人工智能编译器与大模型系统训练营 2024冬

大模型推理系统01：基础知识

讲师　　潘泽众

课程内容

- 大模型推理系统基础知识
- 大模型原理与Transformer结构
- 大模型推理与应用
- 大模型分布式推理与优化（两课时）

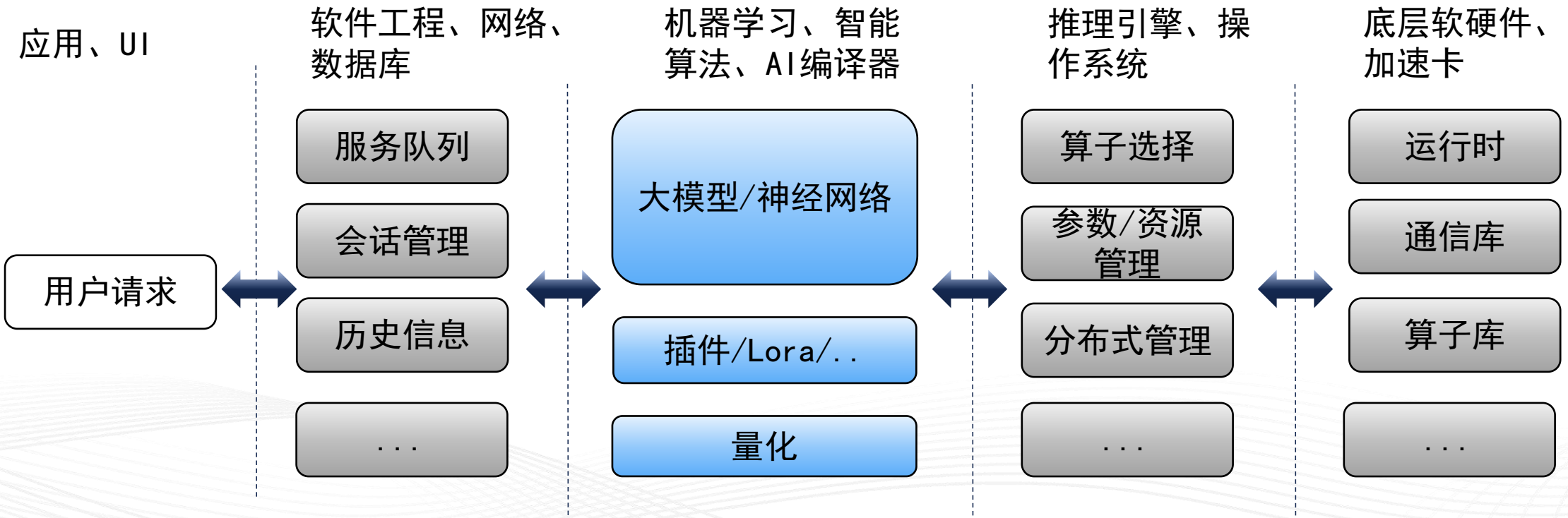
作业：

一个简单语言模型推理系统（Rust）

课程阶段：部分基础功能（算子、模型结构）的实现

项目阶段：简单AI对话功能实现

大模型推理系统



- 以服务的稳定性、高效性、可用性为目的
- 对特定的模型结构、软硬件平台做针对性的优化
- 以大模型为核心，涉及计算机技术上下游各个方面

课程目标

- 了解大模型和神经网络的基本原理（面向新人）
- 学习大模型推理的计算方式（核心）
- 实际尝试搭建大模型应用和服务
- 初步学习一些大模型推理的优化

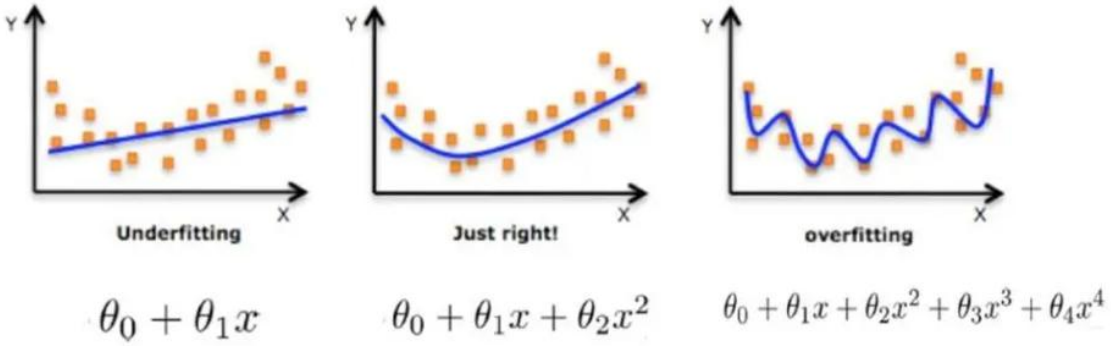
何为大模型?

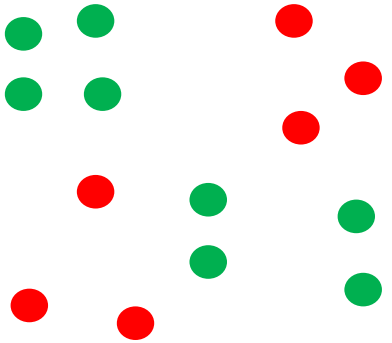
大模型是指具有大规模参数和复杂计算结构的机器学习模型。这些模型通常由深度人工神经网络构建而成。

什么是人工神经网络？

传统模型的局限

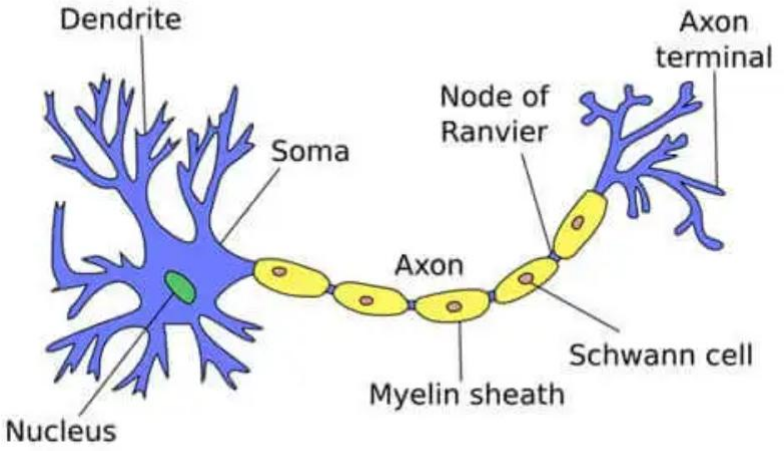
- 用已知的数学公式（如多项式），去拟合采样数据；希望寻找到一个能恰好通过这些数据点的参数集合，来完成建模；如果无法找到，就尽量逼近这些数据。有些时候优参数集合可以通过数学的方式直接求解。

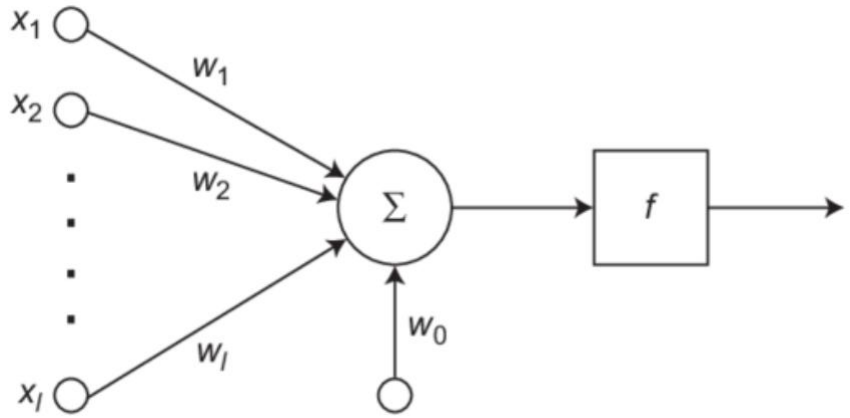




人工神经网络

- 人工神经网络（Artificial Neural Network, ANN）是一种模拟生物神经系统的计算模型。由大量的节点（或称神经元）之间相互联接构成。每个神经元都包含线性计算和非线性计算。线性计算改变了输入信号的权重和偏移，而非线性计算则代表了神经元是否激活。





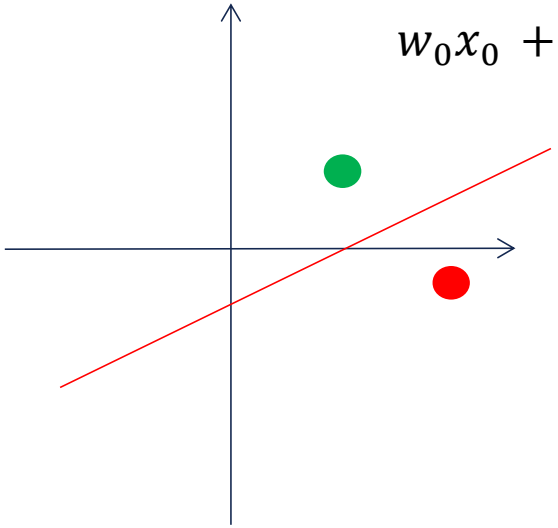
神经网络如何决策

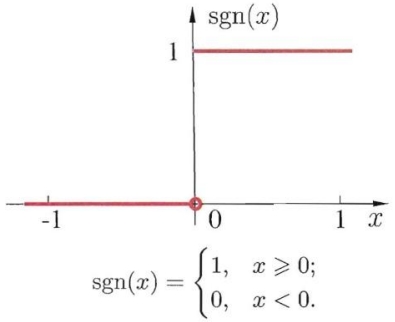
将点的(x0, x1)坐标带入进去，判断结果的正负号

一次线性计算

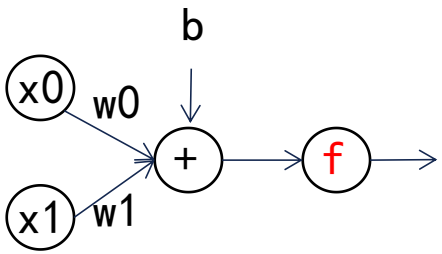
一次非线性判断

$$w_0x_0 + w_1x_1 + b = 0$$





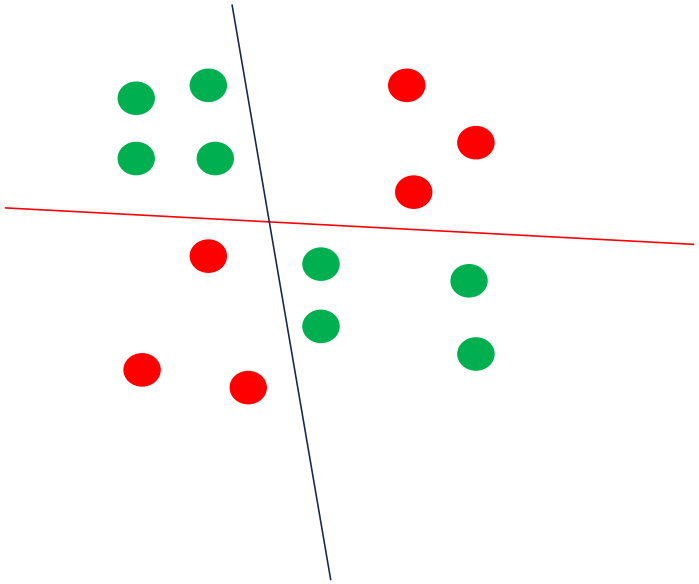
$$\operatorname{sgn}(x) = \begin{cases} 1, & x \geqslant 0; \\ 0, & x < 0. \end{cases}$$

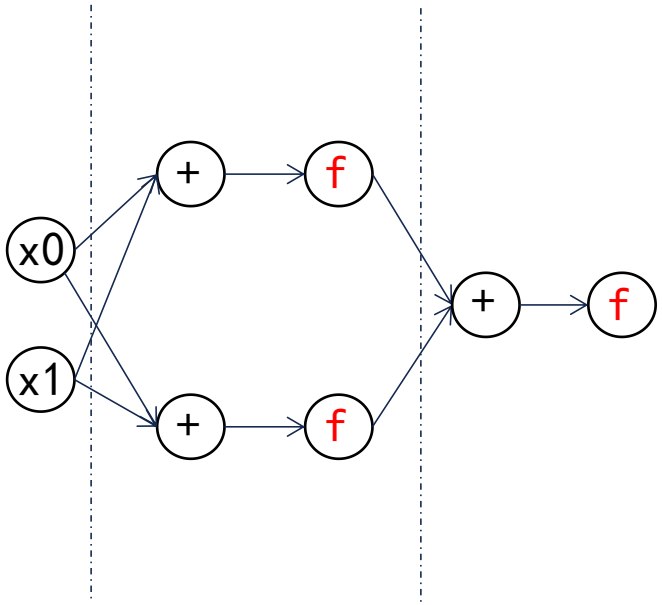


参数：

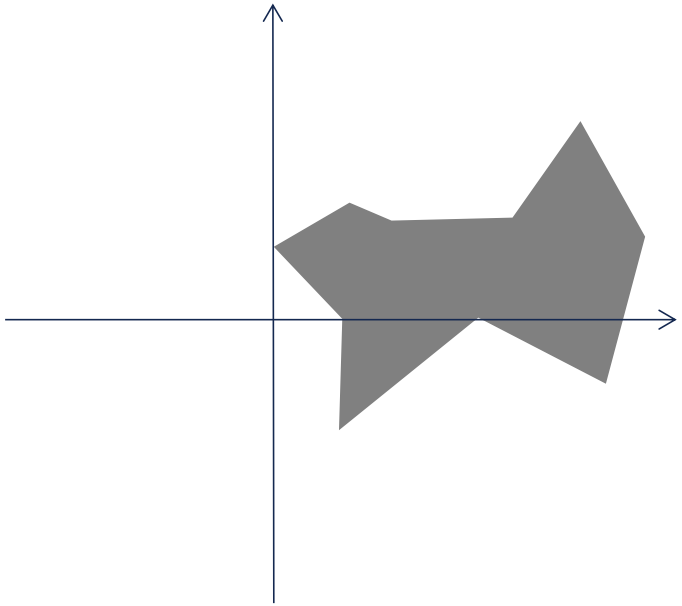
- w：权重
- b：偏移

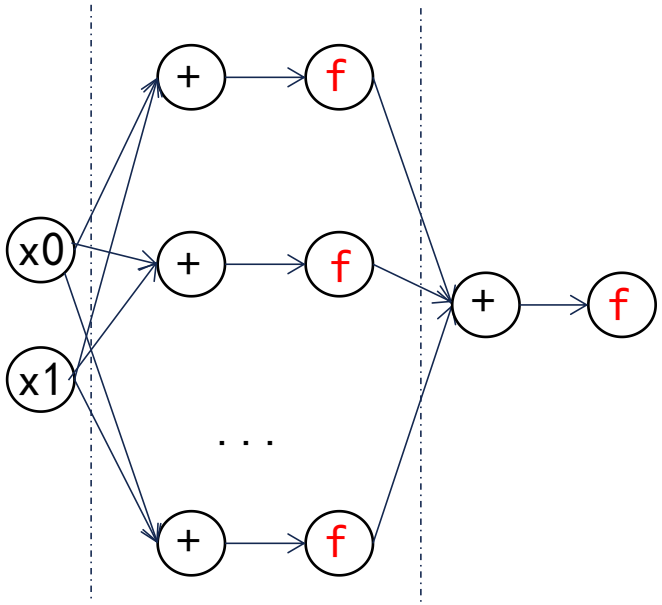
神经网络如何决策





以此类推

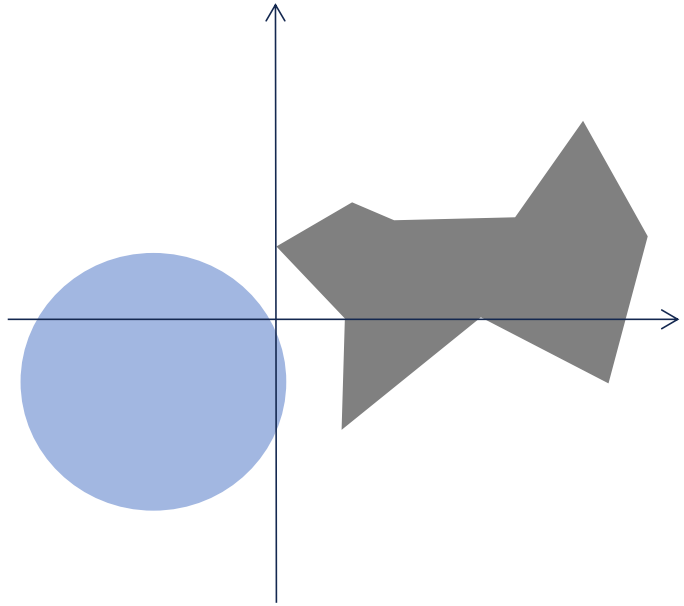


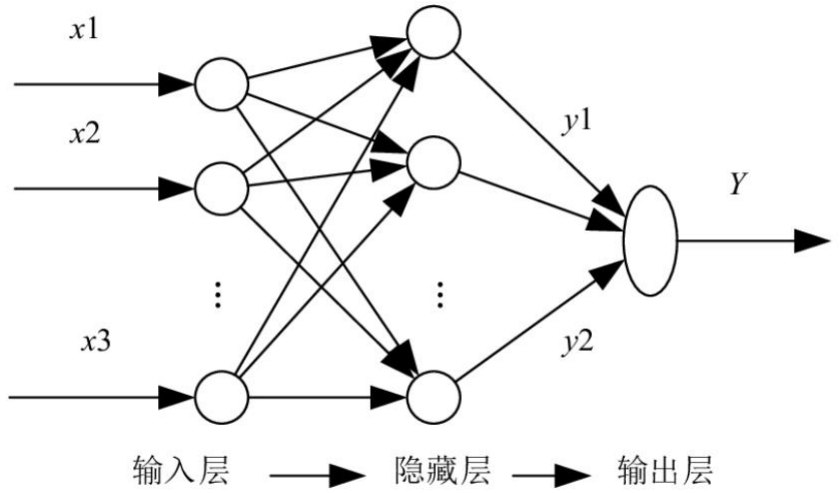


无论多复杂的多边形，你都可以通过画多根直线的方式去拟合

万能近似定理

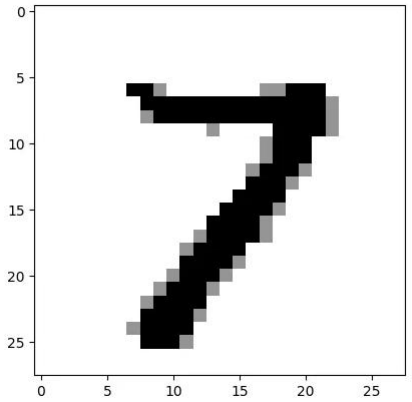
- 三层神经网络（即一个输入层、一个隐藏层和一个输出层）在理论上可以近似任意复杂（有限维度）的决策边界。



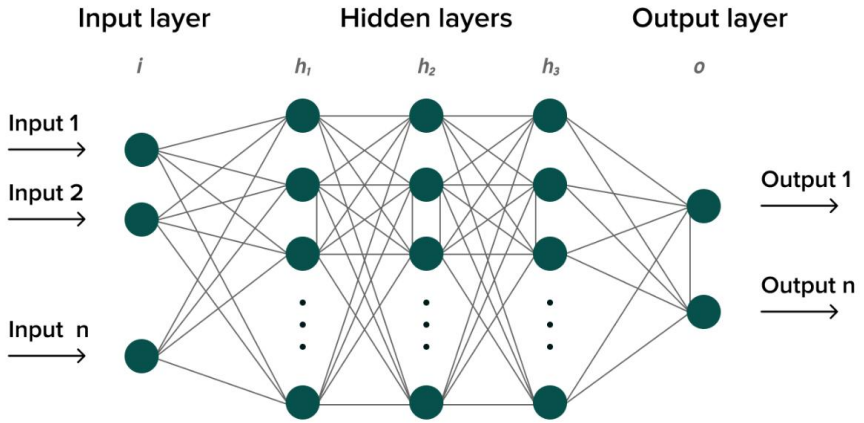


如果输入的维度更高，增加x的数量；如果输出的种类更多，增加y的数量

例子：图像分类



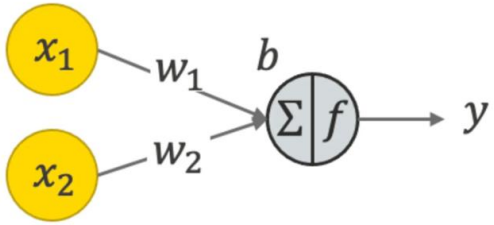
- 每张图都可以被看作一个长度28×28的向量。



0. 02	0. 3	. . . . .				

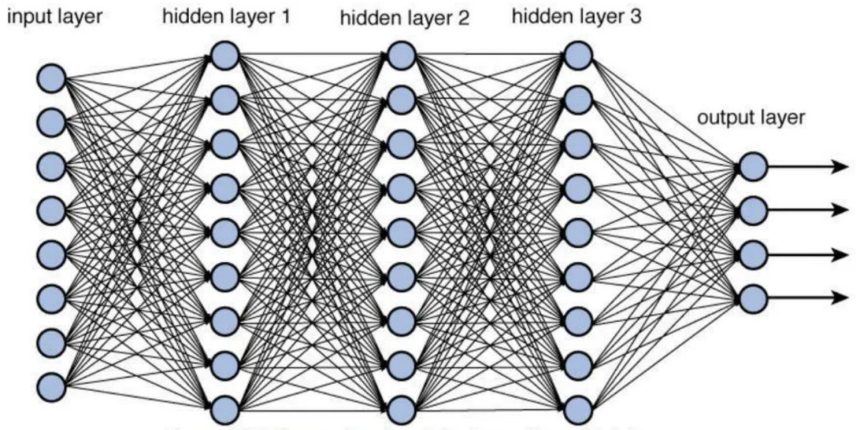
1	0	0	0	0	0	0	0	0	0

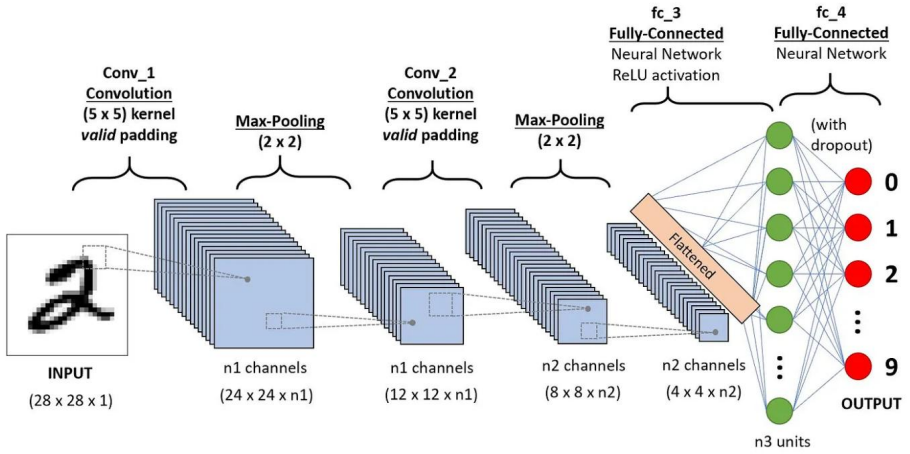
训练 vs. 推理



- 训练
  - 通过大量已知的数据来调整模型参数，使其能够更准确的对新数据进行预测
  - 定义损失函数，按照减少损失的方向调整参数。（梯度下降法）
  - 要有一个好的模型结构
- 推理
  - 使用训练好的参数，对新的数据进行预测

不同结构的深度神经网络

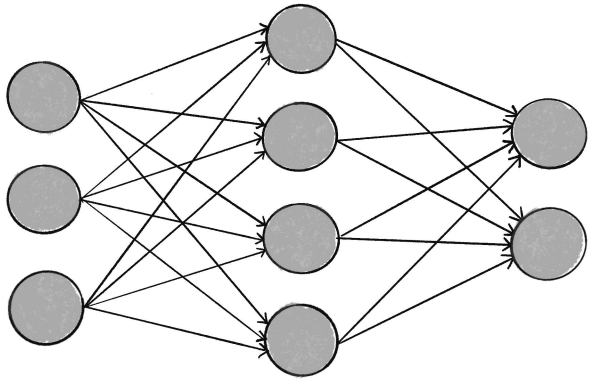




- 深度神经网络通过更多层次的结构，可以用较少的参数捕捉到更为复杂的特征关系，提升参数效率
- 一些特殊的结构，如：CNN、RNN、Transformer等，可以引导模型用更少的参数捕捉特定的特征

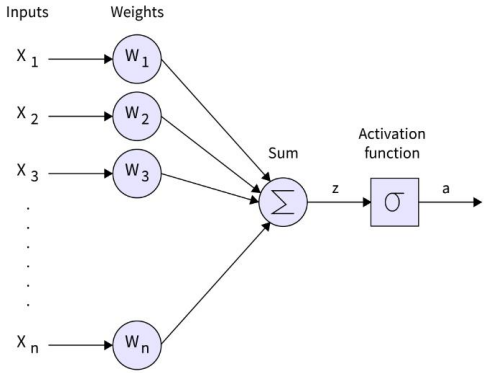
大模型核心算子：矩阵乘

- 批量进行神经元的线性计算：y = xW^T + b
- 是大模型计算中总耗时最长的算子



（每一条边都代表着一个参数）

假如输入x长度为K，中间层维度为N，那么权重矩阵的大小就是K×N。
实践中我们可以对M个输入一起进行计算，即M×K的输入和K×N的权重进行矩阵乘。

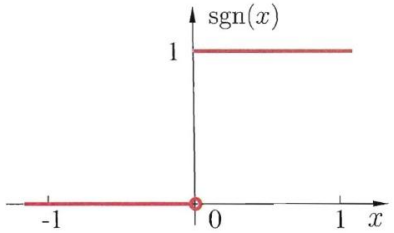


$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} u & v \\ w & x \\ y & z \end{bmatrix} = \begin{bmatrix} au+bw+cy & av+bx+cz \\ du+ew+fy & dv+ex+fz \end{bmatrix}$$

大模型算子：激活函数

Neural Network Activation Functions: a small subset!

ReLU $\max(0, x)$	**GELU** $\frac{x}{2}\left(1+\tanh\left(\sqrt{\frac{2}{\pi}}(x+ax^3)\right)\right)$	**PReLU** $\max(0, x)$
ELU $\begin{cases} x \text{ if } x>0 \\ \alpha(x\exp x-1) \text{ if } x<0 \end{cases}$	**Swish** $\frac{x}{1+\exp -x}$	**SELU** $\alpha(\max(0,x)+\min(0,\beta(\exp x-1)))$
SoftPlus $\frac{1}{\beta}\log(1+\exp(\beta x))$	**Mish** $x\tanh\left(\frac{1}{\beta}\log(1+\exp(\beta x))\right)$	**RReLU** $\begin{cases} x \text{ if } x\geq 0 \\ ax \text{ if } x<0 \text{ with } a\sim\Re(l,u) \end{cases}$
HardSwish $\begin{cases} 0 \text{ if } x\leq -3 \\ x \text{ if } x\geq 3 \\ x(x+3)/6 \text{ otherwise} \end{cases}$	**Sigmoid** $\frac{1}{1+\exp(-x)}$	**SoftSign** $\frac{x}{1+\lvert x\rvert}$
Tanh $\tanh(x)$	**Hard tanh** $\begin{cases} a \text{ if } x\geq a \\ b \text{ if } x\leq b \\ x \text{ otherwise} \end{cases}$	**Hard Sigmoid** $\begin{cases} 0 \text{ if } x\leq -3 \\ 1 \text{ if } x\geq 3 \\ x/6+1/2 \text{ otherwise} \end{cases}$
Tanh Shrink $x-\tanh(x)$	**Soft Shrink** $\begin{cases} x-\lambda \text{ if } x>\lambda \\ x+\lambda \text{ if } x<-\lambda \\ 0 \text{ otherwise} \end{cases}$	**Hard Shrink** $\begin{cases} x \text{ if } x>\lambda \\ x \text{ if } x<-\lambda \\ 0 \text{ otherwise} \end{cases}$



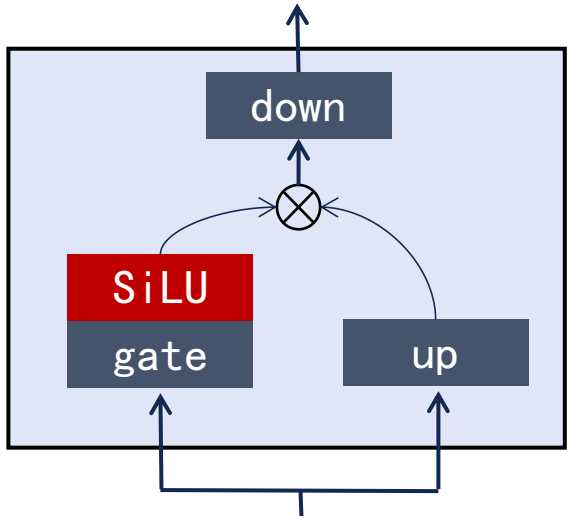
$$\operatorname{sgn}(x)=\begin{cases}1, & x\geqslant 0;\\ 0, & x<0.\end{cases}$$

训练时需要对函数求导才能进行梯度下降

0. 7	0. 1	0. 1	0. 1	0.	0.	0.	0.	0.	0.

大模型中的神经网络

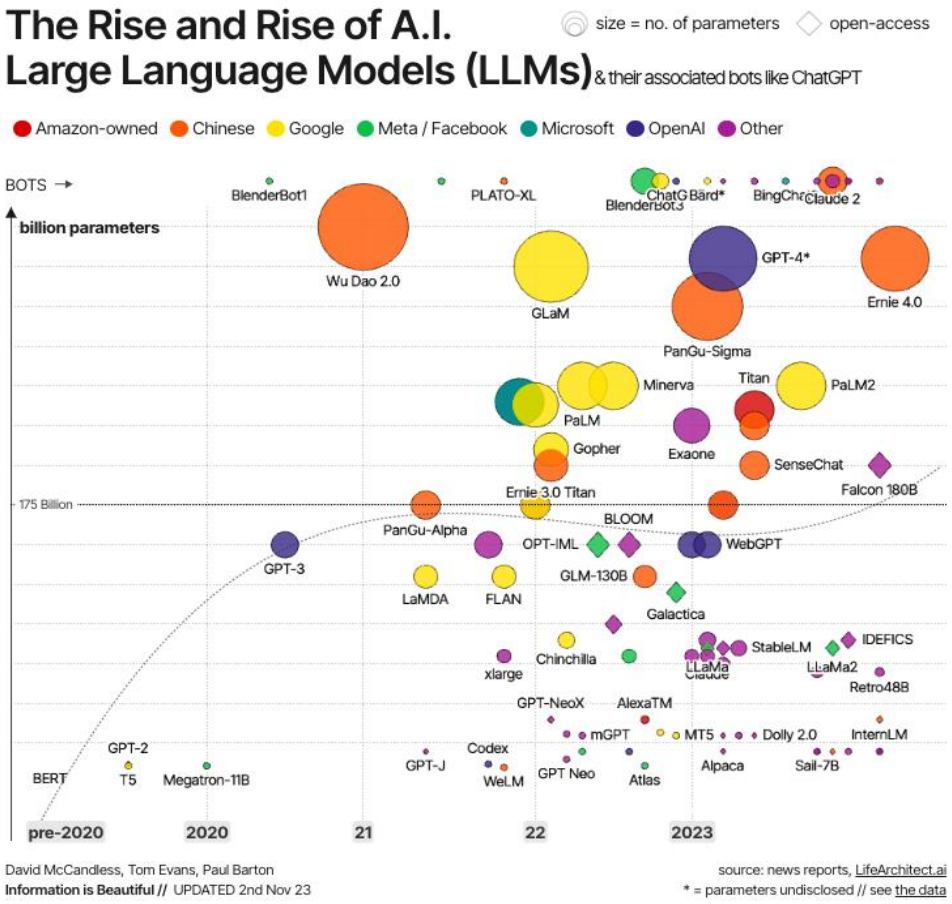
多层感知器（MLP），也被成为前馈神经网络（FFN）



$$Y = (SiLU(X \cdot W_{gate}^{T}) \times X \cdot W_{up}^{T}) \cdot W_{down}^{T}$$

- 推理时参数固定，同样的输入意味着同样的输出
- 多个输入向量之间不会互相影响，没有顺序

大模型有多大?



1B = 十亿（10^9）参数

（还要乘以数据类型大小）

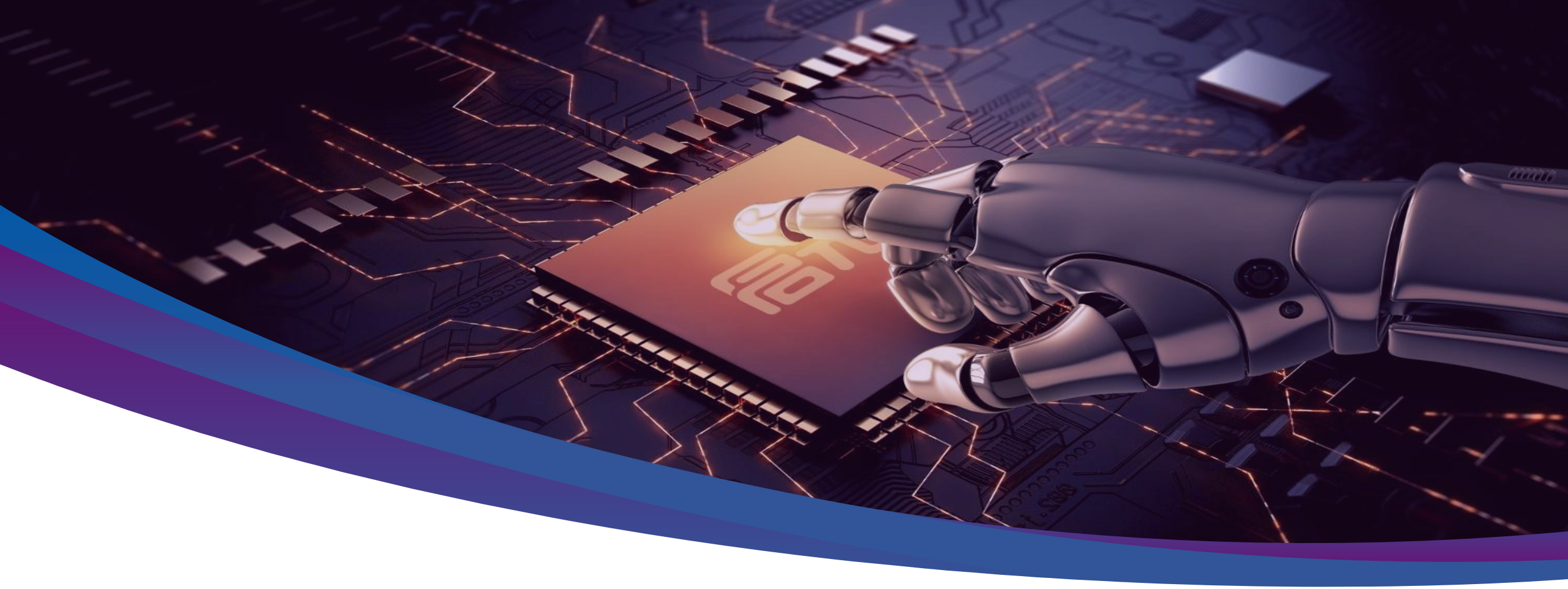
（fp16、bf16是最常见的类型，
更小的量化类型如int8成为趋势）

- 每个权重矩阵都可能有上亿个参数
- 一个模型可以有几十甚至上百层神经网络
- 其实神经网络的激活很稀疏……

作业说明

地址：https://github.com/LearningInfiniTensor/learning-lm-rs

- 项目是用Rust写的，完整项目是一个简易的AI对话程序
- 项目repo里集成了模型文件，数据类型是fp32
- **请仔细阅读Readme文件**
- 作业阶段的代码提供了单元测试，请不要修改
- 项目阶段的代码会依赖作业阶段的代码



感谢聆听

GitHub开源组织：https://github.com/InfiniTensor