SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

VOLUME 3, ISSUE 2
ISSN 2054-7390

Unified Acoustic Modeling using Deep Conditional Random Fields

[1]Yasser Hifny

[1]*Faculty of computers and information systems, University of Helwan, Egypt;*

yhifny@fci.helwan.edu.eg

Abstract

Acoustic models based on Deep Neural Networks (DNNs) lead to significant improvement in the recognition accuracy. In these methods, Hidden Markov Models (HMMs) state scores are computed using flexible discriminant DNNs. On the other hand, Conditional Random Fields (CRFs) are undirected graphical models that maintain the Markov properties of HMMs formulated using the maximum entropy (MaxEnt) principle. CRFs have limited ability to model spectral phenomena since they have single quadratic activation function per state. It is possible and natural to use DNNs to compute the state scores in CRFs. These acoustic models are known as Deep Conditional Random Fields (DCRFs). In this work, a variant of DCRFs is presented and connections with hybrid DNN/HMM systems are established. Under certain assumptions, both DCRFs and hybrid DNN/HMM systems can lead to exact same results for a phone recognition task. In addition, linear activation functions are used in the DCRFs output layer. Consequently, DCRFs and traditional DNN/HMM systems have the same decoding speed.

Keywords: Hidden Markov models; deep conditional random fields; deep neural networks; discriminative training.

1 Introduction

Acoustic modeling based on Hidden Markov Models (HMMs) [1, 2, 3, 4] is employed by state-of-the-art stochastic speech recognition systems. Generative HMMs are well understood models and may be trained efficiently using the Expectation-Maximization (EM) algorithm [5].

An example of an HMM with left-to-right transition topology, which is used to model a phone in an acoustic model, is shown in Fig. 1. This model has one entry state, three emitting states, and one exit state. The left-to-right topology imposes prior information, where speech production is sequential in time.

For every observation at time t, a jump from the current state i to some new state j is allowed with a transition probability:

$$a_{ij} = P(\mathbf{s}_{t+1} = j|\mathbf{s}_t = i), \qquad (1)$$

where $\sum_j^N a_{ij} = 1$, N is the number of states in the HMM model. An acoustic feature vector $\mathbf{o}_t$ may be generated, with an output probability density function

DOI: 10.14738/tmlai.32.1124
Publication Date: 1st May, 2015
URL: http://dx.doi.org/10.14738/tmlai.32.1124

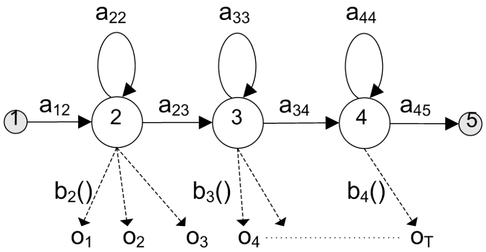

Figure 1: A typical Hidden Markov Model for a phone (a stochastic finite state machine view).

$b_j(\mathbf{o}_t)$, which is associated with state j. A mixture of Gaussian distributions is typically used to model the output distribution for each state,

$$b_j(\mathbf{o}_t) = \sum_{m=1}^{M} c_{jm}\mathcal{N}(\mathbf{o}_t; \mu_{jm}, \Sigma_{jm}), \tag{2}$$

where M is the number of mixture components, c_{jm} is the component weight and $\sum_m^M c_{jm} = 1$. μ_{jm} and Σ_{jm} are the component specific mean vector and covariance matrix respectively. If the acoustic features are statistically independent, then diagonal covariance matrices are used to compute the likelihood of a Gaussian model,

$$\mathcal{N}(\mathbf{o}_t; \mu_{jm}, \Sigma_{jm}) = \prod_{d=1}^{D} \frac{1}{\sqrt{(2\pi)\sigma_{jmd}}} \exp\Big(-\frac{(\mathbf{o}_{td} - \mu_{jmd})^2}{2\sigma_{jmd}^2}\Big), \tag{3}$$

where σ_{jmd} is the variance element of the Gaussian component m for dimension d.

In hybrid ANN/HMM speech recognition systems [6], [7], artificial neural networks (ANN) models are used as flexible discriminant classifiers to estimate a scaled likelihood. In particular, the emission probability score is given by

$$b_j(\mathbf{o}_t) \approx \frac{P_\Lambda(\mathbf{s}_j|\mathbf{o}_t)}{P(\mathbf{s}_j)}, \tag{4}$$

where $b_j(\mathbf{o}_t)$ is the score of state j in the traditional HMM framework, $P_\Lambda(\mathbf{s}_j|\mathbf{o}_t)$ is the posterior probability of a phonetic state estimated by a connectionist estimator [8],[9] and the prior $P(\mathbf{s}_j)$ is estimated from the labeled data. In addition to discriminative training, if the posterior probability $P_\Lambda(\mathbf{s}_j|\mathbf{o}_t)$ is sensitive to acoustic context, $b_j(\mathbf{o}_t)$ score may help to overcome conditional independence assumption and improve the overall recognition performance without changing the basic HMM framework. A graphical representation of the DNN/HMM acoustic model is shown in Fig. 2.

DNNs with several hidden layers that are trained using new methods have been shown to outperform Gaussian mixture models in several tasks [10], [11],

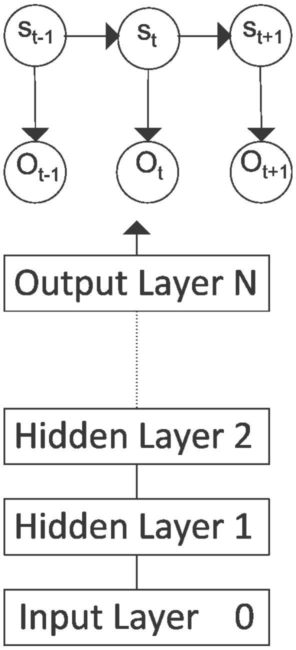

Figure 2: HMM model for phone representation, where the state scores are computed from a DNN.

[12], [13]. DNNs are trained in a generative way to learn the structure in the input data. This "pre-training" step provides a good initialization point to the traditional discriminative training using the backpropagation (BP) algorithm. DNN modeling is an active area of research and there is a lot of effort to improve the training speed of these models [14],[15], [16].

Over the last few years, there is an increased interest to develop acoustic models derived from MaxEnt [17, 18] and Conditional Random Fields [19]. Before CRFs became popular, there were several attempts to develop models similar to HMMs. In particular, the estimation of global posteriors using the forward-backward algorithm was derived in [20], [21]. Recent efforts in the field of MaxEnt/CRF modeling were reviewed and discussed in [22, 23]. Hidden Conditional Random Fields (HCRFs) were introduced to score the states based on a mixture of quadratic activation functions [24]. In [25], a multi-layer CRF model (deep-structured CRF) in which each higher layer's input observation sequence consists of the previous layer's observation sequence was presented. Deep extensions to HCRFs were developed in [26],[27]. A non-linear graphical model for structured prediction was introdcued in [28]. In [29], deep hidden conditional neural fields (Deep-HCNF) which utilized an observation function with deep structure were presented. A segmental version of CRFs was developed in [30].

In [31, 32], a new acoustic modeling paradigm based on Augmented Conditional Random Fields (ACRFs) is investigated and developed. ACRFs paradigm addresses some limitations of HMMs while maintaining many of the aspects which have made them successful. In particular, the acoustic modeling problem is reformulated in a data driven, sparse, augmented space to increase discrimination. Acoustic context modeling is explicitly integrated to handle the sequential phenomena of the speech signal. In the context of ANN field, ACRFs can represent CRFs with one hidden layer constructed from scoring a large number of Gaussians. Rank-based scoring used in maximum entropy direct modeling approaches [33, 34] may be interpreted as a mean to construct an augmented space.

Score-space kernels [35, 36], which are a generalization of the Fisher kernel [37], are used to extract new sufficient statistics, which may relax the conditional independence assumptions in a systematic fashion. These sufficient statistics are used to train MaxEnt models (C-Aug) for post-processing in HMM based speech recognition [38].

Training CRFs on the top of a hidden layer constructed from scoring a large number of sigmoid functions was introduced in [39]. One way to improve this approach is to compute the state scores based on a DNN that has several hidden layers. Hence, this improvement will lead to a deep version of CRFs (DCRFs) [40]. In this work, a mathematical formulation of DCRFs is reviewed and connections with hybrid DNN/HMM systems are established. We will unify the training procedure between DCRFs and hybrid DNN/HMM in order to explore the gains related to different DNN structures used in the two systems. Under this assumption, the paper will show that the two systems can lead to same exact results for a phone recognition task. Consequently, DCRFs may be a natural choice for sequential modeling for speech recognition.

This paper is organized as follows: the basic limitations to use CRF as an acoustic model is addressed in Section 2. A mathematical formulation of DCRFs is described in Section 3. The discriminative training problem of DCRFs is addressed in Section 4. In Section 5, generative training which is used to initilize DNNs is presented. DCRFs and DNN/HMM systems compute the state scores using similar deep architectures. Hence, it is possible to unify and establish connections between DCRFs and DNN/HMM systems. This idea is addressed in Section 6. Section 7 gives experimental results on a phone recognition task. Several issues about the implementation of DCRFs are discussed in Section 8. Finally, a summary of the presented work is given in the conclusions.

2 Conditional Random Fields Limitations

Linear chain Conditional Random Fields are undirected graphical models that maintain the Markov properties of HMMs, formulated using the maximum entropy (MaxEnt) principle [41]. The maximum entropy formalism for sequential modeling results in a probability distribution, which is the log linear or exponential model:

$$P_\Lambda(\mathbf{S}|\mathbf{O}) = \frac{1}{Z_\Lambda(\mathbf{O})}\prod_{t=1}^{T} \exp\Big(\sum_j \lambda^j_{\mathbf{s}_t\mathbf{s}_{t-1}} a_j(\mathbf{s}_t,\mathbf{s}_{t-1}) + \sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{O},\mathbf{s}_t)\Big), \quad (5)$$

where

- $P_\Lambda(\mathbf{S}|\mathbf{O})$ obeys the *Markovian* property $P_\Lambda(\mathbf{s}_t|\{\mathbf{s}_\tau\}_{\tau\neq t},\mathbf{O}) = P_\Lambda(\mathbf{s}_t|\mathbf{s}_{t-1},\mathbf{O})$.
- $\lambda^i_{\mathbf{s}_t}$ and $\lambda^j_{\mathbf{s}_t\mathbf{s}_{t-1}}$ are the Lagrange multipliers (weighting factors) associated to the characterizing functions $b_i(\mathbf{O},\mathbf{s}_t)$ and $a_j(\mathbf{s}_t,\mathbf{s}_{t-1})$.
- $Z_\Lambda(\mathbf{O})$ (Zustandsumme) is a normalization coefficient resulting from the natural constraints over the probabilities summation, commonly called the partition function and given by

$$Z_\Lambda(\mathbf{O}) = \sum_{\mathbf{S}}\prod_{t=1}^{T} \exp\Big(\sum_j \lambda^j_{\mathbf{s}_t\mathbf{s}_{t-1}} a_j(\mathbf{s}_t,\mathbf{s}_{t-1}) + \sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{O},\mathbf{s}_t)\Big),$$

and it is similar to the total probability $p(\mathbf{O}|\mathcal{M})$ in HMMs, which can be calculated using the forward algorithm [19]. The conditional distribution behind the CRF model as shown in Equation (5) implies arbitrary combinations of state scores $b_i(\mathbf{O},\mathbf{s}_t)$ and transition scores $a_j(\mathbf{s}_t,\mathbf{s}_{t-1})$. Hence, it is conceptually similar to HMMs that have *only* two scores; emission probability $p(\mathbf{o}_t|\mathbf{s}_t)$

and transition probabilities $P(\mathbf{s}_t|\mathbf{s}_{t-1})$. CRFs offer a principled framework for combining different state scores in a natural way. The HMMs and CRFs share the first order Markov assumption, which simplifies the training and decoding algorithms.

CRFs have an attractive property: the MaxEnt models (linear chain CRFs are a special case) make little assumptions, as they are the most unbiased distributions that are simultaneously consistent with a set of constraints. Hence, CRF models do not suffer from the observation independence assumption made in the HMM framework, as the characterizing functions may be statistically dependent or correlated. This is very clear in the model equation where the characterizing functions $b_i(\mathbf{O}, \mathbf{s}_t)$ are arbitrary functions over the *entire* observation sequence $\mathbf{O}$. Moreover, CRF models do not constrain the shape of the data generation and the modeling quality is a function of the sufficient statistics represented by the characterizing functions. In speech recognition problems, second order sufficient statistics are extracted from the acoustic observations.

The state characterizing function $b_i(\mathbf{O}, \mathbf{s}_t)$ can depend only on the current observation (i.e. observation $b_i(\mathbf{O}, \mathbf{s}_t) = b_i(\mathbf{o}_t, \mathbf{s}_t)$). For example, frontend speech processing generally extracts MFCC+Δ+$\Delta\Delta$ as the basic acoustic vector, the observation dependent term in Equation (5) is given by

$$\begin{aligned}\sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{O}, \mathbf{s}_t) =& \sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{o}_t, \mathbf{s}_t) \\ =&\lambda^0_{\mathbf{s}_t} b_0 + \sum_{i=1}^{2d} \Big(\lambda^i_{\mathbf{s}_t} \mathbf{o}_{ti} + \lambda^i_{\mathbf{s}_t} \Delta \mathbf{o}_{ti} + \lambda^i_{\mathbf{s}_t} \Delta\Delta \mathbf{o}_{ti} \\ &+ \lambda^i_{\mathbf{s}_t} \mathbf{o}^2_{ti} + \lambda^i_{\mathbf{s}_t} \Delta \mathbf{o}^2_{ti} + \lambda^i_{\mathbf{s}_t} \Delta\Delta \mathbf{o}^2_{ti} \Big),\end{aligned} \tag{6}$$

where b_0 is the bias constraint, d is the vector dimensionality, and $\mathbf{o}_{ti}$, $\mathbf{o}^2_{ti}$ are the first and second order moments of the acoustic features. Equation (6) can be written as

$$\sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{O}, \mathbf{s}_t) = \mathbf{o}_t^T \Lambda_{\mathbf{s}_t} \mathbf{o}_t + \lambda^T_{\mathbf{s}_t} \mathbf{o}_t + b_{\mathbf{s}_t 0}. \tag{7}$$

In addition, with *one* transition characterizing function, the transition dependent term in Equation (5) is given by

$$\sum_j \lambda^j_{\mathbf{s}_t \mathbf{s}_{t-1}} a_j(\mathbf{s}_t, \mathbf{s}_{t-1}) = \lambda_{\mathbf{s}_t \mathbf{s}_{t-1}} a(\mathbf{s}_t, \mathbf{s}_{t-1}), \tag{8}$$

where $a(\mathbf{s}_t, \mathbf{s}_{t-1})$ is a binary function and can be used to define CRF topology and $\lambda_{\mathbf{s}_t \mathbf{s}_{t-1}}$ is related to $\log a_{\mathbf{s}_t \mathbf{s}_{t-1}}$ in HMM modeling. An example of a CRF with left-to-right transition topology, which is used to model a phone in an acoustic model, is shown in Fig. 3.

Equation (7) shows the main limitation of CRF model as used for speech recognition systems. This equation shows that state activation is based on a

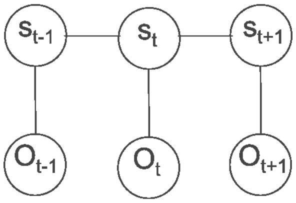

Figure 3: CRF model for phone representation.

single quadratic activation function. In HMM context, this means the state score is based on a single Gaussian component. This low complexity model cannot model the spectral phenomena. Therefore, CRF acoustic models will lead to poor recognition results.

Hidden Conditional Random Fields (HCRFs) were introduced to score the states based on a mixture of quadratic activation functions [24]. This idea extends the CRFs to be similar to HMMs with mixture of Gaussians. However, the exponential quadratic activation functions are more flexible discriminant functions than Gaussian densities, which are used for local observation scoring within the HMM (but the physical meaning of mean and variance is no longer available). Alternatively, deep architectures can be used to compute the state scores. This idea is explored in the following section.

3 Deep Conditional Random Fields

Deep Conditional Random Fields acoustic models are a particular implementation of linear chain CRFs where the state scores are computed based on a DNN that has many hidden layers. The feed-forward phase updates the output value of each neuron. Starting from the first hidden layer, each neuron output is computed as a weighted sum of inputs and applying the sigmoid function to it:

$$\mathbf{o}_{tj}^{h} = \mathrm{sigm}(\sum_{i=1}^{n} \lambda_{ij} \mathbf{o}_{ti}^{h-1}), \tag{9}$$

where $\mathbf{o}_t^h$ is an output of a hidden layer, n is the number of inputs, h is an index to a hidden layer, and sigmoid function is computed as follows:

$$\mathrm{sigm}(x) = \frac{1}{1+e^{-x}}. \tag{10}$$

The output of an hidden layer is passed to the next layer until the output layer is computed as follows:

$$\mathbf{o}_{tj}^{N} = \sum_{i=1}^{n} \lambda_{ij} \mathbf{o}_{ti}^{N-1}, \tag{11}$$

where N is the index of the output layer. Hence, the activation of hidden layers is nonlinear based on a sigmoid function and the output layer activation is linear.

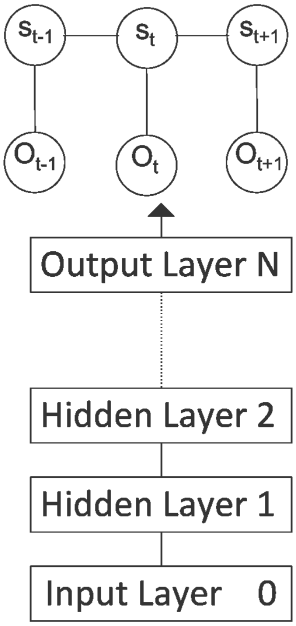

Figure 4: DCRF model for phone representation, where the state scores are computed from a DNN.

A graphical representation of the DCRF acoustic model is shown in Figure 4. The conditional distribution defining DCRFs is given by

$$P_\Lambda(\mathbf{S}|\mathbf{O}) = \frac{1}{Z_\Lambda(\mathbf{O})} \prod_{t=1}^{T} \exp\left(\lambda_{\mathbf{s}_t\mathbf{s}_{t-1}} a(\mathbf{s}_t, \mathbf{s}_{t-1}) + b_{\mathbf{s}_t}(\mathbf{o}_t)\right), \tag{12}$$

where $b_{\mathbf{s}_t}(\mathbf{o}_t) = \mathbf{o}^N_{ts_t}$ is computed from Equation (11). Hence, $b_{\mathbf{s}_t}(\mathbf{o}_t)$ connects DNN output to CRF input.

The partition function, $Z_\Lambda(\mathbf{O})$, is given by

$$Z_\Lambda(\mathbf{O}) = \sum_{\mathbf{S}} \prod_{t=1}^{T} \exp\left(\lambda_{\mathbf{s}_t\mathbf{s}_{t-1}} a(\mathbf{s}_t, \mathbf{s}_{t-1}) + b_{\mathbf{s}_t}(\mathbf{o}_t)\right), \tag{13}$$

and it can be calculated using the forward algorithm [19].

4 DCRF Optimization

For R training observations $\{\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_r, \dots, \mathbf{O}_R\}$ with corresponding transcriptions $\{W_r\}$, DCRFs are trained using the conditional maximum likelihood (CML) criterion to maximize the posterior probability of the correct word sequence given the acoustic observations:

$$\begin{aligned}\mathcal{F}_{\mathrm{CML}}(\Lambda) &= \sum_{r=1}^{R} \log P_\Lambda(\mathcal{M}_{W_r}|\mathbf{O}_r) \\ &= \sum_{r=1}^{R} \log \frac{P(W_r)\sum_{\mathbf{S}|W_r} \exp \sum_t^T \Psi(\mathbf{O},\mathbf{S},c,\Lambda)}{\sum_{\hat{W}} P(\hat{W}) \sum_{\mathbf{S}|\hat{W}} \exp \sum_t^T \Psi(\mathbf{O},\mathbf{S},c,\Lambda)} \\ &\approx \sum_{r=1}^{R} \log Z_\Lambda(\mathbf{O}_r|\mathcal{M}^{\mathrm{num}}) - \log Z_\Lambda(\mathbf{O}_r|\mathcal{M}^{\mathrm{den}}),\end{aligned} \tag{14}$$

where

$$\Psi(\mathbf{O},\mathbf{S},c,\Lambda) = \lambda_{\mathbf{s}_t\mathbf{s}_{t-1}} a(\mathbf{s}_t,\mathbf{s}_{t-1}) + b_{\mathbf{s}_t}(\mathbf{o}_t). \tag{15}$$

The optimal parameters, Λ^*, are estimated by maximizing the CML criterion, which implies minimizing the cross entropy between the correct transcription model and the hypothesized recognition model. In other words, the process maximizes the partition function of the correct models[1] (the numerator term) $Z_\Lambda(\mathbf{O}_r|\mathcal{M}^{\mathrm{num}})$, and simultaneously minimizes the partition function of the recognition model (the denominator term) $Z_\Lambda(\mathbf{O}_r|\mathcal{M}^{\mathrm{den}})$. The optimal parameters are obtained when the gradient of the CML criterion is zero.

4.1 Numerical Optimization for DCRFs

Newton's method can be used to estimate DCRFs based on local quadratic approximation of the CML objective function. These methods rely on local quadratic approximation by expanding the CML *nonlinear* objective function $\mathcal{F}_{\mathrm{CML}}(\Lambda+\delta)$ using Taylor expansion around the current model point Λ in parameter space and is given by

$$\mathcal{F}_{\mathrm{CML}}(\Lambda+\delta) \approx L(\Lambda) + \delta^T \mathbf{g}(\Lambda) + \frac{1}{2}\delta^T \mathbf{H}(\Lambda)\delta + \ldots, \tag{16}$$

where $\mathbf{g}(\Lambda)$ is the local gradient vector defined by

$$\mathbf{g}(\Lambda) = \frac{\partial \mathcal{F}_{\mathrm{CML}}(\Lambda)}{\partial \lambda_i}\Big|_\Lambda, \tag{17}$$

and the $\mathbf{H}(\Lambda)$ is the local Hessian matrix defined by

$$\mathbf{H}_{ij}(\Lambda) \equiv \frac{\partial \mathcal{F}_{\mathrm{CML}}(\Lambda)}{\partial \lambda_i \partial \lambda_j}\Big|_\Lambda. \tag{18}$$

The Newton's Method update rule is given by

$$\lambda^{(\tau)} = \lambda^{(\tau-1)} - \eta^{(\tau)} \mathbf{H}^{-1}(\Lambda)\mathbf{g}(\Lambda). \tag{19}$$

[1] Since a summation over potential functions is commonly called the partition function in undirected graphical modeling, we coin the notation $Z_\Lambda(\mathbf{O}_r|\mathcal{M}^{\mathrm{num}})$ for the summation of all possible state sequences of the correct models.

Since CML is not a quadratic function, taking the full Newton step $\mathbf{H}^{-1}(\Lambda)\mathbf{g}(\Lambda)$ may lead to an overshoot of the maximum. Hence, $\eta^{(\tau)} \neq 1$ will lead to the *damped* Newton step. A *line search* algorithm is used to calculate $\eta^{(\tau)}$. A line search works by evaluating the objective function starting from the current model in the direction of search and choosing $\eta^{(\tau)}$ will lead to an increase of the CML objective function.

Hessian matrix calculation, its inverting and storage, makes Newton's Method useful only for small scale problems. Quasi-Newton or variable metric methods can be used when it is impractical to evaluate the Hessian matrix. Instead of obtaining an estimate of the Hessian matrix at a single point, these methods gradually build up an approximate Hessian matrix by using gradient information from some or all of the previous iterates visited by the algorithm. Limited memory quasi-Newton's methods like L-BFGS are particular realizations of quasi-Newton's methods that cut down the storage for large problems [42].

Truncated-Newton method known as Hessian-Free approach [42, 43, 14], is a second order method for large scale problems. It finds the search direction using an iterative solver and the solver is typically based on conjugate gradient but other alternatives are possible. In this method, Hessian-vector products are computed without explicitly forming the Hessian. Hessian-free methods approximately invert the Hessian while quasi-Newton methods invert an approximate Hessian.

By ignoring the second order derivative, a first order approximation of the CML will lead to the gradient ascent methods and the update is given by

$$\lambda^{(\tau)} = \lambda^{(\tau-1)} + \eta\mathbf{g}(\Lambda). \tag{20}$$

The step size η must be small enough to ensure a stable increase of the CML objective function. It can be shown that the algorithm is convergent provided that η satisfies the condition $0 < \eta < \frac{2}{\lambda_{\max}}$, where $\lambda_{\max}$ is the largest eigenvalue of the Hessian matrix $\mathbf{H}(\mathbf{\Lambda}^{*})$ evaluated at the global maximum of the CML objective function [44]. In practice, second order statistics are not accumulated so $\lambda_{\max}$ is not known and η is chosen in an ad-hoc fashion by trial and error.

The training speed of gradient descent (batch mode) is usually slow. The training process can be accelerated using an online variant known as stochastic gradient descent (SGD).[2] This algorithm can update the learning system on the basis of the objective function measured for a single utterance or batch.

4.2 DCRFs Gradient Computation

For an exponential family activation function based on first-order sufficient statistics, the gradient of the CML objective function for the output layer parameters is given by

$$\nabla\mathcal{F}_{\text{CML}}(\mathbf{O}) = \mathcal{C}_{ji}^{\text{num}}(\mathbf{O}) - \mathcal{C}_{ji}^{\text{den}}(\mathbf{O}), \tag{21}$$

[2] Since CML objective function is maximized in this work, stochastic gradient ascent is used to train DCRFs models.

where the accumulators of the sufficient statistics, $\mathcal{C}_{ji}(\mathbf{O})$, for the j^{th} state and i^{th} constraint are calculated as follows:

$$\mathcal{C}^{\text{num}}_{ji}(\mathbf{O}) = \sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma^r_j(t|\mathcal{M}^{\text{num}})\mathbf{o}^{\text{N}}_{rti}, \tag{22}$$

$$\mathcal{C}^{\text{den}}_{ji}(\mathbf{O}) = \sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma^r_j(t|\mathcal{M}^{\text{den}})\mathbf{o}^{\text{N}}_{rti}, \tag{23}$$

where r is the utterance index and the frame-state alignment probability γ_j, is the probability of being in state j at some time t can be written in terms of the forward score $\alpha_j(t)$ and the backward score $\beta_j(t)$ as in HMMs:

$$\gamma_j(t|\mathcal{M}) = P(\mathbf{s}_t = j|\mathbf{O};\mathcal{M}) = \frac{\alpha_j(t|\mathcal{M})\beta_j(t|\mathcal{M})}{Z_\Lambda(\mathbf{O}|\mathcal{M})}, \tag{24}$$

The delta of the output layer neuron j is given by

$$\delta^N_{tj} = \gamma_j(t|\mathcal{M}^{\text{num}}) - \gamma_j(t|\mathcal{M}^{\text{den}}), \tag{25}$$

and the delta of the hidden layers:

$$\delta^h_{tj} = \mathbf{o}^h_{tj}(1-\mathbf{o}^h_{tj})\sum_{k\in\text{outputs}}\lambda^{h+1}_{kj}\delta^{h+1}_{kt}, \tag{26}$$

and the gradient for the hidden layers parameters is given by:

$$\frac{\partial\mathcal{F}_{\text{CML}}(\Lambda)}{\partial\lambda^h_{ki}} = \sum_{r=1}^{R}\sum_{t=1}^{T_r}\delta^h_{rtj}\mathbf{o}^{h-1}_{rtki}. \tag{27}$$

Based on Equation (27) and Equation (21), a gradient based optimization can be used to estimate the parameters [42]. The transition parameters are given by:

$$\lambda_{\mathbf{s}_t\mathbf{s}_{t-1}} = \log a_{\mathbf{s}_t\mathbf{s}_{t-1}}, \tag{28}$$

where $a_{\mathbf{s}_t\mathbf{s}_{t-1}}$ is the transition probability in HMM modeling and is estimated using the maximum likelihood (MLE) criterion.

5 DCRFs generative training

The training of DNNs is divided into two phases: generative training to initialize the network to a good starting point, which may lead to good results. Fine tuning phase, which basically is the discriminative training described in Section 4. In this section, we will review the restricted Boltzmann machine (RBM), which is the basic building block for generative pretrained DNNs.

5.1 Restricted Boltzmann Machine

Restricted Boltzmann Machines (RBMs) are a special case of Markov random field that have one layer of binary stochastic hidden units and one layer of (Bernoulli or Gaussian) stochastic visible units. As shown in Fig. 5, they are bipartite graphs, where all visible units are connected to all hidden units. An RBM assigns an energy to every configuration of visible and hidden vectors, denoted v and h respectively according to

$$E(\mathrm{v}, \mathrm{h}; \theta) = -\mathrm{b}^T \mathrm{v} - \mathrm{c}^T \mathrm{h} - \mathrm{h}^T \mathrm{W} \mathrm{v}, \tag{29}$$

where W is the matrix of visible/hidden connection weights, b is the visible unit bias, and c is the hidden unit bias. The joint distribution $p(\mathrm{v}, \mathrm{h}; \theta)$ over the visible units v and hidden units h, given the model parameters θ, is defined in terms of an energy function $E(\mathrm{v}, \mathrm{h}; \theta)$ of

$$p(\mathrm{v}, \mathrm{h}; \theta) = \frac{\exp(-E(\mathrm{v}, \mathrm{h}; \theta))}{Z}, \tag{30}$$

where the partition function is given by

$$Z = \sum_{\mathrm{v},\mathrm{h}} \exp(-E(\mathrm{v}, \mathrm{h}; \theta)), \tag{31}$$

and the marginal probability that the model assigns to a visible vector v is

$$p(\mathrm{v}; \theta) = \frac{\sum_{\mathrm{h}} \exp(-E(\mathrm{v}, \mathrm{h}; \theta))}{\sum_{\mathrm{h}} \sum_{\mathrm{u}} \exp(-E(\mathrm{u}, \mathrm{h}; \theta))}. \tag{32}$$

Since there is no hidden-hidden connections, the conditional distribution $p(\mathrm{h}|\mathrm{v}; \theta)$ is given by

$$p(\mathrm{h} = 1|\mathrm{v}; \theta) = \mathrm{sigm}(\mathrm{c} + \mathrm{v}^T \mathrm{W}). \tag{33}$$

Similarly, since there are no visible-visible connections, the conditional distribution $p(\mathrm{v}|\mathrm{h}; \theta)$ is given by

$$p(\mathrm{v} = 1|\mathrm{h}; \theta) = \mathrm{sigm}(\mathrm{b} + \mathrm{h}^T \mathrm{W}^T). \tag{34}$$

Although RBMs with the energy function of Equation (29) are suitable for binary input data, they can not be used for real-valued input data. For example, frontend of a speech recognition system generates real-valued acoustic features. Therefore, the Gaussian- Bernoulli restricted Boltzmann machine (GRBMs) can be used to handle real-valued data. The GRBM energy function

$$E(\mathrm{v}, \mathrm{h}; \theta) = \frac{1}{2}(\mathrm{v} - \mathrm{b})^T (\mathrm{v} - \mathrm{b}) - \mathrm{c}^T \mathrm{h} - \mathrm{v}^T \mathrm{W} \mathrm{h}. \tag{35}$$

Note that Equation (35) implicitly assumes that the visible units have a diagonal covariance Gaussian noise model with variance 1 for each dimension. The corresponding conditional distributions are given by

$$p(\mathrm{h} = 1|\mathrm{v}; \theta) = \mathrm{sigm}(\mathrm{c} + \mathrm{v}^T \mathrm{W}), \tag{36}$$

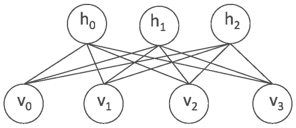

Figure 5: A graphical representation of Restricted Boltzmann Machine (RBM).

$$p(\mathrm{v}|\mathrm{h};\theta) = \mathcal{N}(\mathrm{v}; \mathrm{b} + \mathrm{h}^T \mathrm{W}^T, I), \tag{37}$$

where I is the identity matrix. Apart from these differences, the inference and learning rules for a GRBM are the same as for a binary RBM.

5.2 RBM Training

Exact maximum likelihood learning of large RBM is not feasible because it is exponentially expensive to compute the gradient of the log likelihood of the training data. Instead, an efficient approximate training procedure called "constrastive divergence" (CD) can be used to train an RBM [45]. To compute the log likelihood, let us define a quantity known as the free energy:

$$F(\mathrm{v};\theta) = -\sum_{\mathrm{h}} \exp(-E(\mathrm{v},\mathrm{h};\theta)), \tag{38}$$

Using $F(\mathrm{v};\theta)$, we can write the log likelihood as:

$$L(\theta) = -F(\mathrm{v};\theta) - \log(\sum_{\nu} \exp(-F(\nu;\theta))). \tag{39}$$

Taking the gradient of the log likelihood $L(\theta)$ we can derive the update rule for the RBM weights as:

$$\frac{\partial L(\theta)}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}. \tag{40}$$

The first expectation $\langle v_i h_j \rangle_{data}$, is the frequency which the visible unit v_i and the hidden unit h_i are active together in the training data and $\langle v_i h_j \rangle_{model}$ is that same expectation under the distribution defined by the model. The one step CD approximation for the gradient w.r.t. the visible-hidden weights is:

$$\frac{\partial L(\theta)}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{1}, \tag{41}$$

where $\langle v_i h_j \rangle_1$ is the expectation over-one step reconstructions. In other words, it is the expectation computed with samples generated by running a Gibbs sampler initialized at the data for one full step. A Gibbs sampler can be defined using Equation (33) and Equation (34).

Once the gradient is computed, SGD can be used to update the RBM parameters. The update equation is given by

$$w_{ij}^{\tau} = w_{ij}^{\tau-1} + \alpha \frac{\partial L(\theta)}{\partial w_{ij}}. \tag{42}$$

6 Unified frame based deep acoustic models

DCRFs and DNN/HMM systems compute the state scores using similar deep architectures. Hence, it is possible to establish connection between DCRFs and DNN/HMM systems. The difference between the two systems comes from three issues:

- The training criterion used to train each system.
- The state score of each system and this implies the output layer specifications of each system.
- The transition parameters of each system.

6.1 Training criterion

The traditional DCRFs are trained using sequence level CML training criterion to maximize the posterior probability of the correct word sequence given the acoustic observations as shown in Section 4. On the other hand, most DNN/HMM systems are trained using frame level CML training criterion (known as frame level cross entropy objective function).[3] In order to get comparable results we need to unify the training criterion used to train the two systems. In this work, we will use the frame level CML training criterion to train the two systems. Therefore, the training of DCRFs needs to be modified to be based on frame level CML training criterion. Hence, the $\gamma_j(t|\mathcal{M})$ computation is approximated with state estimates as follows [47]:

$$\gamma_j(t|\mathcal{M}^{\text{den}}) = \frac{\exp\left(\mathbf{o}_{tj}^N\right)}{\sum_{\mathbf{s}} \exp\left(\mathbf{o}_{ts}^N\right)}. \tag{43}$$

Based on this approximation, the training criterion used to train the two systems are identical and differences in the results related to the training criterion are eliminated.

6.2 DNN output layer

The conditional distribution defining hybrid DNN/HMM may be given by

$$P_\Lambda(\mathbf{S}|\mathbf{O}) = \frac{1}{Z_\Lambda(\mathbf{O})} \prod_{t=1}^{T} \exp\Big(\lambda_{\mathbf{s}_t\mathbf{s}_{t-1}} a(\mathbf{s}_t, \mathbf{s}_{t-1}) + b_{\mathbf{s}_t}(\mathbf{o}_t)\Big), \tag{44}$$

where $b_{\mathbf{s}_t}(\mathbf{o}_t) = \frac{P_\Lambda(\mathbf{s}_t|\mathbf{o}_t)}{P(\mathbf{s}_t)}$. It is worth to mention that this conditional distribution is very similar to DCRF conditional distribution described in Equation (12). The only difference is how the state score is computed in each model.

In hybrid DNN/HMM speech recognition systems, the HMM state scores are computed based on Equation (4). This equation implies the calculations

[3] Hybrid DNN/HMM systems can be trained using a sequence training criterion [46].

Table 1: Output layer design in different deep acoustic models.

System	Output layer score	Activation function
DCRF	$b_{\mathbf{s}_t}(\mathbf{o}_t) = \mathbf{o}_{ts_t}^N$	linear
DNN/HMM1	$b_{\mathbf{s}_t}(\mathbf{o}_t) = \frac{P_\Lambda(\mathbf{s}_t\|\mathbf{o}_t)}{P(\mathbf{s}_t)}$	softmax
DNN/HMM2	$b_{\mathbf{s}_t}(\mathbf{o}_t) = P_\Lambda(\mathbf{s}_t\|\mathbf{o}_t)$	softmax

of a softmax activation function for each frame to compute the state posteriors. On the other hand, DCRFs state scores are based on a linear activation function in the output layer based on Equation (11). Hence, it is possible to convert DNN/HMM systems to DCRFs by removing the output softmax layer and decode directly using the linear output activation. Due to the different output layer specifications, the DCRFs and hybrid DNN/HMM system may use different language scaling factor to lead to exact results. Threfore, in order to convert DNN/HMM system to DCRF system:

1. Train the DNN/HMM using frame cross entropy criterion.
2. Remove the softmax output layer.
3. Decode directly using the linear output activation.

Another form of DNN/HMM hybrid system is to assume that the state score is computed directly from the posterior probably of a connectionist estimator. In particular, $b_{\mathbf{s}_t}(\mathbf{o}_t)$ is given by:

$$b_{\mathbf{s}_t}(\mathbf{o}_t) = P_\Lambda(\mathbf{s}_t|\mathbf{o}_t). \tag{45}$$

This implies that $P(\mathbf{s}_t)$ is a uniform distribution. It will be shown in the experimental section that these systems lead to exact results as DCRFs.

Table 1 details the DNN output layer in the different systems under our unified deep acoustic modeling.

6.3 Transition parameters

The transition parameters may be a source of different results between DCRFs and DNN/HMM systems.[4] In order to unify the state scores between the two systems, the transition parameters should be identical for the two systems. This is easily achieved as described in section 4 by setting the transition parameters of the two systems using:

$$\lambda_{\mathbf{s}_t\mathbf{s}_{t-1}} = \log a_{\mathbf{s}_t\mathbf{s}_{t-1}}, \tag{46}$$

where $a_{\mathbf{s}_t\mathbf{s}_{t-1}}$ is the transition probability in HMM modeling and is estimated using the maximum likelihood criterion.

[4] It is known that the transition scores have little impact on the recognition results.

7 Experiments

We have carried out phone recognition experiments on the TIMIT corpus [48]. We used the 462 speaker training set, testing on the 24 speaker core test set, and the development set is based on 50 speakers from the test set [49]. The SA1 and SA2 utterances were not used. The speech was analyzed using a 25ms Hamming window with a 10 ms fixed frame rate. We represented the speech using 12 mel frequency cepstral coefficients (MFCCs), energy, along with their first and second temporal derivatives, resulting in a 39 element feature vector. Another representation is based on using a Log-Fourier-transform-based filterbank with 40 coefficients (plus energy) distributed on a mel-scale, together with their first and second temporal derivatives resulting in a 123 element feature vector. The features are pre-processed to have zero mean and unit variance and acoustic context information is integrated using a window of 9 frames (4 left + current frame+ 4 right) to construct the final frames.

Following Lee and Hon [50], the original 61 phone classes in TIMIT were mapped to a set of 48 labels, which were used for training. This set of 48 phone classes was mapped down to a set of 39 classes [50], after decoding, and phone recognition results are reported on these classes, in terms of the phone error rate (PER), which is analogous to word error rate.

The baseline HMMs have three emitting states and the emission probabilities were modeled with mixtures of Gaussian densities with diagonal covariance matrices. The generative context-dependent HMMs (contained 1127 physical states, with 20 mixture components per state) were trained by the maximum likelihood criterion using the conventional EM algorithm [51]. The system is used only to provide the state alignment of the training data.

Each phone was represented using a three state left-to-right DCRF, all parameters of DNN were initialized to random values and the transition parameters were initialized from trained HMM models forcing left to right DCRFs (the transition parameters are held fixed after the initialization). The training procedure accumulated the $\mathcal{M}^{\text{num}}$ sufficient statistics via a Viterbi pass (forced alignment) of the reference transcription using HMMs trained using maximum likelihood criterion. The language model scaling factor is set to 1.0 during the decoding process. All our experiments used a bigram language model over phones, estimated from the training set. In-house decoder is used to generate the recognition phone sequence.

For training DNNs, the PDNNTK toolkit is used[52] and it is based on Theano library [53], which supports transparent computation for CPUs and GPUs. In addition, the MFCC results is based on an in-house code developed based on Theano. In Table 2, DCRFs recognition performance is reported in terms of PER on TIMIT task (core test set) for MFCC based frontend.

The results based on filterbank frontend are shown in Table 3. Some DCRFs models were pretrained when the number of hidden layers is large. When the number oh hidden layers was 9, the LM scaling factor was set to 1.5.

It is possible to unify deep acoustic models based on a framework presented in section 6. However, the state score is different and may lead to different

Table 2: DCRF decoding results on TIMIT recognition task in terms of PER (MFCC based frontend).

#of Hidden layers	#of neuron	PER
1	8192	25.1%
2	3072	24.4%
3	3072	24.2%
4	3072	23.9%

Table 3: DCRF decoding results on TIMIT recognition task in terms of PER (FBANK based frontend).

#of Hidden layers	#of neuron	PER	Note
2	2048	24.2%	
4	3072	23.1%	
4	3072	23.0%	pretrained
5	3072	22.9%	pretrained
9	2048	**22.7**%	pretrained

decoding results. The DCRFs decoder is modified to support DNN/HMM1 and DNN/HMM2 decoding based on equations summarized in Table 1. As shown in Table 4, the decoding results are sensitive to the value of the language model scaling factor. It is clear that DCRFs and DNN/HMM2 hybrid systems lead to exact PER results.

8 Discussions

In this section we address several issues about the implementation of DCRFs.

8.1 Decoding speed

In hybrid ANN/HMM speech recognition systems, the HMM state scores are computed based on Equation (4). This equation implies the calculations of a softmax activation function for each frame to compute the state posteriors. However, in efficient implementations of DNN/HMM decoders, the softmax cal-

Table 4: Comparison between different acoustic models using a unified framework on TIMIT recognition task in terms of PER (FBANK based frontend).

LM sacling factor	DCRFs	DNN/HMM1	DNN/HMM2
1	23.1%	24.3%	23.1%
1.5	22.7%	23.9%	22.7%
2	23.0%	23.8%	23.0%

culations are ignored and the state scores are based on a linear activation function in the output layer. On the other hand, DCRFs state scores are based on a linear activation function. Consequently, DCRFs and traditional DNN/HMM systems have the same decoding speed.

8.2 Related prior work

The multilayer conditional random field (ML-CRF) was introduced in [39]. In this model, CRF is trained on the top of a single hidden layer constructed from scoring a large number of sigmoid functions. Hence, ML-CRF implies shallow neural networks. In addition, each phone was represented using a single state in the model. The Language model parameters are trained within the ML-CRF framework by defining bi-gram transition constraints. The training algorithm supports error backpropagation.

In deep-structured CRF [25], multi-layer CRF model was developed where the marginal probabilities obtained from the outputs of a lower layer are used as the input of the higher layer. The model can be further extended for phonetic recognition using a variant called deep hidden conditional random field (DHCRF) [26]. In this model, the final layer is a Hidden Conditional Random Field (HCRF) [24] and the intermediate layers are zero-th-order CRFs. The DHCRF supports bi-gram language model features. Although the model has a deep architecture, it does not support DNN and the training algorithm does not support error backpropagation. DHCRFs were further modified to support DNN in [27], where state scores are computed based on DNN setup but the output layer has a softmax activation function. This version of the algorithm supports RBM training for initialization and error backpropagation training algorithm for finetuning.

In [29], deep hidden conditional neural fields (Deep-HCNF) which utilized an observation function with deep structure were presented. The state scores are computed based on DNN setup and the output layer has a linear activation function as in [39] and our work. Deep-HCNF supports bi-gram language model features and Boosted-MMI training criterion (BMMI).

In this work, the state scores are also computed based DNN architecture and the output layer has a linear activation function. In addition, we do not estimate state transition parameters or language model parameters within DCRF framework. The state transition parameters were estimated using traditional HMM framework. Moreover, Maximum Likelihood (ML) criterion is used to estimate bigram language model. Hence, DCRF architecture may be computationally efficient for training and decoding. During the decoding process, a language model scaling factor is used to improve the results. On the other hand, frame level CML criterion is used to estimate DCRFs rather than the full-sequence CML training.

9 Conclusions

In this paper, we present a method to construct deep conditional random fields. In this approach, the state scores are computed based on a DNN that has many hidden layers. The feed-forward phase updates the output value of each neuron. Starting from the first hidden layer, each neuron output is computed as a weighted sum of inputs and applying the sigmoid function to it. The output is forwarded to the next layer until the output layer is updated as a weighted sum of inputs. DCRF state scores are connects the DNN output layer. Hence, the gradient is computed and a back-propagation algorithm is used to compute the gradient of each parameter in the hidden layers.

It was shown in the paper , it is possible to unify the deep acoustic models under a variation of CRF framework. Under certain assumptions presented in the paper, both DCRFs and hybrid DNN/HMM systems can lead to same exact results for a phone recognition task. In addition, linear activation functions are used in the DCRFs output layer. Consequently, DCRFs and traditional DNN/HMM systems have the same decoding speed. In addition, it is possible to convert DNN/HMM hybrid systems to DCRFs using a procedure addressed in the paper. On the other hand, we do not estimate state transition parameters or language model parameters within DCRF framework. The state transition parameters were estimated using traditional HMM framework. Moreover, Maximum Likelihood (ML) criterion is used to estimate bigram language model. Hence, the presented DCRF architecture may be computationally efficient for training and decoding.

References

[1] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. of IEEE 77 (2) (1989) 257–286.

[2] F. Jelinek, Statistical Methods for Speech Recognition, MIT Press, 1997.

[3] X. Huang, A. Acero, H.-W. Hon, Spoken Language Processing: A Guide to Theory, Algorithm and System Development, Prentice Hall, 2001.

[4] J. Bilmes, What HMMs can do, IEICE Transactions on Information and Systems E89-D (3) (2006) 869–891.

[5] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society 39 (1) (1977) 1–38.

[6] S. Renals, N. Morgan, H. Bourlard, M. Cohen, H. Franco, Connectionist probability estimators in HMM speech recognition, IEEE Transactions on Speech and Audio Processing.

[7] N. Morgan, H. Bourlard, Continuous speech recognition: An introduction to the hybrid HMM/connectionist approach, IEEE Signal Processing Magazine 12 (3) (1995) 25–42.

[8] E. Trentin, M. Gori, A survey of hybrid ANN/HMM models for automatic speech recognition, Neurocomputing 37 (1-4) (2001) 91–126.

[9] A. Robinson, An application of recurrent neural nets to phone probability estimation, IEEE Transactions on Neural Networks 5 (2) (1994) 298–305.

[10] A. Mohamed, G. Dahl, G. Hinton, Acoustic modeling using Deep Belief Networks, IEEE Transactions on Audio, Speech and Language Processing 20 (2012) 14–22.

[11] F. Seide, G. Li, D. Y. ., Conversational speech transcription using context-dependent Deep Neural Networks, in: Interspeech, 2011.

[12] G. Dahl, D. Yu, L. Deng, 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 Langauge Processing.

[13] G. Hinton, L. Deng, D. Yu, G. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, , B. Kingsbury, Deep Neural Networks for acoustic modeling in speech recognition, IEEE Signal Processing Magazine.

[14] B. Kingsbury, T. N. Sainath, H. Soltau, Scalable minimum bayes risk training of Deep Neural Network acoustic models using distributed hessian-free optimization, in: INTERSPEECH, 2012.

[15] O. Vinyals, D. Povey, Krylov subspace descent for deep learning, in: AISTATS, 2012.

[16] Y. Hifny, Deep learning using a Manhattan update rule, Deep Learning for Audio, Speech and Language Processing, ICML.

[17] K. Van Horn, A maximum-entropy solution to the frame dependency problem in speech recognition, Tech. rep., Dept. of Computer Science, North Dakota State University (2001).

[18] W. Macherey, H. Ney, A comparative study on maximum entropy and discriminative training for acoustic modeling in automatic speech recognition, in: Proc. EUROSPEECH, Geneva, Switzerland, 2003, pp. 493–496.

[19] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proc. ICML, 2001, pp. 282–289.

[20] J. Hennebert, C. Ris, H. Bourlard, S. Renals, N. Morgan, Estimation of global posteriors and forward-backward training of hybrid HMM/ANN systems, in: Proc. Eurospeech, Rhodes, 1997, pp. 1951–1954.

[21] A. Krogh, S. K. Riis, Hidden neural networks, Neural Computation 11 (2) (1999) 541–563.

[22] M. Gales, S. Watanabe, E. Fosler-Lussier, Structured discriminative models for speech recognition, IEEE Signal Processing Magazine.

[23] E. Fosler-Lussier, Y. He, P. Jyothi, R. Prabhavalkar, Conditional random fields in speech, audio, and language processing, Proceedings of the IEEE 101 (5) (2013) 1054–1075. doi:10.1109/JPROC.2013.2248112.

[24] A. Gunawardana, M. Mahajan, A. Acero, J. Platt, Hidden conditional random fields for phone classification, in: Proc. INTERSPEECH, Lisbon, Portugal, 2005, pp. 1117–1120.

[25] D. Yu, S. Wang, L. Deng, Sequential labeling using deep-structured conditional random fields, IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING.

[26] D. Yu, L. Deng, Deep-structured hidden conditional random fields for phonetic recognition, in: Proc. INTERSPEECH, 2010.

[27] A. Mohamed, D. Yu, L. Deng, Investigation of full-sequence training of Deep Belief Networks for speech recognition, in: Interspeech, 2010.

[28] T.-M.-T. Do, T. Artieres, Neural conditional random fields, in: Proc. of the 13th Interational Conference on Artificial Intelligence and Statistics, (AI-STATS), 2010.

[29] Y. Fujii, K. Yamamoto, S. Nakagawa, Deep-hidden conditional neural fields for continuous phoneme speech recognition, in: Proc. IWSML, 2012.

[30] G. Zweig, P. Nguyen, D. V. Compernolle, K. Demuynck, L. Atlas, P. Clark, G. Sell, M. Wang, F. Sha, H. Hermansky, D. Karakos, A. Jansen, S. Thomas, G. Sivaram, S. Bowman, J. Kao, Speech recognition with segmental conditional random fields: A summary of the JHU CLSP summer workshop, in: Proc. IEEE ICASSP, 2011.

[31] Y. Hifny, Conditional random fields for continuous speech recognition, Ph.D. thesis, University Of Sheffield (2006).

[32] Y. Hifny, S. Renals, Speech recognition using augmented conditional random fields, IEEE Transactions on Audio, Speech and Language Processing 17 (2) (2009) 354–365.

[33] A. Likhododev, Y. Gao, Direct models for phoneme recognition, in: Proc. IEEE ICASSP, Vol. 1, Orlando, FL, USA, 2002, pp. 89–92.

[34] J. K. Hong-Kwang, Y. Gao, Maximum entropy direct models for speech recognition, in: Proc IEEE ASRU Workshop, St. Thomas, U.S. Virgin Islands, 2003, pp. 1– 6.

[35] N. Smith, M. Gales, M. Niranjan, Data dependent kernels in SVM classification of speech patterns, Tech. Rep. CUED/F-INFENG/TR.387, University of Cambridge (2001).

[36] N. Smith, M. Gales, Speech recognition using SVMs, in: Proc. NIPS, Vol. 14, 2002.

[37] T. S. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: Proc. NIPS, Vol. 11, 1998.

[38] M. Layton, M. Gales, Augmented statistical models for speech recognition, in: Proc. IEEE ICASSP, Vol. 1, France, 2006, pp. 129– 132.

[39] R. Prabhavalkar, E. Fosler-Lussier, Backpropagation training for multilayer conditional random field based phone recognition, in: Proc. IEEE ICASSP, Vol. 1, France, 2010, pp. 5534 – 5537.

[40] Y. Hifny, Acoustic modeling based on deep conditional random fields, Deep Learning for Audio, Speech and Language Processing, ICML.

[41] E. T. Jaynes, On the rationale of maximum-entropy methods, Proc. of IEEE 70 (9) (1982) 939–952.

[42] J. Nocedal, S. J. Wright, Numerical Optimization, Springer, 1999.

[43] J. Martens, Deep learning via hessian-free optimization, in: Proc. ICML, 2010.

[44] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice Hal, 1998.

[45] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Computation 14 (2002) 1771–1800.

[46] B. Kingsbury, Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling, in: Proc. IEEE ICASSP, 2009, pp. 3761–3764. doi:10.1109/ICASSP.2009.4960445.

[47] Y. Hifny, S. Renals, N. Lawrence, A hybrid MaxEnt/HMM based ASR system, in: Proc. INTERSPEECH, Lisbon, Portugal, 2005, pp. 3017–3020.

[48] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, V. Zue, TIMIT acoustic-phonetic continuous speech corpus (1990).
URL http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1

[49] A. Halberstadt, J. Glass, Heterogeneous measurements and multiple classifiers for speech recognition, in: Proc. ICSLP, Vol. 3, Sydney, Australia, 1998, pp. 995–998.

[50] K.-F. Lee, H.-W. Hon, Speaker-independent phone recognition using hidden Markov models, IEEE Transactions on Speech and Audio Processing 37 (11) (1989) 1641–1648.

[51] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, P. Woodland, The HTK Book, Version 3.1, 2001.

[52] Y. Miao, PDNN: Yet Another Python Toolkit for Deep Neural Networks. URL http://www.cs.cmu.edu/ ymiao/pdnntk.html

[53] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, Y. Bengio, Theano: a CPU and GPU math expression compiler, in: Proceedings of the Python for Scientific Computing Conference (SciPy), 2010, oral Presentation.