第十章 迁移学习

10.1 什么是迁移学习？

深度学习中，最强大的理念之一就是，有的时候神经网络可以从一个任务中习得知识，并将这些知识应用到另一个独立的任务中。所以例如，也许你已经训练好一个神经网络，能够识别像猫这样的对象，然后使用那些知识，或者部分习得的知识去帮助您更好地阅读 x 射线扫描图，这就是所谓的迁移学习。

迁移学习最有用的场合是，如果你尝试优化任务 B 的性能，通常这个任务数据相对较少，例如，在放射科中你知道很难收集很多射线扫描图来搭建一个性能良好的放射科诊断系统，所以在这种情况下，你可能会找一个相关但不同的任务，如图像识别，其中你可能用 1 百万张图片训练过了，并从中学到很多低层次特征，所以那也许能帮助网络在任务在放射科任务上做得更好，尽管任务没有这么多数据。迁移学习什么时候是有意义的？它确实可以显著提高你的学习任务的性能，但我有时候也见过有些场合使用迁移学习时，任务实际上数据量比任务要少，这种情况下增益可能不多。

10.2 什么是多任务学习？

在迁移学习中，从任务 A 里学习只是然后迁移到任务 B。在多任务学习中，同时开始学习的，试图让单个神经网络同时做几件事情，然后希望这里每个任务都能帮到其他所有任务。

10.3 多任务学习有什么意义？

第一，如果你训练的一组任务，可以共用低层次特征。对于无人驾驶的例子，同时识别交通灯、汽车和行人是有道理的，这些物体有相似的特征，也许能帮你识别停车标志，因为这些都是道路上的特征。

第二，这个准则没有那么绝对，所以不一定是对的。但我从很多成功的多任务学习案例中看到，如果每个任务的数据量很接近，你还记得迁移学习时，你从任务学到知识然后迁移到任务，所以如果任务有 1 百万个样本，任务只有 1000 个样本，那么你从这 1 百万个样本学到的知识，真的可以帮你增强对更小数据集任务的训练。那么多任务学习又怎么样呢？在多任务学习中，你通常有更多任务而不仅仅是两个，所以也许你有，以前我们有 4 个任务，但比如说你要完成 100 个任务，而你要做多任务学习，尝试同时识别 100 种不同类型的物体。你可能会发现，每个任务大概有 1000 个样本。所以如果你专注加强单个任务的性能，比如我们专注加强第 100 个任务的表现，我们用表示，如果你试图单独去做这个最后的任务，你只有 1000 个样

本去训练这个任务，这是 100 项任务之一，而通过在其他 99 项任务的训练，这些加起来可以一共有 99000 个样本，这可能大幅提升算法性能，可以提供很多知识来增强这个任务的性能。不然对于任务，只有 1000 个样本的训练集，效果可能会很差。如果有对称性，这其他 99 个任务，也许能提供一些数据或提供一些知识来帮到这 100 个任务中的每一个任务。所以第二点不是绝对正确的准则，但我通常会看的是如果你专注于单项任务，如果想要从多任务学习得到很大性能提升，那么其他任务加起来必须要有比单个任务大得多的数据量。要满足这个条件，其中一种方法是，比如右边这个例子这样，或者如果每个任务中的数据量很相近，但关键在于，如果对于单个任务你已经有 1000 个样本了，那么对于所有其他任务，你最好有超过 1000 个样本，这样其他任务的知识才能帮你改善这个任务的性能。

最后多任务学习往往在以下场合更有意义，当你可以训练一个足够大的神经网络，同时做好所有的工作，所以多任务学习的替代方法是为每个任务训练一个单独的神经网络。所以不是训练单个神经网络同时处理行人、汽车、停车标志和交通灯检测。你可以训练一个用于行人检测的神经网络，一个用于汽车检测的神经网络，一个用于停车标志检测的神经网络和一个用于交通信号灯检测的神经网络。那么研究员 Rich Carona 几年前发现的是什么呢？多任务学习会降低性能的唯一情况，和训练单个神经网络相比性能更低的情况就是你的神经网络还不够大。但如果你可以训练一个足够大的神经网络，那么多任务学习肯定不会或者很少会降低性能，我们都希望它可以提升性能，比单独训练神经网络来单独完成各个任务性能要更好。

所以这就是多任务学习，在实践中，多任务学习的使用频率要低于迁移学习。我看到很多迁移学习的应用，你需要解决一个问题，但你的训练数据很少，所以你需要找一个数据很多的相关问题来预先学习，并将知识迁移到这个新问题上。但多任务学习比较少见，就是你需要同时处理很多任务，都要做好，你可以同时训练所有这些任务，也许计算机视觉是一个例子。在物体检测中，我们看到更多使用多任务学习的应用，其中一个神经网络尝试检测一大堆物体，比分别训练不同的神经网络检测物体更好。但我说，平均来说，目前迁移学习使用频率更高，比多任务学习频率要高，但两者都可以成为你的强力工具。

所以总结一下，多任务学习能让你训练一个神经网络来执行许多任务，这可以给你更高的性能，比单独完成各个任务更高的性能。但要注意，实际上迁移学习比多任务学习使用频率更高。我看到很多任务都是，如果你想解决一个机器学习问题，但你的数据集相对较小，那么迁移学习真的能帮到你，就是如果你找到一个相关问题，其中数据量要大得多，你就能以它为基础训练你的神经网络，然后迁移到这个数据量很少的任务上来。

今天我们学到了很多和迁移学习有关的问题，还有一些迁移学习和多任务学习的应用。但多任务学习，我觉得使用频率比迁移学习要少得多，也许其中一个例外是计算机视觉，物体检测。在那些任务中，人们经常训练一个神经网络同时检测很多不同物体，这比训练单独的神经网络来检测视觉物体要更好。但平均而言，我认为即使迁移学习和多任务学习工作方式类似。实际上，我看到用迁移学习比多任务学习要更多，我觉得这是因为你很难找到那么多相似且数

据量对等的任务可以用单一神经网络训练。再次，在计算机视觉领域，物体检测这个例子是最显著的例外情况。

所以这就是多任务学习，多任务学习和迁移学习都是你的工具包中的重要工具。最后，我想继续讨论端到端深度学习，所以我们来看下一个视频来讨论端到端学习。

10.4 什么是端到端的深度学习？

深度学习中最令人振奋的最新动态之一就是端到端深度学习的兴起，那么端到端学习到底是什么呢？简而言之，以前有一些数据处理系统或者学习系统，它们需要多个阶段的处理。那么端到端深度学习就是忽略所有这些不同的阶段，用单个神经网络代替它。

而端到端深度学习就只需要把训练集拿过来，直接学到了和之间的函数映射，直接绕过了其中很多步骤。对一些学科里的人来说，这点相当难以接受，他们无法接受这样构建 AI 系统，因为有些情况，端到端方法完全取代了旧系统，某些投入了多年研究的中间组件也许已经过时了。

10.5 端到端的深度学习举例？

这张图上是一个研究员做的人脸识别门禁，是百度的林元庆研究员做的。这是一个相机，它会拍下接近门禁的人，如果它认出了那个人，门禁系统就自动打开，让他通过，所以你不需要刷一个 RFID 工卡就能进入这个设施。系统部署在越来越多的中国办公室，希望在其他国家也可以部署更多，你可以接近门禁，如果它认出你的脸，它就直接让你通过，你不需要带 RFID 工卡。

我们再来看几个例子，比如机器翻译。传统上，机器翻译系统也有一个很复杂的流水线，比如英语机翻得到文本，然后做文本分析，基本上要从文本中提取一些特征之类的，经过很多步骤，你最后会将英文文本翻译成法文。因为对于机器翻译来说的确有很多(英文,法文)的数据对，端到端深度学习在机器翻译领域非常好用，那是因为在今天可以收集对的大数据集，就是英文句子和对应的法语翻译。所以在这个例子中，端到端深度学习效果很好。

10.6 端到端的深度学习有什么挑战？

事实证明，端到端深度学习的挑战之一是，你可能需要大量数据才能让系统表现良好，比如，你只有 3000 小时数据去训练你的语音识别系统，那么传统的流水线效果真的很好。但当你拥有非常大的数据集时，比如 10,000 小时数据或者 100,000 小时数据，这样端到端方法突然开始很厉害了。所以当你的数据集较小的时候，传统流水线方法其实效果也不错，通常做得更

好。你需要大数据集才能让端到端方法真正发出耀眼光芒。如果你的数据量适中，那么也可以用中间件方法，你可能输入还是音频，然后绕过特征提取，直接尝试从神经网络输出音位，然后也可以在其他阶段用，所以这是往端到端学习迈出的一小步，但还没有到那里。

10.7 端到端的深度学习优缺点？

假设你正在搭建一个机器学习系统，你要决定是否使用端对端方法，我们来看看端到端深度学习的一些优缺点，这样你就可以根据一些准则，判断你的应用程序是否有希望使用端到端方法。

这里是应用端到端学习的一些好处，首先端到端学习真的只是让数据说话。所以如果你有足够多的数据，那么不管从 x 到 y 最适合的函数映射是什么，如果你训练一个足够大的神经网络，希望这个神经网络能自己搞清楚，而使用纯机器学习方法，直接从到输入去训练的神经网络，可能更能够捕获数据中的任何统计信息，而不是被迫引入人类的成见。

例如，在语音识别领域，早期的识别系统有这个音位概念，就是基本的声音单元，如 cat 单词的“cat”的 Cu-、Ah-和 Tu-，我觉得这个音位是人类语言学家生造出来的，我实际上认为音位其实是语音学家的幻想，用音位描述语言也还算合理。但是不要强迫你的学习算法以音位为单位思考，这点有时没那么明显。如果你让你的学习算法学习它想学习的任意表示方式，而不是强迫你的学习算法使用音位作为表示方式，那么其整体表现可能会更好。

端到端深度学习的第二个好处就是这样，所需手工设计的组件更少，所以这也许能够简化你的设计工作流程，你不需要花太多时间去手工设计功能，手工设计这些中间表示方式。

那么缺点呢？这里有一些缺点，首先，它可能需要大量的数据。要直接学到这个到的映射，你可能需要大量数据。我们在以前的视频里看过一个例子，其中你可以收集大量子任务数据，比如人脸识别，我们可以收集很多数据用来分辨图像中的人脸，当你找到一张脸后，也可以找得到很多人脸识别数据。但是对于整个端到端任务，可能只有更少的数据可用。所以这是端到端学习的输入端，是输出端，所以你需要很多这样的数据，在输入端和输出端都有数据，这样可以训练这些系统。这就是为什么我们称之为端到端学习，因为你直接学习出从系统的一端到系统的另一端。

另一个缺点是，它排除了可能有用的手工设计组件。机器学习研究人员一般都很鄙视手工设计的东西，但如果你没有很多数据，你的学习算法就没办法从很小的训练集数据中获得洞察力。所以手工设计组件在这种情况，可能是把人类知识直接注入算法的途径，这总不是一件坏事。我觉得学习算法有两个主要的知识来源，一个是数据，另一个是你手工设计的任何东西，可能是组件，功能，或者其他东西。所以当你有大量数据时，手工设计的东西就不太重要了，但是当你没有太多的数据时，构造一个精心设计的系统，实际上可以将人类对这个问题的很多认识直接注入到问题里，进入算法里应该挺有帮助的。

所以端到端深度学习的弊端之一是它把可能有用的人工设计的组件排除在外了，精心设计的人工组件可能非常有用，但它们也有可能真的伤害到你的算法表现。例如，强制你的算法以音位为单位思考，也许让算法自己找到更好的表示方法更好。所以这是一把双刃剑，可能有坏处，可能有好处，但往往好处更多，手工设计的组件往往在训练集更小的时候帮助更大。