

CNN 入门讲解：什么是dropout?

本文首发于[卷积神经网络（CNN）入门讲解](#)

关注公众号：follow_bobo, 加入机器学习交流群

最近有个新闻闹得沸沸扬扬，那就是dropout这个算法竟然被谷歌申请了专利

真是神奇了，加减乘除还能被申请专利？

不过大家不用担心，dropout肯定用还是可以用的，谷歌只是想特定时刻，利用专利保护或者对抗某些竞争对手或者不良企业

那么今天我们就来看看，这个dropout到底是如何一顿操作猛如虎，让谷歌能花费3年时间就为了一个简单的算法申请专利

dropout 从字面意思就是 ‘丢掉’

丢掉？丢掉什么？

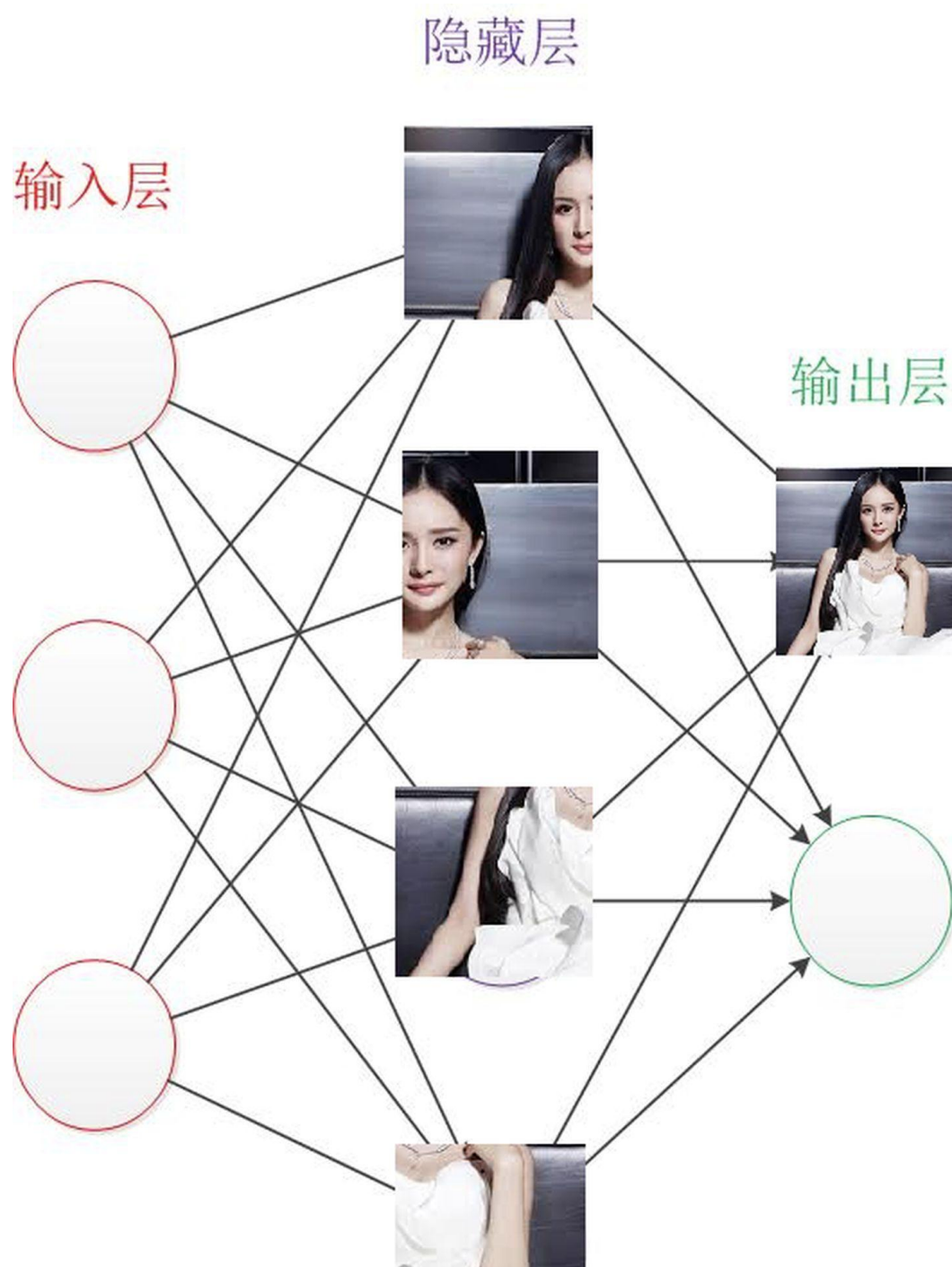
别急，我们来看看？



上图特征清晰明了，想必大家一眼就能认出是谁了把

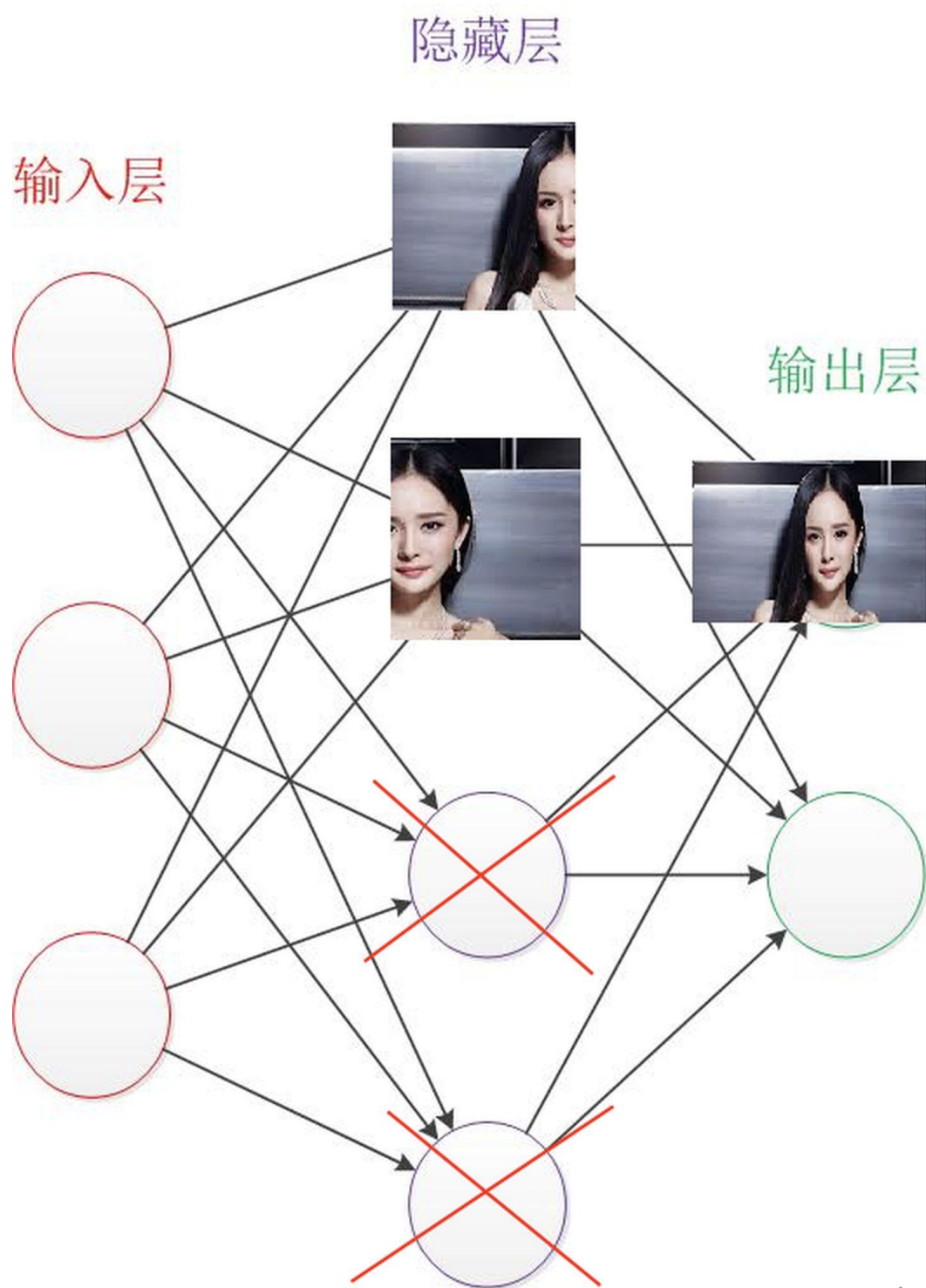
根据这些特征，神经网络也能认出来，so easy





ok, 那我们增加难度, 丢掉一些特征, 如下图





你是不是立马就认出来了？

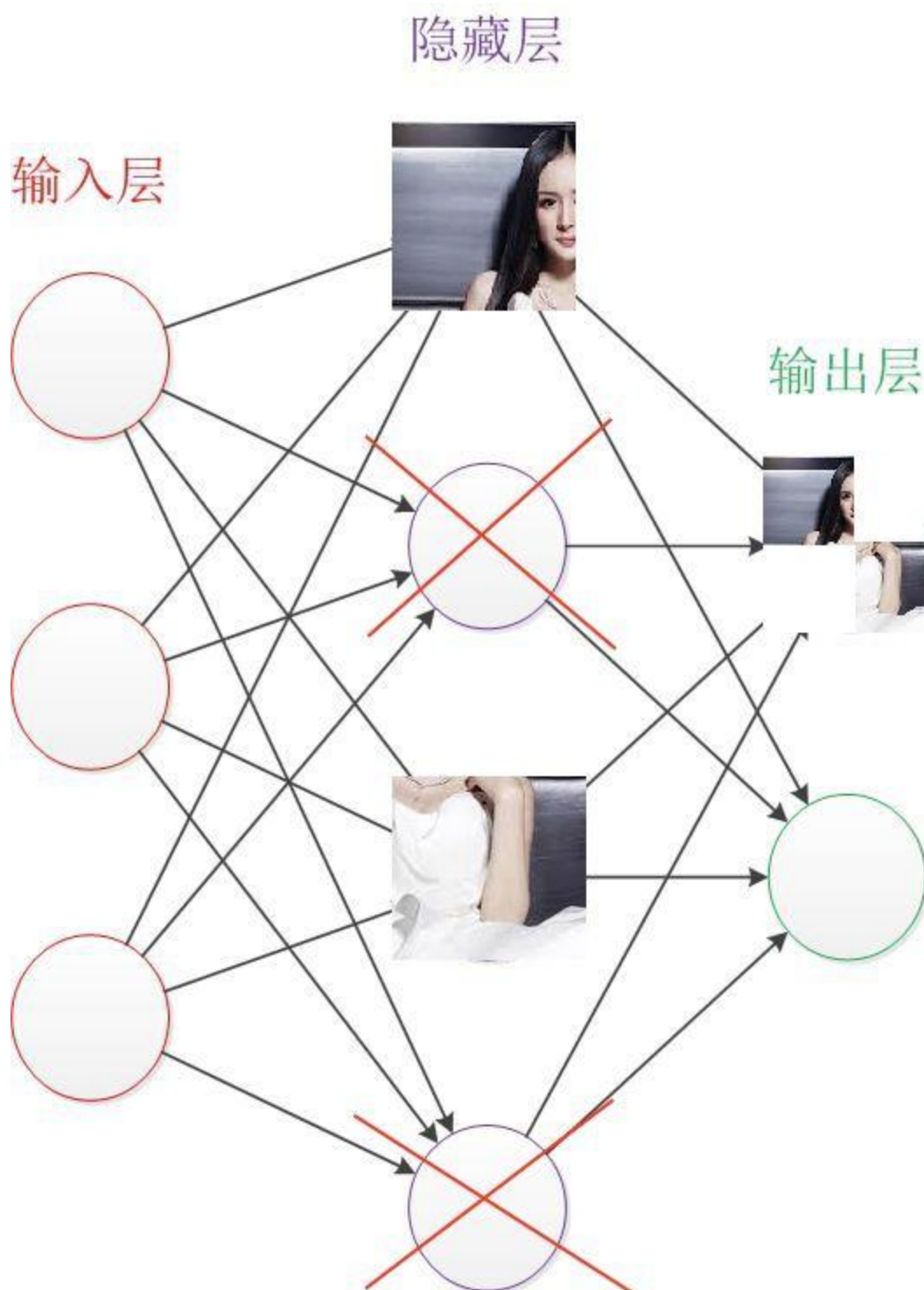
同样的对模型也这么做，我们把提取衣服特征的神经元去掉，模型鲁棒性就更强，为什么？因为模型学到

认人，看脸就行，衣服什么的，无所谓



但这样模型非得看到左右脸才行，神经元提取的特征耦合度太高，我必须左右脸同时出现，模型才能作出判断，神经网络的鲁棒性仍然不够，怎么办？

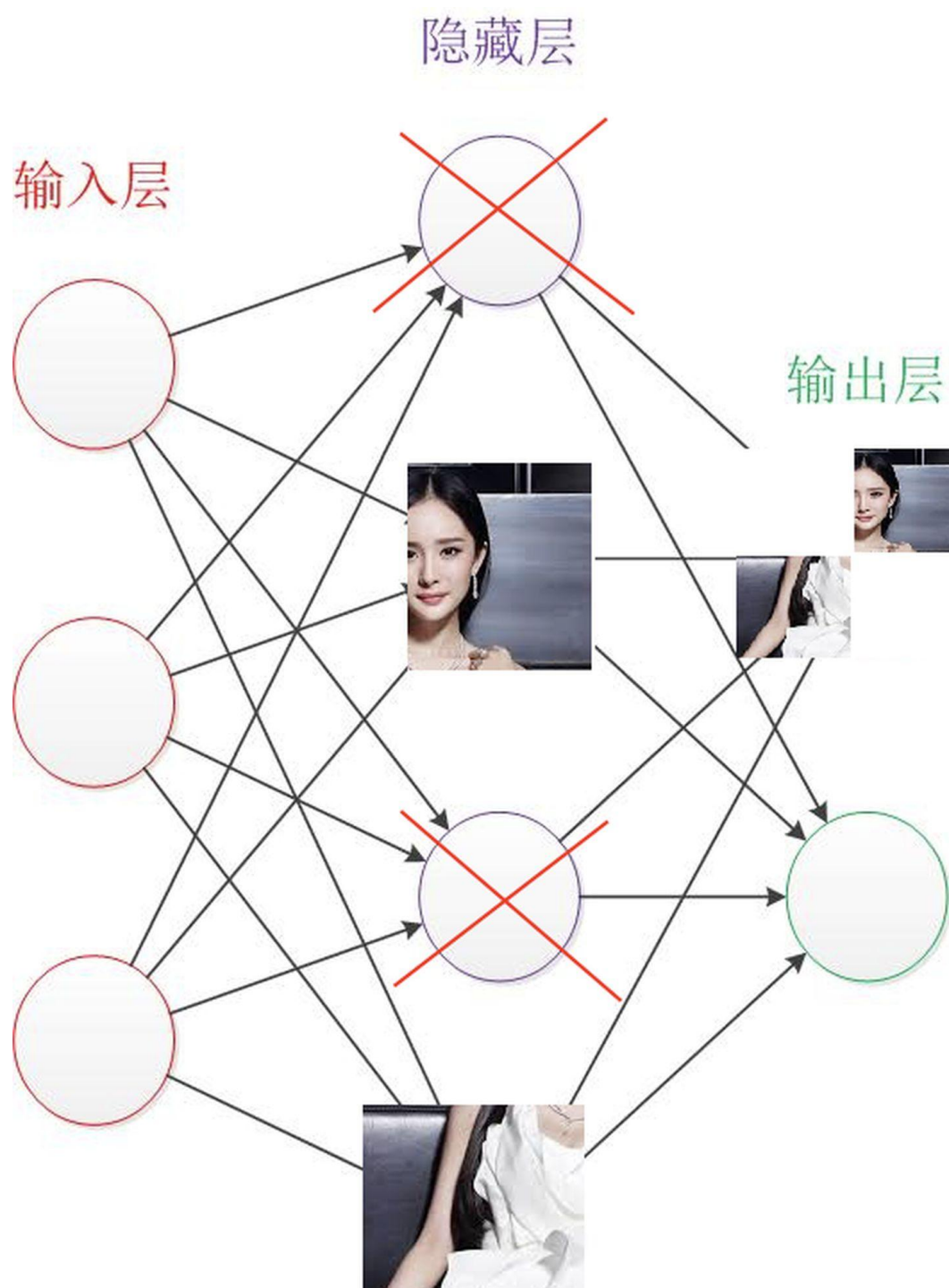
我再随机抽取一半的神经元，让他们不参与任何计算，也不参与到权重的更新，比如这样



(打红叉表示不参与计算，不是被删掉的意思)

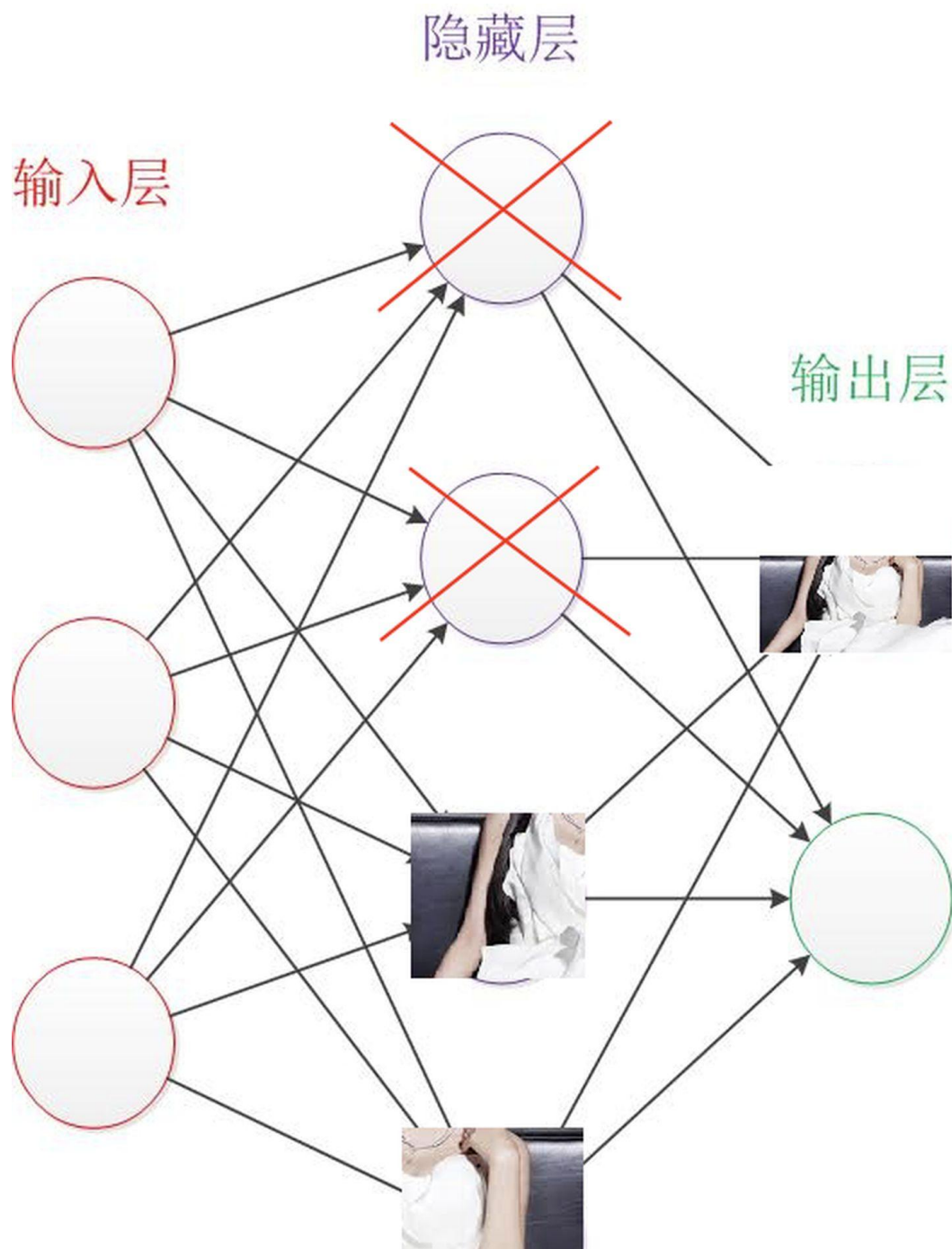
或者这样





当然，还有这种：





一般人光看xiong是认不出这是大幂幂的

这种情况就是dropout rate 设置太大，我们这是0.5，在训练的时候不幸丢掉了重点特征，造成模型overfitting了

当然有些同学说，老师老师，更据我多年阅人无数的经验，我认得出来



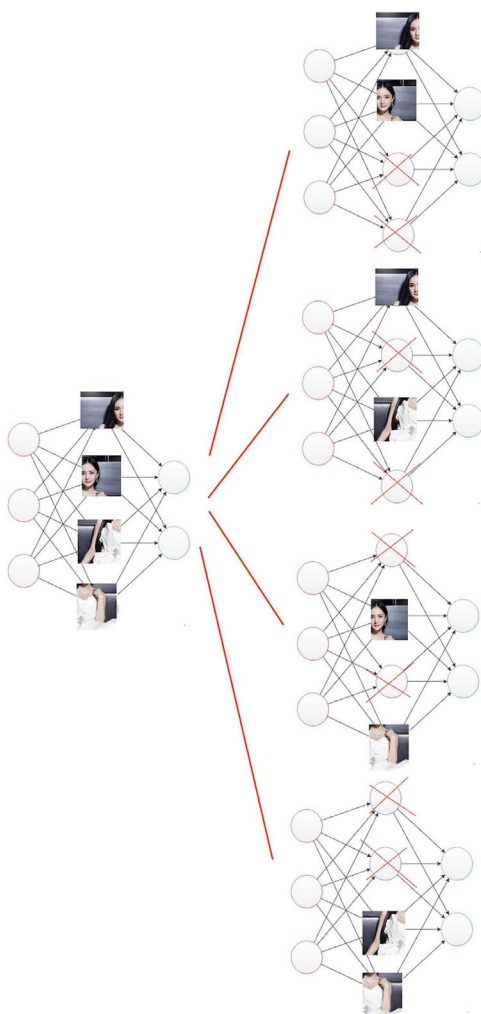


你这司机可以啊

所以控制好dropout rate也是调参的关键，调好了就是加大模型鲁棒性，调不好就是overfitting

一般情况，dropout rate 设为0.3-0.5即可

所以你看，每次训练都随机让一定神经元停止参与运算，简单的操作让我们由一个模型演变成四个模型：



一个模型的作用效果，也转变成4个模型的效果和，更像一个多模型投票

所以，虽然第四个模型overfitting了，但是前三个也模型参与投票判断，第四个模型overfitting就被弱化了

所以，dropout的出现，使得模型稳定性和鲁棒性被大大提高了

到这，你应该是理解dropout 的原理和作用了，但是还有关键的一步

那就是训练的时候会停止训练一些神经元，但是测试的时候，整个模型是完整，可不会dropout任何神经元啊

这样就出现了一个问题，就好比

平时训练200米跑步，突然比赛是400米

茫然不知所措

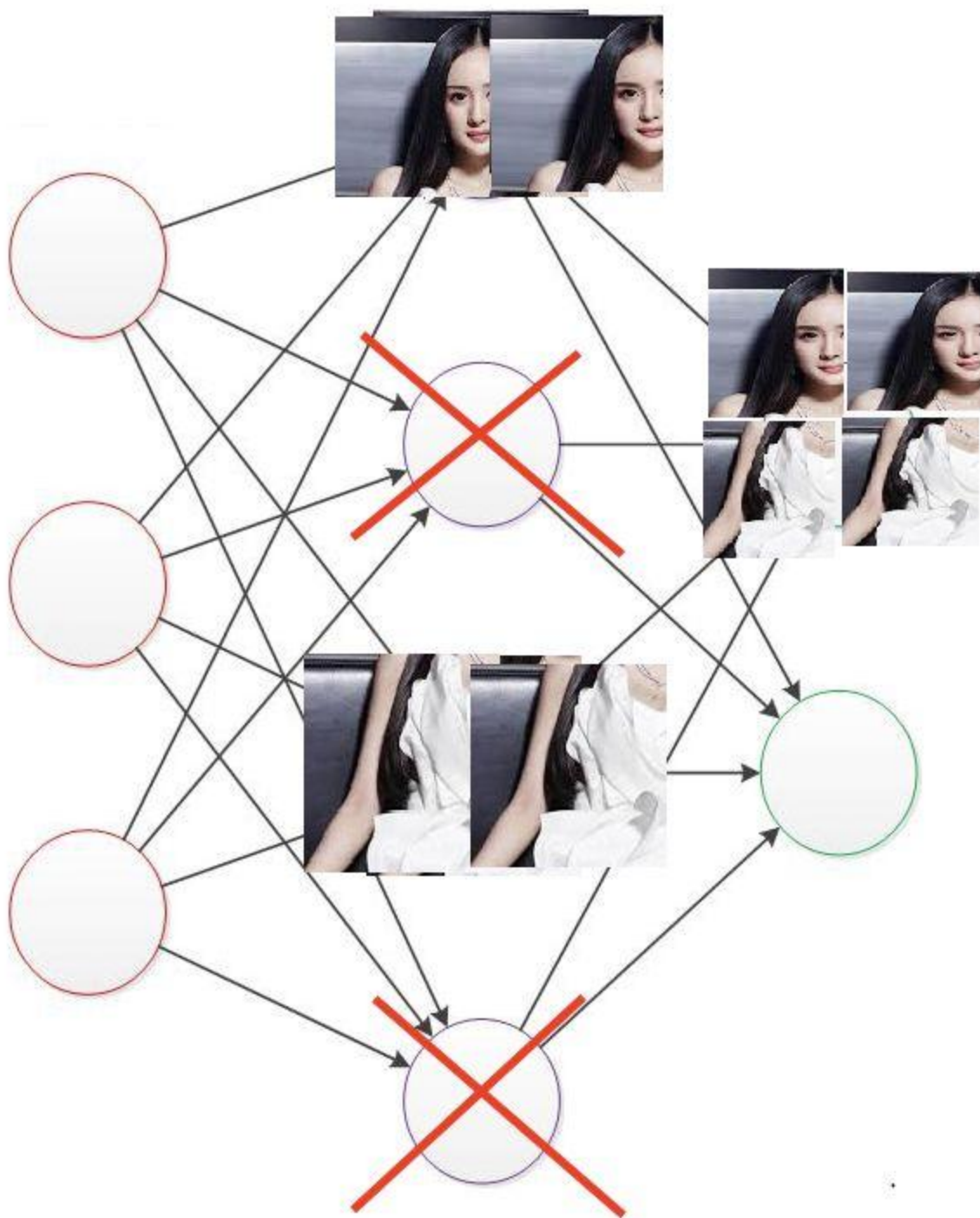


所以，在训练的时候，我们会对**没有被dropout**的神经元权值做一个rescale

$\text{rescale rate} = 1 / (1 - \text{dropout rate})$

我们这dropout rate = 0.5，**为了方便理解**，我画了一个大概的过程如下图：





用数学语言就是：

没用dropout的神经网络前向传播计算公式:

$$\begin{aligned} z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)} \mathbf{y}^l + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}), \end{aligned}$$



用了dropout 的神经网络前向传播计算公式：

$$\begin{aligned}r_j^{(l)} &\sim \text{Bernoulli}(p), \\ \tilde{\mathbf{y}}^{(l)} &= \mathbf{r}^{(l)} * \mathbf{y}^{(l)}, \\ z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)} \tilde{\mathbf{y}}^{(l)} + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}).\end{aligned}$$

因此我在训练的时候就做好rescale, 至少可以一定程度上 ‘弥补’ 测试中没有使用dropout所带来的副作用

OK, 如果有小伙伴们还有不清楚的, 可以看看谷歌官方专利上的解释:

[patentimages.storage.googleapis.com...](http://patentimages.storage.googleapis.com/...)

