深度学习系列培训教程

笔者：汪磊

参考：胡昊天、纪有书

前言

朋友们，深度学习技术听起来高深莫测，但大家或多或少都对深度学习有一些了解，为了各位以后能根据自身需要随时随地训练深度学习模型，顺利跟上任务进度，因此我准备和大家一起学习一下实验室现有的深度学习模型的使用方式。

由于我目前也是仅处于会使用模型的阶段，对于深度学习模型的算法与原理理解尚不透彻，在培训过程中难免会存在疏漏或是错误之处。如果大家发现我有任何解释错误的地方或是有疑问的地方请及时指出，我们一起解决！后面的培训也请大家多多关照！

本次培训预期包含 5 个部分共计 6 个课时（3 周），具体内容如下：

1、Linux 服务器及命令的简单使用（1 课时）

2、训练集、测试集、验证集的介绍及制作（1 课时）

3、深度学习下的文本分类——SVM、BERT、FASTTEXT（1 课时）

4、深度学习下的序列标注——CRF、LSTM、BILSTM、BILSTM+CRF（2 课时）

5、深度学习下的实体识别（1 课时）

最后，请大家注意本群内所传文件仅限 748 学社内部学习交流使用，切勿外传！感谢配合！

第一周 Linux 服务器及命令的简单使用

Linux 服务器的简单使用

1、前期准备

下载远程连接服务器工具：推荐使用**XShell**（命令行工具），**WinSCP**（可视化文件管理工具），便于文件上传与下载；或者可以使用两者集成的**FinalShell**（容易卡顿）。

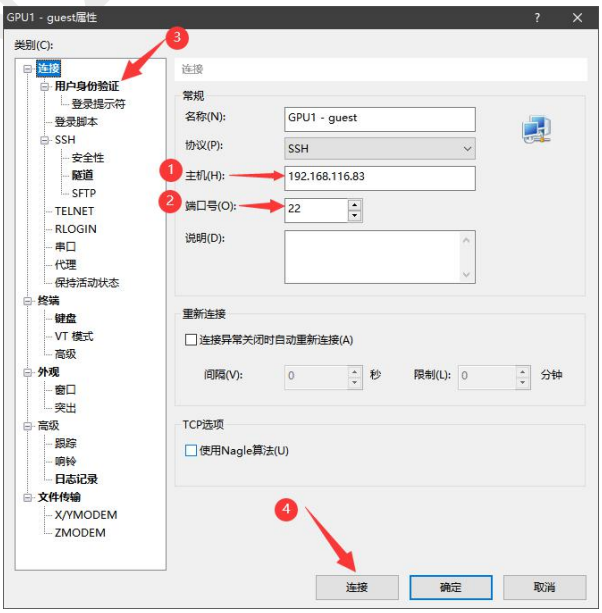
2、登录到 Linux 服务器

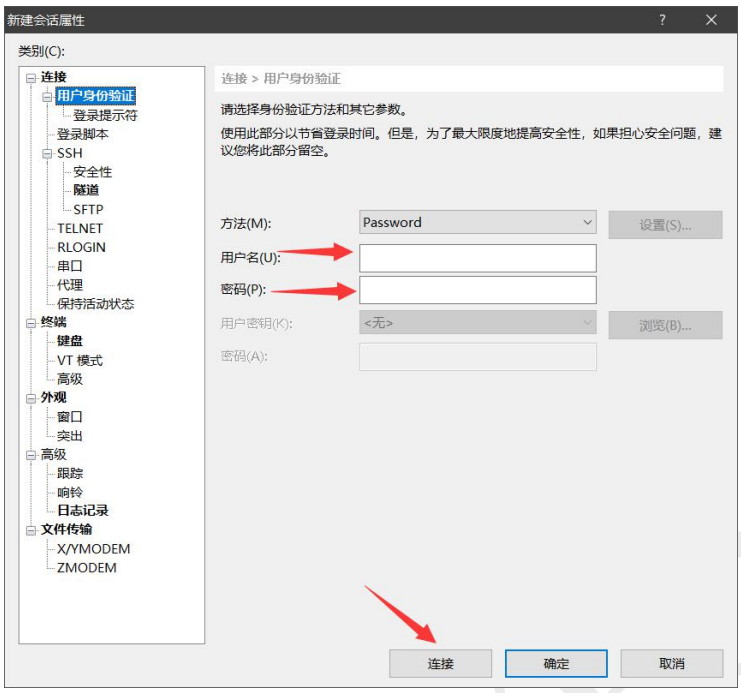
请使用以下IP与账户密码登录到Linux服务器：

姓名	GPU1	姓名	GPU2
韩广坤		刘娟娟	
王红竹		张雅茹	
何珊		朱雅丽	
范呈镭		关卓然	
董满莹		王洁仪	
宋金铃		石小龙	
王博		耿云东	
郑依晴		何洪旭	
沈宸杓	IP：192.168.116.83 端口：22	孙婧楠	IP：192.168.116.99 端口：22
鱼汇沐		翁小颖	
李诚		谢佳琪	
程兆轩		严大钰	
耿冰峰		赵连振	
陈璐鹏		胡叶	
王希羽		孙文龙	
高艺		尹一州	
用户名	guest	密码	guest

打开 cmd，输入 **ssh guest@192.168.116.83** 或 **ssh guest@192.168.116.99**

或使用 XShell 登录：新建（推荐）





Winscp 登录：新建会话

3、新建文件夹，以姓名拼音命名

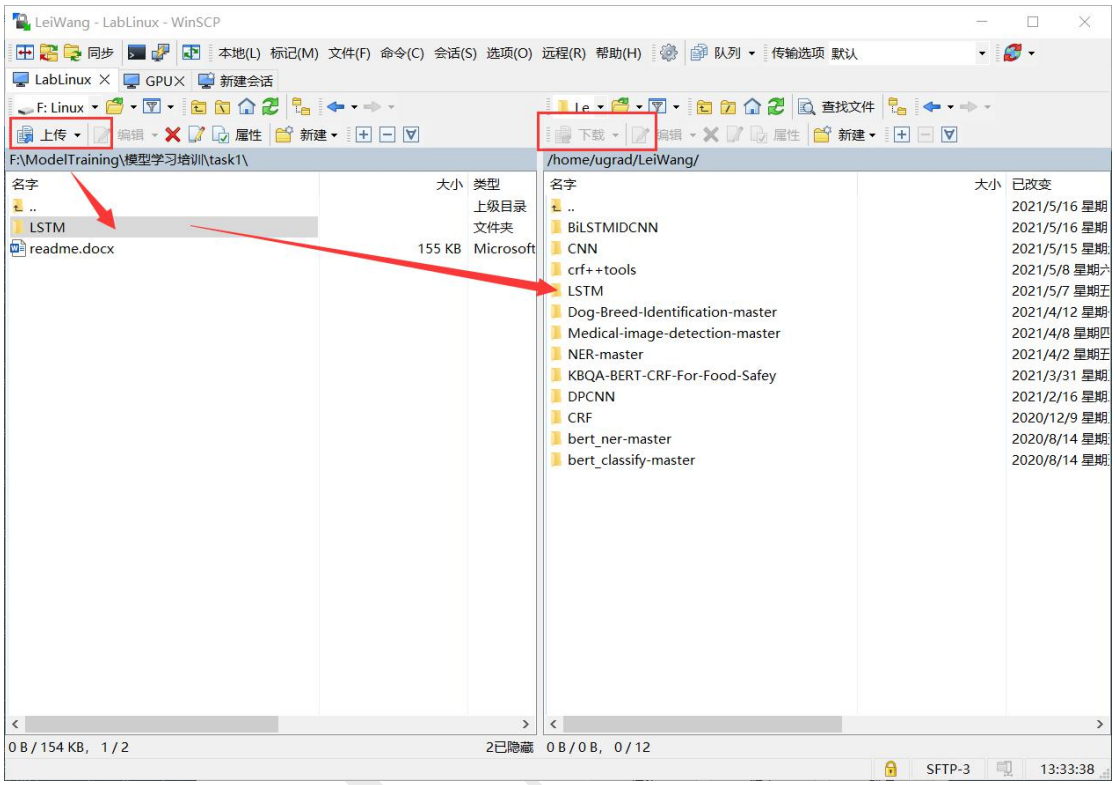
在XShell中操作 或 也可以直接在WinSCP操作

进入**748nlp2021**目录下新建文件夹，命令如下（看不懂先不要管，后面有详细说明）

```
guest@dl-OptiPlex-3020:~$ ls
748nlp2021  examples.desktop  公共的  模板  视频  图片  文档  下载  音乐  桌面
guest@dl-OptiPlex-3020:~$ cd 748nlp2021
guest@dl-OptiPlex-3020:~/748nlp2021$ mkdir LeiWang
guest@dl-OptiPlex-3020:~/748nlp2021$ ls
LeiWang
guest@dl-OptiPlex-3020:~/748nlp2021$
```

4、上传待训练的模型代码及语料（示例）

在 WinSCP 中操作：简单粗暴的方法：直接 Ctrl+C、Ctrl+V，或者用鼠标将文件从本地文件夹拖到待上传目录，从服务器下载同理。规范化的操作如下图：



5、模型训练

以下步骤在XShell中操作：

1）进入深度学习虚拟环境

服务器中已经配置好tensorflow-gpu深度学习的conda虚拟环境：**tensorflow**

可以输入 **conda info -e** 查看（**两个GPU稍微有些不一样，注意区分**）

```
guest@dl-OptiPlex-3020:~$ conda info -e
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:                    GPU1
#
django_envs              /home/dl/anaconda3/envs/django_envs
tensorflow               /home/dl/anaconda3/envs/tensorflow
tf-gpu                   /home/dl/anaconda3/envs/tf-gpu
tfjoe                    /home/dl/anaconda3/envs/tfjoe
root                  *  /home/dl/anaconda3
```

```
guest@748hd_d2:~$ conda info -e
conda: 未找到命令
guest@748hd_d2:~$ source .bashrc      GPU2
guest@748hd_d2:~$ conda info -e
# conda environments:
#
tensorflow               /home/guest/.conda/envs/tensorflow
RGcrawler                /home/dl/anaconda3/envs/RGcrawler
hht_env                  /home/dl/anaconda3/envs/hht_env
hht_env2                 /home/dl/anaconda3/envs/hht_env2
keras_lstmcnn            /home/dl/anaconda3/envs/keras_lstmcnn
```

在训练前首先需要进入虚拟环境，我们选择后面常用的名为tensorflow的虚拟环境，在XShell命令行中输入以下指令进入虚拟环境：

source activate tensorflow

```
guest@dl-OptiPlex-3020:~$ source activate tensorflow
discarding /home/dl/anaconda3/bin from PATH
prepending /home/dl/anaconda3/envs/tensorflow/bin to PATH
(tensorflow) guest@dl-OptiPlex-3020:~$
```

看到命令行前出现（tensorflow）说明已经进入虚拟环境。

2）查看 GPU 使用情况

学习初期，给大家提供的是只有 1 块 GPU 的服务器，用于简单的模型训练学习。

后期若使用安装了多块 GPU 的服务器，训练模型时若不指定 GPU 编号，程序会默认使用编号为 0 的 GPU；如果不指定 GPU，这样多人同时训练模型时只有 0 号 GPU 在满负荷工作，而其他的几块 GPU 依旧闲置。0 号 GPU 做不到啊！最后只能跑崩溃了……

nvidia-smi 查看当前 GPU 使用情况

watch -n 5 nvidia-smi 每隔 5s 查看 GPU 使用情况（退出键按 Ctrl+C）

```
(tensorflow) [ugrad@mgt LeiWang]$ nvidia-smi
Sun May 16 22:25:25 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 440.33.01    Driver Version: 440.33.01    CUDA Version: 10.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P40           Off  | 00000000:05:00.0 Off |                    0 |
| N/A   41C    P0    56W / 250W |   6923MiB / 22919MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla P40           Off  | 00000000:08:00.0 Off |                    0 |
| N/A   48C    P0    60W / 250W |   6659MiB / 22919MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla P40           Off  | 00000000:09:00.0 Off |                    0 |
| N/A   46C    P0    56W / 250W |   7671MiB / 22919MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla P40           Off  | 00000000:85:00.0 Off |                    0 |
| N/A   42C    P0    56W / 250W |   6435MiB / 22919MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+
|   4  Tesla P40           Off  | 00000000:89:00.0 Off |                    0 |
| N/A   40C    P0    57W / 250W |  10268MiB / 22919MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+
|   5  Tesla P40           Off  | 00000000:8A:00.0 Off |                    0 |
| N/A   56C    P0   139W / 250W |  22166MiB / 22919MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+
编号                                显存使用情况
+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0     24978      C   /home/ugrad/.conda/envs/opennmt/bin/python  6913MiB |
|    1     25087      C   /home/ugrad/.conda/envs/opennmt/bin/python  6649MiB |
|    2     25162      C   /home/ugrad/.conda/envs/opennmt/bin/python  7661MiB |
|    3     25303      C   /home/ugrad/.conda/envs/opennmt/bin/python  6425MiB |
|    4     25476      C   /home/ugrad/.conda/envs/opennmt/bin/python  9183MiB |
|    5     25554      C   /home/ugrad/.conda/envs/opennmt/bin/python  6939MiB |
|    5     42284      C   python                                     15213MiB |
+-----------------------------------------------------------------------------+
(tensorflow) [ugrad@mgt LeiWang]$
```

CUDA_VISIBLE_DEVICES = 1 指定使用编号为1的GPU

例如命令行输入指令：**CUDA_VISIBLE_DEVICES = 1 python train.py**，那么程序将使用编号为 1 的 GPU 进行训练。

Linux 命令的简单使用

为方便大家以后在使用过程中更好的操作 Linux 系统，我觉得有必要掌握一些 Linux 系统基本的操作命令，如果有同学后期从事运维或后端开发也会有所涉及，但是这里不会详细讲解，在目前阶段掌握以下命令已足够，具体的应用场景还需诸位日后细细探究……

参考：[菜鸟教程：Linux 常用命令学习] [菜鸟教程：Linux 教程]

1、 目录操作命令

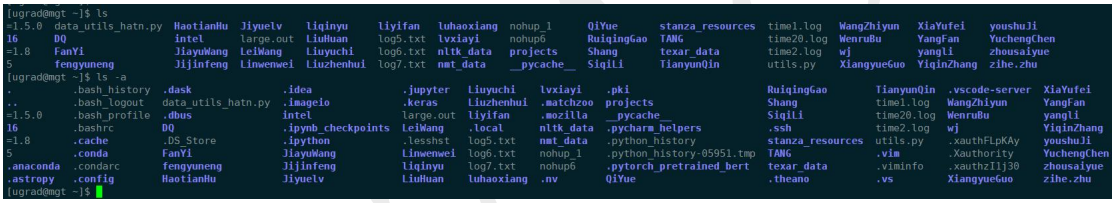
① **ls** 命令

ls 查看当前目录下所有的文件、文件夹等

ls -a 列出目录所有文件，包含以.开始的隐藏文件

ls -1 列出目录所有文件，按列排序

ls -l 除了文件名之外，还将文件的权限、所有者、文件大小等信息详细列出来



② **cd** 命令

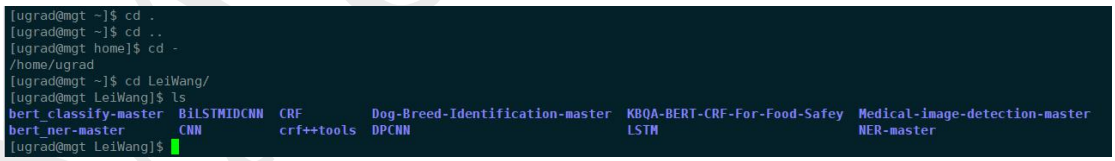
cd . 当前目录

cd .. 回到上层目录

cd - 回到上一次目录

cd ~ 回到用户根目录

cd /LeiWang 进入 **LeiWang** 目录

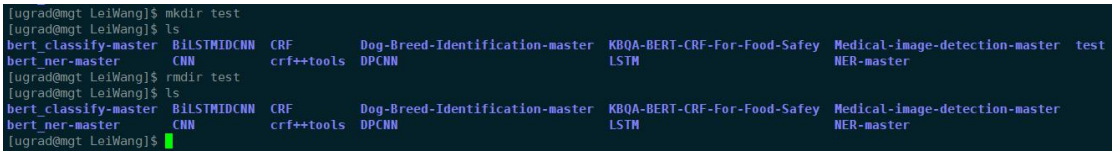


③ **pwd** 命令 查看当前路径

2、 文件夹相关命令

mkdir 创建一个目录

rmdir 删除一个目录



3、 文件相关命令

① 查看文件

见 1、① ls 命令

```
(tensorflow) [ugrad@mgt ~]$ cd LeiWang
(tensorflow) [ugrad@mgt LeiWang]$ cd test
(tensorflow) [ugrad@mgt test]$ ls
test1.txt  test2.txt
(tensorflow) [ugrad@mgt test]$ ls -l
total 8
-rw-rw-r-- 1 ugrad ugrad 78 May 16 23:38 test1.txt
-rw-rw-r-- 1 ugrad ugrad 29 May 16 23:38 test2.txt
(tensorflow) [ugrad@mgt test]$ ls -1
test1.txt
test2.txt
(tensorflow) [ugrad@mgt test]$
```

② 拷贝文件

cp test1.txt test3.txt 拷贝文件

cp test1.txt tmp/test1.txt 拷贝文件至另一个文件夹

```
(tensorflow) [ugrad@mgt test]$ ls
test1.txt  test2.txt
(tensorflow) [ugrad@mgt test]$ cp test1.txt test3.txt
(tensorflow) [ugrad@mgt test]$ mkdir tmp
(tensorflow) [ugrad@mgt test]$ cp test1.txt tmp/test1.txt
(tensorflow) [ugrad@mgt test]$ cd tmp
(tensorflow) [ugrad@mgt tmp]$ ls
test1.txt
(tensorflow) [ugrad@mgt tmp]$
```

③ 重命名/移动文件

mv test1.txt test3.txt 重命名文件

mv test1.txt tmp/test1.txt 移动文件

```
(tensorflow) [ugrad@mgt test]$ mv test1.txt test4.txt
(tensorflow) [ugrad@mgt test]$ ls
test2.txt  test3.txt  test4.txt  tmp
(tensorflow) [ugrad@mgt test]$ mv test4.txt tmp/test4.txt
(tensorflow) [ugrad@mgt test]$ cd tmp
(tensorflow) [ugrad@mgt tmp]$ ls
test1.txt  test2.txt  test3.txt  test4.txt
```

④ 删除文件

rm test4.txt 删除文件

rm -f test3.txt 强制删除文件不询问

rm -i test2.txt 删除文件但询问

rm -r tmp删除文件夹

```
(tensorflow) [ugrad@mgt test]$ rm test4.txt
(tensorflow) [ugrad@mgt test]$ rm -f test3.txt
(tensorflow) [ugrad@mgt test]$ rm -i test2.txt
rm: remove regular file 'test2.txt'? y
(tensorflow) [ugrad@mgt test]$ rm -r tmp
(tensorflow) [ugrad@mgt test]$ ls
(tensorflow) [ugrad@mgt test]$
```

4、 文本编辑器

vim test.txt

a.命令模式——>按 **ESC** 切换到该模式

b.输入模式——>命令模式下按 **insert/i** 切换到该模式

c.某行模式——>命令模式下按 **：** 切换到该模式

:q 退出

:q! 强制退出不报错

:w 保存

:wq 保存并退出

5、 后台运行

nohup python start.py >output 2>&1 & 指定输出到 output 文件（重定向）

用于在系统后台不挂断地运行命令，退出终端不会影响程序的运行，此时会产生一个进程号。

6、 进程相关

ps -aux 查看所有进程

kill -9 进程号 杀死进程

动手试一试

任务要求：

安装基本软件后，尝试连接 Linux 服务器；

连接成功后，在**根目录上创建自己的文件夹**，以**姓名的拼音字母**命名，如 LeiWang；

完成上述 Linux 的基本操作命令，记录**操作截图；**

提交方式：

以文档形式上传到群“**任务一**”文件夹，文档命名格式为：**任务一_汪磊.doc**

截至日期：

2021 年 5 月 21 日（周五）8：00 前

如有任何疑问，请大家及时在群内提出；错误之处，还请诸位见谅！