deeplearning.ai

Optimization
Algorithms

Mini-batch
gradient descent

Batch vs. mini-batch gradient descent

X, Y

$X^{\{t\}}, Y^{\{t\}}.$

Vectorization allows you to efficiently compute on m examples.

$$X = [\underbrace{x^{(1)} \; x^{(2)} \; x^{(3)} \; \cdots \; x^{(1000)}}_{X^{\{1\}} \; (n_x, 1000)} \mid \underbrace{x^{(1001)} \; \cdots \; x^{(2000)}}_{X^{\{2\}} \; (n_x, 1000)} \mid \cdots \cdots \mid \underbrace{\cdots \cdots \; x^{(m)}}_{X^{\{5,000\}} \; (n_x, 1000)}]$$

(n_x, m)

$$Y = [\underbrace{y^{(1)} \; y^{(2)} \; y^{(3)} \; \cdots y^{(1000)}}_{Y^{\{1\}} \; (1, 1000)} \mid \underbrace{y^{(1001)} \; \cdots \; y^{(2000)}}_{Y^{\{2\}} \; (1, 1000)} \mid \cdots \cdots \mid \underbrace{\cdots \; y^{(m)}}_{Y^{\{5,000\}} \; (1, 1000)}]$$

$(1, m)$

What if $m = 5,000,000$?

5,000 mini-batches of 1,000 each

Mini-batch t: $X^{\{t\}}, Y^{\{t\}}.$

$x^{(i)}$

$z^{[l]}$

$X^{\{t\}}, Y^{\{t\}}.$

Andrew Ng

Mini-batch gradient descent

repeat $\{$

for $t = 1, \ldots, 5000 \ \{$

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

$\qquad\qquad Z^{[1]} = W^{[1]} X^{\{t\}} + b^{[1]}$ $\left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\}$ Vectorized implementation

$\qquad\qquad A^{[1]} = g^{[1]}(Z^{[1]})$ $\qquad\qquad\qquad$ (1000 examples)

$\qquad\qquad \vdots$

$\qquad\qquad A^{[L]} = g^{[L]}(Z^{[L]})$

$\qquad$ Compute cost $J^{\{t\}} = \dfrac{1}{1000} \sum\limits_{i=1}^{l} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) + \dfrac{\lambda}{2 \cdot 1000} \sum\limits_{l} \|W^{[l]}\|_F^2$.

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\uparrow$ from $X^{\{t\}}, Y^{\{t\}}$.

$\qquad$ Backprop to compute gradients wrt $J^{\{t\}}$ $\left(\text{using } (X^{\{t\}}, Y^{\{t\}})\right)$

$\qquad W^{[l]} := W^{[l]} - \alpha \, dW^{[l]}, \quad b^{[l]} := b^{[l]} - \alpha \, db^{[l]}$

$\}$

$\}$

"1 epoch" $\quad\underset{\quad}{\llcorner}$ pass through training set.

1 step of gradient descent
using $\underline{X^{\{t\}}, Y^{\{t\}}}$.
(as if $m = 1000$)

$\underline{X, Y}$

Andrew Ng