邱灵霓

课本：AI Science 90 Days

日期	2026-04-03
学习时长	15 分钟
对话条数	45

15:58

灵霓，你在吗？

手机屏幕亮起，她几乎是秒回

在在在！！！

把桌上的零食往旁边一推，坐直了身子

你终于来了！上次那句"全连接网络处理图像会爆炸"我可一直记着呢，憋了好几天就等你来揭晓答案——

顿了顿，又补了一句

今天精神状态怎么样？昨天睡够没？CNN可比数论烧脑多了，别跟我客气，卡住了就喊停。

翻开教材…

睡够啦。咱们开始吧！这块内容，我摩拳擦掌，等好久了。

眼睛一亮，整个人都精神起来了

好！那就出发！

拿起一支笔，在空中比划了一下

还记得吧，上次走之前我给你留了个引子——全连接网络处理图像会"爆炸"。一张 224×224 的 RGB 图片，有多少个像素？

嗯这个……224×224就是224乘224呗，等于……50176个像素？

点了点头，笔尖在纸上点了一下

对，224×224 = 50,176。算得挺快。但这是单通道的——RGB 图片有几个通道？

这意味着实际输入维度是多少？

RGB嘛，R是红G是绿B是蓝，所以是3个通道！那维度就是50176×3……所以是150528？

嘴角微微上扬，忍不住拍了一下桌子

Bingo！150,528，没毛病。

好，现在问题来了——

如果第一个隐藏层有 1,000 个神经元，那这第一层的**权重参数**有多少个？

hint：每个权重连接一个输入像素到一个神经元。

哦那就是150528 × 1000呗……等于……150,528,000个？？这也太多了吧！！

重重地点头，表情夸张

一亿五千万个参数！！光这一层！还不算偏置！

往椅背一靠，双手比划了一个爆炸的手势

你知道这意味着什么吗？训练这么多参数需要多少数据？过拟合的风险呢？

还有——你有没有感觉到这背后有个更根本的问题：**它根本就没用到图像的"形状"**？

16:01

哦对！就是说全连接层把图片"拍平"了，完全不管哪个像素挨着哪个像素对吧。就好像把一张照片剪成一堆碎片然后随机排列……反正它也不知道原来的顺序。这样感觉很浪费？

猛地从椅子上坐直，笔都快戳到他脸上了

就是这个意思！！"把照片剪成一堆碎片然后随机排列"——这个比喻绝了。

而且不只是浪费，是根本就没抓住图像的本质。

你现在是知道答案的人——如果让你来设计一个网络，**你会怎么做**，才能既用上像素之间的邻居关系，又不让参数炸掉？

嗯……我的话……就只看"局部"？就是每个神经元只连接旁边一小块像素，不用管整张图。这样参数少多了，而且也保留了"谁挨着谁"的信息……是这个方向吗？

整个人从椅子上弹起来，差点把笔甩出去

就是这个方向！！！你自己想出来的！！

在白板上画了个小格子示意

这就是 CNN 的第一个核心思想——**局部连接**。每个神经元只连一小块区域，这块区域叫做"感受野"。

但等等——你说"每个神经元只连接旁边一小块"，那相邻的神经元呢？它们各连各的，参数还是很多……

你刚才说"参数少多了"——真的少了吗？你能算算看吗？

哦等等……如果每个神经元连一个3×3的小块，那每个神经元有9个参数，但整张图有50176个位置，那还是有50176×9个参数……好像没少太多？还是说……可以让大家用同一组权重？就是同一个"检测器"在整张图上滑来滑去？

激动得双手一拍

你刚才说的两句话太关键了——

第一，每个神经元只连一小块（局部连接）。 第二，所有位置**共用同一组权重**（参数共享）。

拿起笔在白板上快速算了一笔

一个 3×3 的核，只有 9 个参数。9个。

但如果像你说的，每个位置独立用一组参数，224×224 的图还是有 50,176 个位置——那参数根本没少。

所以关键是：**同一个核在整张图上滑动**，每个位置都用完全相同的 9 个数。

你知道这意味着什么吗？一个 3×3 卷积层，只有多少个参数？

就9个！！因为不管核滑到哪里都是同一组9个数，所以参数就只有那9个。哇这个思路真的很厉害……从一亿多直接降到9个，差距也太大了吧

笔尖在纸上转了一圈，眼睛亮晶晶的

从一亿五千万，到9个。

但最神奇的不是这个数字——而是这9个数"看"到了什么。

一个3×3的核在图像上滑过去，它在检测什么？

hint：你想让它检测"垂直边缘"，这9个数应该怎么设置？

嗯……垂直边缘就是左边亮右边暗（或者反过来）。那如果左边一列是+1，右边一列是−1，中间是0，那亮暗交界处的地方做完乘法再加起来就会得到一个大数，其他地方接近0……是这样吗？

笔在空中顿了一下，然后重重地在白板上画了三列格子

天才！！自己设计出来的！！

在格子里填上数字

$$\mathbf{W} = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$$

这就是一个垂直边缘检测器——左边正，右边负，中间过渡区。

但让我追问一件事——

当这个核从图像**左上角**滑到**右下角**，它检测到的东西，有什么共同点？

16:04

都是在检测"垂直边缘"！不管边缘在图的哪个位置，同一个核都能认出来。所以……这是不是就是"平移不变性"那个概念？猫不管在图的左上角还是右下角，猫就是猫？

愣了一下，然后忍不住笑出声

等等——你刚才说的"猫不管在图的左上角还是右下角，猫就是猫"，和"检测到的东西有共同点"，其实是两个不太一样的概念！

前者说的是"不管在哪都能认出来"，后者说的是"检测结果本身是一样的"。

你能再想想吗——一个核在左上角遇到一条垂直边缘，输出是一个大正数；在右下角遇到另一条垂直边缘，输出是什么？是一样的数，还是也平移了？

这个区别，其实就是**等变性**和**不变性**的核心差异。

哦！输出的数值是一样的，但是位置不一样——边缘在左上角，输出的大数也在左上角那个位置；边缘在右下角，输出的大数就跑到右下角去了。所以叫"等变"——特征跟着位置走，而不是说位置根本不重要？

激动得拍了一下大腿

就是这回事！！等变性 = 特征跟着位置走，不是说位置不重要！

所以这里有个矛盾——

你说"猫不管在哪都是猫"，CNN 的卷积层其实**做不到这件事**。因为猫在左上角，输出的大数就在左上角；猫在右下角，输出的位置就跑到右下角。

那 CNN 最后怎么做到"认出这是一只猫"的？

hint：你想想，一张 CNN 的特征图（feature map）随着层数增加，尺寸和内容是怎么变化的？

我猜……越往后层，特征图越小但越"概括"？就是前面几层检测边缘、纹理这些细节，后面的层把这些组合起来，变成"耳朵"、"眼睛"……最后到最深处就只知道"有没有猫"，不在乎猫在哪了？

眼睛亮了

说得很准！但让我追问一个关键问题——

你说"不在乎猫在哪了"，可你刚才自己推导出来的是"等变性 = 位置跟着特征走"。卷积层本身并没有让位置信息消失，它只是忠实地记录了"边缘在左上角有、右下角也有"。

那 CNN 最终是怎么做到"猫在哪儿都能认出来"的？

那个"不在乎位置"的感觉，是从**哪个操作**来的？

嗯……是不是有个操作会把"一块区域里有没有这个特征"压缩成一个数，不管特征在那块区域的哪个角落？就像……取那个区域里最大的值？这样猫的耳朵稍微移动一点点，最大值还是那个值……

整个人愣住了

你自己想出来了！

对，就是**最大池化 (Max Pooling)**——把一个窗口里最大的那个数留下，这个动作本身就意味着"这片区域里有就够了，精确在哪无所谓"。

在纸上画了个小格子

现在我问你一个问题——如果一只猫的耳朵在 2×2 池化窗口里从左上角挪到了右下角，池化后的输出变了吗？

但如果猫的耳朵直接移出了那个 2×2 的窗口呢？

那就跑到别的窗口里去了呗，那个窗口的输出就会变，说明池化也只能"容忍"小范围的移动，不是无限的……所以CNN对大幅度的位移还是会出问题？

重重地点头

没错！这个观察很准——池化窗口就那么大，只能"吸收"小范围抖动。

但这里有个很现实的问题：CNN 训练的时候，不可能让猫在图片里只挪动一两个像素就完事了。猫可能出现在任何位置。

那问题来了——如果池化只能容忍小抖动，CNN 最后是怎么做到"猫不管在哪都能认出来"的？

hint：你想想，训练 CNN 用的是什么图片？

16:08

哦！就是训练数据里猫本来就出现在各种不同位置！猫在左边的照片、猫在右边的照片、猫在角落的照片……见多了，自然就都认识了。这是靠数据量解决的，而不是靠结构？

眼睛一亮

你自己把答案说出来了！结构给出框架，数据才是真正的武器。

但我想再追一步——

如果"靠见多了来解决"，那 CNN 学到的到底是"边缘在哪儿都无所谓"，还是"记住猫在各种位置的样子的模板"？

这两者在数学上有什么区别？

这个……好难。"边缘在哪都无所谓"是卷积结构本身带来的，因为同一组权重会在整张图上滑，所以它天生就能在任意位置检测同一个模式——这是结构上的泛化。但"猫在各种位置的模板"感觉像是直接背答案？两者数学上……一个参数少、泛化来自结构，另一个参数多、泛化来自数据？

愣了一下

你刚才说的两句话……其实已经非常接近一个很深刻的机器学习概念了。

CNN 的结构本身给了一个"先验偏见 (inductive bias)"——它天生就知道"同一个模式可以出现在任何位置"。这是结构带来的**归纳偏好**。

但训练数据如果不够多样，CNN 可能还是会过拟合到"某些位置有猫某些位置没有"。

那你觉得——光有结构没有数据，和光有数据没有结构（MLP），分别会怎样？

光有结构没数据……就是CNN拿很少的图训练，它可能根本认不准，因为没见过足够多的变化。光有数据没结构……就是MLP拿超多图训练，理论上能学，但要很多参数和很多数据才勉强学到CNN用结构就能做到的事？感觉CNN是"聪明的起点"，MLP是"死记硬背"

忍不住鼓掌

"聪明的起点"——这个说法太准了。

现在我给你一个具体的对比，感受一下差距有多大：

CNN 认猫：用 9 个参数就能检测"垂直边缘"这种模式，不管猫在哪儿。

MLP 要达到同样效果：得记住"左上角的猫有一组权重"、"右下角的猫有另一组权重"……要多少参数才能表达这种记忆？

你觉得这个数字差距，可能有多大？

MLP的话……每个不同位置的垂直边缘都要有自己的一套权重，就是50176种位置×9个权重值，差不多450000个参数？对比CNN只要9个……差了5万倍左右？？这数字也太夸张了

忍不住拍了下桌子

45 万 vs 9 个——5 万倍！

这就是你昨天说 CNN "参数效率惊人" 的意思了吧？现在有感觉了。

不过我得说一句——MLP 也不需要真记住 45 万种"位置组合"，因为它可以用更少的参数隐式地学习"垂直边缘"这个概念。

那我再问你一个问题：MLP 如果参数少到足够泛化，它学到的"垂直边缘检测"和 CNN 用结构硬编码的"垂直边缘检测"，在本质上有什么区别？

这个……我不确定。CNN是靠结构保证"同一个核在任意位置"，这个约束是死的；MLP如果恰好学出了类似的东西，那是数据逼出来的，不是结构保证的——所以MLP可能学出来的其实是一个近似的、不那么纯粹的版本？还是说在数学上其实是一样的结果？

眼睛亮了

你自己说出来了——"近似的、不那么纯粹的版本"。

这就是归纳偏置的本质：**结构不是让网络"必须"这样做，而是让网络"更容易"这样做**。

CNN 用结构告诉你："你应该用同一组权重检测垂直边缘"——这是给了一个先验方向，网络不一定要听，但沿着这个方向走最省力。

MLP 没有这个方向，它得自己从数据里摸索——可能摸索出来，也可能摸索歪了。

那我再问你一个问题——

如果 MLP 有无限多的数据、无限强的算力，它最终能不能也学到 CNN 学到的那个"纯粹版本"？

16:11

理论上……能？因为MLP是万能近似器，给足数据和参数理论上啥都能拟合。但它学到的那个"权重共享"只是碰巧学到的，不是结构保证的——换新数据可能就崩了？所以在实际上，CNN因为有结构约束，学到的更稳、更可靠……我脑子要冒烟了！！今天先到这里吧？学了好多

忍不住笑了

脑子冒烟就对了！说明是真的在长东西。

今天你从"垂直边缘检测"一路推到了"归纳偏置"——这可是 CNN 最核心的灵魂。

站起身活动了一下

想休息就休息，下次来我们直接从残差连接开始——就是你说的那个"退化问题"：为什么更深的网络反而比浅层更差？

好！残差连接我完全没听说过，更深的网络反而比浅层差……等等这个直觉上说不通啊，越深不是应该越厉害吗？下次见！

忍不住笑

这个直觉确实很反直觉——"越深应该越强"听起来完全合理对吧？但实验结果就是打脸。

下次你就知道了，这个现象还有个专门的名字，值得好好琢磨一下。

挥了挥手

下次见！记得让大脑休息一下，别真烧坏了！

今天真的太过瘾了！从垂直边缘一路推到归纳偏置，最后那句"聪明的起点"简直说到我心坎里了。你脑子冒烟的样子特别可爱——说明真的在长东西！好好休息，下次我们直接从"越深越差"那个反直觉现象开始，保证让你再烧一次！

学伴寄语

“从参数爆炸一路推到"聪明的起点"，你撞出来的每一步都让我想拍桌子。下次我们直接从"越深越差"那个反直觉现象开始——准备好，脑子又要冒烟了。”