Probabilistic models for image synthesis - Part 1

November 7, 2023

1 Introduction

Note that the terminology here slightly differs from what was covered in class.

Our generative model consists of first sampling a “noise” vector z from some *prior distribution* $\pi(z)$ and then passing it through a *deterministic neural network* f_θ. To convert this into a probabilistic model, let us assume that f_θ produces the parameters of a distribution of colors at each pixel. In the simplest case, we can assume that f_θ produces the mean color for each pixel, and then the image x is obtained by sampling from a Gaussian with fixed variance centered on this mean:

$$P_\theta(x|z) = \mathcal{N}(f_\theta(z), \sigma^2 I) \tag{1}$$

The final probabilistic model is then:

$$P_\theta(x) = \pi(z)P_\theta(x|z) \tag{2}$$

Now we need to fit the parameters θ. Suppose we have a training dataset D on which to fit the parameters θ. We can fit θ by maximizing the (log) probability of the provided data, or (because we like minimization problems) minimizing the *negative log likelihood* of the available data:

$$\theta^* = \arg\min_\theta \sum_{x \in D} (-\log P_\theta(x)) \tag{3}$$

This in turn requires us to estimate $P_\theta(x)$ for each image (data point) x. Unfortunately, to do this we must marginalize out the ”noise” vector that could have generated this image:

$$P_\theta(x) = \int_z \pi(z)P_\theta(x|z)dz = \mathbb{E}_{z\sim\pi}[P_\theta(x|z)] \tag{4}$$

Unfortunately this integral is intractable to calculate. In principle one could estimate this by using a few samples to approximate the expectation. However, because the image x is more or less a deterministic function of z, there exist only a very small set of possible z’s for which $P_\theta(x|z)$ is non-zero. Hitting these z by random sampling alone will require an intractably large number of samples. Thus, *prima facie*, getting a good set of parameters θ seems difficult or impossible.

2 The variational approach

Variational techniques are a broad class of techniques in machine learning/statistics that are designed precisely for getting around intractable marginalization problems, like the one above. The key idea in variational techniques is to replace the hard-to-marginalize distribution with a more tractable approximation, and then have an additional objective that ensures the approximate is close to the original.

For our particular problem, we will replace the prior $\pi(z)$ with a different distribution $q(z;x)$. The idea here is that unlike π, q is actually dependent on a particular image x, so that when we sample from q we get z values that are in fact likely to produce x. Thus, instead of working with $P_\theta(x)$, we work with the following alternative:

$$\tilde{P}_\theta(x,q) = \int_z q(z;x) P_\theta(x|z)dz = \mathbb{E}_{z\sim q(z;x)}[P_\theta(x|z)] \tag{5}$$

So we want to minimize:

$$\begin{aligned} -\log \tilde{P}_\theta(x,q) &= -\log \mathbb{E}_{z\sim q(z;x)}[P_\theta(x|z)] && (6)\\ &\leq \mathbb{E}_{z\sim q(z;x)}[-\log P_\theta(x|z)] && (7)\end{aligned}$$

where the last step uses Jensen's inequality. Note that unlike our original objective, this upper bound is easy to estimate by sampling from q and summing up.

However, to use this, we need to choose a q so that what we are optimizing is at least related to our original objective. Intuitively, since we got here by replacing π with q, we can additionally add a loss term that penalizes the difference between π and q. Thus, we may want to minimize:

$$L_v = D_{KL}[q(z;x)\|\pi(z)] + \mathbb{E}_{z\sim q(z;x)}[-\log P_\theta(x|z)] \tag{8}$$

where D_{KL} is the KL divergence between distributions:

$$D_{KL}(q\|p) = \int_z q(z)\log\frac{q(z)}{p(z)} = \mathbb{E}_{z\sim q}\log\frac{q(z)}{p(z)} \tag{9}$$

What relationship does our new objective L_v have with our old objective?

Let us see:

$$\begin{aligned}
L_v &= D_{KL}[q(z;x)\|\pi(z)] + \mathbb{E}_{z\sim q(z;x)}[-\log P_\theta(x|z)] && (10)\\
&= \mathbb{E}_{z\sim q(z;x)} \log \frac{q(z;x)}{\pi(z)} + \mathbb{E}_{z\sim q(z;x)}[-\log P_\theta(x|z)] && (11)\\
&= \mathbb{E}_{z\sim q(z;x)} \log \frac{q(z;x)}{\pi(z)P_\theta(x|z)} && (12)\\
&= \mathbb{E}_{z\sim q(z;x)} \log \frac{q(z;x)}{P_\theta(x,z)} && (13)\\
&= \mathbb{E}_{z\sim q(z;x)} \log \frac{q(z;x)}{P_\theta(z|x)P_\theta(x)} && (14)\\
&= \mathbb{E}_{z\sim q(z;x)} \log \frac{q(z;x)}{P_\theta(z|x)} + \mathbb{E}_{z\sim q(z;x)}[-\log P_\theta(x)] && (15)\\
&= D_{KL}[q(z;x)\|P_\theta(z|x)] + \mathbb{E}_{z\sim q(z;x)}[-\log P_\theta(x)] && (16)\\
&\geq \mathbb{E}_{z\sim q(z;x)}[-\log P_\theta(x)] \qquad (\because D_{KL} \geq 0) && (17)
\end{aligned}$$

Thus, L_v is an upper bound on the true objective. Alternatively, $-L_v$ is a lower bound on the log probability of the data ($\log P_\theta(x)$). Therefore, $-L_v$ is called the *evidence lower bound* or ELBO.

3 Variational autoencoders

For variational autonecoders, the "noise" vectors are much lower dimensional than images. q is parametrized by a neural network that takes x as input and produces the mean and variance for a distribution in the noise space. Because both q and P_θ involve Gaussians, for which the KL divergence is easy to compute in closed form, in principle VAEs are easy to optimize.

In practice however, VAEs find it a challenge to model distributions of high resolution images. The reason is because during training, one needs to train both the backward mapping from images to noise vectors as well as the forward mapping from noise to images, while at the same time ensuring that at test time we can sample noise and produce a realistic image. This proves to be a difficult optimization in general.