

1. 为什么要dropout? 其为何能防止过拟合?

在机器学习的模型中，如果模型的参数太多，而训练样本又太少，训练出来的模型很容易产生过拟合的现象。

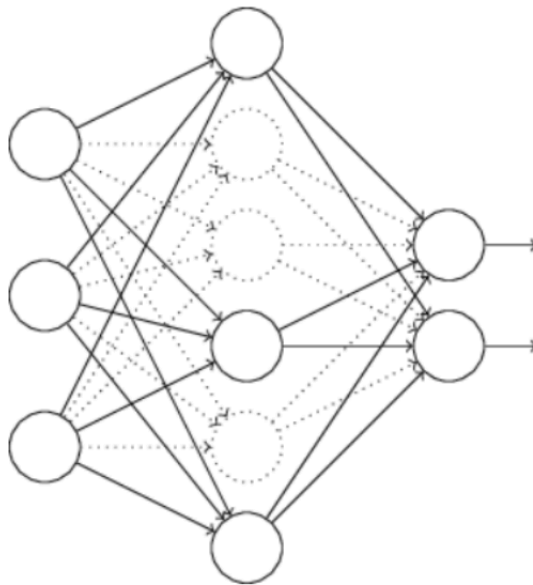
为了解决过拟合问题，一般会采用模型集成的方法，即训练多个模型进行组合。此时，训练模型费时就成为一个很大的问题。

Dropout可以比较有效的缓解过拟合的发生，在一定程度上达到正则化的效果，具体原因如下。

- **取平均**：传统的bagging做法可能是训练n个不同的神经网络模型，用这5个模型的预测结果投票或者取平均，这样不同网络可能在不同地方过拟合，综合起来有一定的抵消作用。dropout其实在过程中不断改变神经网络的结构，整个过程其实就是在对不同的网络取平均。
- **减少神经节点之间的依赖关系**：因为dropout程序导致两个神经元不一定每次都在一个dropout网络中出现。这样权值的更新不再依赖于有固定关系的隐含节点的共同作用，阻止了某些特征仅仅在其它特定特征下才有效果的情况。从这个角度看**dropout**就有点像**L1, L2正则**，减少权重使得网络对丢失特定神经元连接的鲁棒性提高。

2. dropout原理

我们在前向传播的时候，让某个神经元的激活值以一定的概率p停止工作，也就是节点有p概率会被置为0（反向传播时，只更新没有停止工作的节点），这样可以使模型泛化性更强，因为它不会太依赖某些局部的特征。



3. 神经网络Dropout层中为什么dropout后还需要进行rescale?

rescale也被称为**inverted dropout**。

若有n个节点，设置dropout为0.4，则有大约40%的节点被置0，也就是只有60%的节点参与了训练。但是在预测阶段时没有dropout的，所有的节点都会参与预测，这就导致预测阶段的结果比训练结果要偏大。

所以为了使训练和预测阶段的结果规模一致，我们在训练阶段将结果扩大 $1 / 60\%$ ，这就是**rescale**。