Variational Auto Encoders

1 Introduction

∗ Def. **1.0.1 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.

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

∗ Def. **1.0.2 Probabilistic Model:** Assume the observed variable $\boldsymbol{x}$ is a random sample from an unknown underlying process, whose true probability distribution is $p^*(\boldsymbol{x})$. We approximate this underlying process with a chosen model $p_\theta(\boldsymbol{x})$, with parameters $\boldsymbol{\theta}$:

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

Learning is the process of searching for a value of the parameters θ. such that:

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

∗ Def. **1.0.3 Conditional Models:** Often, we are not interested in learning an unconditional model $p_\theta(\boldsymbol{x})$, but a conditional model $p_\theta(\boldsymbol{y}|\boldsymbol{x})$, 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 $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ 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:

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

1.1 Learning in Fully Observed Models with Neural Nets

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

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

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

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:

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

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 θ 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 θ and hill-climb to a local optimum of the Maximum Likelihood.

1.2 Learning and Inference in Deep Latent Variable Models

$*$ Def. **1.2.1 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*.

$*$ Def. **1.2.2 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 θ 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