

Want help with linear algebra? [Take the FREE Mini-Course](#)

Search...



Introduction to Matrices and Matrix Arithmetic for Machine Learning

by Jason Brownlee on February 7, 2018 in [Linear Algebra](#)

Tweet

[Share](#)

[Share](#)

[G+](#)

Matrices are a foundational element of linear algebra.

Matrices are used throughout the field of machine learning in the description of algorithms and processes such as the input data variable (X) when training an algorithm.

In this tutorial, you will discover matrices in linear algebra and how to manipulate them in Python.

After completing this tutorial, you will know:

- What a matrix is and how to define one in Python with NumPy.
- How to perform element-wise operations such as addition, subtraction, and the Hadamard product.
- How to multiply matrices together and the intuition behind the operation.

Let's get started.

Your Start in Machine Learning

You can master applied Machine Learning without the math or fancy degree.

Find out how in this *free* and *practical* email course.

Email Address

START MY EMAIL COURSE



A Gentle Introduction to Matrices for Machine Learning

Photo by [Maximiliano Kolum](#), some rights reserved.

Tutorial Overview

This tutorial is divided into 6 parts; they are:

1. What is a Matrix?
2. Defining a Matrix
3. Matrix Arithmetic
4. Matrix-Matrix Multiplication (Dot Product)
5. Matrix-Vector Multiplication
6. Matrix-Scalar Multiplication

Need help with Linear Algebra

Take my free 7-day email crash course

Click to sign-up and also get a free PDF

[Download Your FREE](#)

Your Start in Machine Learning

You can master applied Machine Learning
without the math or fancy degree.

Find out how in this *free* and *practical* email course.

Email Address

[START MY EMAIL COURSE](#)

What is a Matrix?

A matrix is a two-dimensional array of scalars with one or more columns and one or more rows.

A matrix is a two-dimensional array (a table) of numbers.

— Page 115, No Bullshit Guide To Linear Algebra, 2017

The notation for a matrix is often an uppercase letter, such as A, and entries are referred to by their two-dimensional subscript of row (i) and column (j), such as a_{ij} . For example:

```
1 A = ((a11, a12), (a21, 22), (a31, a32))
```

It is more common to see matrices defined using a horizontal notation.

```
1     a11, a12
2 A = (a21, a22)
3     a31, a32
```

A likely first place you may encounter a matrix in machine learning is in model training data comprised of many rows and columns and often represented using the capital letter “X”.

The geometric analogy used to help understand vectors and some of their operations does not hold with matrices. Further, a vector itself may be considered a matrix with one column and multiple rows.

Often the dimensions of the matrix are denoted as m and n for the number of rows and the number of columns.

Now that we know what a matrix is, let's look at defining one in Python.

Defining a Matrix

We can represent a matrix in Python using a two-dimensional NumPy array.

A NumPy array can be constructed given a list of lists. For example, below is a 3 row, 2 column matrix.

```
1 # create matrix
2 from numpy import array
3 A = array([[1, 2, 3], [4, 5, 6]])
4 print(A)
```

Running the example prints the created matrix showing the expected output.

```
1 [[1 2 3]
2 [4 5 6]]
```

Matrix Arithmetic

In this section will demonstrate simple matrix-matrix arithmetic, which involves performing an operation between two matrices of equal size to result in a new matrix with the same dimensions.

Matrix Addition

Two matrices with the same dimensions can be added together to produce a new matrix with the same dimensions.

Your Start in Machine Learning

You can master applied Machine Learning without the math or fancy degree. Find out how in this free and practical email course.

[START MY EMAIL COURSE](#)

```
1 C = A + B
```

The scalar elements in the resulting matrix are calculated as the addition of the elements in each of the matrices being added.

```
1      a11 + b11, a12 + b12
2 A + B = (a21 + b21, a22 + b22)
3      a31 + b31, a32 + b32
```

Or, put another way:

```
1 C[0,0] = A[0,0] + B[0,0]
2 C[1,0] = A[1,0] + B[1,0]
3 C[2,0] = A[2,0] + B[2,0]
4 C[0,1] = A[0,1] + B[0,1]
5 C[1,1] = A[1,1] + B[1,1]
6 C[2,1] = A[2,1] + B[2,1]
```

We can implement this in python using the plus operator directly on the two NumPy arrays.

```
1 # add matrices
2 from numpy import array
3 A = array([[1, 2, 3], [4, 5, 6]])
4 print(A)
5 B = array([[1, 2, 3], [4, 5, 6]])
6 print(B)
7 C = A + B
8 print(C)
```

The example first defines two 2×3 matrices and then adds them together.

Running the example first prints the two parent matrices and then the result of adding them together.

```
1 [[1 2 3]
2 [4 5 6]]
3
4 [[1 2 3]
5 [4 5 6]]
6
7 [[ 2  4  6]
8 [ 8 10 12]]
```

Matrix Subtraction

Similarly, one matrix can be subtracted from another matrix with the same dimensions.

```
1 C = A - B
```

The scalar elements in the resulting matrix are calculated as the subtraction of the elements in each of the matrices being subtracted.

```
1      a11 - b11, a12 - b12
2 A - B = (a21 - b21, a22 - b22)
3      a31 - b31, a32 - b32
```

Or, put another way:

```
1 C[0,0] = A[0,0] - B[0,0]
2 C[1,0] = A[1,0] - B[1,0]
3 C[2,0] = A[2,0] - B[2,0]
4 C[0,1] = A[0,1] - B[0,1]
5 C[1,1] = A[1,1] - B[1,1]
6 C[2,1] = A[2,1] - B[2,1]
```

Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**.

Find out how in this *free* and *practical* email course.

[START MY EMAIL COURSE](#)

We can implement this in python using the minus operator directly on the two NumPy arrays.

```
1 # subtract matrices
2 from numpy import array
3 A = array([[1, 2, 3], [4, 5, 6]])
4 print(A)
5 B = array([[0.5, 0.5, 0.5], [0.5, 0.5, 0.5]])
6 print(B)
7 C = A - B
8 print(C)
```

The example first defines two 2×3 matrices and then subtracts one from the other.

Running the example first prints the two parent matrices and then subtracts the first matrix from the second.

```
1 [[1 2 3]
2  [4 5 6]]
3
4 [[ 0.5  0.5  0.5]
5  [ 0.5  0.5  0.5]]
6
7 [[ 0.5  1.5  2.5]
8  [ 3.5  4.5  5.5]]
```

Matrix Multiplication (Hadamard Product)

Two matrices with the same size can be multiplied together, and this is often called element-wise matrix multiplication or the Hadamard product.

It is not the typical operation meant when referring to matrix multiplication, therefore a different operator is often used, such as a circle “o”.

```
1 C = A o B
```

As with element-wise subtraction and addition, element-wise multiplication involves the multiplication of elements from each parent matrix to calculate the values in the new matrix.

```
1           a11 * b11, a12 * b12
2 A o B = (a21 * b21, a22 * b22)
3           a31 * b31, a32 * b32
```

Or, put another way:

```
1 C[0,0] = A[0,0] * B[0,0]
2 C[1,0] = A[1,0] * B[1,0]
3 C[2,0] = A[2,0] * B[2,0]
4 C[0,1] = A[0,1] * B[0,1]
5 C[1,1] = A[1,1] * B[1,1]
6 C[2,1] = A[2,1] * B[2,1]
```

We can implement this in python using the star operator directly on the two NumPy arrays.

```
1 # element-wise multiply matrices
2 from numpy import array
3 A = array([[1, 2, 3], [4, 5, 6]])
4 print(A)
5 B = array([[1, 2, 3], [4, 5, 6]])
6 print(B)
7 C = A * B
8 print(C)
```

The example first defines two 2×3 matrices and then multiplies them together.

Your Start in Machine Learning

You can master applied Machine Learning without the math or fancy degree.

Find out how in this *free* and *practical* email course.

[START MY EMAIL COURSE](#)

Running the example first prints the two parent matrices and then the result of multiplying them together with a Hadamard Product.

```
1 [[1 2 3]
2 [4 5 6]]
3
4 [[1 2 3]
5 [4 5 6]]
6
7 [[ 1   4   9]
8 [16  25  36]]
```

Matrix Division

One matrix can be divided by another matrix with the same dimensions.

```
1 C = A / B
```

The scalar elements in the resulting matrix are calculated as the division of the elements in each of the matrices.

```
1      a11 / b11, a12 / b12
2 A / B = (a21 / b21, a22 / b22)
3      a31 / b31, a32 / b32
```

Or, put another way:

```
1 C[0,0] = A[0,0] / B[0,0]
2 C[1,0] = A[1,0] / B[1,0]
3 C[2,0] = A[2,0] / B[2,0]
4 C[0,1] = A[0,1] / B[0,1]
5 C[1,1] = A[1,1] / B[1,1]
6 C[2,1] = A[2,1] / B[2,1]
```

We can implement this in python using the division operator directly on the two NumPy arrays.

```
1 # divide matrices
2 from numpy import array
3 A = array([[1, 2, 3], [4, 5, 6]])
4 print(A)
5 B = array([[1, 2, 3], [4, 5, 6]])
6 print(B)
7 C = A / B
8 print(C)
```

The example first defines two 2×3 matrices and then divides the first from the second matrix.

Running the example first prints the two parent matrices and then

```
1 [[1 2 3]
2 [4 5 6]]
3
4 [[1 2 3]
5 [4 5 6]]
6
7 [[ 1.  1.  1.]
8 [ 1.  1.  1.]]
```

Your Start in Machine Learning

You can master applied Machine Learning without the math or fancy degree.

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Matrix-Matrix Multiplication (Dot Product)

Matrix multiplication, also called the matrix dot product is more complex than matrix addition. There is a rule as not all matrices can be multiplied together.

1 $C = A * B$

or

1 $C = AB$

The rule for matrix multiplication is as follows:

- The number of columns (n) in the first matrix (A) must equal the number of rows (m) in the second matrix (B).

For example, matrix A has the dimensions m rows and n columns and matrix B has the dimensions n and k. The n columns in A and n rows b are equal. The result is a new matrix with m rows and k columns.

1 $C(m, k) = A(m, n) * B(n, k)$

This rule applies for a chain of matrix multiplications where the number of columns in one matrix in the chain must match the number of rows in the following matrix in the chain.

“ One of the most important operations involving matrices is multiplication of two matrices. The matrix product of matrices A and B is a third matrix C. In order for this product to be defined, A must have the same number of columns as B has rows. If A is of shape $m \times n$ and B is of shape $n \times p$, then C is of shape $m \times p$.

— Page 34, Deep Learning, 2016.

The intuition for the matrix multiplication is that we are calculating the dot product between each row in matrix A with each column in matrix B. For example, we can step down rows of column A and multiply each with column 1 in B to give the scalar values in column 1 of C.

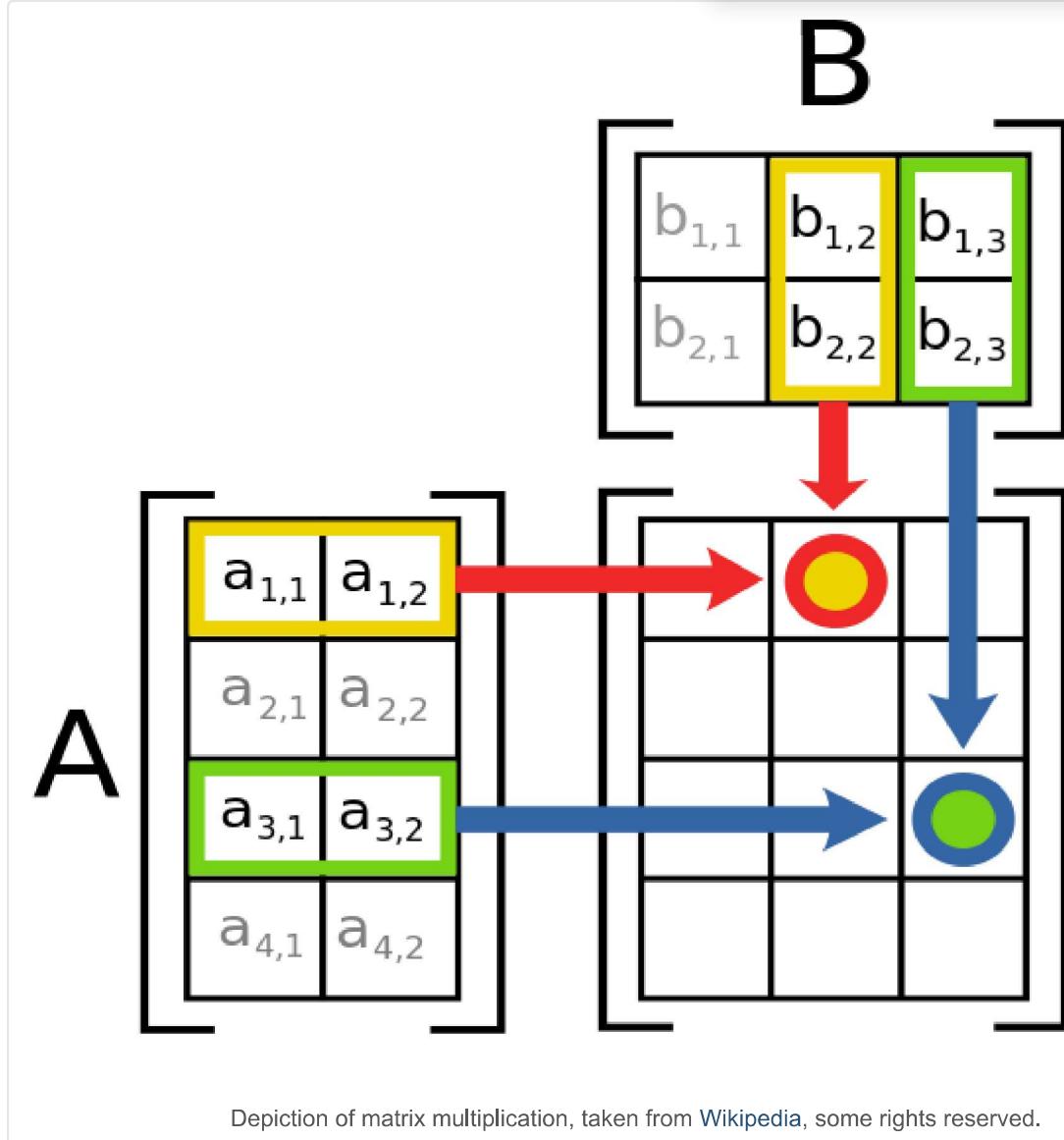
This is made clear with the following image.

Your Start in Machine Learning

You can master applied Machine Learning without the math or fancy degree. Find out how in this free and practical email course.

Email Address

START MY EMAIL COURSE



Depiction of matrix multiplication, taken from Wikipedia, some rights reserved.

Below describes the matrix multiplication using matrix notation.

```

1      a11, a12
2  A = (a21, a22)
3      a31, a32
4
5      b11, b12
6  B = (b21, b22)
7
8      a11 * b11 + a12 * b21, a11 * b12 + a12 * b22
9  C = (a21 * b11 + a22 * b21, a21 * b12 + a22 * b22)
10     a31 * b11 + a32 * b21, a31 * b12 + a32 * b22

```

This can be simplified by removing the multiplication signs as:

```

1      a11b11 + a12b21, a11b12 + a12b22
2  C = (a21b11 + a22b21, a21b12 + a22b22)
3      a31b11 + a32b21, a31b12 + a32b22

```

We can describe the matrix multiplication operation using array notation:

```

1  C[0,0] = A[0,0] * B[0,0] + A[0,1] * B[1,0]
2  C[1,0] = A[1,0] * B[0,0] + A[1,1] * B[1,0]
3  C[2,0] = A[2,0] * B[0,0] + A[2,1] * B[1,0]
4  C[0,1] = A[0,0] * B[0,1] + A[0,1] * B[1,1]
5  C[1,1] = A[1,0] * B[0,1] + A[1,1] * B[1,1]
6  C[2,1] = A[2,0] * B[0,1] + A[2,1] * B[1,1]

```

Your Start in Machine Learning

You can master applied Machine Learning without the math or fancy degree.

Find out how in this *free* and *practical* email course.

[START MY EMAIL COURSE](#)

The matrix multiplication operation can be implemented in NumPy using the dot() function.

```
1 # matrix dot product
2 from numpy import array
3 A = array([[1, 2], [3, 4], [5, 6]])
4 print(A)
5 B = array([[1, 2], [3, 4]])
6 print(B)
7 C = A.dot(B)
8 print(C)
```

The example first defines two 2×3 matrices and then calculates their dot product.

Running the example first prints the two parent matrices and then the result of the dot product.

```
1 [[1 2]
2  [3 4]
3  [5 6]]
4
5 [[1 2]
6  [3 4]]
7
8 [[ 7 10]
9  [15 22]
10 [23 34]]
```

Matrix-Vector Multiplication

A matrix and a vector can be multiplied together as long as the rule of matrix multiplication is observed.

Specifically, that the number of columns in the matrix must equal the number of items in the vector. As with matrix multiplication, the operation can be written using the dot notation. Because the vector only has one column, the result is always a vector.

```
1 c = A . v
```

Or without the dot in a compact form.

```
1 c = Av
```

The result is a vector with the same number of rows as the parent matrix.

```
1      a11, a12
2 A = (a21, a22)
3      a31, a32
4
5      v1
6 v = (v2)
7
8      a11 * v1 + a12 * v2
9 c = (a21 * v1 + a22 * v2)
10     a31 * v1 + a32 * v2
```

Or, more compactly.

```
1      a11v1 + a12v2
2 c = (a21v1 + a22v2)
3      a31v1 + a32v2
```

The matrix-vector multiplication can be implemented in NumPy us

```
1 # matrix-vector multiplication
```

Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**.

Find out how in this *free* and *practical* email course.

[START MY EMAIL COURSE](#)

```
2 from numpy import array
3 A = array([[1, 2], [3, 4], [5, 6]])
4 print(A)
5 B = array([0.5, 0.5])
6 print(B)
7 C = A.dot(B)
8 print(C)
```

The example first defines a 2×3 matrix and a 2 element vector and then multiplies them together.

Running the example first prints the parent matrix and vector and then the result of multiplying them together.

```
1 [[1 2]
2  [3 4]
3  [5 6]]
4
5 [ 0.5  0.5]
6
7 [ 1.5  3.5  5.5]
```

Matrix-Scalar Multiplication

A matrix can be multiplied by a scalar.

This can be represented using the dot notation between the matrix and the scalar.

```
1 C = A . b
```

Or without the dot notation.

```
1 C = Ab
```

The result is a matrix with the same size as the parent matrix where each element of the matrix is multiplied by the scalar value.

```
1      a11, a12
2 A = (a21, a22)
3      a31, a32
4
5 b
6
7      a11 * b, a12 * b
8 C = (a21 * b, a22 * b)
9      a31 * b, a32 * b
```

or

```
1      a11b, a12b
2 C = (a21b, a22b)
3      a31b, a32b
```

We can also represent this with array notation.

```
1 C[0,0] = A[0,0] * b
2 C[1,0] = A[1,0] * b
3 C[2,0] = A[2,0] * b
4 C[0,1] = A[0,1] * b
5 C[1,1] = A[1,1] * b
6 C[2,1] = A[2,1] * b
```

This can be implemented directly in NumPy with the multiplication

Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**.

Find out how in this *free* and *practical* email course.

[START MY EMAIL COURSE](#)

```
1 # matrix-scalar multiplication
2 from numpy import array
3 A = array([[1, 2], [3, 4], [5, 6]])
4 print(A)
5 b = 0.5
6 print(b)
7 C = A * b
8 print(C)
```

The example first defines a 2×3 matrix and a scalar and then multiplies them together.

Running the example first prints the parent matrix and scalar and then the result of multiplying them together.

```
1 [[1 2]
2  [3 4]
3  [5 6]]
4
5 0.5
6
7 [[ 0.5  1. ]
8  [ 1.5  2. ]
9  [ 2.5  3. ]]
```

Extensions

This section lists some ideas for extending the tutorial that you may wish to explore.

- Create 5 examples using each operation using your own data.
- Implement each matrix operation manually for matrices defined as lists of lists.
- Search machine learning papers and find 1 example of each operation being used.

If you explore any of these extensions, I'd love to know.

Further Reading

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

Books

- Section 2.3, Matrix operations. [No Bullshit Guide To Linear Algebra](#), 2017.
- Section 3.3, Matrix multiplication. [No Bullshit Guide To Linear Algebra](#), 2017.
- Section 1.3 Matrices, [Introduction to Linear Algebra](#), Fifth Edition, 2016
- Section 2.4 Rules for Matrix Operations, [Introduction to Linear Algebra](#)
- Section 2.1 Scalars, Vectors, Matrices and Tensors, [Deep Learning](#)
- Section 2.2 Multiplying Matrices and Vectors, [Deep Learning](#)

API

- [numpy.array\(\) API](#)
- [numpy.dot\(\) API](#)

Articles

- [Matrix \(mathematics\)](#)
- [Matrix multiplication on Wikipedia](#)

Your Start in Machine Learning

You can master applied Machine Learning without the math or fancy degree.

Find out how in this *free* and *practical* email course.

[START MY EMAIL COURSE](#)

- Hadamard product (matrices) on Wikipedia
- Dot product on Wikipedia

Summary

In this tutorial, you discovered matrices in linear algebra and how to manipulate them in Python.

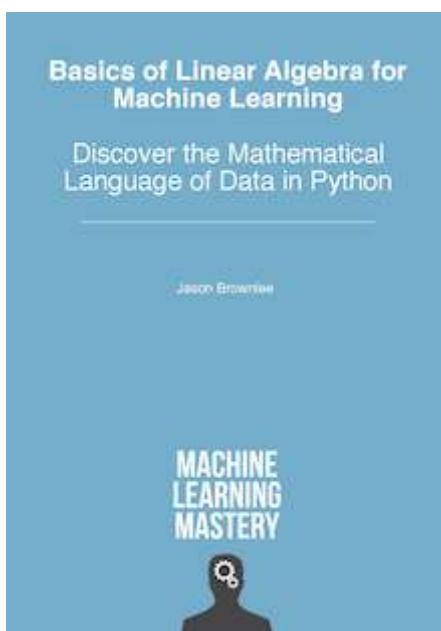
Specifically, you learned:

- What a matrix is and how to define one in Python with NumPy.
- How to perform element-wise operations such as addition, subtraction, and the Hadamard product.
- How to multiply matrices together and the intuition behind the operation.

Do you have any questions?

Ask your questions in the comments below and I will do my best to answer.

Get a Handle on Linear Algebra for Machine Learning!



Develop a working understand of linear algebra

...by writing lines of code in python

Discover how in my new Ebook:
[Linear Algebra for Machine Learning](#)

It provides **self-study tutorials** on topics like:
Vector Norms, Matrix Multiplication, Tensors, Eigendecomposition, SVD, PCA and much more...

Finally Understand the Mathematics of Data

Skip the Academics. Just Results.

[Click to learn more.](#)

Tweet

[Share](#)

[Share](#)

[G+](#)



About Jason Brownlee

Jason Brownlee, PhD is a machine learning specialist who teaches machine learning methods via hands-on tutorials.

[View all posts by Jason Brownlee →](#)

Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**. Find out how in this *free* and *practical* email course.

Email Address

[START MY EMAIL COURSE](#)

7 Responses to *Introduction to Matrices and Matrix Arithmetic for Machine Learning*



Erick February 7, 2018 at 8:16 pm #

REPLY ↗

If someone wants to *understand* what matrices actually do instead of applying them in Python, I truly recommend watching the YouTube channel “Linear Algebra” by 3Blue1Brown. Very insightful, requires very minimal knowledge of vectors and matrices.



Jason Brownlee February 8, 2018 at 8:24 am #

REPLY ↗

Thanks Erick.



Russell February 17, 2018 at 10:07 am #

REPLY ↗

Thanks Erick. Great resource!!



Russell Bigley February 16, 2018 at 12:26 pm #

REPLY ↗

I think there is an error in the code. The Matrix vector Multiplication

```
1      a11 * v1 + a12 + v2
2  c = (a21 * v1 + a22 + v2)
3      a31 * v1 + a32 + v2
```

should be

```
1      a11 * v1 + a12 * v2
2  c = (a21 * v1 + a22 * v2)
3      a31 * v1 + a32 * v2
```

Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**. Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE



meee June 7, 2018 at 9:16 am #

Hi sir can you please guide me. i want to learn absolute al-



Jason Brownlee June 8, 2018 at 6:03 am #

REPLY ↗

You can get started here:

https://machinelearningmastery.com/start-here/#linear_algebra

Leave a Reply

Name (required)

Email (will not be published) (required)

Website

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](#)

Linear Algebra for Machine Learning
Understand linear algebra by writing

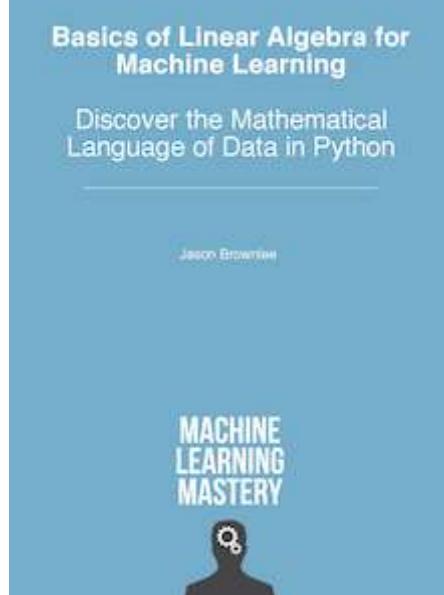
Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**. Find out how in this *free and practical* email course.

Email Address

[START MY EMAIL COURSE](#)

Click to Get Started Now!



POPULAR



So, You are Working on a Machine Learning Problem...

APRIL 4, 2018



How to Develop LSTM Models for Multi-Step Time Series Forecasting of Household Power Consumption

OCTOBER 10, 2018



11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)

AUGUST 6, 2018



How to Make Predictions with Keras

APRIL 9, 2018



How to Develop LSTM Models for Time Series Forecasting

NOVEMBER 14, 2018



You're Doing it Wrong. Why Machine Learning Does Not Have to Be So Hard

MARCH 28, 2018



How to Make Predictions with scikit-learn

APRIL 6, 2018

You might also like...

- 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

Your Start in Machine Learning

You can master applied Machine Learning without the math or fancy degree.

Find out how in this *free* and *practical* email course.

Email Address

START MY EMAIL COURSE

Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**. Find out how in this *free* and *practical* email course.

Email Address

[START MY EMAIL COURSE](#)