Variational Auto-Encoders and Extensions

Diederik (Durk) Kingma

Max Welling

Universiteit van Amsterdam
Parts of this talk

1. Basics
2. Extension with auxiliary variables
3. Recent advances
Variational auto-encoders
Basic problem setup

- We assume:
  - **A huge dataset** of observations
    \[ X = \{x^1, \ldots, x^N\} \]
  - There exists a **simple latent space** \( z \): 
    \[ z \sim p_\Theta(z) \]
    \[ x|z \sim p_\Theta(x|z) \]
  - **Exact posterior distribution** \( p_\Theta(z|x) \) **is intractable** so **EM is intractable**

- We wish to:
  - **efficiently learn** approx. maximum likelihood parameters \( \Theta \)
  - **efficiently infer** latent variables for new observations \( x \)
Deep Latent-Variable Models

- directed latent variables model
  - can represent complicated marginal distributions over $x$

- deep neural nets
  - can represent complicated conditional dependencies

- We combine those strengths

Mathematical notation:

$$p(x, z_1, z_2) = p(x|z_2)p(z_2|z_1)p(z_1)$$
Example model

- \( p(z) = N(0, I) \)
- \( p_\theta(x|z) = N(\mu, \sigma^2) \)
  \[ \mu = f_\theta(z) = \text{multilayer neural net} \]
- With flexible neural net \( f_\theta(z) \), \( p_\theta(x) \) can be almost arbitrarily complicated / multi-modal distribution
- But intractable posterior distribution \( p(z|x) \)
  - Need approximate inference for learning
Non-variational approx. inference methods

- Point estimate of $p(z|x)$ (MAP)
  - **Pro**: simple, fast
  - **Con**: too biased / approximate

- Markov Chain Monte Carlo (MCMC):
  - **Pro**: asymptotically unbiased
  - **Con**: often expensive, hard to assess convergence
Variational inference with Inference Networks

- Introduce **parametric model** $q_\phi(z|x)$ of true posterior
  - $\phi$: **variational parameters**
  - parameterised by neural networks

  - Also learns **parametric inference model** $q(z|x)$ with SGD
  - But wake-sleep uses **different objective for** $q_\phi(z|x)$ which doesn’t optimise a bound log $p(x)$
Auto-Encoding Variational Bayes

[Kingma and Welling, 2013/2014]
[Rezende et al, 2014]

- $q_\phi(z|x) = \mathcal{N}(\mu, \sigma^2)$
  $[\mu, \sigma^2] = f^{(z|x)}(x, \phi) = \text{multilayer neural net}$

- Objective: lower bound of log $p(x)$.
  - Jointly optimized w.r.t. $\phi$ and $\theta$
  - This is approx. maximum likelihood
  - Simple SGD:
    - Sampling small minibatches of data
    - Sampling from approx. posterior

- This also minimizes an expected KL divergence
  $D_{KL}(q_\phi(z|x)||p(z|x))$
  $\rightarrow$ gives us cheap approx. inference for new datapoints
Variational bound

\[
\log p(x) = \mathcal{L}(x) + D_{KL}(q_\phi(z|x)||p(z|x))
\]

Objective per datapoint:

\[
\mathcal{L}(x) = \mathbb{E}_{q_\phi(z|x)} \left[ \log p(x, z) - \log q_\phi(z|x) \right]
\]

Abbreviated:

\[
\mathcal{L}(x) = \mathbb{E}_{q_\phi(z|x)} \left[ f_\phi(x, z) \right]
\]
Relatively new idea: **Stochastic Gradient-based Variational Inference**

Objective per datapoint: $$\mathcal{L}(x) = \mathbb{E}_{q_{\phi}(z|x)}[f_{\phi}(x, z)]$$

- Often no analytical solution to exact gradient $\nabla_{\phi} L$

- Solution: *(doubly) stochastic gradient ascent*

  - Only requires *unbiased estimates* of gradient

  - Can use *small minibatches of data*
Reparameterization trick

Original form

Reparameterised form

\[
\text{Backprop} \quad \frac{\partial f}{\partial \phi_i} \quad \frac{\partial f}{\partial z_j} = \frac{\partial L}{\partial \phi_i}
\]

\[
\sim q(z|\phi, x)
\]

\[
\sim p(\varepsilon)
\]

\[
g(\phi, x, \varepsilon)
\]

\[
\frac{\partial f}{\partial \phi_i} \approx \frac{\partial L}{\partial \phi_i}
\]

Deterministic node

Random node

[Kingma, 2013]
[Bengio, 2013]
[Kingma and Welling 2014]
[Rezende et al 2014]
Reparameterization trick

- Can be performed for a broad class of distributions, e.g.:
  - **Location-scale transforms**
    - Normal, Laplace, Student t’s, Logistic, etc.
  - **Inverse of CDF**
    - Cauchy, Rayleight, Pareto, etc
  - Other strategies exist
    - Gamma, Dirichlet, Beta, Chi-Squared, etc
Stochastic Gradient Variational Inference

- Variational inference by gradient ascent

- “Swiss army knife” for inference:
  - Works with almost any $p(x,z)$
  - Works with almost any $q(z|x)$
  - Just requires gradient ascent on single objective
Connection to auto-encoders

- Variational Auto-Encoder (\textbf{VAE}):
  - \( q(z|x) \): stochastic neural encoder
  - \( p(x|z) \): stochastic neural decoder

Objective function:

\[
L = (\log p(x|z) + \log p(z) - \log q(z|x))|_{z=g(\epsilon)}
\]

- Reconstruction error
- Regularization terms dictated by the bound
Conv. net as encoder/decoder, trained on faces

(trained by Alec Radford 2015)
Classifier vs generative model

Each edge is parameterised as a deep neural net
Generative model
for semi-supervised learning

Inference model

\[ q(y|x) = \text{classifier} \]

Generative model

Variational training includes optimisation of \( q(y|x) \)
VAEs are SOTA on semi-supervised learning on MNIST

“Improving Semi-Supervised Learning with Auxiliary Deep Generative Models”

[Maaløe, Sønderby, Sønderby and Winter, 2015]
Analogy-making

[Diagram of analogies and number sequences]
Inference model

Generative model

\[ p(x,z) \]

Inference model

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

Implicit inference model

\[ q(z|x) = \int q(y,z|x) \, dy \]

- **Pro:** more accurate/flexible inference model \( q(z|x) \)
- **But:** intractable PDF \( q(z|x) = \int q(y,z|x) \, dy \)
Inference model
\( q(y,z|x) \)

Generative model
\( p(x,z) \)

auxiliary model
\( r(y|x,z) \)

\[ \mathcal{L}_{aux} = \mathbb{E}_{q_{\phi}(y,z|x)} \left[ \log p(x, z) - \log q_{\phi}(y, z|x) + \log r_{\phi}(y|z, x) \right] \]
Generative model \( p(x,z) \)

Inference model \( q(z_1, \ldots, z_T | x) \)

Auxiliary model \( r(z_1, \ldots, z_{T-1} | z_T, x) \)

MCMC chain, e.g. Hamiltonian Monte Carlo, as inference model with auxiliary variables
### Table 1.
Comparison of our approach to other recent methods in the literature. We compare the average marginal log-likelihood measured in nats of the digits in the MNIST test set. See section 3.2 for details.

<table>
<thead>
<tr>
<th>Model</th>
<th>( \log p(x) )</th>
<th>( \log p(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \leq - )</td>
<td>( = - )</td>
</tr>
<tr>
<td><strong>HVI + fully-connected VAE:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Without inference network:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 leapfrog steps</td>
<td>90.86</td>
<td>87.16</td>
</tr>
<tr>
<td>10 leapfrog steps</td>
<td>87.60</td>
<td>85.56</td>
</tr>
<tr>
<td>With inference network:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No leapfrog steps</td>
<td>94.18</td>
<td>88.95</td>
</tr>
<tr>
<td>1 leapfrog step</td>
<td>91.70</td>
<td>88.08</td>
</tr>
<tr>
<td>4 leapfrog steps</td>
<td>89.82</td>
<td>86.40</td>
</tr>
<tr>
<td>8 leapfrog steps</td>
<td>88.30</td>
<td>85.51</td>
</tr>
<tr>
<td><strong>HVI + convolutional VAE:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No leapfrog steps</td>
<td><strong>86.66</strong></td>
<td><strong>83.20</strong></td>
</tr>
<tr>
<td>1 leapfrog step</td>
<td>85.40</td>
<td>82.98</td>
</tr>
<tr>
<td>2 leapfrog steps</td>
<td>85.17</td>
<td>82.96</td>
</tr>
<tr>
<td>4 leapfrog steps</td>
<td>84.94</td>
<td>82.78</td>
</tr>
<tr>
<td>8 leapfrog steps</td>
<td>84.81</td>
<td>82.72</td>
</tr>
<tr>
<td>16 leapfrog steps</td>
<td>84.11</td>
<td>82.22</td>
</tr>
<tr>
<td>16 leapfrog steps, ( n_h = 800 )</td>
<td><strong>83.49</strong></td>
<td><strong>81.94</strong></td>
</tr>
<tr>
<td>From (Gregor et al., 2015):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBN 2hl</td>
<td>84.55</td>
<td></td>
</tr>
<tr>
<td>EoNADE</td>
<td>85.10</td>
<td></td>
</tr>
<tr>
<td>DARN 1hl</td>
<td>88.30</td>
<td>84.13</td>
</tr>
<tr>
<td>DARN 12hl</td>
<td>87.72</td>
<td></td>
</tr>
<tr>
<td>DRAW</td>
<td><strong>80.97</strong></td>
<td></td>
</tr>
</tbody>
</table>

Large improvement in nats

DRAW: Gregor et al, 2014
Recurrent VAE with attention
Variational Inference with Normalizing Flows
[Rezende and Mohamed, ICML 2015]
A Recurrent Latent Variable Model for Sequential Data
[Chung et al, 2015]

Generating speech with a variational RNN
Generating Images from Captions with Attention

[Mansimov et al, 2015]
(under submission at ICLR)

A stop sign is flying in blue skies.

A herd of elephants flying in the blue skies.

A toilet seat sits open in the grass field.

A person skiing on sand clad vast desert.
Importance Weighted Autoencoders

[Burda et al 2015]
(under submission at ICLR)

- Objective:

\[ \mathcal{L}_k(x) = \mathbb{E}_{h_1, \ldots, h_k \sim q(h|x)} \left[ \log \frac{1}{k} \sum_{i=1}^{k} \frac{p(x, h_i)}{q(h_i|x)} \right] \]

- Equivalent to VAE objective for \( k=1 \)

- For \( k>1 \), optimizes a tighter bound, at the expense of extra computation
Summary

- Variational Auto-Encoding: **scalable generative modeling**
  - Works with almost any model $p(x,z)$
  - Works with almost any approx. posterior $q(z|x)$
  - Scales to huge datasets
  - Just requires gradient ascent on single objective

- Applications:
  - deep synthesis / analogic reasoning
  - nonlinear PCA
  - semi-supervised learning
  - optimizing MCMC hyper-parameters

https://github.com/dpkingma