

BATCH, MINI-BATCH
AND
STOCHASTIC
GRADIENT DESCENT

What does a learning algorithm do?

The job of a learning algorithm is to take the dataset of inputs/features and outputs and find the optimal values for the parameters of the model.

What does optimal mean?

Optimal values of the parameters of the model are those values that minimize the total penalty - i.e., the cost.

How does the learning algorithm arrive at the optimal values?

By repeatedly cycling through / iterating through the dataset. Each pass through is a step of gradient descent.

How does the learning algorithm
know when to stop iterating?

- We can tell it explicitly to stop after a certain number of iterations.

OR

- We can tell it to stop when the result from one iteration to the next don't change very much.

Data Set

	n features				
	x_0	x_1	x_2	x_3	Output
1					
2					
3					
4					

This feature is "manufactured"

Transforming the features into outputs

$$(w_0 * x_0) + (w_1 * x_1) + (w_2 * x_2) + (w_3 * x_3)$$

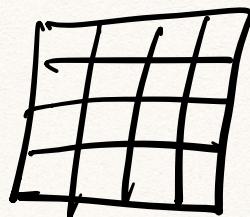
4 features

4 parameters

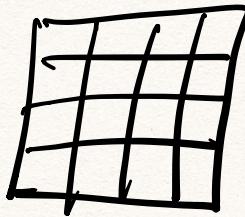
There's always one more feature than what's given to us in the dataset.

Batch Gradient Descent

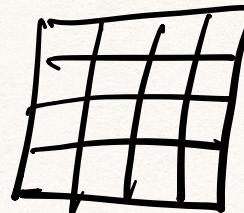
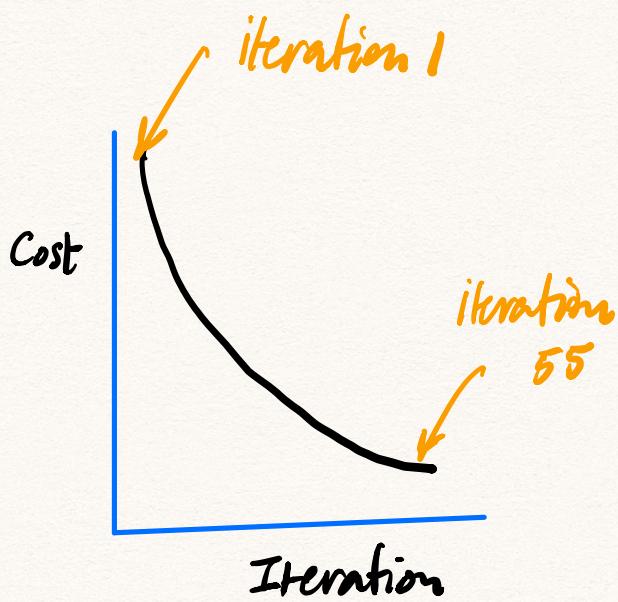
Uses the entire dataset for every iteration



Iteration 1

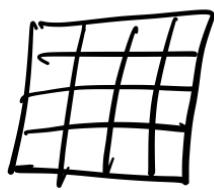


Iteration 2

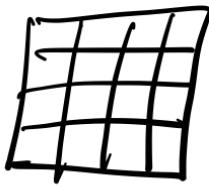


Iteration
55

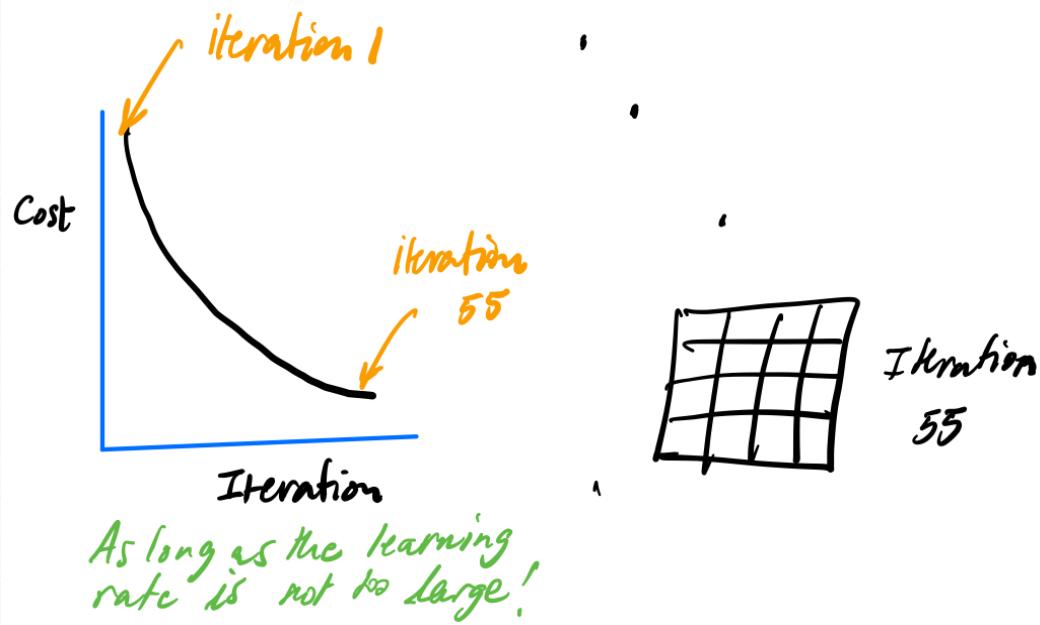
As long as the learning rate is not too large!



Iteration 1

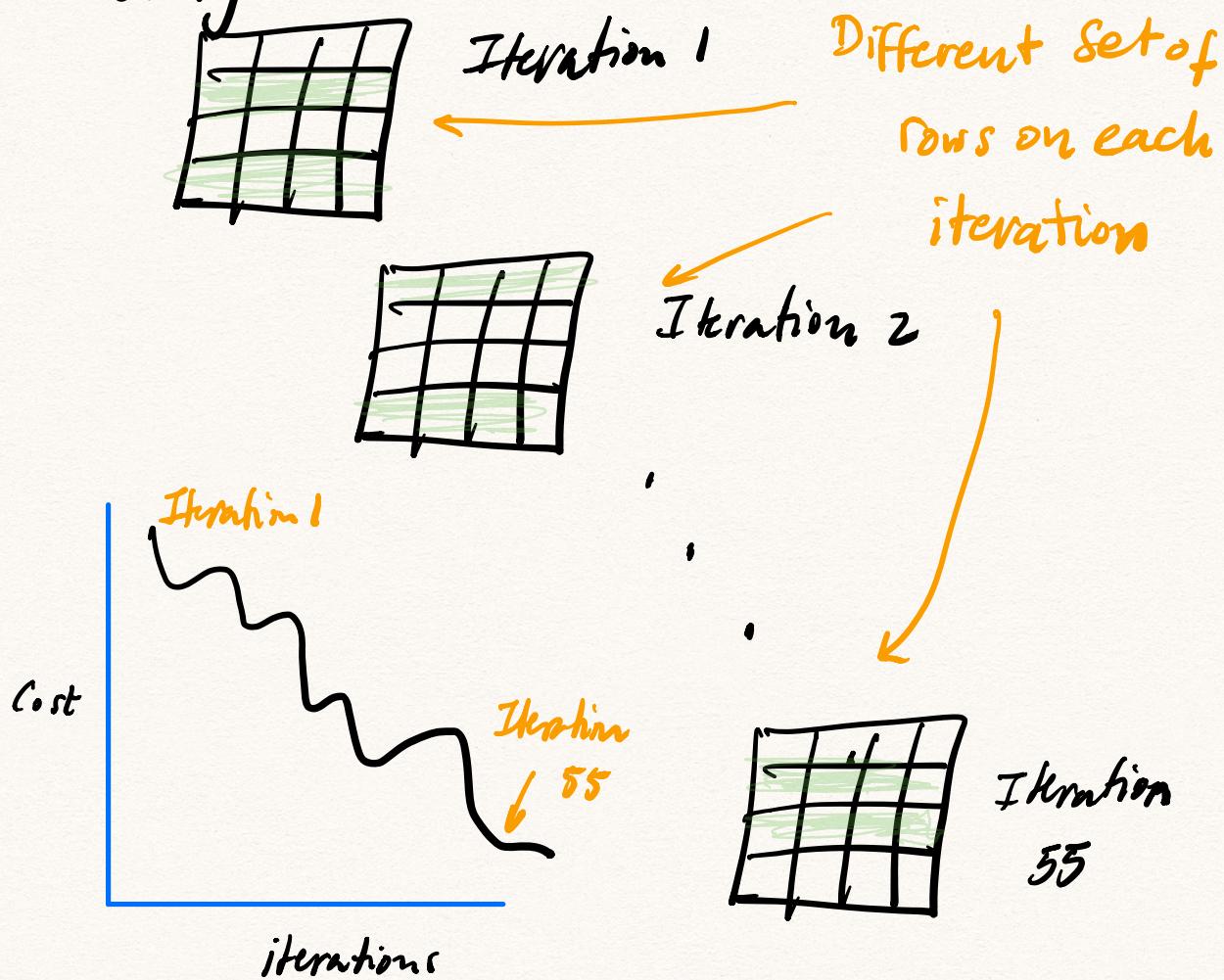


Iteration 2

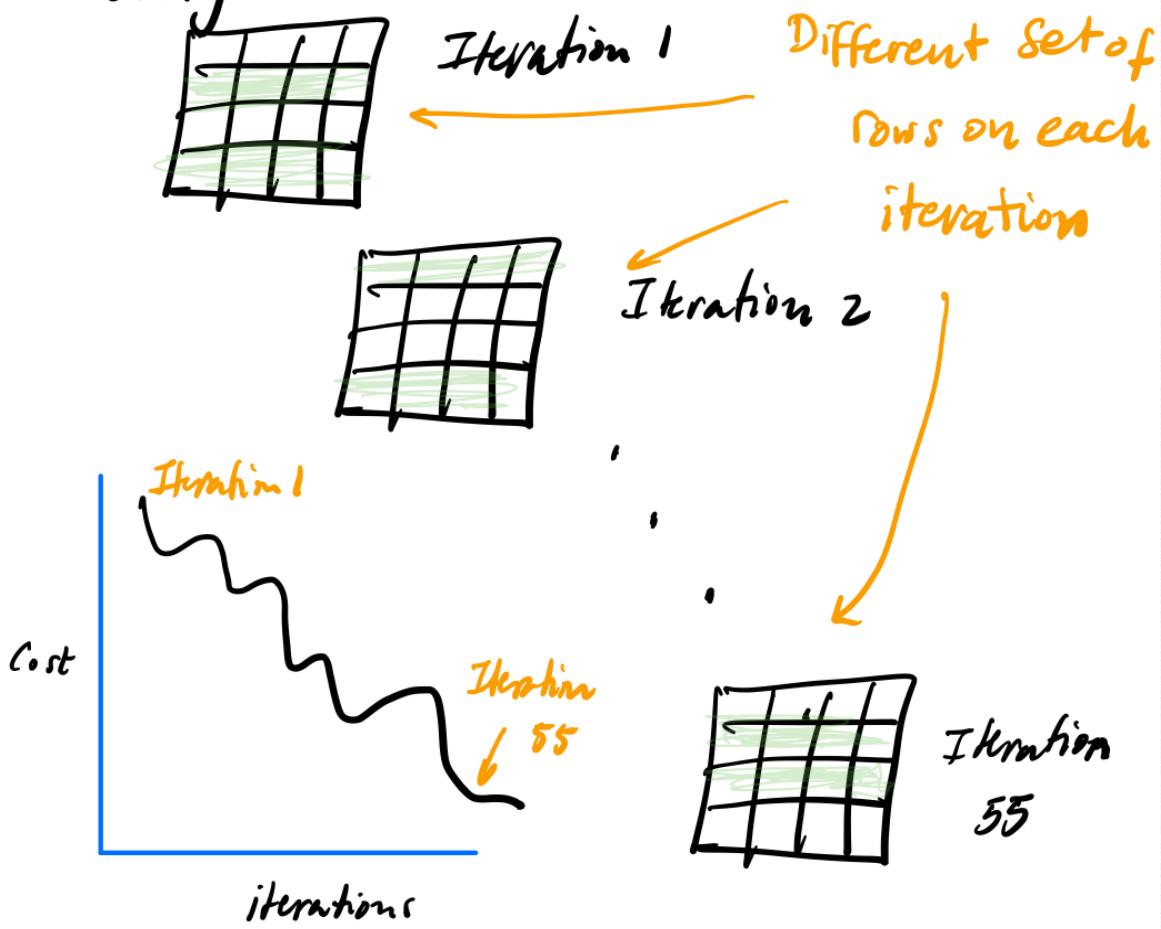


Mini-Batch Gradient Descent

Use a random set of rows
(the same number of rows but the
rows are different each time) for
every iteration.

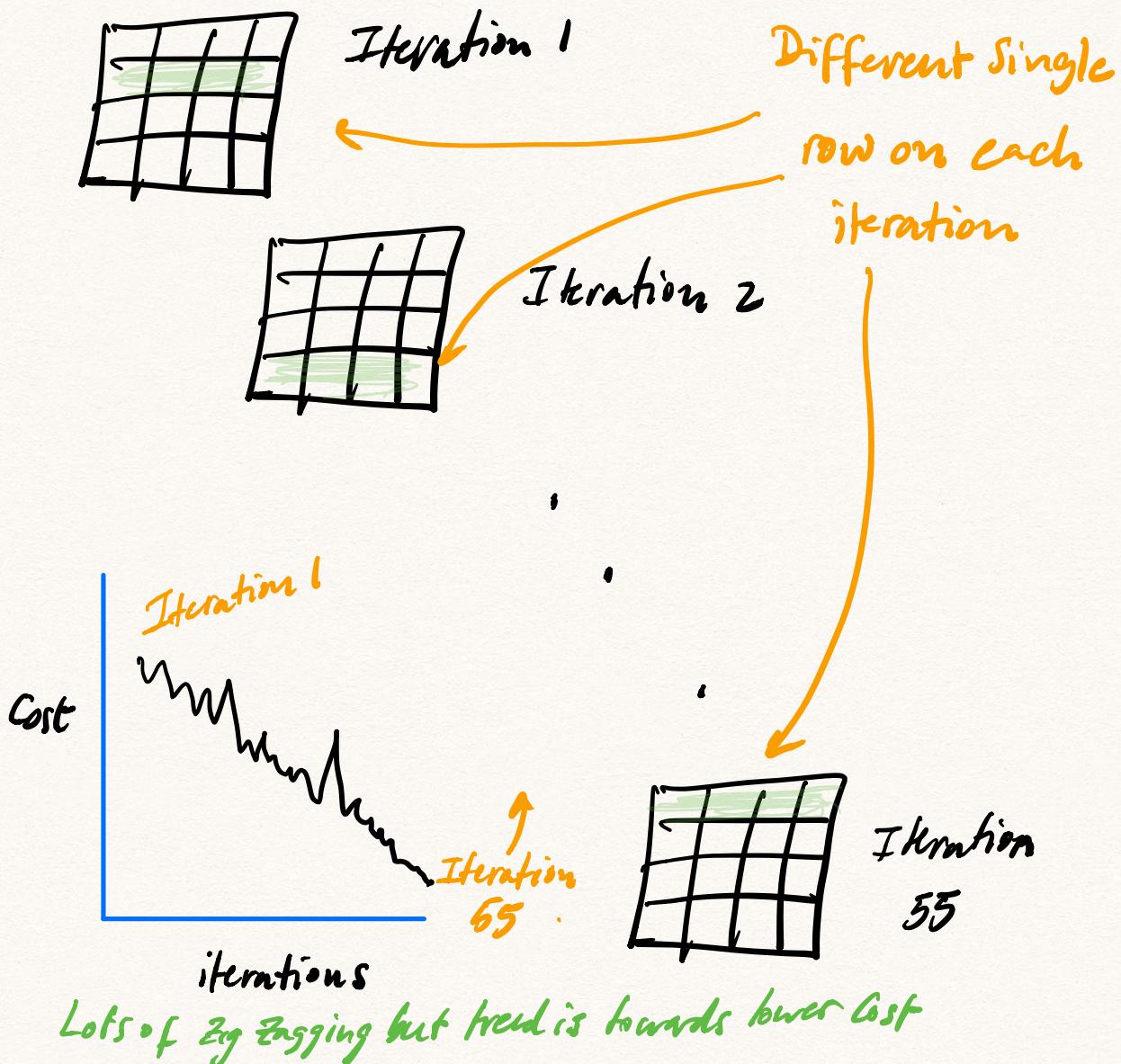


Use a random set of rows
(the same number of rows but the
rows are different each time) for
every iteration.



Stochastic Gradient Descent

Use a single randomly chosen row for each iteration.



Use a single randomly chosen row for each iteration.

