Stochastic Gradient Descent

Have you ever played a game where you have to find a hidden object in a room? The game usually goes like this: you're given a clue and then you have to search the room to find the hidden object. The first clue might be "the object is red" and you start by looking at all the red objects in the room. If you don't find it, you move on to the next clue until you finally find the hidden object.

Well, SGD is like playing that game, but with math! Instead of searching for a hidden object, we're trying to find the best values for the parameters in a machine learning model. And instead of clues, we use data to guide us towards the best parameter values.

Here's how it works: imagine we have a dataset with a bunch of examples and we want to train a model to make predictions on new data. We start by randomly initializing the parameters of the model. Then, for each example in the dataset, we calculate the error between the predicted output of the model and the actual output. We use this error to update the parameters of the model, making small adjustments to nudge it closer to the correct values.

But here's the fun part: instead of going through the entire dataset before updating the parameters, we update the parameters after processing each example! This means that we're constantly adjusting the parameters based on new data, rather than waiting until we've seen everything before making updates. It's like playing the game where you're given one clue at a time and have to search the room after each clue.

This makes SGD really useful for large datasets, because we don't have to wait until we've seen all the examples before making updates to the model. It also makes it easier to escape local minima (points where the error is small but not the smallest possible), because we're constantly adjusting the parameters based on new information.

A practical example of SGD in action is training a model to recognize handwritten digits. We start with a dataset of images of handwritten digits and their corresponding labels (i.e. the digit they represent). We initialize the parameters of the model randomly and start processing each example one at a time. After each example, we update the

parameters to reduce the error between the predicted output of the model and the actual output. We repeat this process for all the examples in the dataset, making small adjustments to the parameters after each one.

Overall, SGD is a powerful and flexible optimization algorithm that's widely used in machine learning. It's like playing a fun game where you use data to guide you towards the best model parameters!