Deep Learning

Lecture 5: Generative models

Chris G. Willcocks

Durham University

Lecture Overview

1. **Introduction**
   - definition
   - probability examples
2. **Density estimation**
   - maximum likelihood estimation
   - cumulative distribution sampling
   - histogram and kernel density estimators
   - problematic densities
3. **Divergence measures**
   - Kullback–Leibler divergence
   - cross entropy
   - optimal transport
4. **Generative networks**
   - definition
   - deep autoencoders
   - variational autoencoders
   - autoregressive approaches

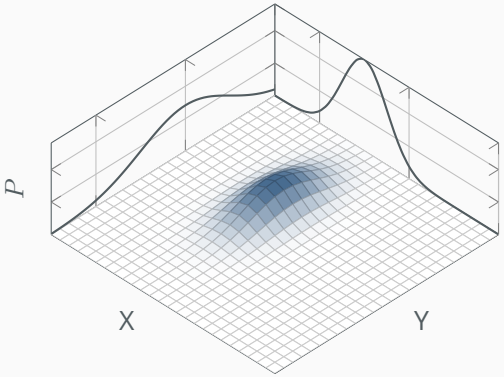
Recap: learning the data distribution

So what is it we want exactly?

- $P(Y|X)$ discriminative model (classification)
- $P(X|Y)$ conditional generative model
- $P(X,Y)$ generative model

We want to learn the probability density function of our data (natures distribution)

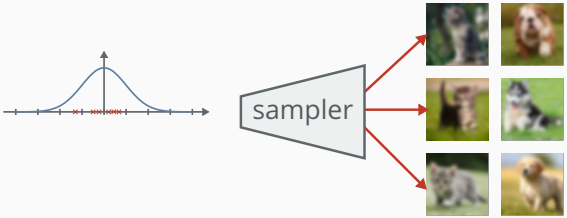
The data distribution $P(X,Y)$



Definition: Generative models learn a joint distribution over the entire dataset. They are mostly used for sampling applications or density estimation:

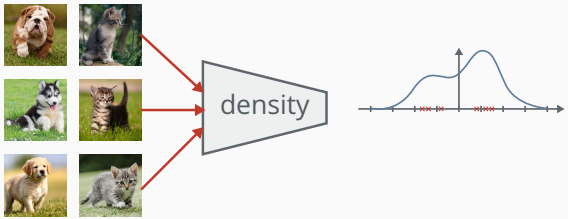
Inference: sampling

A generative model learns to fit a model distribution over observations so we can sample novel data from the model distribution, $\mathbf{x}_{\text{new}} \sim p_{\text{model}}(\mathbf{x})$



Inference: density estimation

Density estimation is estimating the probability of observations. Given a datapoint $\mathbf{x}$, what is the probability assigned by the model, $p_{\text{model}}(\mathbf{x})$?



Examples

Linguists

- What is the probability of a sentence? $P(\text{sentence})$
  - $P(\text{`the dog chased after the ball'})$
  - $P(\text{`printers eat avocados when sad'}) \approx 0$

Meteorologists

- What is the probability of whether it will rain? $P(\text{rain})$

Artists

- What is the probability of this image being a face? $P(\text{face})$

Musicians

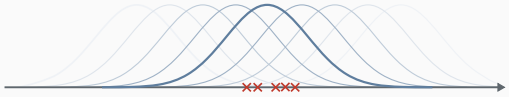
- What is the probability this sounds like Beethoven? $P(\text{Beethoven})$

Definition: maximum likelihood estimation

Maximum likelihood estimation (MLE) is a method for estimating the parameters of a probability distribution by maximizing a likelihood function, so that under the model the observed data is most probable

$$\begin{aligned}\theta^* &= \arg\max_\theta p_{\text{model}}(\mathbb{X};\theta) \\ &= \arg\max_\theta \prod_{i=1}^{n} p_{\text{model}}(\mathbf{x}^i;\theta) \\ &\approx \arg\max_\theta \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log p_{\text{model}}(\mathbf{x};\theta)],\end{aligned}$$

where $\mathbb{X} = \{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^n\}$ are from $p_{\text{data}}(\mathbf{x})$

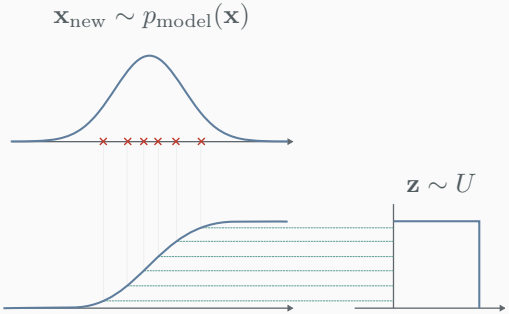


Example: cumulative distribution sampling

Given the CDF $F_X(\mathbf{x})$, the antiderivative of $f_X(\mathbf{x}) = p_{\text{model}}(\mathbf{x})$, e.g. where $F'(\mathbf{x}) = p_{\text{model}}(\mathbf{x})$

$$F_X(\mathbf{x}) = \int_{-\infty}^{\mathbf{x}} f_X(\mathbf{u})\, \mathrm{d}\mathbf{u}$$

we can sample new data by transforming random values $\mathbf{z}$ from the uniform distribution $\mathbf{z} \sim U$ via the inverse of the CDF $F_X^{-1}(\mathbf{z})$.



Definition: histogram density estimator

Histograms divide the space, e.g. $[0,1]^d$, into bins $B_1, B_2, ..., B_N$ and counts the elements inside the bins

$$\hat{p}_{\text{hist}(\mathbf{x})} = \sum_{j=1}^{n} \frac{\hat{\theta}_j}{h^d} I(\mathbf{x} \in B_j),$$

where $\hat{\theta}_j$ is the proportion of observations in the bin, scaled to integrate to one

$$\hat{\theta}_j = \frac{1}{n} \sum_{i=1}^{n} I(\mathbf{x}^i \in B_j)$$

Definition: kernel density estimator

Another popular approach for simple cases is to employ techniques known as kernel density estimators

$$\hat{p}_{\text{kde}}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h^d} K\left(\frac{||\mathbf{x} - \mathbf{x}^i||}{h} \right).$$

where at each point x, $\hat{p}_{\text{kde}}(x)$ is the average of the kernel function centered over the data points X_i, and h is the bandwidth

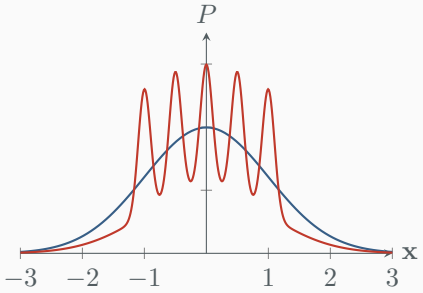
Inevitably there will be a bias/variance trade-off as we fit the data

Example: problematic densities

In a more complex setting, we may have a density such as the 'Bart Simpson' density, as Nando de Freitas likes to call it

$$p(x) = \frac{1}{2}\phi(x; 0, 1) + \frac{1}{10}\sum_{j=0}^{4}\phi(x; (j/2) - 1, 1/10)$$

where ϕ is the normal density with mean μ and standard deviation σ. This density cannot be sufficiently estimated with a normal distribution, as the result is over-smoothed (blue).



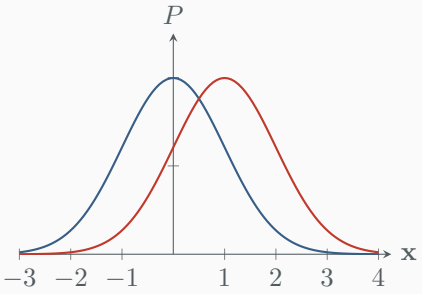
Definition: Kullback–Leibler divergence

We want to measure how different two distributions are. The Kullback-Leibler divergence (also called **relative entropy**) is one such measure that is asymmetric and non-negative:

$$D_{\mathrm{KL}}(p \,||\, q) = \int p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) \mathrm{d}\mathbf{x}$$

where $D_{\mathrm{KL}}(p \,||\, p) = 0$. Practically, the KL divergence is sensitive at the tails of the distribution.

Click to try on Desmos



Definition: Cross entropy

The KL divergence can be rewritten

$$\begin{aligned} D_{\mathrm{KL}}(p||q) &= \int p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) \mathrm{d}\mathbf{x} \\ &= \int p(\mathbf{x}) \log p(\mathbf{x}) \,\mathrm{d}\mathbf{x} - \int p(\mathbf{x}) \log q(\mathbf{x}) \,\mathrm{d}\mathbf{x} \\ &= -H(p) + H(p, q) \end{aligned}$$

where $H(p, q)$ is the cross entropy and $H(p) = H(p, p)$ is the regular entropy. This is minimizing the negative log-likelihood, the same as maximizing the likelihood

$$H(p, q) = -\int p(\mathbf{x}) \log q(\mathbf{x})$$

Definition: Wasserstein metric

It's worth mentioning an elegant metric between two distributions, which is from the optimal transport problem formalised by Gaspard Monge in 1781. If interested, watch Cédric Villani on YouTube discuss this topic and see examples in [1].

In 1D, the first Wasserstein **metric** (or earth mover's distance) can be written between the two CDF's F_1 and F_2 where

$$W(F_1, F_2) = \int |F_1(x) - F_2(x)| \,\mathrm{d}x$$

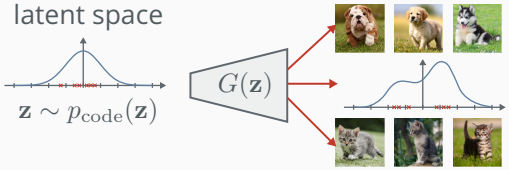
This is the minimal (optimal) cost from moving dirt between the two distributions whose CDF's are F_1 and F_2 respectively.

Generative networks definition

Definition: generative networks

The goal of generative networks is to take some simple distribution, like a normal distribution or a uniform distribution, and apply a non-linear transformation (e.g. a deep neural network) to obtain samples from $p_{\text{data}}(\mathbf{x})$

In 1D, we can say $G = F^{-1}_{\text{data}}(\mathbf{x})$ and sample $\mathbf{z} \sim U$, and similarly in ND — but assuming the determinant of the Jacobian and the inverse of G are computable, which is a large restriction. Ideally we want $\mathbf{z}$ in low dimensions



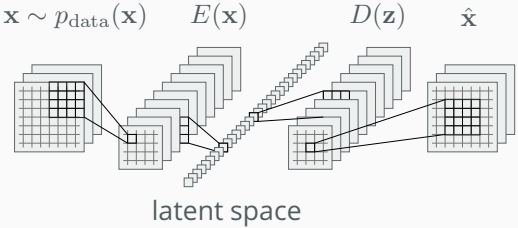
Definition: autoencoder

An autoencoder is a feedforward neural network that is trained to predict its inputs, thus learning an identity

$$\mathcal{L}_{\text{AE}} = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}}[\,\mathcal{L}(\mathbf{x}, D(E(\mathbf{x})))\,]$$

where $\mathcal{L}$ is a loss function such as mean squared error. The encoder function $E : \mathbb{R}^n \to \mathbb{R}^m$ (usually) compresses the dimensionality of the data $n \ll m$ to a latent encoding $\mathbf{z} = E(\mathbf{x})$, which is then recovered by the decoder $D : \mathbb{R}^m \to \mathbb{R}^n$, where $\hat{\mathbf{x}} = D(\mathbf{z})$. It's difficult to sample from the latent space, so its not really a generative model.

Link to Colab example



Definition: variational autoencoders

Variational autoencoders are generative models, as they impose a prior over the latent space $p(\mathbf{z})$, typically $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ which can be sampled from.

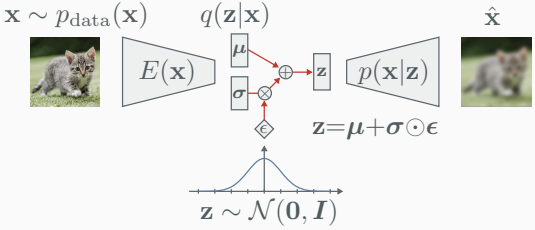
$$\mathbf{z} \sim E(\mathbf{x}) = q(\mathbf{z}|\mathbf{x}), \quad \hat{\mathbf{x}} \sim D(\mathbf{z}) = p(\mathbf{x}|\mathbf{z})$$

The VAE loss is the negated expected log-likelihood (the reconstruction error) and the prior regularization term:

$$\mathcal{L}_{\text{VAE}} = -\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}\right] = \mathcal{L}^{\text{pixel}}_{\text{recon}} + \mathcal{L}_{\text{prior}}$$

where

$$\mathcal{L}^{\text{pixel}}_{\text{recon}} = -\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})]$$
$$\mathcal{L}_{\text{prior}} = D_{\text{KL}}(q(\mathbf{z}|\mathbf{x}) \,||\, p(\mathbf{z}))$$

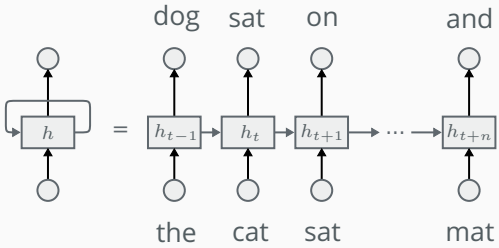


Definition: autoregressive networks

These models assume a natural sequential ordering of the data, then factorize the joint probabilities over symbols (for text) or pixels (for images) as the product of conditional probabilities

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_n \mid x_1, x_2, ..., x_{n-1})$$

Examples: text [2], audio [3], and images [4]



References I

[1] Gabriel Peyré, Marco Cuturi, et al. “Computational Optimal Transport: With Applications to Data Science”. In: Foundations and Trends® in Machine Learning 11.5-6 (2019), pp. 355–607.

[2] Alec Radford et al. “Language models are unsupervised multitask learners”. In: OpenAI Blog 1.8 (2019), p. 9.

[3] Aaron van den Oord et al. “Wavenet: A generative model for raw audio”. In: arXiv preprint arXiv:1609.03499 (2016).

[4] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. “Pixel recurrent neural networks”. In: arXiv preprint arXiv:1601.06759 (2016).

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. Available online, MIT press. 2016.

[6] Kevin P Murphy. Machine learning: a probabilistic perspective. MIT press, 2012.

[7] Ovidiu Calin. Deep learning architectures: a mathematical approach. Springer, 2020.