学习笔记

# Cmd窗口命令

【切换盘符】d:

【创办虚拟环境】conda create -n Name python=3.8

【虚拟环境查看】conda env list

【激活虚拟环境】conda activate Name

【去激活】conda deactivate

【安装幻灯片格式】pip install rise

【打开记事本】jupyter notebook

【查看文件目录】dir

# CUDA安装

【查看CUDA适配版本】win10左下角搜索窗输入“NVIDIA Control Panel”，弹出NVIDIA控制面板对话框，点击左下角“系统信息”，弹出系统信息对话框，点击“组件”，查看NVCUDA64.DLL对应的产品名称，如“NVIDIA CUDA 11.4.108 driver”中的11.4即为CUDA版本号；

【CUDA安装】CUDA下载地址为：https://developer.nvidia.com/zh-cn/cuda-toolkit，依次选择Windows、x86\_64、10、exe(local)，点击下载后下载安装即可；

【cuDNN安装】cuDNN下载地址为：https://developer.nvidia.com/rdp/cudnn-download，选择cuDNN Library for Windows 10版本进行下载；解压后可看到bin、include、lib三个文件夹，将这三个文件夹中的文件分别copy到CUDA安装路径下的bin、include、lib文件夹下面即完成安装（development安装路径下）；

【环境变量配置】CUDA安装、cuDNN安装及二者环境变量配置过程可参考：https://blog.csdn.net/u011473714/article/details/95042856；具体为：搜索win10左下搜索框，输入“编辑系统环境变量”，点击后弹出“系统属性”对话框，选择“高级”，点击“环境变量”，弹出“环境变量”对话框，点击“变量”中的“Path”，点击“编辑”，新建四个环境变量（D:\Softwares\cuda\anzhuangweizhi\development\bin；D:\Softwares\cuda\anzhuangweizhi\development\lib\x64；D:\Softwares\cuda\anzhuangweizhi\samples\bin\win64；D:\Softwares\cuda\anzhuangweizhi\samples\common\lib\x64），点击确定后返回；然后点击“系统变量”下的“新建”按钮，分别新建如下七个“变量名”和“变量值”对，即：

CUDA\_SDK\_PATH = D:\Softwares\cuda\anzhuangweizhi\samples

CUDA\_LIB\_PATH = %CUDA\_PATH%\lib\x64

CUDA\_BIN\_PATH = %CUDA\_PATH%\bin

CUDA\_SDK\_BIN\_PATH = %CUDA\_SDK\_PATH%\bin\win64

CUDA\_SDK\_LIB\_PATH = %CUDA\_SDK\_PATH%\common\lib\x64；

CUDA\_PATH = D:\Softwares\cuda\anzhuangweizhi\development；（该路径自动生成）

CUDA\_PATH\_V11\_4 = D:\Softwares\cuda\anzhuangweizhi\development；（该路径自动生成）

# Jupyter NoteBook技巧

【.ipynb转成.py】在Terminal中打开.ipynb文件所在目录，输入jupyter notebook，在弹出网页界面中点击File，选择Download as，选择Python(.py)，即可下载对应转化好的.py文件。在生成.py的文件当中注释掉get\_ipython对应的代码行，否则对应.py文件运行会报错：get\_ipython().run\_line\_magic('matplotlib', 'inline')，与此同时，修改loss.data[0]为loss.item()即可。

# GPU转为CPU

【GPU代码转为CPU代码】在程序最开始的地方加上device = torch.device('cuda' if torch.cuda.is\_available() else 'cpu')，同时将代码其余地方出现.cuda()的地方改成.to(device)就可以在无gpu的环境中运行了。同时将num\_workers改为0。

# Pycharm

## 环境配置

【创建Project&环境配置】File→New Project弹出create project对话框，在Location后面选择Project的文件夹位置，选中Existing interpreter，点击Interpreter之后的省略号按钮，弹出Add Python Interpreter对话框，选择Conda Environment，点击Interpreter位置后省略号按钮，弹出Select Python Interpreter对话框，选择已经创建的虚拟环境，如envs/MyPyTorch/下的python.exe，然后点击确定，最后点击create进行创建。

【已有Project&环境配置】点击File→Settings→Project→Python Interpreter，选择Project Interpreter后面的齿轮，点击Add弹出Add Python Interpreter对话框，选择Conda Environment，选择Existing environment，点击Interpreter位置后省略号按钮，弹出Select Python Interpreter对话框，选择已经创建的虚拟环境，如envs/MyPyTorch/下的python.exe，然后点击确定即可。

【添加自定义库文件】点击File→Settings→Project→Project Structure，选择sources，点击Add Content Root,弹出Select content root directory对话框，选择自定义库文件目录，如D:/PythonProjects/ProjectsCodes/PublicFunctions，点击ok关闭对话框，点击Apply，然后点击ok关闭对话框。

## 快捷键

【注释/反注释】Ctrl+/ 注释选中的代码。

# 求导法则

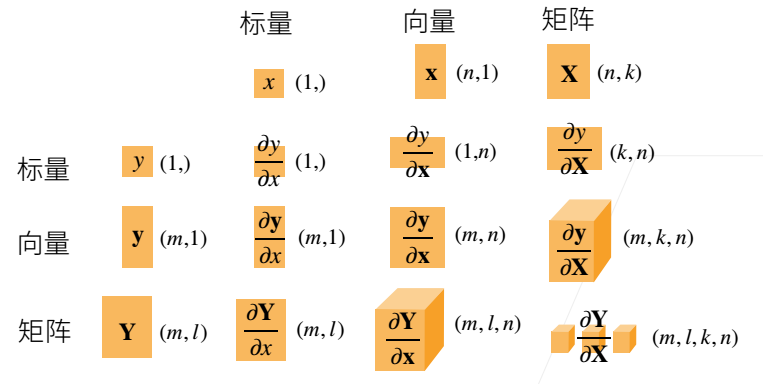
【根本法则】标量和列向量的导数公式为：

【根本法则】列向量和标量的导数公式为：

【根本法则】列向量和列向量的导数公式为：

【根本法则】列向量和列向量的内积与列向量的偏导数公式为：

【法则总结】具体见如下截图：



# Gitbash命令

【查看工作目录变动文件】查看工作目录中所有新建或者修改的文件列表信息，git status；

【文件变动查询】查看工作目录中文件异同命令为git diff；查看缓存区Index中文件与最新文件版本文件异同命令为git diff –staged；

【撤销缓存区文件】撤销缓存区文件，但保留文件内容命令为git reset <file>；

【本地仓库上传远程仓库】

1. 创建新仓库：首先到本地仓库目录下查看是否有.git文件夹，如果没有在，在git的bash输入git init <project-name>，以创建新的git仓库，其中project-name为新的仓库名称，可指定也可省略；
2. 提交更改到缓存区（Index）：然后在本地仓库放需要上传的内容，然后输入git add .或git add <filename>或git add \*；
3. 提交更改到Head区：再输入git commit -m "注释（可以任意）"；
4. 推送到远端主机：最后就推送上去，输入以下命令git push origin <master>，把本地库的内容推送到远程，使用git push -u origin <master>，其中master可替换为想要推送的任何分支；

工作目录

Index

Add

本机

远端

Head

Commit

工作目录

Push



【本地仓库同步远程仓库】如果需要把远程仓库同步到本地仓库 我们就要拉取最新数据到本地仓库，命令如下git pull origin <master>；

【本地仓库关联远程仓库查询】git remote -v；注：一个本地仓库可以关联任意多个主机（即远程仓库）；

【本地仓库关联远程仓库】git remote add origin <server>；

【克隆远程仓库到本地】git clone -o originnn <https://XXX> ‘本地仓库主目录名’;其中-o originnn用来指定本地仓库主机名，可省略，’本地仓库主目录名’可指定也可省略；

【替换本地改动】git checkout --<filename>，此命令会使用Head中的最新内容替换本地工作目录中的文件，而已添加到暂存区Index的改动和新文件都不会受到影响；若想丢弃本地所有改动与提交，可以到远端获取最新版本历史，并将你本地主分支指向它，即命令1，git fetch origin，命令2，git reset –hard origin/master；

【更新与合并】合并改动之前，可以使用如下命令预览差异，git diff <source\_branch> <target\_branch>；

【git帮助】<http://git-scm.com/docs>；

# 工作安排

Git使用方法学会