Why One-Hot Encode Data in Machine Learning?

by **Jason Brownlee** on July 28, 2017 in **Machine Learning Process**

Getting started in applied machine learning can be difficult, especially when working with real-world data.

Often, machine learning tutorials will recommend or require that you prepare your data in specific ways before fitting a machine learning model.

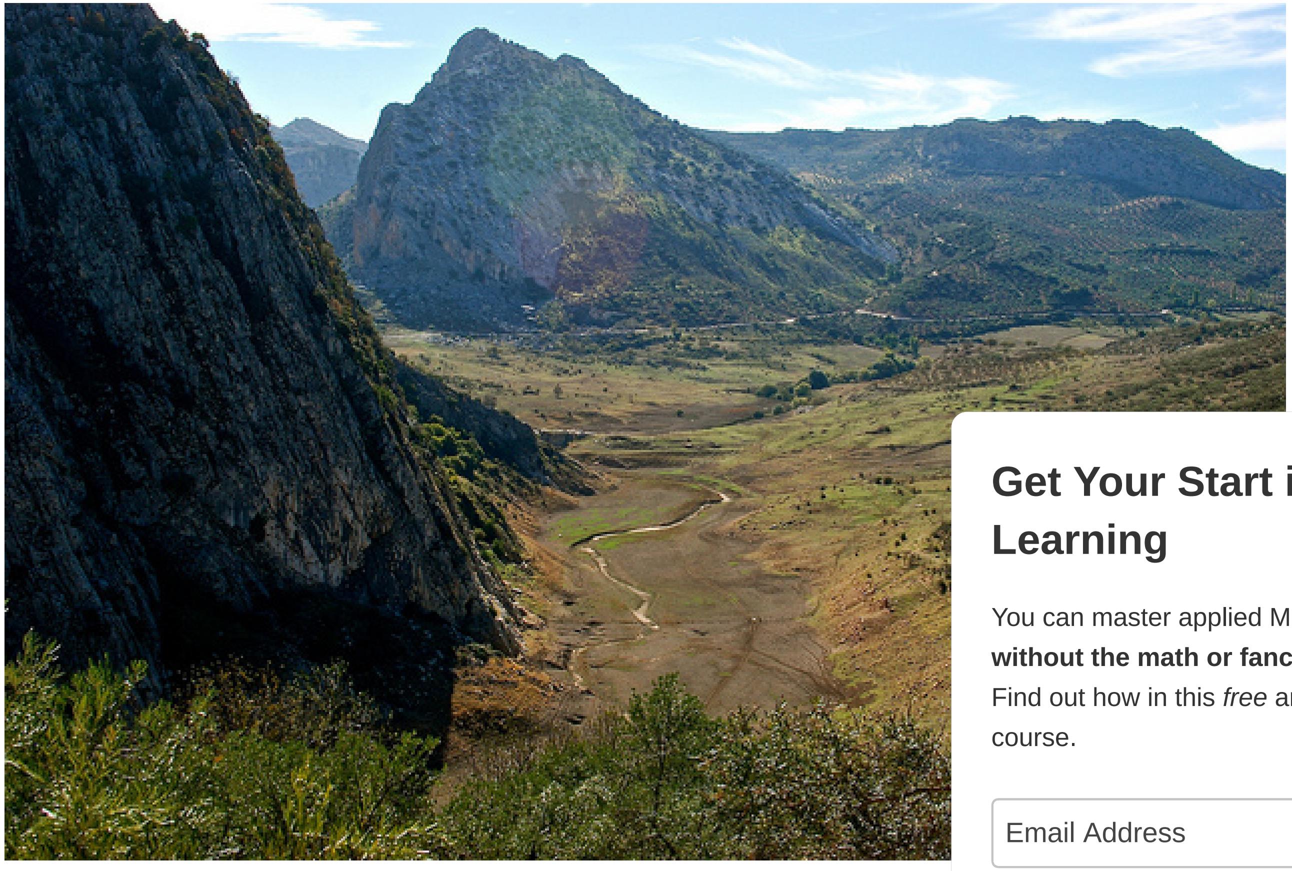
One good example is to use a one-hot encoding on categorical data.

- Why is a one-hot encoding required?
- Why can't you fit a model on your data directly?

In this post, you will discover the answer to these important questions and better understand data pr

Let’s get started.

[redacted]



Why One-Hot Encode Data in Machine Learning?
Photo by Karan Jain, some rights reserved.

What is Categorical Data?

Categorical data are variables that contain label values rather than numeric values.

The number of possible values is often limited to a fixed set.

Categorical variables are often called nominal.

Some examples include:

- A “*pet*” variable with the values: “*dog*” and “*cat*“.
- A “*color*” variable with the values: “*red*“, “*green*” and “*blue*“.
- A “*place*” variable with the values: “first”, “*second*” *and* “*third*“.

Each value represents a different category.

Some categories may have a natural relationship to each other, such as a natural ordering.

The “*place*” variable above does have a natural ordering of values. This type of categorical variable is called an ordinal variable.

What is the Problem with Categorical Data?

Some algorithms can work with categorical data directly.

For example, a decision tree can be learned directly from categorical data with no data transform rec n).

Many machine learning algorithms cannot operate on label data directly. They require all input variab

In general, this is mostly a constraint of the efficient implementation of machine learning algorithms r themselves.

This means that categorical data must be converted to a numerical form. If the categorical variable is predictions by the model back into a categorical form in order to present them or use them in some a

How to Convert Categorical Data to Numerical Data?

This involves two steps:

1. Integer Encoding
2. One-Hot Encoding

1. Integer Encoding

[redacted]

As a first step, each unique category value is assigned an integer value.

For example, “*red*” is 1, “*green*” is 2, and “*blue*” is 3.

This is called a label encoding or an integer encoding and is easily reversible.

For some variables, this may be enough.

The integer values have a natural ordered relationship between each other and machine learning algorithms may be able to understand and harness this relationship.

For example, ordinal variables like the “place” example above would be a good example where a label encoding would be sufficient.

2. One-Hot Encoding

For categorical variables where no such ordinal relationship exists, the integer encoding is not enoug

In fact, using this encoding and allowing the model to assume a natural ordering between categories results (predictions halfway between categories).

In this case, a one-hot encoding can be applied to the integer representation. This is where the integ variable is added for each unique integer value.

In the “*color*” variable example, there are 3 categories and therefore 3 binary variables are needed. color and “0” values for the other colors.

For example:

```
red,    green,  blue
1,      0,  0
0,      1,  0
0,      0,  1
```

The binary variables are often called “dummy variables” in other fields, such as statistics.

Further Reading

- Categorical variable on Wikipedia
- Nominal category on Wikipedia
- Dummy variable on Wikipedia

Summary

In this post, you discovered why categorical data often must be encoded when working with machine learning algorithms.

Specifically:

- That categorical data is defined as variables with a finite set of label values.
- That most machine learning algorithms require numerical input and output variables.
- That an integer and one hot encoding is used to convert categorical data to integer data.

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

About Jason Brownlee

Dr. Jason Brownlee is a husband, proud father, academic researcher, author, professional devel
to helping developers get started and get good at applied machine learning. Learn more.
View all posts by Jason Brownlee →

‹ What is the Difference Between a Parameter and a Hyperparameter?

How to Get Good Results Fast with Deep Learning for Time Series Forecasting ›

41 Responses to *Why One-Hot Encode Data in Machine Learning?*

Varun July 28, 2017 at 6:27 am # REPLY

You didn't mention that if we have a categorical variable with 3 categories, we only need to define 2 one-hot variables to save us from linear dependency.

Navdeep July 28, 2017 at 6:49 am # REPLY

HHi jason.I truly following you alot and really appreciate your effort and ease of tutorials.just a c
multilabel class and in coming tutorials could you help in featureselection of text data for muticlass and r
for 90 datapoints. And used keras for mlp,cnn and rnn where each datapoint is long paragraph with labe
you have any suggestions

Jason Brownlee July 28, 2017 at 8:39 am #

The one hot vector would have a length that would equal the number of labels, but multiple

Thanks for the suggestion.

This post suggests ways to lift deep learning model skill:
http://machinelearningmastery.com/improve-deep-learning-performance/

Espoir July 28, 2017 at 7:18 am # REPLY

What are the cons of one hot encoding ??? Supposed that you have some categorical features with each one with 500 or more differents values !!
So when you do one hot encoding you will have many colums in the dataset does it still good for a mach

[redacted]

Jason Brownlee July 28, 2017 at 8:40 am #

REPLY

Great question!

The vectors can get very large, e.g. the length of all words in your vocab in an NLP problem.

Large vectors make the method slow (increased computational complexity).

In these cases, a dense representation could be used, e.g. word embeddings in NLP.

faadal July 30, 2017 at 1:11 am #

Hi Jason, thanks again for your amazing pedagogy.

Back to the Espoirt question, I face this problem with 84 user_ID. I do a OHE of them and, like y
look like I fall in a infinite loop. So taking in to account the fact that I am not in the NLP case, hov

Thanks.

Jason Brownlee July 30, 2017 at 7:47 am #

What do you mean you fall into an infinite loop?

Vitor July 31, 2017 at 3:11 am #

REPLY

Very helpful post, Jason!
Espoir raised my question here but I did not undestand how to apply your answer to my case. I have 11+ thousand different products id. The database has about 130 thousand entries. This easily leads to MemoryError when using OHE. What approach/solution should I look for?

Jason Brownlee July 31, 2017 at 8:18 am # REPLY

Ouch.

Maybe you can use efficient sparse vector representations to cut down on memory?

Maybe try exploring dense vector methods that are used in NLP. Maybe you can something like a word embedding and let the model (e.g. a neural net) learn the relationship between different input labels, if any.

Sasikanth July 28, 2017 at 11:54 am # REPLY

Hello Jason how do we retrieve the features back after OHE if we need to present it visually?

Jason Brownlee July 29, 2017 at 8:01 am #

You can reverse the encoding with an argmax() (e.g. numpy.argmax())

gezmi July 28, 2017 at 5:20 pm #

Thank you for these wonderful posts!
Does data have to be one-hot encoded for classification trees and random forests as well or they can ha
results?

Jason Brownlee July 29, 2017 at 8:07 am # REPLY

No, trees can deal with categories as-is.

[redacted]

Ravindra July 28, 2017 at 5:31 pm # REPLY

Hi Jason, this post is very helpful, thank you!!
Question- In general what happens to model performance, when we apply One Hot Encoding to a ordinal feature? Would you suggest only to use integer encoding in case of ordinal features?

Jason Brownlee July 29, 2017 at 8:07 am # REPLY

It really depends on the problem and the meaning of the feature being encoded.

If in doubt, test.

Ravindra July 29, 2017 at 4:16 pm #

I see, thanks!

Rajkumar Kaliyaperumal July 28, 2017 at 6:47 pm #

hey Jason,
As usual this is another useful post on feature representation of categorical variables. Since logistic regr
form w1X1 + w2X2 +.. where X are features such as categorical variables- Places,color etc, and w are weights, intuitively X can take only numerical values for the line to fit. Is this a right intuition?

Jason Brownlee July 29, 2017 at 8:11 am # REPLY

Yes, regression algorithms like logistic regression require numeric input variables.

[redacted]

Raj July 31, 2017 at 2:16 pm # REPLY

Thanks a lot for your clarifying. I love your blogs and daily email digests. They help me to understand key concepts & practical tips easily.

Jason Brownlee July 31, 2017 at 3:50 pm # REPLY

Thanks Raj.

PabloRQ July 28, 2017 at 7:15 pm #

nice!

Jason Brownlee July 29, 2017 at 8:11 am #

Thanks.

ritika July 29, 2017 at 8:19 pm #

very well explained..thanks

Jason Brownlee July 30, 2017 at 7:46 am # REPLY

Thanks, I'm glad it helped.

Jie July 31, 2017 at 11:41 am #

REPLY

I love your blog!
One question: if we use tree based methods like decision tree, etc. Do we still need one-hot encoding?
Thanks you very much!

Jason Brownlee July 31, 2017 at 3:49 pm #

REPLY

No Jie. Most decision trees can work with categorical inputs directly.

Jie August 1, 2017 at 1:01 am #

Thank you very much!

Jason Brownlee August 1, 2017 at 8:00 am #

No probs.

Andrew Jabbitt August 3, 2017 at 12:27 am #

REPLY

Hi Jason, loving the blog … a lot!

I'm using your binary classification tutorial as a template (thanks!) for a retail sales data predictor. I'm basically trying to predict future hourly sales using product features and hourly weather forecasts, trained on historical sales and using above/below annual average sales as my binary labels.

I have encoded my categorical data and I get good accuracy when training my data (87%+), but this falls [redacted] and much smaller data set.

As far as I can see my problem is caused by encoding the categorical data – the same categories in my unseen set have different codes than in my model. Could this be the cause of my poor prediction performance: the encoded prediction categories are not aligned to those used to train and test the model? If so how do you overcome these challenges in practice?

Hope it makes sense.

Jason Brownlee August 3, 2017 at 6:53 am #

REPLY

Nice work Andrew!

Your model might be overfitting, try a smaller model, try regularization, try a large dataset, try less tra

Here are more ideas:
http://machinelearningmastery.com/improve-deep-learning-performance/

I hope that helps as a start.

Andrew Jabbitt August 3, 2017 at 4:25 pm #

Hey Jason, didn't think I had 'that' problem, but I probably do 🙂

Many thanks.

Maurice BigMo Flynn August 9, 2017 at 3:05 pm #

REPLY

Appreciable and very helpful post, thank you!!!

Question: What is the best way to one hot encode an array of categorical variables?

I have also startup with a AI post you can also find some knowledge over there: Thebigmoapproach.com

Jason Brownlee August 10, 2017 at 6:50 am # REPLY

There are many ways and “best” is defined by the tools and problem.

Here are a few ways:
http://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/

tom August 28, 2017 at 4:33 pm # REPLY

hi Jason:

One question, take the “color” variable as an example,if the color is ‘red’ , then after one-hot encoding ,it
three features from one feature?
It has been added two columns，is that right ?

Jason Brownlee August 29, 2017 at 5:01 pm #

Correct Tom!

Zhida Li September 4, 2017 at 8:32 pm # REPLY

Hi Jason, if my input data is [1 red 3 4 5], if use one hot encoder, red become [1,0,0],]does it mean that the whole features of the input data is extended?
input data now is [1 1 0 0 3 4 5]

Jason Brownlee September 7, 2017 at 12:35 pm # REPLY

Sorry, I don't follow. Perhaps you can restate your question?

Peter Ken Bediako October 13, 2017 at 6:28 pm # REPLY

Hello DR. Brownlee,

I am training a model to detect attacks and i need someone like you to help me detect the mistakes in my code because my training is not producing any better results. Kindly alert me if you will be interested to help me.
Thank you

Jason Brownlee October 14, 2017 at 5:41 am #

Sorry, I do not have the capacity to review your code.

Peter Ken Bediako October 13, 2017 at 8:20 pm #

I am using Tensorflow developing the mode,and would want to know how your book can help m
you

Jason Brownlee October 14, 2017 at 5:44 am # REPLY

My deep learning book shows how to bring deep learning to your projects using the Keras library. It does not cover tensorflow.

Keras is a library that runs on top of tensorflow and is much easier to use.

[redacted]

Leave a Reply

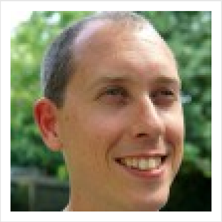
Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT

Welcome to Machine Learning Mastery



Hi, I'm Dr. Jason Brownlee.
My goal is to make practitioners like YOU awesome at applied machine learning.

Read More

You're a Professional (and you need results)!

[redacted]

POPULAR

Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras
JULY 21, 2016

Your First Machine Learning Project in Python Step-By-Step
JUNE 10, 2016

Develop Your First Neural Network in Python With Keras Step-By-Step
MAY 24, 2016

Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras
JULY 26, 2016

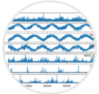
How to Setup a Python Environment for Machine Learning and Deep Learning with Anaconda
MARCH 13, 2017

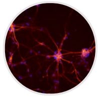
Time Series Forecasting with the Long Short-Term Memory Network in Python
APRIL 7, 2017

Multi-Class Classification Tutorial with the Keras Deep Learning Library
JUNE 2, 2016

Regression Tutorial with the Keras Deep Learning Library in Python

JUNE 9, 2016


Multivariate Time Series Forecasting with LSTMs in Keras
AUGUST 14, 2017


How to Implement the Backpropagation Algorithm From Scratch In Python
NOVEMBER 7, 2016

Privacy | Contact | About