第十五章　表示学习

在本章中，首先我们会讨论学习表示是什么意思，以及表示的概念如何有助于深度框架的设计。我们探讨学习算法如何在不同任务中共享统计信息，包括使用无监督任务中的信息来完成监督任务。共享表示有助于处理多模式或多领域，或是将已学到的知识迁移到样本很少或没有、但任务表示依然存在的任务上。最后，我们回过头探讨表示学习成功的原因，从分布式表示 (Hinton *et al.*, 1986) 和深度表示的理论优势，最后会讲到数据生成过程潜在假设的更一般概念，特别是观测数据的基本成因。

很多信息处理任务可能非常容易，也可能非常困难，这取决于信息是如何表示的。这是一个广泛适用于日常生活、计算机科学及机器学习的基本原则。例如，对于人而言，可以直接使用长除法计算 210 除以 6。但如果使用罗马数字表示，这个问题就没那么直接了。大部分现代人在使用罗马数字计算 CCX 除以 VI 时，都会将其转化成阿拉伯数字，从而使用位值系统的长除法。更具体地，我们可以使用合适或不合适的表示来量化不同操作的渐近运行时间。例如，插入一个数字到有序表中的正确位置，如果该数列表示为链表，那么所需时间是 $O(n)$；如果该列表表示为红黑树，那么只需要 $O(\log n)$ 的时间。

在机器学习中，到底是什么因素决定了一种表示比另一种表示更好呢？一般而言，一个好的表示可以使后续的学习任务更容易。选择什么表示通常取决于后续的学习任务。

我们可以将监督学习训练的前馈网络视为表示学习的一种形式。具体地，网络的最后一层通常是线性分类器，如 softmax 回归分类器。网络的其余部分学习出该分类器的表示。监督学习训练模型，一般会使得模型的各个隐藏层（特别是接近顶层的隐藏层）的表示能够更加容易地完成训练任务。例如，输入特征线性不可分的类别可能在最后一个隐藏层变成线性可分离的。原则上，最后一层可以是另一种模

型，如最近邻分类器 (Salakhutdinov and Hinton, 2007a)。倒数第二层的特征应该根据最后一层的类型学习不同的性质。

前馈网络的监督训练并没有给学成的中间特征明确强加任何条件。其他的表示学习算法往往会以某种特定的方式明确设计表示。例如，我们想要学习一种使得密度估计更容易的表示。具有更多独立性的分布会更容易建模，因此，我们可以设计鼓励表示向量 $\boldsymbol{h}$ 中元素之间相互独立的目标函数。就像监督网络，无监督深度学习算法有一个主要的训练目标，但也额外地学习出了表示。不论该表示是如何得到的，它都可以用于其他任务。或者，多个任务（有些是监督的，有些是无监督的）可以通过共享的内部表示一起学习。

大多数表示学习算法都会在尽可能多地保留与输入相关的信息和追求良好的性质（如独立性）之间作出权衡。

表示学习特别有趣，因为它提供了进行无监督学习和半监督学习的一种方法。我们通常会有巨量的未标注训练数据和相对较少的标注训练数据。在非常有限的标注数据集上监督学习通常会导致严重的过拟合。半监督学习通过进一步学习未标注数据，来解决过拟合的问题。具体地，我们可以从未标注数据上学习出很好的表示，然后用这些表示来解决监督学习问题。

人类和动物能够从非常少的标注样本中学习。我们至今仍不知道这是如何做到的。有许多假说解释人类的卓越学习能力——例如，大脑可能使用了大量的分类器或者贝叶斯推断技术的集成。一种流行的假说是，大脑能够利用无监督学习和半监督学习。利用未标注数据有多种方式。在本章中，我们主要使用的假说是未标注数据可以学习出良好的表示。

15.1 贪心逐层无监督预训练

无监督学习在深度神经网络的复兴上起到了关键的、历史性的作用，它使研究者首次可以训练不含诸如卷积或者循环这类特殊结构的深度监督网络。我们将这一过程称为 **无监督预训练**（unsupervised pretraining），或者更精确地，**贪心逐层无监督预训练**（greedy layer-wise unsupervised pretraining）。此过程是一个任务（无监督学习，尝试获取输入分布的形状）的表示如何有助于另一个任务（具有相同输入域的监督学习）的典型示例。

贪心逐层无监督预训练依赖于单层表示学习算法，例如 RBM、单层自编码器、

稀疏编码模型或其他学习潜在表示的模型。每一层使用无监督学习预训练，将前一层的输出作为输入，输出数据的新的表示。这个新的表示的分布（或者是和其他变量比如要预测类别的关系）有可能是更简单的。如算法 15.1 所示的正式表述。

算法 15.1 贪心逐层无监督预训练的协定

给定如下：无监督特征学习算法 $\mathcal{L}$，$\mathcal{L}$ 使用训练集样本并返回编码器或特征函数 f。原始输入数据是 $\boldsymbol{X}$，每行一个样本，并且 $f^{(1)}(\boldsymbol{X})$ 是第一阶段编码器关于 $\boldsymbol{X}$ 的输出。在执行精调的情况下，我们使用学习者 $\mathcal{T}$，并使用初始函数 f，输入样本 $\boldsymbol{X}$（以及在监督精调情况下关联的目标 $\boldsymbol{Y}$），并返回细调好函数。阶段数为 m。

$f \leftarrow$ 恒等函数
$\tilde{\boldsymbol{X}} = \boldsymbol{X}$
for $k = 1, \ldots, m$ **do**
  $f^{(k)} = \mathcal{L}(\tilde{\boldsymbol{X}})$
  $f \leftarrow f^{(k)} \circ f$
  $\tilde{\boldsymbol{X}} \leftarrow f^{(k)}(\tilde{\boldsymbol{X}})$
end for
if *fine-tuning* ***then***
  $f \leftarrow \mathcal{T}(f, \boldsymbol{X}, \boldsymbol{Y})$
end if
Return f

基于无监督标准的贪心逐层训练过程，早已被用来规避监督问题中深度神经网络难以联合训练多层的问题。这种方法至少可以追溯神经认知机 (Fukushima, 1975)。深度学习的复兴始于 2006 年，源于发现这种贪心学习过程能够为多层联合训练过程找到一个好的初始值，甚至可以成功训练全连接的结构 (Hinton *et al.*, 2006b; Hinton and Salakhutdinov, 2006; Hinton, 2006; Bengio *et al.*, 2007d; Ranzato *et al.*, 2007a)。在此发现之前，只有深度卷积网络或深度循环网络这类特殊结构的深度网络被认为是有可能训练的。现在我们知道训练具有全连接的深度结构时，不再需要使用贪心逐层无监督预训练，但无监督预训练是第一个成功的方法。

贪心逐层无监督预训练被称为**贪心**（greedy）的，是因为它是一个**贪心算法**（greedy algorithm），这意味着它独立地优化解决方案的每一个部分，每一步解决一个部分，而不是联合优化所有部分。它被称为**逐层的**（layer-wise），是因为这些独立的解决方案是网络层。具体地，贪心逐层无监督预训练每次处理一层网络，训练第 k

层时保持前面的网络层不变。特别地，低层网络（最先训练的）不会在引入高层网络后进行调整。它被称为 **无监督**（unsupervised）的，是因为每一层用无监督表示学习算法训练。然而，它也被称为 **预训练**（pretraining），是因为它只是在联合训练算法 **精调**（fine-tune）所有层之前的第一步。在监督学习任务中，它可以被看作是正则化项（在一些实验中，预训练不能降低训练误差，但能降低测试误差）和参数初始化的一种形式。

通常而言，“预训练” 不仅单指预训练阶段，也指结合预训练和监督学习的两阶段学习过程。监督学习阶段可能会使用预训练阶段得到的顶层特征训练一个简单分类器，或者可能会对预训练阶段得到的整个网络进行监督精调。不管采用什么类型的监督学习算法和模型，在大多数情况下，整个训练过程几乎是相同的。虽然无监督学习算法的选择将明显影响到细节，但是大多数无监督预训练应用都遵循这一基本方法。

贪心逐层无监督预训练也能用作其他无监督学习算法的初始化，比如深度自编码器 (Hinton and Salakhutdinov, 2006) 和具有很多潜变量层的概率模型。这些模型包括深度信念网络 (Hinton *et al.*, 2006b) 和深度玻尔兹曼机 (Salakhutdinov and Hinton, 2009a)。这些深度生成模型会在第二十章中讨论。

正如第 8.7.4 节所探讨的，我们也可以进行贪心逐层监督预训练。这是建立在训练浅层模型比深度模型更容易的前提下，而该前提似乎在一些情况下已被证实 (Erhan *et al.*, 2010)。

15.1.1 何时以及为何无监督预训练有效?

在很多分类任务中，贪心逐层无监督预训练能够在测试误差上获得重大提升。这一观察结果始于 2006 年对深度神经网络的重新关注 (Hinton *et al.*, 2006b; Bengio *et al.*, 2007d; Ranzato *et al.*, 2007a)。然而，在很多其他问题上，无监督预训练不能带来改善，甚至还会带来明显的负面影响。Ma *et al.* (2015) 研究了预训练对机器学习模型在化学活性预测上的影响。结果发现，平均而言预训练是有轻微负面影响的，但在有些问题上会有显著帮助。由于无监督预训练有时有效，但经常也会带来负面效果，因此很有必要了解它何时有效以及有效的原因，以确定它是否适合用于特定的任务。

首先，要注意的是这个讨论大部分都是针对贪心无监督预训练而言。还有很多其他完全不同的方法使用半监督学习来训练神经网络，比如第 7.13 节介绍的虚拟对抗

训练。我们还可以在训练监督模型的同时训练自编码器或生成模型。这种单阶段方法的例子包括判别 RBM (Larochelle and Bengio, 2008b) 和梯形网络 (Rasmus *et al.*, 2015)，其中整体目标是两项之和（一个使用标签，另一个仅仅使用输入）。

无监督预训练结合了两种不同的想法。第一，它利用了深度神经网络对初始参数的选择，可以对模型有着显著的正则化效果（在较小程度上，可以改进优化）的想法。第二，它利用了更一般的想法——学习输入分布有助于学习从输入到输出的映射。

这两个想法都涉及到机器学习算法中多个未能完全理解的部分之间复杂的相互作用。

第一个想法，即深度神经网络初始参数的选择对其性能具有很强的正则化效果，很少有关于这个想法的理解。在预训练变得流行时，在一个位置初始化模型被认为会使其接近某一个局部极小点，而不是另一个局部极小点。如今，局部极小值不再被认为是神经网络优化中的严重问题。现在我们知道标准的神经网络训练过程通常不会到达任何形式的临界点。仍然可能的是，预训练会初始化模型到一个可能不会到达的位置——例如，某种区域，其中代价函数从一个样本点到另一个样本点变化很大，而小批量只能提供噪声严重的梯度估计，或是某种区域中的 Hessian 矩阵条件数是病态的，梯度下降必须使用非常小的步长。然而，我们很难准确判断监督学习期间预训练参数的哪些部分应该保留。这是现代方法通常同时使用无监督学习和监督学习，而不是依序使用两个学习阶段的原因之一。除了这些复杂的方法可以让监督学习阶段保持无监督学习阶段提取的信息之外，还有一种简单的方法，固定特征提取器的参数，仅仅将监督学习作为顶层学成特征的分类器。

另一个想法有更好的理解，即学习算法可以使用无监督阶段学习的信息，在监督学习的阶段表现得更好。其基本想法是对于无监督任务有用的一些特征对于监督学习任务也可能是有用的。例如，如果我们训练汽车和摩托车图像的生成模型，它需要知道轮子的概念，以及一张图中应该有多少个轮子。如果我们幸运的话，无监督阶段学习的轮子表示会适合于监督学习。然而我们还未能从数学、理论层面上证明，因此并不总是能够预测哪种任务能以这种形式从无监督学习中受益。这种方法的许多方面高度依赖于具体使用的模型。例如，如果我们希望在预训练特征的顶层添加线性分类器，那么（学习到的）特征必须使潜在的类别是线性可分离的。这些性质通常会在无监督学习阶段自然发生，但也并非总是如此。这是另一个监督和无监督学习同时训练更可取的原因——输出层施加的约束很自然地从一开始就包括在内。

从无监督预训练作为学习一个表示的角度来看，我们可以期望无监督预训练在初始表示较差的情况下更有效。一个重要的例子是词嵌入。使用 one-hot 向量表示的词并不具有很多信息，因为任意两个不同的 one-hot 向量之间的距离（平方 L^2 距离都是 2）都是相同的。学成的词嵌入自然会用它们彼此之间的距离来编码词之间的相似性。因此，无监督预训练在处理单词时特别有用。然而在处理图像时是不太有用的，可能是因为图像已经在一个很丰富的向量空间中，其中的距离只能提供低质量的相似性度量。

从无监督预训练作为正则化项的角度来看，我们可以期望无监督预训练在标注样本数量非常小时很有帮助。因为无监督预训练添加的信息来源于未标注数据，所以当未标注样本的数量非常大时，我们也可以期望无监督预训练的效果最好。无监督预训练的大量未标注样本和少量标注样本构成的半监督学习的优势特别明显。在 2011 年，无监督预训练赢得了两个国际迁移学习比赛 (Mesnil *et al.*, 2011; Goodfellow *et al.*, 2011)。在该情景中，目标任务中标注样本的数目很少（每类几个到几十个）。这些效果也出现在被 Paine *et al.* (2014) 严格控制的实验中。

还可能涉及到一些其他的因素。例如，当我们要学习的函数非常复杂时，无监督预训练可能会非常有用。无监督学习不同于权重衰减这样的正则化项，它不偏向于学习一个简单的函数，而是学习对无监督学习任务有用的特征函数。如果真实的潜在函数是复杂的，并且由输入分布的规律塑造，那么无监督学习更适合作为正则化项。

除了这些注意事项外，我们现在分析一些无监督预训练改善性能的成功示例，并解释这种改进发生的已知原因。无监督预训练通常用来改进分类器，并且从减少测试集误差的观点来看是很有意思的。然而，无监督预训练还有助于分类以外的任务，并且可以用于改进优化，而不仅仅只是作为正则化项。例如，它可以提高去噪自编码器的训练和测试重构误差 (Hinton and Salakhutdinov, 2006)。

Erhan *et al.* (2010) 进行了许多实验来解释无监督预训练的几个成功原因。对训练误差和测试误差的改进都可以解释为，无监督预训练将参数引入到了其他方法可能探索不到的区域。神经网络训练是非确定性的，并且每次运行都会收敛到不同的函数。训练可以停止在梯度很小的点；也可以提前终止结束训练，以防过拟合；还可以停止在梯度很大，但由于诸如随机性或 Hessian 矩阵病态条件等问题难以找到合适下降方向的点。经过无监督预训练的神经网络会一致地停止在一片相同的函数空间区域，但未经过预训练的神经网络会一致地停在另一个区域。图 15.1 可视化了这种现象。经过预训练的网络到达的区域是较小的，这表明预训练减少了估计过程的

方差，这进而又可以降低严重过拟合的风险。换言之，无监督预训练将神经网络参数初始化到它们不易逃逸的区域，并且遵循这种初始化的结果更加一致，和没有这种初始化相比，结果很差的可能性更低。

Erhan *et al.* (2010) 也回答了何时预训练效果最好——预训练的网络越深，测试误差的均值和方差下降得越多。值得注意的是，这些实验是在训练非常深层网络的现代方法发明和流行（整流线性单元，Dropout 和批标准化）之前进行的，因此对于无监督预训练与当前方法的结合，我们所知甚少。

一个重要的问题是无监督预训练是如何起到正则化项作用的。一个假设是，预训练鼓励学习算法发现那些与生成观察数据的潜在原因相关的特征。这也是启发除无监督预训练之外许多其他算法的重要思想，将会在第 15.3 节中进一步讨论。

与无监督学习的其他形式相比，无监督预训练的缺点是其使用了两个单独的训练阶段。很多正则化技术都具有一个优点，允许用户通过调整单一超参数的值来控制正则化的强度。无监督预训练没有一种明确的方法来调整无监督阶段正则化的强度。相反，无监督预训练有许多超参数，但其效果只能之后度量，通常难以提前预测。当我们同时执行无监督和监督学习而不使用预训练策略时，会有单个超参数（通常是附加到无监督代价的系数）控制无监督目标正则化监督模型的强度。减少该系数，总是能够可预测地获得较少正则化强度。在无监督预训练的情况下，没有一种灵活调整正则化强度的方式——要么监督模型初始化为预训练的参数，要么不是。

具有两个单独的训练阶段的另一个缺点是每个阶段都具有各自的超参数。第二阶段的性能通常不能在第一阶段期间预测，因此在第一阶段提出超参数和第二阶段根据反馈来更新之间存在较长的延迟。最通用的方法是在监督阶段使用验证集上的误差来挑选预训练阶段的超参数，如 Larochelle *et al.* (2009) 中讨论的。在实际中，有些超参数，如预训练迭代的次数，很方便在预训练阶段设定，通过无监督目标上使用提前终止策略完成。这个策略并不理想，但是在计算上比使用监督目标代价小得多。

如今，大部分算法已经不使用无监督预训练了，除了在自然语言处理领域中单词作为 one-hot 向量的自然表示不能传达相似性信息，并且有非常多的未标注数据集可用。在这种情况下，预训练的优点是可以对一个巨大的未标注集合（例如用包含数十亿单词的语料库）进行预训练，学习良好的表示（通常是单词，但也可以是句子），然后使用该表示或精调它，使其适合于训练集样本大幅减少的监督任务。这种方法由 Collobert and Weston (2008b)、Turian *et al.* (2010) 和 Collobert *et al.* (2011a)

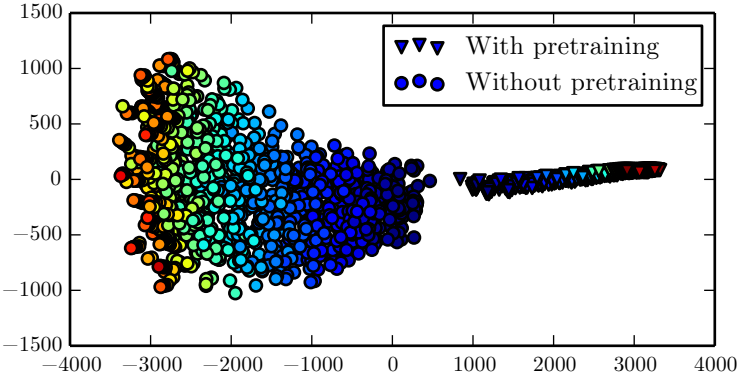

图 15.1: 在函数空间（并非参数空间，避免从参数向量到函数的多对一映射）不同神经网络的学习轨迹的非线性映射的可视化。不同网络采用不同的随机初始化，并且有的使用了无监督预训练，有的没有。每个点对应着训练过程中一个特定时间的神经网络。经Erhan *et al.* (2010) 许可改编此图。函数空间中的坐标是关于每组输入 $\boldsymbol{x}$ 和它的一个输出 $\boldsymbol{y}$ 的无限维向量。Erhan *et al.* (2010) 将很多特定 $\boldsymbol{x}$ 的 $\boldsymbol{y}$ 连接起来，线性投影到高维空间中。然后他们使用 Isomap (Tenenbaum *et al.*, 2000) 进行进一步的非线性投影并投到二维空间。颜色表示时间。所有的网络初始化在上图的中心点附近（对应的函数区域在不多数输入上具有近似均匀分布的类别 y）。随着时间推移，学习将函数向外移动到预测得更好的点。当使用预训练时，训练会一致地收敛到同一个区域；而不使用预训练时，训练会收敛到另一个不重叠的区域。Isomap 试图维持全局相对距离（体积因此也保持不变），因此使用预训练的模型对应的较小区域意味着，基于预训练的估计具有较小的方差。

开创，至今仍在使用。

基于监督学习的深度学习技术，通过 Dropout 或批标准化来正则化，能够在很多任务上达到人类级别的性能，但仅仅是在极大的标注数据集上。在中等大小的数据集（例如 CIFAR-10 和 MNIST，每个类大约有 5,000 个标注样本）上，这些技术的效果比无监督预训练更好。在极小的数据集，例如选择性剪接数据集，贝叶斯方法要优于基于无监督预训练的方法 (Srivastava, 2013)。由于这些原因，无监督预训练已经不如以前流行。然而，无监督预训练仍然是深度学习研究历史上的一个重要里程碑，并将继续影响当代方法。预训练的想法已经推广到 **监督预训练**（supervised pretraining），这将在第 8.7.4 节中讨论，在迁移学习中这是非常常用的方法。迁移学习中的监督预训练流行 (Oquab *et al.*, 2014; Yosinski *et al.*, 2014) 于在 ImageNet 数据集上使用卷积网络预训练。由于这个原因，实践者们公布了这些网络训练出的参数，就像自然语言任务公布预训练的单词向量一样 (Collobert *et al.*, 2011a; Mikolov

et al., 2013a)。

15.2 迁移学习和领域自适应

迁移学习和领域自适应指的是利用一个情景（例如，分布 P_1）中已经学到的内容去改善另一个情景（比如分布 P_2）中的泛化情况。这点概括了上一节提出的想法，即在无监督学习任务和监督学习任务之间转移表示。

在**迁移学习**（transfer learning）中，学习器必须执行两个或更多个不同的任务，但是我们假设能够解释 P_1 变化的许多因素和学习 P_2 需要抓住的变化相关。这通常能够在监督学习中解释，输入是相同的，但是输出不同的性质。例如，我们可能在第一种情景中学习了一组视觉类别，比如猫和狗，然后在第二种情景中学习一组不同的视觉类别，比如蚂蚁和黄蜂。如果第一种情景（从 P_1 采样）中具有非常多的数据，那么这有助于学习到能够使得从 P_2 抽取的非常少样本中快速泛化的表示。许多视觉类别共享一些低级概念，比如边缘、视觉形状、几何变化、光照变化的影响等等。一般而言，当存在对不同情景或任务有用特征时，并且这些特征对应多个情景出现的潜在因素，迁移学习、多任务学习（第 7.7 节）和领域自适应可以使用表示学习来实现。如图 7.2 所示，这是具有共享底层和任务相关上层的学习框架。

然而，有时不同任务之间共享的不是输入的语义，而是输出的语义。例如，语音识别系统需要在输出层产生有效的句子，但是输入附近的较低层可能需要识别相同音素或子音素发音的非常不同的版本（这取决于说话人）。在这样的情况下，共享神经网络的上层（输出附近）和进行任务特定的预处理是有意义的，如图 15.2 所示。

在**领域自适应**（domain adaption）的相关情况下，在每个情景之间任务（和最优的输入到输出的映射）都是相同的，但是输入分布稍有不同。例如，考虑情感分析的任务，如判断一条评论是表达积极的还是消极的情绪。网上的评论有许多类别。在书、视频和音乐等媒体内容上训练的顾客评论情感预测器，被用于分析诸如电视机或智能电话的消费电子产品的评论时，领域自适应情景可能会出现。可以想象，存在一个潜在的函数可以判断任何语句是正面的、中性的还是负面的，但是词汇和风格可能会因领域而有差异，使得跨域的泛化训练变得更加困难。简单的无监督预训练（去噪自编码器）已经能够非常成功地用于领域自适应的情感分析 (Glorot *et al.*, 2011c)。

一个相关的问题是**概念漂移**（concept drift），我们可以将其视为一种迁移学习，

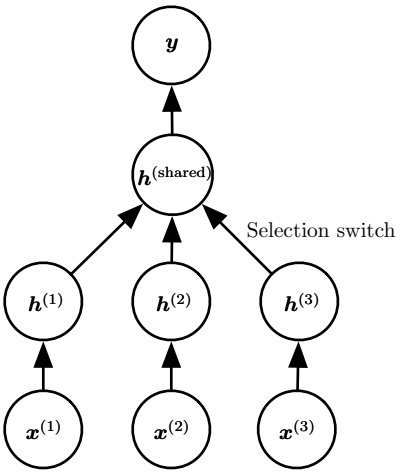

图 15.2: 多任务学习或者迁移学习的架构示例。输出变量 $\mathbf{y}$ 在所有的任务上具有相同的语义；输入变量 $\mathbf{x}$ 在每个任务（或者，比如每个用户）上具有不同的意义（甚至可能具有不同的维度），图上三个任务为 $\mathbf{x}^{(1)}$，$\mathbf{x}^{(2)}$，$\mathbf{x}^{(3)}$。底层结构（决定了选择方向）是面向任务的，上层结构是共享的。底层结构学习将面向特定任务的输入转化为通用特征。

因为数据分布随时间而逐渐变化。概念漂移和迁移学习都可以被视为多任务学习的特定形式。“多任务学习” 这个术语通常指监督学习任务，而更广义的迁移学习的概念也适用于无监督学习和强化学习。

在所有这些情况下，我们的目标是利用第一个情景下的数据，提取那些在第二种情景中学习时或直接进行预测时可能有用的信息。表示学习的核心思想是相同的表示可能在两种情景中都是有用的。两个情景使用相同的表示，使得表示可以受益于两个任务的训练数据。

如前所述，迁移学习中无监督深度学习已经在一些机器学习比赛中取得了成功 (Mesnil *et al.*, 2011; Goodfellow *et al.*, 2011)。这些比赛中的某一个实验配置如下。首先每个参与者获得一个第一种情景（来自分布 P_1）的数据集，其中含有一些类别的样本。参与者必须使用这个来学习一个良好的特征空间（将原始输入映射到某种表示），使得当我们将这个学成变换用于来自迁移情景（分布 P_2）的输入时，线性分类器可以在很少标注样本上训练、并泛化得很好。这个比赛中最引人注目的结果之一是，学习表示的网络架构越深（在第一个情景P_1 中的数据使用纯无监督方式学习），在第二个情景（迁移）P_2 的新类别上学习到的曲线就越好。对于深度表示而言，迁移任务只需要少量标注样本就能显著地提升泛化性能。

迁移学习的两种极端形式是**一次学习**（one-shot learning）和**零次学习**（zero-shot learning），有时也被称为**零数据学习**（zero-data learning）。只有一个标注样本的迁移任务被称为一次学习；没有标注样本的迁移任务被称为零次学习。

因为第一阶段学习出的表示就可以清楚地分离出潜在的类别，所以一次学习 (Fei-Fei *et al.*, 2006) 是可能的。在迁移学习阶段，仅需要一个标注样本来推断表示空间中聚集在相同点周围许多可能测试样本的标签。这使得在学成的表示空间中，对应于不变性的变化因子已经与其他因子完全分离，在区分某些类别的对象时，我们可以学习到哪些因素具有决定意义。

考虑一个零次学习情景的例子，学习器已经读取了大量文本，然后要解决对象识别的问题。如果文本足够好地描述了对象，那么即使没有看到某对象的图像，也能识别出该对象的类别。例如，已知猫有四条腿和尖尖的耳朵，那么学习器可以在没有见过猫的情况下猜测该图像中是猫。

只有在训练时使用了额外信息，零数据学习 (Larochelle *et al.*, 2008) 和零次学习 (Palatucci *et al.*, 2009; Socher *et al.*, 2013b) 才是有可能的。我们可以认为零数据学习场景包含三个随机变量：传统输入 $\boldsymbol{x}$，传统输出或目标 $\boldsymbol{y}$，以及描述任务的附加随机变量 T。该模型被训练来估计条件分布 $p(\boldsymbol{y} \mid \boldsymbol{x}, T)$，其中 T 是我们希望执行的任务的描述。在我们的例子中，读取猫的文本信息然后识别猫，输出是二元变量 y，$y = 1$ 表示 ‘‘是’’，$y = 0$ 表示 ‘‘不是’’。任务变量 T 表示要回答的问题，例如 ‘‘这个图像中是否有猫？’’ 如果训练集包含和 T 在相同空间的无监督对象样本，我们也许能够推断未知的 T 实例的含义。在我们的例子中，没有提前看到猫的图像而去识别猫，所以拥有一些未标注文本数据包含句子诸如 ‘‘猫有四条腿’’ 或 ‘‘猫有尖耳朵’’，对于学习非常有帮助。

零次学习要求 T 被表示为某种形式的泛化。例如，T 不能仅是指示对象类别的one-hot编码。通过使用每个类别词的词嵌入表示，Socher *et al.* (2013b) 提出了对象类别的分布式表示。

我们还可以在机器翻译中发现一种类似的现象 (Klementiev *et al.*, 2012; Mikolov *et al.*, 2013b; Gouws *et al.*, 2014)：我们已经知道一种语言中的单词，还可以学到单一语言语料库中词与词之间的关系；另一方面，我们已经翻译了一种语言中的单词与另一种语言中的单词相关的句子。即使我们可能没有将语言 X 中的单词 A 翻译成语言 Y 中的单词 B 的标注样本，我们也可以泛化并猜出单词 A 的翻译，这是由于我们已经学习了语言 X 和 Y 单词的分布式表示，并且通过两种语言句子的匹配

对组成的训练样本，产生了关联于两个空间的链接（可能是双向的）。如果联合学习三种成分（两种表示形式和它们之间的关系），那么这种迁移将会非常成功。

零次学习是迁移学习的一种特殊形式。同样的原理可以解释如何能执行**多模态学习**（multimodal learning），学习两种模态的表示，和一种模态中的观察结果 $\boldsymbol{x}$ 与另一种模态中的观察结果 $\boldsymbol{y}$ 组成的对 $(\boldsymbol{x}, \boldsymbol{y})$ 之间的关系（通常是一个联合分布）(Srivastava and Salakhutdinov, 2012)。通过学习所有的三组参数（从 $\boldsymbol{x}$ 到它的表示、从 $\boldsymbol{y}$ 到它的表示，以及两个表示之间的关系），一个表示中的概念被锚定在另一个表示中，反之亦然，从而可以有效地推广到新的对组。这个过程如图 15.3 所示。

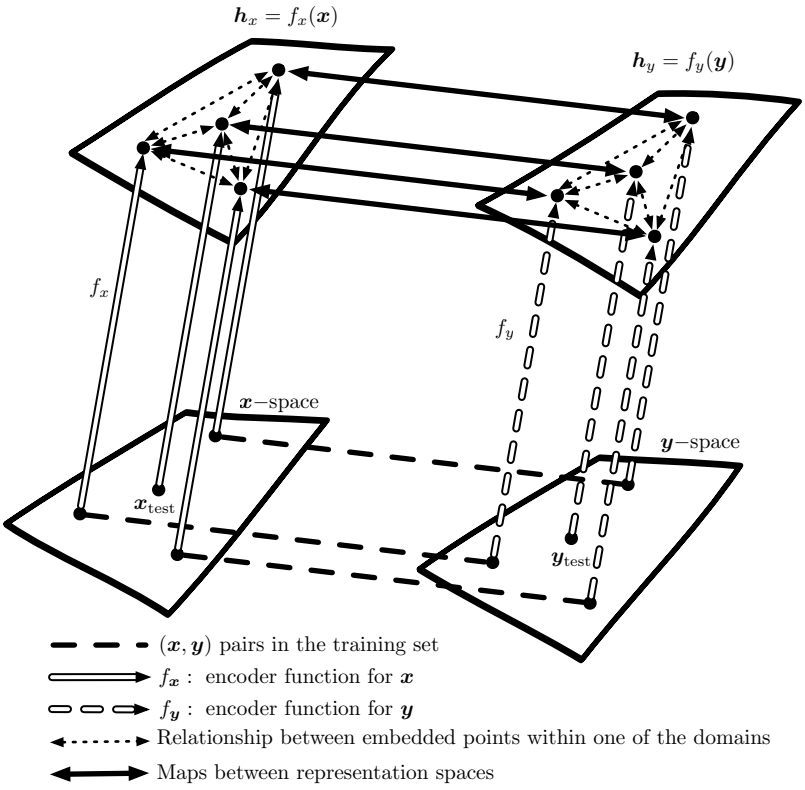

图 15.3: 两个域 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 之间的迁移学习能够进行零次学习。标注或未标注样本 $\boldsymbol{x}$ 可以学习表示函数 $f_{\boldsymbol{x}}$。同样地，样本 $\boldsymbol{y}$ 也可以学习表示函数 $f_{\boldsymbol{y}}$。上图中 $f_{\boldsymbol{x}}$ 和 $f_{\boldsymbol{y}}$ 旁都有一个向上的箭头，不同的箭头表示不同的作用函数。并且箭头的类型表示使用了哪一种函数。$\boldsymbol{h}_{\boldsymbol{x}}$ 空间中的相似性度量表示 $\boldsymbol{x}$ 空间中任意点对之间的距离，这种度量方式比直接度量 $\boldsymbol{x}$ 空间的距离更好。同样地，$\boldsymbol{h}_{\boldsymbol{y}}$ 空间中的相似性度量表示 $\boldsymbol{y}$ 空间中任意点对之间的距离。这两种相似函数都使用带点的双向箭头表示。标注样本（水平虚线）$(\boldsymbol{x}, \boldsymbol{y})$ 能够学习表示 $f_{\boldsymbol{x}}(\boldsymbol{x})$ 和表示 $f_{\boldsymbol{y}}(\boldsymbol{y})$ 之间的单向或双向映射（实双向箭头），以及这些表示之间如何锚定。零数据学习可以通过以下方法实现。像 $\boldsymbol{x}_{\text{test}}$ 可以和单词 $\boldsymbol{y}_{\text{test}}$ 关联起来，即使该单词没有像，仅仅是因为单词表示 $f_{\boldsymbol{y}}(\boldsymbol{y}_{\text{test}})$ 和像表示 $f_{\boldsymbol{x}}(\boldsymbol{x}_{\text{test}})$ 可以通过表示空间的映射彼此关联。这种方法有效的原因是，尽管像和单词没有匹配成队，但是它们各自的特征向量 $f_{\boldsymbol{x}}(\boldsymbol{x}_{\text{test}})$ 和 $f_{\boldsymbol{y}}(\boldsymbol{y}_{\text{test}})$ 互相关联。上图受 Hrant Khachatrian 的建议启发。

15.3 半监督解释因果关系

表示学习的一个重要问题是“什么原因能够使一个表示比另一个表示更好？”一种假设是，理想表示中的特征对应到观测数据的潜在成因，特征空间中不同的特征

或方向对应着不同的原因，从而表示能够区分这些原因。这个假设促使我们去寻找表示 $p(\boldsymbol{x})$ 的更好方法。如果 $\boldsymbol{y}$ 是 $\boldsymbol{x}$ 的重要成因之一，那么这种表示也可能是计算 $p(\boldsymbol{y} \mid \boldsymbol{x})$ 的一种良好表示。从 20 世纪 90 年代以来，这个想法已经指导了大量的深度学习研究工作 (Becker and Hinton, 1992; Hinton and Sejnowski, 1999)。关于半监督学习可以超过纯监督学习的其他论点，请读者参考Chapelle *et al.* (2006) 的第 1.2 节。

在表示学习的其他方法中，我们大多关注易于建模的表示——例如，数据稀疏或是各项之间相互独立的情况。能够清楚地分离出潜在因素的表示可能并不一定易于建模。然而，该假设促使半监督学习使用无监督表示学习的一个更深层原因是，对于很多人工智能任务而言，有两个相随的特点：一旦我们能够获得观察结果基本成因的解释，那么将会很容易分离出个体属性。具体来说，如果表示向量 $\boldsymbol{h}$ 表示观察值 $\boldsymbol{x}$ 的很多潜在因素，并且输出向量 $\boldsymbol{y}$ 是最为重要的原因之一，那么从 $\boldsymbol{h}$ 预测 $\boldsymbol{y}$ 会很容易。

首先，让我们看看 $p(\mathbf{x})$ 的无监督学习无助于学习 $p(\mathbf{y} \mid \mathbf{x})$ 时，半监督学习为何失败。例如，考虑一种情况，$p(\mathbf{x})$ 是均匀分布的，我们希望学习 $f(\boldsymbol{x}) = \mathbb{E}[\mathbf{y} \mid \boldsymbol{x}]$。显然，仅仅观察训练集的值 $\boldsymbol{x}$ 不能给我们关于 $p(\mathbf{y} \mid \mathbf{x})$ 的任何信息。

接下来，让我们看看半监督学习成功的一个简单例子。考虑这样的情况，$\mathbf{x}$ 来自一个混合分布，每个 $\mathbf{y}$ 值具有一个混合分量，如图 15.4 所示。如果混合分量很好地分出来了，那么建模 $p(\mathbf{x})$ 可以精确地指出每个分量的位置，每个类一个标注样本的训练集足以精确学习 $p(\mathbf{y} \mid \mathbf{x})$。但是更一般地，什么能将 $p(\mathbf{y} \mid \mathbf{x})$ 和 $p(\mathbf{x})$ 关联在一起呢?

如果 $\mathbf{y}$ 与 $\mathbf{x}$ 的成因之一非常相关，那么 $p(\mathbf{x})$ 和 $p(\mathbf{y} \mid \mathbf{x})$ 也会紧密关联，试图找到变化潜在因素的无监督表示学习可能像半监督学习一样有用。

假设 $\mathbf{y}$ 是 $\mathbf{x}$ 的成因之一，让 $\mathbf{h}$ 代表所有这些成因。真实的生成过程可以被认为是根据这个有向图模型结构化出来的，其中 $\mathbf{h}$ 是 $\mathbf{x}$ 的父节点：

$$p(\mathbf{h}, \mathbf{x}) = p(\mathbf{x} \mid \mathbf{h}) p(\mathbf{h}). \tag{15.1}$$

因此，数据的边缘概率是

$$p(\boldsymbol{x}) = \mathbb{E}_{\mathbf{h}} p(\boldsymbol{x} \mid \boldsymbol{h}). \tag{15.2}$$

从这个直观的观察中，我们得出结论，$\mathbf{x}$ 最好可能的模型（从广义的观点）是会表示上述 “真实” 结构的，其中 $\boldsymbol{h}$ 作为潜变量解释 $\boldsymbol{x}$ 中可观察的变化。上文讨论的 “理

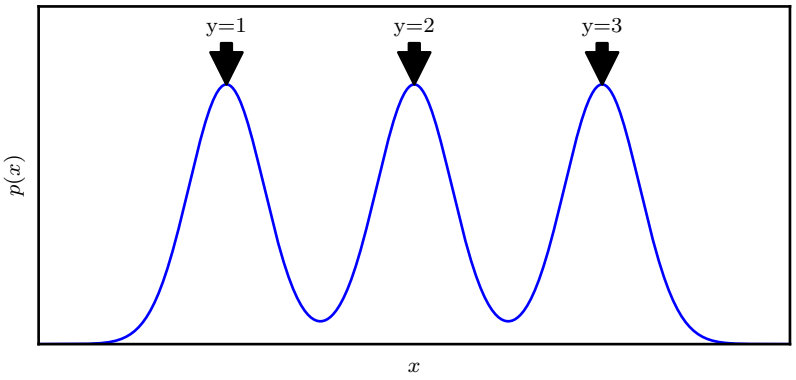

图 15.4: 混合模型。具有三个混合分量的 x 上混合密度示例。混合分量的内在本质是潜在解释因子y。因为混合分量（例如，图像数据中的自然对象类别）在统计学上是显著的，所以仅仅使用未标注样本无监督建模 $p(x)$ 也能揭示解释因子y。

想'' 的表示学习应该能够反映出这些潜在因子。如果 $\mathbf{y}$ 是其中之一（或是紧密关联于其中之一），那么将很容易从这种表示中预测 $\mathbf{y}$。我们会看到给定 $\mathbf{x}$ 下 $\mathbf{y}$ 的条件分布通过贝叶斯规则关联到上式中的分量：

$$p(\mathbf{y} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathbf{y})p(\mathbf{y})}{p(\mathbf{x})}. \tag{15.3}$$

因此边缘概率 $p(\mathbf{x})$ 和条件概率 $p(\mathbf{y} \mid \mathbf{x})$ 密切相关，前者的结构信息应该有助于学习后者。因此，在这些假设情况下，半监督学习应该能提高性能。

关于这个事实的一个重要的研究问题是，大多数观察是由极其大量的潜在成因形成的。假设 $\mathbf{y} = \mathrm{h}_i$，但是无监督学习器并不知道是哪一个 h_i。对于一个无监督学习器暴力求解就是学习一种表示，这种表示能够捕获所有合理的重要生成因子 h_j，并将它们彼此区分开来，因此不管 h_i 是否关联于 $\mathbf{y}$，从 $\mathbf{h}$ 预测 $\mathbf{y}$ 都是容易的。

在实践中，暴力求解是不可行的，因为不可能捕获影响观察的所有或大多数变化因素。例如，在视觉场景中，表示是否应该对背景中的所有最小对象进行编码？根据一个有据可查的心理学现象，人们不会察觉到环境中和他们所在进行的任务并不立刻相关的变化，具体例子可以参考 Simons and Levin (1998)。半监督学习的一个重要研究前沿是确定每种情况下要编码什么。目前，处理大量潜在原因的两个主要策略是，同时使用无监督学习和监督学习信号，从而使得模型捕获最相关的变动因素，或是使用纯无监督学习学习更大规模的表示。

无监督学习的另一个思路是选择一个更好的确定哪些潜在因素最为关键的定义。之前，自编码器和生成模型被训练来优化一个类似于均方误差的固定标准。这些固定标准确定了哪些因素是重要的。例如，图像像素的均方误差隐式地指定，一个潜在因素只有在其显著地改变大量像素的亮度时，才是重要影响因素。如果我们希望解决的问题涉及到小对象之间的相互作用，那么这将有可能遇到问题。如图 15.5 所示，在机器人任务中，自编码器未能学习到编码小乒乓球。同样是这个机器人，它可以成功地与更大的对象进行交互（例如棒球，均方误差在这种情况下很显著）。

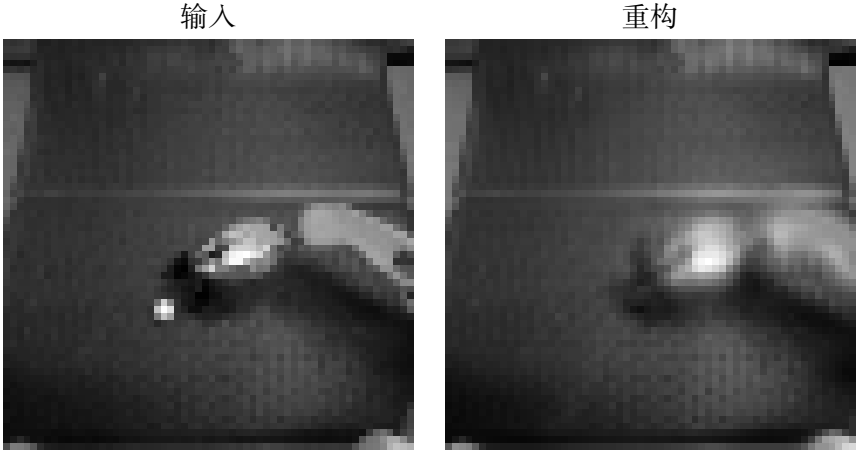

图 15.5: 机器人任务上，基于均方误差训练的自编码器不能重构乒乓球。乒乓球的存在及其所有空间坐标，是生成图像且与机器人任务相关的重要潜在因素。不幸的是，自编码器具有有限的容量，基于均方误差的训练没能将乒乓球作为显著物体识别出来编码。以上图像由 Chelsea Finn 提供。

还有一些其他的显著性的定义。例如，如果一组像素具有高度可识别的模式，那么即使该模式不涉及到极端的亮度或暗度，该模式还是会被认为非常显著。实现这样一种定义显著的方法是使用最近提出的**生成式对抗网络**（generative adversarial network）(Goodfellow *et al.*, 2014c)。在这种方法中，生成模型被训练来愚弄前馈分类器。前馈分类器尝试将来自生成模型的所有样本识别为假的，并将来自训练集的所有样本识别为真的。在这个框架中，前馈网络能够识别出的任何结构化模式都是非常显著的。生成式对抗网络会在第 20.10.4 节中更详细地介绍。为了叙述方便，知道它能学习出如何决定什么是显著的就可以了。Lotter *et al.* (2015) 表明，生成人类头部头像的模型在使用均方误差训练时往往会忽视耳朵，但是对抗式框架学习能够成功地生成耳朵。因为耳朵与周围的皮肤相比不是非常明亮或黑暗，所以根据均方

误差损失它们不是特别突出，但是它们高度可识别的形状和一致的位置意味着前馈网络能够轻易地学习出如何检测它们，从而使得它们在生成式对抗框架下是高度突出的。图 15.6 给了一些样例图片。生成式对抗网络只是确定应该表示哪些因素的一小步。我们期望未来的研究能够发现更好的方式来确定表示哪些因素，并且根据任务来开发表示不同因素的机制。

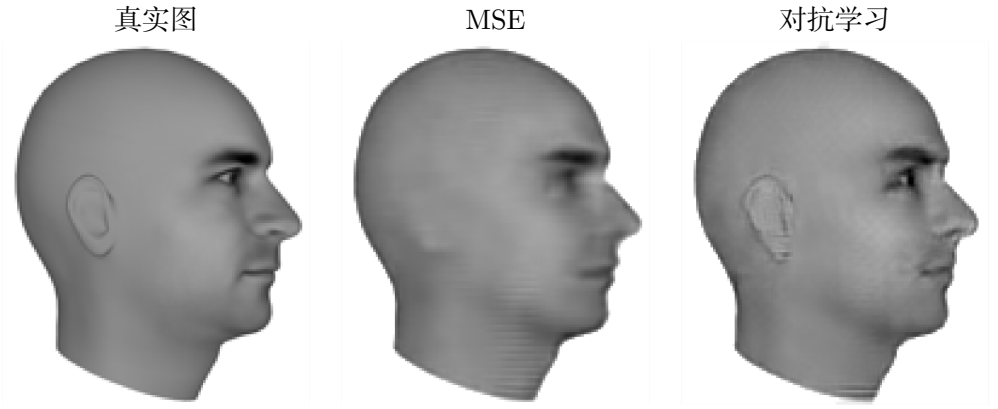

图 15.6: 预测生成网络是一个学习哪些特征显著的例子。在这个例子中，预测生成网络已被训练成在特定视角预测人头的 3D 模型。*(左)* 真实情况。这是一张网络应该生成的正确图片。*(中)* 由具有均方误差的预测生成网络生成的图片。因为与相邻皮肤相比，耳朵不会引起亮度的极大差异，所以它们的显著性不足以让模型学习表示它们。*(右)* 由具有均方误差和对抗损失的模型生成的图片。使用这个学成的代价函数，由于耳朵遵循可预测的模式，因此耳朵是显著重要的。学习哪些原因对于模型而言是足够重要和相关的，是一个重要的活跃研究领域。以上图片由 Lotter *et al.* (2015) 提供。

正如 Schölkopf *et al.* (2012) 指出，学习潜在因素的好处是，如果真实的生成过程中 $\mathbf{x}$ 是结果，$\mathbf{y}$ 是原因，那么建模 $p(\mathbf{x} \mid \mathbf{y})$ 对于 $p(\mathbf{y})$ 的变化是鲁棒的。如果因果关系被逆转，这是不对的，因为根据贝叶斯规则，$p(\mathbf{x} \mid \mathbf{y})$ 将会对 $p(\mathbf{y})$ 的变化十分敏感。很多时候，我们考虑分布的变化（由于不同领域、时间不稳定性或任务性质的变化）时，因果机制是保持不变的（“宇宙定律不变”），而潜在因素的边缘分布是会变化的。因此，通过学习试图恢复成因向量 $\mathbf{h}$ 和 $p(\mathbf{x} \mid \mathbf{h})$ 的生成模型，我们可以期望最后的模型对所有种类的变化有更好的泛化和鲁棒性。

15.4 分布式表示

分布式表示的概念（由很多元素组合的表示，这些元素之间可以设置成可分离的）是表示学习最重要的工具之一。分布式表示非常强大，因为他们能用具有 k 个值的 n 个特征去描述 k^n 个不同的概念。正如我们在本书中看到的，具有多个隐藏单元的神经网络和具有多个潜变量的概率模型都利用了分布式表示的策略。我们现在再介绍一个观察结果。许多深度学习算法基于的假设是，隐藏单元能够学习表示出解释数据的潜在因果因子，就像第 15.3 节中讨论的一样。这种方法在分布式表示上是自然的，因为表示空间中的每个方向都对应着一个不同的潜在配置变量的值。

n 维二元向量是一个分布式表示的示例，有 2^n 种配置，每一种都对应输入空间中的一个不同区域，如图 15.7 所示。这可以与*符号表示*相比较，其中输入关联到单一符号或类别。如果字典中有 n 个符号，那么可以想象有 n 个特征监测器，每个特征探测器监测相关类别的存在。在这种情况下，只有表示空间中 n 个不同配置才有可能在输入空间中刻画 n 个不同的区域，如图 15.8 所示。这样的符号表示也被称为 one-hot 表示，因为它可以表示成相互排斥的 n 维二元向量（其中只有一位是激活的）。符号表示是更广泛的非分布式表示类中的一个具体示例，它可以包含很多条目，但是每个条目没有显著意义的单独控制作用。

以下是基于非分布式表示的学习算法的示例：

- 聚类算法，包含 k-means 算法：每个输入点恰好分配到一个类别。
- k-最近邻算法：给定一个输入，一个或几个模板或原型样本与之关联。在 $k > 1$ 的情况下，每个输入都使用多个值来描述，但是它们不能彼此分开控制，因此这不能算真正的分布式表示。
- 决策树：给定输入时，只有一个叶节点（和从根到该叶节点路径上的点）是被激活的。
- 高斯混合体和专家混合体：模板（聚类中心）或专家关联一个激活的*程度*。和 k-最近邻算法一样，每个输入用多个值表示，但是这些值不能轻易地彼此分开控制。
- 具有高斯核（或其他类似的局部核）的核机器：尽管每个“支持向量”或模板样本的激活程度是连续值，但仍然会出现和高斯混合体相同的问题。

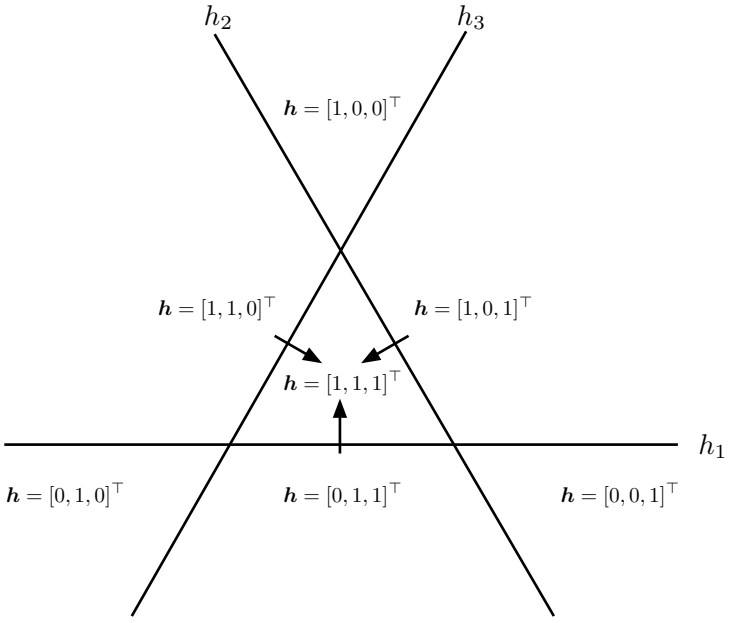

图 15.7: 基于分布式表示的学习算法如何将输入空间分割成多个区域的图示。这个例子具有二元变量 h_1，h_2，h_3。每个特征通过为学成的线性变换设定输出阀值而定义。每个特征将 $\mathbb{R}^2$ 分成两个半平面。令 h_i^+ 表示输入点 $h_i = 1$ 的集合；h_i^- 表示输入点 $h_i = 0$ 的集合。在这个图示中，每条线代表着一个 h_i 的决策边界，对应的箭头指向边界的 h_i^+ 区域。整个表示在这些半平面的每个相交区域都指定一个唯一值。例如，表示值为 $[1, 1, 1]^\top$ 对应着区域 $h_1^+ \cap h_2^+ \cap h_3^+$。可以将以上表示和图 15.8 中的非分布式表示进行比较。在输入维度是 d 的一般情况下，分布式表示通过半空间（而不是半平面）的交叉分割 $\mathbb{R}^d$。具有 n 个特征的分布式表示给 $O(n^d)$ 个不同区域分配唯一的编码，而具有 n 个样本的最近邻算法只能给 n 个不同区域分配唯一的编码。因此，分布式表示能够比非分布式表示多分配指数级的区域。注意并非所有的 $\boldsymbol{h}$ 值都是可取的（这个例子中没有 $\boldsymbol{h} = \mathbf{0}$），在分布式表示上的线性分类器不能向每个相邻区域分配不同的类别标识；甚至深度线性阀值网络的 VC 维只有 $O(w \log w)$（其中 w 是权重数目）(Sontag, 1998)。强表示层和弱分类器层的组合是一个强正则化项。试图学习 ‘‘人’’ 和 ‘‘非人’’ 概念的分类器不需要给表示为 ‘‘戴眼镜的女人’’ 和 ‘‘没有戴眼镜的男人’’ 的输入分配不同的类别。容量限制鼓励每个分类器关注少数几个 h_i，鼓励 $\boldsymbol{h}$ 以线性可分的方式学习表示这些类别。

- 基于 n-gram 的语言或翻译模型：根据后缀的树结构划分上下文集合（符号序列）。例如，一个叶节点可能对应于最后两个单词 w_1 和 w_2。树上的每个叶节点分别估计单独的参数（有些共享也是可能的）。

对于部分非分布式算法而言，有些输出并非是恒定的，而是在相邻区域之间内

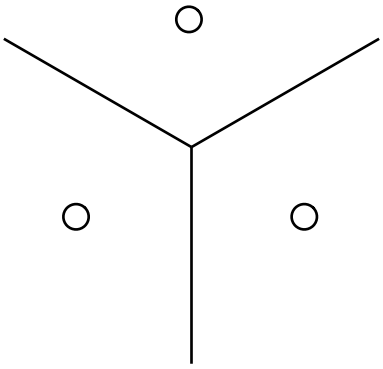

图 15.8: 最近邻算法如何将输入空间分成不同区域的图示。最近邻算法是一个基于非分布式表示的学习算法的示例。不同的非分布式算法可以具有不同的几何形状，但是它们通常将输入空间分成区域，每个区域具有不同的参数。非分布式方法的优点是，给定足够的参数，它能够拟合一个训练集，而不需要复杂的优化算法。因为它直接为每个区域独立地设置不同的参数。缺点是，非分布式表示的模型只能通过平滑先验来局部地泛化，因此学习波峰波谷多于样本的复杂函数时，该方法是不可行的。和分布式表示的对比，可以参照图 15.7 。

插。参数（或样本）的数量和它们能够定义区域的数量之间仍保持线性关系。

将分布式表示和符号表示区分开来的一个重要概念是，由不同概念之间的共享属性而产生的泛化。作为纯符号，“猫” 和 “狗” 之间的距离和任意其他两种符号的距离一样。然而，如果将它们与有意义的分布式表示相关联，那么关于猫的很多特点可以推广到狗，反之亦然。例如，我们的分布式表示可能会包含诸如 “具有皮毛” 或 “腿的数目” 这类在 “猫” 和 “狗” 的嵌入上具有相同值的项。正如第 12.4.2 节所讨论的，作用于单词分布式表示的神经语言模型比其他直接对单词 one-hot 表示进行操作的模型泛化得更好。分布式表示具有丰富的相似性空间，语义上相近的概念（或输入）在距离上接近，这是纯粹的符号表示所缺少的特点。

在学习算法中使用分布式表示何时以及为什么具有统计优势？当一个明显复杂的结构可以用较少参数紧致地表示时，分布式表示具有统计上的优点。一些传统的非分布式学习算法仅仅在平滑假设的情况下能够泛化，也就是说如果 $u \approx v$，那么学习到的目标函数 f 通常具有 $f(u) \approx f(v)$ 的性质。有许多方法来形式化这样一个假设，但其结果是如果我们有一个样本 (x, y)，并且我们知道 $f(x) \approx y$，那么我们可以选取一个估计 $\hat{f}$ 近似地满足这些限制，并且当我们移动到附近的输入 $x + \epsilon$ 时，$\hat{f}$ 尽可能少地发生改变。显然这个假设是非常有用的，但是它会遭受维数灾难：学习出一个

能够在很多不同区域上增加或减少很多次的目标函数 [1]，我们可能需要至少和可区分区域数量一样多的样本。我们可以将每一个区域视为一个类别或符号：通过让每个符号（或区域）具有单独的自由度，我们可以学习出从符号映射到值的任意解码器。然而，这不能推广到新区域的新符号上。

如果我们幸运的话，除了平滑之外，目标函数可能还有一些其他规律。例如，具有最大池化的卷积网络可以在不考虑对象在图像中位置（即使对象的空间变换不对应输入空间的平滑变换）的情况下识别出对象。

让我们检查分布式表示学习算法的一个特殊情况，它通过对输入的线性函数进行阀值处理来提取二元特征。该表示中的每个二元特征将 $\mathbb{R}^d$ 分成一对半空间，如图 15.7 所示。n 个相应半空间的指数级数量的交集确定了该分布式表示学习器能够区分多少区域。空间 $\mathbb{R}^d$ 中的 n 个超平面的排列组合能够生成多少区间？通过应用关于超平面交集的一般结果 (Zaslavsky, 1975)，我们发现 (Pascanu *et al.*, 2014b) 这个二元特征表示能够区分的空间数量是

$$\sum_{j=0}^{d} \binom{n}{j} = O(n^d). \tag{15.4}$$

因此，我们会发现关于输入大小呈指数级增长，关于隐藏单元的数量呈多项式级增长。

这提供了分布式表示泛化能力的一种几何解释：$O(nd)$ 个参数（空间 $\mathbb{R}^d$ 中的 n 个线性阀值特征）能够明确表示输入空间中 $O(n^d)$ 个不同区域。如果我们没有对数据做任何假设，并且每个区域使用唯一的符号来表示，每个符号使用单独的参数去识别 $\mathbb{R}^d$ 中的对应区域，那么指定 $O(n^d)$ 个区域需要 $O(n^d)$ 个样本。更一般地，分布式表示的优势还可以体现在我们对分布式表示中的每个特征使用非线性的、可能连续的特征提取器，而不是线性阀值单元的情况。在这种情况下，如果具有 k 个参数的参数变换可以学习输入空间中的 r 个区域（$k \ll r$），并且如果学习这样的表示有助于关注的任务，那么这种方式会比非分布式情景（我们需要 $O(r)$ 个样本来获得相同的特征，将输入空间相关联地划分成 r 个区域。）泛化得更好。使用较少的参数来表示模型意味着我们只需拟合较少的参数，因此只需要更少的训练样本去获得良好的泛化。

另一个解释基于分布式表示的模型泛化能力更好的说法是，尽管能够明确地编

[1] 一般来说，我们可能会想要学习一个函数，这个函数在指数级数量区域的表现都是不同的：在 d-维空间中，为了区分每一维，至少有两个不同的值。我们想要函数 f 区分这 2^d 个不同的区域，需要 $O(2^d)$ 量级的训练样本

码这么多不同的区域，但它们的容量仍然是很有限的。例如，线性阀值单元神经网络的 VC 维仅为 $O(w \log w)$，其中 w 是权重的数目 (Sontag, 1998)。这种限制出现的原因是，虽然我们可以为表示空间分配非常多的唯一码，但是我们不能完全使用所有的码空间，也不能使用线性分类器学习出从表示空间 $\boldsymbol{h}$ 到输出 $\boldsymbol{y}$ 的任意函数映射。因此使用与线性分类器相结合的分布式表示传达了一种先验信念，待识别的类在 $\boldsymbol{h}$ 代表的潜在因果因子的函数下是线性可分的。我们通常想要学习类别，例如所有绿色对象的图像集合，或是所有汽车图像集合，但不会是需要非线性 XOR 逻辑的类别。例如，我们通常不会将数据划分成所有红色汽车和绿色卡车作为一个集合，所有绿色汽车和红色卡车作为另一个集合。

到目前为止讨论的想法都是抽象的，但是它们可以通过实验验证。Zhou *et al.* (2015) 发现，在 ImageNet 和 Places 基准数据集上训练的深度卷积网络中的隐藏单元学成的特征通常是可以解释的，对应人类自然分配的标签。在实践中，隐藏单元并不能总是学习出具有简单语言学名称的事物，但有趣的是，这些事物会在那些最好的计算机视觉深度网络的顶层附近出现。这些特征的共同之处在于，我们可以设想学习其中的每个特征不需要知道所有其他特征的所有配置。Radford *et al.* (2015) 发现生成模型可以学习人脸图像的表示，在表示空间中的不同方向捕获不同的潜在变差因素。图 15.9 展示表示空间中的一个方向对应着该人是男性还是女性，而另一个方向对应着该人是否戴着眼镜。这些特征都是自动发现的，而非先验固定的。我们没有必要为隐藏单元分类器提供标签：只要该任务需要这样的特征，梯度下降就能在感兴趣的目标函数上自然地学习出语义上有趣的特征。我们可以学习出男性和女性之间的区别，或者是眼镜的存在与否，而不必通过涵盖所有这些值组合的样本来表征其他 $n-1$ 个特征的所有配置。这种形式的统计可分离性质能够泛化到训练期间从未见过的新特征上。

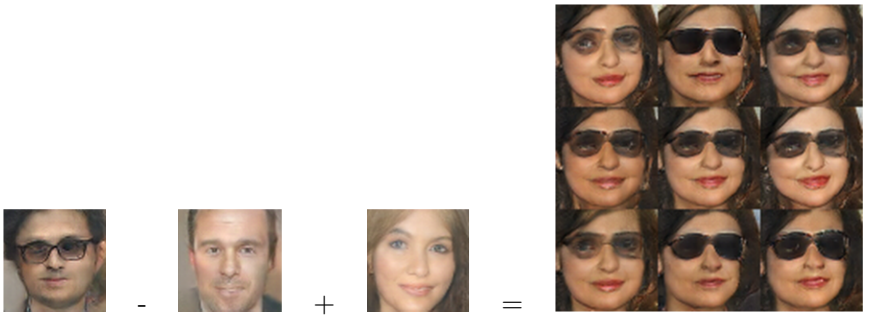

图 15.9: 生成模型学到了分布式表示，能够从戴眼镜的概念中区分性别的概念。如果我们从一个戴眼镜的男人的概念表示向量开始，然后减去一个没戴眼镜的男人的概念表示向量，最后加上一个没戴眼镜的女人的概念表示向量，那么我们会得到一个戴眼镜的女人的概念表示向量。生成模型将所有这些表示向量正确地解码为可被识别为正确类别的图像。图片转载许可自 Radford *et al.* (2015)。

15.5 得益于深度的指数增益

我们已经在第 6.4.1 节中看到，多层感知机是万能近似器，相比于浅层网络，一些函数能够用指数级小的深度网络表示。缩小模型规模能够提高统计效率。在本节中，我们描述如何将类似结果更一般地应用于其他具有分布式隐藏表示的模型。

在第 15.4 节中，我们看到了一个生成模型的示例，能够学习人脸图像的潜在解释因子，包括性别以及是否佩戴眼镜。完成这个任务的生成模型是基于一个深度神经网络的。浅层网络例如线性网络不能学习出这些抽象解释因子和图像像素之间的复杂关系。在这个任务和其他 AI 任务中，这些因子几乎彼此独立地被抽取，但仍然对应到有意义输入的因素，很有可能是高度抽象的，并且和输入呈高度非线性的关系。我们认为这需要深度分布式表示，需要许多非线性组合来获得较高级的特征（被视为输入的函数）或因子（被视为生成原因）。

在许多不同情景中已经证明，非线性和重用特征层次结构的组合来组织计算，可以使分布式表示获得指数级加速之外，还可以获得统计效率的指数级提升。许多种类的只有一个隐藏层的网络（例如，具有饱和非线性，布尔门，和/积，或 RBF 单元的网络）都可以被视为万能近似器。在给定足够多隐藏单元的情况下，这个模型族是一个万能近似器，可以在任意非零允错级别近似一大类函数（包括所有连续函数）。然而，隐藏单元所需的数量可能会非常大。关于深层架构表达能力的理论结果表明，有些函数族可以高效地通过深度 k 层的网络架构表示，但是深度不够（深度

为 2 或 $k-1$）时会需要指数级（相对于输入大小而言）的隐藏单元。

在第 6.4.1 节中，我们看到确定性前馈网络是函数的万能近似器。许多具有单个隐藏层（潜变量）的结构化概率模型（包括受限玻尔兹曼机，深度信念网络）是概率分布的万能近似器 (Le Roux and Bengio, 2008, 2010; Montúfar and Ay, 2011; Montúfar, 2014; Krause *et al.*, 2013)。

在第 6.4.1 节中，我们看到足够深的前馈网络会比深度不够的网络具有指数级优势。这样的结果也能从诸如概率模型的其他模型中获得。**和-积网络**（sum-product network, SPN）(Poon and Domingos, 2011) 是这样的一种概率模型。这些模型使用多项式回路来计算一组随机变量的概率分布。Delalleau and Bengio (2011) 表明存在一种概率分布，对 SPN 的最小深度有要求，以避免模型规模呈指数级增长。后来，Martens and Medabalimi (2014) 表明，任意两个有限深度的 SPN 之间都会存在显著差异，并且一些使 SPN 易于处理的约束可能会限制其表示能力。

另一个有趣的进展是，一系列和卷积网络相关的深度回路族表达能力的理论结果，即使让浅度回路只去近似深度回路计算的函数，也能突出反映深度回路的指数级优势 (Cohen *et al.*, 2015)。相比之下，以前的理论工作只研究了浅度回路必须精确复制特定函数的情况。

15.6 提供发现潜在原因的线索

我们回到最初的问题之一来结束本章：什么原因能够使一个表示比另一个表示更好？首先在第 15.3 节中介绍的一个答案是，一个理想的表示能够区分生成数据变化的潜在因果因子，特别是那些与我们的应用相关的因素。表示学习的大多数策略都会引入一些有助于学习潜在变差因素的线索。这些线索可以帮助学习器将这些观察到的因素与其他因素分开。监督学习提供了非常强的线索：每个观察向量 $\boldsymbol{x}$ 的标签 $\boldsymbol{y}$，它通常直接指定了至少一个变差因素。更一般地，为了利用丰富的未标注数据，表示学习会使用关于潜在因素的其他不太直接的提示。这些提示包含一些我们（学习算法的设计者）为了引导学习器而强加的隐式先验信息。诸如没有免费午餐定理的这些结果表明，正则化策略对于获得良好泛化是很有必要的。当不可能找到一个普遍良好的正则化策略时，深度学习的一个目标是找到一套相当通用的正则化策略，使其能够适用于各种各样的 AI 任务（类似于人和动物能够解决的任务）。

在此，我们提供了一些通用正则化策略的列表。该列表显然是不详尽的，但是

给出了一些学习算法是如何发现对应潜在因素的特征的具体示例。该列表在 Bengio *et al.* (2013d) 的第 3.1 节中提出，这里进行了部分拓展。

- *平滑*：假设对于单位 $\boldsymbol{d}$ 和小量 ϵ 有 $f(\boldsymbol{x}+\epsilon\boldsymbol{d}) \approx f(\boldsymbol{x})$。这个假设允许学习器从训练样本泛化到输入空间中附近的点。许多机器学习算法都利用了这个想法，但它不能克服维数灾难难题。

- *线性*：很多学习算法假定一些变量之间的关系是线性的。这使得算法能够预测远离观测数据的点，但有时可能会导致一些极端的预测。大多数简单的学习算法不会做平滑假设，而会做线性假设。这些假设实际上是不同的，具有很大权重的线性函数在高维空间中可能不是非常平滑的。参看 Goodfellow *et al.* (2014b) 了解关于线性假设局限性的进一步讨论。

- *多个解释因子*：许多表示学习算法受以下假设的启发，数据是由多个潜在解释因子生成的，并且给定每一个因子的状态，大多数任务都能轻易解决。第 15.3 节描述了这种观点如何通过表示学习来启发半监督学习的。学习 $p(\boldsymbol{x})$ 的结构要求学习出一些对建模 $p(\boldsymbol{y} \mid \boldsymbol{x})$ 同样有用的特征，因为它们都涉及到相同的潜在解释因子。第 15.4 节介绍了这种观点如何启发分布式表示的使用，表示空间中分离的方向对应着分离的变差因素。

- *因果因子*：该模型认为学成表示所描述的变差因素是观察数据 $\boldsymbol{x}$ 的成因，而并非反过来。正如第 15.3 节中讨论的，这对于半监督学习是有利的，当潜在成因上的分布发生改变，或者我们应用模型到一个新的任务上时，学成的模型都会更加鲁棒。

- *深度，或者解释因子的层次组织*：高级抽象概念能够通过将简单概念层次化来定义。从另一个角度来看，深度架构表达了我们认为任务应该由多个程序步骤完成的观念，其中每一个步骤回溯到先前步骤处理之后的输出。

- *任务间共享因素*：当多个对应到不同变量 y_i 的任务共享相同的输入 $\mathbf{x}$ 时，或者当每个任务关联到全局输入 $\mathbf{x}$ 的子集或者函数 $f^{(i)}(\mathbf{x})$ 时，我们会假设每个变量 y_i 关联到来自相关因素 $\mathbf{h}$ 公共池的不同子集。因为这些子集有重叠，所以通过共享的中间表示 $P(\mathbf{h} \mid \mathbf{x})$ 来学习所有的 $P(\mathrm{y}_i \mid \mathbf{x})$ 能够使任务间共享统计强度。

- *流形*：概率质量集中，并且集中区域是局部连通的，且占据很小的体积。在连续情况下，这些区域可以用比数据所在原始空间低很多维的低维流形来近似。