2011 International Conference on Document Analysis and Recognition

Joint Optimization of Hidden Conditional Random Fields and Non Linear Feature Extraction

Antoine Vinel
Université Pierre et Marie Curie
LIP6
Paris, France

Trinh Minh Tri Do
IDIAP
Marigny, Switzerland

Thierry Artières
Université Pierre et Marie Curie
LIP6
Paris, France

***Abstract*—We describe an hybrid model that combines deep neural networks (DNN) for nonlinear feature extraction and hidden conditional random fields (HCRF), i.e. conditional random fields with hidden states. The model is globally trained though joint optimization of HCRF and DNN parameters. To deal with this highly non convex optimization criterion, we propose a multi-step training which aims at providing a good initialization before the final joint optimization of all parameters. We investigate then the discriminative power of these models with respect to the architecture of the DNN, and compare our models to HMM and HCRF based algorithms on the IAM database.**

***Keywords*-Deep Neural Networks; Conditional Random Fields; Handwriting recognition;**

I. Introduction

Hidden Markov Models (HMM) have been commonly used to deal with sequential data classification and labeling. HMM are generative models : they define a joint probability distribution on the sequence of observations and the hidden states and are traditionally trained through likelihood maximization which is a non-discriminant criterion. Some works tried to overcome this limitation by learning discriminatively HMM systems through the optimization of discriminant criterion like minimum error classification [1], perceptron [2], maximum mutual information [3] or more recently large margin [4], [5].

Alternatively, an other way to reach higher discriminative power (than HMMs), and a more straightforward one, is to define a model of the posterior conditional probability $p(\mathbf{y}|\mathbf{x})$ where $\mathbf{y} = (y_1, y_2, ..., y_T)$ is the sequence of labels and $\mathbf{x} = (\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_T)$ is the multi-dimensional observable sequence. We focus here on such models : Conditional Random Fields (CRF) [6] and Hidden-CRF (HCRF), a variant which makes use of hidden states to account for the underlying structure of the data (alike HMMs). These models have been used for various signal labeling tasks like speech [7], handwriting [8] or gesture [9], [10] recognition.

First works have focused on linear CRFs exploiting raw features through linear energy function (see section II-A), they have been applied successfully to textual and biological sequences [6], [11], [12]. Yet, these models frequently reach lower accuracies than those using non-linear transformations like kernels [13]. Though [14] showed the feasibility of using a kernelized version of CRF, this extension is not tractable in practice. Some authors chose to bypass this difficulty by using nonlinear transformations of features. For instance a popular strategy has been to use a polynomial extension (of degree 2) of raw features, which allows defining a class of models that include HMMs as a special case [7],[15].

Besides, recent works on deep neural networks (DNNs) training have shown that these models may be trained efficiently on complex data. Indeed, Hinton [16] proposed a new learning algorithm for DNNs that gave a new breath to this domain. It consists in a greedy layer-wise unsupervised pretraining step, where hidden layers are successively learned to extract relevant features from the previous one.

Most of previous works on discriminative tasks using deep networks focused on fixed-sized classification problems (such as [17] or [18]). NNs have already been used for structured prediction tasks. For example, the generic *graph transformer networks* [19] use convolutional networks for check reading. Collobert [20] proposed a multi-tasks framework for natural language processing tasks where a lookup table learns a representation of data before using a convolutional network.

We proposed in [21] a combination of CRF and of DNNs that aimed at combining the pros of the deeps networks as nonlinear feature extractor and a discriminant model for sequences in one unique framework that could be optimized jointly. We build upon this work and propose the combination of DNNs and of Hidden CRFs, allowing a class model to use hidden states instead of using one state per class. Since the training criterion is highly non convex, we propose a three step learning that allows progressively reaching high accuracy solutions. We compare our models to HMM-based algorithms and to more classical HCRF based systems using the polynomial expansion as in [7]. Recently some authors explored too the combination of NNs and CRFs for signal labeling tasks ([22], [23], [24]). Our work differs from them in several points. First, unlike [22], we optimize the DNN and the discriminative model jointly. Unlike [23], we use models with more than one hidden

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.109

layer, and pre-trained them. Second, we exploit hidden states which significantly increase accuracy on real-valued and noisy data (e.g. speech or handwriting recognition) as we will show. Unlike [24] who report results on small datasets only, we report experimental results on a real sized problem.

II. Background

A. Conditional Random Fields

CRFs are a framework for building probabilistic graphical models where independence assumptions between variables are encoded in a Makovian network. While they can be designed with various graph architectures, in the following we restrict ourselves to "chain-structured CRF" (see 1) for dealing with sequence data. The posterior probability of a CRF over a set of Markov network of variables may be defined as the product of functions defined on cliques of the network. In "chain-structured CRF", we can distinguish two kind of cliques at each time t:

- a clique connecting the observation x_t to its corresponding label y_t
- a clique connecting tow successive labels, y_{t-1} and y_t

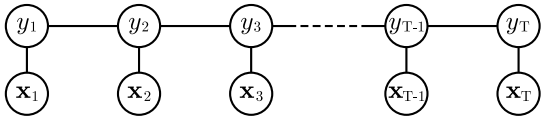

Figure 1. Graphical representation of a CRF on a T-length sequence

Formally, we define two energy equations to model such local and transition relationships:

- $\mathrm{E}_{\mathrm{loc}}(\mathbf{x}, t, y_t, \mathbf{W}) = \langle \mathbf{w}_{y_t} \mathbf{x}_t \rangle$
- $\mathrm{E}_{\mathrm{tra}}(\mathbf{x}, t, y_{t-1}, y_t, \mathbf{\Lambda}) = \lambda_{y_{t-1}, y_t}$

where $\mathbf{W} = \{\mathbf{w}_y | y \in \mathrm{L}\}$ and $\mathbf{\Lambda} = \{\lambda_{y_1, y_2} | (y_1, y_2) \in L^2\}$ stand for the set of local-state and of state-state parameters to be learned (with L denoting the set of possible labels).

A CRF defines the following conditional probability:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \mathbf{\Lambda}) = \frac{e^{\sum_t \mathrm{E}_{\mathrm{loc}}(\mathbf{x}, t, y_t, \mathbf{W}) + \sum_{t>1} \mathrm{E}_{\mathrm{tra}}(\mathbf{x}, t, y_{t-1}, y_t, \mathbf{\Lambda})}}{Z(X)}$$

where $Z(\mathbf{x})$ is a normalization term : the sum over all possible label sequences of the numerator (this makes $\sum_y p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \mathbf{\Lambda}) = 1$). $Z(\mathbf{x})$ can be efficiently computed via dynamic programming.

B. Hidden Conditional Random Fields

When applying CRF to signals such as speech and handwriting a natural idea is to make use of hidden states to account for the underlying structure of the data, alike in HMMs. Hence, to capture the different successive stages in the signal corresponding to a particular class (e.g. character, digit, phoneme, gestures, ...), one can affect a disjoint set of hidden states to each label and allow transitions between those states as proposed in [10].

One can furthermore constraint such a model to allow only few transitions between the hidden states. As is classically done in HMMs for speech and handwriting we consider transitions constraints that lead to a "Left-Right" model.
A HCRF defines the following conditional probability :

$$p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \mathbf{\Lambda}) = \sum_{\mathbf{h} \in s(\mathbf{y})} p(\mathbf{h}|\mathbf{x}, \mathbf{W}, \mathbf{\Lambda}) \quad (1)$$

where $\mathbf{h}$ represents a sequence of hidden states (i.e. a *segmentation*) and $s(\mathbf{y})$ all possible *segmentation* corresponding to a label sequence $\mathbf{y}$.

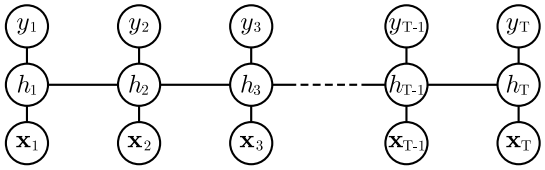

Figure 2. Graphical representation of a HCRF

Unfortunately, introducing hidden states in a CRF makes the optimization (maximizing conditional likelihood) a non-convex problem. A good initialization is then essential to avoid poor learning. Training is usually performed with gradient based algorithms such as SGD or LBFGS.

C. Deep Neural Networks

Deep neural networks have generated renewed interest in recent years as a consequence of the proposition in parallel of a few efficient learning schemes for deep architectures. For instance, Hinton proposed an algorithm for learning deep architectures by pre-training each layer one by one from the bottom one (raw data) to the top one in an unsupervised fashion [16]. The main objective of this pre-training step is to provide a "high-level" data representation as a byproduct of a good (unsupervised) modeling of $\mathbf{x}$. Such a greedy algorithm is shared by a few approaches and relies on a building block for learning each of the hidden layer. Two such building blocks are used, either Restricted Boltzmann Machines (RBM) or Auto-Encoder networks. We focus here on RBM which we used on our experiments. A RBM is an energy-based model that defines a joint-distribution between a "visible" layer $\mathbf{v}$ corresponding to input data and a "hidden" layer (named $\mathbf{c}$ for "code"). In a RBM, one assumes that all units of a layer are conditionally independent. This assumption significantly simplifies sampling as required during training. A RBM defines a probability distribution over visible and hidden units through the definition of an energy function as:

$$p(\mathbf{v}, \mathbf{c}) = \frac{e^{\mathrm{energy}(\mathbf{v}, \mathbf{c})}}{Z}$$

where Z is a normalization term. A popular choice for the energy function (for binary visible units) is the following:

$$\mathrm{energy}(\mathbf{v}, \mathbf{c}) = -\mathbf{v}^\top \tilde{\mathbf{v}} - \mathbf{c}^\top \mathbf{M} \mathbf{v} - \mathbf{c}^\top \tilde{\mathbf{c}}$$

where $\tilde{\mathbf{v}}$ (resp. $\tilde{\mathbf{c}}$) is a bias on the visible (resp. hidden) units, and $\mathbf{M}$ stands for the weights between visible and hidden

units. The normalization term (called "partition function") is unfortunately expensive to compute (may be intractable on real problems) but the gradient of the maximum likelihood objective can be approximated efficiently with the so-called contrastive divergence algorithm [25], making training tractable. Contrastive divergence relies on Gibbs sampling using an interesting feature of RBM: the partition function term is not required to sample $\mathbf{c}$ from $\mathbf{v}$ and vice versa.

III. Mixing DNNs and HCRFs

A. Model

A NeuroHCRF is the combination of a deep neural network and a HCRF. Let us denote:

- $\mathbf{\Theta}$ all the parameters of the DNN
  (i.e. for of each neuron, its bias and the weight of all incoming synaptic connections)
- $\boldsymbol{\phi}(\mathbf{x}_t, \mathbf{\Theta})$ the output of the network computed with $\mathbf{x}_t$ as input.

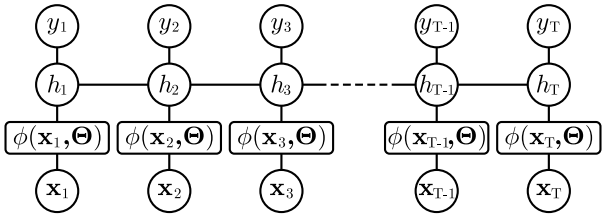

Figure 3. Graphical representation of a NeuroHCRF

A NeuroHCRF extends a NeuroCRF in [21] and defines a conditional probability as:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \mathbf{\Lambda}, \mathbf{\Theta}) = \sum_{\mathbf{h} \in s(\mathbf{y})} p(\mathbf{h}|\boldsymbol{\phi}(\mathbf{x}, \mathbf{\Theta}), \mathbf{W}, \mathbf{\Lambda}) \quad (2)$$

where the conditional probability is computed by summing over all *hidden state segmentation* matching the label sequence $\mathbf{y}$.

B. Training

Training aims at learning all parameters $\mathbf{\Theta}$, $\mathbf{W}$ and $\mathbf{\Lambda}$ to jointly optimize the conditional likelihood criterion. This optimization is difficult since the training criterion is highly non convex. First training DNN weights is a highly non convex problem in itself (this is the reason why DNNs have not been much used for a long time till recent works such as [16]). Second, although learning a linear CRF leads to a convex optimization problem, introducing hidden variables makes the criterion becoming non-convex. Hence training jointly all the parameters of a NeuroHCRF requires a good initialization. To achieve this we investigated a three steps learning scheme that we discuss now.

1) Pre-training the DNN: We first pre-trained a neural network on each frame of the data in an unsupervised way just as is done in [16]. We mainly used deep belief nets, which building block is a RBM, since Stacked Auto Encoders did not allow reaching as good results in preliminary experiments. In the following we will use $\mathbf{\Theta}_0$ to denote this initial set of parameters of the DNN.

2) Initializing the HCRF: Once the DNN is pre-trained, it may be used as a nonlinear feature extractor. We proceed by computing the activation of the last layer of the DNN for each frame of training data. We then get a transformed training set (where all frames have the dimension of the last layer of the DNN). We use it to train the HCRF though the maximization of the conditional likelihood of the training data. This is done using an own optimization procedure based on bundle method (see [26]). Furthermore, to reach an interesting solution, this training is performed in two sub-steps. First we fix the hidden state sequence for a few iterations (according to linear alignment), then freeing the hidden state sequence. Overall this initialization of the HCRF consists in a learning of the NeuroHCRF model, where the weights of the DNN (i.e. $\mathbf{\Theta}$) remain fixed to the value found in the pre-training step of the DNN, $\mathbf{\Theta}_0$.

3) Fine tuning: The last step of the training algorithm is a joint optimization of all the parameters, starting with an initial solution given by above pre-training steps. The last training step consists in minimizing the minus conditional likelihood and an additional quadratic regularization term Ω (around the initial solution for the DNN's weights) in order to avoid over-fitting:

$$\Omega(\mathbf{W}, \mathbf{\Lambda}, \mathbf{\Theta}) = \frac{1}{2}\left[\|\mathbf{\Theta} - \mathbf{\Theta}_0\|^2 + \|\mathbf{W}\|^2 + \|\mathbf{\Lambda}\|^2\right]$$

The gradient of this criterion may be easily back-propagated to compute the gradient of the criterion with respect to HCRF parameters and to the whole DNN parameters. As a final step we perform a rescaling procedure between transitions scores local scores in order to achieve a better accuracy on the development dataset. Such a procedure is classically used in speech recognition to tune HMM parameters to balance insertion or deletion errors (see IV).

C. Inference

Inference in a NeuroHCRF consists in finding the labeling $\widehat{\mathbf{y}}$ which best fits the data sequence $\mathbf{x}$.

$$\widehat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \mathbf{\Lambda}, \mathbf{\Theta})$$

$$\widehat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \sum_{\mathbf{h} \in \mathbf{s}(\mathbf{y})} e^{\sum_t \mathrm{E}_{\mathrm{loc}}(\mathbf{x}, t, h_t) + \sum_{t>1} \mathrm{E}_{\mathrm{tra}}(\mathbf{x}, t, h_{t-1}, h_t)}$$

To do this, we first transform the input sequence, frame by frame, using the DNN . Then we perform a classical dynamic programming procedure to get $\widehat{\mathbf{y}}$.

IV. Experiments

We performed off-line cursive handwritten word recognition experiments on the IAM database [27] preprocessed by computing nine geometrical features on a sliding window as described in [28] who kindly provided us their preprocessed version of the database. We enriched the input of our model by adding contextual information consisting in the two previous and next frames. The size of the input layer of the

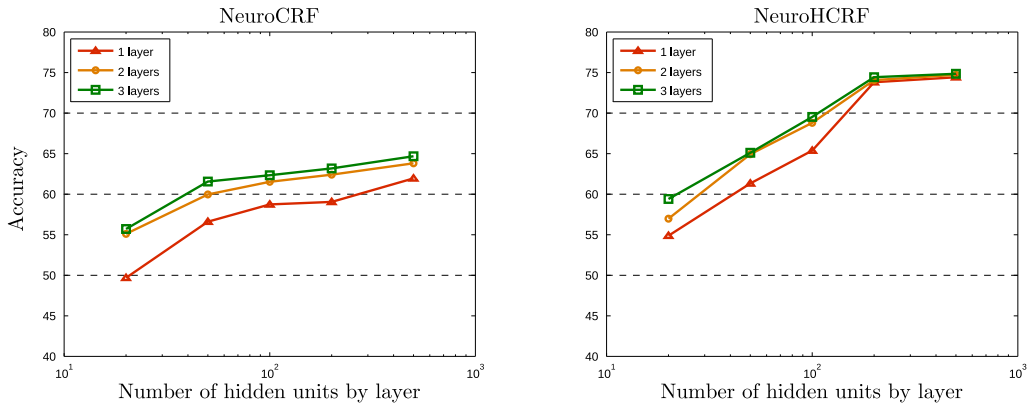

Figure 4. Accuracies of NeuroCRF and NeuroHCRF for models using one to three layers, whose size ranges from 20 to 500 hidden units.

DNN was then 45 units corresponding to x_t, x_{t-1}, x_{t-2}, x_{t+1}, x_{t+2}. We used only medium sized training dataset of 10K words (about 46K characters) to explore various architectures from 20 to 500 units per hidden layer, and from one to three hidden layers. Each character model had five internal hidden states organized in a "Left-Right" way.

We investigated the performance of NeuroHCRF and of NeuroCRF (as proposed in [21]), i.e. single state NeuroHCRF. We compared character error-rates which are computed from the computation of the edit distance between the predicted word and the true word. The edit distance produces four values : the number of "hits" H, of "deletion" D, of "insertion" I, and of "substitution" S.

Reported results consist in character recognition accuracy computed on a 10K words test dataset as follow.

$$\text{accuracy} = \frac{H - I}{H + D + S}$$

We report in Figure 4 comparative results of NeuroCRFs and of NeuroHCRFs with various architectures. In both cases, one can see that the performance can be improved by using larger hidden layers and by adding hidden layers. Also one can see that increasing the hidden layer size and number jointly may lead to increased performance although this seems more true for NeuroCRF than for NeuroHCRF whose performance reaches an upper bound. Finally one see from the two figures that NeuroHCRF consistently outperform corresponding NeuroCRF with the same DNN architecture and that the best NeuroHCRF architecture allows reaching a very interesting performance of about 75% accuracy.

Model		20	50	100	200	500
NeuroCRF	1 layer	49.69	56.61	58.74	59.05	61.95
	2 layers	55.11	59.97	61.53	62.41	63.81
	3 layers	55.71	61.56	62.34	63.18	64.68
NeuroHCRF	1 layer	54.88	61.33	65.38	73.81	74.42
	2 layers	56.98	64.93	68.8	74.07	74.72
	3 layers	59.41	65.11	69.53	74.43	74.85

Table I
ACCURACIES OF NEUROCRF AND NEUROHCRF FOR MODELS FROM 20 TO 500 HIDDEN UNITS IN ONE TO THREE LAYERS.

Models	Specifications	Accuracy
HMM	8 states	51.2
	14 states	59.6
MaxMargin HMM	8 states	70.7
	14 states	70.3
CRF	*raw data*	27.1
	second order data	54.1
NeuroCRF	2 layers of 50 units	60.0 (*36.6*)
	2 layers of 500 units	63.8 (*54.6*)
HCRF	5 states - *raw data*	65.1
	5 states - *second order data*	70.0
NeuroHCRF	5 states - 2 layers of 50 units	64.9 (*56.7*)
	5 states - 2 layers of 500 units	74.7 (*67.3*)

Table II
ACCURACIES OF SOME MODELS. THE PARENTHESIZED ACCURACIES CORRESPONDS TO THOSE OBSERVED BEFORE FINE-TUNING.

Table I report comparative results of NeuroCRF and NeuroHCRF architectures with various models on the same test dataset. We report some baseline results gained with HMM systems, either generatively trained (HMM rows), or discriminatively trained (MaxMargin rows) with a large margin criterion (i.e. all HMM results come from [21] and have been gained on a more than 3-times bigger training database but with the same test set). In addition we report results of linear CRF and HCRF (with 5 hidden states) working on raw features (*raw data* rows) and on polynomial (degree 2) expansion of raw features (*second order data* rows) as proposed in [7].

We can see in these additional results that despite their discriminative learning, CRF (working with *raw* or *second order* data) do not reach the accuracy of HMMs and are far below those from HCRFs and HMMs learned with a large margin criterion.

This fact show how fundamental is the hidden state modeling scheme, even if it leads to more complex optimization schemes (loosing convexity). Comparative performance of the models presented here show the benefits provided by the DNN for "high-level" feature extraction, even when comparing with HCRF working on second order polynomial expansion of raw features.

V. Conclusion

We investigated an hybrid model blending HCRF and deep neural networks which combines the benefits of both approaches : the ability of deep neural network at extracting high-level features, and the discriminative ability of CRF powered by hidden states at labeling complex sequences.

We reported experimental results on the IAM dataset showing the potential of those deep architectures with respect to state of the art approaches.

Acknowledgment

This work has been supported by the PASCAL2 Network of Excellence program.

References

[1] B. Juang and S. Katagiri, "Discriminative learning for minimum error classification," in *IEEE Trans. Signal Processing, Vol.40, No.12*, 1992.

[2] M. Collins, "Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms," in *EMNLP*, 2002.

[3] P. Woodland and D. Povey, "Large scale discriminative training of hidden markov models for speech recognition," *Computer Speech and Language*, 2002.

[4] F. Sha and L. K. Saul, "Large margin hidden markov models for automatic speech recognition," in *NIPS 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. MIT Press, 2007.

[5] T.-M.-T. Do and T. Artières, "Large margin training for hidden Markov models with partially observed states," in *ICML*, 2009.

[6] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Machine Learning-Ineternational Workshop then conference*, 2001.

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

[8] T.-M.-T. Do and T. Artières, "Conditional Random Fields for Online Handwriting Recognition," in *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.

[9] A. Quattoni, M. Collins, and T. Darrel, "Conditional random fields for object recognition," in *Advances in Neural Information Processing Systems (NIPS 2004)*, no. 17, 2005.

[10] L.-P. Morency, A. Quattoni, and T. Darrell, "Latent-dynamic discriminative models for continuous gesture recognition," in *CVPR*, 2007.

[11] Y. Altun, M. Johnson, and T. Hofmann, "Investigating loss functions and optimization methods for discriminative learning of label sequences," in *Empirical methods in natural language processing*, 2003.

[12] K. Sato and Y. Sakakibara, "Rna secondary structural alignment with conditional random fields," in *ECCB/JBI*, 2005.

[13] B. Taskar, C. Guestrin, and D. Koller, "Max-margin markov networks," in *NIPS 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.

[14] J. Lafferty, X. Zhu, and Y. Liu, "Kernel conditional random fields: representation and clique selection," in *ICML*, 2004.

[15] S. Reiter, B. Schuller, and G. Rigoll, "Hidden conditional random fields for meeting segmentation," in *ICME*, 2007.

[16] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, 2006.

[17] H. Larochelle and Y. Bengio, "Classification using discriminative restricted boltzmann machines," *ICML*, 2008.

[18] T. Schmah, G. Hinton, R. Zemel, S. Small, and S. Strother, "Generative versus discriminative training of rbms for classification of fmri images," in *NIPS*, 2008.

[19] Y. L. L. Bottou, Y. Bengio, "Global training of document processing systems using graph transformer networks," in *Computer Vision and Pattern Recognition*, 1997.

[20] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *ICML*, 2008.

[21] T.-M.-T. Do and T. Artières, "Neural conditional random fields," in *AISTAT*, 2010.

[22] J. Morris and E. Fosler-Lussier, "Conditional random fields for integrating local discriminative classifiers," in *EMNLP*, 2008.

[23] J. Peng, L. Bo, and J. Xu, "Conditional neural fields," *NIPS*, 2009.

[24] J. Liu, K. Yu, Y. Zhang, and Y. Huang, "Training conditional random fields using transfer learning for gesture recognition," *ICDM*, 2010.

[25] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, 2002.

[26] T. Do and T. Artieres, "Maximum margin training of gaussian hmms for handwriting recognition," in *ICDAR*, 2009.

[27] U. Marti and H. Bunke, "A full english sentence database for off-line handwriting recognition," in *ICDAR*, 2002.

[28] U.-V. Marti and H. Bunke, "Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system," *IJPRAI*, 2001.