Lecture Notes VI – Auto-Encoders and Generative Models

Marina Meilă
mmp@uwaterloo.ca

With Thanks to Pascal Poupart & Gautam Kamath
Cheriton School of Computer Science
University of Waterloo

March 12, 2026

Autoencoders

Variational AutoEncoders (VAE)

Generative Adversarial Networks (GAN)

Diffusion models

Reading HTF Ch.: 11.3 Neural networks, Murphy Ch.: (16.5 neural nets), Bach Ch.: –, Deep Learning Book (Goodfellow, Bengio, Courville) 6.1-4, ResNet 7.6, ConvNet 9., Autoencoders 14.1, Dive Into Deep Learning 4.1-4.3.

Autoencoders

estion How to learn from data without outputs y?
This is unsupervised learning, not prediction

Idea Learn a low dimensional/sparse representation $h(x)$ of data $x \in \mathbb{R}^d$

$$h(x) \in \mathbb{R}^m, \text{ with } m < d \quad f(h(x)) \approx x! \tag{1}$$

- Optimize $L(x, f(h(x)))$

Variations

- If f linear, L_{LS}, then we "learn" PCA
- Denoising autoencoder
  - Add noise to x input, predict true x

$$\tilde{x} \sim C(\,|x), \quad \min L(x, f(h(\tilde{x}))). \tag{2}$$

- Sparse autoencoder

$$\min L(x, f(h(x)) + \Omega(h) \tag{3}$$

Ω is regularization that makes h sparse

-

Variational Autoencoders

Idea 1 Probabilistic encoder $x \to Enc_\phi(x) = p_\phi(Z|x)$

- Encoder outputs $\mu_\phi(x), \ln \sigma_\phi(x) \in \mathbb{R}^m$
- $p_\phi(Z|x) = \mathcal{N}(\mu_\phi(x), \mathrm{diag}\,\{\sigma^2_\phi(x)\})$
- encoding $z(x)$ is sampled $\sim p_\phi(Z|x)$

- Probabilistic Decoder $Dec_\theta(z) = p_\theta(z) \in [0,1]^d$
  - Assuming $x \in \{0,1\}^d$, $x_j \sim Bernoulli(p_{j,\theta})$ for $j = 1 : d$
- We also set $p(Z) = \mathcal{N}(0, I)$

Step 2 Now consider the joint probability of X, Z (Z is often called latent variable)

- We have $p_\theta(X,Z) = p(Z)p_\theta(X|Z)$. This expression can be computed because it contains the decoder and a gaussian.
- and $p_\theta(x,Z) = p_\theta(Z|x)p_\theta(x)$. From this we get the likelihood of data point x as

$$p_\theta(x) \;=\; \frac{p_\theta(x,Z)}{p_\theta(Z|x)}. \tag{4}$$

- The numerator is computable, but $p_\theta(Z|x)$ is not.

Idea 3.1 **Replace $p_\theta(Z|x)$ with a tractable distribution**, namely $p_\phi(Z|x)$. This is the first step of what is known as a **variational approximation**.

Idea 3.2: Variational lower bound

- Maximize **Likelihood** for single example $x \in \mathcal{D}$

$$\ln p_\theta(x) = \mathbb{E}_\phi[\ln p_\theta(x)] = \mathbb{E}_\phi\left[\ln \frac{p_\theta(x,Z)}{p_\theta(Z|x)}\right] \quad (5)$$

$$= \mathbb{E}_q\left[\ln \frac{p_\theta(x,z)}{p_\theta(z|x)} \cdot \frac{q_\phi(z|x)}{q_\phi(z|x)}\right] \quad (6)$$

$$= \mathbb{E}_\phi\left[\ln p_\theta(x,Z) - \ln q_\phi(Z|x)\right] + \mathbb{E}_\phi\left[\ln \frac{q_\phi(Z|x)}{p_\theta(Z|x)}\right] \quad (7)$$

$$(8)$$

The second term is the KL divergence $\mathrm{KL}(p_\phi(z|x) \,\|\, p_\theta(z|x))$ with $KL(q||p) = \int \ln \frac{q}{p}\, q\, dz \geq 0$

- Hence

$$\ln p_\theta(x) = \underbrace{\mathbb{E}_\phi\left[\ln p_\theta(x,z) - \ln q_\phi(z|x)\right]}_{\text{ELBO}} + \underbrace{\mathrm{KL}(q_\phi(z|x) \,\|\, p_\theta(z|x))}_{\geq 0} \quad (9)$$

- So far: Log-likelihood$(x) \geq ELBO(x)$ – we will maximize ELBO w.r.t. θ, ϕ
- ELBO = Evidence LOwer Bound is a **Variational Lower Bound**

Re-expressing the ELBO

1. Recall $p_\theta(x, Z) = \underbrace{p_\theta(x|Z)}_{\text{decoder}} \underbrace{p(Z)}_{N(0,I)}$
2. Replacing this in the ELBO

$$
\begin{aligned}
ELBO &= \mathbb{E}_\phi \left[\ln p_\theta(x, Z) - \ln p_\phi(Z|x)\right] && (10) \\
&= \mathbb{E}_\phi[\ln p_\theta(x|z)] - \mathbb{E}_\phi \left[\ln p_\phi(z|x) - \ln \mathcal{N}(0, I)\right] && (11) \\
&= \mathbb{E}_\phi[\ln p_\theta(x|z)] - KL(p_\phi(Z|x)||\mathcal{N}(0, I)) && (12)
\end{aligned}
$$

3. KL divergence between two Gaussian distributions has closed form

$$
KL(p_\phi(Z|x)||\mathcal{N}(0, I)) = \frac{1}{2}(\|\mu_\phi\|^2 + \|\sigma_\phi\|^2) - \sum_{j=1}^{m} \ln \sigma_{\phi,j} - \frac{1}{2}m \qquad (13)
$$

This second term of the ELBO only depends on ϕ. Thus it will not affect the maximization w.r.t. θ

4. The first term depends on both ϕ, θ. For this, we first approximate the integral by the Monte-Carlo method
   4.1 sample n_ϵ values $\epsilon \sim \mathcal{N}(0, I_m)$
   4.2 for each ϵ, calculate $z(\epsilon) = \mu_\phi(x) + \sigma_\phi(x)\epsilon$
   Exercise Show that $z \sim \mathcal{N}(\mu_\phi(x), \mathrm{diag}\,\{\sigma^2_\phi(x)\})$
   4.3 $\ln p(Z) = -\frac{1}{2}\|Z\|^2 -$constant
5. Hence, the first term is approximated by

$$\mathbb{E}_\phi[\ln p_\theta(x|z)] \;\approx\; \frac{1}{n_\epsilon}\sum_\epsilon \left[\ln p_\theta(x|\mu_\phi(x) + \sigma_\phi(x)\epsilon)\right] \tag{14}$$

6. Summary: we take a gradient step w.r.t ϕ involving both terms in (12), and w.r.t. θ involving only the first term.

Exercise Complete this derivation. Write the gradient of the log-likelihood of a single x w.r.t. ϕ and θ. What partial derivatives of p_ϕ, p_θ do you need to take? Exercise What kind of (simple) neural networks would you choose for the Encoder and Decoder? What should be $\phi_{\rm out}$ for these networks?

Generative models

- Generative models are models that learn a data distribution
- Given data $\mathcal{D} = \{x^1, \ldots x^n\}$ from an unknown distribution q
- We want to learn $p_\theta \approx q$ and from which we can sample
-
- Notation: $\tilde{x} \sim p_\theta$ is a synthetic (aka fake) sample

Autoencoders (Reconstruct X)	Generative models (Reconstruct $q(X)$)	
Autoencoders	**GAN**	heuristic
VAE	**Diffusion models**	probabilistic

Generative Adversarial Network

- Input $z \sim N(0, I_m)$
- **Generator Network** $p_\theta(z)$ is a probabilitistic decoder; outputs a distribution over X
- $\tilde{x} \sim p_\theta(z)$ the fake data
- **Discriminator Network** Disc is a classifier
- The desired $y = \text{Disc}(X)$ is $+1$ for $x \sim \mathcal{D}$ and -1 for $\tilde{x} \sim p_\theta$
- The input to Disc consists of fake data points and true data points with equal probability

- Training: Gen and Disc are trained simultanously, by backpropagation

- Issues
  - Mode collapse Gen generates only from regions of the data space. For example, it learns to generate only digits 1 and 2 instead of generating all 10
  - Vanishing gradients If Disc becomes very good while Gen is still poor, then every $\tilde{x}$ is recognized as fake. Hence Gen does not get information on how to improve, and this is reflected in small, erratic gradients w.r.t. θ
  - Exercise Could Disc be very good when Gen is good?

Diffusion Models

- Notation $q \equiv Q^{(0)}$ is the true data distribution (as usual, q is given by $\mathcal{D}$)
- The forward process is a probabilistic "encoder" network with $x^{(0)} \equiv x$ and $t = 1:T$
- The backward process is a probabilisitic "decoder" network with $x^{(0)} \equiv \tilde{x}$ and $t = T:0$
- Layers are denoted $1:T$ and indexed by t, and intermediate values are $x^{(t)}$ in both networks; $x^{(0:T)} \in \mathbb{R}^d$
- Output is $x^{(T)} \equiv Z \sim \mathcal{N}(0, I_d)$

$$X^{(0)} \to \cdots \to X^{(t)} \to X^{(t+1)} \to \cdots \to x^{(T)} \equiv z$$

Forward process (F)

$$q\left(x^{(t+1)} \mid x^{(t)}\right) = \mathcal{N}\left(\alpha_t x^{(t)}, \beta_t I\right) \quad (15)$$

$$x^{(t+1)} = \sqrt{\alpha_t}\, x^{(t)} + \sqrt{\beta_t}\, z^{(t)}, \qquad z^{(t)} \sim \mathcal{N}(0, I_d) \quad (16)$$

$$\alpha_t + \beta_t = 1 \quad \Rightarrow \quad \mathrm{Var}\left(x^{(t)}\right) = I_d \quad (17)$$

Backward process (B) $x^{(0)} \leftarrow x^{(1)} \leftarrow \cdots \leftarrow x^{(t)} \leftarrow x^{(T)}, \qquad x^{(T)} \sim \mathcal{N}(0, I)$

$$p_\theta\left(x^{(t)} \mid x^{(t+1)}\right) = \mathcal{N}\left(\mu_t(x^{(t+1)}), \sigma_t^2(x^{(t+1)})\right) \quad (18)$$

$$\mu_t, \sigma_t \leftarrow \theta_t \leftarrow x^{(t+1)} \quad (19)$$

Goal:

$$p\left(x^{(t+1)} \mid x^{(t)}\right)$$