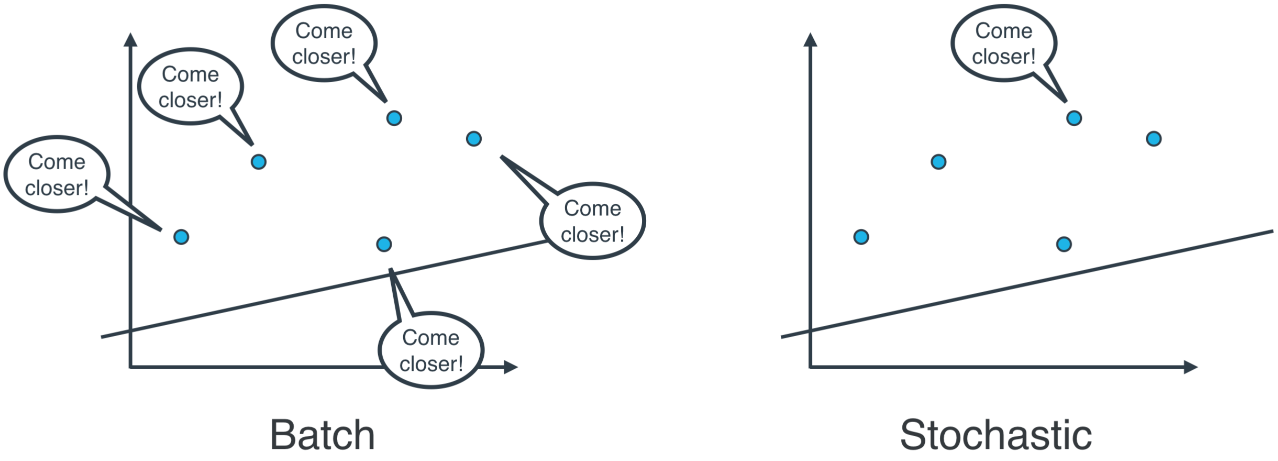
Batch vs Stochastic Gradient Descent

At this point, it seems that we've seen two ways of doing linear regression.

- By applying the squared (or absolute) trick at every point in our data *one by one*, and repeating this process many times.
- By applying the squared (or absolute) trick at every point in our data *all at the same time*, and repeating this process many times.

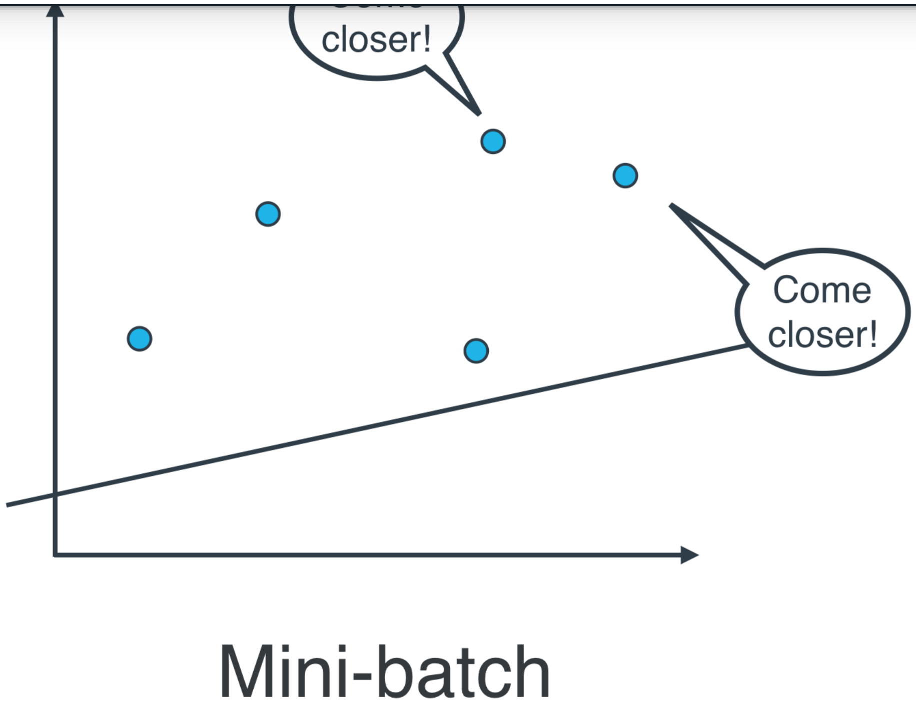
More specifically, the squared (or absolute) trick, when applied to a point, gives us some values to add to the weights of the model. We can add these values, update our weights, and then apply the squared (or absolute) trick on the next point. Or we can calculate these values for all the points, add them, and then update the weights with the sum of these values.

The latter is called *batch gradient descent*. The former is called *stochastic gradient descent*.



The question is, which one is used in practice?

Actually, in most cases, neither. Think about this: If your data is huge, both are a bit slow, computationally. The best way to do linear regression, is to split your data into many small batches. Each batch, with roughly the same number of points. Then, use each batch to update your weights. This is still called *mini-batch gradient descent*.



NEXT