Sequence Discriminative Distributed Training of Long Short-Term Memory Recurrent Neural Networks

Haşim Sak, Oriol Vinyals, Georg Heigold
Andrew Senior, Erik McDermott, Rajat Monga, Mark Mao

Google, USA

{hasim,vinyals,heigold,andrewsenior,erikmcd,rajatmonga,markmao}@google.com

Abstract

We recently showed that Long Short-Term Memory (LSTM) recurrent neural networks (RNNs) outperform state-of-the-art deep neural networks (DNNs) for large scale acoustic modeling where the models were trained with the cross-entropy (CE) criterion. It has also been shown that sequence discriminative training of DNNs initially trained with the CE criterion gives significant improvements. In this paper, we investigate sequence discriminative training of LSTM RNNs in a large scale acoustic modeling task. We train the models in a distributed manner using asynchronous stochastic gradient descent optimization technique. We compare two sequence discriminative criteria – maximum mutual information and state-level minimum Bayes risk, and we investigate a number of variations of the basic training strategy to better understand issues raised by both the sequential model, and the objective function. We obtain significant gains over the CE trained LSTM RNN model using sequence discriminative training techniques.

Index Terms: recurrent neural network, long short-term memory, sequence discriminative training, acoustic modeling.

1. Introduction

Deep neural networks (DNNs) have been very successful for acoustic modeling in large vocabulary speech recognition [1, 2, 3, 4, 5, 6]. More recently, Long Short-Term Memory (LSTM) recurrent neural networks (RNNs) have been shown to beat the state of the art DNN systems [7, 8]. LSTM networks [9, 10, 11] are a type of recurrent neural network, which contain special units called *memory blocks* in the recurrent hidden layer, and which are believed to be easier to train than standard RNNs. The memory blocks contain memory cells with self-connections storing the temporal state of the network. In addition, they have multiplicative units called gates to control the flow of information into the memory cell and out of the cell to the rest of the network.

DNNs and LSTM RNNs for acoustic modeling have commonly used the hybrid approach [12], where the neural networks estimate the hidden Markov model (HMM) state posteriors to be used in HMM based decoders. The models are generally first trained using HMM state aligned acoustic frames, where the initial alignments can be obtained with a Gaussian mixture model (GMM) using the supervised transcripts and can be further refined by realigning with a fully trained neural network. The cross-entropy training criterion is commonly used with a softmax output and the outputs converge to class posteriors. As a frame-level criterion discriminating HMM states, cross-entropy is well suited to the task of labeling individual acoustic frames. However, it is not a good match for the speech recognition objective of word error rate (WER) minimization – which is hard to directly optimize, thus there is a need for an utterance-level criterion discriminating word sequences. A number of alternative sequence discriminative criteria have been proposed to better match the speech recognition objective, including maximum mutual information (MMI) [13], minimum phone error (MPE) [14], and state-level minimum Bayes risk (sMBR) [15]. These techniques have been shown to improve performance of DNN systems bootstrapped with cross-entropy training [15, 16, 17, 18].

Stochastic gradient descent (SGD) [19] is an optimization technique commonly used for training of DNNs and RNNs [2, 1, 3, 5, 6, 7, 8]. Since SGD is sequential, it is inherently slow to train large models on large datasets. However, training can be parallelized through asynchronous stochastic gradient descent (ASGD), which has proven successful in large scale training of neural networks [20, 21, 17, 22]. ASGD has also been used for sequence discriminative training of DNN models [17, 23]. Sequence discriminative training with ASGD has further implications for parameter update asynchrony – the model parameters are updated asynchronously with the gradient computations – and the limited randomization due to the nature of sequence training – the gradients are computed per utterance. We have addressed these issues for sequence discriminative training of DNNs in the DistBelief distributed training framework [23].

In this paper, we explore sequence discriminative training of LSTM RNNs in a large scale acoustic modeling task. LSTM networks naturally improve CE over DNN due to their smoothing capabilities at the frame level (that do not necessarily translate to improved WER). We argue that, by bridging the mismatch of objective functions, the LSTM will focus on long term acoustic dependencies rather than exploiting the language model, thanks to incorporating the smoothing effect of the underlying HMM into the objective function. In order to investigate all the above, we use ASGD optimization for distributed training of the models, which has additional implications for RNNs since we are training over sequences and the truncated backpropagation through time (BPTT) learning algorithm is used to update the model parameters [24]. We compare MMI and sMBR training criteria. We also investigate different training strategies for speed of convergence and performance including starting from not a fully converged cross-entropy trained model, realigning the data with an LSTM model, and using language models of varying power for sequence discriminative training. As far as we know this is the first application of classical sequence discriminative training techniques to RNNs and we show substantial improvement over the CE trained LSTM RNN model.

2. Acoustic Modeling with DNNs and RNNs

Let $X = x_1, \ldots, x_T$ denote a sequence of T feature vectors $x_t \in \mathbb{R}$ and W a word sequence. According to the HMM assumption, the acoustic data likelihood is decomposed as follows (using the Viterbi approximation):

$$p(X|W) = \prod_{t=1}^{T} p(x_t|s_t)p(s_t|s_{t-1}),$$

where $s_1, \ldots, s_T$ is the forced alignment for word sequence W. In the hybrid modeling approach, the emission probability is represented as $p(x_t|s_t) = p(s_t|x_t)p(x_t)/p(s_t)$ (Bayes rule). The state posterior can be modeled by, for example, a DNN [2, 1, 3, 5, 6] or an LSTM RNN over asymmetrically windowed features (this work), $p(s_t|x_1, \ldots, x_t)$ (Section 3.1). The state prior $p(s_t)$ is the relative state frequency. The data likelihood $p(x_t)$ does not depend on states and thus can be ignored for decoding/lattice generation and forced alignment [12].

2.1. Frame-Level Criterion

We assume that alignments are available (generated with an existing model such as Gaussian mixture models or LSTM RNNs) and fixed. Then, the network parameters comprising the weights and biases can be estimated from scratch by maximizing the cross entropy (CE) on all utterances u and frames t

$$F_{CE}(\theta) = \frac{1}{T} \sum_u \sum_{t=1}^{T_u} \sum_s l_{ut}^{(CE)}(s) \log p_\theta(s|x_{ut}). \quad (1)$$

Here, $T = \sum_u T_u$ is the total number of frames. The targets are $l_{ut}^{(CE)}(s) = \delta(s, s_{ut})$ where δ denotes the Kronecker delta. This criterion is simple, can be used to bootstrap a network, and achieves good performance, however it may be suboptimal as it does not consider the lexical and language model constraints we use in the speech engine for decoding, and it does not directly optimize the word error, which is the primary evaluation metric in speech recognition. Sequence-level criteria have been proposed to address these issues.

2.2. Sequence-Level Criteria

A variety of sequence-level criteria have been discussed in the literature, including maximum mutual information (MMI) [13], minimum phone error (MPE) [14], and state-level minimum Bayes risk (sMBR) [15]. In this paper, we shall focus on MMI and sMBR.

Maximum mutual information is defined as:

$$F_{MMI}(\theta) = \frac{1}{T} \sum_u \log \frac{p_\theta(X_u|W_u)^\kappa p(W_u)}{\sum_W p_\theta(X_u|W)^\kappa p(W)}. \quad (2)$$

The logarithm diverges if the argument goes to zero, i.e., if the correct word sequence has zero probability in decoding. To avoid numerical issues with such utterances, we use the frame rejection heuristic described in [18], i.e., discard frames with state occupancy close to zero, $\gamma_{ut}^{(den)}(s) < \epsilon$. As usual for sequence discriminative training, a weak language model $p(W_u)$ is used and κ, the reciprocal of the language model weight, is attached to the acoustic model. State-Level Minimum Bayes Risk (sMBR) is the expected frame state accuracy:

$$F_{sMBR}(\theta) = \frac{1}{T} \sum_u \sum_W \frac{p_\theta(X_u|W)^\kappa p(W)}{\sum_{W'} p_\theta(X_u|W')^\kappa p(W')} \delta(s, s_{ut}). \quad (3)$$

No regularization (for example, ℓ_2-regularization around the initial network) or smoothing such as the H-criterion [17] is used in this paper.

The gradient of the sequence-level criteria MMI and sMBR can be written as

$$\nabla F_{MMI/sMBR}(\theta) = \frac{1}{T} \sum_u \sum_{t=1}^{T_u} \sum_s \kappa l_{\theta,ut}^{(MMI/sMBR)}(s) \nabla \log p_\theta(s|x_{ut}).$$

For MMI, $l_{\theta,ut}^{(MMI)}(s)$ stands for the difference of the numerator and denominator state occupancies for utterance u and frame t, $\gamma_{\theta,ut}^{(num)}(s)$ and $\gamma_{\theta,ut}^{(den)}(s)$ [15, 17, 18]. For sMBR, $l_{\theta,ut}^{(sMBR)}(s)$ denotes the centered state error, $\gamma_{\theta,ut}^{(den)}(s)(\delta(s, s_{ut}) - \sum_s \gamma_{\theta,ut}^{(den)}(s)\delta(s, s_{ut}))$ [14, 18]. Using chain rule terminology, we shall refer to $\kappa l_{\theta,ut}^{(MMI/sMBR)}(s)$ and $\nabla \log p_\theta(s|x_{ut})$ as the outer and inner gradients, respectively. The outer gradients can be efficiently computed using the shortest path algorithm and a suitable expectation semiring [25].

3. Neural Networks & Training

3.1. LSTM RNN

The LSTM network used in this paper is a two layer deep LSTM RNN, where each LSTM layer has 800 memory cells and a dimensionality reducing recurrent projection layer of 512 linear units [8, 26]. The LSTM network has 13 million parameters and uses hyperbolic tangent activation (tanh) for the cell input units and cell output units, and logistic sigmoid for the input, output and forget gate units, and softmax output activation function. The LSTM networks are first trained with cross-entropy and ASGD using distributed training with 500 tasks scheduled on different machines, each working through a partition of the randomly shuffled training data. Each task processes four utterances at a time, using the BPTT algorithm to forward propagate and then backpropagate for T_{bptt} consecutive frames. Each task thus computes a parameter gradient update for a minibatch of $4 \times T_{bptt}$ frames. (We use $T_{bptt} = 20$.) The input to the LSTM at each time step is a single 25ms frame of 40-dimensional log-filterbank energy features. Since information from future frames helps making better decisions for the current frame (similar to having a right context window in DNNs), we delay the output HMM state label by 5 frames.

3.2. Sequence Discriminative Distributed Training with Asynchronous Stochastic Gradient Descent (ASGD)

We use ASGD in a distributed framework [20, 21] for sequence discriminative training, as proposed in [23] and depicted in Fig. 1.

The basic architecture augments the standard distributed training described above for cross-entropy training with a speech module computing outer gradients (Section 2.2) for complete utterances at a time. The basic workflow is as follows: Each worker works through its data partition an utterance at a time. For each utterance, the speech module fetches the model parameters θ from the parameter server, which maintains the current state of all model parameters. It computes the state posteriors for every frame and decodes the speech to compute lattices and occupancy statistics and hence the outer gradients. See Figure 2 for sequence discriminative training pipeline. The

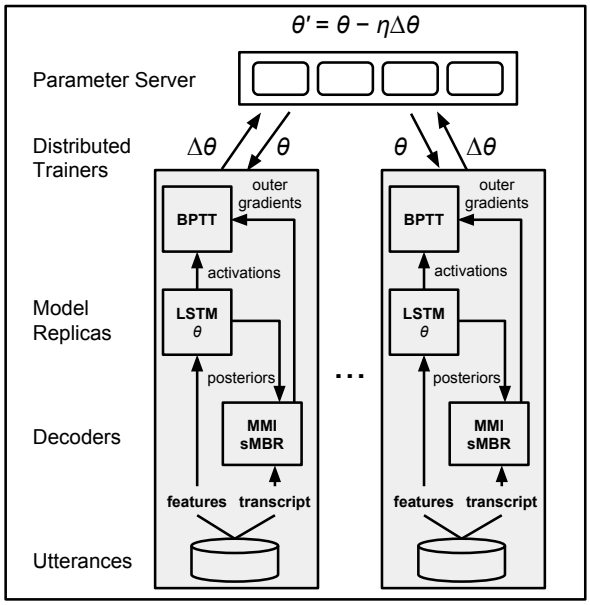

Figure 1: Asynchronous SGD: Model replicas asynchronously fetch parameters θ and push gradients $\Delta\theta$ to the parameter server.

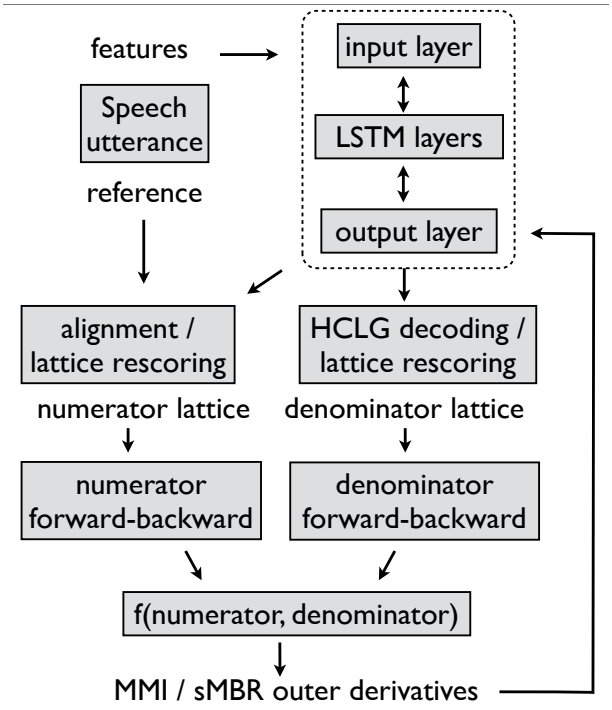

Figure 2: Sequence discriminative training pipeline.

trainer task then proceeds as with cross-entropy training, repeatedly requesting the latest parameters from the parameter server before using BPTT to compute parameter gradients for the next set of T_{bptt} frames and sending those updates to the parameter server. See Fig. 3 for pseudocode. An inherent feature of this architecture is the asynchrony, which means that the gradients are computed on stale parameters and the outer and inner gradients are computed on slightly different parameters but is essential to scale the algorithm.

4. Experiments

4.1. ASR System & Evaluation

All the networks are first trained with cross-entropy criterion on a 3 million utterance (about 1900 hours) dataset consisting of anonymized and hand-transcribed Google voice search and dictation traffic. Then, the networks are trained with MMI or sMBR criterion on the same training data set. The dataset is represented with 25ms frames of 40-dimensional log-filterbank energy features computed every 10ms. The 40-dimensional features are input to the network with no stacking of frames. The utterances are aligned with a 85 million parameter FFNN with 14247 CD states. The weights in all the networks are randomly initialized prior to training. We try to set the learning rate specific to a network architecture and its configuration to the largest value that results in a stable convergence. The learning rates are exponentially decayed during training.

During cross-entropy training, we evaluate the loss and frame accuracy (i.e. HMM state labeling accuracy of acoustic frames) on a held out set of 200,000 frames. During MMI/sMBR training, we evaluate the loss and WER on a held out set. The trained models are evaluated in a large vocabulary speech recognition system on a test set of 22,500 hand-transcribed utterances and the word error rates (WERs) are reported. The vocabulary size of the language model used in the

$\mathcal{U} \leftarrow$ the data set of utterances with transcripts
$\mathcal{U} \leftarrow$ *randomize*($\mathcal{U}$)
θ is the model parameters
for all $u \in \mathcal{U}$ **do**
  $\theta \leftarrow$ read from the parameter server
  calculate $\kappa l_{\theta,ut}^{(MMI/sMBR)}(s)$ for u (see Figure 2)
  for all $s \in$ subsequences(u, bptt_steps) **do**
    $\theta \leftarrow$ read from the parameter server
    forward_pass(s, θ, bptt_steps)
    $\vec{\Delta}\theta \leftarrow$ backward_pass(s, θ, bptt_steps)
    $\Delta\theta \leftarrow$ sum_gradients($\vec{\Delta}\theta$, bptt_steps)
    send $\Delta\theta$ to the parameter server
  end for
end for

Figure 3: Pseudocode for a single task of the sequence discriminative distributed training of RNNs.

decoding is 2.2 million. The language model used in the first pass of decoding is a 5-gram language model heavily pruned to 23 million n-grams. In the second pass, the word lattices output from the first pass are rescored with a 5-gram language model having 1 billion n-grams.

4.2. Results

We experimented with various alternative techniques and strategies for sequence discriminative training and obtained significant improvements in speech recognition accuracy over the LSTM RNN models trained with CE.

The training data used in these experiments was aligned using a large (85M parameter) DNN, itself trained on alignments from a large DNN. Because of the lower WERs from the LSTM,

and because of the different temporal context and structure of the model, we conjectured that the optimal alignment from our LSTM RNN acoustic model might be different from the optimal alignment from the large DNN. Consequently we experimented to see the effect of realignment on the LSTM RNN and in particular on LSTM sequence discriminative training. For this purpose, we trained two LSTM RNN models with CE criterion. The first LSTM RNN model was trained on DNN alignments. Using this LSTM RNN model, we realigned the training data. The resulting alignments were used to train from scratch a new LSTM RNN with the same architecture. We observed improvements in the frame accuracy and in the convergence speed as can be expected, although the new model did not improve the WER. Our motivation was that by realigning the data with an LSTM RNN model, sequence discriminative training would benefit since the numerator computation involves realignment, and the model should be consistent with this alignment, which it is not when trained from DNN alignments. Table 1 compares these two models bootstrapped with CE training on DNN versus LSTM RNN alignments for sMBR sequence discriminative training.

Table 1: *sMBR training of LSTM RNN bootstrapped with CE training on DNN versus LSTM RNN alignments.*

Alignment	CE	sMBR
DNN	10.7	10.1
LSTM RNN	10.7	10.0

We compared the experimental results for language models of increasing power used in sMBR training by varying the n-gram order of the LMs. All the language models used in training are trained over the 3-million transcripts of the training data. Table 2 shows that we obtained the lowest WERs with the bigram LM. We used the bigram LM for all the other experiments in this paper.

Table 2: *sMBR training with LMs of various n-gram orders.*

CE	1-gram	2-gram	3-gram
10.7	10.9	10.0	10.1

There is an obvious match between sequential models that consider all the acoustic observations to predict the HMM state, and sequence discriminative training, which should discourage learning language model specific signals from the acoustic training data (as the HMM has all the language model information readily available). We have observed that adding more acoustic context to the standard DNN acoustic models improves the frame accuracy, but the final WER can be worse. We argue that LSTMs, which have all the context, may suffer from this by learning how to smooth the predictions through a language model learned on the acoustics, which explains in part the much higher (around 10% absolute) frame accuracies of LSTMs versus DNNs.

One of the contributions of this paper is to investigate how sequence discriminative training, which should discourage the model from smoothing its outputs (as the HMM will generally act as the smoothing function), interacts as the training progresses. To this extent, we tried to switch early from CE to MMI/sMBR. In Table 3, we can see how, as CE training progresses, switching to sequence training continues to improve WER (and we generally observe that this improvement can be achieved much faster than CE), but the best WERs can only be achieved after CE training has converged. We have observed this independently with DNN sequence discriminative training, and this may ultimately justify the need of interpolating between CE and sequence discriminative training, as proposed by Su et al. [17].

Table 3: *WERs achieved by MMI/sMBR training for around 3 days when we switch from CE training at different times before convergence. * indicates the best WER achieved after 2 weeks of sMBR training.*

CE WER at switch	MMI	sMBR
15.9	13.8	-
14.9	12.0	-
12.0	10.8	10.7
11.2	10.8	10.3
10.7	10.5	10.0 (9.8*)

Table 3 shows the best performance of 9.8% WER was achieved by 2 weeks of training with sMBR from a 10.7% CE baseline model. In comparison, the best sequence trained DNN with sMBR, with 85 million parameters, on this task achieves 10.4% WER from an 11.3% CE model.

5. Discussion & Conclusions

We investigated sequence discriminative training of LSTM RNNs for large scale acoustic modeling in a distributed framework using ASGD. We experimented with MMI and sMBR training criteria. Given the large number of hyperparameters due to sequence modeling with RNNs and ASGD training, it is hard to conclude sMBR criterion is inherently better than MMI for sequence discriminative training of RNNs. However, in our experimentation, we found that distributed training with sMBR gives better results and is easier to train than MMI. Even though MMI and sMBR costs on the held-out set both continue to improve, after a few days of training the MMI WER starts to get worse. We obtained slightly better results with 2-gram language models in the sMBR training over 3-gram LMs, while a unigram LM did not perform well. Realigning the training data with an LSTM RNN model led to small improvements over the DNN alignments. We investigated starting sequence discriminative training before the convergence of CE training, but we found that it is best to switch only after convergence of CE training. In the end, sMBR training significantly improved a CE trained LSTM RNN model from 10.7% to 9.8% WER, an 8.4% relative improvement.

6. References

[1] G. Dahl, D. Yu, and L. Deng, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

[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, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 20, no. 1, pp. 30–42, Jan. 2012. [Online]. Available: http://dx.doi.org/10.1109/TASL.2011.2134090

[4] A. Rahman Mohamed, G. E. Dahl, and G. E. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

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

[6] 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.

[7] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 273–278.

[8] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition," *ArXiv e-prints*, Feb. 2014.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[10] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[11] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 3, pp. 115–143, Mar. 2003.

[12] H. Bourlard and N. Morgan, *Connectionist speech recognition*. Kluwer Academic Publishers, 1994.

[13] Y. Normandin, "Hidden Markov models, maximum mutual information, and the speech recognition problem," Ph.D. dissertation, McGill University, Montreal, Canada, 1991.

[14] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge, England, 2004.

[15] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 3761–3764.

[16] 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 *INTERSPEECH*, 2012.

[17] 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 *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 6664–6668.

[18] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *INTERSPEECH*, 2013.

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

[20] 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.

[21] 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 (NIPS)*, 2012.

[22] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, Vancouver, Canada, Apr. 2013.

[23] G. Heigold, E. McDermott, V. Vanhoucke, A. Senior, and M. Bacchiani, "Asynchronous stochastic optimization for sequence training of deep neural networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.

[24] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for online training of recurrent network trajectories," *Neural Computation*, vol. 2, pp. 490–501, 1990.

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

[26] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," in *INTERSPEECH 2014*, 2014.