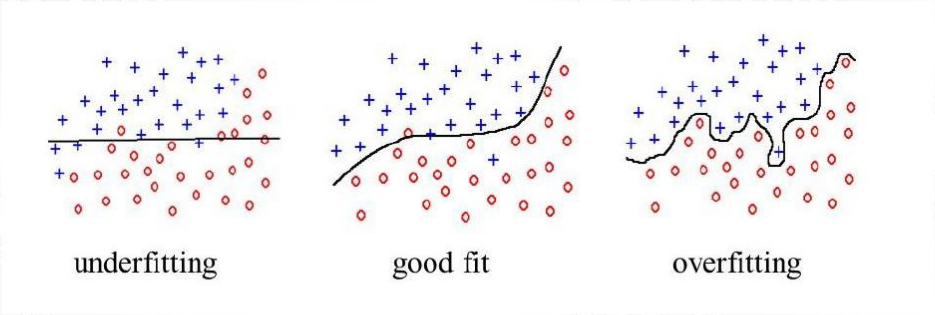
泛化与正则化

目录

什么是泛化与正则化

什么是泛化

- 所谓泛化(Generalization)，模型不仅在训练集表现良好，在未知的数据(测试集)也表现良好，即具有良好的泛化能力



过拟合(overfitting)与欠拟合(underfitting)

泛化不好的后果

- 模型性能不稳定，容易受到攻击

更改一个像素值攻击分类模型

增加贴纸攻击YOLO目标检测模型

增加贴纸攻击人脸识别模型

什么是正则化

- 所谓正则化(Regularization)，目标就是要同时让经验风险和模型复杂度都较小，是对模型的一种规则约束

$$(\ (\ ) \ \ )$$

f即预测结果函数，V即损失函数。R(f)是一个跟模型复杂度相关的单调递增函数，用于约束模型的表达能力。

正则化方法分类

正则化方法分类

◆ 显式正则化(经验正则化，参数正则化)

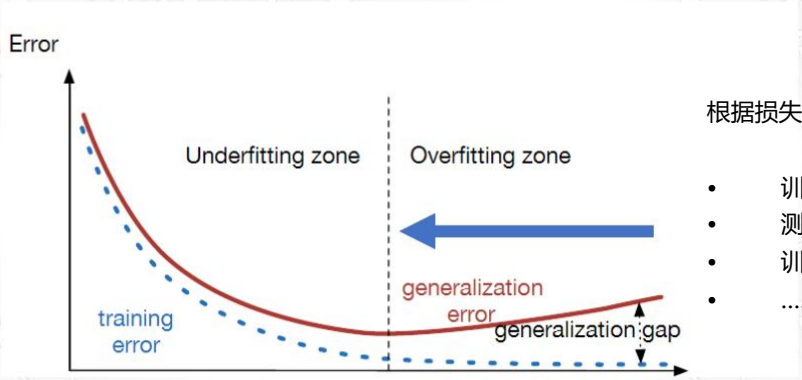
- 网络结构，损失函数的修改，模型使用方法的调整

◆ 隐式正则化

- 没有直接对模型进行正则化约束，但间接获取更好的泛化能力

显式方法-提前终止

◆ 提前终止模型的训练

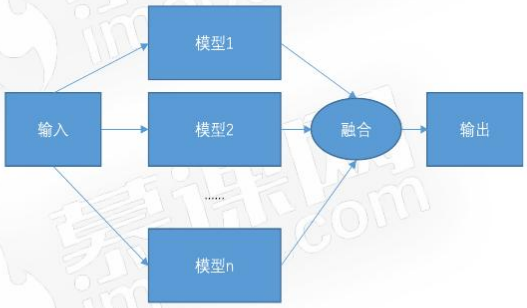


根据损失函数的变化，提前终止模型的训练：

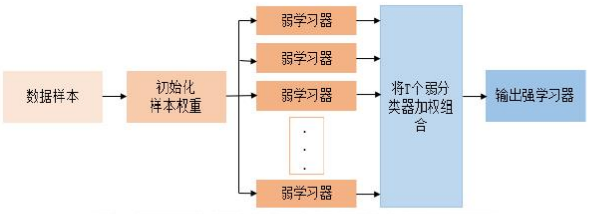
- 训练集误差不再明显下降
- 测试集误差开始持续增大
- 训练集与测试集差距变大
- ……

显式方法-模型集成

◆ 模型集成(Ensemble)：多次训练不同的模型进行结果融合



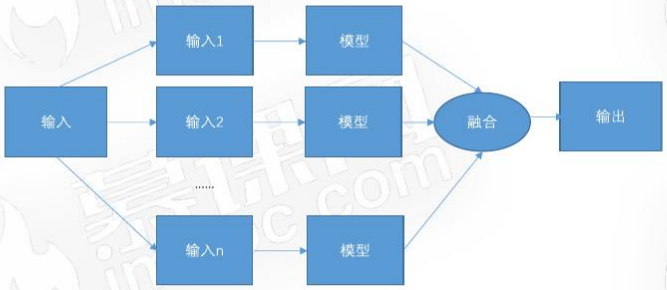
训练多个不同的模型进行结果融合

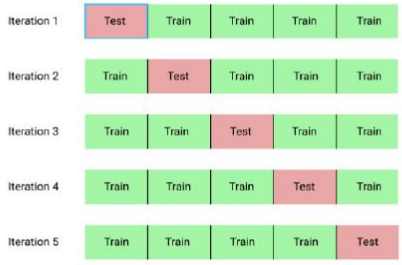


Adaboost

显式方法-模型集成

◆ 模型集成(Ensemble)：多次使用不同的数据训练模型进行结果融合

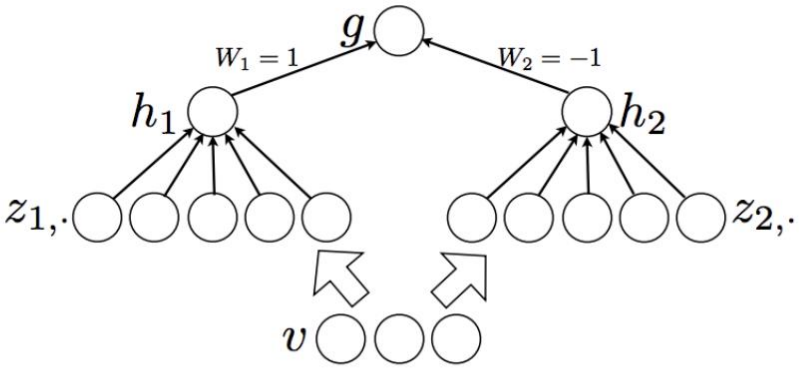




K折验证

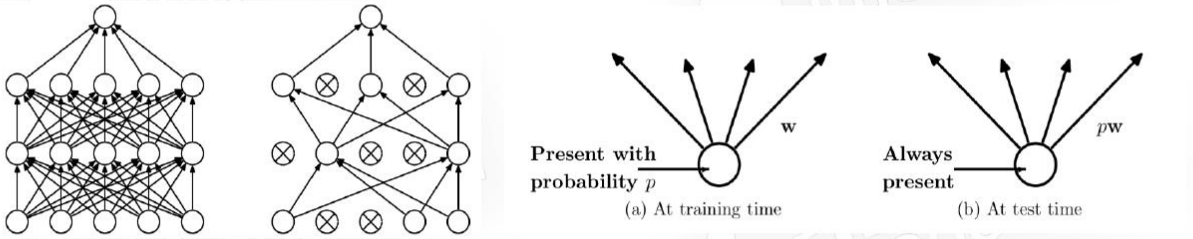
显式方法-Maxout

◆ Maxout，取N个激活的最大值



显式方法-Dropout

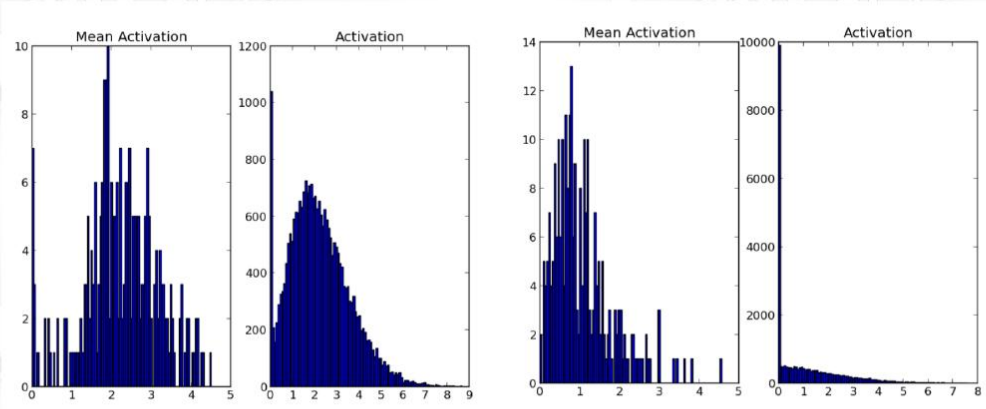
- 训练时按照概率p随机的丢弃一部分节点，测试时不丢弃，输出结果乘以p



有n个节点的神经网络，可以看做是2^n个模型的集成，获得神经元的稀疏性

显式方法-Dropout

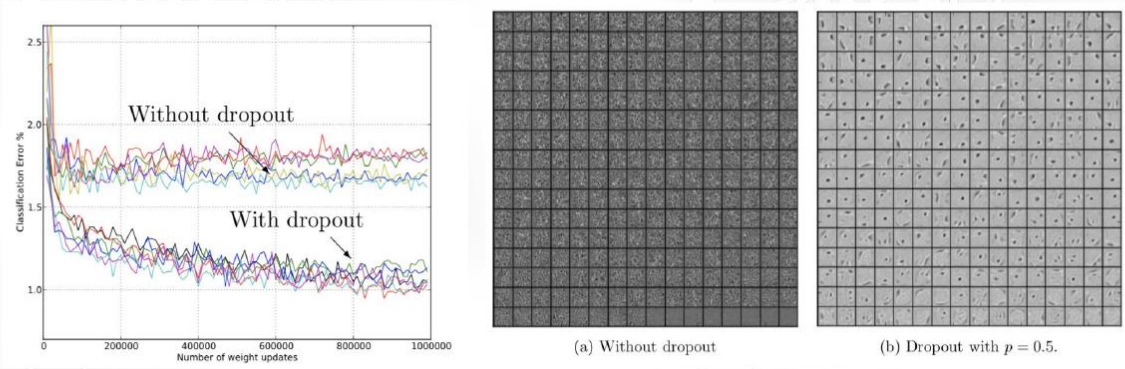
- Dropout带来更稀疏的激活模式，更多接近于0的激活值



(a) Without dropout　　(b) Dropout with $p = 0.5$.

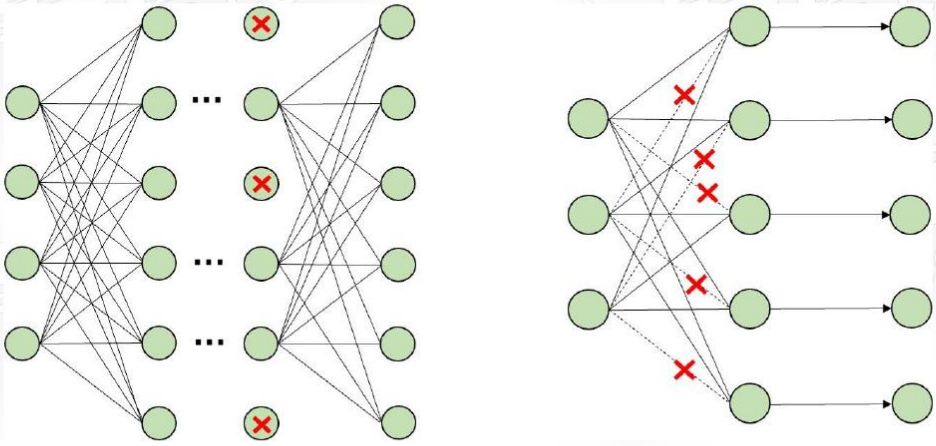
显式方法-Dropout

- Dropout带来更低的泛化误差与更好的特征提取器



显式方法-Dropconnect

- Dropconnect，随机去掉连接

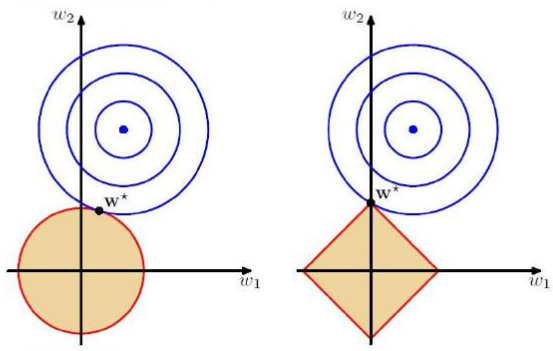


Dropout和Dropconnect对比

显式方法-参数正则化

◆ L1正则化：（　　）（　　）‖ ‖

◆ L2正则化：（　　）（　　）-‖ ‖

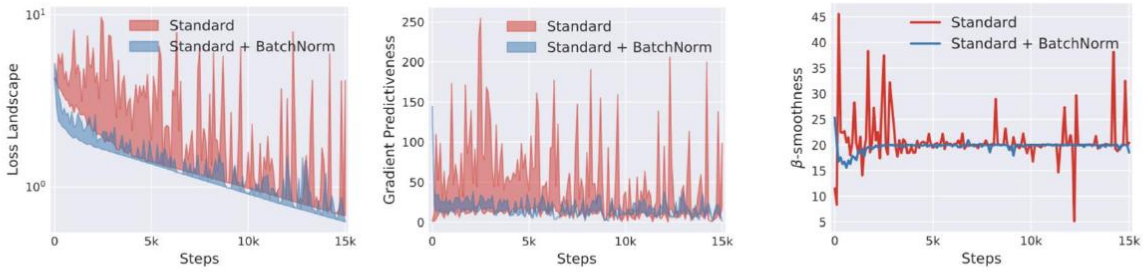


参数空间(w1,w2)是一个二维平面，蓝色部分是一个平方损失函数，黄色部分是正则项。

L2正则化的最优交点使得w1或者w2特别小。
L1正则化的最优交点使得w1或者w2等于0，获得所谓的稀疏化。

隐式正则化方法-数据标准化

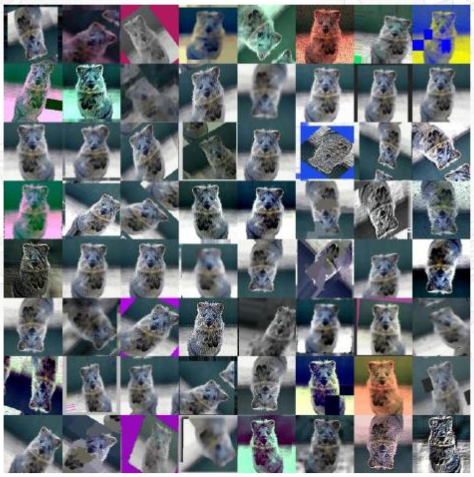
◆ 数据标准化平滑了优化目标函数曲面



(a) loss landscape　(b) gradient predictiveness　(c) "effective" β-smoothness

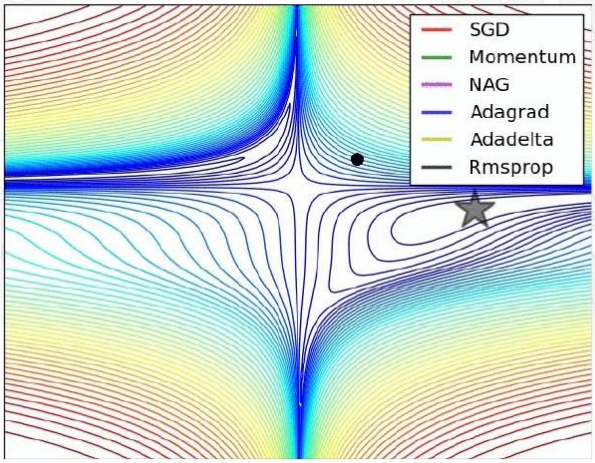
隐式正则化方法-数据增强

- 扩大数据集规模



隐式正则化方法-随机参数更新

- 随机梯度下降算法，每次选取不同的样本，在不同的优化过程会获得不同结果



隐式正则化方法-标签平滑

- 标签平滑(label smoothing)

通过soft one-hot加入噪声，减少了真实样本标签的类别在计算损失函数时的权重，最终起到抑制过拟合的效果。

下次预告：学习率与最优化方法