

TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2023

Variational Auto-Encoders (VAEs)

Generative AI: Autoregression and GANs

For an autoregressive language model we can compute $P_{\text{gen}}(y)$ and train a generative model by cross-entropy loss.

$$\text{gen}^* = \underset{\text{gen}}{\text{argmin}} E_{y \sim P_{\text{op}}} - \ln P_{\text{gen}}(y)$$

But it is not obvious how to this for continuous signals like sounds and images.

GANs replace the cross-entropy loss with an adversarial discrimination loss.

Generative AI for Continuous Data: Flow Models

$$\text{gen}^* = \underset{\text{gen}}{\operatorname{argmin}} E_{y \sim \text{pop}(y)} - \ln p_{\text{gen}}(y)$$

Flow-based generative models work with Jacobians over continuous transformations (no ReLUs) and can be directly trained with cross-entropy loss.

But flow models have not caught on and we will not cover them.

Generative AI for Continuous Data: VAEs

A variational autoencoder (VAE) is defined by three parts:

- An encoder distribution $P_{\text{enc}}(z|y)$.
- A “prior” distribution $P_{\text{pri}}(z)$
- A generator distribution $P_{\text{gen}}(y|z)$

VAE generation uses $P_{\text{pri}}(z)$ and $P_{\text{gen}}(y|z)$ (like a GAN).

VAE training uses a “GAN inverter” $P_{\text{enc}}(z|y)$.

We will rely on expectation notation and will not distinguish discrete distributions from densities.

Cross-Entropy for Continuous Data: L_2 Loss

Define $p_{\text{gen}}(y|z)$ by

$$y = \hat{y}_{\text{gen}}(z) + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

We then get that

$$-\ln p_{\text{gen}}(y|z) = \frac{\|\hat{y}_{\text{gen}}(z) - y\|^2}{2\sigma^2} + \ln Z(\sigma)$$

For a fixed σ we can ignore $\ln Z(\sigma)$ and we get L_2 distortion loss.

Cross-Entropy for Continuous Data: L_2 Loss

$$-\ln p_{\text{gen}}(y|z) = \frac{||\hat{y}_{\text{gen}}(z) - y||^2}{2\sigma^2} + \ln Z(\sigma)$$

For all the VAE models we will discuss, cross-entropy loss for continuous distributions will be measured by L_2 distortion.

When using L_2 distortion loss z should nearly specify y .

Diffusion Model Preview

A diffusion model multi-step (Markovian) VAE where each step adds a small amount of noise.

Each step of a diffusion model is a VAE:

- $P_{\text{enc}}(z|y)$ is defined by adding a small amount of noise to y .
- $P_{\text{pri}}(z)$ is trained to model the marginal onto z of $\text{Pop}(y)P_{\text{enc}}(z|y)$.
- A “denoising” $\hat{y}_{\text{gen}}(z)$ is computed by a U-Net.

Here z contains almost all the information in y .

Fixed Encoder Training

In a diffusion model the encoder is fixed.

$$\text{pri}^*, \text{gen}^* = \underset{\text{pri}, \text{gen}}{\operatorname{argmin}} E_{y \sim \text{Pop}(y), z \sim \text{enc}(z|y)} \left[-\ln P_{\text{pri}}(z) P_{\text{gen}}(y|z) \right]$$

This is a cross-entropy loss from a joint “population distribution” $P_{\text{Pop}, \text{enc}}(y, z)$ to a model distribution $P_{\text{pri}, \text{gen}}(y, z)$.

Assuming universality we get $P_{\text{pri}^*, \text{gen}^*}(z, y) = P_{\text{Pop}, \text{enc}}(z, y)$ which implies $P_{\text{pri}^*, \text{gen}^*}(y) = \text{Pop}(y)$.

Training the Encoder (The Bayesian Interpretation)

VAEs were originally motivated by a Bayesian interpretation:

- $P_{\text{pri}}(z)$ is the Bayesian prior on hypothesis z .
- $P_{\text{gen}}(y|z)$ is the probability of the “evidence” y given hypothesis z .
- $P_{\text{enc}}(z|y)$ is a model approximating the Bayesian posterior on hypothesis z given evidence y .

The Bayesian motivation is to train $P_{\text{enc}}(z|y)$ to approximate Bayesian inference.

Training the Encoder (Bayesian Interpretation)

$$\text{enc}^*, \text{pri}^*, \text{gen}^* = \underset{\text{enc}, \text{pri}, \text{gen}}{\text{argmin}} \ E_{\textcolor{red}{y} \sim \text{Pop}, \textcolor{red}{z} \sim P_{\text{enc}}(z|y)} \mathcal{L}(y, z)$$

$$\mathcal{L}(y, z) = -\ln \frac{P_{\text{pri}}(z) P_{\text{gen}}(y|z)}{P_{\text{enc}}(z|y)}$$

Here we can hope to train the encoder to capture a causal origin for y .

Training the Encoder (Bayesian Interpretation)

Consider training P_{enc} while holding P_{pri} and P_{gen} fixed.

$$\begin{aligned}\text{enc}^* &= \underset{\text{enc}}{\text{argmin}} E_{y \sim \text{Pop}(y), z \sim \text{enc}(z|y)} - \ln \frac{P_{\text{pri}}(z) P_{\text{gen}}(y|z)}{P_{\text{enc}}(z|y)} \\ &= \underset{\text{enc}}{\text{argmin}} E_{y \sim \text{Pop}(y), z \sim \text{enc}(z|y)} - \ln \frac{P_{\text{pri,gen}}(y) P_{\text{pri,gen}}(z|y)}{P_{\text{enc}}(z|y)} \\ &= \underset{\text{enc}}{\text{argmin}} E_{y \sim \text{Pop}(y)} KL(P_{\text{enc}}(z|y), P_{\text{pri,gen}}(z|y)) + E_{y \sim \text{Pop}(y)} [-\ln P_{\text{pri,gen}}(y)]\end{aligned}$$

Training $P_{\text{enc}}(z|y)$ to equal $P_{\text{pri,gen}}(z|y)$ can drive the KL term to zero.

Training $P_{\text{pri}}(z)$ and $P_{\text{gen}}(y|z)$ can drive the cross-entropy term to $H(\text{Pop})$.

The Evidence Lower Bound (ELBO)

The previous derivation can be applied to an arbitrary fixed value of y yielding.

$$\begin{aligned}\ln P_{\text{pri,gen}}(y) &\geq E_{z \sim P_{\text{enc}}(z|y)} \ln \frac{P_{\text{pri}}(z)P_{\text{gen}}(y|z)}{P_{\text{enc}}(z|y)} \\ &= E_{z \sim P_{\text{enc}}(z|y)} [-\mathcal{L}(y, z)]\end{aligned}$$

A Bayesian thinks of y as “evidence” for hypothesis z in the Bayesian model. This method of training $P_{\text{enc}}(z|y)$ is called **variational Bayesian inference**.

Under the Bayesian interpretation the negative of the VAE loss is called **the evidence lower bound (ELBO)** .

Training the Ecoder: Posterior Collapse

$$\text{enc}^*, \text{pri}^*, \text{gen}^* = \underset{\text{enc}, \text{pri}, \text{gen}}{\text{argmin}} \ E_{\textcolor{red}{y} \sim \text{Pop}, \textcolor{red}{z} \sim P_{\text{enc}}(z|y)} \mathcal{L}(y, z)$$

$$\mathcal{L}(y, z) = -\ln \frac{P_{\text{pri}}(z)P_{\text{gen}}(y|z)}{P_{\text{enc}}(z|y)}$$

The objective is fully optimized whenever

$$P_{\text{pri}}(z)P_{\text{gen}}(y|z) = \text{Pop}(y)P_{\text{enc}}(z|y)$$

Training the Ecoder: Posterior Collapse

Under the Bayesian interpretation we would like z to provide useful information about (a causal origin of) y .

However the objective function only produces

$$P_{\text{pri}}(z)P_{\text{gen}}(y|z) = \text{Pop}(y)P_{\text{enc}}(z|y)$$

If the cross-entropy loss is measured by L_2 distortion we need z to nearly specify y .

However, for discrete models, such as language models, we can compute $P_{\text{gen}}(y|z)$ even if z carries no information about y .

Training the Encoder: Posterior Collapse

$$P_{\text{pri}}(z)P_{\text{gen}}(y|z) = \text{Pop}(y)P_{\text{enc}}(z|y)$$

For language models training the encoder allows the encoder to ignore y .

This is called “posterior collapse”.

Optimizing the Encoder: The Reparameterization Trick

$$\text{enc}^* = \underset{\text{enc}}{\operatorname{argmin}} \quad E_{y \sim \text{Pop}(y), z \sim P_{\text{enc}}(z|y)} \left[-\ln \frac{P_{\text{pri}}(z)P_{\text{gen}}(y|z)}{P_{\text{enc}}(z|y)} \right]$$

Gradient descent on the encoder parameters must take into account the fact that we are sampling from the encoder.

To handle this we sample noise ϵ from a fixed noise distribution and replace z with a deterministic function $z_{\text{enc}}(y, \epsilon)$

$$\text{enc}^*, \text{pri}^*, \text{gen}^* = \underset{\text{enc}, \text{pri}, \text{gen}}{\operatorname{argmin}} \quad E_{y, \epsilon, z = z_{\text{enc}}(y, \epsilon)} \left[-\ln \frac{P_{\text{pri}}(z)P_{\text{gen}}(y|z)}{P_{\text{enc}}(z|y)} \right]$$

The Re-Parameterization Trick

$$\text{enc}^*, \text{pri}^*, \text{gen}^* = \underset{\text{enc}, \text{pri}, \text{gen}}{\text{argmin}} \quad E_{y, \epsilon, z=z_{\text{enc}}(y, \epsilon)} \left[-\ln \frac{P_{\text{pri}}(z) P_{\text{gen}}(y|z)}{P_{\text{enc}}(z|y)} \right]$$

To get gradients we must have that $z_{\text{enc}}(y, \epsilon)$ is a smooth function of the encoder parameters and all probabilities must be a smooth function of z .

Commonly we use

$$\epsilon \sim \mathcal{N}(0, I), \quad z_{\text{enc}}(y, \epsilon) = \hat{z}_{\text{enc}}(y) + \sigma \epsilon$$

Optimizing the encoder is tricky for discrete z . Discrete z is handled effectively in EM algorithms and in VQ-VAEs.

EM is Alternating Optimization of the ELBO Loss

Expectation Maximimization (EM) applies in the (highly special) case where the exact posterior $P_{\text{pri,gen}}(z|y)$ is samplable and computable. EM alternates exact optimization of enc and the pair (pri, gen) in:

$$\text{VAE:} \quad \text{pri}^*, \text{gen}^* = \underset{\text{pri,gen}}{\operatorname{argmin}} \min_{\text{enc}} E_{y, z \sim P_{\text{enc}}(z|y)} - \ln \frac{P_{\text{pri,gen}}(z, y)}{P_{\text{enc}}(z|y)}$$

$$\text{EM:} \quad \text{pri}^{t+1}, \text{gen}^{t+1} = \underset{\text{pri,gen}}{\operatorname{argmin}} E_{y, z \sim P_{\text{pri}^t, \text{gen}^t}(z|y)} - \ln P_{\text{pri,gen}}(z, y)$$

Inference
(E Step)

$$P_{\text{enc}}(z|y) = P_{\text{pri}^t, \text{gen}^t}(z|y)$$

Update
(M Step)

Hold $P_{\text{enc}}(z|y)$ fixed

END