https://www.douyin.com/video/7344678214398151946

# 标题:未找到标题  
## 关键字: 未找到关键字  
## 作者: 严伯钧  
## 以下是针对你提供的文本进行标点符号补全和错别字修订的结果：  
  
这两天呢，我在阿开看到了一个清华大学的重要科研成果：就是清华大学姚班的科学家们做出了一个关于矩阵乘法的重大突破。这个文章足足八十七页长，这里面作者居然还有个本科生。哎，这年头的本科生啊，真是一届比一届厉害了。我就纳闷了，这么重要一个计算机算法领域的成果，怎么就没有激起什么水花呢？没事，你们不关注，我来替你们关注。老规矩啊，塞到 t x y z 里面，甭管多长的文章，一样盘你。问几次，就给你盘清楚了。  
  
这篇文章虽然长，但很多呢是证明推导过程，主要核心其实就用计算机去做矩阵乘法的效率，又得到了一个提升。那咋回事呢？这里面逻辑有点长，你听我慢慢给你讲。当然，没有耐心听我说的，你们自己去 t x y z 找这篇文章，叫做 "Faster Matrix Multiplication via Asymmetric Hash"，啊，就是通过非对称哈希实现的更快矩阵乘法。自己问 AI 去啊。  
  
那么，所以啥是矩阵乘法呢？首先，啥是矩阵啊？矩阵呢是啥，我就不多讲了。这一讲，就太长了，直接要把大学里面线性代数讲一遍。我就假设大家知道什么是矩阵。比方，一个二维方阵 a，它有四个元素，从左到右，从上到下分别是 a11, a12, a21, a22 乘以另外一个二阶方阵 b，四个元素分别是 b11, b12, b21, b22。问 a 乘以 b，乘出来的这个新矩阵 c 里面也是四个元素，这四个元素分别等于啥？  
  
那么，这个矩阵乘法的定义很简单，分别是：  
  
c11 = a11 \* b11 + a12 \* b21  
c12 = a11 \* b12 + a12 \* b22  
c21 = a21 \* b11 + a22 \* b21  
c22 = a21 \* b12 + a22 \* b22  
  
那推广一下，对于一个 n 阶矩阵 a 乘以另外一个 n 阶矩阵 b，乘出来的这个新矩阵 c，第 i 行第 j 列的元素 cij 就等于矩阵 a 的第 i 行和矩阵 b 的第 j 列把这个 a 矩阵的 i 行和 b 矩阵的 j 列当成两个矢量做点乘，那就是 cij = Σ (aik \* bkj)，k 从 1 加到 n。啊，这个呢是矩阵乘法的定义。  
  
你别看这个复杂，但里面呢就是做很多次乘法，然后再做很多次的加法。那我们算一下，一个 n 乘以 n 的矩阵乘法里面要算多少次乘法和多少次加法？简单算一下，就知道这里面乘法要做 n 的三次方次，而加法要做 n 的平方乘以 n 减一次。啊，这个数字有啥意义呢？哎，对计算机来说，那就有意义了。因为对于计算机来说，加法是很简单的，但是乘法却没有那么简单。所以，一个矩阵乘法的运算，如果乘法次数少，速度就会快。而对于现在的 AI 来讲，主要是神经网络里面的 AI，里面做的计算就主要是矩阵乘法。因为每个神经元里面的这个 w 参数，它就是个矩阵。数据呢是以矢量的方式进入神经元，然后要算的就是个矩阵乘法。如果这个乘法算的速度特别快，那么 AI 运算的效率就会高。  
  
有人就要问你了，你都知道矩阵乘法的公式了，直接代入算不就完了吗？没那么简单。因为当矩阵的维度很大，也就是 n 很大的时候，你会发现这个差别它就大了去了。比方，现在的大语言模型，神经元里的矩阵都挺大的。比方说，n 等于五百的话，那么做乘法的次数如果是 n 的三次方，那就是要做 1.25 亿次的乘法。如果我们能让矩阵乘法当中，计算机做乘法的次数变成 n 的平方，那你效率就直接提升五百倍。  
  
好了，你可能要问，这个矩阵乘法的定义都清清楚楚的写在那里的，那就是 n 的三次方呀，咋还能算乘法的次数比这个少呢？哎，还真可以，这就是算法的领域了。最早是一九六九年的时候，有个学者叫 Strassen 啊，他提出个 Strassen 算法。比方，一个二乘二的矩阵，正常如果按照定义做乘法，我要做的是二的三次方，也就是八次乘法。但是 Strassen 这个算法呢，就只要做七次就可以了。怎么做的呢？他就是这么搞的，就先搞七个量。我就写在这，我就