GT Deep Learning: 03

Vincent Perrollaz

Institut Denis Poisson

17 Février 2026

Setup

1D Gradient Descent

General Gradient Descent

Improved Algorithms

Stochastic Gradient Descent

Summary

Hyperparameters network architecture, layer sizes,
activation functions
Human Decision, Methodology to come.

Parameters previous session $\implies$ belong to

$$\arg\min_{p\in\mathbb{R}^D} J(p). \tag{1}$$

Remark first $J \in \mathcal{C}^1$, later specific form

Existence AI literature $\cap$ Perron's Paradox $= \emptyset$

$$\boxed{\text{Computation}}$$

Equation Classical calculus $\Longrightarrow$ compute

$$\{p \in \mathbb{R}^D \ : \ \mathrm{d}J(p) = 0\}. \tag{2}$$

Successive Approximation Construct

$$\begin{gathered}\text{recursively } (p_n)_{n\geq 0} \in (\mathbb{R}^D)^{\mathbb{N}} \\ n \to +\infty \implies J(p_n) \to \inf J.\end{gathered} \tag{3}$$

Setup

1D Gradient Descent

General Gradient Descent

Improved Algorithms

Stochastic Gradient Descent

Main Idea

Start No information $\implies p_0 \sim \mathcal{N}(0,1)$

Analysis $p_{n+1} = p_n + (p_{n+1} - p_n)$

.Expansion $J(p_{n+1}) \approx J(p_n) + (p_{n+1} - p_n)J'(p_n)$

Sign condition $\mathrm{sign}(p_{n+1} - p_n) = -\mathrm{sign}(J'(p_n))$

Size condition $p_{n+1} - p_n = -\eta J'(p_n)$

Parameter η learning rate, small

$\boxed{\text{ODE Link}}$

$$\begin{aligned}
\text{Rewriting } & \frac{p_{n+1} - p_n}{\eta} = -\nabla J'(p_n) \\
\text{Limit } & \eta \to 0 \implies \dot{p}(t) = -J'(p(t)) \\
\text{Lyapunov } & \frac{\mathrm{d}}{\mathrm{d}t} J(p(t)) = -\left(J'(p(t))\right)^2 \\
\text{Necessity } & p(t) \to p^* \implies J'(p^*) = 0 \\
\text{Sufficiency } & J \text{ convex } \implies p^* \in \arg\min J.
\end{aligned}$$

Setup

1D Gradient Descent

General Gradient Descent

Improved Algorithms

Stochastic Gradient Descent

Gradient Descent

$$\begin{cases} p_{n+1} = p_n - \eta \nabla J(p_n), \quad \forall n \geq 0 \\ p_0 = \mathcal{N}(0,1)(\omega). \end{cases} \tag{4}$$

$\boxed{\text{Łojasiewicz inequality}}$

1. J μ-PL when $\exists p^*$

$$\forall p \in \mathbb{R}^D, \quad \|\nabla J(p)\|^2 \geq 2\mu(J(p) - J(p^*))$$

2. J uniformly convex $\implies$ J μ-PL
3. Locally true for real analytic functions!
4. ODE $\implies$

$$\frac{\mathrm{d}}{\mathrm{d}t} J(p(t)) \leq -2\mu(J(p(t)) - J(p^*)) \tag{5}$$

5. Gronwall $\implies$

$$J(p(t)) - J(p^*) \leq (J(p_0) - J(p^*))e^{-2\mu t} \tag{6}$$

Setup

1D Gradient Descent

General Gradient Descent

Improved Algorithms

Stochastic Gradient Descent

Momentum

$$\begin{cases} m_{n+1} = \beta m_n + (1-\beta)\nabla J(p_n) \\ p_{n+1} = p_n - \eta m_{n+1} \\ p_0 = \mathcal{N}(0,1)(\omega) \\ m_0 = 0. \end{cases} \tag{7}$$

RMSProp

$$\begin{cases} s_{n+1} = \beta s_n + (1-\beta)\|\nabla J(p_n)\|^2 \\ p_{n+1} = p_n - \eta \frac{\nabla J(p_n)}{\epsilon + \sqrt{s_{n+1}}} \\ p_0 = \mathcal{N}(0,1)(\omega) \\ s_0 = 0. \end{cases} \tag{8}$$

Adam

$$\begin{cases} m_{n+1} = \beta_1 m_n + (1-\beta_1)\nabla J(p_n) \\ s_{n+1} = \beta_2 s_n + (1-\beta_2)\|\nabla J(p_n)\|^2 \\ \hat{m}_{n+1} = \frac{m_{n+1}}{1-\beta_1^{n+1}} \\ \hat{s}_{n+1} = \frac{s_{n+1}}{1-\beta_2^{n+1}} \\ p_{n+1} = p_n - \eta\frac{\hat{m}_{n+1}}{\epsilon+\sqrt{\hat{s}_{n+1}}} \\ p_0 = \mathcal{N}(0,1)(\omega) \\ m_0 = 0 \\ s_0 = 0. \end{cases} \tag{9}$$

Remarks

- Solvers $\implies$ additional hyperparameters
- e.g. Adam: β_1, β_2, η, ϵ.
- ⚠ Theoretical Garantees ⚠
- When do we stop?

Setup

1D Gradient Descent

General Gradient Descent

Improved Algorithms

Stochastic Gradient Descent

$$\boxed{\text{Loss Function}}$$

$$\forall p \in R^D, \quad J(p) := \frac{1}{N} \sum_{0 \le k < N} J_k(p). \tag{10}$$

1. Specific Form to Machine Learning
2. N : sample size, very large (internet size)
3. D : parameter size 10^9 to 10^{12}.
4. Classical Gradient Descent expensive steps!

Stochastic Gradient Descent

1. $N = B \cdot S$
2. E : number of epochs
3. $1 \leq e \leq E$: σ_e random in $\mathbb{S}(N)$
4. $0 \leq s < S$: loss function

$$J_{s,e}(p) := \frac{1}{B} \sum_{sB \leq k < (s+1)B} J_{\sigma_e(k)}(p). \qquad (11)$$

5. One step of some gradient algorithm for $J_{s,e}$.