中国科学院大学

University of Chinese Academy of Sciences

An Introduction to Variational Autoencoders

ZTS

plote5024@gmail.com

Contents

0 Introduction

01 Motivation

One major division in machine learning is generative versus discriminative modeling. Generative Models: Generative models aim to learn the joint distribution of all variables. Generative models simulate the process of generating data in the real world. For example, meteorologists use complex partial differential equations to model weather, and astronomers use equations of motion to model the formation of galaxies. Scientific modeling is often generative modeling, which reveals the generation process by hypothesising and testing theories. Generative models focus on the generation process of data, and Bayesian methods focus on updating beliefs about parameters given data. Discriminative Models: Discriminative models aim to learn predictors from observed data, directly learning the mapping relationship from input to output. Discriminative models make classification or regression predictions directly based on input features.

Details that we don’t know or care about, the nuisance variables, are called noise.

To turn the generative model into a discriminator, we need to use Bayes' rule. This is because the generative model can provide the joint distribution of categories, and Bayes' rule can convert this information into the conditional probability required in the classification task. The generative model (joint distribution) can be converted to the discriminative model (conditional distribution) through Bayes.

Therefore, if the model is wrong (and it is almost always wrong to some extent!), if one is only interested in learning to discriminate, and if one is in an environment with sufficiently large amounts of data, then a purely discriminative model will usually lead to fewer errors in the discriminative task. However, depending on the amount of surrounding data, it may pay off to study the data generation process as a way to guide the training of a discriminator (e.g., a classifier).

Generative modeling can be useful more generally. One can think of it as an auxiliary task.This quest for disentangled,semantically meaningful, statistically independent and causal factors of variation in data is generally known as unsupervised representation learning, and the variational autoencoder (VAE) has been extensively employed for that purpose

The VAE can be viewed as two coupled, but independently parameterized models: the encoder or recognition model, and the decoder or generative model.
These two models support each other. The recognition model delivers to the generative model an approximation to its posterior over latent random variables, which it needs to update its parameters inside an iteration of "expectation maximization" learning. Reversely, the generative model is a scaffolding of sorts for the recognition model to learn meaningful representations of the data, including possibly class-labels. The recognition model is the approximate inverse of the generative model according to Bayes rule.
VAEs marry graphical models and deep learning. The generative model is a Bayesian network of the form $p(x|z)p(z)$, or, if there are multiple stochastic latent layers, a hierarchy such as $p(x|z_L)p(z_L|z_L - 1) \cdots p(z_1|z_0)$. Similarly, the recognition model is also a conditional Bayesiannetwork of the form $q(z|x)$ or as a hierarchy, such as $q(z_0|z_1)...q(z_L|X)$.But inside each conditional may hide a complex (deep) neural network, e.g. $\boldsymbol{z}|\boldsymbol{x} \sim f(x, \varepsilon)$, with f a neural network mapping and ε a noise random variable. Its learning algorithm is a mix of classical (amortized,variational) expectation maximization but through

the reparameterization trick ends up backpropagating through the many layers of the deep neural networks embedded inside of it.

02 Overview

The framework of variational autoencoders (VAEs)provides a principled method for jointly learning deep latent-variable models and corresponding inference models using stochastic gradient descent. The framework has a wide array of applications from generative modeling, semi-supervised learning to representation learning.

03 Probabilistic Models and Variational Inference

Let's use $\boldsymbol{x}$ as the vector representing the set of all observed variables whose joint distribution we would like to model. Note that for notational simplicity and to avoid clutter, we use lower case bold (e.g. x) to denote the underlying set of observed random variables, i.e. flattened and concatenated such that the set is represented as a single vector.

We assume the observed variable $\boldsymbol{x}$ is a random sample from an unknown underlying process, whose true (probability) distribution $p^*(\boldsymbol{x})$ is unknown. We attempt to approximate this underlying process with a chosen model $p_{\theta(\boldsymbol{x})}$, with parameters θ:

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

The learning process is usually to find the value of the parameter θ so that the probability distribution function $p_\theta(\boldsymbol{x})$ given by the model approximates the true distribution of the data $p^*(\boldsymbol{x})$:

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

we wish $p_\theta(\boldsymbol{x})$ to be sufficiently flexible to be able to adapt to the data, such that we have a chance of obtaining a sufficiently accurate model.

031 Conditional Models

Generally speaking, we are trying to predict the output variable $\boldsymbol{y}$ from the input variable $\boldsymbol{x}$. So we want to achieve:

$$p_\theta(\boldsymbol{y}|\boldsymbol{x}) \approx p^*(\boldsymbol{y}|\boldsymbol{x})$$

A relatively common and simple example of conditional modeling is image classification, where x is an image, and y is the image's class, as labeled by a human, which we wish to predict. In this case, $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ is typically chosen to be a categorical distribution, whose parameters are computed from $\boldsymbol{x}$.

Conditional models become more difficult to learn when the predicted variables are very high-dimensional, such as images, video or sound. One example is the reverse of the image classification problem:prediction of a distribution over images, conditioned on the class label. An-other example with both high-dimensional input, and highdimensional output, is time series prediction, such as text or video prediction.

04 Parameterizing Conditional Distributions with Neural Networks

neural networks parameterize a categorical distribution $p_\theta(y|x)$ over a class label $\boldsymbol{y}$, conditioned on an image $\boldsymbol{x}$.
The neural network accepts an input image $\boldsymbol{x}$ and outputs a vector p, where each element p_i in this vector represents the probability that the input image belongs to category i.

$$\boldsymbol{P} = \mathrm{NeuralNet}(\boldsymbol{x})$$

Category distribution

$$p_\theta = \mathrm{Categorical}(y; \boldsymbol{P})$$

05 Directed Graphical Models and Neural Networks

PGMs(Bayesian networks). Directed graphical models are a type of probabilistic models where all the variables are topologically organized into a directed acyclic graph. The joint distribution over the variables of such models factorizes as a product of prior and conditional distributions:

$$p_\theta(\boldsymbol{X}_1, ..., \boldsymbol{X}_M) = \prod_{j=1}^{M} p_\theta(\boldsymbol{X}_j \mid P_a(\boldsymbol{X}_j))$$

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

06 Learning in Fully Observed Models with Neural Nets

Using calculus' chain rule and automatic differentiation tools, we can efficiently compute gradients of this objective, i.e. the first derivatives of the objective $w.r.t.$ its parameters θ.

$$\frac{1}{N_D} \log p_{\theta(D)} \approx \frac{1}{N_M} \log p_{\theta(M)} = \frac{1}{N_M} \sum_{\{x \in M\}} \log p_{\theta(x)}$$

$$\frac{1}{N_D} \nabla_\theta \log p_{\theta(D)} \approx \frac{1}{N_M} \nabla_\theta \log p_{\theta(M)} = \frac{1}{N_M} \sum_{\{x \in M\}} \nabla_\theta \log p_{\theta(x)}$$

07 Learning and Inference in Deep Latent Variable Models

We can extend fully-observed directed models, discussed in the previous section, into directed models with latent variables. Latent variables are variables that are part of the model, but which we don't observe, and are therefore not part of the dataset. We typically use z to denote such latent variables.

In case of unconditional modeling of observed variable $\boldsymbol{x}$, the directed graphical model would then represent a joint distribution $p_{\theta(x,z)}$ over both the observed variables x and the latent variables z.The marginal distribution over the observed variables $p_{\theta(x)}$, is given by:

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

08 Deep Latent Variable Models

We use the term deep latent variable model (DLVM) to denote a latent variable model $p_{\theta(x,z)}$ whose distributions are parameterized by neural networks. Such a model can be conditioned on some context, like $p_{\theta(x,z|y)}$. One important advantage of DLVMs, is that even when each factor (prior or conditional distribution) in the directed model is relatively simple (such as conditional Gaussian)

composition:

- Latent Variables:These are the unobservable implicit variables in the model.
- Observed Variables:Observed data are generated by latent variables through a generative process.
- Generative Model:The generative model defines how to generate observations from latent variables. A deep neural network is usually used to represent the generative process.

$$p(x|z) = f(z; \theta)$$

- Inference Model:Inference models are used to infer latent variables from observed data, usually through a deep neural network.

$$q(z \mid x) = g(x; \varphi)$$

08 Variational Autoencoders

1 Evidence Lower Bound (ELBO)

$$
\begin{aligned}
\log p_{\theta(x)} &= \mathbb{E}_{q_{\varphi(z|x)}}\left[\log p_{\theta(x)}\right] \\
&= \mathbb{E}_{q_{\varphi(z|x)}}\left[\log\left[\frac{p_{\theta(x,z)}}{p_{\theta(z|x)}}\right]\right] \\
&= \mathbb{E}_{q_{\varphi(z|x)}}\left[\log\left[p_{\theta(x,z)}\frac{q_{\varphi(z|x)}}{q_{\varphi(z|x)}p_{\theta(z|x)}}\right]\right] \\
&= \underbrace{\mathbb{E}_{q_{\varphi(z|x)}}\left[\log\left[\frac{p_{\theta(x,z)}}{q_{\varphi(z|x)}}\right]\right]}_{\substack{=L_\theta\varphi(x) \\ \text{ELBO}}} \underbrace{+\mathbb{E}_{q_{\varphi(z|x)}}\left[\log\left[\frac{q_{\varphi(z|x)}}{p_{\theta(z|x)}}\right]\right]}_{=D_{\text{KL}}\left(q_{\varphi(z|x)} \,\|\, p_{\theta(z|x)}\right)}
\end{aligned}
$$

2 EM

Expectation Step:

$$q(Z|\theta^t) = p(Z \mid X, \theta^t)$$

Maximization Step:

$$\theta^{t+1} = \arg\max_\theta \mathbb{E}_{q(Z|\theta^{(t)})}[\log p(X, Z|\theta)]$$

The EM algorithm ensures that after each iteration, the likelihood function increases or remains unchanged until it converges to a local optimal solution or a stable point.

How to understand:

$$p(X|\theta) = \frac{p(X,Z|\theta)}{p(Z|X,\theta)}$$

$$\log P(X|\theta) = \log P(X, Z \mid \theta) - \log P(Z|X,\theta)$$

$$= \log \frac{P(X, Z\ |\theta)}{q(Z)} - \log \frac{P(Z|X,\theta)}{q(Z)}$$

$$\text{left} = \int_z q(Z) \log P(X|\theta)\,\mathrm{d}z = \log P(X|\ \theta) \int_z q(z)\,\mathrm{d}z = \log P(X|\theta)$$

$$\text{right} = \int_Z q(Z) \log \frac{p(X,Z|\theta)}{q(Z)}\,\mathrm{d}Z - \int_z q(Z) \log \frac{p(Z|X,\theta)}{q(Z)}\,\mathrm{d}Z$$

$\log P(X|\theta) = \mathrm{ELBO} + \mathrm{KL}(q\|p).$

So at this point we can get the meaning of : θ

$$\theta = \arg\max_\theta \mathrm{ELBO}$$

$$= \arg\max_\theta \int q(z) \log \frac{p(x,z|\theta)}{q(z)}\,\mathrm{d}z$$