deeplearning.ai

Optimization Algorithms

Mini-batch gradient descent

Applying machine learning is a highly empirical process, is a highly iterative process. In which you just had to train a lot of models to find one that works really well.
So it's really helps to train models quickly.
Training on large dataset is just slow. So what you find is that having fast optimization algorithms, having good optimization algorithm can really speed up your efficiency and your team.

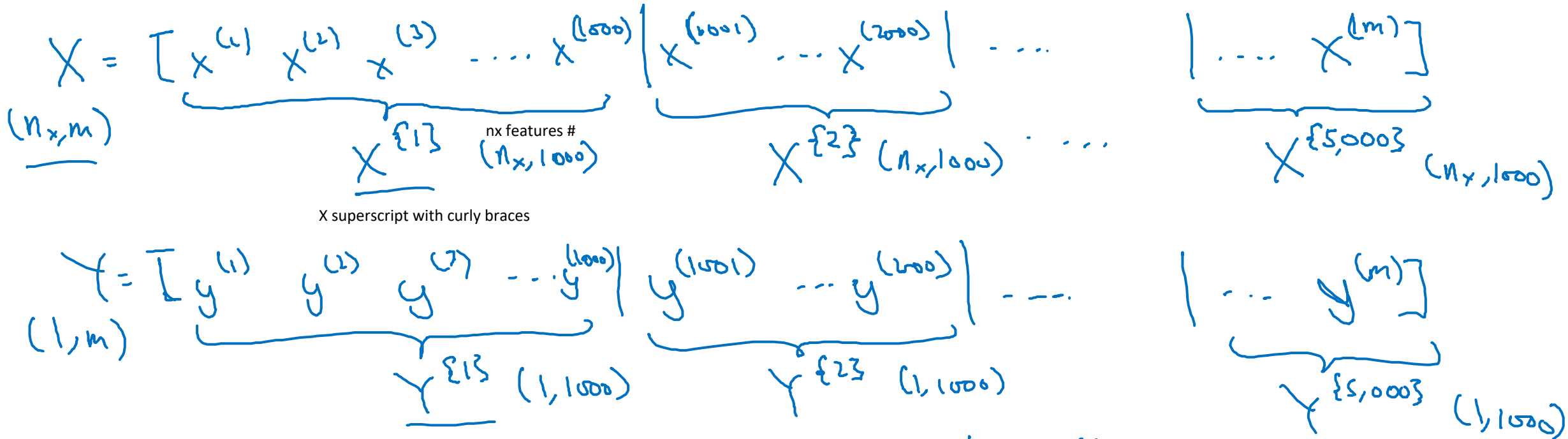
Viewing that as processing your entire batch of training samples all at the same time.

In contrast, process is single mini batch X^{t}, y^{t} at the same time

Batch vs. mini-batch gradient descent

You have to process your entire training set before you take one little step of gradient descent. And the you have to process the entire training sets of five million training samples again before you take another step of gradient descent.

Vectorization allows you to efficiently compute on m examples.



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

5,000 mini-batches of 1,000 each

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

$x^{(i)}$ ith training sample

$z^{[l]}$ index the different layers of the neural network

$X^{\{t\}}, Y^{\{t\}}$. index into different mini batches

Andrew Ng

Mini-batch gradient descent

1 step of gradient descent using $X^{\{t\}}, Y^{\{t\}}$.

(as if $m = 1000$)

rather than having a explicit for loop overall 1,000 examples, you would use vectorization to process 1,000 examples sort of all at the same time.

X, Y

Repeat {

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

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

$Z^{[1]} = W^{[1]} X^{\{t\}} + b^{[1]}$

$A^{[1]} = g^{[1]}(Z^{[1]})$

$\vdots$

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

} Vectorized implementation (1000 examples)

Z Vectorizing connotation

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

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

Frobenius on the way which is square

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

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

} update

} "1 epoch" single pass through training set.

Andrew Ng