

[Start Here](#)[Blog](#)[Books](#)[About](#)[Contact](#)

Need help with Deep Learning? [Take the FREE Mini-Course](#)

# Time Series Prediction With Deep Learning in Keras

by **Jason Brownlee** on July 19, 2016 in **Deep Learning**



Time Series prediction is a difficult problem both to frame and to address with machine learning.

In this post, you will discover how to develop neural network models for time series prediction in Python using the Keras deep learning library.

After reading this post you will know:

- About the airline passengers univariate time series prediction problem.
- How to phrase time series prediction as a regression problem and develop a neural network model for it.
- How to frame time series prediction with a time lag and develop a neural network model for it.

[Get Your Start in Machine Learning](#)

Let's get started.

- **Update Oct/2016:** Replaced graphs with more accurate versions, commented on the limited performance of the first method.
- **Update Mar/2017:** Updated example for Keras 2.0.2, TensorFlow 1.0.1 and Theano 0.9.0.

## Problem Description

The problem we are going to look at in this post is the international airline passengers prediction problem.

This is a problem where given a year and a month, the task is to predict the number of international airline passengers in units of 1,000. The data ranges from January 1949 to December 1960 or 12 years, with 144 observations.

The dataset is available for free from the [DataMarket webpage](#) as a [CSV download](#) with the filename

Below is a sample of the first few lines of the file.

```
1 "Month","International airline passengers: monthly totals in thousands. Jan 49 ? Dec 60"  
2 "1949-01",112  
3 "1949-02",118  
4 "1949-03",132  
5 "1949-04",129  
6 "1949-05",121
```

We can load this dataset easily using the Pandas library. We are not interested in the date, given that we only have data for one month. Therefore when we load the dataset we can exclude the first column.

The downloaded dataset also has footer information that we can exclude with the **skipfooter** argument. Once loaded we can easily plot the whole dataset. The code to load and plot the dataset is listed below.

```
1 import pandas  
2 import matplotlib.pyplot as plt  
3 dataset = pandas.read_csv('international-airline-passengers.csv', usecols=[1], engine='python', skipfooter=3)  
4 plt.plot(dataset)  
5 plt.show()
```

You can see an upward trend in the plot.

You can also see some periodicity to the dataset that probably corresponds to the northern hemisphere

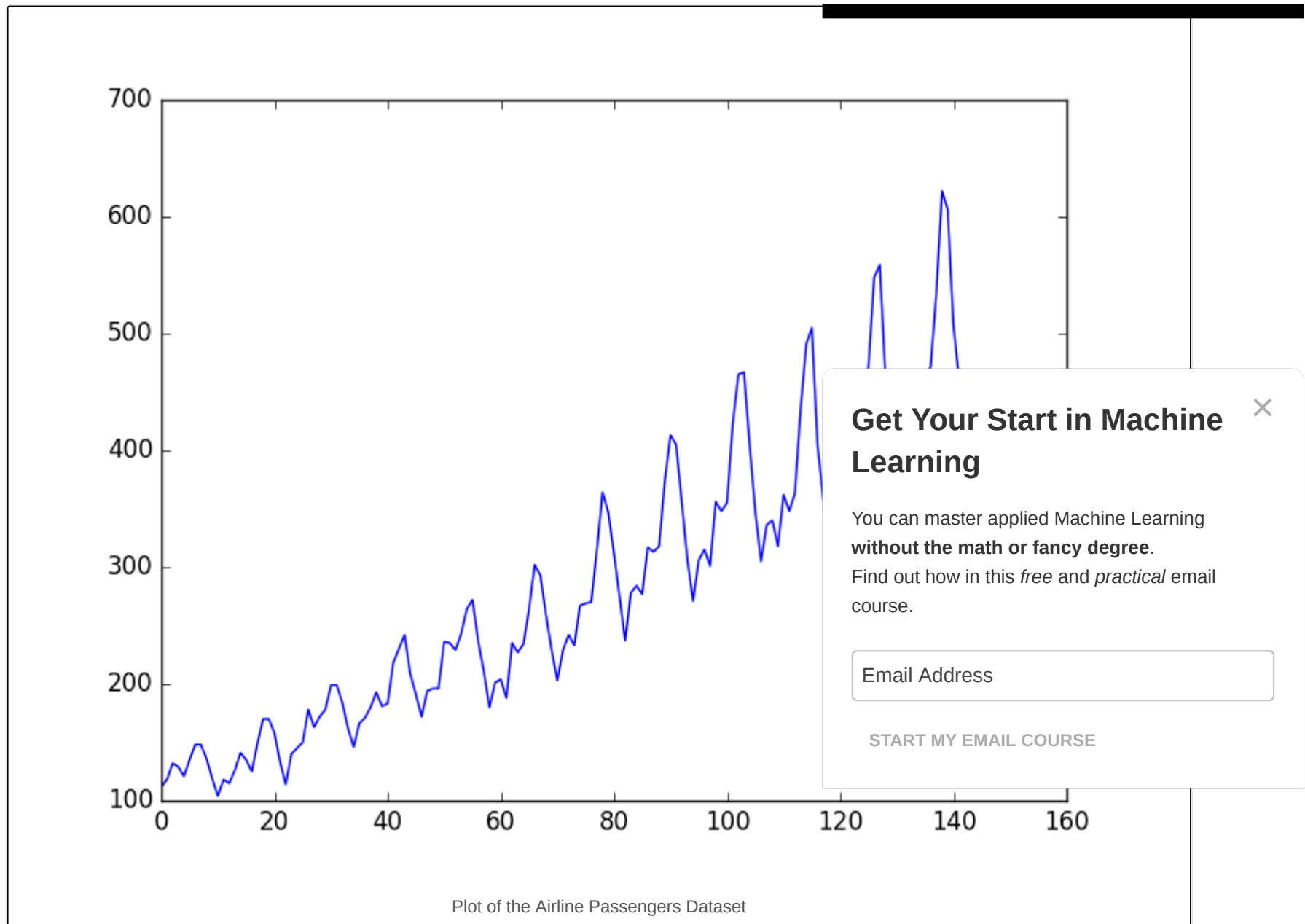
### Get Your Start in Machine Learning

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

Get Your Start in Machine Learning



We are going to keep things simple and work with the data as-is.

Normally, it is a good idea to investigate various data preparation techniques to rescale the data and

**Get Your Start in Machine Learning**

## Need help with Deep Learning in Python?

Take my free 2-week email course and discover MLPs, CNNs and LSTMs (with sample code).

Click to sign-up now and also get a free PDF Ebook version of the course.

Start Your FREE Mini-Course Now!

## Multilayer Perceptron Regression

We want to phrase the time series prediction problem as a regression problem.

That is, given the number of passengers (in units of thousands) this month, what is the number of passengers next month?

We can write a simple function to convert our single column of data into a two-column dataset. The first column containing the current month's passenger count and the second column containing next month's (t+1) passenger count, to be predicted.

Before we get started, let's first import all of the functions and classes we intend to use. This assumes the Keras deep learning library is installed.

```
1 import numpy
2 import matplotlib.pyplot as plt
3 import pandas
4 from keras.models import Sequential
5 from keras.layers import Dense
```

Before we do anything, it is a good idea to fix the random number seed to ensure our results are reproducible.

```
1 # fix random seed for reproducibility
2 numpy.random.seed(7)
```

### Get 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

Get Your Start in Machine Learning

We can also use the code from the previous section to load the dataset as a Pandas dataframe. We can then extract the Numpy array from the dataframe and convert the integer values to floating point values which are more suitable for modeling with a neural network.

```
1 # load the dataset
2 dataframe = pandas.read_csv('international-airline-passengers.csv', usecols=[1], engine='python', skipfooter=3)
3 dataset = dataframe.values
4 dataset = dataset.astype('float32')
```

After we model our data and estimate the skill of our model on the training dataset, we need to get an idea of the skill of the model on new unseen data. For a normal classification or regression problem we would do this using cross validation.

With time series data, the sequence of values is important. A simple method that we can use is to split the ordered dataset into train and test datasets. The code below calculates the index of the split point and separates the data into the training datasets with 67% of the observations that we can use to train our model, leaving the remaining 33% for testing the model.

```
1 # split into train and test sets
2 train_size = int(len(dataset) * 0.67)
3 test_size = len(dataset) - train_size
4 train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]
5 print(len(train), len(test))
```

Now we can define a function to create a new dataset as described above. The function takes two arguments: the dataset we want to convert into a dataset and the **look\_back** which is the number of previous time steps to use in this case, defaulted to 1.

This default will create a dataset where X is the number of passengers at a given time (t) and Y is the number of passengers at time (t+1).

It can be configured and we will look at constructing a differently shaped dataset in the next section.

```
1 # convert an array of values into a dataset matrix
2 def create_dataset(dataset, look_back=1):
3     dataX, dataY = [], []
4     for i in range(len(dataset)-look_back-1):
5         a = dataset[i:(i+look_back), 0]
6         dataX.append(a)
7         dataY.append(dataset[i + look_back, 0])
8     return numpy.array(dataX), numpy.array(dataY)
```

Let's take a look at the effect of this function on the first few rows of the dataset.

## Get 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

Get Your Start in Machine Learning

	X	Y
1	112	118
2	118	132
3	132	129
4	129	121
5	121	135

If you compare these first 5 rows to the original dataset sample listed in the previous section, you can see the  $X=t$  and  $Y=t+1$  pattern in the numbers.

Let's use this function to prepare the train and test datasets ready for modeling.

```
1 # reshape into X=t and Y=t+1
2 look_back = 1
3 trainX, trainY = create_dataset(train, look_back)
4 testX, testY = create_dataset(test, look_back)
```

We can now fit a Multilayer Perceptron model to the training data.

We use a simple network with 1 input, 1 hidden layer with 8 neurons and an output layer. The model square root gives us an error score in the units of the dataset.

I tried a few rough parameters and settled on the configuration below, but by no means is the network

```
1 # create and fit Multilayer Perceptron model
2 model = Sequential()
3 model.add(Dense(8, input_dim=look_back, activation='relu'))
4 model.add(Dense(1))
5 model.compile(loss='mean_squared_error', optimizer='adam')
6 model.fit(trainX, trainY, epochs=200, batch_size=2, verbose=2)
```

Once the model is fit, we can estimate the performance of the model on the train and test datasets. models.

```
1 # Estimate model performance
2 trainScore = model.evaluate(trainX, trainY, verbose=0)
3 print('Train Score: %.2f MSE (%.2f RMSE)' % (trainScore, math.sqrt(trainScore)))
4 testScore = model.evaluate(testX, testY, verbose=0)
5 print('Test Score: %.2f MSE (%.2f RMSE)' % (testScore, math.sqrt(testScore)))
```

Finally, we can generate predictions using the model for both the train and test dataset to get a visual indication of the skill of the model.

## Get 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

Get Your Start in Machine Learning

Because of how the dataset was prepared, we must shift the predictions so that they align on the x-axis with the original dataset. Once prepared, the data is plotted, showing the original dataset in blue, the predictions for the train dataset in green the predictions on the unseen test dataset in red.

```
1 # generate predictions for training
2 trainPredict = model.predict(trainX)
3 testPredict = model.predict(testX)
4
5 # shift train predictions for plotting
6 trainPredictPlot = numpy.empty_like(dataset)
7 trainPredictPlot[:, :] = numpy.nan
8 trainPredictPlot[look_back:len(trainPredict)+look_back, :] = trainPredict
9
10 # shift test predictions for plotting
11 testPredictPlot = numpy.empty_like(dataset)
12 testPredictPlot[:, :] = numpy.nan
13 testPredictPlot[len(trainPredict)+(look_back*2)+1:len(dataset)-1, :] = testPredict
14
15 # plot baseline and predictions
16 plt.plot(dataset)
17 plt.plot(trainPredictPlot)
18 plt.plot(testPredictPlot)
19 plt.show()
```

We can see that the model did a pretty poor job of fitting both the training and the test datasets. It ba

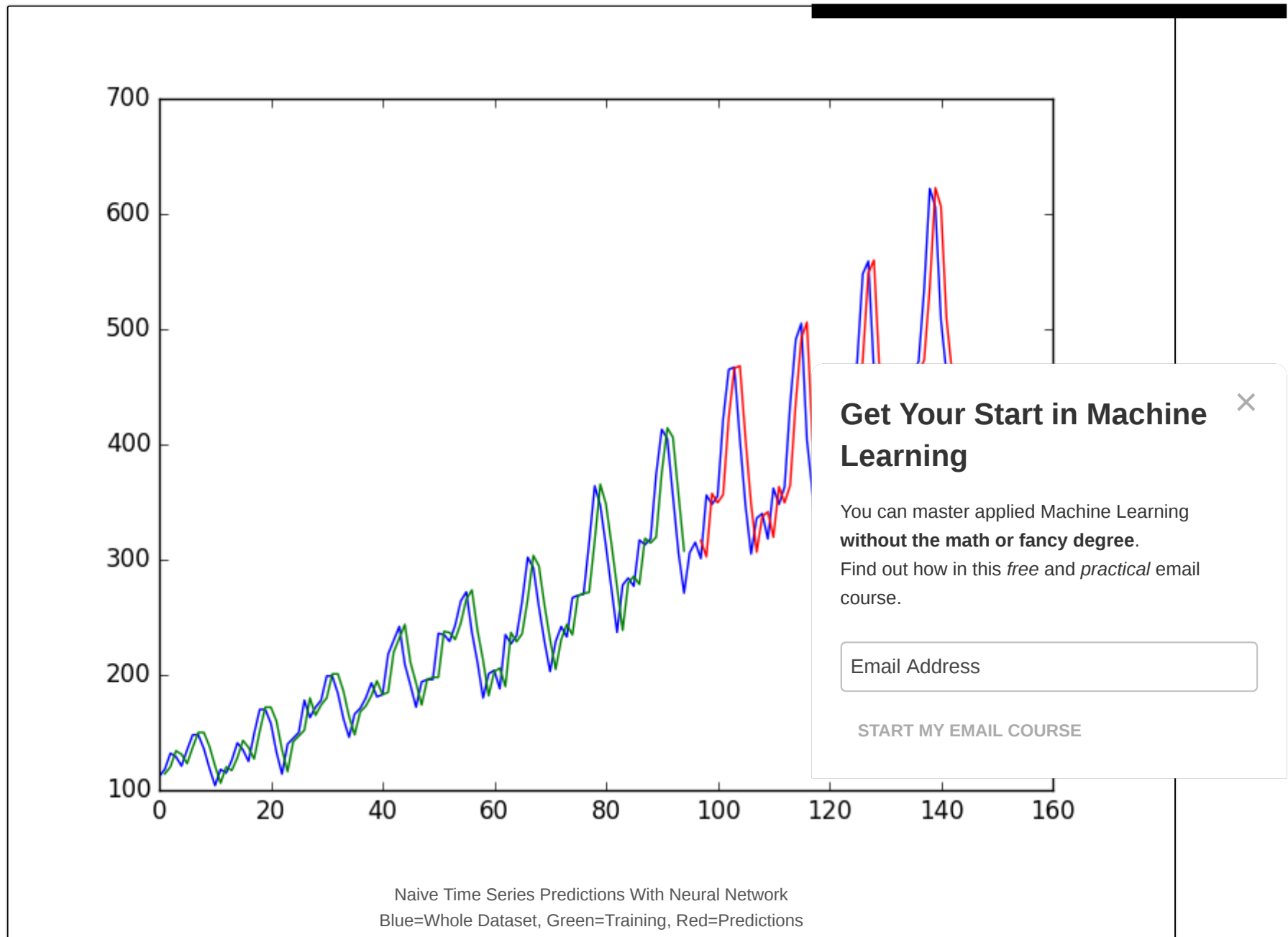
## Get Your Start in Machine Learning

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

Get Your Start in Machine Learning



For completeness, below is the entire code listing.



```

1 # Multilayer Perceptron to Predict International Airline Passengers (t+1, given t)
2 import numpy
3 import matplotlib.pyplot as plt
4 import pandas
5 import math
6 from keras.models import Sequential
7 from keras.layers import Dense
8 # fix random seed for reproducibility
9 numpy.random.seed(7)
10 # load the dataset
11 dataframe = pandas.read_csv('international-airline-passengers.csv', usecols=[1], engine='python', skipfooter=3)
12 dataset = dataframe.values
13 dataset = dataset.astype('float32')
14 # split into train and test sets
15 train_size = int(len(dataset) * 0.67)
16 test_size = len(dataset) - train_size
17 train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]
18 print(len(train), len(test))
19 # convert an array of values into a dataset matrix
20 def create_dataset(dataset, look_back=1):
21     dataX, dataY = [], []
22     for i in range(len(dataset)-look_back-1):
23         a = dataset[i:(i+look_back), 0]
24         dataX.append(a)
25         dataY.append(dataset[i + look_back, 0])
26     return numpy.array(dataX), numpy.array(dataY)
27 # reshape into X=t and Y=t+1
28 look_back = 1
29 trainX, trainY = create_dataset(train, look_back)
30 testX, testY = create_dataset(test, look_back)
31 # create and fit Multilayer Perceptron model
32 model = Sequential()
33 model.add(Dense(8, input_dim=look_back, activation='relu'))
34 model.add(Dense(1))
35 model.compile(loss='mean_squared_error', optimizer='adam')
36 model.fit(trainX, trainY, epochs=200, batch_size=2, verbose=2)
37 # Estimate model performance
38 trainScore = model.evaluate(trainX, trainY, verbose=0)
39 print('Train Score: %.2f MSE (%.2f RMSE)' % (trainScore, math.sqrt(trainScore)))
40 testScore = model.evaluate(testX, testY, verbose=0)
41 print('Test Score: %.2f MSE (%.2f RMSE)' % (testScore, math.sqrt(testScore)))
42 # generate predictions for training
43 trainPredict = model.predict(trainX)
44 testPredict = model.predict(testX)
45 # shift train predictions for plotting
46 trainPredictPlot = numpy.empty_like(dataset)
47 trainPredictPlot[:, :] = numpy.nan

```

## Get 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

Get Your Start in Machine Learning

```
48 trainPredictPlot[look_back:len(trainPredict)+look_back, :] = trainPredict
49 # shift test predictions for plotting
50 testPredictPlot = numpy.empty_like(dataset)
51 testPredictPlot[:, :] = numpy.nan
52 testPredictPlot[len(trainPredict)+(look_back*2)+1:len(dataset)-1, :] = testPredict
53 # plot baseline and predictions
54 plt.plot(dataset)
55 plt.plot(trainPredictPlot)
56 plt.plot(testPredictPlot)
57 plt.show()
```

Running the model produces the following output.

```
1 Epoch 195/200
2 0s - loss: 536.7014
3 Epoch 196/200
4 0s - loss: 555.4216
5 Epoch 197/200
6 0s - loss: 552.2841
7 Epoch 198/200
8 0s - loss: 541.2220
9 Epoch 199/200
10 0s - loss: 542.3288
11 Epoch 200/200
12 0s - loss: 534.2096
13 Train Score: 532.59 MSE (23.08 RMSE)
14 Test Score: 2358.07 MSE (48.56 RMSE)
```

Taking the square root of the performance estimates, we can see that the model has an average error of 23.08 on the training dataset and 48 passengers (in thousands) on the test dataset.

## Multilayer Perceptron Using the Window Method

We can also phrase the problem so that multiple recent time steps can be used to make the prediction for the next time step.

This is called the window method, and the size of the window is a parameter that can be tuned for each problem.

For example, given the current time ( $t$ ) we want to predict the value at the next time in the sequence ( $t + 1$ ), we can use the current time ( $t$ ) as well as the two prior times ( $t-1$  and  $t-2$ ).

When phrased as a regression problem the input variables are  $t-2$ ,  $t-1$ ,  $t$  and the output variable is  $t+1$ .

### Get 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

Get Your Start in Machine Learning

The `create_dataset()` function we wrote in the previous section allows us to create this formulation of the time series problem by increasing the `look_back` argument from 1 to 3.

A sample of the dataset with this formulation looks as follows:

```
1 X1 X2 X3 Y
2 112 118 132 129
3 118 132 129 121
4 132 129 121 135
5 129 121 135 148
6 121 135 148 148
```

We can re-run the example in the previous section with the larger window size. We will increase the network capacity to handle the additional information. The first hidden layer is increased to 14 neurons and a second hidden layer is added with 8 neurons. The number of epochs is also increased to 400.

The whole code listing with just the window size change is listed below for completeness.

```
1 # Multilayer Perceptron to Predict International Airline Passengers (t+1, given t, t-1,
2 import numpy
3 import matplotlib.pyplot as plt
4 from pandas import read_csv
5 import math
6 from keras.models import Sequential
7 from keras.layers import Dense
8
9 # convert an array of values into a dataset matrix
10 def create_dataset(dataset, look_back=1):
11     dataX, dataY = [], []
12     for i in range(len(dataset)-look_back-1):
13         a = dataset[i:(i+look_back), 0]
14         dataX.append(a)
15         dataY.append(dataset[i + look_back, 0])
16     return numpy.array(dataX), numpy.array(dataY)
17
18 # fix random seed for reproducibility
19 numpy.random.seed(7)
20 # load the dataset
21 dataframe = read_csv('international-airline-passengers.csv', usecols=[1], engine='python', skipfooter=3)
22 dataset = dataframe.values
23 dataset = dataset.astype('float32')
24 # split into train and test sets
25 train_size = int(len(dataset) * 0.67)
26 test_size = len(dataset) - train_size
27 train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]
```

## Get 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

Get Your Start in Machine Learning

```
28 # reshape dataset
29 look_back = 3
30 trainX, trainY = create_dataset(train, look_back)
31 testX, testY = create_dataset(test, look_back)
32 # create and fit Multilayer Perceptron model
33 model = Sequential()
34 model.add(Dense(12, input_dim=look_back, activation='relu'))
35 model.add(Dense(8, activation='relu'))
36 model.add(Dense(1))
37 model.compile(loss='mean_squared_error', optimizer='adam')
38 model.fit(trainX, trainY, epochs=400, batch_size=2, verbose=2)
39 # Estimate model performance
40 trainScore = model.evaluate(trainX, trainY, verbose=0)
41 print('Train Score: %.2f MSE (%.2f RMSE)' % (trainScore, math.sqrt(trainScore)))
42 testScore = model.evaluate(testX, testY, verbose=0)
43 print('Test Score: %.2f MSE (%.2f RMSE)' % (testScore, math.sqrt(testScore)))
44 # generate predictions for training
45 trainPredict = model.predict(trainX)
46 testPredict = model.predict(testX)
47 # shift train predictions for plotting
48 trainPredictPlot = numpy.empty_like(dataset)
49 trainPredictPlot[:, :] = numpy.nan
50 trainPredictPlot[look_back:len(trainPredict)+look_back, :] = trainPredict
51 # shift test predictions for plotting
52 testPredictPlot = numpy.empty_like(dataset)
53 testPredictPlot[:, :] = numpy.nan
54 testPredictPlot[len(trainPredict)+(look_back*2)+1:len(dataset)-1, :] = testPredict
55 # plot baseline and predictions
56 plt.plot(dataset)
57 plt.plot(trainPredictPlot)
58 plt.plot(testPredictPlot)
59 plt.show()
```

Running the example provides the following output.

```
1 Epoch 395/400
2 0s - loss: 485.3482
3 Epoch 396/400
4 0s - loss: 479.9485
5 Epoch 397/400
6 0s - loss: 497.2707
7 Epoch 398/400
8 0s - loss: 489.5670
9 Epoch 399/400
10 0s - loss: 490.8099
11 Epoch 400/400
12 0s - loss: 493.6544
```

## Get 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

Get Your Start in Machine Learning

13 Train Score: 564.03 MSE (23.75 RMSE)

14 Test Score: 2244.82 MSE (47.38 RMSE)

We can see that the error was not significantly reduced compared to that of the previous section.

Looking at the graph, we can see more structure in the predictions.

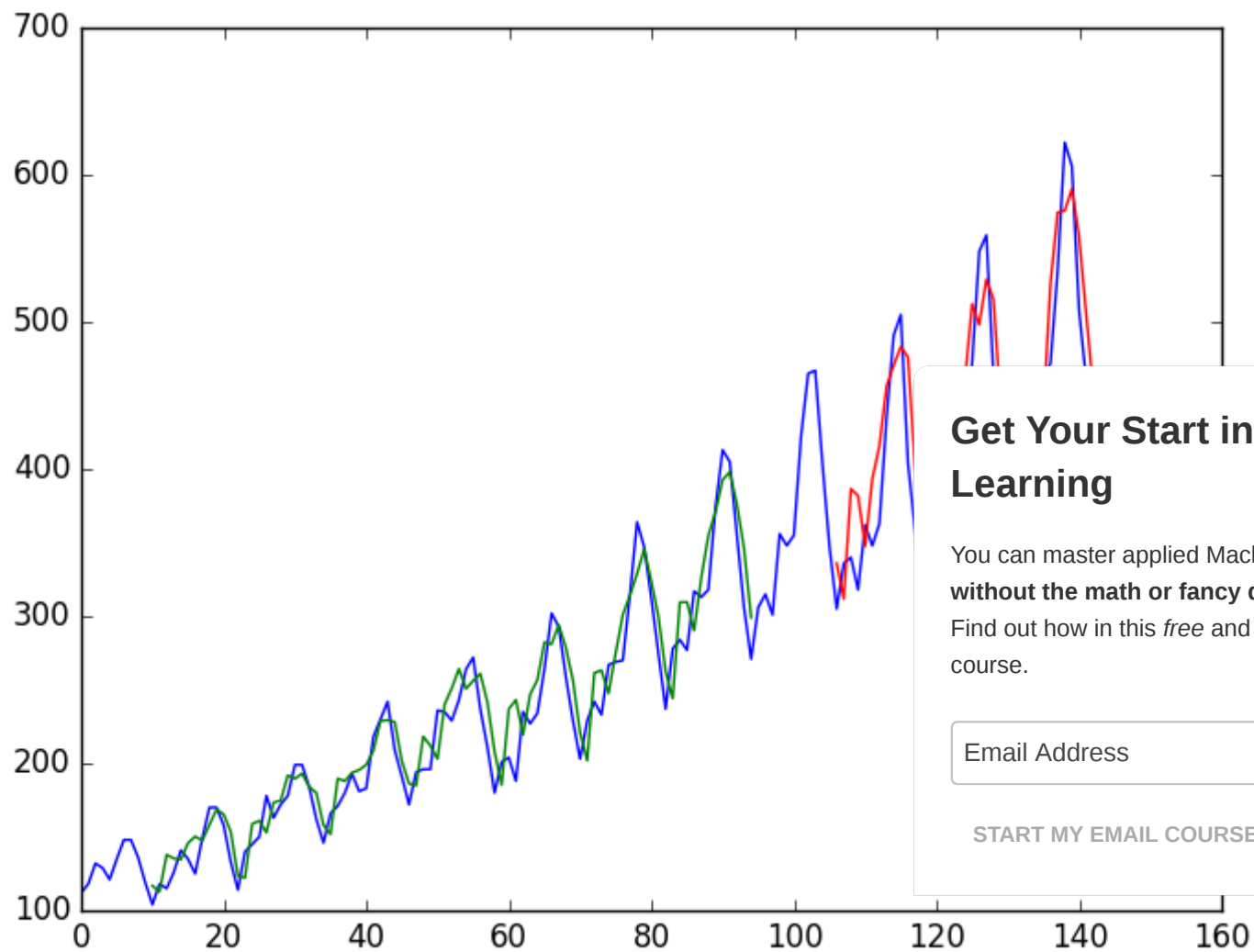
Again, the window size and the network architecture were not tuned, this is just a demonstration of how to frame a prediction problem.

Taking the square root of the performance scores we can see the average error on the training dataset was 23 passengers (in thousands per month) and the average error on the unseen test set was 47 passengers (in thousands per month).

## Get 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



Window Method For Time Series Predictions With Neural Networks  
Blue=Whole Dataset, Green=Training, Red=Predictions

## Get 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

## Summary

Get Your Start in Machine Learning

In this post, you discovered how to develop a neural network model for a time series prediction problem using the Keras deep learning library.

After working through this tutorial you now know:

- About the international airline passenger prediction time series dataset.
- How to frame time series prediction problems as a regression problems and develop a neural network model.
- How use the window approach to frame a time series prediction problem and develop a neural network model.

Do you have any questions about time series prediction with neural networks or about this post?

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

## Frustrated With Your Progress In Deep Learning?

### What If You Could Develop A

...with just a few lines of

Discover how in my new Ebook: [Deep Learning in Python](#)

It covers **self-study tutorials** and **end-to-end projects** including *Multilayer Perceptrons, Convolutional Nets and Recurrent Neural Networks*

### Finally Bring Deep Learning To Your Own Projects

Skip the Academics. Just

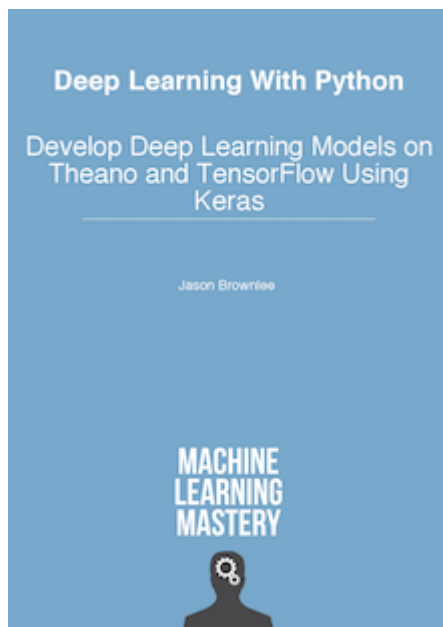
[Click to learn more.](#)

## Get 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



Get Your Start in Machine Learning



### About Jason Brownlee

Dr. Jason Brownlee is a husband, proud father, academic researcher, author, professional developer and a machine learning practitioner. He is dedicated to helping developers get started and get good at applied machine learning.

[Learn more.](#)

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

< [How To Estimate The Performance of Machine Learning Algorithms in Weka](#)

[How To Estimate A Basic Performance Metric For Machine Learning Models In Weka](#) >

## 140 Responses to *Time Series Prediction With Deep Learning in Keras*



**Steve Buckley** August 13, 2016 at 2:12 am #

Hi Jason,

This is a new tool for me so an interesting post to get started!

It looks to me like your plot for the first method is wrong. As you're only giving the previous time point to straight line and won't pull out the periodicity your plot suggests. The almost perfect fit of the red line to the blue line also doesn't reflect the much worse fit suggested in the model score!

Hope that's helpful.

### Get 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](#)



**Curious George** August 17, 2016 at 2:29 am #

[REPLY](#) ↩

[Get Your Start in Machine Learning](#)



Hi Jason,

How can you use this technique to forecast into the future?

Thanks!



**Jason Brownlee** August 17, 2016 at 9:52 am #

REPLY ↩

This example is forecasting  $t+1$  in the future.



**Curious George** August 25, 2016 at 1:18 am #

In order to forecast  $t+2$ ,  $t+3$ ,  $t+n, \dots$ , is it recommended to use the previous prediction ( $t$ )? For example, if I wanted to forecast  $t+2$ , I would use the available data including my prediction at  $t+1$ . I understand that the error would increase the further out the forecast due to relying on prediction. Thoughts?



**Jason Brownlee** August 25, 2016 at 5:05 am #

Yes, using this approach will provide multiple future data points. As you suggest, the errors are to compound.

Give it a go, it's good to experiment with these models and see what they are capable of.



**Curious George** August 25, 2016 at 1:51 am #

REPLY ↩

Also, when running the full code snippet using the window method, the graph produced

## Get 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

Get Your Start in Machine Learning

This is what I'm getting

<http://imgur.com/a/NaoYE>



**Jason Brownlee** August 25, 2016 at 5:06 am #

REPLY ↩

I did update the plotting code with a minor change and did not update the images accordingly. I will update them ASAP.



**Andy** March 1, 2017 at 6:03 am #

REPLY ↩

Could you show an example where maybe there was a couple more features. So, say you knew about temperature and day of the week (Mon-Sun).



**Jason Brownlee** March 1, 2017 at 8:46 am #

Hi Andy,

Yes, I am working on more sophisticated time series tutorials at the moment, they should be



**Soren Pallesen** June 9, 2017 at 6:18 pm #

Look forward to these time series forecast with multiple features examples – when do you expect to post them to your blog?

As always thx for this valuable resource and for sharing your experience !



**Jason Brownlee** June 10, 2017 at 8:19 am #

## Get 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

Get Your Start in Machine Learning

Perhaps a month. No promises. I am taking my time to ensure they are good.

**Ramzan Shahid** November 10, 2017 at 4:51 am #

REPLY ↩

Sir please, share some tutorial on tensorflow and what are the differences to make models in tensorflow and keras. thanks



**Jason Brownlee** November 10, 2017 at 10:41 am #

REPLY ↩

Tensorflow is like coding in assembly, Keras is like coding in Python.

Keras is so much simpler and makes you more productive, but gives up some speed and flexibility.



**Keshav Mathur** August 30, 2016 at 7:20 am #

Hello,

Thank you for a great article. I have a big doubt and also related to the plot posted in the earlier comment. We are training the model on  $t$  to get predictions for  $t+1$ .

Given this I would assume that when the model sees an input of 112 it should predict around 118 (first 5 train predictions show). Copying the top 5 train points and their subsequent predictions generated by the code

```
trainX[:5] trainPredict[:5]
[ 112.], [112.56],
[ 118.], [118.47],
[ 132.], [132.26],
[ 129.], [129.55],
[ 121.] [121.57],
```

I am trying to understand from a model perspective as to why is it predicting with a lag?

## Get 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

Get Your Start in Machine Learning



**Jason Brownlee** October 9, 2016 at 10:54 am #

REPLY ↩

Thanks Keshav, I have updated the description and the graphs.



**Jev** September 5, 2016 at 7:17 am #

REPLY ↩

Just as Steve Buckley pointed out, your first method seems to be wrong. The model indeed just fits a straight line ( $y_{Pred} = a \cdot X + b$ ), which can be verified by calculating predictions on an input such as `arange(200)`. Because you shift the results afterwards before plotting, the outcome seems very good. However, from a conceptual point of view, it should be impossible to predict  $X_{t+1}$  correctly based on only  $X_t$ , as the latter contains no trend or seasonal information.

Here is what I've got after trying to reproduce your results:

```
X Y yPred
0 112.0 118.0 112.897537
1 118.0 132.0 118.847107
2 132.0 129.0 132.729446
3 129.0 121.0 129.754669
....
```

as you can see, the  $y_{Pred}$  is way off (it should be equal to  $Y$ ), but looks good when shifted one period.

## Get 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



**Jason Brownlee** October 9, 2016 at 10:55 am #

Yep, right on Jev, thanks. I have updated the description and the graphs.



**Max Clayer** September 14, 2016 at 4:01 am #

REPLY ↩

Hi, Jason

Get Your Start in Machine Learning

I also have to agree with Jev, I would expect using `predict(trainX)` would give values closer to `trainY` values not `trainX` values.



**Jason Brownlee** October 9, 2016 at 10:56 am #

REPLY ↩

They do Max, you're right. I have updated the graphs to better reflect the actual predictions made.



**Himadri** September 24, 2016 at 11:56 pm #

REPLY ↩

Hi Jason,

Thanks for such a wonderful tutorial!

I was just wondering if in function `create_dataset`, there should be `range(len(dataset)-1)` in the loop. Here

...

```
trainPredictPlot[lb:len(train),:] = trainPredict
```

...

```
testPredictPlot[len(train)+lb:len(dataset),:] = testPredict
```

I am just in a big confusion with the index and getting somewhat difference plot for `look_back=3` : <http://i>

**Veltzer Doron** September 26, 2016 at 6:06 pm #

Hey, thanks for a most helpful tutorial, any ideas why this seems to work better than the time series predictions using `TRAIN` and `EST` in the sister tutorial? My intuition predicts the opposite.

## Get 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



**Jason Brownlee** September 27, 2016 at 7:41 am #

REPLY ↩

I'm glad you like it Veltzer.

Get Your Start in Machine Learning

Great question, the LSTMs probably require more fine tuning I expect.



**Newbtothis** September 29, 2016 at 12:02 pm #

REPLY ↩

Hey there! Great blog and articles – the examples really help a lot! I'm new to this so excuse the stupid question if applicable – I want to predict the next three outputs based on the same input. Is that doable in the LSTM framework? This is for predicting the water temperature for the next 3 days.



**Jason Brownlee** September 30, 2016 at 7:48 am #

REPLY ↩

Yes, this is called sequence to sequence prediction.

I see two main options:

- Run the LSTM 3 times and feed output as input.
- Change the LSTM to output 3 numbers.



**Han** September 30, 2016 at 11:37 am #

This particular time-series has strong seasonality and looks exponential in trend. In reality, the growth rate is decreasing. Could you plot the year-on-year growth rate?

## Get 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



**Jason Brownlee** October 1, 2016 at 8:00 am #

REPLY ↩

There would be benefit in modeling a stationary version of the data, I agree.

**Han** October 1, 2016 at 2:44 am #

Get Your Start in Machine Learning



I agree with Steve Buckley. The code is predicting  $x[i+1] = x[i]$  (approximately), that why the last part of code, which is supposed to fix the shift, couldn't get the shift part right.

Try the following: pick any point in your testX, say testX[i], use the model to predict testY[i], then instead of using testX[i+1], use testY[i] as the input parameter for model.predict(), and so on. You will end up with a nearly straight line.

I'd thank you for your wonderful posts on neural network, which helped me a lot when learning neural network. However, this particular code is not correct.



**Jeremy** October 5, 2016 at 1:58 pm #

REPLY ↩

Thanks for great article! It is really helpful for me. I have one question. If I have two more variable, how can i do? Take example, my data looks like follow,

date windspeed rain price

20160101 10 100 1000

20160102 10 80 1010

...

I'd like to predict the price.

## Get 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



**Jason Brownlee** October 6, 2016 at 9:26 am #

Hi Jeremy, each input would be a feature. You could then use the window method to frame features.

For example:

```
1 x1(t), x2(t), x1(t-1), x2(t-1) to predict y(t+1)
```



**Shimin** November 22, 2016 at 10:53 pm #

REPLY ↩

Hi Jason,

Get Your Start in Machine Learning

Thanks for your great explanation!

I have one question like Jeremy's. Is there any suggestion for me if I want to predict 2 variables? Data frame shown as below:

Date X1 X2 X3 X4 Y1 Y2

I want to predict Y1 and Y2. Also, Y1 and Y2 have some correlations.



**Jason Brownlee** November 23, 2016 at 8:59 am #

REPLY ↩

hi Shimin,

Yes, this is often called a sequence prediction problem in deep learning or a multi-step prediction problem.

You can use an LSTM with two outputs or you can use an MLP with two outputs to model the form.

I hope that helps.

## Get 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



**Sunny** October 18, 2016 at 9:09 am #

Jason,

Great writeup on using Keras for TS data. My dataset is something like below: # print the

```
Date Time Power1 Power2 Power3 Meter1 Meter2
12/02/2012 02:53:00 2.423 0.118 0.0303 0.020 1.1000
```

My feature vectors/predictors are Date, Time, Power1, Power2, Power3, Meter1. i am trying to predict Meter 2.

I would like to instead of using MLP use RNN/LSTM for the above time series prediction.

Can you pl. suggest is this is possible? and if yes, any pointers would help

thanks

Sunny

Get Your Start in Machine Learning





**Jason Brownlee** October 19, 2016 at 9:10 am #

REPLY ↩

Hi Sunny, this tutorial will help to get you started:

<http://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>



**nicoad** October 31, 2016 at 7:52 pm #

REPLY ↩

Hello , nice tutorial .

I have one question : it would be usefull to have similar stuff on live data. let s say I have access to some real time data (software downloads, stock price ...) , would it requires to train the model each time new data is available ?



**Jason Brownlee** November 1, 2016 at 7:59 am #

I agree nicoad, a real-time example would be great. I'll look into it.

A great thing about neural networks is that they can be updated with new data and do not have to be retrained.



**sherlockatszx** November 8, 2016 at 3:36 am #

Hi, your original post code is to use 1(or 3) dimension X to predict the later 1 dimension Y. how about 50th. what i mean is i increase the time unit i want to predict , predict 3 or even 10 time unit . under such condition : does that mean i just change the output\_dim of the last output layer :

```
model.add(Dense(
output_dim=3))
```

Is that right?

## Get 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

Get Your Start in Machine Learning



**Jason Brownlee** November 8, 2016 at 9:58 am #

REPLY ↩

Yes, that looks right. Let me know how you go.



**sherlockatszx** November 8, 2016 at 6:56 pm #

REPLY ↩

Hi jason, I make a quick experiment in jupyter notebook and published in the github  
github:<https://github.com/sherlockhoatszx/TimeSeriesPredctionUsingDeeplearning>  
the code could work.

However If you look very carefully of the trainPredict data(IN[18] of the notebook).

the first 3 array is:

```
array([[ 128.60112 , 127.5030365 ],  
 [ 121.16256714, 122.3662262 ],  
 [ 144.46884155, 145.67802429])
```

the list inside [ 128.6,127.5 ] [121,2,122,3] does not like t+1 and t+2.

**\*\*Instead,\*\*** It looks like 2 probaly prediction for 1 unit.

What i means is [128.6,127.5] doesn't mean t+1 and t+2 prediction, it most possibly mean 2 pos  
one output cell with 2dimension and 2 output cell with 1 dimension is different.

I discussed it with other guy in github .

[https://github.com/VictOrSch/deep\\_learning/issues/11](https://github.com/VictOrSch/deep_learning/issues/11)

It seems i should use seq2seq or use timedistributed wrapper .

I stilll explored this and have not got one solution .

What is your suggestion?

## Get 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



**Jason Brownlee** November 9, 2016 at 9:49 am #

REPLY ↩

That does sound like good advice. Treat the problem as sequence to sequence problem.

Get Your Start in Machine Learning



**sherlockatszx** November 8, 2016 at 8:28 pm #

REPLY ↩

hi jason , I made a experiment on the jupyter notebook and published on the github .The code could output 2 columns data.

<https://github.com/sherlockhoatszx/TimeSeriesPredctionUsingDeeplearning/blob/master/README.md>

However! If you look very carefully of the trainPredict data(IN[18] of the notebook).

the first 3 array is:

```
array([[ 128.60112 , 127.5030365 ],  
 [ 121.16256714, 122.3662262 ],  
 [ 144.46884155, 145.67802429]
```

the list inside [ 128.6,127.5 ] [121,2,122,3] does not like t+1 and t+2.

**\*\*Instead,\*\*** It looks like 2 probaly prediction for 1 unit.

What i means is [128.6,127.5] doesn't mean t+1 and t+2 prediction, it most possibly mean 2 possible pre  
1 output cell with 2 dimension and 2 output cell with 1 dimension is different.

The input dimension and the output dimension will be tricky for the NN.

## Get 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



**Xiao** November 16, 2016 at 1:59 am #

Thanks Jason for the conceptual explaining. I have one question about the KERAS package:

It looks you input the raw data (x=118 etc) to KERAS. Do you know whether KERAS needs to standardize distribution with mean of 0?

— Xiao

REPLY ↩



**Jason Brownlee** November 16, 2016 at 9:32 am #

Great question Xiao,

It is a good idea to standardize data or normalize data when working with neural networks. Try it on your model.

Get Your Start in Machine Learning



**Satoshi Report** November 19, 2016 at 2:24 pm #

REPLY ↩

Wasn't the data normalized in an early version of this post?



**Jason Brownlee** November 22, 2016 at 6:46 am #

REPLY ↩

I don't believe so Satoshi.

Normalization is a great idea in general when working with neural nets, though.



**charith** December 12, 2016 at 8:11 pm #

I keep getting this error `dt = datetime.datetime.fromordinal(ix).replace(tzinfo=UTC)`

`ValueError: ordinal must be >= 1`



**Jason Brownlee** December 13, 2016 at 8:05 am #

Sorry charith, I have not seen this error before.

## Get 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



**Trex** January 7, 2017 at 8:07 am #

REPLY ↩

In Your text you say, the window size is 3, But in Your Code you use `loop_back = 10` ?

Get Your Start in Machine Learning



**Jason Brownlee** January 7, 2017 at 8:41 am #

REPLY ↩

Thanks Trex.

That is a typo from some experimenting I was doing at one point. Fixed.



**Trex** January 7, 2017 at 11:44 am #

REPLY ↩

No problem,

I have another question:

what the algorithm now does is predict 1 value. I want to predict with this MLP like n-values.

How should this work?



**Jason Brownlee** January 8, 2017 at 5:17 am #

Reframe your training dataset to match what you require and change the number of neurons to your desire.

## Get 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



**Mansolo** January 7, 2017 at 12:47 pm #

Hey Sir,

great Tutorial.

I am trying to build a NN for Time-Series-Prediction. But my Datas are different than yours.

I want to predict a whole next day. But a whole day is defined as 48 values.

Some lines of the Blank datas:

2016-11-10 05:00:00.000 0

Get Your Start in Machine Learning

2016-11-10 05:30:00.000 0  
2016-11-10 06:00:00.000 1  
2016-11-10 06:30:00.000 3  
2016-11-10 07:00:00.000 12  
2016-11-10 07:30:00.000 36  
2016-11-10 08:00:00.000 89  
2016-11-10 08:30:00.000 120  
2016-11-10 09:00:00.000 209  
2016-11-10 09:30:00.000 233  
2016-11-10 10:00:00.000 217  
2016-11-10 10:30:00.000 199  
2016-11-10 11:00:00.000 244

There is a value for each half an hour of a whole day.

i want to predict the values for every half an hour for the next few days. How could this work?



**Hem** January 8, 2017 at 3:43 pm #

Could you do an example for a Multivariate Time Series? 😊



**Jason Brownlee** January 9, 2017 at 7:48 am #

Yes, there are some tutorials scheduled on the blog. I will link to them once they're out.



**Bonje** January 12, 2017 at 12:19 am #

Why doesnt need the reLu Activation function that the input datas are normalized between 0 and 1?

If i use the sigmoid activation function, there is a must, that the input datas are normalized.

## Get 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

REPLY ↩

Get Your Start in Machine Learning

But why reLu doesnt need that?



**Jason Brownlee** January 12, 2017 at 9:28 am #

REPLY ↩

Generally, because of the bounds of the sigmoid function imposes hard limits values outside of 0-1.

The Rectifier function is quite different, you can read up on it here:

[https://en.wikipedia.org/wiki/Rectifier\\_\(neural\\_networks\)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks))

I'd recommend implementing in excel or Python and having a play with inputs and outputs.



**Bonje** January 12, 2017 at 1:10 am #

Another Question:

Your Input Layer uses reLu as activation Function.

But why has your Output Layer no activation Function? Is there a default activation function which keras uses if no, why is it possible to have a Layer without a activation function in it?

Thanks 😊



**Jason Brownlee** January 12, 2017 at 9:31 am #

Yes the default is Linear, this is a desirable activation function on regression problems.



**Bonje** January 12, 2017 at 1:11 am #

dont give one as parameter\*

REPLY ↩

## Get 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

Get Your Start in Machine Learning



**Dmitry N. Medvedev** February 4, 2017 at 1:11 pm #

REPLY ↩

A stupid question, sir.

Suppose I have a dataset with two fields: "date" (timestamp), "amount" (float32) describing a year.

on the first day of each month the amount is set to -200.

This is true for 11 months, except for the 12th (December).

Is there a way to train a NN so that it returns 12, marking the December as not having such and amount on its first day?



**Jason Brownlee** February 5, 2017 at 5:15 am #

Sorry Dmitry, I'm not sure I really understand your question.

Perhaps you're able to ask it a different way or provide a small example?



**Thomas Durant** February 9, 2017 at 3:36 pm #

Is it common to only predict the single next time point? Or are there times/ways to predict 2,3, and more? How do you assess performance metrics for those predictions?

## Get 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



**Jason Brownlee** February 10, 2017 at 9:50 am #

REPLY ↩

Good question Thomas.

The forecast time horizon is problem specific. You can predict multiple steps with a MLP or LSTM using multiple neurons in the output layer.

Evaluation is problem specific but could be RMSE across the entire forecast or per forecast lead time.

Get Your Start in Machine Learning





**zhou** February 28, 2017 at 8:34 pm #

REPLY ↩

thanks for Jason's post, I benefit a lot from it. now I have a problem:how can I get the passengers in 1961-01? anticipates your reply.



**Jason Brownlee** March 1, 2017 at 8:36 am #

REPLY ↩

You can train your model on all available data, then call `model.predict()` to forecast the next out of sample observation.



**zhou** March 3, 2017 at 1:42 pm #

it seems the model can't forecast the next month in future ?



**Jason Brownlee** March 6, 2017 at 10:41 am #

What do you mean exactly zhou?



**zhou** March 8, 2017 at 4:39 pm #

sorry. I want to forecast the passengers in future, what should I do?

## Get 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



**Viktor** March 2, 2017 at 12:08 am #

REPLY ↩

Thanks for the tutorial, Jason. it's very useful. It would be nice to also know how you chose the optimizing them.

Get Your Start in Machine Learning



**Jason Brownlee** March 2, 2017 at 8:17 am #

REPLY ↩

Thanks Viktor, I hope to cover more tutorials on this topic.

You can see this post on how to best tune an MLP:

<http://machinelearningmastery.com/improve-deep-learning-performance/>



**0xKA** March 6, 2017 at 7:35 pm #

REPLY ↩

In the first case. If I shift model to the left side, it will be a good model for forecasting because possible to do that ?



**Jason Brownlee** March 7, 2017 at 9:35 am #

Can you give an example of what you mean?



**Sphurti** March 22, 2017 at 4:29 pm #

Is there any specific condition to use activation functions? how to decide which activation function is more suitable for linear or non-linear datasets?



**Jason Brownlee** March 23, 2017 at 8:47 am #

REPLY ↩

There are some rules.

Relu in hidden because it works really well. Sigmoid for binary outputs, linear for regression outputs,

**Get Your Start in Machine Learning**

**Get 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

Often you can transform your data for the bounds of a given activation function (e.g. 0,1 for sigmoid, -1,1 for tanh, etc.)

I hope that helps as a start.



**Sphurti** March 23, 2017 at 5:51 pm #

REPLY ↩

how to decide the optimizer? Is there any relevance with activation function?



**Jason Brownlee** March 24, 2017 at 7:53 am #

REPLY ↩

Not really. It's a matter of taste it seems (speed vs time).

## Get 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



**John** March 29, 2017 at 1:37 am #

What kind of validation are you using in this tutorial? is it cross validation?



**Jason Brownlee** March 29, 2017 at 9:08 am #

No, a train-test split.



**Sphurti** March 29, 2017 at 3:31 pm #

REPLY ↩

Is there any another deep learning algorithms that can be used for time series prediction? why to prefer multilayer perceptron for time series prediction?

Get Your Start in Machine Learning



**Jason Brownlee** March 30, 2017 at 8:47 am #

REPLY ↩

Yes, you can use Long Short-Term Memory (LSTM) networks.



**Qiushi Wang** April 3, 2017 at 5:22 pm #

REPLY ↩

Hi Jason,

I always have a question, if we only predict 1 time step further ( $t+1$ ), the accurate predicted result is just copy the value of  $t$ , as the first figure shows. When we add more input like ( $t-2$ ,  $t-1$ ,  $t$ ), the predicted result get worse. Even compare with other prediction method like ARIMA, RNN, this conclusion perhaps is still correct. To better exhibit the power of these prediction methods, should we try to predict more time steps?

Thanks



**Jason Brownlee** April 4, 2017 at 9:13 am #

It is a good idea to make the input data stationary and scale it. Then the network needs to be trained on the scaled data.



**Stephan Oelze** April 10, 2017 at 1:56 am #

Dear Jason.

Thanks for sharing your information here. Anyway i was not able to reproduce your last figure. On my machine it still looks like the “bad” figure.

<https://www2.pic-upload.de/img/32978063/auto.png>

I used the code as stated above. Where is my misunderstanding here?

<https://pastebin.com/EzvjnvGv>

Thank You!

silly me 😊

## Get 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

Get Your Start in Machine Learning



**Jason Brownlee** April 10, 2017 at 7:38 am #

REPLY ↩

Perhaps try fitting the network for longer?



**trupti** April 11, 2017 at 3:22 pm #

REPLY ↩

thanks for this post..actually I am referring this for my work. my dataset is linear. Can I use softplus or elu as an activation function for linear data?



**Jason Brownlee** April 12, 2017 at 7:50 am #

Yes, but your model may be more complex than is needed. In fact, you may be better off with Linear Regression.

## Get 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



**ikok** April 20, 2017 at 8:31 am #

Firstly thanks Jason, I try MLP and LSTM based models on my time series data, and I get some example count 1400, min value:21, max value 210 ) What is acceptance value of RMSE. ?



**Jason Brownlee** April 20, 2017 at 9:35 am #

REPLY ↩

Nice work!

An acceptable RMSE depends on your problem and how much error you can bear.

Get Your Start in Machine Learning



**Dmitry** April 21, 2017 at 6:58 pm #

REPLY ↩

Great article, thank you.

Is it possible to make a DNN with several outputs? For example the output layer has several neurons responsible for different flight directions. What difficulties can arise?



**Jason Brownlee** April 22, 2017 at 9:25 am #

REPLY ↩

Yes, try it.

Skill at future time steps often degrades quickly.



**piemonsparrow** April 21, 2017 at 10:50 pm #

Hello, Jason, i am a student, recently i am learning from your blog. Could you make a display d be very appreciated if you can, because i am a newer. Thank you!



**Jason Brownlee** April 22, 2017 at 9:26 am #

This example shows you how to display training history:

<http://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/>

## Get 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

**Hans** April 23, 2017 at 2:09 pm #

REPLY ↩

Does anybody have an idea/code snippet how to store observations of this example code in a variable, so that the variable can be used to to make predictions beyond the airline dataset (one step in the future)?

Get Your Start in Machine Learning

**Hans** April 24, 2017 at 3:20 pm #

REPLY ↩

Would it be logical incorrect to extend the testX-Array with for example [0,0,0] to forecast unseen data/ a step in the future?



**Jason Brownlee** April 25, 2017 at 7:45 am #

REPLY ↩

It would not be required.

Fit your model on all available data. When a new observation arrives, scale it appropriately, gather it with the other lag observations your model requires as input and call `model.predict()`.

**Hans** April 24, 2017 at 4:45 pm #

Is there a magic trick to get the right array-format for a prediction based on observations?  
I always get the wrong format:

```
obsv1 = testPredict[4]
obsv2 = testPredict[5]
obsv3 = testPredict[6]

dataset = obsv1, obsv2, obsv3
dataX = []
dataX.append(dataset)
#dataX.append(obsv2)
#dataX.append(obsv3)
myNewX = numpy.array(dataX)
```

## Get 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

Get Your Start in Machine Learning

**Hans** April 24, 2017 at 5:05 pm #

REPLY ↩

Update:

After several days I managed to make a prediction on unseen data in this example (code below).

Is this way correct?

How many observations should be used to get a good prediction on unseen data.

Are there standard tools available to measure corresponding performances and suggest the amount of observations?

Would this topic be the same as choosing the right window-size for time-series analysis, or where would be the difference?

Code:

```
obsv1 = float(testPredict[4])
obsv2 = float(testPredict[5])
obsv3 = float(testPredict[6])

dataX = []
myNewX = []
dataX.append(obsv1)
dataX.append(obsv2)
dataX.append(obsv3)
myNewX.append(dataX)
myNewX = numpy.array(myNewX)

futureStepPredict = model.predict(myNewX)
print(futureStepPredict)
```

## Get 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



**Jason Brownlee** April 25, 2017 at 7:48 am #

REPLY ↩

Looks fine.

The number of obs required depends on how you have configured your model.

The “best” window size for a given problem is unknown, you must discover it through trial and error

<http://machinelearningmastery.com/a-data-driven-approach-to-machine-learning/>

Get Your Start in Machine Learning



**Hans** April 28, 2017 at 1:04 pm #

REPLY ↩

Is there a method or trial and error-strategy to find out how many lag observations are 'best' for a forecast of unseen data?  
Is there a relation between look\_back (window size) and lag observations?  
In theory I could use all observations to predict one step of unseen data. Would this be useful?



**Jason Brownlee** April 29, 2017 at 7:20 am #

Look-back defines the lag.

You can use ACF and PACF plots to discover the most relevant lag obs:

<http://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/>

The "promise" of LSTMs is that they can learn the appropriate time dependence structure

**Hans** May 31, 2017 at 8:38 pm #

If I fill the model with 3 obs, I get 3 predictions/data points of unseen data.

If I only want to predict one step in the future, should I build an average of the resulting 3 predictions or should I simply use the last of the 3 prediction steps?

Thank you.



**Jason Brownlee** June 2, 2017 at 12:46 pm #

REPLY ↩

I would recommend changing the model to make one prediction if only one time step prediction is required

Get Your Start in Machine Learning

## Get 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

**Hans** June 2, 2017 at 7:49 pm #

How would you change the Multilayer Perceptron model of this site in this regard?

**Hans** June 9, 2017 at 9:04 pm #

REPLY ↩

I have a misconception here. Don't do the same fellow reader!

With "obsv(n) = float(testPredict[n])" I took predictions of the test dataset as observations.

THAT'S WRONG!

Instead we take a partition of the original raw data as x/observations to predict unseen data, with

Like in R:

<http://machinelearningmastery.com/finalize-machine-learning-models-in-r/#comment-401949>

Is this right Jason?

## Get 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



**Jason Brownlee** April 25, 2017 at 7:46 am #

If you need a 2D array with 1 row and 2 columns, you can do something like:

```
1 data = [0.1, 0.1]
2 data = numpy.array(data)
3 data = data.resize(1, 2)
```



**Md. Armanur Rahman** April 28, 2017 at 2:48 pm #

REPLY ↩

Hello Sir,

Get Your Start in Machine Learning

This is Arman from Malaysia. I am a student of Multimedia University. I want to do "Self-Tuning performance of Hadoop using Deep Learning". So which framework I will consider for this sort of problem. as like DBM, DBN, CNN, RNN ?

I need your suggestion.

With best regards

Arman



**Jason Brownlee** April 29, 2017 at 7:21 am #

REPLY ↩

I would recommend following this process:

<http://machinelearningmastery.com/start-here/#process>

**Hans** May 1, 2017 at 6:41 pm #

Are there any more concerns about this code. Or is it updated and technical correct now?



**Jason Brownlee** May 2, 2017 at 5:57 am #

We can always do things better.

For this example, I would recommend exploring providing the data as time steps and explore larger

## Get 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

**Hans** May 2, 2017 at 11:31 pm #

REPLY ↩

Hm, I'm not sure if I understand it right.

I believe I'm already feeding it with time-step like so:

Get Your Start in Machine Learning

```
return datetime.strptime(x, '%Y-%m-%d')
```

My raw data items have a decent date column. Is this what you meant?

How do we explore larger networks fit for more epochs?

I have everything parameterized in a central batch file now (pipeline).

Should I increase the epochs for...

```
model.fit(trainX, trainY, epochs=myEpochs, batch_size=myBatchSize, verbose=0)
```

Thank you.

**Hans** May 1, 2017 at 6:58 pm #

I'm trying to adapt some code from:

<http://machinelearningmastery.com/time-series-forecasting-long-short-term-memory-network-python/>

...and build the variable EXPECTED in the context of this script.

Unfortunately I don't know how to do it right. I'm a little bit frustrated at this point.

```
for i in range(len(test)): <-- what should I better use here?
expected = dataset[len(train) + i + 1] <-- what should I better use here?
print(expected)
```

This looks cool so far, could I use the index to retrieve a var called EXPECTED?

```
for i in range(len(testPredict)):
pre = '%.3f' % testPredict[i]
print(pre)
```

A code example would help to solve my index-confusions.

## Get 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



**Stefan** June 15, 2017 at 2:06 am #

REPLY ↩

This is a great example that machine learning is often much more than knowing how to use the algorithms / libraries. It's always important to understand the data we are working with. For this example as it is 1 dimensional this is luckily quite easily done. In the first example we are giving the the algorithm one previous value and ask it "What will the next value be?".

Since we use a neural net not taking into account any time behavior, this system is strongly overdetermined. There are a lot of values at the y value 290 for example. For half of them the values decline, for half of them the values increase. If we don't give the algorithm any indication, how should it know which direction this would be for the test datapoint? There is just not enough information.

One idea could be to additionally give the algorithm the gradient which would help in the decision whether we a rising or a falling value follows (which is somehow what we do when adding a lookback of 2). Yet, the results do obviously not improve significantly.

Here I want to come back to "understand the data you are dealing with". If we look at the plot, there are a rising trend and a periodicity. We want the algorithm to cover both. Only then, will the prediction be accurate. If we want the algorithm to cover that periodicity without ANN we have to at least provide it the data in a format to deduct this property.

Hence: Extending the lookback to 12 month (12 datapoints in the X) will lead to a significantly improved prediction. A higher feature dimension, which might not be desired due to computational reasons (doesn't matter for this example). We can take only 3 month steps at lookback (still look back 12 month but skip 2 months in the data). We still cover the trend and the algorithm provides almost the same performance for the "1 month ahead" prediction.

Another possibility would surely be to add the month (Jan, Feb, etc.) as a categorical feature.

## Get 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



**Jason Brownlee** June 15, 2017 at 8:46 am #

Thanks Stefan, very insightful.



**Paul** July 11, 2017 at 10:59 am #

REPLY ↩

Get Your Start in Machine Learning

Hello Jason! Thanks for the great example! I was looking for this kind of example.

I'm learning Neural Network these days and trying to predict the number which is temperature like this example, but I have more inputs to predict temperature. Then should I edit on the `pandas.read_csv(...,usecols[1],...)` to `usecols[0:4]` if I have 5 inputs?

Thanks in advance!

Best,  
Paul



**Paul** July 11, 2017 at 11:13 am #

REPLY ↩

I mean something like below

X1 X2 X3 X4 X5 Y1

380 17.00017 9.099979 4 744 889.7142

Thank you!

## Get 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

REPLY ↩



**Jason Brownlee** July 12, 2017 at 9:37 am #

This post might help you frame your prediction problem:

<http://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/>



**Paul** July 24, 2017 at 3:13 pm #

Thanks for replying me back! 😊 And sorry for late response.

When I clicked the link that you wrote, it requires username and password.. :(

**Jason Brownlee** July 25, 2017 at 9:28 am #

Get Your Start in Machine Learning



Sorry about that, fixed. Please try again.



**Paul** July 24, 2017 at 3:16 pm #

REPLY ↩

NVM. I figured out it was machinelearningmastery. instead of mlmastery.staging.wpengine.com 😊

Thanks. 😊

Best,  
Paul



**Jason Brownlee** July 25, 2017 at 9:30 am #

Yes, for some reason I liked to the staging version of my site, sorry about that.



**Barkin Tuncer** August 2, 2017 at 7:49 am #

Hey, I am trying to make a case where the test case is not given but the model should predict the code which takes the last row of the train data and predict a value from it then put the predicted value at the end of the test data. After doing this procedure for let say len(testX) times. It ended up like an exponential graph. I can upload the code. I dont understand why it works like that. I hope you can enlighten me.

```
prediction=np.zeros((testX.shape[0],1))
test_initial=trainX[-1].copy()
testPredictFirst = model.predict(test_initial.reshape(1,3))
new_=create_pred(test_initial,testPredictFirst[0][0])
prediction[0]=testPredictFirst
```

```
for k in range(1,len(testX)):
    testPredict=model.predict(new_.reshape(1,3))
    new_=create_pred(new_,testPredict[0][0]) #this code does if new_ is [1,2,3] and testPredict[0][0] is 4 the
```

## Get 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

Get Your Start in Machine Learning

prediction[k]=testPredict



**rohini** August 2, 2017 at 8:55 pm #

REPLY ↩

really awesome and useful to0



**Jason Brownlee** August 3, 2017 at 6:49 am #

REPLY ↩

Thanks.



**Jay Shah** August 15, 2017 at 7:34 pm #

Hi,

It's awesome article. Very Helpful. I implemented these concepts in my Categorical Time Series Forecasting. My Time Series can take only 10 values from 0 to 9. I've approx 15k rows of data. I want to predict next values. But the issue is '1' appears in time series most of the time. So starting from 2nd or 3rd epoch LSTM prediction Hyperparameter but it's not working out. Can you please point out what could be the approach to solve this?

## Get 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



**Jason Brownlee** August 16, 2017 at 6:32 am #

REPLY ↩

Perhaps your problem is too challenging for the chosen model.

Try testing with an MLP with a large window size. The search hyperparameters of the model.

Get Your Start in Machine Learning





**Patt** September 10, 2017 at 5:48 am #

REPLY ↩

I'm new to coding. How can I predict  $t+1$  from your example code? I mean from your code I want the value of  $t+1$  or can you more explanation about the code where it predicts  $t+1$ .



**Jason Brownlee** September 11, 2017 at 12:05 pm #

REPLY ↩

Perhaps start with something simpler if you are new to coding, for example simpler linear models:  
<https://machinelearningmastery.com/start-here/#timeseries>



**Dogan** September 14, 2017 at 2:57 am #

Hi Jason,

Why do you think making the data stationary is a good idea in this approach? I know ARIMA assumes the data is stationary. Do you think normalization is enough for neural networks in general? I thought normalization would be enough.



**Jason Brownlee** September 15, 2017 at 12:08 pm #

Yes, it will make the problem easier to model.

## Get 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



**karan** September 24, 2017 at 10:55 pm #

REPLY ↩

I am getting this error:  
Help me please i am new here. i am using tensorflow

Get Your Start in Machine Learning

Traceback (most recent call last):

File "international-airline-passengers.py", line 49, in

testPredictPlot[len(trainPredict)+(look\_back\*2)+1:len(dataset)-1, :] = testPredict

ValueError: could not broadcast input array from shape (94,1) into shape (46,1)



**karan** September 24, 2017 at 11:35 pm #

REPLY ↩

I got my error. It was silly mistake.

thanks



**Jason Brownlee** September 25, 2017 at 5:38 am #

Glad to hear you worked it out.



**Sanam** September 26, 2017 at 10:47 pm #

Hi Jason,

Thankyou so much for all this . I have a question ! Why the obtained accuracy of regression using theano, tensorflow or keras. However , if we try to train MLP or anyother model by using very good accuraccy in terms of e power negative values. why is that so ?

## Get 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



**Jason Brownlee** September 27, 2017 at 5:41 am #

Accuracy is a score for classification algorithms that predict a label, RMSE is a score for regression algorithms that predict a quantity.

Get Your Start in Machine Learning



**MaCa** October 5, 2017 at 3:21 am #

REPLY

Hi Jason,

Maybe I am not understanding something.

You say something like

“We can see that the model did a pretty poor job of fitting both the training and the test datasets. It basically predicted the same input value as the output.” when talking about the first image. I don’t understand how that prediction is bad. It looks very very good to me. I am asking because I tried your code with my own dataset and I obtained something similar, i.e. it looked perfect except it was slightly shifted. But how is it bad?

Also in the following section you say

“Looking at the graph, we can see more structure in the predictions.”

How do we see the structure? To me it looks like it is less precise than the first one.

Apologies if I quoted you twice, but I don’t really understand...



**Jason Brownlee** October 5, 2017 at 5:27 am #

If the model cannot do better than predicting the input as the output, then the model is not a model:

<https://machinelearningmastery.com/persistence-time-series-forecasting-with-python/>



**Wawan** November 2, 2017 at 1:19 am #

Hi Jason

Do you how train data in PyCharm with Dynamic CNN

Please give us more explanation..

thank you

## Get 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

Get Your Start in Machine Learning



**Alessandro** December 15, 2017 at 2:21 am #

REPLY ↩

Hi Jason,

I think I'm a little confused.

Your post seems to address how to forecast  $t+1$  from  $t$ .

The output however looks pretty poor as it ends up performing as a persistence model.

What is the value of using keras to achieve the same goal as a persistence model then?

How would you modify your network to try to perform better than a common persistence model?

What would the model structure look like?

Thanks in advance!



**Jason Brownlee** December 15, 2017 at 5:37 am #

I would recommend an MLP tuned to the problem with many lag variables as input.



**Volodymyr** December 15, 2017 at 7:40 am #

Hi Jason, thx for great tutorial, but i cant find value  $t+1$ . And can we use it for predicting stock p



**Jason Brownlee** December 15, 2017 at 3:31 pm #

Use a persistence model to predict short-term stock prices:

<https://machinelearningmastery.com/gentle-introduction-random-walk-times-series-forecasting-python/>

REPLY ↩

## Get 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

Get Your Start in Machine Learning

## Leave a Reply

Name (required)

Email (will not be published) (required)

Website

### 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](#)

## Get 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.

Finally Get Started With Deep Learning

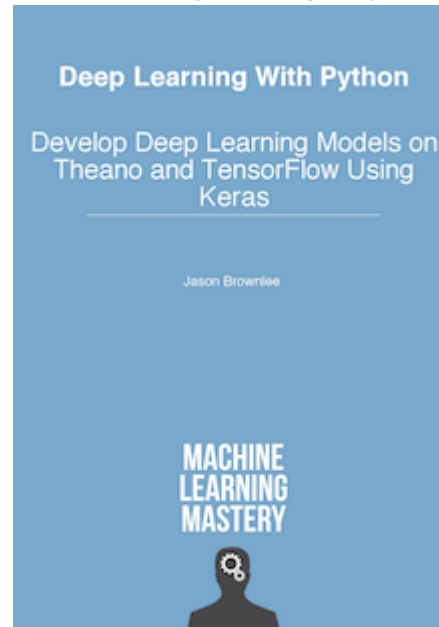
Get Your Start in Machine Learning

Sick of the fancy math and need for super computers?

Looking for step-by-step tutorials?

Want end-to-end projects?

Get Started With Deep Learning in Python Today!



## Get 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

### 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

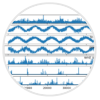


**Develop Your First Neural Network in Python With Keras Step-By-Step**

MAY 24, 2016

**Multivariate Time Series Forecasting with LSTMs in Keras**

**Get Your Start in Machine Learning**



AUGUST 14, 2017



How to Setup a Python Environment for Machine Learning and Deep Learning with Anaconda

MARCH 13, 2017



Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras

JULY 26, 2016



Time Series Forecasting with the Long Short-Term Memory Network in Python

APRIL 7, 2017



Regression Tutorial with the Keras Deep Learning Library in Python

JUNE 9, 2016



Multi-Class Classification Tutorial with the Keras Deep Learning Library

JUNE 2, 2016



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

AUGUST 9, 2016

## Get 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](#)

© 2018 Machine Learning Mastery. All Rights Reserved.

[Privacy](#) | [Contact](#) | [About](#)

Get Your Start in Machine Learning