



deeplearning.ai

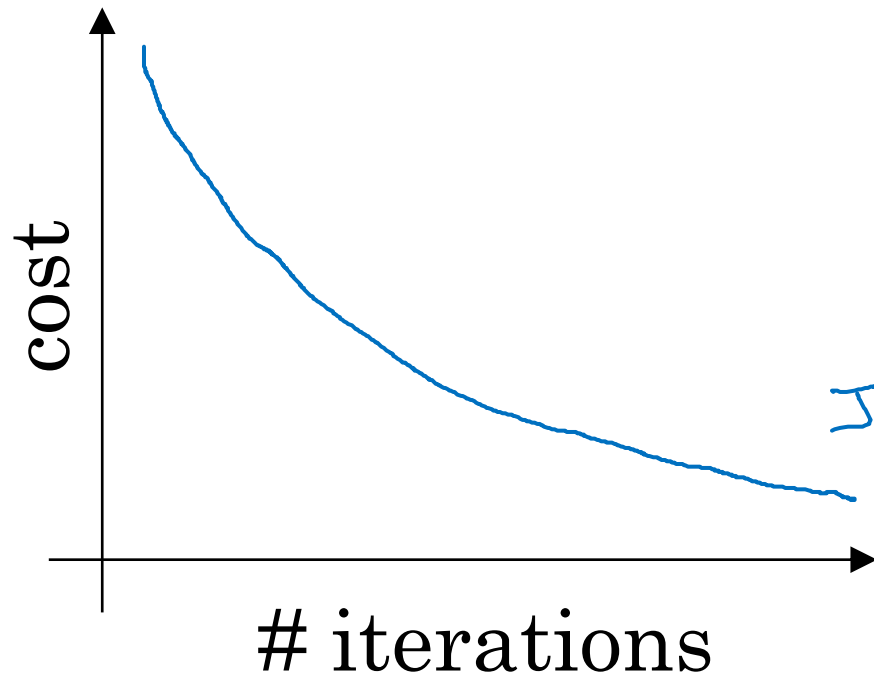
# Optimization Algorithms

---

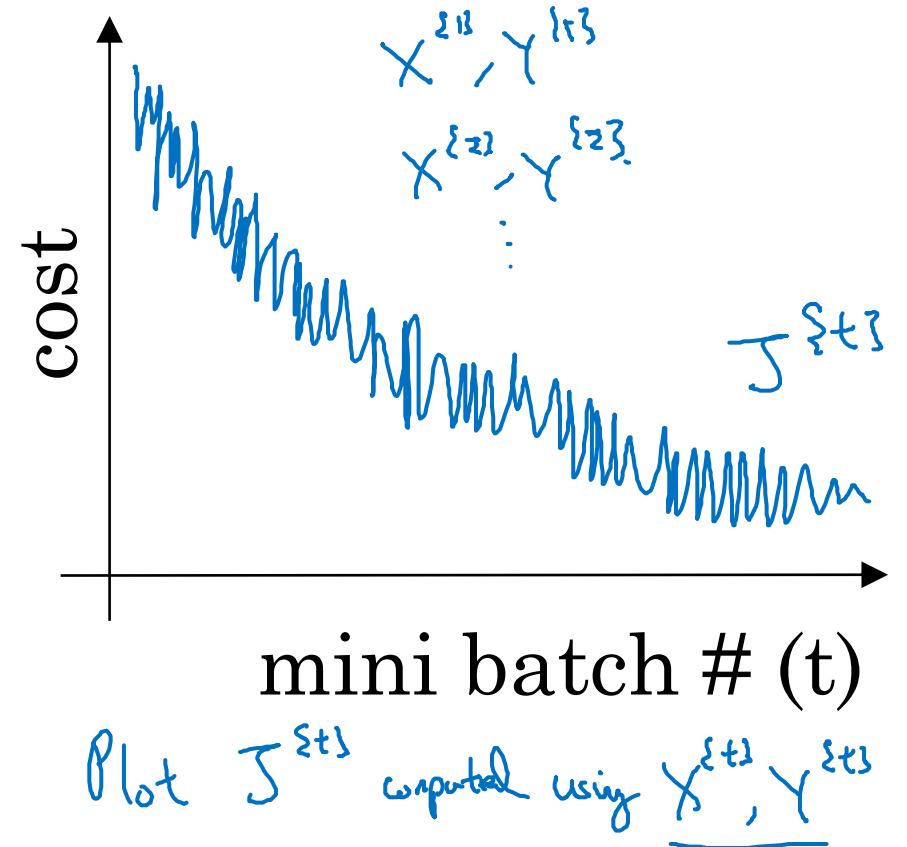
Understanding  
mini-batch  
gradient descent

# Training with mini batch gradient descent

## Batch gradient descent



## Mini-batch gradient descent



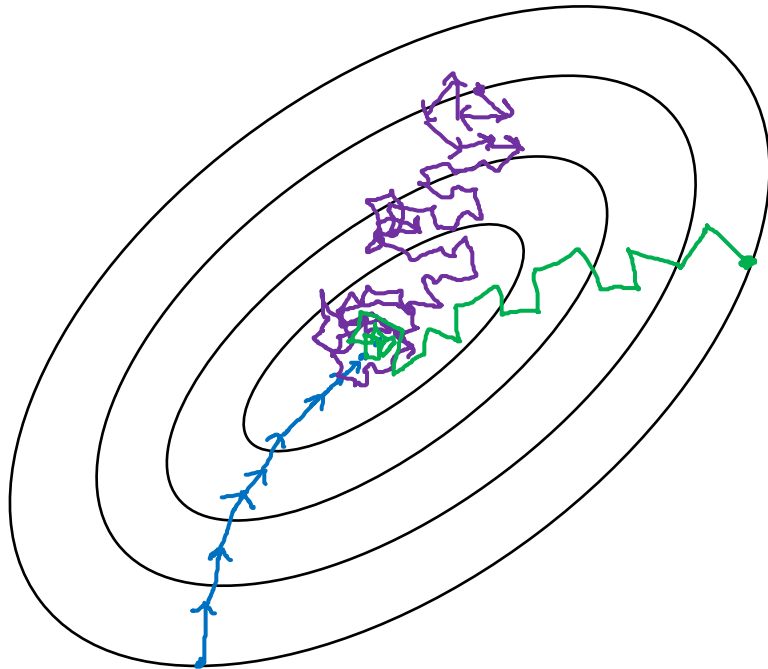
# Choosing your mini-batch size

→ If mini-batch size =  $m$  : Batch gradient descent.

$$(X^{(13)}, Y^{(13)}) = (X, Y).$$

→ If mini-batch size = 1 : Stochastic gradient descent. Every example is its own mini-batch.  
 $(X^{(13)}, Y^{(13)}) = (x^{(1)}, y^{(1)}) \dots (x^{(n)}, y^{(n)})$  mini-batch.

In practice: Somewhere in-between 1 and  $m$



Stochastic  
gradient  
descent

↓  
Loss spikes  
from vectorization

In-between  
(mini-batch size  
not too big/small)

↓  
Fastest learning.

- Vectorization.  
( $n=1000$ )
- Make passes without  
processing entire training set.

Batch  
gradient descent  
(mini-batch size =  $m$ )


↓  
Too long  
per iteration

# Choosing your mini-batch size

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

Typical mini-batch sizes:

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



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