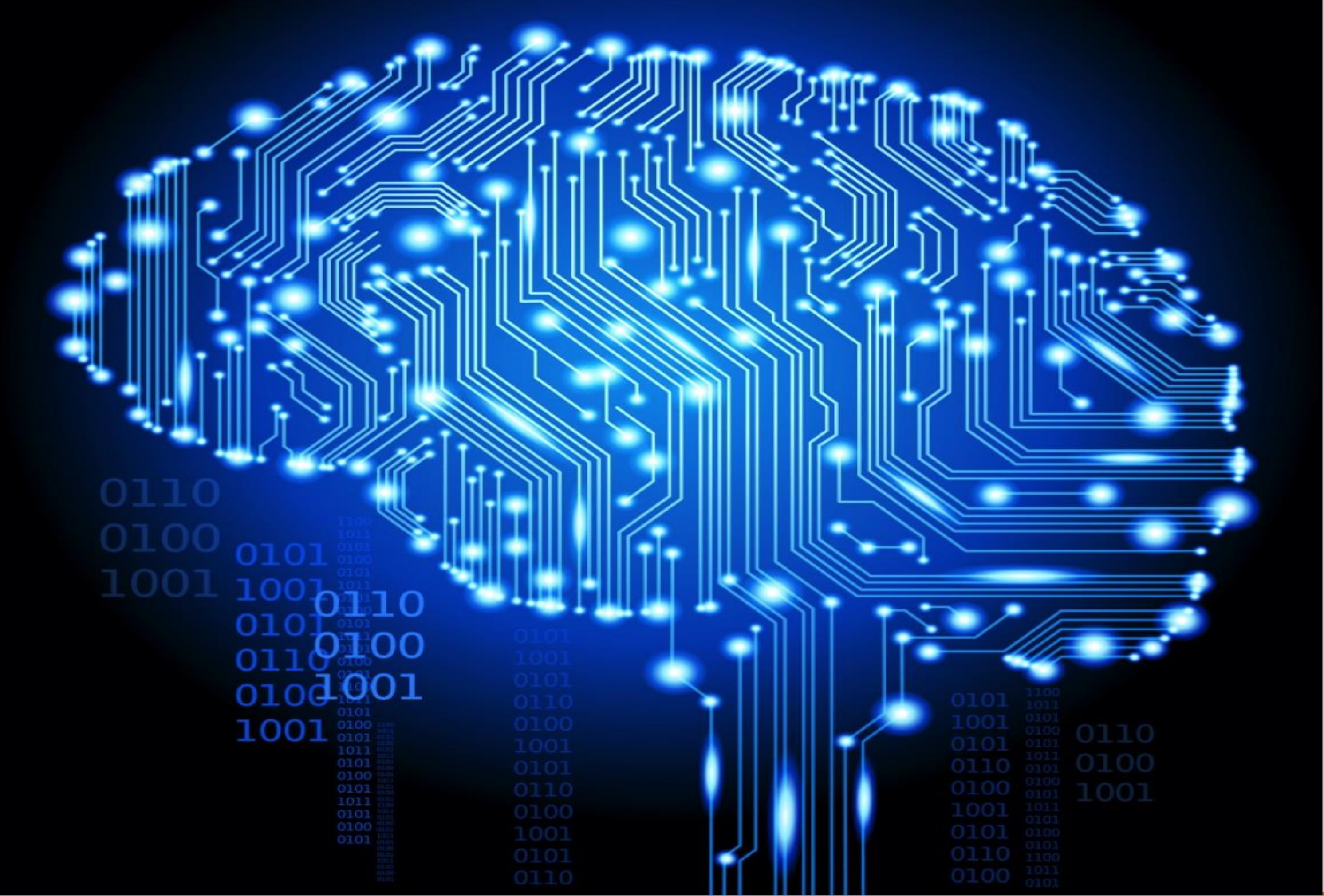


Lecture 8: Deep Generative & Energy Models

Efstratios Gavves

Lecture overview

- Generative models: motivation
- Energy-based models
- Hopfield networks
- Boltzmann machines
- Deep belief networks
- Modern energy models

Discriminative models: summary

- So far we have explored discriminative models mainly
- Given an individual input ***x*** predict
  - the correct label (classification)
  - the correct score (regression)
- Learning by maximizing the probability of individual classifications/regressions

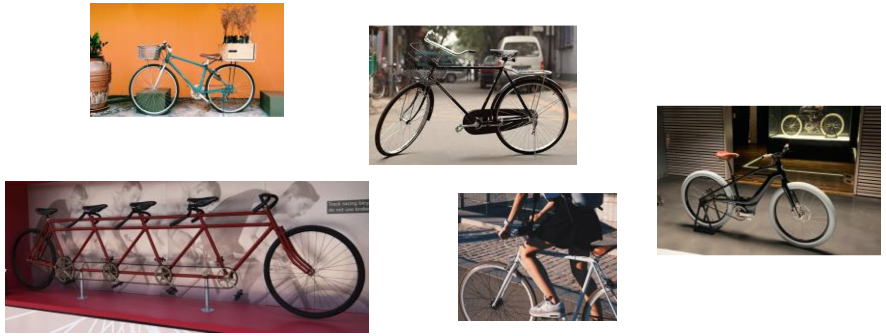
Prediction: "bicycle"

Prediction: "8.7" on IMDb

Generative models: main idea

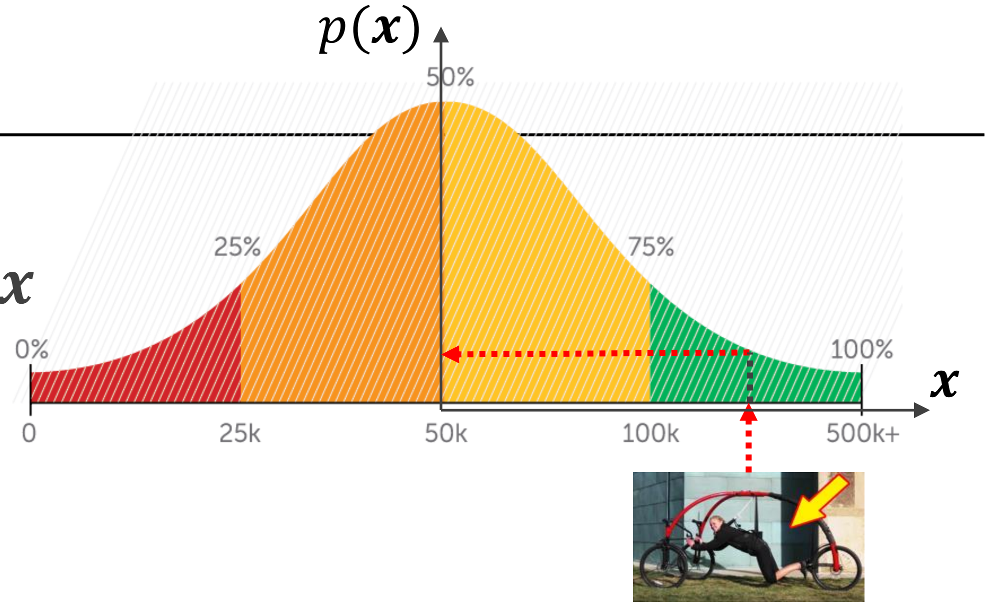
- Discriminative learning does not model data jointly
- Rephrasing: we want to know what is the distribution of data
- For instance: we want to know how likely is $\boldsymbol{x}_a$
  - Or if it is more likely than $\boldsymbol{x}_b$

Our observations

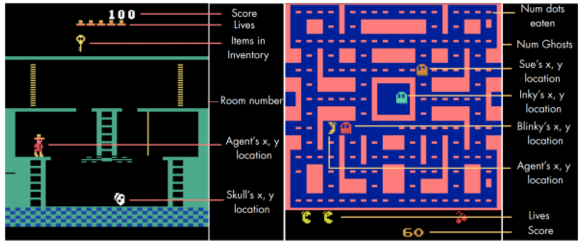


"What are the chances this is a bicycle"?

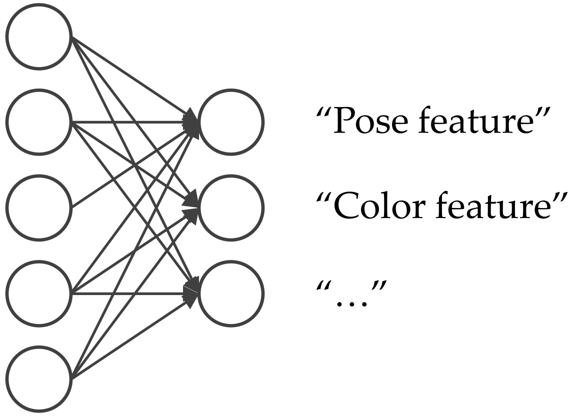
Why/when to learn a distribution?



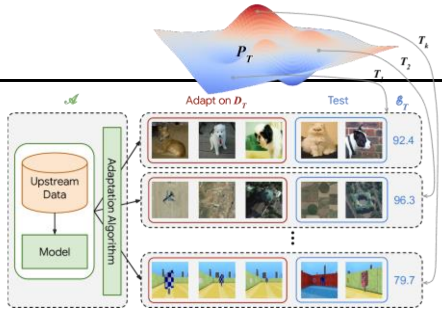
- Density estimation: estimate the probability of $\boldsymbol{x}$
- Sampling: generate new plausible $\boldsymbol{x}$
  - *E.g.*, model-based reinforcement learning



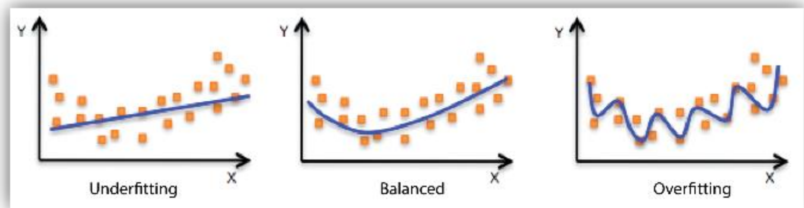
- Structure/representation learning: learn good features of $\boldsymbol{x}$ <u>unsupervised</u>



Why/when to learn a distribution?

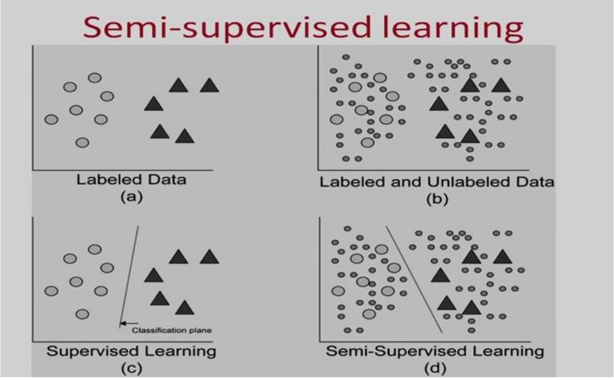


- Generative models to pretrain for downstream tasks



- Generative models to ensure generalization
  - *E.g.*, model-based reinforcement learning

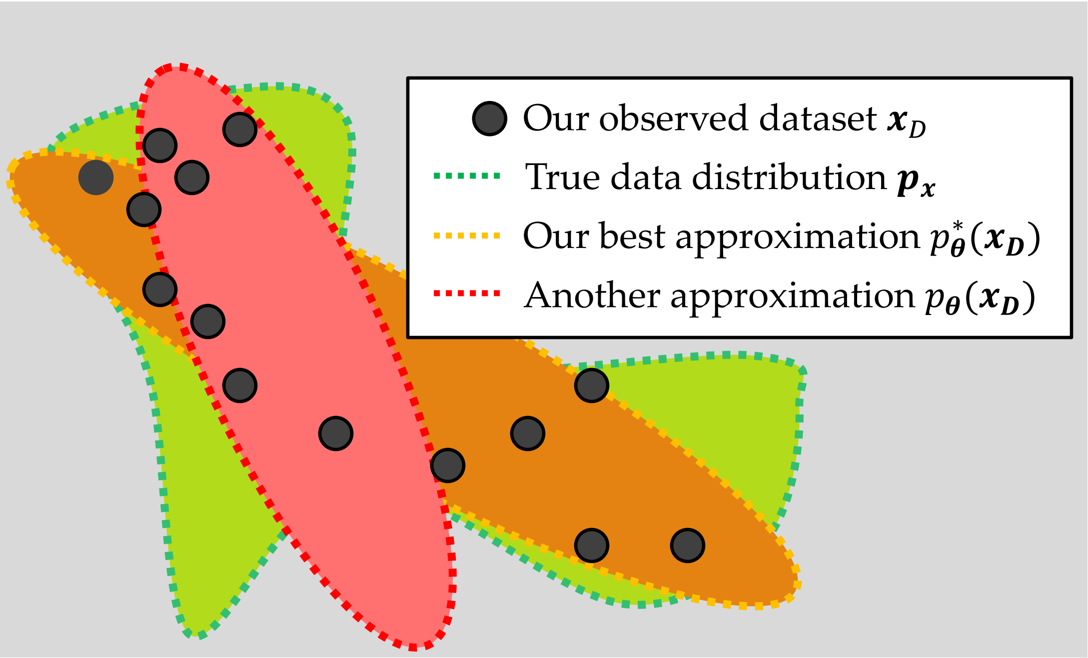
- Semi-supervised learning



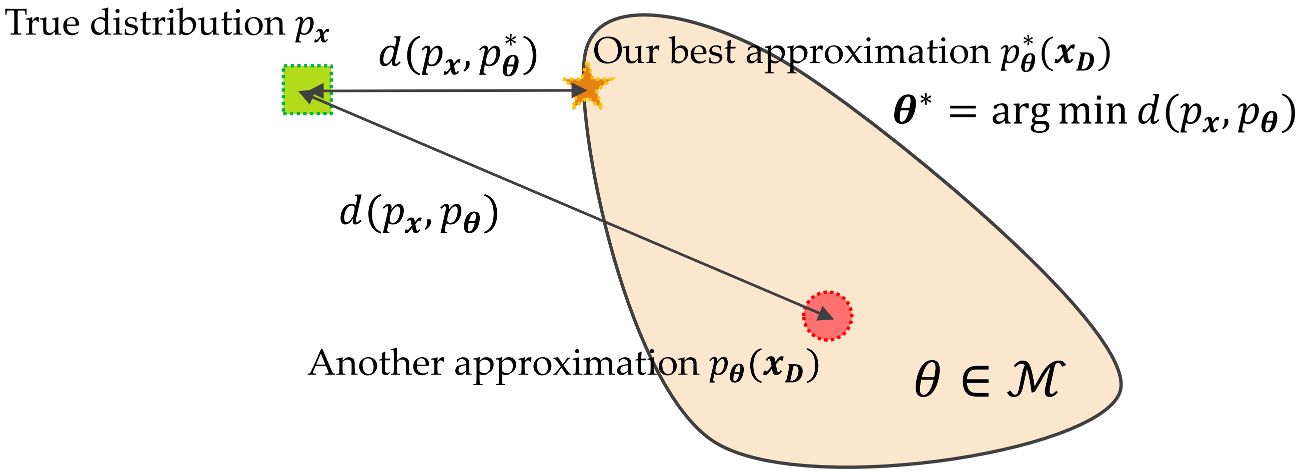
- Simulations

The world as a distribution

In data space: all possible data $\boldsymbol{x}$ (a.k.a. "The world")

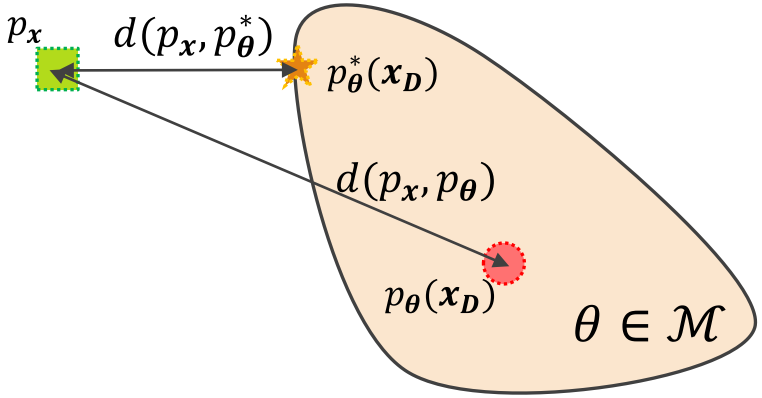


In the distribution space



Generative models: main challenges

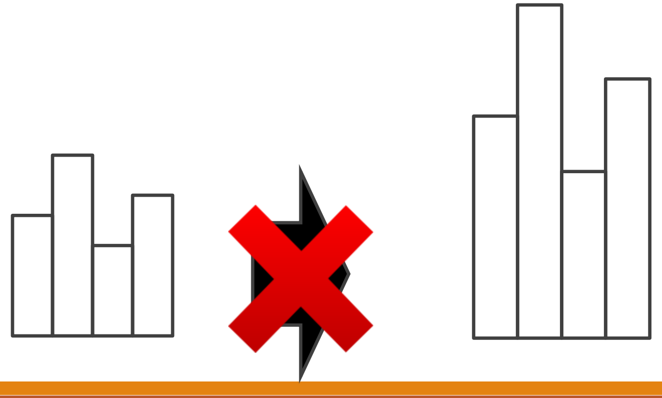
- We are interested in parametric models from a family of models $\mathcal{M}$



- How to pick the right family of models $\mathcal{M}$?
- How to know which θ from $\mathcal{M}$ is a good one?
- How to learn/optimize our models from family $\mathcal{M}$?

Properties for modelling distributions

- We want to learn distributions $p_{\boldsymbol{\theta}}(\boldsymbol{x})$
- Our model must therefore have the following properties
  - Non-negativity: $p_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq 0 \; \forall \; \boldsymbol{x}$
  - Probabilities of all events must sum up to 1: $\int_{\boldsymbol{x}} p_{\boldsymbol{\theta}}(\boldsymbol{x}) \, d\boldsymbol{x} = 1$
- Summing up to 1 (normalization) makes sure predictions improve relatively
  - Model cannot trivially get better scores by predicting higher numbers
  - The pie remains the same → model forced to make non-trivial improvements



Parameterizing models for distributions: non-negativity

- Our model must therefore have the following properties
  - Non-negativity: $p_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq 0 \forall\ \boldsymbol{x}$
  - Probabilities of all events must sum up to 1: $\int_{\boldsymbol{x}} p_{\boldsymbol{\theta}}(\boldsymbol{x})\, d\boldsymbol{x} = 1$

- Easy to obtain non-negativity
  - Consider: $g_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}^2(\boldsymbol{x})$ where $f_{\boldsymbol{\theta}}$ is a neural network
  - Or $g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \exp(f_{\boldsymbol{\theta}}(\boldsymbol{x}))$
  - But they do not sum up to 1

Energy-based models for distributions

- Normalize by the total volume of the function

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{\text{volume}(g_{\boldsymbol{\theta}})} g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{\int_{\boldsymbol{x}} g_{\boldsymbol{\theta}}(\boldsymbol{x}) d\boldsymbol{x}} g_{\boldsymbol{\theta}}(\boldsymbol{x})$$

  - In simple words, equivalent to normalizing $[3, 1, 4]$ as $\frac{1}{3+1+4}[3, 1, 4]$

- Examples
  - $g_{\boldsymbol{\theta}=(\boldsymbol{\mu},\boldsymbol{\sigma})}(\boldsymbol{x}) = \exp(-(\boldsymbol{x}-\boldsymbol{\mu})^2/2\boldsymbol{\sigma}^2) \Rightarrow \text{Volume}(g_{\boldsymbol{\theta}}) = \sqrt{2\pi\boldsymbol{\sigma}^2} \Rightarrow$ Gaussian
  - $g_{\boldsymbol{\theta}=\lambda}(\boldsymbol{x}) = \exp(-\lambda\boldsymbol{x}) \Rightarrow \text{Volume}(g_{\boldsymbol{\theta}}) = \frac{1}{\lambda} \Rightarrow$ Exponential

- Must find convenient $g_{\boldsymbol{\theta}}$ to be able to compute the integral analytically
  - Otherwise we cannot make sure of valid probabilities

Why is learning a distribution hard?

- The integrals mean that learning distributions becomes harder with scale
- Think of 300x400 color images with $[0, 256)$ color range
  - The number of possible images $\boldsymbol{x}$ is $256^{3 \cdot 300 \cdot 400}$
  - In principle must assign a probability to all of them

- While easy to *define* a family of models, we got a $\int_{\boldsymbol{x}} g_{\boldsymbol{\theta}}(\boldsymbol{x}) d\boldsymbol{x}$
  - Not always easy how to sample (needed for evaluating)
  - Not always easy how to optimize (needed for training)
  - Not always data efficient (long training times)
  - Not always sample efficient (many samples needed for accuracy)

Why/when not to learn a distribution?

- *"One should solve the [classification] problem directly and never solve a more general [and harder] problem as an intermediate step."*
  V. Vapnik, father of SVMs.
- Generative models to be preferred
  - when probabilities are important
  - when you got no human annotations and want to learn features
  - when you want to generalize to (many) downstream tasks
  - when the answer to your question is not: "more data"
- If you have a very specific classification task and lots of data
  - no need to make things complicated

A map of generative models