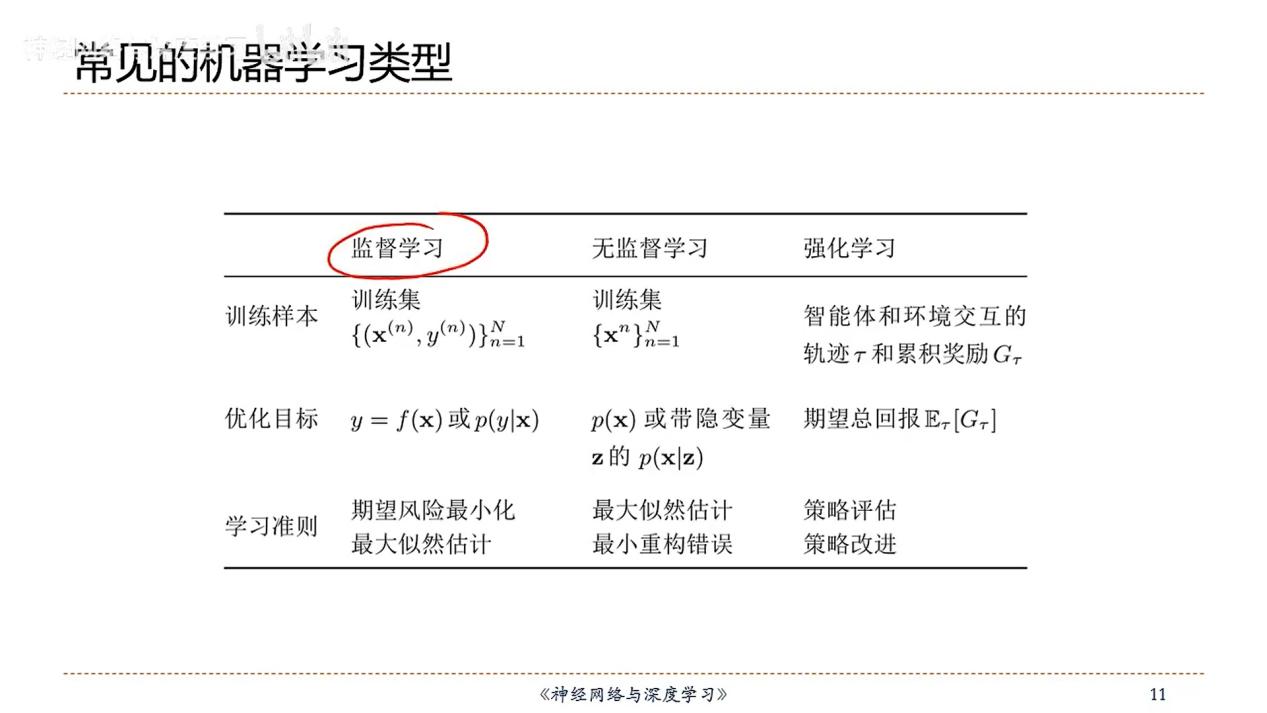
**一、机器学习概述**

1.机器学习类型

监督学习：有x（训练集）和y（输出）——回归，分类

无监督学习：有x无y，自动学习归类——聚类，降维，密度估计

强化学习：不断尝试试错，通过与环境交互学习



1. 机器学习要素

数据，模型，学习准则，优化算法

损失函数：用来量化模型预测和真实标签之间的差异

期望风险：通过大数定律进行近似，得到经验风险

最后应该让经验风险最小化

最优化问题：

梯度下降法计算最优解

随机梯度下降法：每次迭代的时候只采集一个样本，经过足够的迭代次数也可以收敛到局部最优解，优点是不需要所有的数据集，计算开销小，但是无法充分利用计算机的并行计算

小批量-随机梯度下降法：少选点儿，可以提高训练效率（常用！）

1. 泛化与正则化

欠拟合问题：由于模型能力不够，模型错误率过高

过拟合问题：由于模型能力太好，模型错误率过低，一般由于训练数据少和噪声多

正则化（损害优化的方法）减少泛化（过拟合）误差：增加优化约束或者干扰优化过程，可以引入验证集，当验证集上的错误率不下降的时候就停止随机梯度下降迭代！

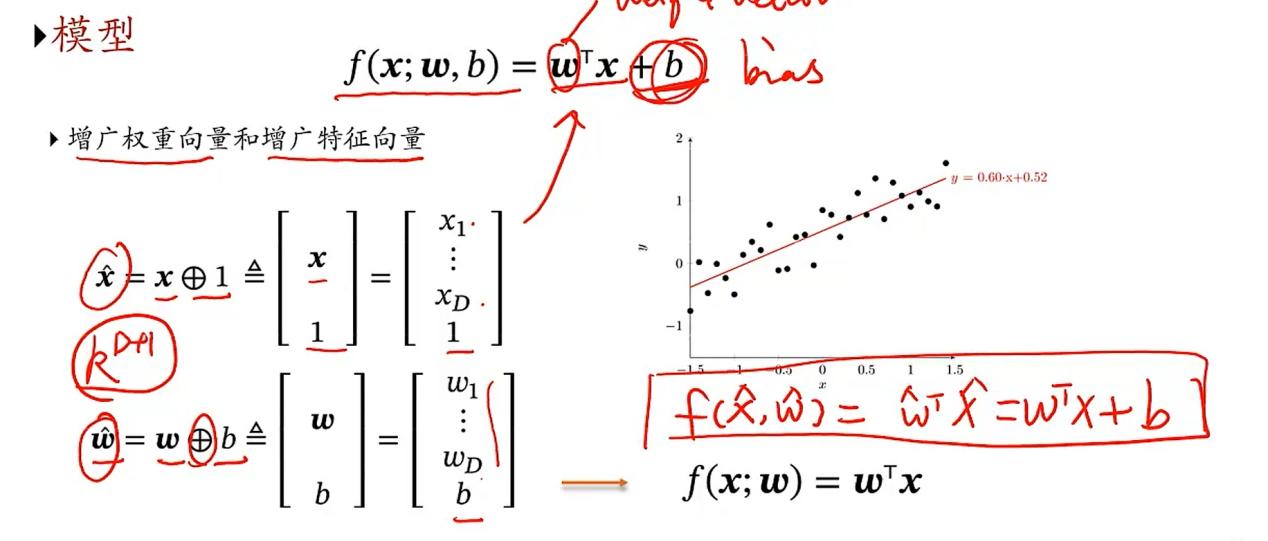
1. 线性回归

线性回归模型（linear regression），训练集{x，y}

用“y-hat”表示预测数值

用f\_w,b（）表示线性函数y=wx+b，w是权重向量，b是偏置

增广权重向量（w-hat：w拼接b）和增广特征向量（x-hat：x拼接1）



经验风险最小化（ERM）：采用平方损失函数计算

结构风险最小化（SRM）：在经验风险的基础上增加对w的限制，

矩阵微积分

1. 多项式回归

建模非线性，如果采用线性会欠拟合

F(x,w)=w0+w1x+w2x^2……+wnx^n

损失函数采用平方损失函数

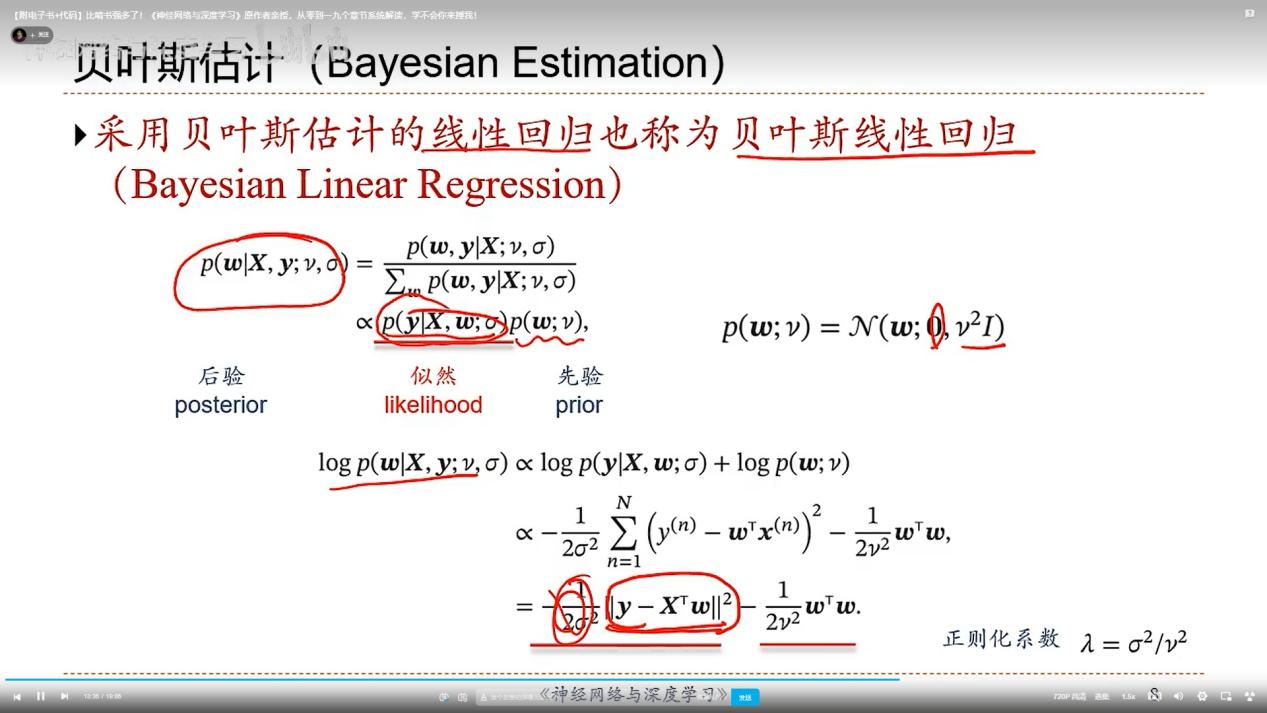
正则化系数λ：控制模型正则化程度（lnλ = -18）

1. 线性回归的概率视角

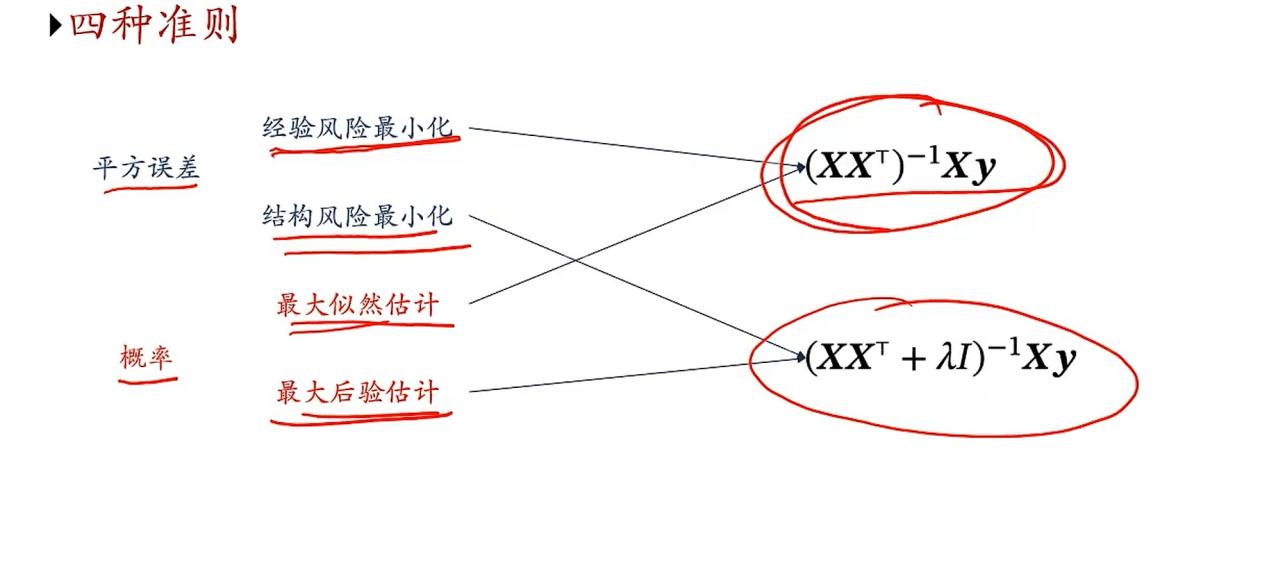
最大似然函数

贝叶斯学习：将w也看做随机变量，给定一组观测数据X，求P（w|X）【后验分布】

贝叶斯公式，贝叶斯线性回归【最大后验估计】



总结:



1. 选择模型：引入验证集（Validation set）

交叉验证：解决数据稀疏问题，将训练集分为S组，每次使用S-1组作为训练集，1组作为验证集，取验证集上平均性能最好的一组

赤池信息量准则，贝叶斯信息准则

1. 常用的定理
2. 没有免费午餐定理：没有对于所有问题都有效的算法
3. 丑小鸭定理：具体问题具体分析，特征不同
4. 奥卡姆剃刀定理：能用简单模型就不用复杂模型
5. 归纳偏置：对学习的问题做一些假设，假设称为归纳偏置（先验）
6. PAC学习：经验风险趋近于期望风险，泛化误差趋近于0【训练集趋向无穷大】

**二、有监督学习**

**线性模型**

1. 分类问题【CIFAR-10】

图像分类，目标检测，实例分割

1. 线性分类模型

！Y是离散的标签，通过阈值函数将线性数据转化为离散的

1. 二分类问题

采用0-1损失函数，标签y只有俩，将空间划为2份

1. 多分类问题

标签y多于2，一对其余方式；一对一方式；**argmax方式**

1. 交叉熵和对数似然

信息论中的熵 ：熵越高随机变量信息越多，反之越少

自信息：一个随机事件所包含的信息量，具有可加性

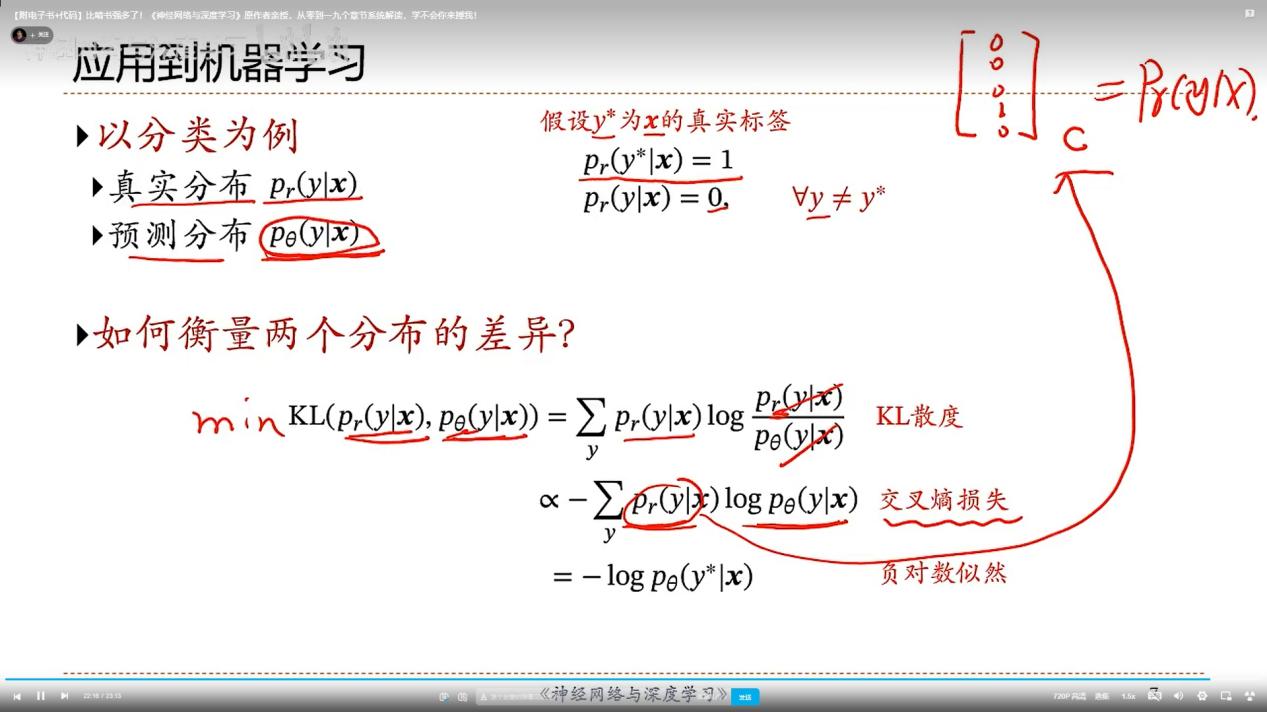
熵：随机变量X的自信息的数学期望

熵编码【类似概率论中的哈夫曼编码】

交叉熵按照概率分布q的最优编码对真实分布为p的信息进行编码的长度，给定q的情况下，p和q越接近，交叉熵越小，反之越大

KL散度用q来近似p的时候所造成的信息损失量

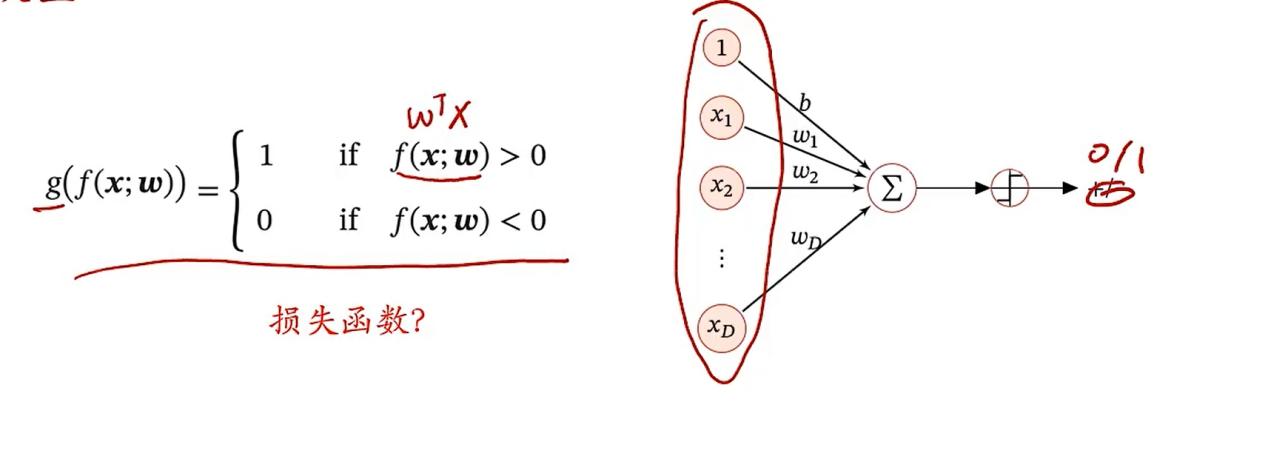
应用到机器学习：



1. Logistic回归

解决二分类问题

对数几率回归，对率回归

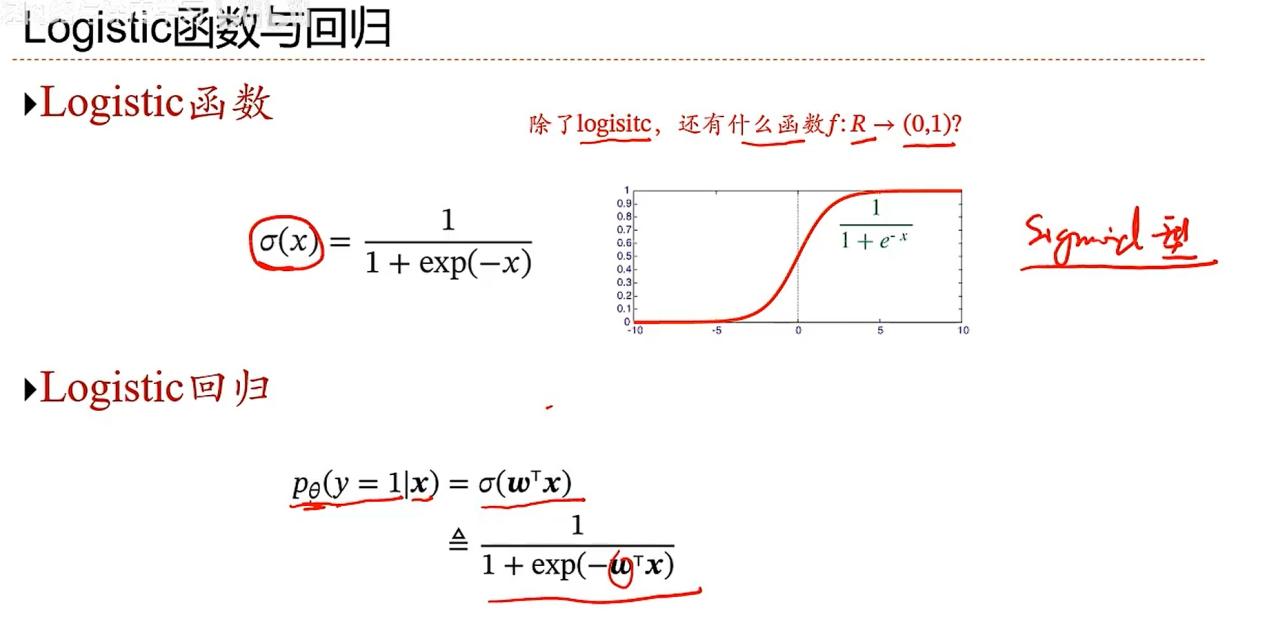


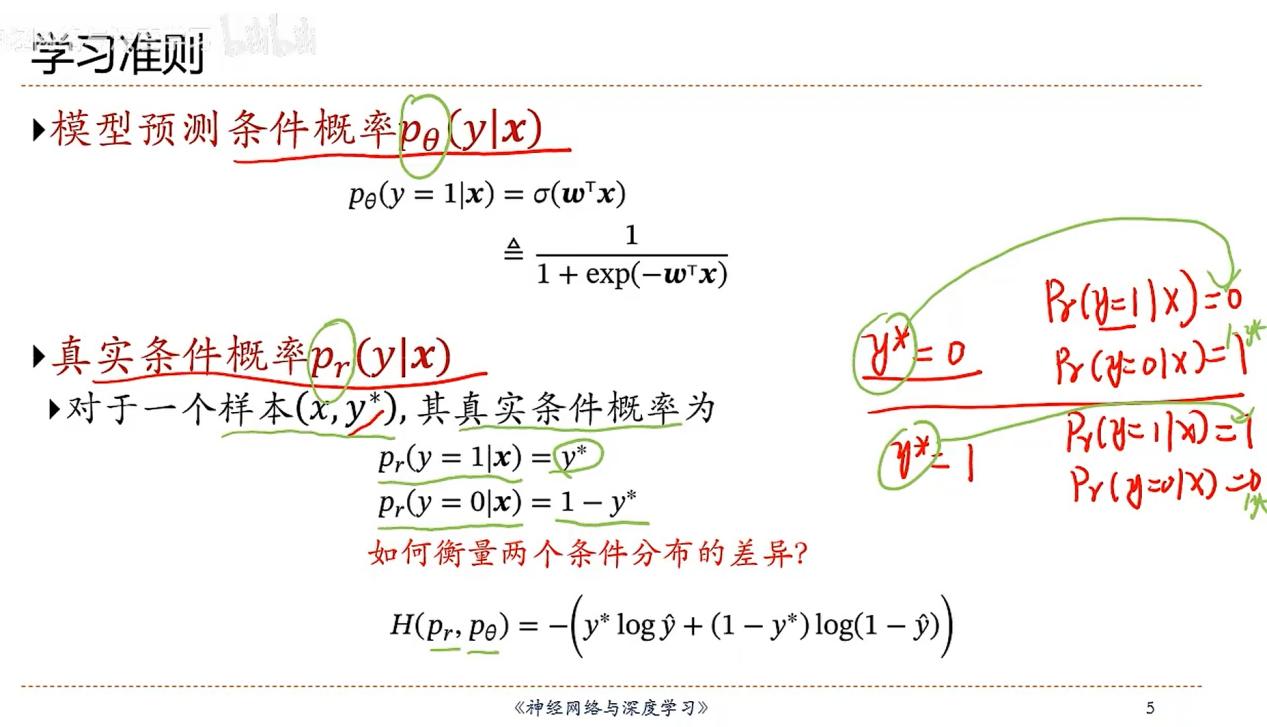
利用交叉熵构造损失函数

判别函数f：一个线性函数

激活函数g:把线性函数的值域从实数区间挤压到（0,1）之间以表示概率

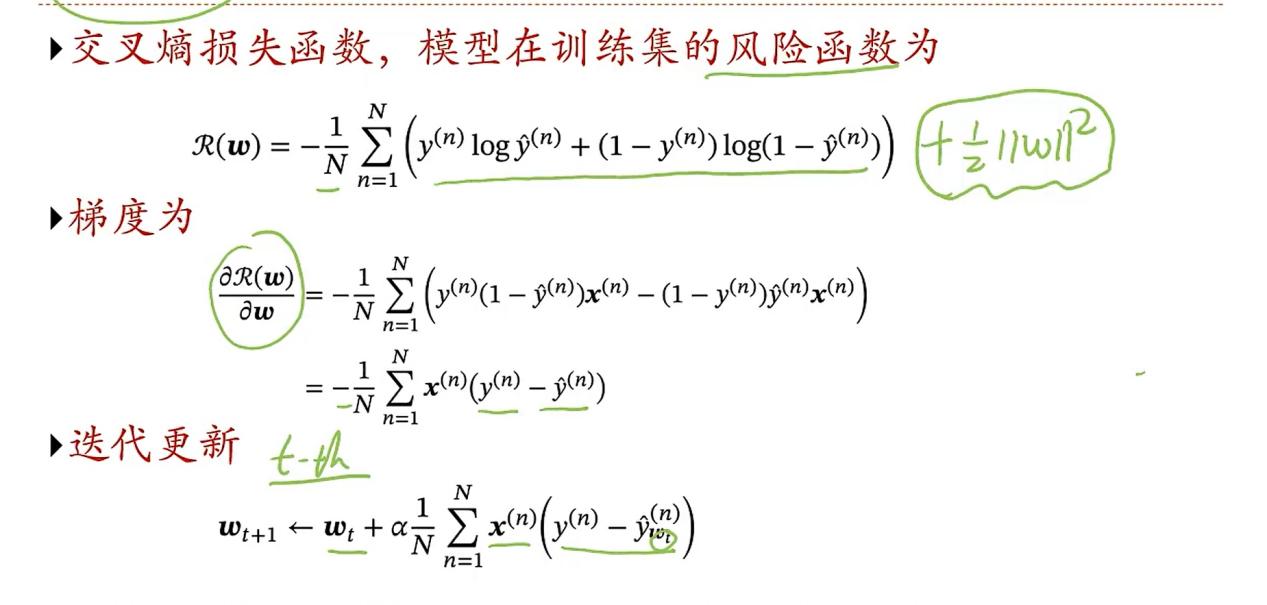
Logistic函数：

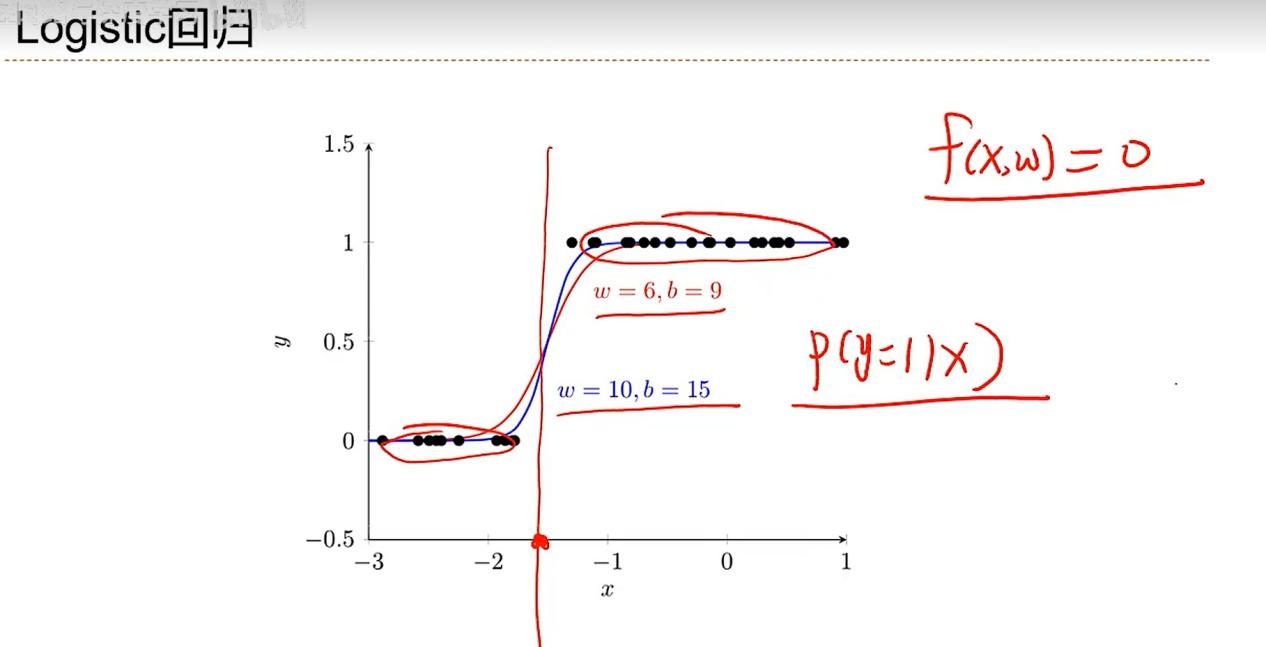




H就是交叉熵

梯度下降：





1. Softmax回归

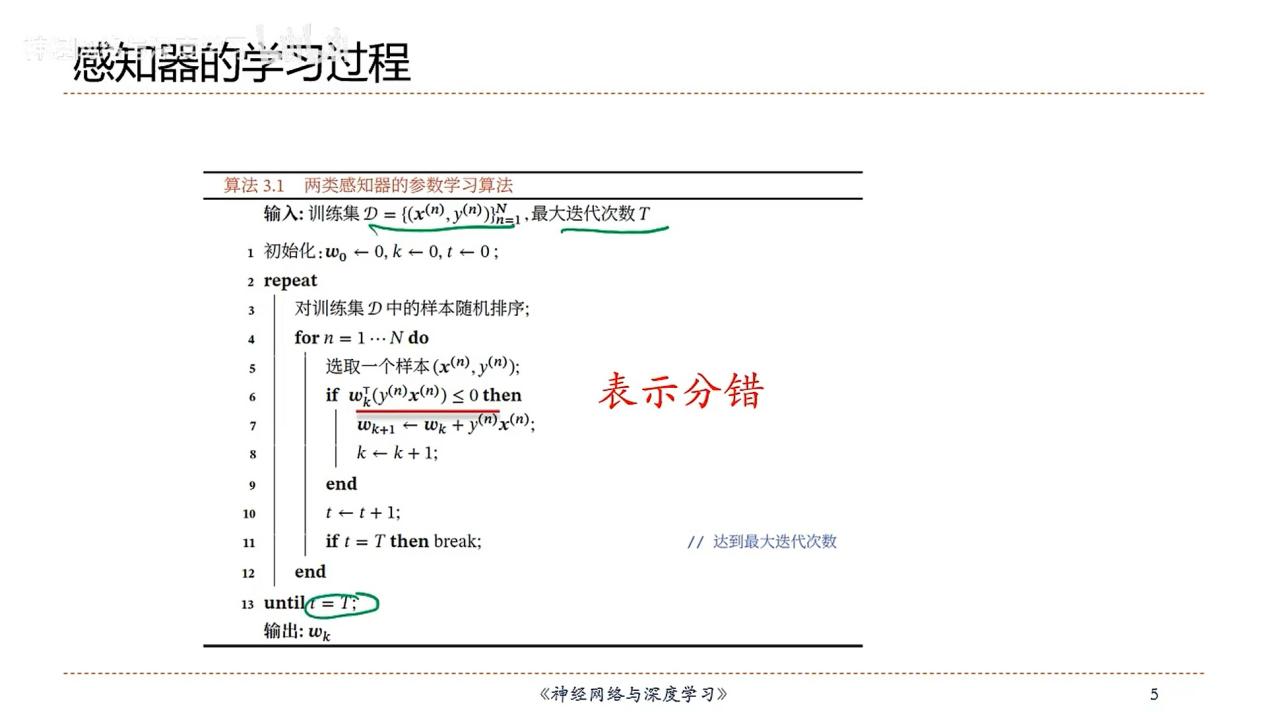
负责解决多分类问题，标签y可以有多个取值

Softmax函数

1. 感知器

模拟深恶神经元行为的机器，输出+1和-1

错误驱动的在线学习算法



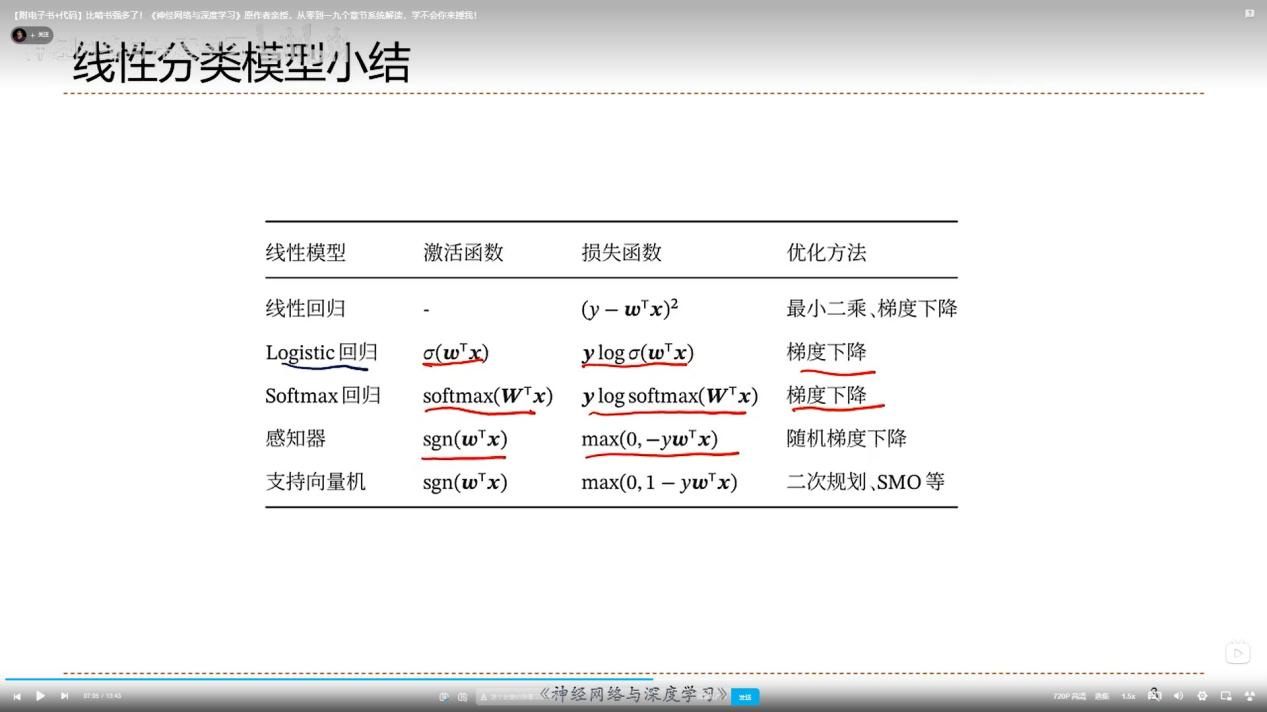
相比logistic回归，感知器更新收敛速度快，对于犯错差没有太在意

1. 支持向量机

比较好的模型，所有的样本离分界面越远（间隔大），泛化性越好

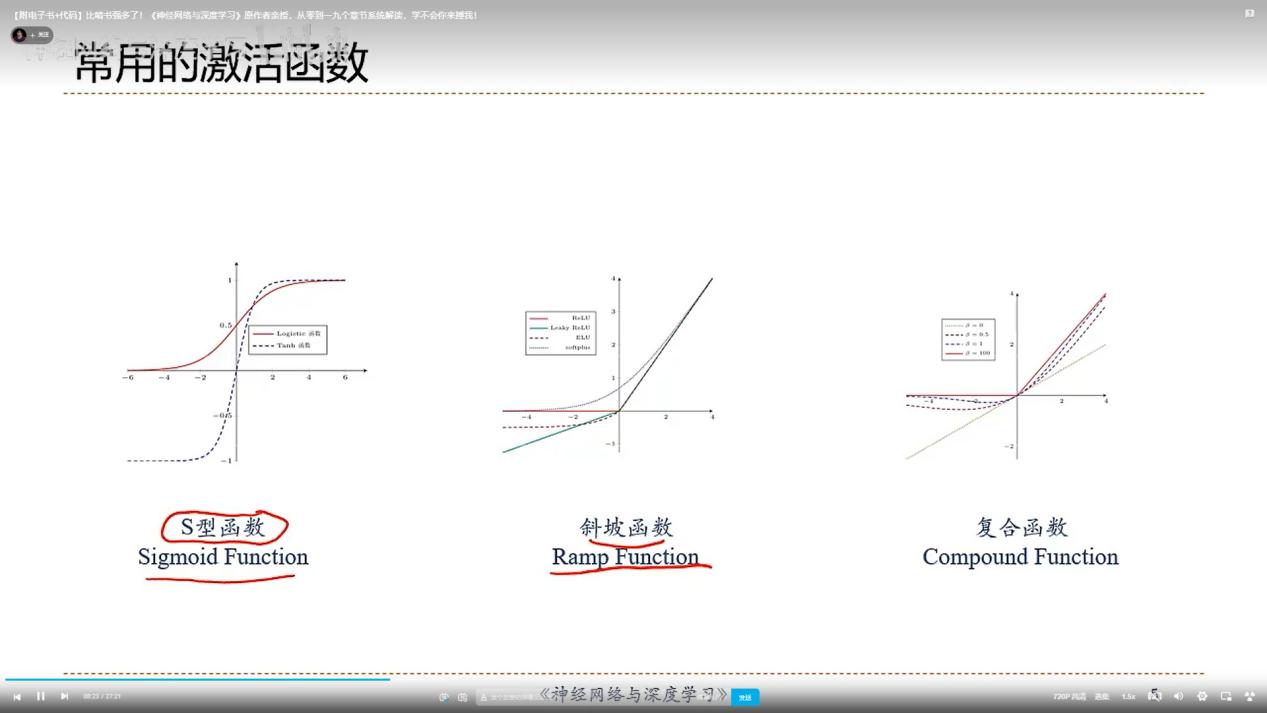
优化准则为间隔

软间隔：如果不是线性可分的无法找到最优解，引入松弛变量让尽可能小



**前馈神经网路（FNN）**

1. 激活函数的性质：
   1. 连续并可导的非线性函数
   2. 激活函数和导函数尽可能简单，可以提高计算效率
   3. 导函数值域需要在一个合适的区间，否则影响训练的效率和稳定性
2. 常用激活函数：



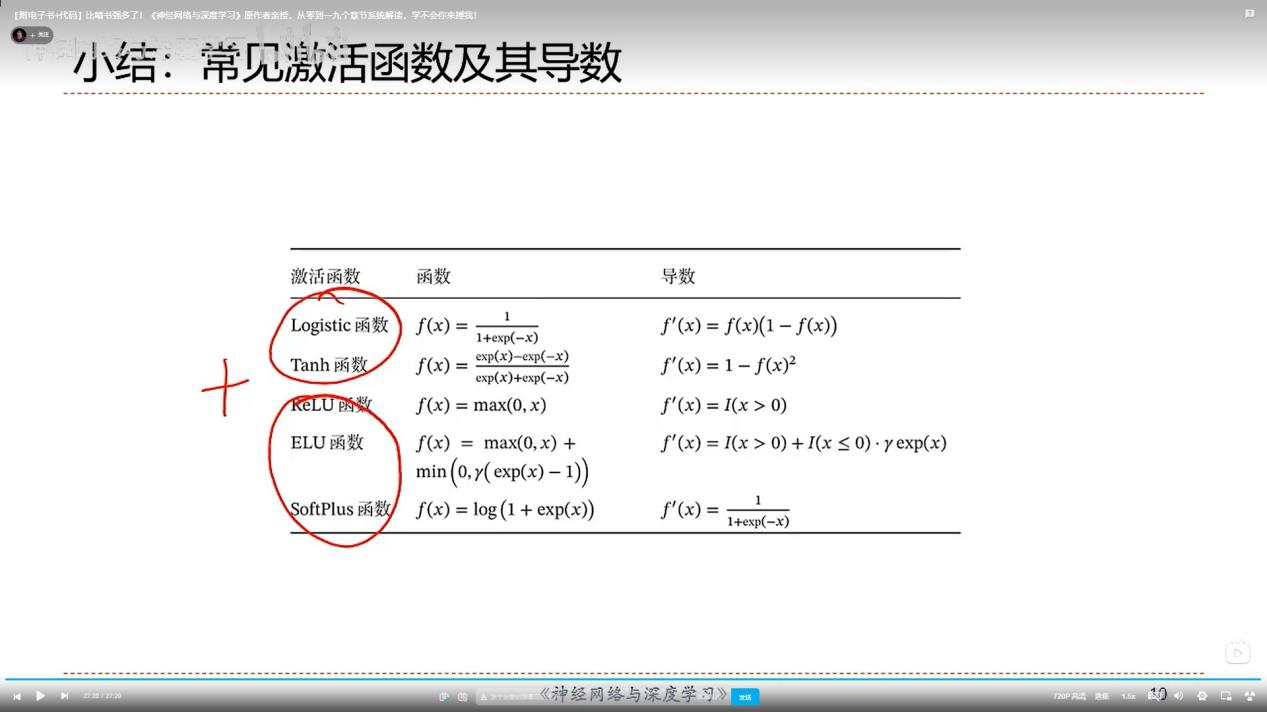
S型函数Sigmoid

斜坡函数ReLU：计算高效，一定程度上缓解梯度消失问题，存在死亡ReLU问题

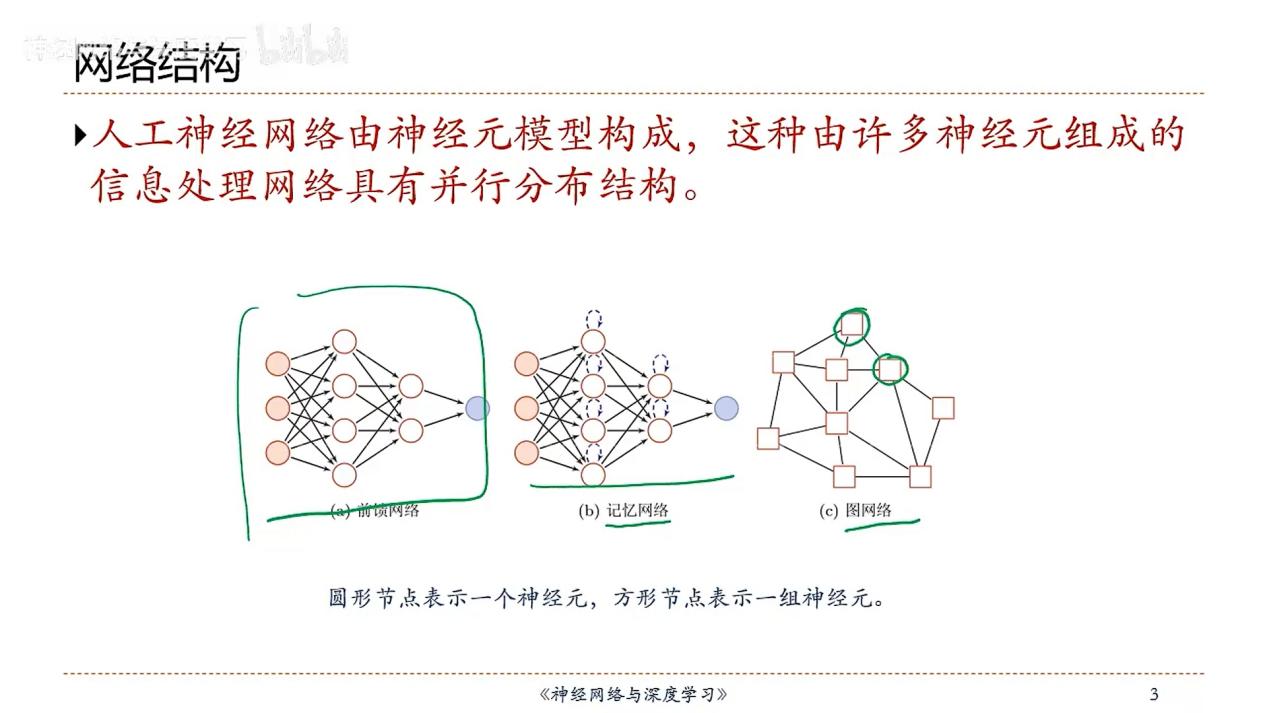
复合函数：

Switch函数：一种自门控激活函数，变化β让性质在线性函数和复合函数之间

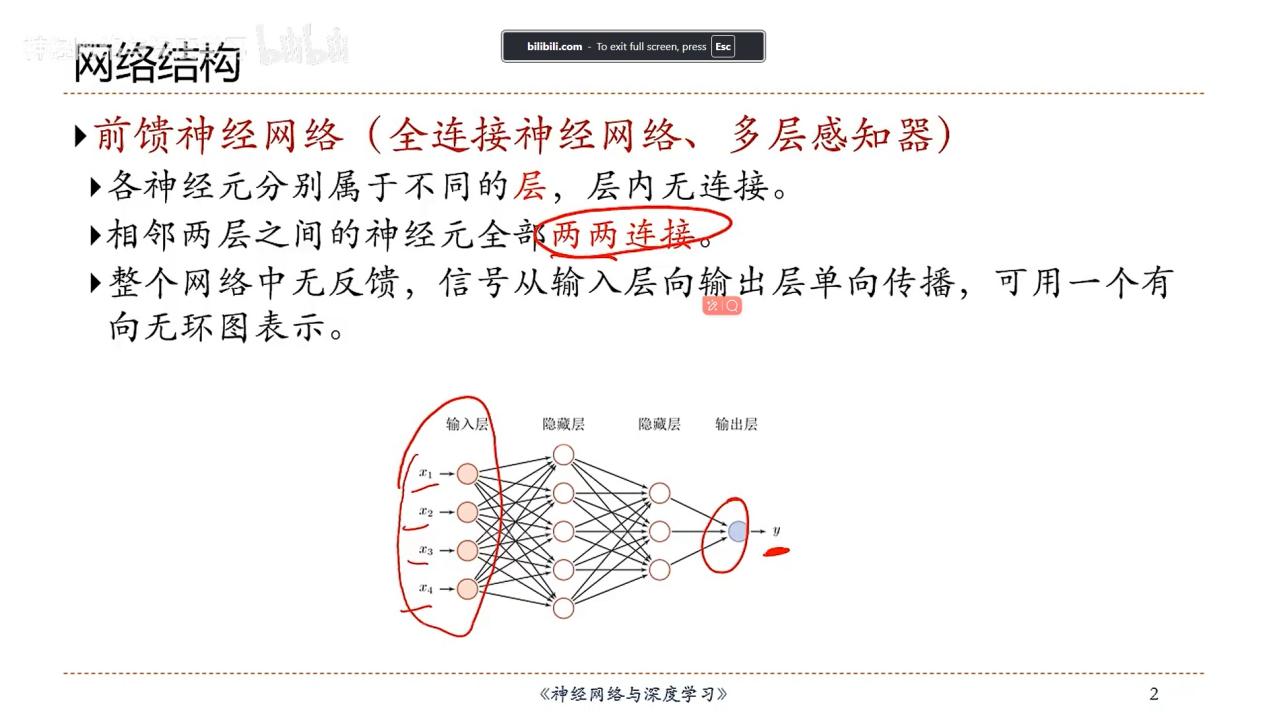
GELU函数：高斯误差线性单元，一般用Tanh函数或Logistic函数近似

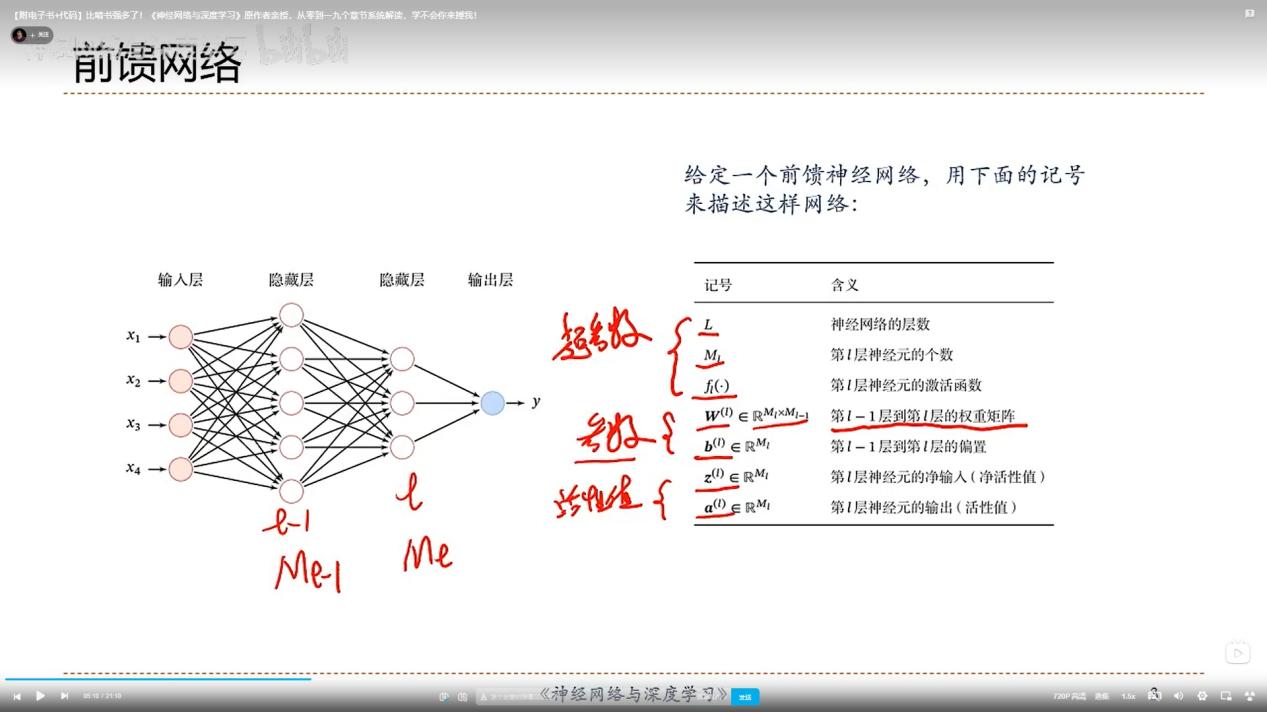


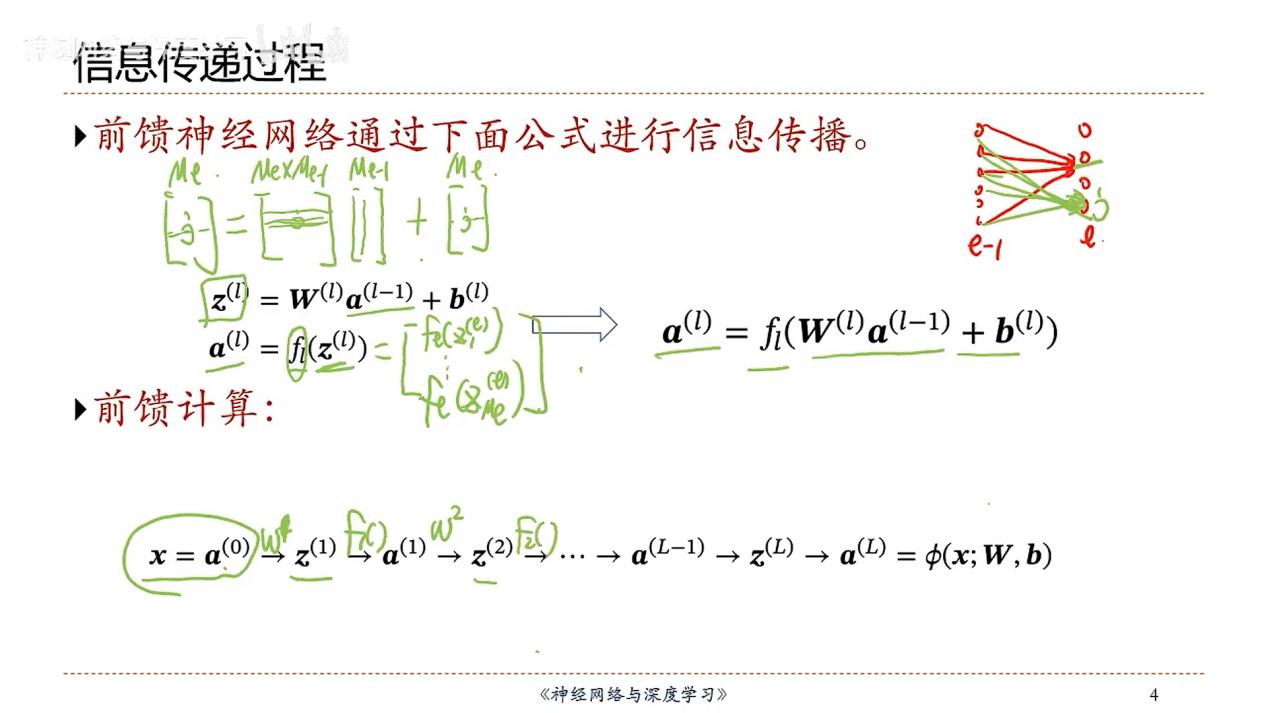
1. 神经网络



1. 前馈神经网络







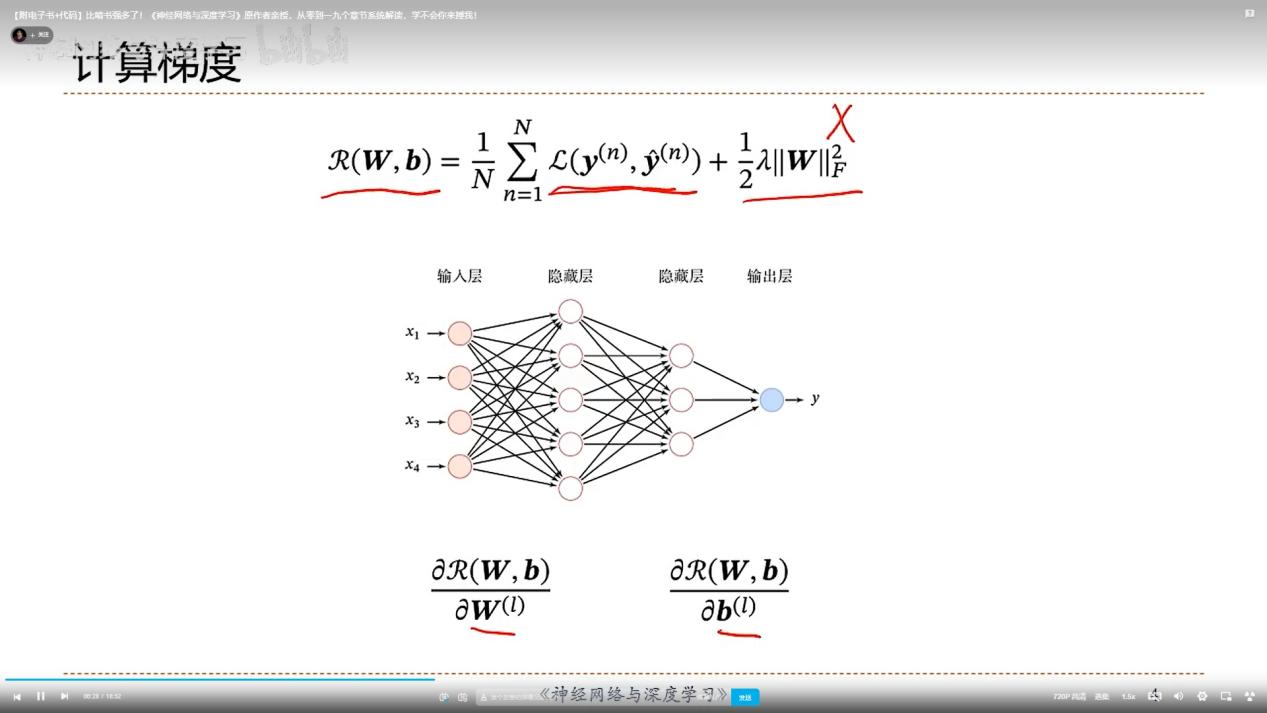
每一次处理就是前置的神经元权重矩阵的行乘以神经元的活性值

计算梯度

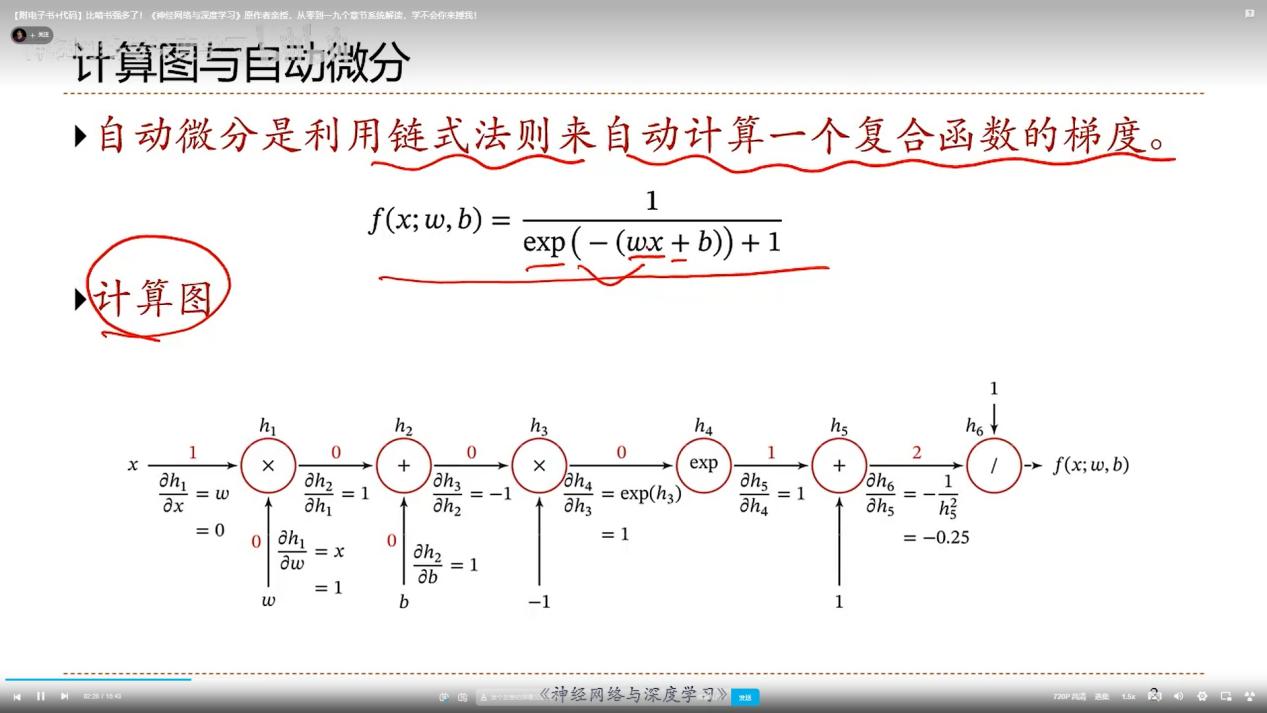
反向传播算法！！！——计算损失函数关于参数函数的梯度

链式法则

[【深度学习】基于计算图的反向传播详解-CSDN博客](https://blog.csdn.net/Daycym/article/details/83756693?ops_request_misc=%7B%22request%5Fid%22%3A%22403508dac01d2602b59d3b868a5eba8c%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=403508dac01d2602b59d3b868a5eba8c&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-83756693-null-null.142%5ev100%5epc_search_result_base4&utm_term=%E8%8B%B9%E6%9E%9C%E5%92%8C%E6%A9%98%E5%AD%90%E5%8F%8D%E5%90%91%E4%BC%A0%E6%92%AD)



计算图与自动微分



静态计算图和动态计算图

静态使用Theano和TensorFlow

动态使用DyNet，Chainer和PyTorch

！！！定义网络->损失函数->优化算法

1. 优化问题
   1. 非凸优化问题
   2. 梯度消失：梯度有小于1的导致连乘后为0，梯度消失不动效率下降，最好在1左右，所以推荐使用ReLU函数

[详解机器学习中的梯度消失、爆炸原因及其解决方法\_梯度消失的原因及解决方法-CSDN博客](https://blog.csdn.net/qq_25737169/article/details/78847691?ops_request_misc=%7B%22request%5Fid%22%3A%220900871b997e3d2428dd56a4b6963008%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=0900871b997e3d2428dd56a4b6963008&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-78847691-null-null.142%5ev100%5epc_search_result_base4&utm_term=%E6%A2%AF%E5%BA%A6%E6%B6%88%E5%A4%B1&spm=1018.2226.3001.4187)

**卷积神经网络（CNN）**

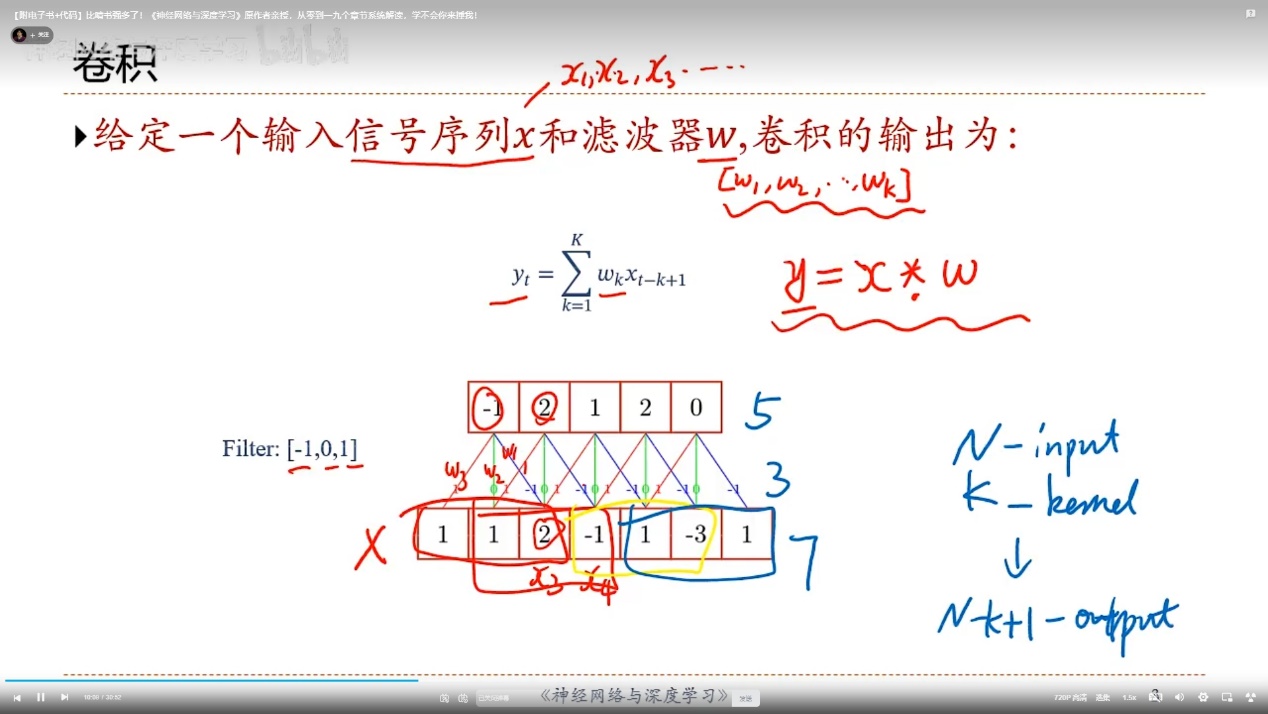
1. 一种前馈神经网络

2. 卷积

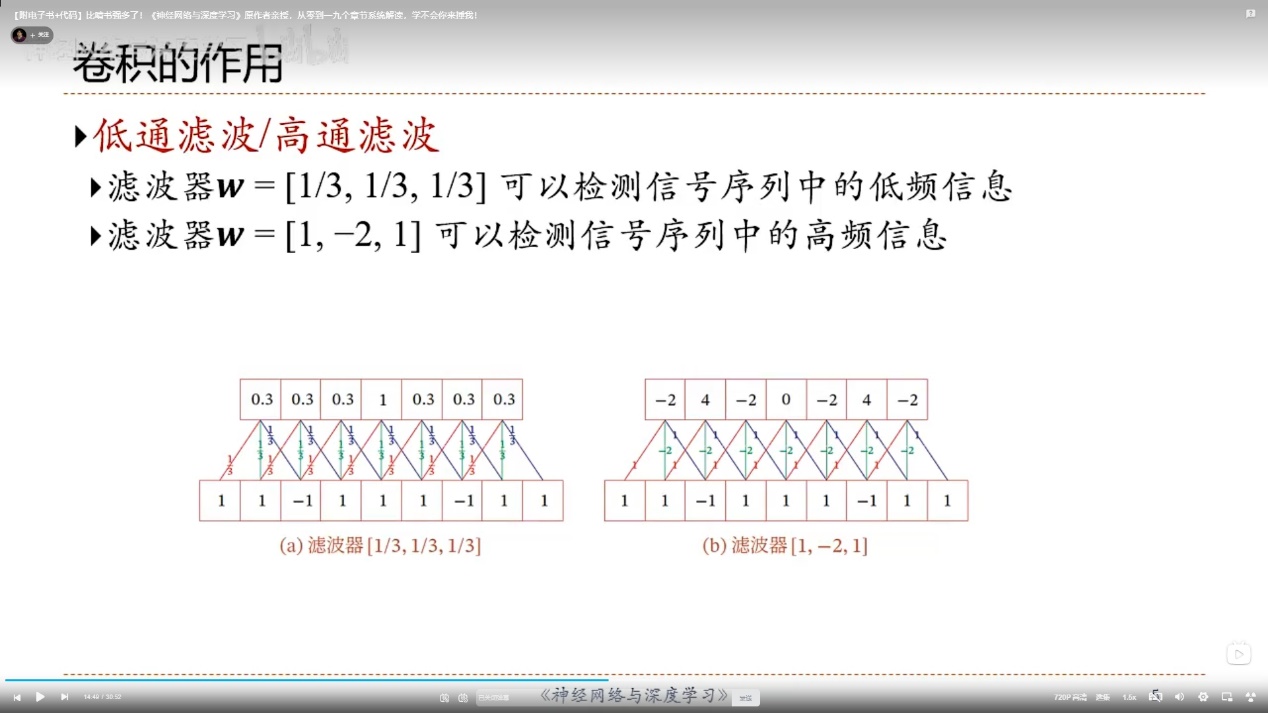
用于计算信号的延迟累积

信号会随着时间步长衰减，计算当前时刻的信息和之前时刻的延迟信息的累加

给定一个信号序列x和滤波器w得到卷积

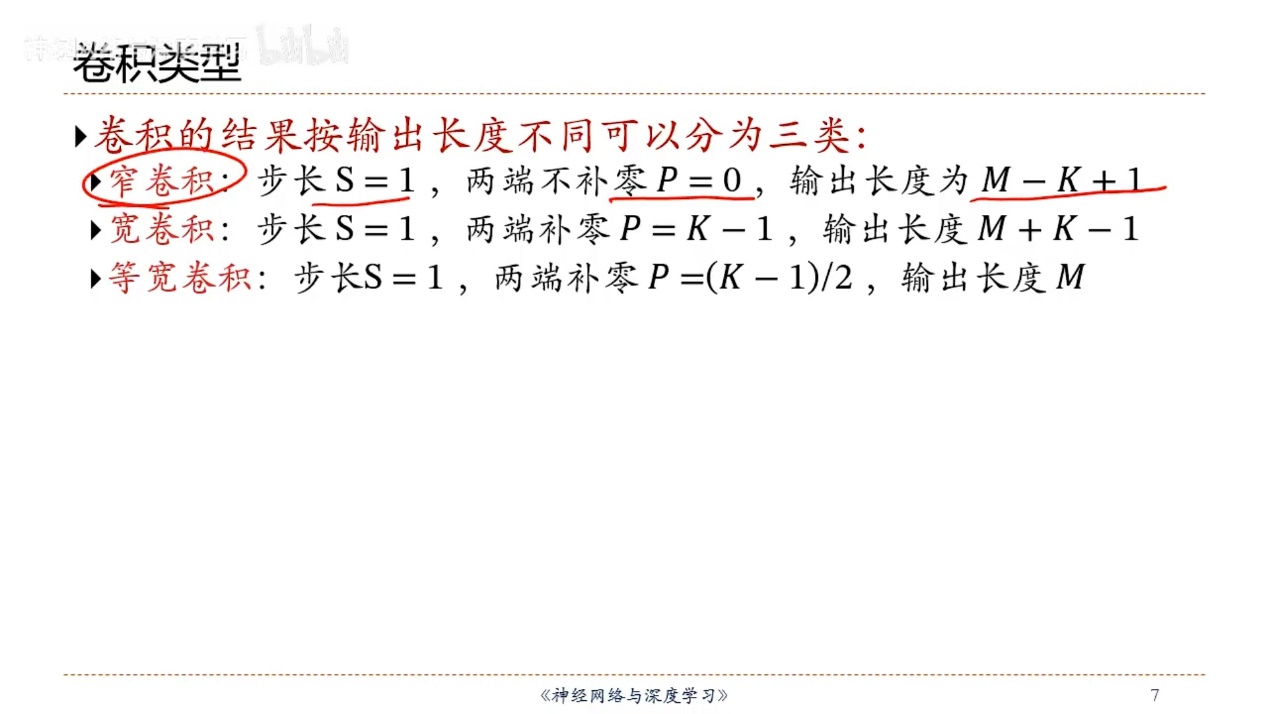


低频信息一般指信号中的均值（变化慢） ，高频一般指局部显著变化的信息



滤波器滑动步长：减少输出序列长度

零填充 ：向输入信号两段补俩零

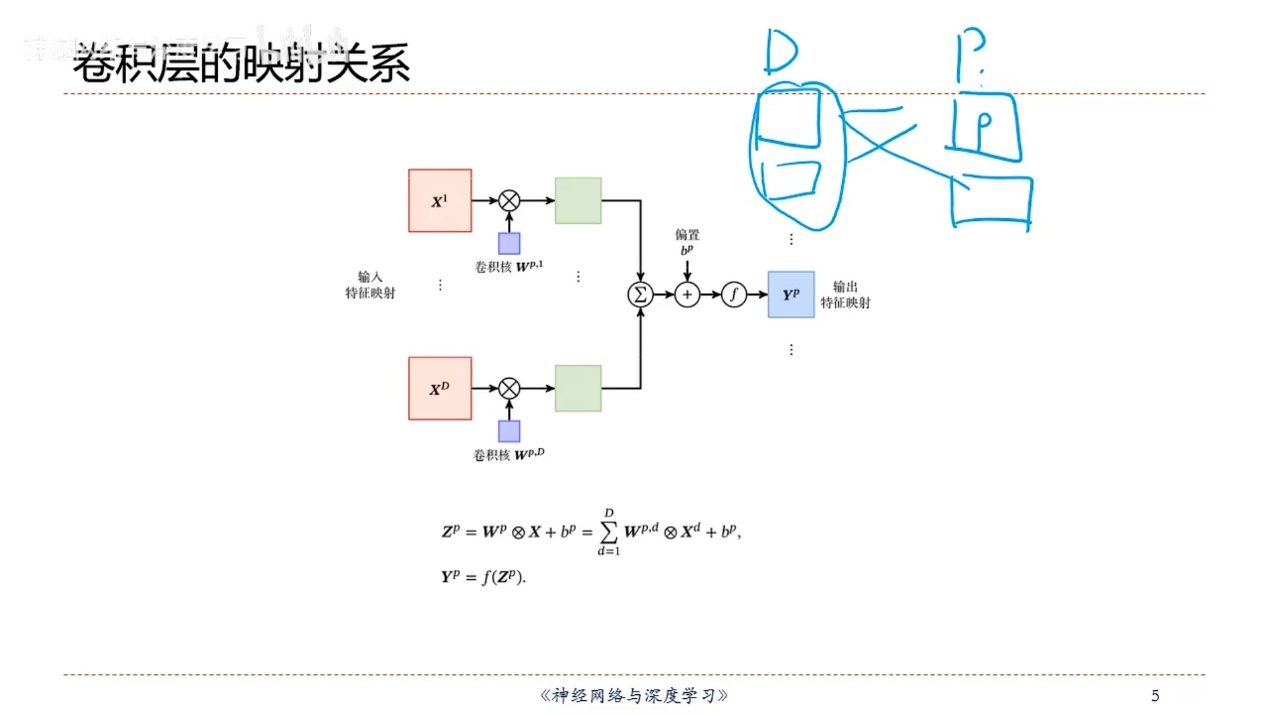


3. 卷积神经网络

用卷积替代全连接层（例如把一个隐含层h改为h和滤波器w的卷积），减少连接数量

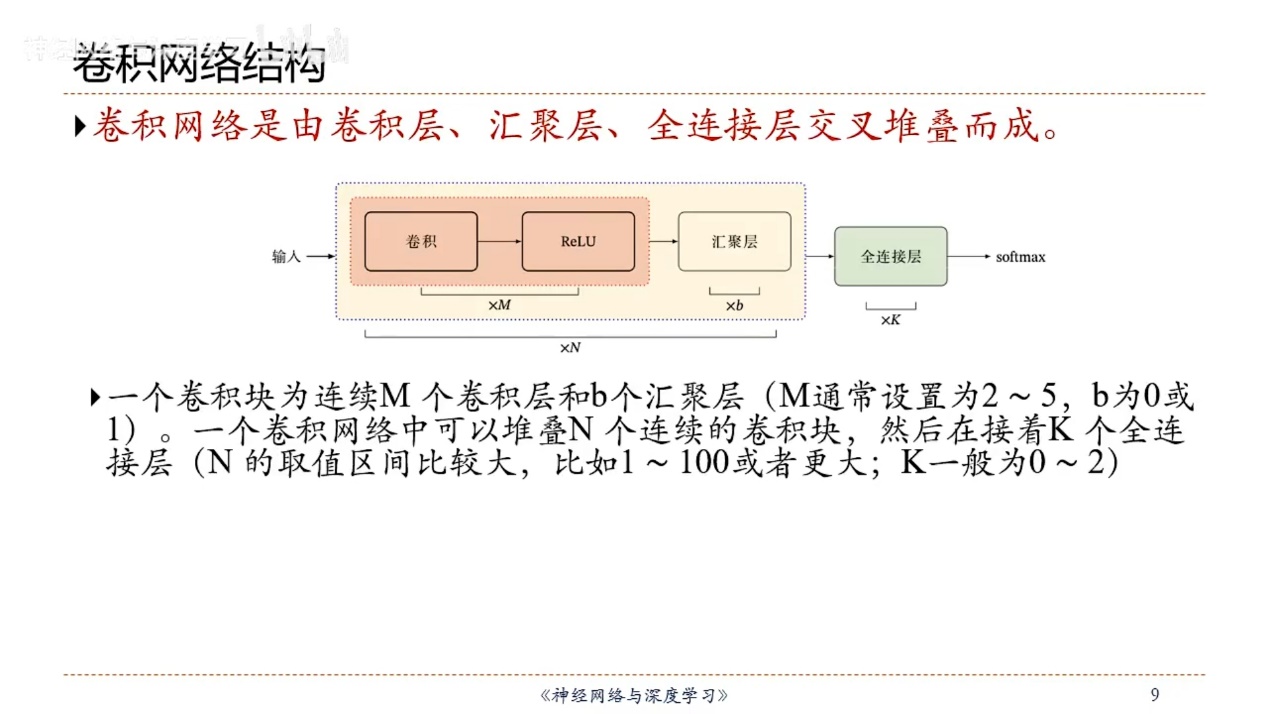
互相关：计算卷积需要将卷积核翻转w顺序和t相反，但是互相关不需要翻转卷积核

每个卷积核看做一个特征提取器，采用多个卷积核增强卷积层的能力



卷积层是三维结构

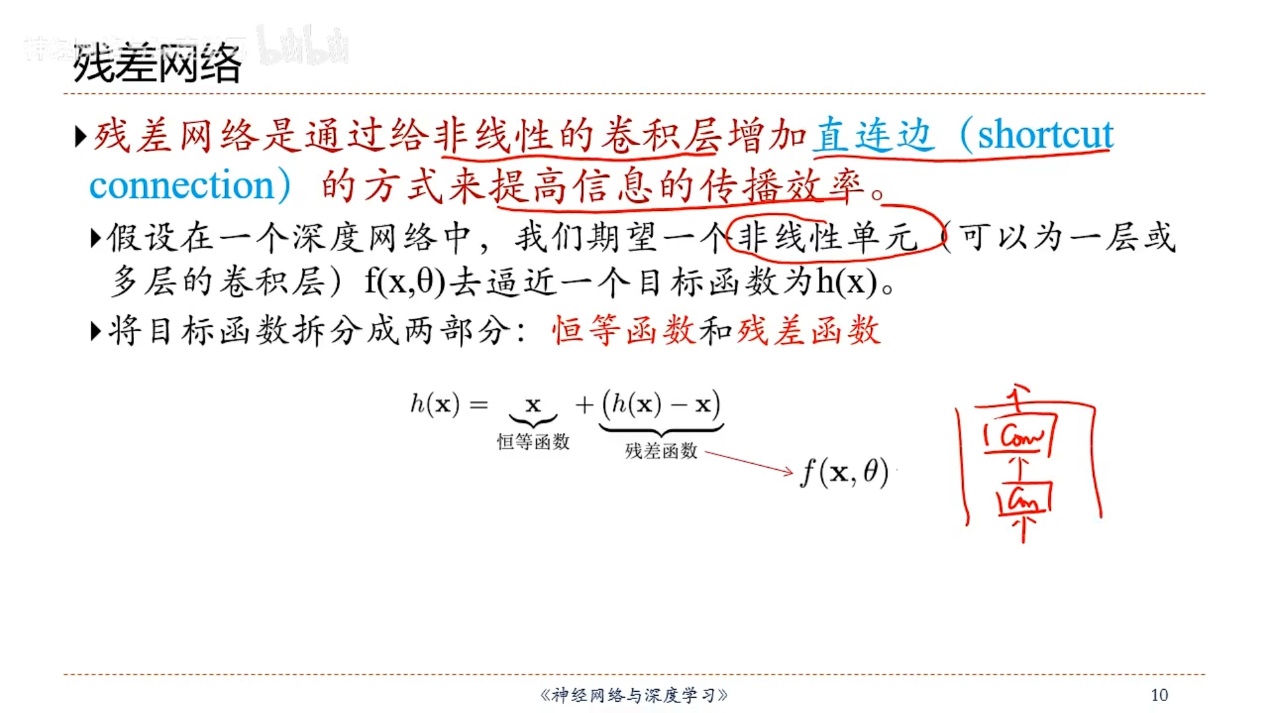
汇聚层：每个特征映射的神经元不会因为卷积层减少，通过引入汇聚层减少



\*\*\* 空洞卷积：通过给卷积核插入“空洞”以增加大小从而增加输出单元的感受野

\*\*\* 转置卷积：在输入插入“0”，扩大输入

AlexNet等卷积网络



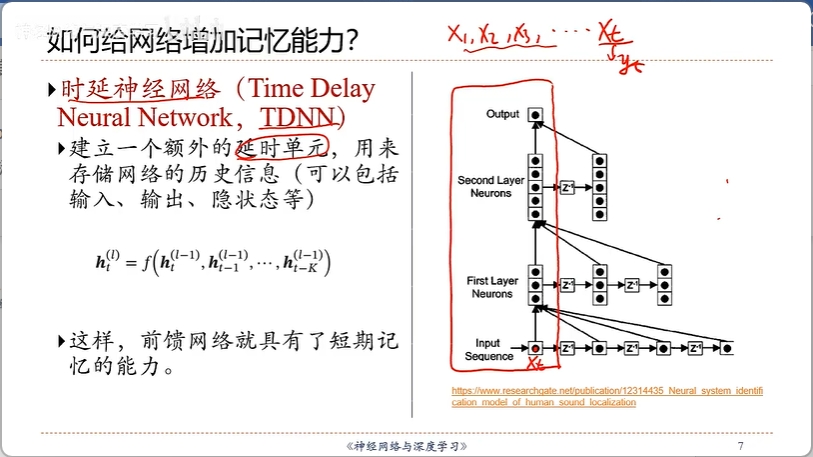
卷积神经网络的应用

1. AlphaGo，基于强化学习的
2. 目标检测
3. OCR，文字识别

**循环神经网络**

1. 如何给网络增加记忆能力：

·利用时延神经网络（TDNN），增加一个延时单元

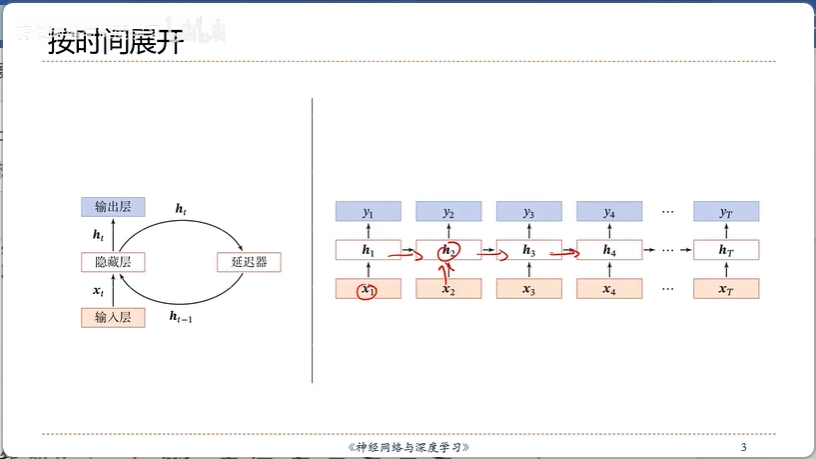


Z就是一个时延单元，每经过一层就更新一次

·自回归模型

1. 循环神经网络（RN）

通过使用带自反馈的神经元，处理任意长度的时序数据



一个完全连接的循环神经网路可以近似解决所有可计算问题（RN），图灵完备

作用：

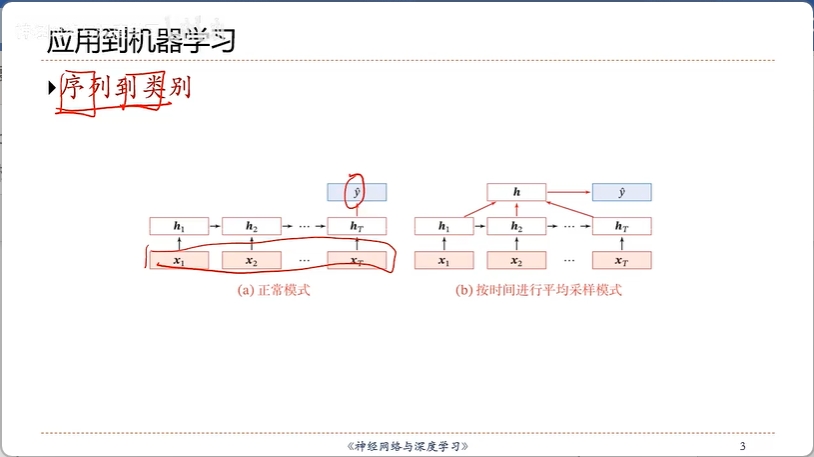
输入-输出映射（机器学习模型）

作为存储器（联想记忆模型）

应用到机器学习：

1. 序列到类别

输出只有一个

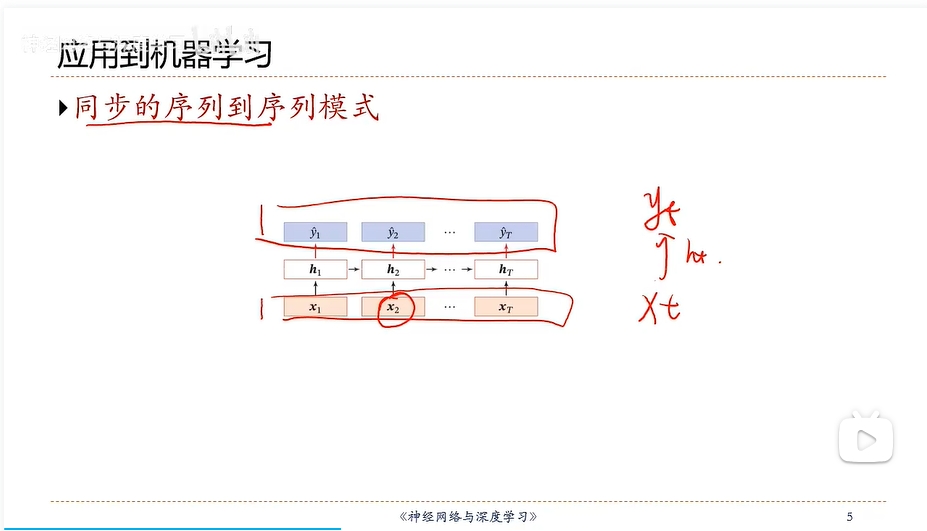


一般常用第二种

常用于情感分类

1. 同步的序列到序列

输出还是一个序列，只和当前时刻的数据相关

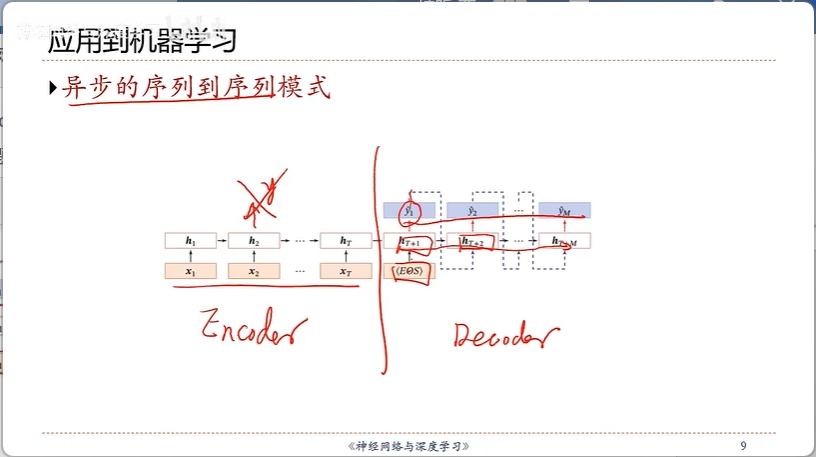


比如中文分词（自然语言处理）

语音识别

1. 异步的序列到序列

一种自回归模型，与上一时刻的数据也有关



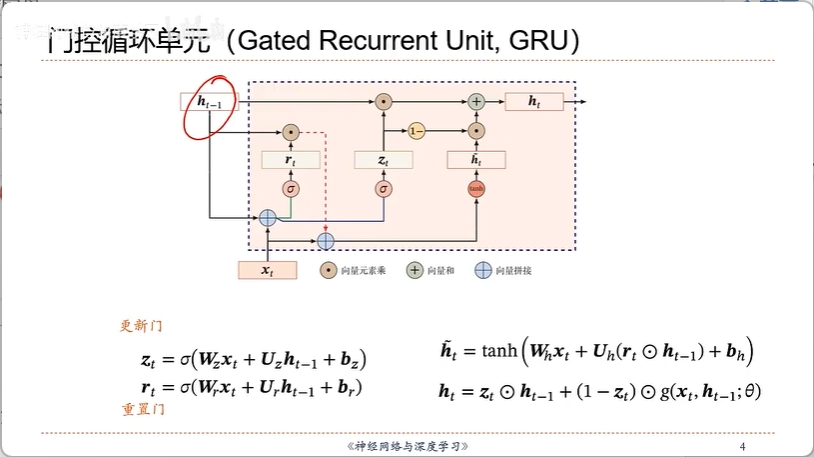
用于机器翻译

！长程依赖问题：由于梯度爆炸或梯度消失问题，实际上只能学习到短周期的依赖关系

由于循环神经网络在时间维度上非常深，如果梯度爆炸可以进行权重衰减或者梯度阶段；如果梯度消失就只能改变模型（把循环边改为线性依赖关系，但是模型能力降低）了

！！！常用的实现

1. 门控循环单元（GRU）



1. 长短期记忆神经网络（LSTM）

根据当前输入Xt和上一次X(t-1)，计算出当前Ct，用上一次C(t-1)乘一个遗忘门算出输出Ct，最后乘一个算出Ht

1. 双向循环神经网络（常用模型）

优点：

1. 引入（短期）记忆
2. 图灵完备

缺点：

1. 长程依赖问题
2. 记忆容量问题
3. 并行能力

应用场景：

1. 语言模型（自然语言理解）：理解一个语句的可能性和合理性

学习一堆语言然后自己生成（代码，语言啥的）

1. 统计机器翻译
2. 对话系统

拓展到图结构

1. 递归神经网络（RNN）：增加了树结构