# Variational Auto Encoders

## 1 Introduction

$\ast$ Def. **1.0.1 Statistical Inference:** Determining properties of the underlying probability distribution based on observed data. This could include estimating parameters, latent variables, or making predictions. In Bayesian inference, the primary goal is to compute the posterior $p(z|x)$:

$$p(z|x) = \frac{p(x,z)}{p(x)} = \frac{p(x|z)p(z)}{\int_{\mathcal{Z}} p(x,z)\,dz}$$

Since the marginal likelihood $p(x)$ is intractable, we instead approximate the posterior $p(z|x)$ with a simpler, tractable distribution $q(z|\lambda)$, and we wish to minimize the KL divergence $KL\Big(q(z|\lambda)\,||\,p(z|x)\Big)$, or equivalently maximize the Evidence Lower Bound (ELBO):

$\ast$ Def. **1.0.2 Density Estimation:** Approximating PDF $p(x)$ of a random variable $X$ based on observed data $\{x_1,\ldots,x_n\}$ drawn i.i.d from the underlying distribution. Can be either parametric, e.g.

$$p(x|\theta) \sim \mathcal{N}(\mu_x, \sigma_x)$$

or nonparametric density estimation, for example Kernel Density Estimation:

$$p(x) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)$$

$\ast$ Def. **1.0.3 Modeling:** Modeling is unveiling the underlying ruling processes, by posing hypotheses and predictions, based on observations. For instance, physicists model how fluids flow, and biologists the structure of organisms.

Modeling often involves representation, where we describe a phenomena using specific qualities and quantities related to the process we are interested in. We describe an object by its shape, color, position, volume, etc. When looking at data - a large collection of samples - in some cases, it is reasonable to believe that those representations follow some distributions. For example, human height clearly follow some probability distribution. In such cases, we can think of the samples as being "generated" from those distributions. When such a hypothesis is true, we can generate new samples in the population, provided we estimated its probabilities. Alternatively, we may say that we have uncertainty about those variables, and we specify the degree and nature of this uncertainty in terms of probability distributions.

**Remark ()**: A complete probabilistic model captures both the distributions of its components and the relations/dependencies between them. Usually the linear relations, given by the covariances.

$\ast$ Def. **1.0.4 Probabilistic Model:** Assume the observed variables $\boldsymbol{x}$ are a random sample from an unknown underlying process, whose true probability distribution is $p^*(\boldsymbol{x})$. We approximate this underlying process with a chosen probabilistic model $p_{\boldsymbol{\theta}}(\boldsymbol{x})$, with parameters $\boldsymbol{\theta}$:

$$\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Learning is the process of searching for a value of the parameters $\theta$, such that the probability function given by the mode $p_{\boldsymbol{\theta}}(\boldsymbol{x})$, approximates the true distribution of the data, denoted by $p^{^*}(\boldsymbol{x})$, such that for any observed $\boldsymbol{x}$:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) \approx p^{\star}(\boldsymbol{x})$$

$\ast$ Def. **1.0.5 Conditional Models:** Often, we are not interested in learning an unconditional model $p_{\boldsymbol{\theta}}(\boldsymbol{x})$, but a conditional model $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$, that approximates the underlying conditional distribution $p^*(\boldsymbol{y}|\boldsymbol{x})$: A distribution over the values of variable $\boldsymbol{y}$, conditioned on the value of an observed variable $\boldsymbol{x}$. $\boldsymbol{x}$ is often called the *input* of the model.

A common example is image classification, where $\boldsymbol{x}$ is the image, and $\boldsymbol{y}$ is the label, and $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ is chosen to be the categorical distribution, whose parameters are computed from $\boldsymbol{x}$.

**Remark ()**: We can use neural networks to parameterize a distribution. For example for the categorical distribution Categorical$(y;\boldsymbol{p})$ over a class label $y$, we have:

$$\boldsymbol{p} = \mathrm{NeuralNet}(\boldsymbol{x})$$
$$p_{\boldsymbol{\theta}}(y|\boldsymbol{x}) = \mathrm{Categorical}(y;\boldsymbol{p})$$

## 1.1 Directed PGMs and NNs

We work with *directed* PGMs, where all variables are organized into a directed acyclic graph, and the joint distribution over the variables factorizes as a product of prior and conditional probabilities:

$$p_\theta(\boldsymbol{x}_1, \ldots \boldsymbol{x}_m) = \sum_{j=1}^{m} p_\theta(\boldsymbol{x}_j | Pa(\boldsymbol{x}_j))$$

where $Pa(\boldsymbol{x}_j)$ is the set of parent variables of node $j$ in the directed graph.

Traditionally, each conditional probability distribution $p_\theta(\boldsymbol{x}_j|Pa(\boldsymbol{x}_j))$ is parameterized by a lookup table or a linear model. A more flexible way is to parametrize such conditional distributions using a neural network. In this case, the NN takes as input the parents of a variable, and produce the distributional parameters $\boldsymbol{\eta}$ over that variable:

$$\boldsymbol{\eta} = \text{NeuralNet}(Pa(\boldsymbol{x}))$$
$$p_\theta(\boldsymbol{x}|Pa(\boldsymbol{x})) = p_\theta(\boldsymbol{x}|\boldsymbol{\eta})$$

for example, in VAE, when learning $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, note that $Pa(\boldsymbol{x}) = \boldsymbol{z}$, and the recognition model learn the distributional parameters $\boldsymbol{\eta} = (\mu, \sigma)$ to parametrize $p_\theta(\boldsymbol{x}|\boldsymbol{\eta})$

## 1.2 Learning in Fully Observed Models with Neural Nets

If all variables in the directed model are observed in the data, then we can compute and differentiate the log-probability of the data under the model, leading to a relatively straightforward optimization.

Dataset: We collect a dataset $\mathcal{D}$ consisting of $N$ datapoints:

$$\mathcal{D} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\}$$

The datapoints are assumed to be independent samples from an unchanging underlying distribution, i.e. i.i.d. Then the probability of the datapoints given the parameters factorizes as a product of individual datapoint probabilities:

$$\log p_\theta(\mathcal{D}) = \sum_{\boldsymbol{x} \in \mathcal{D}} \log p_\theta(\boldsymbol{x}) \qquad \text{(Maximum Likelihood)}$$

A common criterion for probabilistic models is maximum log-likelihood (MLL). We attempt to find the parameters $\theta$ that maximize the above sum, or equivalently, the average, of the log-probabilities assigned to the data by the model. Since our model is limited by its expressiveness, and our data is noisy, this can not be trivially solved.

Using stochastic gradient descent, we draw a minibatch $\mathcal{M}$ of the data, and since:

$$\nabla_\theta \log p_\theta(\mathcal{D}) \simeq \nabla_\theta \log p_\theta(\mathcal{M}) = \sum_{\boldsymbol{x} \in \mathcal{M}} \nabla_\theta \log p_\theta(\boldsymbol{x})$$

where $\simeq$ indicates an unbiased estimator, we can use $\sum_{\boldsymbol{x} \in \mathcal{M}} \nabla_\theta \log p_\theta(\boldsymbol{x})$ to iteratively update $\theta$ and hill-climb to a local optimum of the Maximum Likelihood.

## 1.3 Learning and Inference in Deep Latent Variable Models

$\ast$ Def. **1.3.1 Latent Variables:** Variables which are part of the model, but which we do not observe.

For an observed variable $\boldsymbol{x}$ and an unobserved variable $\boldsymbol{z}$, the joint distribution is denoted $p_\theta(\boldsymbol{x}, \boldsymbol{z})$ over both variables. The marginal distribution over the observed variables $p_\theta(\boldsymbol{x})$ is given by:

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}, \boldsymbol{z})\, d\boldsymbol{z} \tag{1}$$

This is also called the *evidence*.

$\ast$ Def. **1.3.2 Deep Latent Variable Model (DLVM):** latent variable model $p_\theta(\boldsymbol{x}, \boldsymbol{z})$ whose distributions are parameterized by neural nets. Then, even when each factor (prior or conditional distribution) in the directed model is relatively simple (such as a conditional Gaussian), the marginal distribution $p_\theta(\boldsymbol{x})$ can be very complex. This expressivity makes DLVM attractive for approximating complicated underlying distributions $p^\star(\boldsymbol{x})$.

**@ Example:** A simple DLVM is:

$$p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p_\theta(\boldsymbol{z}) p_\theta(\boldsymbol{x}|\boldsymbol{z})$$

we call $p_\theta(\boldsymbol{z})$ the *prior distribution* over $\boldsymbol{z}$.

For example, assume, for binary data $\boldsymbol{x}$ (such as 0-1 MNIST images), a spherical Gaussian latent space, and a factorized Bernoulli observation model. Then:

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; 0, I)$$
$$\boldsymbol{p} = \text{DecoderNeuralNet}_\theta(\boldsymbol{z})$$
$$\log p(\boldsymbol{x}|\boldsymbol{z}) = \sum_{j=1}^{D} \log p(x_j|\boldsymbol{z}) = \sum_{j=1}^{D} \log \text{Bernoulli}(x_j; p_j)$$

where $D$ is the dimensionality of $\boldsymbol{x}$.

**Remark ()**: As $\boldsymbol{z}$ is not observed, computing Eq. (1) is intractable, and hence we cannot directly optimize it.

For a Fully observed model, like $p_\theta(\boldsymbol{x}) \sim \mathcal{N}(\mu, \sigma^2)$, we can easily compute $\log p_\theta(\boldsymbol{x})$, and optimize $\theta$ using the gradients of Maximum Likelihood.

For latent variable models, we have:

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}, \boldsymbol{z})\, d\boldsymbol{z}$$

The integral over $\boldsymbol{z}$ often lacks a closed-form solution, due to the complexity of taking its gradient, and the dimensionality of $\boldsymbol{z}$.

Note - we often do know $p_\theta(\boldsymbol{x}, \boldsymbol{z})$, the joint distribution, since we have $p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p_\theta(\boldsymbol{x}|z) p_\theta(\boldsymbol{z})$, and we often assume $p_\theta(\boldsymbol{z})$, and $p_\theta(\boldsymbol{x}|z)$, the *likelihood*

The intractability of $p_\theta(\boldsymbol{x})$ is related to the intractability of the *posterior* distribution $p_\theta(\boldsymbol{z}|\boldsymbol{x})$, via the identity:

$$p_\theta(\boldsymbol{z}|\boldsymbol{x}) = \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{p_\theta(\boldsymbol{x})}$$

since $p_\theta(\boldsymbol{x}, \boldsymbol{z})$ is tractable to compute. We approximate $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ using variational inference, by a similar distribution $q\theta(\boldsymbol{z}|\boldsymbol{x})$, called the **variational distribution**.

## 2 Variational Auto Encoders

**Remark (**Motivation**)**: Suppose we have observed data $\mathbf{x}$, for instance, pixels of dog images. These data points are drawn from some true but unknown distribution $p^*(\mathbf{x})$. Our goal is to approximate $p^*(\boldsymbol{x})$ using a parameterized family $p_\theta(\mathbf{x})$, enabling tasks like generation (sampling new images) or downstream inference (classification, etc.).

Directly modeling $p_\theta(\mathbf{x})$ can be extremely challenging, especially for complex, high-dimensional data. One common approach is to introduce latent variables $\mathbf{z}$ that describe underlying factors or structures, and then express the data distribution through marginalization:

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x} \mid \mathbf{z}) p_\theta(\mathbf{z})\, d\mathbf{z}.$$

Here, $p_\theta(\mathbf{x} \mid \mathbf{z})$ is often simpler to model than $p_\theta(\mathbf{x})$ itself. For example, $\mathbf{z}$ might represent attributes like a dog's breed, pose, or lighting, making it easier to build a model that generates images $\mathbf{x}$ given $\mathbf{z}$. We also choose a prior $p_\theta(\mathbf{z})$ that is tractable, such as a Gaussian.

The challenge is that computing $p_\theta(\mathbf{x})$ requires integrating over $\mathbf{z}$, which is typically intractable for complex models. This makes it hard to directly optimize parameters $\theta$ by maximum likelihood.

**Role of the Posterior and Variational Inference**: Remember our goal is to find parameters $\theta$ that will optimize $\log p_\theta(\boldsymbol{x})$. Using mathematical tricks, we use posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ - in fact an approximation $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ - and by choosing $\phi$ such that $q_\phi(\boldsymbol{z}|\boldsymbol{z})$ is close to $p_\theta(\boldsymbol{z})$, we increase $\log p_\theta(\boldsymbol{x})$.

Specifically, note that:

$$\log p_\theta(\boldsymbol{x}) = \log \int p_\theta(\boldsymbol{x}|\boldsymbol{z}) p_\theta(\boldsymbol{z})\, d\boldsymbol{z}$$

We Multiply and divide by $q_\phi(\boldsymbol{z}|\boldsymbol{x})$:

$$\log p_\theta(\boldsymbol{x}) = \log \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \, d\boldsymbol{z}$$

Recall Jensen inequality for concave function states that $f(E[X]) \geq E[f(X)]$, and since log is concave, we get that $\log E[X] \geq E[\log X]$. Define $X = \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}$, and we get:

$$\log p_\theta(\boldsymbol{x}) \geq \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \, d\boldsymbol{z}$$

Rewrite the integrand using $p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p_\theta(\boldsymbol{x}|\boldsymbol{z}) p_\theta(\boldsymbol{z})$ and we get:

$$\log p_\theta(\boldsymbol{x}) \geq \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z}) p_\theta(\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \, d\boldsymbol{z}$$

This gives us:

$$\begin{aligned}
\log p_\theta(\boldsymbol{x}) &\geq E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\big[p_\theta(\boldsymbol{x}|\boldsymbol{z})\big] + E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\big[\log p_\theta(\boldsymbol{z})\big] - E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\big[\log q_\phi(\boldsymbol{z}|\boldsymbol{x})\big] \\
&= E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\big[p_\theta(\boldsymbol{x}|\boldsymbol{z})\big] - \Big[E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\big[\log q_\phi(\boldsymbol{z}|\boldsymbol{x})\big] - E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\big[\log p_\theta(\boldsymbol{z})\big]\Big] \\
&= E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\big[p_\theta(\boldsymbol{x}|\boldsymbol{z})\big] - KL\Big(q_\phi(\boldsymbol{z}|\boldsymbol{x}) || p_\theta(\boldsymbol{z})\Big)
\end{aligned}$$

This is the Evidence Lower BOund (ELBO):

$$\log p_\theta(\mathbf{x}) \geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - \mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z}))}_{\mathrm{ELBO}(\theta,\phi)}.$$

If we fix $\theta$ and adjust $\phi$, making $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ closer to $p_\theta(\boldsymbol{z})$ we increase $\log p_\theta(\boldsymbol{x})$.

In summary, latent variable models and the introduction of a variational posterior allow us to deal with intractable marginal likelihoods by converting the problem into one of optimizing a tractable lower bound, thereby enabling effective parameter learning even for very complex data like images.

$\ast$ Def. **2.0.1 VAE:** The VAE can be viewed as two coupled, but independently parameterized models: The encoder or the recognition model,and the generation model (aka decoder). The recognition model provides (an approximation to) the posterior for latent variables conditioned on observed data $p_\theta(z|x)$.