

# CPS实验室入门教程(恒哥之迷弟迷妹专用)-Alpha.v2

## Catalogue? ? ? ? ? ? ?

跳转锚点

- 1.Foreword
  - 1.1 Self-Introduction
  - 1.2 Contact
  - 1.3 Level List
  - 1.4 Guidance
- 2.Anylysis
  - 2.1 Introduction
  - 2.2 Prior Foundation (L2)
    - 2.1.1 硬件常识
    - 2.1.2 计算机基础
    - 2.1.3 基本意识
  - 2.2 Basics (L1)
  - 2.3 Advanced (L3)
- 3.Epilogue

## 1.Foreword

### 1.1 Self-Introduction

大家好，我是 广东工业大学-计算机学院-软件工程专业3班 黄泊凯，现23级（大二）。  
现在在 广工物理信息融合重点实验室(CPS)，跟着黄老师学习。撰写本教程文档是为了给后来的各位师弟师妹一个入门指导，包括对老师提供资料的一些汇总解释以及个人经历看法来作为借fan鉴mian参jiao考cai。

### 1.2 Contact

联系方式	Content
Phone/WeChat	13600323338
WeChat_name	ILVACM
Email	3347620766@qq.com
Address	综合楼602

### 1.3 Level List

Level	Specialty
1	生医工 生工 制药 土木学院全专业
2	信息工程 机电工程专业
3	计算机学院全专业 自动化学院全专业

L1 < L2 < L3（数字大的包含数字小的）

PS:本列表后期会继续维护，根据实际情况来调整。

## 1.4 Guidance

- 由于本人就读于计算机学院，所撰写教程也总体偏向于电子信息。但很明显的是，咱实验室家大业大，所涉及的方向也非常多，包括但不限于：自动化、硬件机械、医工结合等等，就算是电子信息领域也难以实现基础入门全覆盖。所以本教程更多的是提供一个思路方向(因为其实有很多我自己在写这个文档的时候都wo不shi明fei白wu，红豆泥私密马赛+90°鞠躬)，给各位找到一个切入点便于各位上手熟悉。大家根据自己的实际情况灵活变通，人是活的，不要一根筋。
- 另外，本教程主要面向后来的师弟师妹，应该大抵都是在黄老师名下。黄老师主要专注与人工智能及其结合应用研究，不过各位师弟师妹并不一定都来自 计算机学院（还有 信息工程学院 生物医药学院 ..... 甚至 土木与交通工程学院 都有可能），所以本教程会做一个 学习等级 划分，这个等级是用于标识某个要点在电子信息领域的专业程度(基于个人的主观判断)，像来自生物医药学院生医工专业的同学，就不需要特别深入的了解一些专业的计算机知识，有需要可以找实验室计算机学院的同学乃至师兄师姐来帮忙。
- 如果各位能够看到这份文档，那么大抵至少和黄老师面谈过一次了，关于就业与升学以及具体的发展方向，后面"Epilogue"部分还会有一些个人的看法见解供各位参考，如果一时半会想不明白可以先做入门学习，然后上手一些简单的项目配套，一边做一边调整。另外，截止到2024/11/25，目前本教程文档应该处于内部测试阶段(Alpha)，里面会有很多错漏的地方，大家看的时候如果觉有问题直接过来压力我就行（求轻喷），线上线下都行。-

## 2.Anylysis

- 在正式开始教程的技术入门说明之前，关于需要准备的东西这里先提来说明一下。好吧其实就是有一台笔记本电脑，依照目前的形势来看，我相信大部分上大学的人，也就是看文档的各位，理论上家庭都是能够担负得起一台笔记本电脑的，如果不幸你属于例外情况，请立刻联系辅导员等（勤工俭学 /寒暑假兼职 /... ）。如果你因为有电子竞技方面的需求(dog.jpg)而已经入手了一个游戏本，那你已经可以跳过这段话直接开始看教程文档的技术内容说明了。
- 如果到现在还没有自己的笔记本电脑的或者准备买个新电脑的，建议购买游戏本，这里有一些品牌推荐：[/宏碁(Acer) /华硕(ASUS) /神州(HASEE) /机械革命(MECHREVO) /七彩虹(COLORFUL) ]。上述品牌笔记本产品都是搭载Windows系统，同时游戏本产品的品控和性价比都很不错。至于为什么推荐购买游戏本这里确实也有很多说法，但是由于比较繁琐复杂，暂时先不赘述，后面有需要的话在更新教程版本的时候另开一节说明。
- 补充一点，建议Windows系统的原因是，这边大部分人（包括我和师兄师姐）都是使用Windows系统的产品，出现问题能帮得上忙，我用不起水果机所以MacOS用户爱莫能助。其他师兄师姐不一定是用不起，但是也是没有用。
- 另外，如果各位师弟师妹有 Linux\ ACM\相关基础，本教程文档的很大一部分甚至全部都可以直接跳过，直接看看最后（如果有心情看的话），然后就可以上手做任务了。

## ！使用说明！

- 开始之前，讲一下如何阅读使用本文档。
- 各位按照自己专业根据前面"1.3 Level List"部分中对应的学习等级来参考，注意等级数字大的包含小的，比如 计算机类专业 或者 软件工程专业 对应 L3(Level-3)，那么所有不大于L3标识的部分都要涉及参考。
- 阅读文档的时候。注意！文档中大部分是思路方向的讲解和学习要点的提示！ 虽然当中带有一些讲解，但是只是起到一个引导辅助的作用
- !!! 纸上得来终觉浅，要知此事须躬行!!!
- 每一小节看完之后，自行根据要点查找资料学习了解。关于如何查找资料，见后文"2.1.1 基本意识"章节内容。

## 2.1 Prior Foundation

- 现在正式开始对入门技术内容进行说明。在正式讲基本知识概念之前，首先需要给大家铺垫一些基本的常识概念。
- 能读到本教程文档的各位师弟师妹大概都是跟着黄老师吧，黄老师主要做人工智能结合研究的。
- 既然是涉及人工智能结合研究，一些最基本的常识概念是需要知道的。（主要针对计算机方面）

### 2.1.1 基本概念(L1)

- 所谓基本常识概念，其实就是在以后研究开发的过程中，一些基本的意识。
- 这些所谓的基础更准确的来说应该叫电子信息领域的常识，就好比去菜市场买菜知道怎么挑菜，什么样的菜叫什么名字（其实我不会，我这就是属于没有生活常识的人，我没去过几次菜市场 红豆泥私密马赛 x2 ）。
- 换一个说法就好比 你打 英雄联盟(LOL)端游排位，前期对线要补刀，经济优势要拉开，辅助要保自家AD 同时在有利条件情况下敢于先手开团... 然后一般情况下没有AD贴脸单挑上单的等等的。
- 这些就是最基础的意识，更高级的不展开（有兴趣的同学可以私聊学长我深入交流，学长我玩的一手垃圾AD，等着被带）。

那么接下来正式开始讲解

!! 那么接下来正式开始讲解 !!

开发环境搭建 存储分盘\ 文件管理\ 360\ Microsoft office 365\ 基本IDE( VScode VS Pycharm-Jetbrains? )  
readme\官方文档  
资料收集查找 > CSDN/ GitHub/ 菜鸟教程/ AI  
工具软件（专业的事）  
官方、纯净  
镜像源、默认源（VPN 魔法上网）

??????

- 电子信息基础，主要针对电子信息类专业的同学。理论上来说，这些基础知识应该在大学入学之前，或者至少在入学第一个学期全部了解明白（最好是熟悉） • 不过我相信各位其实多少都懂，毕竟我们是信息时代的原住民，只不过不会特别提出来专门学习，特别是如果在座各位有去 B站 知乎 等一些网站看专业的产品测评习惯的话，基本上瞟两眼就可以跳过了。我这里主要是做一个提示，有不会的大家自己稍微上网查查就懂了。

.....专业导论  
????????

## 2.1.x 硬件常识(L2)

硬件常识这里主要讲解电脑，其实标准名称应该叫做个人微型工作站（我是这么觉得），这也就是常说的 PC端 或者 本地端(local)。因为如果要全面了解电子硬件的常识，需要系统的学习包括嵌入式硬件在内的一整套体系，对于普通学生来说本科都不一定搞得定（各位要是有意向可以自行尝试）。这里算是快速入门，所以就从最常见最简单的地方开始，前面也说了文档最核心的是起到一个引导的作用。

接下来是具体讲解，主要关注其中的概念和当前市面上主流产品以及技术趋势。

### 2.1.1.x 计算机基本组成

#### 2.1.1.x CPU

- 首先是电脑中最核心的组件，全称"中央处理器"，英文"**Central Processing Unit**"，缩写就是我们常说的 **CPU**。他是电脑中最重要的部分，它的参数决定了一台设备主要性能的上下限，不管是笔记本（轻薄本/ 游戏本/ 全能本/）还是台式机乃至服务器。本节我们主要关注桌面端处理器，也就是台式机和笔记本电脑。极少数使用移动处理器的产品，比如微软(**Microsoft**)自己出的采用 高通骁龙(**Snapdragon**) X Elite 的 **surface**系列（目前最新的啊）等此类产品我们暂时不做讨论，学有余力乃至财力的同学可以自行了解。

- 当前在处理器领域，包含桌面端和移动端乃至嵌入式，主流厂商有 英特尔(**Intel**)/ 超威半导体(**AMD- Advanced Micro Devices**)/ 美国高通公司(**Qualcomm**)/ 安谋国际科技(**ARM- Advanced RISC Machine**)/

- 这里大家注意，我介绍的是芯片架构设计品牌厂商，而不是实际代工生产的厂家

- 诸如大家所熟知的 台积电 是代工生产厂家，不是设计厂商，这里有个例外就是截止至本教程文档撰写的时间，英特尔(**Intel**)长期以来一直是自主设计生产一条龙的。

.....

- 第一家我相信大家都知道，毕竟英特尔(**Intel**)是最老牌的厂商，曾经也一度是这个领域统治者，不过现在不如当年了，根据一些网络上的评价我比较喜欢叫他 牙膏厂（产品更新迭代喜欢挤牙膏）。但是后几家可能不一定很清楚，可能听说过他们的英文缩写（比如上网网购的时候，大抵会有人说用**AMD**锐龙系列**CPU**性价比很好，不要买牙膏厂的产品），但是要问他们的英文全名乃至中文名称，或者更进一步说他们的主要市场业务及其区别之类的，估计很多人一下子就懵了，当然各位查一查其实就知道了，只不过可能平常忽略了这些细节而已。

- 先点一下第二家 **AMD**（这是缩写），英文全名"**Advanced Micro Devices**"，中文叫"超威半导体"。从英文全名其实一眼就可以看的明白这家公司的主营业务（微型高集成电路）及其面向市场（**CPU/ GPU**）。**GPU**，也就是显卡方面放一下，后面讲，这节主要是介绍**CPU**。**AMD**经过前几年的技术革新（主要因为有一位硅谷芯片大佬的加入），更新了其处理器方面拳头产品的架构设计，另外通过与台积电合作采用更加先进的制造工艺（前面说了**Intel**是极其特殊的研发设计生产一条龙，生产也要靠自己，还是比较费时费力的）。产品整体性能参数迅速追平**Intel**，甚至在部分方面实现超越，另外通过比**Intel**更加亲民的定价策略，使其在大部分场景中性能差距并不明显的情况下，具有了极高的性价比乃至极强的市场竞争力，所以各位以后采购相关设备涉及到桌面端处理器的时候，现阶段**AMD**确实不失为一个优秀的选择。具体技术细节我就不展开了毕竟这里只是简单入门，有兴趣的同学自行去B站观看相关的专业**CPU**测评讲解，里面说的比我更加详细且专业。

#### 2.1.1.2 GPU

#### 2.1.1.3 内存(RAM)

外存(SSD /HDD /M2 )

IO接口

外设(键盘、屏幕)

## 2.1.x 计算机基础(L3)

名称前后缀 代号 缩写

计算机网络概念(局域网、广域网)

破解

SSH FTP HTTPS.HTTP Serial Shell

操作系统(Windows /Linux /MacOS /Android /iOS ...) 系统安装配置(刷系统)

系统结构(驱动、环境变量、注册表、任务管理器、磁盘管理)

系统基本参数(Windows /linux )-IP.远程桌面.

程序编译运行

系统架构-前端后台数据库硬件人工智能

## 2.2 Basics

- 经过前面基本常识的说明,各位就算是零基础,也至少有一个基本的认识了。
- 要是说记不住很正常,看这份文档的大抵都是零基础,一下子塞这么多理念指望立刻理解吸收是不现实的(而且全是文字,本来是想找些图的,这样可以直观很多,但是懒,红豆泥私密马赛x3)。记不住可以记个笔记,日后重复复习或者找相关资料进一步理解(有关资料查找见"2.1.3.2 资料收集查找?????")
- 所谓熟能生巧,前面的那些基础都是在日常学习工作过程中逐步积累并建立的一个属于自己的体系理解,那么基于自身的体系自然会形成一套自己的发展路线思路,其实我前面关于基础的学习路线的说明也是主观的基于我的知识体系一个看法观点,所以才说本文档主要提供参考和借鉴思路,并不是真的说教学(我何德何能,没那个能耐)。
- "2.2 Basics (L1)"是全文档最主要的部分。假设各位现处于大一且没有其他事务干扰的情况下,只有上课和作业(这里的事务指的是还有其他工作,比方说各位除了 实验室 又是 ACM集训队成员,或者说又兼任有 团委 等这种行政工作)。那么在每天晚上(包括周末)空一个晚上的情况下(19:00-23:00),最久大约三周(21-Day)能够完成。其他不同情况参考这个来安排时间。
- 那么进入正题(废话多是这样的,感谢各位的耐心,红豆泥私密马赛x4),由于本教程是为了给各位刚跟着黄老师的来自不同专业的师弟师妹快速上手能够进入 工程项目 或者 论文实验 当中担任工作,所以更多的是一种速成。加入各位学有余力想进一步提升的话,欢迎过来和各位师兄师姐讨论(愿意找我也可以哦, doge.jpg)。

### 2.2.1 算法概论基础(L3)

(非电子信息类专业的同学跳过本节) → 传送锚点

- 在开始一切具体的技能说明之前,我觉得有必要把算法这个概念先提出来,就算时间来不及去具体学习,也应该树立一个概念。
- 算法(Algorithm),简单来说,就是解决某一个问题的一种方法或者说思路。
- 打个比方就是:各位高中做数学题的时候,一道题的解题方法就实质上是算法,注意是解题方法不是具体的数学公式。对应现在的程序设计,也就是打代码,那么算法就是你解决问题的流程思路(画成一张图就是流程图),然后通过代码来实现你的算法思路。
- 务必注意先后顺序,算法是一个程序的灵魂,是一个软件工程项目的骨骼(一定程度上应该也可以等效为框架),有了框架骨骼才好往里面填充血肉(具体开发,打代码什么的)。别跟我杠软体动物!!!软体动物长不大(工程项目也做不大,论文也做不好)

- 这一个方面的具体学习路线话是很简单的。倘若各位拥有ACM相关基础（包括OI生→有牌最好，学习过是不作数的\参加过相关赛事→打到ACM新生赛决赛或者更好的比赛才算\！！！尤其有广工校ACM集训队服役经历的！！！），那么这一段后面的瞄两眼复习一下就可以过了。如果各位没有相关基础也没关系，零基础就从头开始，没有谁是天生就会的，都是从零基础走过来的，只不过有先有后罢了。
- 插一句，零基础然后不熟练的最好反复复习，熟能生巧！！！
- 
- 按照学习先后顺序，各位自行查找资料学习相关概念：

1. 时空复杂度\
2. 暴力枚举\
3. 模拟\
4. 递归\ 迭代\
5. 分治\ 二分\
6. 排序\
7. 贪心\
8. 前缀和\ 差分\
9. 倍增\
10. 构造\
11. 搜索\ DFS\ BFS\
12. 简单数据结构（链表 栈 队列）\

2.2.2 基础程序语言( L1\ L2\)

- 经过上一节的讲解，这里默认各位已经有了基本的算法概念体系，没有的没关系，回头去复习就好了，最差的也不过就是熟能生巧，倘若各位要是有隐藏天赋，大可以觉醒开挂直接速通。
- 需要学习的基础语言有：

语言	种类
C/ C++	编程语言
Python	编程语言
MarkDown	标签语言

- 做一个补充说明：非电子信息类专业的 C\C++ 这两个跳过暂时可以不学（其实是一门语言）；Python 语言上手实验大约是5个晚上（也就是五天），最长不要过一周，超过了那就肯定有问题，速找师兄师姐帮忙；Markdown同理，只不过时间更短，大约30min不超过1h。
- 下面关于语言的学习做一个具体说明：这里是以尽可能快速入门上手为主要目的的文档，我们不需要很深入学习语言的内部机制，我们只需要了解其语法（会用就行）。
- 下面列了每种语言的基础要点，学会了就可以了。只要达到给一段代码能够看明白，给一个需求能写出相应的代码就算过关（什么方法都行，查资料，借chao鉴别人的，问AI都行，只要程序起作用效果过关）。

2.2.2.1 C/C+(L2)

1. 代码语法结构
2. 变量
3. 常量



4. 数据类型
5. 运算符
6. 逻辑分支
7. 循环
8. 函数
9. 数组
10. 指针
11. 位运算
12. 结构体\ 联合体\ 枚举\
13. 类\ 对象\
14. 头文件库

### 2.2.2.2 Python(L1)

1. 代码语法缩进规则
2. 变量
3. 常量
4. 数据类型
5. 运算符
6. 逻辑分支
7. 循环
8. 函数
9. with-as 语句
10. 类\ 对象\
11. 依赖库\ 依赖模块\

### 2.2.2.3 Markdown(L1)

这个就不写了，30min多就能上手的，真的。

### 2.2.3 人工智能基本概念(L1)

- 恭喜各位坚持到这一步，离胜利已经不远了，大家再坚持一下。
- 人工智能基本概念并不是大家可能想象的那种需要很深入学习深奥晦涩的东西。不需要啊，再强调一下，文档的目的是让各位最快速度入门上手，建立一套基本的体系概念。如果各位有心进一步深入学习，搞定基本要点之后可以自行学习理解（大力鼓励且支持）。
- 再解释一下学习思路：由于阅读教程文档的各位大概率还是大一的（用咱们计院的话就叫新东西），不是的话基础可能也比较薄弱（不然也不用看这份文档不是嘛,dog.jpg）。所以这部分关于人工智能概念的学习其实更加接近于 科普，大家只要对基本概念有一个形象直观的理解就行了，每天花点时间看看就好，两三天就差不多了其实。因为根据个人经历，大家真的上手接触实际工作的时候，这里学习的东西根本不够看，基本上都是边做边学，或者根据自己的发展方向针对特定领域进行深入学习的，所以先期只需要像科普那样子有一个简单的概念就可以了。
- 这里列出一些基本概念，大家差不多了解就行，不需要纠结：

1. 强化学习
2. 机器学习
3. 深度学习
4. 有监督学习\ 无监督学习\ 联邦学习\
5. 神经网络\ RNN\ CNN\

- 6. 梯度下降\
- 7. 反向传播\

## 2.2.4 python环境搭建(Anaconda) (L1)

- 最后一个基础要求，行百里者半九十。到这里就会轻松很多啦！
- 本节学习有两种方法：
  1. 继续阅读本文档，自行查找相关资料学习理解
  2. 联系一下带教师兄师姐，让师兄师姐手把手教学（各位第一次来实验室找老师的时候肯定会加的）
- 如果选择第二种，那么直接和带教的师兄（或者师姐）讲清楚，要教这些：教教怎么 通过 **Anaconda** 搭建python环境（不要用 python IDEL 之类的这种），以及 **Anaconda**软件 的基本使用，然后再教如何部署运行yolo模型。
- 如果找师兄师姐来教，建议和师兄师姐说，采用YOLO-v5，这是开源版本中最新最好的，也便于教学展示；截止至本版本(Alpha.v2)教程文档的撰写，YOLO系列模型更新到第11代→ yolo-v11，最新的不是不行，主要是不开源，另外其实内部机理大差不差没什么所谓的。
- 学完上面那些，各位可以直接跳过"2.2.5 工程模型部署运行-yolo/fcn"部分到"2.2.6"部分。
- 如果选择第一种，首先无论如何都要赞扬各位的毅力与独立自主的精神，这将在后面的学习研究过程中带来很大帮助。
- 那么请继续按顺序阅读本文档。
- **anaconda**的学习主要就是三样：安装\ 配置\ 使用\
- 学习过程中注意理解 **anaconda**软件 的作用，因为前期在给人讲解的时候我发现很多人完全不理解为什么要使用**anaconda**，尤其是他的特点优势
- 建议去CSDN搜索相关教程（关键词：**anaconda\ 安装\ 配置\ 使用\**）
- 下面提示一些要点：

### •Install

先通过 官网<https://www.anaconda.com/download> 下载最新版本安装包  
再通过安装包安装

**!!!安装过程中注意安装位置!!!**，也就是常说的安装路径

之所以要注意安装路径，其实只是因为天天说的存储空间的问题，也就是传说中的C盘爆炸。因为分盘以及工作区的缘故（关于工作区，详情见"2.1.3.x"），C盘不一定有很充足的空间，另外**anaconda**软件在运行开发过程中会占用大量存储空间，所以需要根据实际来选择安装路径，并不是说不能默认安装。

### •Config

安装完成之后，主要关注两个地方的配置：

1. 依赖库安装下载源(**conda\ pip\**)，就是传说中的镜像站，一般默认推荐 清华大学镜像站，当然各位要是已经明白如何科学上网那这一条无所谓。
2. 环境默认配置（也可以叫安装）地址，就是当新建一个Python环境的时候，这个python解释器及环境所配置的库或者包的默认安装位置。有时候会出现，软件安装路径与环境配置路径不在同一个根目录的情况



•Use

软件的具体使用教程会写的很清楚  
主要是注意，建议以命令行界面为主(Anaconda Powershell Prompt\ Anaconda Prompt\)  
!!! >>>不要使用GUI界面(Anaconda Navigator)<<<!!!  
.  
使用命令行界面，主要了解下面几种命令及其参数格式（表格中是命令示范，请根据实际情况编辑命令）：

Command	Example
config download source (add)	conda config --add channels <a href="https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgsrc/free/">https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgsrc/free/</a>
create environment	conda create -n python311-example python=3.11 -y
delete environment	conda remove -n python311-example
install by conda	conda install torch=2.5.1 -c pytorch -y
install by pip	pip install torch=2.5.1
uninstall by conda	conda uninstall torch
uninstall by pip	pip uninstall torch
back up environment	conda env export > backup.yml

补充强调一点，创建环境的时候，!!!环境命名最好遵循一个固定的格式!!!（可以按照自己的，我的是 **python**版本号+连字符+环境使用目的），类似程序设计中变量名命名的命名规范

2.2.5 工程模型部署运行( yolo/ fcn/) (L1)

- The last one !!! 做完就可以上手了哦。
- 不同于"2.2 Basics"部分前面的内容，前面都相当于是基础知识，这一部分更加倾向于实际应用以及一种思路流程的示范，并非某个具体的知识要点的学习。
- 这一节的目的是让大家直观形象的感受真实人工智能领域中的计算机实验流程如何开展执行，增进大家对实验或者项目的流程思路的理解。
- 这一节以示范为主，我们就以yolo系列目标检测模型中的第5代版本(yolo-v5)来示范，fcn是经典的语义分割模型，学有余力的同学可以自行了解尝试。

- 模型的整个部署流程大体也是分三步：下载\ 配置\ 运行\
- Download

通过 官方网站[GitHub](#) 或者 镜像网站[Gitee](#) 下载代码仓库
- Config

根据 项目介绍说明(Readme.md)以及官方文档，通过Anaconda搭建python环境。另外根据代码以及说明下载相应的参数文件
- Run → Detect

通过IDE(VScode\ Pycharm\ )链接激活环境，运行detect.py，让其调用模型执行推理并输出处理效果（图片目标检测识别）

!vvvvvvvvvvvvvvvvvvvvv!

-----  
|看到这里，所有的必要基础其实就已经完成啦!!! 如果你实在心急，太想进步了(doge.jpg)。...|  
|你可以立即联系你的带教师兄师姐，或者是你所在 工程项目组 论文实验组 的负责人师兄师姐安排。|  
|然后直接跳到"3.Epilogue"部分，写了一些其他的补充，这些补充虽然不是些很直接的知识概念·|  
|但是有鉴于进到实验室之后的经历和一些个人的主观理解，有必要写一些理念和思路的补充作为参考·|  
|如果你没有那么急，或者是师兄师姐那边一时半会没有一个具体的安排（很正常啊，后面会说的）...|  
我还是建议你继续向下按顺序阅读。.....

!AAAAAAAAAAAAAAAAAAAAA!

2.2.6 文档撰写(optional) (L1)

这里关于一些基本技能再做一个补充，都是10min左右就可以熟悉的。  
由于各种各样的原因和需求，我们在后面研究以及工作中，必定会需要撰写各种类型的文档或者文章（主要是论文）  
下面列几种常见的文档类型，大家上网搜索一下模板格式了解即可：

- 1. 论文
- 2. 项目报告文档
- 3. 技术文档
- 4. 项目申报
- 5. 科研计划申报

2.2.7 服务器(optional) (L3)

另外在补充一个，服务器。虽然大家可能自己有一个带了一张显卡的游戏本，但是不可能真的用自己的电脑来训练模型。  
游戏本里的那张显卡虽然能够方便各位在本地做一些简单的验证测试，但是正如其名称，这玩意主要还是让各位在电子游戏世界中畅游。  
真要是用这个来训练模型，电脑寿命估计少说得打个对折。  
那么这个时候，就需要用到专门的设备设施，也就是 服务器(server)。

我们这里以 **AutoDL** 为例，这是平常租用算力资源最常见的平台服务选择。同样，大家阅读官网 <https://test.autodl.com:33443/home> 上的说明文档学习，下面列几个要点。

我们主要就关注三个：连接\ 配置\ 使用\

### Connect

在 **AutoDL** 官网 选好配置，下单，等待平台配置激活。

实例配置完成并激活之后，按照平台提供的 **SSL连接凭证**，通过 **VScode** 或者 其他 **SSL** 终端工具 连接服务器。

### Config

连接成功之后，通过终端环境配置运行环境。

注意这里的配置针对的是模型的运行环境，前面平台对实例的配置是系统本身以及驱动等底层设置

包括 **anaconda\ python** 依赖库\ 预训练参数文件\ 等等

### Use

通过命令行终端界面操作系统，运行模型，收集运行结果

## Linux

### / Shell

## 2.3 Advanced (L3)

- 经过上一节的基础讲解入门，大家已经有一个基本的思路和理解了（不行的话，比如忘了回过头复习复习就是了，熟能生巧）。
- 但是毕竟说了，这是一个以速成为主的入门文档，主要是为了让大家快速上手。
- 随便读过几本武侠小说或者网文（比如斗罗大陆，这本放到现在应该很老了吧，**doge.jpg**）都知道，基础不牢地动山摇，基础不行上限高不到哪里去的。
- 但是由于各位出身的学院以及专业不同，不同情况要具体分析，我自己连计算机都学不明白更不指望来教人了。
- 所以，这里根据我个人的发展方向列出一些学习思路，给大家参考，大家结合自己的专业实际来调整（包括每个要点的学习等级标定也是）。

论文实验 项目思路

算法进阶（ **ACM/ 硬件算法/ 人工智能基本算法**）

语言进阶（ **Java/ SQL/ 前端三剑客-JS.CSS.HTML/ Latex/ CUDA/ ...**）

正则表达式

**Node.js**（ **npm/ ...**）

应用进阶（ **Git/ 数据库/ 编译器.CMake.Maven/ 油猴脚本/ NAS/ ...**）

开发进阶（ **脚本/ 工程/ 工具/ 软件绿化(portable)/ ...**）

**Pytorch/ Tensorflow/ Transformer/ ...**

**/openCV /NLP /SLAM**

## 3.Epilogue

写在最后

个人经历 重应用，要点突破

工位 过来做

自主自动，不会没事干

方向选择 行政× 学术√

敢于问

## References

...