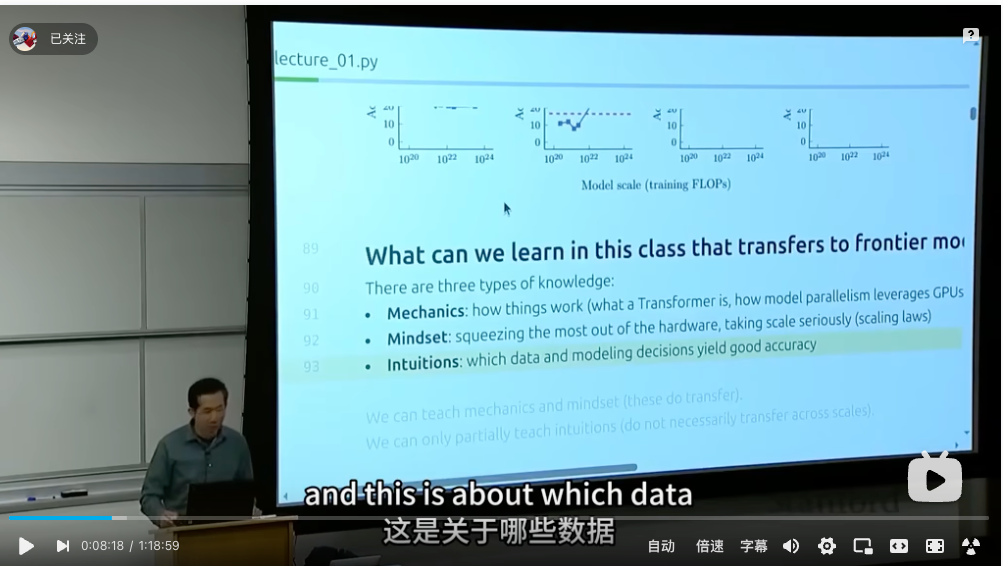
# 【README】

大模型课程(中英字幕完结)斯坦福 CS336

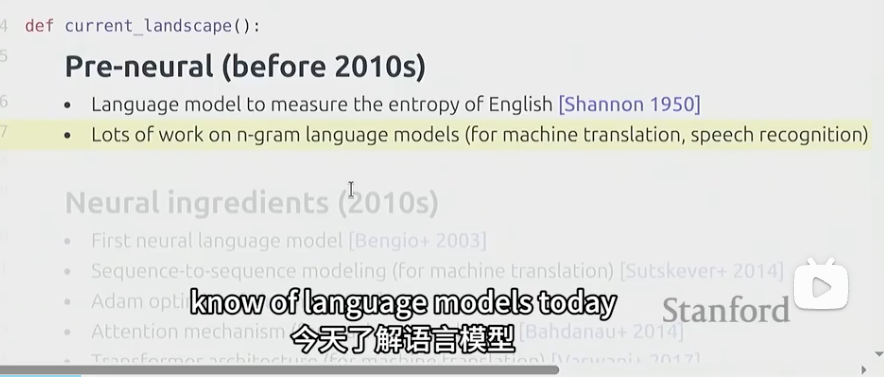
# 【1】第1章-入门

1. 有三类知识：
   1. 机制；
   2. 思维模式；
   3. 直觉；



## 【1.1】大模型发展

### 【第1阶段】2010s之前（2010-2019）



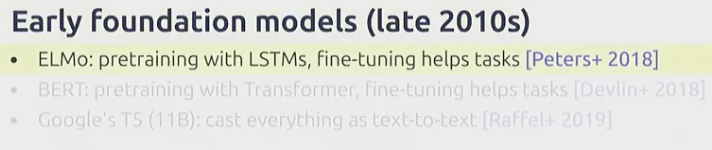
1 衡量英语熵的语言模型

2 n克语言模型，针对机器翻译，语音识别；

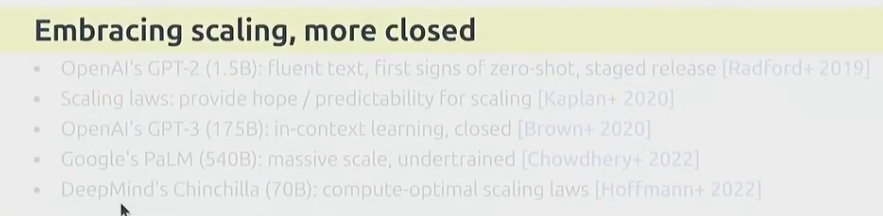
### 【第2阶段】神经网络因素-2010s



### 【第3阶段】2010s之后-早期基础模型

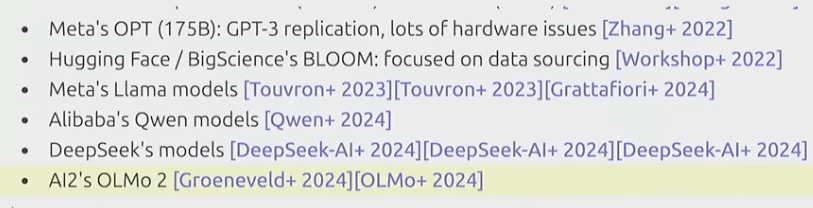


## 【1.2】拥抱扩展，闭源模型

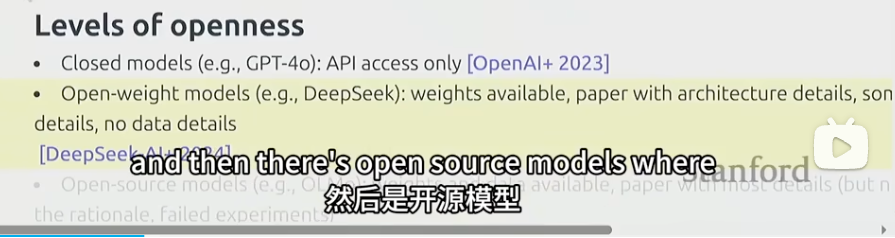


## 【1.3】开源模型

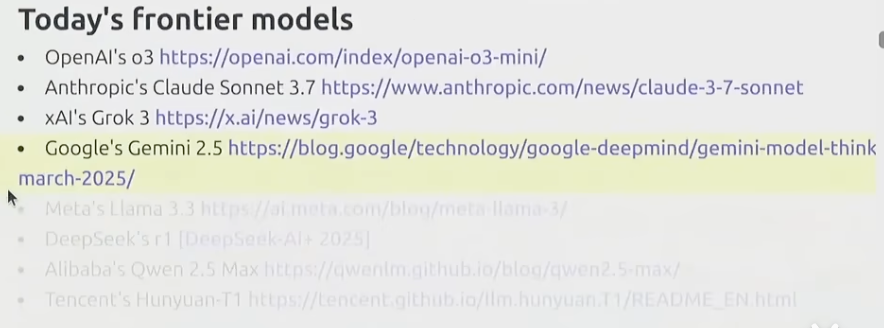




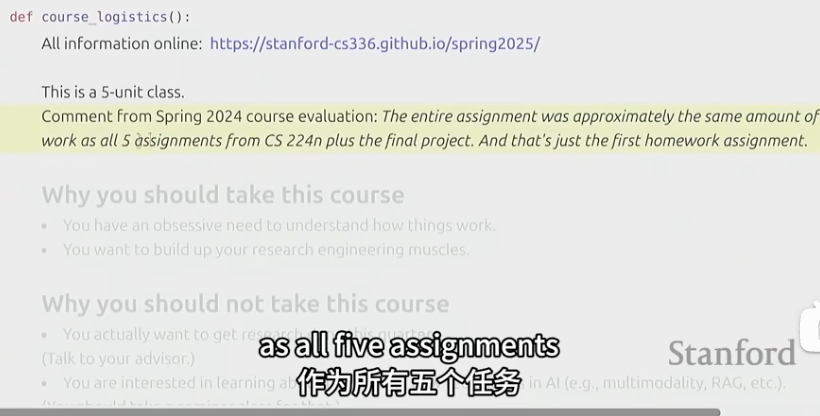
### 【1.3.1】开源级别



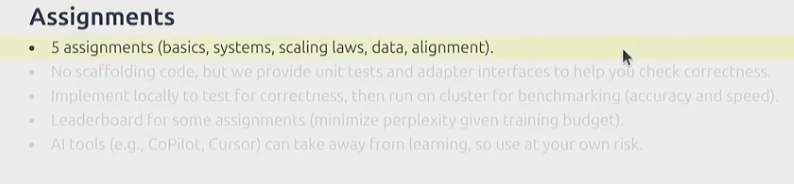
### 【1.3.2】现在前沿模型



## 【1.4】课程逻辑

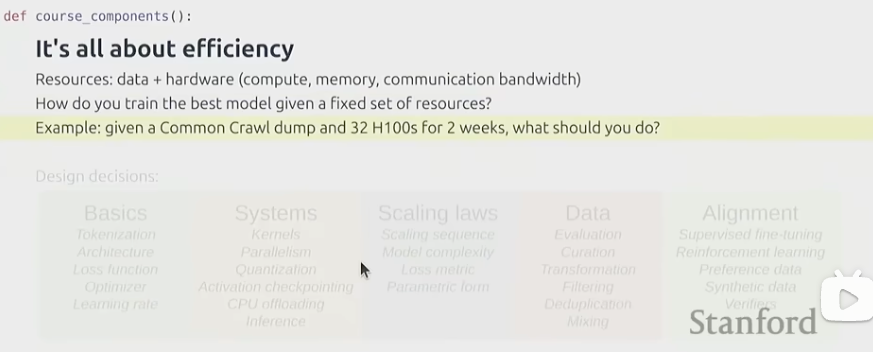


### 【1.4.1】5个家庭作业



## 【1.5】课程组成部分

### 【1.5.1】关于效率

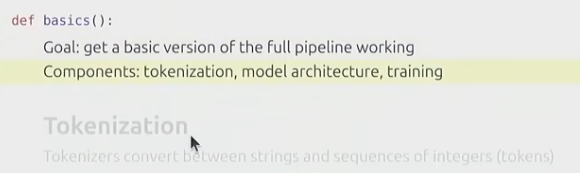


有很不同的设计决策，包括基础知识，系统，扩展规则，数据，对齐等。



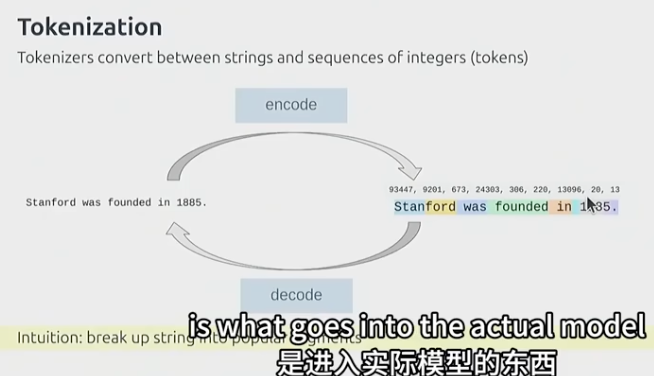
### 【1.5.2】基础知识-basic

组成部分：分词，模型架构，训练。



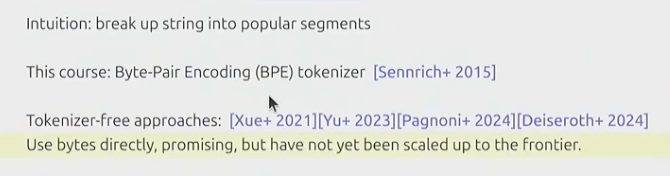
#### 【1.5.2.1】分词

分词器实现字符串与整形数字的相互转换。



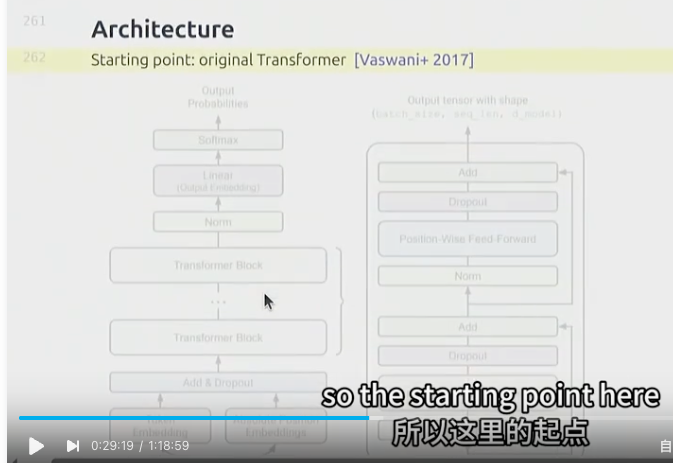
#### 【1.5.2.2】本节讨论双向编码BPE。

这节课，我们讨论双向编码（byte-pair encoding 字节对编码）。

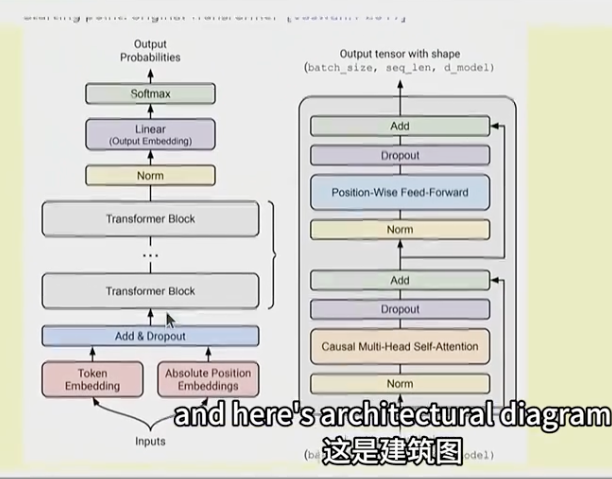


#### 【1.5.2.3】定义模型架构

在把字符串标记为整数序列后，现在我们定义一个整体模型架构。



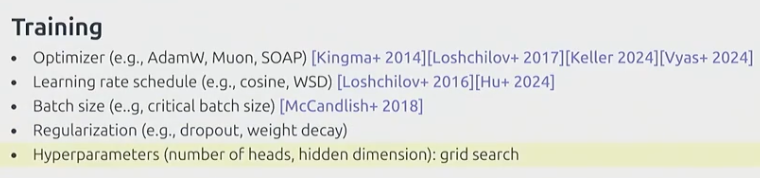
起点是：原始转换器。



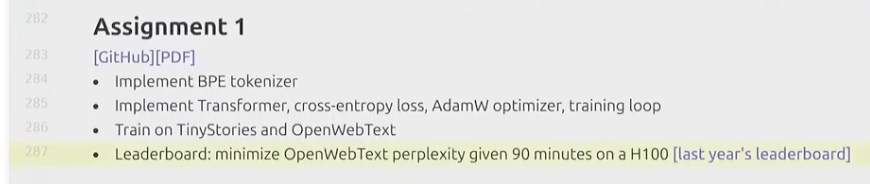
#### 【1.5.2.4】变体架构



#### 【1.5.2.5】训练模型

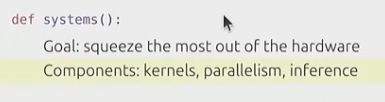


#### 【1.5.2.6】作业1



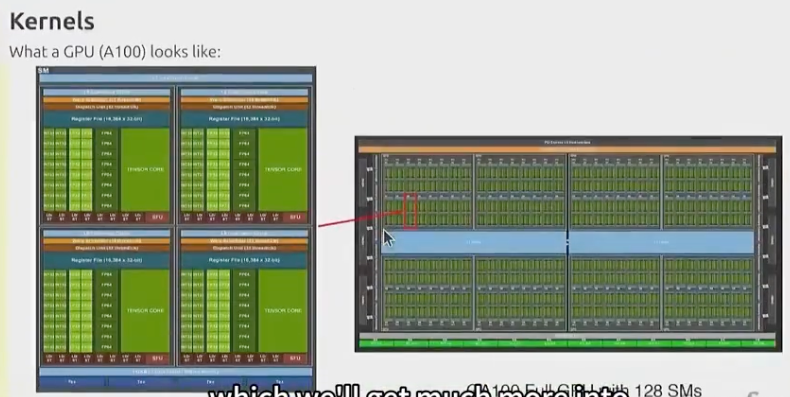
### 【1.5.3】系统知识system

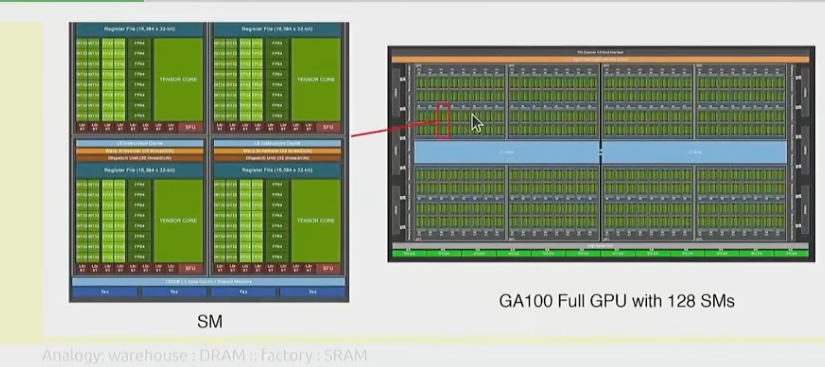
如何充分利用硬件； 包括内核，并行，推理。



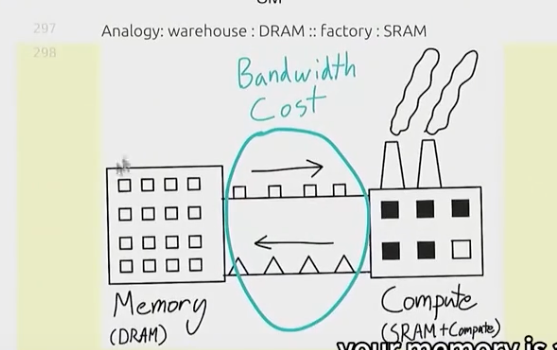
#### 【1.5.3.1】内核kernel

GPU（A100）是什么样子？





一个简单的类比： 仓库：



最大成本是数据移动成本。

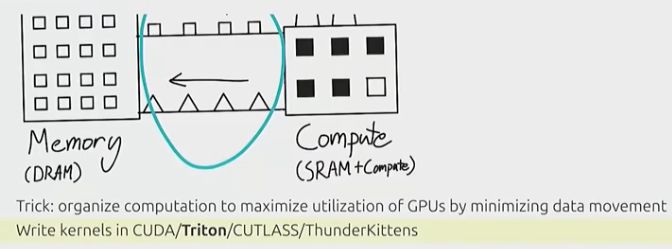
你的内存，你可以存储数据，像个仓库。

你的电脑，像个工厂。

问题：

如何组织计算能够最小化数据移动实现最大化使用GPU。

使用CUDA/Triton/CUTASS/ThunderKittens 编写内核（即GPU）。



CHAPTER-01 00:36:19

#### 【1.5.3.2】并行运行

如果我们有多个GPU将会怎样？（如8个 A100）



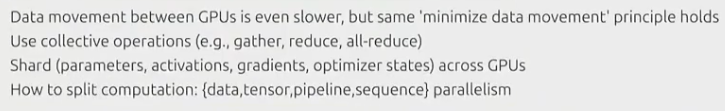
多个GPU连接到CPU。

多个GPU可以通过MV开关直接连接。

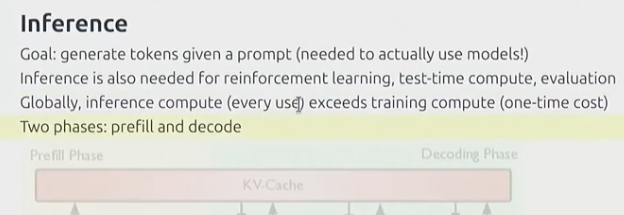
【问题】

GPU间的数据移动非常慢，所以我们需要清楚如何将模型参数，激活和梯度，并将它们放在GPU上，并进行计算并最小化数量。

然后我们要探索不同的数据并行等技术类型和张量并行性。



#### 【1.5.3.3】推理-Inference



目标：根据给定提示词生成词条token；

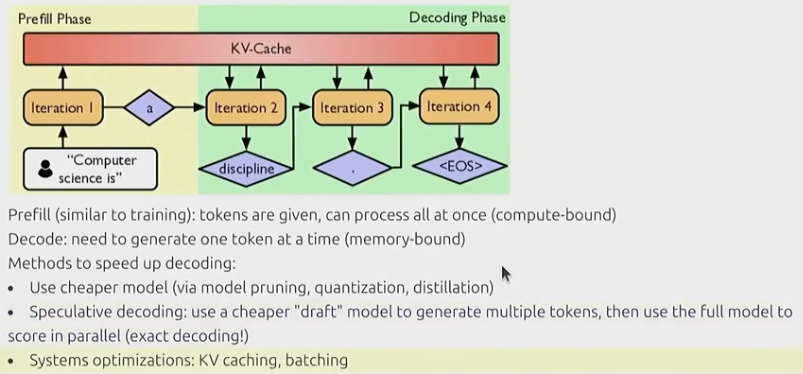
推理也被强化学习，测试计算时间，模型评估所需要；

整体上，推计算量超出了训练计算量。

推理分两个阶段： 预填充与解码。

预填充：接受提示词；你可以在模型中运行它并得到一些激活；

解码：自动生成词条token；



预填充（与训练类似）：给定词条，可以一次性所有给定的词条；

解码：一次生成一个词条；

加速解码的方法：

1. 使用便宜模型（通过模型剪枝，量化，下钻）
2. 推测解码： 使用更便宜的草稿模型，来生成多个词条，然后，我们使用这个完整模型计算并行分数；
3. 系统优化：kv缓存，批处理；

00:37:25 [on 2025-01-07]