deeplearning.ai

Optimization Algorithms

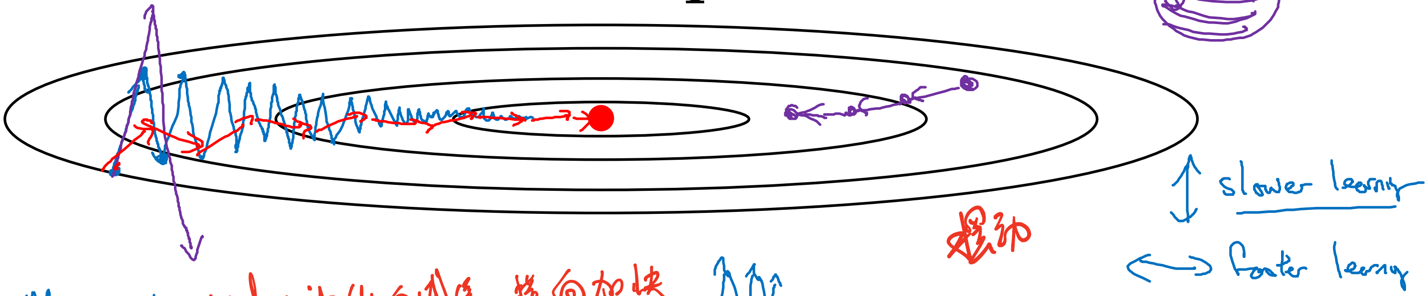
Gradient descent with momentum

Basic idea:

to compute an exponentially weighted average of your gradients, and then use that gradient to update your

weight instead.

Gradient descent example



Momentum: red: 让纵向减慢, 横向加快

On iteration t:

Compute dW, db on current mini-batch.

$$V_{dW} = \beta V_{dW} + (1-\beta) dW$$

$$V_{db} = \beta V_{db} + (1-\beta) db$$

"$V_\theta = \beta V_\theta + (1-\beta)\theta_t$"

bowl example: friction 阻力 → β ↑ velocity 速度, ↑ acceleration 加速度

$$W := W - \alpha V_{dW}, \quad b := b - \alpha V_{db}$$

deeplearning.ai

Andrew Ng

Implementation details

$V_{dW} = 0$, $V_{db} = 0$

On iteration t:

Compute dW, db on the current mini-batch

$v_{dW} = \beta v_{dW} + (1-\beta) dW$ (keep)

$v_{db} = \beta v_{db} + (1-\beta) db$ (keep)

$V_{dW} = \beta V_{dW} + dW$ ← some essays, adjust α with about $\frac{1}{1-\beta}$

$W = W - \alpha v_{dW}, \quad b = b - \alpha v_{db}$

Hyperparameters: α, β

$\beta = 0.9$ Common choice

average over last ≈ 10 gradients