Structured Prediction with Neural Networks in Speech Recognition

Liang Lu

TTIC

19 April 2016

Outline

- Speech Recognition as a Structured Prediction problem
- Hidden Markov Models
- Connectionist Temporal Classification
- Neural Segmental Conditional Random Field
- Encoder-Decoder with Attention

Structured Prediction

General supervised training:

$$\text{input: } x \quad \longrightarrow \quad \text{output: } y \tag{1}$$

- Classification
  - Input (x): scalar or vector,
  - Output(y): discrete class label
  - Loss: (usually) **0-1** loss
- Regression
  - Input (x): scalar or vector
  - Output (y): real number
  - Loss: (usually) **mean square error**

Structured Prediction

General supervised training:

$$\text{input: } x \quad \longrightarrow \quad \text{output: } y \tag{2}$$

- Structured Prediction
  - Input (x): set or sequence,
  - Output (y): sequence, tree, or graph
  - Loss: ?

Structured Prediction

General sequence transduction:

$$\text{input: } x_{1:T} \quad \longrightarrow \quad \text{output: } y_{1:L} \tag{3}$$

- Speech Recognition
  - Input (x): a sequence of vectors (length = T)
  - Output (y): a sequence of class labels (length = L)
  - Loss: **edit distance** (optimal, but not differentiable)
- Challenges
  - $T > L$: segmentation problem
  - $x_t \rightarrow ?$: alignment problem

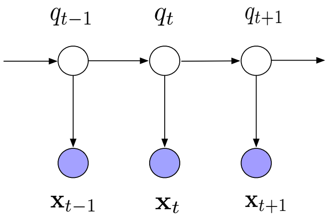
Hidden Markov Model

- General sequence transduction:

$$\text{input: } x_{1:T} \quad \longrightarrow \quad \text{output: } y_{1:L} \tag{4}$$

- Frame-level classification problem:

$$\text{input: } x_{1:T} \quad \longrightarrow \quad \text{hidden:} q_{1:T} \longrightarrow \quad \text{output: } y_{1:L} \tag{5}$$

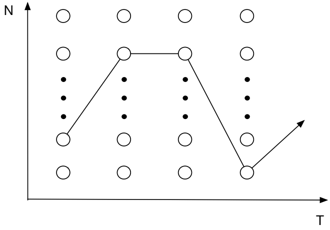


Hidden Markov Model

- Given $(x, q)_{1:T}$, mini-batch training of NN is straightforward
- Problem: how to get the hidden labels $q_{1:T}$?
- Expectation-Maximization algorithm
  - E: Given $x_{1:T}, y_{1:L}, \theta_{old}$, compute $\underbrace{P(q_{1:T}|x_{1:T}, y_{1:L}; \theta_{old})}_{\text{constrained decoding}}$
  - M: Given $x_{1:T}, q_{1:T}$, update model $\theta_{new} \leftarrow \theta_{old} + \delta\theta$
- Usually do many iterations

Hidden Markov Model

- Decoding and Constrained Decoding



- T is the number of time steps
- N is the number of HMM states

Hidden Markov Model

- Decoding graph: $H \circ C \circ L \circ G$
  - H: HMM transition ids to context dependent phones
  - C: context dependent phones to context independent phones
  - L: context independent phones to words
  - G: words to sequences of words
- Example: http://vpanayotov.blogspot.com/2012/06/kaldi-decoding-graph-construction.html

Hidden Markov Model

- Limitations:
  - Conditional independence: given $q_{1:T}$, every pair of x are independent
  - Local (frame-level) normalization: $P(q_t|x_t)$
  - Not end-to-end, many iterations to update $q_{1:T}$

Connectionist Temporal Classification

- Enumerate all the hidden labels (paths)

$$\text{input: } x_{1:T} \quad \longrightarrow \quad \text{hidden: } \begin{bmatrix} \color{blue}{q_{1:T}} \\ \color{green}{q_{1:T}} \\ \vdots \\ \color{red}{q_{1:T}} \end{bmatrix} \longrightarrow \quad \text{output: } y_{1:L} \tag{6}$$

- Marginalize out the hidden variables

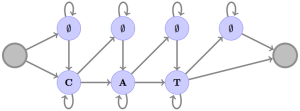
$$P(y_{1:L}|x_{1:T}) = \sum_{q_{1:T} \in \psi(y)} P(q_{1:T}|x_{1:T}) \tag{7}$$

- Again, local normalization

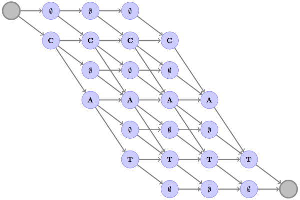
$$P(q_{1:T}|x_{1:T}) = \prod_t P(q_t|x_t) \tag{8}$$

Connectionist Temporal Classification

- How to enumerate all the paths?



(a)



(b)

- Can you enumerate all the paths for MT?

[1] R. Collobert, et al, "Wav2Letter: an End-to-End ConvNet-based Speech Recognition System", arXiv 2016

Connectionist Temporal Classification

- Role of the blank state (–), separating duplicated labels

  y: abbc $\longrightarrow$ q: {a, b} – {b, c}

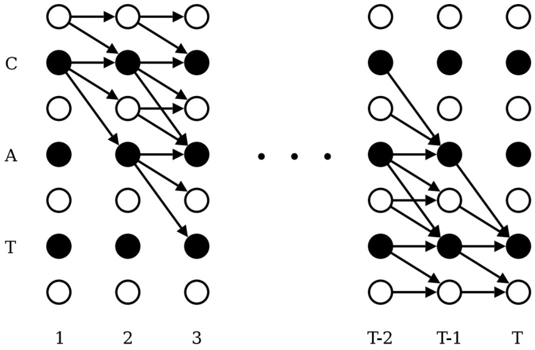
  q: –aaa–bb–bbb-cc– $\longrightarrow$ y: abbc

- Conditional maximum likelihood training

$$P(y_{1:L}|x_{1:T}) = \sum_{q_{1:T} \in \psi(y)} P(q_{1:T}|x_{1:T}) \tag{9}$$

- Forward-backward algorithm to compute the summed probability

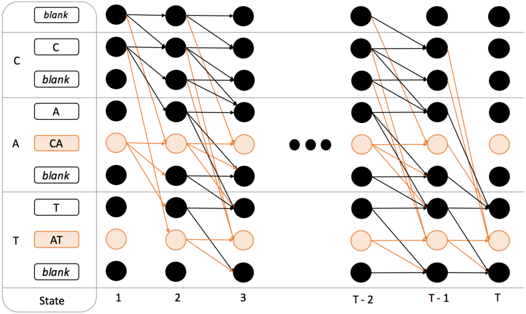
Connectionist Temporal Classification



[1] A. Graves, et al, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks", ICML 2006

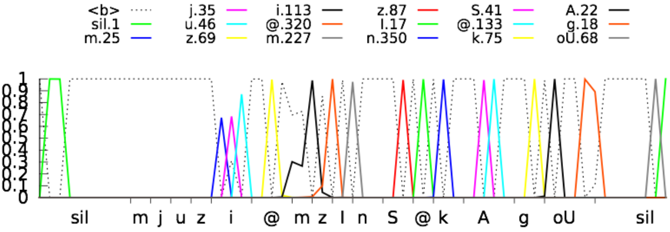
Connectionist Temporal Classification

- Gram-CTC: CTC with character n-grams



[1] H. Liu, et al, "Gram-CTC: Automatic Unit Selection and Target Decomposition for Sequence Labelling", arXiv 2017

Connectionist Temporal Classification



Q: Why most of the frames are labelled as blank?

[1] A. Senior, et al, “Acoustic Modelling with CD-CTC-sMBR LSTM RNNs”, ASRU 2015.

Connectionist Temporal Classification

Remarks:

- Suitable for end-to-end training
- Independence assumption: $P(q_{1:T}|x_{1:T}) = \prod_t P(q_t|x_t)$
- Scalable to large dataset
- Works with LSTM, CNN, but not DNN

(Segmental) Conditional Random Field

Sequence transduction for speech:

$$\text{input: } x_{1:T} \quad \longrightarrow \quad \text{output: } y_{1:L}$$

- CRF still require an alignment model for speech recognition
- Segmental CRF is equipped with implicit alignment

(Segmental) Conditional Random Field

- CRF [Lafferty et al. 2001]

$$P(y_{1:L} \mid x_{1:T}) = \frac{1}{Z(x_{1:T})} \prod_j \exp\left(w^\top \Phi(y_j, x_{1:T})\right) \qquad (10)$$

where $L = T$.

- Segmental (semi-Markov) CRF [Sarawagi and Cohen 2004]

$$P(y_{1:L}, E, \mid x_{1:T}) = \frac{1}{Z(x_{1:T})} \prod_j \exp\left(w^\top \Phi(y_j, e_j, \boldsymbol{x}_{1:T})\right) \qquad (11)$$

where $e_j = \langle s_j, n_j \rangle$ denotes the beginning (s_j) and end (n_j) time tag of y_j; $E = \{e_{1:L}\}$ is the latent segment label.

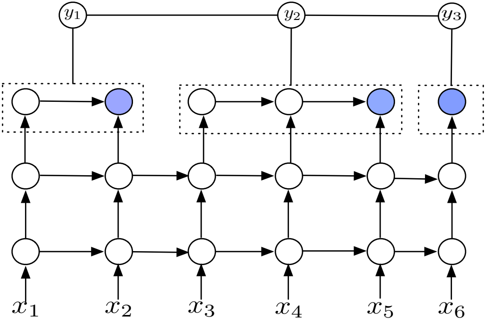
(Segmental) Conditional Random Field

$$\frac{1}{Z(x_{1:T})} \prod_j \exp\left(w^\top \Phi(y_j, x_{1:T})\right)$$

- Learnable parameter w
- Engineering the feature function $\Phi(\cdot)$
- Designing $\Phi(\cdot)$ is much harder for speech than NLP

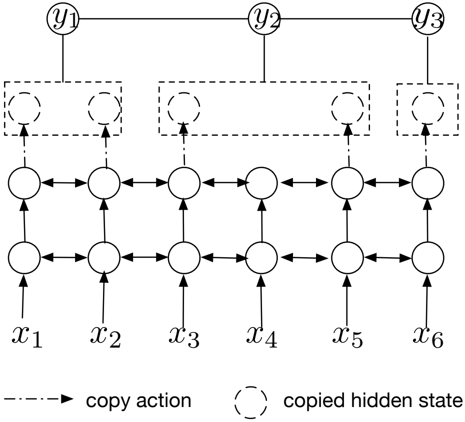
Segmental Recurrent Neural Network

- Using (recurrent) neural networks to learn the feature function $\Phi(\cdot)$.



Segmental Recurrent Neural Network

- More memory efficient



Segmental Recurrent Neural Network

- Comparing to previous segmental models
  - M. Ostendorf et al., "From HMM's to segment models: a unified view of stochastic modeling for speech recognition", IEEE Trans. Speech and Audio Proc. 1996
  - J. Glass, "A probabilistic framework for segment-based speech recognition", Computer Speech & Language, 2002
- Markovian framework vs. CRF framework (local vs. global normalization)
- Neural network feature (and end-to-end training)

Related works

- (Segmental) CRFs for speech
- Neural CRFs
- Structured SVMs
- Two good review papers
  - M. Gales, S. Watanabe and E. Fosler-Lussier, "Structured Discriminative Models for Speech Recognition", IEEE Signal Processing Magazine, 2012
  - E. Fosler-Lussier et al. "Conditional random fields in speech, audio, and language processing, Proceedings of the IEEE, 2013

Segmental Recurrent Neural Network

- Training criteria
  - Conditional maximum likelihood

$$\begin{aligned}\mathcal{L}(\theta) &= \log P(y_{1:L} \mid x_{1:T}) \\ &= \log \sum_{E} P(y_{1:L}, E \mid x_{1:T}) \end{aligned} \tag{12}$$

  - Hinge loss – similar to structured SVM
  - Marginalized hinge loss

[1] H. Tang, et al, "End-to-end training approaches for discriminative segmental models", SLT, 2016

Segmental Recurrent Neural Network

- Viterbi decoding
  - Partially Viterbi decoding

$$y^*_{1:L} = \arg\max_{y_{1:L}} \log \sum_{E} P(y_{1:L}, E \mid x_{1:T}) \tag{13}$$

  - Full Viterbi decoding

$$y^*_{1:L}, E^* = \arg\max_{y_{1:L}, E} \log P(y_{1:L}, E \mid x_{1:T}) \tag{14}$$

Segmental Recurrent Neural Network

Remarks:

- No independence assumption
- Globally (sequence-level) normalized model
- Computationally expensive, not very scalable

Scale to Large Vocabulary ASR

- Why Segmental CRF expensive?

$$P(y_{1:L}, E, \mid x_{1:T}) = \frac{1}{Z(x_{1:T})} \prod_j \exp\left(w^\top \Phi(y_j, e_j, x_{1:T})\right) \quad (15)$$

where $e_j = \langle s_j, n_j \rangle$ denotes the beginning (s_j) and end (n_j) time tag.

$$Z(x_{1:T}) = \sum_{y,E} \prod_{j=1}^{J} \exp f\left(y_j, e_j, x_{1:T}\right). \quad (16)$$

- Computation complexity is $O(T^2|\mathcal{V}|)$

Scale to Large Vocabulary ASR

- Analogous to large softmax for language modeling

$$P(w) = \frac{\exp(z_w)}{\sum_{w' \in \mathcal{V}} \exp(z_{w'})} \tag{17}$$

- Noise Contrastive Estimation
- Importance Sampling
- Can we try similar ideas for SCRF?

Attention Model

Sequence transduction for speech:

$$\text{input: } x_{1:T} \quad \longrightarrow \quad \text{output: } y_{1:L}$$

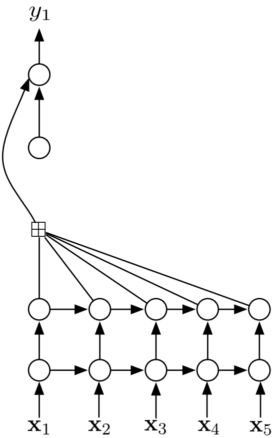
Compute the conditional probability

$$P(y_{1:L}|x_{1:T}) = \prod_{l=1}^{L} P(y_l|y_{<1}, x_{1:T}) \tag{18}$$

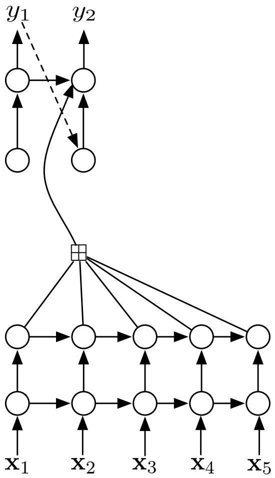
$$\approx \prod_{l=1}^{L} P(y_l|y_{<1}, c_l) \tag{19}$$

$$c_l = \text{attEnc}(y_{<1}, x_{1:T}) \tag{20}$$

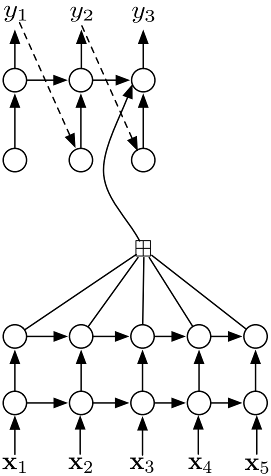
Attention Model



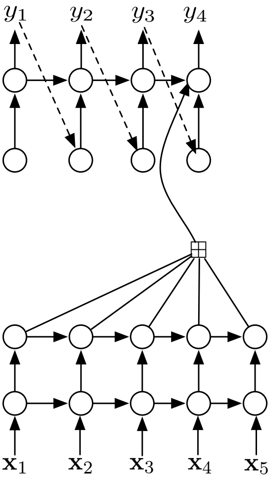
Attention Model



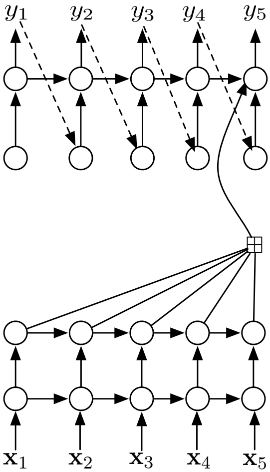
Attention Model



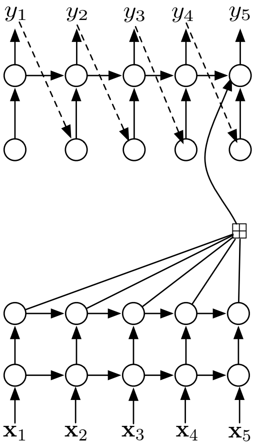
Attention Model



Attention Model



Attention Model



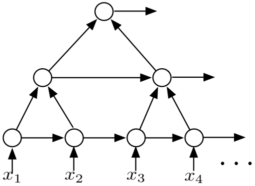
Decoder $P(y_j \mid y_1, \cdots, y_{j-1}, \mathbf{c}_j)$

Attention $\mathbf{c}_j = \mathrm{Attend}(\mathbf{h}_{1:T})$

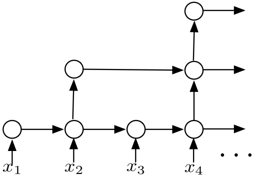
Encoder $\mathbf{h}_{1:T} = \mathrm{RNN}(\mathbf{x}_{1:T})$

Attention Model

Encoder with pyramid RNN



a) concatenate / add



b) skip

Attention Model

- Remarks
  - monotonic alignment $\times$
  - independence assumption for inputs $\times$
  - long input sequence $\surd$
  - length mismatch $\surd$
  - Locally normalized for each output token

$$P(y_{1:L}|x_{1:T}) \approx \prod_{l} P(y_l|y_{<l}, c_l) \tag{21}$$

Attention Model

- Locally normalized models:
  - conditional independence assumption
  - label bias problem
  - We care more about the sequence level loss in speech recognition
  - . . .

[1] D. Andor, et al, "Globally Normalized Transition-Based Neural Networks", ACL, 2016

Speech Recognition

- Locally to globally normalized models:
  - HMMs: CE $\rightarrow$ sequence training
  - CTC: CE $\rightarrow$ sequence training
  - Attention model: Minimum Bayes Risk training

$$\mathcal{L} = \sum_{y \in \Omega} P(y|x) A(y, \hat{y}) \tag{22}$$

  - Would be interesting to look at this for speech

[1] S. Shen, et al, "Minimum Risk Training for Neural Machine Translation", ACL, 2016
[2]S. Wiseman, A. Rush, "Sequence-to-Sequence Learning as Beam-Search Optimization", EMNLP, 2016

Experimental Results

- TIMIT dataset ($\sim$ 1 million frames)
- WSJ ($\sim$ 30 million frames)
- SWBD ($\sim$ 100 millon frames)

Experiments on TIMIT

Table: Results on TIMIT. LM = language model, SD = speaker dependent feature

System	LM	SD	PER
HMM-DNN	$\surd$	$\surd$	18.5
CTC [Graves 2013]	$\times$	$\times$	18.4
RNN transducer [Graves 2013]	–	$\times$	17.7
Attention model [Chorowski 2015]	–	$\times$	17.6
Segmental RNN	$\times$	$\times$	18.9
Segmental RNN	$\times$	$\surd$	17.3

Experiments on WSJ

Table: Results on WSJ. LM = language model

System	LM	WER(%)
HMM-DNN (phone)	$\surd$	3 - 4
CTC [Graves & Jaitly 2014]	$\times$	30.1
CTC [Graves & Jaitly 2014]	$\surd$	8.7
CTC [Miao 2015]	$\surd$	7.3
Gram-CTC [Liu 2017]	$\surd$	6.8
Attention model [Chan 2016]	–	9.6
Attention model [Chorowski 2016]	$\surd$	6.7

Experiments on SWBD

Table: Results on SWBD. LM = language model

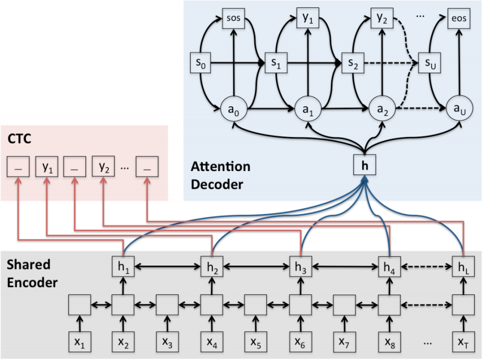
System	LM	WER(%)
HMM-DNN (phone)	$\surd$	9.6
HMM-DNN (phone) (2000h)	$\surd$	5.5
CTC [Zweig 2016]	$\times$	24.7
CTC [Zweig 2016]	$\surd$	14.0
Gram-CTC [Liu 2017] (2000h)	$\surd$	7.3
Attention model [Lu 2016]	$\times$	26.8
Attention model [Toshniwal 2017]	$\times$	23.1

Multitask Learning

- Weaknesses of end-to-end models
  - Attention model – alignment problem in the early stage of training
  - CTC model – conditional independence assumption
  - SRNN model – large computational cost
- Multitask learning to mitigate the weaknesses

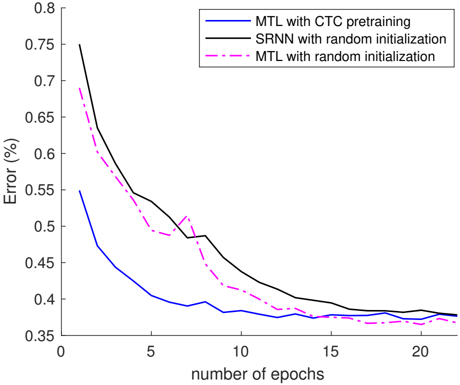
[1] S. Kim, T. Hori, S. Watanabe, "Joint CTC-Attention based End-to-End Speech Recognition using Multi-task Learning ", ICASSP 2017.

Multitask Learning



[1] S. Kim, T. Hori, S. Watanabe, "Joint CTC-Attention based End-to-End Speech Recognition using Multi-task Learning ", ICASSP 2017.

Multitask Learning



[1] L. Lu et al., "Multi-task Learning with CTC and Segmental CRF for Speech Recognition", arXiv 2017.

Conclusion

- Structured prediction for speech recognition
- End-to-end training models
- Flexibility vs. Scalability
- Other deep learning architectures