优化模型

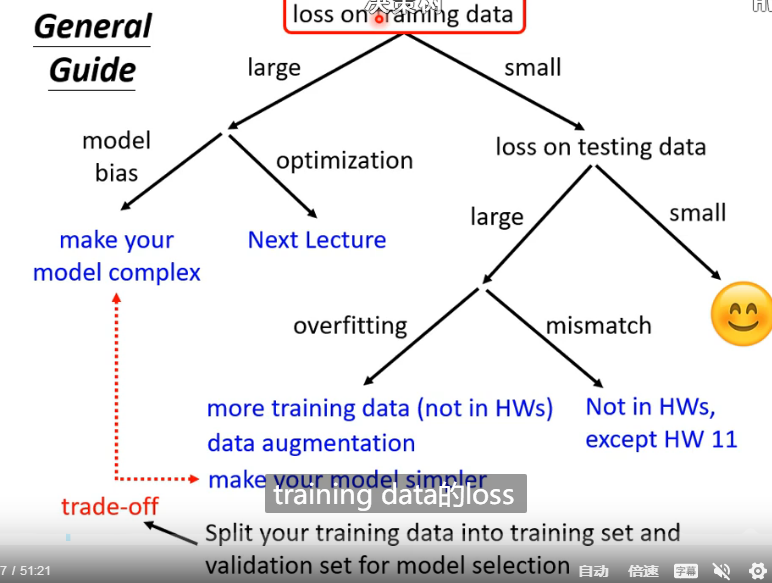
当训练的损失很大的时候，有可能是模型太简单，就需要重新设计模型，提高模型的弹性，通过增加输入的维度。

还有一个可能是模型的优化有问题，就是在训练的过程中梯度下降到了一个局部最优解，梯度为0，使得梯度消失，导致训练的损失不够低。

通过比较不同的模型来得知模型到底够不够大，通过检查训练数据上的结果，当更高层的模型的损失比低层的模型损失大，那么就说明高层的模型的优化没有做好。

当训练的损失很小的时候，在测试集上的损失也很小的时候那就没有什么问题了

但如果在测试集上的损失很大，那就有可能是过拟合，实际上就是模型过于复杂，过度拟合了训练集上的数据，使得在测试集上的表现变差。可以通过减少模型的参数、共享参数、droupout、权重衰退、early features、正则化来缓解过拟合。



数据集的划分

将数据集分为训练集和测试集

训练集又被分为训练集和验证集

测试集又被分为公共测试集和私有测试集

训练时通过验证集来选择模型，那么在公共数据集上的分数就能够来反映私有测试集上的分数

N折交叉验证

将数据集进行切分切成N份，将其中一份作为验证集，然后重复N次，每一次的验证集都不一样。

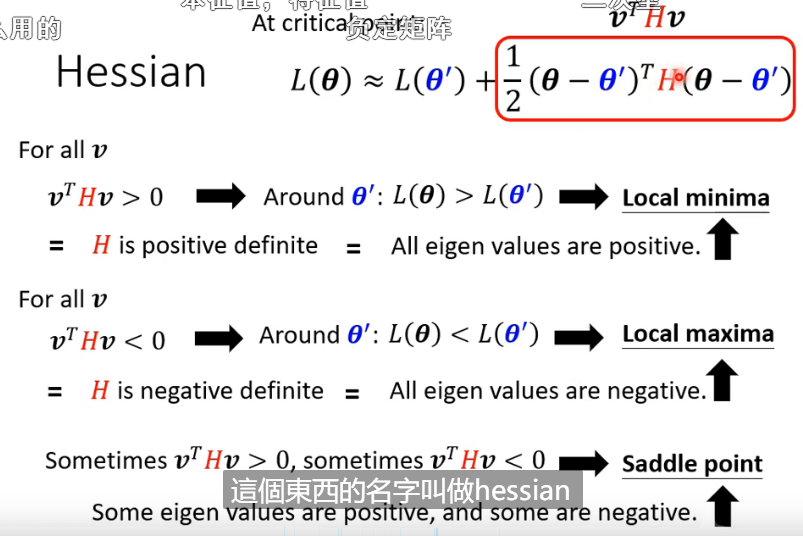
类神经网络训练不起来

当训练的损失无法下降，那么就有可能卡在局部最优的点或者是鞍点。

如果是卡在局部最优解的点，那么就没有什么办法了。

但如果是卡在鞍点的话，还是有路可以走，只要逃离鞍点那么就可以往损失更低的地方走。

通过二阶泰勒展开式子的第二阶的那一项来判断这个点是鞍点还是极值点。



Batch

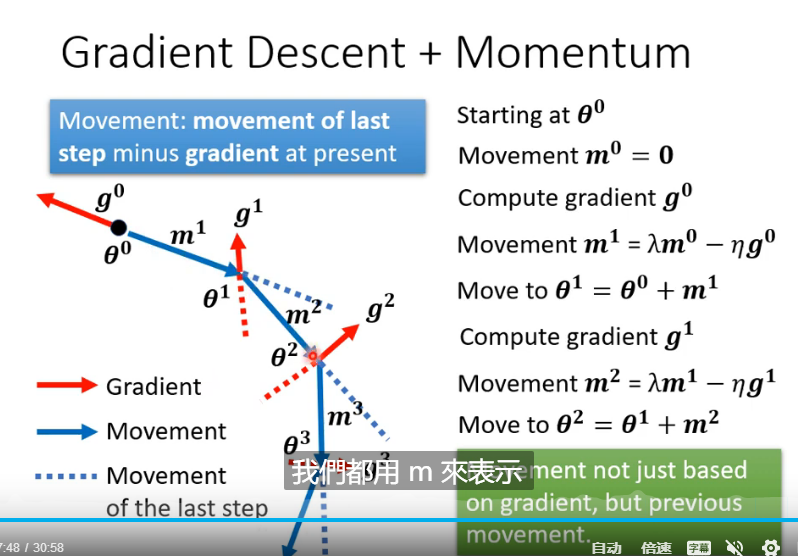
当Batch\_size设为N(Full batch)就相当于没有设置Batch\_size，那么在训练的过程中模型就要读完全部的数据才能够计算损失然后再计算梯度然后更新参数。

当Batch\_size设为1那么就只要看一次数据就可以计算损失和梯度然后更新参数。

Batch\_size过大那么训练精度就会变低，是由于优化过程出现了问题，Batch\_size小了优化反而变好。

Momentum

给梯度下降施加一个惯性，使得模型在更新的过程中就能够跳出鞍点和局部最优点。



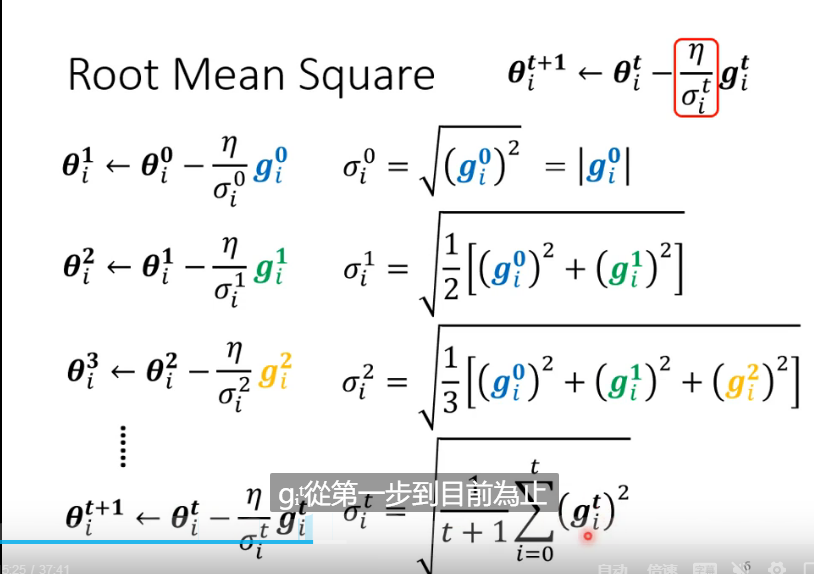
自动调整学习率

在训练时梯度下降的时候损失始终无法下降那么就有可能是到了局部最优点或者鞍点，但是还有一种可能就是发生了振荡无法到达最低点。

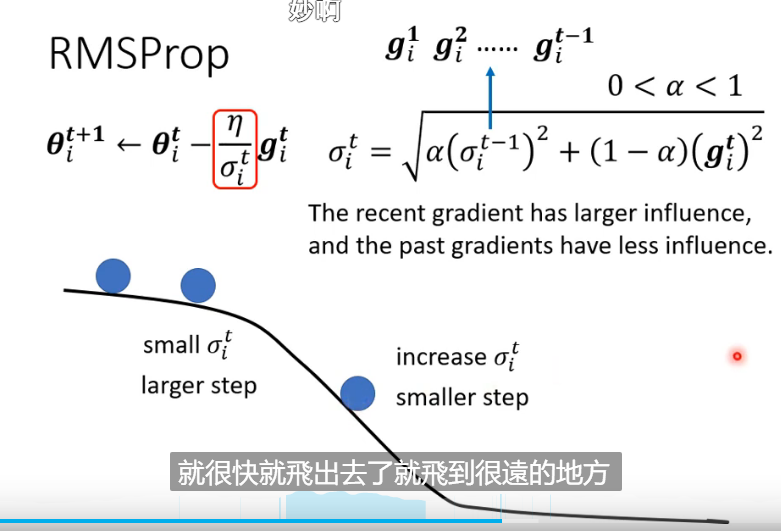
学习率设太大那么就有可能会造成反复横跳无法收敛。

学习率太小那么收敛速度很慢，有可能无法到达最优点。

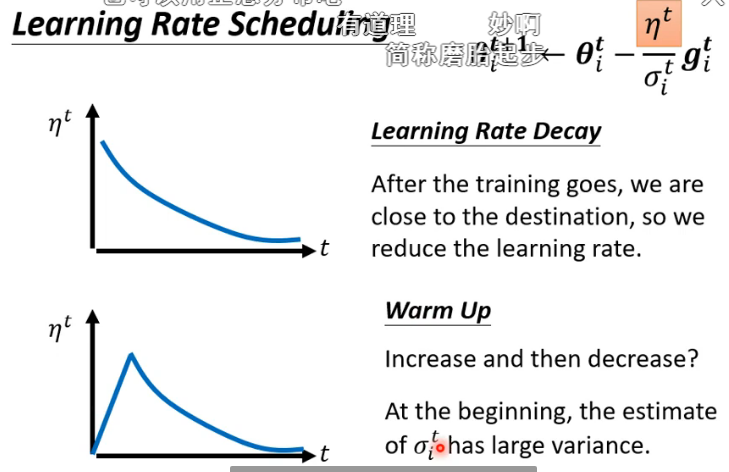
Root Mean Square方法



RMSProp



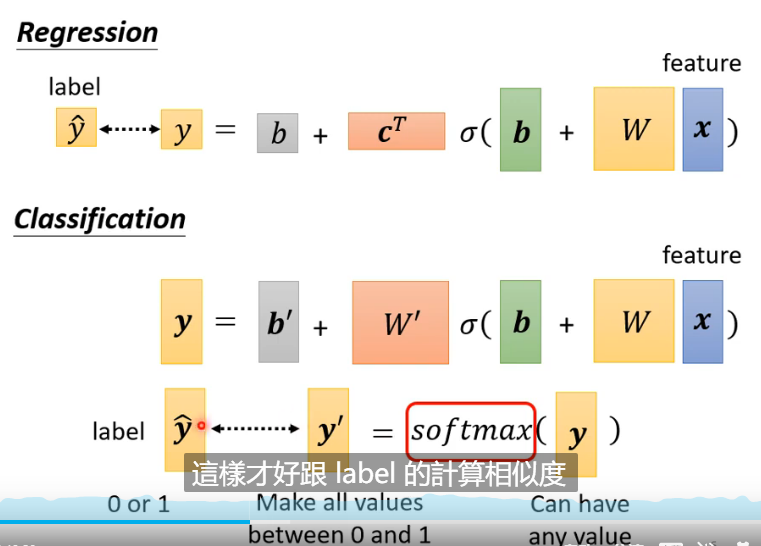
warm up是一种学习率优化方法。在模型训练之初选用比较小的学习率，在训练了一段时间后使用预设的学习率进行训练。



one-hot编码

在计算损失的时候使用one-hot编码就可以避免类别之间的距离影响了损失的计算，使用one-hot编码就可以让每个类别两两之间的距离都是一样的。

在分类任务中通过复杂网络后都需要再通过一个线性层和一个softmax层将输出结果进行归一化，那么输出的结果就是某一个类别的概率，只要从中选出概率最大的对应的类别就是最终的输出结果。



分类任务的损失函数最好使用Cross-entropy，使用交叉熵在分类任务中会更加的容易梯度下降收敛，使用MSE的话就会使得收敛变的更难。

