

2. Understanding Mini-batch Gradient Descent

Tags

mini-batch gradient descent

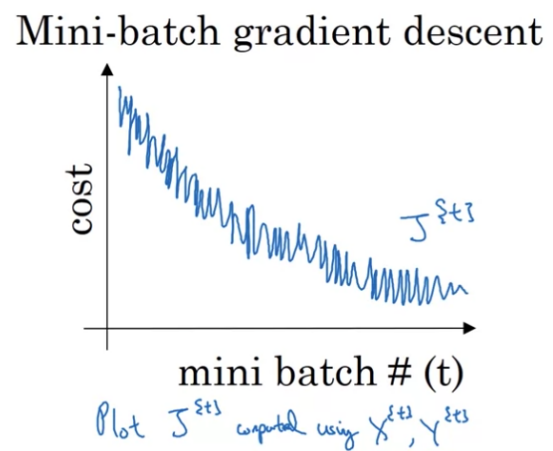
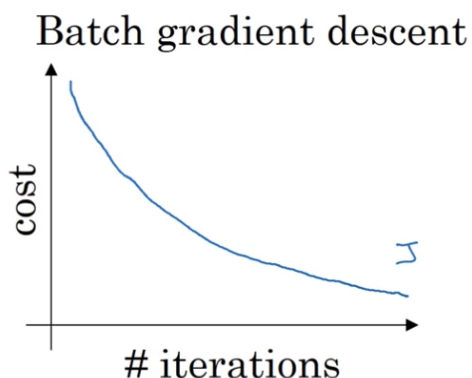
[Training with mini-batch gradient descent](#)

[Choosing your mini-batch size](#)

[Choosing your mini-batch size](#)

Training with mini-batch gradient descent

Training with mini batch gradient descent



Andrew Ng

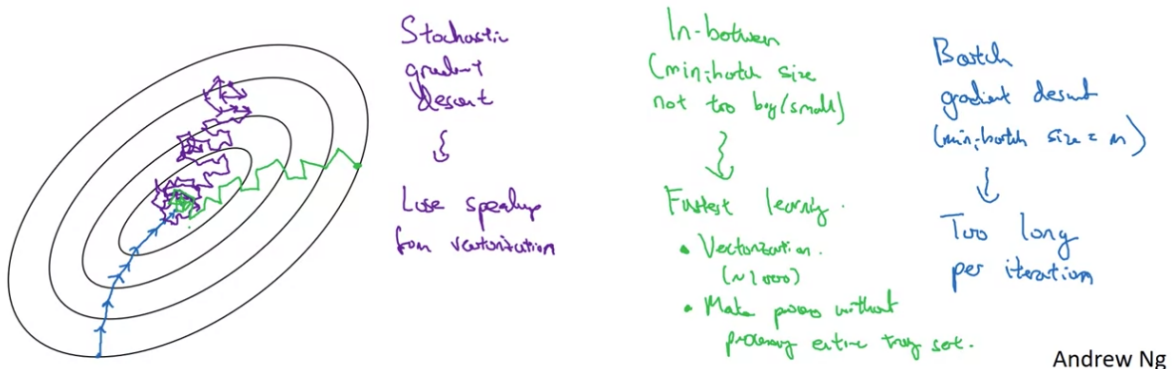
mini-batch : it may not decrease every iteration

- oscillations(진동화): 미니배치의 경우, 특정 배치에서는 학습이 쉬워서 cost function 값이 낮을 수 있다(반대도 마찬가지!)

Choosing your mini-batch size

Choosing your mini-batch size

- If mini-batch size = m : Batch gradient descent. $(X^{(1)}, Y^{(1)}) = (X, Y)$
- If mini-batch size = 1 : Stochastic gradient descent. Every example is its own mini-batch. $(X^{(1)}, Y^{(1)}) = (x^{(1)}, y^{(1)}) \dots (x^{(n)}, y^{(n)})$ mini-batch.
- In practice: Somewhere in-between 1 and m



- batch gradient descent: Too long per iteration
- stochastic gradient descent: Lose almost all your speed up from vectorization

if mini-batch size == 1 then Every example is its own mini-batch

⇒ 미니배치 사이즈가 1이라면 모든 샘플들이 미니배치가 되는 것이다?!



We talk about learning rate decay later

Choosing your mini-batch size

Choosing your mini-batch size

If small toy set : Use batch gradient descent.
($m \leq 2000$)

Typical mini-batch sizes:

→ 64, 128, 256, 512, 1024
 $2^6, 2^7, 2^8, 2^9, 2^{10}$

Make sure mini-batch fit in CPU/GPU memory.
 $X^{(k)}, y^{(k)}$

Andrew Ng

train data가 2000개 이하라면 batch gradient descent를 사용하라
그 외의 경우 64, 128, 256, 512의 미니배치 사이즈를 적용한 mini-batch gradient를 사용하라



Make sure mini-batch fit in CPU/GPU memory