



deeplearning.ai

Optimization Algorithms

Mini-batch
gradient
descent

Batch vs. mini-batch gradient descent

Vectorization allows you to efficiently compute on m examples.

$$\begin{array}{c}
 \text{X, Y} \\
 \text{X}^{t+3}, \text{Y}^{t+3}
 \end{array}$$

$$\begin{array}{c}
 \text{X} = \left[\underbrace{\text{x}^{(1)} \text{ x}^{(2)} \text{ x}^{(3)} \dots \text{x}^{(1000)}}_{\text{X}^{\{1\}} \text{ (n_x, 1000)}} \mid \underbrace{\text{x}^{(1001)} \dots \text{x}^{(2000)}}_{\text{X}^{\{2\}} \text{ (n_x, 1000)}} \mid \dots \mid \underbrace{\dots \text{x}^{(m)}}_{\text{X}^{\{5,000\}} \text{ (n_x, 1000)}} \right] \\
 \text{(n_x, m)}
 \end{array}$$

$$\begin{array}{c}
 \text{Y} = \left[\underbrace{\text{y}^{(1)} \text{ y}^{(2)} \text{ y}^{(3)} \dots \text{y}^{(1000)}}_{\text{Y}^{\{1\}} \text{ (1, 1000)}} \mid \underbrace{\text{y}^{(1001)} \dots \text{y}^{(2000)}}_{\text{Y}^{\{2\}} \text{ (1, 1000)}} \mid \dots \mid \underbrace{\dots \text{y}^{(m)}}_{\text{Y}^{\{5,000\}} \text{ (1, 1000)}} \right] \\
 \text{(1, m)}
 \end{array}$$

What if $m = \underline{5,000,000}$?

5,000 mini-batches of 1,000 each

Mini-batch t : $\text{X}^{t+3}, \text{Y}^{t+3}$

$$\begin{array}{c}
 \text{x}^{(i)} \\
 \text{z}^{[l]} \\
 \text{X}^{t+3}, \text{Y}^{t+3}
 \end{array}$$

Mini-batch gradient descent

repeat {
for $t = 1, \dots, 5000$ {

Forward prop on $X^{\{t\}}$.

$$Z^{\{t\}} = W^{\{t\}} X^{\{t\}} + b^{\{t\}}$$

$$A^{\{t\}} = \sigma^{\{t\}}(Z^{\{t\}})$$

$$\vdots$$

$$A^{\{t\}} = \sigma^{\{t\}}(Z^{\{t\}})$$

Vectorized implementation
(1000 examples)

Compute cost $J^{\{t\}} = \frac{1}{1000} \sum_{i=1}^L \ell(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2 \cdot 1000} \sum_{\mathbf{a}} \|W^{\{t\}}\|_F^2$.

Backprop to compute gradients w.r.t $J^{\{t\}}$ (using $(X^{\{t\}}, Y^{\{t\}})$)

$$W^{\{t+1\}} := W^{\{t\}} - \alpha dW^{\{t\}}, \quad b^{\{t+1\}} := b^{\{t\}} - \alpha db^{\{t\}}$$

"1 epoch"

pass through training set.

1 step of grad desc
using $X^{\{t+1\}}, Y^{\{t+1\}}$.
(as if $m=1000$)

X, Y