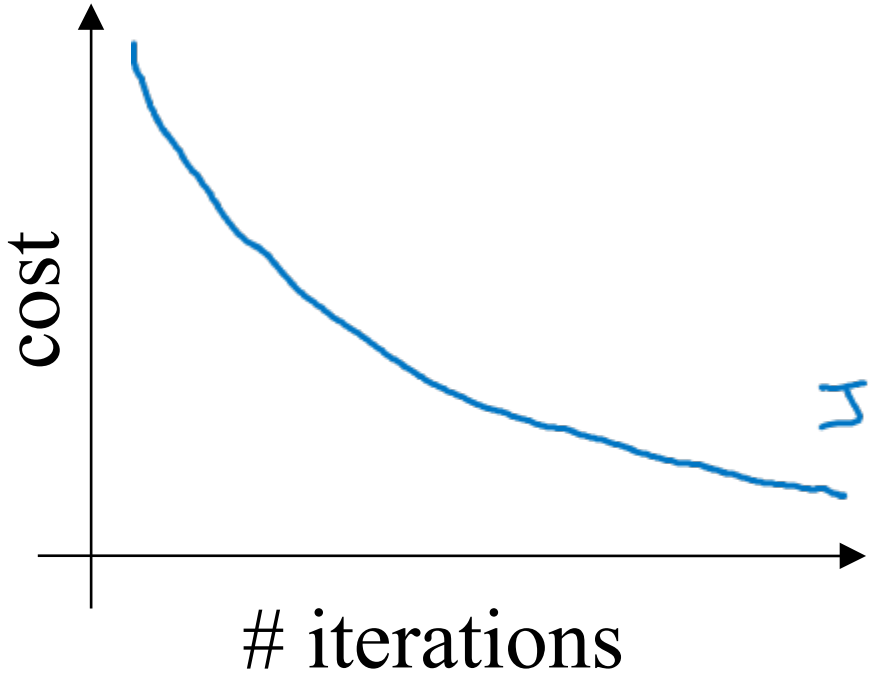
deeplearning.ai

Optimization Algorithms

Understanding mini-batch gradient descent

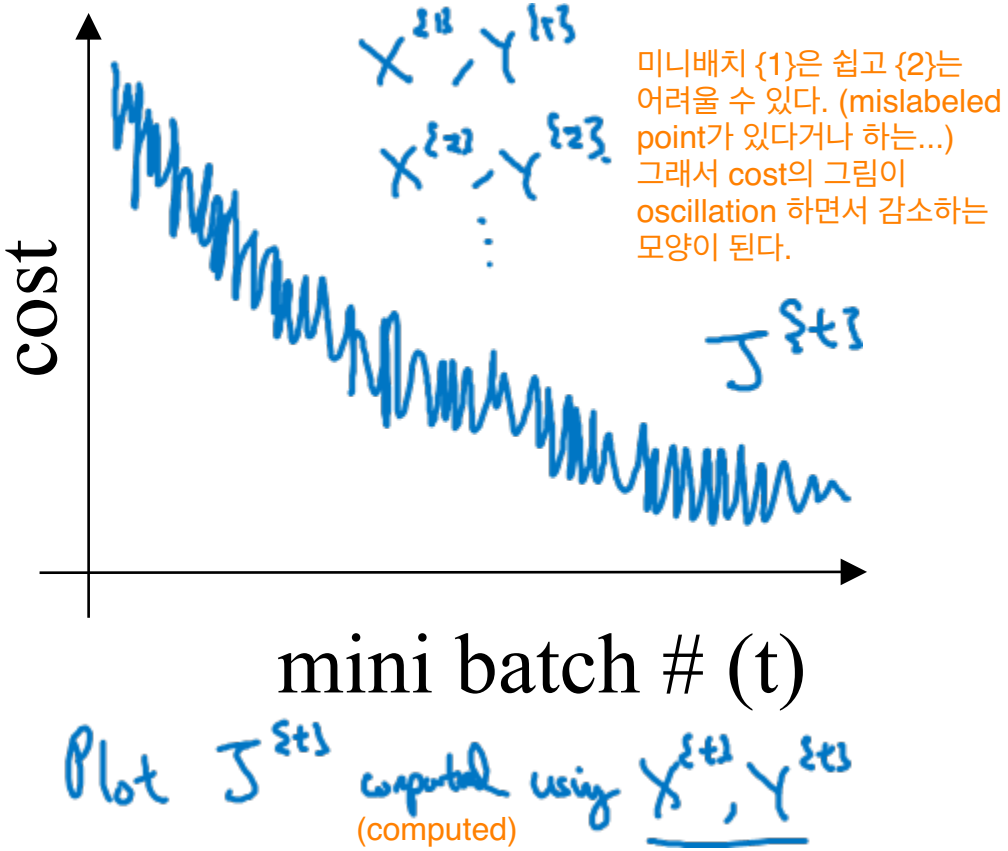
Training with mini batch gradient descent

Batch gradient descent



batch gradient descent를 쓰면 매 반복마다 항상 cost가 감소하게 된다. 한번이라도 증가하면 뭔가 이상이 있는 것임. learning rate가 너무 크다거나...

Mini-batch gradient descent



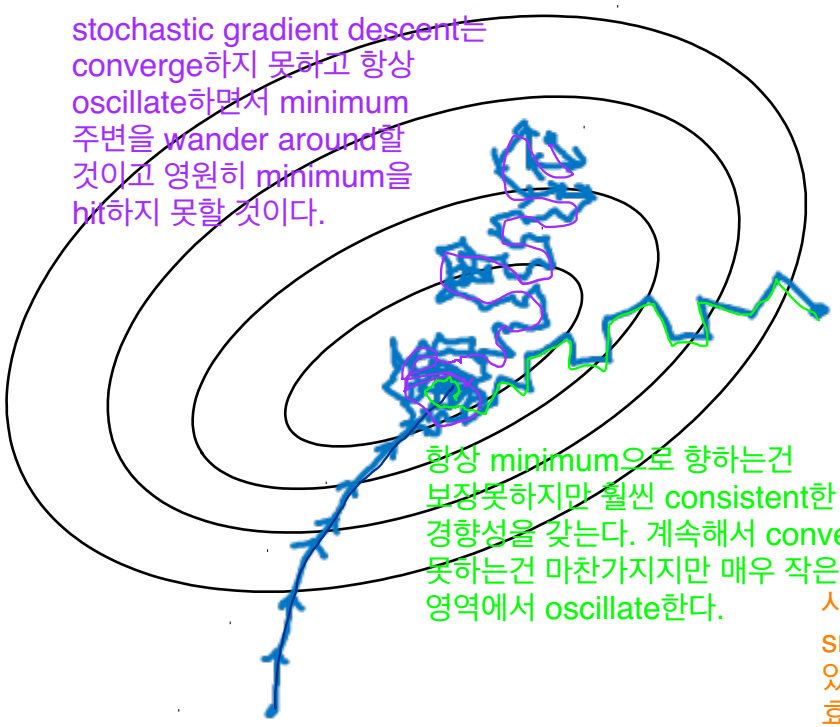
Andrew Ng

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 it own mini-batch. $(X^{\{1\}}, Y^{\{1\}}) = (x^{(1)}, y^{(1)}) \dots (x^{(m)}, y^{(m)})$

→ In practice: Somewhere in-between 1 and m



stochastic gradient descent는 converge하지 못하고 항상 oscillate하면서 minimum 주변을 wander around할 것이고 영원히 minimum을 hit하지 못할 것이다.

항상 minimum으로 향하는건 보장못하지만 훨씬 consistent한 경향성을 갖는다. 계속해서 converge 못하는건 마찬가지지만 매우 작은 영역에서 oscillate한다.

Stochastic gradient descent
↓
Lose speedup (speed up) from vectorization

사실 noisiness는 큰 문제가 아니다 걍 smaller learning rate 쓰거나 해서 줄일 수 있다. 근데 더 큰 문제는 vectorization 효과를 볼 수 없다는 것

In-between (mini-batch size not too big/small)
↓
Fastest learning.
- Vectorization. (n 1000)
- Make progress (make progress) without (without) processing (processing) entire (entire) training set (training set).

Batch gradient descent (mini-batch size = m)
↓
Too long per iteration

Andrew Ng

Choosing your mini-batch size

If small training 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})

common size

Make sure mini-batch fit in CPU/GPU memory.
$X^{\{t\}}, Y^{\{t\}}$

근데 애플리케이션마다 상황이 달라서 최적의 mini-batch size가 무엇인지는 해봐야 안다.
GD나 MGD보다 even more efficient algorithm이 있는데 한번 알아보자.