

[Migrate to Big Query.](#)

## Difference Between a Batch and an Epoch in a Neural Network

by **Jason Brownlee** on [August 15, 2022](#) in **Deep Learning**  **229**

Share

Post

Share

Stochastic gradient descent is a learning algorithm that has a number of hyperparameters.

Two hyperparameters that often confuse beginners are the batch size and number of epochs. They are both integer values and seem to do the same thing.

In this post, you will discover the difference between batches and epochs in stochastic gradient descent.

After reading this post, you will know:

- Stochastic gradient descent is an iterative learning algorithm that uses a training dataset to update a model.
- The batch size is a hyperparameter of gradient descent that controls the number of training samples to work through before the model's internal parameters are updated.
- The number of epochs is a hyperparameter of gradient descent that controls the number of complete passes through the training dataset.

**Kick-start your project** with my new book [Deep Learning With Python](#), including *step-by-step tutorials* and the *Python source code* files for all examples.

Let's get started.



What is the Difference Between a Batch and an Epoch in a Neural Network?  
Photo by Graham Cook, some rights reserved.

### Overview

This post is divided into five parts; they are:

1. Stochastic Gradient Descent
2. What Is a Sample?
3. What Is a Batch?
4. What Is an Epoch?
5. What Is the Difference Between Batch and Epoch?

## Stochastic Gradient Descent

Stochastic Gradient Descent, or SGD for short, is an optimization algorithm used to train machine learning algorithms, most notably artificial neural networks used in deep learning.

The job of the algorithm is to find a set of internal model parameters that perform well against some performance measure such as logarithmic loss or mean squared error.

Optimization is a type of searching process and you can think of this search as learning. The optimization algorithm is called “*gradient descent*”, where “*gradient*” refers to the calculation of an error gradient or slope of error and “*descent*” refers to the moving down along that slope towards some minimum level of error.

The algorithm is iterative. This means that the search process occurs over multiple discrete steps, each step hopefully slightly improving the model parameters.

Each step involves using the model with the current set of internal parameters to make predictions on some samples, comparing the predictions to the real expected outcomes, calculating the error, and using the error to update the internal model parameters.

This update procedure is different for different algorithms, but in the case of artificial neural networks, the [backpropagation update algorithm](#) is used.

Before we dive into batches and epochs, let's take a look at what we mean by sample.

Learn more about gradient descent here:

- [Gradient Descent For Machine Learning](#)

## What Is a Sample?

A sample is a single row of data.

It contains inputs that are fed into the algorithm and an output that is used to compare to the prediction and calculate an error.

A training dataset is comprised of many rows of data, e.g. many samples. A sample may also be called an instance, an observation, an input vector, or a feature vector.

Now that we know what a sample is, let's define a batch.

## What Is a Batch?

The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters.

Think of a batch as a for-loop iterating over one or more samples and making predictions. At the end of the batch, the predictions are compared to the expected output variables and an error is calculated. From this error, the update algorithm is used to improve the model, e.g. move down along the error gradient.

A training dataset can be divided into one or more batches.

When all training samples are used to create one batch, the learning algorithm is called batch gradient descent. When the batch is the size of one sample, the learning algorithm is called stochastic gradient descent. When the batch size is more than one sample and less than the size of the training dataset, the learning algorithm is called mini-batch gradient descent.

- **Batch Gradient Descent.** Batch Size = Size of Training Set
- **Stochastic Gradient Descent.** Batch Size = 1
- **Mini-Batch Gradient Descent.**  $1 < \text{Batch Size} < \text{Size of Training Set}$

In the case of mini-batch gradient descent, popular batch sizes include 32, 64, and 128 samples. You may see these values used in models in the literature and in tutorials.

### What if the dataset does not divide evenly by the batch size?

This can and does happen often when training a model. It simply means that the final batch has fewer samples than the other batches.

Alternately, you can remove some samples from the dataset or change the batch size such that the number of samples in the dataset does divide evenly by the batch size.

For more on the differences between these variations of gradient descent, see the post:

- [A Gentle Introduction to Mini-Batch Gradient Descent and How to Configure Batch Size](#)

For more on the effect of batch size on the learning process, see the post:

- [How to Control the Speed and Stability of Training Neural Networks Batch Size](#)

A batch involves an update to the model using samples; next, let's look at an epoch.

## What Is an Epoch?

The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset.

One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters. An epoch is comprised of one or more batches. For example, as above, an epoch that has one batch is called the batch gradient descent learning algorithm.

You can think of a for-loop over the number of epochs where each loop proceeds over the training dataset. Within this for-loop is another nested for-loop that iterates over each batch of samples, where one batch has the specified "batch size" number of samples.

The number of epochs is traditionally large, often hundreds or thousands, allowing the learning algorithm to run until the error from the model has been sufficiently minimized. You may see examples of the number of epochs in the literature and in tutorials set to 10, 100, 500, 1000, and larger.

It is common to create line plots that show epochs along the x-axis as time and the error or skill of the model on the y-axis. These plots are sometimes called learning curves. These plots can help to diagnose whether the model has over learned, under learned, or is suitably fit to the training dataset.

For more on diagnostics via learning curves with LSTM networks, see the post:

- [A Gentle Introduction to Learning Curves for Diagnosing Model Performance](#)

In case it is still not clear, let's look at the differences between batches and epochs.

## What Is the Difference Between Batch and Epoch?

The batch size is a number of samples processed before the model is updated.

The number of epochs is the number of complete passes through the training dataset.

The size of a batch must be more than or equal to one and less than or equal to the number of samples in the training dataset.

The number of epochs can be set to an integer value between one and infinity. You can run the algorithm for as long as you like and even stop it using other criteria besides a fixed number of epochs, such as a change (or lack of change) in model error over time.

They are both integer values and they are both hyperparameters for the learning algorithm, e.g. parameters for the learning process, not internal model parameters found by the learning process.

You must specify the batch size and number of epochs for a learning algorithm.

There are no magic rules for how to configure these parameters. You must try different values and see what works best for your problem.

## Worked Example

Finally, let's make this concrete with a small example.

Assume you have a dataset with 200 samples (rows of data) and you choose a batch size of 5 and 1,000 epochs.

This means that the dataset will be divided into 40 batches, each with five samples. The model weights will be updated after each batch of five samples.

This also means that one epoch will involve 40 batches or 40 updates to the model.

With 1,000 epochs, the model will be exposed to or pass through the whole dataset 1,000 times. That is a total of 40,000 batches during the entire training process.

## Further Reading