When Bayes meets Deep Learning
towards robust and interpretable DL

Jun Zhu
dcszj@mail.Tsinghua.edu.cn
Department of Computer Science and Technology
Tsinghua University
Deep Learning

“Big Model + Big Data + Big/Super Cluster”

9 layers sparse autoencoder with:
- local receptive fields to scale up;
- local L2 pooling and local contrast normalization for invariant features
- 1B parameters (connections)
- 10M 200x200 images
- train with 1K machines (16K cores) for 3 days

-able to build high-level concepts, e.g., cat faces and human bodies
-15.8% accuracy in recognizing 22K objects (70% relative improvements)
Overfitting in Big Data

 Relevant information grows slower than linear (Bialek et al., 2001)
Some Counter-intuitive Properties of DL

- Stability w.r.t small perturbations to inputs
  - Imperceptible non-random perturbation can arbitrarily change the prediction (adversarial examples exist!)

[Szegedy et al., Intriguing properties of neural nets, 2013]
Overfitting in Big Data

- Surprisingly, regularization to prevent overfitting is *increasingly important*, rather than increasingly irrelevant!

- Increasing research attention, e.g., dropout training (Hinton, 2012)

- More theoretical understanding and extensions
  - Dropout as a Bayesian approximation (Gal & Ghahramani, 2016)
  - MCF (van der Maaten et al., 2013); Logistic-loss (Wager et al., 2013);
  - Dropout SVM (Chen, et al., 2014; Zhuo et al., 2015)
Bayesian methods in DL

- Bayesian neural networks (MacKay, 1992; Neal, 1992)
- Gaussian process in machine learning (MacKay, 1998; Williams, 1999; Rasmussen & Williams, 2006)
- Nonparametric Bayesian deep networks (Adams et al., 2010)
Bayesian methods in DL

- Distilling the knowledge (Hinto et al., 2015) & Bayesian dark knowledge (Korattikara et al., 2015)
Bayesian methods in DL

- Neural variational Bayes (Kingma & Welling, 2014; Rezende et al., 2014)
- Learning to generate with memory (Li et al., 2016)
- RegBayes for deep generative models (Li et al., 2015)
- Doubly stochastic gradient MCMC (Du et al., 2015)
Learning to Generate with Memory
(Li, Zhu, Zhang, ICML 2016)
Probabilistic Generative Models

Assumption: data is described by some factors, which are often hidden

\[
\begin{align*}
z & \sim p(z|\alpha) \\
x & \sim p(x|z, \beta)
\end{align*}
\]

Inference with top-down & bottom-up cues: infer the posterior distribution

\[
p(z|x) \propto p(z|\alpha)p(x|z, \beta)
\]

Learning: estimate the parameters

\[
\hat{\theta} = \text{argmax } p(D|\theta)
\]

Bayesian inference: infer the posterior distribution of parameters

\[
p(\theta|D) \propto p_0(\theta)p(D|\theta)
\]
Deep Generative Models

- Multi-layer *latent-feature* representations with nonlinear transformations

\[ x \sim p(x|z, \beta) \quad z \sim p(z|\alpha) \]

- Many variants by combining different building blocks

A horse sitting on the grass
Symmetric Q-P Network

- **P-network** with two deterministic layers

\[ p(z|x) \propto p(z|\alpha)p(x|z, \beta) \]
Symmetric Q-P Network

- **Q-network** approximates the posterior (Kingma & Welling, 2014; Rezende et al. 2014)

\[
q(z|x) \approx p(z|x)
\]
Symmetric Q-P Network

Problem: detail information is lost during abstraction
Symmetric Q-P Network

- **Ideal case**: get the lost information back!
A Layer with Memory and Attention

[Li, Zhu & Zhang. Learning to Generate with Memory, ICML 2016]
A Stacked Deep Model with Memory

- Asymmetric architecture
Some Results

Density estimation

<table>
<thead>
<tr>
<th>MODELS</th>
<th>MNIST</th>
<th>OCR-LETTERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE</td>
<td>-85.69</td>
<td>-30.09</td>
</tr>
<tr>
<td>MEM-VAE (ours)</td>
<td>-84.41</td>
<td>-29.09</td>
</tr>
<tr>
<td>IWAE-5</td>
<td>-84.43</td>
<td>-28.69</td>
</tr>
<tr>
<td>MEM-IWAE-5 (ours)</td>
<td>-83.26</td>
<td>-27.65</td>
</tr>
<tr>
<td>IWAE-50</td>
<td>-83.58</td>
<td>-27.60</td>
</tr>
<tr>
<td>MEM-IWAE-50 (ours)</td>
<td>-82.84</td>
<td>-26.90</td>
</tr>
</tbody>
</table>
Some Results

Density estimation

<table>
<thead>
<tr>
<th>Models</th>
<th>MNIST</th>
<th>OCR-LETTERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE</td>
<td>-85.69</td>
<td>-30.09</td>
</tr>
<tr>
<td>MEM-VAE (ours)</td>
<td>-84.41</td>
<td>-29.09</td>
</tr>
<tr>
<td>IWAE-5</td>
<td>-84.43</td>
<td>-28.69</td>
</tr>
<tr>
<td>MEM-IWAE-5 (ours)</td>
<td>-83.26</td>
<td>-27.65</td>
</tr>
<tr>
<td>IWAE-50</td>
<td>-83.58</td>
<td>-27.60</td>
</tr>
<tr>
<td>MEM-IWAE-50 (ours)</td>
<td>-82.84</td>
<td>-26.90</td>
</tr>
<tr>
<td>DBN</td>
<td>-84.55</td>
<td>-</td>
</tr>
<tr>
<td>S2-IWAE-50</td>
<td>-82.90</td>
<td>-</td>
</tr>
<tr>
<td>RWS-SBN/SBN*</td>
<td>-85.48</td>
<td>-29.99</td>
</tr>
<tr>
<td>RWS-NADE/NADE*</td>
<td>-85.23</td>
<td>-26.43</td>
</tr>
<tr>
<td>NADE*</td>
<td>-88.86</td>
<td>-27.22</td>
</tr>
<tr>
<td>DARN*</td>
<td>-84.13</td>
<td>-28.17</td>
</tr>
</tbody>
</table>
Missing Value Imputation

(a) Data
(b) Noisy data

(c) Results of VAE
(d) Results of MEM-VAE
Learnt Memory Slots

Average preference over classes of the first 3 slots:

<table>
<thead>
<tr>
<th></th>
<th>0”</th>
<th>1”</th>
<th>2”</th>
<th>3”</th>
<th>4”</th>
<th>5”</th>
<th>6”</th>
<th>7”</th>
<th>8”</th>
<th>9”</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.27</td>
<td>0.82</td>
<td>0.33</td>
<td>0.11</td>
<td>0.34</td>
<td>0.15</td>
<td>0.49</td>
<td>0.27</td>
<td>0.09</td>
<td>0.28</td>
<td></td>
</tr>
<tr>
<td>0.24</td>
<td>0.09</td>
<td>0.06</td>
<td>0.11</td>
<td>0.30</td>
<td>0.13</td>
<td>0.12</td>
<td>0.27</td>
<td>0.09</td>
<td>0.21</td>
<td></td>
</tr>
<tr>
<td>0.18</td>
<td>0.05</td>
<td>0.06</td>
<td>0.11</td>
<td>0.07</td>
<td>0.07</td>
<td>0.05</td>
<td>0.11</td>
<td>0.09</td>
<td>0.18</td>
<td></td>
</tr>
</tbody>
</table>

Corresponding images:
Max-margin Deep Generative Models
(Li, Zhu, Zhang, NIPS 2015)
Learning Problem

- A supervised deep generative model

\[
\begin{align*}
Z_K & \\
\vdots & \\
Z_1 & \\
X & \\
\end{align*}
\]

\[
z \sim p(z|\alpha) \quad x \sim p(x|z, \beta)
\]

\[
\min_{\theta, q(\eta, Z)} \mathcal{L}(\theta, q(\eta, Z); X) + \mathcal{CR}(q(\eta, Z; X))
\]
Classification Loss

- Hinge-loss of an averaging classifier

\[ f(y; x) = \mathbb{E}_q[\eta_y^T z] \]

\[ R(q(\eta, Z); X) = \sum_{n=1}^{N} \max_y (\Delta \ell_n(y) + f(y; x_n) - f(y_n; x_n)) \]
Learning with a Q-P Network

- Characterize the variational distribution with an inference network

For example (Kingma & Welling, 2013):

\[ q_\phi(z|x, y) = \mathcal{N}(\mu(x, y; \phi), \sigma^2(x, y; \phi)) \]

- where both mean and variance are nonlinear functions of data by a DNN
- Representation lemma ensures that the parameterization often exists
Doubly stochastic subgradient descent

- A doubly stochastic generalization of Pegasos (Shalev-Shwartz et al., 2011)

- Stochastic **unbiased** estimate of the objective by random mini-batches:

\[
\hat{\mathcal{L}}_m := \frac{N}{m} \sum_{n=1}^{m} \mathcal{L}(\theta, \phi; x_n) + \frac{\|\lambda\|^2}{2\sigma^2} + \frac{NC}{m} \sum_{n=1}^{m} \ell(\lambda, \phi; x_n)
\]

- Stochastic estimate of the per-sample variational bound and its subgradient (**unbiased**)
2-Layer MLP: Q-P network architecture

*Same as in Auto-Encoding Variational Bayes (VA) [Kingma & Welling, 2014]*
5-Layer CNN: Q-P network architecture
# Classification Performance

## MNIST

<table>
<thead>
<tr>
<th>Model</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>VA + Pegasos</td>
<td>1.04</td>
</tr>
<tr>
<td>VA + Class-conditionVA</td>
<td>0.96</td>
</tr>
<tr>
<td>MMVA</td>
<td>0.90</td>
</tr>
<tr>
<td>CVA + Pegasos</td>
<td>1.35</td>
</tr>
<tr>
<td>CMMVA</td>
<td>0.45</td>
</tr>
</tbody>
</table>

*Table: Error rates (%) on MNIST dataset.*

## SVHN

<table>
<thead>
<tr>
<th>Model</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVA + Pegasos</td>
<td>25.3</td>
</tr>
<tr>
<td>CMMVA</td>
<td>3.09</td>
</tr>
<tr>
<td>CNN Sermanet et al. [2012]</td>
<td>4.9</td>
</tr>
<tr>
<td>Stochastic Pooling Zeiler and Fergus [2013]</td>
<td>2.80</td>
</tr>
<tr>
<td>Maxout Network Goodfellow et al. [2013]</td>
<td>2.47</td>
</tr>
<tr>
<td>Network in Network Lin et al. [2014]</td>
<td>2.35</td>
</tr>
<tr>
<td>DSN Lee et al. [2015]</td>
<td>1.92</td>
</tr>
</tbody>
</table>
Figure: t-SNE embedding results for both (a) VA and (b) MMVA with a standard Gaussian prior and pre-training.
Generative Capability

MNIST & SVHN:
Imputation examples

Considering Rect\((12 \times 12)\) noise, CMMVA makes fewer mistakes and refines the images better, which accords with the MSE results.

(a) Original data    (b) Noisy data    (c) CVA    (d) CMMVA

**Figure:** (a): original test data; (b) test data with missing value; (c-d): results inferred by CVA and CMMVA respectively for 100 epochs.
Classification with Missing Values

CNN makes prediction on the incomplete data directly. CVA and CMMVA with default settings infer missing data for 100 iterations at first and then make prediction on the refined data.

**Table:** Error rates(%) with missing values on MNIST.

<table>
<thead>
<tr>
<th>NOISE LEVEL</th>
<th>CNN</th>
<th>CVA</th>
<th>CMMVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECT (6 x 6)</td>
<td>7.5</td>
<td>2.5</td>
<td>1.9</td>
</tr>
<tr>
<td>RECT (8 x 8)</td>
<td>18.8</td>
<td>4.2</td>
<td>3.7</td>
</tr>
<tr>
<td>RECT (10 x 10)</td>
<td>30.3</td>
<td>8.4</td>
<td>7.7</td>
</tr>
<tr>
<td>RECT (12 x 12)</td>
<td>47.2</td>
<td>18.3</td>
<td>15.9</td>
</tr>
</tbody>
</table>

CMMVA outperforms both VA and CNN in this scenario.
Gray the Black-Box

(Liu, Shi, Li, Li, Zhu, Liu, VAST 2016)
Gray the Black-Box

Towards better analysis of deep CNN (joint with Shixia Liu)

http://cgcad.thss.tsinghua.edu.cn/mengchen/video/CNNVis-final.mp4
http://shixialiu.com/publications/cnnvis/demo/index.html
Gray the Black-Box
Summary

- Bayesian methods are highly relevant for deep learning
  - Protect from overfitting
  - Structure learning
  - Uncertainty
- Inference is a challenge
  - Inference with Q-P network
- Learning to generate with memory & attention
- Max-margin learning with DGMs
Thanks!

Some code available at:

http://ml.cs.tsinghua.edu.cn/~jun