**Epoch:** One pass over the training set.

**Batch size:** In Batch Gradient Descent (see below), we split the training set into batches and train the model by feeding it data in batches. After each batch, the model adjusts the weights of the neurons. (e.g. 8, 16, 32, 64, etc.)

**Optimizer Step:** One weight update. (e.g. for SGD, optimizer step = no. of data samples)

**So, how many weight updates per epoch?**

No. of weight updates (steps) per epoch = No. of data in training set / Batch size

**Iterations:** The total number of weight updates after *n* Epochs.

Iterations = *n* Epochs \* No. of weight updates per epoch

**decay\_steps (in Keras schedules):** It is the number of weight updates (steps) after which the learning rate will be decayed. It is a positive integer (duh!).

**decay\_rate (in Keras schedules):** It is the rate at which the learning rate will be decayed. (e.g. if 0.5, the LR will be halved every weight update).

New learning rate = Old learning rate \* Decay rate

**Note:** When using a learning rate schedule, by default, Keras updates (decays) the learning rate after every batch update, NOT after every epoch! This is called the **decay step**, which the user can define when using a learning rate scheduler.

**Gradient Descent vs. Stochastic Gradient Descent vs. Batch Gradient Descent**

Suppose you have 1000 data samples in your training set.

The main difference between these three optimization techniques is in the number of weight updates (steps) per epoch.

In Gradient Descent, the model updates the weights once every epoch.

In SGD, the model updates the weights once each data sample is fed, so in this case, 1000 times per epoch.

In Batch Gradient Descent, the model updates the weights once each batch is fed. So, with a batch size of 100, the model will update the weights 10 times.

\*The Batch Gradient Descent is **faster** than Gradient Descent, and **more accurate** than SGD!

Source: <https://datascience.stackexchange.com/questions/27421/when-are-weights-updated-in-cnn>

**Note:** Whenever you define a batch size in Keras, you are actually using Batch Gradient Descent and not SGD, even when you use the SGD optimizer!