[4.28-5.8（准备数模校赛）](#header-n40)  
 [5.9python语法简单梳理](#header-n83)  
 [5.10](#header-n97)  
 [监督学习](#header-n98)  
 [回归是什么？](#header-n101)  
 [分类是什么？=》标记问题](#header-n103)  
 [搜索=》推荐系统](#header-n107)  
 [序列学习是什么？](#header-n109)  
 [无监督学习](#header-n111)  
 [离线学习与环境互动=》强化学习](#header-n113)  
 [5.10](#header-n115)  
 [深度学习框架](#header-n117)  
 [配置框架](#header-n123)  
 [5.11（数据操作）](#header-n129)  
 [5.12（数据预处理，NVIDIA比赛了解）](#header-n149)  
 [[2:5] [1:5:2]](#header-n151)  
 [读取数据集](#header-n159)  
 [处理缺失值（fillnan，dummies这个不太会用）](#header-n167)  
 [5.13](#header-n172)  
 [降维](#header-n173)  
 [微积分](#header-n178)  
 [5.14（写计网实验报告+小程序比赛去了）](#header-n186)  
 [5.15](#header-n187)   
 [5.16-5.19（云服务器知识的学习）](#header-n195)  
 [5.20-5.23](#header-n204)  
 [线性回归的从零开始实现和简洁实现（基于框架再做了一遍）](#header-n208)  
 [图像分类数据集（框架内置函数下载数据集并读取到内存中）](#header-n217)  
 [softmax回归的从零开始实现和简洁（框架）实现](#header-n218)

本周工作：深度学习线性回归部分

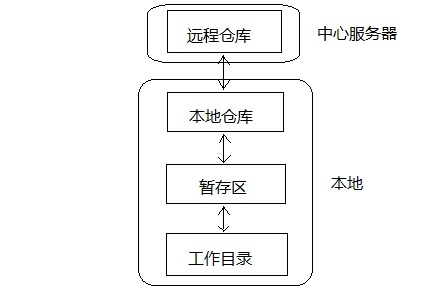
下周工作：深度学习多层感知机

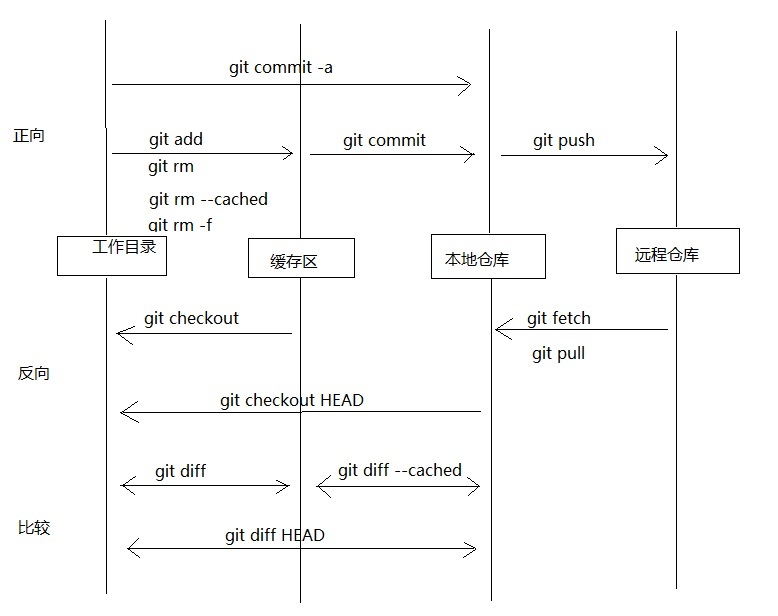
待学：

开源之夏

1.云服务器，docker

* 镜像你可以把它看成Java中的类，而容器可以看做是类的实例化对象。
* 一个类可以有多个对象，同理，一个镜像可以有多个容器。
* 容器是基于镜像创建的，即容器中的进程依赖于镜像中的文件
* docker提供了一个很简单的机制来创建镜像或更新现有的镜像。用户甚至可以从其他人那里下载一个已经做好的镜像直接使用。（镜像是只读的，可以理解为静态文件）





2.虚拟机

虚拟机（VM）是一种创建于物理硬件系统（位于外部或内部）、充当虚拟计算机系统的虚拟环境，它模拟出了自己的整套硬件，包括 **CPU、内存、网络接口和存储器**。通过名为[虚拟机监控程序](https://www.redhat.com/zh/topics/virtualization/what-is-a-hypervisor)的软件，用户可以将机器的资源与硬件分开并进行适当置备，以供虚拟机使用。

**为什么要使用虚拟机？**

**服务器整合是使用虚拟机的首要原因**。部署到裸机时，大多数操作系统和应用部署都只会使用少量的物理资源。通过虚拟化服务器，您可以在每个物理服务器上设置大量虚拟服务器，从而提高硬件利用率。

这样您就无需购买额外的物理资源（例如硬盘驱动器或硬盘），也不用压缩数据中心对电能、空间和冷却能力的需求。通过支持故障转移和冗余，虚拟机提供了额外的**灾难恢复选项**，而这以前只能通过增加硬件才能实现。

虚拟机可以提供一个与系统其余部分隔离开的环境。这样，**无论虚拟机内部运行什么，都不会干扰主机硬件上运行的其他内容**。

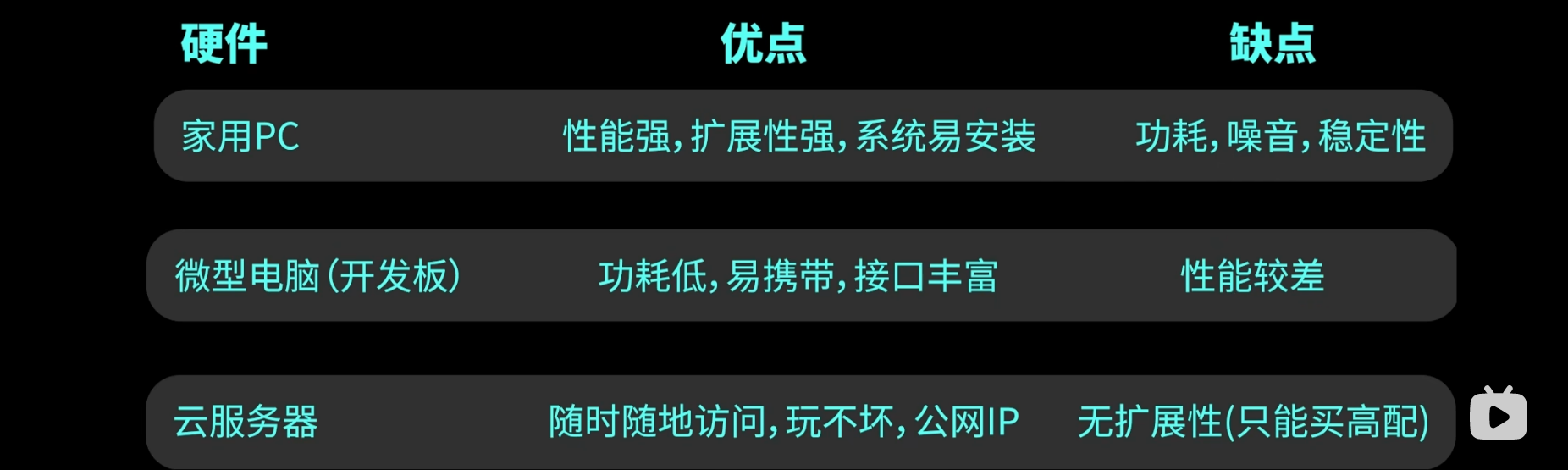
由于虚拟机处于隔离状态，因此堪称是测试新应用或设置生产环境的理想之选。此外，针对特定的进程，您还可以运行单用途虚拟机。

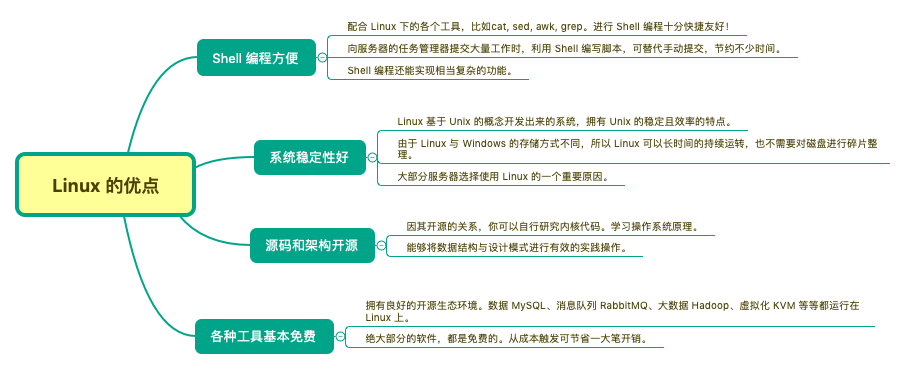
3.linux系统

debian->ubuntu()

redhat(红帽，收费)->centos

微型电脑树莓派（接口多，支持linux）





**SSH**是一种网络协议，用于计算机之间的加密登录



4.Nginx (engine x) 是一个高性能的 HTTP 和 反向代理 web服务器

正向代理是代理了客户端，而反向代理则是代理服务器端

**imagenet数据集**

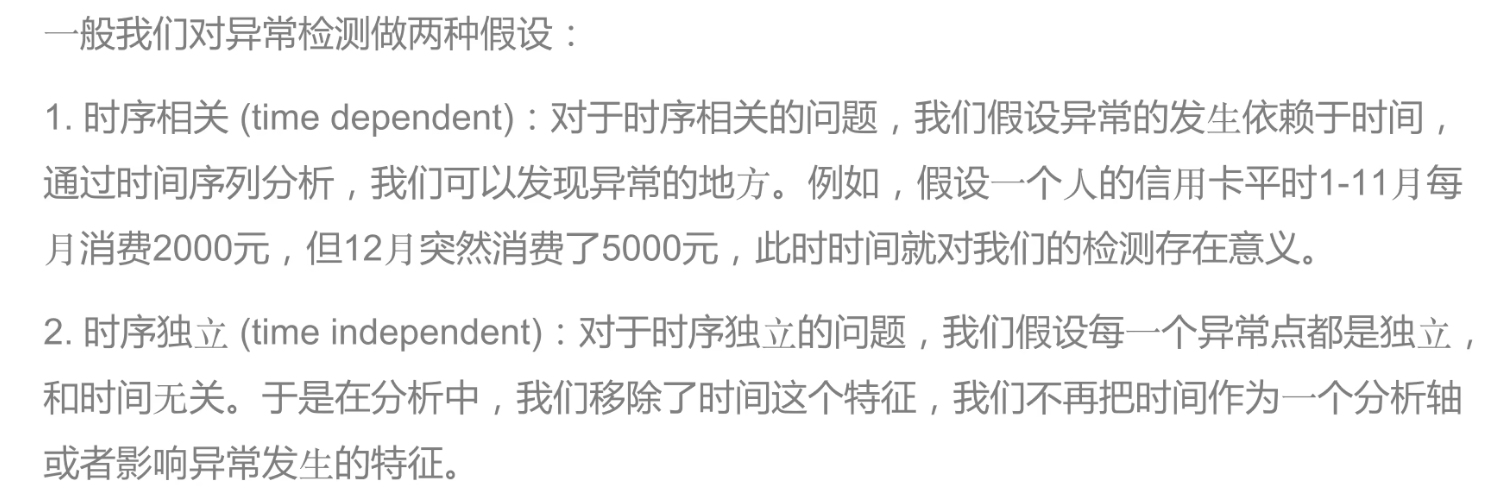
### 4.28-5.8（准备数模校赛）

异常检测：单点（达到某个界定值），上下文（特定场景下出现异常），集体异常（集体情况发生改变）

难点：1.无标签，监督学习不适用 **2.需要区别噪声和异常点，比较抽象**

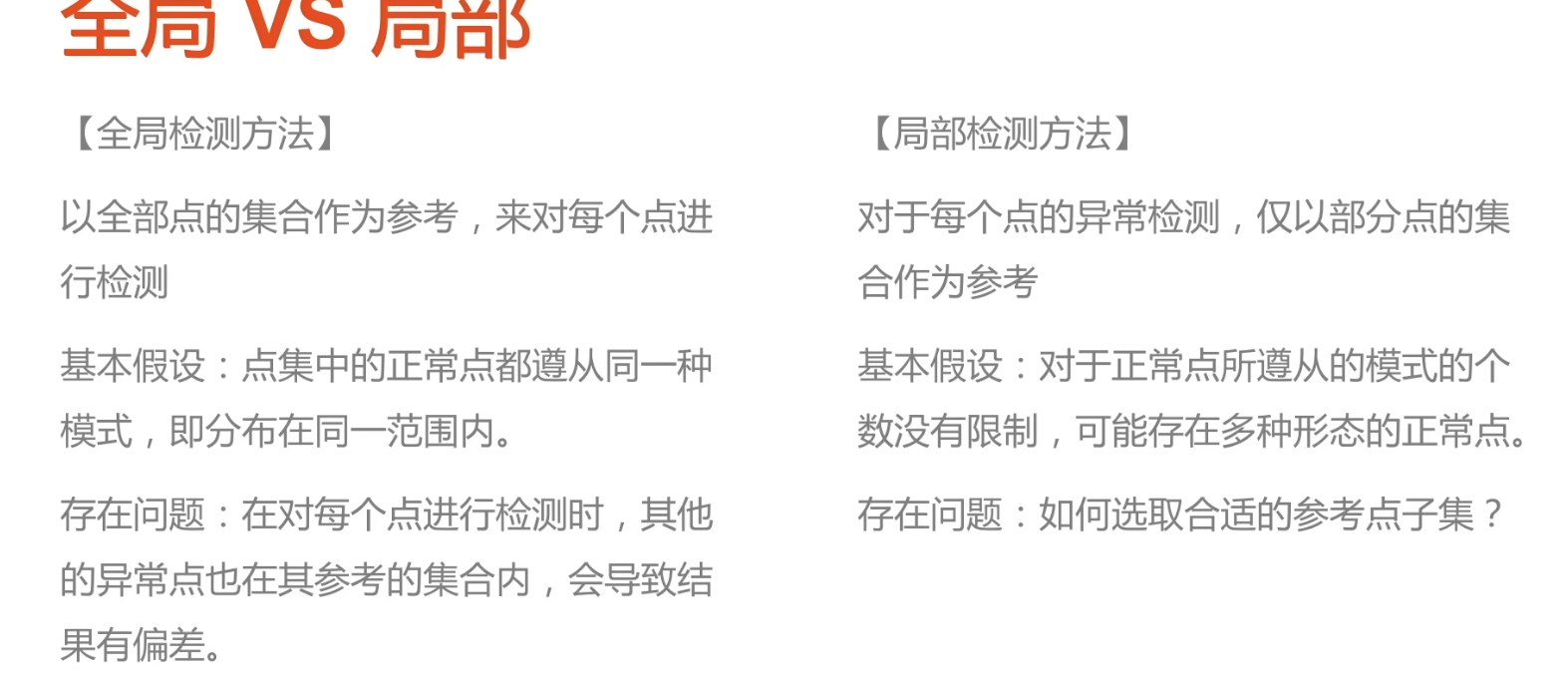
方法：无监督，专家认证

时序相关和时序独立：取决于是否依赖时间

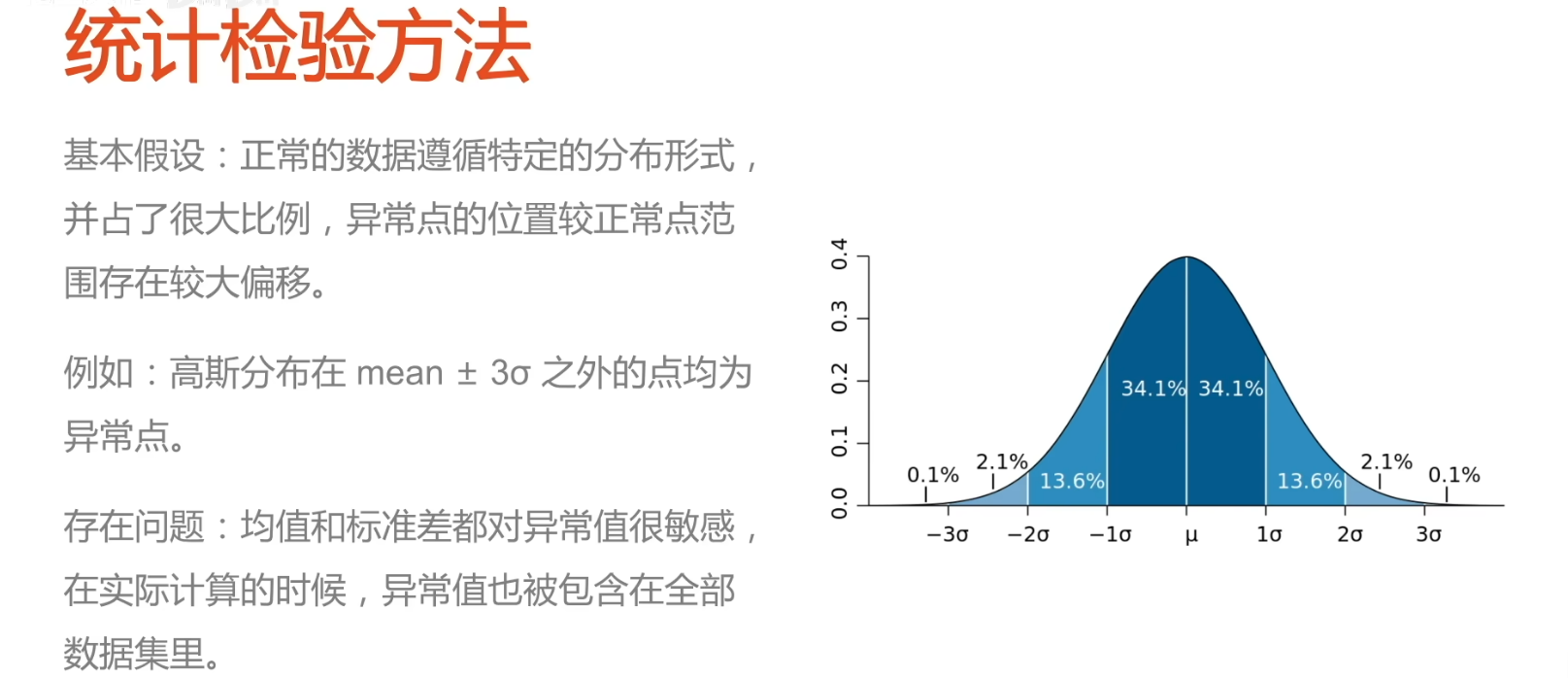


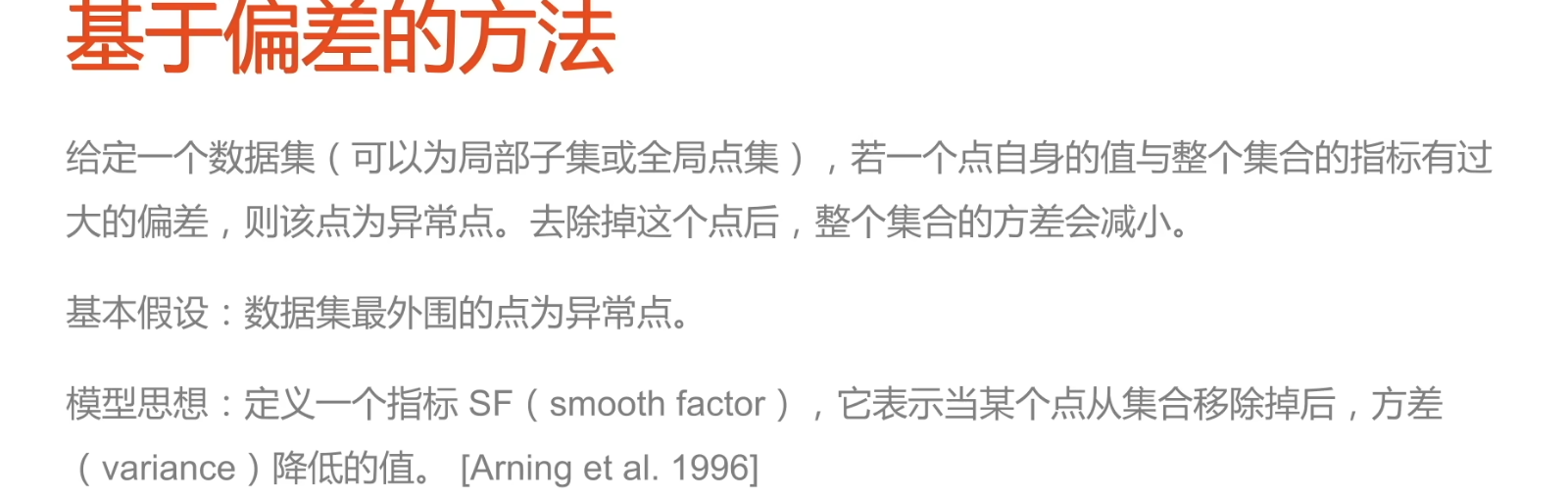


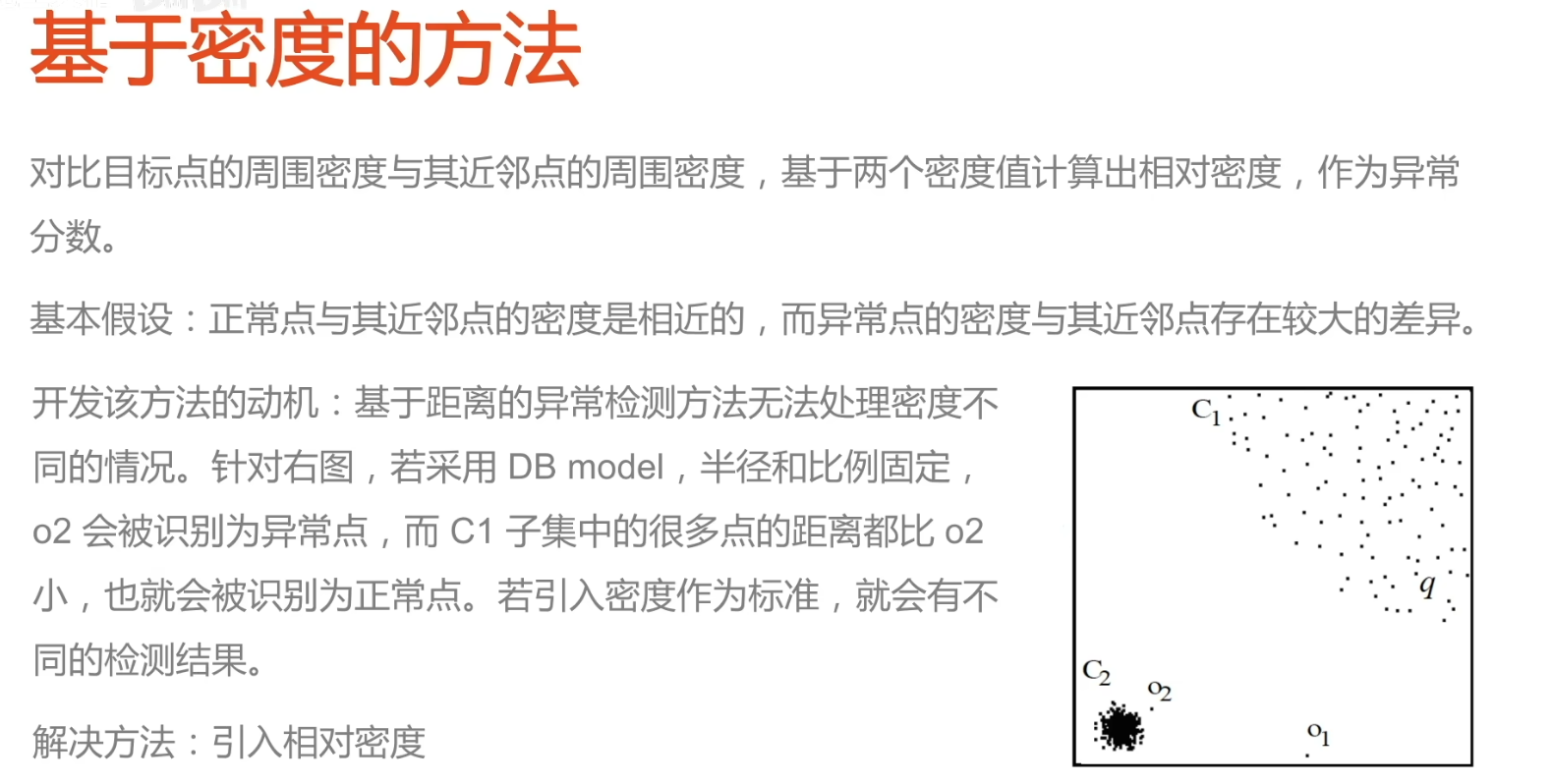
全局检测和局部检测



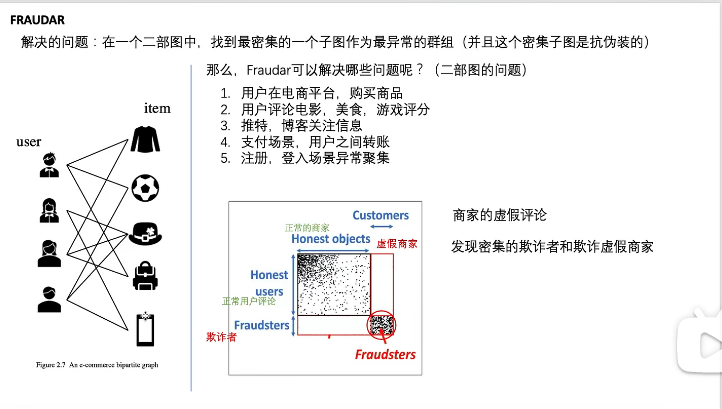




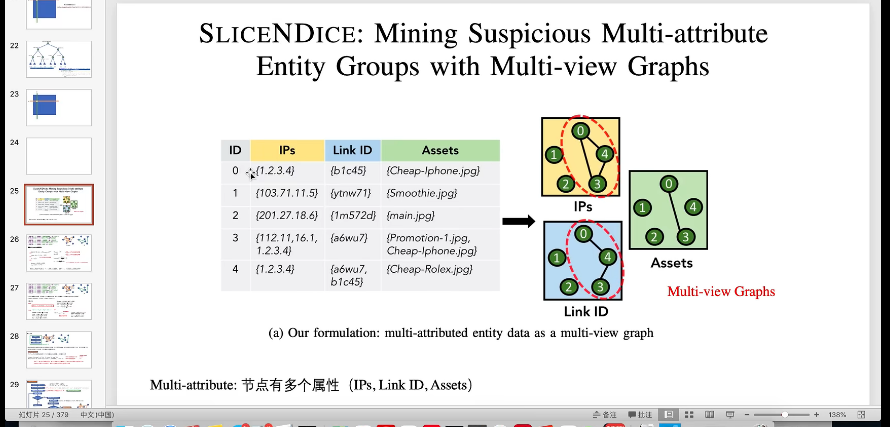




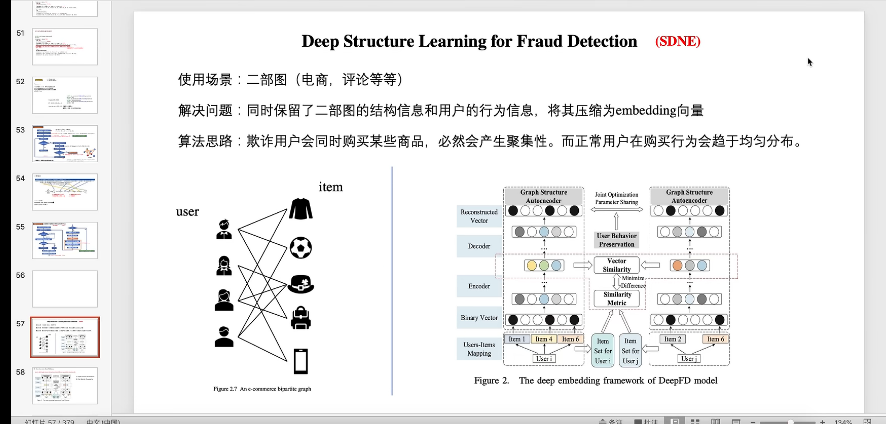
**fraudar**

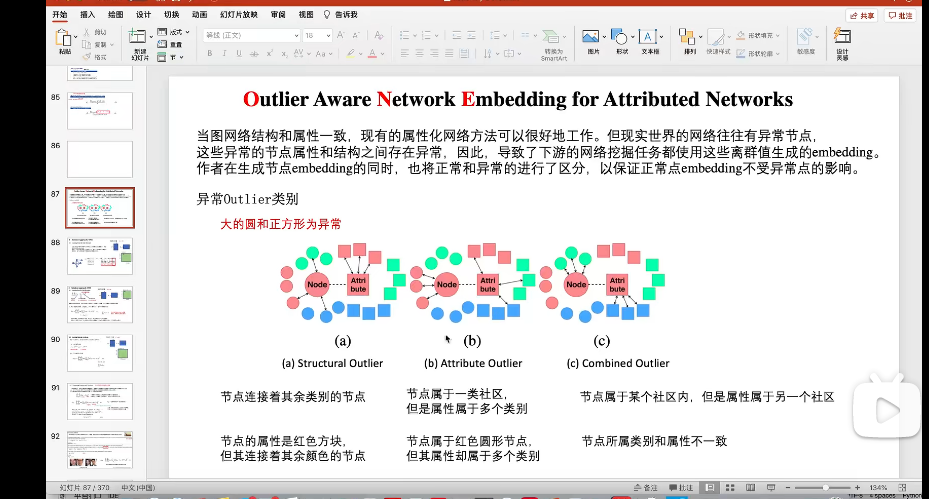


SliceNDice

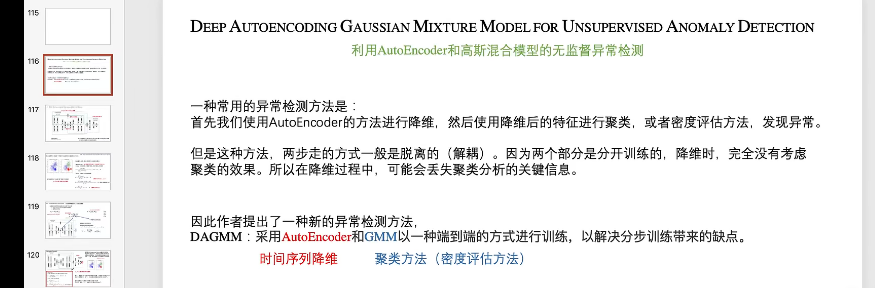


deepFD





DAGMM（降维）



MSCRED（时间序列，有点像）

而到了真正写论文的三天，我们遇到了以下问题：

**虚拟区域是什么？**

一开始我以为是检测一条水管，将中间分成了8个区域，求和之后发现这些值围绕0上下波动，

以为就是用多元线性回归方程建立，y（求和值）与x（各个区域流量差值）

**流量差值数据表中数据？**

我们理解过程如下：这个差值是否应该围绕0上下波动，一开始在围绕3上下波动的情况

是否说明已经发生泄露（题目表述也不太清楚，我们就认为这是不同流量计的特性差异了）；

**异常值的定义与后面的对检测异常结果解释？**

我们给数据按照标准打上异常标签，又用划定异常的标准制定了异常检测模型，然后还要说明异常

检测结果的合理性，"过拟合"是无法避免的，我们认为我们小组读不懂题目，建立异常检测标准和

检测异常似乎是同一件事情，我们最后也是做的稀里糊涂的。

**模型优化结果不明显？**

我们采取了很多我们理解范围内的很多方法，正态分布，箱线图，求梯度的五点求导法......最后我们发现

无论我们采用那种方法，无非围绕平均值、方差标准差、斜率这些基本的要素，认知和方法储备还是远远不够。

有时一些看似合理的优化还有着背道而驰的作用。

**回顾**

我们真正建模时间满打满算也就三天，期间还要上课，其实真正坐在电脑前的估计也就一天半，我们没有实现什么，

只是接受网上繁杂的相关信息，将其杂糅输出实现了一遍。花费时间最多的其实是在写论文过程上，找到思路，

将方法理解实现，再用有逻辑一点的语言表述出来，再向其中掺杂一点自己不切实际的思考，便交出了我们的答卷......

### 5.9python语法简单梳理

Python 模块(Module)，是一个 Python 文件，以 .py 结尾，包含了 Python 对象定义和Python语句。import

缩进敏感，换行不敏感

if：

XXX

else：

XXX

//函数定义

def function( 参数 ):

print("")

需要读取excel

### 5.10

#### 监督学习

告诉模型在每种情况下应该做什么，直到模型学会从情况到行动的映射

识别出模式并模仿他们的行为

##### 回归是什么？

“有多少”问题，围绕y去对x的数学分析，数值问题

##### 分类是什么？=》标记问题

划定1-n种类别

层次分类：宁可划分错也要避免没划分的情况

分类问题的常见损失函数被称为*交叉熵*（cross-entropy）

##### 搜索=》推荐系统

基于相关系数评分和筛选过滤的一种算法

##### 序列学习是什么？

输入是连续的，我们需要考虑“前后文”的联系

#### 无监督学习

聚类、主成分分析、因果关系和概率图、生成对抗性网络

#### 离线学习与环境互动=》强化学习

交互性，对环境做出趋优化的一种行为

### 5.10

n维数组，也称为*张量*（tensor）

#### 深度学习框架

PyTorch 是一个优化的张量库，Facebook人工智能研究团队开发

TensorFlow 是由 Google 团队开发的深度学习框架之一

MXNET 关心性能，开发和移植的便利性的

paddle 百度

Keras、Caffe...

#### 配置框架

下载配置

[(54条消息) pytorch超详细安装教程，Anaconda、PyTorch和PyCharm整套安装流程*pytorch安装*李问号的博客-CSDN博客](https://blog.csdn.net/Bluebro/article/details/127161165)

下载过慢，http error：没用anaconda

添加解释器

[(54条消息) pycharm.2023.1配置python解释器时找不到conda环境*OnePiece*zym的博客-CSDN博客](https://blog.csdn.net/OnePiece_zym/article/details/130426615)

### 5.11（数据操作）

arange创建行向量

shape属性访问张量各个轴，size访问大小

reshape改变形状

zeros、ones建立张量，randn随机取样，tensor直接赋值

torch.cat((X, Y), dim=0), torch.cat((X, Y), dim=1)//连结

**广播机制**

1. 通过适当复制元素来扩展一个或两个数组，以便在转换之后，两个张量具有相同的形状；

* 2.对生成的数组执行按元素操作。

**索引和切片**

**节省内存**

新结果会重新分配内存

执行原地操作非常简单。 我们可以使用切片表示法将操作的结果分配给先前分配的数组

如果在后续计算中没有重复使用X， 我们也可以使用X[:] = X + Y或X += Y来减少操作的内存开销。

**转换为其它python对象**

就地操作更改一个张量也会同时更改另一个张量。

A = X.numpy()  
B = torch.tensor(A)

要将大小为1的张量转换为Python标量，我们可以调用item函数或Python的内置函数。

### 5.12（数据预处理，NVIDIA比赛了解）

Omniverse RTX Renderer：是一种基于物理的实时光线追踪渲染器

#### [2:5] [1:5:2]

print(str) # 输出字符串

print(str[0:-1]) # 输出第一个到倒数第二个的所有字符

print(str[0]) # 输出字符串第一个字符

print(str[2:5]) # 输出从第三个开始到第六个的字符（不包含第六个，即 "[ n , m )" ）

print(str[2:]) # 输出从第三个开始后的所有字符

print(str[1:5:2]) # 输出从第二个开始到第五个且每隔一个的字符（步长为2）

print(str + '你好') # 连接字符串\

#### 读取数据集

os.makedirs(os.path.join('..', 'data'), exist\_ok=True)

exist\_ok：只有在目录不存在时创建目录，目录已存在时不会抛出异常。

os.makedirs() 方法用于递归创建目录。

data\_file = os.path.join('..', 'data', 'house\_tiny.csv')

os.path.join()：连接目录和路径

data = pd.read\_csv(data\_file)  
print(data)

读取文件

#### 处理缺失值（fillnan，dummies这个不太会用）

inputs, outputs = data.iloc[:, 0:2], data.iloc[:, 2]  
inputs = inputs.fillna(inputs.mean())  
print(inputs)

iloc：索引 fillna：填补空缺值

inputs = pd.get\_dummies(inputs, dummy\_na=True)  
print(inputs)

pd.get\_dummies（数据离散化，独热码）

### 5.13

#### 降维

调用**求和**函数会沿所有的轴降低张量的维度，使它变为一个标量

计算平均值的函数也可以沿指定轴降低张量的维度

点积（dot），矩阵向量积

范数（不太懂）

#### 微积分

导包没有导入其中某个句子，**d2l不是自己定义的包，要单独下**

在线安装（cmd访问）和离线安装（安装包）

今日问题：

1.module 'd2l' has no attribute 'set\_figsize'

2.#@save

解决方法：github上找到了d2l的内容，但是又不能直接保存下来，就把里面

的d2l.py内容copy到本地，目前能跑......

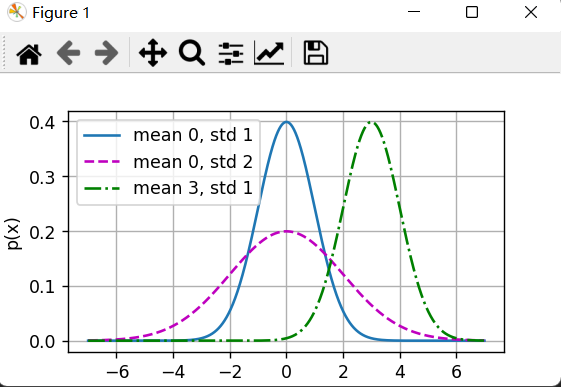
### 5.14（写计网实验报告+小程序比赛去了）

### 5.15

1.还是d2l的问题，现在的解决方法是哪里需要哪里搬，因为d2l的内容也是基于包上的。。。

然后自己塞进d2l里，在其它py文件导包，然后d2l.plot使用即可

（已解决）



不过不要忘记plt.show()才能显示

2.时间的基准测试，定义一个计时器

3.**CUDA(Compute Unified Device Architecture)**，是一种新的操作GPU计算的硬件和软件架构，是建立在NVIDIA的GPUs上的一个通用并行计算平台和编程模型，它提供了GPU编程的简易接口，基于CUDA编程可以构建基于GPU计算的应用程序，利用GPUs的并行计算引擎来更加高效地解决比较复杂的计算难题。它将GPU视作一个数据并行计算设备，而且无需把这些计算映射到图形API。操作系统的多任务机制可以同时管理CUDA访问GPU和图形程序的运行库，其计算特性支持利用CUDA直观地编写GPU核心程序。

### 5.16-5.19（云服务器知识的学习）

腾讯云租的公费cpu，云服务器包括CPU、内存、系统盘、带宽、网络

学习遇到的问题：

1.由于是自己摸索，不太能理解云服务器的概念，在最初的登录上面就遇到了问题，因为服务器是在舍友的账号上，自己的

账号不能很好地看见服务器信息，以为一个服务器的信息能让多个用户同时管理。在网上看了视频后，自己却怎么也找不到【新建】按钮，折腾了很久。

解决情况：按照理解后，其实腾讯云提供的是一个平台，我们还是需要在别的终端管理软件上管理我们的服务器，我们只需要

创建并配置 -> 远程连接使用 -> 开机、关机、文件传输、重装系统、配置域名、防火墙、密钥 -> 一些应用与操作实例

2.思路很清晰，but...，因为我们的系统是linux，后面在Xshell中终端连接成功



### 5.20-5.23

线性神经网络

**矢量化加速**，可以直接相加，免去了for循环的开销浪费

**正态分布与平方损失的关系**：均方误差损失函数（简称均方损失）可以用于线性回归的一个原因是： 我们假设了观测中包含噪声，其中噪声服从正态分布。

#### 线性回归的从零开始实现和简洁实现（基于框架再做了一遍）

生成数据集-》读取数据集-》初始化模型参数-》定义模型-》定义损失函数-》定义优化算法-》训练

**线性模型参数噪声项生成数据集及其标签**：y=Xw+b+e

**读取**：定义一个data\_iter函数， 该函数接收批量大小、特征矩阵和标签向量作为输入，生成大小为batch\_size的小批量

**初始化模型**：更新这些参数，梯度下降自动微分

**定义模型**：Xw+b 定义损失函数：均方损失

**优化算法**：然后根据参数计算损失的梯度。 接下来，朝着减少损失的方向更新我们的参数

**训练**：重复 1.初始化参数 2.计算梯度、更新参数

#### 图像分类数据集（框架内置函数下载数据集并读取到内存中）

#### softmax回归的从零开始实现和简洁（框架）实现

初始化模型参数-》定义softmax操作、模型、损失函数、分类精度-》训练、预测

**初始化模型参数**：按一定大小的矩阵输入

**定义softmax操作**：如何解决张量直接相加降维？不折叠

1. 对每个项求幂（使用exp）；
2. 对每一行求和（小批量中每个样本是一行），得到每个样本的规范化常数；
3. 将每一行除以其规范化常数，确保结果的和为1。

**定义模型**：wX+b

**分类精度**：性能衡量标准

**训练**：

**预测**：