

超快速！10分钟入门Keras指南



章华燕 北京交通大学 计算机硕士

116 人赞同了该文章

码字不易，欢迎给个赞！

欢迎交流与转载，文章会同步发布在公众号：机器学习算法全栈工程师(Jeemy110)

作者：李中梁 @ZL LI

Keras框架介绍

在用了一段时间的Keras后感觉真的很爽，所以特意祭出此文与我们公众号的粉丝分享。Keras是一个非常方便深度学习框架，它以TensorFlow或Theano为后端。用它可以快速地搭建深度网络，灵活地选取训练参数来进行网路训练。总之就是：灵活+快速！！

安装Keras

首先你需要有一个Python开发环境，直接点就用Anaconda，然后在CMD命令行中安装：

```
# GPU 版本
>>> pip install --upgrade tensorflow-gpu

# CPU 版本
>>> pip install --upgrade tensorflow

# Keras 安装
>>> pip install keras -U --pre
```

第一个例子：手写数字识别

首先我们在Keras中定义一个单层全连接网络，进行线性回归模型的训练：

```
# -*- coding: utf-8 -*-
# Regressor example
import numpy as np
np.random.seed(1337)
from keras.models import Sequential
from keras.layers import Dense
import matplotlib.pyplot as plt

# 创建数据集
X = np.linspace(-1, 1, 200)
np.random.shuffle(X) # 将数据集随机化
Y = 0.5 * X + 2 + np.random.normal(0, 0.05, (200,))
# 绘制数据集plt.scatter(X, Y)
plt.show()
X_train, Y_train = X[:160], Y[:160] # 把前160个
X_test, Y_test = X[160:], Y[160:] # 把后40个点

# 定义一个model,
model = Sequential() # Keras有两种类型的模型，序贯模型(Sequential)和函数式模型
# 比较常用的是Sequential，它是单输入单输出的
```

登录即可查看 **超5亿** 专业优质内容

超 5 千万创作者的优质提问、专业回答、深度文章和精彩视频尽在知乎。

立即登录/注册

第二层无需指定输入，一般第二层把第一层#

```

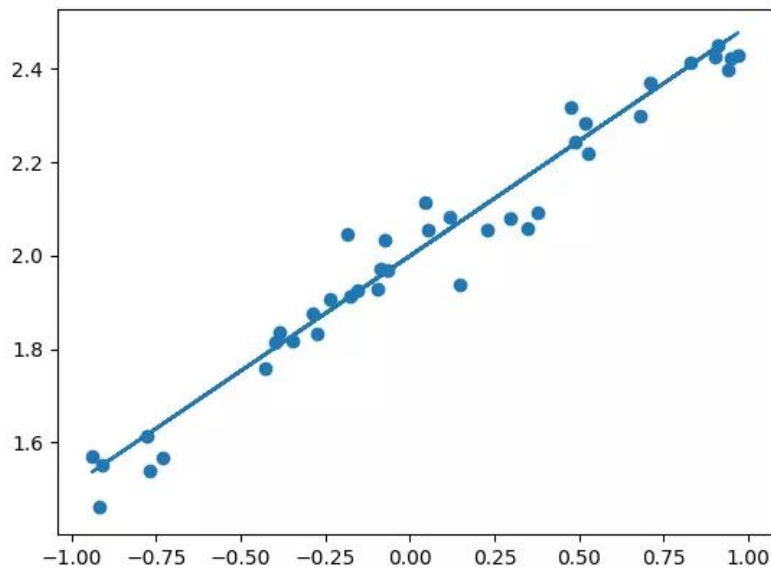
# 定义完模型就需要训练了，不过训练之前我们需要指定一些训练参数
# 通过compile()方法选择损失函数和优化器
# 这里我们用均方误差作为损失函数，随机梯度下降作为优化方法
model.compile(loss='mse', optimizer='sgd')
# 开始训练
print('Training -----')
for step in range(301):
    cost = model.train_on_batch(X_train, Y_train) # Keras有很多开始训练的函数，这里
    if step % 100 == 0:
        print('train cost: ', cost)
# 测试训练好的模型
print('\nTesting -----')
cost = model.evaluate(X_test, Y_test, batch_size=40)
print('test cost:', cost)
W, b = model.layers[0].get_weights() # 查看训练出的网络参数
# 由于我们网络只有一层，且每次训练的输入只有一
# 因此第一层训练出Y=WX+B这个模型，其中W, b为i

print('Weights=', W, '\nbiases=', b)
# plotting the prediction
Y_pred = model.predict(X_test)
plt.scatter(X_test, Y_test)
plt.plot(X_test, Y_pred)
plt.show()

```

训练结果：

最终的测试cost为：0.00313670327887，可视化结果如下图：



第二个例子：手写数字识别

MNIST数据集可以说是在业内被搞过次数最多的数据集了，毕竟这里我们也简单说一下在Keras下如何训练这个数据集：

```

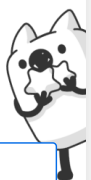
# *_ coding: utf-8 *_
# mnist
import numpy as np
np.random.seed(1337)
from keras.datasets import mnist

```

登录即可查看 **超5亿** 专业优质内容

超 5 千万创作者的优质提问、专业回答、深度文章和精彩视频尽在知乎。

立即登录/注册



```

from keras.layers import Dense, Activation
from keras.optimizers import RMSprop
# 下载数据集
(X_train, y_train), (X_test, y_test) = mnist.load_data()
# 数据预处理
X_train = X_train.reshape(X_train.shape[0], -1) / 255.
X_test = X_test.reshape(X_test.shape[0], -1) / 255.
y_train = np_utils.to_categorical(y_train, num_classes=10)
y_test = np_utils.to_categorical(y_test, num_classes=10)
# 不使用model.add(), 用以下方式也可以构建网络
model = Sequential([
    Dense(400, input_dim=784),
    Activation('relu'),
    Dense(10),
    Activation('softmax'),
])
# 定义优化器
rmsprop = RMSprop(lr=0.001, rho=0.9, epsilon=1e-08, decay=0.0)
model.compile(optimizer=rmsprop,
              loss='categorical_crossentropy',
              metrics=['accuracy']) # metrics赋值为'accuracy', 会在训练过程中输出正
# 这次我们用fit()来训练网路
print('Training -----')
model.fit(X_train, y_train, epochs=4, batch_size=32)
print('\nTesting -----')
# 评价训练出的网络
loss, accuracy = model.evaluate(X_test, y_test)
print('test loss: ', loss)
print('test accuracy: ', accuracy)

```

训练后得到: test loss: 0.0970609934615, test accuracy: 0.9743

5

第三个例子: 加经典网络的预训练模型 (以VGG16为例)

1. 当服务器不能联网时, 需要把模型*.h5文件下载到用户目录下的~/keras/model, 模型的预训练权重:
2. 通过以下代码加载VGG16:

```

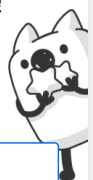
# 使用VGG16模型
from keras.applications.vgg16 import VGG16
print('Start build VGG16 -----')
# 获取vgg16的卷积部分, 如果要获取整个vgg16网络需要设置:include_top=True
model_vgg16_conv = VGG16(weights='imagenet', include_top=False)
model_vgg16_conv.summary()
# 创建自己的输入格式
# if K.image_data_format() == 'channels_first':
#     input_shape = (3, img_width, img_height)
# else:
#     input_shape = (img_width, img_height, 3)
input = Input(input_shape, name = 'image_input') #
# 将vgg16模型原始输入转换成自己的输入
output_vgg16_conv = model_vgg16_conv(input)
# output_vgg16_conv是包含了vgg16的卷积层, 下面我需要做二
x = Flatten(name='flatten')(output_vgg16_conv)
x = Dense(4096, activation='relu', name='fc1')(x)
x = Dense(512, activation='relu', name='fc2')(x)
x = Dense(128, activation='relu', name='fc3')(x)
x = Dense(1, activation='softmax', name='prediction')
# 最终创建出自己的vgg16模型
my_model = Model(input=input, output=x)

```

登录即可查看 超5亿 专业优质内容

超 5 千万创作者的优质提问、专业回答、深度文章和精彩视频尽在知乎。

立即登录/注册



```
my_model.summary()
```

其他Keras使用细节

1.查看GPU使用情况语句 (Linux)

```
# 1秒钟刷新一次  
watch -n 1 nvidia-smi
```

2.指定显卡

```
import os  
os.environ["CUDA_VISIBLE_DEVICES"] = "2"
```

这里指定了使用编号为2的GPU，大家可以根据需要和实际情况来指定使用的GPU

3.GPU并行

```
from model import unet  
G = 3 # 同时使用3个GPU  
with tf.device("/cpu:0"):  
    M = unet(input_rows, input_cols, 1)  
    model = keras.utils.training_utils.multi_gpu_model(M, gpus=G)  
    model.compile(optimizer=Adam(lr=1e-5), loss='binary_crossentropy', metrics =  
    model.fit(X_train, y_train,  
            batch_size=batch_size*G, epochs=nb_epoch, verbose=0, shuffle=True,  
            validation_data=(X_valid, y_valid))  
    model.save_weights('/path_to_save/model.h5')
```

4. 查看网络结构

```
print (model.summary())
```

效果如图:

登录即可查看 **超5亿** 专业优质内容

超 5 千万创作者的优质提问、专业回答、
深度文章和精彩视频尽在知乎。

立即登录/注册



```

conv2d_1 (Conv2D)      (None, 254, 254, 32)      896
activation_1 (Activation) (None, 254, 254, 32)      0
max_pooling2d_1 (MaxPooling2D) (None, 127, 127, 32)      0
conv2d_2 (Conv2D)      (None, 127, 127, 32)      25632
activation_2 (Activation) (None, 127, 127, 32)      0
max_pooling2d_2 (MaxPooling2D) (None, 43, 43, 32)      0
conv2d_3 (Conv2D)      (None, 41, 41, 16)      4624
activation_3 (Activation) (None, 41, 41, 16)      0
max_pooling2d_3 (MaxPooling2D) (None, 20, 20, 16)      0
flatten_1 (Flatten)    (None, 6400)              0
dense_1 (Dense)        (None, 1024)              6554624
activation_4 (Activation) (None, 1024)              0
dense_2 (Dense)        (None, 1)                 1025
activation_5 (Activation) (None, 1)                 0
=====
Total params: 6,586,801
Trainable params: 6,586,801
Non-trainable params: 0
None
Found 21032 images belonging to 2 classes.
Found 8477 images belonging to 2 classes.
Found 3761

```

5.保存网络结构图

```

# 你还可以用plot_model()来讲网络保存为图片
plot_model(my_model, to_file='my_vgg16_model.png')

```

小结：写在最后

本文介绍了一个灵活快速的深度学习框架——Keras，并且通过三个例子讲解了如何利用Keras搭建深度网络进行训练、如何使用预训练模型，还介绍了在使用Keras训练网络中的一些tricks。最后，祝各位炼丹师玩的愉快~

PS:

欢迎follow我的[GitHub](#)

还有我的[博客](#)

参考资料

- keras中文文档
- Keras开发者的github
- 莫烦keras教程代码

码字不易，欢迎给个赞!

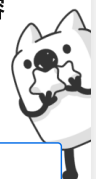
欢迎交流与转载，文章会同步发布在公众号：[机器学习算法全栈](#)

编辑于 2018-03-15 10:32

登录即可查看 **超5亿** 专业优质内容

超 5 千万创作者的优质提问、专业回答、深度文章和精彩视频尽在知乎。

立即登录/注册



写下你的评论...



10 条评论

默认 时间

- 
曲晓峰  
 监控 gpu 可以用 glances。可以同时看显卡占用率和显存占用率。
 2018-03-12 👍 3
- 
李中梁
 感谢分享~
 由于实验室服务器安装的是nvidia-smi，所以在文章中就用它来举例了。
 2018-03-15 👍 1
- 
Rinawale
 写得很棒 还会更新吗
 2018-03-16 👍 1
- 
David 9
[nooverfit.com/wp/3%e7%a...](#)
 2019-11-08 👍 赞
- 
ChenJ0212
 感觉keras的单卡训练和多卡训练在model保存和load时候还是蛮坑的
 2019-11-01 👍 赞
- 
小残
 不说说怎么个坑法?
 05-26 👍 赞
- 
NoOne
 第一个例子的标题应该是线性回归吧
 2019-10-10 👍 赞
- 
赵小生
 Dense 函数里面的参数是什么含义没有讲
 2019-06-03 👍 赞
- 
回眸一笑倒苍生 
 数字是神经元个数，第二个是激励函数
 2021-01-15 👍 赞
- 
帕齐尼
 您好，导入mnist数据集的时候为什么会提示ValueError: too many values to unpack (expected 2)错误呢?
 2019-01-16 👍 赞

写下你的评论...

文章被以下专栏收录

- 
机器学习算法工程师
 微信公众号，内容更精彩！
- 
机器学习算法工程师
 欢迎关注同名微信公众号

×

登录即可查看 超5亿 专业优质内容

超 5 千万创作者的优质提问、专业回答、深度文章和精彩视频尽在知乎。

立即登录/注册

