DEEP-HIDDEN CONDITIONAL NEURAL FIELDS FOR CONTINUOUS PHONEME SPEECH RECOGNITION

Yasuhisa Fujii, Kazumasa Yamamoto, Seiichi Nakagawa*

Toyohashi University of Technology
Department of Computer Science and Engineering
1-1 Hibarigaoka, Tenpaku-cho, Toyohashi, Aichi 441-8580, JAPAN

ABSTRACT

We have proposed Hidden Conditional Neural Fields (HCNF) for automatic speech recognition and shown the effectiveness by continuous phoneme recognition experiments on the TIMIT and the Japanese ASJ+JNAS corpora. In this paper, we propose to use an observation function with a deep structure in HCNF. The proposed deep observation function enables to use the deep neural networks in HCNF, which have recently achieved remarkable success in Automatic Speech Recognition. We call the HCNF with the observation function with a deep structure *Deep-HCNF*. Experimental results of continuous phoneme speech recognition on the TIMIT and the ASJ+JNAS corpora showed that the Deep-HCNFs with monophone structure outperformed traditional tied-state triphone HMMs trained in MPE manner.

Index Terms— hidden conditional neural fields, hidden conditional random fields, HMM, speech recognition

1. INTRODUCTION

Current Automatic Speech Recognition (ASR) systems employ a Hidden Markov Model (HMM) together with a Gaussian Mixture Model (GMM) as the emission probability for an acoustic model. However, the HMM has two major drawbacks to use as an acoustic model. First, it relies on a strong independency assumption whereby frames are independent in a given state, and thus, lacks the ability to deal with features that straddle several frames. Second, it is not suitable for discriminating sequences because the HMM is a generative model. To solve the former problem, features that can deal with phenomena straddling multiple frames have been developed, such as the delta coefficient [1], segmental statistics [2], and modulation spectrum [3]. For the latter problem, discriminative training methods such as the minimum phone error (MPE) have been investigated [4].

We have proposed an ASR method using Hidden Conditional Neural Fields (HCNF) to overcome the above two problems [5, 6]. HCNF can handle features that straddle several frames and has high discriminative power since it is a discriminative model. Experimental results on the TIMIT and Japanese ASJ+JNAS corpora showed the effectiveness of HCNF for ASR in a continuous phoneme recognition experiment [5, 6].

Recently, deep densely connected neural networks have achived remarkable success [7]. The observation function used in previous works [5, 6] had only one gate (hidden) layer and corresponded to a traditional three-layered MLP (input, hidden, output). In this paper, we propose to use an observation function with a deep structure in HCNF. The proposed deep observation function enables to use the deep neural networks in HCNF.

This paper is organized as follows. In the next section, HCNF for ASR is reviewed. After that, the deep observation function for HCNF is introduced in Section 3. The experimental setup and results are shown in Sections 4. Finally, Section 5 presents our conclusions and future works.

2. HIDDEN CONDITIONAL NEURAL FIELDS

2.1. Formulation

Given an observation sequence $X = (x_1, x_2, \dots, x_T)$, HCNF computes a score of a label sequence $Y = (y_1, y_2, \dots, y_T)$ as follows:

$$P(Y|X) = \frac{\sum_S \exp(\Lambda(X, Y, S))}{Z(X)}, \tag{1}$$

where

$$\Lambda(X, Y, S) = \kappa\Phi_n(X, Y, S) + \kappa\Psi_n(X, Y, S), \tag{2}$$

and $Z(X)$ is a partition function computed as

$$Z(X) = \sum_{Y'} \sum_S \exp(\Lambda(X, Y', S)), \tag{3}$$

This work was supported by Global COE Program " Frontiers of Intelligent Sensing " from the Ministry of Education,Culture,Sports, Science and Technology, Japan. *Research Fellow at Japan Society for the Promotion of Science

where $S = (s_1, s_2, \dots, s_T)$ is a hidden variable sequence that represents a state sequence, κ is a state-flattening coefficient [8]. $\Phi_n(X, Y, S)$ and $\Psi_n(X, Y, S)$ are an observation function and a transition function, respectively and are given below.

$$\Phi_n(X, Y, S) = \sum_t \sum_g^K w_{y_t, s_t, g} h(\theta_g^T \phi(X, Y, S, t)), \quad (4)$$

$$\Psi_n(X, Y, S) = \sum_j u_j \sum_t \psi_j(X, Y, S, t, t-1), \quad (5)$$

where $\psi_j(X, Y, S, t, t-1)$ is a transition feature extracted at frame t and $t-1$, and u_j is a corresponding weight. $w_{y,s,g}$ is a weight specific to the triple y, s, and g. $\phi(X, Y, S, t)$ is a vector representation of features such as MFCCs, and θ_g is the corresponding weight vector specific to the gate g [1]. $h(x)$ is a gate function defined as

$$h(x) = \frac{c}{1 + \exp(-\alpha(x - \beta))} - b, \quad (6)$$

where c and d are terms to change the range of values, and α and β are terms to control the shape of the gate function.

2.2. Training

Given training data $D = \{X^i, Y^i\}, i = 0, \dots, N$, the training of HCNF can be formulated as a problem to find $\lambda = \{w_{y,s,g}, \theta_g, u_j\}$ which minimizes the following objective function:

$$\ell_{MMI}(\lambda; D) = - \sum_i \log \frac{\sum_S \exp(\Lambda(X, Y, S))}{\sum_{Y'} \sum_S \exp(\Lambda(X, Y', S))}. \quad (7)$$

where MMI stands for Maximum Mutual Information.

If we can compute the partial derivatives of $\ell_{MMI}(\lambda; D)$, we can utilize arbitrary gradient-based optimization methods to train HCNF. The partial derivatives of $\ell_{MMI}(\lambda; D)$ by $w_{y,s,g}$, θ_g and u_j can be computed as follows:

$$\frac{\partial \ell_{MMI}(\lambda; D)}{\partial w_{y,s,g}} = - \kappa \sum_i E\left[\sum_t h(\theta_g^T \phi(X^i, Y^i, S, t))\right]_{S|X^i, Y^i} + \kappa \sum_i E\left[\sum_t h(\theta_g^T \phi(X^i, Y, S, t))\right]_{Y, S|X^i}, \quad (8)$$

$$\frac{\partial \ell_{MMI}(\lambda; D)}{\partial \theta_g} = - \kappa \sum_i E\left[\sum_t w_{y,s,g} \frac{\partial h(\theta_g^T \phi(X^i, Y^i, S, t))}{\partial \theta_g}\right]_{S|X^i, Y^i} + \kappa \sum_i E\left[\sum_t w_{y,s,g} \frac{\partial h(\theta_g^T \phi(X^i, Y, S, t))}{\partial \theta_g}\right]_{Y, S|X^i}, \quad (9)$$

$$\frac{\partial \ell_{MMI}(\lambda; D)}{\partial u_j} = - \kappa \sum_i E\left[\sum_t \psi_j(X^i, Y^i, S, t, t-1)\right]_{S|Y^i, X^i} + \kappa \sum_i E\left[\sum_t \psi_j(X^i, Y, S, t, t-1)\right]_{Y, S|X^i}. \quad (10)$$

The partial derivative of Eq. (6) can be computed as follows:

$$\frac{dh(x)}{dx} = \frac{\alpha}{c}(b + h(x))(c - b - h(x)). \quad (11)$$

In [6], we proposed a Hidden Boosted-MMI (HB-MMI) as a training criterion of HCNF to consider training errors in a more direct way than posterior maximization even if state sequences are not known (fixed).

$$\ell_{HB-MMI}(\lambda; D) = - \sum_i \log \frac{\sum_S \exp(\Lambda(X, Y, S))}{\sum_{Y'} \sum_S \exp(\Lambda(X, Y, S)) \exp(-b\mathrm{Acc}(S, D^i))}. \quad (12)$$

where $\mathrm{Acc}(S, D^i)$ is the expected correct state count of a state sequence S given a reference $D^i = (X^i, Y^i)$. If $\mathrm{Acc}(S, D^i)$ is regarded as a constant, the partial derivative of (12) can be computed in common with Eqs. (8), (9) and(10).

2.3. Inference

We used Viterbi algorithm to find the most likely sequence of hidden states instead of searching for the most likely output sequence that maximizes Eq. (1).

3. DEEP OBSERVATION FUNCTION

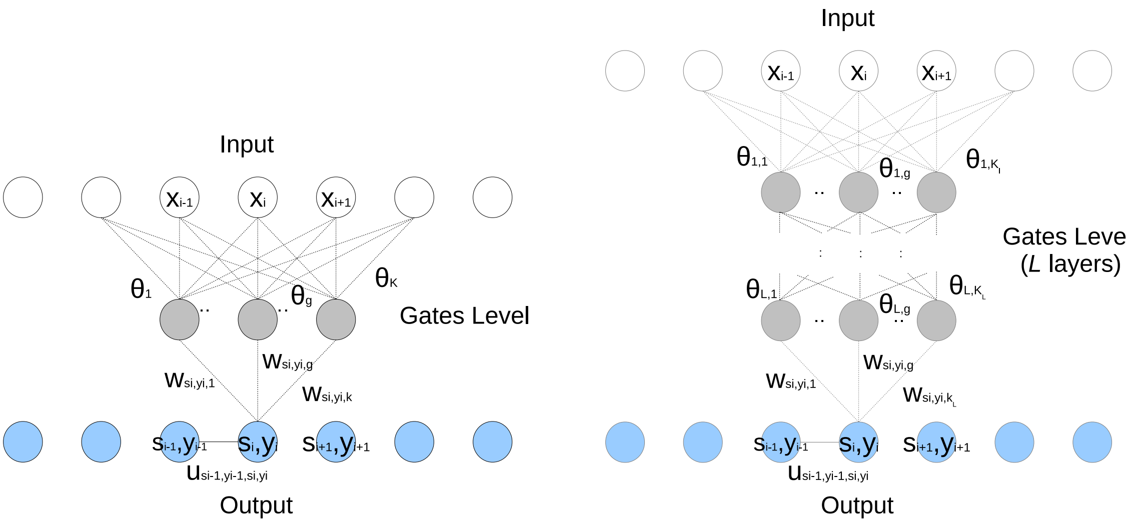
The proposed deep observation function enables us to use the deep neural networks in HCNF. To use the deep observation function in HCNF, the following equation is used instead of Eq. (4) as the observation function:

$$\Phi_n(X, Y, S) = \sum_t \sum_g^{K_L} w_{y_t, s_t, g} G_{L,g}(X, Y, S, t), \quad (13)$$

where L is the number of layers and $G_{l,g}$ is defined recursively as follows:

$$G_{l,g}(X, Y, S, t) = \begin{cases} h(\sum_v^{K_{l-1}} \theta_{l,v} G_{l-1,v}(X, Y, S, t)) & l > 1, \\ h(\theta_{l,g}^T \phi(X, Y, S, t)) & l = 1. \end{cases} \quad (14)$$

[1] In [5, 6], we used label and state specific gate functions since it was robust in our initial experiments where we used only one gate layer. However, with the deep observation function described in Section 3, gate functions which are independent of label and state are more efficient and effective. Therefore, we used label- and state-independent gate functions θ_g instead of $\theta_{y,s,g}$ throughout this paper.



(a) HCNF

(b) Deep-HCNF

Fig. 1. The structure of HCNF and Deep-HCNF.

When $L = 1$, Eq. (13) reduced to Eq. (4). Partial derivatives of Eqs. (7) and (12) with the deep observation function in terms of $\theta_{l,v}$ can be derived similarly to Eq. (9). We call the HCNF with the observation function with a deep structure *Deep-HCNF*. Figure 1 shows the difference between the structures of HCNF and Deep-HCNF.

We found that Deep-HCNF easily suffer overfitting due to the great expressive power, especially when trained on a small corpus like TIMIT. To mitigate the problem, weight averaging conducted in [9] for HCRF was performed for HCNF. We used the following equation for the averaging.

$$\lambda_{\text{ave}} = \frac{\sum_{i=0}^{\#epoch} \lambda_i}{\#epoch} \tag{15}$$

where λ_i means a model parameter after epoch i and λ_0 is a initial parameter. We used this averaging for the experiments on the TIMIT corpus only.

4. EXPERIMENTS

4.1. Setup

We used the TIMIT corpus to examine the effectiveness of our proposed method because it offers a good test bed to study algorithmic improvements [10]. The training set in the TIMIT corpus consists of 3696 utterances by 462 speakers ($\approx 3h$). For the evaluation, we used the core test set consisting of 192 utterances by 24 speakers. A subset of remaining data in the TIMIT corpus, which consists of 400 utterances by 50 speakers, was used as the development set. We also used the ASJ+JNAS corpus[2], which is about 11 times larger than the TIMIT corpus. The training set in the ASJ+JNAS corpus consists of 20337 utterances by 133 speakers ($\approx 33h$). For evaluation, we used the IPA100 test set consisting of 100 utterances by 23 speakers. We extracted 13 MFCC features for the TIMIT corpus. Log energy was used instead of the 0-th MFCC for the ASJ+JNAS corpus. The delta features were not used since they may harm the generalization ability of HCNF [6]. The speech was analyzed using a 25 ms Hamming window with a pre-emphasis coefficient of 0.97 and shifted with a 10 ms fixed frame advance. The acoustic features were normalized to have zero mean and unit variance. For the TIMIT corpus, the 61 TIMIT phonemes were mapped into 48 phonemes for training and further collapsed from 48 phonemes to 39 phonemes for evaluation [11]. For the ASJ+JNAS corpus, 43 Japanese phonemes were used. All phonemes were represented as 3 state left-to-right monophone models. We defined two observation functions as given below:

$$\phi^{M1}_{s,d,f}(X, Y, S, t) = \delta(s_t = s)x_{d,t+f}, \tag{16}$$

$$\phi^{M2}_{s,d,f}(X, Y, S, t) = \delta(s_t = s)x^2_{d,t+f}, \tag{17}$$

where $x_{d,t}$ is the $d-$th component of x_t and f is used to consider the surrounding frames. In the experiments, we used $-7 \leq f \leq 7$, which means we used the number of observations in the fifteen frames centered at the current frame. Moreover, we defined the following transition functions:

$$\phi^{Occ}_{s}(X, Y, S, t) = \delta(s_t = s), \tag{18}$$

$$\phi^{Uni}_{y}(X, y, s, t) = \delta(y_t = y), \tag{19}$$

$$\psi^{Tr}_{s,s'}(X, Y, S, t, t-1) = \delta(s_t = s)\delta(s_{t-1} = s'), \tag{20}$$

$$\psi^{Bi}_{y,y'}(X, y, y', s, s', t, t-1) = \delta(y_t = y)\delta(y_{t-1} = y'). \tag{21}$$

[2] http://www.mibel.cs.tsukuba.ac.jp/_090624/jnas/instruct.html

All parameters of the HCNF were randomly initialized between -0.5 and 0.5. The state-flattening parameter κ was set to 0.1. The parameters of the gate function were set to $\alpha = 0.1$, $\beta = 0.0$, $c = 6.0$, and $d = 3.0$. The parameters for the HCNF were trained by 50 epochs for the TIMIT corpus and 15 epochs for the ASJ+JNAS corpus by Stochastic Gradient Descent without regularization.

For comparison, we prepared tied-state triphone HMMs to obtain the best results with HMMs. For the TIMIT experiment, the HMMs consisted of 792 states which had a 8 mixture GMM with diagonal covariance matrices and while for the ASJ+JNAS experiment, they consisted of 2087 states which had a 32 mixture GMM with diagonal covariance matrices. As a language model, a bigram phone language model was trained from the training corpora.

4.2. Results

4.2.1. Investigation on the number of layers

Figure 2 shows the experimental results with the deep observation function on the TIMIT corpus. In this experiment, we used 512 gates for each layer. As the number of layers increases, we can observe that the Phoneme Error Rates (PERs) decrease in all sets. This result indicates that having deep structure in observation function increases the expressive ability of the model while preserving the generalization ability. By increasing the number of layers, the size of the parameters also increases. In the case of Figure 2, the size of the parameters increased from 278960 (#layer=1) to 2642864 (#layer=10). To confirm that the improvement was not caused by only the increment of the parameter size, we conducted an additional experiment where the number of gates is increased while the number of layers is fixed. The results are shown in Table 1. As can be seen in the table, increasing the number of gates, thereby increasing the number of parameters did not improve the PERs so much. From this result, we can conclude that the improvements of the PERs were not due to the increase of the parameter size but due to adding more layers.

4.2.2. Comparison with triphone HMMs

We compared Deep-HCNFs to the triphone-HMMs. In this experiment, a layer size of 10 and a gate size of 512 were used for the experiment on the TIMIT corpus and a layer size of 5 and a gate size of 1024 were used for the experiment on the ASJ+JNAS corpus, respectively. Table 2 shows the result on the TIMIT corpus. On the corpus, the Deep-HCNFs outperformed triphone-HMMs regardless of the use of HB-MMI. HB-MMI provided an additional improvement of PER and yielded the result of PER=24.3. Table 3 shows the result on the ASJ+JNAS corpus. With HB-MMI, Deep-HCNF was superior to the result of triphone-HMM trained in MPE manner even though it employed the monophone structure. These results confirm the effectiveness of the deep observation function in HCNF.

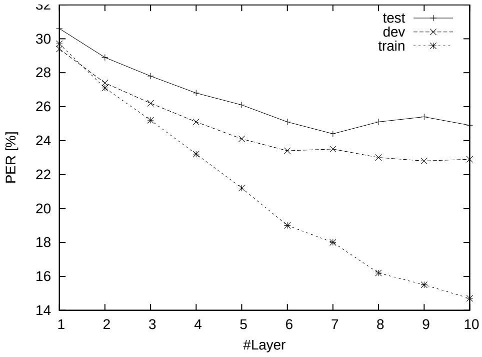


Fig. 2. *Phoneme recognition results with deep observation function on the TIMIT corpus.*

Table 1. *Phoneme recognition results with single layer observation function on the TIMIT corpus [%]*

#Gate	PER	Train-PER	#Param
1000	32.7	31.8	383040
2000	32.3	30.9	761040
4000	31.7	30.3	1895040

5. CONCLUSION

In this paper, we proposed Deep-HCNF which utilized an observation function with deep structure. The Deep-HCNFs with monophone structure was comparable to the MPE-trained tied-state triphones. We achieved the best result PER=24.3 on the TIMIT core test set and PER=12.1 on the IPA100 test set, respectively. The result on the TIMIT core test was much worse than recently reported best results where they achieved PERs under 20% [12, 13]. However, Deep-HCNF use a deep feedforward neural network in the observation function, and therefore, we can use a sophisticated pre-training algorithm such as DBN to provide the deep observation function with reasonable initialization in Deep-HCNF, which might yield PERs under 20% to Deep-HCNF on the TIMIT core test set. This would be our future work.

Table 2. *Comparison of phoneme recognition results between Deep-HCNF and triphone-HMM on the TIMIT core test set [%]*

Model	PER	#Param
Deep-HCNF (MMI)	24.9	2642864
Deep-HCNF (HB-MMI, $b = 1.0$)	24.3	
Triphone-HMM (MLE)	27.3	504823
Triphone-HMM (MPE)	27.6	

Table 3. *Comparison of phoneme recognition results between Deep-HCNF and triphone-HMM on the IPA100 test set [%]*

Model	PER	#Param
Deep-HCNF (MMI)	13.1	4734965
Deep-HCNF (HB-MMI, $b = 1.0$)	12.1	
Triphone-HMM (MLE)	14.1	5640178
Triphone-HMM (MPE)	12.4	

6. REFERENCES

[1] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Transactions of Acoustics Speech and Signal Processing*, vol. 34, no. 1, pp. 52 – 59, Feb. 1986.

[2] S. Nakagawa and K. Yamamoto, "Speech recognition using hidden markov models based on segmental statistics," *Systems and Computers in Japan*, vol. 28, no. 7, pp. 31–38, Jun. 1997.

[3] N. Kanedera, T. Arai, H. Hermansky, and M. Pavel, "On the relative importance of various components of the modulation spectrum for automatic speech recognition," *Speech Communication*, vol. 28, pp. 43–55, 5 1999.

[4] D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. thesis, Cambridge University Engineering Dept, 2003.

[5] Y. Fujii, K. Yamamoto, and S. Nakagawa, "Automatic speech recognition using hidden conditional neural fields," in *Proc. ICASSP*, Mar. 2011, pp. 5036 – 5039.

[6] Y. Fujii, K. Yamamoto, and S. Nakagawa, "Hidden boosted mmi and hierarchical state posterior feature for automatic speech recognition based on hidden conditional neural fields," in *Proc. Interspeech*, Aug. 2011, pp. 1001 – 1004.

[7] A. Mohamed, G. Dahl, and G. Hinton, "Deep Belief Networks for phone recognition," in *Proc. NIPS*, 2010.

[8] M. Mahajan, A. Gunawardana, and Alex Acero, "Training algorithms for hidden conditional random fields," in *Proc. ICASSP*, May 2006, pp. I–273–I–276.

[9] Y.-H. Sung and D. Jurafsky, "Hidden conditional random fields for phone recognition," in *Proc. ASRU*, 2009, pp. 107 – 112.

[10] T. N. Sainath, B. Ramabhadran, and M. Picheny, "An exploration of large vocabulary tools for small vocabulary phonetic recognition," in *Proc. ASRU*, 2009, pp. 359 – 364.

[11] K.-F. Lee and H.-W. HON, "Speaker-Independent Phone Recognition Using Hidden Markov Models," *IEEE Transactions of Acoustics Speech and Signal Processing*, vol. 37, no. 11, pp. 1641 – 1648, Nov. 1989.

[12] A. Mohamed, T. N. Sainath, G. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny, "Deep belief networks using discriminative features for phone recognition," in *Proc. ICASSP*, 2011, pp. 5060 – 5063.

[13] T. N. Sainath, D. Nahamoo, B. Ramabhadran, D. Kanevsky, V. Goel, and P. M. Shah, "Exemplar-based sparse representation phone identification features," in *Proc. ICASSP*, 2011, pp. 4492 – 4495.