Depth Qualification Exam Presentation

Li Wan, Dept. of Computer Science,
Courant Institute,
New York University
Overview of Talk

1. Literature Survey
   (a) Approaches to train deep neural network
   (b) Topic models and its application to computer vision

2. Research Result: How to effectively combine neural network model with topic model.
Non-linear activation function: \( h_i = f(w^T_i x + b) \) where function \( f \) could be sigmoid function \( \sigma(x) = 1/(1 + \exp(-x)) \) or \( tanh \) function (normalized to \((0, 1)\) or \((-1, 1)\) scale) neural network [1].

[1] Bishop, Neural Network for Pattern Recognition, 1995
Deep network is better than shallow. However, standard random initialization leads poor training and generalization error in deep neural networks except deep CovNets.

how to Pre-train Deep Neural Networks: greedy layer-wise pre-training[1](\(x\): input data, \(h\) hidden layer, and \(w\) parameters)

Generative model[2][3](Restricted Boltzmann Machines):

\[
w = \arg \max_w \sum_h p(x, h; w)
\]

Encoder(\(f\))-decoder(\(g\)) model[4][5][6][7]:

\[
w = \arg \min_w \|h - f(x; w)\| + \|x - g(h; w)\| + \lambda \|w\|_1
\]

Restricted Boltzmann Machines

Undirected graphical model (Bipartite graph $U = \{x\}$ and $V = \{h\}$) with energy function (binary case):

$$E(x, h) = -b^T x - c^T h - h^T W x$$

Fast inference: $p(h|x; w) = \prod_j p(h_j|x; w) = \prod_j \sigma(W_j x + b)$

Fast sampling: $p(x|h; w) = \prod_i p(x_i|h; w) = \prod_i \sigma(W_i^T h + c)$.

Neural network feed forward operation: $f(x; w) = p(h|x; w)$

Initialize $W$ in neural network via maximize $p(x; w) = \sum_h p(x, h; w)$ as follows:

$$\Delta W = E_{data}[hx^T] - E_{model}[hx^T]$$

However, $E_{model}[hx^T]$ is intractable[1] because number of possible $h$ is exponential to its size. Contrast Divergence[2][3] and its extensions[4] proposed to approximate model expectation with a few samples.

[1] Long et al. Restricted Boltzmann Machines are Hard to Approximately Evaluate or Simulate
Encoder-Decoder Model

- Encoding operation should preserve essential information of data $x$. Verify it by reconstruct $x$ with decoder($g$) based on $h = f(x; w)$.

- Minimize encoding error $\|h - f(x; w)\|$ and decoding error $\|x - g(h; w)\|$ with proper penalty on $\lambda\|w\|_1$ to encourage local filters. $t(h)$ is penalty term of code $h$ to encourage special property such as sparestness [3].

$$L(w, h) = \|h - f(x; w)\| + \|x - g(h; w)\| + \lambda\|w\|_1 + \alpha t(h)$$

- Neural network feed forward operation is an encoding operation

- Learning $W$ by repeat the following steps[1][2][3]: (with random initialize $w$)
  - $h_0 \leftarrow f(x; w)$
  - $h_t \leftarrow h_{t-1} + \eta \frac{\partial L(w, h_{t-1})}{\partial h_{t-1}}$ a few steps with initial condition at $h_0$
  - $w \leftarrow w + \eta \frac{\partial L(w, h_t)}{\partial w}$

Applications of Deep Neural Network

- Natural Image patches modeling [1][8]
- Image classification [2][5]
- Text Modeling [3]
- Human Pose Tracking [4]
- Digit Recognition [6][7]

Given a data set $x$ with label $y$, we are interested in the following probabilistic regression model:

$$ y = f(x) + \epsilon \text{ with } f(x) \sim N(0, K) \text{ and } \epsilon \sim N(0, \sigma^2) $$

Here $K_{ij} = \alpha \exp(-\beta(x_i - x_j)^T(x_i - x_j))$ is covariance function. Loss function $-\log p(y|x)$ could be defined by integrate $f(x)$ as follows:

$$ L = - \log p(y|x) = -\frac{1}{2} \log \|K + \sigma^2 I\| - \frac{1}{2} y^T (K + \sigma^2 I)^{-1} y + C $$

1. Gradient $\partial L/\partial x$ could be written down from definition

2. If $x$ is response of neural network with input $v$, $\partial x/\partial v$ could be defined.

3. Back-propagation of the joint model is defined based on $\partial L/\partial x$ and $\partial x/\partial v$ [1].

Latent Semantic Analysis [1]: map document to latent semantic space of reduced dimensionality.

Given co-occurrence table $X$ where each row is histogram of words

Apply SVD to $X$: $X = U\Sigma V^T$.

Approximate $X$ by a few top singular values in $\Sigma$:

$\tilde{X} = U\tilde{\Sigma}V^T \approx U\Sigma V^T = X$

co-occurrence table in latent space: $U\tilde{\Sigma}$ because inner product space is: $XX^T \approx U\tilde{\Sigma}^2U^T$.

Topic Models: pLSA

Probabilistic Latent Semantic Analysis [1]

- Joint distribution \( p(d, w) = p(d) \sum_z p(w|z)p(z|d) = \sum_z p(z)p(d|z)p(w|z) \)
- Relationship with LSA: \( U_{ik} = p(d_i|z_k), V_{jk} = P(w_j|z_k) \) and \( \Sigma_{kk} = p(z_k) \)
- Learn with EM by alternate update \( p(z|w, d) \) and \( p(w|z), p(d|z), p(z) \).


Graphical model representation of the aspect model in the asymmetric (a) and symmetric (b) parameterization.
Each document is a random mixture of corpus-wide topics

Each word is drawn from one of those topics
Latent Dirichlet Allocation [1]

- Fully generative model: extension of pLSA
- Joint distribution: \( p(w|\alpha, \beta) = \int p(\theta|\alpha) \left( \prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta \)
- Learn with variational EM algorithm

Object Recognition in Computer Vision

Image classification: how it works

Input: dense sampling

Image feature: SIFT

Dictionary learning

K-means

rescoring: Combining detection results

Binary classification: SVM

Pyramid pooling histogram of words
Extract Image Features

- Extract features from image patches (SIFT [1], HOG [2], etc.)
- Learn dictionary from visual features ($K$-means, sparse coding [3], etc.)
- Represent images by combining features (histogram, global/local pooling [3][4])

[1] Lowe, Distinctive image features from scale-invariant keypoints, IJCV, 2004
Model Image Features

- Discriminative model: SVM with linear/hist-intersection/$\chi^2$ kernel [1]

- Generative model: Hierarchical Bayesian model could be applied, such as extension of naïve Bayesian model [2], pLSA model [3][4], LDA model [5].

Bayesian Model for Object Recognition

An extension of Bayesian topic model by including location information[1]-[4]

<table>
<thead>
<tr>
<th>symbol</th>
<th>description</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_{ji}$</td>
<td>$K$-means index (patch appearance)</td>
<td>$w_{ji} \sim \text{Multi}(\eta_z)$</td>
</tr>
<tr>
<td>$v_{ji}$</td>
<td>object part (patch location)</td>
<td>$v_{ji} \sim N(\mu_k, \Lambda_k)$</td>
</tr>
<tr>
<td>$z_{ji}$</td>
<td>topic index</td>
<td>$z_{ji} \sim \text{Multi}(\pi_o)$</td>
</tr>
<tr>
<td>$\rho_j$</td>
<td>object center location</td>
<td>$\rho_j \sim N(\gamma, \varsigma)$</td>
</tr>
<tr>
<td>$o_j$</td>
<td>image label</td>
<td></td>
</tr>
</tbody>
</table>

Bayesian Model for Object Recognition

- Each object is a mixture of topics
- Each appearance and location pair are draw from one of those topics
My Research Result

Combine neural network model with topic model

Neural network: nonlinear transformation
My Research Result

Combine neural network model with topic model

- Neural network: nonlinear transformation
- Bayesian Topic Model: transparent to human
My Research Result

Combine neural network model with topic model

- Neural network: nonlinear transformation
- Bayesian Topic Model: transparent to human
- Replace regression component of neural network with Bayesian model (topic model)
My Research Result

Combine neural network model with topic model

- Neural network: nonlinear transformation
- Bayesian Topic Model: transparent to human
- Replace regression component of neural network with Bayesian model (topic model)
- Bayesian model with input from the response of neural network
What we want to learn

Given the input data $v$ with label $y$, $x = f_w(v)$ is output of neural network given input $v$. The likelihood function is given by:

$$p_v(v|y) = p_x(f_w(v)|y) | \det \frac{\partial(f_w)}{\partial(v)}|$$

- $p_x(f_w(v)|y)$ defined by generative model
- $\frac{\partial(f_w)}{\partial(v)}$ is the Jacobian matrix
What we want to learn

Given the input data $v$ with label $y$, $x = f_w(v)$ is output of neural network given input $v$. The likelihood function is given by:

$$p_v(v|y) = p_x(f_w(v)|y) \cdot \det \frac{\partial (f_w)}{\partial (v)}$$

- $p_x(f_w(v)|y)$ defined by generative model
- $\frac{\partial (f_w)}{\partial (v)}$ is the Jacobian matrix

Applying Bayesian rule, we have the loss function:

$$p(y|v) = \frac{p_v(v|y)}{\sum \tilde{y} p_v(v|\tilde{y})} = \frac{p_x(f_w(v)|y)}{\sum \tilde{y} p_x(f_w(v)|\tilde{y})} = p_x(y|f_w(v))$$
Model overview
1. We first initialize the parameters \( \{ w^0, \pi^0, \eta^0, \phi^0 \} \) by pre-training of neural network and graphical model.
1. We first initialize the parameters \( \{w^0, \pi^0, \eta^0, \phi^0\} \) by pre-training of neural network and graphical model. 
2. Jointly updated according to the gradient descent: Convert generative model into extra layers of neural network (assume there is a closed form inference in top graphical model).
Generative Model

1. Draw latent topic $z_j \sim Multi(\pi_{y_i})$

2. Draw latent word $u_j \sim Multi(\eta_{z_i})$

3. Draw feature vector $x_j \sim Gaussian(\phi_{u_j})$. 
Joint Optimization

Overall loss function:

\[ L = - \sum_j \log p(f_w(v_j)|y, \pi, \eta, \phi) + \log \sum_{i=1}^S \prod_j p(f_w(v_j)|y = i, \pi, \eta, \phi) \]
Joint Optimization

Overall loss function:

\[ L = - \sum_j \log p(f_w(v_j)|y, \pi, \eta, \phi) + \log \sum_{i=1}^{S} \prod_j p(f_w(v_j)|y = i, \pi, \eta, \phi) \]

Generative model likelihood function:

\[ p(f_w(v_j)|y, \pi, \eta, \phi) = \sum_{z_j=1}^{M} \left( \sum_{u_j=1}^{K} p(f_w(v_j)|u_i, \phi) p(u_j|z_j, \eta) \right) p(z_j|y, \pi) \]
Joint Optimization

Overall loss function:

\[ L = - \sum_j \log p(f_w(v_j)|y, \pi, \eta, \phi) + \log \sum_i \prod_j p(f_w(v_j)|y = i, \pi, \eta, \phi) \]

Generative model likelihood function:

\[ p(f_w(v_j)|y, \pi, \eta, \phi) = \sum_{z_j=1}^M \left( \sum_{u_j=1}^K p(f_w(v_j)|u_i, \phi) p(u_j|z_j, \eta) \right) p(z_j|y, \pi) \]

Trick: decompose likelihood function into small piece
Unified model

Gaussian Likelihood Layer ($F_2 : f_w(v_j) \rightarrow p(f_w(v_j)|u_j, \phi)$):

$$p(f_w(v_j)|y, \pi, \eta, \phi) = \sum_{z_j=1}^{M} \left( \sum_{u_j=1}^{K} p(f_w(v_j)|u_i, \phi) p(u_j|z_j, \eta) \right) p(z_j|y, \pi)$$
Unified model

- Gaussian Likelihood Layer ($F_2 : f_w(v_j) \rightarrow p(f_w(v_j)|u_j, \phi)$):

- Integration Layer on $u(F_1(., \eta))$:

\[
p(f_w(v_j)|y, \pi, \eta, \phi) = \sum_{z_j=1}^{M} \left( \sum_{u_j=1}^{K} p(f_w(v_j)|u_i, \phi) p(u_j|z_j, \eta) \right) p(z_j|y, \pi)
\]

\[
F_1(., \eta)
\]
Unified model

- Gaussian Likelihood Layer ($F_2 : f_w(v_j) \rightarrow p(f_w(v_j)|u_j, \phi)$):
- Integration Layer on $u(F_1(., \eta))$:
- Integration Layer on $z(F_1(., \pi))$:

\[
p(f_w(v_j)|y, \pi, \eta, \phi) = \sum_{z_j=1}^{M} \left( \sum_{u_j=1}^{K} p(f_w(v_j)|u_i, \phi) p(u_j|z_j, \eta) \right) p(z_j|y, \pi)
\]

$F_1(., \pi)$
Unified model

- Gaussian Likelihood Layer($F_2 : f_w(v_j) \rightarrow p(f_w(v_j)|u_j, \phi)$):

- Integration Layer on $u(F_1(., \eta))$:

- Integration Layer on $z(F_1(., \pi))$:

- Bayesian Layer($F_0 : p(f_w(v_j)|y) \rightarrow p(y|f_w(v_j)))$:

\[
L = - \sum_{j} \log p(f_w(v_j)|y, \pi, \eta, \phi) + \log \sum_{i=1}^{S} \prod_{j} p(f_w(v_j)|y = i, \pi, \eta, \phi)
\]
2D data with 5 latent cluster drawn from 4 classes

- shape: class label (cross, dot, square, circle)
- color: model prediction
- visualization of input after neural network transformation
## Scene classification result

<table>
<thead>
<tr>
<th>Method</th>
<th>Classification Rate ± Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>pLSA SVM</td>
<td>63.3 ± 1.1</td>
</tr>
<tr>
<td>LDA</td>
<td>65.2</td>
</tr>
<tr>
<td>Neural network</td>
<td>51.6 ± 1.1</td>
</tr>
<tr>
<td>HTM</td>
<td>64.9 ± 1.2</td>
</tr>
<tr>
<td>HTM SVM</td>
<td>65.5 ± 1.5</td>
</tr>
<tr>
<td>Hybrid model pre-trained</td>
<td>65.7 ± 0.4</td>
</tr>
<tr>
<td>Hybrid model fully trained</td>
<td>70.1 ± 0.6</td>
</tr>
</tbody>
</table>

### Table 1: Classification rates of different methods on scene classification dataset