Variational approaches to maximum likelihood

Course of Machine Learning
Master Degree in Computer Science
University of Rome "Tor Vergata"
a.a. 2024-2025

Giorgio Gambosi

1 The case of untreatable $p(\mathbf{z}|\mathbf{x})$ and variational inference

1.1 Situation recap

Let us reassume the situation for what concerns inference in a variable latent model.

We wish to find the value $\boldsymbol{\theta}^*$ which maximizes the likelihood $p(\bar{\mathbf{x}}; \boldsymbol{\theta})$ (or $p(\overline{\mathbf{X}}; \boldsymbol{\theta})$ in the general case) of observed data, that is

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}} \ \log p(\bar{\mathbf{x}}; \boldsymbol{\theta})$$

However, this may be unfeasible, and we may only refer to the joint distribution $p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta})$ while ignoring the value $\bar{\mathbf{z}}$ of the latent variable corresponding to $\bar{\mathbf{x}}$. That is, we cannot maximize instead the joint likelihood

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}} \ \log p(\bar{\mathbf{x}}, \bar{\mathbf{z}}; \boldsymbol{\theta})$$

even marginalizing the latent variable

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}} \int_{\mathcal{Z}} \log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}) d\mathbf{z}$$

is unfeasible due to the multidimensional integral introduced.

Our approach has been then to consider, instead of $\bar{\mathbf{z}}$, the expectation of $\mathbf{z}$ wrt the distribution $p(\mathbf{z}; \hat{\boldsymbol{\theta}})$ of $\mathbf{z}$ as defined for a given (current) value of the parameter $\boldsymbol{\theta}$. That is,

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}} \ \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}; \hat{\boldsymbol{\theta}})} [\, \log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}) \,] = \underset{\boldsymbol{\theta}}{\text{argmax}} \int_{\mathcal{Z}} p(\mathbf{z}; \hat{\boldsymbol{\theta}}) \log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}) d\mathbf{z}$$

or even better, since by hypothesis $\mathbf{x}$ and $\mathbf{z}$ are not independent, the expectation of $p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta})$ wrt to $\mathbf{z} \sim p(\mathbf{z}|\bar{\mathbf{x}}; \hat{\boldsymbol{\theta}})$

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}} \ \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}|\bar{\mathbf{x}}; \hat{\boldsymbol{\theta}})} [\, \log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}) \,] = \underset{\boldsymbol{\theta}}{\text{argmax}} \int_{\mathcal{Z}} p(\mathbf{z}|\bar{\mathbf{x}}; \hat{\boldsymbol{\theta}}) \log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}) d\mathbf{z}$$

In the simpler cases, $p(\mathbf{z}|\mathbf{x}; \boldsymbol{\theta})$ can be computed and also $\mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}|\bar{\mathbf{x}}; \hat{\boldsymbol{\theta}})} [\, \log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}) \,]$ can be optimized, either analytically or by means of gradient methods: this is the case of the base expectation maximization algorithm. If computing $p(\mathbf{z}|\mathbf{x}; \boldsymbol{\theta})$ is unfeasible, a distribution $q(\mathbf{z}|\bar{\mathbf{x}}; \hat{\boldsymbol{\phi}})$ must be derived which approximates sufficiently well $p(\mathbf{z}|\bar{\mathbf{x}}; \hat{\boldsymbol{\theta}})$.

In the case that the maximization of $\mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}|\bar{\mathbf{x}}; \hat{\boldsymbol{\theta}})} [\, \log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}) \,]$ (or of $\mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\bar{\mathbf{x}}; \hat{\boldsymbol{\phi}})} [\, \log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}) \,]$) cannot be performed analytically, a gradient based iterative method can be applied for a certain number of steps to return an approximate maximum: this is called **Generalized EM**.

1.2 Variational inference

The main idea of variational methods is to turn inference into an optimization problem.

Suppose we are given an intractable probability distribution $p(\mathbf{z})$ that we need then to approximate in some way, that is derive a method which makes it possible to compute approximate values of $p(\bar{\mathbf{z}})$ for any $\bar{\mathbf{z}}$. Such an approximation of p could be pursued by applying different approaches:

- numerically, by sampling methods
- analytically, by introducing approximate distributions (the case considered here)

Variational techniques apply the second approach, trying to solve an optimization problem over a class of tractable distributions $\mathcal{F}$ in order to find a $q \in \mathcal{F}$ that is most similar (according to some measure, such as for example KL divergence) to p. We will then refer to q (rather than p) in order to get an approximate solution.

1.3 Mean field theory

There are two main hypotheses that, when applied for choosing the form of distributions in $\mathcal{F}$, make the problem simpler.

1. assuming that $\mathcal{F}$ is a parametric family of distributions with the same functional form, that is $q(\mathbf{z};\omega)$. Here, the restriction turns the problem into a simpler parameter optimization one. It provides a particular framework in dependance of the specified parametric family: for example, if $q(\mathbf{z};\omega)$ is assumed to be multivariate Gaussian, we are dealing with the Gaussian variational inference.
2. assuming that $q(\mathbf{z})$ factorizes on disjoint subsets of components of the latent variable, that is $q(\mathbf{z}) = \prod_{i=1}^{m} q_i(\mathbf{z}_i)$ for some partition $\mathbf{z}_1, \dots, \mathbf{z}_m$ of $\mathbf{z}$. This is known as **mean field variational inference** since it is rooted in a framework of statistical physics called mean field theory. Observe that maximizing the ELBO is still a variational optimization task here, but adding the first hypothesis above results into a parametric framework where $q(\mathbf{z};\omega) = \prod_{j=1}^{m} q_j(\mathbf{z}_j;\omega_j)$.

The variable $\mathbf{z}$ is then governed by its own variational factors, the distributions $q_i(\mathbf{z}_i)$. All the distributions are independent of each other and we have to optimize each of them. The final optimal distribution which approximates the posterior $p(\mathbf{z})$ is the product of all such independent distributions.

2 Variational EM

In the case we consider here, we are dealing with a latent variable model $p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta})$ and we are interested to maximize the log-likelihood $\log p(\bar{\mathbf{x}}; \boldsymbol{\theta})$ wrt $\boldsymbol{\theta}$. In this framework, we introduced an iterative algorithm where each step has the following structures:

1. Given a current value $\boldsymbol{\theta}^{(k)}$ of $\boldsymbol{\theta}$, we compute a good approximation of $p(\mathbf{z}|\bar{\mathbf{x}}; \boldsymbol{\theta}^{(k)}$ consider the ELBO

$$\mathcal{L}(q, \bar{\mathbf{x}}, \boldsymbol{\theta}^{(k)}) = \mathop{\mathbb{E}}_{q(\mathbf{z})}\Big[\log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)})\Big] - \mathop{\mathbb{E}}_{q(\mathbf{z})}[\log q(\mathbf{z})] \leq \log p(\bar{\mathbf{x}}; \boldsymbol{\theta}^{(k)})$$

   as a functional of q and find a distribution $q^{(k)}$ which provides the best possible ELBO wrt the log-likelihood $\log p(\bar{\mathbf{x}}; \boldsymbol{\theta}^{(k)})$, that is the one such that $\mathcal{L}(q^{(k)}, \bar{\mathbf{x}}, \boldsymbol{\theta}^{(k)})$ is maximum, or equivalently $\mathcal{K}(q^{(k)}, \bar{\mathbf{x}}, \boldsymbol{\theta}^{(k)})$ is minimum. As already shown, this is equivalent to finding the best approximation to the conditional distribution $p(\mathbf{z}|\bar{\mathbf{x}}; \boldsymbol{\theta}^{(k)})$.

2. Maximize the expectation

$$Q(q^{(k)}, \bar{\mathbf{x}}, \boldsymbol{\theta}) = \mathop{\mathbb{E}}_{q^{(k)}(\mathbf{z})}[\log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta})]$$

   with respect to $\boldsymbol{\theta}$.

The distribution of interest here is then conditional distribution of the latent variable given the observations $p(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\theta})$ which has to be approximated in the first step and which is strictly related to the likelihood of the observations since $p(\mathbf{x};\boldsymbol{\theta}) = \dfrac{p(\mathbf{x},\mathbf{z};\boldsymbol{\theta})}{p(\mathbf{z}|\mathbf{x};\boldsymbol{\theta})}$.

According to the variational approach sketched above, the basic idea here is defining a family $\mathcal{F}$ of tractable distributions over the latent variable $\mathbf{z}$. Each $q(\mathbf{z}) \in \mathcal{F}$ is a candidate approximation to the posterior. Our goal is to find the best candidate which maximizes the ELBO $\mathcal{L}(q,\bar{\mathbf{x}},\hat{\boldsymbol{\theta}})$, that is the distribution $q^*(\mathbf{z}) \in \mathcal{F}$ such that $\mathcal{L}(q^*,\bar{\mathbf{x}},\hat{\boldsymbol{\theta}})$ is as close as possible to the log-likelihood of $\bar{\mathbf{x}}$ given $\hat{\boldsymbol{\theta}}$.

In the description of the EM algorithm provided above, the k-th E-step returns the distribution $q^{(k)}(\mathbf{z})$ maximizing $\mathcal{L}(q,\bar{\mathbf{x}},\boldsymbol{\theta}^{(k)})$, and thus minimizing $D_{KL}\left(q(\mathbf{z})||p(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\theta}^{(k)})\right)$, which would be obtained by setting

$$q^{(k)}(\mathbf{z}) = p(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\theta}^{(k)})$$

This is not possible, however, in the case that the posterior distribution $p(\mathbf{z}|\mathbf{x};\boldsymbol{\theta})$ is untreatable. In this case, we have to maximize $\mathcal{L}(q(\mathbf{z}),\bar{\mathbf{x}},\boldsymbol{\theta}^{(k)}))$ by varying $q(\mathbf{z})$ only within some predefined family $\mathcal{F}$ of tractable functions and, as a consequence, obtaining $q^{(k)} \in \mathcal{F}$

That is, the E-step is defined as

$$q^{(k)}(\mathbf{z}) = \underset{q\in\mathcal{F}}{\operatorname{argmax}}\ \mathcal{L}(q,\bar{\mathbf{x}},\boldsymbol{\theta}^{(k)})$$

Since, in most cases, $p(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\theta}^{(k)})$ is not included in the family $\mathcal{F}$ of distributions, then

$$D_{KL}\left(q^{(k)}(\mathbf{z})||p(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\theta}^{(k)})\right) > 0$$

and

$$\log p(\bar{\mathbf{x}};\boldsymbol{\theta}^{(k)}) > \mathcal{L}(q^{(k)},\bar{\mathbf{x}},\boldsymbol{\theta}^{(k)})$$

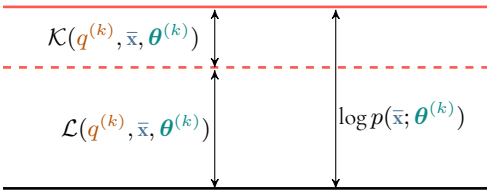

Figure 1: After E-step

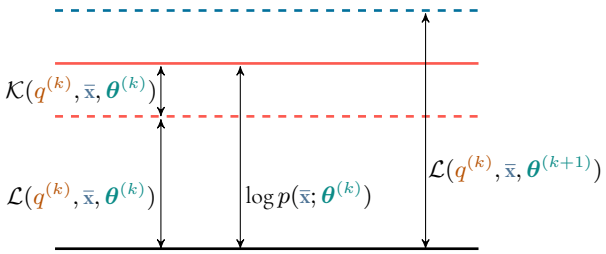

Figure 2: After M-step

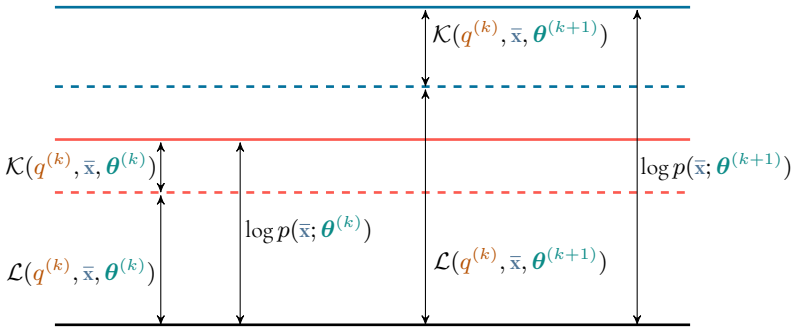

Figure 3: New log-likelihood decomposition

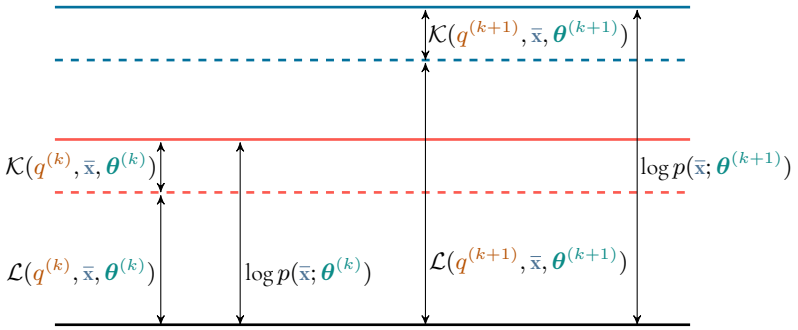

Figure 4: After new E-step

For what regards the M-step of the algorithm, we maximize the lower bound

$$Q(q^{(k)}, \bar{\mathrm{x}}, \boldsymbol{\theta}) = \mathop{\mathbb{E}}_{q^{(k)}(\mathrm{z})} [\, p(\bar{\mathrm{x}}, \mathrm{z}; \boldsymbol{\theta}) \,]$$

with respect to $\boldsymbol{\theta}$.

By the same arguments seen in the case of EM, we have the situation reported in figures 1-4

2.1 Mean field theory and variational EM

Among all distributions $q(\mathbf{z})$ having the mean field form, we now seek that distribution for which the ELBO $\mathcal{L}(q, \mathbf{x}, \hat{\boldsymbol{\theta}})$ is largest. This is done by optimizing with respect to each of the factors in turn. To achieve this, let us define

$$\mathbf{z}_{-j} \stackrel{\Delta}{=} (\mathbf{z}_1, \ldots, \mathbf{z}_{j-1}, \mathbf{z}_{j+1}, \ldots, \mathbf{z}_m)$$

and

$$q_{-i}(\mathbf{z}_{-i}) \stackrel{\Delta}{=} \prod_{j \neq i} q_j(\mathbf{z}_j)$$

let us also denote as $\mathcal{Z}$ the domain of $\mathbf{z}$, as $\mathcal{Z}_i$ the domain of $\mathbf{z}_i$ and as $\mathcal{Z}_{-i}$ the domain of $\mathbf{z}_{-i}$.

It is possible to prove that, for any factor $\mathbf{z}_j$:

$$\begin{aligned}\mathcal{L}(q, \bar{\mathbf{x}}, \boldsymbol{\theta}^{(k)}) &= \mathop{\mathbb{E}}_{q(\mathbf{z})}\left[\log \frac{p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)})}{q(\mathbf{z})}\right] \\ &= \cdots \\ &= \int_{\mathcal{Z}_j} q_j(\mathbf{z}_j)\left(\int_{\mathcal{Z}_{-j}} q_{-j}(\mathbf{z}_{-j}) \log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)}) d\mathbf{z}_{-j}\right) d\mathbf{z}_j - \int_{\mathcal{Z}_j} q_j(\mathbf{z}_j) \log q_j(\mathbf{z}_j) d\mathbf{z}_j + f_j\end{aligned}$$

where f_j is a constant with respect to $q_j(\mathbf{z}_j)$

The term

$$\int_{\mathcal{Z}_{-j}} q_{-j}(\mathbf{z}_{-j}) \log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)}) d\mathbf{z}_{-j}$$

is the expectation of the log likelihood of the complete dataset $\log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)})$ wrt all factors $\mathbf{z}_s$ except $\mathbf{z}_j$, each distributed according to the current distribution q_s. That is,

$$\int_{\mathcal{Z}_{-j}} q_{-j}(\mathbf{z}_{-j}) \log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)}) d\mathbf{z}_{-j} = \mathop{\mathbb{E}}_{q_{-j}(\mathbf{z}_{-j})}\left[\log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)})\right]$$

Clearly, being all factors except $\mathbf{z}_j$ marginalized out in the integral, the term is a function of $\mathbf{z}_j$.

Let us now define the distribution $\tilde{p}(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}^{(k)})$ as

$$\tilde{p}(\bar{\mathbf{x}}, \mathbf{z}_j; \boldsymbol{\theta}^{(k)}) \triangleq \frac{e^{\mathbb{E}_{q_{-j}(\mathbf{z}_{-j})}\left[\log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)})\right]}}{\int_{\mathcal{Z}_j} e^{\mathbb{E}_{q_{-j}(\mathbf{z}_{-j})}\left[\log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)})\right]} d\mathbf{z}_j}$$

As a consequence, we have

$$\int_{\mathcal{Z}_{-j}} q_{-j}(\mathbf{z}_{-j}) \log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)}) d\mathbf{z}_{-j} = \log \tilde{p}(\bar{\mathbf{x}}, \mathbf{z}_j; \boldsymbol{\theta}^{(k)}) + C$$

where $C = \log \int_{\mathcal{Z}_j} e^{\mathbb{E}_{q_{-j}(\mathbf{z}_{-j})}\left[\log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)})\right]} d\mathbf{z}_j$

We wish to maximize the ELBO $\mathcal{L}(q, \bar{\mathbf{x}}, \boldsymbol{\theta}^{(k)}) = \mathcal{L}(q_1, \ldots, q_m, \bar{\mathbf{x}}, \boldsymbol{\theta}^{(k)})$ wrt q_j, hence we consider it as a functional of q_j, obtaining

$$\begin{aligned}\mathcal{L}(q_j, \bar{\mathbf{x}}, \boldsymbol{\theta}^{(k)}) &= \int_{\mathcal{Z}_j} q_j(\mathbf{z}_j)\left(\log \tilde{p}(\bar{\mathbf{x}}, \mathbf{z}_j; \boldsymbol{\theta}^{(k)}) + C\right) d\mathbf{z}_j - \int_{\mathcal{Z}_j} q_j(\mathbf{z}_j) \log q_j(\mathbf{z}_j) d\mathbf{z}_j + f_j \\ &\cdots \\ &= -D_{KL}\left(q_j(\mathbf{z}_j) || \tilde{p}(\bar{\mathbf{x}}, \mathbf{z}_j; \boldsymbol{\theta}^{(k)})\right) + f_j + C\end{aligned}$$

$\mathcal{L}(q, \bar{\mathbf{x}}, \boldsymbol{\theta}^{(k)})$ is then maximized, with respect to q_j, when $D_{KL}\left(q_j(\mathbf{z}_j) || \tilde{p}(\bar{\mathbf{x}}, \mathbf{z}_j; \boldsymbol{\theta}^{(k)})\right)$ is minimal and in particular when

$$D_{KL}\left(q_j(\mathbf{z}_j) || \tilde{p}(\bar{\mathbf{x}}, \mathbf{z}_j; \boldsymbol{\theta}^{(k)}\right) = 0$$

that is

$$q_j(\mathbf{z}) = \tilde{p}(\bar{\mathbf{x}}, \mathbf{z}_j; \boldsymbol{\theta}^{(k)}) = \frac{e^{\mathbb{E}_{q_{-j}(\mathbf{z}_{-j})}\left[\log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)})\right]}}{\int_{\mathcal{Z}_j} e^{\mathbb{E}_{q_{-j}(\mathbf{z})_{-j}}\left[\log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)})\right]} d\mathbf{z}_j} \tag{1}$$

Algorithm 1: CAVI

Input: A model $p(\mathbf{x}, \mathbf{z})$, an observation $\bar{\mathbf{x}}$
Output: A variational density $q(\mathbf{z}) = \prod_{j=1}^{m} q_j(\mathbf{z}_j)$
Initialize: Variational factors $q_j(\mathbf{z}_j)$
while *the ELBO has not converged* **do**
  for $j \in \{1, \dots, m\}$ **do**
    Set $q_j(\mathbf{z}) \propto \exp\{\mathbb{E}_{q_{-j}(\mathbf{z}_j)}\left[\log p(\mathbf{z}_j|\mathbf{z}_{-j}, \bar{\mathbf{x}}; \hat{\boldsymbol{\theta}})\right]\}$
  end
  Compute $\text{ELBO}(q) = \mathbb{E}_{q(\mathbf{z})}\left[\log p(\bar{\mathbf{x}}, \mathbf{z}; \hat{\boldsymbol{\theta}})\right] - \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})]$
end
return $q(\mathbf{z})$

Observe now that since

$$\log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)}) = \log p(\mathbf{z}_j|\mathbf{z}_{-j}, \bar{\mathbf{x}}; \boldsymbol{\theta}^{(k)}) + \log p(\mathbf{z}_{-j}, \bar{\mathbf{x}}; \boldsymbol{\theta}^{(k)})$$

it results

$$e^{\mathbb{E}_{q_{-j}(\mathbf{z}_{-j})}\left[\log p(\bar{\mathbf{x}}, \mathbf{z}; \boldsymbol{\theta}^{(k)})\right]} = e^{\mathbb{E}_{q_{-j}(\mathbf{z}_{-j})}\left[\log p(\mathbf{z}_j|\mathbf{z}_{-j}, \bar{\mathbf{x}}; \boldsymbol{\theta}^{(k)})\right]} e^{\mathbb{E}_{q_{-j}(\mathbf{z}_{-j})}\left[\log p(\mathbf{z}_{-j}, \bar{\mathbf{x}}; \boldsymbol{\theta}^{(k)})\right]}$$

and

$$\tilde{p}(\bar{\mathbf{x}}, \mathbf{z}_j; \boldsymbol{\theta}^{(k)}) = \frac{e^{\mathbb{E}_{q_{-j}(\mathbf{z}_{-j})}\left[\log p(\mathbf{z}_j|\mathbf{z}_{-j}, \bar{\mathbf{x}}; \boldsymbol{\theta}^{(k)})\right]}}{\int_{\mathcal{Z}_j} e^{\mathbb{E}_{q_{-j}(\mathbf{z}_{-j})}\left[\log p(\mathbf{z}_j|\mathbf{z}_{-j}, \bar{\mathbf{x}}; \boldsymbol{\theta}^{(k)})\right]} d\mathbf{z}_j}$$

the condition

$$D_{KL}\left(q_j(\mathbf{z}_j)||\tilde{p}(\bar{\mathbf{x}}, \mathbf{z}_j; \boldsymbol{\theta}^{(k)}\right) = 0$$

is verified also when

$$q_j(\mathbf{z}) = \frac{e^{\mathbb{E}_{q_{-j}(\mathbf{z}_{-j})}\left[\log p(\mathbf{z}_j|\mathbf{z}_{-j}, \bar{\mathbf{x}}; \boldsymbol{\theta}^{(k)})\right]}}{\int_{\mathcal{Z}_j} e^{\mathbb{E}_{q_{-j}(\mathbf{z}_{-j})}\left[\log p(\mathbf{z}_j|\mathbf{z}_{-j}, \bar{\mathbf{x}}; \boldsymbol{\theta}^{(k)})\right]} d\mathbf{z}_j} \tag{2}$$

This observations lead to the definition of the Coordinate ascent variational inference (CAVI), one of the most commonly used algorithms for variational inference in mean field models, which iteratively optimizes each factor q_j while others are fixed.

The algorithm is applied at each E-step, and iterates through factors, at each step updating the current factor $q_j(\mathbf{z}_j)$ using Equation (2). CAVI then goes uphill on the ELBO, towards a local optimum wrt to $q(\mathbf{z})$ of the ELBO for the given observation $\bar{\mathbf{x}}$ with parameter value $\boldsymbol{\theta}^{(k)}$, assuming the set of distributions which factorize as $q(\mathbf{z}) = \prod_{i=1}^{m} q_j(\mathbf{z}_j)$ is considered.

Variational Autoencoder

As noticed, computing a different $q_i(\mathbf{z}_i) = q(\mathbf{z}_i; \boldsymbol{\phi}_i)$ for each item $\bar{\mathbf{x}}_i$ in the dataset can be costly, since it requires computing a different parameter value for each item in a usually large dataset. Amortization consider instead a single conditional distribution $q(\mathbf{z}|\bar{\mathbf{x}}; \boldsymbol{\phi})$, with a single parameter value to be inferred: we explicitly apply this approach here.

Let us first observe that since

$$\log p(\bar{\mathbf{x}};\boldsymbol{\theta}) = \int_{\mathcal{Z}} q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi}) \log \frac{p(\bar{\mathbf{x}},\mathbf{z};\boldsymbol{\theta})}{q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})} d\mathbf{z} - \int_{\mathcal{Z}} q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi}) \log \frac{p(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\theta})}{q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})} d\mathbf{z}$$

then

$$\begin{aligned}
\log p(\bar{\mathbf{x}};\boldsymbol{\theta}) + \int_{\mathcal{Z}} q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi}) \log \frac{p(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\theta})}{q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})} d\mathbf{z} &= \int_{\mathcal{Z}} q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi}) \log \frac{p(\bar{\mathbf{x}},\mathbf{z};\boldsymbol{\theta})}{q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})} d\mathbf{z} \\
&= \int_{\mathcal{Z}} q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi}) \log \frac{p(\bar{\mathbf{x}}|\mathbf{z};\boldsymbol{\theta}) p(\mathbf{z};\boldsymbol{\theta})}{q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})} d\mathbf{z} \\
&= \int_{\mathcal{Z}} q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi}) \log p(\bar{\mathbf{x}}|\mathbf{z};\boldsymbol{\theta}) d\mathbf{z} + \int_{\mathcal{Z}} q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi}) \log \frac{p(\mathbf{z};\boldsymbol{\theta})}{q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})} d\mathbf{z}
\end{aligned}$$

which results into

$$\log p(\bar{\mathbf{x}};\boldsymbol{\theta}) - D_{KL}\left(q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi}) || p(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\theta})\right) = \mathop{\mathbb{E}}_{\mathbf{z}\sim q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})} \left[\log p(\bar{\mathbf{x}}|\mathbf{z};\boldsymbol{\theta})\right] - D_{KL}\left(q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi}) || p(\mathbf{z};\boldsymbol{\theta})\right)$$

Observe that we wish to maximize the log likelihood $\log p(\bar{\mathbf{x}};\boldsymbol{\theta})$ while approximating $p(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\theta})$ with $q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})$ as good as possible, that is minimizing their divergence, hence maximizing $-D_{KL}\left(q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi}) || p(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\theta})\right)$. In summary, we wish to find $\boldsymbol{\theta}^*$ and $\boldsymbol{\phi}^*$ which maximize the left hand side of the equation above.

Equivalently, we may then aim to maximize the right hand side

$$\mathop{\mathbb{E}}_{\mathbf{z}\sim q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})} \left[\log p(\bar{\mathbf{x}}|\mathbf{z};\boldsymbol{\theta})\right] - D_{KL}\left(q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi}) || p(\mathbf{z};\boldsymbol{\theta})\right)$$

with respect to both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$.

In order to compute the first term, we may consider the approximation of the expectation provided by the average of m values $\hat{\mathbf{z}}_1 \ldots, \hat{\mathbf{z}}_m$ sampled from $q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})$. That is,

$$\mathop{\mathbb{E}}_{\mathbf{z}\sim q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})} \left[\log p(\bar{\mathbf{x}}|\mathbf{z};\boldsymbol{\theta})\right] \approx \frac{1}{m} \sum_{i=1}^{m} \log p(\bar{\mathbf{x}}|\hat{\mathbf{z}}_i;\boldsymbol{\theta}) \qquad \text{where} \quad \hat{\mathbf{z}}_i \sim q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi}), i = 1, \ldots, m$$

In particular, we may consider the case $m = 1$, when only one value $\hat{\mathbf{z}}$ is sampled, thus obtaining a rough approximation of the original expectation

$$\mathop{\mathbb{E}}_{\mathbf{z}\sim q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})} \left[\log p(\bar{\mathbf{x}}|\mathbf{z};\boldsymbol{\theta})\right] \approx \log p(\bar{\mathbf{x}}|\hat{\mathbf{z}};\boldsymbol{\theta}) \qquad \text{where} \quad \hat{\mathbf{z}} \sim q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})$$

which results in the following function to be maximized

$$\log p(\bar{\mathbf{x}}|\hat{\mathbf{z}};\boldsymbol{\theta}) - D_{KL}\left(q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi}) || p(\mathbf{z};\boldsymbol{\theta})\right)$$

where $\hat{\mathbf{z}}$ is assumed sampled from $q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})$.

The first term requires that, given an observed item $\bar{\mathbf{x}}$, its probability is high even when conditioned by the latent value $\hat{\mathbf{z}}$ that the model associates to it (at least probabilistically). That is, we wish to have values $\boldsymbol{\phi}^*, \boldsymbol{\theta}^*$ that make it likely to associate (through $\boldsymbol{\phi}^*$) a latent variable value $\hat{\mathbf{z}}$ to $\bar{\mathbf{x}}$, that makes the conditional probability (also through $\boldsymbol{\theta}^*$) of $\bar{\mathbf{x}}$ as high as possible.

The second term requires that distributions $q(\mathbf{z}|\bar{\mathbf{x}};\boldsymbol{\phi})$ are not very different each other for different observed values $\bar{\mathbf{x}}$ and in particular that all of them are as similar as possible to a prior distribution $p(\mathbf{z};\boldsymbol{\theta})$, independent from $\bar{\mathbf{x}}$, whose parameters are often considered as constant, that is of type $p(\mathbf{z})$. This provides a regularizing effect, since it combats the tendency of the model, during inference, to excessively adapt to observed data.

The whole situation at inference time can be seen as an encoding-decoding process, where:

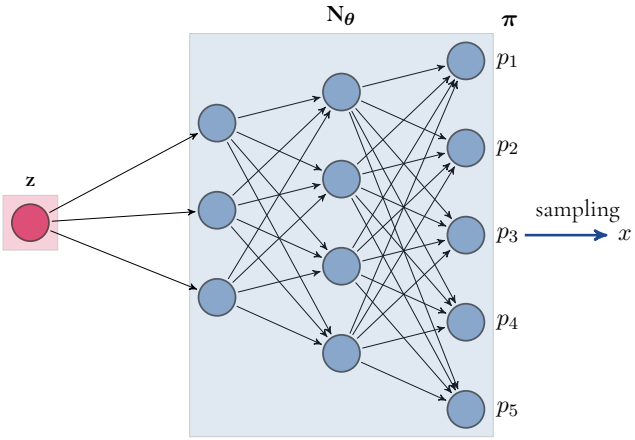

Figure 5: Categorical DLVM with $x \in \{1, \ldots, 5\}$

1. given the observation $\bar{\mathbf{x}}$, a latent variable value $\hat{\mathbf{z}}$ is produced by sampling from $q(\mathbf{z}|\bar{\mathbf{x}}; \boldsymbol{\phi})$: we call this **encoding** $\bar{\mathbf{x}}$ as $\hat{\mathbf{z}}$

2. given the latent value $\hat{\mathbf{z}}$, a value $\hat{\mathbf{x}}$ in the observations space is produced by sampling from $p(\bar{\mathbf{x}}|\hat{\mathbf{z}}; \boldsymbol{\theta})$: we call this **decoding** $\hat{\mathbf{z}}$ as $\hat{\mathbf{x}}$

We wish that $\boldsymbol{\phi}, \boldsymbol{\theta}$ make the probability that $\hat{\mathbf{x}}$ is equal to $\bar{\mathbf{x}}$ as high as possible, while also having simple, not too specialized, distributions $q(\mathbf{z}|\bar{\mathbf{x}}; \boldsymbol{\phi})$. Observed that this second requirement, by resulting in $p(\mathbf{z})$ similar to all $q(\mathbf{z}|\bar{\mathbf{x}}; \boldsymbol{\phi}^*)$, makes it possible to safely produce new values in observations space which are not statistically distinguishable from the ones available and used at inference time, by simply first sampling $\hat{\mathbf{z}}$ from $p(\mathbf{z})$ and then, as before, $\hat{\mathbf{x}}$ from $p(\bar{\mathbf{x}}|\hat{\mathbf{z}}; \boldsymbol{\theta})$.

Deep latent variable models

In a DLVM we assume that the distributions involved in the model are of a given type, with parameters computed by predefined parametric functions, computed by (deep) neural networks.

For example, if the space of observations is defined on a discrete set of possible values, $p(x|\mathbf{z})$ could be a Categorical distribution, with the value of the corresponding parameters (i.e. the posterior probabilities of each value) is obtained by applying a given function $\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{z})$, parametric on $\boldsymbol{\theta}$, to $\hat{\mathbf{z}}$. According to the approach, the function $\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{z})$ is assumed computed by a deep neural network $\mathfrak{D}_{\boldsymbol{\theta}}$.

More in detail, let $p(x|\mathbf{z}) = \mathcal{C}(x; \boldsymbol{\pi})$, where $\mathcal{C}(x; \boldsymbol{\pi})$ is a categorical distribution with vector $\boldsymbol{\pi}$ of probability values. We assume $\boldsymbol{\pi} = N_{\boldsymbol{\theta}}(\mathbf{z})$ where $N_{\boldsymbol{\theta}}$ is a neural network with parameters $\boldsymbol{\theta}$ (Figure 5).

Moreover, if we assume that $p(\mathbf{z})$ is also given, for example a standard 3-variate gaussian, we may generate new values for $\mathbf{x}$ by (Figure 6):

1. sampling $\hat{\mathbf{z}}$ from $p(\mathbf{z})$

2. sampling $\hat{x}$ from $p(x|\hat{\mathbf{z}}) = \mathcal{C}(\mathbf{x}; N_{\boldsymbol{\theta}}(\hat{\mathbf{z}}))$

The same considerations can be done for the approximate conditional distribution $q(\mathbf{z}|\mathbf{x}; \boldsymbol{\phi})$, that could be assumed to be of a given type according to the definition of the latent space. For example, if the latent space is $\mathbb{R}^k$, $q(\mathbf{z}|\mathbf{x})$ could be a k Gaussian distribution, with the value of the corresponding parameters (i.e. expectation vector and covariance matrix) obtained by applying given functions $\mathcal{M}_{\boldsymbol{\phi}}(\cdot)$ and $\mathcal{V}_{\boldsymbol{\phi}}(\cdot)$, parametric on $\boldsymbol{\phi}$, to $\bar{\mathbf{x}}$. According to the approach, the functions are assumed computed by a deep neural network $\mathfrak{E}_{\boldsymbol{\phi}}$.

This results into the situation given in Figure 7, denoted as a **variational autoencoder** at inference time. Observed that the covariance matrix of $q(\mathbf{z}|\mathbf{x}; \boldsymbol{\phi}) = \mathcal{N}(\mathbf{z}; \mathcal{M}_{\boldsymbol{\phi}}(\mathbf{x}), \mathcal{V}_{\boldsymbol{\phi}}(\mathbf{x}))$ is assumed diagonal.

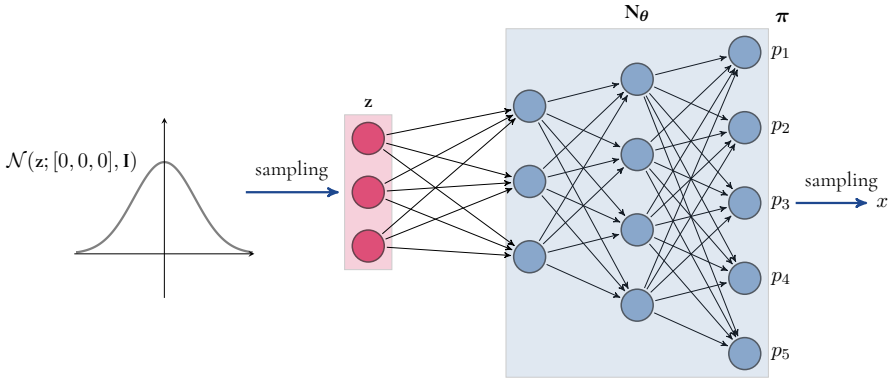

Figure 6: Sampling new values in a Gaussian-categorical DLVM with $\mathbf{z} \in \mathbb{R}^3$ and $x \in \{1, \ldots, 5\}$

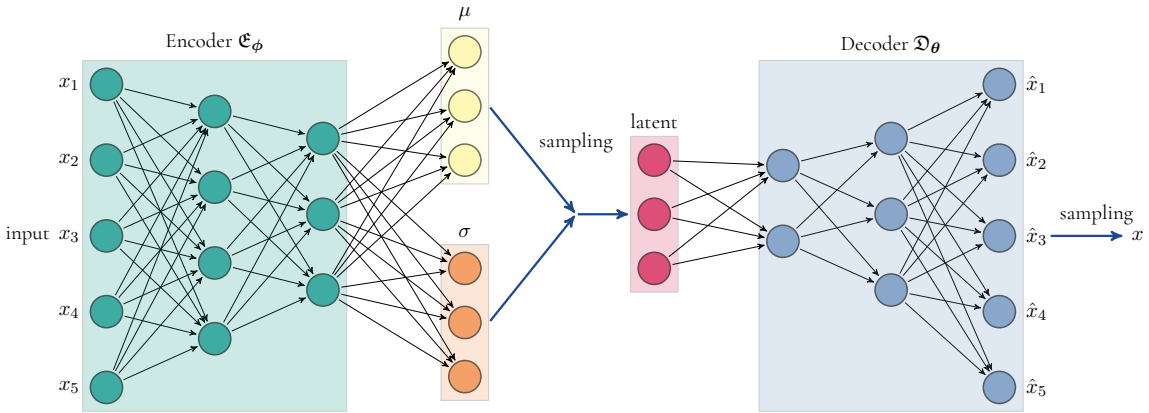

Figure 7: Variational encoder as DLVM

Autoencoders

The variational encoder introduced above can be seen as a probabilistic version of a neural network architecture named autoencoder, which is mainly designed to encode the input into a compressed and meaningful representation, and then decode it back such that the reconstructed input is similar as possible to the original one. Their main purpose is learning in an unsupervised manner an "informative" representation of the data that can be used for various implications such as clustering.

Formally, an $n/p/n$ autoencoder can be defined in terms of

- a class of encoder functions $\mathcal{E}$, from $\mathbb{R}^n$ to $\mathbb{R}^p$
- a class of decoder functions $\mathcal{D}$, from $\mathbb{R}^p$ to $\mathbb{R}^n$
- a reconstruction loss function $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$

The problem is finding a pair of encoder and decoder functions $e^* \in \mathcal{E}$, $d^* \in \mathcal{D}$ such that

$$(e^*, d^*) = \underset{e \in \mathcal{E}, d \in \mathcal{D}}{\operatorname{argmin}} \; E_{\mathbf{x}}[\mathcal{L}(\mathbf{x}, d(e(\mathbf{x})))] \tag{3}$$

that is, we are interested in finding the encode-decoder pair which minimizes the (expected) difference between the input $\mathbf{x}$ and the result of the sequence of encoding and decoding $d(e(\mathbf{x}))$. The reconstruction loss is usually set to be the ℓ_2-norm, that is $\mathcal{L}(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||^2 = \sum_{i=1}^{n} (x_i - y_i)^2$.

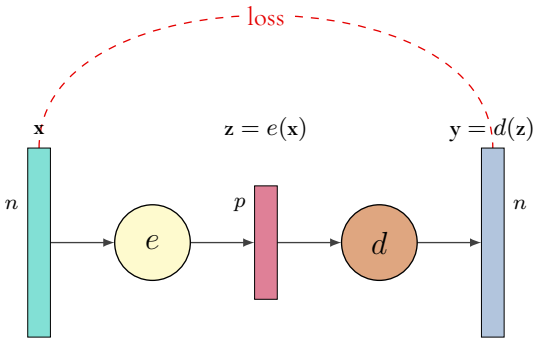

Figure 8: General structure of an autoencoder.

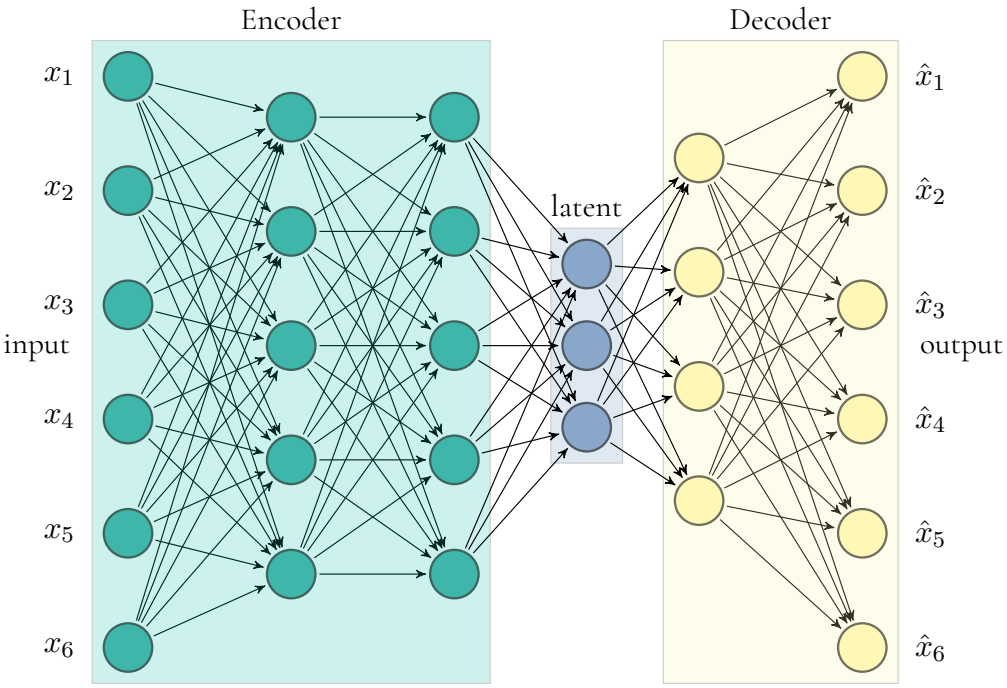

Figure 9: Neural network implementation of an autoencoder.

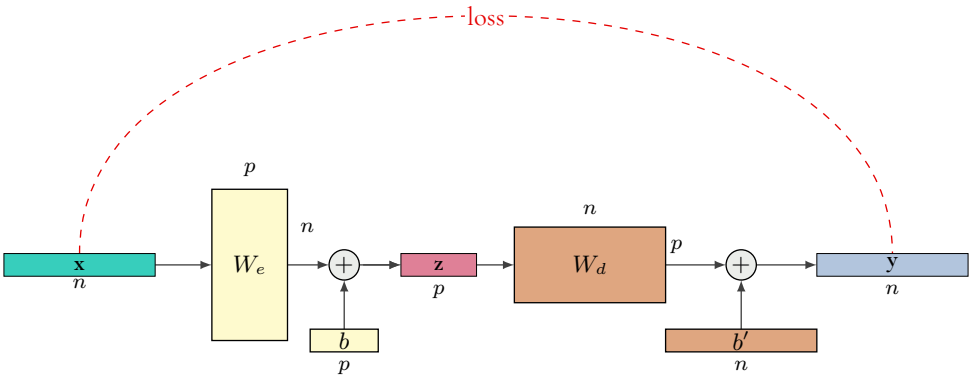

Figure 10: Structure of a linear autoencoder.

Figure 8 provides an illustration of the autoencoder model.

As we may see, starting from $\mathbf{x}$, the encoder derives a new vector $\mathbf{z}$ of a different (usually smaller) size: this is called a **latent representation** of $\mathbf{x}$ In the most popular form of autoencoders, e and d are implemented by neural networks.

In the special case that $\mathcal{E}$ and $\mathcal{D}$ are classes of linear operations, we get a linear autoencoder: in this case, each component of $\mathbf{z}$ is a linear combination of components of $\mathbf{x}$, and each component of $\mathbf{y}$ is a linear combination of components of $\mathbf{z}$. In the case of a linear autoencoder the same latent representation as Principal Component Analysis (PCA) is achieved. Therefore, an autoencoder is in fact a generalization of PCA, where instead of finding a low dimensional hyperplane in which the data lies, it is able to learn a non-linear manifold. In addition, the autoencoder is explicitly optimized for the data reconstruction from the code. A good intermediate representation not only can capture latent variables, but also benefits a full decompression process.

As usual in Machine Learning, the set of functions to be considered are defined as classes of parametric functions with parameters $\boldsymbol{\theta}$ for the encoder and $\boldsymbol{\phi}$ for the decoder. The loss function is then minimized wrt the encoder and the decoder parameters:

$$(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*) = \underset{\boldsymbol{\theta},\boldsymbol{\phi}}{\operatorname{argmin}}\, E_{\mathbf{x}}[\mathcal{L}(\mathbf{x}, d(e(\mathbf{x};\boldsymbol{\theta}), \boldsymbol{\phi}))]$$

The parametric functions are usually implemented as neural networks.

Since in training, one may just get the identity operator for e and d, which keeps the achieved representation the same as the input, some additional regularization is required. The most common option is to make the dimension of the representation smaller than the input. This way, a **bottleneck** is imposed. This option also directly serves the goal of getting a low dimensional representation of the data. This representation can be used for purposes such as data compression, feature extraction, etc. Its important to note that even if the bottleneck is comprised of only one node, then overfitting is still possible if the capacity of the encoder and the decoder is large enough to encode each sample to an index.

In cases where the size of the hidden layer is equal or greater than the size of the input, there is a risk that the encoder will simply learn the identity function. To prevent it without creating a bottleneck (i.e. smaller hidden layer) several options exists for regularization that would enforce the autoencoder to learn a different representation of the input.

Inference in VAE

Inference in a VAE is made as usual for all neural networks, by applying backpropagation, that is computing the gradient of the loss function wrt all parameters (arc weights) in the backward direction, starting from the final

layer towards the input layer. Due to the complexity of the loss function, gradients are computed numerically, i.e. without deriving an analytic form of the gradient, by applying **automatic differentiation**, a technique which makes it possible to compute gradient values starting from a procedural (that is algorithmic) description of the loss function.

Applying backpropagation is also a critical issue due to the presence of a sampling step in the computation pipeline, which does not allow to backpropagate values. This problem is tackled by expressing the sampling operation in a more suitable way, where sampling is a side, unparameterized, operation.

Reparameterization Trick

The expectation term in the loss function invokes generating samples from $\mathbf{z} \sim q(\mathbf{z}|\bar{\mathbf{x}}; \boldsymbol{\phi})$. Sampling is a stochastic process and therefore we cannot backpropagate the gradient to update $\boldsymbol{\phi}$. To make it trainable, the reparameterization trick is introduced.

The trick consists in modifying the network to perform the stochastic choice of ε wrt a parameterless distribution, such as $\mathcal{N}(\cdot; \mathbf{0}, \mathbf{I})$, and then applying a parameterized function to the value obtained.

It is in fact often possible to express the random variable $\mathbf{z}$ as produced by the application of a deterministic parametric function $\mathbf{z} = \mathcal{T}_{\boldsymbol{\phi}}(\mathbf{x}, \varepsilon)$, where ε is an auxiliary independent random variable, and the transformation function $\mathcal{T}_{\boldsymbol{\phi}}$ parameterized by $\boldsymbol{\phi}$ converts ε to $\mathbf{z}$. This makes the stochastic choice a constant component in the computation performed by the network, which is then not involved into backpropagation.

For example, in the common case where $q(\mathbf{z}|\bar{\mathbf{x}}; \boldsymbol{\phi})$ is a multivariate Gaussian with a diagonal covariance structure with variances in vector $\boldsymbol{\sigma}^2$

$$\hat{\mathbf{z}} \sim q(\mathbf{z}|\bar{\mathbf{x}}; \boldsymbol{\phi}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \mathbf{I})$$

the trick computes $\hat{\mathbf{z}}$ as

$$\hat{\mathbf{z}} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \varepsilon \quad \varepsilon \sim \mathcal{N}(0, \mathbf{I})$$

where $\odot$ is the element-wise product.

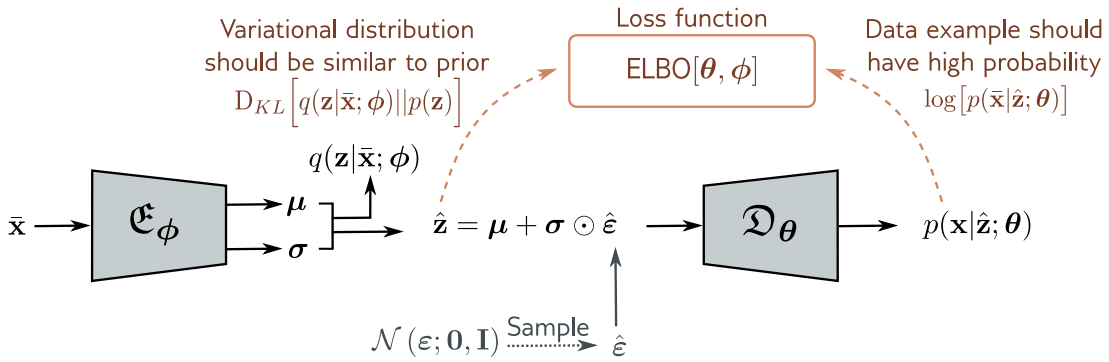

Figure 11: Reparametrization trick makes sampling a side unparameterized step with respect to the main computation pipeline