Deep Generative Models

Durk Kingma

Max Welling

Deep Probabilistic Models Workshop
Wednesday, 1st of Oct, 2014
D.P. Kingma
Deep generative models

- Transformations between Bayes nets and Neural nets
  “Transformation between Bayes nets and Neural Nets”. *ICML’14*

- Deep generative models of images
  Auto-Encoding Variational Bayes, *ICLR’14*

- Semi-Supervised Learning with Deep Generative Models
  *NIPS’14*; with Shakir & Danilo (Deepmind)
Part 1:
Transformations between Bayes nets and Neural nets
Directed latent variables models

- Can model complex (multimodal) distributions over observed variables

Feedforward neural nets:

- Typically used for learning a single conditional: \( \log p(y|x) = f(x,y) \)
  - e.g. Classification/regression with deep net

But: not good for modelling complex distributions, e.g. multi-modal distributions
Neural Nets and Deep Generative Models (2)

- Natural idea: neural nets as components of deep latent-variable models
  - Learn complex distributions over data

- Inner loop of learning requires **posterior inference**
  - Exact inference is intractable

- Efficient gradient-based approximate inference methods:
  - Stochastic Variational inference (biased but fast)
    [Hoffman & Blei 2012] [Kingma & Welling 2013] [Deepmind 2014]
  - Sampling-based methods (unbiased but slower), e.g., HMC

Can we increase the efficiency of gradient-based posterior inference?

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

\[
\nabla_z \log p(z|x) = \nabla_z \log p(x, z)
\]
Idea: Reparameterizations (Gaussian example)

Centered form (CP)

\[ p_\theta(z|pa) = \mathcal{N}(z; \mu, \sigma^2 I) \]
\[ \mu = f_\theta(pa) \]
\( (\text{e.g. neural net}) \)

Differentiable Non-Centered Form (DNCP)

\[ \epsilon \sim \mathcal{N}(0, I) \]
\[ \tilde{z} = f_\theta(pa) + \sigma \cdot \epsilon \]

Neural net perspective: hidden unit with injected noise
Idea: Reparameterizations

Centered form (CP)

\[ p_{\theta}(z|p) \]

Differentiable Non-Centered Form (DNCP)

\[ \epsilon \sim p(\epsilon) \]
\[ z = g(p, \epsilon, \theta) \]

Neural net perspective: hidden unit with injected noise
Reparameterizations

- Can be performed for a broad class of distributions

<table>
<thead>
<tr>
<th>Example</th>
<th>$q_\phi(z)$</th>
<th>$p(\varepsilon)$</th>
<th>$g(\phi, \varepsilon)$</th>
<th>Also...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal dist.</td>
<td>$z \sim \text{N}(\mu, \sigma)$</td>
<td>$\varepsilon \sim \text{N}(0,1)$</td>
<td>$z = \mu + \sigma \times \varepsilon$</td>
<td>Location-scale family: Laplace, Elliptical, Student’s t, Logistic, Uniform, Triangular, ...</td>
</tr>
<tr>
<td>Exponential</td>
<td>$z \sim \exp(\lambda)$</td>
<td>$\varepsilon \sim \text{U}(0,1)$</td>
<td>$z = -\log(1 - \varepsilon)/\lambda$</td>
<td>Invertible CDF: Cauchy, Logistic, Rayleigh, Pareto, Weibull, Reciprocal, Gompertz, Gumbel and Erlang, ...</td>
</tr>
<tr>
<td>Other</td>
<td>$z \sim \text{logN}(\mu, \sigma)$</td>
<td>$\varepsilon \sim \text{N}(0,1)$</td>
<td>$z = \exp(\mu + \sigma \times \varepsilon)$</td>
<td>Gamma, Dirichlet, Beta, Chi-Squared, and F distributions</td>
</tr>
</tbody>
</table>
So: we have a choice

Which form to use for learning / inference?
Posterior correlation analysis (1)

Squared posterior correlation $\rho^2$
second-order metric of posterior dependency
between pair of latent variables A and B

$$C = \begin{pmatrix} \sigma_A^2 & \sigma_{AB}^2 \\ \sigma_{AB}^2 & \sigma_B^2 \end{pmatrix} = -\mathbf{H}^{-1} = \frac{1}{\det(\mathbf{H})} \begin{pmatrix} -H_B & H_{AB} \\ H_{AB} & -H_A \end{pmatrix}$$

$$\rho^2 = (\sigma_{AB}^2)^2 / (\sigma_A^2 \sigma_B^2)$$
$$= (H_{AB}/\det(\mathbf{H}))^2 / ((-H_A/\det(\mathbf{H}))( -H_B/\det(\mathbf{H})))$$
$$= H_{AB}^2 / (H_A H_B)$$
Posterior correlation analysis (2)

• Squared correlations for CP:

\[ \rho^2_{pa(z),z} = \frac{(H_{pa(z)z})^2}{H_{pa(z)} H_{zz}} = \frac{w_i^2/\sigma^4}{(\alpha - w_i^2/\sigma^2)(\sigma^{-2}_{z|ch(z)} - 1/\sigma^2)} \]

• Squared correlations for DNCP:

\[ \rho^2_{pa(z),\epsilon} = \frac{(H_{pa(z)\epsilon})^2}{H_{pa(z)} H_{\epsilon\epsilon}} = \frac{\sigma^2 w_i^2 (\sigma^{-2}_{z|ch(z)})^2}{(\alpha + w_i^2(\sigma^{-2}_{z|ch(z)})) (\sigma^2 \sigma^{-2}_{z|ch(z)} - 1)} \]
Inequality: all terms cancel with very simple result:

\[ \rho_{pa(z),z}^2 > \rho_{pa(z),\epsilon}^2 \]

\[ \Leftrightarrow \]

\[ \sigma_{z|pa(z)}^2 < \sigma_{z|ch(z)}^2 \]

i.e., DNCP leads to more efficient inference when latent variable 'z' is more strongly bound to its parents then to its children.
Posterior correlation analysis (4)

| Condition                        | \( \rho^2_{y_i, z} \) (CP) | \( \rho^2_{y_i, e} \) (DNCP) | \( \sigma^2_{z|pa(z)} \) |
|----------------------------------|-----------------------------|-------------------------------|-----------------------------|
| \( \lim \sigma_{z|pa(z)} \to 0 \) | 1                           | 0                             |                             |
| \( \lim \sigma_{z|pa(z)} \to \infty \) | 0                           | \( \frac{\sigma_{z|ch(z)} w_i^2}{\sigma_{z|ch(z)} w_i^2 + \alpha} \) |                             |
| \( \lim \sigma_{z|ch(z)} \to 0 \) | 0                           | 1                             |                             |
| \( \lim \sigma_{z|ch(z)} \to \infty \) | \( \frac{w_i^2}{w_i^2 - \alpha \sigma^2_{z|pa(z)}} \) | 0                             |                             |

"beauty-and-beast pair"

© Disney
2D linear-Gaussian posterior example

\[ \sigma_Z = 50 \]

\[ \sigma_Z = 1 \]

\[ \sigma_Z = 0.02 \]
Proposal distribution is mix between two proposal distributions:

\[ Q(z'|z) = \rho \cdot Q_{CP}(z'|z) + (1 - \rho) \cdot Q_{DNCP}(z'|z) \]

Where we use \( \rho = 0.5 \) in experiments
Experiment: HMC sampling with Dynamic Bayesian network (DBN)

\[ p_\theta(z_t | z_{t-1}) = \mathcal{N}(z; \tanh(W z_{t-1}), \sigma_z^2 I) \]

\[ p_\theta(x_t | z_t) = \mathcal{N}(z; \tanh(W z_t), \sigma_x^2 I) \]
**Autocorrelation results on DBN**

<table>
<thead>
<tr>
<th>log $\sigma_z$</th>
<th>CP</th>
<th>DNCP</th>
<th>ROBUST</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>2</td>
<td>305</td>
<td>640</td>
</tr>
<tr>
<td>-4.5</td>
<td>26</td>
<td>348</td>
<td>498</td>
</tr>
<tr>
<td>-4</td>
<td>10</td>
<td>570</td>
<td>686</td>
</tr>
<tr>
<td>-3.5</td>
<td>225</td>
<td>417</td>
<td>624</td>
</tr>
<tr>
<td>-3</td>
<td>386</td>
<td>569</td>
<td>596</td>
</tr>
<tr>
<td>-2.5</td>
<td>542</td>
<td>608</td>
<td>900</td>
</tr>
<tr>
<td>-2</td>
<td>406</td>
<td>972</td>
<td>935</td>
</tr>
<tr>
<td>-1.5</td>
<td>672</td>
<td>1078</td>
<td>918</td>
</tr>
<tr>
<td>-1</td>
<td>1460</td>
<td>1600</td>
<td>1082</td>
</tr>
</tbody>
</table>

---

![Graphs](image)

- **CP**: Terribly inefficient
- **DNCP**: Very efficient
Part 2:
The Variational Auto-Encoder
Deep latent variable model
Monte Carlo EM
Monte Carlo EM - End result
MAP inference with L-BFGS
The Variational Auto-Encoder (ICLR’14)

$$L = -D_{KL}(\tilde{p}(x)q(z|x)\|p(x, z))$$
$$\leq \log p(x)$$
Why is this an auto-encoder?

Reparameterized $Q(z|x)$  

$P(x|z)$  

$x \rightarrow h \rightarrow h \rightarrow z \rightarrow h \rightarrow h \rightarrow x$

$L = D_{KL}(\widetilde{p}(x)q(z|x)\|p(x,z))$

$= \mathbb{E}_{\widetilde{p}(x)} \left[ \mathbb{E}_{p(\epsilon)} \left[ \log p(x|z) + \log p(z) - \log q(z|x) \right] \right]$  

Reconstruction error  

Regularization terms (dictated by the bound)
Variational Auto-Encoder trained on MNIST
(2D Latent space)
3D latent space
Labeled faces in the wild
Semi-supervised learning with deep generative models ("Model 2.5")
Approach 1

DLGM / VAE as feature extractor for semi-supervised classifier

\[ L = -D_{KL}(\tilde{p}(x)q(z|x) \parallel p(x, z)) \]
Approach 2
DLGM / VAE as regularizer of neural net classifier

\[
q(y|x) q(z|x, y) \quad p(y) p(z)
\]

\[L = \mathbb{E}_{\tilde{p}_i(x, y)} [\log q(y|x)] + \beta L^{reg}\]
\[L^{reg} = - D_{KL}( \tilde{p}_i(x, y) q(z|x, y) \parallel p(x, y, z) )\]
\[- D_{KL}( \tilde{p}_u(x) q(y|x) q(z|x, y) \parallel p(x, y, z) )\]
Approach 3
Table 1: Benchmark results of semi-supervised classification on MNIST with few labels.

<table>
<thead>
<tr>
<th>$N_{labeled}$</th>
<th>NN</th>
<th>CNN</th>
<th>TSVM</th>
<th>EmbNN</th>
<th>CAE</th>
<th>MTC</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>25.81</td>
<td>22.98</td>
<td>16.81</td>
<td>16.86</td>
<td>13.47</td>
<td>12.03*</td>
<td>11.82 (± 0.25)</td>
<td>11.97 (± 1.71)</td>
<td>3.54 (± 0.03)</td>
</tr>
<tr>
<td>600</td>
<td>11.44</td>
<td>7.68</td>
<td>6.16</td>
<td>5.97</td>
<td>6.3</td>
<td>5.13*</td>
<td>5.72 (± 0.049)</td>
<td>4.94 (± 0.13)</td>
<td>2.85 (± 0.1)</td>
</tr>
<tr>
<td>1000</td>
<td>10.7</td>
<td>6.45</td>
<td>5.38</td>
<td>5.73</td>
<td>4.77</td>
<td>3.64*</td>
<td>4.24 (± 0.07)</td>
<td>3.60 (± 0.56)</td>
<td>2.76 (± 0.30)</td>
</tr>
<tr>
<td>3000</td>
<td>6.04</td>
<td>3.35</td>
<td>3.45</td>
<td>3.59</td>
<td>3.22</td>
<td>2.57*</td>
<td>3.49 (± 0.04)</td>
<td>3.92 (± 0.63)</td>
<td>2.63 (± 0.24)</td>
</tr>
</tbody>
</table>
Analogic reasoning

(will improve)
Next steps

- SSL on larger images
  - SVHN, CIFAR-10 (this NIPS paper)
  - Imagenet (Future papers)
- SSL on video’s
  - Youtube
- Applications of anagogic reasoning
Thanks!
RMSProp

Initialization:
\[ m \leftarrow 0 \]
\[ v \leftarrow 0 \]

Update parameters \((w, g)\):
\[ m \leftarrow \beta_1 \cdot g + (1 - \beta_1) \cdot m \]
\[ v \leftarrow \beta_2 \cdot g^2 + (1 - \beta_2) \cdot v \]
\[ w \leftarrow w - \alpha \cdot m / \sqrt{v} \]
Estimation based on exp. mov. avg.

Assume: $g$ is a random variable
Goal: from exponential moving averages of i.i.d. draws $g_i$, estimate moments $E[g]$ and $E[g^2]$

$$m_t = \beta_1 \cdot g_t + (1 - \beta_1) \cdot m_{t-1}$$

$$= \beta_1 \sum_{i=1}^{t} (1 - \beta_1)^{t-i} \cdot g_i$$

(As function of past samples)

$$E[m_t] = E \left[ \beta_1 \sum_{i=1}^{t} (1 - \beta_1)^{t-i} \cdot g_i \right]$$

(Taking expectations)

$$= E[g] \cdot \beta_1 \sum_{i=1}^{t} (1 - \beta_1)^{t-i}$$

(Because i.i.d.)

$$= E[g] \cdot (1 - (1 - \beta_1)^t)$$

(Further simplification)
Fixed RMSProp (AdaM)

Initialization:
\[ m \leftarrow 0 \]
\[ v \leftarrow 0 \]

Update \_parameters(\(w, g\)) :
\[ m \leftarrow \beta_1 \cdot g + (1 - \beta_1) \cdot m \]
\[ v \leftarrow \beta_2 \cdot g^2 + (1 - \beta_2) \cdot v \]
\[ w \leftarrow w - \alpha \cdot \gamma_t \cdot m / \sqrt{v} \]

where \[ \gamma_t = \sqrt{1 - (1 - \beta_2)^t} / (1 - (1 - \beta_1)^t) \]
VAE, 10 epochs

\[ \beta_1 = 1 \]

\[ \beta_1 = 0.1 \]

\[ \beta_2 = 0.01 \quad \beta_2 = 0.001 \quad \beta_2 = 0.0001 \]

More like AdaGrad (infinite memory)
VAE, 100 epochs

\[ \beta_1 = 1 \]

\[ \beta_1 = 0.1 \]

\[ \beta_2 = 0.01 \]
\[ \beta_2 = 0.001 \]
\[ \beta_2 = 0.0001 \]

More like AdaGrad (infinite memory)
Now possible: Interpolation between Adagrad and RMSProp

Initialization:
\[ \mathbf{m} \leftarrow 0 \]
\[ \mathbf{v} \leftarrow 0 \]

Update parameters \((\mathbf{w}, \mathbf{g})\):
\[ \mathbf{m} \leftarrow \beta_1 \cdot \mathbf{g} + (1 - \beta_1) \cdot \mathbf{m} \]
\[ \mathbf{v} \leftarrow \beta_2 \cdot \mathbf{g}^2 + (1 - \beta_2) \cdot \mathbf{v} \]
\[ \mathbf{w} \leftarrow \mathbf{w} - \alpha \cdot \gamma_t \cdot \mathbf{m} / \sqrt{t} \cdot \mathbf{v} \]

where \[ \gamma_t = \sqrt{1 - (1 - \beta_2)^t / (1 - (1 - \beta_1)^t)} \]

This algorithm AdaGrad when \(\beta_1 = 1\) and \(\beta_2 = \varepsilon\) (infinitesimal)