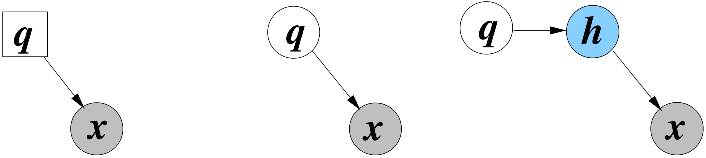
UNIVERSITY OF CAMBRIDGE

4F10: Variational AutoEncoder

Mark Gales

Michaelmas 2018



Gaussian Mixture Model   Factor Analysis   VAE

- Latent variables enables rich models: indicator variable $\boldsymbol{q}$
  - introduce discrete latent variable c_m
  - introduce continuous latent variable $\boldsymbol{z}$
- For discrete case possible to write

$$p(\boldsymbol{x}) = \sum_{m=1}^{M} P(c_m) p(\boldsymbol{x}|c_m)$$

Continuous Latent Variable Models

- Generative latent variable models have form

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$$

  - $\boldsymbol{x}$ is the d-dimensional observation
  - $\boldsymbol{z}$ is the p-dimensional latent variable
- Focus on Gaussian distributions of the form

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{x}; \boldsymbol{f}(\boldsymbol{z}), \sigma^2\mathbf{I}) \\ p(\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \mathbf{I}) \end{aligned}$$

Factor Analysis [6]

- Consider linear relationship

$$\boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{A}\boldsymbol{z}$$

  - this is a restricted form of factor analysis
  - also called probabilistic PCA
- The overall distribution can be written as

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \mathbf{0}, \boldsymbol{A}\boldsymbol{A}^{\mathsf{T}} + \sigma^2\mathbf{I})$$

  - parameters can be estimated using Maximum Likelihood (ML)
  - EM can be used: auxiliary function in this case has the form

$$\begin{aligned} \mathcal{Q}(\boldsymbol{\lambda}, \tilde{\boldsymbol{\lambda}}) &= \int p(\boldsymbol{z}|\boldsymbol{x}; \tilde{\boldsymbol{\lambda}}) \log(p(\boldsymbol{x}|\boldsymbol{z}; \boldsymbol{\lambda}))d\boldsymbol{z} \\ p(\boldsymbol{z}|\boldsymbol{x}) &\propto p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z}) \quad \text{Gaussian distribution} \end{aligned}$$

General Form

- Rather than restricting relationship to be linear
  - consider general mappings from $\boldsymbol{z}$ to the mean
  - use a (deep) neural network with parameters $\boldsymbol{\lambda}$

$$p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{x};\boldsymbol{f}(\boldsymbol{z};\boldsymbol{\lambda}),\sigma^2\mathbf{I})$$

- In general no simple closed-form solutions to integral
  - cannot compute the log-likelihood for inference
  - cannot compute gradient to estimate $\boldsymbol{\lambda}$ and σ^2
- To address this variational approaches are adopted
  - first revision of Kullback-Leibler Divergence
  - then EM and variational EM discussed

Kullback-Leibler Divergence

- Consider two PDFs, $p(\mathbf{x})$ and $q(\mathbf{x})$.
  - Kullback-Leibler divergence, $\mathcal{KL}(p(\mathbf{x})||q(\mathbf{x}))$, is

$$\mathcal{KL}(p(\mathbf{x})||q(\mathbf{x})) = \int p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) d\mathbf{x} = -\int p(\mathbf{x}) \log\left(\frac{q(\mathbf{x})}{p(\mathbf{x})}\right) d\mathbf{x}$$

Using $\log(y) \leq y - 1$, we can write

$$\begin{aligned} \int p(\mathbf{x}) \log\left(\frac{q(\mathbf{x})}{p(\mathbf{x})}\right) d\mathbf{x} &\leq \int p(\mathbf{x}) \left(\frac{q(\mathbf{x})}{p(\mathbf{x})} - 1\right) d\mathbf{x} \\ &= \int (q(\mathbf{x}) - p(\mathbf{x}))\, d\mathbf{x} = 0 \end{aligned}$$

- This gives the following inequality

$$\mathcal{KL}(p(\mathbf{x})||q(\mathbf{x})) = \int p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) d\mathbf{x} \geq 0$$

KL Divergence for Gaussians

- Consider two Gaussian distributions

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1); \quad q(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

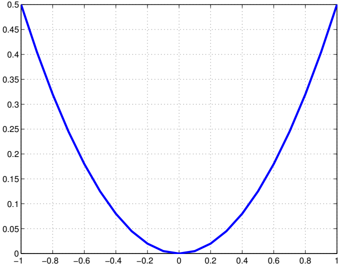
- The KL divergence between the two is given by

$$\mathcal{KL}(p(\boldsymbol{x})||q(\boldsymbol{x})) = \frac{1}{2}\left(\mathrm{tr}(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1 - \boldsymbol{I}) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^{\mathsf{T}}\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \log\left(\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}\right)\right)$$

- Take simple example:

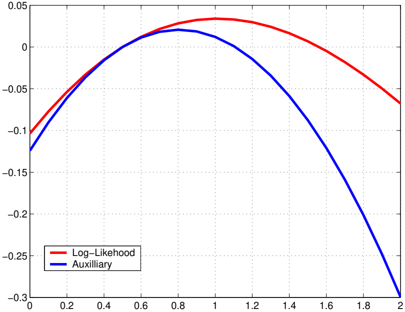
$$\begin{aligned} p(x) &= \mathcal{N}(x; 0, 1); \\ q(x) &= \mathcal{N}(x; \mu, 1) \end{aligned}$$

- Vary μ - plot KL



(Variational) EM

Expectation Maximisation (EM)



- Maximise an auxiliary function rather than log-likelihood
  - iterative optimisation for iteration $k+1$, parameters $\boldsymbol{\lambda}^{(k+1)}$

$$\boldsymbol{\lambda}^{(k+1)} = \arg\max_{\boldsymbol{\lambda}} \left\{ \mathcal{Q}(\boldsymbol{\lambda}^{(k)}, \boldsymbol{\lambda}) \right\} = \arg\max_{\boldsymbol{\lambda}} \left\{ \int p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\lambda}^{(k)}) \log \left(p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\lambda}) \right) d\boldsymbol{z} \right\}$$

Auxiliary Function Optimisation

- Maximising auxiliary function simpler than log-likelihood
  - expected value of log joint distribution, $\log\left(p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\lambda})\right)$
  - consider factor analysis (expectation over $\boldsymbol{z}$)

$$
\begin{aligned}
\log\left(p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\lambda})\right) &= \log\left(p(\boldsymbol{x}|\boldsymbol{z}; \boldsymbol{\lambda})\right) + \log\left(p(\boldsymbol{z})\right) \\
&= \log\left(\mathcal{N}(\boldsymbol{x}; \boldsymbol{A}\boldsymbol{z}, \boldsymbol{\Sigma}_{\texttt{diag}})\right) + \log\left(\mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \mathbf{I})\right) \\
&= -\frac{1}{2}\left(\boldsymbol{z}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}_{\texttt{diag}}^{-1}\mathbf{A}\boldsymbol{z} - 2\boldsymbol{z}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\Sigma}_{\texttt{diag}}^{-1}\boldsymbol{x} + \boldsymbol{x}^{\mathsf{T}}\boldsymbol{\Sigma}_{\texttt{diag}}^{-1}\boldsymbol{x}\right) \\
&\quad -\frac{1}{2}\log\left(|\boldsymbol{\Sigma}_{\texttt{diag}}|\right) + C
\end{aligned}
$$

  - C is a constant not a function of $\boldsymbol{\lambda}$
  - a function of the first and second moments of $\boldsymbol{z}$

Log-Likelihood Expression

- Estimate parameters, $\boldsymbol{\lambda}$, based on log-likelihood, $\mathcal{L}(\boldsymbol{\lambda})$
  - for simplicity only consider a single training observation $\boldsymbol{x}$

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\lambda}) &= \log(p(\boldsymbol{x};\boldsymbol{\lambda})) \\
&= \int p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda}) \log\left(p(\boldsymbol{x};\boldsymbol{\lambda})\right) d\boldsymbol{z} \\
&= \int p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda}) \log\left(\frac{p(\boldsymbol{x};\boldsymbol{\lambda})p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda})}{p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda})}\right) d\boldsymbol{z} \\
&= \left\langle \log\left(\frac{p(\boldsymbol{x},\boldsymbol{z};\boldsymbol{\lambda})}{p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda})}\right)\right\rangle_{p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda})}
\end{aligned}
$$

- Need posterior distribution of the latent variables $p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda})$

Log-Likelihood Expression (cont)

- BUT what happens if not possible to compute $p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda})$
- Consider any valid distribution $q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})$

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\lambda}) &= \log(p(\boldsymbol{x};\boldsymbol{\lambda})) \\
&= \int q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}}) \log\left(p(\boldsymbol{x};\boldsymbol{\lambda})\right) d\boldsymbol{z} \\
&= \int q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}}) \log\left(\frac{p(\boldsymbol{x};\boldsymbol{\lambda})p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda})}{p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda})}\right) d\boldsymbol{z} \\
&= \left\langle \log\left(\frac{p(\boldsymbol{x},\boldsymbol{z};\boldsymbol{\lambda})}{p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda})}\right)\right\rangle_{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})}
\end{aligned}
$$

Variational Approximation

- For any parameter values, e.g. $\tilde{\boldsymbol{\lambda}}$, and distribution $q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})$,

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\lambda}) &= \left\langle \log\left(\frac{p(\boldsymbol{x},\boldsymbol{z};\boldsymbol{\lambda})}{p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda})}\right)\right\rangle_{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})} \\
&= \left\langle \log\left(\frac{p(\boldsymbol{x},\boldsymbol{z};\boldsymbol{\lambda})}{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})}\right)\right\rangle_{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})} + \left\langle \log\left(\frac{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})}{p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda})}\right)\right\rangle_{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})} \\
&\geq \left\langle \log\left(\frac{p(\boldsymbol{x},\boldsymbol{z};\boldsymbol{\lambda})}{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})}\right)\right\rangle_{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})} = \mathcal{F}\left(q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}}),\boldsymbol{\lambda}\right)
\end{aligned}
$$

  - uses KL-divergence to yield a lower-bound
  - equality when $q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}}) = p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda})$

EM revisited

- EM expressed based in this form - at iteration $k+1$
  - initially $q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)}) = p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\lambda}^{(k)})$

$$\mathcal{L}(\boldsymbol{\lambda}^{(k)}) = \mathcal{F}(q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)}), \boldsymbol{\lambda}^{(k)})$$

- Maximise $\mathcal{F}(q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)}), \boldsymbol{\lambda})$ to find parameters $\boldsymbol{\lambda}^{(k+1)}$

$$\boldsymbol{\lambda}^{(k+1)} = \arg\max_{\boldsymbol{\lambda}} \left\{ \left\langle \log\left(\frac{p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\lambda})}{q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)})} \right) \right\rangle_{q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)})} \right\}$$

  - this maximises a lower-bound on $\mathcal{L}(\boldsymbol{\lambda}^{(k)})$
- Minimise KL divergence for “variational” approximation

$$q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k+1)}) = \arg\min_{q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}})} \left\{ \left\langle \log\left(\frac{q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}})}{p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\lambda}^{(k+1)})} \right) \right\rangle_{q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}})} \right\}$$

  - occurs when $q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}) = p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\lambda}^{(k+1)})$

EM (cont)

- Yields the following sequence of inequalities for iteration $k+1$

$$\mathcal{L}(\boldsymbol{\lambda}^{(k)}) = \mathcal{F}\left(q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}}^{(k)}),\boldsymbol{\lambda}^{(k)}\right) \leq \mathcal{F}\left(q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}}^{(k)}),\boldsymbol{\lambda}^{(k+1)}\right) \leq \mathcal{L}(\boldsymbol{\lambda}^{(k+1)})$$

  - provided $q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}}^{(k)}) = p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda}^{(k)})$
- Iterate until convergence:
  - each iteration guaranteed not to decrease the likelihood
  - finds a local maximum of the likelihood
  - final solution depends on initial parameters $\boldsymbol{\lambda}^{(0)}$

General Form for EM [1, 2]

- Start from the same basic equation as before at iteration k

$$\mathcal{L}(\boldsymbol{\lambda}^{(k)}) = \left\langle \log \left(\frac{p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\lambda}^{(k)})}{q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)})} \right) \right\rangle_{q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)})} + \left\langle \log \left(\frac{q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)})}{p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\lambda}^{(k)})} \right) \right\rangle_{q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)})}$$

- General form of $q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)})$
    - second-term is greater than or equal to zero initially
    - yields an initial inequality

$$\mathcal{L}(\boldsymbol{\lambda}^{(k)}) \geq \mathcal{F}\left(q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)}), \boldsymbol{\lambda}^{(k)} \right)$$

- Repeat same process as previous slide
    - estimate $\boldsymbol{\lambda}^{(k+1)}$ using $\mathcal{F}\left(q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)}), \boldsymbol{\lambda} \right)$
    - estimate $q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k+1)})$ by minimising KL-divergence

Variational EM

- Yields the following inequalities

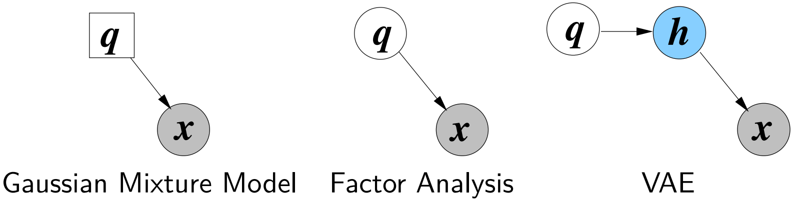
$$\mathcal{L}(\boldsymbol{\lambda}^{(k)}) \geq \mathcal{F}\left(q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)}), \boldsymbol{\lambda}^{(k)}\right) \leq \mathcal{F}\left(q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)}), \boldsymbol{\lambda}^{(k+1)}\right) \leq \mathcal{L}(\boldsymbol{\lambda}^{(k+1)})$$

  - no guarantees of not decreasing log-likelihood
  - but allows any form of expression for $q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}})$

- One standard form is the mean-field approximation where

$$q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}) = \prod_{i=1}^{n} q_i(z_i; \tilde{\boldsymbol{\lambda}})$$

Variational AutoEncoder



- Consider a neural network with parameters $\boldsymbol{\lambda}$

$$p(\boldsymbol{x};\boldsymbol{\lambda}) = \int p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\lambda})p(\boldsymbol{z})d\boldsymbol{z} = \int \mathcal{N}(\boldsymbol{x};\boldsymbol{f}(\boldsymbol{z};\boldsymbol{\lambda}),\sigma^2\mathbf{I})p(\boldsymbol{z})d\boldsymbol{z}$$

  - $p(\boldsymbol{z})$ known - standard Gaussian $\mathcal{N}(\mathbf{0},\mathbf{I})$
- Cannot compute integral - $\boldsymbol{f}(\boldsymbol{z};\boldsymbol{\lambda})$ non-linear

Variational EM Maximisation

- Iterative maximisation involves

$$
\begin{aligned}
\boldsymbol{\lambda}^{(k+1)} &= \arg\max_{\boldsymbol{\lambda}} \left\{ \mathcal{F}(q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)}), \boldsymbol{\lambda}) \right\} \\
&= \arg\max_{\boldsymbol{\lambda}} \left\{ \left\langle \log \left(\frac{p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\lambda})}{q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)})} \right) \right\rangle_{q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}}^{(k)})} \right\}
\end{aligned}
$$

  - need to consider form of $q(\boldsymbol{z}; \tilde{\boldsymbol{\lambda}})$
  - needs to approximate $p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\lambda})$
  - influences how tight the bound is for optimisation

- VEM allows EM to be applied for a wider range of models
  - useful when maximising auxiliary function is easy
  - not (normally) possible for deep learning

(Stochastic) Gradient Descent

- Rather than variational EM, use gradient descent
  - approximate gradient by lower-bound gradient

$$\begin{aligned}\nabla\mathcal{L}(\boldsymbol{\lambda}) &\approx \nabla\left(\left\langle \log\left(\frac{p(\boldsymbol{x},\boldsymbol{z};\boldsymbol{\lambda})}{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})}\right)\right\rangle_{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})}\right)\\ &= \nabla\left(\langle\log(p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\lambda}))\rangle_{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})} + \left\langle \log\left(\frac{p(\boldsymbol{z})}{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})}\right)\right\rangle_{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})}\right)\end{aligned}$$

  - $p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\lambda})$ is Gaussian $\mathcal{N}(\boldsymbol{x};\boldsymbol{f}(\boldsymbol{z};\boldsymbol{\lambda}),\sigma^2\mathbf{I})$
  - $p(\boldsymbol{z})$ is Gaussian (and known)

- Iterative optimisation
  - optimise $\boldsymbol{\lambda}$ (model parameters)
  - optimise $\tilde{\boldsymbol{\lambda}}$ (variational approximation)

Variational Form

- Use a density network as the variational approximation
  - introduce a dependence on $\boldsymbol{x}$ for variational approximation

$$q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}}) \rightarrow q(\boldsymbol{z}|\boldsymbol{x};\tilde{\boldsymbol{\lambda}}) = \mathcal{N}(\boldsymbol{z};\boldsymbol{f}_{\mu}(\boldsymbol{x};\tilde{\boldsymbol{\lambda}}),\boldsymbol{f}_{\Sigma}(\boldsymbol{x};\tilde{\boldsymbol{\lambda}}))$$

  - everything is Gaussian!
- Likelihood can now be written as:

$$\mathcal{L}(\boldsymbol{\lambda}) = \left\langle \log\left(\frac{q(\boldsymbol{z}|\boldsymbol{x};\tilde{\boldsymbol{\lambda}})}{p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\lambda})}\right) + \log\left(p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\lambda})\right) + \log\left(\frac{p(\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x};\tilde{\boldsymbol{\lambda}})}\right)\right\rangle_{q(\boldsymbol{z}|\boldsymbol{x};\tilde{\boldsymbol{\lambda}})}$$

- Consider the three terms (left-to-right)
  1. error: need variational approximation to be close to $\boldsymbol{z}$ posterior
  2. decoding: compute probability given $\boldsymbol{z}$
  3. encoding: encode information about $\boldsymbol{x}$ into $\boldsymbol{z}$

Optimisation

- Gradient lower-bound for optimisation

$$\nabla\left(\left\langle \log\left(p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\lambda})\right) + \log\left(\frac{p(\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x};\tilde{\boldsymbol{\lambda}})}\right)\right\rangle_{q(\boldsymbol{z}|\boldsymbol{x};\tilde{\boldsymbol{\lambda}})}\right)$$

- First term is tricky - remember

$$\langle\log(p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\lambda}))\rangle_{q(\boldsymbol{z}|\boldsymbol{x};\tilde{\boldsymbol{\lambda}})} = \left\langle\log\left(\mathcal{N}(\boldsymbol{x};\boldsymbol{f}(\boldsymbol{z};\boldsymbol{\lambda}),\sigma^2\mathbf{I})\right)\right\rangle_{q(\boldsymbol{z}|\boldsymbol{x};\tilde{\boldsymbol{\lambda}})}$$

  - need to compute (for example) $\langle\boldsymbol{f}(\boldsymbol{z})\rangle_{q(\boldsymbol{z}|\boldsymbol{x};\tilde{\boldsymbol{\lambda}})}$
  - difficult if $\boldsymbol{f}(\boldsymbol{z})$ highly non-linear
- Second term on RHS (-) KL-Divergence between Gaussians
  - simple closed-form solution - see earlier slide

Monte-Carlo Approximation

- Use standard integration approximation

$$
\begin{aligned}
\left\langle \log(p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\lambda}))\right\rangle_{q(\boldsymbol{z}|\boldsymbol{x};\tilde{\boldsymbol{\lambda}})} &\approx \frac{1}{K}\sum_{i=1}^{K} \log(p(\boldsymbol{x}|\boldsymbol{z}^{(i)};\boldsymbol{\lambda})) \\
\boldsymbol{z}^{(i)} &\sim \mathcal{N}(\boldsymbol{f}_{\mu}(\boldsymbol{x};\tilde{\boldsymbol{\lambda}}), \boldsymbol{f}_{\Sigma}(\boldsymbol{x};\tilde{\boldsymbol{\lambda}})) \\
q(\boldsymbol{z}|\boldsymbol{x};\tilde{\boldsymbol{\lambda}}) &= \mathcal{N}(\boldsymbol{z};\boldsymbol{f}_{\mu}(\boldsymbol{x};\tilde{\boldsymbol{\lambda}}), \boldsymbol{f}_{\Sigma}(\boldsymbol{x};\tilde{\boldsymbol{\lambda}}))
\end{aligned}
$$

  - using SGD will approximate gradient using a subset of $\mathcal{D}$
  - can use a single sample of $\boldsymbol{z}$ ($K = 1$)
- This approximation allows $\boldsymbol{\lambda}$ gradients to be estimated
  - but this is using samples from $q(\boldsymbol{z}|\boldsymbol{x};\tilde{\boldsymbol{\lambda}})$
  - not possible to get gradients for $\tilde{\boldsymbol{\lambda}}$

Reparameterisation Trick

- To introduce the dependence on $\tilde{\boldsymbol{\lambda}}$ rewrite

$$\begin{aligned} \boldsymbol{z}^{(i)} &= \boldsymbol{f}_{\mu}(\boldsymbol{x};\tilde{\boldsymbol{\lambda}}) + \boldsymbol{f}_{\Sigma}(\boldsymbol{x};\tilde{\boldsymbol{\lambda}})^{1/2}\boldsymbol{\epsilon}^{(i)} \\ \boldsymbol{\epsilon}^{(i)} &\sim \mathcal{N}(\mathbf{0},\mathbf{I}) \end{aligned}$$

- Now the expression becomes

$$\begin{aligned} \mathcal{L}(\boldsymbol{\lambda}) \geq & \left\langle \log(p(\boldsymbol{x}|\boldsymbol{f}_{\mu}(\boldsymbol{x};\tilde{\boldsymbol{\lambda}}) + \boldsymbol{f}_{\Sigma}(\boldsymbol{x};\tilde{\boldsymbol{\lambda}})^{1/2}\epsilon;\boldsymbol{\lambda}))\right\rangle_{\mathcal{N}(\mathbf{0},\mathbf{I})} \\ & + \left\langle \log\left(\frac{p(\boldsymbol{z})}{q(\boldsymbol{z};\tilde{\boldsymbol{\lambda}})}\right)\right\rangle_{q(\boldsymbol{z}|\boldsymbol{x};\tilde{\boldsymbol{\lambda}})} \end{aligned}$$

  - the first term is a function of $\boldsymbol{\lambda}$ and $\tilde{\boldsymbol{\lambda}}$
  - second term - closed-form solution ((-)KL-divergence) - function of $\tilde{\boldsymbol{\lambda}}$

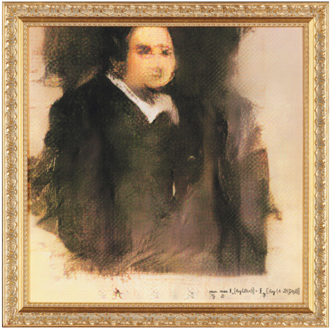
Generated Examples

- "Fictional Celebrities" - Alec Radford (search YouTube)

Conclusions

Conclusions [4, 3, 7]

- Deep generative models are popular at the moment
- Other variants developed, for example
  - (restricted) Boltzmann machines (RBMs)
  - generative adversarial networks (GANs)
  - WaveNet (speech synthesis)
- GAN-generated art ... expensive (©Obvious)



[1] L. Baum and J. Eagon, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology," *Bull Amer Math Soc*, vol. 73, pp. 360–363, 1967.

[2] J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, M. W. (eds, M. J. Beal, and Z. Ghahramani, "The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures," 2003.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[4] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002. [Online]. Available: https://doi.org/10.1162/089976602760128018

[5] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, no. 2014, 2013.

[6] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[7] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw auaio," *CoRR*, 2016. [Online]. Available: https://arxiv.org/pdf/1609.03499.pdf