^T \in \mathcal{L}(\mathbf{v})} \exp\Big(\sum_{t=1}^T \lambda_{s_t}^T \phi_W(x_t) + \alpha_{s_t} + \log p(s_t|s_{t-1})\Big) \ . \quad (7)$$

It is a common practice in discriminative training (DT) to use acoustic model scaling, i.e., to set the LM scale to one and the acoustic model scale β to the inverse of the LM scale, which is used in recognition. Using a *weak* LM, typically a unigram, is another heuristic which is commonly used in discriminative training of GMMs [17, 18, 19]. Its aim is to ensure that the set of competing hypothesis represented by the lattice has enough variation.

Note that in the sequence-discriminative framework, there is no need to apply the softmax normalization or divide by the state prior. The softmax normalization cancels in Equation (7) and the bias parameter is already trained properly.

2.4. Minimum phone error

MPE [7] is commonly regarded as the criterion of choice for discriminative training of GMMs [18, 19]. It optimizes the *expected* error of the reference on the training data according to the model. The error is defined as a local approximation to the Levenshtein distance on phoneme level. Let $E_\mathcal{L}$ denote the local distance measure. Then the MPE objective function is defined as

$$\mathcal{F}^{(\mathrm{MPE})}(\theta) = \sum_{\mathbf{v}\in\mathcal{L}} p_\theta(\mathbf{v}|\mathbf{x}) E_\mathcal{L}(\mathbf{w}, \mathbf{v}) \ . \quad (8)$$

The locality of the string distance measure is required to enable the use of word lattices, which is crucial for the application to large vocabulary continuous speech recognition (LVCSR). The state-level minimum Bayes risk (sMBR) criterion, which has recently been used for sequence training of DNNs [9, 11], is obtained with a slightly different definition of the error measure [20, 21].

In contrast to MMI, the MPE objective function is bounded and is therefore more robust to outliers. Furthermore, it is generally considered to be advantageous over MMI because it is more closely related to the word error rate (WER) – the typical evaluation measure in ASR.

3. ENHANCEMENTS

Several authors have reported stability problems of sequence training [10, 11, 12]. The following modifications have been proposed to improve the performance of sequence training.

3.1. Cross-entropy smoothing

Su et al. [10] emphasize that lattice sparsity is an inherent cause of instability of lattice-based sequence training. Even when very dense lattices are employed, only a fraction of the classes are represented at every frame. Unfavorable scores of unrepresented classes do not affect the objective function at all. In other words, when the model changes too much from the one used for generating the lattices, there is a strong mismatch between objective function and recognition WER. This problem is especially severe with stochastic optimization algorithms because of their frequent model updates.

As a solution to this problem, Su et al. [10] proposed smoothing the sequence-discriminative objective function with the CE objective function. This yields for example the smoothed MMI criterion:

$$\mathcal{F}^{(\mathrm{sm-MMI})} = (1-\gamma)\mathcal{F}^{(\mathrm{MMI})} + \gamma\mathcal{F}^{(\mathrm{CE})} \ , \quad (9)$$

with an interpolation factor $0 < \gamma < 1$.

The lattice sparsity can be circumvented completely by computing the lattices on-demand for every utterance [12]. This approach however is only feasible within a complex software framework with a parallelized implementation of asynchronous SGD.

3.2. Frame-rejection heuristic

MMI training is sensitive to outliers, because the objective function is unbounded. Veselý et al. [11] proposed a *frame-rejection heuristic* to make MMI training more robust. According to the heuristic, all frames t where the probability of the reference state at time t given the whole observation sequence (known as the state occupancy in MMI) is smaller than a small threshold are discarded.

In principle, one can discard these frames in MPE training as well, but we have not found this to give any improvement and do not detail the results here.

4. OPTIMIZATION

The gradient of sequence-discriminative criteria is computed by backpropagation as in CE training, only the error signal at the output layer changes. The error signal is accumulated on a word lattice in the same way as in conventional discriminative training.

In general, any gradient-based numerical optimizer can be used for sequence training. One approach is to use stochastic optimization, usually SGD, which is the de-facto standard for CE training of DNNs. SGD scales well to large datasets and can be implemented efficiently on a GPU. This approach has been taken by [8, 10, 11, 12]. Neural networks with multiple linear bottlenecks can not be trained

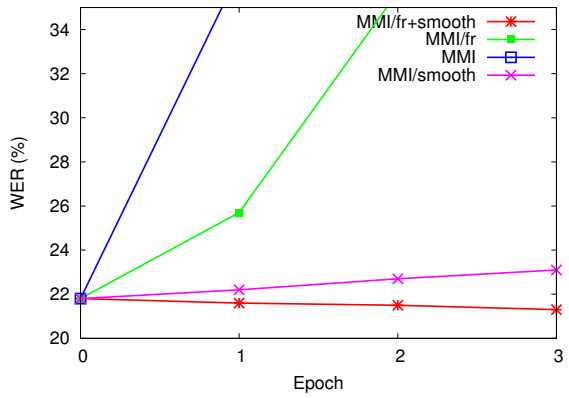

Fig. 1. Evolution of the WER with different variants of MMI. `fr` stands for frame rejection and `smooth` for CE-smoothing

well with SGD. In our previous work [15], we derived a stochastic algorithm called mean-normalized stochastic gradient descent (MN-SGD) and showed that it is capable of optimizing such *bottleneck networks* from scratch

For sequence training, the advantages of stochastic optimization are less compelling. First, because of their frequent model updates, the lattices deviate quickly from the search space of the current model. Second, with SGD on utterance-level, the data can be shuffled worse than on frame-level. Also, the updates are more heterogeneous due to the varying utterance length. In contrast to CE training, sequence training is initialized with a good model. In this case, batch algorithms benefit strongly from second-order information. Further, batch algorithms can be parallelized straight-forwardly and do not require fiddling with learning rates.

We use the batch algorithm Rprop [16] (more specifically, the iRprop$^+$ variant proposed in [22]) as an alternative to SGD. Rprop has separate learning rates for all parameters, which are computed from sign changes of the gradient. The use of separate learning rates per parameter corresponds to a diagonal second-order model of the objective function. So far, the only work where Rprop has been applied to DNN sequence training is by Kubo et al. [23], but there, a non-standard training procedure has been used and only the output layer of the DNN has been trained. In contrast to the HF algorithm used in [9], Rprop has only very few tuning parameters, is simple to implement, and has no computational overhead beyond the gradient computation.

5. EXPERIMENTAL RESULTS

We validated our proposed approach on the English Quaero corpus [24], a broadcast conversations recognition task characterized by highly spontaneous speech. We use a 50-hour subset of the corpus for training, and the evaluation corpora from 2010 and 2011 as development and test sets. The development and test sets consist of 3.7 and 3.3 hours of speech respectively. We implemented the sequence training in the RASR toolkit [25].

The general training setup of the CE baseline systems is the same as in [15]. The input to the models is a 493-dimensional vector, which is derived from Mel-frequency cepstral coefficients (MFCC) in a sliding window of size 17. The networks have a softmax output layer representing the 4501 context-dependent states of the GMM baseline. The recognition lexicon has 150k words. The LM is a smoothed four-gram, trained on roughly four billion words.

Table 1. WER of the shallow network with different training criteria and lattices. GM arcs stands for "garbage model arcs"

Criterion	Lattices		WER [%]	
	LM	GM arcs	Dev	Test
CE	-	-	21.8	**28.5**
MPE	unigram	yes	no impr.	no impr.
		no	21.0	27.4
	4-gram		20.3	**26.8**
MMI			21.0	27.5

We trained three different CE baseline modes, all with sigmoid units in the hidden layers. The first one is a *shallow NN* with just one 2048-dimensional hidden layer. The second is a *DNN* with six hidden layers of the same size. The third is a *bottleneck network* with the same topology as the DNN, but with a 256-dimensional linear bottleneck placed after every sigmoid layer. The number of parameters of the models are 10.2 million, 31.2 million, and 7.9 million for the shallow NN, the DNN, and the bottleneck network respectively.

The weights of the deep networks are initialized with discriminative pre-training [26]. CE training of the shallow NN and the DNN is performed with SGD. The bottleneck network is trained with MN-SGD. In [15], we observed that the WER continues decreasing after the validation frame error stagnates. Therefore, we adapt the learning rate based on the training error and use early stopping for regularization.

The error rates of the baseline models are shown in the first rows of Table 1, 2, and 3. Note that the bottleneck network outperforms the DNNalthough it has even less parameters than the shallow model. The CE results here are slightly better than in [15] because of a technical detail. Our phoneme set contains a garbage model which is used for pronunciations of short word fragments, which appear frequently in spontaneous speech, for example in contexts like "nat- natural" or "wou- should". Including the garbage model in the NN training unexpectedly improved the WER of the DNN and the bottleneck model by 0.2 and respectively 0.4 percent.

The lattices are generated with the CE models and kept fix for the whole training. We used the training lexicon for lattice generation. The same LM is used for lattice generation and sequence training. The force-aligned reference is merged into the lattice. The average number of lattice arcs per reference word is between 300 and 500.

The sequence trainings are initialized with the CE models with adapted bias parameters. We use fixed learning rates between 10^{-3} and 10^{-5} for the experiments with SGD and MN-SGD. Rprop is applied with the standard hyperparameters from [16]. The initial step size is set to a small value which ensures stable optimization. For all experiments, the best epoch is selected on the validation set.

5.1. Initial experiments with the shallow network

In our first experiments with sequence training, we observed a strong degradation of the models. Our first results are therefore obtained in the most controlled setup – we only train the output layer of a shallow network. We use SGD for optimization.

Figure 1 shows the evolution of the WER with different MMI variants. Without smoothing, the WER increases quickly. The same behavior is observed with MPE training (not shown in the figure). The frame rejection heuristic discards roughly five percent of the training data with a threshold of 10^{-6}. We only obtain improvements with

Table 2. WERs of the DNN with CE training and MPE training with Rprop and SGD

Algorithm	Layers trained	#Epochs	WER [%] Dev	WER [%] Test
CE baseline	all	26	18.5	**24.6**
SGD	all	11	17.6	23.5
	output layer	14	17.8	23.7
Rprop	all	29	17.5	**23.3**
	output layer	24	17.5	23.3

Table 3. WERs of the bottleneck-DNN with CE training and MPE training with Rprop and (MN-)SGD

Algorithm	Layers trained	#Epochs	WER [%] Dev	WER [%] Test
CE baseline	all	26	18.0	**23.7**
MN-SGD	all	10	17.5	23.2
SGD	output layer	11	17.5	23.2
Rprop	all	31	17.4	**23.0**
	output layer	18	17.6	23.1

MMI when both, the frame rejection heuristic and CE smoothing are active. All results shown in the following are obtained with CE smoothing with interpolation factor 0.1 and additionally the frame rejection in the case of MMI.

The lattice creation is a technical but crucial step for getting improvements with sequence training. With our standard lattice generation setup, i.e., with unigram LM and training lexicon, some phonemes are strongly overrepresented. In particular, words with garbage model pronunciation blow up the lattices. Therefore, sequence training focuses on discriminating the garbage model from the reference state, although the garbage model is not used in the recognition lexicon at all. We fixed this by adding a large penalty on the garbage model of the network used for lattice generation. The results in Table 1 show that this fix is necessary for getting improvements on this task. In addition, we observed that arcs with very short words like "I" or "a" and interjections and disfluencies like "um" or "huh" dominate the lattices. This is a consequence of using a unigram LM and can be avoided with a higher-order LM, see Table 1. Note that this observation is in contrast to results in discriminative training literature, e.g. [17, 18, 19], where a unigram LM is preferred. We conjecture that a higher-order LM is preferable on spontaneous speech tasks, where the word boundaries are not acoustically distinct.

Finally, we found MPE to perform clearly better than MMI – with the best lattice configuration MPE achieves a WER of 26.8 percent WER in comparison to 27.5 percent WER with MMI training. This is an improvement of 2.3 percent absolute over the CE baseline.

5.2. DNN results

According to the findings in the experiments with the shallow network, we used four-gram lattices and CE-smoothed MPE for the DNN sequence trainings. In this set of experiments, we compare the performance of SGD and Rprop. This question might also depend on whether the complete network or only the output layer is trained. The results are shown in Table 2.

One could suspect that it is important to include the hidden layers in sequence training, because they amount to more than seventy percent of the parameters. However, we only observed a small improvement of 0.2 percent WER in the case of SGD by training the complete network.

The best results are obtained with Rprop. As expected, Rprop requires more epochs than SGD. On the other hand, we ran SGD with two different learning rates in parallel, which is not necessary for Rprop. In addition, the gradient computation required for Rprop can be distributed.

The aim of CE smoothing is to avoid problems occurring when the lattices do not fit well to the model. This raises the question whether CE smoothing is only beneficial for stochastic gradient training – the case which Su et al. [10] studied – or for batch training as well. Therefore, we ran an unsmoothed MPE training with Rprop. We observed that the WER does not diverge directly as with SGD, but already after two epochs. The WER at this point is 18.1 percent on the development data and 24.3 percent on test, which is far from the improvement obtained with smoothed MPE.

5.3. Bottleneck network results

In our final experiments, we investigate whether the improvements obtained with bottleneck networks persist after sequence training. The results are shown in Table 3. The improvements from sequence training are smaller than with the DNN. Still, a small but consistent improvement of the bottleneck networks remains. The best result is again achieved with Rprop. The improvement over the full DNN is 0.3 percent WER on the test data.

6. CONCLUSION

Sequence training of DNN-HMMs has been shown to give substantial improvements on state-of-the-art speech recognition systems. However, sequence training is a technically complex technique and there is no common agreement on the best training configuration.

We presented experiments with our implementation of sequence training, which is part of the freely available RASR toolkit [25]. Our experiments on a broadcast conversations recognition task provide more empirical evidence on the best choice of the training criterion, training enhancements, and the optimization algorithm.

In particular, we found the CE smoothing proposed by Su et al. [10] to be essential for getting improvements. Su et al. only evaluated this technique with SGD optimization and only applied it to MMI. So far, their idea has not been taken up by other authors. It is not clear why the lattice sparsity has not caused similar problems in the works by Kingsbury et al. [9] and Veselý et al. [11]. One reason might be that their lattices are are generated with a WFST decoder, while we use a dynamic decoder. In our opinion, this is an important issue, which requires further analysis.

The question which optimization algorithm should be used does not have a simple answer. McDermott et al. [13] applied sequence training on very large datasets and found that asynchronous SGD converges already before the data is processed even once. The behavior with larger acoustic models however might be different. Rprop on the other hand can be parallelized straight-forwardly, and does not require learning rate tuning. Furthermore, we observed improvements with Rprop over SGD in WER.

Finally, we investigated neural networks with a linear bottleneck structure, based on our previous work [15]. We found that a small gain due to the bottleneck structure persists after sequence training. The bottleneck network in this experiment has only a quarter of the parameters of the original DNN without linear bottlenecks.

7. REFERENCES

[1] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 20, no. 1, pp. 30–42, 2012.

[2] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks." in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc. (Interspeech)*, Florence, Italy, Aug. 2011, pp. 437–440.

[3] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novák, and A. rahman Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Waikoloa, HI, USA, Dec. 2011, pp. 30–35.

[4] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer, 1994, vol. 247.

[5] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. of the Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Tokyo, Japan, May 1986, pp. 49–52.

[6] V. Valtchev, J. Odell, P. C. Woodland, and S. J. Young, "MMIE training of large vocabulary recognition systems," *Speech Commun.*, vol. 22, no. 4, pp. 303–314, 1997.

[7] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. of the Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Orlando, FL, USA, May 2002, pp. 105 – 108.

[8] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proc. of the Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 3761–3764.

[9] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc. (Interspeech)*, Portland, OR, USA, Sep. 2012.

[10] H. Su, G. Li, D. Yu, and F. Seide, "Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription," in *Proc. of the Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Vancouver, Canada, May 2013, pp. 6664–6668.

[11] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc. (Interspeech)*, Lyon, France, Aug. 2013, pp. 2345–2349.

[12] G. Heigold, E. McDermott, V. Vanhoucke, A. Senior, and M. Bacchiani, "Asynchronous stochastic optimization for sequence training of deep neural networks," in *Proc. of the Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 5624–5628.

[13] E. McDermott, G. Heigold, P. Moreno, A. Senior, and M. Bacchiani, "Asynchronous stochastic optimization for sequence training of deep neural networks: Towards big data," in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc. (Interspeech)*, Singapore, Sep. 2014, pp. 1224–1228.

[14] J. Martens, "Deep learning via hessian-free optimization," in *Proc. of the Int. Conf. on Mach. Learning (ICML)*, Haifa, Israel, Jun. 2010, pp. 735–742.

[15] S. Wiesler, A. Richard, R. Schlüter, and H. Ney, "Mean-normalized stochastic gradient for large-scale deep learning," in *Proc. of the Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 180–184.

[16] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Proc. of Int. Conf. on Neural Networks (ICNN)*, San Francisco, CA, USA, Mar. 1993, pp. 586–591.

[17] R. Schlüter, B. Müller, F. Wessel, and H. Ney, "Interdependence of language models and discriminative training," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Keystone, CO, USA, Dec. 1999, pp. 119–122.

[18] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge University, Cambridge, UK, 2004.

[19] G. Heigold, "A log-linear discriminative modeling framework for speech recognition," Ph.D. dissertation, RWTH Aachen University, Aachen, Germany, Jun. 2010.

[20] J. Zheng and A. Stolcke, "Improved discriminative training using phone lattices," in *Proc. of the European Conf. on Speech Commun. and Technology (Eurospeech)*, Lisbon, Portugal, Sep. 2005, pp. 2125–2128.

[21] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition," in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc. (Interspeech)*, Pittsburgh, PA, USA, Sep. 2006, pp. 2406–2409.

[22] C. Igel and M. Hüsken, "Empirical evaluation of the improved Rprop learning algorithms." *Neurocomputing*, vol. 50, pp. 105–123, 2003.

[23] Y. Kubo, T. Hori, and A. Nakamura, "Integrating deep neural networks into structural classification approach based on weighted finite-state transducers," in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc. (Interspeech)*, Portland, OR, USA, Sep. 2012.

[24] Quaero Programme, http://www.quaero.org.

[25] S. Wiesler, A. Richard, P. Golik, R. Schlüter, and H. Ney, "RASR/NN: The RWTH neural network toolkit for speech recognition," in *Proc. of the Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 3313–3317.

[26] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Waikoloa, HI, USA, Dec. 2011, pp. 24–29.