Machine Learning Mastery
Making Developers Awesome at Machine Learning

Difference Between a Batch and an Epoch in a Neural Network

by **Jason Brownlee** on August 10, 2022 in **Deep Learning**

Last Updated on August 15, 2022

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.

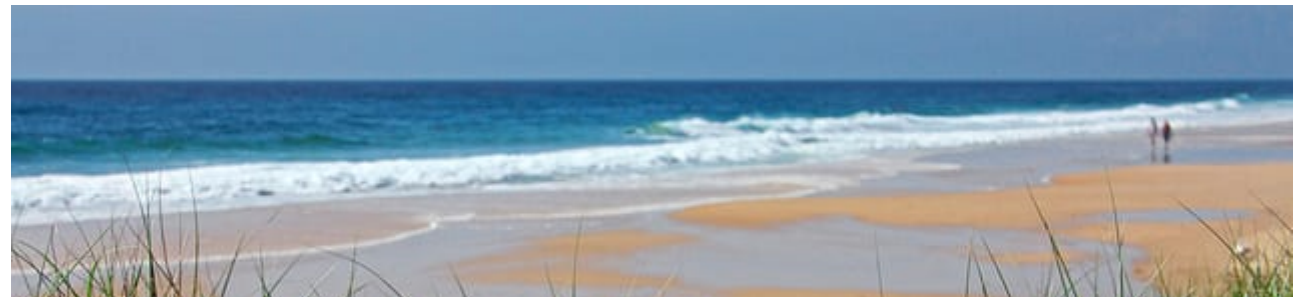
In this post, you will discover the difference between batches and epochs in stochastic gradient descent.

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.



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 < Batch Size < 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

This section provides more resources on the topic if you are looking to go deeper.

- Gradient Descent For Machine Learning
- How to Control the Speed and Stability of Training Neural Networks Batch Size
- A Gentle Introduction to Mini-Batch Gradient Descent and How to Configure Batch Size
- A Gentle Introduction to Learning Curves for Diagnosing Model Performance
- Stochastic gradient descent on Wikipedia
- Backpropagation on Wikipedia

Summary

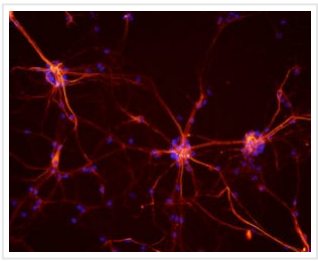
In this post, you discovered the difference between batches and epochs in stochastic gradient descent.

Specifically, you learned:

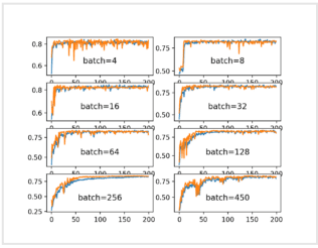
- Stochastic gradient descent is an iterative learning algorithm that uses a training dataset to update a model.
- The batch size is a hyperparameter of gradient descent that controls the number of training samples to work through before the model's internal parameters are updated.
- The number of epochs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

Do you have any questions?
Ask your questions in the comments below and I will do my best to answer.

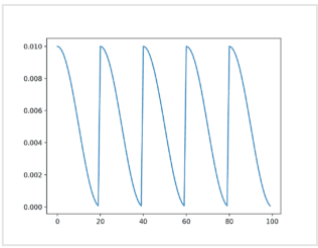
More On This Topic



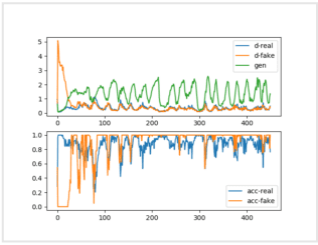
How to Code a Neural Network with Backpropagation In…



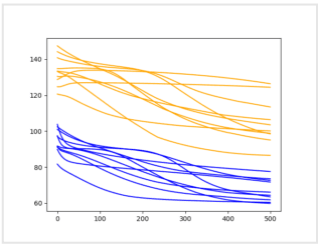
How to Control the Stability of Training Neural…



Snapshot Ensemble Deep Learning Neural Network in Python



How to Identify and Diagnose GAN Failure Modes



How to Tune LSTM Hyperparameters with Keras for Time…

Time Series Prediction with LSTM Recurrent Neural…

About Jason Brownlee

Jason Brownlee, PhD is a machine learning specialist who teaches developers how to get results with modern machine learning methods via hands-on tutorials.

View all posts by Jason Brownlee →

‹ Using Depthwise Separable Convolutions in Tensorflow

When to Use MLP, CNN, and RNN Neural Networks ›

210 Responses to *Difference Between a Batch and an Epoch in a Neural Network*

Zakarie July 20, 2018 at 5:54 am #

REPLY

Very informative and well explained.

Jason Brownlee July 20, 2018 at 6:01 am #

REPLY

Thanks.

JAIKISHAN December 29, 2020 at 1:48 pm #

REPLY

Thank you Jason for the explanation with absolute clarity.

Jason Brownlee December 30, 2020 at 6:30 am #

REPLY

You're welcome.

Sameer July 20, 2018 at 6:34 am #

REPLY

Great Explanation Jason..I have been your big fan and have read all of your books..it's great to learn from u.

Jason Brownlee July 21, 2018 at 6:25 am #

REPLY

Thanks.

Eucharia Onyekwere May 18, 2021 at 9:59 pm #

REPLY

Thanks Jason. Great explanations, your works are quite resourceful.

Jason Brownlee May 19, 2021 at 6:34 am #

REPLY

Thanks!

Elia July 10, 2019 at 8:04 am #

REPLY

I love your articles ,good explanation and i enjoy from the reading

Jason Brownlee July 10, 2019 at 8:22 am #

REPLY

Thanks.

Jesse April 25, 2021 at 7:22 am #

REPLY

Mark Littlewood July 20, 2018 at 7:23 am #

REPLY

You nailed it with the last paragraph, a small simple toy example always trumps a description

Jason Brownlee July 21, 2018 at 6:26 am #

REPLY

Thanks Mark.

med July 24, 2019 at 2:45 pm #

REPLY

does the content of batches change frome an epoch to another ?

Jason Brownlee July 25, 2019 at 7:38 am #

REPLY

Yes. The samples are shuffled at the end of each epoch and batches across epochs differ in terms of the samples they contain.

Craig March 24, 2020 at 7:35 am #

If one is making a time series forecasting model (say something with an lstm layer) will the batch observations of the training set be kept in “chunks” (meaning groups of time will not be broken up, and thus the underlying pattern disrupted)? This matters, right?

Your articles are great! thank you!

Jason Brownlee March 24, 2020 at 8:00 am #

It may. It often does not matter.

You can evaluate this by shuffling samples vs not shuffling samples fed into the LSTM during training or inference.

djm July 20, 2020 at 8:11 pm #

Thank you very much for your precise explanation. If all samples are shuffled at the end of each epoch, is it possible that we may find a single sample in the datasets to be evaluated so many times and some might not be evaluated at all? Or is it possible to make once evaluated sample not to be evaluated again?

Jason Brownlee July 21, 2020 at 6:01 am #

No. Each sample gets one opportunity to be used to update the model each epoch.

Obaid Ashraf July 20, 2018 at 3:42 pm #

REPLY

Good explanation and good example .. Thankyou and keep up the good work sir !!

Jason Brownlee July 21, 2018 at 6:29 am #

REPLY

Thanks.

Durgesh Kumar Trivedi July 20, 2018 at 5:25 pm #

REPLY

Very well explained and most simple possible way !!!.

Jason Brownlee July 21, 2018 at 6:31 am #

REPLY

I'm glad it helped.

Vijay July 20, 2018 at 5:46 pm #

REPLY

Absolutely, thanks for making this Dr. Jason – this eases life without hammering head and time for some on exploring several sources

I have quick question based on (below excerpt from your post)…could you please name / refer other procedures used to update parameters in the case of other algorithms.
****
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.
****

Jason Brownlee July 21, 2018 at 6:31 am #

REPLY

Glad it helped.

You can learn more about other algorithms here:
https://machinelearningmastery.com/start-here/#algorithms

Pospai July 20, 2018 at 7:53 pm #

REPLY

Très bien. Mais j'aurais aimé voir plus d'exemples. En tout cas GRAND MERCI !

Jason Brownlee July 21, 2018 at 6:33 am #

REPLY

Thanks. Did the example at the end help?

Emrah July 21, 2018 at 4:23 am #

REPLY

in modern deeeep learning approaches, i almost always encounter that people save their models after some number of epoches (or some time period) while visualizing some kind of performance metrics to evaluate the next values for the hyperparams, thereafter do they carry out their experiments for the next epochs. So we can call this procedure as ‘mini epoch stochactic deep learning’. Thanks.

REPLY

Jason Brownlee July 21, 2018 at 6:39 am #

Thanks for sharing.

REPLY

Jerry Dowetin July 22, 2018 at 8:46 pm #

This is brilliant and straight forward. Thanks for the mini course Dr. Brownlee

REPLY

Jason Brownlee July 23, 2018 at 6:08 am #

I’m glad it helped.

REPLY

Desi Mazdur August 1, 2018 at 5:58 pm #

Hello Dr Jason,

Thank you again for a great blog post. For time series data in LSTM, does it ever makes sense to have the size of a batch more than one?
I have searched and searched and I could not find any example where the batch size is more than one but I have also not found anyone saying that it does not make sense.

REPLY

Jason Brownlee August 2, 2018 at 5:59 am #

Yes, when you want the model to learn across multiple subsequences.

I have some posts that demonstrate this scheduled.

REPLY

youssef August 30, 2018 at 10:47 am #

thank you for your explanation really very cair thanks again

REPLY

Jason Brownlee August 30, 2018 at 4:50 pm #

I'm happy that it helped.

Strange Xue October 30, 2018 at 12:26 am #

REPLY

I've read many blogs written by you about such things. It does help me a lot, thank you! 抱拳

Jason Brownlee October 30, 2018 at 6:02 am #

REPLY

Thanks, I'm glad to hear that the post helped.

Zsolt November 3, 2018 at 10:07 am #

REPLY

Thanks, great explanation. So far your blog is the best source for learning ML I've found (for beginners like me).

Jason Brownlee November 4, 2018 at 6:24 am #

REPLY

Thanks!

Jenna Ma November 10, 2018 at 10:47 pm #

REPLY

It is very clear. Thank you.
I also see 'steps_per_epoch' in some cases, what is that mean? Is it same as batches?

Jason Brownlee November 11, 2018 at 6:05 am #

REPLY

The number of batches to retrieve from a generator in order define an epoch.

Francisco November 22, 2018 at 9:09 pm #

REPLY

We love examples! Thank you so much!

Jason Brownlee November 23, 2018 at 7:48 am #

REPLY

Thanks.

Mohammad November 26, 2018 at 10:19 am #

REPLY

Thank you so much for your crystal clear explanation

Jason Brownlee November 26, 2018 at 2:01 pm #

REPLY

I'm happy you found it useful.

Fatma December 6, 2018 at 5:53 pm #

REPLY

Great explanation in an easy way. Thanks.

Jason Brownlee December 7, 2018 at 5:18 am #

REPLY

Thanks, I'm glad it helped.

sangeeth January 21, 2019 at 1:48 pm #

REPLY

Hi,
Updates are performed after each batches are over. I just used one sample and gave different batch_sizes in model.fit, why does the value change every time?…it should be able to take one batch size if there is only one sample, isn't it?

Jason Brownlee January 22, 2019 at 6:18 am #

REPLY

Sorry, I don't understand your question, can you elaborate please?

Alaa February 8, 2019 at 5:25 am #

REPLY

What a great explanation!
Never sent a reply to a tutorial, but cannot leave without saying Thanks Jason.
God bless you!

Jason Brownlee February 8, 2019 at 8:05 am #

REPLY

Thanks, I'm glad it helped!

ANita February 26, 2019 at 9:16 pm #

REPLY

Fantastic explanation !!!

Jason Brownlee February 27, 2019 at 7:26 am #

REPLY

Thanks.

sampath April 1, 2019 at 5:56 am #

REPLY

Well Explained Thanks!!

Jason Brownlee April 1, 2019 at 7:53 am #

REPLY

Thanks, I'm glad it helped.

Momo April 12, 2019 at 7:34 pm #

REPLY

Hello there,
I am currently working with Word2Vec. In connection with Epochs and batchSize I still don't understand exactly what a sample is. Above you describe that a sample is a single row of data. In my program I first edit my text file with a SentenceIterator so that I get one sentence per line and then I use a tokenizer to get single words in these lines. Is a sample in Word2Vec a word from the data set or is it a line (containing a sentence)? Thank you very much in Advance 🙂

Jason Brownlee April 13, 2019 at 6:27 am #

REPLY

The samples/epoch/batch terminology does not map onto word2vec. Instead you just have a training dataset of text from which you learn statistics.

Momo April 16, 2019 at 1:12 am #

REPLY

But with the program Word2Vec you also have the hyperparameters Epochs, Iterations and Batch Size, which you can set… Don't you think that they also influence the results from Word2Vec.
As I understood it now, a set passed as a Batch contains one sentence. However, I'm surprised that the number of iterations doesn't change if I vary the number of epochs and batch sizes but don't define iterations concretely. Do you know how that works?

Jason Brownlee April 16, 2019 at 6:51 am #

REPLY

Not really, I recommend this tutorial:
https://machinelearningmastery.com/develop-word-embeddings-python-gensim/

Sagara Sumathipala May 2, 2019 at 2:24 am #

REPLY

Very well explained in simple terms. Thanks.

Jason Brownlee May 2, 2019 at 8:06 am #

REPLY

Thanks.

Kaligambe Abraham May 10, 2019 at 2:27 pm #

REPLY

It's finally clear. Thank you

Jason Brownlee May 11, 2019 at 6:04 am #

REPLY

I'm happy to hear that.

Suraj Bhagat May 12, 2019 at 12:59 am #

REPLY

You are super Dr,
Thank you so much for writing in easy way to understand…. Also, try to add pics or graph or schematic diagram to represent your text. As I have seen here you gave one example, it makes many things with super clarity. In some previous post you added graph as well…

Thanks again
Please keep continue

Best regards
Suraj

Jason Brownlee May 12, 2019 at 6:44 am #

REPLY

Thanks for the suggestion!

Karan May 14, 2019 at 6:37 pm #

REPLY

Hi Jason. After every epoch, the accuracy either improves or sometimes not. For example, epoch 1 achieved accuracy of 94 and epoch 2 achieved an accuracy of 95. After the end of epoch 1 we get new weights (i.e updated after final epoch 1 batch). Is that the new weights used in epoch 2 begining to improve it from 94% to 95%? If yes, is that the reason for some epoch getting lower accuracy from the previous epoch due to the generalization of weights for the entire dataset? That's why we get good accuracy after running so many epochs due to better generalization?

Jason Brownlee May 15, 2019 at 8:12 am #

REPLY

Typically more training means better accuracy, but not always.

Sometimes it can be a good idea to stop training early, see this post on the topic:
https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/

Zakariya May 14, 2019 at 9:58 pm #

REPLY

Thanks! That was simple and easy to understand.

Jason Brownlee May 15, 2019 at 8:15 am #

REPLY

Thanks, I'm happy it helped!

saad wazir May 27, 2019 at 6:02 am #

REPLY

very well explained thankyou boy

Jason Brownlee May 27, 2019 at 6:52 am #

REPLY

Thanks.

bright stern May 28, 2019 at 5:54 am #

REPLY

Well explained with easy to understand example. thank you

Jason Brownlee May 28, 2019 at 8:21 am #

REPLY

Thanks, I'm glad it helped.

Rohaifa June 18, 2019 at 9:00 pm #

REPLY

Indeed, in the last example, the total number of mini-batches is 40,000, but this is true only if the batches are selected without shuffling the training data or selected with data shuffling but without repetition. Otherwise, if within one epoch the mini batches are constructed by selecting training data with repetition, we can have some points that appear more than once in one epoch (they appear in different mini batches in one epoch) and others only once. Therefore, the total number of mini-batches, in this case, may exceed 40,000.

Jason Brownlee June 19, 2019 at 7:54 am #

REPLY

Typically data is shuffled prior to each epoch.

Typically we do not select samples with replacement as it will bias the training.

Hani Younis July 15, 2019 at 4:44 pm #

REPLY

You Deserve Big Thank You letter for this explanation

Jason Brownlee July 16, 2019 at 8:12 am #

REPLY

Thanks, I'm glad it helped.

Pavan July 27, 2019 at 9:42 pm #

REPLY

thanks for this amazing blog post 🙂

If i have 1000 training samples and my batchsize=400 then i have to remove 200 samples from my training data , always my training data should be mulitple of the batchsize

Jason Brownlee July 28, 2019 at 6:44 am #

REPLY

Thanks.

No, the samples will be shuffled before each epoch, then you will get 3 batches, 300, 300 and 200.

It is better to choose a batch size that divides the samples evenly, if possible, e.g. 100, 200, or 500 in your case.

Yu Bai September 16, 2019 at 12:34 pm #

REPLY

Thank you so much! Such a nice explanation with an intuitive example in the end! Thank you!

Jason Brownlee September 16, 2019 at 2:13 pm #

REPLY

Thanks, I'm glad it helped.

ahmedhesham33 October 2, 2019 at 12:10 am #

REPLY

thanks for your great article , and i have a question
if i have the following settings and i am using fit_generator function
epochs =100
data=1000 images
batch = 10
step_per_epochs = 20
i know i should set the step_per_epochs = (1000/10)= 100 but if i set it to 20

Are these settings mean that the model will be trained using only part of the training data (at each epoch will use the same 200 images(batch*step_per_epochs)) and not used the all 1000 images ?
or it will use first 200 images in dataset in first epoch then the following 200 images in the second epoch and so on (will divided the 1000 images on each 5 epochs) and model will be trained 20 times using the whole training dataset in the 100 epochs
Thanks

Jason Brownlee October 2, 2019 at 8:00 am #

REPLY

Yes, only 200 images per epoch will be used.

Carol October 2, 2019 at 5:13 am #

REPLY

olá, tudo bem? Muito obrigada pela explicação. Gostaria de saber se o senhor sabe o que é Batch Accumulation, Random seed e Validation Interval (em epocas)

Jason Brownlee October 2, 2019 at 8:06 am #

REPLY

Yes.

Batch accumulation is the error collected from the samples in one match used to update the weights.

Random seed is the starting point for the random number generator:
https://machinelearningmastery.com/introduction-to-random-number-generators-for-machine-learning/

What exactly do you mean by validation interval? What context? Perhaps you mean validation dataset:
https://machinelearningmastery.com/difference-test-validation-datasets/

Bipasha October 5, 2019 at 5:29 pm #

REPLY

Sir thank you so much for this excellent tutorial.
Can you tell me how to run the model on a similar test dataset after training the model?

Jason Brownlee October 6, 2019 at 8:15 am #

REPLY

Yes, you can use model.predict(), see examples here:
https://machinelearningmastery.com/make-predictions-scikit-learn/

sahil October 7, 2019 at 11:42 pm #

REPLY

Great explanation, keep sharing your knowledge,
Thank you very much.

Jason Brownlee October 8, 2019 at 8:03 am #

REPLY

Thanks!

Milind Dalvi October 15, 2019 at 10:27 am #

REPLY

Hello Jason,

If I were to create my own custom batches say within the model.fit_generator() method.

Do we create new randomly sampled batches for each epoch or do we just create batches at __init__ and use them without any changes throughout the training?

What's the recommended way?

P.S. If I randomly sample batches each epoch I see spikes in val_acc, not sure it's bcoz of that though!

Jason Brownlee October 15, 2019 at 1:46 pm #

REPLY

Great question.

It is important to ensure that each batch is representative (within reason), and that each epoch of batches is broadly representative of the problem.

If not, you will push the weights all over the place or back/forth on each update not not generalize well.

Milind Dalvi October 16, 2019 at 12:44 am #

REPLY

Hello Jason,

Thank you for your response.

I also just confirmed that Keras would separate the provided X in mini-batches only once before entering the epoch loop.

Here is the link to code https://github.com/keras-team/keras/blob/f242c6421fe934680644415 51cdab66e70f631d8/keras/engine/training_generator.py#L160

Jason Brownlee October 16, 2019 at 8:06 am #

REPLY

Yes.

Milind Dalvi October 17, 2019 at 5:03 am #

REPLY

Good Morning Jason,

A question came in my mind today.

What happens while training a Neural Network in mini-batches when the class labels are imbalanced. Are we suppose to stratify the batches?

Becoz it seems like my NN is only predicting dominant class no matter what I do!

Jason Brownlee October 17, 2019 at 6:42 am #

REPLY

Great question. We get bad times!

Sometimes the experts would say to alternate classes in each batch. Sometimes stratify. It might be problem/model dependent. I'm thinking back to this book:

https://machinelearningmastery.com/neural-networks-tricks-of-the-trade-review/

Nevertheless, imbalanced data is a pain regardless of your update strategy. Oversampling the training set is a great solution.

Milind Dalvi October 19, 2019 at 7:17 am #

REPLY

Thanks, Jason.

I will surely take a look at the book.

Btw I am actually in ranking business. So I got very few 1st and 2nd rankers but a lot of 3rd and above, somewhere as (10%, 10%, 80%) respectively.

What I did is, I took a different perspective on the problem and converted my imbalanced multiclass dataset to an equalized binary dataset.

I converted.

Racing Car 1: 1st Rank
Racing Car 2: 2nd Rank
Racing Car 3: 3rd Rank
Racing Car 4: 4th Rank

to,

Racing Car1, Racing Car2 = 0
Racing Car2, Racing Car1 = 1
Racing Car1, Racing Car3 = 0
Racing Car3, Racing Car1 = 1
Racing Car1, Racing Car4 = 0
Racing Car4, Racing Car1 = 1
Racing Car2, Racing Car3 = 0
Racing Car3, Racing Car2 = 1
Racing Car2, Racing Car4 = 0
Racing Car4, Racing Car2 = 1

and so on… where Target is now the winning side!

Jason Brownlee October 20, 2019 at 6:12 am #

REPLY

Fascinating! Thanks for sharing.

Thiyagarajan Paramadayalan November 15, 2019 at 6:18 pm #

REPLY

very well explained, Jason. thanks.

Jason Brownlee November 16, 2019 at 7:21 am #

REPLY

Thanks!

Victor December 18, 2019 at 5:57 pm #

REPLY

Well explained. Thanks.

Jason Brownlee December 19, 2019 at 6:25 am #

REPLY

Thanks.

Rajib Das December 29, 2019 at 12:15 pm #

REPLY

HI Jason – I have a question – If I understood it correctly , the weights and bias are updated after running through the batch , so any change after the batch is run is applied to the next batch ? And it continues so on.

Jason Brownlee December 30, 2019 at 5:56 am #

REPLY

Correct.

Ei Ei Mon January 6, 2020 at 11:27 pm #

REPLY

Well explained. Thanks Dr. Jason.

Jason Brownlee January 7, 2020 at 7:23 am #

REPLY

Thanks!

Ali January 12, 2020 at 8:38 am #

REPLY

Thanks Dr. Jason

Jason Brownlee January 13, 2020 at 8:16 am #

REPLY

You're welcome.

Yiding January 17, 2020 at 1:38 am #

REPLY

Super straightforward and helpful. Thank you!

Jason Brownlee January 17, 2020 at 6:03 am #

REPLY

Thanks, I'm happy it was useful.

mingxing wang February 4, 2020 at 8:59 pm #

REPLY

Hi. Sir.
I am very thankful to you.
Now I am in the middle of studying hands on machine learning and Part 2 in chapter 11 I can't understand the meaning of batch. At first I think neural network must train by sample one by one. But they said "batch" and I can't understand on earth.
But your article gives me a good sense about batch.
I understand batch completely with only one question.
How can I use gradient method with batch?
I mean in one sample it is understandable.
But with batch I don't understand how to evaluate error.
Thank you.

Jason Brownlee February 5, 2020 at 8:08 am #

REPLY

It does go one by one, but after "batch" number of samples the weights are updated with accumulated error.

Moe February 17, 2020 at 9:32 pm #

REPLY

Greetings, Dr. Brownlee

I was hoping you would be able to help me with my rather long confusing questions(sorry). I am very new to deep learning.

I do think I understand the definition for iteration, batch, and epoch but I am not so sure about them in regards to them in-practice.
So I will give an example and hope that you could help me that way.

Now, I (hopefully) understand that iteration is the parameter in which it will pass through a set of samples through and back the model where Epoch will pass through (and back) all of the samples.

Assuming I have a dataset of 50,000 points.

The following parameters are set in Python/Keras as.
batch_size = 64
iterations = 50
epoch = 35

So, my assumption on what the code is doing is as follows:

50,000 samples will be divided by the batch size (=781.25 =~ 781).
So now I have 64 blocks (batches) of the whole dataset, with each containing 781 samples.

For iteration 1:
All of the blocks from 1 to 64 will be passed through the model. Each block/batch resulting in its own accuracy metric, resulting in 64 accuracy numbers that will be averaged at the end.
This above process will be repeated 35 times (the number of Epochs) resulting in 35 averaged accuracies, and as the Epoch increases along the iteration, the accuracy is (theoretically) going to be better than the previous accuracy.
After the iteration is done, the weights of the nodes will be updated and be used for iteration 2.

The above process will be repeated 50 times, as I have 50 iterations.

Is what I said true so far?

That is my major confusion at the moment.

My other rather question is in regards to the accuracy and the 3 mentioned hyperparameters.

I have been playing around with the Addition RNN example over at Keras, where they set batch to 128, iterations to 200 and epochs to 1.

My question is, if you set batch to 2048, iterations to 4 with 50 epochs. Not only will you not reach an Accuracy of 0.999x at the end (you almost always reach this accuracy in other combinations of the parameters). However, your accuracy will actually dip substantially.
I have put the results in the following pastebin link [https://pastebin.com/gV1QKxH3]
and would like to bring your attention to Epoch 41/50 where the accuracy almost halved itself.
Is there any reason at all to this?
My only thought process is maybe the weights were somehow reseted but that seems extremely unlikely.

Thank you greatly for your time, as always

I hope to hear from you soon

Regards,
Moe

Jason Brownlee February 18, 2020 at 6:20 am #

REPLY

What is an iteration?

Mohamad Almustafa February 18, 2020 at 4:45 pm #

REPLY

An iteration in deep learning, is when all of the batches are passed through the model.
The epochs will repeat this process (35 times).
At the end of this process, the model will be updated with new weights.

For iteration 2, the same process will happen again, but this time the model will be using its new weights from the previous iteration.

I hope this helped

Jason Brownlee February 19, 2020 at 7:58 am #

REPLY

I would call it one epoch.

Yves February 18, 2020 at 8:02 pm #

REPLY

Nice explanation, thank you!
Just to make sure i understood:
if one would do a Batch GD, then one would not need any epoch, right?
Namely, it is really the different compositions of the mini-batches in each epoch, that make the epochs different, right?

Jason Brownlee February 19, 2020 at 8:01 am #

REPLY

No. One epoch would equal one batch. Still need many epochs.

mah.max March 26, 2020 at 3:37 am #

REPLY

Thank you very much Jason. I saw that you used sometimes epoch in this way
model.fit(X_train,y_train,epochs=50) and sometimes in a for loop like this
for iter in range(50):
model.fit(X_train,y_train,epochs=1)
according to the definition of epoch in both cases, 50 times the learning algorithm will work through the entire training dataset. Is it correct? Are they doing the same? if not could you please tell me the difference?

Jason Brownlee March 26, 2020 at 8:00 am #

REPLY

They are the same thing.

The manual loop gives more control in case you want to do something each epoch.

mah.max March 27, 2020 at 7:22 am #

REPLY

Thank you very much.

Jason Brownlee March 27, 2020 at 8:03 am #

REPLY

You're welcome.

Akilu Rilwan March 29, 2020 at 10:46 pm #

REPLY

Thank you Jason, you always save my search.

Jason Brownlee March 30, 2020 at 5:33 am #

REPLY

You're welcome.

Rajrudra April 21, 2020 at 2:44 am #

REPLY

Thanks , I had heard stochastic gradient descent but here, just with one line, you have cleared the basic concept. I am just a novice but this might be a good starting point

Jason Brownlee April 21, 2020 at 6:02 am #

REPLY

Thanks, I'm happy it helped!

Rajrudra April 21, 2020 at 2:45 am #

REPLY

Also thanks for the batch concept

Jason Brownlee April 21, 2020 at 6:03 am #

REPLY

You're welcome.

nishan April 21, 2020 at 7:21 pm #

REPLY

How should epoch, batch size affect the weight? How can you describe the relation between

Jason Brownlee April 22, 2020 at 5:53 am #

REPLY

What do you mean exactly, can you please elaborate?

Mira April 21, 2020 at 7:29 pm #

REPLY

How many times the weight update, does that depends on the batch_Size and number of epochs? or it should stop when it reaches the best weight?

Jason Brownlee April 22, 2020 at 5:53 am #

REPLY

Yes, the number of times the weights are update depends on the batch size and epochs – this is mentioned in the tutorial.

There is no best weight – we stop when the model stops improving or when we have run out of time.

Sakshi April 23, 2020 at 5:09 pm #

REPLY

Hi Sir

Thank you for helping us with your tutorials.
I just love your site. Thanks for making me a better data scientist 🙂

Jason Brownlee April 24, 2020 at 5:37 am #

REPLY

Thanks!

Ehsan April 25, 2020 at 12:10 am #

REPLY

Thank you Jason for good explanation.
Please let me know about the following issue:
What happen when one bach feed to network. Error of each sample calculated, and then get the average of error of all samples, and then gradient descent use this average to update the weights or it works in another way?

Jason Brownlee April 25, 2020 at 6:51 am #

REPLY

Something like that.
https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/

Behnaam April 26, 2020 at 2:57 am #

REPLY

Jason, many thanks for the explanation. Is this statement correct?

Increasing the batch size makes the error surface smoother; so, the mini-batch gradient descent is preferable over the stochatic one. On the other hand, we might want to keep the batch size not so large so that the network has enough number of update chances (iterations) using the whole samples.

Jason Brownlee April 26, 2020 at 6:18 am #

REPLY

Maybe.

Sometimes we want a noisy estimate of the gradient to bounce around the parameter space and find a good/better set of parameters.

Ricardo April 26, 2020 at 3:22 pm #

REPLY

You are too much sir. Thank you.

Jason Brownlee April 27, 2020 at 5:29 am #

REPLY

You're welcome.

Isic May 5, 2020 at 2:32 am #

REPLY

Thank you for this amazing explanation.

I wanted to ask a question before I figured out the answer :)!
Actually, the question was, "why we need to go through the entire training dataset more than once ?", and I think the answer is that, in the first epoch, weight are randomly initialized, but in the 2nd one, weight are already updated in every batch, and so on. In other words: the weights of the epoch t are "transferred" to the epoch t+1. This way, the learning curve of the training set is going down.

Please correct me if I'm wrang.

Jason Brownlee May 5, 2020 at 6:33 am #

REPLY

Great question!

The models learn iteratively, slowly. If we learn too fast, we over learn and cannot generalize well to new data. We – the community – have learned this over 40+ years.

Raj May 14, 2020 at 6:58 pm #

REPLY

Thank you so much for clarifying the concept of epoch and batch size, in very simple and easy terminology.

Jason Brownlee May 15, 2020 at 5:57 am #

REPLY

You're welcome!

Riyaz December 28, 2020 at 9:25 am #

REPLY

Hello sir, thank you for your amazing posts.

So to make things simpler for me, say I have a linear regression model and we are in epoch 3 (it has gone through the entire dataset twice and now it's doing it a third time), we still have the same dataset but the only difference is that we have updated our parameters (the coefficient values) twice. When we updated it the first time, we developed a better model with coefficient more suited to the training data, then when in epoch 2, we used the updated model from epoch 1 and used that to train on the same dataset and then again use our model with updated coeffieicients from epoch 2 in epoch 3. So summing up, at each epoch we have a slightly better model than the previous one which allows us to lower down the error rate? Is my understanding right?

Kind regards

Jason Brownlee December 28, 2020 at 9:51 am #

REPLY

You're welcome.

Exactly!

Sanjay Talbar May 16, 2020 at 5:15 pm #

REPLY

excellent explanation!
My question is"For SGD, traing time will be more as compared to mini batch and batch gradient descent Algo?

Jason Brownlee May 17, 2020 at 6:33 am #

REPLY

Batch gradient descent will be faster to execute than mini-batch gradient descent.

arman May 21, 2020 at 10:05 am #

REPLY

hi tanks for great content
sorry for asking irrelevant question
is steps_per_epoch in Imagedatagenerator just for saving time ?
thank you

Jason Brownlee May 21, 2020 at 1:41 pm #

REPLY

No. From memory, it is a proxy for the number of samples in an epoch or the number of updates, I don't recall which.

Pravin Girase July 18, 2020 at 4:00 pm #

REPLY

Jason

I have question about shuffelimg the data during training. What I have observed that if I run the same code multiple times the results are not the same ifbi am using shuffled data. So how do I get confidence that my code is correct when the accuracy and training losses keep changing.

So sometimes I end up fixing the training data set and validation data set. I want to know if this is correct practice. If not then how to believe that whatever results I am getting are good enough?

Jason Brownlee July 19, 2020 at 6:24 am #

REPLY

Good question, you must evaluate your model many times and report the mean and standard deviation of the model's performance.

This will help:
https://machinelearningmastery.com/faq/single-faq/why-do-i-get-different-results-each-time-i-run-the-code

Omar September 2, 2020 at 12:42 am #

REPLY

Jason, you are a great teacher. I like the way you explain.
I also like that despite trying to "de-academize" the teaching, you still put references to the original papers. You strike a fairly nice balance there.

Thanks a lot.

Omar

Jason Brownlee September 2, 2020 at 6:30 am #

REPLY

Thanks Omar.

John September 8, 2020 at 1:26 pm #

REPLY

Good to know. Thanks for the clear explanation!

Jason Brownlee September 8, 2020 at 1:39 pm #

REPLY

You're welcome.

Shivan October 11, 2020 at 1:22 am #

REPLY

Hello sir
If we use four different pre-trained network, can we set different number of epoch for every networks? For example: 15 epoch for Alexnet, 20 epoch for vgg16 and so on? I will make a comparison among these networks

Thanks

Jason Brownlee October 11, 2020 at 6:53 am #

REPLY

Sure. Train them separately and save to file, then load them later for your ensemble. I have tutorials on this, search for deep learning ensemble.

Unnikrishnan November 4, 2020 at 6:42 pm #

REPLY

Thank You Jason.
Your articles on Machine Learning and Deep Learning are informative and great.
For me , its the go-to-site on these topics. Thanks.

Jason Brownlee November 5, 2020 at 6:32 am #

REPLY

Thank you!

saran November 5, 2020 at 9:05 pm #

REPLY

what will be the maximin total possible number of epochs for your examples of 200 samples and batch size of 5 any standard formula

Jason Brownlee November 6, 2020 at 5:54 am #

REPLY

Infinite. There's no maximum. You train until the model stops improving.

Mohammad Kasra Habib December 11, 2020 at 1:43 am #

REPLY

Hi,

Regarding what you said "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."

It means that the loss is computed only when one batch passed through the net, and then the gradient update takes place.

So, how the loss is carried for 5 samples inside the batch?

E.g., my net is processing the 1st batch and my loss function is MAE. Basically, the neural network calculates MAE for each individual instance in the batch, then average it, and eventually pass it to the optimizer (in this example lets say it is SGD) and SGD multiplies it by the learning rate and subtract it from the net's weights to accomplish the gradient update. Is this correct!

If it is correct, then the loss is not computed at the end of each epoch and it only specifies how many iterations should be done on each batch.

Thanks in advance!
I like your tutorials, they are really great. KEEP THEM UP!

Jason Brownlee December 11, 2020 at 6:39 am #

REPLY

The loss is averaged over the five samples in the batch.

Swetha Balaji December 14, 2020 at 4:47 pm #

REPLY

Finally understood with the example stated above d/b epoch and Batch. Thank you

Jason Brownlee December 15, 2020 at 6:16 am #

REPLY

I'm happy that it helped.

SuSa December 21, 2020 at 2:48 am #

REPLY

Dear Jason,

Thanks for the simple explanation. I had read so many articles on ANN, without any clarity on the subject. This suddenly made everything clear.

Jason Brownlee December 21, 2020 at 6:40 am #

REPLY

You're welcome, I'm happy that it helped.

Giriraj Pawar January 11, 2021 at 2:43 am #

REPLY

"an epoch that has one batch is called the batch gradient descent learning algorithm".

Batch Gradient Descent. Batch Size = Size of Training Set
Stochastic Gradient Descent. Batch Size = 1
Mini-Batch Gradient Descent. 1 < Batch Size < Size of Training Set

as per the above explanation:
if Batch Size = 1 then it should be called Stochastic Gradient Descent, why it is being called batch gradient descent learning algorithm.

Jason Brownlee January 11, 2021 at 6:20 am #

REPLY

Here we are saying if the batch size equals the entire training dataset, this is called "batch gradient descent".

If the batch size equals one row/sample, this is called "stochastic gradient descent".

Questioner January 18, 2021 at 4:04 pm #

REPLY

If more Epochs make more learning, why don't we set it to a large number (eg. 10,000 or 100,000) like that? So the result we be better and better

Jason Brownlee January 19, 2021 at 6:34 am #

REPLY

Good question.

Diminishing returns – e.g. no further improvement or even making the model worse (overfitting) after some point.

REPLY

Vesto January 21, 2021 at 9:45 pm #

Thank you for this great work Jason. I found myself asking that question today after using these terms for a while now!

REPLY

Jason Brownlee January 22, 2021 at 7:19 am #

You're welcome!

REPLY

Gabba February 10, 2021 at 5:13 am #

Amazing.
Thanks

REPLY

Jason Brownlee February 10, 2021 at 8:12 am #

You're welcome!

REPLY

MER February 15, 2021 at 3:26 pm #

Your articles are amazing, it's so clear. Thank you so much

REPLY

Jason Brownlee February 15, 2021 at 4:01 pm #

Thank you!

REPLY

Bilew February 27, 2021 at 7:25 pm #

it is a short and brief explanation thanks.

REPLY

Jason Brownlee February 28, 2021 at 4:33 am #

You're welcome!

Nilu R Salim March 1, 2021 at 4:19 pm #

REPLY

That's why we go for early stopping, to avoid overfitting.

Jason Brownlee March 2, 2021 at 5:42 am #

REPLY

Yes, if we have sufficient additional data, early stopping can be very effective: https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/

Ajoku Kingsley kelechi March 22, 2021 at 8:14 am #

REPLY

Thank you Jason. Please does the weight get updated after each batch sample have been passed or after an epoch cycle have been made?

Jason Brownlee March 23, 2021 at 4:53 am #

REPLY

Yes, model weights are updated after each batch.

David Espinosa April 9, 2021 at 8:42 am #

REPLY

One of the nicest summaries I have read so far about this subject!

Thank you very much Dr. Bronwlee!

Jason Brownlee April 10, 2021 at 5:57 am #

REPLY

Thanks, I'm happy it helped.

Shaima' May 24, 2021 at 8:37 pm #

REPLY

Thanks a lot for your wonderful explanation.

Jason Brownlee May 25, 2021 at 6:07 am #

REPLY

You're welcome.

Roshan Basnet June 16, 2021 at 4:35 am #

REPLY

Very helpful content with clear explanation.

Jason Brownlee June 16, 2021 at 6:22 am #

REPLY

Thanks.

chirag palan June 23, 2021 at 6:11 am #

REPLY

Befor googling this question, i thought I will be only one who is looking for "difference between epoch and batch size" but after looking at all the comments I was very much surprised.
But it is very clear now.
Thanks Jason

Jason Brownlee June 24, 2021 at 5:55 am #

REPLY

I'm happy it helped!

Taher August 13, 2021 at 12:01 pm #

REPLY

Thanks a lot Jason! Very well explained. These terminologies are very confusing for beginners. Your article solved this mystery for me.

Adrian Tam August 14, 2021 at 2:33 am #

REPLY

Thank you. Glad you like it.

MS August 26, 2021 at 4:01 pm #

REPLY

Thank you so much for this article. Your articles on machine learning are so easy to grasp and I always follow them.

Adrian Tam August 27, 2021 at 6:00 am #

REPLY

Hope you enjoyed it! Thank you.

Ahmed Hamadto September 2, 2021 at 6:35 pm #

REPLY

Your articles are amazing, and a testament to your understanding of machine learning, your work is vital to the community and is very highly appreciated, please keep it up. I may not write on every article but know that each and every one of them is appreciated.

Thank you, once again.

Jason Brownlee September 3, 2021 at 5:29 am #

REPLY

Thank you deeply!

Aitha Sahith October 25, 2021 at 4:43 am #

REPLY

Yeah it helped thanks!!!!

Nico H. November 5, 2021 at 2:01 pm #

REPLY

It helped me a lot! Thank you! 🙂

Adrian Tam November 7, 2021 at 8:03 am #

REPLY

Glad you liked it!

Fahad November 6, 2021 at 9:01 pm #

REPLY

Amazing blog! Explained the difference between the two so clearly.

Amit December 29, 2021 at 10:13 pm #

REPLY

Great explanation, you nailed it..

James Carmichael December 30, 2021 at 10:06 am #

REPLY

Thank you for the feedback and kind words Amit!

Aditya February 5, 2022 at 1:46 am #

REPLY

Dear Sir,
Thank you for the example at the end of this blog. I have a question regarding epochs. I am posing the question using the same example you used.

As per the example, we have 1000 epochs with each epoch having 40 batches. I am curious whether the composition of these batches are the same for each epoch or are is shuffled with every epoch.

Regards,
Aditya

James Carmichael February 5, 2022 at 11:02 am #

REPLY

Hi Aditya…This can be controlled. The following discussion may be of interest:

https://datascience.stackexchange.com/questions/53927/are-mini-batches-sampled-randomly-in-keras-sequential-fit-method

Julie Smith May 9, 2022 at 10:34 am #

REPLY

This is very helpful but I'm a little confused about batch size vs optimizers. I'm using pytorch code I got online and it uses a mini-batch gradient descent (i.e. they define a batch size of 128) and later in the code they call a SGD (stochastic gradient descent) optimizer. Can one use a mini batch gradient descent with a SGD optimizer? Is that common? Also, I was playing around with batch size and used a batch size of 1 (which is defined above as a stochastic gradient descent) and left the optimizer the same. Does this make sense to do?

James Carmichael May 9, 2022 at 10:57 am #

REPLY

Hi Julie…the following is a great resource that may prove helpful:

https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/

ADUGE August 19, 2022 at 9:08 pm #

REPLY

Waooo this is incisive and wonderful

Percy August 24, 2022 at 12:01 pm #

REPLY

I understand. Thank you.

James Carmichael August 25, 2022 at 7:46 am #

REPLY

You are very welcome Percy! Keep up the great work!

Leave a Reply

Name (required)

Email (will not be published) (required)

SUBMIT COMMENT

Welcome!
I'm *Jason Brownlee* PhD
and I **help developers** get results with **machine learning**.
Read more

Never miss a tutorial:

Picked for you:

Your First Deep Learning Project in Python with Keras Step-by-Step

How to Grid Search Hyperparameters for Deep Learning Models in Python with Keras

Regression Tutorial with the Keras Deep Learning Library in Python

Multi-Class Classification Tutorial with the Keras Deep Learning Library

How to Save and Load Your Keras Deep Learning Model