Stochastic gradients

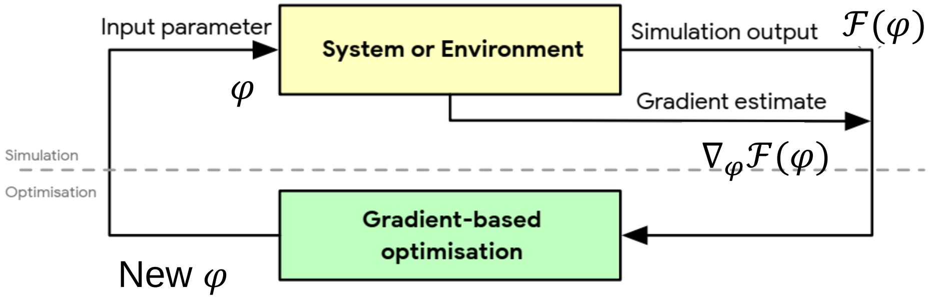

Figure 1: Stochastic optimisation loop comprising a simulation phase and an optimisation phase. The simulation phase produces a simulation of the stochastic system or interaction with the environment, as well as unbiased estimators of the gradient.

[Monte Carlo Gradient Estimation in Machine Learning]

Stochastic learning

- We have a general probabilistic objective

$$\mathcal{F}(\varphi) = \int p_\varphi(\boldsymbol{x}) f_\theta(\boldsymbol{x}) d\boldsymbol{x}$$

  - $p_\varphi(x)$: continuous probability distribution differentiable w.r.t. φ
  - $f_\theta(x)$: the structured cost with structural parameters θ
- Learning means we want to optimize $\mathcal{F}$ w.r.t. φ (and θ)
- The learning objective already looks like an MC estimator

MC learning ⇔ Stochastic gradient estimation

- To optimize the learning objective we must take gradients $\frac{d}{d\varphi}\mathcal{F}(\varphi)$
- The learning objective is stochastic → the gradients are stochastic

$$\frac{d}{d\varphi}\mathcal{F}(\varphi) = \nabla_{\varphi}\mathbb{E}_{\boldsymbol{x}\sim p_{\varphi}(\boldsymbol{x})}[f_{\theta}(\boldsymbol{x})]$$

- Except for simple cases, the stochastic gradients cannot be computed analytically
- We must resort to MC estimation instead
  - Although now we do not prescribe a specific pdf $p(\boldsymbol{x})$
  - We prescribe a family of $p_{\varphi}(\boldsymbol{x})$ and learn the best possible φ in the process

Challenges

$$\eta = \nabla_\varphi \mathcal{F}(\varphi) = \nabla_\varphi \mathbb{E}_{\boldsymbol{x} \sim p_\varphi(\boldsymbol{x})}[f_\theta(\boldsymbol{x})]$$

- $\boldsymbol{x}$ is typically high dimensional
- The parameters φ are often in the order of thousands
- The cost function is often not differentiable or even unknown
- That is, the expectation (integral) is often intractable
  - We must estimate it instead, with MC integration

Desired properties of MC estimators for gradients

- Consistency
  - When sampling more samples the estimator $\hat{y}$ should get closer to the true y
- Unbiasedness
  - Guarantees convergence of stochastic optimization
- Low variance
  - Few samples should suffice
  - Less jiggling → gradient updates in consistent direction → more efficient learning
- Computational efficiency
  - Should be easy to sample and estimate

Stochastic gradients: A pipeline

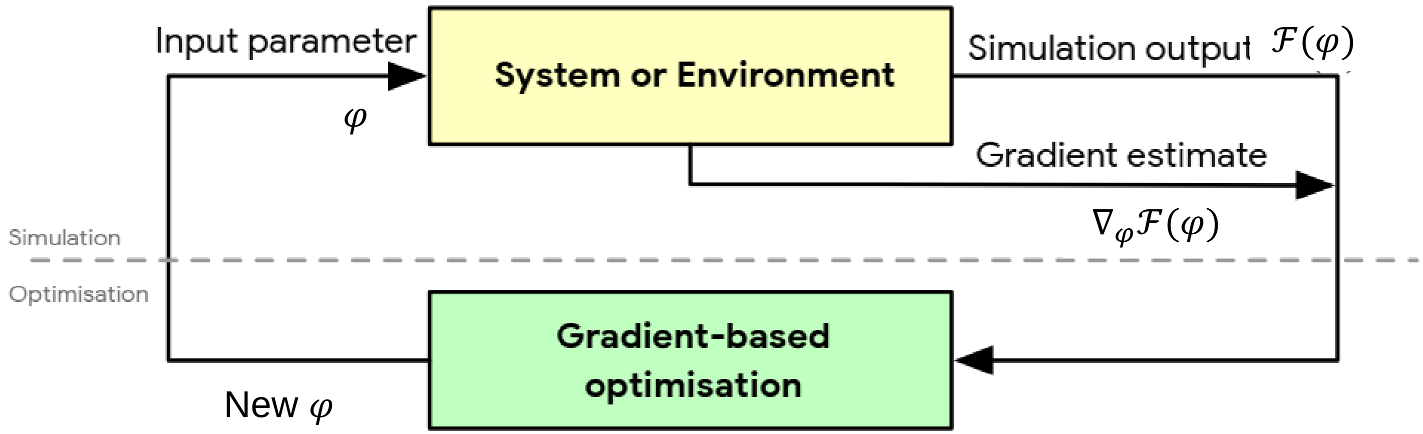

Figure 1: Stochastic optimisation loop comprising a simulation phase and an optimisation phase. The simulation phase produces a simulation of the stochastic system or interaction with the environment, as well as unbiased estimators of the gradient.

Monte Carlo Gradient Estimation in Machine Learning

Applications of stochastic gradients

- Variational inference

$$\nabla_\varphi \mathbb{E}_{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\boldsymbol{x}|\boldsymbol{z}) - \log \frac{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})}\right]$$

- Reinforcement learning

$$\nabla_\varphi \mathbb{E}_{p_\varphi(\boldsymbol{\tau})}\left[\sum_t \gamma_t\, r(\boldsymbol{s}_t, \boldsymbol{a}_t)\right]$$

  - Where $\boldsymbol{\tau} = (\boldsymbol{s}_1, \boldsymbol{a}_1, \boldsymbol{s}_2, \dots)$ are trajectories over time t
  - γ_t are discount factors and r is the reward
- Outside ML and DL
  - Sensitivity analysis
  - Discrete event systems and queuing theory
  - Experimental design

Monte Carlo Gradient Estimation in Machine Learning