**南大 一面 Lambda 实验室**

1. 图的表示方式 邻接矩阵和邻接表的优劣
2. 图神经网络 相关
3. 对机器学习的认识？ 如果你遇到科研挫折了，论文屡投不中，你怎么处理？

**南大 二面**

1. 为什么想要接触机器学习
2. 如何看待人工智能领域
3. 什么是半正定矩阵

1)设A为实对称矩阵，若对于每个非零实向量X，都有X'AX≥0，则称A为半正定矩阵，称X'AX为半正定二次型。（其中，X'表示X的转置。）

2）设A为实对称矩阵，若对于每个非零实向量X，都有X'AX>0，则称A为正定矩阵，称X'AX为正定二次型。

3）设A为实对称矩阵，若对于每个非零实向量X，都有X'AX<0，则称A为负定矩阵，称X'AX为负定二次型。

4） 设A为实对称矩阵，若对于每个非零实向量X，都有X'AX≤0，则称A为半负定矩阵，称X'AX为半负定二次型。

1. 什么是大数定律

答 ： 在随机事件的大量重复出现中，往往呈现几乎[必然](https://baike.baidu.com/item/%E5%BF%85%E7%84%B6/4325015)的规律，这个规律就是大数定律。通俗地说，这个定理就是，在试验不变的条件下，重复试验多次，随机事件的频率近似于它的概率。偶然中包含着某种必然。

表现形式：[切比雪夫](https://baike.baidu.com/item/%E5%88%87%E6%AF%94%E9%9B%AA%E5%A4%AB)大数定理

随着[样本容量](https://baike.baidu.com/item/%E6%A0%B7%E6%9C%AC%E5%AE%B9%E9%87%8F)n的增加，[样本](https://baike.baidu.com/item/%E6%A0%B7%E6%9C%AC)平均数将接近于[总体](https://baike.baidu.com/item/%E6%80%BB%E4%BD%93)平均数。从而为统计推断中依据样本平均数估计总体平均数提供了理论依据。

[伯努利](https://baike.baidu.com/item/%E4%BC%AF%E5%8A%AA%E5%88%A9)大数定律

该定律是[切比雪夫大数定律](https://baike.baidu.com/item/%E5%88%87%E6%AF%94%E9%9B%AA%E5%A4%AB%E5%A4%A7%E6%95%B0%E5%AE%9A%E5%BE%8B)的特例，其含义是，当n足够大时，事件A出现的频率将几乎接近于其发生的概率，即频率的稳定性。

1. 有工程能力吗

读研动机？

从研究某个领域到发表论文，这期间大致的一个研究流程是怎么样的?

**中科院计算所**

**提问方向： 数学问题 + 算法问题 + 谈人生**

**60-70% 提问项目经历，需要好好准备，期间穿插基本知识点**

1. N！ 末尾有多少个零 2\*5可以得到一个0，所以也就是求质因数5的个数
2. 求扩展N！的二进制表示中最低位1中的位置。相当于求质因数的2的个数
3. 做过机器学习的项目吗？
4. 对于机器学习了解过那些优化的算法函数？
5. 什么是正交矩阵？

如果AAT=E（E为单位矩阵，AT表示“矩阵A的转置矩阵”）或ATA=E，则n阶[实矩阵](https://baike.baidu.com/item/%E5%AE%9E%E7%9F%A9%E9%98%B5/10691442)A称为正交矩阵

1. 什么是凸优化问题？

凸优化问题（OPT，convex optimization problem）指定义在凸集中的凸函数最优化的问题。

<https://www.jianshu.com/p/6a962fb1b4e0>

**北航**

**1.解释一下DMA**

DMA的出现就是为了解决批量数据的输入/输出问题。DMA是指外部设备不通过CPU而直接与系统内存交换数据的接口技术。这样数据的传送速度就取决于存储器和外设的工作速度。

2.流水线中的三种相关

（1）结构相关：当硬件资源满足不了指令重叠执行的要求，而发生资源冲突时，就发生了结构相关。

（2）数据相关：当一条指令需要用到前面指令的执行结果，而这些指令均在流水线中重叠执行时，就可能引起数据相关。

（3）控制相关：当流水线遇到分支指令和其它能够改变PC值的指令时，就会发生控制相关。

3. 说下五级流水CPU的各阶段

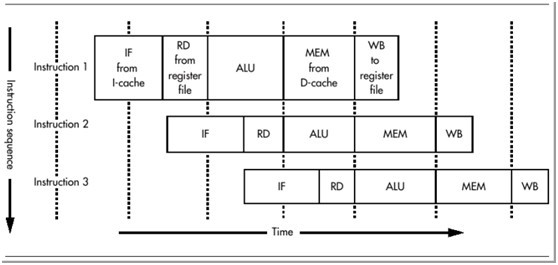
IF(instruction fetch)，从指令高速缓存中获取下一条指令。

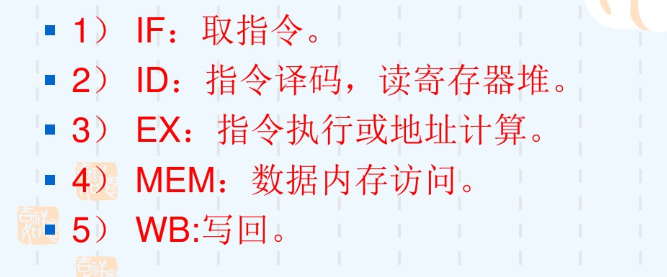
RD(Read register)，从指定的源寄存器域中指定的cpu寄存器中读取数据。

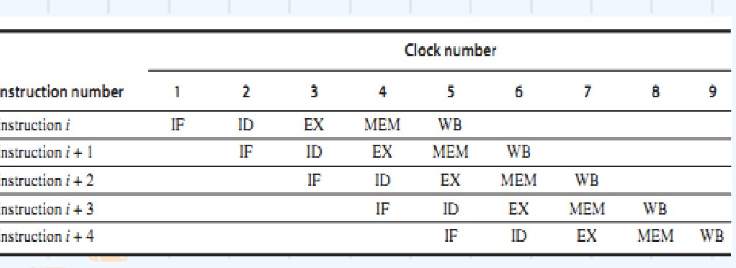
ALU(arithmetic/logic  unit)，算术和逻辑运算操作。

MEM(memory R/W)，读写数据高速缓存中的内存变量。

WB(Write back)，将结果写入寄存器堆中。







进程间通讯方式有：管道，信号，信号量，消息队列，共享内存，套接字共六种

1.分页和分段

  用户程序的地址空间被划分成若干固定大小的区域，称为“页”，相应地，内存空间分成若干个物理块，页和块的大小相等。可将用户程序的任一页放在内存的任一块中，实现了离散分配。

将用户程序地址空间分成若干个大小不等的段，每段可以定义一组相对完整的逻辑信息。存储分配时，以段为单位，段与段在内存中可以不相邻接，也实现了离散分配。

页是信息的物理单位，分页的主要目的是为了实现离散分配，提高内存利用率，分页仅仅是系统管理上的需要，完全是系统行为，对用户不可见

段是信息的逻辑单位。分页的主要目的是更好地满足用户需求。一个段通常包含着一组属于一个逻辑模块的信息。分段对用户是可见的，用户编程时需要显式地给出段名

2.TCP与UDP

1） TCP提供面向连接的传输，通信前要先建立连接（三次握手机制）； UDP提供无连接的传输，通信前不需要建立连接。

2） TCP提供可靠的传输（有序，无差错，不丢失，不重复）； UDP提供不可靠的传输。

3） TCP面向字节流的传输，因此它能将信息分割成组，并在接收端将其重组； UDP是面向数据报的传输，没有分组开销。

1. TCP提供拥塞控制和流量控制机制； UDP不提供拥塞控制和流量控制机制。

3.输入一个URL会经历什么

→ 1- 输入网址

→ 2- 缓存解析

→ 3- 域名解析

→ 4- tcp连接，三次握手

→ 6- 页面渲染

服务器域名解析成ip地址

通过DNS服务器获取到ip地址。

发数据包时就发给服务器ip。

数据包+源地址ip+目标地址ip

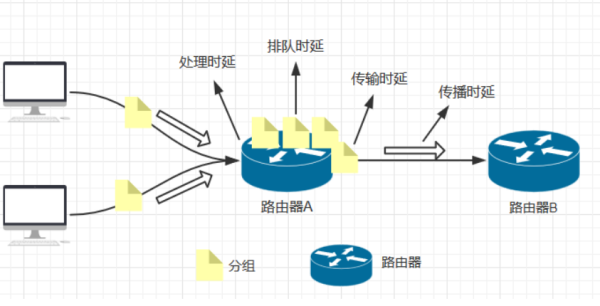
数据包+源地址ip+目标地址ip+源mac地址+目标mac地址

源mac地址和目标mac地址在不断动态重写，重新封装。在每段链路中，包没变，帧在变。

mac地址 （物理地址）决定了下一跳到哪去，ip地址决定了最终到哪去。

1. 计算机网络中的时延

计算机bai网络中的四种延迟分别是：节点处理du延迟 、排队延迟、发zhi送延迟、传播延迟。



**发送时延是主机或路由器发送数据帧所需要的时间，** 也就是从发送数据帧的第一个比特算起，到该帧的最后一个比特发送完毕所需要的时间

  传播时延是电磁波在信道中传播一定的距离需要花费的时间

主机或路由器在收到分组时要花费一定的时间进行处理，例如分析分组的首部、从分组中提取数据部分、进行差错检验或查找适当的路由等，这就产生了**处理时延**

 分组通过网络传输时，要经过很多路由器。分组在进入路由器后要先在输入队列中排队等待处理。在路由器确定了转发接口后，还要在输出队列中 排队等待转发。这两个因素就造成了**排队时延** 。

1. 特征向量和特征值的意义

矩阵乘法对应了一个变换，是把任意一个向量变成另一个方向或长度都大多不同的新向量。在这个变换的过程中，原向量主要发生旋转、伸缩的变化。如果矩阵对某一个向量或某些向量只发生伸缩变换，不对这些向量产生旋转的效果，那么这些向量就称为这个矩阵的特征向量，伸缩的比例就是特征值。

实际上，上述的一段话既讲了矩阵变换特征值及特征向量的几何意义（图形变换）也讲了其物理含义

1. 机器学习和深度学习

人工智能就是研究如何使计算机去做过去只有人才能做的智能工作

机器学习的广泛定义：

对于某类任务 T 和性能度量 P，一个计算机程序被认为可以从经验 E 中学习是指，通过经验 E 改进后，它在任务 T 上由性能度量 P 衡量的性能有所提升。

深度学习：

。基本上，深度学习用于创建人工“神经网络” ，可以自己学习和做出明智的决策。我们可以说深度学习是机器学习的一个子领域

区别：

数据依赖性

性能是两种算法之间的主要关键区别。虽然，当数据很小时，深度学习算法表现不佳。这就是是深度学习算法需要大量数据才能完美理解的原因。

硬件依赖

通常，深度学习依赖于高端机器，而传统学习依赖于低端机器。因此，深度学习要求包括GPU。这是它工作中不可或缺的一部分。它们还进行大量的矩阵乘法运算。

问题解决方式

当应用传统机器学习算法解决问题的时候，传统机器学习通常会将问题分解为多个子问题并逐个子问题解决最后结合所有子问题的结果获得最终结果。相反，深度学习提倡直接的端到端的解决问题。

执行时间

通常情况下，训练一个深度学习算法需要很长的时间。这是因为深度学习算法中参数很多，因此训练算法需要消耗更长的时间。最先进的深度学习算法 ResNet完整地训练一次需要消耗两周的时间，而机器学习的训练会消耗的时间相对较少，只需要几秒钟到几小时的时间。

可解释性

至关重要的一点，我们把可解释性作为比较机器学习和深度学习的一个因素。

我们看个例子。假设我们适用深度学习去自动为文章评分。深度学习可以达到接近人的标准，这是相当惊人的性能表现。但是这仍然有个问题。深度学习算法不会告诉你为什么它会给出这个分数。当然，在数学的角度上，你可以找出来哪一个深度神经网络节点被激活了。但是我们不知道神经元应该是什么模型，我们也不知道这些神经单元层要共同做什么。所以无法解释结果是如何产生的。另一方面，为了解释为什么算法这样选择，像决策树(decision trees)这样机器学习算法给出了明确的规则，所以解释决策背后的推理是很容易的。因此，决策树和线性/逻辑回归这样的算法主要用于工业上的可解释性。

梯度下降法：

我们有一个可微分函数，这个函数就像是这座大山，我们的目标就是找到这个函数的最小值，也就是下山。最快的下山方式就是找到这个山最陡峭的地方，然后下去，对应到函数里就是找到定点的梯度，然后朝着梯度相反的方向，就能让函数值下降的最快。

梯度是一个向量，向量有方向，梯度的方向就指出了函数在给定点的上升最快的方向

什么是结构化程序？

结构化程序设计采用自顶向下、逐步求精的设计方法，各个模块通过“顺序、选择、循环”的控制结构进行连接，并且只有一个入口、一个出口。

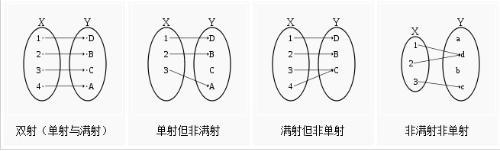


讲一下开机过程？

https://www.cnblogs.com/shao-ye/p/11202680.html

首先加电开机，等到供电稳定了，找到BIOS程序的启动代码，BIOS启动，显示进行POST自检，即对系统的关键设备，硬件（如显卡，内存）等进行检查，并进行相关初始化，如果硬件出错，主板会发出不同的蜂鸣，启动终止。硬件自检结束后，BIOS这个时候将控制权交给下一阶段的启动程序，将控制权交给排在第一位的存储设备，即开始从第一位设备中读取设备的MBR（主引导记录），其主要作用是找到操作系统的位置。然后是硬盘启动，加载操作系统。

单射、双射、满射？



单射就是只能一对一，不能多对一

满射只要Y中的元素在X中都能找到原像就行了(一对一，多对一都行).

双射就是既是单射又是满射(一个对一个,每个都不漏掉).

生成可执行文件过程

1) 预处理  
    gcc -E test.c -o test.i  
在当前目录下会多出一个预处理结果文件 test.i，打开 test.i 可以看到，在 test.c 的基础上把stdio.h和stdlib.h的内容插进去了。  
  
2) 编译为汇编代码  
    gcc -S test.i -o test.s  
其中-S参数是在编译完成后退出，-o为指定文件名。

编译过程就是将预处理后得到的预处理文件（如test.i）进行词法分析，语法分析，语义分析并优化，生成汇编代码文件  
  
3) 汇编为目标文件  
    gcc -c test.s -o test.o  
.o就是目标文件。目标文件与可执行文件类似，都是机器能够识别的可执行代码，但是由于还没有链接，结构会稍有不同。  
  
3) 链接并生成可执行文件  
    gcc test.o -o test

**5. 梯度消失/梯度爆炸（Vanishing / Exploding gradients）**

[**https://blog.csdn.net/qq\_41813258/article/details/100389967**](https://blog.csdn.net/qq_41813258/article/details/100389967)

**训练神经网络，尤其是深度神经所面临的一个问题就是梯度消失或梯度爆炸，也就是你训练神经网络的时候，导数或坡度有时会变得非常大，或者非常小，甚至于以指数方式变小，这加大了训练的难度。**

**本质上，梯度消失和爆炸是一种情况。在深层网络中，由于网络过深，如果初始得到的梯度过小，或者传播途中在某一层上过小，则在之后的层上得到的梯度会越来越小，即产生了梯度消失。梯度爆炸也是同样的。一般地，不合理的初始化以及激活函数，如sigmoid等，都会导致梯度过大或者过小，从而引起消失/爆炸。**

**3、ReLu、leakReLu等激活函数**

**ReLu：其函数的导数在正数部分是恒等于1，这样在深层网络中，在激活函数部分就不存在导致梯度过大或者过小的问题，缓解了梯度消失或者爆炸。同时也方便计算。当然，其也存在存在一些缺点，例如过滤到了负数部分，导致部分信息的丢失，输出的数据分布不在以0为中心，改变了数据分布。**

**leakrelu：就是为了解决relu的0区间带来的影响，其数学表达为：leakrelu=max(k\*x,0)其中k是leak系数，一般选择0.01或者0.02，或者通过学习而来。**

**4.Batch Normalization**

**Batch Normalization是深度学习发展以来提出的最重要的成果之一了，目前已经被广泛的应用到了各大网络中，具有加速网络收敛速度，提升训练稳定性的效果，Batch Normalization本质上是解决反向传播过程中的梯度问题。Batch Normalization，简称BN，即批规范化，通过规范化操作将输出信号x规范化到均值为0，方差为1保证网络的稳定性。**

Dropout 借鉴了 集成学习的思想，如bagging和boosting，就是裁剪掉一些神经元，这样就相当于变成了弱学习器，这样这多个弱学习器组合成强学习器

为什么计算机采用二进制

1. 电路中容易实现 ：当计算机工作的时候，电路通电工作，于是每个输出端就有了电压。  
     
   电压的高低通过模数转换即转换成了二进制：  
     
   高电平是由1表示，低电平由0表示。也就是说将模拟电路转换成为数字电路。这里的高电平与低电平可以人为确定，一般地，2.5伏以下即为低电平，3.2伏以上为高电平。二进制数码只有两个(“0”和“1”)。电路只要能识别低、高就可以表示“0”和“1”。  
     
   2、物理上最易实现存储 ：  
     
   （1）基本道理：二进制在物理上最易实现存储，通过磁极的取向、表面的凹凸、光照的有无等来记录。  
     
   （2）具体道理：对于只写一次的光盘，将激光束聚住成1--2um的小光束，依靠热的作用融化盘片表面上的碲合金薄膜，在薄膜上形成小洞（凹坑），记录下“1”，原来的位置表示记录“0”。  
     
   3、便于进行加、减运算和计数编码。易于进行转换，二进制与十进制数易于互相转换。  
     
   简化运算规则：  
     
   两个二进制数和、积运算组合各有三种，运算规则简单，有利于简化计算机内部结构，提高运算速度。  
     
   电子计算机能以极高速度进行信息处理和加工，包括数据处理和加工，而且有极大的信息存储能力。数据在计算机中以器件的物理状态表示，采用二进制数字系统，计算机处理所有的字符或符号也要用二进制编码来表示。  
     
   用二进制的优点是容易表示， 运算规则简单，节省设备。人们知道，具有两种稳定状态的元件（如晶体管的导通和截止，继电器的接通和断开，电脉冲电平的高低等）容易找到，而要找到具有10种稳定状态的元件来对应十进制的10个数就困难了  
     
   4、便于逻辑判断（是或非）。适合逻辑运算：逻辑代数是逻辑运算的理论依据，二进制只有两个数码，正好与逻辑代数中的“真”和“假”相吻合。二进制的两个数码正好与逻辑命 题中的“真(Ture)”、“假(False)或称为”是(Yes)、“否(No)相对应。  
     
   5、用二进制表示数据具有抗干扰能力强，可靠性高等优点。因为每位数据只有高低两个状态，当受到一定程度的干扰时，仍能可靠地分辨出它是高还是低。