Variational Inference and Learning II
Latent Variable Models and Variational Autoencoders

Michael U. Gutmann

Probabilistic Modelling and Reasoning (INFR1134)
School of Informatics, The University of Edinburgh

Spring Semester 2024
Assumptions

- Model: $p(v, h; \theta)$
- Data: $\mathcal{D} = \{v_1, \ldots, v_n\}$, $v_i \overset{iid}{\sim} p_*$
- The model is a latent variable model: we have observations for all dimensions of $v$ but no observations of the latents $h$.
- For each observation $v_i$, there is a latent $h_i$.
- Because of iid assumption,

$$p(v_1, \ldots, v_n, h_1, \ldots, h_n; \theta) = \prod_{i=1}^{n} p(v_i, h_i; \theta) \quad (1)$$

- We do not deal with the case of unobserved variables due to missing data, i.e. incomplete observations of $v$. (For VI work on this topic, see e.g. Simkus et al, Variational Gibbs Inference for Statistical Model Estimation from Incomplete Data, https://arxiv.org/abs/2111.13180)
1. Scalable generic variational learning of latent variable models

2. Deep latent variable models and variational autoencoders
1. Scalable generic variational learning of latent variable models
   - ELBO for iid data
   - Amortised variational inference
   - Reparameterisation and stochastic optimisation

2. Deep latent variable models and variational autoencoders
Lower bound on the likelihood for iid data

- We had

\[ \mathcal{L}_x(q) = \mathbb{E}_{q(y|x)} \left[ \log \frac{p(x, y)}{q(y|x)} \right] \tag{2} \]

- Substitute

\[ x \rightarrow (v_1, \ldots, v_n) \quad p(x, y) \rightarrow \prod_{i=1}^{n} p(v_i, h_i; \theta) \tag{3} \]

\[ y \rightarrow (h_1, \ldots, h_n) \tag{4} \]

- Since the true conditional factorises, we use

\[ q(h_1, \ldots, h_n|v_1, \ldots, v_n) = \prod_{i=1}^{n} q(h_i|v_i) \tag{5} \]

- We have one conditional variational distribution \( q(h|v) \).
The ELBO $\mathcal{L}_D$ for iid data $\mathcal{D} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ becomes a sum of per data-point ELBOs $\mathcal{L}_{\mathbf{v}_i}$, denoted by $\mathcal{L}_i$:

$$\mathcal{L}_D(\theta, q) = \sum_{i=1}^{n} \mathcal{L}_i(\theta, q)$$

(6)

$$\mathcal{L}_i(\theta, q) = \mathbb{E}_{q(h_i | \mathbf{v}_i)} \left[ \log \frac{p(\mathbf{v}_i, h_i; \theta)}{q(h_i | \mathbf{v}_i)} \right]$$

(7)

Technical detail: In $\mathcal{L}_i$, we can drop the index $i$ from $h_i$ since it is just the random variable $h \sim q(h | \mathbf{v}_i)$. Hence:

$$\mathcal{L}_i(\theta, q) = \mathbb{E}_{q(h | \mathbf{v}_i)} \left[ \log \frac{p(\mathbf{v}_i, h; \theta)}{q(h | \mathbf{v}_i)} \right]$$

(8)
Lower bound on the likelihood for iid data

- From the basic properties of the ELBO, we have
  \[ \mathcal{L}_i(\theta, q) = \log p(v_i; \theta) - \text{KL}(q(h|v_i)||p(h|v_i; \theta)) \] (9)

- This gives
  \[ \mathcal{L}_D(\theta, q) = \sum_{i=1}^{n} \left[ \log p(v_i; \theta) - \text{KL}(q(h|v_i)||p(h|v_i; \theta)) \right] \] (10)

- With \( \ell(\theta) = \sum_i \log p(v_i; \theta) \) we obtain
  \[ \mathcal{L}_D(\theta, q) = \ell(\theta) - \sum_{i=1}^{n} \text{KL}(q(h|v_i)||p(h|v_i; \theta)) \] (11)

- Maximum likelihood estimation
  \[ \max_{\theta} \ell(\theta) = \max_{\theta, q} \mathcal{L}_D(\theta, q) \] (12)
ELBO maximisation for large sample sizes

▶ For iid data, we have seen a connection between maximum likelihood estimation and minimisation of $\text{KL}(p^*(v)||p(v; \theta))$ if the sample size $n$ is large:

$$\argmax_{\theta} \ell(\theta) \approx \argmin_{\theta} \text{KL}(p^*(v)||p(v; \theta))$$  \hspace{1cm} (13)

▶ A similar result can be shown for $\mathcal{L}_D$:

$$\argmax_{\theta, q} \mathcal{L}_D(\theta, q) \approx \argmin_{\theta, q} \text{KL}(p^*(v)q(h|v)||p(v, h; \theta))$$  \hspace{1cm} (14)

▶ Note: $\theta$ and $q$ enter the KL divergence on different sides: $\theta$ on the right; $q$ on the left.
Potential failure modes

\[
\argmax_{\theta, q} \mathcal{L}_D(\theta, q) \approx \argmin_{\theta, q} \text{KL}(p_*(v)q(h|v)||p(v, h; \theta))
\]

- For fixed \(q\), maximising the ELBO wrt \(\theta\) fits the model \(p(v, h; \theta)\) to augmented data \((v, h)\), with \(v \sim p_*\) and \(h \sim q(h|v)\).
- For fixed \(\theta\), maximising the ELBO wrt \(q\) may lead to mode seeking behaviour.
- By changing \(q\), we change the training data / the target distribution \(p_*(v)q(h|v)\) that we approximate with our model \(p(v, h; \theta)\).
Potential failure modes

\[
\arg\max_{\theta, q} \mathcal{L}_D(\theta, q) \approx \arg\min_{\theta, q} \text{KL}(p_* (v) q(h | v) || p(v, h; \theta))
\]

- An example is the learning of representations in \( h \) space.
- Because of mode-seeking property, \( q(h | v) \) may only cover a small space in \( h \) (for sake of argument, a single mode).
- It thus produces “reduced” training data for \( p(v, h; \theta) \).
- If \( p(v, h; \theta) \) is sufficiently flexible, the KL div can be minimised and we do have \( p_* (v) q(h | v) \approx p(v, h; \hat{\theta}) \) and hence

\[
p_* (v) \approx p(v; \hat{\theta}) = \int p(v, h; \hat{\theta}) dh
\]  \hspace{1cm} (15)

- This means that the marginal \( p(v; \hat{\theta}) \) is meaningful and approximates the distribution of the observed data.
- But the joint \( p(v, h; \hat{\theta}) \) and learned \( q \) may not be meaningful at all since trained with “reduced” \( h \) samples.
For large sample sizes \( n \) we have

\[
\frac{1}{n} \ell(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p(v_i; \theta) \rightarrow \mathbb{E}_{p^\ast(v)} [\log p(v; \theta)] \tag{16}
\]

Similarly

\[
\frac{1}{n} \mathcal{L}_D(\theta, q) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{v_i}(\theta, q) \rightarrow \mathbb{E}_{p^\ast(v)} \mathcal{L}_v(\theta, q) \tag{17}
\]

Dividing Equation (11) by \( n \) thus gives:

\[
\frac{1}{n} \mathcal{L}_D(\theta, q) = \frac{1}{n} \ell(\theta) - \frac{1}{n} \sum_{i=1}^{n} \text{KL}(q(h|v_i)||p(h|v_i; \theta)) \tag{18}
\]

\[
\rightarrow \mathbb{E}_{p^\ast(v)} \mathcal{L}_v(\theta, q) = \mathbb{E}_{p^\ast(v)} [\log p(v; \theta)] - \mathbb{E}_{p^\ast(v)} [\text{KL}(q(h|v)||p(h|v; \theta))] \tag{19}
\]
ELBO max for large sample sizes: proof (not examinable)

\[
\mathbb{E}_{p_*(v)} \mathcal{L}_v(\theta, q) = \mathbb{E}_{p_*(v)} [\log p(v; \theta)] - \mathbb{E}_{p_*(v)} [\text{KL}(q(h|v)||p(h|v; \theta))] \\
= \mathbb{E}_{p_*(v)} [\log p(v; \theta)] - \mathbb{E}_{p_*(v)} \mathbb{E}_q(h|v) \left[ \log \frac{q(h|v)}{p(h|v; \theta)} \right] \\
= -\mathbb{E}_{p_*(v)} \mathbb{E}_q(h|v) \left[ \log \frac{q(h|v)}{p(h|v; \theta)p(v; \theta)} \right]
\]

(20)

Subtract \( \mathbb{E}_{p_*(v)}[\log p_*(v)] \) on both sides:

\[
\mathbb{E}_{p_*(v)} [\mathcal{L}_v(\theta, q) - \log p_*(v)] = -\mathbb{E}_{p_*(v)} \mathbb{E}_q(h|v) \left[ \log \frac{q(h|v)}{p(h|v; \theta)p(v; \theta)} \right] \\
- \mathbb{E}_{p_*(v)} \log p_*(v)
\]

(23)

\[
= -\mathbb{E}_{p_*(v)} \mathbb{E}_q(h|v) \left[ \log \frac{p_*(v)q(h|v)}{p(h|v; \theta)p(v; \theta)} \right]
\]

(24)

\[
= -\text{KL} (p_*(v)q(h|v)||p(h|v; \theta)p(v; \theta))
\]

(25)

\[
= -\text{KL} (p_*(v)q(h|v)||p(h, v; \theta))
\]

(26)

Hence: \( \arg\max_{\theta, q} \mathcal{L}_D(\theta, q) \approx \arg\min_{\theta, q} \text{KL}(p_*(v)q(h|v)||p(v, h; \theta)) \)
Key technical difficulties

- Let us return to the case of finite samples.
- We have to maximise $L_D(\theta, q) = \sum_i L_i(\theta, q)$ with respect to $\theta$ and the conditional $q(h|v)$.
- We had

$$L_i(\theta, q) = \mathbb{E}_{q(h|v_i)} \left[ \log \frac{p(v_i, h; \theta)}{q(h|v_i)} \right] \quad (27)$$

Analytical closed form expression only available in special cases.
- We do not want to restrict the model class but solve the optimisation problem for large $n$ and generic $p(v, h; \theta)$.
- Key technical difficulties are:
  1. Learning of conditional variational distribution $q(h|v)$
  2. Maximisation when the objective involves the $\mathbb{E}_{q(h|v_i)}$
Issue 1: Learning the conditional variational distribution

- Learning the conditional $q(h|v)$ is hard since we have to effectively learn infinitely many pdfs/pmfs (one for each $v$!).
- $\mathcal{L}_i$ only involves $q(h|v_i)$. Hence we could optimise $\mathcal{L}_D$ by optimising each $\mathcal{L}_i$ with respect to $q_i(h) = q(h|v_i)$

$$\max_q \mathcal{L}_D(\theta, q) \Leftrightarrow \max_{q_i} \mathcal{L}_i(\theta, q_i) \quad \text{for} \; i = 1, \ldots, n \quad (28)$$

- We typically make some parametric assumptions. Let $q_i(h)$ be parameterised as $q_i(h; \lambda_i) \in Q_i$.
- Different $q_i(h; \lambda_i)$ may belong to different parametric families.
- Optimisation with respect to $q_i$ then becomes optimisation with respect to $\lambda_i$. 
Issue 1: Learning the conditional variational distribution

- Closed form solution typically not available. This means that we have to iteratively optimise $\mathcal{L}_i$ with respect to $\lambda_i$ for all data points.
- Feasible if $n$ is very small. But too costly otherwise.
Amortisation

- Let us parameterise the conditional distribution $q(h|v)$ directly as
  \[ q(h|v) = q_\phi(h|v) = q(h; \lambda_\phi(v)) \] (29)
  where $\lambda_\phi(v)$ is a nonlinear function parameterised by $\phi$. It is called inference or encoder network, or simply encoder.
- This means that we assume that each $q(h|v_i)$ belongs to the same parametric family $Q = \{q(h; \lambda)\}_\lambda$.
- The function $\lambda_\phi(v)$ maps each $v$ to its corresponding parameter value $\lambda$.
- Note: $\lambda$ are the parameters of the variational distribution while $\phi$ are the parameters of the encoder network.
- Denote $\mathcal{L}_i(\theta, q_\phi)$ by $\mathcal{L}_i(\theta, \phi)$ and $\mathcal{L}_D(\theta, q_\phi)$ by $\mathcal{L}_D(\theta, \phi)$.
- We learn $\phi$ by maximising
  \[ \mathcal{L}_D(\theta, \phi) = \sum_{i=1}^{n} \mathcal{L}_i(\theta, \phi) \] (30)
Amortisation (example)

* A popular choice for $q_\phi(h|v)$ is
  \[
  q_\phi(h|v) = \prod_{k=1}^{H} q_\phi(h_k|v) \quad (31)
  \]
  \[
  q_\phi(h_k|v) = \mathcal{N}(h_k; \mu_k(v; \phi^\mu_k), \sigma^2_k(v; \phi^\sigma_k)) \quad (32)
  \]
  $\phi$ denotes parameters needed to parameterise all mean and var functions.

* Often used for variational autoencoders (see later).

* Makes both an independence and parametric assumption.

* This means that $Q = \{q(h; \lambda)\}_\lambda$ equals the factorised Gaussian family with parameters
  \[
  \lambda = (\mu_1, \ldots, \mu_H, \sigma^2_1, \ldots, \sigma^2_H) \quad (33)
  \]

* The mapping $\lambda_\phi(v)$ maps $v$ to the means and variances,
  \[
  (\mu_1, \ldots, \mu_H, \sigma^2_1, \ldots, \sigma^2_H) = \lambda_\phi(v) \quad (34)
  \]
Amortisation gap

- $\mathcal{L}_D$ is maximised if all individual per data-point $\mathcal{L}_i$ are maximised.
- When learning $\phi$, we hope that after learning

$$q(\mathbf{h}; \lambda_\phi(\mathbf{v}_i)) \approx \arg\max_{q_i \in \mathcal{Q}_i} \mathcal{L}_i(\theta, q_i) \quad \text{for all } i$$ (35)

- The optimisation $\arg\max_{q_i} \mathcal{L}_i$ maps $\mathbf{v}_i$ to the optimal $q_i$, and the idea of amortised inference is to approximate this mapping.
- However, the approximation will not be perfect because
  - $\lambda_\phi(\mathbf{v})$ is learned by maximising the sum $\sum_i \mathcal{L}_i(\theta, \phi)$ and not a single $\mathcal{L}_i(\theta, \phi)$ for a given $\mathbf{v}_i$.
  - We assume that all $q(\mathbf{h}|\mathbf{v}_i)$ belong to the same parametric family, i.e. $\mathcal{Q} = \mathcal{Q}_i$ for all $i$, which may not be the case.
- The approximation will be better for some $\mathbf{v}_i$ than for others.
Amortisation gap

- The approximation error due to amortisation is
  \[ q^*_i(h|v_i) - q(h; \lambda \hat{\phi}(v_i)) = \arg\max_{q_i \in Q} L_i(\theta, q_i) \quad (36) \]

  (If \( Q = Q_i \), we can also compare the amortised with the optimal parameter \( \lambda \))

- Difference between corresponding ELBOs is called the amortisation gap
  \[ L_i(\theta, q^*_i) - L_i(\theta, \hat{\phi}) = \arg\max_{\phi} L_D(\theta, \phi) \quad (37) \]

- After learning, the encoder network \( \lambda \hat{\phi}(v) \) can be applied to test inputs \( v_{\text{test}} \) thereby bypassing an optimisation of the ELBO \( L_{v_{\text{test}}} \).

- The approximation error and amortisation gap will likely be larger for \( v_{\text{test}} \) than for the training data \( v_1, \ldots, v_n \).

For methods to reduce the amortisation gap, see e.g. Marino et al, *Iterative amortised inference*, ICML 2018, https://arxiv.org/abs/1807.09356
Amortisation gap

- Example in two dimensions where $q_i$ is assumed Gaussian with parameters $\lambda = (\mu_1, \mu_2)$.
- The contour plot shows $\mathcal{L}_i(\theta, q_i)$ as a function of $\lambda$.
- The blue line shows the gradient ascent optimisation path when the ELBO is optimised without amortisation.
- The cyan diamond shows the amortised estimate $\hat{\lambda}_\phi(v_i)$.

Figure 1 from Marino et al, ICML 2018.
The optimisation problem is

\[
\hat{\theta}, \hat{\phi} = \arg\max_{\theta, \phi} \mathcal{L}_D(\theta, \phi)
\]  

where

\[
\mathcal{L}_D(\theta, \phi) = \sum_{i=1}^{n} \mathcal{L}_i(\theta, \phi)
\]

\[
= \sum_{i=1}^{n} \mathbb{E}_{q_{\phi}(h|v_i)} \left[ \log \frac{p(v_i, h; \theta)}{q_{\phi}(h|v_i)} \right]
\]

We would like to solve it using gradient ascent.

Difficulties:

1. We generally cannot compute the expectations in closed form.
2. The parameter \( \phi \) occurs in the expectation so that we cannot pull \( \nabla_{\phi} \) inside.
Important special case

▶ For some $q_\phi$, part of the ELBO is available in closed form.
▶ From the basic properties of the ELBO

$$\mathcal{L}_i(\theta, \phi) = \mathbb{E}_{q_\phi(h|v_i)} \left[ \log p(v_i, h; \theta) \right] + \mathcal{H}(q_\phi)$$  \hspace{1cm} (41)

where $\mathcal{H}(q_\phi)$ is the entropy of $q_\phi$.
▶ The entropy can sometimes be computed in closed form.
▶ For factorised Gaussian:

$$\mathcal{H}(q_\phi) = \sum_{k=1}^{H} \frac{1}{2} \left( 1 + \log(2\pi\sigma_k^2(v)) \right)$$  \hspace{1cm} (42)

▶ However, the $\mathbb{E}_{q_\phi(h|v_i)}$ issue remains for the first term.
Reparameterisation

- Consider again the general case.
- We can approximate the expectation as a sample average, but we have to keep track of the $\phi$-dependency of the samples.
- For that, let us consider variational distributions $q_\phi(h|v)$ that can be obtained via a transformation of a random variable $\epsilon$ that we can sample from.

$$ h \sim q_\phi(h|v) \iff h = t_\phi(\epsilon, v), \quad \epsilon \sim p(\epsilon) \quad (43) $$

- Examples:
  - $h \sim \mathcal{N}(h; \mu(v), \sigma^2(v)) \iff h = \mu(v) + \sigma(v)\epsilon$ with $\epsilon \sim \mathcal{N}(\epsilon, 0, 1)$.
  - Inverse transform sampling
  - Factor analysis or ICA model where factor or mixing matrix depends on $v$.
  - ...
Reparameterisation

By the law of the unconscious statistician, we then obtain

\[
\mathbb{E}_{q_\phi(h|v_i)} \left[ \log \frac{p(v_i, h; \theta)}{q_\phi(h|v_i)} \right] = \mathbb{E}_{p(\epsilon)} \left[ \log \frac{p(v_i, t_\phi(\epsilon, v_i); \theta)}{q_\phi(t_\phi(\epsilon, v_i)|v_i)} \right] \tag{44}
\]

We can now pull the gradients inside

\[
\nabla_{\theta, \phi} \mathbb{E}_{q_\phi(h|v_i)} [\cdots] = \nabla_{\theta, \phi} \mathbb{E}_{p(\epsilon)} [\cdots] = \mathbb{E}_{p(\epsilon)} [\nabla_{\theta, \phi} \cdots]
\]

The gradient can then be computed via auto-differentiation.

Note: Alternative to reparameterisation is to use an approach called score function gradient estimation (not examinable).
The gradient of $\mathcal{L}_D(\theta, \phi)$ thus becomes

$$\nabla_{\theta, \phi} \mathcal{L}_D(\theta, \phi) = \sum_{i=1}^{n} \mathbb{E}_{p(\epsilon_i)} \left[ \nabla_{\theta, \phi} \log \frac{p(v_i, t_\phi(\epsilon_i, v_i); \theta)}{q_\phi(t_\phi(\epsilon_i, v_i)|v_i)} \right]$$  \hspace{1cm} (45)$$

We can approximate $\mathbb{E}_{p(\epsilon_i)}$ with a sample average (Monte Carlo integration) with $S$ samples.

For large $n$ and $S$, evaluation of the gradient is expensive.

Computing the gradient for all $v_i$ and using a large $S$ is not necessary. We can use stochastic optimisation instead.

This means we only evaluate the gradient for a random subset (minibatch) of the $v_i$ and set $S$ to a small number (e.g. 1!).

We gloss over technical details here; for an introduction to stochastic optimisation, see *Introduction to Stochastic Search and Optimization* by James Spall.

Eq (45) can be manipulated to reduce the variance of the stochastic gradient, see Roeder et al, *Sticking the Landing: Simple, Lower-Variance Gradient Estimators for Variational Inference*, NeuRIPS 2017.
Program

1. Scalable generic variational learning of latent variable models
   - ELBO for iid data
   - Amortised variational inference
   - Reparameterisation and stochastic optimisation

2. Deep latent variable models and variational autoencoders
1. Scalable generic variational learning of latent variable models

2. Deep latent variable models and variational autoencoders
   - Deep latent variable model
   - Variational autoencoder (VAE)
   - Gaussian and Bernoulli VAE
Deep directed graphical models

- Parametric directed graphical models are sets of pdfs/pmfs that factorise as

\[ p(x; \theta) = \prod_{k=1}^{d} p(x_k | pa_k; \theta) \quad (46) \]

where \( pa_k \) denotes the parents of \( x_k \) in a given directed acyclic graph (DAG).

- We say that the model is a deep directed graphical model if

\[ p(x_k | pa_k; \theta) = p(x_k; \eta_k) \quad \text{with} \quad \eta_k = \eta^k_{\theta}(pa_k) \quad (47) \]

where \( p(x_k; \eta) \) is a parametric model and \( \eta^k_{\theta}(pa_k) \) a parameterised nonlinear function (deep neural network) that maps the parents \( pa_k \) to the model-parameters \( \eta_k \).
Example

- **Chain rule** \( p(x; \theta) = \prod_{k=1}^{d} p(x_k|\text{pre}_k; \theta) \) with
  \[
p(x_k|\text{pre}_k; \theta) = \mathcal{N}(x_k; \mu_k, \sigma_k^2), \quad (\mu_k, \sigma_k^2) = \eta_{\theta}^k(\text{pre}_k)
  \]

- **Markov chain** \( p(x; \theta) = \prod_{k=1}^{d} p(x_k|x_{k-1}; \theta) \) with
  \[
p(x_k|x_{k-1}; \theta) = \mathcal{N}(x_k; \mu_k, \sigma_k^2), \quad (\mu_k, \sigma_k^2) = \eta_{\theta}^k(x_{k-1})
  \]
Deep latent variable model

▶ A deep (directed) latent variable model is a deep directed graphical model with latent variables.
▶ Often (but not always), they are models of the form

\[ p(v, h; \theta) = p(v|h; \theta)p(h) \]  \hspace{1cm} (48)

where \( p(h) \) does not depend on \( \theta \) and \( p(v|h; \theta) \) is

\[ p(v|h; \theta) = \prod_{k=1}^{d} p(v_k|\tilde{pa}_k, h; \theta) \]  \hspace{1cm} (49)

with

\[ p(v_k|\tilde{pa}_k, h; \theta) = p(v_k; \eta_k) \hspace{1cm} \eta_k = \eta^k_{\theta}(\tilde{pa}_k, h) \]  \hspace{1cm} (50)

▶ The latents \( h \) affect the distribution of all the visibles; \( \tilde{pa}_k \) denotes the parents of \( v_k \) restricted to the visibles, i.e. without the \( h \).
▶ Note: parameterised models \( p(h; \theta) \) may also be used.
Graphical model for variational autoencoders

Reconsider the directed acyclic graph for FA and ICA:

The visibles $\mathbf{v} = (v_1, \ldots, v_d)$ are independent from each other given the latents $\mathbf{h} = (h_1, \ldots, h_H)$.

Different assumptions on $p(v_k|\mathbf{h})$ and $p(\mathbf{h})$ give different methods, e.g. FA and ICA.

Working with $H < d$ and $p(v_k|\mathbf{h}; \theta) = p(v_k; \eta_k)$, where $\eta_k = \eta_{\theta}(\mathbf{h})$, gives variational autoencoders (VAE).

The function $\eta_k = \eta_{\theta}^k(\mathbf{h})$ is called the decoder or decoder network.
VAE: overview

- Depending on the data, different parametric families are chosen for the univariate distributions $p(v_k; \eta_k)$
- For example:
  - Gaussian pdf for $v_k \in \mathbb{R}$: Here $\eta_k = (m_k, \nu^2_k)$ are the mean and variance.
  - Bernoulli pmf for $v_k \in \{0, 1\}$: Here $\eta_k = p_k$ is the probability for $v_k = 1$.
- Note: The parametric families may be simple but the parameter $\eta_k$ is a nonlinear transformation of $h$: $\eta_k = \eta_{\theta}(h)$
Example: Gaussian VAE

Nonlinear mean function (NN with random weights and ReLu), constant variance:

Nonlinear mean and variance functions:
The variational distribution \( q_\phi(h|v) \) is often assumed to be a factorised Gaussian.

Variational distribution \( q_\phi(h|v) \) goes under several names: encoder, inference model, or recognition model are used; the model \( p(v|h; \theta) \) is called the decoder or generative model.

Note: the encoder/decoder names may refer to the distribution or the mapping to their parameters.
We now derive the ELBO for the VAE using that:

- \( p(v, h; \theta) = p(v|h; \theta)p(h) \) with \( p(h) = \mathcal{N}(h; 0, I) \)
- Factorised Gaussian for the variational distribution \( q_\phi(h|v) \)

As before:

\[
q_\phi(h|v) = \prod_k q(h_k|v) \tag{51}
\]
\[
q_\phi(h_k|v) = \mathcal{N}(h_k; \mu_k(v), \sigma^2_k(v)) \tag{52}
\]

That is, \( \lambda_\phi(v) \) maps \( v \) to \( (\mu_1, \ldots, \mu_H, \sigma^2_1, \ldots, \sigma^2_H) \).

(\( \phi \)-dependency of \( \mu_k(v), \sigma^2_k(v) \) is suppressed.)

With the Gaussianity assumption on \( p(h) \) and \( q_\phi(h|v) \), part of the ELBO can be computed in closed form.
VAE: learning

► We have seen that if $q_\phi(h|v)$ is a factorised Gaussian

$$
\mathcal{L}_i = \mathbb{E}_{q_\phi(h|v_i)}[\log p(v_i, h; \theta)] + \sum_{k=1}^{H} \frac{1}{2} \left(1 + \log(2\pi\sigma_k^2(v_i))\right)
$$

► Inserting further that $p(v, h; \theta) = p(v|h; \theta)\mathcal{N}(h; 0, I)$, we have

$$
\mathbb{E}_{q_\phi(h|v_i)} \log p(v_i, h; \theta) = \mathbb{E}_{q_\phi(h|v_i)}[\log p(v_i|h; \theta)] + \mathbb{E}_{q_\phi(h|v_i)}[\log \mathcal{N}(h; 0, I)]
$$

► We can compute the second term in closed form

$$
\mathbb{E}_{q_\phi(h|v_i)}[\log \mathcal{N}(h; 0, I)] = -\frac{H}{2} \log(2\pi) - \frac{1}{2} \mathbb{E}_{q_\phi(h|v_i)} \left[\sum_{k=1}^{H} h_k^2\right]
$$

$$
= -\frac{H}{2} \log(2\pi) - \frac{1}{2} \sum_{k=1}^{H} \left[\sigma_k^2(v_i) + \mu_k^2(v_i)\right]
$$
VAE: learning

- Hence

\[ L_i = \mathbb{E}_{q(\mathbf{h}|\mathbf{v}_i)} [\log p(\mathbf{v}_i|\mathbf{h}; \theta)] - \frac{H}{2} \log(2\pi) \]

\[ - \frac{1}{2} \sum_{k=1}^{H} \left[ \sigma_k^2(\mathbf{v}_i) + \mu_k^2(\mathbf{v}_i) \right] + \sum_{k=1}^{H} \frac{1}{2} \left( 1 + \log(2\pi \sigma_k^2(\mathbf{v}_i)) \right) \]

\[ = \mathbb{E}_{q(\mathbf{h}|\mathbf{v}_i)} [\log p(\mathbf{v}_i|\mathbf{h}; \theta)] \]

\[ + \frac{1}{2} \sum_{k=1}^{H} \left( 1 + \log(\sigma_k^2(\mathbf{v}_i)) - \sigma_k^2(\mathbf{v}_i) - \mu_k^2(\mathbf{v}_i) \right) \]

- Same expression can be obtained from

\[ L_i = \mathbb{E}_{q(\mathbf{h}|\mathbf{v}_i)} [\log p(\mathbf{v}_i|\mathbf{h}; \theta)] - \text{KL}(q(\mathbf{h}|\mathbf{v}_i)||\mathcal{N}(\mathbf{h}; 0, I)) \]

and using the closed-form expression for the KL divergence.

- First term: reconstruction/fit; second term: regularisation
With the conditional independence assumption for $p(v_i|h; \theta)$:

\[
\mathbb{E}_{q_\phi(h|v_i)} \left[ \log p(v_i|h; \theta) \right] = \sum_{k=1}^d \mathbb{E}_{q_\phi(h|v_i)} \left[ \log p(v_{ik}; \eta^k_\theta(h)) \right]
\]

where $v_{ik}$ denotes the $k$-th element of $v_i$.

We thus have for the VAE:

\[
\mathcal{L}_i(\theta, \phi) = \sum_{k=1}^d \mathbb{E}_{q_\phi(h|v_i)} \left[ \log p(v_{ik}; \eta^k_\theta(h)) \right] + \\
+ \frac{1}{2} \sum_{k=1}^H \left( 1 + \log(\sigma^2_k(v_i)) - \sigma^2_k(v_i) - \mu^2_k(v_i) \right) \tag{53}
\]

Optimisation problem

\[
\hat{\theta}, \hat{\phi} = \arg\max_{\theta, \phi} \mathcal{L}_D(\theta, \phi) = \arg\max_{\theta, \phi} \sum_{i=1}^n \mathcal{L}_i(\theta, \phi) \tag{54}
\]

Solved with stochastic gradient ascent and the reparam. trick.
Gaussian VAE

- The Gaussian VAE is obtained for
  \[ p(v_k|h; \theta) = \mathcal{N}(v_k; m_k, s^2_k) \quad (m_k, s^2_k) = \eta^k_\theta(h) \quad (55) \]

- Generative model \( p(v|h; \theta) \) equivalent to
  \[
  v = \begin{pmatrix}
  m_1(h) \\
  \vdots \\
  m_D(h)
  \end{pmatrix} + \begin{pmatrix}
  s_1(h) \\
  \vdots \\
  s_D(h)
  \end{pmatrix} n, \quad n \sim \mathcal{N}(n; 0, I)
  \]

- FA obtained for \( m = (m_1, \ldots, m_D)^\top = Fh + c \) and \( s^2_k = \Psi_k \).
- Gaussian VAE is a nonlinear generalisation of FA.
The Bernoulli VAE with $v_k \in \{0, 1\}$ is obtained for

$$p(v_k|h; \theta) = p_k^{v_k} (1 - p_k)^{(1 - v_k)} \quad p_k = \eta_k(h) \quad (56)$$

This is often also used for $v_k \in [0, 1]$. While the ELBO can be evaluated, it is formally wrong since $v_k$ is not binary.

For $v_k \in [0, 1]$, use the so-called continuous Bernoulli distribution or the beta distribution instead.

(see Loaiza-Ganem and Cunningham, *The continuous Bernoulli: fixing a pervasive error in variational autoencoders*, NeuRIPS 2019)
Program recap

1. Scalable generic variational learning of latent variable models
   - ELBO for iid data
   - Amortised variational inference
   - Reparameterisation and stochastic optimisation

2. Deep latent variable models and variational autoencoders
   - Deep latent variable model
   - Variational autoencoder (VAE)
   - Gaussian and Bernoulli VAE