Contrastive Divergence learning, Product models, and Deep Belief Nets

Daichi Mochihashi

NTT Communication Science Laboratories

daichi@cslab.kecl.ntt.co.jp

SVM 2008 「夏の終りに」

Aug 31, 2008

NAIST

Overview

- 混合モデルから Product モデルへ
  - Products of Experts (PoE) (Hinton, 2002)
- NIPS 2007: Deep Learning ワークショップ
  - http://www.iro.umontreal.ca/~lisa/twiki/bin/view.cgi/Public/DeepLearningWorkshopNIPS2007
  - 「Deep Belief Net」と次々に言わせるという Banquet ネタ
- Contrastive Divergence 学習
- 自然言語処理との (深い) 関係と展望

Mixture models and Product models

- Mixture Model

$$p(\mathbf{x}|\Theta;\Lambda) = \sum_{m=1}^{M} \lambda_m \, p(\mathbf{x}|\theta_m) \tag{1}$$

  - データは「どれか 1 つ」のモデルから生成される

- Product Model

$$p(\mathbf{x}|\Theta) = \frac{\prod_{m=1}^{M} p(\mathbf{x}|\theta_m)}{\sum_{\mathbf{x}} \prod_{m=1}^{M} p(\mathbf{x}|\theta_m)} \tag{2}$$

  - データは「すべての制約」を満たされて生成
  - 現実のデータには, 多くの場合こちらが適当
  - 自然言語の場合,
    - PCFG, n-gram, ジェンダー, 文脈, リズム, …など
      多くの条件を満たして文が生成されている (cf. 短歌や俳句)
    - 商品を選ぶにも, 「内容」「デザイン」「緊急性」「広告効果」
      etc..

Loglinear and Product of Experts

$$p(\mathbf{x}|\Theta) = \frac{\prod_m p(\mathbf{x}|\theta_m)}{\sum_{\mathbf{x}} \prod_m p(\mathbf{x}|\theta_m)} \iff p(\mathbf{x}|\Lambda) = \frac{\prod_m e^{\lambda_m f_m(\mathbf{x})}}{\sum_{\mathbf{x}} \prod_m e^{\lambda_m f_m(\mathbf{x})}} \tag{3}$$

- Product of Experts (PoE) … 1 つ 1 つのモデルが, パラメータ θ を持つ確率分布
- Loglinear は, 各モデルが (スケールされた) δ 関数となる特別な場合

$$e^{\lambda_m f_m(\mathbf{x})} = \begin{cases} e^{\lambda_m} & \text{if } f_m(\mathbf{x}) = 1 \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

- 学習の目標 (Unsupervised) :
  - 各モデル $p(\cdot|\theta_1) \cdots p(\cdot|\theta_M)$ ができるだけ直交して (独立に) データを表現するように, パラメータ Θ を最適化する
  - 「積数」M の可変長化 (later)

Problem with Estimating PoE

$$p(\mathbf{x}|\Theta) = \frac{\prod_m p(\mathbf{x}|\theta_m)}{\sum_{\mathbf{x}} \prod_m p(\mathbf{x}|\theta_m)} \tag{5}$$

- 分配関数 $Z = \sum_{\mathbf{x}} \prod_m p(\mathbf{x}|\theta_m)$ が容易には求まらない！
  - $\sum_{\mathbf{x}}$ はたとえば, 「可能な文 $\mathbf{x}$ すべてについての厖大な和」
- Loglinear と違い, 一般に凸でない (× L-BFGS)
- 教師なし学習.

Contrastive Divergence learning (Hinton 2000; 2002)

- 一般に,

$$p(\mathbf{x}|\theta) = \frac{1}{Z} f(\mathbf{x}|\theta) \quad ; \; Z = \sum_{\mathbf{x}} f(\mathbf{x}|\theta) \tag{6}$$

  となるモデルを考えよう.

- データ $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ について,

$$L = \left\langle \log p(\mathbf{X}|\theta) \right\rangle_{\hat{p}(\mathbf{x})} \tag{7}$$

$$= \sum_{i=1}^{N} \hat{p}(\mathbf{x}_i) \log p(\mathbf{x}_i|\theta) \tag{8}$$

  を最大化することを考える.

- このとき, $\log p(\mathbf{x}|\theta) = \log f(\mathbf{x}|\theta) - \log Z$ なので,

Contrastive Divergence learning (2)

$$\frac{\partial}{\partial\theta}\log p(\mathbf{x}|\theta) = \frac{\partial}{\partial\theta}\left[\log f(\mathbf{x}|\theta) - \log Z\right] \tag{9}$$

$$= \frac{\partial}{\partial\theta}\log f(\mathbf{x}|\theta) - \frac{1}{Z}\frac{\partial}{\partial\theta}Z \tag{10}$$

ここで

$$\frac{\partial}{\partial\theta}Z = \frac{\partial}{\partial\theta}\sum_{\mathbf{x}} f(\mathbf{x}|\theta) = \sum_{\mathbf{x}}\frac{\partial}{\partial\theta}f(\mathbf{x}|\theta) \tag{11}$$

かつ

$$\frac{\partial}{\partial\theta}\log f(\mathbf{x}|\theta) = \frac{1}{f(\mathbf{x}|\theta)}\frac{\partial}{\partial\theta}f(\mathbf{x}|\theta) \tag{12}$$

$$\therefore \quad \frac{\partial}{\partial\theta}f(\mathbf{x}|\theta) = f(\mathbf{x}|\theta)\frac{\partial}{\partial\theta}\log f(\mathbf{x}|\theta) \quad \text{なので,} \tag{13}$$

Contrastive Divergence learning (3)

$$\frac{\partial L}{\partial \theta} = \left\langle \frac{\partial}{\partial \theta} \log f(\mathbf{x}|\theta) - \sum_{\mathbf{x}} \frac{f(\mathbf{x}|\theta)}{Z} \frac{\partial}{\partial \theta} \log f(\mathbf{x}|\theta) \right\rangle_{\hat{p}(\mathbf{x})} \tag{14}$$

$$= \left\langle \frac{\partial}{\partial \theta} \log f(\mathbf{x}|\theta) - \sum_{\mathbf{x}} p(\mathbf{x}|\theta) \frac{\partial}{\partial \theta} \log f(\mathbf{x}|\theta) \right\rangle_{\hat{p}(\mathbf{x})} \tag{15}$$

$$= \left\langle \frac{\partial}{\partial \theta} \log f(\mathbf{x}|\theta) \right\rangle_{\hat{p}(\mathbf{x})} - \left\langle \frac{\partial}{\partial \theta} \log f(\mathbf{x}|\theta) \right\rangle_{p(\mathbf{x}|\theta)} \tag{16}$$

$$= \left\langle \frac{\partial}{\partial \theta} \log f(\mathbf{x}|\theta) \right\rangle_{p_0} - \left\langle \frac{\partial}{\partial \theta} \log f(\mathbf{x}|\theta) \right\rangle_{p_\infty} \tag{17}$$

- ここで, 経験分布 $\hat{p}(\mathbf{x}) = p_0$, モデル分布 $p(\mathbf{x}|\theta) = p_\infty$ とおいた.
  - モデル分布は, MCMC を ∞ 回動かしてサンプルした分布と同じ

Contrastive Divergence learning (4)

$$\frac{\partial L}{\partial \theta} = \left\langle \frac{\partial}{\partial \theta} \log f(\mathbf{x}|\theta) \right\rangle_{p_0} - \left\langle \frac{\partial}{\partial \theta} \log f(\mathbf{x}|\theta) \right\rangle_{p_\infty} \tag{18}$$

この差を計算して θ を最適化する代わりに,

$$\left\langle \frac{\partial}{\partial \theta} \log f(\mathbf{x}|\theta) \right\rangle_{p_0} - \left\langle \frac{\partial}{\partial \theta} \log f(\mathbf{x}|\theta) \right\rangle_{p_1} \tag{19}$$

で微分を計算することにする

- p_1 は, データ $\mathbf{x}$ から MCMC 1 回分の reconstruction (confabulation)
- こうしても, 結果はほとんど変わらない

Contrastive Divergence learning (5)

ここで,

$$D(p_n||p_\infty) = \sum_{\mathbf{x}} p_n(\mathbf{x}) \log p_n(\mathbf{x}) - \sum_{\mathbf{x}} p_n(\mathbf{x}) \log p_\infty(\mathbf{x}) \tag{20}$$

$$= -H(p_n) - \langle \log p_\infty \rangle_{p_n} \propto -\langle \log p_\infty \rangle_{p_n} \tag{21}$$

に注意すると,

$$\begin{aligned} -\frac{\partial}{\partial \theta_m}\left(p_0||p_\infty - p_1||p_\infty\right) &= \left\langle \frac{\partial}{\partial \theta_m} \log p_\infty \right\rangle_{p_0} - \left\langle \frac{\partial}{\partial \theta_m} \log p_\infty \right\rangle_{p_1} \\ &= \left\langle \frac{\partial}{\partial \theta_m} \log p(\mathbf{x}|\theta_m) \right\rangle_{p_0} - \left\langle \frac{\partial}{\partial \theta} \log p(\mathbf{x}|\theta) \right\rangle_{p_\infty} \\ &\quad - \left\langle \frac{\partial}{\partial \theta_m} \log p(\mathbf{x}|\theta_m) \right\rangle_{p_1} + \left\langle \frac{\partial}{\partial \theta} \log p(\mathbf{x}|\theta) \right\rangle_{p_\infty} \end{aligned} \tag{22}$$

Contrastive Divergence learning (6)

$$= \left\langle \frac{\partial}{\partial \theta_m} \log p(\mathbf{x}|\theta_m) \right\rangle_{p_0} - \left\langle \frac{\partial}{\partial \theta_m} \log p(\mathbf{x}|\theta_m) \right\rangle_{p_1} . \tag{23}$$

よって,

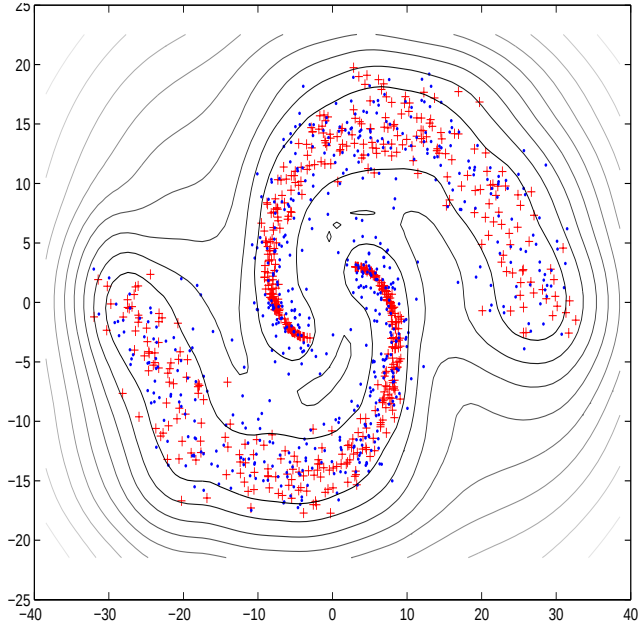
$$\frac{\partial D}{\partial \theta_m} = \left\langle \frac{\partial}{\partial \theta_m} \log p(\mathbf{x}|\theta_m) \right\rangle_{p_0} - \left\langle \frac{\partial}{\partial \theta_m} \log p(\mathbf{x}|\theta_m) \right\rangle_{p_1} \tag{24}$$

を MCMC 1 ステップから計算して, θ_m を更新すればよい. □

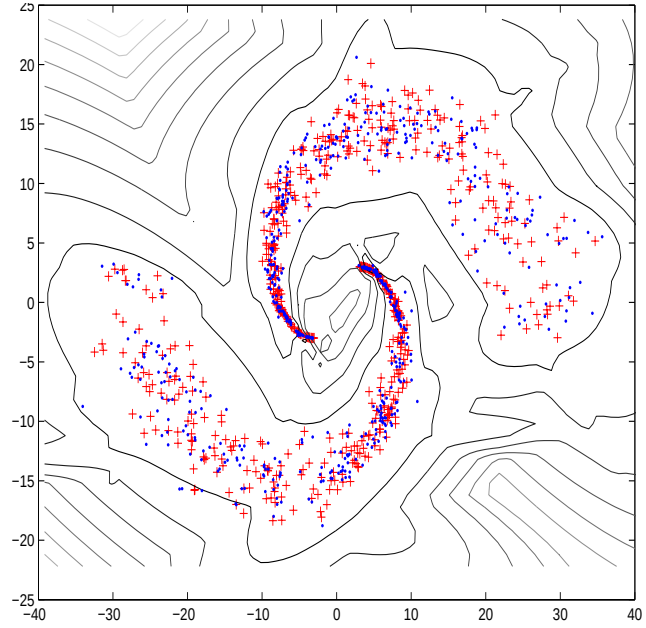
Contrastive Divergence learning (7): 図解

- “Self supervised boosting” (Welling, Zemel, Hinton 2001; SVM 勉強会, 2002) からの図

(a)



(c)



- 1 次元の場合は, ホワイトボードで説明

Relationship to other works (1/2)

- “Contrastive Estimation” (Smith and Eisner, ACL 2005)
  … CD とほとんど同じ (本人は浅い理解で違うと主張している)
  - CE は

$$\frac{\partial L}{\partial \lambda_j} = \sum_i \langle f_j | \mathbf{x}_i \rangle_\lambda - \langle f_j | \mathcal{N}(\mathbf{x}_i) \rangle_\lambda \tag{25}$$

    を解いて λ_j を最適化
  - これは

$$\prod_i p(\mathbf{x}_i | \mathcal{N}(\mathbf{x}_i), \Lambda) \tag{26}$$

    を最大化していることに相当
  - $\mathcal{N}$ は “Neighborhood” 関数で,
    - 1 語削除/前後 2 単語入れ替え/部分単語列削除/任意の単語置換
    - などを考える

Relationship to other works (2/2)

- “A Discriminative Language Model with Pseudo-Negative Examples” (Okanohara, ACL 2007)
  … CD と基本的に同じ
  - “擬似負例” をモデルから生成して, 全文最大エントロピー法のパラメータを学習
  - 学習には, オンラインマージン最大化法を使う (ここが違う)
  - クラス bigram を使った拡張
  - 全文 ME には, 誰も適用していなかった

Text modeling through Products of Experts

- Rate Adapting Poisson (RAP) Model
  (Gehler, Holub, Welling: ICML 2006)
  http://www.kyb.tuebingen.mpg.de/bs/people/pgehler/rap/

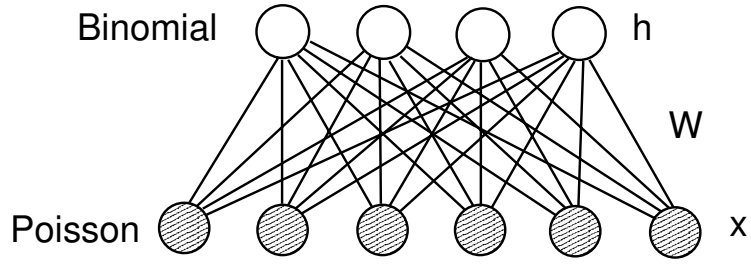


Figure 1. Markov random field representation of the RAP model.

$$p(\mathbf{x},\mathbf{h}) \propto \prod_i \lambda_i \frac{e^{x_i}}{x_i!} \prod_j \frac{p_j}{1-p_j} \frac{e^{h_j}}{h_j!(M_j - h_j)!} \cdot \underbrace{\prod_i \prod_j w_{ij} x_i h_j}_{\text{Energy}} \qquad (27)$$

- 単語観測ベクトル: $\mathbf{x}$, 隠れトピック層: $\mathbf{h}$
- Restricted Boltzmann Machine というニューラルネットの一種
- “Prior” というものはない
  - 正規化定数 Z は計算不可能 (に近い)

RAP (2): Conditional distribution

- $\mathbf{x}$ と $\mathbf{h}$ の条件つき分布は Poisson,Binomial になる

$$p(\mathbf{x}|\mathbf{h}) = \prod_i \mathrm{Po}\big(\lambda_i \exp(\sum_j w_{ij} h_j)\big) \tag{28}$$

$$p(\mathbf{h}|\mathbf{x}) = \prod_j \mathrm{Bin}\big(\sigma(\beta_j + \sum_i w_{ij} x_i), M_j\big) \tag{29}$$

  - $\sigma(x) = 1/(1+\exp(-x))$：シグモイド関数
  - $\beta_j = \log p_j/(1-p_j)$ とおいた
- 観測値 $\mathbf{x}_n$ から隠れ層 $\mathbf{h}_n$ がサンプルでき, $\mathbf{h}_n$ から “Reconstruction” $\tilde{\mathbf{x}}_n$ がサンプルできる

RAP (3): Marginal distribution

- 潜在トピック層 $\mathbf{h}$ を周辺化して消去すると,

$$p(\mathbf{x}) \propto \prod_i \lambda_i \frac{e^{x_i}}{x_i!} \cdot \underbrace{\prod_j e^{M_j} (1 + \exp(\underbrace{\sum_i w_{ij} x_i - \beta_j}_{\text{トピック } j \text{ に関する } \mathbf{x} \text{ の “activation”}}))}_{\text{トピック } j \text{ の確率密度} \geq 1} \tag{30}$$

  - $\mathbf{x}$ の確率 $\simeq$ Poisson 出現 $\times$ ニューラルネットの Activation の積
    - $\mathbf{x}$ が「スポーツ」トピックで適当かつ,「政治」の activation が 0 でもよい ($\because e^0 = 1$)

RAP (4): 学習

- 正規化定数 Z はわからないので, Contrastive Divergence で学習 ($M_j = 1$ とした)

$$\begin{cases} \delta \log \lambda_i \propto \langle x_i \rangle_{\hat{p}} - \langle x_i \rangle_{\tilde{p}} \\ \delta \beta_j \propto -\langle \sigma(\mathbf{w}_j \cdot \mathbf{x} - \beta_j) \rangle_{\hat{p}} - \langle \sigma(\mathbf{w}_j \cdot \mathbf{x} - \beta_j) \rangle_{\tilde{p}} \\ \delta w_{ij} \propto \langle x_i \sigma(\mathbf{w}_j \cdot \mathbf{x} - \beta_j) \rangle_{\hat{p}} - \langle x_i \sigma(\mathbf{w}_j \cdot \mathbf{x} - \beta_j) \rangle_{\tilde{p}} \end{cases} \tag{31}$$

- "Reconstruction" $\tilde{p}$ は $\mathbf{x} \to \mathbf{h} \to \tilde{\mathbf{x}}$ で作った, モデルからの擬似データ

RAP (5): 実験

- 20 Newsgroup :

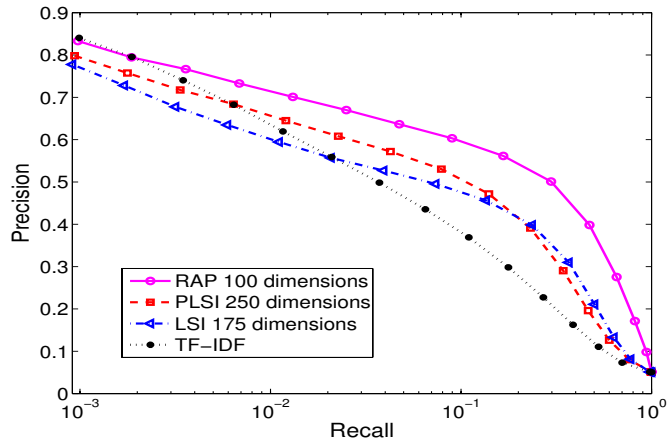
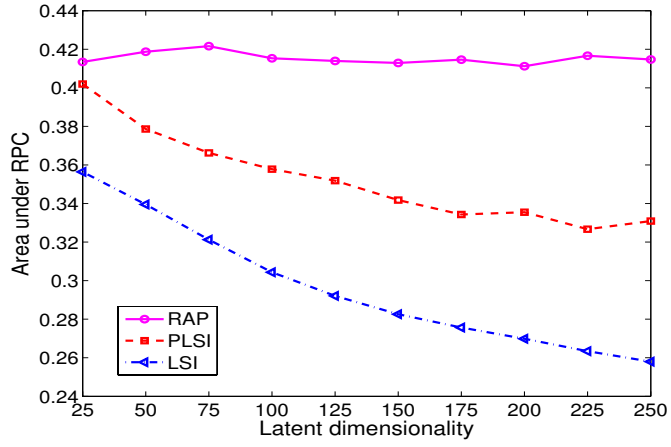


Figure 4. RPC plot on a log-scale of the 20 Newsgroups dataset for various models. As a baseline the retrieval results with tf-idf reweighed word-counts are shown. Number of topics for each model was chosen by optimizing 1-NN classification performance on the test set corresponding to the average precision for retrieving a single document (left most marker).

- Reuters-21578 :



RAP (6): 長所

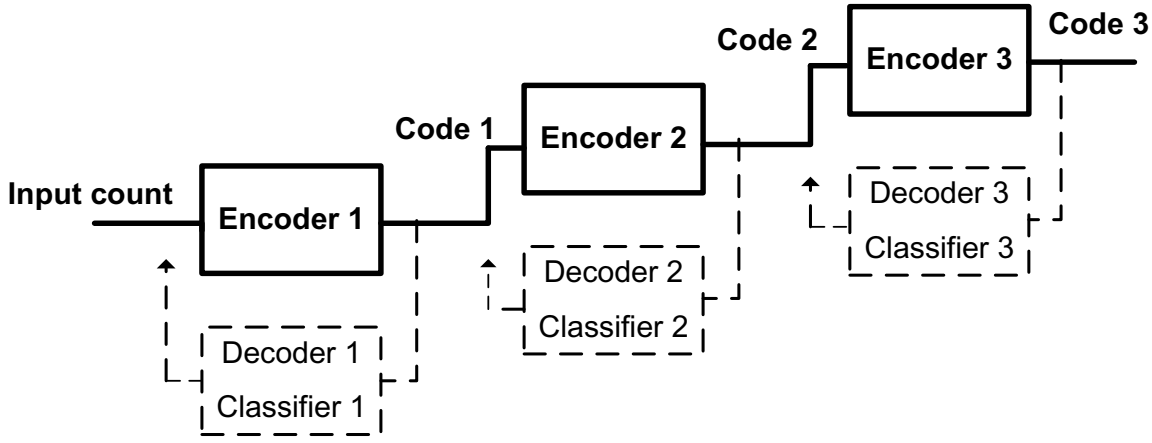
- RAP の長所: 文書 $\mathbf{x}$ が与えられると, “Latent representation” は

$$\boldsymbol{\beta} + W\mathbf{x} \tag{32}$$

  として簡単に求まる (行列・ベクトルの積は超高速)
  - LDA や HDP では複雑な最適化が必要
  - Gaussian 表現されている →カーネルマシン等への応用; 後で
  - 正確な統計的意味づけがある
    - 単なるニューラルネットではない
- RBM は, ニューラルネットの世界では “Harmonium” として知られていたらしい (Smolensky 1986)
  - ただし, Hinton(2002) のような効率的な学習法はなかった
- 指数分布族に一般化可能
  - “Exponential Family Harmoniums” (Welling, Rosen-Zvi, Hinton: NIPS 2004)

From Harmoniums to Deep Belief Nets

- Deep Belief Nets (Hinton+ 2006, *Neural Computation*)
  - RBM を階層化
  - “Encoder” と “Decoder” の組み合わせで, ある隠れ層の結果を次の層の入力として順番に CD 学習
  - Variational bound を順に最大化している (らしい)
- “Semi-supervised Learning of Compact Document Representations with Deep Networks” (Ranzato, Szummer (Microsoft), ICML 2008)



Deep Document Representation (Ranzato&Szummer 2008)

- 第 1 層は, Poisson(Decoder)+Gaussian(Encoder) ⋯ RAP と同じ
- 第 2 層以降では, Gaussian(Decoder)+Gaussian(Encoder)

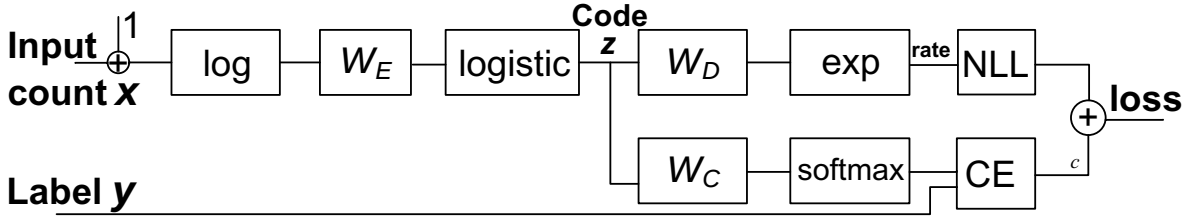


Figure 2. The architecture of the first stage has three components: (1) an encoder, (2) a decoder (Poisson regressor), and (3) a classifier. The loss is the weighted sum of cross-entropy (CE) and negative log-likelihood (NLL) under the Poisson model.

- Encoder: $\mathbf{z} \sim \sigma(W \log(\mathbf{x}+1) + b_E)$: 全層
- Decoder: $\mathbf{x} \sim \mathrm{Po}(\beta \exp(W_D \mathbf{z} + b_D))$: 第 1 層
  $\mathbf{x} \sim \mathrm{N}(W_D \mathbf{z} + b_D, \sigma)$: 第 2 層以降

Semi-supervised Learning

- 文書ラベルを考慮した Semi-Supervised Learning

$$L = E_R + \alpha E_C \tag{33}$$

  を最小化. ($\alpha = 0$ で生成モデル)

- E_R : Reconstruction error

$$E_R = \sum_i \left[\beta \exp(W_D \mathbf{z} + b_{Di}) - (x_i W_D \mathbf{z} + x_i b_{Di} - \log(x_i!)) \right] \tag{34}$$

- E_C : Classification error
  - 隠れ層 $\mathbf{z}$ からラベルを予測する

$$\begin{cases} \hat{y}_i &= \frac{\exp(W_{Ci}\mathbf{z} + b_{Ci})}{\sum_i \exp(W_{Ci}\mathbf{z} + b_{Ci})} \\ E_C &= -\sum_{i=1}^{K} y_i \log \hat{y}_i \end{cases} \tag{35}$$

分類問題への適用

- “Using Deep Belief Nets to Learn Covariance kernels for Gaussian Processes”, Salakhutdinov and Hinton, NIPS 2007
- Gaussian Process: ベイズカーネルマシン
  - 線形回帰の重みベクトル $\mathbf{w}$ を積分消去
  - データ: $D = \{(y_i, \mathbf{x}_i)\}$ $(i = 1 \ldots N)$
  - ガウス分布で予測

$$p(y|\mathbf{x}, D, \sigma^2) = \mathrm{N}(\mathbf{k}^T(\mathbf{K} + \sigma^2 I)^{-1}\mathbf{y}, \Sigma) \tag{36}$$

    - $\mathbf{k}, \mathbf{K}$: カーネル行列
    - $\mathbf{k} = K(\mathbf{x}, \mathbf{X})$
    - $\mathbf{K} = K(\mathbf{X}, \mathbf{X})$
  - y が離散の場合は, 予測にシグモイドをかませる

分類問題への適用 (2)

- 通常のカーネルマシンの問題点:
  - 確率モデルでない (ex. SVM)
  - 要素 (ex. 単語) 間のカーネルが ID カーネルだったりする
    - Tree kernel 等も最後は単語の比較に帰着
- Deep Belief Nets の隠れ層 (Gaussian) を Gaussian Process の入力とする

$$K_{ij} = \alpha \exp\left(-\frac{1}{2\beta} \| \mathbf{h}_i|\mathbf{x}_i - \mathbf{h}_j|\mathbf{x}_j \|^2\right) \tag{37}$$

  - $\mathbf{h}|\mathbf{x}, W$: DBN による入力 $\mathbf{x}$ の map
  - α, β は GP 分類器を作った後, 予測に関して Optimize する

Future Agenda

- 隠れ層の数/素子数の自動決定?
- 時系列モデルへの拡張?
  - 言語モデルとしては一応 (tRBM; Mnih, Hinton 2007) があるが‥
- Structured Output の場合?

  ⇓
- GP 等のカーネルマシンとの連繋が興味深い

最後に

- 京阪奈に来て, 今年でちょうど 10 年
  - 1998 年 NAIST M1 入学
  - 色々ありました.. (ATR etc.)
- 当時興味を持っていたことは, 未だ光を失っていない
  - 統計的な洗練 (でもまだ充分ではない)
  - 現役の人は, 自分の問題意識を大事にしてほしい
- そろそろ京阪奈を出たい?

Fin

- ご静聴ありがとうございました。