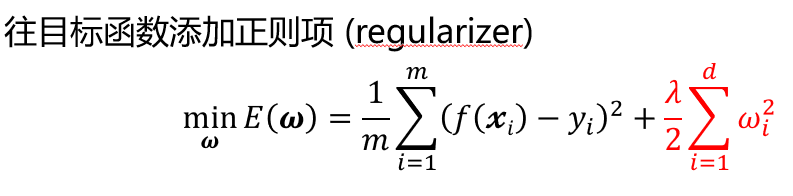
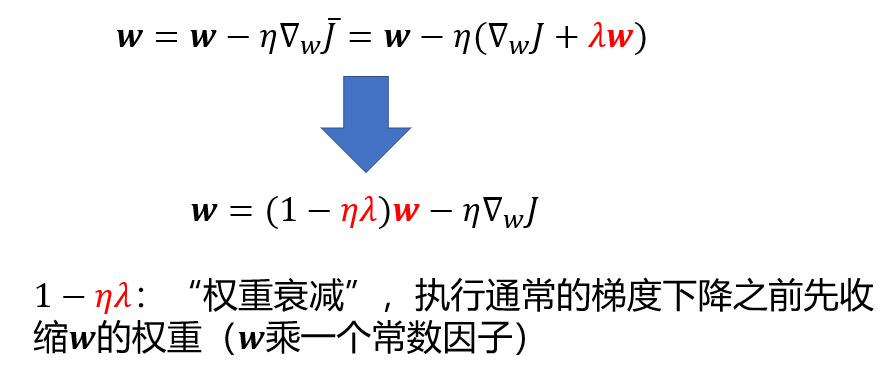
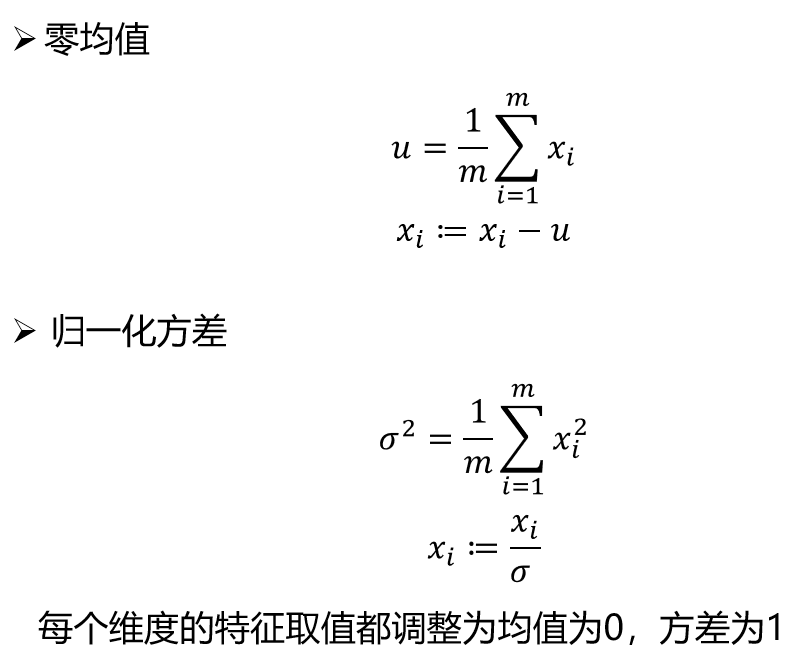
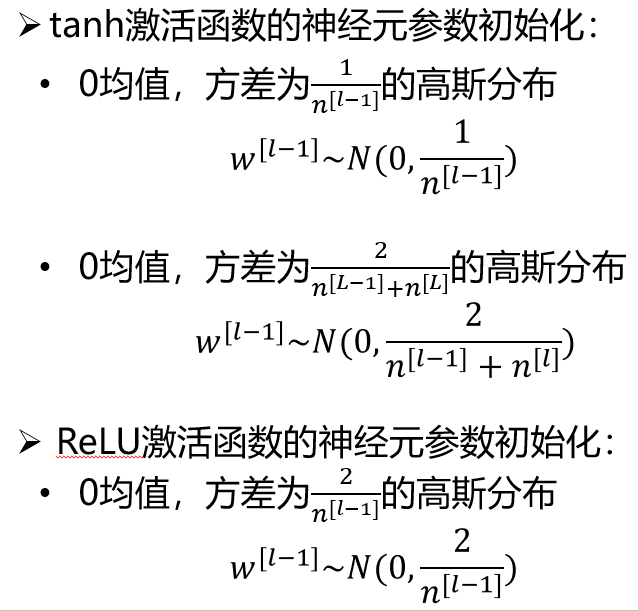
**正则化**：通过限制神经网络复杂度，来减少过拟合

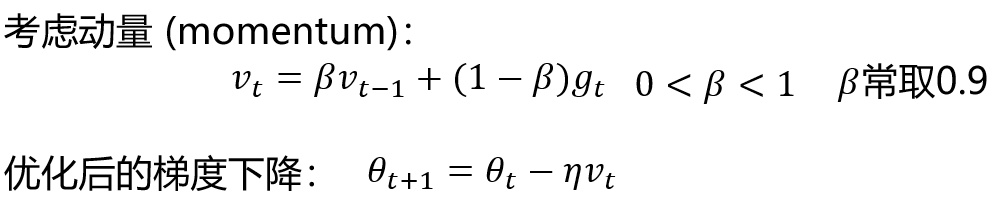
1. 𝐿1和𝐿2正则化
2. Dropout：在每一次迭代网络训练时，以一定的概率丢弃一些神经元
3. 数据增强：对数据进行旋转、翻转、缩放、平移、加噪等操作增加数据量

**输入归一化**：若输入样本不同维度取值范围差别巨大会影响梯度下降法的效率，所以需要将样本每个维度都调整为相同尺度或者取值范围

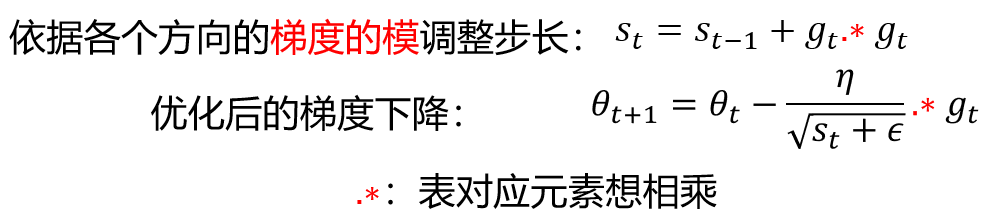
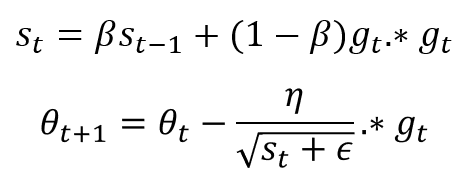
**参数初始化**：随着网络层数的增加，梯度以指数方式变小或者增大。为了需要保持神经元的输入和输出方差大体一致（why？），提高网络训练效率，参数w需要合理初始化。

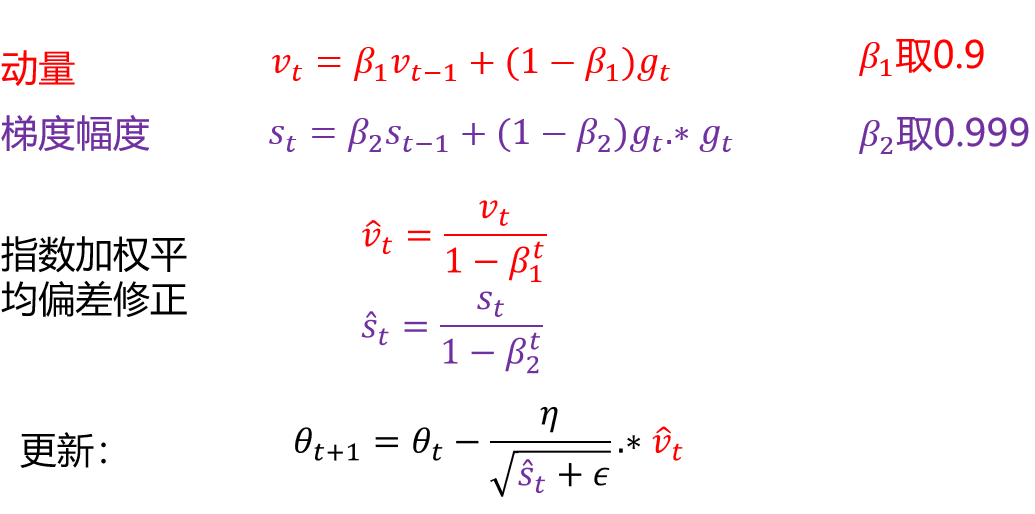
**梯度下降优化算法**：

数据量：小批量、随机批量

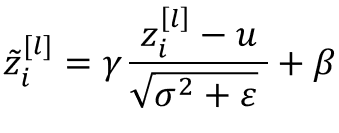
梯度：引入梯度动量

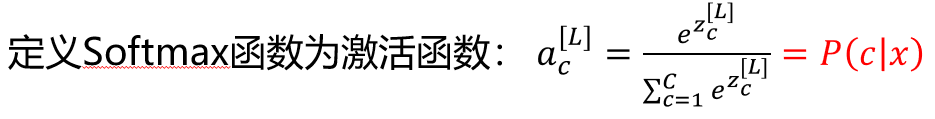
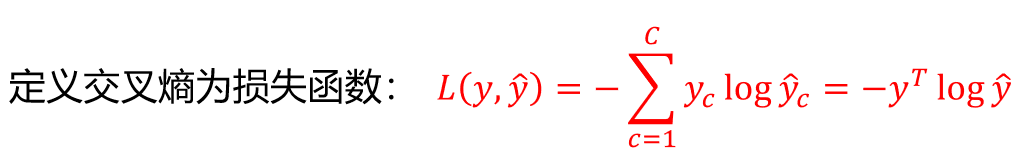
学习率：

* 学习率随训练次数t进行衰减
* AdaGra 算法
* RMSprop算法：指数加权平均优化后的AdaGrad

集大成者：Adam算法：梯度动量+RMSprop

**批量归一化**：数据在传入激活函数之前进行归一化处理，以提高训练效率。

Batch算法：（u为均值 σ为方差 𝛾和𝛽可通过梯度下降法学习）

**Softmax 分类**：Softmax层：多个神经元输出对应类别的概率。