TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2018

The Evidence Lower Bound (the ELBO)

Variational Autoencoders

Big Picture: Latent Variables

We are often interested in models of the form

$$P_\Phi(y) = \sum_z P_\Phi(z) P_\Phi(y|z).$$

Note that CTC has this form.

Probabilistic grammar models also have this form where y is a sentence and z is a parse tree.

Rate-Distortion Autoencoders also have this form where z is the compression of y.

In these cases the sum over z can be computed exactly.

Big Picture: Friendly Distributions

A distribution $P(u)$ will be called **friendly** if we can both draw samples from it and compute $P(u)$ for any value u.

$$P_\Phi(y) = \sum_z P_\Phi(z) P_\Phi(y|z).$$

It is often the case that $P_\Phi(z)$ is friendly, and $P_\Phi(y|z)$ is friendly, but $P_\Phi(y)$ is not friendly (the sum over z is intractible).

For example z might be an assignment of truth values to Boolean variables and y might be the value of a fixed Boolean formula Φ. In this case determining if $P_\Phi(y) > 0$ is the SAT problem which is NP hard.

Superpixel Colorization

SLIC superpixels, Achanta et al.

x is black and white, y color, z a segmentation.

$$P_\Phi(y|x) = \sum_z P_\Phi(z|x) P_\Phi(y|z,x).$$

Superpixel Colorization

$P_\Phi(z|x)$ is defined by a deep network computing a friendly graphical model on segmentations – perhaps a independent distribution over segment indeces for each superpixel.

$P_\Phi(y|z,x)$ is a deep network taking a particular segmentation (a segment index at each pixel) and computing a distribution over colors for each segment.

Although $P(z|x)$ is friendly, and $P_\Phi(y|z,x)$ is friendly, $P(y|x)$ not friendly (similar to the SAT example).

Big Picuture: ELBO Replaces Search with Generation

$$\begin{aligned}
P_\Phi(y) &= E_{z\sim P_\Phi(z)} P_\Phi(y|z) \quad \text{sampling } z \text{ is ineffective} \\
\ln P_\Phi(y) &= E_{z\sim P_\Psi(z|y)} \ln P_\Phi(y) \quad \text{introduce } z \text{ generator using } y \\
&= E_{z\sim P_\Psi(z|y)} \left(\ln P_\Phi(y) P_\Phi(z|y) + \ln \frac{P_\Psi(z|y)}{P_\Phi(z|x)} + \ln \frac{1}{P_\Psi(z|y)} \right) \\
&= E_{z\sim P_\Psi(z|y)} P_\Phi(z,y) + H(P_\Psi(z|y)) + KL(P_\Psi(z|y), P_\Phi(z|y)) \\
&\geq E_{z\sim P_\Psi(z|y)} P_\Phi(z) P_\Phi(y|z) + -\ln P_\Psi(z|y)
\end{aligned}$$

Superpixel Colorization

SLIC superpixels, Achanta et al.

x is black and white, y color, z a segmentation.

$P_\Phi(z|x)$ is friendly and $P_\Phi(y|z,x)$ is friendly but $P(y|x)$ is not friendly.

$P_\Psi(z|y,x)$ computes a friendly graphical model for z given y.

Measuring ELBO Loss

$$\mathcal{L}_{\mathrm{ELBO}}(y, \Phi, \Psi) = E_{z \sim P_\Psi(y)} \; \ln P_\Psi(y) - \ln P_\Phi(z) P_\Phi(y|z)$$

If $P_\Phi(z)$, $P_\Phi(y|z)$, and $P_\Psi(z|y)$ are friendly (even whwn when $P_\Phi(y)$ is not friendly) we can measure ELBO loss through sampling.

If we can measure it, we can do gradient descent on it (but perhaps with difficulty).

A General ELBO Architecture

$P_\Phi(y|x)$

$$
\begin{aligned}
& \text{intput } x \\
& \vdots \\
z &= \operatorname*{expsoftmax}_z \ldots \\
& \text{input } z \\
& \vdots \\
y &= \operatorname*{expsoftmax}_y \ \ldots
\end{aligned}
$$

$P_\Psi(z|y,x)$

$$
\begin{aligned}
& \text{intput } x, y \\
& \vdots \\
z &= \operatorname*{expsoftmax}_z \ldots
\end{aligned}
$$

The exponential softmaxes are friendly (they produce a friendly graphical model).

EM is Alternating Maximization of the ELBO

Forward-backward EM for HMMs and inside-outside EM for PCFGs (or any EM) can be written as

$$\begin{aligned}
\text{ELBO} &= E_{z\sim P_\Psi(z|y)} \ln P_\Phi(z,y) + H(P_\Psi(z|y)) \quad (1) \\
&= \ln\; P_\Phi(y) - KL(P_\Psi(z|y), P_\Phi(z|y)) \quad (2)
\end{aligned}$$

$$\text{by (2)} \quad \Psi^{t+1} = \operatorname*{argmin}_\Psi\; E_{y\sim\text{Train}}\, KL(P_\Psi(z|y), P_{\Phi^t}(z|y)) = \Phi^t$$

$$\text{by (1)} \quad \Phi^{t+1} = \operatorname*{argmax}_\Phi E_{y\sim\text{Train}}\, E_{z\sim P_{\Phi^t}(z|y)} \ln P_\Phi(z,y)$$

ELBO Loss Consistency

$$\mathcal{L}_{\mathrm{ELBO}}(y, \Phi, \Psi) = E_{z \sim P_\Psi(y)} \ln P_\Psi(y) - \ln P_\Phi(z) P_\Phi(y|z)$$

$$\min_Q E_{y \sim \mathrm{Pop}} \mathcal{L}_{\mathrm{ELBO}}(y, \Phi, Q) = H(\mathrm{Pop}, P_\Phi)$$

Hard ELBO

Hard ELBO is to ELBO as hard EM is to EM.

$$\mathcal{L}_{\mathrm{HELBO}}(y, \Phi, \Psi) = E_{z \sim P_\Phi(z|y)} - \ln P_\Phi(z, y)$$

$$\min_{P,Q} E_{y \sim \mathrm{Pop}}\, \mathcal{L}_{\mathrm{RELBO}}(y, P, Q) \leq H(\mathrm{Pop}) + \ln 2$$

This can be proved from Shannon's source coding theorem where z is the code for y.

Variational Auto Encoders (VAEs)

In these slides I am reserving the term VAE for applications of the ELBO where $p_\Phi(z)$ a continuous Gaussian density and the ELBO expression is used directly on continuous densities with differential entropy.

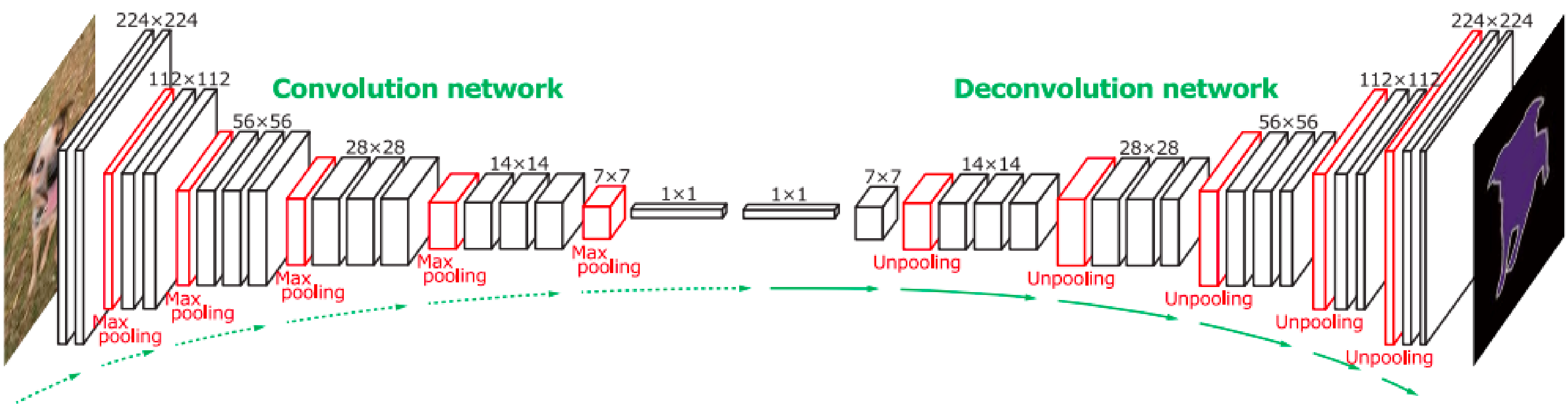
Working with continuous densities greatly simplifies gradient descent.

I will describe the architecture and discuss possible problems with continuous distributions later.

A VAE for Images

Auto-Encoding Variational Bayes, Diederik P Kingma, Max Welling, 2013.

y $\quad z_\Psi(y, \epsilon) \quad z \; p_\Phi(z) \quad \hat{y}_\Phi(z) \quad ||y - \hat{y}||^2$



[Hyeonwoo Noh et al.]

Distortion as Probability Density

We will assume that we have a function (deconvolution) mapping z to $\hat{y}_\Phi(z)$.

For rate-distortion autoencoders we assume a distortion function $D(y,\ \hat{y}_\Phi(z(y)))$.

L_1 or L_2 distortion can be converted to a continuous probability density.

$$p(y|\hat{y}) = \frac{1}{Z}\, e^{-D(y,\hat{y})}$$

This gives a density $p_\Phi(y|z) = p(y|\hat{y}_\Phi(z))$.

The Reparameterization Trick

$$\mathcal{L}_{\mathrm{ELBO}}(y, \Phi, \Psi) = E_{z \sim p_\Psi(z|y)} \;\; \ln\; p_\Psi(z|y) - \ln p_\Phi(z) + \lambda ||y - \hat{y}_\Phi(z)||^2$$

All variables are now continuous.

How do we differentiate the sampling?

The Reparameterization Trick

$$\mathcal{L}_{\text{ELBO}}(y, \Phi, \Psi) = E_{z \sim p_\Psi(z|y)} \;\; \ln\; p_\Psi(z|y) - \ln p_\Phi(z) + \lambda ||y - \hat{y}_\Phi(z)||^2$$

We note that in practice all sampling is computed by a deterministic function of (pseudo) random numbers.

We can make this explicit.

Model $P_\Psi(z|y)$ by $\epsilon \sim$ noise, $z = z_\Psi(y, \epsilon)$

The Reparameterization Trick

$$E_{z \sim p_\Psi(z|y)} \;\ln\; p_\Psi(z|y) - \ln p_\Phi(z) + \lambda ||y - \hat{y}_\Phi(z)||^2$$

becomes

$$E_{\epsilon \sim \mathcal{N}(0,I)} \;\; z := z_\Psi(y) + \sigma \odot \epsilon;\; \ln\; p_\Psi(z|y) - \ln p_\Phi(z) + \lambda ||y - \hat{y}_\Phi(z)||^2$$

Decoding with L_2 Distortion

Switching back to minimization, we can now rewrite the objective as

$$\min\; E_{y,\epsilon}\;\left(|z_\Psi(y,\epsilon)|_\Phi + \frac{1}{2}\lambda||y - \hat{y}_\Phi(z_\Psi(y,\epsilon))||^2\right) - |z_\Psi(y,\epsilon)|_{\Psi,y}$$

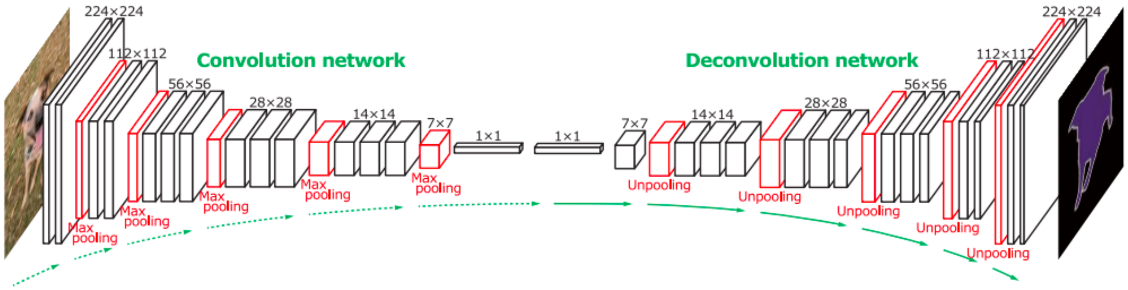
$$|z|_\Phi = -\log_2 P_\Phi(z) \qquad\qquad |z|_{\Psi,y} = -\log_2 P_\Psi(z|y)$$

For z discrete, $|z|_\Phi$ is the code length of $z(y,\epsilon)$ under an optimal code for P_Φ.

$|z|_{\Psi,y}$ is the code length for z under the code for $P_\Psi(z|y)$.

Sampling

$$P_\Psi(z|y) \qquad z \qquad P_\Phi(z,y)$$



[Hyeonwoo Noh et al.]

Sampling uses just the second half $P_\Phi(z,y)$.

Sampling from Gaussian Variational Autoencoders

[Alec Radford]

Why Blurry?

A common explanation for the blurryness of images generated from VAEs is the use of L_2 as the distortion measure.

It does seem that L_1 works better (see the slides on image-to-image GANs).

However, training on L_2 distortion can produce sharp images in rate-distortion autoencoders (see the slides on rate-distortion autoencoders).

END