

CSDN新首页上线啦，邀请你来立即体验！(http://feed.csdn.net/)

立即体验

CSDN

博客 (http://feed.csdn.net/?ref=toolbar)学院 (http://edu.csdn.net/?ref=toolbar)

下载 (http://download.csdn.net/?ref=toolbar)更多 ▾

登录 (http://passport.csdn.net/account/login?ref=toolbar)注册 (http://passport.csdn.net/account/mobileregister?ref=toolbar&action=mobileRegister)

深度学习Keras 库 跑例子

原创

2015年11月10日 21:37:55

18469

跑lmbd\_lstm.py 因为需要用lstm，所以就先跑 lstm例子，

1、官网下载后，直接运行lmbd\_lstm.py。总是提示无法下载，打开程序有看到，通过load\_data来下载数据，但是这个数据没法在线下载，导致跑不通。

```
print("Loading data...")
(X_train, y_train), (X_test, y_test) = imdb.load_data(nb_words=max_features, test_split=0.2)
```

解决：在lmbd.py 中路径更改一下。如下边所示，直接给路径。

```
# path = get_file(path, origin="https://s3.amazonaws.com/text-datasets/imdb.pkl")
path = "E:\\project\\deep learning\\RNN\\leeg rnn\\theano code\\keras-master\\imdb.pkl"
```

再次运行，即可跑通

二、下边是官网 例子说明，讲的很清楚，看完这个才真正发现，这个库确实 很好用，很简单啊。就是速度有点慢。

官网地址：<http://keras.io/examples/>

Here are a few examples to get you started!

Multilayer Perceptron (MLP):

```
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation
from keras.optimizers import SGD

model = Sequential()
# Dense(64) is a fully-connected layer with 64 hidden units.
# in the first layer, you must specify the expected input data shape:
# here, 20-dimensional vectors.
model.add(Dense(64, input_dim=20, init='uniform')) // 全连接层， 64个神经元
model.add(Activation('tanh'))
model.add(Dropout(0.5))
model.add(Dense(64, init='uniform'))
model.add(Activation('tanh'))
model.add(Dropout(0.5))
model.add(Dense(2, init='uniform'))
model.add(Activation('softmax')) // 最后一个全连接层用 softmax当激活函数

sgd = SGD(lr=0.1, decay=1e-6, momentum=0.9, nesterov=True) // 用随机梯度下降优化, nesterov????
model.compile(loss='mean_squared_error', optimizer=sgd)

model.fit(X_train, y_train, nb_epoch=20, batch_size=16)
score = model.evaluate(X_test, y_test, batch_size=16)
```

Alternative implementation of MLP:

deep\_lerner (http://b...

+关注

(http://blog.csdn.net/u014114990)

原创

粉丝

喜欢

未开通

149

204

1

(https://gite

他的最新文章

更多文章 (http://blog.csdn.net/u014114990)

LeNet、AlexNet、GoogLeNet、VGG、ResNetInception-ResNet-v2、FractalNet、DenseNet (http://blog.csdn.net/u014114990/article/details/53905649)

SqueezeNet模型参数降低50倍，压缩461倍 (http://blog.csdn.net/u014114990/article/details/53905600)

caffe 里面GPU 选择设置，通过配置文件 读取GPU 的编号。 (http://blog.csdn.net/u014114990/article/details/52776990)

学校平面设计

在线课程

SDCC 2017 智能制造技术实训线上峰会 (http://edu.csdn.net)

MVVM在美团点评漫游移动端的最佳实践 (http://edu.csdn.net/huiyiCourse/detail/603?utm\_source=blog9)

```

model = Sequential()
model.add(Dense(64, input_dim=20, init='uniform', activation='tanh'))
model.add(Dropout(0.5))
model.add(Dense(64, init='uniform', activation='tanh'))
model.add(Dropout(0.5))
model.add(Dense(2, init='uniform', activation='softmax'))

sgd = SGD(lr=0.1, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='mean_squared_error', optimizer=sgd)

```

### VGG-like convnet:

```

from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation, Flatten
from keras.layers.convolutional import Convolution2D, MaxPooling2D
from keras.optimizers import SGD

model = Sequential()
# input: 100x100 images with 3 channels -> (3, 100, 100) tensors.
# this applies 32 convolution filters of size 3x3 each.
model.add(Convolution2D(32, 3, 3, border_mode='full', input_shape=(3, 100, 100)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='valid'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
# Note: Keras does automatic shape inference.
model.add(Dense(256))
model.add(Activation('relu'))
model.add(Dropout(0.5))

model.add(Dense(10))
model.add(Activation('softmax'))

sgd = SGD(lr=0.1, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)

model.fit(X_train, Y_train, batch_size=32, nb_epoch=1)

```

### Sequence classification with LSTM:

```

from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation
from keras.layers.embeddings import Embedding
from keras.layers.recurrent import LSTM

model = Sequential()
model.add(Embedding(max_features, 256, input_length=maxlen))
model.add(LSTM(output_dim=128, activation='sigmoid', inner_activation='hard_sigmoid'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='rmsprop')

model.fit(X_train, Y_train, batch_size=16, nb_epoch=10)
score = model.evaluate(X_test, Y_test, batch_size=16)

```

### Architecture for learning image captions with a convnet and a Gated Recurrent Unit:

(word-level embedding, caption of maximum length 16 words).

Note that getting this to work well will require using a bigger convnet, initialized with pre-trained weights.



(<http://edu.csdn.net>)

C语言大型软件设计的面向对象  
[http://edu.csdn.net/huiyiCourse/detail/594?utm\\_source=blog9](http://edu.csdn.net/huiyiCourse/detail/594?utm_source=blog9)

### 热门文章

深度学习Keras 库 跑例子 (<http://blog.csdn.net/u014114990/article/details/4943939>)

18455

多通道(比如RGB三通道)卷积过程 (<http://blog.csdn.net/u014114990/article/details/4943939>)

17304

Deep Residual Learning for Image Recognition 笔记 (<http://blog.csdn.net/u014114990/article/details/4943939>)

16376

caffe Sigmoid cross entropy loss 交叉熵损失函数 (<http://blog.csdn.net/u014114990/article/details/4943939>)

13767

matlab自带各种分类器的使用示例 (<http://blog.csdn.net/u014114990/article/details/4943939>)

10585

```

max_caption_len = 16
vocab_size = 10000

# first, let's define an image model that
# will encode pictures into 128-dimensional vectors.
# it should be initialized with pre-trained weights.
image_model = Sequential()
image_model.add(Convolution2D(32, 3, 3, border_mode='full', input_shape=(3, 100, 100)))
image_model.add(Activation('relu'))
image_model.add(Convolution2D(32, 3, 3))
image_model.add(Activation('relu'))
image_model.add(MaxPooling2D(pool_size=(2, 2)))

image_model.add(Convolution2D(64, 3, 3, border_mode='full'))
image_model.add(Activation('relu'))
image_model.add(Convolution2D(64, 3, 3))
image_model.add(Activation('relu'))
image_model.add(MaxPooling2D(pool_size=(2, 2)))

image_model.add(Flatten())
image_model.add(Dense(128))

# let's load the weights from a save file.
image_model.load_weights('weight_file.h5')

# next, let's define a RNN model that encodes sequences of words
# into sequences of 128-dimensional word vectors.
language_model = Sequential()
language_model.add(Embedding(vocab_size, 256, input_length=max_caption_len))
language_model.add(GRU(output_dim=128, return_sequences=True))
language_model.add(Dense(128))

# let's repeat the image vector to turn it into a sequence.
image_model.add(RepeatVector(max_caption_len))

# the output of both models will be tensors of shape (samples, max_caption_len, 128).
# let's concatenate these 2 vector sequences.
model = Sequential()
model.add(Merge([image_model, language_model], mode='concat', concat_axis=-1))
# let's encode this vector sequence into a single vector
model.add(GRU(256, 256, return_sequences=False))
# which will be used to compute a probability
# distribution over what the next word in the caption should be!
model.add(Dense(vocab_size))
model.add(Activation('softmax'))

model.compile(loss='categorical_crossentropy', optimizer='rmsprop')

# "images" is a numpy float array of shape (nb_samples, nb_channels=3, width, height).
# "captions" is a numpy integer array of shape (nb_samples, max_caption_len)
# containing word index sequences representing partial captions.
# "next_words" is a numpy float array of shape (nb_samples, vocab_size)
# containing a categorical encoding (0s and 1s) of the next word in the corresponding
# partial caption.
model.fit([images, partial_captions], next_words, batch_size=16, nb_epoch=100)

```

In the examples folder, you will find example models for real datasets: - CIFAR10 small images classification: Convolutional Neural Network (CNN) with realtime data augmentation - IMDB movie review sentiment classification: LSTM over sequences of words - Reuters newswires topic classification: Multilayer Perceptron (MLP) - MNIST handwritten digits classification: MLP & CNN - Character-level text generation with LSTM

...and more.

三、看到网上一篇博客注释Istm，这个可能是老版本上注释的，但是参数还是有可借鉴的地方  
参考博客地址：<http://www.jianshu.com/p/3992fe7bb847>





## Keras Recurrent Layers 解析

字数1179 阅读120 评论0 喜欢1

## GRU

```
keras.layers.recurrent.GRU(input_dim, output_dim=128, init='glorot_uniform', inner_init='orthogonal',
activation='sigmoid', inner_activation='hard_sigmoid', weights=None, truncate_gradient=-1, return_sequences=False)
```


Gated Recurrent Unit - Cho et al. 2014.

-  输入形状：3D 张量：(nb\_samples, timesteps, input\_dim)。
-  输出形状：
  -  如果 return\_sequences：3D 张量形如：(nb\_samples, timesteps, output\_dim)。
  -  否则：2D 张量形如：(nb\_samples, output\_dim)。
- 参数：
  - input\_dim：输入的维数
  - output\_dim：内部投影的维数和最终输出的维数
  - init：权重初始函数。可以是任何已经存在的函数（str），或者是一个 Theano 的函数（参见：初始化 (<http://keras.io/initializations/>)）
  - inner\_init：对内部元件的权重初始化函数
  - activation：激活函数。可以是任何已经存在的函数（str），或者是一个 Theano 的函数（参见：初始化 (<http://keras.io/activations/>)）
  - inner\_activation：内部元件的激活函数
  - weights：numpy 数组的列表用以设置初始权重。这个列表应该有 9 个元素
  - truncate\_gradient：BPTT 的截断时间步。参见：Theano scan (<http://deeplearning.net/software/theano/library/scan.html>)
  - return\_sequences：Boolean。是否返回输出序列的最后一个，或者返回全部序列。
- References:
  - On the Properties of Neural Machine Translation: Encoder–Decoder Approaches (<http://www.aclweb.org/anthology/W14-4012>)
  - Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling (<http://arxiv.org/pdf/1412.3555v1.pdf>)

## LSTM

```
keras.layers.recurrent.LSTM(input_dim, output_dim=128, init='glorot_uniform', inner_init='orthogonal',
forget_bias_init='one', activation='tanh', inner_activation='hard_sigmoid', weights=None, truncate_gradient=-1, return_sequences=False)
```

Long Short-Term Memory unit - Hochreiter et al. 1997

-  输入形状：3D 张量：(nb\_samples, timesteps, input\_dim)。

- 输出形状：

- 如果 `return_sequences` : 3D 张量形如：(nb\_samples, timesteps, output\_dim)。
- 否则：2D 张量形如：(nb\_samples, output\_dim)。

- 参数：

4



- `input_dim`：输入的维数
- `output_dim`：内部投影的维数和最终输出的维数
- `init`：权重初始函数。可以是任何已经存在的函数（str），或者是一个 Theano 的函数（参见：初始化 (<http://keras.io/initializations/>)）
- `inner_init`：对内部元件的权重初始化函数
- `forget_bias_init`：用作遗忘门的偏差初始函数。Jozefowicz 等人推荐使用 1 来初始化
- `activation`：激活函数。可以是任何已经存在的函数（str），或者是一个 Theano 的函数（参见：初始化 (<http://keras.io/activations/>)）
- `inner_activation`：内部元件的激活函数
- `weights`：numpy 数组的列表用以设置初始权重。这个列表应该有 9 个元素
- `truncate_gradient`：BPTT 的截断时间步。参见：Theano scan (<http://deeplearning.net/software/theano/library/scan.html>)
- `return_sequences`：Boolean。是否返回输出序列的最后一个，或者返回全部序列。

- References:

- Long short-term memory ([http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97\\_lstm.pdf](http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97_lstm.pdf))
- Learning to forget: Continual prediction with LSTM (<http://www.mitpressjournals.org/doi/pdf/10.1162/089976600300015015>)
- Supervised sequence labelling with recurrent neural networks (<http://www.cs.toronto.edu/~graves/preprint.pdf>)

## JZS1, JZS2, JZS3

```
keras.layers.recurrent.JZS1(input_dim, output_dim=128, init='glorot_uniform', inner_init='orthogonal',
activation='tanh', inner_activation='sigmoid', weights=None, truncate_gradient=-1, return_sequences=False)
```

全连接的 RNN 其中输出被重回输入。不是特别有用，仅供参考。

- 输入形状：3D 张量：(nb\_samples, timesteps, input\_dim)。

- 输出形状：

- 如果 `return_sequences` : 3D 张量形如：(nb\_samples, timesteps, output\_dim)。
- 否则：2D 张量形如：(nb\_samples, output\_dim)。

- 参数：

- input\_dim
  - output\_dim
  - init：权重初始函数。可以是任何已经存在的函数（str），或者是一个 Theano 的函数（参见：初始化 (<http://keras.io/initializations/>)）
  - inner\_init：内部元件的初始化的函数
  - activation：激活函数。可以是任何已经存在的函数（str），或者是一个 Theano 的函数（参见：初始化 (<http://keras.io/activations/>)）
  - weights：numpy 数组的列表用以设置初始权重。这个列表应该有 3 个元素，形如：  
[(input\_dim, output\_dim), (output\_dim, output\_dim), (output\_dim, output\_dim)]
  - truncate\_gradient：BPTT 的截断时间步。参见：Theano scan (<http://deeplearning.net/software/theano/library/scan.html>)
  - return\_sequences：Boolean。是否返回输出序列的最后一个，或者返回全部序列。
- 参考文献：
- An Empirical Exploration of Recurrent Network Architectures (<http://www.jmlr.org/proceedings/papers/v37/jozefowicz15.pdf>)

## SimpleDeepRNN

```
keras.layers.recurrent.SimpleDeepRNN(input_dim, output_dim, depth=3, init='glorot_uniform', inner_init='orthogonal', activation='sigmoid', inner_activation='hard_sigmoid', weights=None, truncate_gradient=-1, return_sequences=False)
```

全连接的 RNN 其中多个时间步的输出重回输入中（使用 depth 参数来控制步数）。

```
output = activation( W.x_t + b + inner_activation(U_1.h_tm1) + inner_activation(U_2.h_tm2) + ... )
```

也不是常用的模型，仅供参考。

- 输入形状：3D 张量：(nb\_samples, timesteps, input\_dim)。
- 输出形状：
  - 如果 return\_sequences：3D 张量形如：(nb\_samples, timesteps, output\_dim)。
  - 否则：2D 张量形如：(nb\_samples, output\_dim)。
- 参数：
  - input\_dim
  - output\_dim
  - init：权重初始函数。可以是任何已经存在的函数（str），或者是一个 Theano 的函数（参见：初始化 (<http://keras.io/initializations/>)）
  - inner\_init：对内部元件的权重初始化函数
  - activation：激活函数。可以是任何已经存在的函数（str），或者是一个 Theano 的函数（参见：初始化 (<http://keras.io/activations/>)）

- inner\_activation : 内部元件的激活函数
- weights : numpy 数组的列表用以设置初始权重。这个列表应该有 3 个元素, 形如 :  

```
[(input_dim, output_dim), (output_dim, output_dim), (output_dim, )]
```
- truncate\_gradient : BPTT 的截断时间步。参见 : Theano scan (<http://deeplearning.net/software/theano/library/scan.html>)
- return\_sequences : Boolean。是否返回输出序列的最后一个, 或者返回全部序列。



4



版权声明 本文为博主原创文章, 未经博主允许不得转载。



wwwttnnnyy (wwwttnnnyy) 2016-10-11 17:15

1楼

运行例子的时候出错: model.fit(X\_train, y\_train, nb\_epoch=20, batch\_size=16)  
NameError: name 'X\_train' is not defined,请问这是什么问题?谢谢

回复

## 相关文章推荐

### 深度学习 (十) keras学习笔记 (<http://blog.csdn.net/hjimce/article/details/49095199>)

keras与torch7的使用非常相似, 是最近才火起来的深度学习开源库, 底层是用了theano。keras可以说是python版的torch7, 对于快速构建CNN模型非常方便。同时也包含了一些最新文献...



hjimce (<http://blog.csdn.net/hjimce>) 2015年10月13日 10:38 28410

### python keras (一个超好用的神经网络框架) 的使用以及实例 ([http://blog.csdn.net/Star\\_Bob](http://blog.csdn.net/Star_Bob))

先吐槽一下这个基于theano的keras有多难装, 反正我是在windows下折腾到不行, 所以自己装了一个双系统。这才感到linux系统的强大之初, 难怪大公司都是用这个做开发, 妹的, 谁用谁知道啊!!! ...



Star\_Bob ([http://blog.csdn.net/Star\\_Bob](http://blog.csdn.net/Star_Bob)) 2015年09月20日 17:18 57997




### 11.11 程序员该买些什么犒劳自己?

这是一份11.11程序员版爆款剁手清单! 万万没想到, 他们买了这些...

([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjDLnjT0IZ0qnfK9ujYzP1nsrjDz0Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1Y1uhNbhuhuW-WrHI9nWTv0AwY5HDdnH0vnWRzPHc0lgF\\_5y9YIZ0IQzq-uZR8mLPbUB48ugfEXyN9T-KzUvdEIA-EUBqbugw9pysEn1qdIAdxTvqdThP-5yF\\_UvTkn0KzujY4rHb0mhYqn0KsTWYs0ZNGUjYkPHTYn1mk0AqGuJYknWb3rjDY0APGuJYLnWm4n1c0ULI85H00TZbqnW0v0APzm1YdrjTsn6](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjDLnjT0IZ0qnfK9ujYzP1nsrjDz0Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1Y1uhNbhuhuW-WrHI9nWTv0AwY5HDdnH0vnWRzPHc0lgF_5y9YIZ0IQzq-uZR8mLPbUB48ugfEXyN9T-KzUvdEIA-EUBqbugw9pysEn1qdIAdxTvqdThP-5yF_UvTkn0KzujY4rHb0mhYqn0KsTWYs0ZNGUjYkPHTYn1mk0AqGuJYknWb3rjDY0APGuJYLnWm4n1c0ULI85H00TZbqnW0v0APzm1YdrjTsn6))


### keras的例子 (入手) ([http://blog.csdn.net/m0\\_37407756/article/details/78117498](http://blog.csdn.net/m0_37407756/article/details/78117498))

keras的例子（入手）： 1 基于多层感知器的 softmax 多分类: from keras.models import Sequential from keras.layers imp  
O...

 m0\_37407756 ([http://blog.csdn.net/m0\\_37407756](http://blog.csdn.net/m0_37407756)) 2017年09月27日 21:31 119


## Keras上实现简单线性回归模型 (<http://blog.csdn.net/marsjhao/article/details/67042392>)

神经网络可以用来模拟回归问题 (regression), 实质上是单输入单输出神经网络模型, 例如给下面一组数据, 用一条线来对数据进行拟合, 并可以预测新输入 x 的输出值。一、详细解读我们通过这...

 marsjhao (<http://blog.csdn.net/marsjhao>) 2017年03月27日 22:31 2639

## 深度学习框架Keras安装 (<http://blog.csdn.net/u012556077/article/details/50364640>)

之前本打算安装Caffe, 但是依赖包太多了, 大大小小的问题真是头疼, 也是因为机器的原因, 本来想在自己笔记本上就在实体机上安装Ubuntu然后装caffe等深度学习框架, 但是固态硬盘最近出了问题, 拿去返...

 u012556077 (<http://blog.csdn.net/u012556077>) 2015年12月20日 16:22 26709




### 人人都能看懂的 AI 入门课

本课程将讲述人工智能的现状、应用场景和入门方法, 并通过运用 TensorFlow, 使得受众能清晰了解人工智能的运作方式。

([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjcvPjm0lZ0qnfK9ujYzP1ndnHfz0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1YsrjbdrAF-nhmkuyRznHKB0AwY5HDdnH0vnWRzPHn0lgF\\_5y9YIZ0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq\\_lyd8Q1R4uWI-n16kPWKWrrHnvnHRvnnNBuYD4PHqdlAdxTvqdThP-5HDknHK4mhkEusKzujY4rHb0mhYqn0KstWYs0ZNGUjYkPHTYn1mk0AqGujYkn10snjf10APGujYLnWm4n1c0ULI85H00TZbqnW0v0APzm1YkP1bvns](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjcvPjm0lZ0qnfK9ujYzP1ndnHfz0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1YsrjbdrAF-nhmkuyRznHKB0AwY5HDdnH0vnWRzPHn0lgF_5y9YIZ0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq_lyd8Q1R4uWI-n16kPWKWrrHnvnHRvnnNBuYD4PHqdlAdxTvqdThP-5HDknHK4mhkEusKzujY4rHb0mhYqn0KstWYs0ZNGUjYkPHTYn1mk0AqGujYkn10snjf10APGujYLnWm4n1c0ULI85H00TZbqnW0v0APzm1YkP1bvns))


## 基于 Theano 的深度学习 (Deep Learning) 框架 Keras 学习随笔 -02-Example (<http://blog.cs...>)

基于 Theano 的深度学习 (Deep Learning) 框架 Keras 学习随笔 -02-Example -- 下面来看几个例子, 来了解一下 Keras 的便捷之处。不需要具体去研究代码的意思, 只需要看一下这...

 niuwei22007 (<http://blog.csdn.net/niuwei22007>) 2015年10月11日 21:47 14975

## Keras cnn example(总结自莫凡) (<http://blog.csdn.net/taree/article/details/66973866>)

"" To know more or get code samples, please visit my website: <https://morvanzhou.github.io/tutorial...>

 taree (<http://blog.csdn.net/taree>) 2017年03月27日 17:24 943

## 关于 Keras 2.0 版本运行 demo 出错的问题 (<http://blog.csdn.net/JohinieLi/article/details/6...>)

关于 Keras 2.0 版本运行 demo 出错的问题由于是神经网络小白, 在运行 demo 的时候不懂 keras 的版本问题, 出现了一下的警告: 1、UserWarning: Update your `Conv2D...

 JohinieLi (<http://blog.csdn.net/JohinieLi>) 2017年04月05日 09:42 3012





## Delphi7 高级应用开发随书源码 (<http://download.csdn.net/detail/chexh...>)

(<http://download.csdn.net/detail/chexh...>) 2003年04月30日 00:00 676KB 下载

## [5]深度学习和 Keras----一个图像识别的简单 Demo (<http://blog.csdn.net/chancein007/arti...>)





在第三篇文章中。笔者给出了如何用Keras的神经网络训练一个线性函数，在这篇文章，笔者给出如何用Keras的神经网络，对mnist图库中的60000张手写的1-9的数字进行识别的一个例子。本代码是参考...

 chancein007 (<http://blog.csdn.net/chancein007>) 2017年06月04日 22:48  2374



在 keras 上实践,通过 keras例子来理解 lstm循环神经网络 ([/ma416539432/article/details/...](http://blog.csdn.net/ma416539432/article/details/497...))

本文是对这篇博文的翻译和实践: <http://machinelearningmastery.com/understanding-stateful-lstm-recurrent-neural-netw...>

 ma416539432 (<http://blog.csdn.net/ma416539432>) 2016-12-07 19:05  7087

LSTM与GRU的一些比较--论文笔记 ([/meanme/article/details/48845793](http://blog.csdn.net/meanme/article/details/48845793))

reference: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling1.概要: 传统的RNN在训练...

 meanme (<http://blog.csdn.net/meanme>) 2015-10-01 13:29  32140



基于Theano的深度学习(Deep Learning)框架Keras学习随笔-14-递归层 ([/niuwei22007/ar...](http://blog.csdn.net/niuwei22007/article/details/48845793))

基于Theano的深度学习(Deep Learning)框架Keras学习随笔-14-递归层 recurrent neural network 上一篇介绍了卷基层，可以用来构建很常见的卷积神经网络等模...

 niuwei22007 (<http://blog.csdn.net/niuwei22007>) 2015-10-24 00:39  11304



Keras 时序模型 ([/thinking\\_boy1992/article/details/53207177](http://blog.csdn.net/thinking_boy1992/article/details/53207177))

本文翻译自时序模型就是层次的线性叠加。你能够通过向构造函数传递层实例的列表构建序列模型: from keras.models import Sequential from keras.layer...

 Thinking\_boy1992 ([http://blog.csdn.net/Thinking\\_boy1992](http://blog.csdn.net/Thinking_boy1992)) 2016-11-17 21:18  1976



详细解读简单的lstm的实例 ([/zjm750617105/article/details/51321889](http://blog.csdn.net/zjm750617105/article/details/51321889))

本文是初学keras这两天来，自己仿照addition\_rnn.py，写的一个实例，数据处理稍微有些不同，但是准确性相比addition\_rnn.py 差一点，下面直接贴代码，解释和注释都在代码里...

 zjm750617105 (<http://blog.csdn.net/zjm750617105>) 2016-05-05 15:31  11452



DeepLearning tutorial (6) 易用的深度学习框架Keras简介 ([/zhoubi668/article/details/4...](http://blog.csdn.net/zhoubi668/article/details/497...))

之前我一直在使用Theano，前面五篇Deep learning相关的文章也是学习Theano的一些笔记，当时已经觉得Theano用起来略显麻烦，有时想实现一个新的结构，就要花很多时间去编程，所以想过将...

 zhoubi668 (<http://blog.csdn.net/zhoubi668>) 2015-05-07 13:40  25848


keras的一些例子理解 ([/b\\_c\\_wang/article/details/74885654](http://blog.csdn.net/B_C_Wang/article/details/74885654))

keras的一些例子理解来自我的github页面: <https://github.com/B-C-WANG/AI.Learning/tree/master/AI.Learning.Notes.III...>

 B\_C\_Wang ([http://blog.csdn.net/B\\_C\\_Wang](http://blog.csdn.net/B_C_Wang)) 2017-07-09 15:00  645




## python keras （一个超好用的神经网络框架）的使用以及实例 (</dlaicxf/article/details/5...>)

先吐槽一下这个基于theano的keras有多难装，反正我是在windows下折腾到不行，所以自己装了一个双系统。这才感到linux系统的强大之初，难怪大公司都是用这个做开发，妹的，谁用谁知道啊！！...

 dlaicxf (<http://blog.csdn.net/dlaicxf>) 2016-10-18 11:05  1222





## 我的keras小例子及问题 (</wd1603926823/article/details/52223373>)

  wd1603926823 (<http://blog.csdn.net/wd1603926823>) 2016-08-16 17:41  15207

## DeepLearning tutorial（6）易用的深度学习框架Keras简介 (</u012162613/article/details/...>)

之前我一直在使用Theano，前面五篇Deeplearning相关的文章也是学习Theano的一些笔记，当时已经觉得Theano用起来略显麻烦，有时想实现一个新的结构，就要花很多时间去编程，所以想过将...

 u012162613 (<http://blog.csdn.net/u012162613>) 2015-04-30 17:03  45143