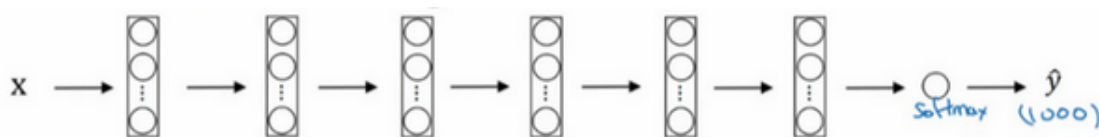


2.9 迁移学习（Transfer Learning）

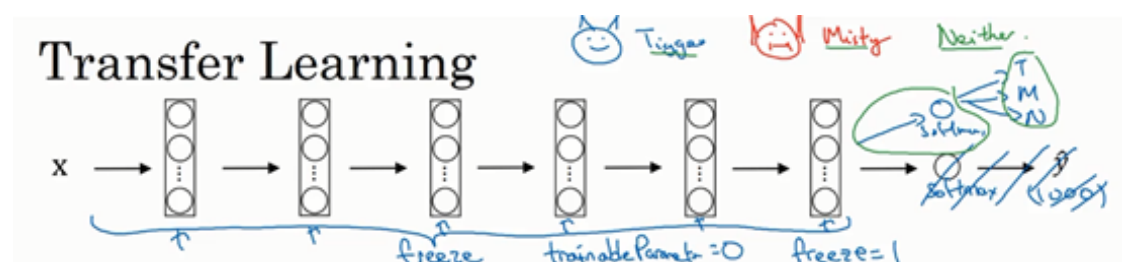
如果你要做一个计算机视觉的应用，相比于从头训练权重，或者说从随机初始化权重开始，如果你下载别人已经训练好网络结构的权重，你通常能够进展的相当快，用这个作为预训练，然后转换到你感兴趣的任务上。计算机视觉的研究社区非常喜欢把许多数据集上传到网上，如果你听说过，比如 **ImageNet**，或者 **MS COCO**，或者 **Pascal** 类型的数据集，这些都是不同数据集的名字，它们都是由大家上传到网络的，并且有大量的计算机视觉研究者已经用这些数据集训练过他们的算法了。有时候这些训练过程需要花费好几周，并且需要很多的 **GPU**，其它人已经做过了，并且经历了非常痛苦的寻最优过程，这就意味着你可以下载花费了别人好几周甚至几个月而做出来的开源的权重参数，把它当作一个很好的初始化用在你自己的神经网络上。用迁移学习把公共的数据集的知识迁移到你自己的问题上，让我们看一下怎么做。



举个例子，假如说你要建立一个猫咪检测器，用来检测你自己的宠物猫。比如网络上的 **Tigger**，是一个常见的猫的名字，**Misty** 也是比较常见的猫名字。假如你的两只猫叫 **Tigger** 和 **Misty**，还有一种情况是，两者都不是。所以你现在有一个三分类问题，图片里是 **Tigger** 还是 **Misty**，或者都不是，我们忽略两只猫同时出现在一张图片里的情况。现在你可能没有 **Tigger** 或者 **Misty** 的大量的图片，所以你的训练集会很小，你该怎么办呢？

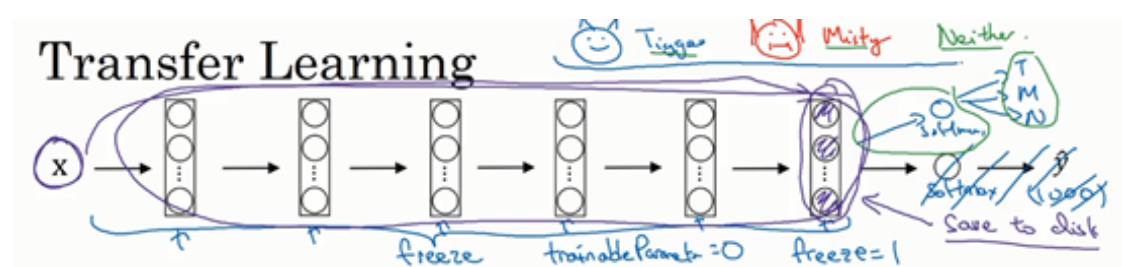


我建议你从网上下载一些神经网络开源的实现，不仅把代码下载下来，也把权重下载下来。有许多训练好的网络，你都可以下载。举个例子，**ImageNet** 数据集，它有 1000 个不同的类别，因此这个网络会有一个 **Softmax** 单元，它可以输出 1000 个可能类别之一。

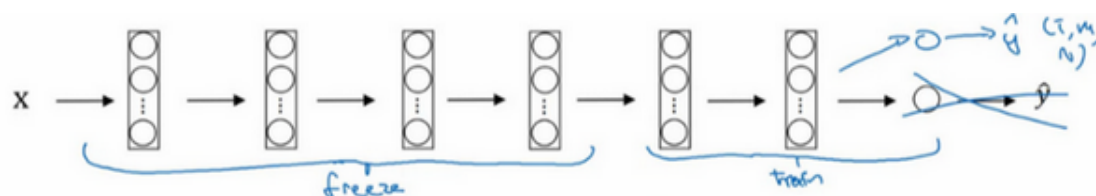


你可以去掉这个 **Softmax** 层，创建你自己的 **Softmax** 单元，用来输出 **Tigger**、**Misty** 和 **neither** 三个类别。就网络而言，我建议你把所有的层看作是冻结的，你冻结网络中所有层的参数，你只需要训练和你的 **Softmax** 层有关的参数。这个 **Softmax** 层有三种可能的输出，**Tigger**、**Misty** 或者都不是。

通过使用其他人预训练的权重，你很可能得到很好的性能，即使只有一个小的数据集。幸运的是，大多数深度学习框架都支持这种操作，事实上，取决于用的框架，它也许会有 `trainableParameter=0` 这样的参数，对于这些前面的层，你可能会设置这个参数。为了不训练这些权重，有时也会有 `freeze=1` 这样的参数。不同的深度学习编程框架有不同的方式，允许你指定是否训练特定层的权重。在这个例子中，你只需要训练 **softmax** 层的权重，把前面这些层的权重都冻结。

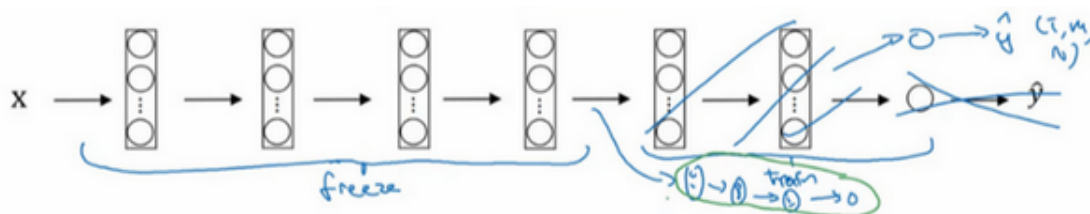


另一个技巧，也许对一些情况有用，由于前面的层都冻结了，相当于一个固定的函数，不需要改变。因为你不需要改变它，也不训练它，取输入图像 X ，然后把它映射到这层（**softmax** 的前一层）的激活函数。所以这个能加速训练的技巧就是，如果我们先计算这一层（紫色箭头标记），计算特征或者激活值，然后把它们存到硬盘里。你所做的就是用这个固定的函数，在这个神经网络的前半部分（**softmax** 层之前的所有层视为一个固定映射），取任意输入图像 X ，然后计算它的某个特征向量，这样你训练的就是一个很浅的 **softmax** 模型，用这个特征向量来做预测。对你的计算有用的一步就是对你的训练集中所有样本的这一层的激活值进行预计算，然后存储到硬盘里，然后在此之上训练 **softmax** 分类器。所以，存储到硬盘或者说预计算方法的优点就是，你不需要每次遍历训练集再重新计算这个激活值了。

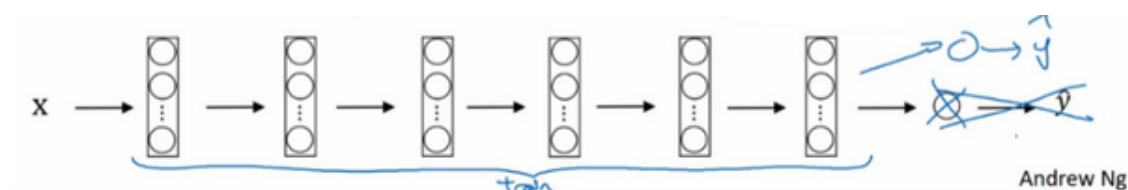


因此如果你的任务只有一个很小的数据集，你可以这样做。要有一个更大的训练集怎么办呢？根据经验，如果你有一个更大的标定的数据集，也许你有大量的 **Tigger** 和 **Misty** 的照片，还有两者都不是的，这种情况，你应该冻结更少的层，比如只把这些层冻结，然后训练

后面的层。如果你的输出层的类别不同，那么你需要构建自己的输出单元，**Tigger**、**Misty** 或者两者都不是三个类别。有很多方式可以实现，你可以取后面几层的权重，用作初始化，然后从这里开始梯度下降。



或者你可以直接去掉这几层，换成你自己的隐藏单元和你自己的 **softmax** 输出层，这些方法值得一试。但是有一个规律，如果你有越来越多的数据，你需要冻结的层数越少，你能够训练的层数就越多。这个理念就是，如果你有一个更大的数据集，也许有足够多的数据，那么不要单单训练一个 **softmax** 单元，而是考虑训练中等大小的网络，包含你最终要用的网络的后面几层。



最后，如果你有大量数据，你应该做的就是用开源的网络和它的权重，把这、所有的权重当作初始化，然后训练整个网络。再次注意，如果这是一个 1000 节点的 **softmax**，而你只有三个输出，你需要你自己的 **softmax** 输出层来输出你要的标签。

如果你有越多的标定的数据，或者越多的 **Tigger**、**Misty** 或者两者都不是的图片，你可以训练越多的层。极端情况下，你可以用下载的权重只作为初始化，用它们来代替随机初始化，接着你可以用梯度下降训练，更新网络所有层的所有权重。

这就是卷积网络训练中的迁移学习，事实上，网上的公开数据集非常庞大，并且你下载的其他人已经训练好几周的权重，已经从数据中学习了很多了，你会发现，对于很多计算机视觉的应用，如果你下载其他人的开源的权重，并用作你问题的初始化，你会做的更好。在所有不同学科中，在所有深度学习不同的应用中，我认为计算机视觉是一个你经常用到迁移学习的领域，除非你有非常非常大的数据集，你可以从头开始训练所有的东西。总之，迁移学习是非常值得你考虑的，除非你有一个极其大的数据集和非常大的计算量预算来从头训练你的网络。