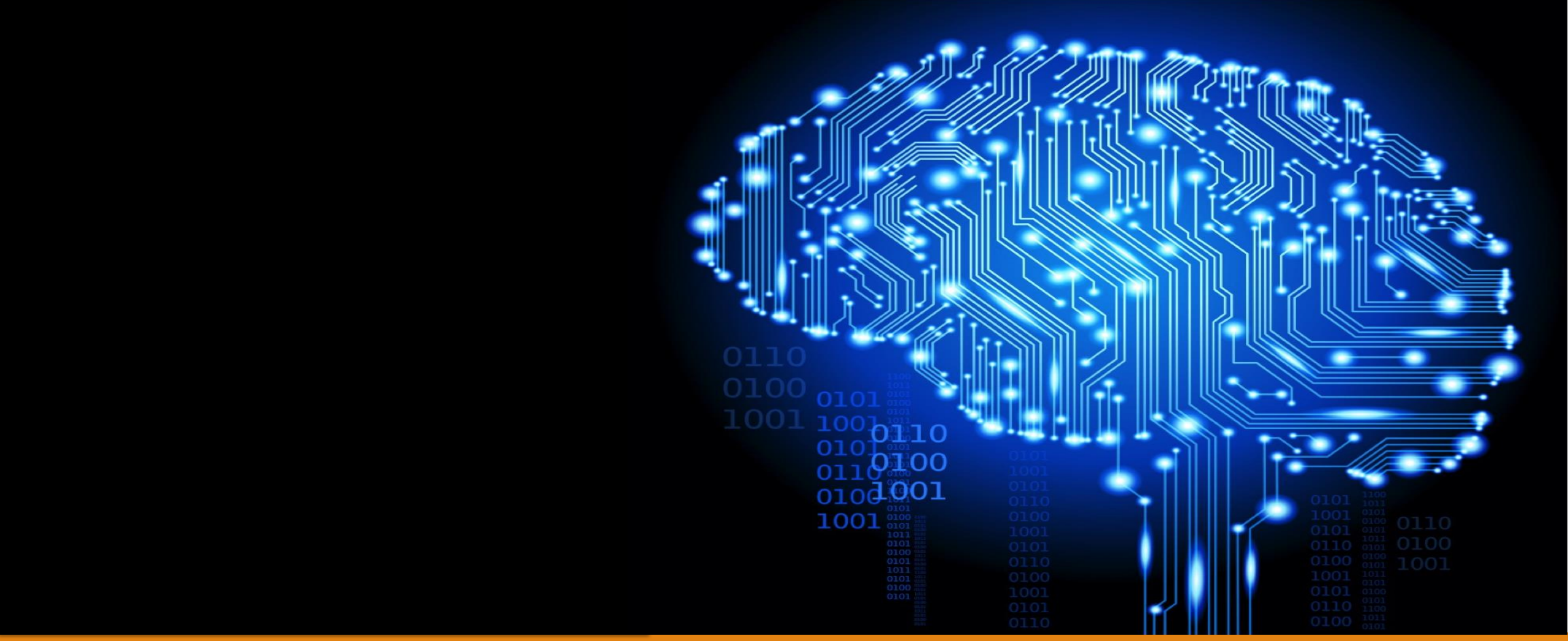


Lecture 12: Deep Stochastic Estimation

Efstratios Gavves

Lecture overview

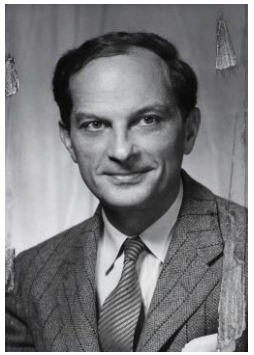
- Monte Carlo simulation
- Stochastic gradients
- MC gradient estimators
- Bias and variance in gradients

How it started

John von Neumann

⇔

Stanislav Ulam

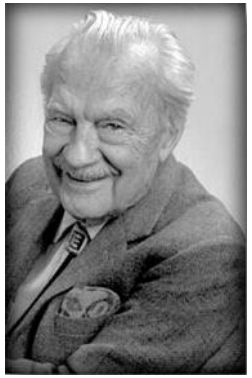


⇔

Manhattan project

⇕

Nicholas Metropolis

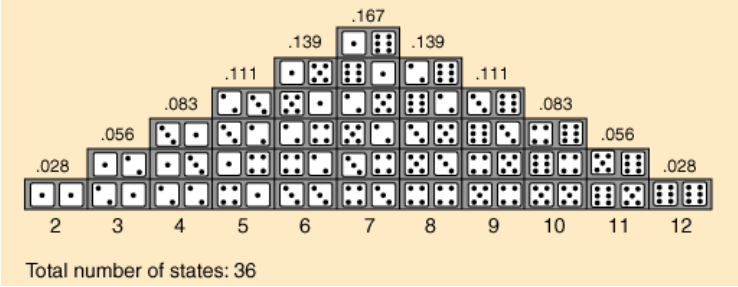


Applications

- High-energy Physics
- Finance
- All sort of simulations
- Machine Learning
- And of course Deep Learning

Motivation

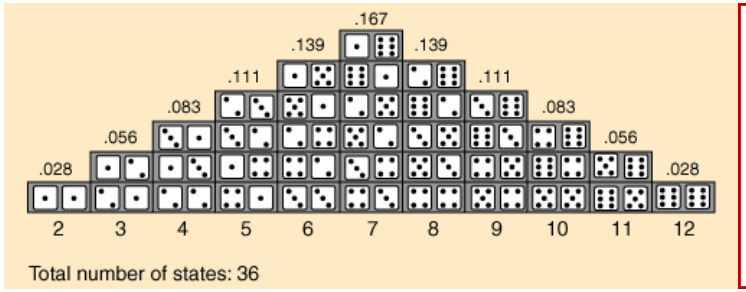
- We are often interested to compute quantities (statistics) on random variables
  - The average response to a drug
  - Or the probability of a particular sum when throwing two dice
  - Or the average reconstructions in my VAE given an input
- These statistics often intractable to compute
  - Cannot derive a perfect drug response model (too complex)
  - Cannot enumerate all possible dice combinations (too lazy)
  - Computationally infeasible (intractable integrals)

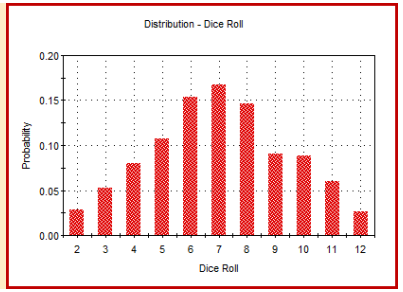


https://www.goldsim.com/Web/Introduction/MonteCarlo/

Monte Carlo integration

- Use random sampling instead of analytical computation
  - A single random sample might not be enough
  - Many random samples can give us a reliable quantification
- *E.g,* by throwing dice many times we can obtain a histogram of probabilities for each possible sum
  - If we throw dice once, the histogram will be very wrong (just a single bar)
  - But if we repeat hundreds of times and average, we are gonna get very close





https://www.goldsim.com/Web/Introduction/MonteCarlo/

Monte Carlo integration

- More formally, in MC integration we treat inputs x as RVs with pdf $p(x)$
  - Our desired statistic y is the output and integrate over all possible x

$$y = \int_x f(x)p(x)dx$$

- This integral is equivalent to an expectation

$$y = \mathbb{E}_{x\sim p(x)}[f(x)] = \int_x f(x)p(x)dx$$

- This is an expectation (integral) we can approximate it by random sampling and summation

$$y = \mathbb{E}_{x\sim p(x)}[f(x)] \approx \frac{1}{n}\sum_i f(x_i) = \hat{y}, \text{where } x_i \text{ is sampled from } p(x)$$

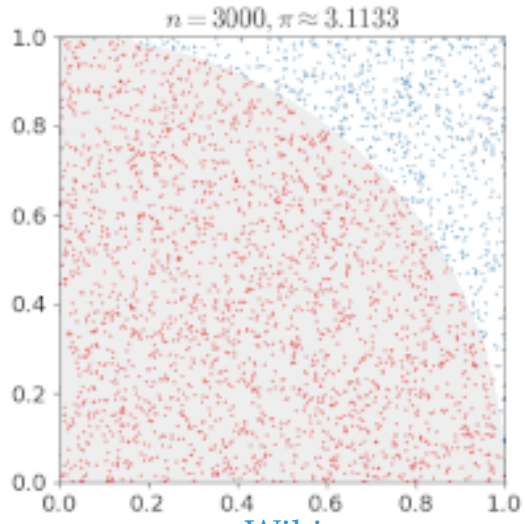
- $\hat{y}$ is an estimator because it only approximately estimates the value of y

Toy example: estimating π

- One can estimate the value of π numerically
  - Only the upper right quadrant suffices
- We count points x_c inside the circle (distance < 1 from (0,0) – red area)
  - And points x_s in the square (red and blue area)
  - Our estimator $\hat{y} = \frac{x_c}{x_s}$ estimates circle quadrant area over square area

$$\frac{\frac{1}{4}\pi r^2}{r^2} = \frac{\pi}{4}$$

  - Thus, with our estimator estimates $\hat{y} \approx \frac{\pi}{4} \Rightarrow \pi \approx \hat{\pi} = 4\hat{y}$
- If we repeat another time this experiment
  - We get a different $\hat{\pi}$



Wiki

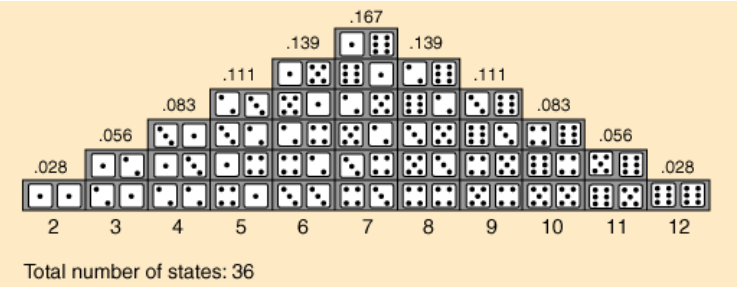
Estimator mean and variance

$$y = \mathbb{E}_{x \sim p(x)}[f(x)] \approx \frac{1}{n}\sum_i f(x_i) = \hat{y}, \text{where } x_k \text{ is sampled from } p(x)$$

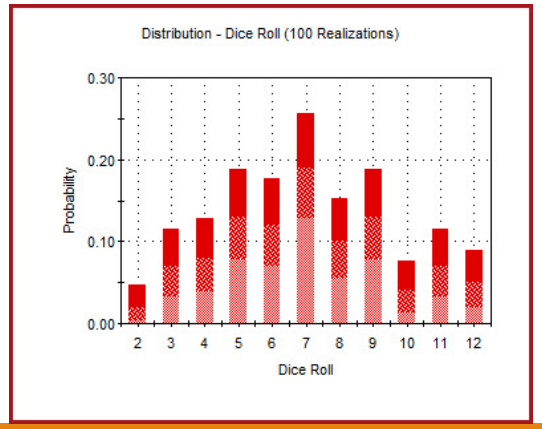
- Our estimator is itself a random variable

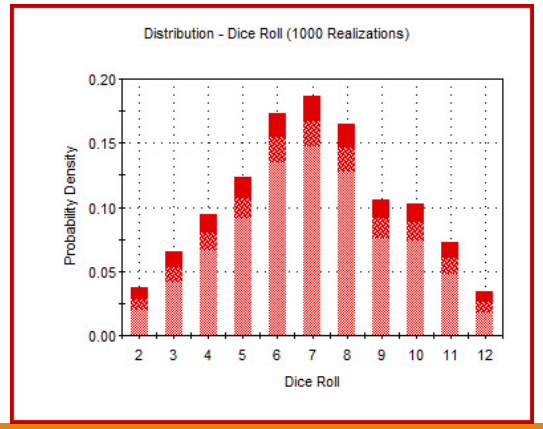
  → It has its own mean $\mu_{\hat{y}} = \mathbb{E}[\hat{y}]$ and variance $\mathrm{Var}[\hat{y}] = \mathbb{E}[(\hat{y} - \mu_{\hat{y}})^2]$

- The higher the variance, the more the estimation fluctuates after every new experiment



https://www.goldsim.com/Web/Introduction/MonteCarlo/





Estimator bias

- An estimator is unbiased it in expectation it matches the true statistic

$$\mathbb{E}[\hat{y}] = y$$

- Otherwise, biased with bias

$$\text{bias} = \mathbb{E}[\hat{y}] - y$$

- Better to have unbiased estimators
  - Although in cases a bit of bias is ok
  - Trade tractability for less accurate solutions (than what could be)
- The MC estimators are unbiased due to law of large numbers
  - "As the number of identically distributed, randomly generated variables increases, their sample mean (average) approaches their theoretical mean."

Standard error of MC estimator

- The MC estimator is a sample mean

$$\mathbb{E}_{x\sim p(x)}[f(x)] \approx \frac{1}{n}\sum_i f(x_i)$$

- The standard error of a sample mean is

$$\sigma_{\hat{f}} = \frac{\sigma}{\sqrt{n}}$$

- The more samples we take the less the estimator deviates
  - But the deviation reduces only as $\sqrt{n}$
  - With 4x more samples we only improve our error 2x

To sum up

- If we want to compute a quantity y
  - that we can express it as an integral of a function f over a probability space x
  - that has a known and easy to sample pdf $p(x)$
  - we can replace the exact but intractable computation with a tractable MC estimator

$$y = \mathbb{E}_{x \sim p(x)}[f(x)] \approx \frac{1}{n} \sum_i f(x_i), x_i \sim p(x)$$

- If we can't translate the quantity as such an integral, we can't estimate it
  - For instance, we cannot use MC on the following because neither the $\log p(\boldsymbol{x}|\boldsymbol{z})$ nor the $\nabla_\varphi q_\varphi(\boldsymbol{z}|\boldsymbol{x})$ are probability densities

$$\nabla_\varphi \mathbb{E}_{\boldsymbol{z} \sim q_\varphi(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})] = \int_z \log p(\boldsymbol{x}|\boldsymbol{z}) \nabla_\varphi q_\varphi(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z}$$

Why do we care?

- In Deep Learning many computations are intractable
  - Complex integrals that cannot be solved analytically
  - Extremely expensive sums, e.g., summing over all 2^{50} possible binary latent vectors z to obtain the marginal likelihood $p(\boldsymbol{x}) = \sum_z p(\boldsymbol{x}, \boldsymbol{z})$
  - We can make many of these computations tractable with MC estimators
- Examples of MC estimation
  - Stochastic gradient descent can be seen as an MC estimator
  - Sampling from a VAE is an MC estimator
  - And many other operations involving integrations,
    - generative models
    - gradient estimation
    - …