

纪念第一个Pytorch/TensorFlow程序

作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/>

截止今日，写了17篇有关[Deep Learning](#)的博文，如下所示

深度多视图子空间聚类

摘要：深度多视图子空间聚类 作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/> 1. Deep Multi-view Subspace Clustering with Unified and Discriminative Learning 这部分是对Deep M [阅读全文](#)
posted @ [2021-03-26 15:33](#) 凯鲁嘎吉 阅读(510) 评论(0) 推荐(0) [编辑](#)

具有协同训练的深度嵌入多视图聚类

摘要：具有协同训练的深度嵌入多视图聚类 作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/> 本文对Deep Embedded Multi-view Clustering with Collaborative Training这篇文章进行总结，前提请了解DEC、I [阅读全文](#)
posted @ [2021-03-25 22:18](#) 凯鲁嘎吉 阅读(255) 评论(0) 推荐(0) [编辑](#)

结构深层聚类网络

摘要：结构深层聚类网络 作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/> 本文是对Structural Deep Clustering Network这篇文章的展开与叙述。了解这篇文章的前提需要知道GCN图卷积神经网络，DEC深度嵌入聚类，自编码器等。这是 [阅读全文](#)
posted @ [2021-03-25 17:39](#) 凯鲁嘎吉 阅读(246) 评论(0) 推荐(0) [编辑](#)

宽度学习系统用于数据聚类

摘要：宽度学习系统用于数据聚类 作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/> 阅读本文的前提：宽度学习系统(Broad Learning System, BLS)。了解什么是宽度学习系统，有一篇文章是关于BLS的综述，适合BLS入门的初学者看，参看参考文献 [阅读全文](#)
posted @ [2021-03-25 17:14](#) 凯鲁嘎吉 阅读(129) 评论(0) 推荐(0) [编辑](#)

从对比学习(Contrastive Learning)到对比聚类(Contrastive Clustering)

摘要：从对比学习(Contrastive Learning)到对比聚类(Contrastive Clustering) 作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/> 想要了解对比聚类，首先应该清楚对比学习的整个过程。最经典的对比学习的文章是Hinton团队 [阅读全文](#)
posted @ [2021-03-05 19:13](#) 凯鲁嘎吉 阅读(1330) 评论(4) 推荐(1) [编辑](#)

多视图子空间聚类/表示学习(Multi-view Subspace Clustering/Representation Learning)

摘要：多视图子空间聚类/表示学习(Multi-view Subspace Clustering/Representation Learning) 作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/> “横看成岭侧成峰，远近高低各不同。”多视图聚类是最近一个较为热门 [阅读全文](#)
posted @ [2021-01-21 17:30](#) 凯鲁嘎吉 阅读(1388) 评论(4) 推荐(0) [编辑](#)

关于“Unsupervised Deep Embedding for Clustering Analysis”的优化问题

摘要：关于“Unsupervised Deep Embedding for Clustering Analysis”的优化问题 作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/> Deep Embedding Clustering (DEC)和Improved [阅读全文](#)
posted @ [2021-01-19 22:07](#) 凯鲁嘎吉 阅读(225) 评论(1) 推荐(0) [编辑](#)

ClusterGAN: 生成对抗网络中的潜在空间聚类

摘要：ClusterGAN: 生成对抗网络中的潜在空间聚类 作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/> 生成对抗网络(GANs)在许多无监督学习任务中取得了显著的成功，毫无疑问，聚类是一个重要的无监督学习问题。虽然可以利用GANs中的潜在空间反向投影进行 [阅读全文](#)
posted @ [2020-11-20 10:38](#) 凯鲁嘎吉 阅读(1106) 评论(0) 推荐(0) [编辑](#)

宽度学习系统(Broad Learning System, BLS)

摘要：宽度学习系统(Broad Learning System, BLS) 作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/> 宽度学习系统(Broad Learning System, BLS)由Chen等[1]提出，其基于“平展型”神经网络，因其高效性、结构 [阅读全文](#)
posted @ [2020-10-28 13:02](#) 凯鲁嘎吉 阅读(2484) 评论(0) 推荐(1) [编辑](#)

MATLAB实例：BP神经网络用于回归任务

摘要：MATLAB实例：BP神经网络用于回归(非线性拟合)任务 作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/> 问题描述 给定多元(多维)数据X，有真实结果Y，对这些数据进行拟合(回归)，得到拟合函数的参数，进而得到拟合函数，现在进来一些新样本，对这些新样本 [阅读全文](#)
posted @ [2020-10-18 17:37](#) 凯鲁嘎吉 阅读(2273) 评论(8) 推荐(1) [编辑](#)

基于图嵌入的高斯混合变分自编码器的深度聚类(Deep Clustering by Gaussian Mixture Variational Autoencoders with Graph Embedding, DGG)

摘要：基于图嵌入的高斯混合变分自编码器的深度聚类 Deep Clustering by Gaussian Mixture Variational Autoencoders with Graph Embedding, DGG 作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kail/> [阅读全文](#)
posted @ [2020-07-07 16:21](#) 凯鲁嘎吉 阅读(1382) 评论(0) 推荐(0) [编辑](#)

没想到吧，我的电脑连Python都没安装，之前一直用的Notepad++看的程序

变分深度嵌入(Variational Deep Embedding, VaDE)

摘要：变分深度嵌入(Variational Deep Embedding, VaDE) 作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/> 这篇博文主要是对论文 "Variational Deep Embedding: An Unsupervised and Generative Model" 的解读。 阅读全文
posted @ 2020-05-13 15:47 凯鲁嘎吉 阅读(1689) 评论(7) 推荐(1)

变分推断与变分自编码器

摘要：变分推断与变分自编码器 作者：凯鲁嘎吉 - 博客园
推导过程，但变分自编码器涉及一些概率统计的基础知识。
posted @ 2020-03-11 17:49 凯鲁嘎吉 阅读(1036) 评论(0) 推荐(0)

A Survey of Deep Clustering Algorithms

摘要：A Survey of Deep Clustering Algorithms 作者：Kailugaji
and New Methods 2. A Survey of Deep Clustering Algorithms 阅读全文
posted @ 2020-02-29 16:51 凯鲁嘎吉 阅读(823) 评论(1) 推荐(0)

Deep Clustering Algorithms

摘要：Deep Clustering Algorithms 作者：凯鲁嘎吉 - 博客园
Embedded Clustering(DEC)->Improved Deep Embedded Clustering(DEC) 阅读全文
posted @ 2019-12-27 10:22 凯鲁嘎吉 阅读(2504) 评论(0) 推荐(0)

没想到吧!



要介绍变分自编码器(Variational Auto-Encoder, VAE)及其
[阅读全文](#)

/kailugaji/ 1. Clustering with Deep Learning: Taxonomy

一文研究路线：深度自编码器(Deep Autoencoder)->Deep

说到Notepad++，这是一款免费的开源文本和源代码编辑器，无论电脑上有没有安装编程软件，用这个软件可以打开各种程序文件，比如MATLAB的.m文件，python的.py文件，C语言的.c文件等等，而且可以高亮文本，显示行号等等，看程序特别方便。

```
1 """ Code for loading data. """
2 import numpy as np
3 import os
4 import random
5 import tensorflow as tf
6
7 from tensorflow.python.platform import flags
8 from utils import get_images
9
10 FLAGS = flags.FLAGS
11
12 class DataGenerator(object):
13     """
14     Data Generator capable of generating batches of sinusoid or Omniglot data.
15     A "class" is considered a class of omniglot digits or a particular sinusoid function.
16     """
17     def __init__(self, num_samples_per_class, batch_size, config={}): # num_samples_per_class: K-way中的K
18         """
19         Args:
20             num_samples_per_class: num samples to generate per class in one batch 每个batch中每个类别所包含的K个样本数
21             batch_size: size of meta batch size (e.g. number of functions) 每次元更新抽样的任务数
22         """
23         self.batch_size = batch_size
24         self.num_samples_per_class = num_samples_per_class
25         self.num_classes = 1 # by default 1 (only relevant for classification problems)
26
```

趁着昨天周日，没有其他要紧的事情，自己动手从零开始安装Python以及Pytorch/TensorFlow框架



当然，中途由于各种版本问题，一直重复着安装卸载安装卸载的过程

简单回顾一下安装流程

1) 安装python：在[python官网](#)找到Windows的版本，选择Windows x86-64 executable installer系列的，我下了Python 3.8.5，直接就是.exe文件(当然，可以是其他版本，可能我后续还会再重复一次卸载安装。。。)



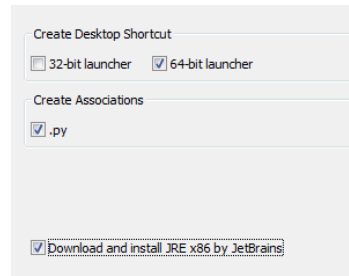
双击运行该程序，自定义安装，勾选Add Python 3.8 to PATH，Add Python to environment variables，更改安装路径，之后一路默认，完成安装。在cmd命令行输入 python 或者 python --version，能够看到python的版本，即完成安装。

```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [版本 10.0.19042.1165]
(c) Microsoft Corporation。保留所有权利。

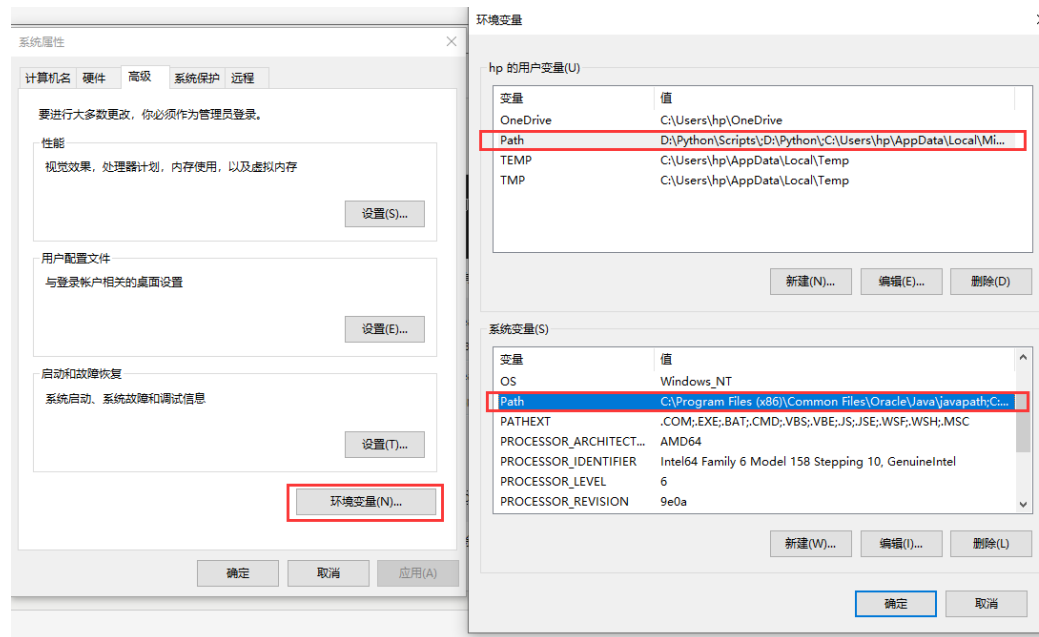
C:\Users\hp>python --version
Python 3.8.5

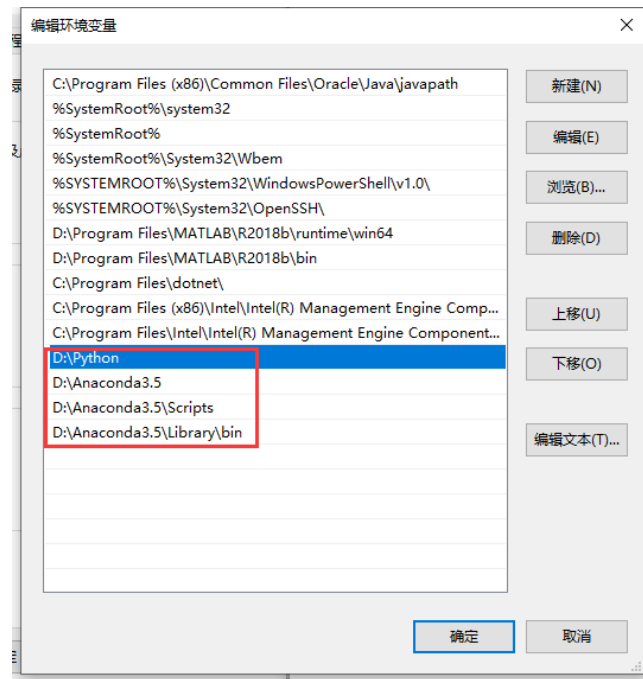
C:\Users\hp>python
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> _
```

2) 安装pycharm编辑器：去[pycharm官网](#)下载，社区版虽然免费，但功能不全，最好安装专业版(收费，网上也有一些破解方法)，我的是PyCharm 2018.3.5。安装时，自定义安装路径，再勾选这几个，然后一路默认，即可完成安装。

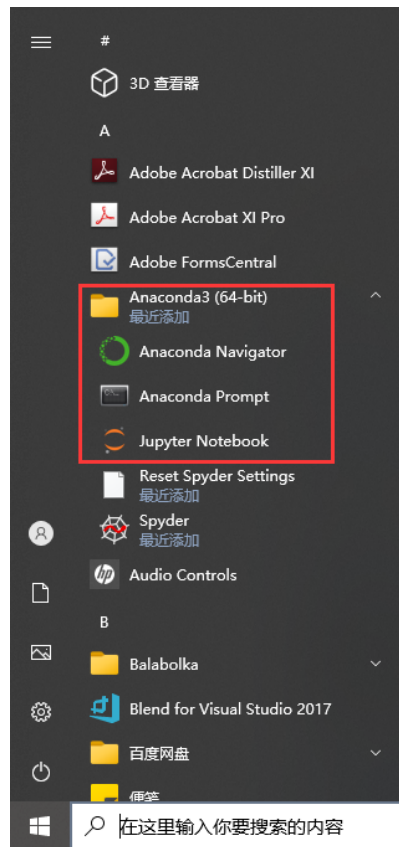


3) 安装Anaconda: 在[清华镜像源](#)找到合适的版本, 我用的Anaconda3-5.2.0-Windows-x86_64.exe。安装时, 除了自定义路径之外, 其余一路默认。安装好之后, 需要添加路径, 在"设置"->"关于"->右上角"高级系统设置"->"系统属性"->"高级"->"环境变量"里面添加路径。(由于我不知道往哪个里面添加, 于是用户变量与系统变量都添上了)





4) Anaconda Prompt操作: 打开电脑"开始", 可以看到最近添加了Anaconda3。



找到里面的Anaconda Prompt，输入 `conda info --envs`，可以查看用户定义的虚拟环境，*表示当前所处的环境。

```

Anaconda Prompt

(base) C:\Users\hp>conda info --envs
# conda environments:
#
base                    * D:\Anaconda3.5
Python code             D:\Anaconda3.5\envs\Python code
Pythonproject           D:\Anaconda3.5\envs\Pythonproject
Pythonproject2          D:\Anaconda3.5\envs\Pythonproject2
pytorch                 D:\Anaconda3.5\envs\pytorch
tensorflow              D:\Anaconda3.5\envs\tensorflow

(base) C:\Users\hp>
```

创建虚拟环境：创建pytorch环境 `conda create -n pytorch python=3.8`，激活该环境 `conda activate pytorch`，如下所示

```
(base) C:\Users\hp>conda activate pytorch

(pytorch) C:\Users\hp>python
Python 3.8.11 (default, Aug 6 2021, 09:57:55) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

退出该环境 `conda deactivate`，如下所示

```
(base) C:\Users\hp>conda activate pytorch

(pytorch) C:\Users\hp>python
Python 3.8.11 (default, Aug 6 2021, 09:57:55) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

(pytorch) C:\Users\hp>conda deactivate

(base) C:\Users\hp>_
```

5) 搭建pytorch: 在[pytorch官网](https://pytorch.org/)选择适合自己的版本

PyTorch Build	Stable (1.9.0)		Preview (Nightly)		LTS (1.8.1)		
Your OS	Linux		Mac		Windows		
Package	Conda		Pip		LibTorch		Source
Language	Python				C++ / Java		
Compute Platform	CUDA 10.2		CUDA 11.1		ROCm 4.2 (beta)		CPU
Run this Command:	conda install pytorch torchvision torchaudio cpuonly -c pytorch						

并在Anaconda Prompt中pytorch环境下安装

```
(base) C:\Users\hp>conda activate pytorch

(pytorch) C:\Users\hp>conda install pytorch torchvision torchaudio cpuonly -c pytorch
```

输入 `import torch` 不报错即为安装完成

```
(base) C:\Users\hp>conda activate pytorch

(pytorch) C:\Users\hp>python
Python 3.8.11 (default, Aug 6 2021, 09:57:55) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> _
```

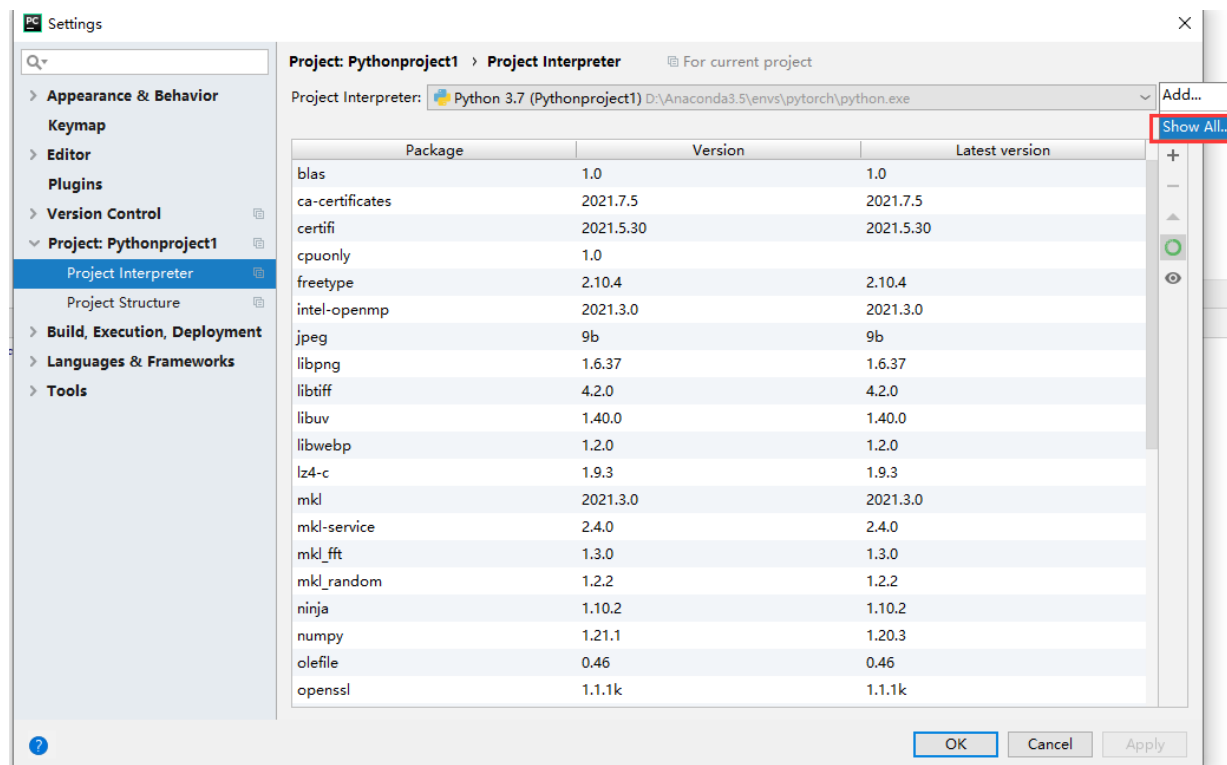
我也在pycharm中python环境下通过 `pip install torch==1.6.0 torchvision==0.7.0 -f https://download.pytorch.org/whl/torch_stable.html` 安装torch包与torchvision包。

输入 `conda list`，显示已经安装的包，可以看到torch包。

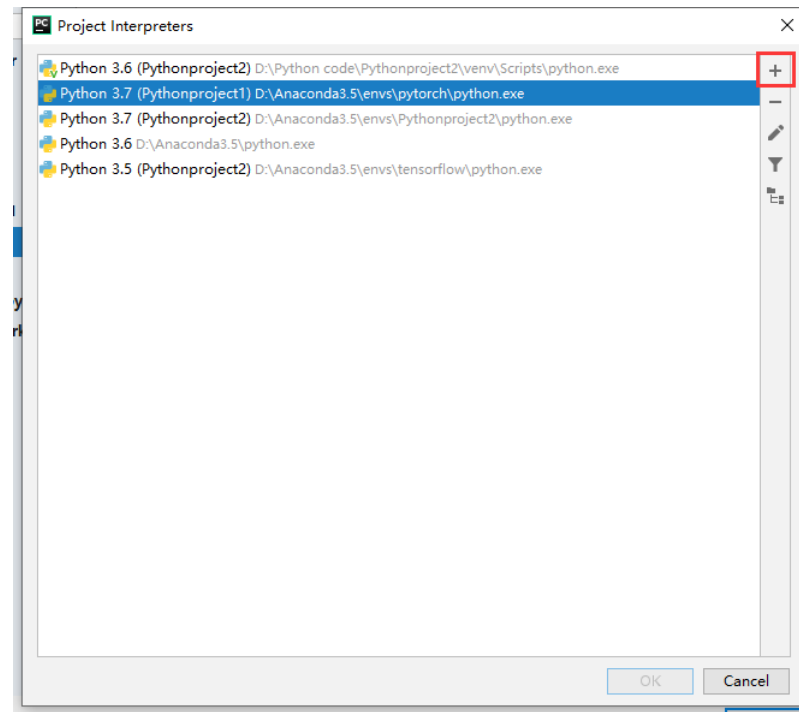

```
(base) C:\Users\hp>conda list
# packages in environment at D:\Anaconda3.5:
#
# Name                      Version           Build    Channel
_ipyw_jlab_nb_ext_conf     0.1.0             py36he6757f0_0 defaults
absl-py                     0.13.0            pypi_0    pypi
alabaster                   0.7.10            py36hcd07829_0 defaults
anaconda                    5.2.0             py36_3    defaults
anaconda-client             1.6.14            py36_0    defaults
anaconda-navigator          1.8.7             py36_0    defaults
anaconda-project            0.8.2             py36hfad2e28_0 defaults
asn1crypto                  0.24.0            py36_0    defaults
astroid                     1.6.3             py36_0    defaults
astropy                     3.0.2             py36h452e1ab_1 defaults
attrs                       18.1.0            py36_0    defaults
babel                       2.5.3             py36_0    defaults
backcall                    0.1.0             py36_0    defaults
backports                   1.0               py36h81696a8_1 defaults
backports.shutil_get_terminal_size 1.0.0             py36h79ab834_2 defaults
beautifulsoup4              4.6.0             py36hd4cc5e8_1 defaults
btstack                      0.8.1             py36h66f69d1_1 defaults
```

```
Anaconda Prompt - conda deactivate - conda deactivate
sphinx                      1.7.4             py36_0    defaults
sphinxcontrib               1.0               py36hbba3d2_1 defaults
sphinxcontrib-websupport    1.0.1             py36hb5e5916_1 defaults
spyder                      3.2.8             py36_0    defaults
sqlalchemy                  1.2.7             py36ha85dd04_0 defaults
sqlite                      3.36.0            h2bbff1b_0 defaults
statsmodels                 0.9.0             py36h452e1ab_0 defaults
sympy                       1.1.1             py36h96708e0_0 defaults
tblib                       1.3.2             py36h30f5020_0 defaults
tensorboard-data-server     0.6.1             pypi_0    pypi
tensorboard-plugin-wit      1.8.0             pypi_0    pypi
terminado                   0.8.1             py36_1    defaults
testpath                    0.3.1             py36h2698cfe_0 defaults
tk                           8.6.7             hcb92d03_3 defaults
toolz                       0.9.0             py36_0    defaults
torch                       1.6.0             pypi_0    pypi
torchvision                 0.7.0             pypi_0    pypi
tornado                     5.0.2             py36_0    defaults
tqdm                        4.82.0            pyhd3eb1b0_1 defaults
traitlets                   4.3.2             py36h096827d_0 defaults
typing                      3.6.4             py36_0    defaults
typing-extensions           3.10.0.0          pypi_0    pypi
unicodcsv                   0.14.1            py36h6450c06_0 defaults
urllib3                     1.22              py36h276f60a_0 defaults
vc                           14.2              h21ff451_1 defaults
vs2015_runtime              14.27.29016       h5e58377_2 defaults
wcwidth                     0.1.7             py36h3d5aa90_0 defaults
webencodings                0.5.1             py36h67c50ae_1 defaults
werkzeug                    0.14.1            py36_0    defaults
```

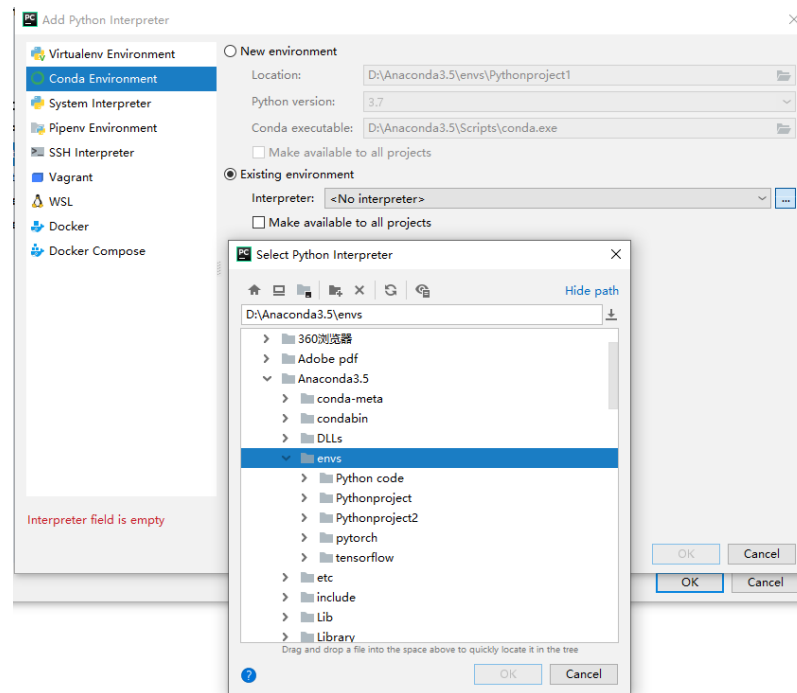
6) pycharm设置: "File"-">"Settings"-">"Project Interpreter"-">"Show All"



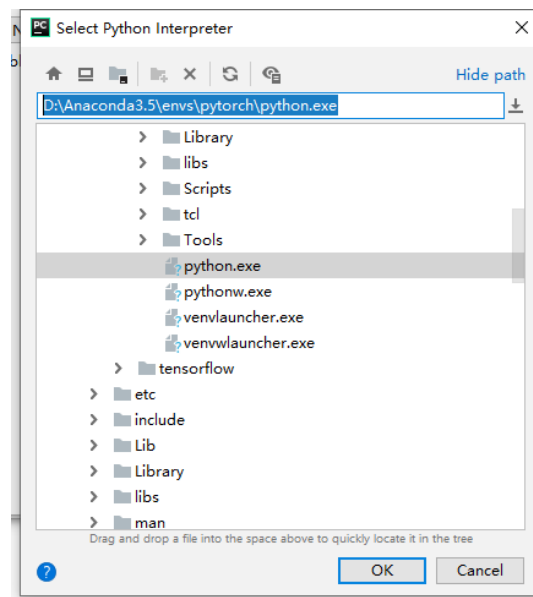
点击加号



选择"Conda Environment"-">"Existing environment"->找到Anaconda安装路径里面的环境, D:\Anaconda3.5\envs

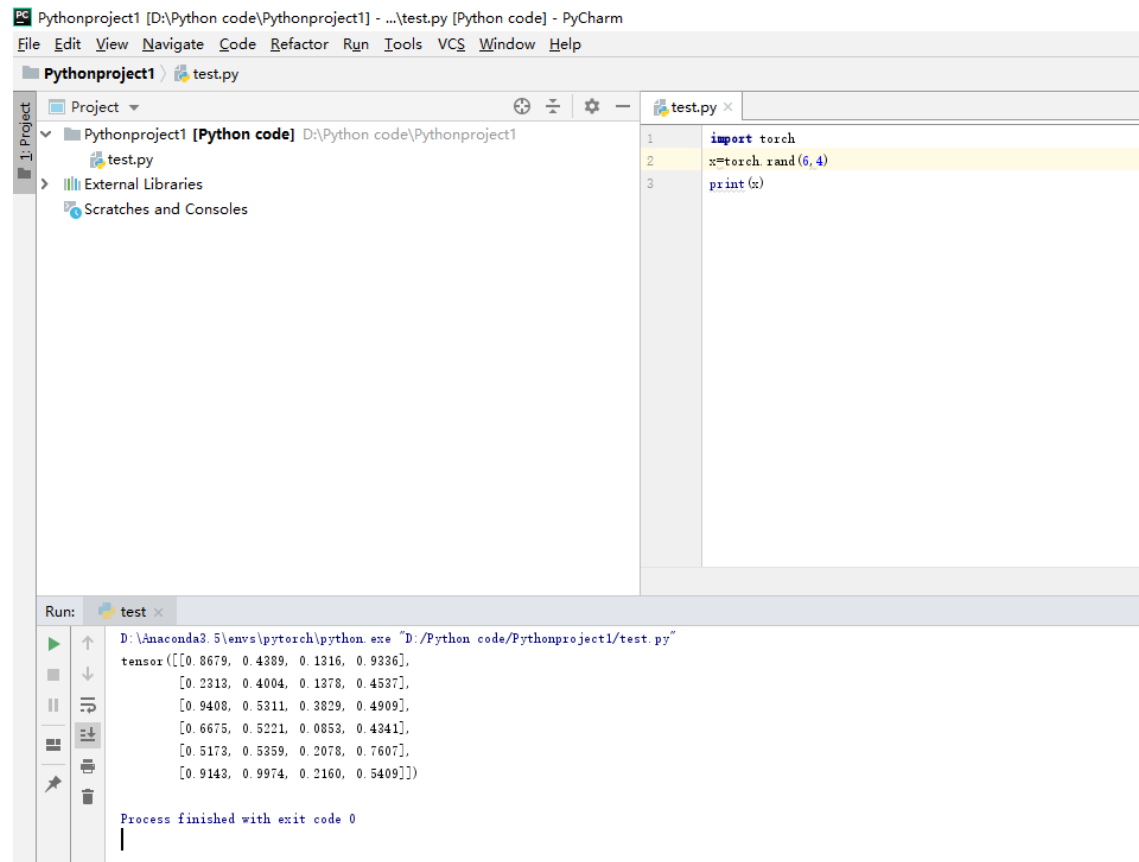


找到 D:\Anaconda3.5\envs\pytorch\python.exe ， 点击ok， 一路确定即可。



7) 完成， 运行第一个pytorch环境下的python程序。

```
import torch
x=torch.rand(6,4)
print(x)
```



补充：安装GPU版本的pytorch：

.condarc文件中的内容(添加清华镜像源)

channels:

- <http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/main/>
- <http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/free/>
- <http://mirrors.bfsu.edu.cn/anaconda/pkg/free/>
- <http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/>

show_channel_urls: true

ssl_verify: true

report_errors: true

quiet: true

在<https://pytorch.org/get-started/previous-versions/>找到与CUDA版本匹配的文件，并在创建的环境中输入`conda install pytorch==1.7.1 torchvision==0.8.2 torchaudio==0.7.2 cudatoolkit=10.1`

```
(D:\ProgramData\Anaconda3) C:\Users\24410>nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Sun_Jul_28_19:12:52_Pacific_Daylight_Time_2019
Cuda compilation tools, release 10.1, V10.1.243
```

注意去掉最后的-c pytorch。

进入Python，输入下面程序：

```
import torch
torch.cuda.is_available()
```

■ Anaconda Prompt - python

```
(D:\ProgramData\Anaconda3) C:\Users\24410>activate RL
(RL) C:\Users\24410>python
Python 3.7.1 (default, Oct 28 2018, 08:39:03) [MSC v.1912 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>>
>>> torch.cuda.is_available()
True
>>>
```

返回True，则成功。

下面是TensorFlow搭建。

1) 在Anaconda Prompt中修改清华镜像源，在base这一环境上修改。

```
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgsrc/free/
```

```
conda config --set show_channel_urls yes
```

2) 创建TensorFlow虚拟环境：输入 conda create -n tensorflow python=3.5.2

3) 安装完成后，输入 activate tensorflow 进入该环境，并安装TensorFlow

安这个1.8版本的：

```
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple/ https://mirrors.tuna.tsinghua.edu.cn/pypi/web/packages/3f/bb/dd01844cf88d15264d92e12a8b89526e1d805c082b8e945b632d4a1989a4/tensorflow-1.8.0-cp35-cp35m-win_amd64.whl#sh
```

(旧的，舍掉) pip install -i https://pypi.tuna.tsinghua.edu.cn/simple/ https://mirrors.tuna.tsinghua.edu.cn/tensorflow/windows/cpu/tensorflow-1.1.0-cp35-cp35m-win_amd64.whl

```
(base) C:\Users\hp>activate tensorflow

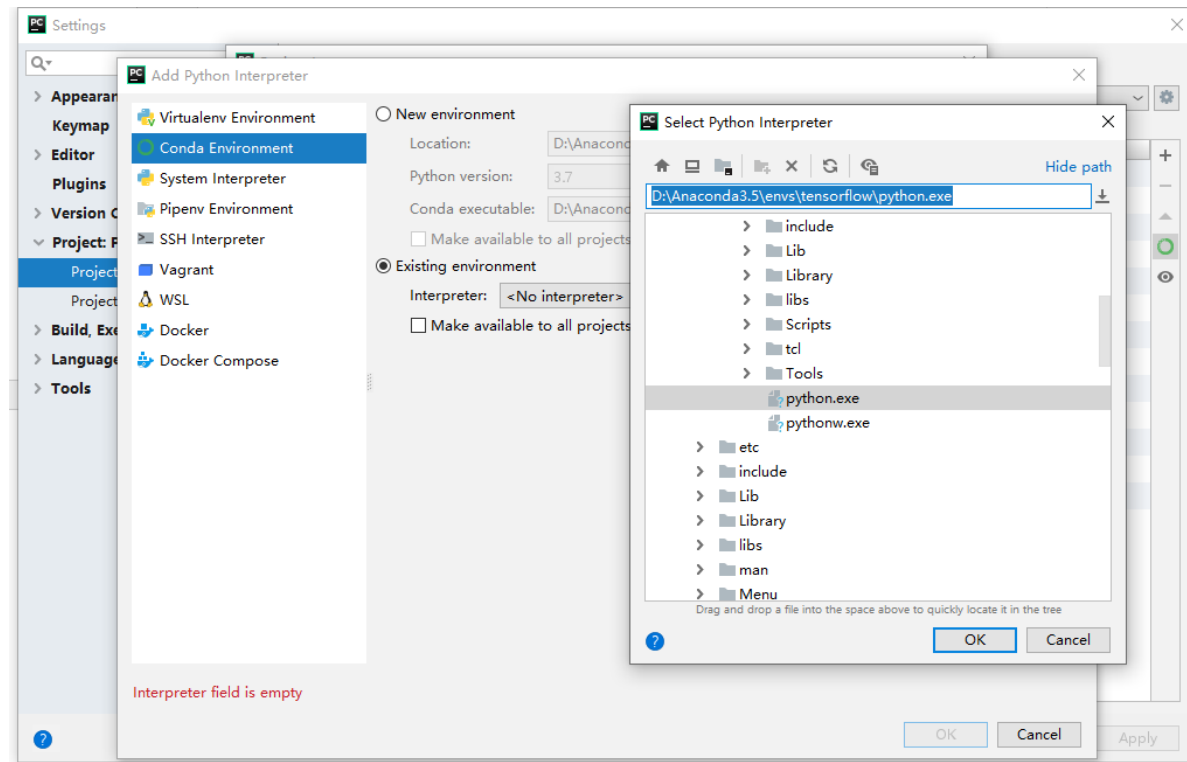
(tensorflow) C:\Users\hp>pip install -i https://pypi.tuna.tsinghua.edu.cn/simple/ https://mirrors.tuna.tsinghua.edu.cn/tensorflow/windows/cpu/tensorflow-1.1.0-cp35-cp35m-win_amd64.whl
DEPRECATION: Python 3.5 reached the end of its life on September 13th, 2020. Please upgrade your Python as Python 3.5 is no longer maintained. pip 21.0 will drop support for Python 3.5 in January 2021. pip 21.0 will remove support for this functionality.
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple/
Collecting tensorflow==1.1.0
  Downloading https://mirrors.tuna.tsinghua.edu.cn/tensorflow/windows/cpu/tensorflow-1.1.0-cp35-cp35m-win_amd64.whl (19.4 MB)
    | 1.9 MB 6.4 MB/s eta 0:00:03
```

4) 进入python, 输入 `import tensorflow as tf`, 如果没报错, 说明安装完成

```
(base) C:\Users\hp>activate tensorflow

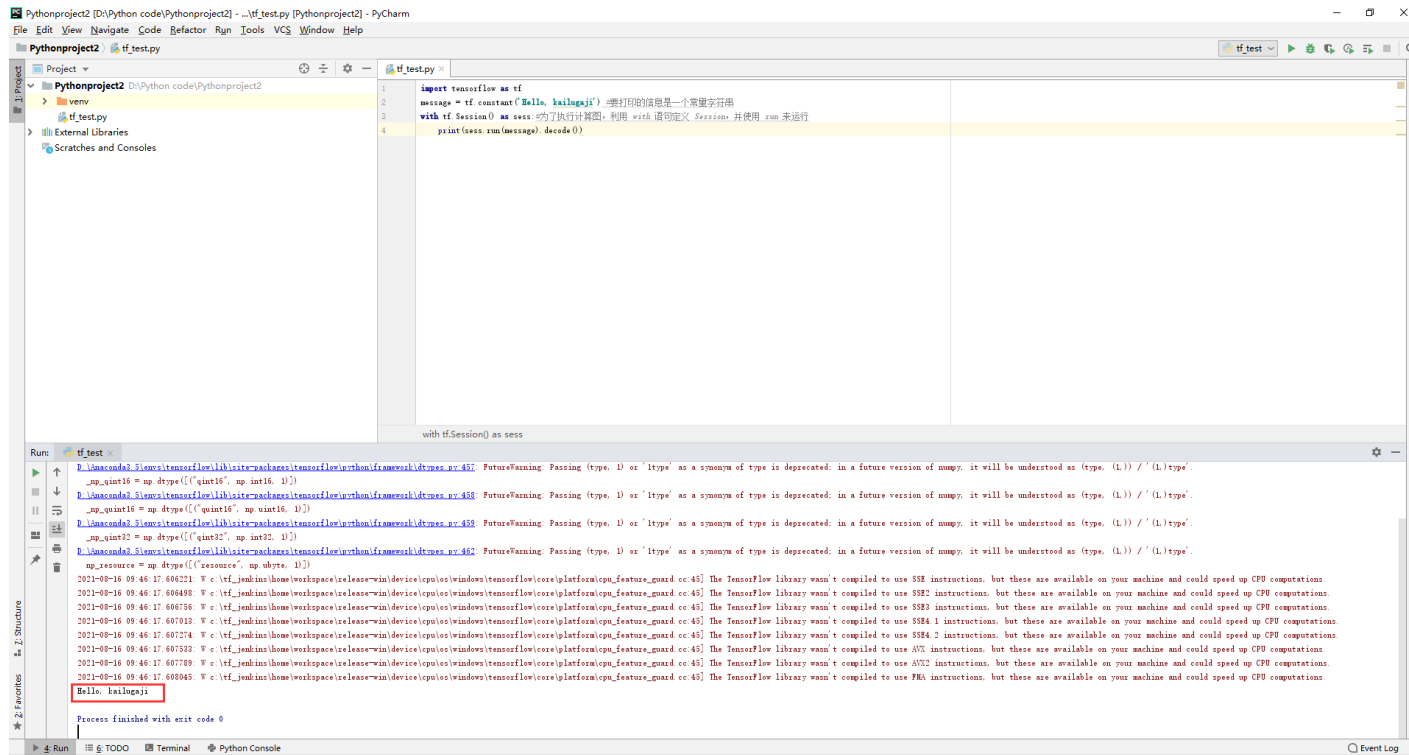
(tensorflow) C:\Users\hp>python
Python 3.5.2 |Continuum Analytics, Inc.| (default, Jul 5 2016, 11:41:13) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import tensorflow as tf
D:\Anaconda3.5\envs\tensorflow\lib\site-packages\tensorflow\python\framework\dtypes.py:455: FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,))
  / (1,)type.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
D:\Anaconda3.5\envs\tensorflow\lib\site-packages\tensorflow\python\framework\dtypes.py:456: FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,))
  / (1,)type.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
D:\Anaconda3.5\envs\tensorflow\lib\site-packages\tensorflow\python\framework\dtypes.py:457: FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,))
  / (1,)type.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
D:\Anaconda3.5\envs\tensorflow\lib\site-packages\tensorflow\python\framework\dtypes.py:458: FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,))
  / (1,)type.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
D:\Anaconda3.5\envs\tensorflow\lib\site-packages\tensorflow\python\framework\dtypes.py:459: FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,))
  / (1,)type.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
D:\Anaconda3.5\envs\tensorflow\lib\site-packages\tensorflow\python\framework\dtypes.py:462: FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,))
  / (1,)type.
  _np_resource = np.dtype([("resource", np.ubyte, 1)])
>>>
```

5) pycharm设置: 与pytorch设置一样, "File"->"Settings"->"Project Interpreter"->"Show All", 点击加号, 选择"Conda Environment"->"Existing environment", 找到 D:\Anaconda3.5\envs\tensorflow\python.exe, 点击ok, 即可。



6) 运行第一个TensorFlow框架下的python程序

```
import tensorflow as tf
message = tf.constant('Hello, kailugaji') #要打印的信息是一个常量字符串
with tf.Session() as sess: #为了执行计算图，利用 with 语句定义 Session，并使用 run 来运行
    print(sess.run(message).decode())
```

至此，所有的都已安装完毕。

下面给出Anaconda Prompt一些常用命令

- 1 #打开Anaconda Prompt
- 2 #检查conda版本
- 3 conda -V
- 4 #查看已安装包
- 5 conda list
- 6 #安装包
- 7 conda install numpy
- 8 #删除包
- 9 conda remove numpy
- 10 #查看当前存放的虚拟环境
- 11 conda env list
- 12 #创建python虚拟环境
- 13 conda create -n pytorch python=3.8
- 14 #激活虚拟环境
- 15 conda activate tensorflow
- 16 conda activate pytorch
- 17 #关闭虚拟环境
- 18 conda deactivate

19 #删除虚拟环境

20 conda remove -n tensorflow --all

21 conda remove -n pytorch --all

升级pip: python -m pip install --upgrade pip

安装包: pip install numpy

解决Anaconda navigator打不开的问题:

在cmd窗口输入命令语句: tasklist | findstr "pythonw"

找到pythonw的PID, 如: 25376, 在cmd窗口输入命令语句: taskkill /F /PID 25376, 将所有的pythonw全部关掉。

在安装路径下找到 anaconda3\Lib\site-packages\anaconda_navigator\api\conda_api.py, 在1300行左右, 把data=yaml.load(f) 改为 data=yaml.safe_load(f), 保存, 并重启电脑。

GPU版本用这个: <https://zhuanlan.zhihu.com/p/124231172>

CUDA: 10.1.105

cuDNN: v7.6.5 for CUDA 10.1

tensorflow==1.15.0

参考文献:

[1] windows环境下tensorflow安装过程详解_小白_努力-CSDN博客_tensorflow安装 <https://blog.csdn.net/laobai1015/article/details/84580243>

[2] WIN10下pytorch环境配置 (安装了半天的血泪史)_熊孩纸_的博客-CSDN博客_pytorch环境搭建 https://blog.csdn.net/weixin_39487353/article/details/104666953