2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

Tutorial

Energy-Based Models with Applications to Speech and Language Processing

Zhijian Ou

Speech Processing and Machine Intelligence (SPMI) Lab, Tsinghua University

http://oa.ee.tsinghua.edu.cn/ouzhijian/

May 22, 2022

Introduction

- **Energy-Based Models (EBMs)** are an important class of probabilistic models, also known as **random fields (RFs)** and **undirected graphical models (UGMs)**.
- EBMs have **unique properties** and are radically different from some other popular probabilistic models such as
  - hidden Markov models (HMMs), auto-regressive models, Generative Adversarial Nets (GANs) and Variational Auto-encoders (VAEs), which are self-normalized (i.e., sum to one).
- EBMs have attracted **increasing interests** not only from core machine learning but also from application domains such as vision, speech, natural language processing
  - with significant theoretical and algorithmic progress
- **The sequential nature of speech and language** also presents special challenges and needs treatment different from processing fix-dimensional data (e.g., images).

The purpose of this tutorial is to present a **systematic** introduction to energy-based models, including both algorithmic progress and applications in speech and language processing.

Content

I. Basics for EBMs (45 min)

1. Probabilistic graphical modeling (PGM) framework and EBM model examples (classic & modern)
2. Learning EBMs by Monte Carlo methods
3. Learning EBMs by noise-contrastive estimation (NCE)

II. EBMs for language modeling (45 min)

1. Trans-dimensional random field (TRF) LMs for speech recognition
2. Residual energy-based models for text generation
3. Electric: an energy-based cloze model for representation learning over text

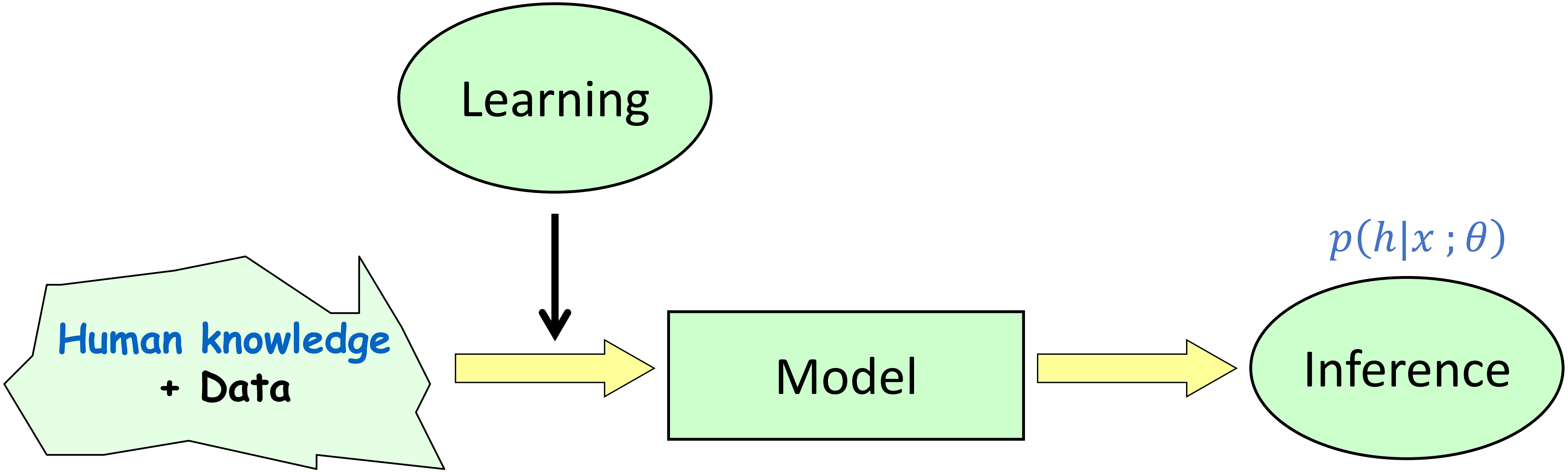
III. EBMs for speech recognition and natural language labeling(45 min)

1. CRFs as conditional EBMs
2. CRFs for speech recognition
3. CRFs for sequence labeling in NLP

IV. EBMs for semi-supervised natural language labeling (45 min)

1. Upgrading EBMs to Joint EBMs (JEMs) for fixed-dimensional data
2. Upgrading CRFs to Joint random fields (JRFs) for sequential data
3. JRFs for semi-supervised natural language labeling

Probabilistic Framework



$p(x, h;\ \theta)$: Generative model, e.g., Hidden Markov Model (HMM)

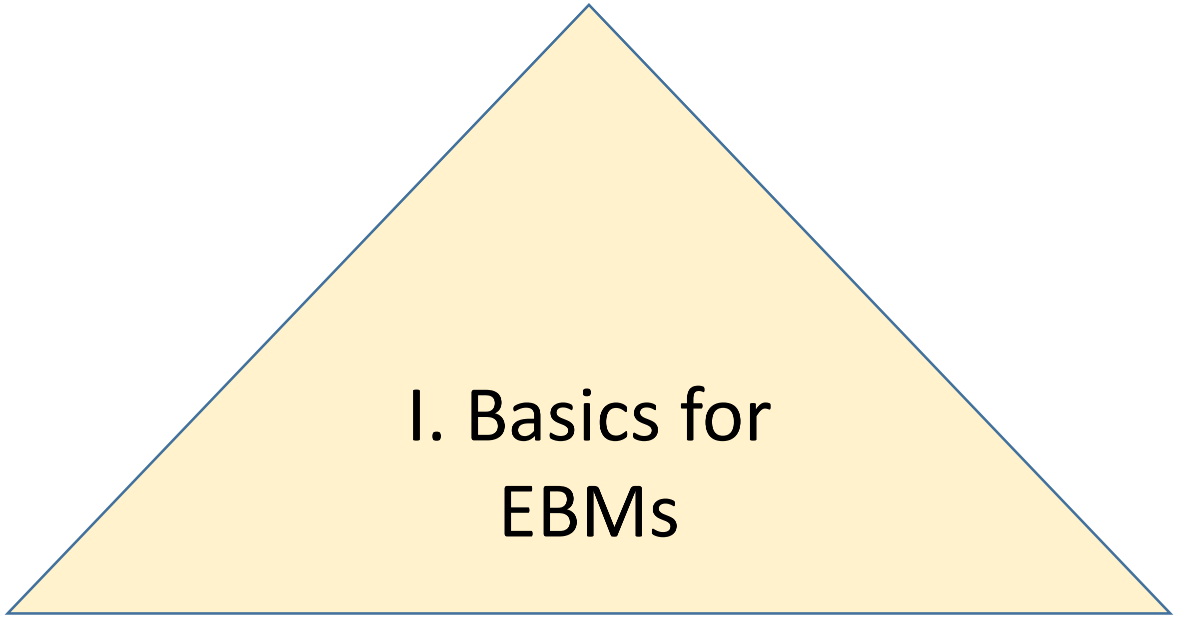
$p(h|x;\ \theta)$: Discriminative model, e.g., Conditonal Random Field (CRF)

We need probabilistic models, besides neural nets.

Roadmap

II. EBMs for language modeling

$p_\theta(x)$



$p_\theta(h|x)$

III. EBMs for speech recognition and natural language labeling

$p_\theta(x,h)$

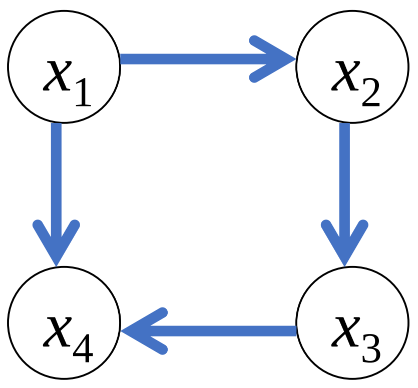
IV. EBMs for semi-supervised natural language labeling

Probabilistic Graphical Modeling (PGM) Framework

A general framework for describing and applying probabilistic models

- A graphical model is a family of probability distributions defined in terms of a directed or undirected graph.
- Semantics: how the family of distributions is defined.

- Semantics of Directed Graphical Models (DGMs)



Consider a directed acyclic graph (DAG)

- x_V : a collection of random variables indexed by the nodes
- $pa(v)$: the parent nodes of v

$$p(x_V) = \prod_{v \in V} p\left(x_v | x_{pa(v)}\right)$$

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_1, x_3)$$

DGM example - Neural Net (NN) based classifier

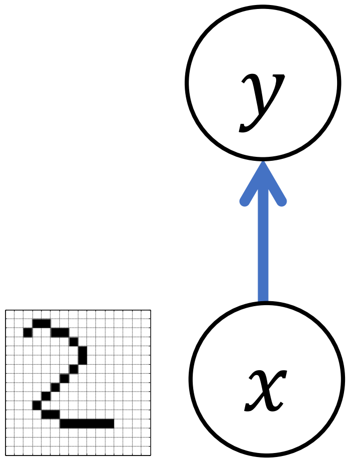
- Multi-class logistic regression

Consider observation/features $x \in \mathbb{R}^d$, class label $y \in \{1, \cdots, K\}$

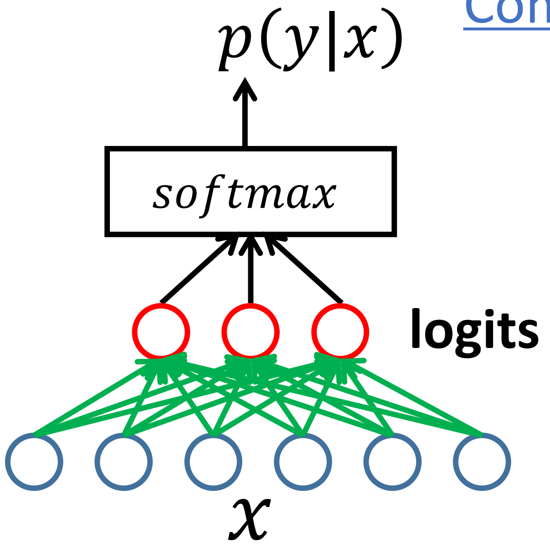
$$p(y = k|x) = \frac{exp(z_k)}{\sum_{j=1}^{K} exp(z_j)} \triangleq softmax(z_k)$$

where $z_k = w_k^T x + b_k, k = 1, \cdots, K,$ often called **logits**

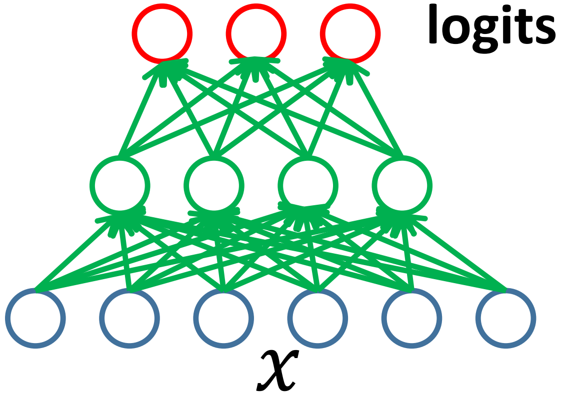
GM Representation



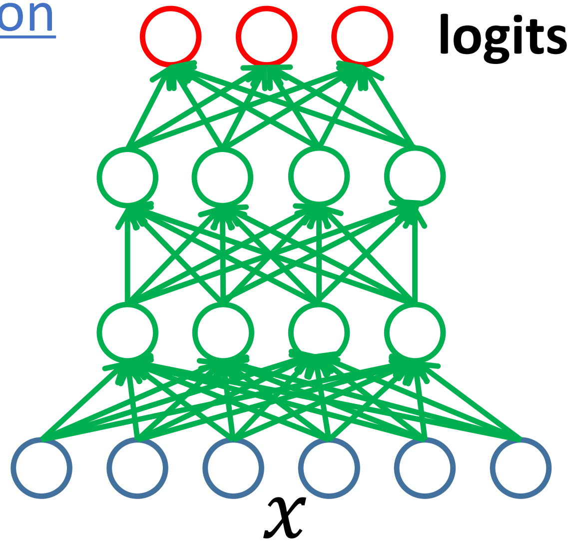
Computational Graph Representation



1-layer NN



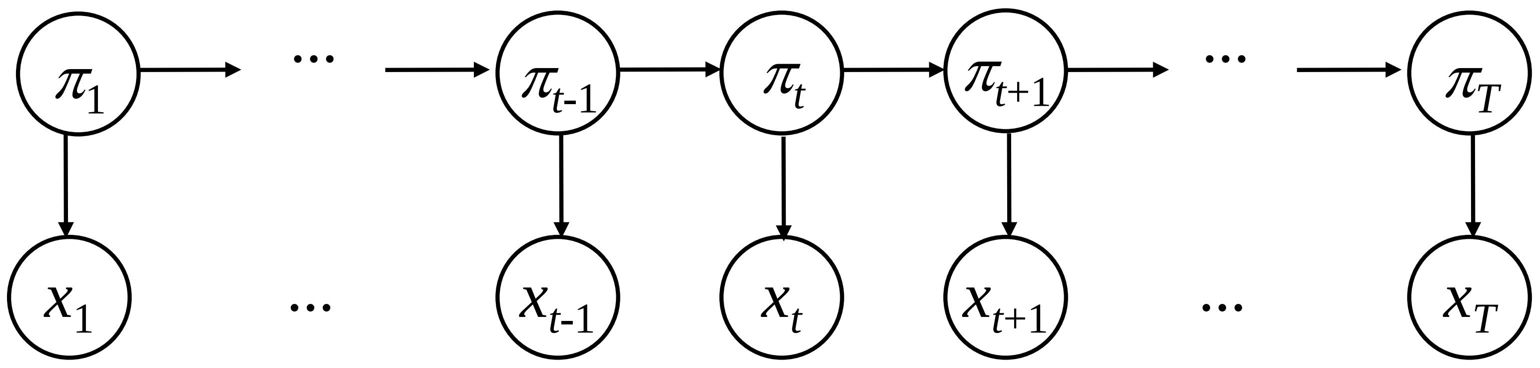
2-layer NN



3-layer NN

(NNs as feature extractors)

HMM Viewed as Directed Graphical Model



The joint probability distribution of a hidden Markov model (HMM) :

$$p(\pi_{1:T}, x_{1:T}) = p(\pi_1) \prod_{t=1}^{T-1} p(\pi_{t+1}|\pi_t) \prod_{t=1}^{T} p(x_t|\pi_t)$$

State Initial Distr.

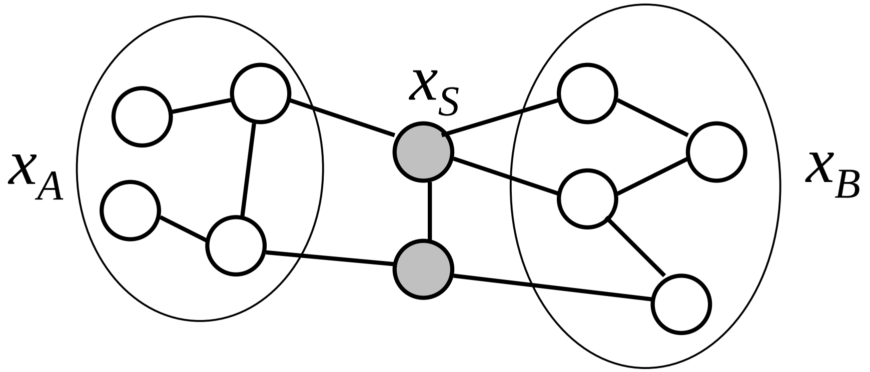
State Transition Distr.

State Observation Distr.

Undirected Graphical Model (UGM) Semantics - G property

- A probability distribution $p(x_V)$ is said to obey the Global Markov property, relative to a undirected graph g , if for any triple (A, B, S) of disjoint subsets of V such that S *separates* A *from* B,

$$x_A \perp x_B \mid x_S$$



S *separates* A *from* B : if all trails from A to B intersect S

UGM Semantics - Factorization property

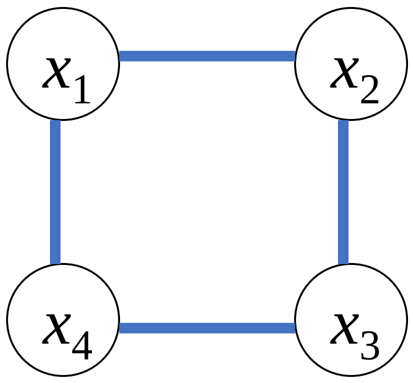
Hammersley-Clifford Theorem: If p is strictly positive, (F)⟺(G).

- A probability distribution $p(x_V)$ is said to factorize according to g, if there exist non-negative functions (called potential functions) $\phi_C(x_C)$ for all cliques C such that

$$p(x_V) = \frac{1}{Z}\prod_{C\in\mathcal{C}}\phi_C(x_C) \quad \text{or} \quad p(x_V) \propto \prod_{C\in\mathcal{C}}\phi_C(x_C)$$

where Z is the normalizing constant (partition function)

$$Z = \sum_{x_V}\prod_{C\in\mathcal{C}}\phi_C(x_C)$$



$$p(x_1,x_2,x_3,x_4) = \frac{1}{Z}\phi(x_1,x_2)\phi(x_2,x_3)\phi(x_3,x_4)\phi(x_1,x_4)$$

A subset of nodes C is called a clique, if every pair of nodes in C is joined.

UGMs and Energy-Based Models (EBMs)

- Let every clique potential be associated with a clique energy $E(x_C)$

$$E_C(x_C) = -log\phi_C(x_C)$$

$$\text{energy} = -\,log\ \text{potential}$$

- The resulting distribution is known as the Gibbs (or Boltzmann) distribution, originating from statistical physics

$$p(x_V) \propto exp\left[-\sum_{C\in\mathcal{C}} E_C(x_C)\right]$$

High probability states correspond to low energy configurations.

UGMs and Log-Linear Models

- Let each clique potential be a log-linear function

$$log\phi_C(x_C) = \theta_C^T f_C(x_C)$$

where $f_C(x_C)$ is a feature vector derived from (the values of) the variables x_C, θ_C is the associated feature weight vector.

- The resulting distribution has the form

$$p(x_V) = \frac{1}{Z(\theta)} exp\left[\sum_C \theta_C^T f_C(x_C)\right]$$

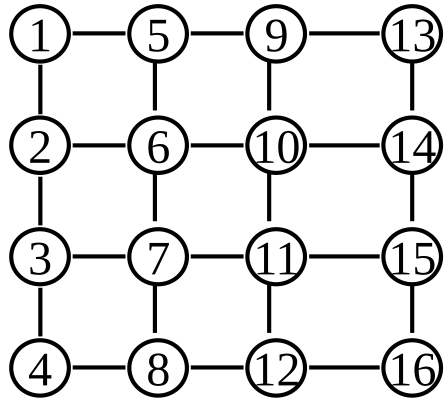
i.e., the distribution with the maximum entropy s.t. empirical expectation of f_C = model expectation f_C

This is known as a Log-Linear Model or a Maximum Entropy Model.

It can be proved that the maxent distribution is the same as the maximum likelihood distribution from the closure of the set of log-linear distributions.

S. D. Pietra, V. D. Pietra, and J. Lafferty, “Inducing features of random fields”, IEEE PAMI, 1997.

UGM Example - Ising model



- Consider a lattice of binary RV's, $x_i \in \{-1,1\}$

$$p(x_{1:N^2}) \propto exp\left\{-\sum_{i\sim j} E(x_i, x_j)\right\} = exp\left\{\beta \sum_{i\sim j} x_i x_j\right\} \quad \beta > 0$$

  - β : how much neighboring variables take identical values is favored.
  - Samples of Ising models on a lattice with different β:

β = 0.1 or 10 ?

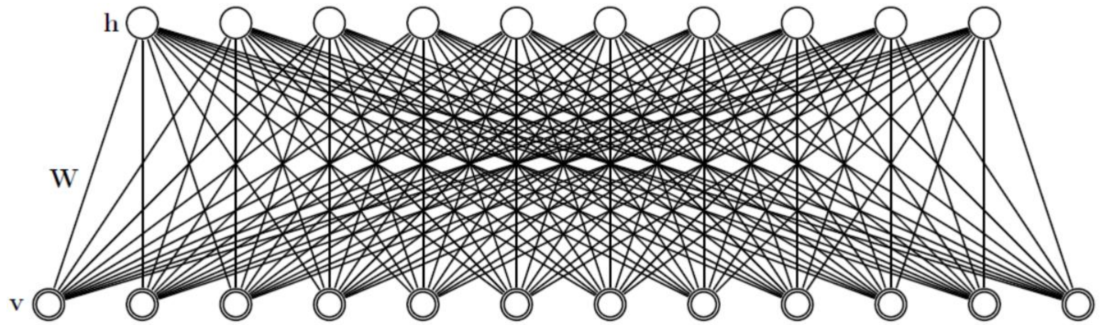
β = 1

β = 0.1 or 10 ?

EBMs are natural for modeling interactions (mutual influences), where the directions of edges cannot be clearly defined.

UGM Example - Restricted Boltzmann Machines (RBMs)

- RBM is the main building block of Deep Belief Network, which ignites Deep Learning
- RBM is a UGM over a bipartite graph
  - Binary visible variables $v \in \{0,1\}^D$
  - Binary hidden variables $h \in \{0,1\}^F$
  - $\theta = \{W, b, a\}$



RBM: a stochastic version of a NN

$$p(v,h;\theta) = \frac{1}{Z(\theta)} exp[-E(v,h;\theta)]$$

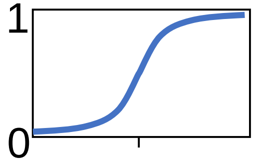
$$E(v,h;\theta) = -v^T Wh - b^T v - a^T h$$

$$= -\sum_{i=1}^{D}\sum_{j=1}^{F} v_i W_{ij} h_j - \sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{F} a_j h_j$$

$$p(h|v;\theta) = \prod_j p(h_j|v), \qquad p(h_j = 1|v) = \sigma\left(\sum_i W_{ij} v_i + a_j\right)$$

$$p(v|h;\theta) = \prod_i p(v_i|h), \qquad p(v_i = 1|h) = \sigma\left(\sum_j W_{ij} h_j + b_i\right)$$

Sigmoid function : $\sigma(x) = \frac{1}{1+e^{-x}}$

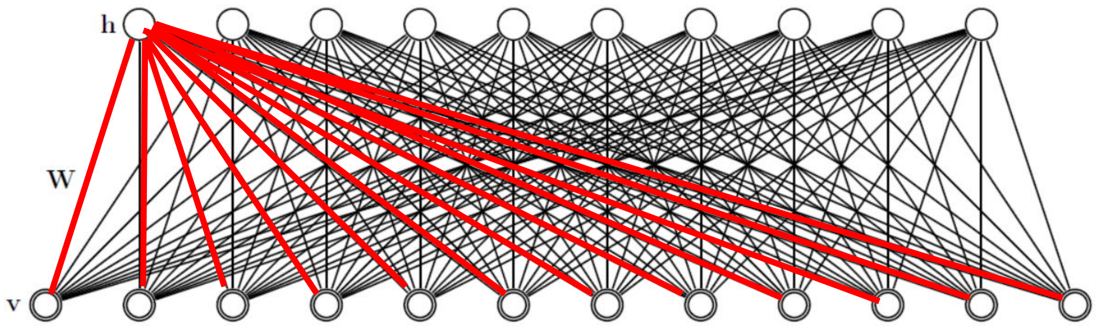


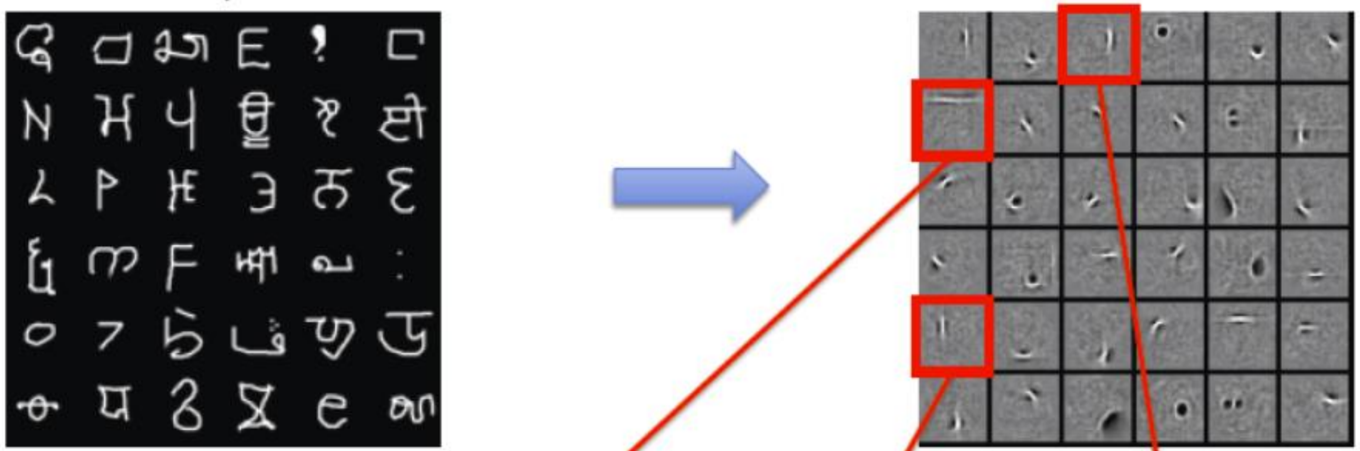
Hinton and Salakhutdinov. "Reducing the dimensionality of data with neural networks." Science, 2006.

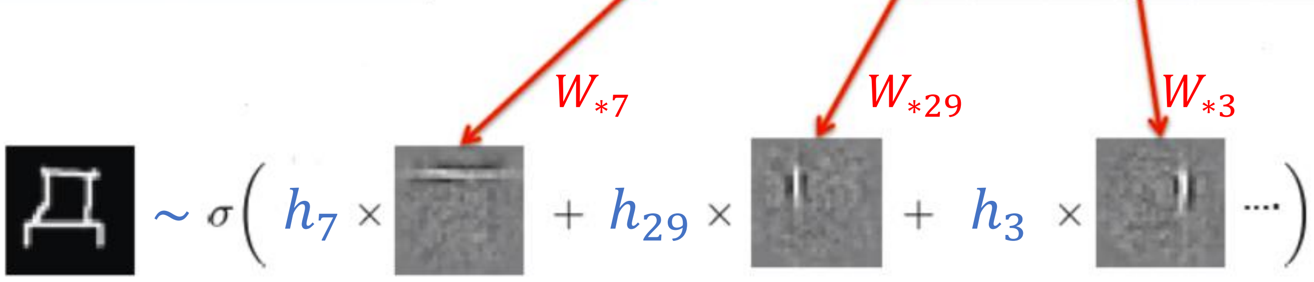
Learned features W_{*j}

Learned receptive fields for unit h_j

h: higher-level encoding of v





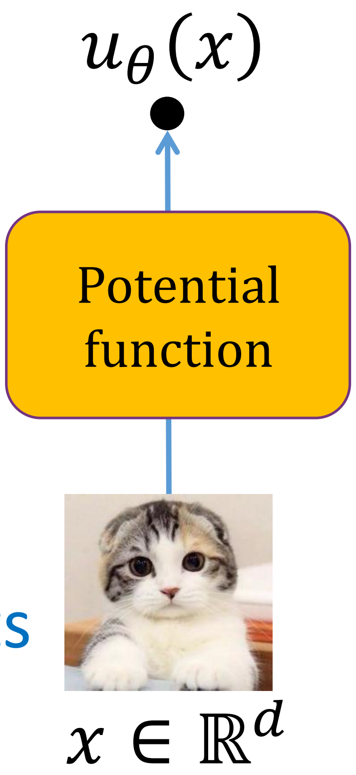


$$p(v|h;\theta) = \prod_i p(v_i|h),$$

$$p(v_i = 1|h) = \sigma\left(\sum_j W_{ij} h_j + b_i\right)$$

$$v|h \sim \sigma(h_1 \cdot W_{*1} + h_2 \cdot W_{*2} + \cdots + b)$$

Neural Random Fields (NRFs) - Basics



- NRFs are defined by using NNs to implement $u_\theta(x)$: $\mathbb{R}^d \rightarrow \mathbb{R}$

$$p_\theta(x) = \frac{1}{Z(\theta)} exp[u_\theta(x)]$$

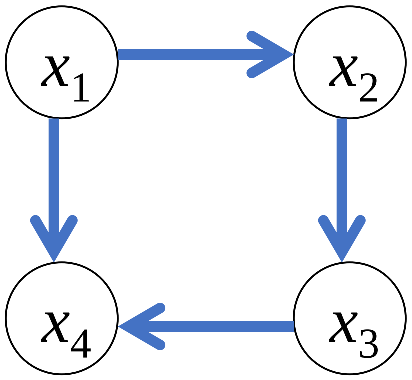
  - $u_\theta(x)$ can be very flexibly defined

- This type of EBMs has been studied several times in different contexts
  - Deep energy models (DEMs)
    - Ngiam et al., 2012
    - Kim & Bengio, 2016 - includes linear and squared terms in $u_\theta(x)$
  - Descriptive models / Generative ConvNet
    - Xie et al., 2016 / Dai et al., 2014 - defines in the form of exponential tilting of a reference distribution (Gaussian white noise)
  - Neural random field language models
    - Wang & Ou, 2017 - defines over sequences

See references in: Yunfu Song, Zhijian Ou. Learning Neural Random Fields with Inclusive Auxiliary Generators. arXiv:1806.00271, 2018.

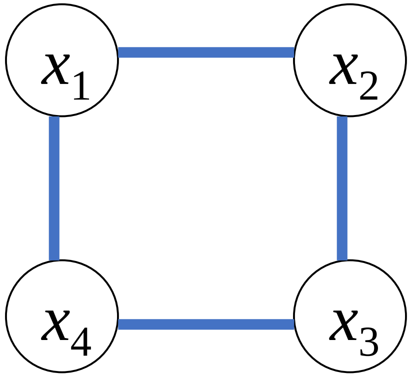
Probabilistic Graphical Modeling (PGM) Framework - Summary

- **Directed Graphical Models / Bayesian Networks (BNs)**
  - Self-normalized/Local-normalized
  - e.g. Hidden Markov Models (HMMs), Neural network (NN) based classifiers, Variational AutoEncoders (VAEs), Generative Adversarial Networks (GANs), auto-regressive models (e.g. RNNs/LSTMs)



$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_1, x_3)$$

- **Undirected Graphical Models / Random Fields (RFs) / Energy-based models**
  - Involves the normalizing constant Z / Globally-normalized
  - e.g. Ising model, Conditional Random Fields (CRFs)



$$p(x_1, x_2, x_3, x_4) = \frac{1}{Z}\Phi(x_1, x_2)\Phi(x_2, x_3)\Phi(x_3, x_4)\Phi(x_1, x_4)$$

Content

MCMC example: the Metropolis–Hastings algorithm

Problem: We want to draw samples from a target distribution $p(x)$?

Solution: Construct a Markov chain that has $p(x)$ as the stationary distribution.

1. Randomly initialize x_0

2. For $t = 1, \cdots$

   Generates x^* from a proposal $q(x^* | x_{t-1})$,

   Accept $x_t = x^*$ with probability $min\left\{1, \frac{p(x^*)q(x_{t-1}|x^*)}{p(x_{t-1})q(x^*|x_{t-1})}\right\}$,

   otherwise set $x_t = x_{t-1}$

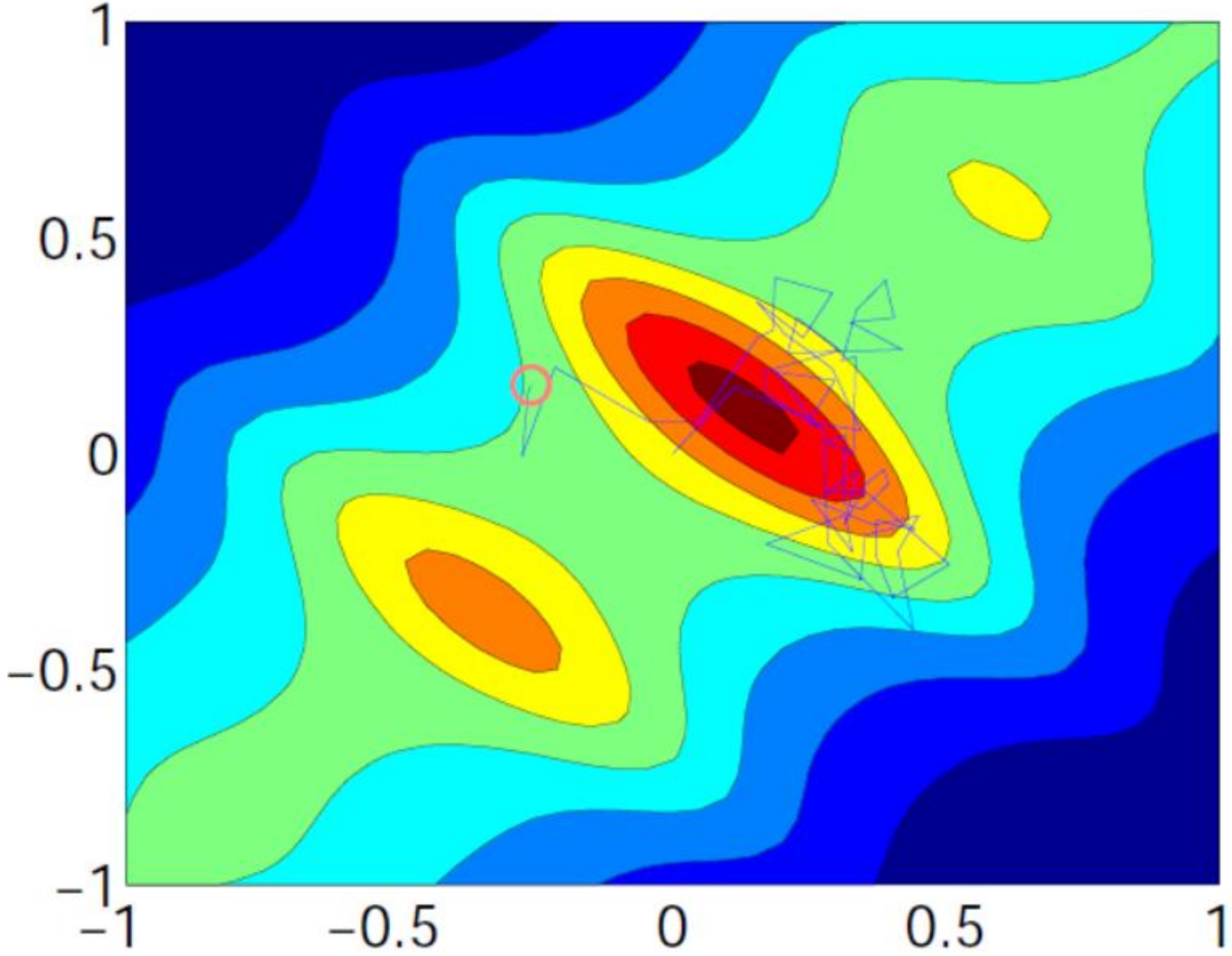
   Burn-in : first few samples are discarded.

- For an irreducible & ergodic Markov chain, there exist stationary distribution π, which satisfies equation $\pi = \pi \mathrm{P}$.
- A sufficient condition: satisfy the detailed balance equation

$$\pi_i \mathrm{P}_{ij} = \pi_j \mathrm{P}_{ji}$$

Metropolis–Hastings example

e.g. $q(x^*|x_{t-1}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x^*-x_{t-1})^2}{2\sigma^2}}$



Training of EBMs in general

$$p(x;\theta) = \frac{1}{Z(\theta)} \exp[u(x;\theta)]$$

Normalizing constant:

$$Z(\theta) = \sum_x \exp[u(x;\theta)]$$

- Maximum likelihood (ML) training

The scaled log-likelihood of observations $\{x_i, i = 1, \cdots, N\}$

$$L(\theta) \triangleq \frac{1}{N}\sum_{i=1}^{N} logp(x_i;\theta) = \left[\frac{1}{N}\sum_{i=1}^{N} u(x_i;\theta)\right] - logZ(\theta)$$

$$\frac{\partial L(\theta)}{\partial \theta} = E_{\tilde{p}(x)}\left[\frac{\partial u(x;\theta)}{\partial \theta}\right] - E_{p(x;\theta)}\left[\frac{\partial u(x;\theta)}{\partial \theta}\right] = 0$$

Maximum Entropy

Expectation under empirical distribution $\tilde{p}(x) = \frac{1}{N}\sum_{i=1}^{N} 1(x = x_i)$

Expectation under model distribution $p(x;\theta)$

Learning EBMs by Monte Carlo methods

gradient = empirical expectation – model expectation

1. Approximate the model expectations using Monte Carlo sampling.
   - We can use MCMC to generate the samples, but running MCMC to convergence at the inner loop would be extremely slow.
   - Fortunately, it was shown by Younes (1989) that we can start the MCMC chain at its previous value from the outer loop, and just take a few steps in the inner loop.

```
For mini-batch iterations
    Obtain empirical expectations;
    For Monte Carlo iterations
        Obtain model expectations;
    End
    update parameters;
End
```

2. We can combine this with stochastic gradient descent (SGD), which takes mini-batches of samples from the empirical distribution.

Both ideas are applications of the Stochastic Approximation (SA) methodology!

- Robbins and Monro. A stochastic approximation method. The annals of mathematical statistics, 1951.
- L. Younes, "Parametric inference for imperfectly observed Gibbsian fields," Probability Theory and Related Fields, 1989.

Stochastic Approximation (SA)

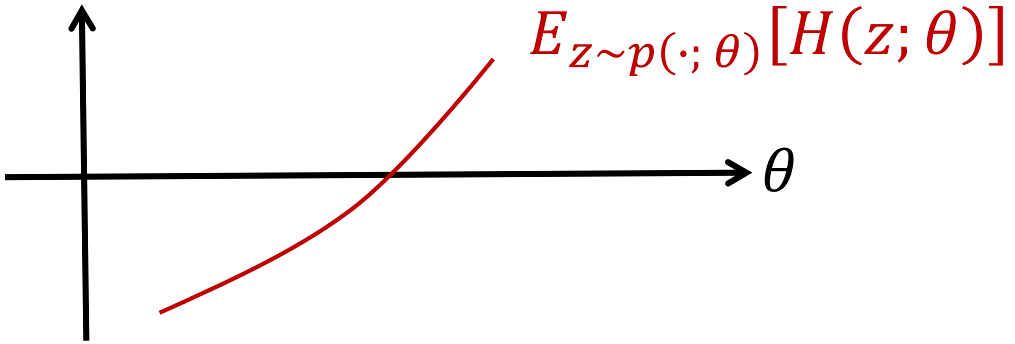
Problem: find a solution θ to $f(\theta) \triangleq E_{z\sim p(\cdot;\,\theta)}[F(z;\theta)] = 0$,
where $\theta \in R^d$, noisy measurement $H(z;\theta) \in R^d$

Method:

(1) Sampling: Generate $z_t \sim K(z_{t-1}, \cdot\,; \theta_{t-1})$, a Markov transition kernel that admits $p(\cdot;\ \theta_{t-1})$ as the invariant distribution.

(2) Updating: Set $\theta_t = \theta_{t-1} + \gamma_t H(z_t; \theta_{t-1})$

When $f(\theta)$ corresponds to the gradient of some objective function, then under certain regularity conditions, θ_t will converge to a optimal solution.



- Robbins and Monro. A stochastic approximation method. The annals of mathematical statistics, 1951.
- Chen (2002), Stochastic Approximation and Its Applications, Kluwer Academic Publishers.

Connection between existing RF training methods

- Stochastic Approximation (SA), Robbins and Monro 1951.
- aka Stochastic Maximum Likelihood (SML), Younes 1989.
- This was independently discovered by Tieleman in 2008, who called it persistent contrastive divergence (PCD).
- In regular contrastive divergence (CD), proposed by Hinton 2002, we restart the Markov chain at the training data rather than at the previous state. This will not converge to the MLE.
- “Clearly, the widely used practice of CD1 learning is a rather poor “substitute” for maximum likelihood learning. ” (Salakhutdinov phd thesis 2009).

Content

Learning EBMs by noise-contrastive estimation (NCE)

- The target RF model $p_\theta(x) = \frac{1}{Z(\theta)} e^{u_\theta(x)}$
- Treat $\log Z(\theta)$ as a parameter ζ and rewrite $p_{\theta,\zeta}(x) \propto e^{u_\theta(x)-\zeta}$
- Introduce a **noise distribution** $q(x)$, and consider a binary classification

$$P(C=0|x) = \frac{p_{\theta,\zeta}(x)}{p_{\theta,\zeta}(x) + \nu q(x)}, where \ \nu = \frac{P(C=1)}{P(C=0)}$$

$$P(C=1|x) = 1 - P(C=0|x)$$

- Noise Contrastive Estimation (NCE):

$$\max_{\theta,\zeta} E_{x\sim p_0(x)}[\log P(C=0|x)] + \nu E_{x\sim q(x)}[\log P(C=1|x)]$$

Consistency: $p_\theta \rightarrow p_0$ (oracle), under infinite amount of data and infinite capacity of p_θ.

Michael Gutmann, Aapo Hyvarinen. “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” AIStat, 2010.

NCE discussion

- In applying NCE, a natural question to ask is: from a statistical standpoint, what the best choice of q and ν would be?

To get estimates with a small estimation error, the foregoing discussion suggests the following

1. Choose noise for which an analytical expression for $\ln p_n$ is available.
2. Choose noise that can be sampled easily.
3. Choose noise that is in some aspect, for example with respect to its covariance structure, similar to the data.
4. Make the noise sample size as large as computationally possible.

Michael Gutmann, Aapo Hyvarinen. "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," AIStat, 2010.

Dynamic noise-contrastive estimation (DNCE)

- Reliable NCE needs a large $\nu \approx 20$, which almost linearly increases the training cost.
- The model estimated by NCE could be overfitted to the empirical distribution.

$x \sim \alpha p_0(x) + (1-\alpha) q_\phi(x)$

$x \sim q_\phi(x)$

Binary discriminator $\rightarrow$ $C = 0/1$

Dynamic noises help optimization, by gradually increasing the difficulty of the two-class discrimination task

Introduce a dynamic noise distribution, simultaneously optimized to be close to the data distribution, so that small ν could be used.

$$
\begin{cases}
\min_{\phi} KL\left(p_0 || q_\phi\right) \\
\max_{\theta,\zeta} E_{x \sim \alpha p_0(x) + (1-\alpha) q_\phi(x)}[\log P(C=0|x)] + \nu E_{x \sim q_\phi(x)}[\log P(C=1|x)]
\end{cases}
$$

Using interpolation to alleviate overfitting

Consistency under infinite amount of data and infinite capacity of p_θ and q_ϕ

Bin Wang, Zhijian Ou. Improved training of neural trans-dimensional random field language models with dynamic noise-contrastive estimation. SLT, 2018.

Content

I. Basics for EBMs (45 min)

1. Probabilistic graphical modeling (PGM) framework and EBM model examples
2. Learning EBMs by Monte Carlo methods
3. Learning EBMs by noise-contrastive estimation (NCE)

II. EBMs for language modeling (45 min)

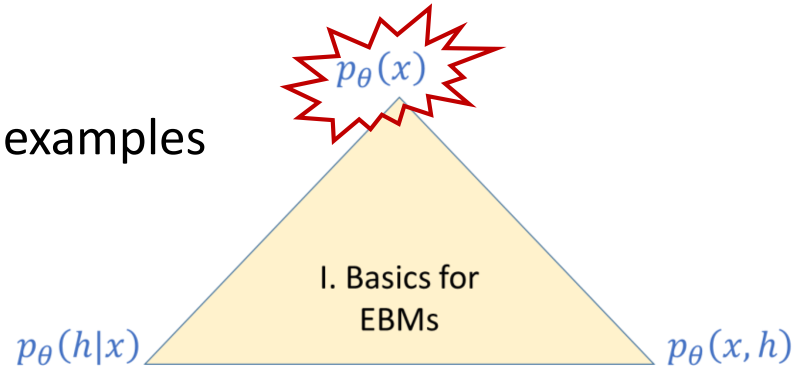
1. Trans-dimensional random field (TRF) LMs for speech recognition
2. Residual energy-based models for text generation
3. Electric: an energy-based cloze model for representation learning over text

III. EBMs for natural language labeling and speech recognition (45 min)

1. CRFs as conditional EBMs
2. CRFs for speech recognition
3. CRFs for sequence labeling in NLP

IV. EBMs for semi-supervised natural language labeling (45 min)

1. Upgrading EBMs to Joint EBMs (JEMs) for fixed-dimensional data
2. Upgrading CRFs to Joint random fields (JRFs) for sequential data
3. JRFs for semi-supervised natural language labeling



Section Content

1. Motivation
2. Related work
3. Method: **RF LMs**
4. Experiments
5. Conclusion

- Bin Wang, Zhijian Ou, Zhiqiang Tan. Trans-dimensional Random Fields for Language Modeling. ACL Long Paper, 2015.
- Bin Wang, Zhijian Ou, Zhiqiang Tan. Learning Trans-dimensional Random Fields with Applications to Language Modeling. TPAMI, 2018.
- Bin Wang, Zhijian Ou. Language modeling with neural trans-dimensional random fields. ASRU, 2017.
- Bin Wang, Zhijian Ou. Learning neural trans-dimensional random field language models with noise-contrastive estimation. ICASSP, 2018.
- Bin Wang, Zhijian Ou. Improved training of neural trans-dimensional random field language models with dynamic noise-contrastive estimation. SLT, 2018.
- Silin Gao, Zhijian Ou, Wei Yang, Huifang Xu. Integrating discrete and neural features via mixed-feature trans-dimensional random field language models. ICASSP, 2020. [Oral]

N-gram LMs

- Language modeling (LM) is to determine the joint probability of a sentence, i.e. a word sequence.
- Dominant: Directed modeling approach

Current word

All previous words/history

$$p(x_1, x_2, \cdots, x_l) = \prod_{i=1}^{l} p(x_i | x_1, \cdots, x_{i-1})$$

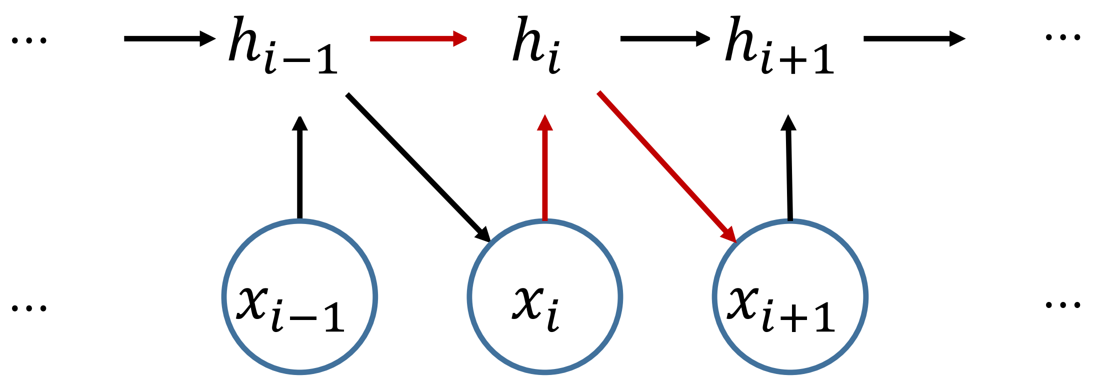
Previous $n-1$ words

$$\approx \prod_{i=1}^{l} \underline{p(x_i | x_{i-n+1}, \cdots, x_{i-1})}$$

- Using Markov assumption leads to the N-gram LMs
  – One of the state-of-the-art LMs

Chen and Goodman. "An empirical study of smoothing techniques for language modeling", Computer Speech & Language, 1999.

Recurrent Neural Nets (RNNs)/LSTM/Transformer LMs



$$p(x_i|x_1,\cdots,x_{i-1}) \approx p\left(x_i|h_{i-1}(x_1,\cdots,x_{i-1})\right) \approx \frac{h_{i-1}^T w_k}{\sum_{k=1}^V h_{i-1}^T w_k}$$

☹.1 Computational expensive in both training and testing [1]

e.g. $V = 10^4 \sim 10^6, w_k \in \mathbb{R}^{250\sim1024}$

☹.2 As directed sequential model /Auto-regressive model, potentially suffers from Exposure Bias and Label Bias

[1] Partly alleviated by using un-normalized models (e.g., through NCE) or a small set of tokens (e.g., BPE).

Trans-dimensional Random Field (TRF) LM: motivation

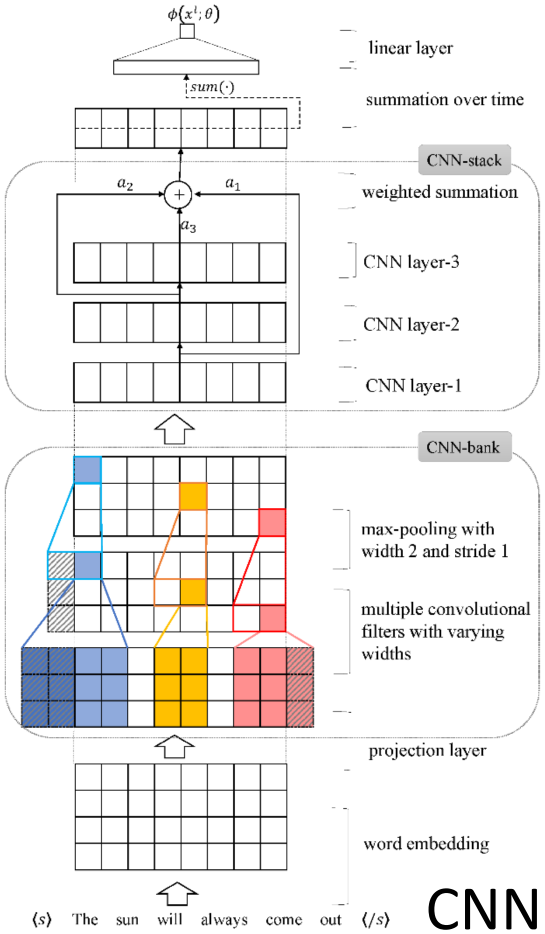
☺.1 Avoid local normalization

$$p_\theta(x^l) = \frac{1}{Z_l(\theta)} e^{u_\theta(x^l)}, x^l \triangleq x_1, x_2, \cdots, x_l$$

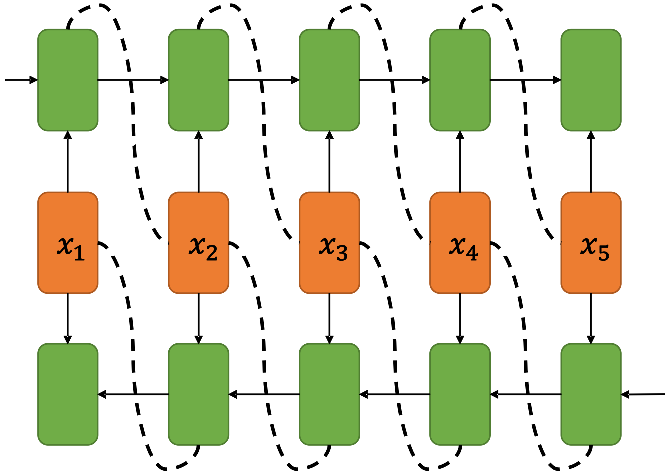
☺.2 Flexible

Type	Features
w	$(w_{-3}w_{-2}w_{-1}w_0)(w_{-2}w_{-1}w_0)(w_{-1}w_0)(w_0)$
c	$(c_{-3}c_{-2}c_{-1}c_0)(c_{-2}c_{-1}c_0)(c_{-1}c_0)(c_0)$
ws	$(w_{-3}w_0)(w_{-3}w_{-2}w_0)(w_{-3}w_{-1}w_0)(w_{-2}w_0)$
cs	$(c_{-3}c_0)(c_{-3}c_{-2}c_0)(c_{-3}c_{-1}c_0)(c_{-2}c_0)$
wsh	$(w_{-4}w_0)\ (w_{-5}w_0)$
csh	$(c_{-4}c_0)\ (c_{-5}c_0)$
cpw	$(c_{-3}c_{-2}c_{-1}w_0)\ (c_{-2}c_{-1}w_0)(c_{-1}w_0)$
tied	$(c_{-9:-6}, c_0)\ (w_{-9:-6}, w_0)$

Discrete features



CNN features



BLSTM features

Trans-dimensional random fields (TRFs): model

- Assume the sentences of length $\boldsymbol{l}$ are distributed as follows:

$$p_l(x^l;\lambda) = \frac{1}{Z_l(\lambda)} e^{\lambda^T f(x^l)} \qquad l = 1, \dots, l_{max}$$

$x^l \triangleq x_1, x_2, \cdots, x_l$ is a word sequence with length l;

$f(x^l) = \left(f_1(x^l), \dots, f_d(x^l)\right)^T$ is the feature vector;

$\lambda = (\lambda_1, \dots, \lambda_d)^T$ is the parameter vector;

$Z_l(\lambda) = \sum_{x^l} e^{\lambda^T f(x^l)}$ is the normalization constant.

Needed to be estimated

- Assume length $\boldsymbol{l}$ is associated with priori probability $\boldsymbol{\pi_l}$. Therefore the pair $(\boldsymbol{l}, \boldsymbol{x^l})$ is jointly distributed as:

$$p(l, x^l; \lambda) = \pi_l \cdot p_l(x^l; \lambda)$$

Feature definition

$$p_l(x^l;\lambda) = \frac{1}{Z_l(\lambda)} e^{\lambda^T f(x^l)}$$

- $f_i(x^l)$ returns the count of a specific phrase observed in the input sentence x^l

  $\underline{x^l = he\ is\ a\ teacher\ and\ he\ is\ also\ a\ good\ father.}$

  $f_{he\ is}(x^l)$ = count of "$he\ is$" observed in $x^l = 2$
  $f_{a\ teacher}(x^l)$ = count of "$a\ teacher$" observed in $x^l = 1$
  $f_{she\ is}(x^l)$ = count of "$she\ is$" observed in $x^l = 0$
  … …

- For example, n-grams and skip n-grams (tied or not) of orders ranging from 1 to 10, observed in the training set are added to the features.

Review the development of TRF LMs

ACL-2015 TPAMI-2018	• Discrete features • Augmented stochastic approximation (**AugSA**) for model training
ASRU-2017	• Potential function as a deep CNN. • Model training by **AugSA plus JSA** (joint stochastic approximation)
ICASSP-2018	• Potential function in the form of **exponential tilting** (revisited in residual EBMs) • Use LSTM on top of CNN • **NCE** is introduced to train TRF LMs
SLT-2018	• Simplify the potential definition by using only Bidirectional LSTM • Propose **Dynamic NCE** for improved model training
ICASSP-2020	• **Mixed-feature** TRFs, by integrating discrete and neural features

WSME - Introduction

- Whole-sentence maximum entropy (WSME)
  - Rosenfeld, Chen, Zhu. “Whole-sentence exponential language models: a vehicle for linguistic-statistical integration”. Computer Speech & Language, 2001.

$$p(x;\lambda) = \frac{1}{Z(\lambda)} \exp[\lambda^T f(x)]$$

- The empirical results of previous WSME models are not satisfactory
  - After incorporating lexical and syntactic information, 1% and 0.4% respectively in perplexity and in WER is reported for the resulting WSME (Rosenfeld et al., 2001).
  - Amaya and Benedi. “Improvement of a whole sentence maximum entropy language model using grammatical features”, ACL 2001.
  - Ruokolainen, Alumae, Dobrinkat. “Using dependency grammar features in whole sentence maximum entropy language model for speech recognition”. HLT 2010.

RFLMs vs WSME

- Whole-sentence maximum entropy (WSME)

$$p(l, x^l; \lambda) = \frac{1}{Z(\lambda)} \exp[\lambda^T f(x^l)], \qquad x \triangleq (l, x^l), \qquad x^l \triangleq (x_1, x_2, \cdots, x_l)$$

Essentially a mixture distribution with unknown weights (differ from each other greatly, 10^{40}) !

Poor sampling → poor estimate of gradient → poor fitting

$$p(l, x^l; \lambda) = \frac{Z_l(\lambda)}{Z(\lambda)} \cdot \frac{1}{Z_l(\lambda)} \cdot \exp[\lambda^T f(x^l)], \; Z_l(\lambda) = \sum_{x^l} \exp[\lambda^T f(x^l)]$$

RFLMs vs WSME

- Whole-sentence maximum entropy (WSME)

$$p(l, x^l; \lambda) = \frac{1}{Z(\lambda)} \exp[\lambda^T f(x^l)], \qquad x \triangleq (l, x^l), \qquad x^l \triangleq (x_1, x_2, \cdots, x_l)$$

Essentially a mixture distribution with unknown weights (differ from each other greatly, 10^{40}) !

Poor sampling → poor estimate of gradient → poor fitting

$$p(l, x^l; \lambda) = \frac{Z_l(\lambda)}{Z(\lambda)} \cdot \frac{1}{Z_l(\lambda)} \cdot \exp[\lambda^T f(x^l)], \; Z_l(\lambda) = \sum_{x^l} \exp[\lambda^T f(x^l)]$$

- We propose a trans-dimensional RF model

$$p(l, x^l; \lambda) = \pi_l \cdot \frac{1}{Z_l(\lambda)} \cdot \exp[\lambda^T f(x^l)], \qquad l = 1, \cdots, m$$

Empirical length probabilities in the training data

Serve as a control device to improve sampling from multiple distributions !

Apply SA to RFLM training

- The trans-dimensional RF model $$p(l, x^l; \lambda) = \pi_l \cdot \frac{1}{Z_l(\lambda)} \cdot \exp\left[\lambda^T f(x^l)\right] \quad (1)$$

$$E_{\tilde{p}(x)}[f_i(x)] - E_{p(x;\lambda)}[f_i(x)] = 0, \qquad x \triangleq (l, x^l)$$

- Consider the joint distribution of the pair (l, x^l) $$p(l, x^l; \lambda, \zeta) \propto \pi_l \cdot \frac{1}{e^{\zeta_l}} \cdot \exp\left[\lambda^T f(x^l)\right] \quad (2)$$

where ζ_l is hypothesized values of the true $\zeta_l^*(\lambda) = logZ_l(\lambda)$

The marginal probability of length l is: $p(l; \lambda, \zeta) = \frac{\pi_l e^{-\zeta_l + \zeta_l^*(\lambda)}}{\sum_j \pi_l e^{-\zeta_j + \zeta_j^*(\lambda)}}$

- SA is used to find $\zeta_l^* = \zeta_l^*(\lambda^*)$ and λ^* that solves

$$\begin{cases} \pi_l = p(l; \lambda, \zeta), \qquad l = 1, \cdots, m \\ 0 = E_{\tilde{p}(x)}[f_i(x)] - E_{p(l, x^l; \lambda, \zeta)}[f_i(x)] \end{cases}$$

Zhiqiang Tan. 2015. Optimally adjusted mixture sampling and locally weighted histogram. In Technical Report, Dept. of Statistics, Rutgers Univ.

RFLMs – Breakthrough in training (1)

- Propose Augmented Stochastic Approximation (AugSA) Training Algorithm
  - Simultaneously updates the model parameters and normalizing constants

Input: training set

set initial values $\lambda^{(0)} = (0, \ldots, 0)^T$ and
$\zeta^{(0)} = \zeta^*(\lambda^{(0)}) - \zeta_1^*(\lambda^{(0)})$

for $t = 1, 2, \ldots, t_{max}$ **do**

  set $B^{(t)} = \emptyset$

  set $(j^{(t,0)}, x^{(t,0)}) = (j^{(t-1,K)}, x^{(t-1,K)})$

  Step I: MCMC sampling

  for $k = 1 \rightarrow K$ **do**

    $(j^{(t,k)}, x^{(t,k)}) = \text{sample}((j^{(t,k-1)}, x^{(t,k-1)}))$ (See Tab.1)

    set $B^{(t)} = B^{(t)} \cup \{(j^{(t,k)}, x^{(t,k)})\}$

  end for

  Step II: SA updating

  Compute $\lambda^{(t)}$ based on (16)

  Compute $\zeta^{(t)}$ based on (17) and (18)

end for

Fig. 1. Augmented stochastic approximation (AugSA)

Bin Wang, Zhijian Ou, Zhiqiang Tan. Learning Trans-dimensional Random Fields with Applications to Language Modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2018.

RFLMs – Breakthrough in training (2)

- Propose Trans-dimensional mixture sampling
  - Sampling from $p(l, x^l; \lambda, \zeta)$, a mixture of RFs on subspaces of different dimensions.
  - Formally like RJ-MCMC (Green, 1995).

function $\textsc{Sampling}((L^{(t-1)}, X^{(t-1)}))$
  set $k = L^{(t-1)}$
  set $L^{(t)} = k$
  set $X^{(t)} = X^{(t-1)}$
  Stage I: Local jump
  generate $j \sim \Gamma(k, \cdot)$
  if $j = k + 1$ **then**
    generate $Y \sim g_{k+1}(y|X^{(t-1)})$ (equ.24)
    set $L^{(t)} = j$ and $X^{(t)} = \{X^{(t-1)}, Y\}$ with probability equ.22
  end if
  if $j = k - 1$ **then**
    set $L^{(t)} = j$ and $X^{(t)} = X^{(t-1)}_{1:k-1}$ with probability equ.23
  end if
  Stage II: Markov move
  for $i = 1 \to L^{(t)}$ **do**
    $a \sim p(L^{(t)}, \{X^{(t)}_{1:i-1}, \cdot, X^{(t)}_{i+1:L^{(t)}}\}; \Lambda, \zeta)$
    $X^{(t)}_i \leftarrow a$
  end for
  return $(L^{(t)}, X^{(t)})$
end function

Motivation: Integrating discrete and neural features

- Language models using discrete features (N-gram LMs, Discrete TRF LMs)
  - Mainly capture local lower-order interactions between words
  - Better suited to handling symbolic knowledges
- Language models using neural features (LSTM LMs, Neural TRF LMs)
  - Able to learn higher-order interactions between words
  - Good at learning smoothed regularities due to word embeddings
- Interpolation of LMs[1, 2]: usually achieves further improvement
  - Discrete and neural features have complementary strength. ☺
  - Two-step model training is sub-optimal. ☹

[1]Xie Chen, Xunying Liu, Yu Wang, Anton Ragni, Jeremy HM Wong, and Mark JF Gales, "Exploiting future word contexts in neural network language models for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1444–1454, 2019.

[2]Bin Wang, Zhijian Ou, Yong He, and Akinori Kawamura, "Model interpolation with trans-dimensional random field language models for speech recognition," *arXiv preprint arXiv:1603.09170*, 2016.

Mixed-feature TRF LMs: Definition

- Mixed-feature TRF LMs:
  - $p(l, x^l; \eta) = \frac{\pi_l}{Z_l(\eta)} e^{V(x^l, \eta)}, \; V(x^l, \eta) = \lambda^T f(x^l) + \phi(x^l; \theta), \; \eta = (\lambda, \theta)$

Greater flexibility in potential definition than examples shown below!

Discrete n-gram features, with parameter λ:

$$f(x^l) = (f_1(x^l), f_2(x^l), \cdots, f_d(x^l))$$

d: the total number of types of n-grams

$$f_k(x^l) = c$$

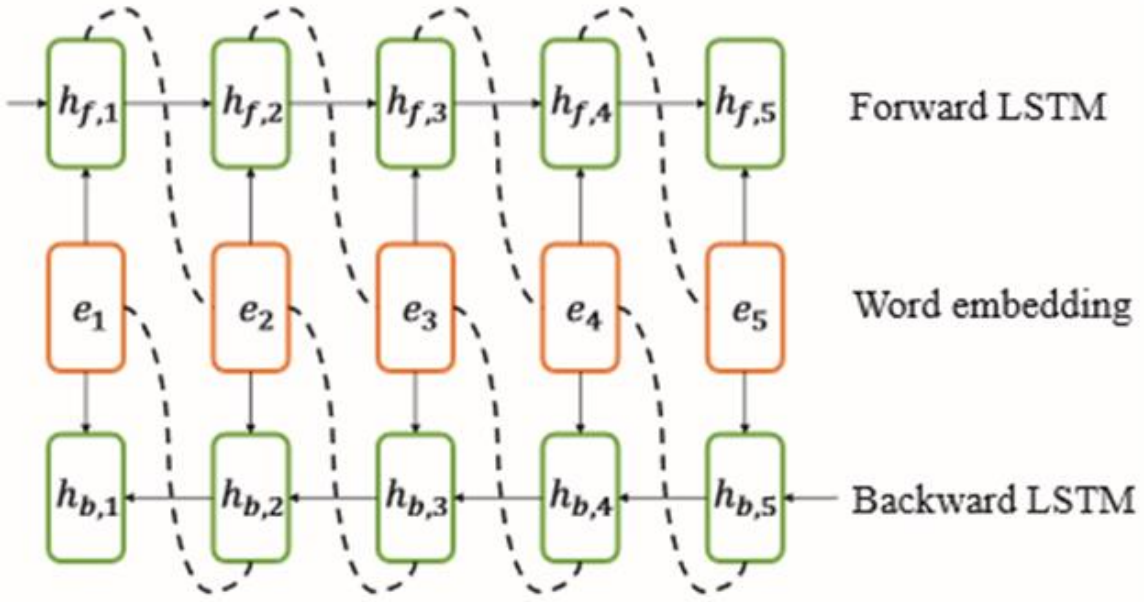
where c is the count of the kth n-gram type in x^l

$\underline{x^l = \textit{he is a teacher and he is also a good father.}}$

$f_{he\ is}(x^l) =$ count of "*he is*" in $x^l = 2$

$f_{a\ teacher}(x^l) =$ count of "*a teacher*" in $x^l = 1$

Neural network features, with parameter θ



$$\phi(x^l; \theta) = \sum_{i=1}^{l-1} h_{f,i}{}^T e_{i+1} + \sum_{i=2}^{l} h_{b,i}{}^T e_{i-1}$$

Mixed-feature TRF LMs: Training

- Treat $\log Z_l(\eta)$ as a parameter ζ_l and rewrite

$$p(l, x^l; \eta) = \frac{\pi_l}{Z_l(\eta)} e^{V(x^l,\eta)} \longrightarrow p(x; \xi) = \pi_l e^{V(x^l,\eta)-\zeta_l}, x = (l, x^l), \xi = (\eta, \zeta)$$

- Introduce a **noise distribution** $q_\phi(x)$, and consider a binary classification

$q_\phi(x) = \pi_l \times NNLM(x^l)$, $NNLM$ could be LSTM/Transformer autoregressive LM

$$P(C = 0|x) = \frac{p(x; \xi)}{p(x; \xi) + \nu q_\phi(x)}, where\ \ \nu = \frac{P(C = 1)}{P(C = 0)}$$

$$P(C = 1|x) = 1 - P(C = 0|x)$$

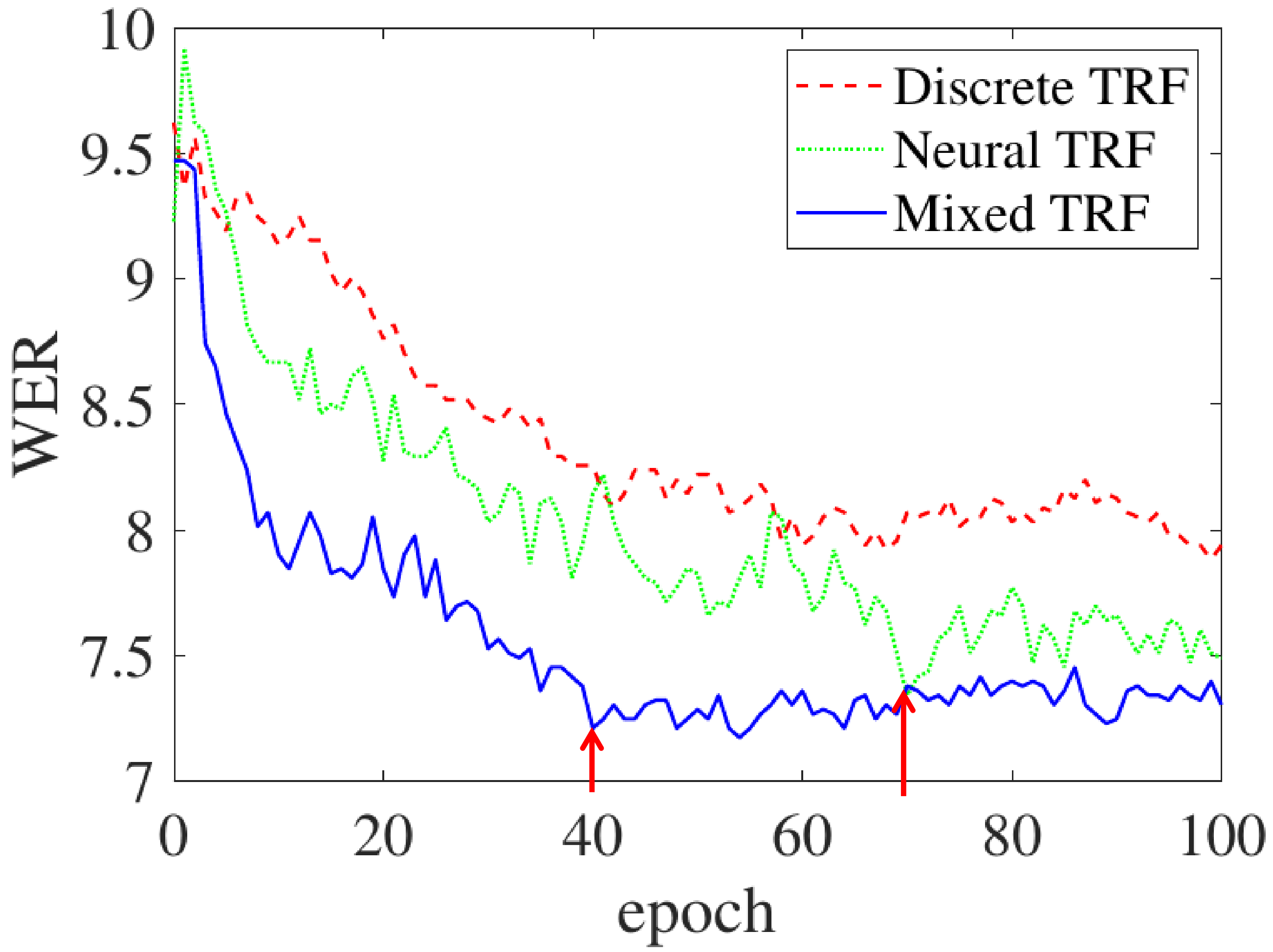
- Noise Contrastive Estimation (NCE):

$$\max_\xi E_{x\sim p_0(x)}[\log P(C = 0|x)] + \nu E_{x\sim q_\phi(x)}[\log P(C = 1|x)]$$

Reliable NCE needs a large $\nu \approx 20$; Dynamic-NCE works well with $\nu = 1$

Experiments: PTB dataset

WER curves of the three TRF LMs during the first 100 training epochs:



- Mixed TRF converges faster than the state-of-the-art Neural TRF, using only 58% training epochs.

☺ The discrete features in Mixed TRF lower the non-convexity of the optimal problem, and reduce the amount of patterns for neural features to capture.

On Google one-billion word benchmark

Training: Google One-Billion word benchmark, 0.8 billion words, 568K vocabulary
Testing: WSJ’92 test data, 330 utterances, rescoring 1000-best lists

Model	WER (%)	#Param (M)	Training time	Inference Time
KN5	6.13	133	2.5 h (1 CPU)	0.491 s (1 CPU)
LSTM-2x1024	5.55	191	144 h (2 GPUs)	0.909 s (2 GPUs)
discrete-TRF basic	6.04	102	131 h (8 cores and 2 GPUs)	0.022 s (1 CPU)
neural-TRF	5.47	114	336 h (2 GPUs)	0.017 s (2 GPUs)
mix-TRF	5.28 5%	216 33%	297 h (8 cores and 2 GPUs)	0.024 s (1 core and 2 GPUs) 38x
LSTM-2x1024+KN5	5.38	324		

Open-source LM toolkit

https://github.com/thu-spmi/SPMILM

Section Conclusion

- Language models play an important role for ASR and more applications!
- Random Field language models
  - Avoid local normalization
  - Being flexible to integrate rich features (both discrete and neural)
  - Overcome “label bias” and “Exposure bias”
- More related work
  - Residual energy-based models for text generation
  - Electric: an energy-based cloze model for representation learning over text

- Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. Residual energy-based models for text generation, ICLR 2020.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Pre-training transformers as energy-based cloze models, EMNLP 2020.

Content

I. Basics for EBMs (45 min)

1. Probabilistic graphical modeling (PGM) framework and EBM model examples (classic & modern)
2. Learning EBMs by Monte Carlo methods
3. Learning EBMs by noise-contrastive estimation (NCE)

II. EBMs for language modeling (45 min)

1. Trans-dimensional random field (TRF) LMs for speech recognition
2. Residual energy-based models for text generation
3. Electric: an energy-based cloze model for representation learning over text

III. EBMs for speech recognition and natural language labeling (45 min)

1. CRFs as conditional EBMs
2. CRFs for speech recognition
3. CRFs for sequence labeling in NLP

IV. EBMs for semi-supervised natural language labeling (45 min)

1. Upgrading EBMs to Joint EBMs (JEMs) for fixed-dimensional data
2. Upgrading CRFs to Joint random fields (JRFs) for sequential data
3. JRFs for semi-supervised natural language labeling

Motivation

- Text generation is ubiquitous in many NLP tasks, from summarization, to dialogue and machine translation.
- Locally normalized LMs are plagued by exposure bias and label bias.
- EBMs are ideal for modeling text as … , but seldom explored
  - they can score the whole input at once,
  - they are not prone to exposure bias and label bias ,
  - they may enable generation of large chunks of text, which should help improve coherency (e.g., resampling from the large set of candidates produced by the base locally normalized LM).

Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. Residual energy-based models for text generation, ICLR 2020.

Model - Residual EBMs

- Deng et al. investigate an EBM trained on the residual of a pretrained autoregressive LM [a, b]

$$P_\theta(x) \propto P_{LM}(x) exp[-E_\theta(x)]$$

  - $P_{LM}(x)$: a pretrained autoregressive LM and fixed
  - $E_\theta(x)$: the residual energy function parameterized by θ
  - Call P_θ the joint model
- Consider the problem of conditional generation of discrete sequences

$$P_\theta(x_{p+1}, \cdots, x_T | x_1, \cdots, x_p) = \frac{P_{LM}(x_{p+1}, \cdots, x_T | x_1, \cdots, x_p) \exp(-E_\theta(x_1, \cdots, x_T))}{Z_\theta(x_1, \cdots, x_p)} \quad (2)$$

  - Given a prefix $x_1, \cdots, x_p$, generate a sequence of total length $T > p$

a. Bin Wang, Zhijian Ou. Learning neural trans-dimensional random field language models with noise-contrastive estimation. ICASSP, 2018.
b. Tetiana Parshakova, Jean-Marc Andreoli, and Marc Dymetman. Global autoregressive models for data-efficient sequence learning. CoNLL, 2019.

Training of Residual EBMs

- Trained using NCE, and more specifically its conditional version
  - Using $P_{LM}(x)$ as the noise distribution
  - $E_\theta(x)$ initialized with BERT/RoBERTa; in the final layer we project the mean-pooled hidden states to a scalar energy value.
  - x_+ : positive sentence taken from the training data
  - x_- : negative sentence drawn from $P_{LM}(x)$ (for a given ground truth prefix)

$$\max \mathbb{E}_{x_+ \sim P_{data}} \log \frac{1}{1+\exp(E_\theta(x_+))} + \mathbb{E}_{x_- \sim P_{LM}} \log \frac{1}{1+\exp(-E_\theta(x_-))}$$

In all our experiments we use a prefix of size 120 tokens and we generate the following 40 tokens; with the notation of Eq. 2, $p = 120$ and $T = 160$. For training the joint models, for efficiency we generated 16/128 samples per prefix for CC-News/Book Corpus offline, and sample uniformly from those samples at training time.

Generation by Residual EBMs

- Use self-normalizing importance sampling

  a) Sampling from the proposal - the auto-regressive language model $P_{LM}(x)$

  b) Resampling according to the energy function.

Algorithm 1: Top-k Joint Sampling

Input: number of samples n drawn from P_{LM}, value of k in top-k

`// Get a set of samples from` P_{LM}

sample n samples $\{x^1, \cdots, x^n\}$ from P_{LM} with top-k sampling

calculate energies $s^i = E_\theta(x^i)$ for each $x^i \in \{x^1, \cdots, x^n\}$

`// Resample from the set of LM samples`

sample $x = x^i$ with probability $\frac{\exp(-s^i)}{\sum_{j=1}^{n} \exp(-s^j)}$

return x

Evaluation

PPL = exp(- Log Likelihood per token)
The average number of tokens the model needs to guesst at every time.

- PPL (perplexity) evaluation needs to estimate the partition function

$$Z_\theta = \sum_x P_{LM}(x) exp[-E_\theta(x)] = \mathbb{E}_{x \sim P_{LM}(x)} \{exp[-E_\theta(x)]\}$$

- Two estimators for the lower and upper bounds of the partition function

Theorem 2. *Denote T_n as the empirical estimate of $\log \mathbb{E}_{x \sim P_{LM}} \exp(-E(x))$ with n samples $x_i \sim P_{LM}(i = 1, \cdots, n)$: $T_n = \log \frac{1}{n} \sum_{i=1}^{n} \exp(-E(x_i))$, then $\forall \epsilon > 0$, $\exists N > 0$ such that $\forall n > N$ we have*

$$Z_\theta - \epsilon < \mathbb{E}[T_n] < Z_\theta < \mathbb{E}[(2n-1)T_n - 2(n-1)T_{n-1}] < Z_\theta + \epsilon \tag{4}$$

- Per-step probabilities

$$P(x_t|x_{<t}) = P_{LM}(x_t|x_{<t}) \frac{\mathbb{E}_{x'_{t+1}, \cdots, x'_T \sim P_{LM}(\cdot|x_{\leq t})}[\exp(-E_\theta(x_{\leq t}, x'_{t+1}, \cdots, x'_T))]}{\mathbb{E}_{x'_t, \cdots, x'_T \sim P_{LM}(\cdot|x_{\leq t-1})}[\exp(-E_\theta(x_{\leq t-1}, x'_t, \cdots, x'_T))]}. \tag{5}$$

  - The basic PLM distribution is adjusted

A central technical contribution of this paper!

Experiments

- On two large datasets, residual EBMs have demonstrated improved generation ability against very strong auto-regressive baselines, both in terms of estimated perplexity and through human evaluation.

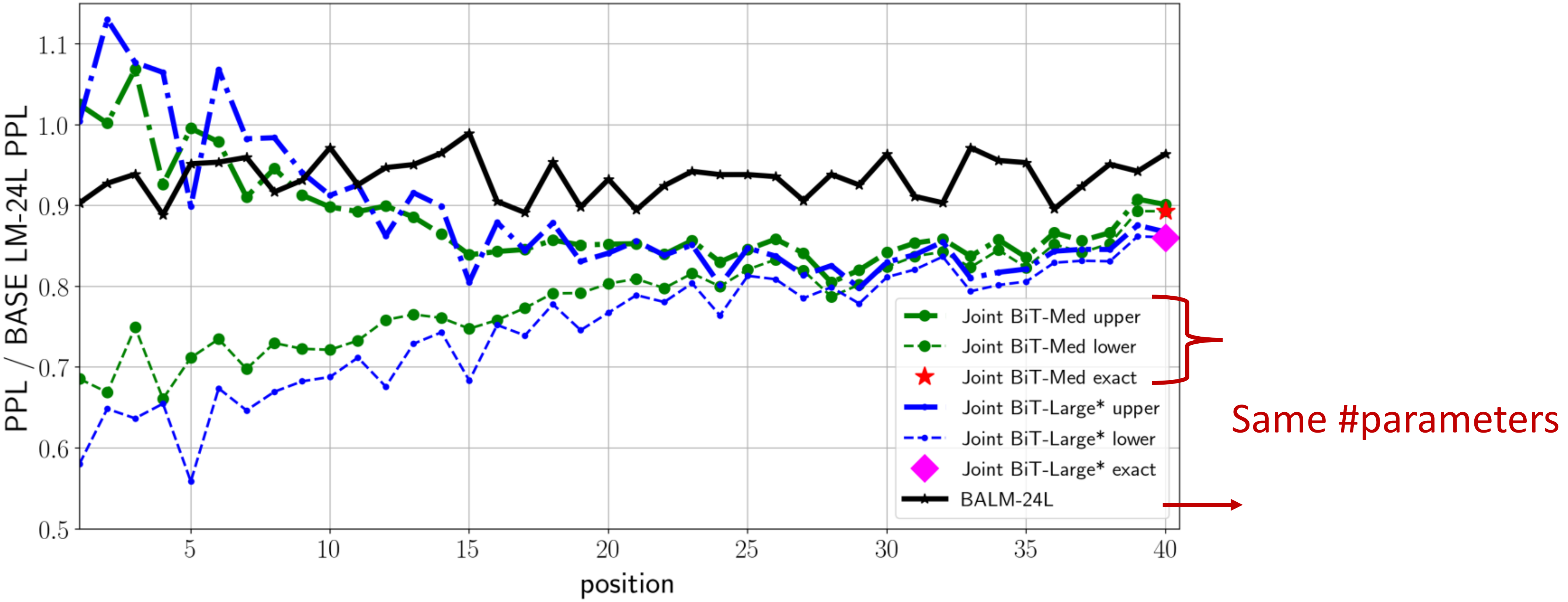


Figure 1: Perplexity gain of **JOINT BIT-MED** and JOINT BIT-LARGE∗ (using BASE LM-24L) at each position relative to **BASE LM-24L** on the test set of CC-News.

Content

I. Basics for EBMs (45 min)

1. Probabilistic graphical modeling (PGM) framework and EBM model examples (classic & modern)
2. Learning EBMs by Monte Carlo methods
3. Learning EBMs by noise-contrastive estimation (NCE)

II. EBMs for language modeling (45 min)

1. Trans-dimensional random field (TRF) LMs for speech recognition
2. Residual energy-based models for text generation
3. Electric: an energy-based cloze model for representation learning over text

III. EBMs for speech recognition and natural language labeling (45 min)

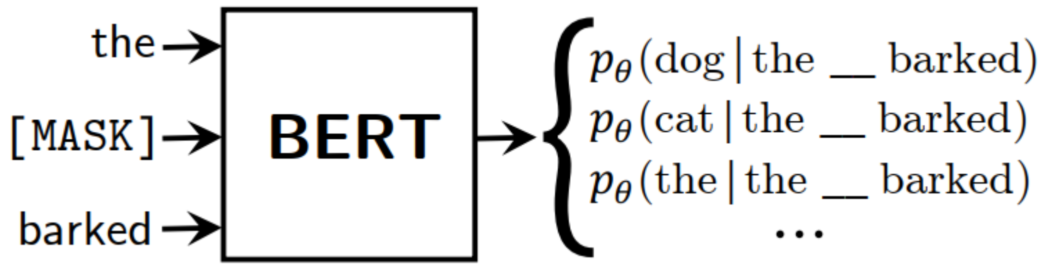
1. CRFs as conditional EBMs
2. CRFs for speech recognition
3. CRFs for sequence labeling in NLP

IV. EBMs for semi-supervised natural language labeling (45 min)

1. Upgrading EBMs to Joint EBMs (JEMs) for fixed-dimensional data
2. Upgrading CRFs to Joint random fields (JRFs) for sequential data
3. JRFs for semi-supervised natural language labeling

Motivation

- The cloze task of predicting the identity of a token given its surrounding context has proven highly effective for representation learning over text.
- BERT implements the cloze task by replacing input tokens with [MASK], but
  - Drawback in efficiency (only 15% of tokens are masked out at a time)
  - A pre-train/fine-tune mismatch where BERT sees [MASK] tokens in training but not in fine-tuning
  - Less concerned with producing (pseudo-)likelihood score for text



Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Pre-training transformers as energy-based cloze models, EMNLP 2020.

The Electric Model

- Learn $p_{data}(x_t|x_{\setminus t})$ of a token x_t occurring in the surrounding context

$$x_{\setminus t} = [x_1, \cdots, x_{t-1}, x_{t+1}, \cdots, x_n]$$

  - Maps the unmasked input $x = [x_1, \cdots, x_n]$ into contextualized vector representations $h(x) = [h_1, \cdots, h_n]$ using a Transformer net.
  - Define a conditional EBM, with a learned weight vector w

$$P_\theta(x_t|x_{\setminus t}) \propto exp[-w^T h_t], \qquad \text{where energy function } E(x)_t \triangleq w^T h_t$$

BERT

- Mask 15% of the input sequence
- Calculate the distribution over the vocabulary using a softmax layer for each masked position

Electric

- Using unmasked input
- Likelihood scores $P_\theta(x_t|x_{\setminus t})$ can be computed simultaneously for all positions rather than only for a small masked-out subset.

Training of the Electric Model

- Trained using NCE, and more specifically its conditional version
  - Define the un-normalized output

$$\hat{p}_\theta(x_t|x_{\setminus t}) = exp[-w^T h_t]$$

  - A binary classifier is trained to distinguish positive x_t vs negative $\hat{x}_t$, with k negatives sampled for every n positive data points.

Formally, the NCE loss $\mathcal{L}(\theta)$ is

$$n \cdot \mathop{\mathbb{E}}_{\boldsymbol{x},t}\left[-\log \frac{n \cdot \hat{p}_\theta(x_t|\boldsymbol{x}_{\setminus t})}{n \cdot \hat{p}_\theta(x_t|\boldsymbol{x}_{\setminus t}) + k \cdot q(x_t|\boldsymbol{x}_{\setminus t})}\right] + k \cdot \mathop{\mathbb{E}}_{\substack{\boldsymbol{x},t \\ \hat{x}_t \sim q}}\left[-\log \frac{k \cdot q(\hat{x}_t|\boldsymbol{x}_{\setminus t})}{n \cdot \hat{p}_\theta(\hat{x}_t|\boldsymbol{x}_{\setminus t}) + k \cdot q(\hat{x}_t|\boldsymbol{x}_{\setminus t})}\right]$$

  - Noise distribution: a two-tower cloze model

$$\overrightarrow{\boldsymbol{h}} = T_{\text{LTR}}(\boldsymbol{x}), \quad \overleftarrow{\boldsymbol{h}} = T_{\text{RTL}}(\boldsymbol{x})$$

$$q(x_t|\boldsymbol{x}_{\setminus t}) = \text{softmax}(\boldsymbol{W}[\overrightarrow{\boldsymbol{h}}_{t-1}, \overleftarrow{\boldsymbol{h}}_{t+1}])_{x_t}$$

Two causal transformers
(left-to-right & right-to-left)

Training of the Electric Model

$$n \cdot \mathop{\mathbb{E}}_{\boldsymbol{x},t}\left[-\log \frac{n \cdot \hat{p}_\theta(x_t|\boldsymbol{x}_{\setminus t})}{n \cdot \hat{p}_\theta(x_t|\boldsymbol{x}_{\setminus t}) + k \cdot q(x_t|\boldsymbol{x}_{\setminus t})}\right] + k \cdot \mathop{\mathbb{E}}_{\substack{\boldsymbol{x},t \\ \hat{x}_t \sim q}}\left[-\log \frac{k \cdot q(\hat{x}_t|\boldsymbol{x}_{\setminus t})}{n \cdot \hat{p}_\theta(\hat{x}_t|\boldsymbol{x}_{\setminus t}) + k \cdot q(\hat{x}_t|\boldsymbol{x}_{\setminus t})}\right]$$

- Naïve calculation is expensive: $k+1$ forward passes through the transformer to compute the $\hat{p}_\theta$s (once for the positive samples $x_t|x_{\setminus t}$ and once for each negative sample $\hat{x}_t|x_{\setminus t}$)

Algorithm 2 Efficient NCE loss estimation

Given: Input sequence $\boldsymbol{x}$, number of negative samples k, noise distribution q, model $\hat{p}_\theta$.

1. Pick k unique random positions $R = \{r_1, ..., r_k\}$ where each r_i is $1 \leq r_i \leq n$.

2. Replace the k random positions with negative samples: $\hat{x}_i \sim q(x_i|\boldsymbol{x}_{\setminus i})$ for $i \in R$, $\boldsymbol{x}^{\text{noised}} = \text{REPLACE}(\hat{\boldsymbol{x}}, R, \hat{X})$.

3. For each position $t = 1$ to n: add to the loss

$$-\log \frac{k \cdot q(\hat{x}_t|\boldsymbol{x}_{\setminus t})}{(n-k) \cdot \hat{p}_\theta(\hat{x}_t|\boldsymbol{x}^{\text{noised}}_{\setminus t}) + k \cdot q(\hat{x}_t|\boldsymbol{x}_{\setminus t})} \quad \text{if } t \in R$$

$$-\log \frac{(n-k) \cdot \hat{p}_\theta(x_t|\boldsymbol{x}^{\text{noised}}_{\setminus t})}{(n-k) \cdot \hat{p}_\theta(x_t|\boldsymbol{x}^{\text{noised}}_{\setminus t}) + k \cdot q(x_t|\boldsymbol{x}_{\setminus t})} \quad \text{otherwise}$$

Trick:

- Simultaneously choose $k = [0.15n]$ random positions
- One pass through the transformer over x^{noised}
- Assume $\hat{p}_\theta(\cdot\,|x_{\setminus t}) \approx \hat{p}_\theta(\cdot\,|x^{\text{noised}}_{\setminus t})$

Experiments

Model	MultiNLI	SQuAD 2.0	GLUE Avg.
BERT	84.3	73.7	82.2
XLNet	85.8	78.5	–
ELECTRA	86.2	80.5	85.1
Electric	85.7	80.1	84.1

Table 1: Dev-set scores of pre-trained models on downstream tasks. To provide direct comparisons, we only show base-sized models pre-trained on WikiBooks.

Model	Pre-train Data	Clean WER	Other WER	Runtime
None	–	7.26	20.37	0
BERT	WikiBooks	5.41	17.41	n
Electric	WikiBooks	5.65	17.42	1
GPT-2	OWT	5.64	17.60	1
TwoTower	OWT*	5.32	17.25	1
ELECTRA-TT	OWT*	5.22	17.01	1
Electric	OWT*	5.18	16.93	1

Table 2: Test-set word error rates on LibriSpeech after rescoring with base-sized models. None, GPT-2, and BERT results are from Salazar et al. (2020). Runtime is measured in passes through the transformer. "Clean" and "other" are easier and harder splits of the data. *We use a public re-implementation of OpenWebText.

- Electric slightly underperforms ELECTRA on downstream tasks, better than BERT.

- Pseudo-log-likelihood (PLL) scores, are used to re-rank the outputs of a speech recognition system, perform better and faster than masked models

$$\mathrm{PLL}(\boldsymbol{x}) = \sum_{t=1}^{n} \log(\hat{p}_\theta(x_t|\boldsymbol{x}_{\setminus t})) = \sum_{t=1}^{n} -E(\boldsymbol{x})_t$$

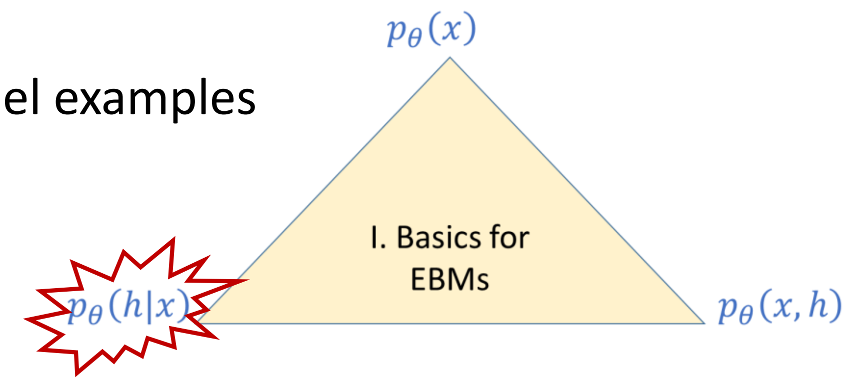
Content

I. Basics for EBMs (45 min)
1. Probabilistic graphical modeling (PGM) framework and EBM model examples
2. Learning EBMs by Monte Carlo methods
3. Learning EBMs by noise-contrastive estimation (NCE)

II. EBMs for language modeling (45 min)
1. Trans-dimensional random field (TRF) LMs for speech recognition
2. Residual energy-based models for text generation
3. Electric: an energy-based cloze model for representation learning over text

III. EBMs for speech recognition and natural language labeling (45 min)
1. CRFs as conditional EBMs
2. CRFs for speech recognition
3. CRFs for sequence labeling in NLP

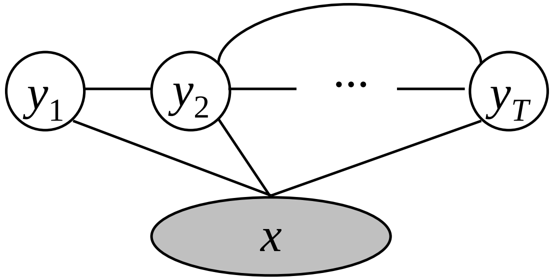
IV. EBMs for semi-supervised natural language labeling (45 min)
1. Upgrading EBMs to Joint EBMs (JEMs) for fixed-dimensional data
2. Upgrading CRFs to Joint random fields (JRFs) for sequential data
3. JRFs for semi-supervised natural language labeling



UGM Example - Conditional Random Fields (CRFs)

- A CRF is a conditional distribution $p(y|x)$ defined as a UGM/RF/EBM

$$p(y|x) = \frac{1}{Z(x)} exp\left[\sum_C \phi_C(y_C, x)\right]$$



- x is observed sequence, which is always given;
- y is hidden sequence;
- $\phi_C(y_C, x)$: Clique (log-)potential function over clique C in the subgraph induced by y

☺ CRFs can overcome "label bias" and "exposure bias" suffered by locally-normalized models

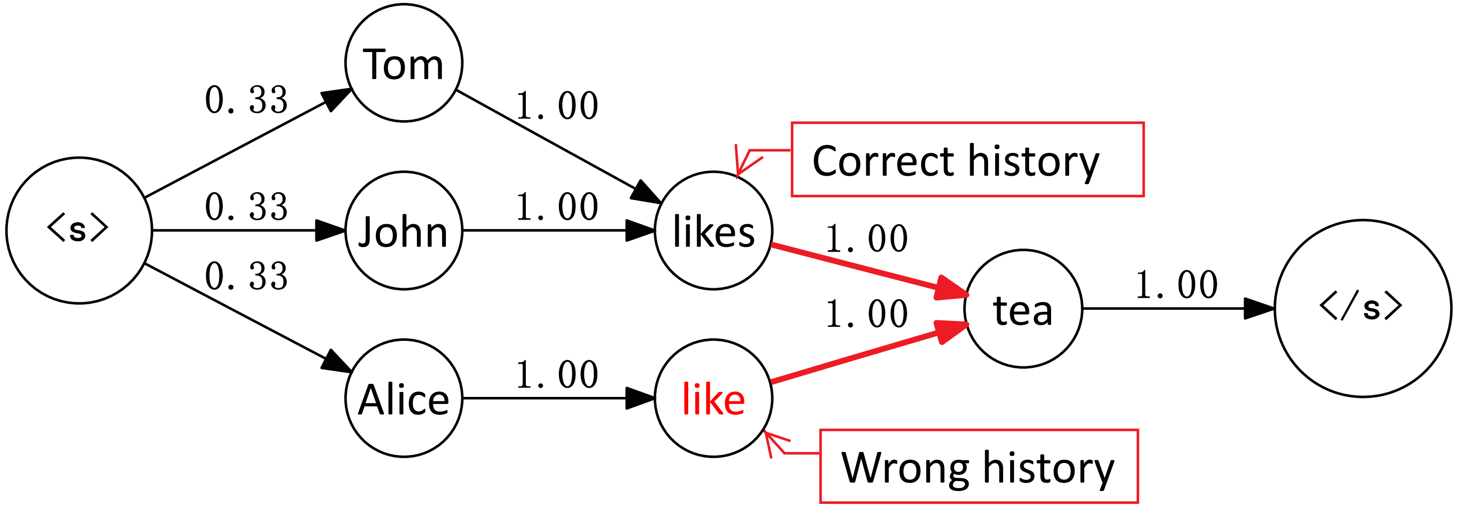
J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", ICML 2001.

Label bias

▶ Word probabilities at each time-step are locally normalized, so successors of incorrect histories receive the same mass as do the successors of the true history. [Wiseman, et al., 2016]

Training data

Tom likes tea
John likes tea
Alice like tea



▶ [Andor, et al., 2016]

- "Intuitively, we would like the model to be able to revise an earlier decision made during search, when later evidence becomes available that rules out the earlier decision as incorrect."
- "the label bias problem means that locally normalized models often have a very weak ability to revise earlier decisions."
- A proof that globally normalized models are strictly more expressive than locally normalized models.

- Wiseman, et al., "Sequence-to-sequence Learning as Beam-Search Optimization", EMNLP, 2016.
- Andor, et al., "Globally Normalized Transition-Based Neural Networks", ACL, 2016.

Exposure bias

▶ Mismatch between **training** (teacher forcing) and **testing** (prediction) of locally-normalized sequence models:

- **Training**: maximize the likelihood of each successive target word, conditioned on the gold history of the target word.
- **Testing**: the model predict the next step, using its own predicted samples in testing.

▶ The model is never exposed to its own errors during training, and so the inferred histories at test-time do not resemble the gold training histories. [Wiseman, et al., 2016]

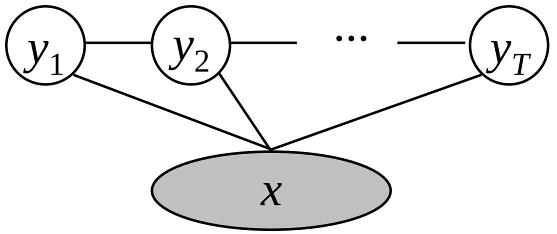
▶ **Exposure bias** results from training in a certain way (maybe alleviated by scheduled sampling), **Label bias** results from properties of the model itself.

- Wiseman, et al., "Sequence-to-sequence Learning as Beam-Search Optimization", EMNLP, 2016.
- Andor, et al., "Globally Normalized Transition-Based Neural Networks", ACL, 2016.

Linear-chain CRFs

for sequence tagging, e.g. POS tagging, shallow parser, Chinese word segmentation, …

$$p(y_{1:T}|x) \propto exp\left\{\sum_{t=1}^{T-1} \phi_t(y_t, x) + \sum_{t=1}^{T-1} \psi_t(y_{t-1}, y_t, x)\right\}$$



Traditional: **log-linear models** with discrete features

$$p(y_{1:T}|x) \propto exp\left\{\sum_{t=1}^{T-1} \sum_i \lambda_i f_i(y_t, x, t) + \sum_{t=1}^{T-1} \sum_j \mu_j f_j(y_{t-1}, y_t, x, t)\right\}$$

- Node potential

  $$\lambda_1 f_1(y_t, x, t) = \lambda_1 \cdot 1(y_t = \text{prep}, x_t = \text{on})$$

  $$\lambda_2 f_i(y_t, x, t) = \lambda_2 \cdot 1(y_t = \text{adv}, x_t \; ends \; in \; \text{ly})$$

- Edge potential

  $$\mu_1 f_1(y_{t-1}, y_t, x, t) = \mu_1 \cdot 1(y_t = \text{prep}, y_{t+1} = \text{non})$$

Recently: **neural CRFs**

Use NN to extract features

$$\text{LSTM}(x_{1:T}): x_{1:T} \rightarrow h_{1:T}$$

- Node potential, calculated via a linear layer

  $$\phi_t(y_t = k, x) = w_k^{\mathrm{T}} h_t \triangleq \phi_t^k$$

  w_k is the weight vector for label k

- Edge potential, mostly implemented as a matrix A

Training of CRFs in general

$$p_\theta(y|x) = \frac{1}{Z_\theta(x)} \exp[u_\theta(x, y)]$$

Normalizing constant:

$$Z_\theta(x) = \sum_y \exp[u_\theta(x, y)]$$

- Conditional Maximum likelihood (CML) training

The scaled log conditional likelihood of training data $\{(x_i, y_i), i = 1, \cdots, N\}$

$$L(\theta) \triangleq \frac{1}{N} \sum_{i=1}^{N} log p_\theta(y_i|x_i) = \frac{1}{N} \sum_{i=1}^{N} \{u_\theta(x_i, y_i) - \log Z_\theta(x_i)\}$$

$$\frac{\partial L(\theta)}{\partial \theta} = \frac{1}{N} \sum_{i=1}^{N} \left\{ \frac{\partial u_\theta(x_i, y_i)}{\partial \theta} - E_{p_\theta(y|x_i)} \left[\frac{\partial u_\theta(x_i, y)}{\partial \theta} \right] \right\}$$

Expectation under empirical distribution

Expectation under model distribution $p_\theta(y|x_i)$

For linear-chain CRFs, this expectation can be exactly calculated.

Training of Neural CRFs

Model $p_\theta(y|x) = \frac{1}{Z_\theta(x)} \exp[u_\theta(x,y)]$, where $u_\theta(x,y) = \sum_t \phi_t^{y_t} + \sum_t A_{y_{t-1},y_t}$

For potential value $\phi_t^k, 1 \le t \le T, 1 \le k \le K$

$$\frac{\partial logp(y|x)}{\partial \phi_t^k} = \delta(y_t = k) - E_{p(y|x)}[\delta(y_t = k)]$$

$$= \delta(y_t = k) - p(y_t = k|x)$$

i.e., the **error signal** received by the NN feature extractor during training

i.e., γ_t^k, the posterior **state occupation probability**, calculated using the alpha-beta variables from the forward-backward algorithm [Rabiner, 1989]

Conditional random field (CRF) - Summary

A CRF define a conditional distribution over output sequence y^l given input sequence x^l of length l :

$$p_\theta(y^l|x^l) = \frac{1}{Z_\theta(x^l)} \exp(u_\theta(x^l, y^l)) \qquad Z_\theta(x^l) = \sum_{y^l} \exp(u_\theta(x^l, y^l))$$

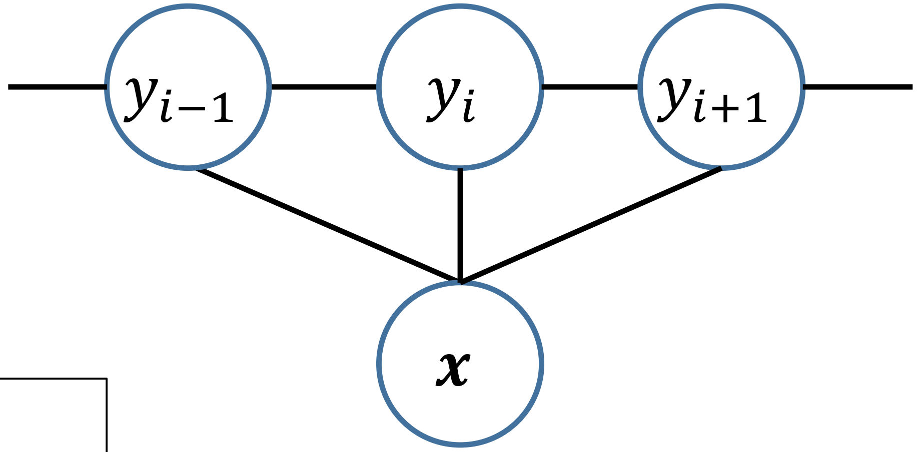
Potential for linear-chain:

Node potential

Edge potential

$$u_\theta(x^l, y^l) = \sum_{i=1}^{l} \phi_i(y_i, x^l) + \sum_{i=1}^{l} \psi_i(y_{i-1}, y_i, x^l)$$

☺ CRFs can overcome "label bias" and "exposure bias".



Example of a linear-chain CRF

Successfully applied for sequence labeling in NLP, less so for ASR

- CRFs was explored for phone classification, using zero, first and second order features [Gunawardana, et al., 2005].
- CTC-CRF: the first CRF successfully developed for end-to-end ASR

A. Gunawardana, et al.,"Hidden conditional random fields for phone classification", Europspeech, 2005.

Content

I. Basics for EBMs (45 min)
   1. Probabilistic graphical modeling (PGM) framework and EBM model examples (classic & modern)
   2. Learning EBMs by Monte Carlo methods
   3. Learning EBMs by noise-contrastive estimation (NCE)

II. EBMs for language modeling (45 min)
   1. Trans-dimensional random field (TRF) LMs for speech recognition
   2. Residual energy-based models for text generation
   3. Electric: an energy-based cloze model for representation learning over text

III. EBMs for speech recognition and natural language labeling (45 min)
   1. CRFs as conditional EBMs
   2. CRFs for speech recognition
   3. CRFs for sequence labeling in NLP

IV. EBMs for semi-supervised natural language labeling (45 min)
   1. Upgrading EBMs to Joint EBMs (JEMs) for fixed-dimensional data
   2. Upgrading CRFs to Joint random fields (JRFs) for sequential data
   3. JRFs for semi-supervised natural language labeling

End-to-end ASR: Basics

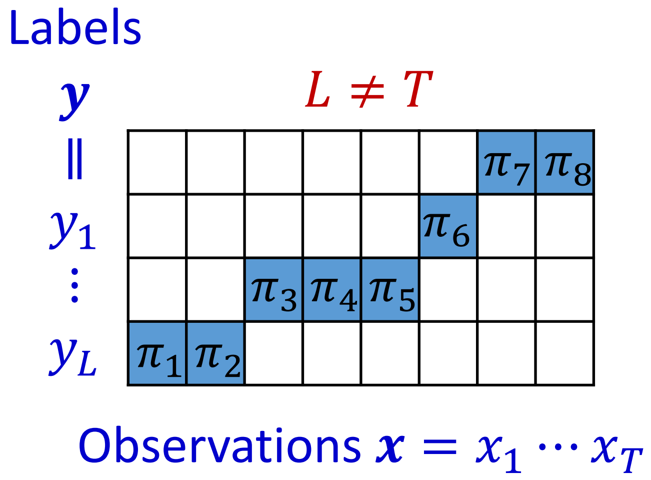
ASR is a *sequence discriminative* problem

- For acoustic observations $\boldsymbol{x} \triangleq x_1, \cdots, x_T$, find the most likely labels $\boldsymbol{y} \triangleq y_1, \cdots, y_L$

1. How to obtain $p(\boldsymbol{y} \mid \boldsymbol{x})$
2. How to handle alignment, since $L \neq T$

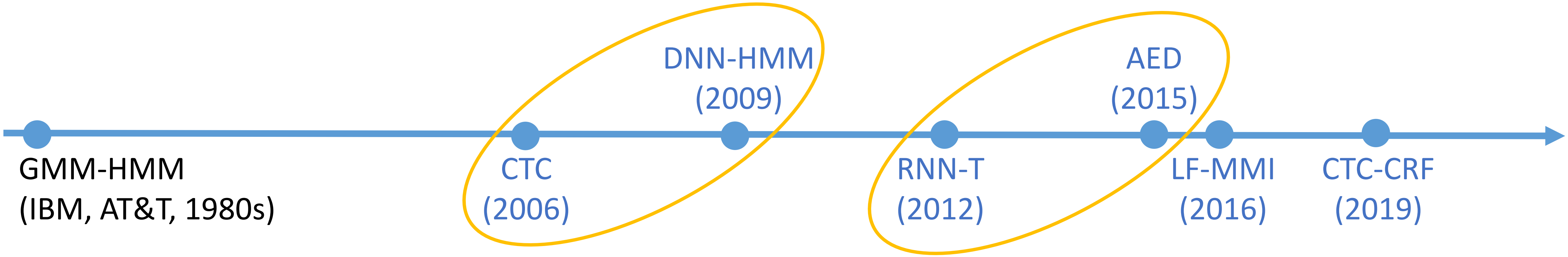
- Need a differentiable sequence-level loss of mapping acoustic sequence $\boldsymbol{y}$ to label sequence $\boldsymbol{x}$

- **Explicitly**: introduce hidden state sequence $\boldsymbol{\pi}$, as in Connectionist Temporal Classification (CTC), RNN Transducer (RNNT), CRF
- **Implicitly**: as in Attention based Encoder-Decoder (AED)



Example of explicit alignment

Brief ASR History

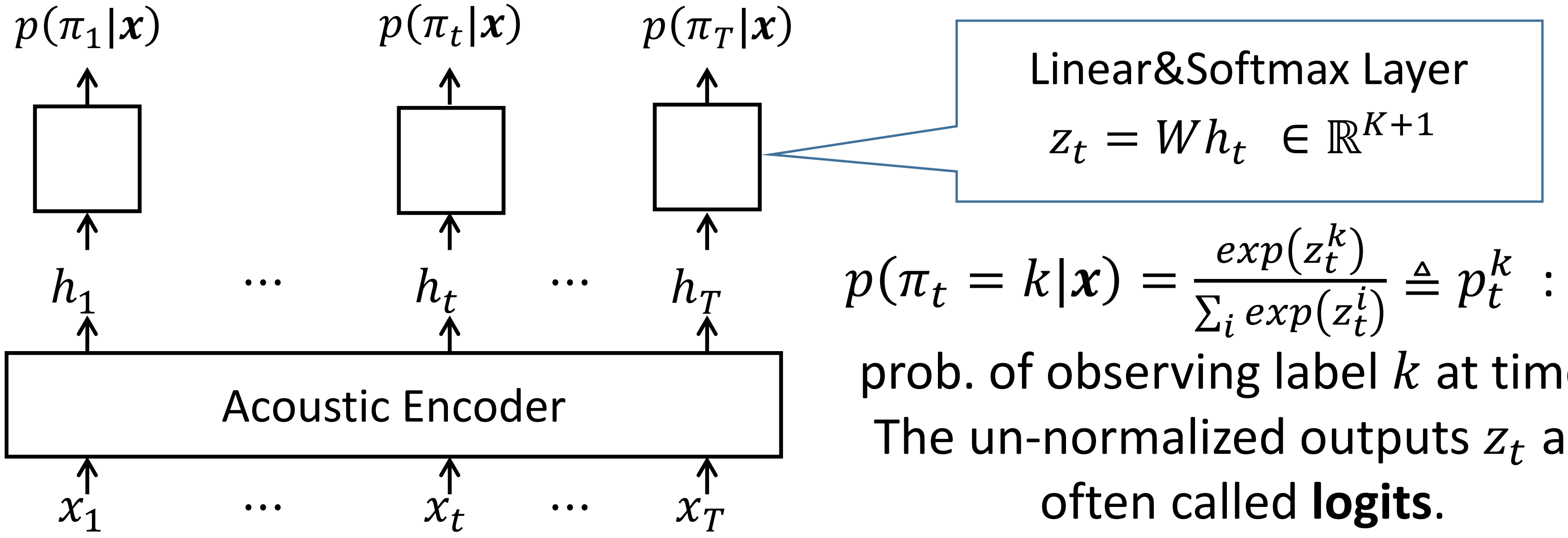


- [CTC] Graves, et al., “Connectionist Temporal Classification: Labelling unsegmented sequence data with RNNs”, ICML 2006.
- [DNN-HMM] A. Mohamed, et al., “Deep belief networks for phone recognition”, NIPS Workshop Deep Learning for Speech Recognition and Related Applications, 2009.
- [RNNT] A. Graves, “Sequence transduction with recurrent neural networks”, ICML 2012 Workshop on Representation Learning.
- [AED] D. Bahdanau, et al., “Neural machine translation by jointly learning to align and translate”, ICLR 2015.
- [LF-MMI] D. Povey, et al., "Purely sequence-trained neural networks for ASR based on lattice-free MMI", INTERSPEECH 2016.
- [CTC-CRF] Xiang&Ou. "CRF-based Single-stage Acoustic Modeling with CTC Topology", ICASSP, 2019.

CTC: introducing blank symbol

- Motivation: training $p(\boldsymbol{y} \mid \boldsymbol{x})$ without the need for frame-level alignments between the acoustics $\boldsymbol{x}$ and the transcripts $\boldsymbol{y}$
  - Introduce a state sequence $\boldsymbol{\pi} \triangleq \pi_1, \cdots, \pi_T$, where $\pi_t \in$ the-alphabet-of-labels $\cup$ <b>

Path posterior $p(\boldsymbol{\pi} \mid \boldsymbol{x}) = \prod_{t=1}^{T}$ State posterior $p(\pi_t \mid \boldsymbol{x})$



$p(\pi_t = k|\boldsymbol{x}) = \frac{exp(z_t^k)}{\sum_i exp(z_t^i)} \triangleq p_t^k$: the prob. of observing label k at time t

The un-normalized outputs z_t are often called **logits**.

Graves, et al., “Connectionist Temporal Classification: Labelling unsegmented sequence data with RNNs”, ICML 2006.

CTC topology

- **State topology** refers to the state transition structure in $\boldsymbol{\pi}$, which basically determines the mapping $\mathcal{B}_{CTC}$ from $\boldsymbol{\pi}$ to $\boldsymbol{y}$

CTC topology : a mapping $\mathcal{B}_{CTC}$ maps $\boldsymbol{\pi}$ to $\boldsymbol{y}$ by

1. reducing repetitive symbols to a single symbol;
2. removing all blank symbols.

$$\mathcal{B}(-CC--AA-T-) = CAT$$

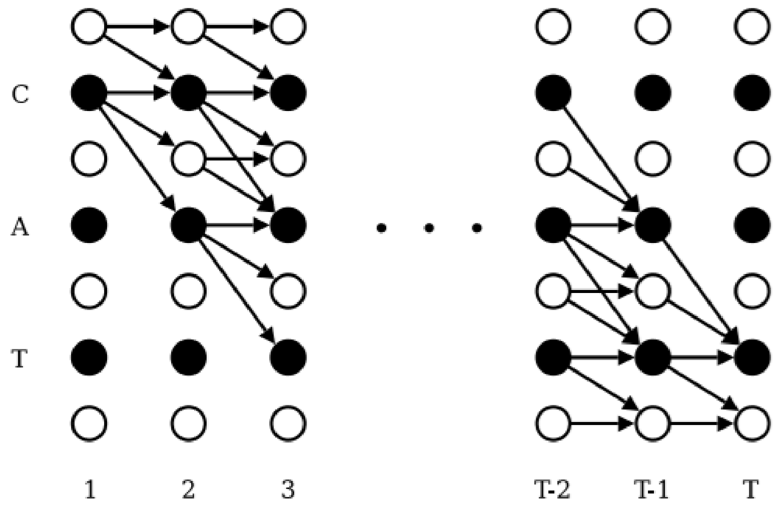
Path posterior

$$p(\boldsymbol{\pi}|\boldsymbol{x}) = \prod_{t=1}^{T} p(\pi_t|\boldsymbol{x})$$

Label-seq posterior

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi}:\mathcal{B}_{CTC}(\boldsymbol{\pi})=\boldsymbol{y}} p(\boldsymbol{\pi}|\boldsymbol{x})$$

Summing over all possible paths, which map to $\boldsymbol{y}$



B B **c** B B **a** **a** B B **t**
B **c** **c** B **a** B B B B **t**
. . .
B **c** B B **a** B B **t** **t** B

CTC: shortcoming

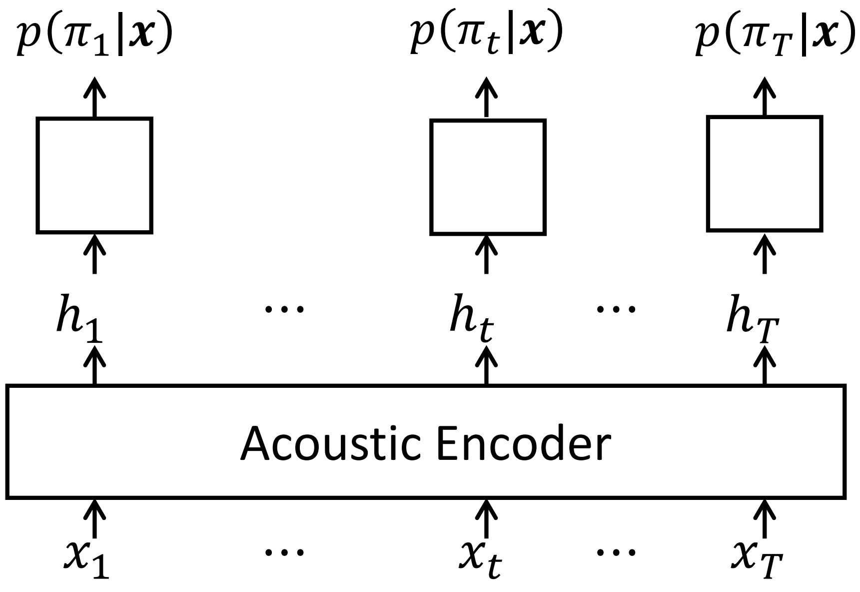
- Conditional independence assumption

Overcome

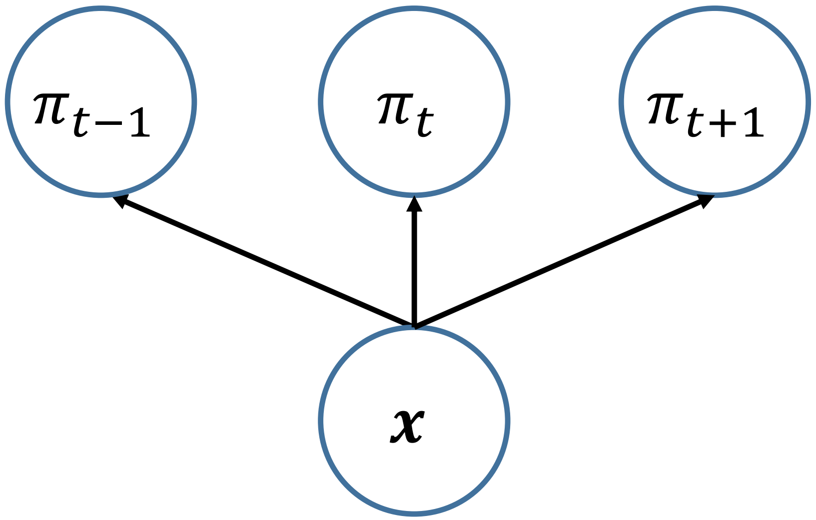
RNN-T

CTC-CRF

$$p(\boldsymbol{\pi}|\ \boldsymbol{x}) = \prod_{t=1}^{T} p(\pi_t|\boldsymbol{x})$$



Computational flow



Graphical Model Representation

Section Content

1. Motivation
2. Related work
3. Method: **CTC-CRF**
4. Experiments
5. Conclusion

- H. Xiang, Z. Ou. "CRF-based Single-stage Acoustic Modeling with CTC Topology", ICASSP, 2019.
- K. An, H. Xiang, Z. Ou. "CAT: A CTC-CRF based ASR Toolkit Bridging the Hybrid and the End-to-end Approaches towards Data Efficiency and Low Latency", INTERSPEECH, 2020.
- Fan, et al., "The SLT 2021 children speech recognition challenge: Open datasets, rules and baselines", SLT, 2021.
- H. Zheng, W. Peng, Z. Ou, J. Zhang. "Advancing CTC-CRF Based End-to-End Speech Recognition with Wordpieces and Conformers", arXiv:2107.03007, 2021.

Motivation: data-efficient end2end

- End-to-end system:
  - Eliminate the construction of GMM-HMMs and phonetic decision-trees, and can be trained from scratch (flat-start or single-stage)
- In a more strict/ambitious sense:
  - Remove the need for a pronunciation lexicon and, even further, train the acoustic and language models jointly rather than separately
  - Data-hungry

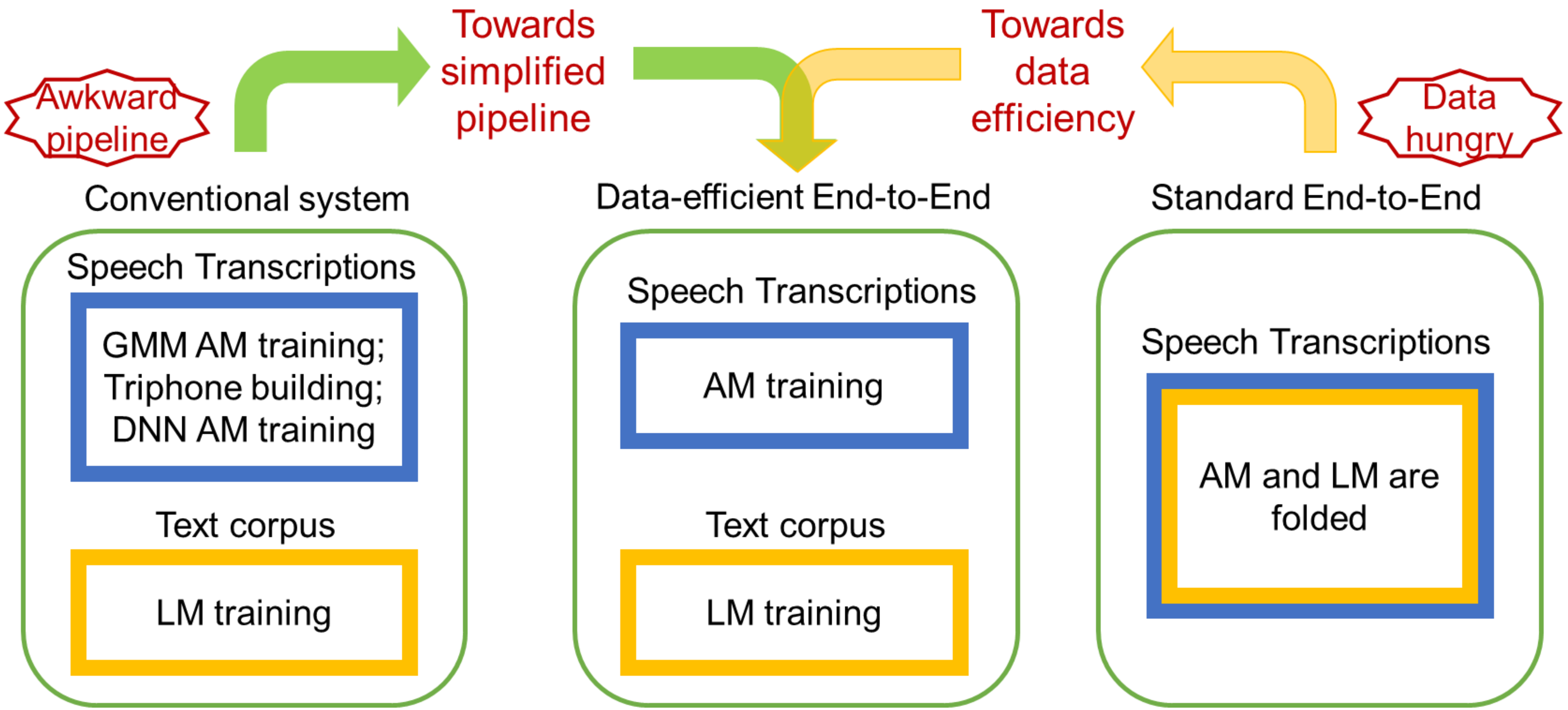
We need data-efficient end2end speech recognition, which can flexibly use a separate language model (LM) with or without a pronunciation lexicon.

- Text corpus for language modeling are cheaply available.
- Data-efficient

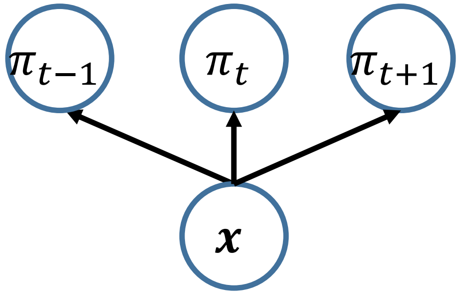
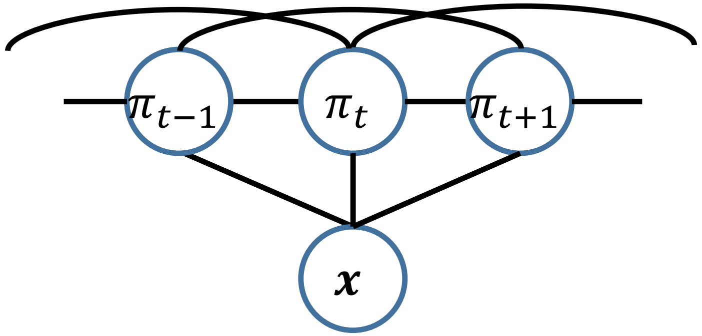
Motivation: bridging

Modularization promote Data-efficiency

✓ Keep necessary factorization of AM and LM



CTC vs CTC-CRF

CTC	CTC-CRF
$p(\boldsymbol{y}\|\boldsymbol{x}) = \sum_{\boldsymbol{\pi}:\mathcal{B}(\boldsymbol{\pi})=\boldsymbol{y}} p(\boldsymbol{\pi}\|\boldsymbol{x})$, using CTC topology $\mathcal{B}$	
State Independence $p(\boldsymbol{\pi}\|\boldsymbol{x};\boldsymbol{\theta}) = \prod_{t=1}^{T} p(\pi_t\|\boldsymbol{x})$	$p(\boldsymbol{\pi}\|\boldsymbol{x};\boldsymbol{\theta}) = \frac{e^{\phi(\boldsymbol{\pi},\boldsymbol{x};\boldsymbol{\theta})}}{\sum_{\boldsymbol{\pi}'} e^{\phi(\boldsymbol{\pi}',\boldsymbol{x};\boldsymbol{\theta})}}$ $\phi(\boldsymbol{\pi},\boldsymbol{x};\boldsymbol{\theta}) = \left(\begin{array}{l}\sum_{t=1}^{T}\log p(\pi_t\|\boldsymbol{x}) \\ +\log p_{LM}(\mathcal{B}(\boldsymbol{\pi}))\end{array}\right)$ Node potential, by NN Edge potential, by n-gram denominator LM of labels, like in LF-MMI
$\frac{\partial \log p(\boldsymbol{y}\|\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = \mathbb{E}_{p(\boldsymbol{\pi}\|\boldsymbol{y},\boldsymbol{x};\boldsymbol{\theta})}\left[\frac{\partial \log p(\boldsymbol{\pi}\|\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right]$	$\frac{\partial \log p(\boldsymbol{y}\|\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = \mathbb{E}_{p(\boldsymbol{\pi}\|\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta})}\left[\frac{\partial \phi(\boldsymbol{\pi},\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right] - \mathbb{E}_{p(\boldsymbol{\pi}'\|\boldsymbol{x};\boldsymbol{\theta})}\left[\frac{\partial \phi(\boldsymbol{\pi}',\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right]$
	

Training of CTC-CRFs

Model $\quad p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi}:\mathcal{B}(\boldsymbol{\pi})=\boldsymbol{y}} p(\boldsymbol{\pi}|\boldsymbol{x})$, where $p(\boldsymbol{\pi}|\boldsymbol{x}) = \dfrac{e^{\phi(\boldsymbol{\pi},\boldsymbol{x})}}{\sum_{\boldsymbol{\pi}'} e^{\phi(\boldsymbol{\pi}',\boldsymbol{x})}}$

Node potential   Edge potential

$$\phi(\boldsymbol{\pi},\boldsymbol{x}) = \Sigma_{t=1}^{T} \log p(\pi_t|\boldsymbol{x}) + \log p_{LM}\,(\mathcal{B}(\boldsymbol{\pi}))$$

Denote $p(\pi_t = k|\boldsymbol{x}) = \phi_t^k$, then for potential value $\phi_t^k, 1 \le t \le T, 1 \le k \le K+1$

$$\frac{\partial \log p(\boldsymbol{y}|\boldsymbol{x})}{\partial \phi_t^k} = \mathbb{E}_{p(\boldsymbol{\pi}|\boldsymbol{x},\boldsymbol{y})}\left[\frac{\partial \phi(\boldsymbol{\pi},\boldsymbol{x})}{\partial \phi_t^k}\right] - \mathbb{E}_{p(\boldsymbol{\pi}'|\boldsymbol{x})}\left[\frac{\partial \phi(\boldsymbol{\pi}',\boldsymbol{x})}{\partial \phi_t^k}\right]$$

$$= E_{p(\boldsymbol{\pi}|\boldsymbol{x},\boldsymbol{y})}[\delta(\boldsymbol{\pi}_t = k)] - E_{p(\boldsymbol{\pi}'|\boldsymbol{x})}[\delta(\boldsymbol{\pi}'_t = k)]$$

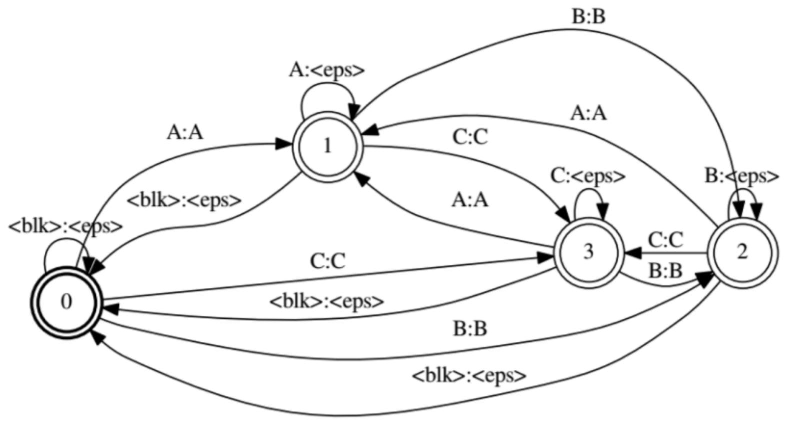
i.e., the **error signal** received the acoustic encoder NN during training

i.e., the posterior **state occupation probability**, by running the FB algorithm over the WFST determined by $\boldsymbol{y}$

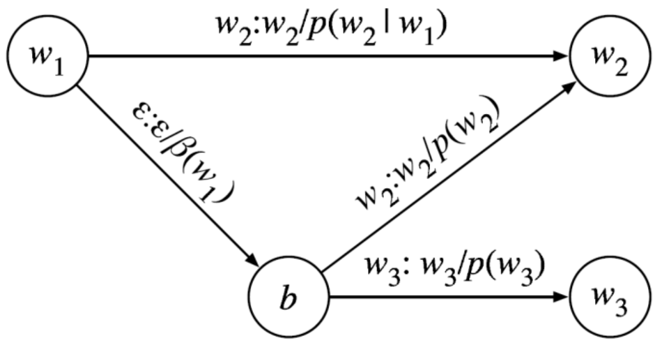
i.e., the posterior **state occupation probability**, by running the FB algorithm over the WFST determined by n-gram LM of labels

Edge potential in CTC-CRF $\log p_{LM}\left(\mathcal{B}(\boldsymbol{\pi})\right)$

WFST representing CTC topology: T

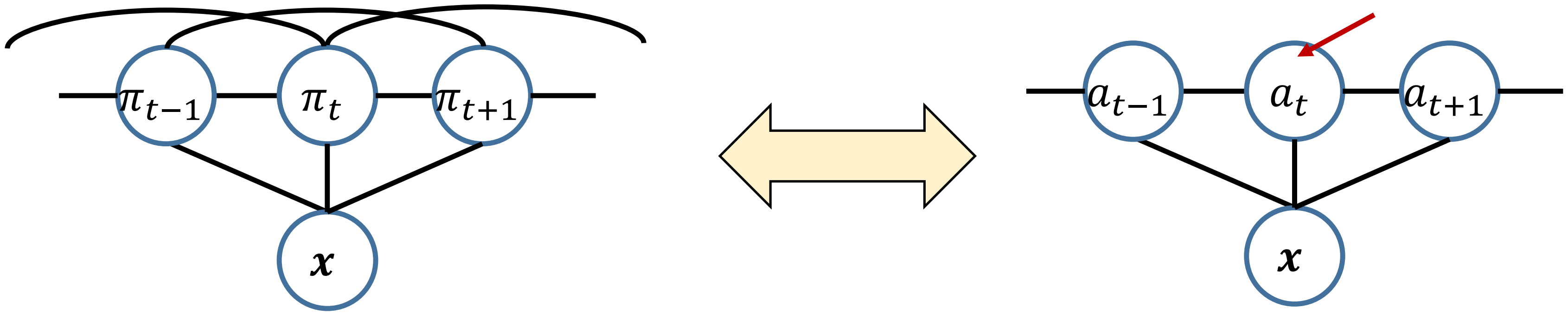


n-gram denominator LM of labels: G



$T \circ G$: Composed into a single WFST

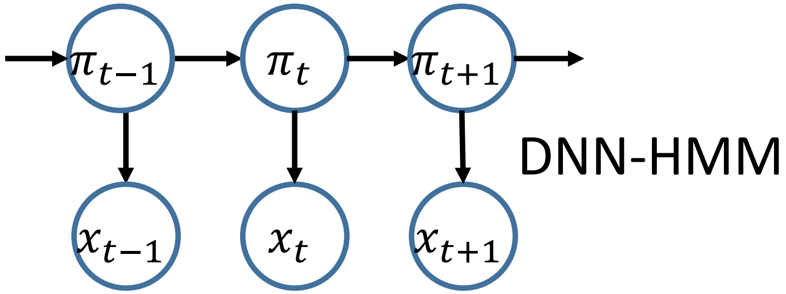
The arc in the WFST is treated as the state

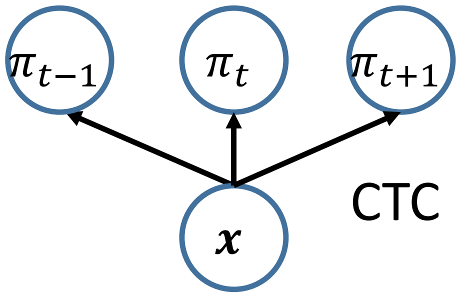


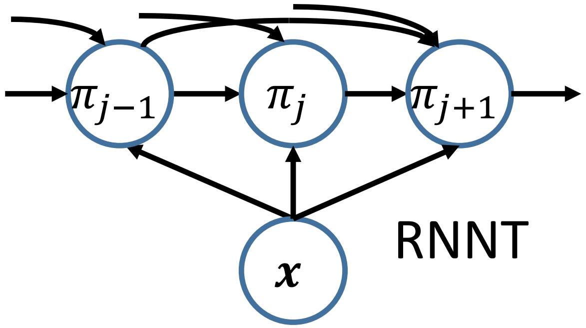
$$p(\pi_t = k|\boldsymbol{x}) = \sum\nolimits_{\text{input symbol of } a \text{ is } k} p(a_t = a|\boldsymbol{x})$$

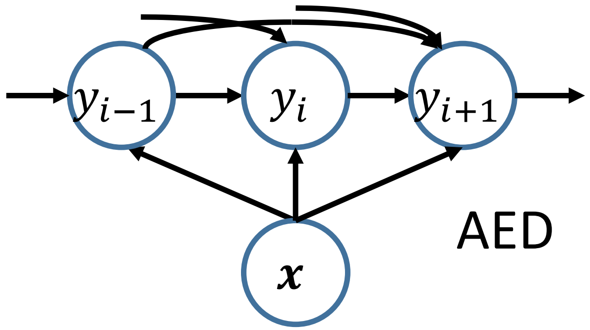
Related work

- Directed Graphical Model/Locally normalized
  - ➢ DNN-HMM : Model $p(\boldsymbol{\pi}, \boldsymbol{x})$ as an HMM, could be discriminatively trained, e.g. by $\max_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x})$
  - ➢ CTC : $p(\boldsymbol{\pi} | \boldsymbol{x}) = \prod_{t=1}^{T} p(\pi_t | \boldsymbol{x})$
  - ➢ RNNT : $p(\pi_{1:T+U} | x_{1:T}) = \prod_{j=1}^{T+U} p\left(\pi_j | \pi_{1:j-1}\right)$
  - ➢ AED : $p(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{i=1}^{L} p(y_i | y_1, \cdots, y_{i-1}, \boldsymbol{x})$



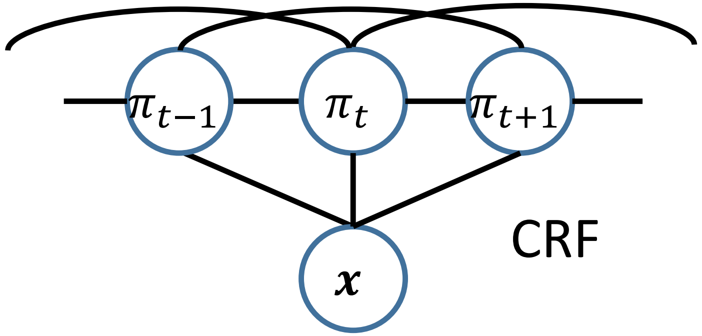






- Undirected Graphical Model/Globally normalized
  - ➢ CRF : $p(\boldsymbol{\pi} | \boldsymbol{x}) \propto exp[\phi(\boldsymbol{\pi}, \boldsymbol{x})]$

CTC-CRF is fundamentally different from all history models!

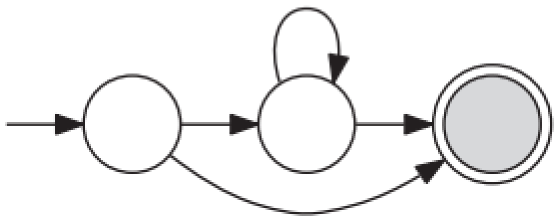


Related work (SS-LF-MMI/EE-LF-MMI)

- Single-Stage (SS) Lattice-Free Maximum-Mutual-Information (LF-MMI)
  - 10 - 25% relative WER reduction on 80-h WSJ, 300-h Switchboard and 2000-h Fisher+Switchboard datasets, compared to CTC, Seq2Seq, RNN-T.
  - Cast as MMI-based discriminative training of an HMM (generative model) with

    Pseudo state-likelihoods calculated by the bottom DNN,

    Fixed state-transition probabilities.
  - 2-state HMM topology
  - Including a silence label



CTC-CRF

- Cast as a CRF;
- CTC topology;
- No silence label.

Hadian, et al., "Flat-start single-stage discriminatively trained HMM-based models for ASR", T-ASLP 2018.

SS-LF-MMI vs CTC-CRF

	SS-LF-MMI	CTC-CRF
State topology	HMM topology with two states	CTC topology
Silence label	Using silence labels. Silence labels are randomly inserted when estimating denominator LM.	No silence labels. Use <blk> to absorb silence. ☺ No need to insert silence labels to transcripts.
Decoding	No spikes.	The posterior is dominated by <blk> and non-blank symbols occur in spikes. ☺ Speedup decoding by skipping blanks.
Implementation	Modify the utterance length to one of 30 lengths; use leaky HMM.	☺ No length modification; no leaky HMM.

Experiments

- We conduct our experiments on three benchmark datasets:
  - WSJ 80 hours
  - Switchboard 300 hours
  - Librispeech 1000 hours
- Acoustic model: 6 layer BLSTM with 320 hidden dim, 13M parameters
- Adam optimizer with an initial learning rate of 0.001, decreased to 0.0001 when cv loss does not decrease
- Implemented with Pytorch.
- Objective function (use the CTC objective function to help convergences):

$$\mathcal{J}_{CTC-CRF} + \alpha \mathcal{J}_{CTC}$$

- Decoding score function (use word-based language models, WFST based decoding):

$$\log p(\boldsymbol{l}|\boldsymbol{x}) + \beta \log p_{LM}(\boldsymbol{l})$$

H. Xiang, Z. Ou. "CRF-based Single-stage Acoustic Modeling with CTC Topology", ICASSP, 2019.

Experiments (Comparison with CTC, phone based)

WSJ 80h

Model	Unit	LM	SP	dev93	eval92
CTC	Mono-phone	4-gram	N	10.81%	7.02%
CTC-CRF	Mono-phone	4-gram	N	6.24%	3.90%

44.4%

Switchboard 300h

Model	Unit	LM	SP	SW	CH
CTC	Mono-phone	4-gram	N	12.9%	23.6%
CTC-CRF	Mono-phone	4-gram	N	11.0%	21.0%

14.7% 11%

Librispeech 1000h

Model	Unit	LM	SP	Dev Clean	Dev Other	Test Clean	Test Other
CTC	Mono-phone	4-gram	N	4.64%	13.23%	5.06%	13.68%
CTC-CRF	Mono-phone	4-gram	N	3.87%	10.28%	4.09%	10.65%

19.1% 22.1%

SP: speed perturbation for 3-fold data augmentation.

Experiments (Comparison with STOA)

Switchboard 300h

Model	SW	CH	Average	Source
Kaldi chain triphone	9.6	19.3	14.5	IS 2016
Kaldi e2e chain monophone	11.0	20.7	15.9	ASLP 2018, 26M
Kaldi e2e chain biphone	9.8	19.3	14.6	ASLP 2018, 26M
CTC-CRF monophone	10.3	19.7	15.0	ICASSP 2019, BLSTM, 13M
CTC-CRF monophone	**9.8**	**18.8**	**14.3**	**IS 2020, VGG BLSTM, 16M**

10%

RWTH IS 2018, “Improved training of end-to-end attention models for speech recognition”.
RWTH IS 2019, “RWTH ASR Systems for LibriSpeech Hybrid vs Attention -- Data Augmentation”.
IBM IS19, “Forget a Bit to Learn Better Soft Forgetting for CTC-based Automatic Speech Recognition”.
Espnet ASRU19, “Espresso: A Fast End-to-end Neural Speech Recognition Toolkit”.
Google IS19, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition”.

Experiments (Comparison with STOA)

Librispeech 1000h

Model	Test Clean	Test Other	Source
Kaldi chain triphone	4.28	-	IS 2016
CTC-CRF monophone	**4.0**	**10.6**	**ICASSP 2019, BLSTM (6,320), 13M**

RWTH IS 2018, "Improved training of end-to-end attention models for speech recognition".
RWTH IS 2019, "RWTH ASR Systems for LibriSpeech Hybrid vs Attention -- Data Augmentation".
IBM IS19, "Forget a Bit to Learn Better Soft Forgetting for CTC-based Automatic Speech Recognition".
Espnet ASRU19, "Espresso: A Fast End-to-end Neural Speech Recognition Toolkit".
Google IS19, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition".

Mandarin Aishell-1 results

- 170 hours mandarin speech corpus
- 400 speakers from different accent areas
- 15% CER reduction compared with LF-MMI
- 5% CER reduction compared with end-to-end transformer

Model	%CER
LF-MMI with i-vector [1]	7.43
Transformer [2]	6.7
CTC-CRF [3]	6.34

[1] D. Povey, A. Ghoshal, and et al, "The Kaldi speech recognition toolkit," ASRU 2011.
[2] S. Karita, N. Chen, and et al, "A comparative study on transformer vs RNN in speech applications," ASRU 2019.
[3] Keyu An, Hongyu Xiang, and **Zhijian Ou**, "CAT: A CTC-CRF based ASR toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency," INTERSPEECH 2020.

- 400 hours of data, targeting to boost children speech recognition research.
- Evaluated on 10 hours of children's reading and conversational speech.
- 3 baselines (Chain model, Transformer and CTC-CRF) are provided.

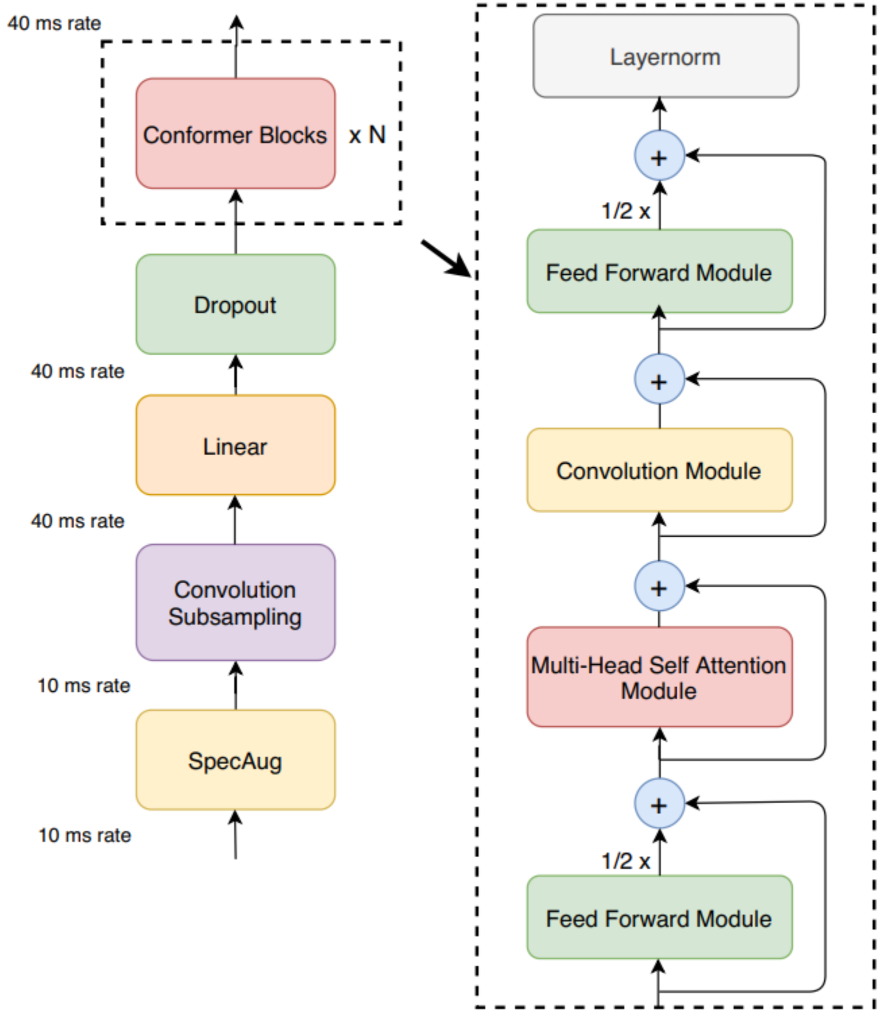
model	Kaldi Chain model	Espnet Transformer	CTC-CRF
CER%	28.75	27.28	**25.34**

Fan Yu, Zhuoyuan Yao, Xiong Wang, Keyu An, Lei Xie, **Zhijian Ou**, Bo Liu, Xiulin Li, Guanqiong Miao. The SLT 2021 children speech recognition challenge: Open datasets, rules and baselines. SLT 2021.

Advancing CTC-CRF Based End-to-End Speech Recognition with Wordpieces and Conformers

Huahuan Zheng, Wenjie Peng, **Zhijian Ou** and Jinsong Zhang, arXiv:2107.03007

Basic Units of Labels	Label Sequence
phoneme	DH AE1 T N IY1 DH ER0 AH1 V DH EH1 M HH AE1 D K R AO1 S T DH AH0 TH R EH1 SH OW2 L D S IH1 N S DH AH0 D AA1 R K D EY1
character /grapheme	t h a t _ n e i t h e r _ o f _ t h e m _ h a d _ c r o s s e d _ t h e _ t h r e s h o l d _ s i n c e _ t h e _ d a r k _ d a y _
subword /wordpiece	that_ ne i ther_ of_ them_ had_ cro s sed_ the_ th re sh old_ sin ce_ the_ d ar k_ day_
word	that neither of them had crossed the threshold since the dark day



Experiments (Comparison between different units, WER%)

Switchboard 300h

Model	Unit	LM	Augmentation	Eval2000	SW	CH
Conformer (this work)	monophone	4-gram	SP, SA	12.1	7.9	16.1
	monophone	Trans.*	SP, SA	10.7	6.9	14.5
	wordpiece	4-gram	SP, SA	12.7	8.7	16.5
	wordpiece	Trans.*	SP, SA	11.1	7.2	14.8

Librispeech 1000h

Model	Unit	LM	Augmentation	Test Clean	Test Other
Conformer (this work)	monophone	4-gram	SA	3.61	8.10
	monophone	Trans.**	SA	2.51	5.95
	wordpiece	4-gram	SA	3.59	8.37
	wordpiece	Trans.**	SA	2.54	6.33

SP: speed perturbation for 3-fold data augmentation.
SA: our implementation of SpecAug with ratio
* Latest Kaldi Transformer LM rescoring
** RWTH 42-layer Transformer

English: a low degree of grapheme-phoneme correspondence

Experiments (Comparison between different units, WER%)

CommonVoice German 700h

Model	#params	unit	LM	Augmentation	Test
Conformer (This work)	25.03	char	4-gram	SP, SA	12.7
	25.03	char	Trans.	SP, SA	11.6
	25.03	monophone	4-gram	SP, SA	10.7
	25.03	monophone	Trans.	SP, SA	10.0
	25.06	wordpiece	4-gram	SP, SA	10.5
	25.06	wordpiece	Trans.	SP, SA	9.8

German: a high degree of grapheme-phoneme correspondence

Experiments (Comparison with STOA)

Switchboard 300h

Model	#params	LM	unit	SW	CH	Eval2000
RNN-T, 2021 [10]	57	RNN LM	char	6.4	13.4	9.9
Conformer [9]	44.6	Trans.	bpe	6.8	14.0	10.4
TDNN-F [11]	-	Trans.*	triphone	7.2	14.4	10.8
TDNN-F [11]	-	Trans.**	triphone	6.5	13.9	10.2
VGGBLSTM [2]	39.15	RNN LM	monophone	8.8	17.4	[13.0]
Conformer (This work)	51.82	Trans.	monophone	6.9	14.5	10.7
	51.85	Trans.	wordpiece	7.2	14.8	11.1

* N-best rescoring, ** Iterative lattice rescoring

[2] "CAT: A CTC-CRF based ASR toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency," INTERSPEECH 2020.
[9] "Conformer: Convolution-augmented Transformer for Speech Recognition", Interspeech 2020.
[10] "Advancing RNN transducer technology for speech recognition," ICASSP 2021.
[11] "A paralleliz- able lattice rescoring strategy with neural language models," ICASSP, 2021

Section Conclusion

- The CTC-CRF framework inherits the data-efficiency of the hybrid approach and the simplicity of the end-to-end approach.
- CTC-CRF significantly outperforms regular CTC on a wide range of benchmarks, and is on par with other state-of-the-art end-to-end models.
  - English WSJ-80h, Switchboard-300h, Librispeech-1000h; Mandarin Aishell-170h; …
- Flexibility
  - Streaming ASR <- INTRESPEECH 2020
  - Neural Architecture Search <- SLT 2021
  - Children Speech Recognition <- SLT 2021
  - Wordpieces, Conformer architectures
  - Multilingual and Crosslingual <- ASRU2021
  - …

CAT

https://github.com/thu-spmi/cat

Content

I. Basics for EBMs (45 min)
1. Probabilistic graphical modeling (PGM) framework and EBM model examples (classic & modern)
2. Learning EBMs by Monte Carlo methods
3. Learning EBMs by noise-contrastive estimation (NCE)

II. EBMs for language modeling (45 min)
1. Trans-dimensional random field (TRF) LMs for speech recognition
2. Residual energy-based models for text generation
3. Electric: an energy-based cloze model for representation learning over text

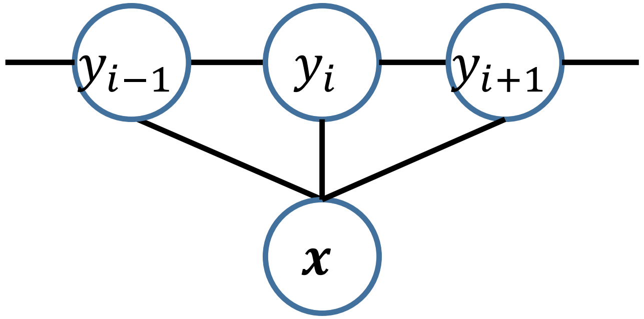
III. EBMs for speech recognition and natural language labeling (45 min)
1. CRFs as conditional EBMs
2. CRFs for speech recognition
3. CRFs for sequence labeling in NLP

IV. EBMs for semi-supervised natural language labeling (45 min)
1. Upgrading EBMs to Joint EBMs (JEMs) for fixed-dimensional data
2. Upgrading CRFs to Joint random fields (JRFs) for sequential data
3. JRFs for semi-supervised natural language labeling

Motivation

- Conditional random fields (CRFs) have been shown to be one of the most successful approaches to sequence labeling.

- Various linear-chain neural CRFs (NCRFs) have been developed
  - Node potential modeling is improved by using NNs
  - But the linear-chain structure is still kept, i.e. using a bigram table as the edge potential
  - Linear-chain NCRFs capture only first-order [1] interactions and neglect higher-order dependencies between labels, which can be potentially useful in real-world sequence labeling applications



Linear-chain CRF

[1] *Fixed n-th order can be cast as first-order.*

How can we improve CRFs to capture long-range dependencies in the label sequence (preferably non-Markovian)?

Related work

- Infinite-Order CRFs (based on the Hierarchical Pitman-Yor Process) [a], Semi-Markov CRFs [b], Latent-dynamic CRFs [c], but not by using NNs
- Attention-based encoder-decoder (AED) and RNNT exploit non-Markovian dependences between labels, but both are locally normalized models and thus suffer from the label bias and exposure bias problems
- [d] extends AED, by removing the final softmax in the RNN decoder to learn global sequence scores, but cast as a non-probabilistic variant of the seq2seq model
- [e] proposes **neural CRF transducers**: RNNT+CRF

a. Sotirios P. Chatzis and Yiannis Demiris, “The Infinite-Order Conditional Random Field Model for Sequential Data Modeling,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, pp. 1523–1534, 2013.
b. Sunita Sarawagi and William W. Cohen, “Semi-Markov Conditional Random Fields for Information Extraction,” in NIPS, 2004.
c. Louis-Philippe Morency, Ariadna Quattoni, and Trevor Darrell, “Latent-Dynamic Discriminative Models for Continuous Gesture Recognition,” in CVPR, 2007.
d. Wiseman, et al., "Sequence-to-sequence Learning as Beam-Search Optimization", EMNLP, 2016.
e. Kai Hu, Zhijian Ou, et al. Neural CRF Transducers for Sequence Labeling. ICASSP, 2019.

Related work - RNNT

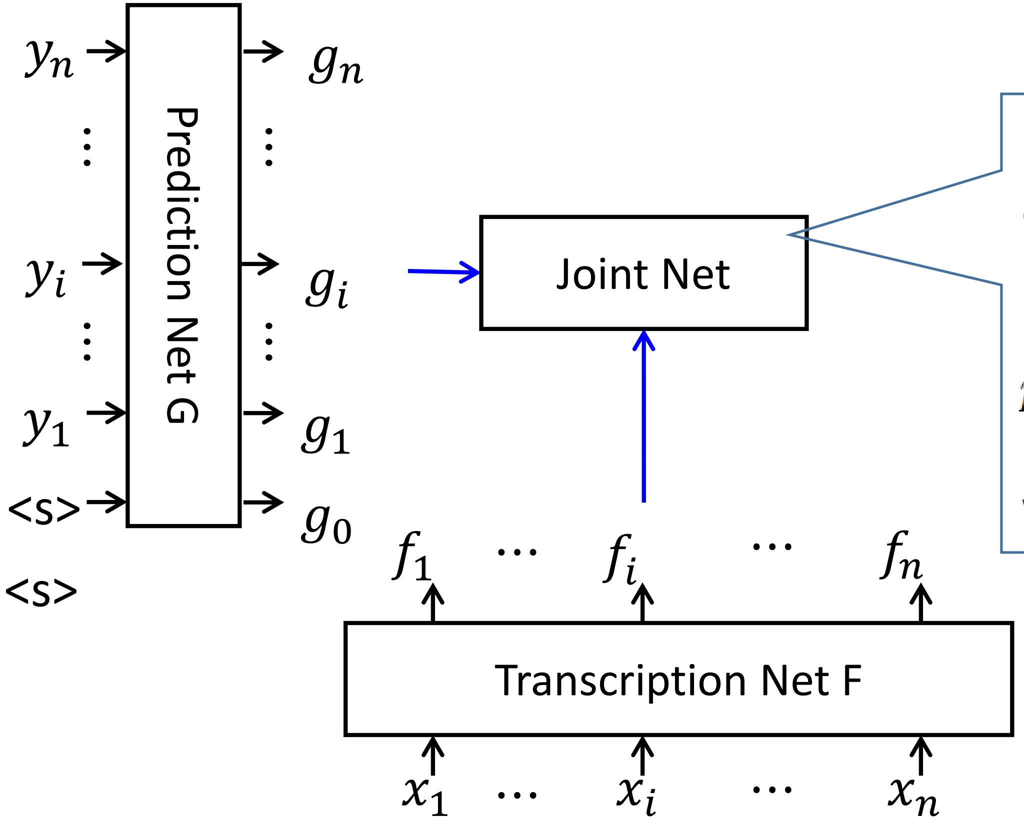
- RNN Transducers (RNNT)
  - Originally developed for general sequence-to-sequence learning, which do not assume that the input and output sequences are of equal lengths and aligned, e.g., in speech recognition
  - In the following, we introduce RNN transducers in a simple form for applications in sequence labeling: i.e., for the aligned setting: one label for one observation in each position
  - Similar idea can be used to revise general RNNT

A. Graves, "Sequence transduction with recurrent neural networks," ICML 2012 Workshop on Representation Learning.

RNNT: introducing prediction network for labels

- Motivation: extending CTC by considering output-output dependencies
- Introduce the **prediction network**, which attempts to model each output in $y_{1:n}$ given the previous ones

$$p(y|x) = \prod_{i=1}^{n} p(y_i|y_{0:i-1}, x)$$



Defines the conditional output distribution:

$$JointNet(\cdot\,|i)$$

$$p(y_i = k|y_{0:i-1}, x) = \frac{\exp(f_i^k + g_i^k)}{\sum_{k'=1}^{K} \exp(f_i^{k'} + g_i^{k'})}$$

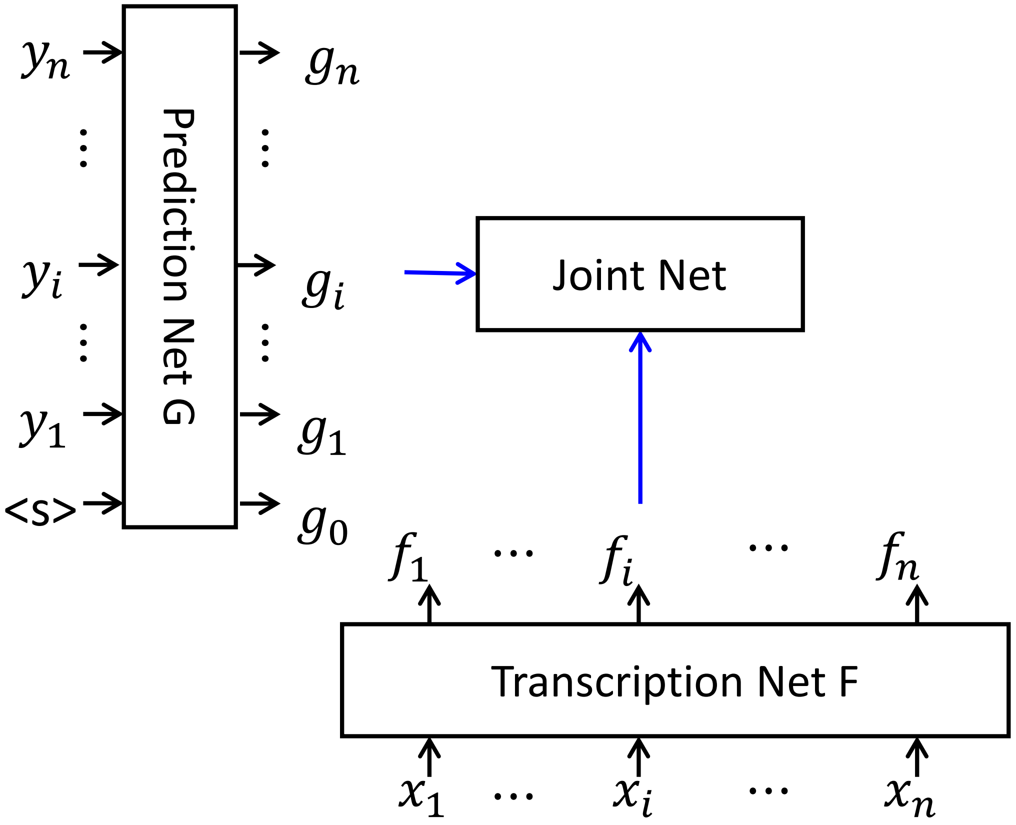
which is softmax over K labels.

y_0 is the special token <s>

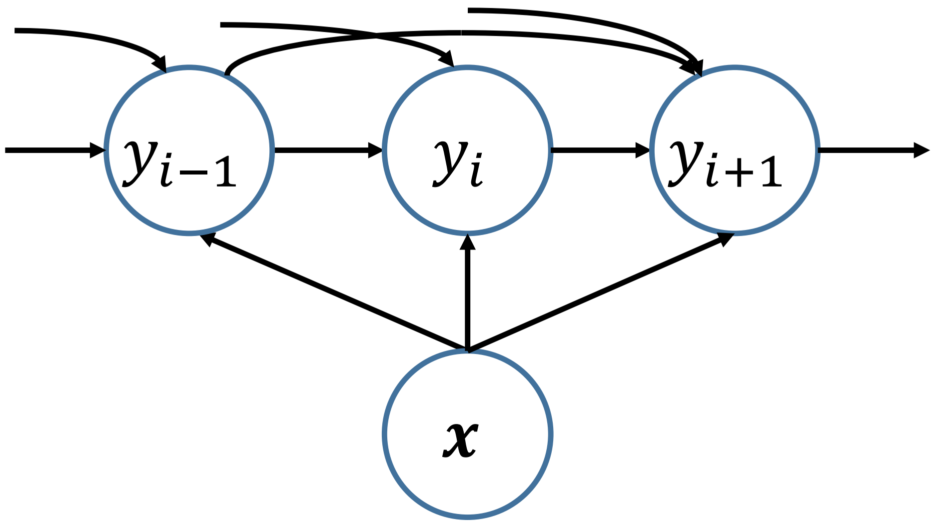
$f_i, g_i \in \mathbb{R}^K$

A. Graves, "Sequence transduction with recurrent neural networks," ICML 2012 Workshop on Representation Learning.

RNNT: shortcoming



$$p(y|x) = \prod_{i=1}^{n} p(y_i|y_{0:i-1}, x)$$



Graphical Model Representation

- As directed sequential model /Auto-regressive model, RNNT potentially suffers from Exposure Bias and Label Bias. A recent effort in [Cui, et al., 2021].

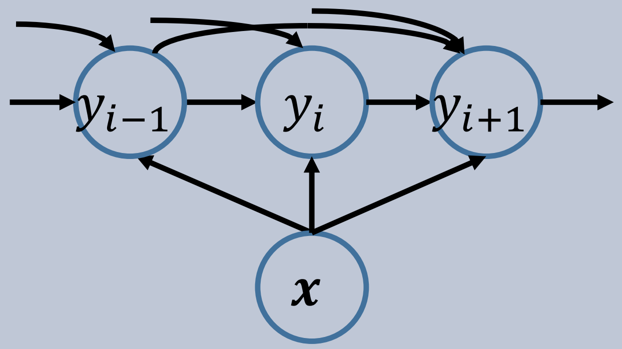
X. Cui, et al., "Reducing Exposure Bias in Training Recurrent Neural Network Transducers", INTERSPEECH 2021.

Neural CRF Transducer

$$p(y|x) = \prod_{i=1}^{n} p(y_i|y_{0:i-1}, x)$$

$$p(y|x;\theta) = \frac{\exp\{u(y,x;\theta)\}}{Z(x;\theta)}, \text{where } Z(x;\theta) = \sum_{y'} \exp\{u(y',x;\theta)\}$$

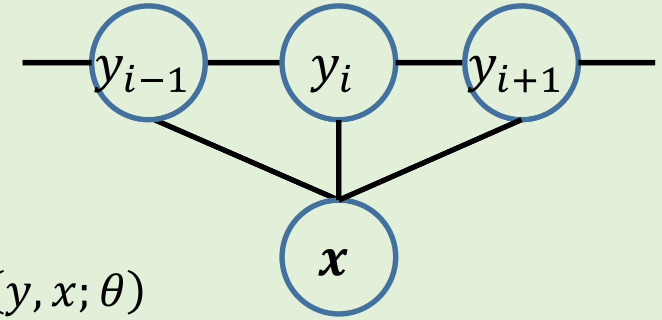
RNNT



$$p(y_i = k|y_{0:i-1}, x) = \frac{\exp(f_i^k + g_i^k)}{\sum_{k'=1}^{K} \exp(f_i^{k'} + g_i^{k'})}$$

Local conditional

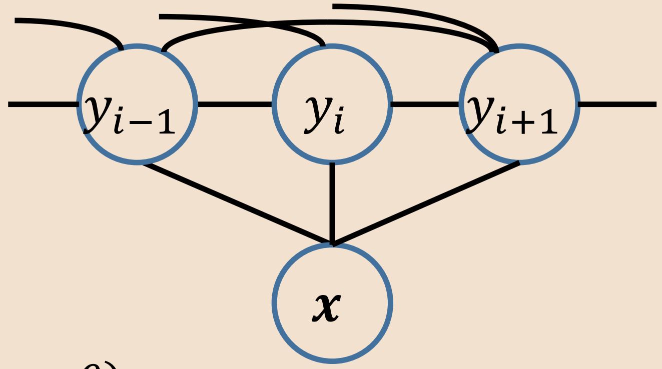
Linear-chain CRF



$$u(y,x;\theta) = \sum_{i=1}^{n} \{\phi_i(y_i,x;\theta) + \psi_i(y_{i-1}, y_i;\theta)\}$$

Local log-potential: $f_i^k + A_{jk}$
for $y_{i-1}=j$, $y_i=k$

Neural CRF Transducer



$$u(y,x;\theta) = \sum_{i=1}^{n} \{\phi_i(y_i,x;\theta) + \psi_i(y_{0:i-1}, y_i;\theta)\}$$

Local log-potential: $f_i^k + g_i^k$ for $y_i=k$

Kai Hu, Zhijian Ou, et al. Neural CRF Transducers for Sequence Labeling. ICASSP, 2019.

Neural CRF Transducer: training and decoding

Training

- Negative log-likelihood over input seq. x and oracle label seq. y^*

$$L(y^*;\theta) = -u(y^*,x;\theta) + logZ(x;\theta)$$

- Monte Carlo Method

$$\nabla_\theta logZ(x;\theta) = E_{p(y'|x;\theta)}[\nabla_\theta u(y',x;\theta)]$$

- Beam search with early updates [b]

$$L(y^*_{1:j};\theta) = -u(y^*_{1:j};\theta) + log\sum_{y'\epsilon\mathcal{B}_j}\exp\{u(y'_{1:j};\theta)\}$$

$\mathcal{B}_j$ contains all paths in the beam at step j, together with the oracle path prefix $y^*_{1:j}$

Decoding: beam search

a. Kai Hu, Zhijian Ou, et al. Neural CRF Transducers for Sequence Labeling. ICASSP, 2019.
b. Andor, Alberti, et al., "Globally Normalized Transition-Based Neural Networks", ACL 2016.

Experiment and Conclusion

Task / Model	POS (Accuracy)	Chunking (F1 score)	English NER (F1 score)	Dutch NER (F1 score)	Globally normalized	Long-range dependencies
Linear-chain CRF	97.52	95.01	91.11	81.53	√	×
RNN Transducer	97.50	95.02	91.02	81.59	×	√
CRF Transducer	97.52	95.14	91.40	81.84	√	√

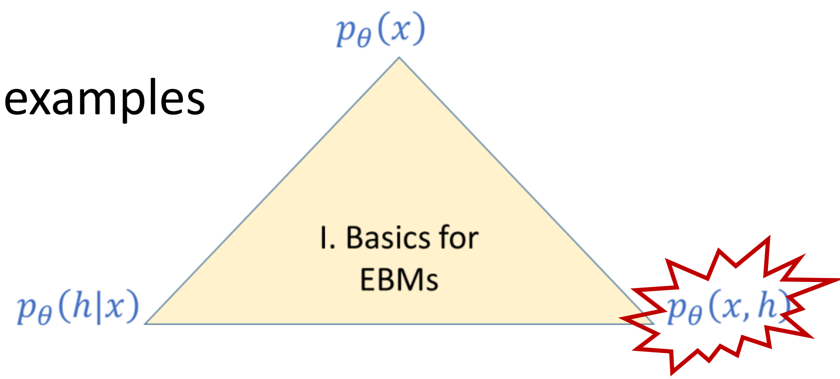
Experiment results show that CRF transducers achieve consistent improvements over linear-chain CRFs and RNN transducers across four sequence labeling tasks, and obtain state-of-theart results.

Reproducible code is at https://github.com/thu-spmi/SPMISeq

Kai Hu, Zhijian Ou, et al. Neural CRF Transducers for Sequence Labeling. ICASSP, 2019.

Content

I. Basics for EBMs (45 min)
1. Probabilistic graphical modeling (PGM) framework and EBM model examples
2. Learning EBMs by Monte Carlo methods
3. Learning EBMs by noise-contrastive estimation (NCE)



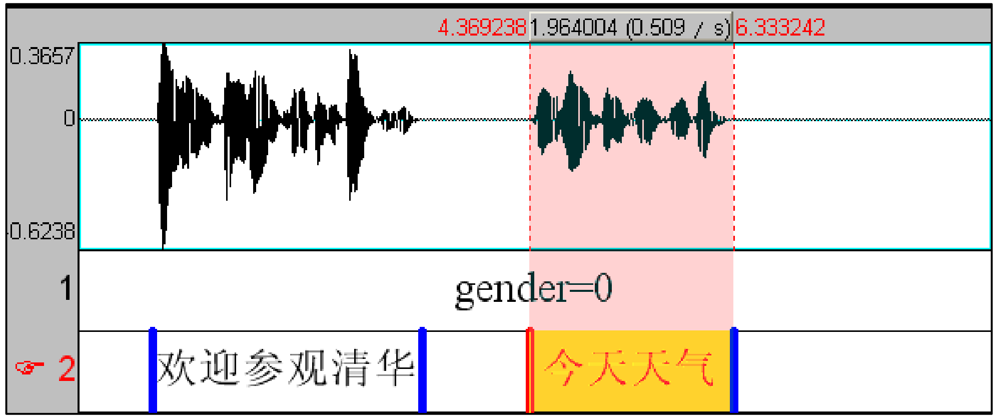
II. EBMs for language modeling (45 min)
1. Trans-dimensional random field (TRF) LMs for speech recognition
2. Residual energy-based models for text generation
3. Electric: an energy-based cloze model for representation learning over text

III. EBMs for speech recognition and natural language labeling (45 min)
1. CRFs as conditional EBMs
2. CRFs for speech recognition
3. CRFs for sequence labeling in NLP

IV. EBMs for semi-supervised natural language labeling (45 min)
1. Upgrading EBMs to Joint EBMs (JEMs) for fixed-dimensional data
2. Upgrading CRFs to Joint random fields (JRFs) for sequential data
3. Comparison of joint-training and pre-training for semi-supervised learning via EBMs

Supervised learning from Labeled data $\{(x_j, y_j), j = 1, 2, \cdots\}$

Tremendous Success!

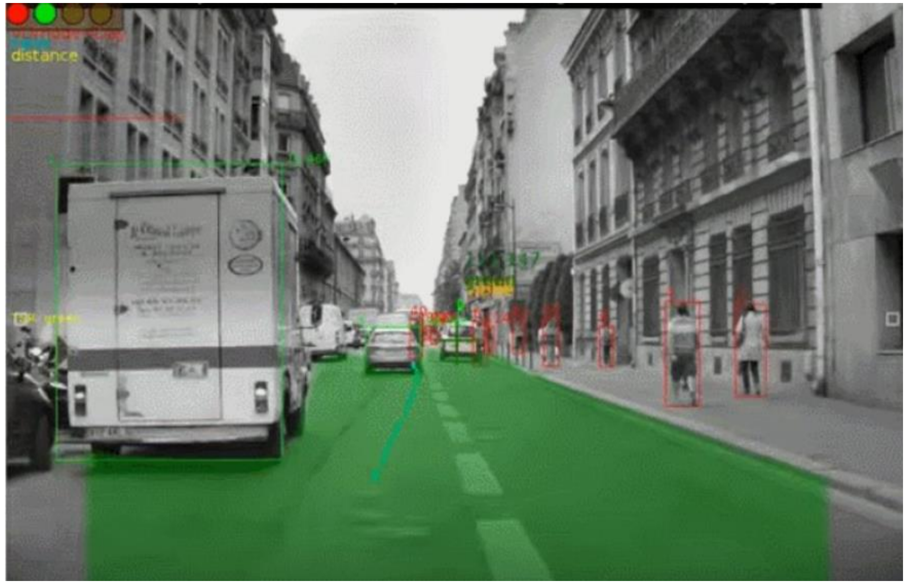


Speech Recognition

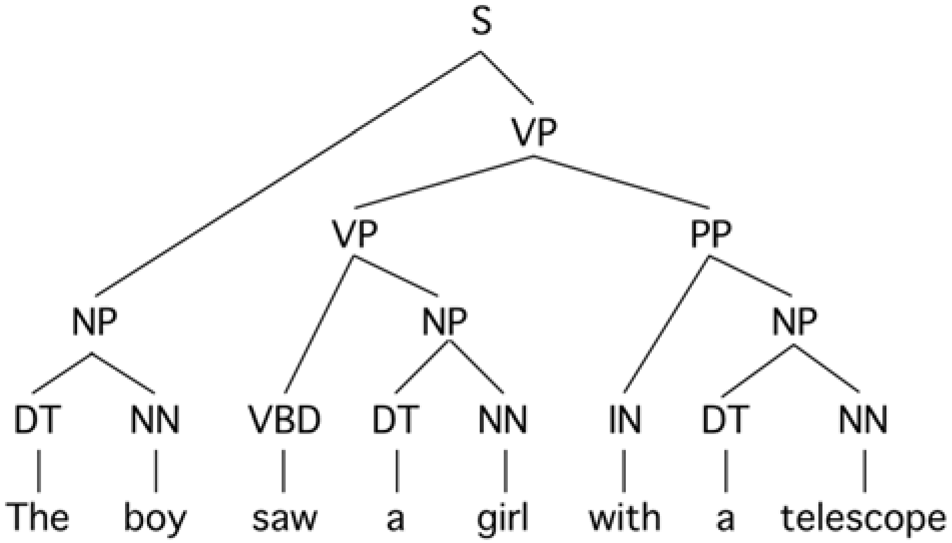
ATIS UTTERANCE EXAMPLE IOB REPRESENTATION

Sentence	*show*	*flights*	*from*	*Boston*	*To*	*New*	*York*	*today*
Slots/Concepts	O	O	O	B-dept	O	B-arr	I-arr	B-date
Named Entity	O	O	O	B-city	O	B-city	I-city	O
Intent	*Find_Flight*							
Domain	*Airline Travel*							

Intent Detection, Slot Filling, Named Entity Recognition

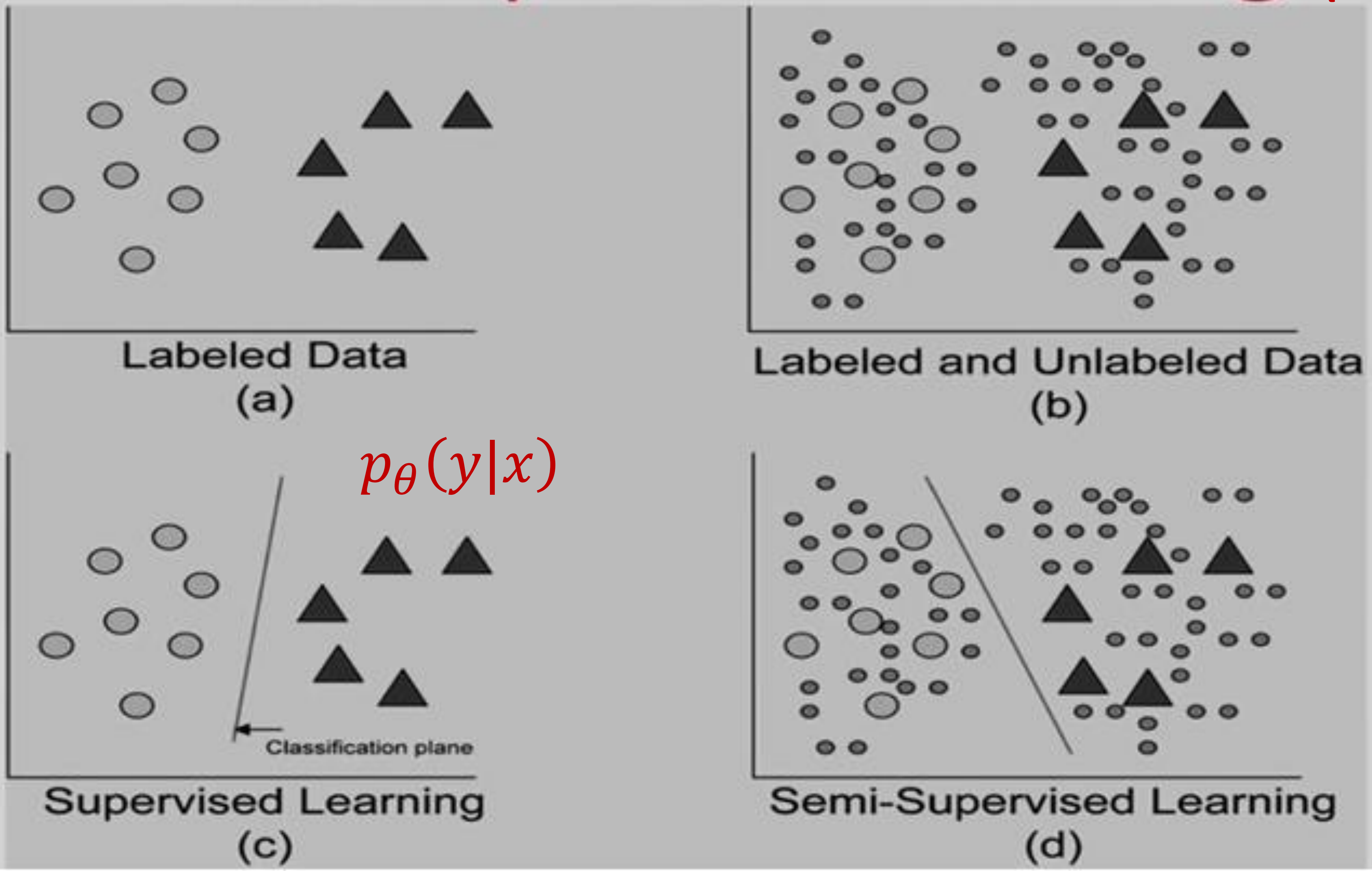


Object Detection and Tracking



Syntactic Parsing

Semi-supervised learning (SSL)



The key to designing SSL methods is:
How to effectively exploit the information contained in the unlabeled data $\{x\}$, which can provide <u>priors/regularizations/inductive biases</u> for finding the posterior $p_\theta(y|x)$.

Xiaojin Zhu, "Semi-supervised learning literature survey," Technical report, University of Wisconsin-Madison, 2006.

SSL methods (for using DNNs)

- Recent SSL methods with DNNs can be distinguished by the priors they adopt, and, can be divided into two classes.
  - **Generative SSL**
  - **Discriminative SSL**: The outputs from the discriminative classifier are smooth with respect to local and random perturbations of the inputs [1-5].

[1] Takeru Miyato, et al, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," TPAMI, 2018.
[2] Samuli Laine and Timo Aila, "Temporal ensembling for semisupervised learning," ICLR, 2017.
[3] Antti Tarvainen and Harri Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," NIPS, 2017.
[4] Kihyuk Sohn, David Berthelot, Chun-Liang Li, and et al, "FixMatch: Simplifying semi-supervised learning with consistency and confidence," arXiv:2001.07685, 2020.
[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," arXiv:2002.05709, 2020.

Discriminative SSL

- Recent SSL methods with DNNs can be distinguished by the priors they adopt, and, can be divided into two classes.
  - **Generative SSL**
  - **Discriminative SSL**: The outputs from the discriminative classifier are smooth with respect to local and random perturbations of the inputs.

☹ heavily rely on domain-specific data augmentations, which are tuned intensively for images leading to impressive performance in some image domains
☹ less successful for other domains where these augmentations are less effective (e.g., medical images and text). For instance, random input perturbations are more difficult to apply to discrete data like text [6].

[6] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc Le, "Semi-supervised sequence modeling with cross-view training," in EMNLP, 2018

Generative SSL - Basics

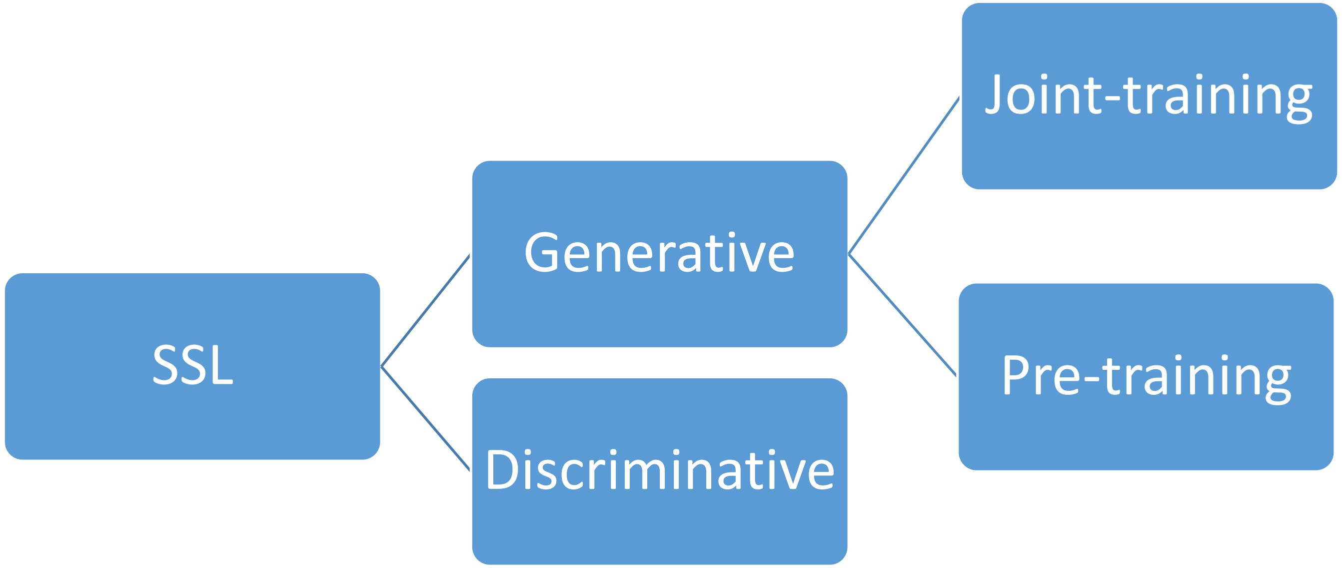
- Exploit unsupervised learning of generative models over unlabeled data, blend unsupervised learning and supervised learning.

☺ inherently not require data augmentations and generally can be applied to a wider range of domains.
☺ make fewer domain-specific assumptions and tend to be domain-agnostic.

Generative SSL - Two Different Approaches

- **Joint-training**
  - A joint model of $p(x,y)$ is defined.
  - When we have label y, we maximize $p(y|x)$ (the supervised objective), and when the label is unobserved, we marginalize it out and maximize $p(x)$ (the unsupervised objective).
  - Semi-supervised learning over a mix of labeled and unlabeled data is formulated as maximizing the (weighted) sum of $\log p(y|x)$ and $\log p(x)$.
- **Pre-training**
  - Only defines $p(x)$ without y.
  - Perform unsupervised representation learning (called pre-training) on unlabeled data, followed by supervised training (called fine-tuning) on labeled data.
  - This manner of pre-training followed by fine-tuning has received increasing application in Natural Language Processing.

There are many open questions in designing semi-supervised methods for particular tasks !



Directed models	Undirected models
Latent Variable Model (LVM), e.g. LABES	**Joint Random Field (JRF), Joint Energy-based model (JEM)**
Auto-Regressive Language Model, e.g. GPT **Masked Language Model**, e.g. BERT	**Random-field Language Model**, e.g. TRF, Electric

[LABES] Y. Zhang, Z. Ou, et al. A Probabilistic End-To-End Task-Oriented Dialog Model with Latent Belief States towards Semi-Supervised Learning. EMNLP, 2020.
[JRF] Y. Song, Z. Ou, et al. Upgrading CRFs to JRFs and its benefits to sequence modeling and labeling. ICASSP, 2020.
[JEM] S. Zhao, J.H. Jacobsen, et al. Joint energy-based models for semi-supervised classification. ICML Workshop on Uncertainty and Robustness in Deep Learning, 2020.
[TRF] B. Wang, Z. Ou. Improved training of neural trans-dimensional random field language models with dynamic noise-contrastive estimation. SLT, 2018.
[Electric] K. Clark, M.T. Luong, et al. Pre-Training Transformers as Energy-Based Cloze Models. EMNLP, 2020.

Table 1. Applications of EBMs across different domains: comparison and connection (See text for details).

	Image classification	Natural language labeling
Observation	$x \in \mathbb{R}^D$ continuous, fixed-dimensional	$x \in \bigcup_l \mathbb{V}^l$ discrete, sequence
Label	$y \in \{1, 2, \cdots, K\}$	$y \in \bigcup_l \{1, 2, \cdots, K\}^l$
Pre-training	① $u_\theta(x) = w^T h$	③ $u_\theta(x)$ in Eq.(3)
Joint-training	② $u_\theta(x, y) = \Psi_\theta(x)[y]$	④ $u_\theta(x, y)$ in Eq.(6)

Yunfu Song, Huahuan Zheng, Zhijian Ou. An empirical comparison of joint-training and pre-training for domain-agnostic semi-supervised learning via energy-based models. MLSP, 2021.

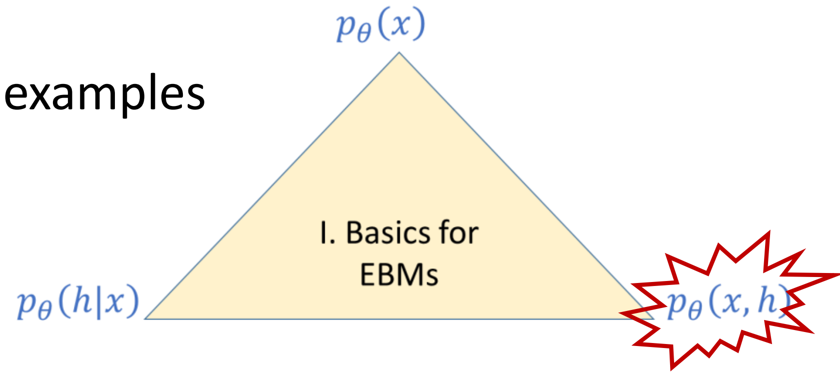
Content

I. Basics for EBMs (45 min)
1. Probabilistic graphical modeling (PGM) framework and EBM model examples
2. Learning EBMs by Monte Carlo methods
3. Learning EBMs by noise-contrastive estimation (NCE)

II. EBMs for language modeling (45 min)
1. Trans-dimensional random field (TRF) LMs for speech recognition
2. Residual energy-based models for text generation
3. Electric: an energy-based cloze model for representation learning over text

III. EBMs for speech recognition and natural language labeling (45 min)
1. CRFs as conditional EBMs
2. CRFs for speech recognition
3. CRFs for sequence labeling in NLP

IV. EBMs for semi-supervised natural language labeling (45 min)
1. Upgrading EBMs to Joint EBMs (JEMs) for fixed-dimensional data
2. Upgrading CRFs to Joint random fields (JRFs) for sequential data
3. Comparison of joint-training and pre-training for semi-supervised learning via EBMs



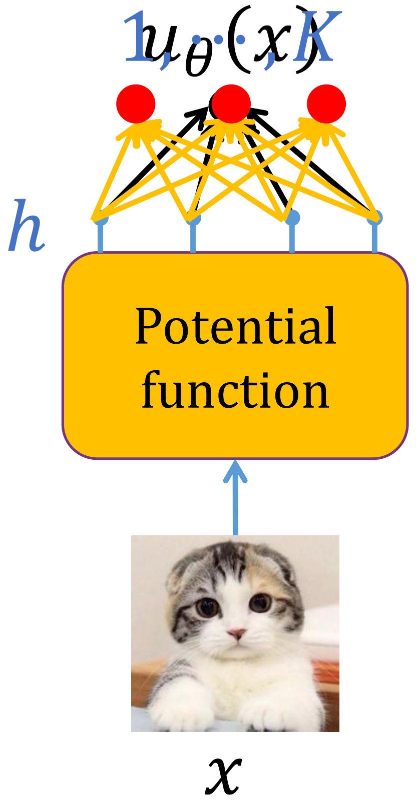
① Pre-training of an EBM for semi-supervised image classification

1) Pre-training: estimate $p_\theta(x)$ over unlabeled images

$$p_\theta(x) = \frac{1}{Z(\theta)} exp[u_\theta(x)]$$

Use a feedforward NN to implement $u_\theta(x): \mathbb{R}^d \to \mathbb{R}$

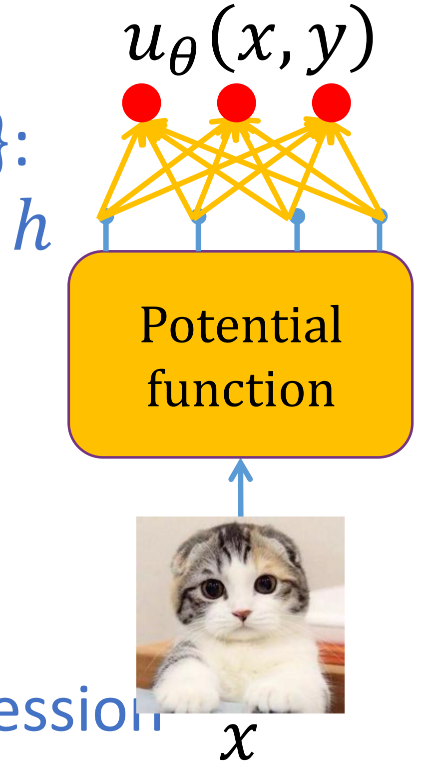
which, in the final layer, calculates $u_\theta(x) = w^T h$ via a linear layer.



2) Fine-tuning: throw w and fed h into an new linear output layer, followed by softmax(Wh), to predict $y \in \{1, \cdots, K\}$, where $W \in \mathbb{R}^{K \times H}$

Yunfu Song, Zhijian Ou. Learning Neural Random Fields with Inclusive Auxiliary Generators. arXiv:1806.00271, 2018.

② Joint-training of an EBM for semi-supervised image classification



- Joint modeling of observation $x \in \mathbb{R}^d$ and class label $y \in \{1, \cdots, K\}$:

$$p_\theta(x,y) = \frac{1}{Z(\theta)} exp[u_\theta(x,y)]$$

- Consider a NN $\Psi_\theta(x): \mathbb{R}^d \to \mathbb{R}^K$ and define:

$$u_\theta(x,y) = \Psi_\theta(x)[y]$$

- **Classifier**: $p_\theta(y|x) = \frac{p_\theta(x,y)}{p_\theta(x)} = \frac{exp[u_\theta(x,y)]}{\sum_y exp[u_\theta(x,y)]}$, like a K-class logistic regression

Marginal density: $p_\theta(x) = \frac{1}{Z(\theta)} exp[u_\theta(x)]$, where $u_\theta(x) \triangleq log \sum_y exp[u_\theta(x,y)]$

$$\begin{cases} \min_\theta KL[\tilde{p}(\tilde{x})||p_\theta(\tilde{x})] - \alpha \sum_{(\tilde{x},\tilde{y})\sim\mathcal{L}} log p_\theta(\tilde{y}|\tilde{x}) \\ \min_\phi KL\left[p_\theta(x)||q_\phi(x)\right] \end{cases}$$

Yunfu Song, Zhijian Ou. Learning Neural Random Fields with Inclusive Auxiliary Generators. arXiv:1806.00271, 2018.

[Submitted on 1 Jun 2018 (this version), latest version 22 Jul 2020 (v5)]

Learning Neural Random Fields with Inclusive Auxiliary Generators

Yunfu Song, Zhijian Ou

In this paper we develop Neural Random Field learning with Inclusive-divergence minimized Auxiliary Generators (NRF-IAG), which is under-appreciated in the literature. The contributions are two-fold. First, we rigorously apply the stochastic approximation algorithm to solve the joint optimization and provide theoretical justification. The new approach of learning NRF-IAG achieves superior unsupervised learning performance competitive with state-of-the-art deep generative models (DGMs) in terms of sample generation quality. Second, semi-supervised learning (SSL) with NRF-IAG gives rise to strong classification results comparable to state-of-art DGM-based SSL methods, and simultaneously achieves superior generation. This is in contrast to the conflict of good classification and good generation, as observed in GAN-based SSL.

Published as a conference paper at ICLR 2020

YOUR CLASSIFIER IS SECRETLY AN ENERGY BASED MODEL AND YOU SHOULD TREAT IT LIKE ONE

Will Grathwohl
University of Toronto & Vector Institute
Google Research
wgrathwohl@cs.toronto.edu

Kuan-Chieh Wang*& Jörn-Henrik Jacobsen*
University of Toronto & Vector Institute
wangkua1@cs.toronto.edu
j.jacobsen@vectorinstitute.ai

David Duvenaud
University of Toronto & Vector Institute
duvenaud@cs.toronto.edu

Kevin Swersky & Mohammad Norouzi
Google Research
{kswersky, mnorouzi}@google.com

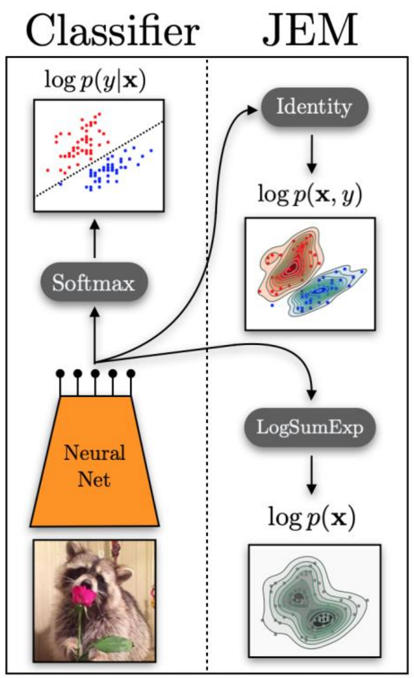
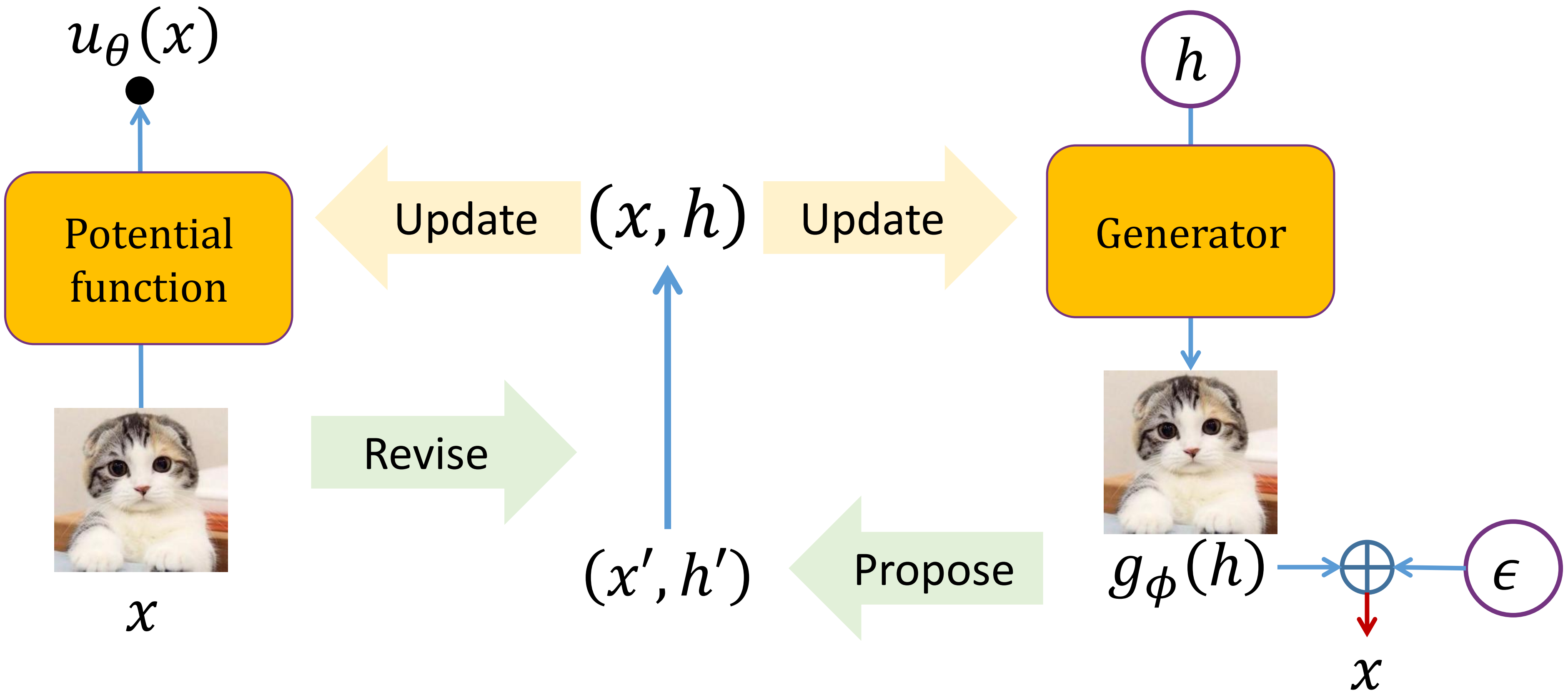


Figure 1: Visualization of our method, JEM, which defines a joint EBM from classifier architectures.

Inclusive-NRF algo. for learning from continuous data, e.g., Images.

simultaneously training a random field and a generator.



$$\begin{cases} \min_\theta KL[\tilde{p}(\tilde{x})||p_\theta(\tilde{x})] \\ \min_\phi KL\big[p_\theta(x)||q_\phi(x)\big] \end{cases} \Rightarrow \begin{cases} \nabla_\theta = E_{\tilde{p}(\tilde{x})}[\nabla_\theta log p_\theta(\tilde{x})] = E_{\tilde{p}(\tilde{x})}[\nabla_\theta u_\theta(\tilde{x})] - E_{p_\theta(x)}[\nabla_\theta u_\theta(x)] \\ \nabla_\phi = E_{p_\theta(x)}\big[\nabla_\phi log q_\phi(x)\big] = E_{p_\theta(x) q_\phi(h|x)}\big[\nabla_\phi log q_\phi(x,h)\big] \end{cases}$$

Yunfu Song, Zhijian Ou. Learning Neural Random Fields with Inclusive Auxiliary Generators. arXiv:1806.00271, 2018.

Content

I. Basics for EBMs (45 min)

1. Probabilistic graphical modeling (PGM) framework and EBM model examples (classic & modern)
2. Learning EBMs by Monte Carlo methods
3. Learning EBMs by noise-contrastive estimation (NCE)

II. EBMs for language modeling (45 min)

1. Trans-dimensional random field (TRF) LMs for speech recognition
2. Residual energy-based models for text generation
3. Electric: an energy-based cloze model for representation learning over text

III. EBMs for speech recognition and natural language labeling (45 min)

1. CRFs as conditional EBMs
2. CRFs for speech recognition
3. CRFs for sequence labeling in NLP

IV. EBMs for semi-supervised natural language labeling (45 min)

1. Upgrading EBMs to Joint EBMs (JEMs) for fixed-dimensional data
2. Upgrading CRFs to Joint random fields (JRFs) for sequential data
3. Comparison of joint-training and pre-training for semi-supervised learning via EBMs

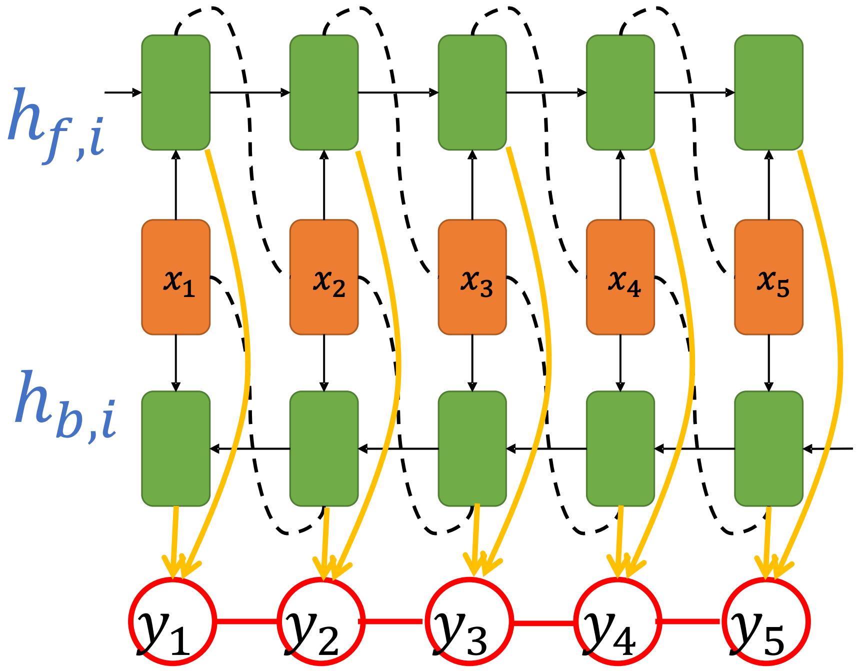
③ Pre-training of an EBM for semi-supervised natural language labeling

1) Pre-training: estimate $p_\theta(x)$ over unlabeled sentences $x = (x_1, \cdots, x_l)$

$$p_\theta(x) = \frac{1}{Z(\theta)} exp[u_\theta(x)]$$

Use a B-LSTM to implement $u_\theta(x)$: $\mathbb{V}^l \rightarrow \mathbb{R}$

$$u_\theta(x) = \sum_{i=1}^{l-1} h_{f,i}^T e_{i+1} + \sum_{i=2}^{l} h_{b,i}^T e_{i-1}$$



2) Fine-tuning: we add a CRF on top of the extracted representations $\{(h_{f,i}, h_{b,i}), i = 1, \cdots, l\}$ to predict label sequence $y = (y_1, \cdots, y_l)$.

Bin Wang, Zhijian Ou. Improved training of neural trans-dimensional random field language models with dynamic noise-contrastive estimation. IEEE Workshop on Spoken Language Technology (SLT), 2018.

④ Joint-training of an EBM for semi-supervised natural language labeling

- JRF: Define a joint distribution over $x = (x_1, \cdots, x_l)$ and $y = (y_1, \cdots, y_l)$

$$p_\theta(l, x^l, y^l) = \pi_l p_\theta(x^l, y^l; l) = \frac{\pi_l}{Z_\theta(l)} \exp\left(u_\theta(x^l, y^l)\right)$$

- Consider a NN $\Psi_\theta(x): \mathbb{V}^l \rightarrow \mathbb{R}^{l \times K}$ and define:

$$u_\theta(x, y) = \sum_{i=1}^{l} \Psi_\theta(x)[i, y_i] + \sum_{i=1}^{l} A[y_{i-1}, y_i]$$

- From JRF we have:

$$p_\theta(y^l | x^l) = \frac{1}{\sum_{y^l} \exp(u_\theta(x^l, y^l))} \exp\left(u_\theta(x^l, y^l)\right)$$

which is a CRF

- From JRF we have

$$p_\theta(l, x^l) = \frac{\pi}{Z_\theta}$$

$$= \frac{\pi_l}{Z_\theta(l)} \exp\big($$

where $u_\theta(x^l) = \log \sum$

which is a trans-dim

Recently: neural CRFs

Use NN to extract features

$$\mathrm{LSTM}(x_{1:l}): x_{1:l} \rightarrow h_{1:l}$$

- Node potential, calculated via a linear layer

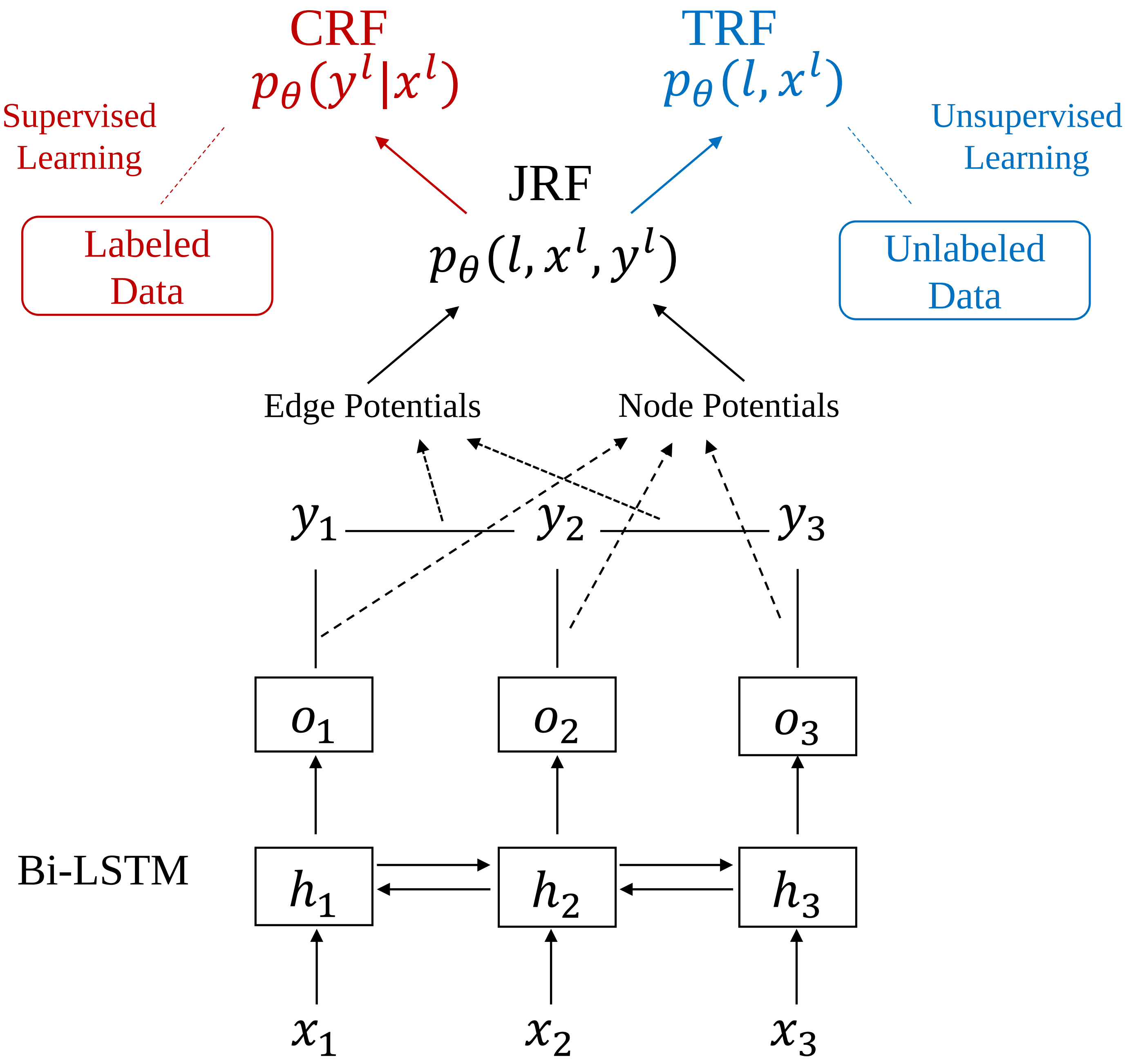
$$\phi_t(y_t = k, x) = w_k^{\mathrm{T}} h_t \triangleq \phi_t^k$$

w_k is the weight vector for label k

- Edge potential, mostly implemented as a matrix A

Yunfu Song, Zhijian Ou, et al. Upgrading CRFs to JRFs and its benefits to sequence mode

Upgrading CRFs to Joint random fields (JRFs) for sequential data



Dynamic NCE algo. for learning from discrete data, e.g., texts.

Simultaneously train a random field and a generator.

- The target RF model $p_\theta(x) = \frac{1}{Z(\theta)} e^{u_\theta(x)}$
- Treat $\log Z(\theta)$ as a parameter ζ and rewrite $p_{\theta,\zeta}(x) \propto e^{u_\theta(x)-\zeta}$
- Introduce a **noise distribution** $q(x)$, and consider a binary classification

$$P(C=0|x) = \frac{p_{\theta,\zeta}(x)}{p_{\theta,\zeta}(x) + \nu q(x)}, where\ \ \nu = \frac{P(C=1)}{P(C=0)}$$

$$P(C=1|x) = 1 - P(C=0|x)$$

- Noise Contrastive Estimation (NCE):

$$\max_{\theta,\zeta} E_{x\sim p_0(x)}[\log P(C=0|x)] + E_{x\sim q(x)}[\log P(C=1|x)]$$

- Consistency: $p_\theta \to p_0$ (oracle), under infinite amount of data and infinite capacity of p_θ.
- Reliable NCE needs a large ν≈20; Dynamic-NCE works well with ν=1.

Content

I. Basics for EBMs (45 min)

1. Probabilistic graphical modeling (PGM) framework and EBM model examples (classic & modern)
2. Learning EBMs by Monte Carlo methods
3. Learning EBMs by noise-contrastive estimation (NCE)

II. EBMs for language modeling (45 min)

1. Trans-dimensional random field (TRF) LMs for speech recognition
2. Residual energy-based models for text generation
3. Electric: an energy-based cloze model for representation learning over text

III. EBMs for speech recognition and natural language labeling (45 min)

1. CRFs as conditional EBMs
2. CRFs for speech recognition
3. CRFs for sequence labeling in NLP

IV. EBMs for semi-supervised natural language labeling (45 min)

1. Upgrading EBMs to Joint EBMs (JEMs) for fixed-dimensional data
2. Upgrading CRFs to Joint random fields (JRFs) for sequential data
3. Comparison of joint-training and pre-training for semi-supervised learning via EBMs

EBM models can be very flexibly defined for SSL, by either of joint-training and pre-training.

… previously known in the literature†, but it is unclear which is better when evaluated in a common experimental setup.

To the best of our knowledge, this paper‡ is the first to systematically compare joint-training and pre-training for EBM-based for SSL, across domains (image classification and natural language labeling).

† EBM based SSL results have been reported across different data modalities (images, natural languages, an protein structure prediction and year prediction from the UCI dataset repository) [12,13,14].

‡ Yunfu Song, Huahuan Zheng, Zhijian Ou. An empirical comparison of joint-training and pre-training for domain-agnostic semi-supervised learning via energy-based models. MLSP, 2021.

Table 1. Applications of EBMs across different domains: comparison and connection (See text for details).

	Image classification	Natural language labeling
Observation	$x \in \mathbb{R}^D$ continuous, fixed-dimensional	$x \in \bigcup_l \mathbb{V}^l$ discrete, sequence
Label	$y \in \{1, 2, \cdots, K\}$	$y \in \bigcup_l \{1, 2, \cdots, K\}^l$
Pre-training	① $u_\theta(x) = w^T h$	③ $u_\theta(x)$ in Eq.(3)
Joint-training	② $u_\theta(x, y) = \Psi_\theta(x)[y]$	④ $u_\theta(x, y)$ in Eq.(6)

Pre-training aims to learn representations that may be useful for multiple downstream tasks, and any information about the labels is not utilized until the fine-tuning stage.

Yunfu Song, Huahuan Zheng, Zhijian Ou. An empirical comparison of joint-training and pre-training for domain-agnostic semi-supervised learning via energy-based models. MLSP, 2021.

Table 2. SSL for image classification over CIFAR-10 with 4,000 labels. The upper/lower blocks show generative/discriminative SSL methods respectively. The means and standard deviations are calculated over ten independent runs with randomly sampled labels.

Methods	error (%)
CatGAN [30]	19.58±0.46
Ladder network [31]	20.40±0.47
Improved-GAN [32]	18.63±2.32
BadGAN [33]	14.41±0.30
Sobolev-GAN [34]	15.77±0.19
Supervised baseline	25.72±0.44
Pre-training+fine-tuning EBM	21.40±0.38
Joint-training EBM	15.12±0.36
Results below this line cannot be directly compared to those above.	
VAT small [1]	14.87
Temporal Ensembling [2]	12.16±0.31
Mean Teacher [3]	12.31±0.28

Joint-training EBMs outperform pre-training+fine-tuning EBMs by a large margin in this task.

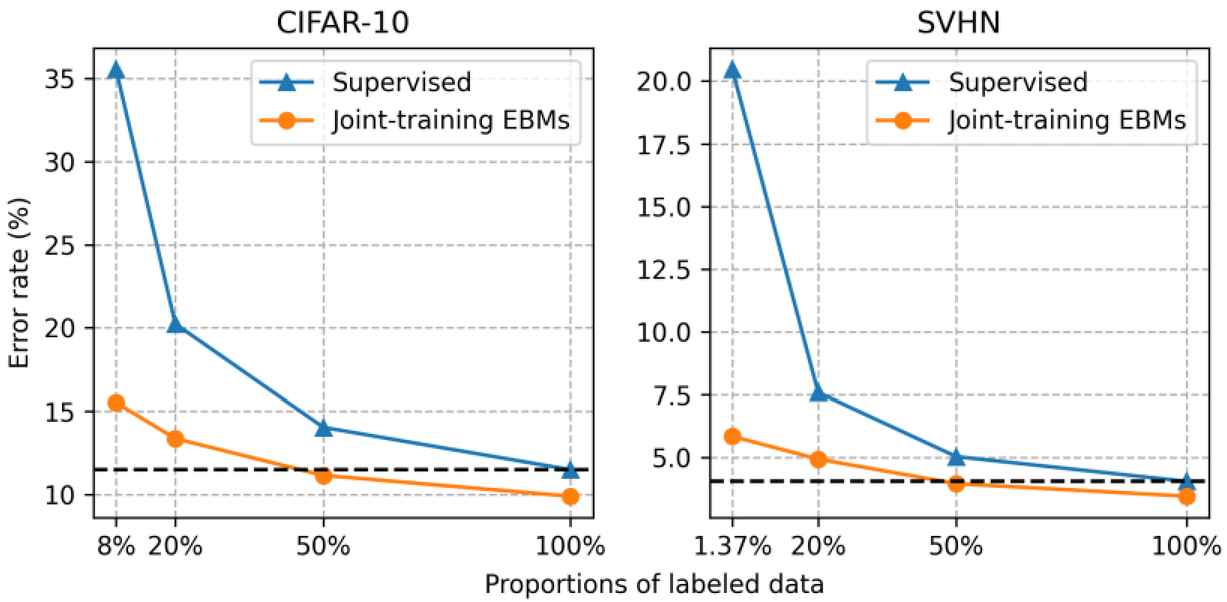


Fig. 1. Error rates of supervised baseline and joint-training EBMs as the amount of labels varies on SVHN and CIFAR-10 datasets. The dash line is the supervised result trained with 100% labeled data.

Can reduce 50% of labels without losing performance.

Table 3. Natural language labeling results. The evaluation metric is accuracy for POS and F_1 for chunking and NER. "Labeled" denotes the amount of labels in terms of the proportions w.r.t. the full set of labels. "U/L" denotes the ratio between the amount of unlabeled and labeled data. "U/L=0" denotes the supervised baseline. "pre." and "joint" denote the results by pre-training+fine-tuning EBMs and joint-training EBMs, respectively.

Labeled	U/L	POS tagging pre.	POS tagging joint	Chunking pre.	Chunking joint	NER pre.	NER joint
2%	0	95.57		78.73		78.19	
	50	95.72	95.92	81.62	82.24	76.74	77.61
	250	95.96	96.13	82.10	82.26	78.49	78.51
	500	96.08	96.24	83.10	83.05	79.47	79.17
10%	0	96.81		90.06		86.93	
	50	96.87	96.99	91.60	91.85	86.37	87.05
	250	96.88	97.00	91.09	91.93	86.86	86.77
	500	96.92	97.08	91.93	92.23	87.57	87.06
100%	0	97.41		94.77		90.74	
	50	97.40	97.49	95.05	95.31	91.24	91.34
	250	97.45	97.54	95.12	95.48	91.19	91.51
	500	97.46	97.57	95.19	95.50	91.30	91.52

Table 4. Relative improvements by joint-training EBMs compared to the supervised baseline (abbreviated as sup.) and pretraining+fine-tuning EBMs respectively. Refer to Table 3 for notations.

		joint over sup.			joint over pre.		
Labeled	U/L	POS	Chunking	NER	POS	Chunking	NER
2%	50	7.9	16.5	-2.7	4.7	3.4	3.7
	250	12.6	16.6	1.5	4.2	0.9	0.1
	500	15.1	20.3	4.5	4.1	-0.3	-1.5
10%	50	5.6	18.0	0.9	3.8	3.0	5.0
	250	6.0	18.3	-1.2	3.8	9.4	-0.7
	500	8.5	21.8	1.0	5.2	3.7	-4.1
100%	50	3.1	10.3	6.5	3.5	5.3	1.1
	250	5.0	13.6	8.3	3.5	7.4	3.6
	500	6.2	14.0	8.4	4.3	6.4	2.5

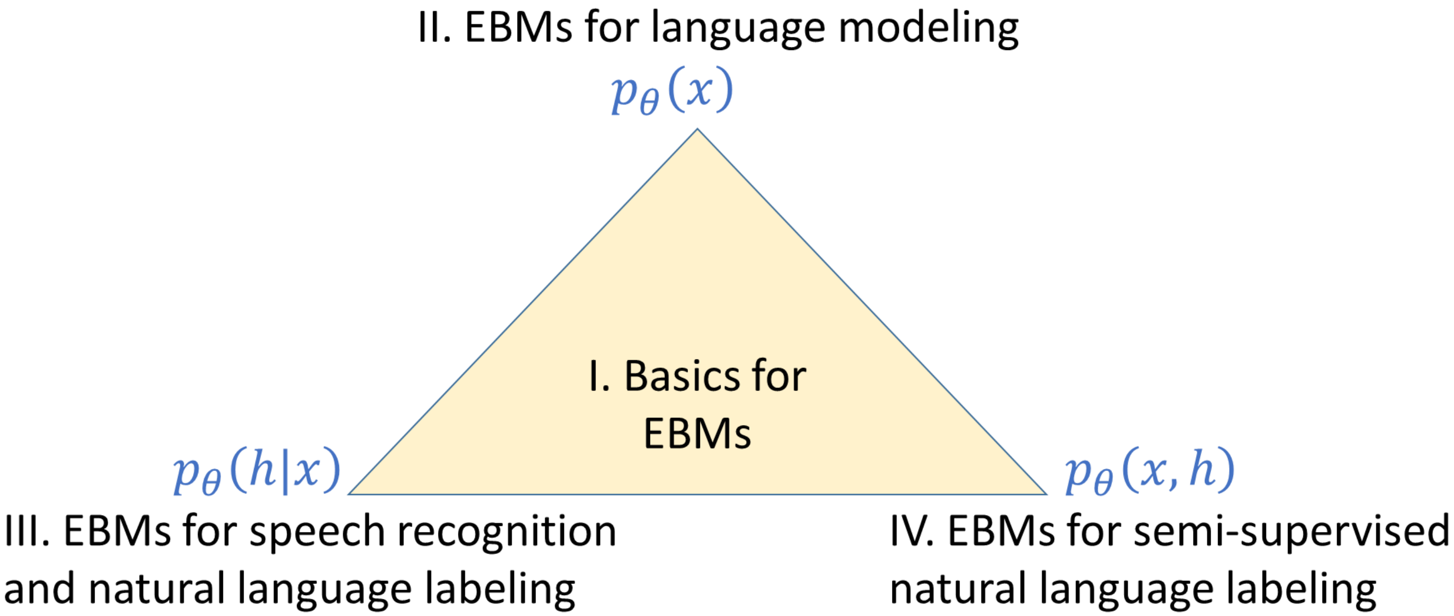
- Joint-training EBMs outperform pre-training EBMs in 23 out of the 27 settings marginally but nearly consistently.
- A possible explanation is that pretraining is not aware of the labels for the targeted task and is thus weakened for representation learning.

Section Conclusion

- We systematically evaluate and compare joint-training and pre-training for EBM-based domain-agnostic SSL, through a suite of experiments across a variety of domains such as image classification and natural language labeling.
- **Joint-training EBMs outperform pre-training EBMs marginally but nearly consistently.**
  - ▶Presumably, this is because that the optimization of joint-training is directly related to the targeted task, but pre-training is not aware of the labels for the targeted task.
- This new finding would be helpful for future work to further explore better methods to leverage unlabeled data.

Reproducible code is at https://github.com/thu-spmi/semi-EBM

Summary



Take-home messages for EBMs/UBMs/RFs

1. Flexibility in modeling
2. Computation efficiency in inference
3. Overcome label bias and exposure bias suffered by locally-normalized models
4. Joint EBMs for generative semi-supervised learning
5. Difficult in model training

We are making progress, and there are many interesting open questions...

Main References

1. D. Koller and N. Friedman. Probabilistic graphical models: principles and techniques. MIT press, 2009.
2. Eric Fosler-Lussier, et al. Conditional random fields in speech, audio, and language processing. Proceedings of the IEEE, 2013.
3. Zhijian Ou. A Review of Learning with Deep Generative Models from Perspective of Graphical Modeling. arXiv:1808.01630, 2018.
4. Bin Wang, Zhijian Ou, Zhiqiang Tan. Trans-dimensional Random Fields for Language Modeling. Annual Meeting of the Association for Computational Linguistics (ACL Long Paper), 2015.
5. Bin Wang, Zhijian Ou, Zhiqiang Tan. Learning Trans-dimensional Random Fields with Applications to Language Modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2018, vol.40, no.4, pp.876-890. https://github.com/thu-spmi/SPMILM
6. Bin Wang, Zhijian Ou. Language modeling with neural trans-dimensional random fields. ASRU, 2017.
7. Bin Wang, Zhijian Ou. Learning neural trans-dimensional random field language models with noise-contrastive estimation. ICASSP, 2018.
8. Bin Wang, Zhijian Ou. Improved training of neural trans-dimensional random field language models with dynamic noise-contrastive estimation. SLT, 2018.
9. Silin Gao, Zhijian Ou, et al. Integrating discrete and neural features via mixed-feature trans-dimensional random field language models. ICASSP (Oral), 2020.
10. Yunfu Song, Zhijian Ou. Learning Neural Random Fields with Inclusive Auxiliary Generators. arXiv:1806.00271, 2018. https://github.com/thu-spmi/Inclusive-NRF
11. Hongyu Xiang, Zhijian Ou. CRF-based Single-stage Acoustic Modeling with CTC Topology. ICASSP (Oral), 2019. https://github.com/thu-spmi/CAT
12. Kai Hu, Zhijian Ou, Min Hu, Junlan Feng. Neural CRF Transducers for Sequence Labeling. ICASSP, 2019. https://github.com/thu-spmi/SPMISeq
13. Yunfu Song, Zhijian Ou, Zitao Liu, Songfan Yang. Upgrading CRFs to JRFs and its benefits to sequence modeling and labeling. ICASSP, 2020. https://github.com/thu-spmi/semi-EBM
14. Yunfu Song, Huahuan Zheng, Zhijian Ou. An empirical comparison of joint-training and pre-training for domain-agnostic semi-supervised learning via energy-based models. IEEE Machine Learning for Signal Processing Workshop (MLSP), 2021. https://github.com/thu-spmi/semi-EBM
15. Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. ICLR 2020.
16. Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. Residual energy-based models for text generation, ICLR 2020.
17. Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Pre-training transformers as energy-based cloze models, EMNLP 2020.

Thanks for your attention !

Thanks to my collaborators and students :

Zhiqiang Tan, Bin Wang, Hongyu Xiang, Yunfu Song, Kai Hu,

Keyu An, Huahuan Zheng, Silin Gao, Wenjie Peng