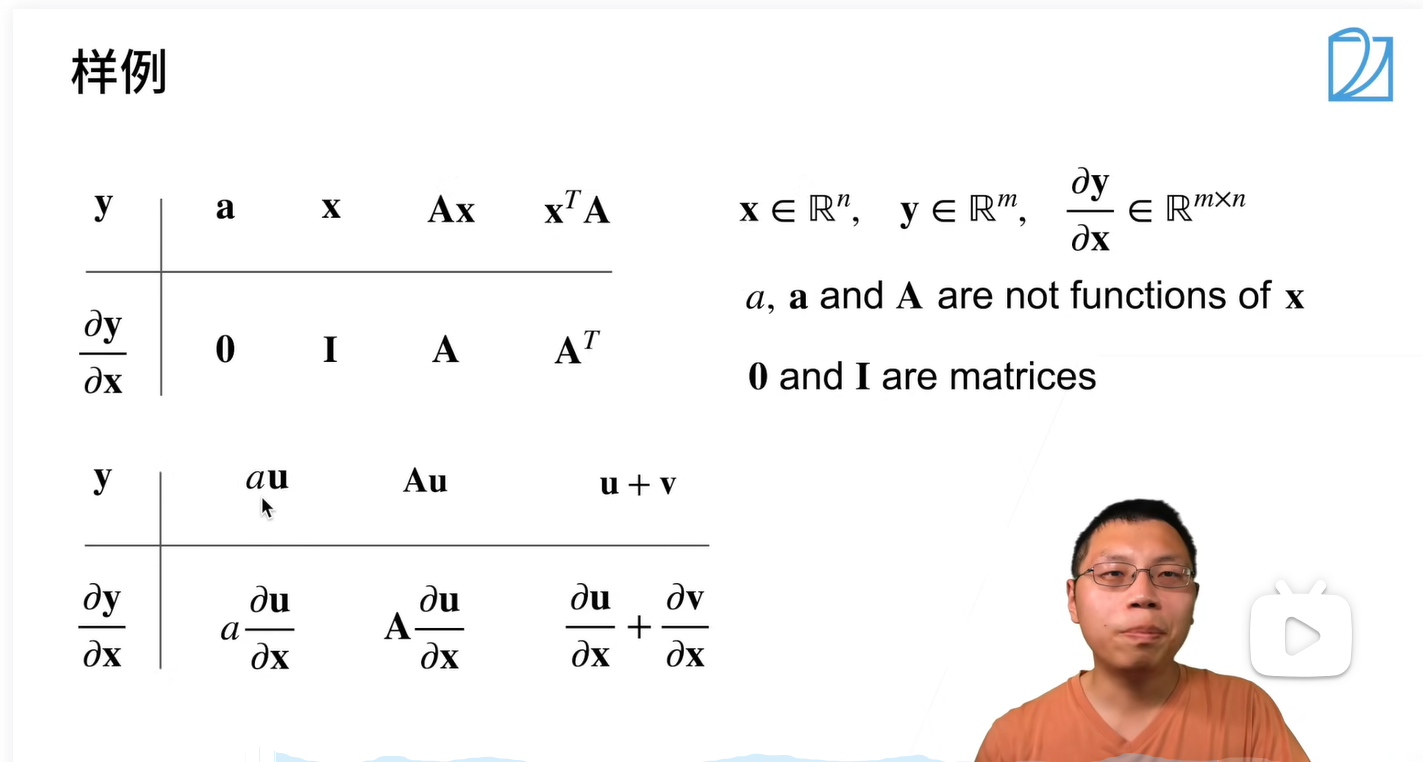
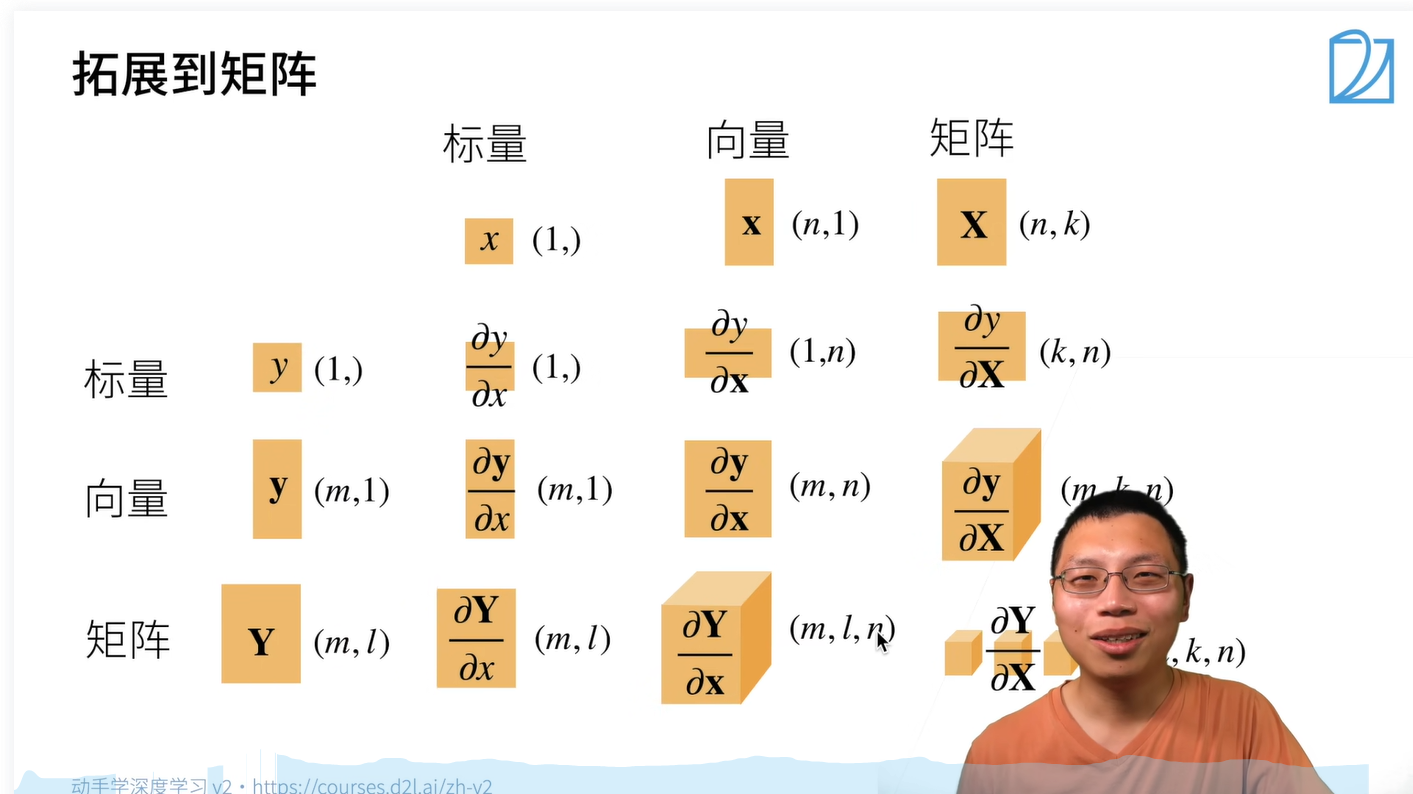
1. **矩阵计算**

向量对向量求导





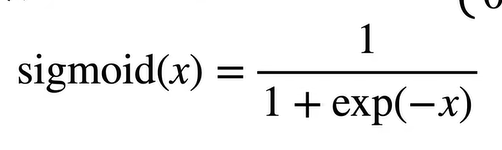
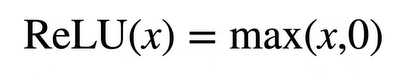
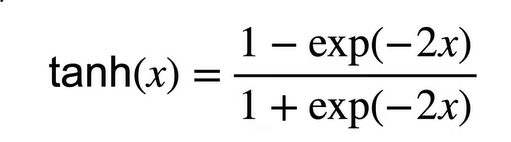
全部情况的总结

1. **损失函数**

了解损失函数，通过函数形状和梯度形状，以分析模型预测值变化的特性（平滑性

1. **激活函数**

用于隐藏层，防止层数塌陷（使多个隐藏层的变换非线性

常见的： 、（最常用）

1. **训练集、验证数据集、测试集**
2. 验证集：用来判断模型的性能，验证后可更改参数、学习率等

测试集：只用一次，用来预测，不能用来调参

不过平时的训练中，测试集被当作成了验证集。

1. 当数据集内容不足时，使用k则交叉验证方法（常用k=5或10）
2. **控制模型复杂度的方法**：
3. 权重衰退。

使用L2正则化损失函数，引入超参数 ，越大复杂度越小



1. Dropout

每一层加入噪音，一些输出项随机置0（超参数概率p）



1. **数值稳定性**

神经网络的输出为o







则输出对第l层的权重求偏导：



文本

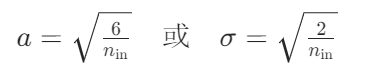
AI 生成的内容可能不正确。



表现为中间L-l个矩阵相乘，因此若数值过大或过小，会导致梯度爆炸或梯度消失，这会导致：

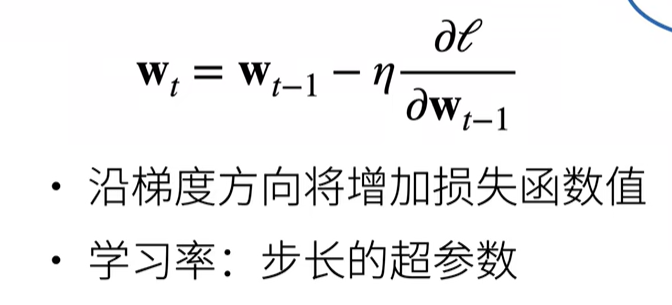
1. 梯度爆炸：梯度值超出值域；对学习率敏感
2. 梯度消失：梯度太小，训练没有进展；层数越深，影响越大，仅仅顶部层训练的好。

解决： Xavier初始化参数方法

1. **激活函数匹配**：
   * Xavier 初始化更适合 sigmoid、tanh 等饱和激活函数。
   * 对于 ReLU 及其变种（如 LeakyReLU），建议使用 **He 初始化**（Kaiming 初始化），其公式为：**
2. **避免偏置项影响**：  
   通常将偏置项初始化为 0，因为其对梯度方差的影响较小。
3. **批量归一化（BatchNorm）的补充**：  
   结合 BatchNorm 可以进一步稳定训练，但初始化仍需合理设置
4. **卷积层**
5. **问题**

1. 模型的优化方法

2．确定学习率大小的方法（学习率不能太小消耗计算资源，也不能太大引起震荡）



3.常见的神经网络模型：线性回归、softmax回归（分类模型）

4.另一个超参数：b，批量大小，为了减小计算量抽取一些样本计算梯度

选取b的方法

