TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2019

Latent Variable Models

Expectation Maximization (EM)

The Evidence Lower Bound (the ELBO)

Variational Autoencoders (VAEs)

Latent Variable Models

We are often interested in models of the form

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

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

Here z is the latent variable.

Reasons for Latent Variables

- Measuring cross entropy.
- Inserting domain knowledge.
- Improving interpretability.

Dangers of Latent Variables

Domain "knowledge" is often wrong.

Measuring Cross Entropy (TZ)

For structured label spaces (sentences or images), cross-entropy is currently only measured for "autoregressive" (directed) graphical models.

For images this is done with "pixel RNNs".

But compression models can be viewed as latent variable models with measurable cross-entropy.

Cross Entropy of a Compression Model

A compression model Φ maps a label y to a latent value (code) $z_\Phi(y)$ such that there exists a decompression algorithm $y_\Phi(z)$ satisfying

$$y_\Phi(z_\Phi(y)) = y.$$

Let $|z_\Phi(y)|$ be the bit length of the compression of y.

Cross Entropy of a Compression Model

$$\begin{aligned}\Phi^* &= \operatorname*{argmin}_{\Phi} \; E_{y\sim \mathrm{Pop}} \; -\ln P_\Phi(y) \\ &= \operatorname*{argmin}_{\Phi} \; E_{y\sim \mathrm{Pop}} \; |z_\Phi(y)|\end{aligned}$$

Assuming universal expressive power, Shannon's source coding theorem implies

$$E_{y\sim \mathrm{Pop}} \; |z_{\Phi^*}(y)| \leq H_2(y) + 1$$

Measuring Rate-Distortion Points

A noisy-channel RDA

$$\Phi^* = \operatorname*{argmin}_{\Phi} I(y, z) + \lambda E_{y \sim \text{Pop},\, z \sim p_\Phi(z|y)} \text{Dist}(y, y_\Phi(z))$$

has a measurable rate-distortion trade off provided we can measure the bandwidth $I(y, z)$ of the noisy channel.

Sparse Labeling Compression (TZ)

Given a graphical model Φ on semantic segmentations we can code a segmentation y by a sparse segmentation $z_\Phi(y)$ assigning a label to only a small fraction of the pixels.

We can define a decoding $y_\Phi(z)$ to be the result of running a deterministic local search over the labels of the unspecified pixels to find a locally best-scoring full semantic segmentation.

We can then define the encoding $z_\Phi(y)$ to be

$$z_\Phi(y) = \operatorname*{argmin}_{z:\, y_\Phi(z)=y} |z|$$

where $|z|$ is the number of pixels assigned by z.

Encoding Domain Knowledge

In CTC the latent variable z (the latent sequence of phonemes and blanks) and the blank removal operation $y(z)$ constitute useful engineered domain knowledge.

There have been many attempts to build standard latent linguistic structure, such as parse trees, into deep language models. These attempts have largely failed to improve performance on end-to-end applications.

Attempts to use classical graphical models in computer vision have also largely failed.

But I believe that we should not give up on engineering.

Improved Interpretabilty

In CTC we can see where in the input signal system the gold labels are coming from.

If we have latent parse trees in an NLP model we can see what parse the system is using.

If the latent variables are engineered to have semantics then, by construction, we have a semantic interpretation of internal processing.

Attention and Latent Variables

In machine translation attention is used to handle a latent alignment between the input sentence and the gold label translation.

In general, attention can be viewed as defining a probability distribution over a latent choice.

Attention is the central mechanism in the transformer network (to be discussed later).

The transformer network can be viewed as constructing latent trees over an input sentence.

Expectation Maximization (EM)

Mixture of Gaussian Modeling

$$\Phi = \pi[z], \mu[z], \Sigma[z], \quad z \in \{1, \ldots, k\}$$

$$\begin{aligned} p_\Phi(y) &= \sum_z P(z)p(y|z) \\ &= \sum_{z=1}^{k} \pi[z] \frac{1}{Z[z]} \exp\left(-\frac{1}{2}(y-\mu[z])^\top \Sigma[z]^{-1}(y-\mu[z])\right) \end{aligned}$$

Expectation Maximization (EM)

Mixture of Gaussian Modeling

$$\Phi = \pi[z], \mu[z], \Sigma[z], \quad z \in \{1, \ldots, k\}$$

$$\text{Train} = \{y_1, \ldots, y_N\}$$

Until Convergence:

$$P_\Phi(z|y_j) = \frac{\pi[z]P(y_j|z)}{\sum_z \pi[z]P(y_j|z)} \quad \text{Inference (E step)}$$

$$\left.\begin{aligned} \pi^{t+1}[z] &= \frac{1}{N} \sum_j P_{\Phi^t}(z|y_j) \\ \mu^{t+1}[z] &= \frac{1}{N} \sum_j P_{\Phi^t}(z|y_j) y_j \\ \Sigma^{t+1}[z] &= \frac{1}{N} \sum_j P_{\Phi^t}(z|y_j) y_j y_j^\top \end{aligned}\right\} \text{Model Update (M step)}$$

General EM

$$\Phi^* = \operatorname*{argmin}_{\Phi} E_{y \sim \text{Train}} - \ln P_\Phi(y)$$

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

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

Update (M Step)

Inference (E Step)

Colorization

x $\hat{y}$ y

Larsson et al., 2016

x is a grey level image.

y is a color image drawn from $\text{Pop}(y|x)$.

$\hat{y}$ is an arbitrary color image.

$P_\Phi(\hat{y}|x)$ is the probability that model Φ assigns to the color image $\hat{y}$ given grey level image x.

Colorization with Latent Semantic Segmentation (TZ)

x $\hat{y}$ y

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

input x

$P_\Phi(z|x) = \ldots$ semantic segmentation

$P_\Phi(\hat{y}|z,x) = \ldots$ segment colorization

Maybe EM?

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

$$\Phi^{t+1} = \underset{\Phi}{\operatorname{argmin}} \; E_{y \sim \text{Train}} \; E_{z \sim P_{\Phi^t}(z|y)} \; - \ln P_\Phi(z, y)$$

Update Inference

In most cases the inference is intractible!

Variational Inference:

The Evidence Lower Bound (The ELBO)

We introduce a friendly model $P_\Psi(z|y)$ to approximate $P_\Phi(z|y)$.

$$
\begin{aligned}
\ln P_\Phi(y) &= E_{z\sim P_\Psi(z|y)} \ln P_\Phi(y) \\
&= E_{z\sim P_\Psi(z|y)} \left(\ln P_\Phi(y) \frac{P_\Phi(z|y)}{P_\Psi(z|y)} + \ln \frac{P_\Psi(z|y)}{P_\Phi(z|y)} \right) \\
&= \left(E_{z\sim P_\Psi(z|y)} \ln \frac{P_\Phi(z,y)}{P_\Psi(z|y)} \right) + KL(P_\Psi(z|y), P_\Phi(z|y)) \\
&= \qquad \text{ELBO} \qquad + KL(P_\Psi(z|y), P_\Phi(z|y))
\end{aligned}
$$

Optimization of the ELBO

$$\ln P_\Phi(y) \geq \mathrm{ELBO}(\Phi, \Psi, y)$$

$$= E_{z \sim P_\Psi(z|y)} \ln \frac{P_\Phi(z, y)}{P_\Psi(z|y)}$$

$$\Phi^*, \Psi^* = \operatorname*{argmax}_{\Phi,\Psi} E_{y \sim \mathrm{Pop}} \mathrm{ELBO}(\Phi, \Psi, y)$$

$$\Phi^*, \Psi^* = \operatorname*{argmax}_{\Phi,\Psi} E_{y \sim \mathrm{Pop}} E_{z \sim P_\Psi(z|y)} \ln \frac{P_\Phi(z, y)}{P_\Psi(z|y)}$$

EM is Alternating Maximization of the ELBO

$$\text{ELBO} = E_{z \sim P_\Psi(z|y)} \ln \frac{P_\Phi(z,y)}{P_\Psi(z|y)} \quad (1)$$

$$= \ln P_\Phi(y) - KL(P_\Psi(z|y), P_\Phi(z|y)) \quad (2)$$

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

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

Hard ELBO

In hard EM we use only the single most likely z rather than the expectation of $z \sim P_{\Phi^t}(z|y)$.

K-means is hard EM for mixtures of Gaussians (when all covariances matrices are fixed at I).

$$\text{ELBO} = \left(E_{z \sim P_\Psi(z|y)} \ \ln \frac{P_\Phi(z,y)}{P_\Psi(z|y)} \right)$$

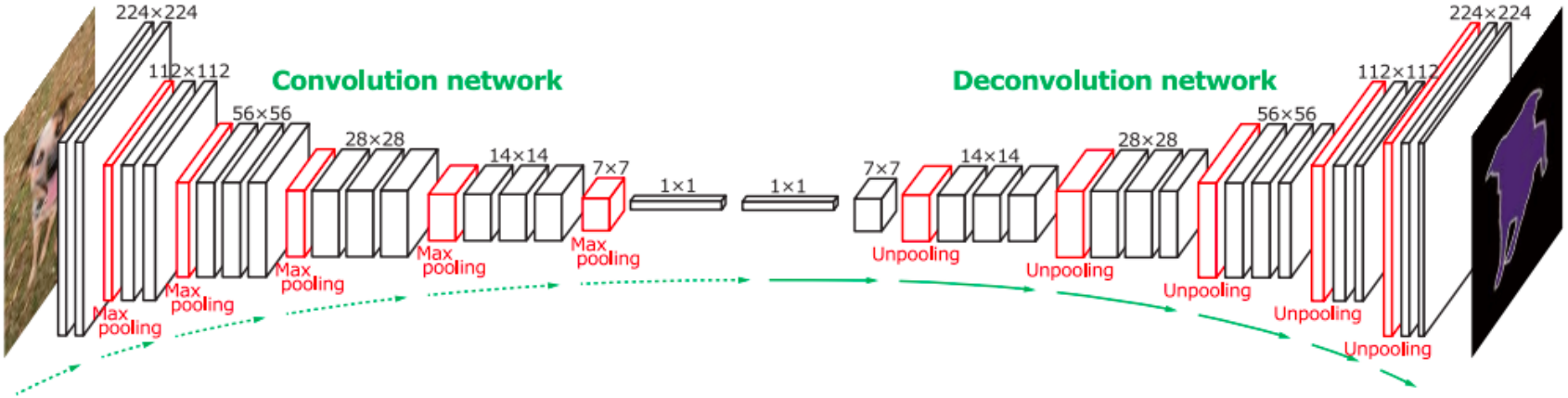
$$\text{ELBO} = \left(E_{z \sim P_\Psi(z|y)} \ \ln P_\Phi(z,y) \right) + H(P_\Psi(z|y))$$

$$\text{hard ELBO} = E_{z \sim P_\Psi(z|y)} - \ln P_\Phi(z,y)$$

A VAE for Images

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

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



[Hyeonwoo Noh et al.]

Gaussian VAEs

$$p_\Phi(z) \propto \exp\left(\sum_i (z[i] - \mu[i])^2/(2\sigma[i]^2)\right)$$

$$p_\Phi(y|z) \propto \exp\left(\sum_j (y[j] - y_\Phi(z)[j])^2/(2\gamma[j]^2)\right)$$

$$p_\Psi(z|y) \propto \exp\left(\sum_i (z[i] - z_\Psi(y)[i])^2/(2\sigma_\Psi(y)[i]^2)\right)$$

Gaussian VAEs

$$\begin{aligned}
\text{ELBO} &= E_{z\sim p_\Psi(z|y)} \ln \frac{p_\Phi(z,y)}{p_\Psi(z|y)} \\
\mathcal{L}_{\text{ELBO}} &= -\text{ELBO} \\
&= KL(p_\Psi(z|y), p_\Phi(z)) + E_{z\in p_\Psi(z|y)} - \ln p_\Phi(y|z)
\end{aligned}$$

Continuous KL-divergence is ok.

Continuous cross-entropy has issues ...

Gaussian VAEs vs. Gaussian RDAs

$$\mathcal{L}_{\text{ELBO}}(y) = KL(p_\Psi(z|y), p_\Phi(z)) + E_{z \in p_\Psi(z|y)} - \ln p_\Phi(y|z)$$

$$\mathcal{L}_{\text{RDA}}(y) = KL(p_\Phi(z|y), p_\Psi(z)) + \lambda\, E_{z \in p_\Phi(z|y)}\, D(y, y_\Phi(z))$$

Through a reparameterization these can be written as

$$\mathcal{L}_{\text{ELBO}}(y) = KL(p_\Psi(z|y), \mathcal{N}(0, I)) + E_{z \in p_\Psi(z|y)} - \ln p_\Phi(y|z)$$

$$\mathcal{L}_{\text{RDA}}(y) = KL(p_\Phi(z|y), \mathcal{N}(0, I))) + \lambda\, E_{z \in p_\Phi(z|y)}\, D(y, y_\Phi(z))$$

Sampling

Sample $z \sim \mathcal{N}(0, I)$ and compute $y_\Phi(z)$

[Alec Radford]

Reasons for Latent Variables

- Measuring cross entropy.

- Inserting domain knowledge.

- Improving interpretability.

Dangers of Latent Variables

Domain "knowledge" is often wrong.

END