Neural Higher-Order Factors in Conditional Random Fields for Phoneme Classification

Martin Ratajczak[1], Sebastian Tschiatschek[2], Franz Pernkopf[1]

[1]Graz University of Technology, Signal Processing and Speech Communication Laboratory
martin.ratajczak@tugraz.at, pernkopf@tugraz.at
[2]ETH Zurich, Learning & Adaptive Systems Group
sebastian.tschiatschek@inf.ethz.ch

Abstract

We explore neural higher-order input-dependent factors in linear-chain conditional random fields (LC-CRFs) for sequence labeling. It is a fusion of two powerful models as higher-order LC-CRFs with linear factors are well-established for sequence labeling tasks, but they lack to model non-linear dependencies. Therefore, we present neural higher-order input-dependent factors which map sub-sequences of inputs to sub-sequences of outputs using distinct multilayer perceptron sub-networks. This is important in many tasks, in particular, for phoneme classification where the phone representation strongly depends on the context phonemes. Experimental results for phoneme classification with LC-CRFs and neural higher-order factors confirm this fact and we achieve the best ever reported phoneme classification performance on TIMIT, i.e. a phoneme error rate of 15.8%. Furthermore, we show that the success is not obvious as linear high-order factors degrade phoneme classification performance on TIMIT.

Index Terms: Neural higher-order factors, conditional random field, multi-layer perceptron networks

1. Introduction

In *sequence labeling*, we assign some given input sequence $\mathbf{x}$, e.g. a time series, to an output label sequence $\mathbf{y}$. *Linear-chain conditional random fields (LC-CRFs)* are established models for sequence labeling [1], e.g. speech recognition [2], optical character recognition and natural language processing [3]. Due to several advantages, LC-CRFs achieve better performance compared to their generative counterparts, i.e. hidden Markov models (HMMs): First, observed variables can be dependent given the label sequence. This increases the model's expressiveness. Second, the discriminative objective function, the conditional likelihood, directly optimizes the relationship between input and output variables, i.e. it focuses on the prediction of the best output label sequence instead of estimating the joint probability distribution over the input and output variables. Third, normalization is performed over the whole output sequence. In contrast, HMMs are normalized locally. The normalization over the whole sequence counteracts the *label bias* problem.

First-order LC-CRFs typically consist of transition factors, modeling the relationship between two consecutive output labels, and local factors, modeling the relationship between input observations (usually a sliding window over input frames) and one output label. But CRFs are not limited to such types of factors: *Higher-order LC-CRFs (HO-LC-CRFs)* allow for arbitrary input-independent (such factors depend on the output labels only) [3] and input-dependent (such factors depend on both the input and output variables) higher-order factors [4, 5]. That means both types of factors can include more than two output labels. Clearly, the Markov order of the largest factor (on the output side) dictates the order of LC-CRF.

Unfortunately, higher-order CRFs increase the model complexity as the number of features grows exponentially with the number of the output variables considered in higher order factors [6]. Consequently, to avoid over-fitting, the amount of training data has to be sufficiently large for training. Alternatively, transforming the data into a proper representation may reduce the over-fitting problem. It is common practice to represent the higher-order factors by linear functions which can reduce the model's expressiveness. To overcome this limitation, a widely used approach is to represent non-linear dependencies by parametrized models and to learn these models from data. Several approaches have been suggested to parametrize *first-order* factors in LC-CRFs, mainly kernel methods [7] and *neural models* [8, 9, 10, 11, 12]. In summary, most work in the past focused either on (a) higher-order factors represented by simple linear models, or (b) first-order input-dependent factors represented by neural networks. In this work, we explore neural *and* higher-order input-dependent factors in LC-CRFs.

Our main contributions are: (i) We introduce *neural higher-order input-dependent factors* in LC-CRFs depending on both sub-sequences of the input and the output labels. These factors are represented by *distinct multi-layer perceptron (MLP) networks* which are first *discriminatively pre-trained for prediction of label sub-sequences*. The activations of the last hidden layer are used as features. Afterwards, the HO-LC-CRF is optimized on the conditional likelihood for full label sequence. (ii) Experimental results for phoneme classification are presented. LC-CRFs with neural higher-order factors achieve the best classification performance of 15.8% on the TIMIT phoneme classification task. Further, we show that linear higher-order factors degrade classification performance on TIMIT.

The remainder of this paper is structured as follows: In Section 2 we briefly review related work. In Section 3 we introduce the model. In Section 4 we evaluate our model on the TIMIT phone classification task. Section 5 concludes the paper.

2. Related Work

For sequence labeling, HO-LC-CRFs have been applied in tagging tasks and handwriting recognition [3]. Their HO-LC-CRFs consist of overlapping higher-order factors, i.e. fac-

tors which represent n-gram model dependencies of different length. In another work, overlapping input-dependent higher-order factors have been applied in handwriting recognition [4]. In a large scale task, HO-LC-CRFs with non-overlapping input-dependent factors have been applied to statistical machine translation [5].

In these works, higher-order factors have not been modeled by neural networks which is the gap we fill. However, first-order factors have been already modeled by several types of neural networks. Conditional neural fields (CNFs) [9] and multi-layer CRFs [10], propose a direct method to optimize MLP networks and LC-CRFs under the conditional likelihood criterion based on error back-propagation. Another approach is to pre-train an unsupervised representation with a deep belief network, transform it into an MLP network and finally fine-tune the network in conjunction with the LC-CRF [13]. *Hidden-unit conditional random field* (HU-CRF) [11] replaces local factors by *discriminative RBMs* (DRBM) [8], CNN triangular CRF [14] by convolutional neural networks (CNNs) and *context-specific deep CRFs* [12] by sum-product networks [15, 16] which can be interpreted as *discriminative deep Gaussian mixture models* generalizing discriminative Gaussian mixture models to multiple layers of hidden variables.

In more detail, we contrast our work from [13]: First, although formulated quite general that work focused on neural first-order factors in contrast to neural higher-order factors in our work. Second, they used one shared neural network for all factors in contrast to distinct neural networks as in our case. Third, that work utilized generative and unsupervised pre-training using deep belief networks as initialization of their MLP network. We discriminatively pre-trained distinct higher-order MLP factors using an individual conditional likelihood for each label sub-sequences length.

Finally, in computer vision higher-order factors in Markov random fields [17] and conditional random fields [18, 6, 19] are much more common than in sequence labeling. Most of that work focus on higher-order factors represented by products of experts [20]. Typically, approximate inference such as belief propagation or a sampling method is utilized. We use exact inference in this paper.

3. Higher-Order Conditional Random Fields

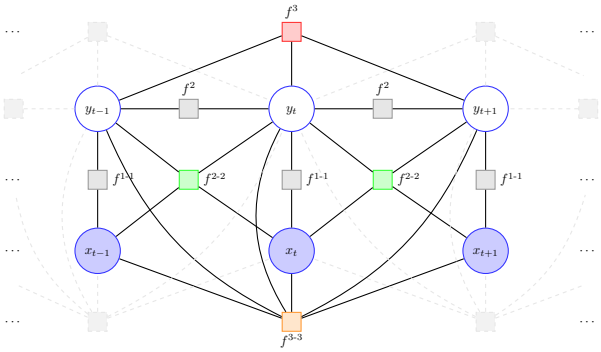

Figure 1: Factor graph of LC-CRF using input-dependent uni-gram features $f^{1\text{-}1}$ and bi-gram transition features f^2 (typical) and additionally n-gram features f^3 as well as input-dependent higher-order features $f^{2\text{-}2}$ and $f^{3\text{-}3}$ (higher-order).

We consider HO-LC-CRFs for sequence labeling. The HO-LC-CRF defines a conditional distribution

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \prod_{n=1}^{N} \Phi_t(\mathbf{y}_{t-n+1:t}; \mathbf{x}), \tag{1}$$

for an output sequence $\mathbf{y}$ of length T given an input sequence $\mathbf{x}$ of length T where $\Phi_t(\mathbf{y}_{t-n+1:t}; \mathbf{x})$ are non-negative factors that can depend on the label sub-sequence $\mathbf{y}_{t-n+1:t}$ and the whole input sequence $\mathbf{x}$, and where $Z(\mathbf{x})$ is an input-dependent normalization computed as

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{t=1}^{T} \prod_{n=1}^{N} \Phi_t(\mathbf{y}_{t-n+1:t}; \mathbf{x}). \tag{2}$$

An $(\mathrm{N}-1)^{\text{th}}$-order CRF models label sub-sequences of maximal span N in the corresponding factors. The factors $\Phi_t(\mathbf{y}_{t-n+1:t}; \mathbf{x})$ are assumed to be given in log-linear form, i.e.

$$\Phi_t(\mathbf{y}_{t-n+1:t}; \mathbf{x}) = \exp\left(\sum_k \mathbf{w}_k^{t,n} \mathbf{f}_k(\mathbf{y}_{t-n+1:t}; t, \mathbf{x})\right), \tag{3}$$

where $\mathbf{f}_k(\mathbf{y}_{t-n+1:t}; t, \mathbf{x})$ are arbitrary vector-valued and (possibly) position-dependent feature functions and $\mathbf{w}_k^{t,n}$ are the weights. These functions can be any functions ranging from simple indicator functions, linear functions, up to functions computed using neural networks. We distinguish the following types of feature functions:

n-gram input-independent features. These features are observation-independent, i.e. $\mathbf{f}_k(\mathbf{y}_{t-n+1:t}; t, \mathbf{x}) = \mathbf{f}^n(\mathbf{y}_{t-n+1:t})$. They compute vectors where every entry corresponds to the indicator function of a certain label sub-sequence $\mathbf{a}_i$, i.e. $\mathbf{f}^n(\mathbf{y}_{t-n+1:t}) = [\mathbf{1}(\mathbf{y}_{t-n+1:t} = \mathbf{a}_1), \mathbf{1}(\mathbf{y}_{t-n+1:t} = \mathbf{a}_2), \ldots]^T$. Typically $\mathbf{a}_1, \mathbf{a}_2, \ldots$ enumerate all possible label sub-sequences of length n. These transition functions are denoted as f^n in Figure 1.

m-n-gram input-dependent MLP features. These features generalize local factors to longer label sub-sequences. In this way, these feature functions can depend on the label sub-sequence $\mathbf{y}_{t-n+1:t}$ and an input sub-sequence of $\mathbf{x}$, i.e. $\mathbf{f}^{m-n}(\mathbf{y}_{t-n+1:t}; t, \mathbf{x}) = [\mathbf{1}(\mathbf{y}_{t-n+1:t} = \mathbf{a}_1)\, \mathbf{g}^m(\mathbf{x}, t), \ldots]^T$ where $\mathbf{g}^m(\mathbf{x}, t)$ is an arbitrary function. This function maps an input sub-sequence into a new feature space. In this work, we choose to use MLP networks for this function being able to model complex interactions among the variables. More specific, the hidden activations of the last layer $\mathbf{h}^m(\mathbf{x}, t)$ of the MLP network are used, i.e. $\mathbf{g}^m(\mathbf{x}, t) = \mathbf{h}^m(\mathbf{x}, t)$. We call these features *m-n-gram MLP features*. They are denoted as $f^{m\text{-}n}$ in Figure 1, assuming that they only depend on input output sub-sequences. Although possible, we do not consider the full input sequence $\mathbf{x}$ to counteract over-fitting, but only use a symmetric and centered contextual window of length m around position t or time interval $t - n + 1 : t$. Exemplary, in case of two labels and four input sub-sequences we include the inputs from time interval $t - 2 : t + 1$. An important extension to prior work is that the m-n-gram MLP features are modeled by separate networks to represent different non-linear interdependences between input and output sub-sequences.

Figure 1 shows LC-CRF as factor graph. A typical LC-CRF consists of input-dependent uni-gram features f^{1-1} and input-independent bi-gram features f^2. In this work, we consider a rare used extension using higher-order input-dependent m-n-gram features, for example f^{3-3}, shown in Figure 1.

The benefit of input-dependent higher-order factors for phoneme classification is substantiated by the fact that the spectral properties of phonemes are strongly influenced by neighboring phonemes. This is illustrated in Figure 2. In conventional speech recognition systems, this well-known fact is tackled by introducing meta-labels in form of tri-phone models [21]. Input-dependent higher-order factors in HO-LC-CRF support this by mapping an input sub-sequence to an output subsequence, i.e. several output labels, without introducing meta-labels. Further in HO-LC-CRF, we are able to model arbitrary input sub-sequences of length m into arbitrary output subsequences of length n, i.e we can also model mono-phones, bi-phones and tri-phones within the same model.

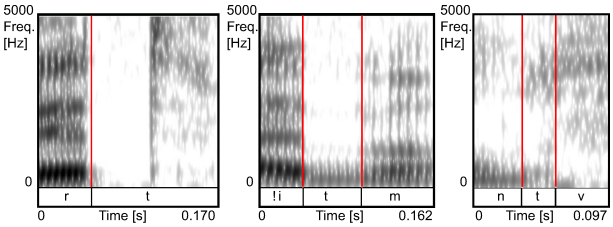

Figure 2: Three realizations of word-final /t/ in spontaneous Dutch. Left panel: Realization of /rt/ in *gestudeerd* 'studied'. Middle panel: Realization of /ˈɛitm/ in *leeftijd mag* 'age is allowed'. Right panel: Realization of /ntv/ in *want volgens* 'because according' [22].

3.1. Parameter Learning

Parameters $\mathbf{w} = \{\mathbf{w}_k^{t,n} \mid \forall k, t, n\}$ are optimized to maximize the conditional log-likelihood of the training-data, i.e.

$$\mathcal{F}(\mathbf{w}) = \sum_{j=1}^{J} \log p(\mathbf{y}^{(j)}|\mathbf{x}^{(j)}), \tag{4}$$

where $((\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(J)}, \mathbf{y}^{(J)}))$ is a collection of J input-output sequence pairs drawn i.i.d. from some data distribution. The partial derivatives of (4) with respect to the weights $\mathbf{w}_k^{t,n}$ can be computed as described in [3, 4]. To perform parameter learning using gradient ascent all marginal posteriors of the form $p(\mathbf{y}_{t-n+1:t}|t, \mathbf{x}^{(j)})$ are required. This can be achieved efficiently using the *forward-backward algorithm* which we describe for 2[nd]-order CRFs in the following. The algorithm can be easily extended to CRFs of order $(N-1) > 2$.

3.2. Forward-backward Algorithm for 2[nd]-order CRFs.

The ingredients of the forward-backward algorithm are the forward and backward recursions. For a given input-output sequence pair $(\mathbf{x}, \mathbf{y})$, using these recursions, quantities $\alpha_t(\mathbf{y}_{t-1:t})$ and $\beta_t(\mathbf{y}_{t-1:t})$ can be determined. By properly combining these quantities, the marginal posteriors can be computed. The forward recursion is given by

$$\begin{aligned}\alpha_t(\mathbf{y}_{t-1:t}) =& \Phi_t(y_t; \mathbf{x})\Phi_t(\mathbf{y}_{t-1:t}; \mathbf{x}) \times \\ & \sum_{y_{t-2}} \Phi_t(\mathbf{y}_{t-2:t}; \mathbf{x})\alpha_{t-1}(\mathbf{y}_{t-2:t-1}),\end{aligned} \tag{5}$$

where for $t = 2$, the recursion is initialized as

$$\alpha_2(\mathbf{y}_{1:2}) = \Phi_2(y_2; \mathbf{x})\Phi_1(\mathbf{y}_{1:2}; \mathbf{x})\Phi_1(y_1; \mathbf{x}). \tag{6}$$

The backward recursion is similarly defined [3, 4]. The normalization constant can easily computed as

$$Z(\mathbf{x}) = \sum_{\mathbf{y}_{T-1:T}} \alpha_T(\mathbf{y}_{T-1:T}). \tag{7}$$

The most probable label sequence is found by Viterbi algorithm generalized to HO-LC-CRFs. For details and for time complexities, we refer to [3, 4].

3.3. Pre-Training of the m-n-gram MLP Features

First, we pre-trained MLP networks which represent the probabilities of the corresponding label sub-sequences. So, we optimized the conditional likelihoods, i.e.

$$p^{\text{MLP}}(\mathbf{y}_{t-n+1:t}|t, m, \mathbf{x}) = \frac{\exp\left(\mathbf{w}_n^T \mathbf{f}^{m-n}(\mathbf{y}_{t-n+1:t}; t, \mathbf{x})\right)}{Z^{\text{MLP}}(\mathbf{x})} \tag{8}$$

where

$$Z^{\text{MLP}}(\mathbf{x}) = \sum_{\mathbf{y}_{t-n+1:t}} \exp\left(\mathbf{w}_n^T \mathbf{f}^{m-n}(\mathbf{y}_{t-n+1:t}; t, \mathbf{x})\right) \tag{9}$$

is the normalization constant. Back-propagation is used to compute the partial derivatives for gradient ascent. The non-linear activations of the MLPs are either rectifier units or tanh units.

4. Experiments

We evaluated the performance of the proposed models on the TIMIT phoneme classification task. We compared isolated phone classification (without information on previous labels) with MLP networks to phone labeling with neural HO-LC-CRFs. This comparison substantiates the effectiveness of neural higher-order factors.

4.1. TIMIT Data Set

The TIMIT data set [23] contains recordings of 5.4 hours of English speech from 8 major dialect regions of the United States. The recordings were manually segmented at phone level. We use this segmentation for phone classification. Note that phone classification should not be confused with phone recognition [21] where no segmentation is provided. We collapsed the original 61 phones into 39 phones. All frames of MFCC, delta and double-delta coefficients of a phonetic segment are mapped into one feature vector. Details on pre-processing and data set are presented in [24]. The task is, given an utterance and a corresponding segmentation, to infer the phoneme within every segment. The development set is used for parameter tuning. The performance measure is the phone error rate (PER) in [%].

4.2. Experimental Setup

In all experiments, input features were normalized to zero mean and unit standard deviation. Optimization of our models was in all cases performed using stochastic gradient ascent using a batch-size of one sample. An ℓ_2-norm regularizer on the model weights was used. We utilized early stopping determined on the development data set.

4.3. Labeling Results Using Only MLP Networks

In the first experiment, we trained MLP networks with a single hidden layer to predict the phone label of each segment. We tuned the number of hidden units $H \in \{100, 150, 200, 300, 400, 500\}$ and their activation functions (rectifier, tanh). Furthermore, we analyzed the effect of the

Table 1: Isolated Phone Classification using MLP networks ($n = 1$) with different number of hidden units H and lengths of the contextual input window m.

		rectifier		tanh	
H	m	dev	test	dev	test
150	1	22.6	23.0	22.6	23.0
150	3	21.4	22.2	21.8	22.3
150	5	22.4	22.9	23.2	23.9
200	1	22.5	23.2	22.4	22.6
200	3	21.3	**21.8**	21.4	22.6
200	5	22.3	22.7	22.7	22.9
500	1	22.1	22.6	22.1	22.9
500	3	**20.9**	22.1	**20.6**	**21.0**
500	5	22.3	22.7	21.9	22.7

number of input segments, i.e. we used the current segment, three or five segments centered at the current position index as input. Results in Table 1 (only a sub-set is reported) show that more hidden units result in better performance. For tanh activations, the best performance of 21.0% is achieved with $m = 3$ number of input segments and using $H = 500$ neurons. Larger number of input segments reduces the performance. In preliminary experiments, we found that more than one hidden layer decreased the performance.

4.4. Labeling Results Using LC-CRFs With Linear or Neural Higher-Order Factors

Experiments with linear HO-LC-CRFs as shown in Table 2 indicate that classification performance degrade with linear 3-3 gram factors.

Table 2: Linear higher-order CRFs. All results with $m = 1$ and $n = 1$ already include input-independent 2-gram factors.

m=n	1	+2	+3
dev	25.8	**20.4**	20.7
test	25.9	**20.5**	21.6

In the next set of experiments, we consider LC-CRFs with neural input-dependent higher-order factors and will show their effectiveness in contrast to their linear counterparts. As described in Section 3.3, we discriminatively pre-trained these higher-order MLP factors and used the activations of the last hidden layer of MLP networks as features to train the HO-LC-CRFs. In Table 3, we explore the combination of higher-order factors up to the order $n = m = 3$ as described in Section 3. The plus sign indicates additional higher-order factors on top to the ones from previous line in the table. In addition, we use different number of hidden units H and different activation functions. For factors with $m = 1$ and $n = 1$, the performance slightly increases when using more and more neurons and is best when using 200 neurons. The best overall result of 15.8% is achieved with 150 neurons and factors up to order $n = m = 3$.

Table 3: Combination of factors of different order using tanh and rectifier activations. All results with $m = n = 1$ already include input-independent 2-gram factors.

				rectifier		tanh	
	H	m	n	dev	test	dev	test
	150	1	1	19.8	20.5	20.2	21.2
+	150	2	2	16.4	16.7	16.4	16.5
+	150	3	3	**15.9**	**16.4**	**15.5**	**15.8**
	200	1	1	19.7	20.3	20.0	20.7
+	200	2	2	16.5	17.2	16.3	16.7
+	200	3	3	15.9	16.7	15.5	16.0

Table 4: Summary of labeling results. Results marked by (†) are from [25], by (††) are from [26], by (†††) are from [24] by (††††) are from [12], and by (*) are from [27].

Model	PER [%]
GMMs ML††	25.9
HCRFs†	21.5
Large-Margin GMM††	21.1
Heterogeneous Measurements†††	21.0
CNF	20.67
Linear HO-LC-CRF	20.5
GMM+LC-CRF (1st order)††††	22.10
CS-DCRF+MEMM (8th order)††††	22.15
CS-DCRF+LC-CRF (1st order)††††	19.95
Hierarchical Large-Margin GMMs*	16.7
Proposed model in this paper	**15.8**

To the best of our knowledge, deep scattering spectrum [28], a recently published preprocessing, in combination with support vector machines achieved state-of-the-art performance of 15.9%. Still, we achieved better performance of 15.8%, although 0.1% absolute improvement is statistically not significant. However, because of different pre-processing the results are not directly comparable. We are confident that deep scattering spectrum pre-processing will boost our model. We leave this to future work. Finally, we compare our best result to other state-of-the-art methods based on MFCC features as shown in Table 4. Using the software of [9] we tested CNFs with 50, 100 and 200 hidden units as well as one and three input segments. We achieved the best result with 100 hidden units and one segment as input. Furthermore, hierarchical large-margin GMMs achieve a performance of 16.7% and outperform most other referenced methods but exploit human-knowledge and committee techniques. However, our best model, the HO-LC-CRF augmented by m-n-gram MLP factors achieves a performance of 15.8% and outperforms the other state-of-the-art methods.

5. Conclusion

We considered LC-CRFs with higher-order input-dependent factors for sequence labeling and achieved a phoneme error rate of 15.8% on the TIMIT phoneme classification task. To the best of our knowledge, this is the best performance ever reported. The used higher-order factors were modeled using MLP networks that map sub-sequences of inputs to sub-sequences of outputs. Future work includes joint training of the neural higher-order factors and LC-CRF as well as testing of different types of neural networks.

6. Acknowledgments

This work was supported by the Austrian Science Fund (FWF) under the project number P25244-N15. Furthermore, we acknowledge NVIDIA for providing GPU computing resources.

7. References

[1] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *International Conference on Machine Learning (ICML)*, 2001, pp. 282–289.

[2] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden conditional random fields for phone classification," in *Interspeech*, 2005, pp. 1117–1120.

[3] N. Ye, W. S. Lee, H. L. Chieu, and D. Wu, "Conditional random fields with high-order features for sequence labeling," in *Neural Information Processing Systems (NIPS)*, 2009, pp. 2196–2204.

[4] X. Qian, X. Jiang, Q. Zhang, X. Huang, and L. Wu, "Sparse higher order conditional random fields for improved sequence labeling," in *International Conference on Machine Learning (ICML)*, 2009, pp. 849–856.

[5] T. Lavergne, A. Allauzen, J. M. Crego, and F. Yvon, "From n-gram-based to CRF-based Translation Models," in *Workshop on Statistical Machine Translation*, 2011, pp. 542–553.

[6] L. Stewart, X. He, and R. S. Zemel, "Learning flexible features for conditional random fields." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 8, pp. 1415–1426, 2008.

[7] J. Lafferty, X. Zhu, and Y. Liu, "Kernel conditional random fields: Representation and clique selection," in *International Conference on Machine Learning (ICML)*, 2004, pp. 64–.

[8] H. Larochelle and Y. Bengio, "Classification using discriminative restricted Boltzmann machines," in *International Conference on Machine Learning (ICML)*, 2008, pp. 536–543.

[9] J. Peng, L. Bo, and J. Xu, "Conditional neural fields," in *Neural Information Processing Systems (NIPS)*, 2009, pp. 1419–1427.

[10] R. Prabhavalkar and E. Fosler-Lussier, "Backpropagation training for multilayer conditional random field based phone recognition." in *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 5534–5537.

[11] L. van der Maaten, M. Welling, and L. K. Saul, "Hidden-unit conditional random fields." in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011, pp. 479–488.

[12] M. Ratajczak, S. Tschiatschek, and F. Pernkopf, "Sum-product networks for structured prediction: Context-specific deep conditional random fields," in *International Conference on Machine Learning (ICML) Workshop on Learning Tractable Probabilistic Models Workshop*, 2014.

[13] T. M. T. Do and T. Artières, "Neural conditional random fields," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, pp. 177–184.

[14] P. Xu and R. Sarikaya, "Convolutional neural network based triangular crf for joint intent detection and slot filling." in *ASRU*. IEEE, 2013, pp. 78–83. [Online]. Available: http://dblp.uni-trier.de/db/conf/asru/asru2013.html#XuS13

[15] H. Poon and P. Domingos, "Sum-product networks: A new deep architecture," in *Uncertainty in Artificial Intelligence (UAI)*, 2011, pp. 337–346.

[16] R. Gens and P. Domingos, "Learning the structure of sum-product networks," in *International Conference on Machine Learning (ICML)*, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. JMLR Workshop and Conference Proceedings, May 2013, pp. 873–880.

[17] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 860–867.

[18] X. He, R. S. Zemel, and M. A. Carreira-Perpiñán, "Multiscale conditional random fields for image labeling," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004, pp. 695–703.

[19] Y. Li, D. Tarlow, and R. S. Zemel, "Exploring compositional high order pattern potentials for structured output learning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 49–56.

[20] G. E. Hinton, "Products of experts," in *International Conference on Artificial Neural Networks (ICANN)*, 1999, pp. 1–6.

[21] 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," *Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[22] B. Schuppler, "Automatic analysis of acoustic reduction in spontaneous speech," Ph.D. dissertation, PhD Thesis, Radboud University Nijmegen, Netherlands, 2011.

[23] V. Zue, S. Seneff, and J. R. Glass, "Speech database development at MIT: Timit and beyond," *Speech Communication*, vol. 9, no. 4, pp. 351–356, 1990.

[24] A. K. Halberstadt and J. R. Glass, "Heterogeneous acoustic measurements for phonetic classification," in *EUROSPEECH*, 1997, pp. 401–404.

[25] Y.-H. Sung, C. Boulis, C. Manning, and D. Jurafsky, "Regularization, adaptation, and non-independent features improve hidden conditional random fields for phone classification," in *Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2007, pp. 347–352.

[26] F. Sha and L. Saul, "Large margin Gaussian mixture modeling for phonetic classification and recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006, pp. 265–268.

[27] H.-A. Chang and J. R. Glass, "Hierarchical large-margin Gaussian mixture models for phonetic classification," in *Workshop on Automatic Speech Recognition Understanding (ASRU)*, 2007, pp. 272–277.

[28] J. Andn and S. Mallat, "Deep scattering spectrum," *CoRR*, vol. abs/1304.6763, 2013.