权重衰退

1. 过拟合的概念

过拟合是指模型在训练数据上表现良好，但在测试数据或新数据上表现不佳。这通常是因为模型过于复杂，能够捕捉到训练数据中的噪声

2. 权重衰退的原理

**权重衰退通过在损失函数中添加一个惩罚项来限制模型权重的大小。这样可以防止模型过于依赖某些特征，从而减少过拟合的风险**

3. 数学表达

权重衰退的损失函数可以表示为：



* L：原始损失函数（如均方误差、交叉熵等）
* λ：权重衰退系数，控制惩罚的强度。值越大，惩罚越强
* wi：模型的权重

4. 实现方式

在训练模型时，优化算法（如梯度下降）会同时考虑原始损失和惩罚项，从而更新权重。例如，更新规则可以写成：



其中 η是学习率

5. 超参数调节

* 选择合适的 λ：通常需要通过交叉验证来选择合适的权重衰退系数。过小可能无法有效防止过拟合，过大则可能导致欠拟合

6. 应用场景

权重衰退广泛应用于各种机器学习模型，尤其是深度学习模型，如神经网络。它在训练过程中有助于提高模型的稳定性和鲁棒性

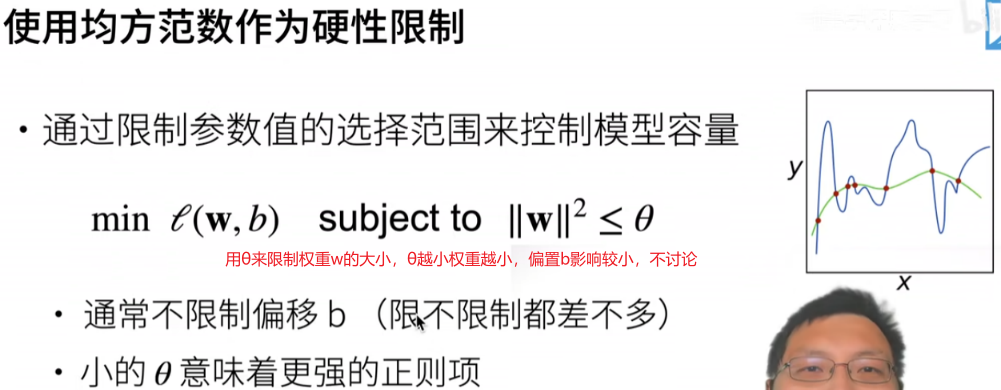
7. 与其他正则化技术的比较

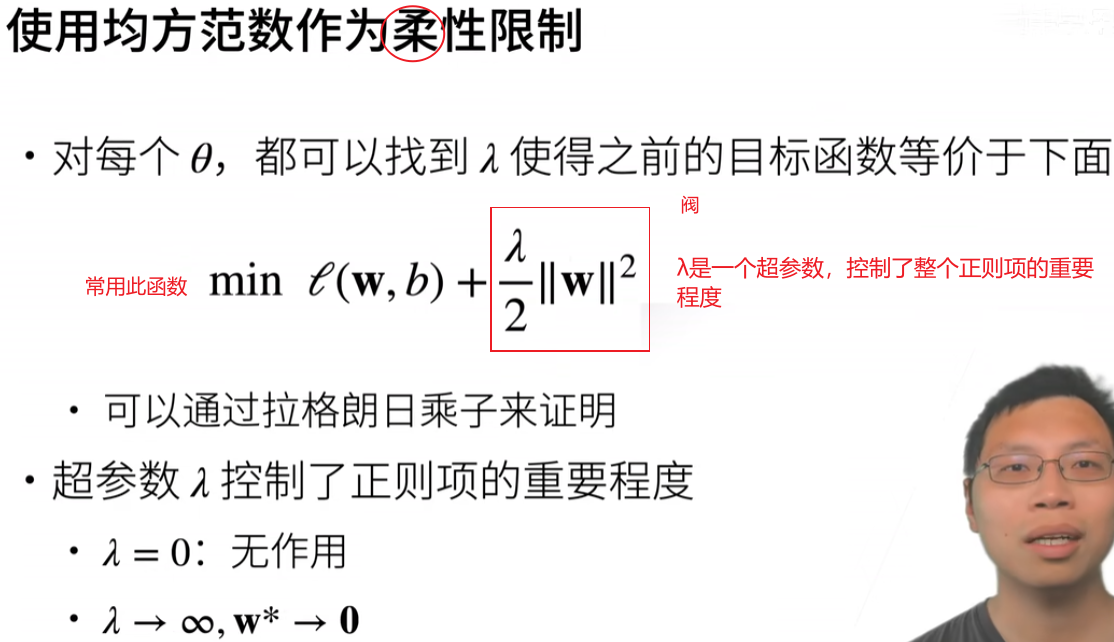
* **L1正则化**（Lasso）：通过绝对值惩罚项促使某些权重变为零，能进行特征选择
* **L2正则化**（权重衰退）：通过平方惩罚项减少权重值，但不会使权重完全为零

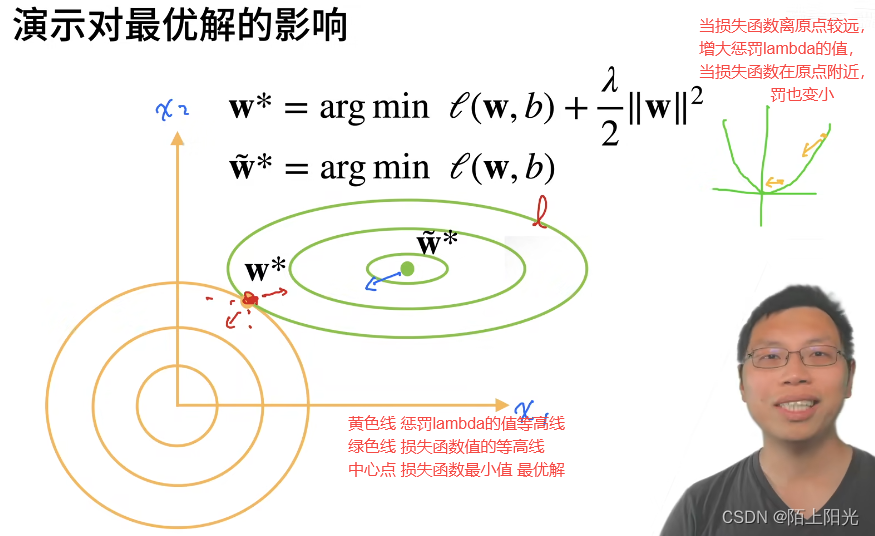
8. 总结

权重衰退是一种简单而有效的正则化技术，能够帮助模型在面对复杂数据时保持良好的泛化能力。适当的使用权重衰退可以显著提高模型的性能

怎么控制一个模型的容量？一个是把模型变得比较小，里面参数比较少，第二个是使得每个参数选择的值的范围比较小，权重衰退就是通过控制整个值的选择范围来进行的



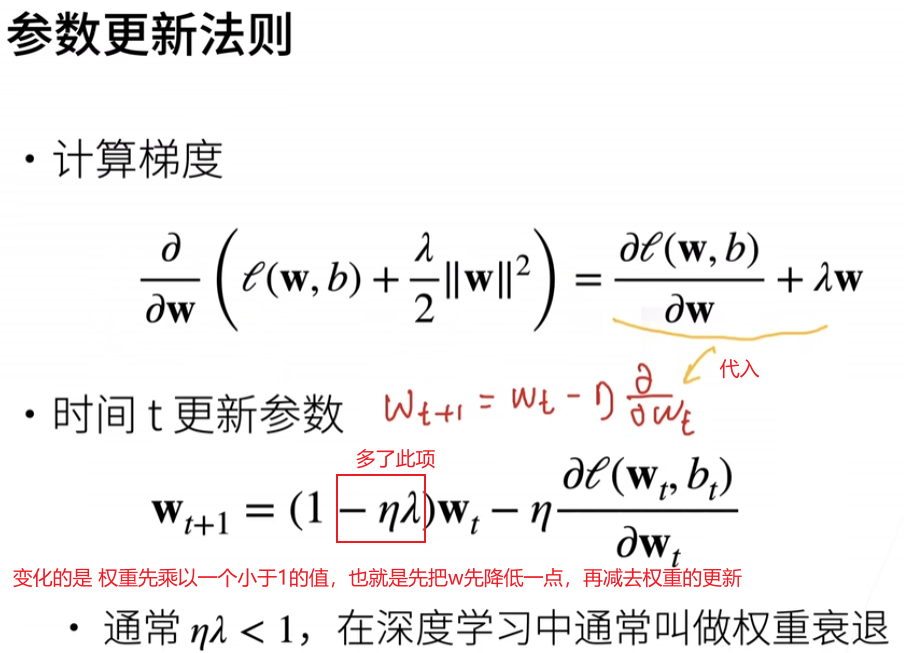




为了便于讨论，我们假设训练的模型中只有w1和w2两个参数，但是我们觉得100这个数还是太大了，怎么办？我们在损失函数中添加一项，这一项是w1的平方+w2的平方。罚的项是L2正则项，因为离原点越近，正则项越小。绿线是损失函数的取值, 黄线是惩罚项的取值, 两者都是圈越大取值越大

两个圆锥的交点可能是最优点！！！！！！！

**其实就是新的损失函数由两项组成，此时求导后，梯度有两项了，一项将w向绿线中心拉，一项将w向原点拉进，最后将在w\*点达到一个平衡**



每次都在权重更新之前对w做了一次放小(ηλ<1)， 所以叫做权重衰退。  
λ是控制模型参数的超参数

Dropout丢弃法

**Dropout** 是一种有效的正则化技术，旨在防止神经网络的过拟合。它通过随机“丢弃”神经元（即在训练过程中将其输出设为零），来减少模型对特定神经元的依赖，从而增强模型的泛化能力

1. 原理

**在每个训练迭代中，以一定的概率（通常在 0.2 到 0.5 之间）随机选择一部分神经元不参与前向传播和反向传播。**这意味着这些神经元在该次训练中不会影响模型的学习过程

2. 数学表达

假设有一个神经元的输出为 h*h*，在应用 Dropout 后，输出变为：

**h′=h⋅r**

其中 **r**是一个随机变量，如果以概率 **p** 保留神经元（**r=1**），以概率 **1−p**丢弃神经元（**r=0**）

3. 训练与推理

* **训练阶段**：在每个训练批次中，随机丢弃一定比例的神经元
* **推理阶段**：不再丢弃神经元，而是将所有神经元的输出缩放，以补偿训练时的丢弃。通常通过将输出乘以保留概率 p*p* 来实现

4. 超参数

* 丢弃率 (Dropout Rate)：决定了在训练过程中丢弃的神经元比例。常见的值为 **0.2 到 0.5**

5. 优点

* **减少过拟合**：通过引入随机性，Dropout 可以显著降低模型的复杂度，减少对特定神经元的依赖。
* **提高鲁棒性**：增强模型对输入噪声和变化的适应能力。

6. 应用场景

Dropout 通常用于全连接层和卷积层中，尤其是在深度学习模型中，如卷积神经网络（CNN）和递归神经网络（RNN）。

7. 注意事项

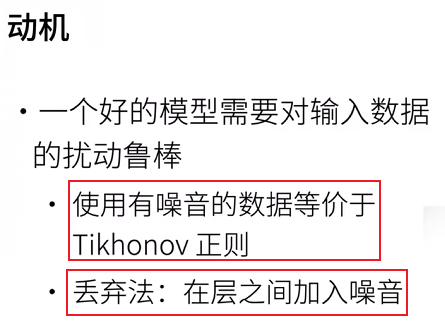
* Dropout 可能会导致训练时间增加，因为每次训练都需要进行随机选择。
* 在某些情况下（如小数据集），Dropout 可能会导致欠拟合，因此需要根据具体情况调整。

总结

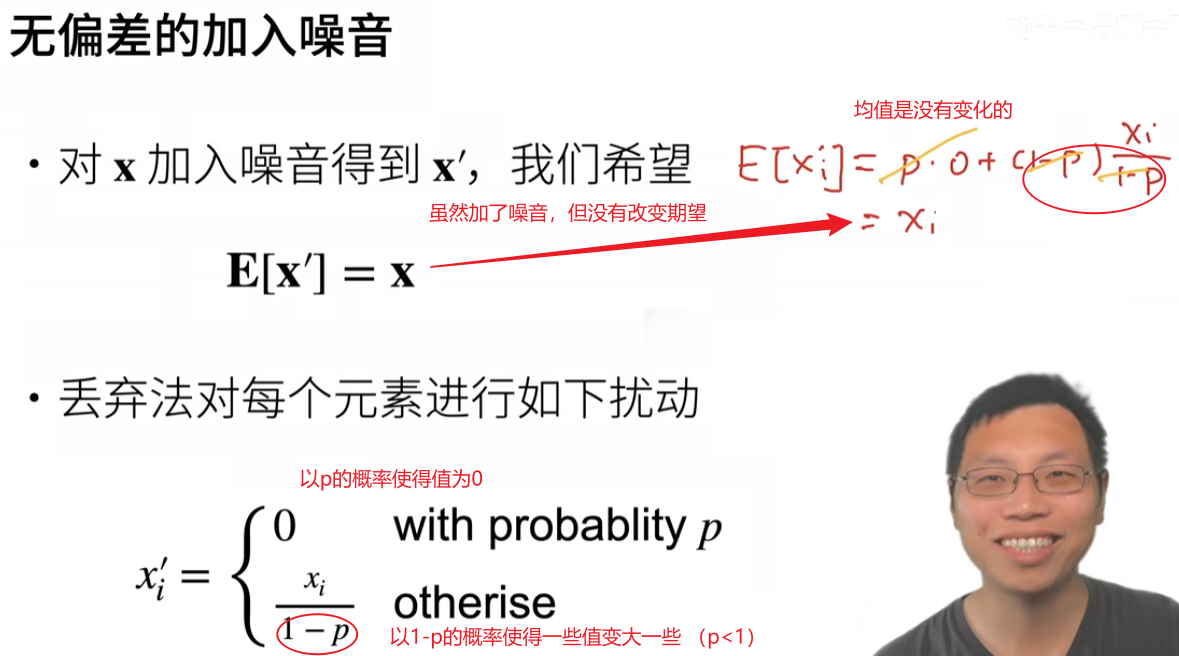
Dropout 是一种简单而有效的正则化方法，通过随机丢弃神经元来增强模型的泛化能力，广泛应用于各种深度学习任务中。

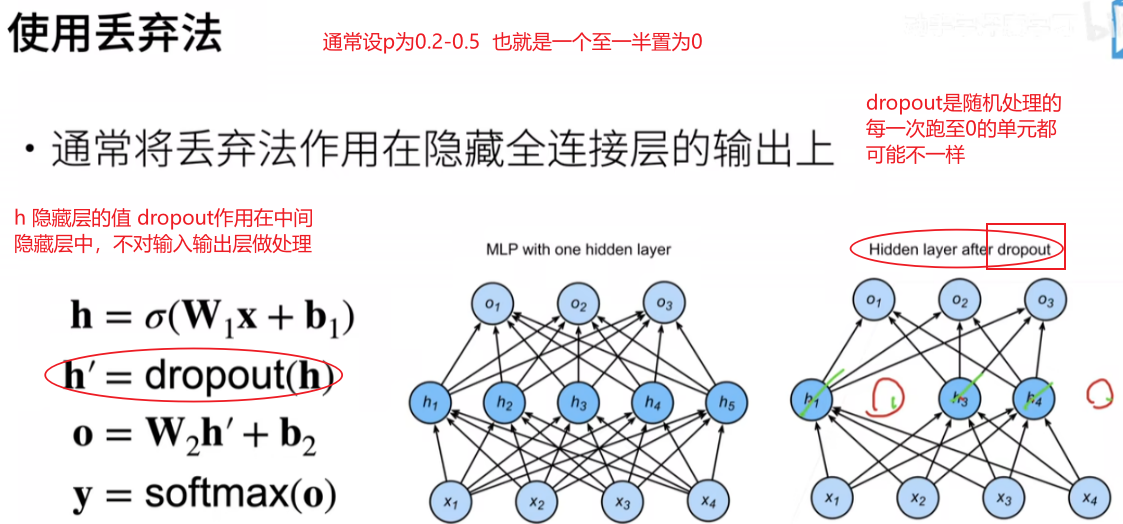
**扰动鲁棒性**是指模型在面对输入数据的微小变化或噪声时，仍能保持其性能和输出稳定的能力

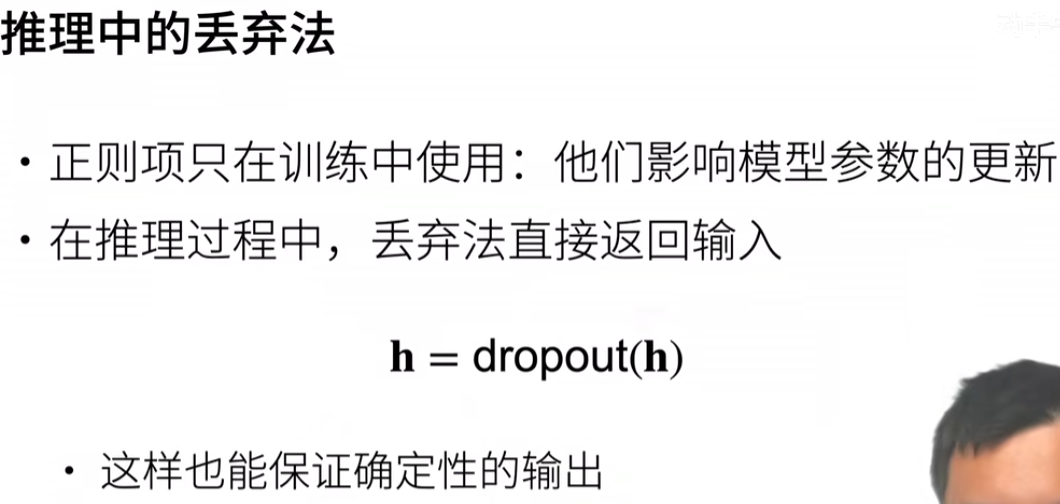
**防止过拟合**：鲁棒性有助于模型在训练集与测试集之间保持一致性，避免对特定输入的过度依赖



在层之间加入噪音，不对输入层做处理。不是在输入数据上加噪音







以上是训练过程使用的。把dropout看作是一个处理中介结果的正则项，**正则项只在训练中使用，是对权重做的影响**。在推理(inference 预测过程 权重不需要发生变化)的时候是不对h做处理的，dropout输出的是数据本身

