Tutorial on Deep Generative Models

Shakir Mohamed and Danilo Rezende

@shakir_za @deepsipiker
Abstract

This tutorial will be a review of recent advances in deep generative models. Generative models have a long history at UAI and recent methods have combined the generality of probabilistic reasoning with the scalability of deep learning to develop learning algorithms that have been applied to a wide variety of problems giving state-of-the-art results in image generation, text-to-speech synthesis, and image captioning, amongst many others. Advances in deep generative models are at the forefront of deep learning research because of the promise they offer for allowing data-efficient learning, and for model-based reinforcement learning. At the end of this tutorial, audience member will have a full understanding of the latest advances in generative modelling covering three of the active types of models: Markov models, latent variable models and implicit models, and how these models can be scaled to high-dimensional data. The tutorial will expose many questions that remain in this area, and for which there remains a great deal of opportunity from members of the UAI community.
Beyond Classification

- Move beyond associating inputs to outputs
- Understand and imagine how the world evolves
- Recognise objects in the world and their factors of variation
- Detect surprising events in the world
- Establish concepts as useful for reasoning and decision making
- Anticipate and generate rich plans for the future
What is a Generative Model?

- A model that allows us to learn a simulator of data
- Models that allow for (conditional) density estimation
- Approaches for unsupervised learning of data

Characteristics are:
- **Probabilistic** models of data that allow for uncertainty to be captured.
- **Data distribution** $p(x)$ is targeted.
- **High-dimensional** outputs.
Why Generative Models?
Why Generative Models

Generative models have a role in many problems.

- Super-resolution, Compression, Text-to-speech
- Proteomics, Drug Discovery, Astronomy, High-energy physics
- Planning, Exploration, Intrinsic motivation, Model-based RL
Drug Design and Response Prediction

Proposing candidate molecules and for improving prediction through semi-supervised learning.

Locating Celestial Bodies

Generative models for applications in astronomy and high-energy physics.

Regier et al., 2015
Image super-resolution

Photo-realistic single image super-resolution

original

bicubic
(21.59dB/0.6423)

SRGAN
(20.34dB/0.6562)

Ledig et al., 2016
Text-to-speech Synthesis

Generating audio conditioned on text

Oord et al., 2016
Image and Content Generation

Generating images and video content.

DRAW

Pixel RNN

ALI

Gregor et al., 2015, Oord et al., 2016, Dumoulin et al., 2016
Communication and Compression

Hierarchical compression of images and other data.

Original images

Compression rate: 0.2bits/dimension

JPEG

JPEG-2000

RVAE v1

RVAE v2

Gregor et al., 2016
One-shot Generalisation

Rapid generalisation of novel concepts

Rezende et al., 2016
Visual Concept Learning

Understanding the factors of variation and invariances.

Higgins et al., 2017
Future Simulation

Simulate future trajectories of environments based on actions for planning

Atari simulation

Robot arm simulation

Chiappa et al, 2017, Kalchbrenner et al., 2017
Scene Understanding

Understanding the components of scenes and their interactions

Wu et al., 2017
Probabilistic Deep Learning
Two Streams of Machine Learning

**Deep Learning**

+ Rich non-linear models for classification and sequence prediction.
+ Scalable learning using stochastic approximation and conceptually simple.
+ Easily composable with other gradient-based methods.

- Only point estimates.
- Hard to score models, do selection and complexity penalisation.

**Probabilistic Reasoning**

- Mainly conjugate and linear models.
- Potentially intractable inference, computationally expensive or long simulation time.

+ Unified framework for model building, inference, prediction and decision making.
+ Explicit accounting for uncertainty and variability of outcomes.
+ Robust to overfitting; tools for model selection and composition.

Complementary strengths, making it natural to combine them
Thinking about Machine Learning

1. Models
2. Learning Principles
3. Algorithms
Types of Generative Models

**Fully-observed models**
Model observed data directly without introducing any new unobserved local variables.

**Latent Variable Models**
Introduce an unobserved random variable for every observed data point to explain hidden causes.

- **Prescribed models**: Use observer likelihoods and assume observation noise.
- **Implicit models**: Likelihood-free models.
Spectrum of Fully-observed Models

Directed

Discrete

NADE, EoNADE
Fully-visible sigmoid belief networks
Pixel CNN/RNN
RNN Language mod.
Context tree switching

Boltzmann Machines
Discrete Markov
Random Fields
Ising, Hopfield
and Potts Models

Normal Means
Continuous
Markov Models
N-AR(p)
RNADE

Gaussian MRFs
Log-linear models

Continuous

Undirected
Building Generative Models

Equivalent ways of representing the same DAG

\[
p(x_1, \ldots, N) = \prod_{i=1}^{N} p(x_i | x_1, \ldots, (i-1)) \quad \quad \quad \quad \quad p(x_1, \ldots, N) = \prod_{i=1}^{N} p(x_i | s_i(s_{i-1}, x_{i-1}))
\]
Fully-observed Models

\[ p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i | x_1, \ldots, x_{i-1}) \]

+ Can directly encode how observed points are related.
+ Any data type can be used
+ For directed graphical models: Parameter learning simple
  * Log-likelihood is directly computable, no approximation needed.
+ Easy to scale-up to large models, many optimisation tools available.

- Order sensitive.
- For undirected models, parameter learning difficult: Need to compute normalising constants.
- Generation can be slow: iterate through elements sequentially, or using a Markov chain.

All conditional probabilities described by deep networks.
Spectrum of Latent Variable Models

Cascaded Indian Buffet process
Hierarchical Dirichlet process
Deep Nonparametric Discrete

Sigmoid Belief Net
Deep auto-regressive networks (DARN)
Deep Parametric Discrete

Indian buffet process
Dirichlet process mixture
Direct Nonparametric Discrete

Hidden Markov Model
Discrete LVM
Sparse LVMs
Linear Parametric Discrete

Deep Gaussian processes
Recurrent Gaussian Process
GP State space model
Deep Nonparametric Continuous

Nonlinear factor analysis
Nonlinear Gaussian belief network
Deep Latent Gaussian (VAE, DRAW)
Deep Parametric Continuous

PCA, factor analysis
Independent components analysis
Gaussian LDS
Latent Gauss Field
Linear Parametric Continuous

Gaussian process LVM
Direct Nonparametric Continuous
Building Generative Models

\[ p(x, z, \theta) = \rho(\theta) \prod_{i=1}^{N} p(x_i | z_i, \theta) \pi(z_i) \]

\[ \pi(z) = \mathcal{N}(0, \mathbb{I}_{d_z}) \]

\[ \rho(\theta) = \mathcal{N}(0, \kappa^2 \mathbb{I}_{d_\theta}) \]

\[ p(x | z, \theta) = \mathcal{N}(\theta_0 + \theta_1 z, \exp(\theta_2)) \]

\[ \theta = \{ \theta_0 \in \mathbb{R}^{d_x}, \theta_1 \in \mathbb{R}^{d_x \times d_z}, \theta_2 \in \mathbb{R}^{d_x} \} \]
Building Generative Models

Graphical Models + Computational Graphs (aka NNets)

\[ \pi(z) = \mathcal{N}(0, \mathbb{I}_{d_z}) \]
\[ \rho(\theta) = \mathcal{N}(0, \kappa^2 \mathbb{I}_{d_\theta}) \]
\[ p(x|z, \theta) = \mathcal{N}(\theta_0 + \theta_1 z, \exp(\theta_2)) \]

\[ \pi(z) = \mathcal{N}(0, \mathbb{I}_{d_z}) \]
\[ \rho(\theta) = \mathcal{N}(0, \kappa^2 \mathbb{I}_{d_\theta}) \]
\[ h_1 = \theta_0 + \theta_1 z \]
\[ h_2 = \exp(\theta_2) \]
\[ p(x|z, \theta) = \mathcal{N}(h_1, h_2) \]
Latent Variable Models

+ Easy sampling.
+ Easy way to include hierarchy and depth.
+ Easy to encode structure
+ Avoids order dependency assumptions: marginalisation induces dependencies.
+ Provide compression and representation.
+ Scoring, model comparison and selection possible using the marginalised likelihood.

- Inversion process to determine latents corresponding to an input is difficult in general.
- Difficult to compute marginalised likelihood requiring approximations.
- Not easy to specify rich approximations for latent posterior distribution.

\[ p(x, z, \theta) = \rho(\theta) \prod_{i=1}^{N} p(x_i | z_i, \theta) \pi(z_i) \]

Introduce an unobserved local random variable that represents hidden causes.
Choice of Learning Principles

For a given model, there are many competing inference methods.

- Exact methods (conjugacy, enumeration)
- Numerical integration (Quadrature)
- Generalised method of moments
- Maximum likelihood (ML)
- Maximum a posteriori (MAP)
- Laplace approximation
- Integrated nested Laplace approximations (INLA)
- Expectation Maximisation (EM)
- Monte Carlo methods (MCMC, SMC, ABC)
- Contrastive estimation (NCE)
- Cavity Methods (EP)
- Variational methods
Combining Models and Inference

A given model and learning principle can be implemented in many ways.

### Convolutional neural network + penalised maximum likelihood
- Optimisation methods (SGD, Adagrad)
- Regularisation (L1, L2, batchnorm, dropout)

### Implicit Generative Model + Two-sample testing
- Method-of-moments
- Approximate Bayesian Computation (ABC)
- Generative adversarial network (GAN)

### Latent variable model + variational inference
- VEM algorithm
- Expectation propagation
- Approximate message passing
- Variational auto-encoders (VAE)

### Restricted Boltzmann Machine + maximum likelihood
- Contrastive Divergence
- Persistent CD
- Parallel Tempering
- Natural gradients
<table>
<thead>
<tr>
<th>Objective</th>
<th>Quantity of Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction</td>
<td>$p(x(t+1), \ldots, \infty</td>
</tr>
<tr>
<td>Planning</td>
<td>$J = \mathbb{E}_p \left[ \int_0^{\infty} dt C(x_t) \mid x_0, u \right]$</td>
</tr>
<tr>
<td>Parameter estimation</td>
<td>$p(\theta \mid x_0, \ldots, N)$</td>
</tr>
<tr>
<td>Experimental Design</td>
<td>EIG = D$[p(f(x_t, \ldots, \infty) \mid u); p(f(x_{-\infty}, \ldots, t))]$</td>
</tr>
<tr>
<td>Hypothesis testing</td>
<td>$\frac{p(f(x_{-\infty}, \ldots, t) \mid H_0)}{p(f(x_{-\infty}, \ldots, t) \mid H_1)}$</td>
</tr>
</tbody>
</table>
Approximate Inference
Latent Variable Models

\[ x \in \mathbb{R}^{d_x} \quad z \in \mathbb{R}^{d_z} \quad \theta \in \mathbb{R}^{d_\theta} \]

\[ \mathcal{D} = \{ x_i \} \quad i \in \{ 1, \ldots, N \} \]

\[ \log p_\theta(x) = \log \int p_\theta(x|z)p(z)dz = \log \mathbb{E}_{p(z)}[p_\theta(x|z)] \]

\[ \log p_\theta(\mathcal{D}) = \sum_{i=1}^{N} \log \mathbb{E}_{p(z)}[p_\theta(x_i|z)] \]
Methods for Approximate Inference

- Laplace approximations
- Importance sampling
- Variational approximations
- Perturbative corrections
- Other methods: MCMC, Langevin, HMC, Adaptive MCMC
Laplace Approximation

\[ \log E_{\theta}(x) = \log \int p_\theta(x | z) p(z) \, dz \]
\[ = \log \int e^{-u(x, z)} \, dz \]
\[ u(x, z) = -\log p_\theta(x | z) p(z) \]
\[ u(x, z) \approx u(x, \mu) + \frac{1}{2}(z - \mu)^T H(\mu)(z - \mu) \]

\[ \log E_{\theta}(x) \approx \log \int e^{-u(x, \mu) - \frac{1}{2}(z - \mu)^T H(\mu)(z - \mu)} \, dz \]
\[ = -u(x, \mu) - \frac{1}{2} \ln \det(2\pi H^{-1}(\mu)) \]

Other names
Saddle-point approximation, Delta-method
Importance Sampling

\[
\log p(x_i) = \log \mathbb{E}_{p(z)}[p_\theta(x_i \mid z)]
\]

\[
= \log \mathbb{E}_{q_\phi(z \mid x_i)} \left[ \frac{p_\theta(x_i \mid z)p(z)}{q_\phi(z \mid x_i)} \right]
\]

\[
= \log \mathbb{E}_{q_\phi(z \mid x_i)} \left[ e^{-\mathcal{F}(x_i, z)} \right]
\]

\[
\approx \log \sum_{k=1}^{K} e^{-\mathcal{F}(x_i, z_k)} - \log K
\]

\[
\mathcal{F}(x, z) = \ln q(z \mid x) - \ln p(z) - \ln p(x \mid z)
\]

\[
\log p(x) \geq \mathbb{E}_{q_\phi(z \mid x_i)} \left[ \log \sum_{k=1}^{K} e^{-\mathcal{F}(x_i, z_k)} \right] - \log K
\]
Importance sampling provides a bound in expectation

\[
\log p(x) \geq \mathbb{E}_{q(z|x)} \left[ \log \sum_{k=1}^{K} e^{-\mathcal{F}(x, z_k)} \right] - \log K
\]
Variational Inference

\[
\log p_\theta(\mathcal{D}) = \sum_{i=1}^{N} \log \mathbb{E}_{p(z)}[p_\theta(x_i | z)]
\]

\[
\log \mathbb{E}_{p(z)}[p_\theta(x_i | z)] = \log \mathbb{E}_{q_i(z)} \left[ \frac{p_\theta(x_i | z) p(z)}{q_i(z)} \right], \quad \forall q_i > 0
\]

\[
\log \mathbb{E}_{q_i(z)} \left[ \frac{p_\theta(x_i | z) p(z)}{q_i(z)} \right] \geq \mathbb{E}_{q_i(z)} \left[ \log \frac{p_\theta(x_i | z) p(z)}{q_i(z)} \right]
\]

\[
\log p_\theta(\mathcal{D}) \geq \sum_{i=1}^{N} \mathbb{E}_{q_i(z)} \left[ \log \frac{p_\theta(x_i | z) p(z)}{q_i(z)} \right]
\]
Variational Inference

\[
\log p_\theta(D) \geq \sum_{i=1}^{N} \mathbb{E}_{q_i(z)} \left[ \log \frac{p_\theta(x_i|z)p(z)}{q_i(z)} \right]
\]

\[
\mathbb{E}_{q_i(z)} \left[ \log \frac{p_\theta(x_i|z)p(z)}{q_i(z)} \right] = \mathbb{E}_{q_i(z)} \left[ \log p_\theta(x_i|z) \right] - \text{KLD}(q_i \parallel p)
\]

Reconstruction

Regularizer
Perturbative Corrections

$$\log \mathbb{E}_{p(z)}[p_\theta(x|z)] = \log \int e^{-u(x,z)} \, dz$$

$$\mathcal{F}(x,z) = \ln q(z|x) + u(x,z)$$

$$\mathcal{F}(x) = \mathbb{E}_{q(z|x)}[\mathcal{F}(x,z)]$$

$$\Delta = -\mathcal{F}(x,z) + \mathcal{F}(x)$$

$$-\mathcal{F}(x) + \log \mathbb{E}_{q(z|x)}[e^{\Delta(x,z)}]$$

$$-\mathcal{F}(x) + \log \mathbb{E}_{q(z|x)} \left[ \sum_{k=0}^{\infty} \frac{\Delta(x,z)^k}{k!} \right]$$

$$= -\mathcal{F}(x) + \log \sum_{k=0}^{\infty} \frac{1}{k!} \mathbb{E}_{q(z|x)}[\Delta(x,z)^k]$$

$$e^y = \sum_{k=0}^{\infty} \frac{y^k}{k!}$$
Design Choices

Choice of Model
Computation graphs, Renderers, simulators and environments

Variational Optimisation
- Variational EM
- Stochastic VEM
- Monte Carlo gradient estimators

Approximate Posteriors
- Mean-field
- Structured approx
- Aux. variable methods
Variational EM Algorithm

Fixed-point iterations between variational and model parameters

\[ E \quad q_i^*(z) = \text{argmax}_{q_i} \mathbb{E}_{q_i^*}(z) \left[ \log \frac{p_\theta(x_i|z)p(z)}{q_i^*(z)} \right] \Leftrightarrow q_i^*(z) = \frac{p_\theta(x_i|z)p(z)}{p(x_i)} \]

\[ M \quad \theta^* = \text{argmax}_\theta \sum_{i=1}^N \mathbb{E}_{q_i^*}(z) \left[ \log \frac{p_\theta(x_i|z)p(z)}{q_i^*(z)} \right] \]
Amortised Inference

\[ q_i^*(z) = \arg\max_{q_i} \mathbb{E}_{q_i^*}(z) [-F(x_i, z)] \]

Introduce a parametric family of conditional densities

\[ \arg\max_{q_i} \mathbb{E}_{q_i^*}(z) [-F(x_i, z)] \Rightarrow \arg\max_{\phi} \mathbb{E}_{q_\phi(z|x)} [-F_\phi(x_i, z)] \]

Rezende et al., 2015
Variational Auto-encoders

Simplest instantiation of a VAE

Deep Latent Gaussian Model $p(x, z)$
- prior sample $z \sim \mathcal{N}(0, I)$
- data sufficient statistics $\eta = f_\theta(z)$
- data conditional likelihood $x \sim \mathcal{N}(\eta)$

Gaussian Recognition Model $q(z)$
- data sample $x \sim \mathcal{D}$
- latent sufficient statistics $\eta = f_\phi(x)$
- posterior sample $z \sim \mathcal{N}(\eta)$

We then optimise the free-energy wrt model and variational parameters

Kingma and Welling, 2014, Rezende et al., 2014
Richer VAES

DRAW: Recurrent/Dependent Priors

Recurrent/Dependent Inference Networks

AIR: Structured Priors

Volumetric and Sequence data

Semi-supervised Learning
Applications of Generative Models

- AI
- Science
- Products

Planning, Exploration
Intrinsic motivation
Model-based RL

Super-resolution,
Compression,
Text-to-speech

Proteomics,
Drug Discovery,
Astronomy,
High-energy physics

Probabilistic Deep Learning

Variational Principles

Amortised Inference

Types of Generative Models

Summary so far
END OF FIRST HALF
Stochastic Optimisation
Classical Inference Approach

\[ \begin{align*}
\mathbb{E} & \quad \int (\ldots) q_{\phi}(z|x) \, dz \\
\n\mathbb{M} & \quad \nabla_{\phi}
\end{align*} \]

Compute expectations then M-step gradients
Stochastic Inference Approach

In general, we won’t know the expectations.
Gradient is of the parameters of the distribution w.r.t. which the expectation is taken.
Stochastic Gradient Estimators

Score-function estimator:
Differentiate the density \( q(z|x) \)

Pathwise gradient estimator:
Differentiate the function \( f(z) \)

Typical problem areas:
- Generative models and inference
- Reinforcement learning and control
- Operations research and inventory control
- Monte Carlo simulation
- Finance and asset pricing
- Sensitivity estimation

Fu, 2006
Score Function Estimators

\[ \nabla_{\phi} \mathbb{E}_{q_{\phi}(z)}[f_{\theta}(z)] = \nabla \int q_{\phi}(z)f_{\theta}(z)dz \]

\[ = \mathbb{E}_{q(z)}[f_{\theta}(z)\nabla_{\phi} \log q_{\phi}(z)] \]

Gradient reweighted by the value of the function

Other names:
- Likelihood-ratio trick
- Radon-Nikodym derivative
- REINFORCE and policy gradients
- Automated inference
- Black-box inference

When to use:
- Function is not differentiable.
- Distribution \( q \) is easy to sample from.
- Density \( q \) is known and differentiable.
Reparameterisation

\[ \nabla \phi \mathbb{E}_{q_\phi(z)}[f_\theta(z)] = \nabla \int q_\phi(z) f_\theta(z) \, dz \]

Find an invertible function \( g(.) \) that expresses \( z \) as a transformation of a base distribution.

\[ z = g_\phi(\epsilon) \quad \epsilon \sim p(\epsilon) \]

\[ \mathbb{E}_{q_\phi(z|x)}[f(z)] = \mathbb{E}_{p(\epsilon)}[f(g_\phi(x, \epsilon))] \]

Kingma and Welling, 2014, Rezende et al., 2014
Pathwise Derivative Estimator

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

\[ \nabla_{\phi} \mathbb{E}_{q_\phi(z)}[f_\theta(z)] = \nabla \int q_\phi(z) f_\theta(z) \, dz \]

\[ = \mathbb{E}_{p(\epsilon)}[\nabla_{\phi} f_\theta(g(\epsilon, \phi))] \]

**Other names:**
- Reparameterisation trick
- Stochastic backpropagation
- Perturbation analysis
- Affine-independent inference
- Doubly stochastic estimation
- Hierarchical non-centred parameterisations.

**When to use**
- Function \( f \) is differentiable
- Density \( q \) can be described using a simpler base distribution: inverse CDF, location-scale transform, or other co-ordinate transform.
- Easy to sample from base distribution.
Gaussian Stochastic Gradients

$$\nabla_{\phi} \mathbb{E}_{\mathcal{N}(\mu, CC^T)}[f_{\theta}(z)]$$

**First-order Gradient**

$$p(\epsilon) = \mathcal{N}(0, 1) \quad g(\epsilon, \phi) = \mu_{\phi}(x) + C_{\phi}(x)\epsilon$$

$$\mathbb{E}_{p(\epsilon)}[J^\top (\nabla_{\phi} \mu_{\phi} + \nabla_{\phi} C_{\phi}^\top \epsilon)]$$

**Second-order Gradient**

$$\mathbb{E}_{q(z)}[J^\top \nabla_{\phi} \mu_{\phi} + \text{Tr}[HC_{\phi} \nabla_{\phi} C_{\phi}]]$$

We can develop low-variance estimators by exploiting knowledge of the distributions involved when we know them

Rezende et al., 2014
Beyond the Mean Field
Mean Field Approximations

Key part of variational inference is choice of approximate posterior distribution $q$.

$$
\mathcal{F}(q, \theta) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \text{KL}[q_\phi(z|x) \| p(z)]
$$
Mean-Field Posterior Approximations

Mean-field or fully-factorised posterior is usually not sufficient
Real-world Posterior Distributions

Deep Latent Gaussian Model

Complex dependencies · Non-Gaussian distributions · Multiple modes
Richer Families of Posteriors

Two high-level goals:
- Build richer approximate posterior distributions.
- Maintain computational efficiency and scalability.

Same as the problem of specifying a model of the data itself.

\[
q^*(z|x) \propto p(x|z)p(z)
\]

\[
q_{MF}(z|x) = \prod_k q(z_k)
\]
Structured Approximations

True Posterior

Structured Approx.

Fully-factorised

\[ q^* (z | x) \propto p(x | z) p(z) \]

\[ q(z) = \prod_k q_k(z_k | \{z_j\}_{j \neq k}) \]

\[ q_{MF} (z | x) = \prod_k q(z_k) \]
Families of Approximate Posteriors

Covariance Models
- \text{Mean-field: } \text{diag}(\alpha_1, \ldots, \alpha_K)
- \text{Rank-1: } \text{diag}(\alpha_1, \ldots, \alpha_K) + uu^T
- \text{Rank-J: } \text{diag}(\alpha_1, \ldots, \alpha_K) + \sum_j u_j u_j^T
- \text{Full: } UU^T

Mixture model
\[ q_{mm}(z; \nu) = \sum_r \rho_r q_r(z_r | \nu_r) \]

Copula Methods
\[ q_{lm}(z; \nu) = \left( \prod_k q_k(z_k | \nu_k) \right) C(z; \nu_{k+1}) \]
Normalising Flows

Exploit the rule for change of variables:
- Begin with an initial distribution
- Apply a sequence of $K$ invertible transforms

\[ z_K = f_K \circ \ldots \circ f_2 \circ f_1(z_0) \]
\[ \log q_K(z_K) = \log q_0(z_0) - \sum_{k=1}^{K} \log \det \left| \frac{\partial f_k}{\partial z_k} \right| \]

Sampling and Entropy

Distribution flows through a sequence of invertible transforms

Rezende and Mohamed, 2015
Normalising Flows

$q_0$

Planar

K=1

K=2

K=10

Unit Gaussian

Uniform
Normalising Flows
Choice of Transformation

\[ \mathcal{L} = \mathbb{E}_{q_0(z_0)}[\log p(x, z_K)] - \mathbb{E}_{q_0(z_0)}[\log q_0(z_0)] - \mathbb{E}_{q_0(z_0)} \left[ \sum_{k=1}^{K} \log \det \frac{\partial f_k}{\partial z_k} \right] \]

Begin with a fully-factorised Gaussian and improve by change of variables.

Triangular Jacobians allow for computational efficiency.

Planar Flow

\[ z_{k+1} = z_k + uh(w^\top z_k + b) \]

Real NVP

\[ y_{1:d} = z_{k-1,1:d} \]
\[ y_{d+1:D} = t(z_{k-1,1:d}) + z_{d+1:D} \odot \exp(s(z_{k-1,1:d})) \]

Inverse AR Flow

\[ z_k = z_{k-1} - \mu_k(z_{<k}, x) \]
\[ \sigma_k(z_{<k}, x) \]

Linear time computation of the determinant and its gradient.

Rezende and Mohamed, 2015; Dinh et al, 2016, Kingma et al, 2016
Normalising Flows on Non-Euclidean Manifolds

\[ \log q_K(z_K) = \log q_0(z_0) - \frac{1}{2} \sum_{k=1}^{K} \log \det |J_\phi^T J_\phi| \]

Gemici et al., 2016
Normalising Flows on non-Euclidean Manifolds
True Posterior

Families of Posterior Approximations

Fully-factorised

Normalising flows

Structured mean-field

Covariance models

Auxiliary variables

Mixtures

Most Expressive

Least Expressive

\[ q^*(z|x) \propto p(x|z)p(z) \]

\[ q_{MF}(z|x) = \prod_k q(z_k) \]
Learning in Implicit Generative Models
Learning by Comparison

For some models, we only have access to an unnormalised probability, partial knowledge of the distribution, or a simulator of data.

We compare the estimated distribution $q(x)$ to the true distribution $p^*(x)$ using samples.
Density Estimation by Comparison

\[ \mathcal{L}(\theta, \phi) \]

**Probability Difference**

\[ r_\phi = p^* - q_\theta \]

- Max Mean Discrepancy
- Moment Matching
- Bregman Divergence

**Probability Ratio**

\[ r_\phi = \frac{p^*}{q_\theta} \]

- Class Probability Estimation
- \( f \)-Divergence

\[ f(u) = u \log u - (u + 1) \log(u + 1) \]

Mohamed and Lakshminarayanan, 2017.
Learning by Comparison

Comparison

Use a hypothesis test or comparison to build an auxiliary model to indicate how data simulated from the model differs from observed data.

Estimation

Adjust model parameters to better match the data distribution using the comparison.
Density Ratios and Classification

\[ \frac{p^*(x)}{q(x)} \]

Bayes’ Rule

\[ p(x|y) = \frac{p(y|x)p(x)}{p(y)} \]

Combine data

\[ \{x_1, \ldots, x_N\} = \{\hat{x}_1, \ldots, \hat{x}_n\} \]

Assign labels

\[ \{y_1, \ldots, y_N\} = \{+1, \ldots, +1, -1, \ldots, -1\} \]

Equivalence

\[ p^*(x) = p(x|y = 1) \quad q(x) = p(x|y = -1) \]

Real Data

Simulated Data

Sugiyama et al., 2012
Density Ratios and Classification

Computing a density ratio is equivalent to class probability estimation.

\[
\frac{p^*(x)}{q(x)} = \frac{p(x | y = 1)}{p(x | y = -1)}
\]

\[
= \frac{p(y = +1 | x)p(x)}{p(y = +1)} / \frac{p(y = -1 | x)p(x)}{p(y = -1)}
\]

\[
\frac{p^*(x)}{q(x)} = \frac{p(y = 1 | x)}{p(y = -1 | x)}
\]
Unsupervised-as-Supervised Learning

**Scoring Function**

\[ p(y = +1|x) = D_\theta(x) \quad p(y = -1|x) = 1 - D_\theta(x) \]

**Bernoulli Loss**

\[ \mathcal{F}(x, \theta, \phi) = \mathbb{E}_{p^*(x)}[\log D_\theta(x)] + \mathbb{E}_{q_\phi(x)}[\log(1 - D_\theta(x))] \]

**Alternating optimisation**

\[ \min_\phi \max_\theta \mathcal{F}(x, \theta, \phi) \]

- Use when we have differentiable simulators and models
- Can form the loss using any proper scoring rule.

**Other names and places:**
- Unsupervised and supervised learning
- Continuously updating inference
- Classifier ABC
- Generative Adversarial Networks

*Friedman et al. 2001*
Generative Adversarial Networks

\[ z \sim p(z) \]
\[ x_{\text{gen}} = f_\phi(z) \]

Comparison loss

\[ \mathcal{F}(x, \theta, \phi) = \mathbb{E}_{p^*(x)}[\log D_\theta(x)] + \mathbb{E}_{q_\phi(x)}[\log(1 - D_\theta(x))] \]

(Alt) Generative loss

\[ \theta \propto \nabla_\theta \mathbb{E}_{p^*(x)}[\log D_\theta(x)] + \nabla_\theta \mathbb{E}_{q_\phi(x)}[\log(1 - D_\theta(x))] \]

\[ \phi \propto -\nabla_\phi \mathbb{E}_{q(z)}[\log D_\theta(f_\phi(z))] \]

Alternating optimisation

\[ \min_\phi \max_\theta \mathcal{F}(x, \theta, \phi) \]

Goodfellow et al. 2014
Integral Probability Metrics

\[ M_f (p, q) = \sup_{f \in \mathcal{F}} |\mathbb{E}_p(x)[f] - \mathbb{E}_{q_\theta}(x)[f]| \]

\( f \) sometimes referred to as a test function, witness function or a critic.

Many choices of \( f \) available: classifiers or functions in specified spaces.

\[ \| f \|_L < 1 \quad \| f \|_\infty < 1 \]

Wasserstein

\[ \| f \|_\mathcal{H} < 1 \quad \left\| \frac{df}{dx} \right\|_L < 1 \]

Max Mean Discrepancy

Cramer
Generative Models and RL
Probabilistic Policy Learning

\[ u(s, a) \sim \text{Environment}(a) \quad p(R(s) | a) \propto \exp(u(s, a)) \]

\[ \mathcal{F}(\theta) = \mathbb{E}_{\pi(a|s)}[R(s, a)] - \text{KL}[\pi_\theta(a|s) \| p(a)] \]

Policy gradient update:
- Uniform prior on actions
- Score-function gradient estimator (aka Reinforce)

\[ \nabla_\theta \mathcal{F}(\theta) = \mathbb{E}_{\pi(a|s)}[(R(s, a) - c) \nabla_\theta \log \pi_\theta(a|s)] + \nabla_\theta \mathbb{H}[\pi_\theta(a|s)] \]

Other algorithms:
- Relative entropy policy search
- Generative adversarial imitation learning
- Reinforced variational inference

Other names and instantiations:
- Planning-as-inference
- Variational MDPs
- Path-integral control
The Future
Applications of Generative Models

Variational Principles

Probabilistic Deep Learning

Types of Generative Models

Rich Distributions

Stochastic Optimisation

Learning by Comparison
Challenges

- Scalability to large images, videos, multiple data modalities.
- Evaluation of generative models.
- Robust conditional models.
- Discrete latent variables.
- Support-coverage in models, mode-collapse.
- Calibration.
- Parameter uncertainty.
- Principles of likelihood-free inference.
References: Applications


References: Applications

- Wu J, Tenenbaum JB, Kohli P. Neural Scene De-rendering., CVPR 2017
References: Fully-observed Models

References: Latent Variable Models

References: Stochastic Gradients

- Pierre L'Ecuyer, Note: On the interchange of derivative and expectation for likelihood ratio derivative estimators, Management Science, 1995

- Peter W Glynn, Likelihood ratio gradient estimation for stochastic systems, Communications of the ACM, 1990

- Michael C Fu, Gradient estimation, Handbooks in operations research and management science, 2006

- Ronald J Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine learning, 1992

- Paul Glasserman, Monte Carlo methods in financial engineering, 2003


- Michael C Fu, Gradient estimation, Handbooks in operations research and management science, 2006


References: Amortised Inference

References: Structured Mean Field

References: Normalising Flows

References: Other Variational Objectives

References: Discrete Latent Variable Models

References: Implicit Generative Models

References: Prob. Reinforcement Learning