

://service.weibo.com/share/share.php?



(https://yq.aliyun.com/webinar/play/469)

基于Keras的LSTM多变量时间序列预测

【方向】 (/users/1834978585139856) 2017-08-17 23:04:31 浏览31005 评论6

- 云栖社区 (/tags/type_blog-tagid_1/)
- python (/tags/type_blog-tagid_14/)
- 测试 (/tags/type_blog-tagid_1002/)
- test (/tags/type_blog-tagid_1773/)
- LSTM (/tags/type_blog-tagid_13670/)
- 时间序列 (/tags/type_blog-tagid_13923/)
- Keras (/tags/type_blog-tagid_18314/)

摘要： 还在为设计多输入变量的神经网络模型发愁？来看看大神如何解决基于Keras的LSTM多变量时间序列预测问题！文末附源码！

更多深度文章，请关注：https://yq.aliyun.com/cloud (https://yq.aliyun.com/cloud)

LSTM是一种时间递归神经网络，它出现的原因是为了解决RNN的一个致命的缺陷。原生的RNN会遇到一个很大的问题，叫做The vanishing gradient problem for RNNs，也就是后面时间的节点会出现老年痴呆症，也就是忘事儿，这使得RNN在很长一段时间内都没有受到关注，网络只要一深就没法训练。后来有些大牛们开始使用递归神经网络来对时间关系进行建模。而根据深度学习三大牛的阐述，LSTM网络已被证明比传统的RNNs更加有效。

适合多输入变量的神经网络模型一直让开发人员很头痛，但基于（LSTM）的循环神经网络能够几乎可以完美的解决多个输入变量的问题。

基于（LSTM）的循环神经网络可以很好的利用在时间序列预测上，因为很多古典的线性方法难以适应多变量或多输入预测问题。

在本教程中，你会看到如何在Keras深度学习库中开发多变量时间序列预测的LSTM模型。

读完本教程后，你将学会：

- 如何将原始数据集转换为可用于时间序列预测的数据集。
- 如何准备数据并创建适应多变量时间序列预测问题的LSTM。
- 如何做出预测并将结果重新调整到原始单位。

本教程分为3部分：

- 1.空气污染预报。
- 2.基本数据准备。
- 3.多变量LSTM预测模型。

Python环境

本教程假设你已安装Python SciPy环境，你可以在本教程中使用Python 2或3。你必须使用TensorFlow或Theano后端安装Keras（2.0或更高版本）。本教程还假设你已经安装了scikit-learn，Pandas，NumPy和Matplotlib。如果你需要帮助你的环境，请参阅这篇文章：

如何为Anaconda设置机器学习和深度学习的Python环境 (http://machinelearningmastery.com/setup-python-environment-machine-learning-deep-learning-anaconda/)

空气污染预报

达人介绍



【方向】 (/users/1834978585139856)

文章 563篇 | 关注

关注

目录

- 1 Python环境
- 2 空气污染预报
- 3 基本数据准备
- 4 多变量LSTM预测模型建立：
- 5 LSTM数据准备
- 6 定义和拟合模型
- 7 评估模型
- 8 完整例子
- 9 进一步阅读
- 10 总结：

文中提到的云产品

- 移动测试 (/go/1/113?postion=2)

移动测试（Mobile Testing）是为广大企业客户者提供真机测试服务的云平台... 查看详情 (/go/1/113?postion=2)
- 云效 (/go/1/176?postion=2)

业内领先的面向企业的一站式研发提效平台（通过项目流程管理和专项自动化提效工具，能... (/go/1/176?postion=2)
- 性能测试 (/go/1/42?postion=2)

全球领先的SaaS性能测试平台，具有强大的分力，可模拟海量用户真实的业务场景，让应... (/go/1/42?postion=2)
- 云服务器ECS (/go/1/3?postion=2)

为您提供简单高效、处理能力弹性伸缩的计算服务。助您快速构建更稳定、安全的应用，提升运维效... (/go/1/3?postion=2)

博主其他文章

[更多>](#) (/users/1834978585

cles/613771)

文本挖掘方法么，过

articles/610496)

马?

?

的性能吗?

数据分析 (Part I)

中找到心仪的工作

开始构建图像搜索

开始构建图像搜索

相关话题

[更多>](#)

程序媛比码大赛 不秀美颜 秀代码 (/roundtat
关于代码的那些低级错误 (/roundtable/1201-
为什么DevOps很好，80%的公司却无法落址
所理解的DevOps (/roundtable/120182)
云栖社区内容开发平台伙伴招募计划，用户：
地! (/roundtable/119885)
8012年了，云栖社区首页长这样了，你觉得
(/roundtable/119769)

网站建设特惠

建站流程详解，优选建站低至4折

立即抢购

(<https://promotion.aliyun.com/ntms/act/jianzhaagentsource=yqneirongbanner>)

在这个小例子中，我们将使用空气质量数据集。这是一个数据集，在美国驻北京大使馆五年内报告天气和污染水平。数据类型包括日期时间、称为PM2.5浓度的污染物、以及天气信息、包括露点（露点温度）、温度、压力、风向、风速和累积的降雪小时数。原始数据中的完整功能列表如下：

- 1.No：行号。
- 2.year：年。
- 3.month：月。
- 4.day：日。
- 5.hour：小时。
- 6.pm2.5：PM2.5浓度。
- 7.DEWP：露点温度。
- 8.TEMP：温度。
- 9.PRES：压力。
- 10.cbwd：风向。
- 11.lws：风速。
- 12.Is：积雪的时间
- 13.Ir：累积的下雨时数

我们可以使用这些数据并构建一个预测问题，基于天气条件和前几个小时的污染，我们预测在下一个小时的污染。你可以从UCI Machine Learning Repository下载数据集—北京PM2.5数据集 (<https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data>)。下载数据集并将其放在你当前的工作目录中，文件名为“raw.csv”。

基本数据准备

第一步，我们必须清洗数据。

以下是原始数据集的前几行。

```
No,year,month,day,hour,pm2.5,DEWP,TEMP,PRES,cbwd,lws,Is,Ir
1,2010,1,1,0,NA,-21,-11,1021,NW,1.79,0,0
2,2010,1,1,1,NA,-21,-12,1020,NW,4.92,0,0
3,2010,1,1,2,NA,-21,-11,1019,NW,6.71,0,0
4,2010,1,1,3,NA,-21,-14,1019,NW,9.84,0,0
5,2010,1,1,4,NA,-20,-12,1018,NW,12.97,0,0
```

第一步是将日期时间信息整合为一个日期时间，以便我们可以将其用作Pandas的索引。我们需要快速显示前24小时的pm2.5的NA值。因此，我们需要删除第一行数据。在数据集中还有几个分散的“NA”值；我们现在可以用0值标记它们。

以下脚本加载原始数据集，并将日期时间信息解析为Pandas Data Frame索引。“No”列被删除，然后为每列指定更清晰的名称。最后，将NA值替换为“0”值，并删除前24小时。

```
from pandas import read_csv
from datetime import datetime
# load data
def parse(x):
    return datetime.strptime(x, '%Y %m %d %H')
dataset = read_csv('raw.csv', parse_dates = [['year', 'month', 'day', 'hour']], index_col=0,
dataset.drop('No', axis=1, inplace=True)
# manually specify column names
dataset.columns = ['pollution', 'dew', 'temp', 'press', 'wnd_dir', 'wnd_spd', 'snow', 'rain']
dataset.index.name = 'date'
# mark all NA values with 0
dataset['pollution'].fillna(0, inplace=True)
# drop the first 24 hours
dataset = dataset[24:]
# summarize first 5 rows
print(dataset.head(5))
# save to file
dataset.to_csv('pollution.csv')
```

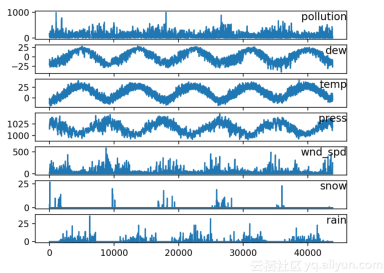
运行该示例打印转换的数据集的前5行，并将数据集保存到“pollution.csv”。

Date	pollution	dew	temp	press	wnd_dir	wnd_spd	snow	rain
2010-01-02 00:00:00	129.0	-16	-4.0	1020.0	SE	1.79	0	0
2010-01-02 01:00:00	148.0	-15	-4.0	1020.0	SE	2.68	0	0
2010-01-02 02:00:00	159.0	-11	-5.0	1021.0	SE	3.57	0	0
2010-01-02 03:00:00	181.0	-7	-5.0	1022.0	SE	5.36	1	0
2010-01-02 04:00:00	138.0	-7	-5.0	1022.0	SE	6.25	2	0

下面的代码加载了“pollution.csv”文件，并将每个系列作为单独的子图绘制，除了风速dir，这是分类的。

```
from pandas import read_csv
from matplotlib import pyplot
# load dataset
dataset = read_csv('pollution.csv', header=0, index_col=0)
values = dataset.values
# specify columns to plot
groups = [0, 1, 2, 3, 5, 6, 7]
i = 1
# plot each column
pyplot.figure()
for group in groups:
    pyplot.subplot(len(groups), 1, i)
    pyplot.plot(values[:, group])
    pyplot.title(dataset.columns[group], y=0.5, loc='right')
    i += 1
pyplot.show()
```

运行示例创建一个具有7个子图的程序，显示每个变量的5年数据。



云栖社区 yq.aliyun.com

多变量LSTM预测模型建立：

LSTM数据准备

第一步是为LSTM准备污染数据集。这涉及将数据集视为监督学习问题并对输入变量进行归一化处理。考虑到上一个时间段的污染测量和天气条件，我们将把监督学习问题作为预测当前时刻（t）的污染情况。根据过去24小时的天气情况和污染，预测下一个小时的污染，并给予下一个小时的“预期”天气条件。

我们可以使用在博客中开发的series_to_supervised（）函数来转换数据集：如何将时间序列转换为Python中的监督学习问题（<http://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/>）

首先，加载“pollution.csv”数据集。风速特征是标签编码（整数编码）。如果你有兴趣探索，也可以使用热编码。

接下来，所有功能都被规范化，然后将数据集转换为监督学习问题。然后删除要预测的小时的天气变量（t）。

完整的代码清单如下：

```
# convert series to supervised learning
def series_to_supervised(data, n_in=1, n_out=1, dropnan=True):
    n_vars = 1 if type(data) is list else data.shape[1]
    df = DataFrame(data)
    cols, names = list(), list()
    # input sequence (t-n, ... t-1)
    for i in range(n_in, 0, -1):
        cols.append(df.shift(i))
        names += [('var%d(t-%d)' % (j+1, i)) for j in range(n_vars)]
    # forecast sequence (t, t+1, ... t+n)
    for i in range(0, n_out):
        cols.append(df.shift(-i))
        if i == 0:
            names += [('var%d(t)' % (j+1)) for j in range(n_vars)]
        else:
            names += [('var%d(t+%d)' % (j+1, i)) for j in range(n_vars)]
    # put it all together
    agg = concat(cols, axis=1)
    agg.columns = names
    # drop rows with NaN values
    if dropnan:
        agg.dropna(inplace=True)
    return agg

# load dataset
dataset = read_csv('pollution.csv', header=0, index_col=0)
values = dataset.values
# integer encode direction
encoder = LabelEncoder()
values[:,4] = encoder.fit_transform(values[:,4])
# ensure all data is float
values = values.astype('float32')
# normalize features
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(values)
# frame as supervised learning
reframed = series_to_supervised(scaled, 1, 1)
# drop columns we don't want to predict
reframed.drop(reframed.columns[[9,10,11,12,13,14,15]], axis=1, inplace=True)
print(reframed.head())
```

运行示例打印转换后的数据集的前5行。我们可以看到8个输入变量（输入序列）和1个输出变量（当前小时的污染水平）。

	var1(t-1)	var2(t-1)	var3(t-1)	var4(t-1)	var5(t-1)	var6(t-1)	\
1	0.129779	0.352941	0.245902	0.527273	0.666667	0.002290	
2	0.148893	0.367647	0.245902	0.527273	0.666667	0.003811	
3	0.159960	0.426471	0.229508	0.545454	0.666667	0.005332	
4	0.182093	0.485294	0.229508	0.563637	0.666667	0.008391	
5	0.138833	0.485294	0.229508	0.563637	0.666667	0.009912	
	var7(t-1)	var8(t-1)	var1(t)				
1	0.000000	0.0	0.148893				
2	0.000000	0.0	0.159960				
3	0.000000	0.0	0.182093				
4	0.037037	0.0	0.138833				
5	0.074074	0.0	0.109658				

注意：我们必须提供超过一小时的输入时间步长。因为在解决序列预测问题时，LSTMs通过时间进行反向传播。

定义和拟合模型

在本节中，我们将通过多变量输入数据拟合LSTM。

首先，我们必须将准备好的数据集分成训练集和测试集。为了加快对这次示范模型的训练，我们将仅使用第一年的数据来拟合模型，然后对其余4年的数据进行评估。下面的示例将数据集分成训练集和测试集，然后将训练集和测试集分成输入和输出变量。最后，将输入（X）重构为LSTM预期的3D格式，即[样本，时间步长，特征]。

```
# split into train and test sets
values = reframed.values
n_train_hours = 365 * 24
train = values[:n_train_hours, :]
test = values[n_train_hours:, :]
# split into input and outputs
train_X, train_y = train[:, :-1], train[:, -1]
test_X, test_y = test[:, :-1], test[:, -1]
# reshape input to be 3D [samples, timesteps, features]
train_X = train_X.reshape((train_X.shape[0], 1, train_X.shape[1]))
test_X = test_X.reshape((test_X.shape[0], 1, test_X.shape[1]))
print(train_X.shape, train_y.shape, test_X.shape, test_y.shape)
```

运行此示例打印训练集的形状：

```
(8760, 1, 8) (8760,) (35039, 1, 8) (35039,)
```

现在我们可以定义和拟合我们的LSTM模型。

我们将在第一个隐层中定义具有50个神经元的LSTM和用于预测污染的输出层中的1个神经元。输入形状是1个时间步长，具有8个特征。

我们将在我们的模型中使用平均绝对误差（MAE）损失函数。该模型将拟合50个批量大小为72的训练时期。最后，我们通过在fit（）函数中设置validation_data参数来跟踪训练过程中的训练和测试失败。在运行结束时，绘制训练和测试损失。

```
# design network
model = Sequential()
model.add(LSTM(50, input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(Dense(1))
model.compile(loss='mae', optimizer='adam')
# fit network
history = model.fit(train_X, train_y, epochs=50, batch_size=72, validation_data=(test_X, test_y))
# plot history
pyplot.plot(history.history['loss'], label='train')
pyplot.plot(history.history['val_loss'], label='test')
pyplot.legend()
pyplot.show()
```

评估模型

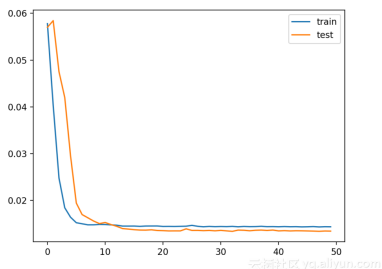
模型拟合后，我们可以预测整个测试数据集。

我们将预测的数据集与测试数据集相结合，并反演缩放。我们还可以用预期的污染数字来反演测试数据集的缩放。使用预测值和实际值，我们可以计算模型的误差分数。并且我们还可以计算出与变量本身相同的单位产生误差的均方根误差（RMSE）。

完整例子

完整的示例如下所示：源程序附件中下载！

运行示例首先创建一个绘图，显示训练中的训练损失和测试损失：



有趣的是，我们可以看到测试损失低于训练损失，该模型可能过度拟合训练数据。

训练损失和测试损失在每个训练时期结束时打印。在运行结束时，打印测试数据集上模型的最终RMSE。我们可以看到，该模型实现了可观的RMSE：3.836。

```
Epoch 46/50
0s - loss: 0.0143 - val_loss: 0.0133
Epoch 47/50
0s - loss: 0.0143 - val_loss: 0.0133
Epoch 48/50
0s - loss: 0.0144 - val_loss: 0.0133
Epoch 49/50
0s - loss: 0.0143 - val_loss: 0.0133
Epoch 50/50
0s - loss: 0.0144 - val_loss: 0.0133
Test RMSE: 3.836
```

进一步阅读

如果你想学到更多，请参考一下资源：

北京PM2.5数据集在UCI机器学习库中 (<https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data>)

克拉斯长期记忆模型的5步生命周期 (<http://machinelearningmastery.com/5-step-life-cycle-long-short-term-memory-models-keras/>)

使用Python中的长时间内存网络进行时间序列预测 (<http://machinelearningmastery.com/time-series-forecasting-long-short-term-memory-network-python/>)

用Python中的长时间内存网络进行多步时间序列预测 (<http://machinelearningmastery.com/multi-step-time-series-forecasting-long-short-term-memory-networks-python/>)

总结：

在本教程中，你发现了如何将LSTM适用于多变量时间序列预测问题。

具体来说，你学到了：

如何将原始数据集转换为可用于时间序列预测的内容。

如何准备数据并适应多变量时间序列预测问题的LSTM。

如何做出预测并将结果重新调整到原始单位。

本文由@阿里云云栖社区组织翻译。

文章原标题《Multivariate Time Series Forecasting with LSTMs in Keras》

作者：Jason Brownlee

作者个人博客地址：<http://machinelearningmastery.com/blog/>

译者：袁虎 审阅：主题曲哥哥

文章为简译，更为详细的内容，请查看原文 (<http://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/>)

附件下载：[new 1.zi...\[【方向】\].1503015691.zip \(/attachment/download/?id=1948\)](#)

▶ 版权声明：本文内容由互联网用户自发贡献，版权归作者所有，本社区不拥有所有权，也不承担相关法律责任。如果您发现本社区中有涉嫌抄袭的内容，欢迎发送邮件至：yqgroup@service.aliyun.com (<mailto:yqgroup@service.aliyun.com>) 进行举报，并提供相关证据，一经查实，本社区将立刻删除涉嫌侵权内容。



用云栖社区APP，舒服~

分享到:

上一篇: [【深度学习之美】LSTM长短记, 长序依赖可追忆 \(入门篇\)](#) 下一篇: [如何开发机器学习模型? \(机器学习\)](#)

(http://service.weibo.com/sh
title=%E5%9F%BA%E4%BA%

相关文章

- | | |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------|
| 基于Keras的LSTM多变量时间序列预测 (/articles/174435) | 教你搭建多变量时间序列预测模型LSTM（附代码、数据集） (/a... |
| 深度学习必备手册（上） (/articles/221660) | 深度学习基础之LSTM (/articles/292728) |
| Sequential与Model模型、keras基本结构... (/articles/569479) | 程序员技术进阶手册（二） (/articles/471537) |
| keras中最常用深度学习的API (/articles/228109) | 深度学习将会变革NLP中的中文分词 (/articles/176473) |
| 手把手：教你如何用深度学习模型预测加密货币价格 (/articles/5... | Keras编解码模型序列预测 (/articles/236381) |

网友评论



(/users/1048076181520534)

following (/users/1048076181520534)

2017-09-22 16:18:28

厉害, 正在学习LSTM, 谢谢分享

 0

 5

【方向】 (/users/1834978585139856)

2017-09-22 16:34:06

正在整理更多优质好文, 关注, 希望您能在社区获得更多知识!

 0



following (/users/1048076181520534)

2017-09-23 10:49:02

谢谢!

 0



1590116783094772 (/users/1590116783094772)

2018-01-24 16:39:08

为什么要删除前24小时的数据呢

 0



【方向】 (/users/1834978585139856)

2018-01-24 17:48:08

@1590116783094772 您可以点击查看原文跟作者联系, 至于技术上的事情小编也解释不清。

 0



1590116783094772 (/users/1590116783094772)


2018-01-25 15:05:30

@【方向】 懂了, 前24小时数据是空的~

 0



写下你的评论...



1565521098176831 (/users/1565521098176831)

2018-03-15 15:30:38

请问这是多层的还是一层的神经网络呢?

👍 0

💬 1

【方向】 (/users/1834978585139856)


2018-03-15 19:53:42

一个隐藏层

👍 0

💬

写下你的评论...



(/users/1719822198809476)

1719822198809476 (/users/1719822198809476)

2018-03-28 09:00:45

楼主有MATLAB环境的吗？最近参加一个比赛需要多变量的LSTM预测，但是我只熟悉MATLAB，如果有的话，麻烦您给我分享一份，邮箱dgf_dllg_yjs@sina.com。

跪谢了！

👍 0

💬 1

【方向】 (/users/1834978585139856)


2018-06-21 15:50:32

没有啊~~

👍 0

💬

写下你的评论...



(/users/1075368289373786)

1075368289373786 (/users/1075368289373786)

2018-05-23 10:46:38

楼主，您好，非常感谢您的代码，有个疑问，我把这个reframed = series_to_supervised(scaled, 1, 1)改成reframed = series_to_supervised(scaled, 2, 1)，即我想让前面的2个时间点预测后一个时间点，可是的出来的结果很差，有点不理解，为什么用前一个时间点预测后一个时间点的效果这么好，感觉是过拟合了。

👍 0

💬 1

【方向】 (/users/1834978585139856)


2018-06-21 15:57:38

呃~这是从国外一个大牛个人网站上，翻译过来的文章。你可以在本文末点击原文跟真楼主取得联系。这篇文章在他的个人网站上，讨论者也很多。你也可以在下面咨询一些问题。

👍 0

💬

写下你的评论...



(/users/1082224205281575)


1082224205281575 (/users/1082224205281575)

2018-06-18 20:31:41

啊啊啊，我也需要多变量的matlab版本额

👍 0

💬 0



(/users/1931793796270569)

1931793796270569 (/users/1931793796270569)

2018-06-21 15:25:25

楼主，在做预测的时候，如果将test = values[n_train_hours,:]的数据里面的pollution列全部置为0的话，预测的所有结果就会趋于0，我在想我们的训练数据不是train = values[:n_train_hours,:]吗？它和测试数据是没有关系的，我们测试数据应该只用到除去pollution列的其他列吧，那应该pollution的值是多少或者有没有值应该不会影响最后的预测结果呀！

👍 0

💬 5

【方向】 (/users/1834978585139856)

2018-06-21 15:57:27

呃~这是从国外一个大牛个人网站上，翻译过来的文章。你可以在本文末点击原文跟真楼主取得联系。这篇文章在他的个人网站上，讨论者也很多。你也可以在下面咨询一些问题。

👍 0

💬

1931793796270569 (/users/1931793796270569)

2018-06-21 16:34:10

@【方向】 好的 谢谢

👍 0

💬

1071029658286500 (/users/1071029658286500)

2018-06-22 17:15:28

其实在测试的时候用到了pollution列的数据

👍 1

💬

【方向】 (/users/1834978585139856)

2018-06-22 19:05:08

@1071029658286500 多谢！

👍 0

💬

1931793796270569 (/users/1931793796270569)

2018-06-25 14:56:23

@1071029658286500 预测的时候确实是用到了，感觉这种结果只能做自测（验证）吧，而在实际应用中事先是不知道预测结果的。想请教大佬有没有做过或者有没有思路通过多变量时间序列预测一个未知的结果（未来一段时间区间的）。。。

👍 0

💬

写下你的评论...

登录后可评论，请 登录 (<https://account.aliyun.com/login/login.htm?>)

- 热点导航
- [闲时流量包 \(https://promotion.aliyun.com/ntms/act/flowbagidle.html\)](https://promotion.aliyun.com/ntms/act/flowbagidle.html)[云计算 \(https://www.aliyun.com/\)](https://www.aliyun.com/)[网络安全 \(https://market.aliyun.com/security\)](https://market.aliyun.com/security)[互联网架构 \(https://www.aliyun.com/aliware\)](https://www.aliyun.com/aliware)[ECS升级配置 \(https://yq.aliyun.com/ask/53742\)](https://yq.aliyun.com/ask/53742)[物联网 \(https://www.aliyun.com/product/iot\)](https://www.aliyun.com/product/iot)[教程 \(https://edu.aliyun.com/jiaocheng\)](https://edu.aliyun.com/jiaocheng)[PHP \(https://yq.aliyun.com/php\)](https://yq.aliyun.com/php)
- 用户关注
- [自动化测试 \(https://bbs.aliyun.com/read/301499.html\)](https://bbs.aliyun.com/read/301499.html)[解决方案 \(https://www.aliyun.com/solution/all\)](https://www.aliyun.com/solution/all)[linux命令 \(https://yq.aliyun.com/articles/34777\)](https://yq.aliyun.com/articles/34777)[云服务 \(https://www.aliyun.com/\)](https://www.aliyun.com/)
- 更多推荐
- [JavaScript 函数 \(https://yq.aliyun.com/articles/92145\)](https://yq.aliyun.com/articles/92145)[服务器监控 \(https://yq.aliyun.com/articles/48786\)](https://yq.aliyun.com/articles/48786)[Python语言 \(https://yq.aliyun.com/roundtable/56407\)](https://yq.aliyun.com/roundtable/56407)[移动数据分析 \(https://www.aliyun.com/\)](https://www.aliyun.com/)[用户体验 \(https://yq.aliyun.com/articles/132294\)](https://yq.aliyun.com/articles/132294)[云数据库Rds \(https://help.aliyun.com/product/26090.html\)](https://help.aliyun.com/product/26090.html)[负载均衡 \(https://www.aliyun.com/product/slb/\)](https://www.aliyun.com/product/slb/)[域名注册 \(https://wanwang.aliyun.com/\)](https://wanwang.aliyun.com/)[Whois查询 \(https://whois.aliyun.com\)](https://whois.aliyun.com)[数据可视化 \(https://help.aliyun.com/product/43570.html\)](https://help.aliyun.com/product/43570.html)[ICP备案查询 \(https://beian.aliyun.com\)](https://beian.aliyun.com)[主题地图 \(https://yq.aliyun.com/zt\)](https://yq.aliyun.com/zt)[阿里云大学 \(https://ed.cn域名 \(https://wanwang.aliyun.com/domain/cn/\)](https://ed.cn域名 (https://wanwang.aliyun.com/domain/cn/)[Js \(https://yq.aliyun.com/jsarticle\)](https://yq.aliyun.com/jsarticle)[Mysql \(https://yq.aliyun.com/sqlarticle\)](https://yq.aliyun.com/sqlarticle)[移动站 \(https://m.aliyun.com/yunqi/\)](https://m.aliyun.com/yunqi/)[IT论坛 \(https://bbs.aliyun.com/\)](https://bbs.aliyun.com/)[企业邮箱 \(https://mail.aliyun.com/\)](https://mail.aliyun.com/)[签名文件 \(https://www.aliyun.com/jiaocheng/1075.html\)](https://www.aliyun.com/jiaocheng/1075.html)

关于我们 (/www.aliyun.com/about) 法律声明及隐私权政策 (http://terms.aliyun.com/legal-agreement/terms/suit_bu1_ali_cloud/suit_bu1_ali_cloud201710161525_98396.html) 廉正举报 (https://jubao.alibaba.com/index.html?site=ALIYUN)

(/www.aliyun.com/contact) 加入阿里云 (/www.aliyun.com/careers)

阿里巴巴集团 (<http://www.alibabagroup.com/cn/global/home>) 淘宝网 (<http://www.taobao.com/>) 天猫 (<http://www.tmall.com/>) 聚划算 (<http://ju.taobao.com/>) 全球速卖通 (<http://www.aliexpress.com/>)

阿里巴巴国际交易市场 (<http://www.alibaba.com/>) 1688 (<http://www.1688.com/>) 阿里妈妈 (<http://www.alimama.com/index.htm>) 飞猪 (<http://www.fliggy.com>) 阿里云计算 (<http://www.aliyun.com/>)

AliOS (<http://www.aliyun.com/>) 阿里通信 (<http://aliqin.tmall.com/>) 万网 (<http://wanwang.aliyun.com/>) 高德 (<http://www.autonavi.com/>) UC (<http://www.uc.cn/>) 友盟 (<http://www.umeng.com/>)

虾米 (<http://www.xiami.com/>) 阿里星球 (<http://www.alibabaplanet.com>) 钉钉 (<http://www.dingtalk.com/?ifrom=20150205111943449>) 支付宝 (<http://www.alipay.com/>)

© 2009-2018 Aliyun.com 版权所有 ICP证：浙B2-20080101

 (<http://idinfo.zjaic.gov.cn/bscx.do?method=lzxx&id=3301963301080000025024>)

 浙公网安备 33010002000099号(<http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=33010002000099>)