Gradient Descent For Machine Learning

by **Jason Brownlee** on March 23, 2016 in **Understand Machine Learning Algorithms**

Optimization is a big part of machine learning. Almost every machine learning algorithm has an optimization algorithm at it's core.

In this post you will discover a simple optimization algorithm that you can use with any machine learning algorithm. It is easy to understand and easy to implement. After reading this post you will know:

- What is gradient descent?
- How can gradient descent be used in algorithms like linear regression?
- How can gradient descent scale to very large datasets?
- What are some tips for getting the most from gradient descent?

Let's get started.

Gradient Descent For Machine Learning
Photo by Grand Canyon National Park, some rights reserved.

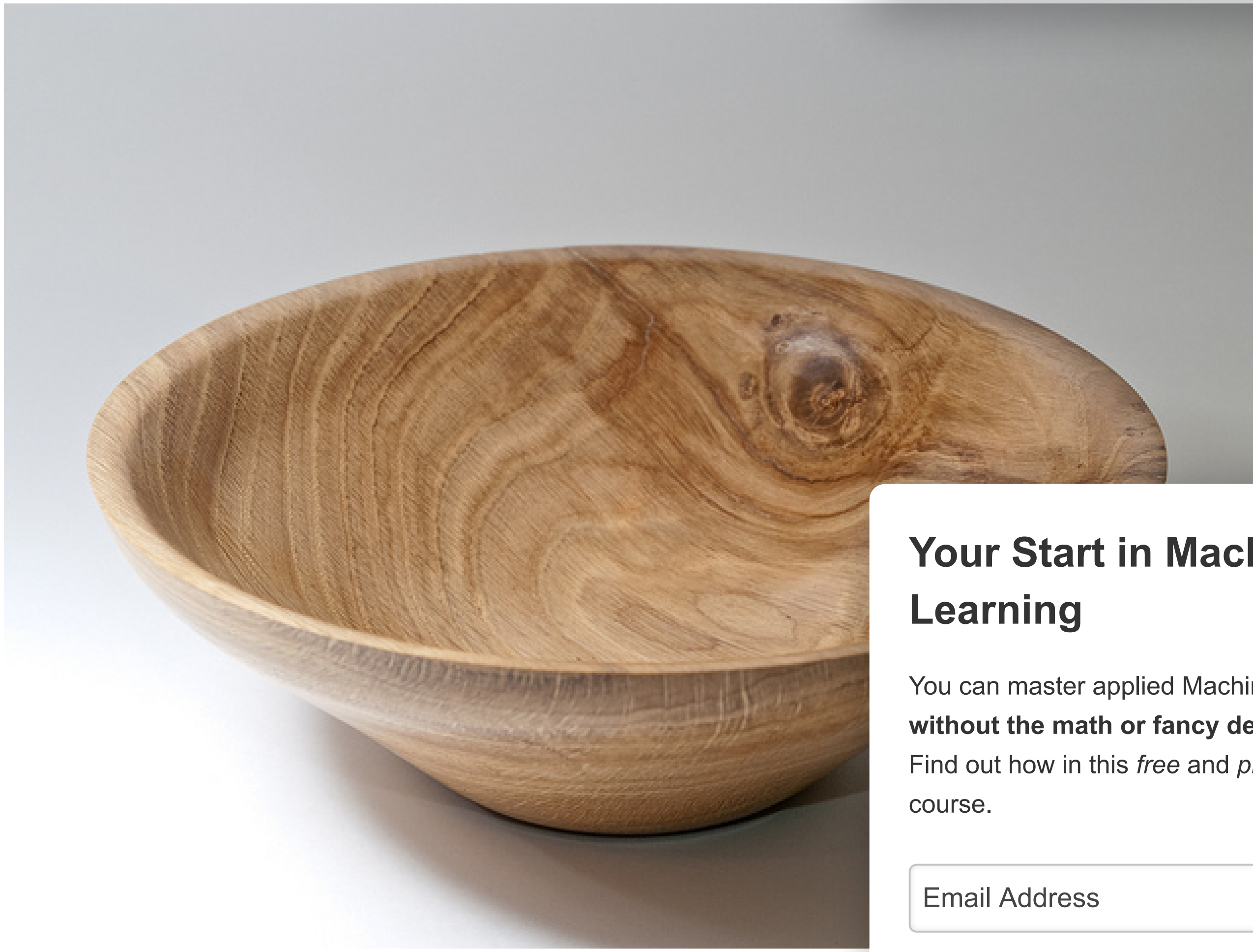
Gradient Descent

Gradient descent is an optimization algorithm used to find the values of parameters (coeffi ion (cost).

Gradient descent is best used when the parameters cannot be calculated analytically (e.g. y an optimization algorithm.

Intuition for Gradient Descent

Think of a large bowl like what you would eat cereal out of or store fruit in. This bowl is a p



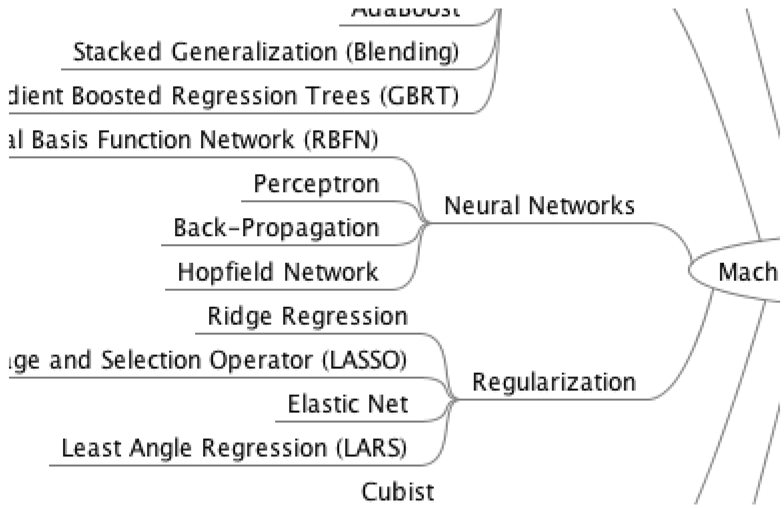
Large Bowl
Photo by William Warby, some rights reserved

A random position on the surface of the bowl is the cost of the current values of the coefficients (cost).

The bottom of the bowl is the cost of the best set of coefficients, the minimum of the function.

The goal is to continue to try different values for the coefficients, evaluate their cost and select new coefficients that have a slightly better (lower) cost.

Repeating this process enough times will lead to the bottom of the bowl and you will know the values of the coefficients that result in the minimum cost.



Sample of the handy machine learning algorithms mind map.

Gradient Descent P

The procedure starts off with initial values for the coefficient or coefficients for the function.

coefficient = 0.0

The cost of the coefficients is evaluated by plugging them into the function and calculating

cost = f(coefficient)

or

cost = evaluate(f(coefficient))

The derivative of the cost is calculated. The derivative is a concept from calculus and refers to the slope of the function at a given point. We need to know the slope so that we know the direction (sign) to move the coefficient values in order to get a lower cost on the next iteration.

delta = derivative(cost)

Now that we know from the derivative which direction is downhill, we can now update the c a) must be specified that controls how much the coefficients can change on each update.

coefficient = coefficient – (alpha * delta)

This process is repeated until the cost of the coefficients (cost) is 0.0 or close enough to zero to be good enough.

You can see how simple gradient descent is. It does require you to know the gradient of your cost function or the function you are optimizing, but besides that, it's very straightforward. Next we will see how we can use this in machine learning algorithms.

Batch Gradient Descent for Machine Learning

The goal of all supervised machine learning algorithms is to best estimate a target function (f) that maps input data (X) onto output variables (Y). This describes all classification and regression problems.

Some machine learning algorithms have coefficients that characterize the algorithms estim have different representations and different coefficients, but many of them require a proces t result in the best estimate of the target function.

Common examples of algorithms with coefficients that can be optimized using gradient de ion.

The evaluation of how close a fit a machine learning model estimates the target function c specific to the machine learning algorithm. The cost function involves evaluating the coeffi g a prediction for the model for each training instance in the dataset and comparing the predic sum or average error (such as the Sum of Squared Residuals or SSR in the case of linear

From the cost function a derivative can be calculated for each coefficient so that it can be u bed above.

The cost is calculated for a machine learning algorithm over the entire training dataset for each iteration of the gradient descent algorithm. One iteration of the algorithm is called one batch and this form of gradient descent is referred to as batch gradient descent.

Batch gradient descent is the most common form of gradient descent described in machine learning.

Stochastic Gradient Descent for Machine Learning

Gradient descent can be slow to run on very large datasets.

Because one iteration of the gradient descent algorithm requires a prediction for each instance in the training dataset, it can take a long time when you have many millions of instances.

In situations when you have large amounts of data, you can use a variation of gradient descent called stochastic gradient descent.

In this variation, the gradient descent procedure described above is run but the update to the coefficients is performed for each training instance, rather than at the end of the batch of instances.

The first step of the procedure requires that the order of the training dataset is randomized. This is to mix up the order that updates are made to the coefficients. Because the coefficients are updated after every training instance, the updates will be noisy jumping all over the place, and so will the corresponding cost function. By mixing up the order for the updates to the coefficients, it harnesses this random walk and avoids it getting distracted or stuck.

The update procedure for the coefficients is the same as that above, except the cost is not calculated for one training pattern.

The learning can be much faster with stochastic gradient descent for very large training da passes through the dataset to reach a good or good enough set of coefficients, e.g. 1-to-1

Tips for Gradient Descent

This section lists some tips and tricks for getting the most out of the gradient descent algor

- **Plot Cost versus Time**: Collect and plot the cost values calculated by the algorithm e g gradient descent run is a decrease in cost each iteration. If it does not decrease, try re
- **Learning Rate**: The learning rate value is a small real value such as 0.1, 0.001 or 0.0 which works best.
- **Rescale Inputs**: The algorithm will reach the minimum cost faster if the shape of the cost function is not skewed and distorted. You can achieved this by rescaling all of the input variables (X) to the same range, such as [0, 1] or [-1, 1].
- **Few Passes**: Stochastic gradient descent often does not need more than 1-to-10 passes through the training dataset to converge on good or good enough coefficients.
- **Plot Mean Cost**: The updates for each training dataset instance can result in a noisy plot of cost over time when using stochastic gradient descent. Taking the average over 10, 100, or 1000 updates can give you a better idea of the learning trend for the algorithm.

Summary

In this post you discovered gradient descent for machine learning. You learned that:

- Optimization is a big part of machine learning.
- Gradient descent is a simple optimization procedure that you can use with many machine learning algorithms.
- Batch gradient descent refers to calculating the derivative from all training data before calculating an update.
- Stochastic gradient descent refers to calculating the derivative from each training data instance and calculating the update immediately.

Do you have any questions about gradient descent for machine learning or this post? Leave a comment and ask your question and I will do my best to answer it.

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 →

‹ Overfitting and Underfitting With Machine Learning Algorithms

Linear Regression for Machine Learning ›

68 Responses to *Gradient Descent For Machine Learning*

Victor Garcia March 23, 2016 at 6:48 pm #

Thanks for all your work, Jason.
It's very helpful

Jason Brownlee July 16, 2016 at 9:08 am #

You're welcome Victor

Sara Ebrahim June 15, 2016 at 6:10 pm #

REPLY

It is the best and simplest explaination I have ever read.

Keep it up.

Jason Brownlee June 16, 2016 at 5:35 am # REPLY

Thanks Sara.

Aravind Krishnakumar July 16, 2016 at 7:17 am # REPLY

Beautifully Explained ! Thank you very much

Jason Brownlee July 16, 2016 at 9:08 am #

You're welcome Aravind, I'm glad it was useful.

krishna August 29, 2016 at 8:37 am #

Hey Jason,

Can you explain what do you mean by update?

Jason Brownlee August 30, 2016 at 8:23 am # REPLY

Hi Krishna, I mean change the coefficients that are being optimized.

chakradhara rao September 8, 2016 at 12:02 pm # REPLY

Hi Brownlee,
Thanks for sharing the topic . your explanation is easy and crystal clear .

Jason Brownlee September 9, 2016 at 7:17 am # REPLY

I'm glad you found it useful.

Carmen September 17, 2016 at 3:10 am # REPLY

The post mentions "The first step of the procedure requires that the order of the train
be done? At the start of each epoch or before running the entire operation?

I've tried both but cannot see any noticeable difference looking simply at the plot.

Thanks

Jason Brownlee September 17, 2016 at 9:33 am #

Hi Carmen, generally, it is a good idea to randomize prior to each epoch. The ch
other out if sample order is not changed, resulting in worse performance.

Carmen September 17, 2016 at 3:17 am # REPLY

I also have another question.

I've been going through the tutorial for linear regression and it seems to me that alpha and the number of epochs are what you'd call tuning parameters, is that right? Meaning we need to try different values to see which ones yield the best prediction (on the training dataset, at least).

Using the number of epochs as an example, I've created a loop to sequentially try 4, 5, 6… 100 epochs and for each epoch setting I've saved the
RMSE value obtained and then plotted RMSE as a function of epoch setting chosen. This als

the lowest RMSE on the training dataset at least.

Would you say that this is a generic way in which 1) we can get a feel for how tuning parameters affect prediction accuracy in a model and 2) identify the best combination of tuning parameters?

Thanks

Jason Brownlee September 17, 2016 at 9:35 am #

REPLY

Yep, nice approach.

Generally I like to try broad brush numbers of epochs (10, 100, 1000 on a log scale) and zoom in from there.

Searching blocks of pre-defined parameters is called grid searching, a topic I write a bit al

Carmen September 17, 2016 at 3:27 am #

Sorry, me again 🙂

On the topic of alpha, the learning parameter. Is there a benefit to setting this to a really small of epochs must be that much greater in order to obtain a very small RMSE. So the tradeoff se benefit to being a "slow learner"? Intuitively, it feels like it should reduce variance but I'm not s training dataset affect how this value should be set?

Thanks

Jason Brownlee September 17, 2016 at 9:36 am #

REPLY

Great question. The number of epochs and learning rate are linked.

Small alpha needs more training, and the reverse.

You will see this pattern in algorithm after algorithm, the amount of update and how long t

zerious95 October 2, 2016 at 7:35 am #

REPLY

The Explanation is useful
Thanks jason

Jason Brownlee October 2, 2016 at 8:21 am #

REPLY

I'm glad you liked it zerious95, thanks.

Dapo October 7, 2016 at 11:32 pm #

hi jason could you give an example of how to use this method on some data set? jus

Jason Brownlee October 8, 2016 at 10:38 am #

Sure Dapo, see this tutorial:
http://machinelearningmastery.com/linear-regression-tutorial-using-gradient-descent-for-m

Dima December 26, 2016 at 4:31 am #

REPLY

Thank you for your article it is very useful for new ones but I still have a question. What should I do if my coordinate is on the top of the wavy function, it would be easier to show in the pic but I cannot paste it, so imagine a sin function and its waves . I mean derivative at this point equals zero and this point is the maximum so gradient descent probably won`t work here.

Andy December 27, 2016 at 3:18 pm #

Jason,
Really great and simple explanation. Been reading up on Gradient Descent, and this by far made the most sense. I've read that partial derivatives is used in Gradient Descent. It's been a while since I took calculus, and I feel like a need a refresh. Do you think taking a Calculus 1 course would be the best bet? Or should I just read up on partial derivatives or derivatives? Any suggestions for resources, or what type of Calculus course to focus on would be appreciated. Thanks

Jason Brownlee December 28, 2016 at 7:04 am # REPLY

Hi Andy,

If you plan on doing research on new gradient descent methods or implementing gradient descent yourself for operational use, then getting familiar with the "why" of the algorithm is a good idea.

If you want to use it and deliver results, I would suggest it might not be the best use of you

Nuwan Chathuranga May 24, 2017 at 12:17 pm #

Hi Json,
Nice explanation about gradient decent and it is great help for pple like us,novis to the machir

My question is how we find derivative in a multi dimension situation.

Eg:

f(x1,x2) = x1^2 + x2^3+7

In kind of this situation what woud be the derivative subject?

Is it df(x1,x2)/x1 or df(x1,x2)/dx2. Or would it be a partial differential approach.

Thank you

Nidhi May 22, 2019 at 1:18 am # REPLY

Yes, when ever the function has more than one independent variables such as x_1, x_2 etc, then we use the concept of partial derivatives.
For updating the coefficients, you will find partial derivative of f w.r.t x_1 keeping x_2 as constant and next par der w.r.t x_2 keeping x_1 as constant and so on

---

Sami July 23, 2017 at 11:29 pm # REPLY

Thanks a Lot Sir! Your effort is really appreciable!

Jason Brownlee July 24, 2017 at 6:54 am #

Thanks, I'm glad you found the post useful.

Zeeshan Ul Islam July 24, 2017 at 11:54 pm #

Dear Mr. Jason sir,

Can I have a list of algorithms which perform gradient descent or gradient optimization, some
gradient?

Also, sir, I am doing pattern recognition using an autoencoder model on Brani maps, and I am
Can you suggest me some better algorithms for this purpose??

Thanks a lot sir

Jason Brownlee July 25, 2017 at 9:46 am # REPLY

Sorry, I do not have a list.

Take a look at any modern neural net library for a good list, for example, here is the list in Keras:
https://keras.io/optimizers/

Adam is excellent:
http://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

Chandrajit Pal August 1, 2017 at 4:57 am # REPLY

Just excellent explanation…brilliant …..explained in such a simple way that a layman can easily understand…

Jason Brownlee August 1, 2017 at 8:13 am #

Thanks.

Abubakar September 26, 2017 at 5:38 am #

Thanks very much, a very good and comprehensive explanation

Jason Brownlee September 26, 2017 at 7:27 am #

Thanks.

Andrea de Luca October 18, 2017 at 6:45 am # REPLY

Jason,your teaching efforts are to be considered commendable.

However, students should not be encouraged to avoid the math behind ML. Rather, they should be advised to embrace its study with enthusiasm. Any non-mathematicized study of such matters will produce a crippled and wanting knowledge of the subject.

Jason Brownlee October 18, 2017 at 3:49 pm # REPLY

Thanks for your thoughts Andrea.

Aniket Saxena October 29, 2017 at 2:53 am # REPLY

Hello Jason,

Firstly, I want to say that this post is awesome for everyone looking for information about optir

Secondly, I have a query regarding minimizing the cost function of linear regression using gra

In the documentation of scikit learn at this link scikit-learn.org/stable/modules/linear_model.htı
I saw a statement mentioning that,

LinearRegression fits a linear model with coefficients w = (w_1, …, w_p) to minimize the resid in the dataset, and the responses predicted by the linear approximation. Mathematically it solv \underset{w}{min\,} {|| X w – y||_2}^2

So, my question is that if I use LinearRegression class of sklearn.Linear Model package for m squares), how does this class minimizes it and obtain the model coefficients(i.e., theta0, theta that minimizes this cost function? What do you think which optimization algorithm does Linear and obtain model coefficients?

Can I use SGDRegressor class of sklearn library to minimize the cost function and obtain the model coefficients(i.e., theta0, theta1) that minimizes this cost function using optimization algorithm like stochastic gradient descent?

Please help…….

Jason Brownlee October 29, 2017 at 5:58 am #

sklearn and other libraries do not use gradient descent, instead linear algebra methods are often used. You can do this when all data can easily fit into memory.

We use linear regression here to demo gradient descent because it is an easy algorithm to understand. We might not use this algorithm to fit a linear regression if all data can fit into memory and we can use linalg methods instead.

Surendra November 9, 2017 at 1:19 am # REPLY

Hi Jason

Excellent article I can say at least what I have gone through about Gradient descent.

Thank you so much for a nice article.

I hope "SGDRegressor" from "sklearn" using this cost function to calculate the best coefficien
using normal equation to solve the same.

However, here I doubt how will we start with initial set of coefficients in SGD so that we will st

Jason Brownlee November 9, 2017 at 10:01 am #

SGD does not guarantee the best coefficients, only good enough coefficients.

You can try and run the process many times and take the best across all runs.

Surendra November 9, 2017 at 12:32 pm # REPLY

Thank you Jason for your answer, could you please let me know what is the best way to start with some random coefficients?

Jason Brownlee November 10, 2017 at 10:29 am # REPLY

Small random numbers in the range 0-1

israt November 12, 2017 at 5:39 pm #

REPLY

Why we change delta w on gradient decent?
What is convergence?
Why we done gradient decent in convergence?
When gradient decent not working as convergence?why we need appropriate learning rate?

Shaan Kevin December 9, 2017 at 9:47 pm #

can i create an ai using python and implement it into xcode and android studio and u
Is there a way to create an os that is an ai? if so how to do it

Jason Brownlee December 10, 2017 at 5:20 am #

Sorry, I don't know about xcode or android studio, I cannot give you good advice

Aakash January 2, 2018 at 11:53 pm #

Hi Jason,

Thank you for your explanation of gradient descent. I have a question regarding stochastic gradient descent.

For this explanation:

"The update procedure for the coefficients is the same as that above, except the cost is not summed over all training patterns, but instead calculated for one training pattern."

Regarding this, I am just confused that how the cost is calculated for one training pattern? Isn

Thank You

Jason Brownlee January 3, 2018 at 5:38 am # REPLY

The difference is that the gradient is estimated from one sample rather than a batch of samples and therefore is more noisy.

codeflight January 10, 2018 at 11:59 pm # REPLY

can you refer any article to learn implementation of this algorithm in python?

Jason Brownlee January 11, 2018 at 5:50 am #

Yes, see here:
https://machinelearningmastery.com/implement-linear-regression-stochastic-gradient-des

Arnish February 5, 2018 at 4:18 pm #

Hey Jason,
Great explanation, One question though. I am particularly new to calculus but how does the s
I mean if the slope is negative we decrease the values of coefficients and vice versa?
Or it depends on each curve/function.

Jason Brownlee February 6, 2018 at 9:10 am # REPLY

Yes, it is as simple as that. Moving left or right on the number line.

foxadilly February 16, 2018 at 7:39 am #

REPLY

now how does all these calculus of gradients apply in natural language processing text classification text summarisation ?

Jason Brownlee February 16, 2018 at 8:37 am #

REPLY

Gradient descent is the same regardless of the general problem being solved by the network.

Hector Alavro Rojas April 8, 2018 at 1:22 am #

Thanks a lot for all your explanations, Jason. It has been very helpful for me.

Any chance to get examples of how to apply Gradient Descent and Stochastic Gradient Desc
(Caret) and Python (Sklearn) by using cross-validation and tuning the parameters?

Jason Brownlee April 8, 2018 at 6:23 am #

Sure, start here:
https://machinelearningmastery.com/start-here/#deeplearning

Badri April 17, 2018 at 2:56 pm #

REPLY

Superb. Very useful article and it makes it clear and simple to understand about gradient descent for machine learning.

Jason Brownlee April 17, 2018 at 2:57 pm #

REPLY

Thanks.

MUNI KUMAR N August 7, 2018 at 7:02 pm #

REPLY

Superb Explanation. I am a big fan of your Machine Learning pages.

Jason Brownlee August 8, 2018 at 6:17 am #

REPLY

Thanks.

Ramya September 1, 2018 at 11:43 am #

really good explanation of the topic and could understand thoroughly. Look forward t
One comment on the making it more interesting – try adding examples and visuals.

Jason Brownlee September 2, 2018 at 5:27 am #

Thanks for the suggestion.

mothy September 8, 2018 at 7:11 pm #

REPLY

It was very helpful .Thanks jason

Jason Brownlee September 9, 2018 at 6:00 am #

I'm happy to hear that!

AKRITO MANDAL September 26, 2018 at 10:31 pm #

REPLY

Informative page, thanks Sir.

Jason Brownlee September 27, 2018 at 6:00 am #

REPLY

Thanks.

Chandrakant Patil November 15, 2018 at 10:37 pm #

Neat & clean clarification. Thanks for sharing.

Jason Brownlee November 16, 2018 at 6:15 am #

Thanks.

Jolaoso Lateef Olakunle May 22, 2019 at 7:56 pm #

REPLY

Hi Jason, you have done a great work. Thanks very much. Hope you dont mind if I contact you for help on my future work on machine learning. I am a Mathematician working on algorithms for optimization problems. Regards

Jason Brownlee May 23, 2019 at 5:59 am #

Your Start in Machine Learning

I'm here to help, if I can.

Leave a Reply

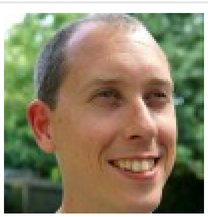
Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT

Welcome to Machine Learning Mastery!



Hi, I'm **Jason Brownlee**, PhD.
I write tutorials to help developers (*like you*) get results with machine learning.

Read More

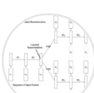
Your Start in Machine Learning

POPULAR


How to Develop LSTM Models for Multi-Step Time Series Forecasting of Household Power Co
OCTOBER 10, 2018


How to Develop LSTM Models for Time Series Forecasting
NOVEMBER 14, 2018


11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)
AUGUST 6, 2018


A Gentle Introduction to LSTM Autoencoders
NOVEMBER 5, 2018

How to Develop RNN Models for Human Activity Recognition Time Series Classification
SEPTEMBER 24, 2018

When to Use MLP, CNN, and RNN Neural Networks
JULY 23, 2018

How to Develop Convolutional Neural Network Models for Time Series Forecasting
NOVEMBER 12, 2018

Top beginner tutorials:

- How to Install Python for Machine Learning
- Your First Machine Learning Project in Python
- Your First Neural Network in Python
- Your First Classifier in Weka
- Your First Time Series Forecasting Project

Top project tutorials:

- Photo Captioning Project
- Neural Machine Translation Project
- Sentiment Analysis Project
- Power Forecasting Project

Privacy | Disclaimer | Terms | Contact