Introduction

22-808: Generative models
Sharif University of Technology
Fall 2025

Fatemeh Seyyedsalehi

Course info.

- Lecturer: Fatemeh Seyyedsalehi
  - Contact: fateme.ssalehi@gmail.com
- Head TA: Maryam Rezaie
  - Contact: ?
- Course website: On Quera - Github
  - Tentative schedule, lectures
  - Policies and rules
  - Discussions
  - HWs & solutions

Grading policy

- Mid-term exam: 4
- Final exam: 6
- Homework (5 practical and conceptual HWs): 10+1

Text books and related courses

- Books:
  - Bishop, Christopher M. and Hugh Bishop, Deep Learning: Foundations and Concepts, Springer
  - Murphy, Kevin P, Probabilistic Machine Learning: Advanced Topics, The MIT Press
  - Tomczak, Jakub M., Deep Generative Modeling, Springer
  - Koller D., Friedman N., Probabilistic Graphical Models, Principles and Techniques, The MIT Press

- Courses with similar topics from other institutions:
  - Stanford CS-236: Deep Generative Models
  - CMU18-789:Deep Generative Models
  - Washington CSE-599: Generative Models
  - Berkeley CS 294-158: Deep Unsupervised Learning

Introduction

- We should understand complex and unstructured phenomenon to be able to generate them

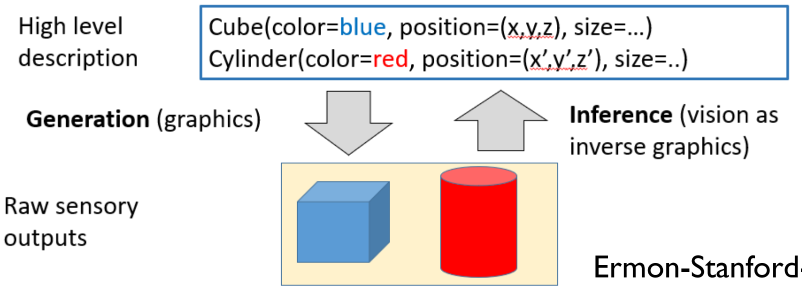
- Audio signals

- Natural images
- Natural languages

به نام آنکه جان را فکرت آموخت

Introduction

- Our tools to understand phenomenon is statistics
  - In contrast to rule based techniques

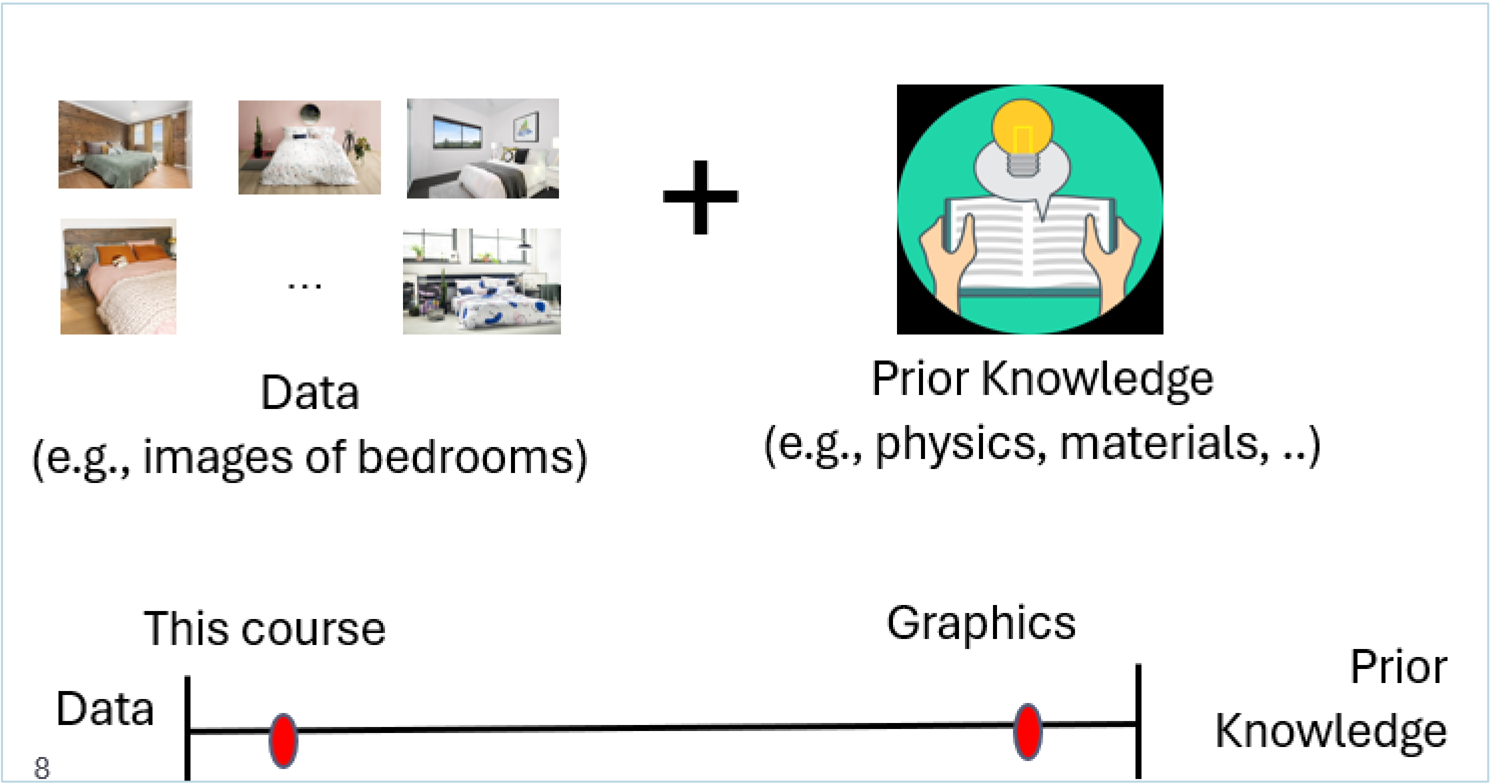


Ermon-Stanford-CS236

- We use probability distributions to describe any thing
  - Images
  - Sentences
  - Videos
  - Audios
  - ...

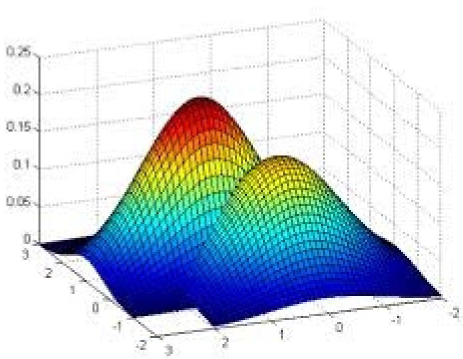
Statistical generative models

- Statistical generative models are learned from data
- Priors are always necessary, but there is a spectrum



Statistical generative models

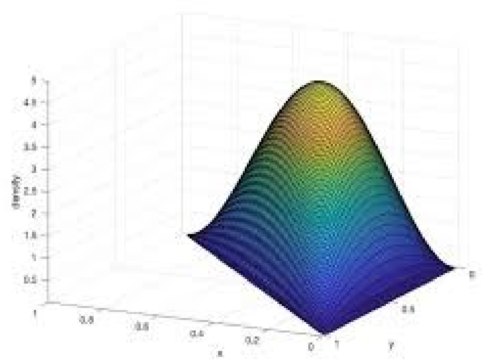
- A statistical generative model is a probability distribution $P(x)$
- It is generative because sampling from p(x) generates new images



$P_{\theta}(x)$

Statistical generative models

- We also can control the process of generation
  - Conditional generative models



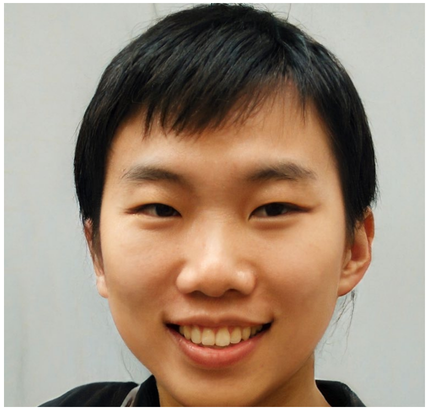
$P_{\theta}(x)$

Control signal

9

Generative model examples
Image generation

- This person does not exist!
  - https://thispersondoesnotexist.com/






Generative model examples
Image generation

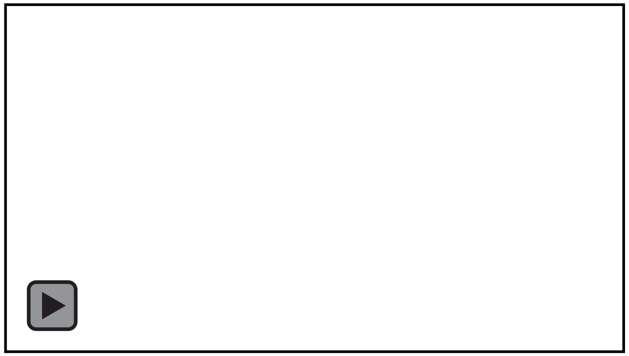
- OpenAI Dall-E
  - https://openai.com/index/dall-e/

- Prompt: “A photorealistic image of an astronaut riding a horse”

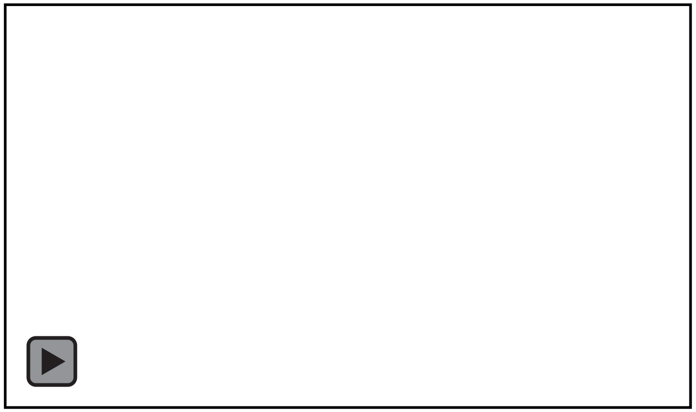
- Prompt: "A store front that has the word ‘OpenAI’ written on it”

Generative model examples
Video generation

- Prompt: “A couple sledding down a snowy hill on a tire roman chariot style”

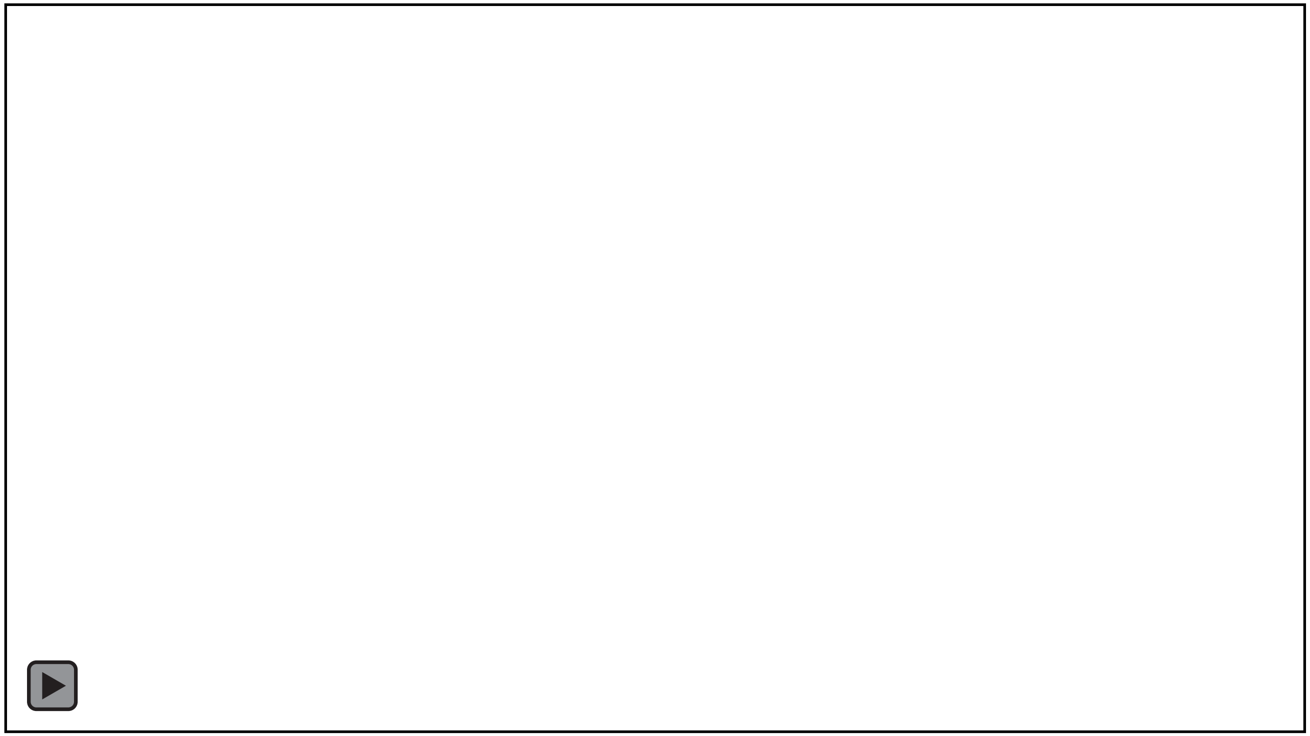


- Prompt: “Suddenly, the walls of the embankment broke and there was a huge flood”



Generative model examples
Video generation

- Deepfake
  - https://deepfakesweb.com/projects



Generative model examples
Code generation

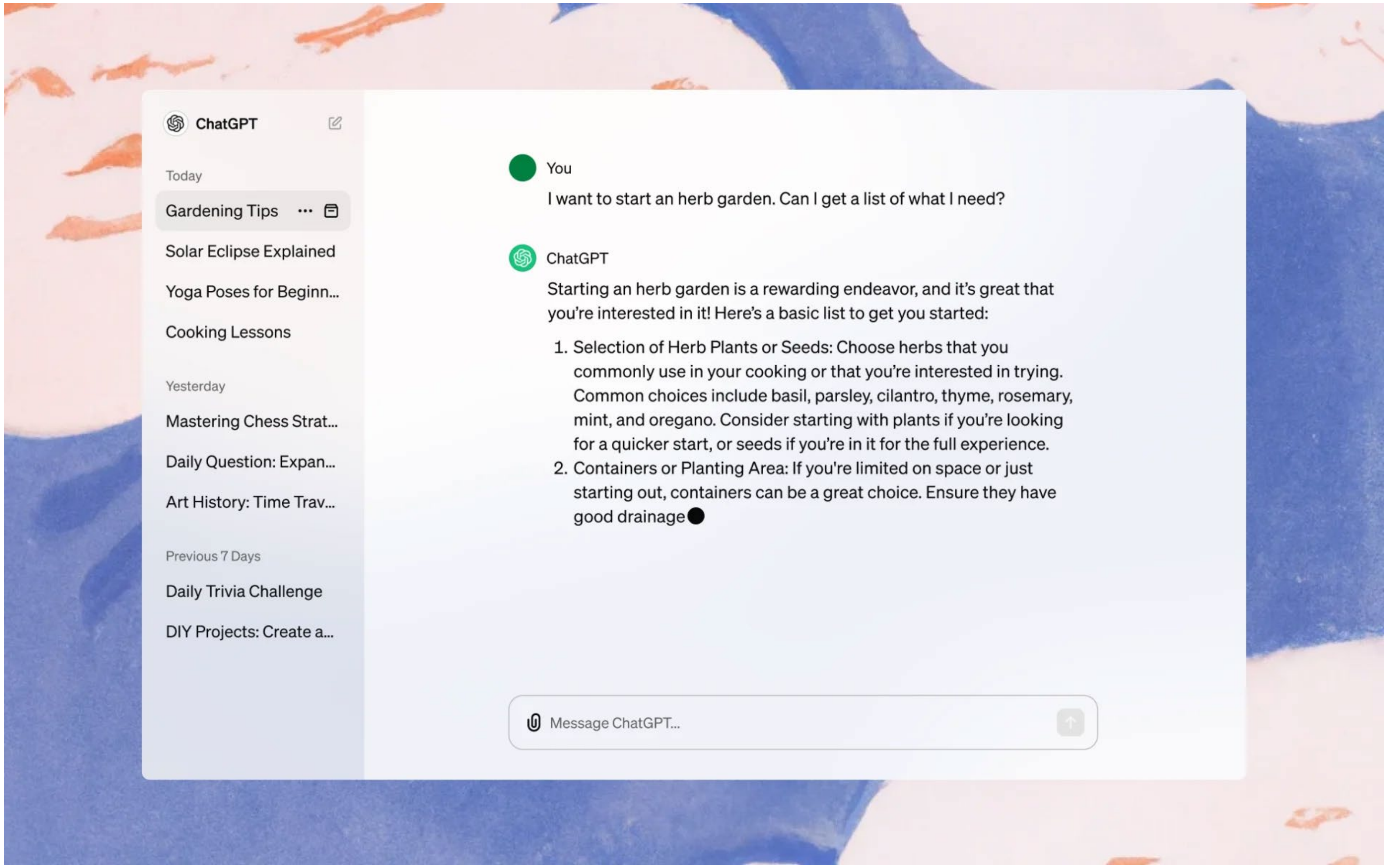
- OpenAI Codex
  - https://openai.com/index/openai-codex/

parse_expenses.py | write_sql.go | sentiment.ts | addresses.rb

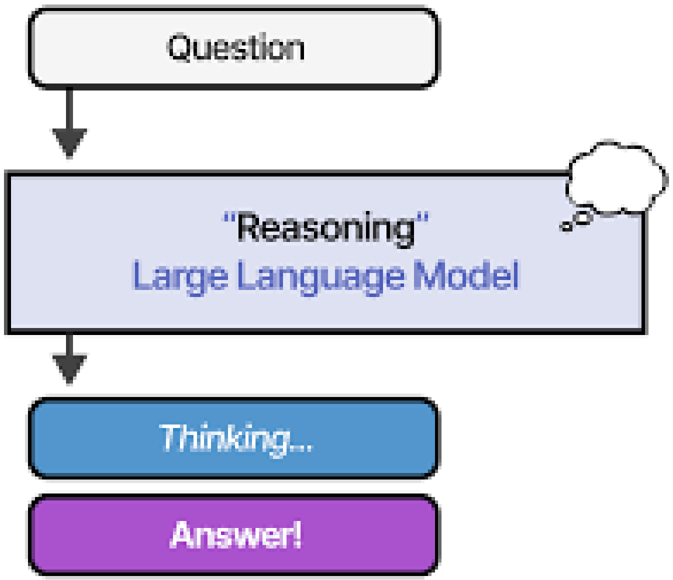
```python
import datetime

def parse_expenses(expenses_string):
    """Parse the list of expenses and return the list of triples (date, va
```

Generative model examples
Language generation

- OpenAI ChatGPT
  - https://openai.com/chatgpt/



Reasoning



Generative model examples
Speech generation

- مدل متن به گفتار فارسی

"برای موفقیت در درس یادگیری مولد، حضور در کلاس، مطالعه فردی و انجام به موقع تمرینات لازم است. این حوزه امروزه یکی از موضوعات به روز هم در دنیای تحقیقات و هم در حوزه صنعت به شمار می رود."

Foundation models

- The term of *foundation model* describes large ML models trained on a broad spectrum of generalized and unlabeled data
  - The ability of performing a wide variety of general tasks such as understanding language, generating text and images, and conversing in natural language

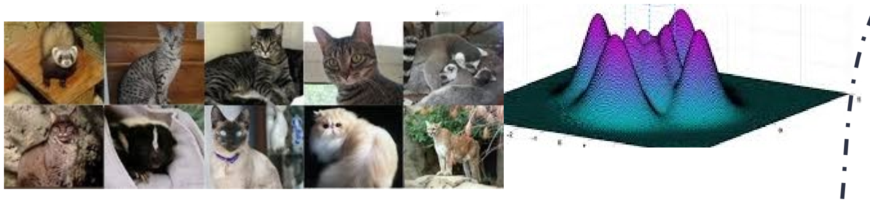
- They changed the way data scientists approach machine learning
  - Rather than developing from scratch, a foundation model can be used as a starting point to develop ML models that power new applications more quickly and cost-effectively.

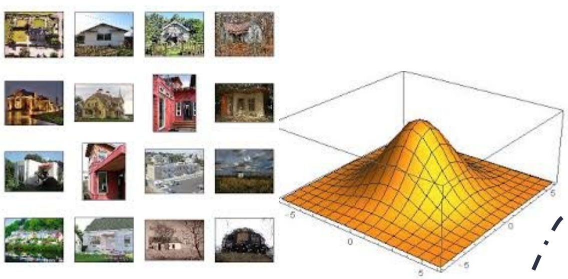
- A good paper:
  - "On the Opportunities and Risks of Foundation Models"

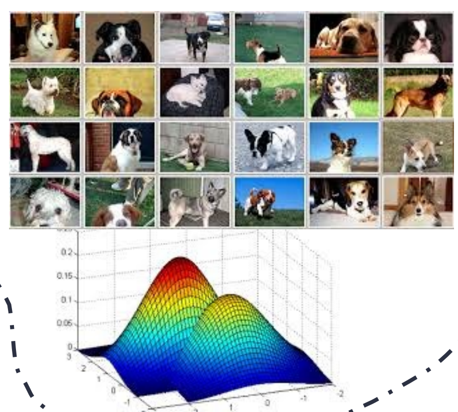
Course overview

- **(Probabilistic graphical models)** Directed (Bayesian networks) and undirected (Markov random fields))
  - Exact and approximate inference - Learning from complete and incomplete data
- **(Deep generative models)** Autoregressive Models
  - The NADE Framework
  - Text modeling, LSTM and Transformers, Intro. to large language models
- Variational Autoencoders
- Generative Adversarial Nets
  - f-GANs & Wasserstein GANs
- Generative Flow
- Energy-Based Models
  - Stein's Method and Score Matching
- Langevin Dynamics and Diffusions
- Flow Matching
- LLM and LMM
- LLM emergent abilities and reasoning

The generation problem







A family of distributions we can tackle with. Each instance of $\boldsymbol{\theta}$ represent a specific distribution.

$\boldsymbol{P_{\theta}(x)}$

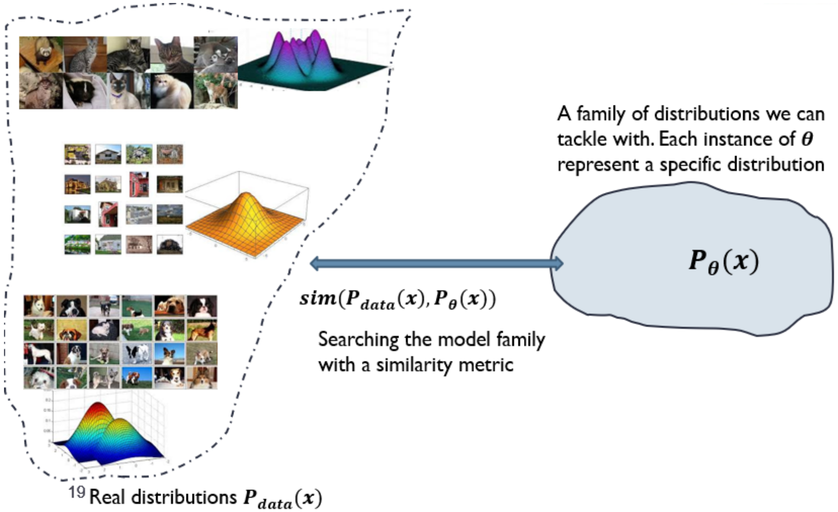
$\boldsymbol{sim(P_{data}(x), P_{\theta}(x))}$

Searching the model family with a similarity metric

Real distributions $\boldsymbol{P_{data}(x)}$

The generation problem

- Once we learn $\boldsymbol{P_\theta(x)}$,
  - We can **generate** $x \sim \boldsymbol{P_\theta(x)}$ that should look like a real sample
  - We can approximate the function (**density estimation** problem) the density of a sample, i.e. the value of $\boldsymbol{P_\theta(x)}$
    - Useful in anomaly detection
    - However, generation is usually easier than density estimation problem
  - We can discover features of the space in an unsupervised manner.



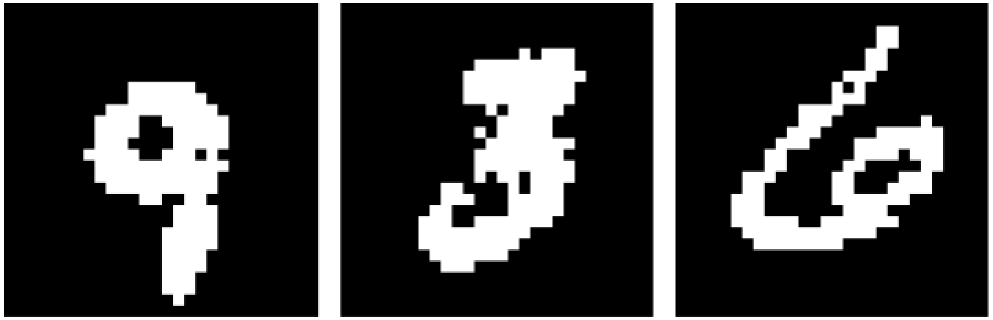
Representation of model space

- We should represent distribution spaces in a way we could tackle with
  - Optimization, generation, density estimation, inference, …
- Our approaches
  - Basic parametric distributions
  - Probabilistic graphical models
  - Deep neural networks
- Restricting to a parametric family of functions regularizes the problem.

Representation of model space

- Bernoulli distribution: (biased) coin flip
  - $D = \{Heads, Tails\}$
  - Specify $P(X = Heads) = p$. Then $P(X = Tails) = 1 - p$.
  - Write: $X \sim Ber(p)$
  - Sampling: flip a (biased) coin
- Categorical distribution: (biased) m-sided dice
  - $D = \{1, \cdots, m\}$
  - Specify $P(Y = i) = p_i$, such that $\sum p_i = 1$
  - Write: $Y \sim Cat(p_1, \cdots, p_m)$
  - Sampling: roll a (biased) die

Representation of model space



- Suppose $X_1, \ldots, X_n$ are binary (Bernoulli) random variables, i.e., $\text{Val}(X_i) = \{0, 1\} = \{\text{Black}, \text{White}\}$.
- How many possible images (states)?

$$\underbrace{2 \times 2 \times \cdots \times 2}_{n \text{ times}} = 2^n$$

- Sampling from $p(x_1, \ldots, x_n)$ generates an image
- How many parameters to specify the joint distribution $p(x_1, \ldots, x_n)$ over n binary pixels?

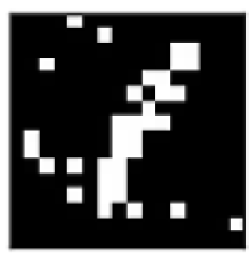
$$2^n - 1$$

Representation of model space

- If $X_1, \ldots, X_n$ are independent, then

$$p(x_1, \ldots, x_n) = p(x_1)p(x_2)\cdots p(x_n)$$

- How many possible states? 2^n
- How many parameters to specify the joint distribution $p(x_1, \ldots, x_n)$?
  - How many to specify the marginal distribution $p(x_1)$? 1
- 2^n **entries can be described by just** n **numbers** (if $|\mathrm{Val}(X_i)| = 2$)!
- Independence assumption is too strong. Model not likely to be useful
  - For example, each pixel chosen independently when we sample from it.



Representation of model space

- Using Chain Rule

$$p(x_1, \ldots, x_n) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \cdots p(x_n \mid x_1, \cdots, x_{n-1})$$

- How many parameters? $1 + 2 + \cdots + 2^{n-1} = 2^n - 1$
  - $p(x_1)$ requires 1 parameter
  - $p(x_2 \mid x_1 = 0)$ requires 1 parameter, $p(x_2 \mid x_1 = 1)$ requires 1 parameter Total 2 parameters.
  - $\cdots$
- $2^n - 1$ is still exponential, chain rule does not buy us anything.
- Now suppose $X_{i+1} \perp X_1, \ldots, X_{i-1} \mid X_i$, then

$$\begin{aligned} p(x_1, \ldots, x_n) &= p(x_1)p(x_2 \mid x_1)p(x_3 \mid \cancel{x_1}, x_2) \cdots p(x_n \mid \cancel{x_1, \cdots}, x_{n-1}) \\ &= p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2) \cdots p(x_n \mid x_{n-1}) \end{aligned}$$

- How many parameters? $2n - 1$. Exponential reduction!

Representation of model space

- By factorizing joint distributions with independency assumption,
  - We **assume a structure for the problem domain**
  - We regularize the problem
  - We simplify the hypothesis (distributions) space
- Probabilistic graphical models
  - A way to represent a factorized joint distribution over a system of random variables with independency assumptions
  - We will introduce them first
- Deep neural network
  - More complicated and descriptive tool for representation of a model family.

Next Session

- Probabilistic graphical models
  - Directed (Bayesian networks)
  - Undirected (Markov random fields)