# 问题

我们在机器学习文件夹的问题“11_三种集成学习思想简介”中大体上介绍了 bagging 思想，在这个问题中，我们便具体讨论下这种思想，且与DNN中的dropout思想做下对比。

# bagging

要得到泛化性能强的集成，集成中的个体学习器应尽可能表现好且相互独立，即“好而不同”。但是“独立”的学习方法在现实任务中无法做到，因为同一个数据集，训练得到的学习器肯定不会完全独立，但可以设法使基学习器尽可能具有较大的差异。给定一个训练数据集，一种可能的做法是对训练样本进行采样，产生出若干个不同的子集，再从每个数据子集中训练出一个基学习器。这样，由于训练数据的不同，我们获得的基学习器可望具有较大的差异。然而，为了获得好的集成，我们同时还希望个体学习器不能太差，如果采样出的每个子集都完全不同，则每个学习器都只用到了一小部分的训练数据，甚至不足以进行有效学习。为解决这个问题，我们可考虑使用相互有交叠的采样子集。

bagging对训练数据集的采样使用的是 **bootstrap 自助采样法**，因此这里先对这个方法进行简单介绍：

> 给定包含 m 个样本的数据集 $D$ ，我们对它进行采样产生数据集 $D'$ ：每次随机从 $D$ 中挑选一个样本，将其拷贝放入 $D'$ ，然后再将该样本放回初始数据集 $D$ 中，使得该样本在下次采样时仍有可能被采到；这个过程反复执行 m 次后，我们就得到了包含 m 个样本的数据集 $D'$ ，这就是自助采样法的结果。显然，$D$ 中有一部分样本会在 $D'$ 中多次出现，而另一部分样本不出现，可以做个简单的统计，样本在 m 次采样中始终不被采到的概率是 $(1-\frac{1}{m})^m$ ，取极限得到：
>
> $$\lim_{m\to\infty}(1-\frac{1}{m})^m=\frac{1}{e}\approx 0.368 \tag{1}$$

照上面的自助采样法，我们可以采样出 T 个含有 m 个训练样本的采样集，然后基于每个采样集训练出一个基学习器，再将这些基学习器进行结合，这便是 bagging 方法的基本流程。**在对预测进行结合时，Bagging 通常对分类任务使用简单投票法，对回归任务使用简单平均法。**

bagging方法之所以有效，是因为并非所有的分类器都会产生相同的误差，只要有不同的分类器产生的误差不同，就会对减小泛化误差有效。

**与 Adaboost 的区别：**

> 标准 AdaBoost 只适用于二分类任务，而 Bagging 能不经修改地用于多分类与回归任务。

# Bagging 与 Dropout 的联系

dropout 思想继承自 bagging方法。bagging是每次训练一个基分类器的时候，都有一些样本对该基分类器不可见，而dropout是每次训练的时候，都有一些神经元对样本不可见。

我们可以把 dropout 类比成将许多大的神经网络进行集成的一种 bagging 方法。但是每一个神经元的训练是非常耗时和占用内存的，训练很多的神经网络进行集合分类就显得太不实际了，但是 dropout可以看做是训练所有子网络的集合，这些子网络通过去除整个网络中的一些神经元来获得。

**dropout 具体怎么去除一个神经元呢？**可以在每个神经元结点处独立采样一个二进制掩膜，采样一个掩膜值为 0 的概率是一个固定的超参数，则掩膜值为 0 的被去除，掩膜值为 1 的正常输出。

## bagging与dropout训练的对比

- 在bagging中，所有的分类器都是独立的，而在dropout中，所有的模型都是共享参数的。

- 在bagging中，所有的分类器都是在特定的数据集下训练至收敛，而在dropout中没有明确的模型训练过程。网络都是在一步中训练一次（输入一个批次样本，随机训练一个子网络）

## dropout的优势

- very computationally cheap。在dropout训练阶段，每一个样本每一次更新只需要O(n)，同时要生成n个二进制数字与每个状态相乘。除此之外，还需要O(n)的额外空间存储这些二进制数字，直到反向传播阶段。
- 没有很显著的限制模型的大小和训练的过程。

# 参考资料

《机器学习》西瓜书
从bagging到dropout（deep learning笔记）