CSLT

现代机器学习概论

2016 年 9 月 19 日

Springer

Contents

Chapter 1
深度学习

1.1 计算图

计算图是把复合函数表达成节点互相连接的网络，其中每个节点代表一个操作或者函数。在形式上，它和前向神经网络比较相似，不过它们之间有本质的区别。当把一个复合函数转化为计算图，我们可以很容易辨识复合函数的嵌套结构并且可以按照基本的求导规则对任意的节点进行求导。

1.1.1 计算图的一个实例

下面是展示一个简单的复合函数转化为计算图：

设$f(x) = e^{sin(x^2)}$，我们能把这个复合函数分解为三个简单函数，$f(x) = e^x$，$g(x) = sin(x)$和$h(x) = x^2$，然后可以把这个复合函数看成之这三个函数的嵌套函数$f(g(h(x))) = e^{g(h(x))}$。

现在把$f(x)$用计算图的方式展示出来，如下图1.1所示。

图 1.1 复合函数可视化为计算图

正如上图所示，这个计算图很清晰地显示f(x)嵌套结构。每条有向边的弧头为下一个函数，弧尾为输入。现在就可以在这个计算图上可视化链式规则，我们只需在每个节点对其输入求导就可以实现，如下图所示1.2.

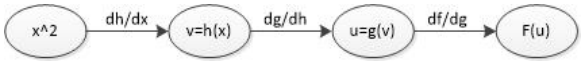

图 1.2 在计算图上实现链式求导

现在我们可以计算任何的计算图上存在的嵌套函数的导数通过所求嵌套函数路径上的导数相乘。比如，我们现在想得到df/dx，只要把从f到x路径上的导数相乘就可以得到。这符合求导的链式规则，如下所示：

$$\frac{df}{dx} = \frac{df}{dg} * \frac{dg}{dh} * \frac{dh}{dx} \tag{1.1}$$

1.1.2 基于神经网络的计算图

一个神经网络就可以看做一个巨多层嵌套的复合函数。比如，前向神经网络的每一层可以看做一个简单函数，其输入为权重W和上一层的输出。这就意味着我们可以为每一个神经网络建立一个计算图并通过BP算法计算梯度。图**??**是一个典型的前向神经网络基本结构图，图**??**是其对应的计算图。

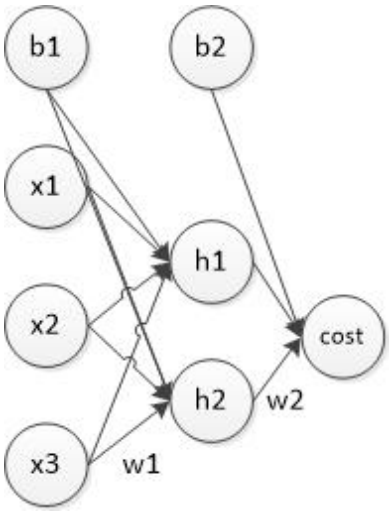

图 1.3 一个典型神经网络的结构图

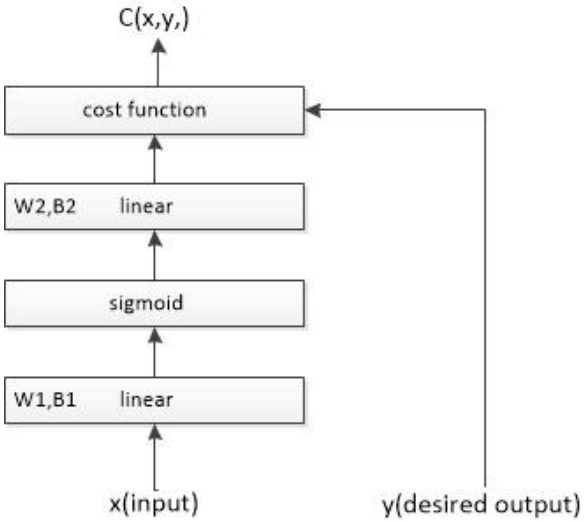

图 1.4 计算图表示一个多层神经网络

图**??**是在基于神经网络的计算图上对其参数进行求导的过程，公式是具体的求导步骤。

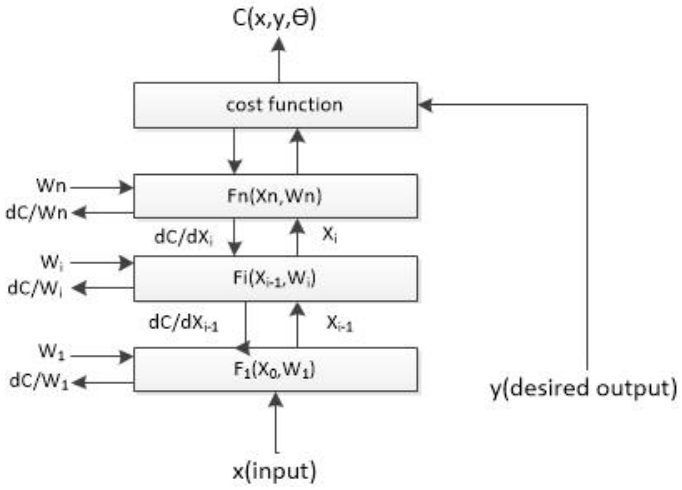

图 1.5 计算图表示一个多层神经网络

绝大多数神经网络计算平台实现了计算图，其中节点代表各种函数，比如linear、relu、cost等等，有向边代表数据的流向。在这些计算平台上，我们

可以构建计算图实现任何的神经网络结构，如下图**??**所示，不过计算图必须是有向无环图或者有向无环图（能在时间维度上展开）；我们可以在节点设置任意的函数，只要这个函数对于参数和非终端输入可微分。这些计算平台提供自动微分，极大地提高我们的工作效率，使我们的注意力集中在神经网络结构上。

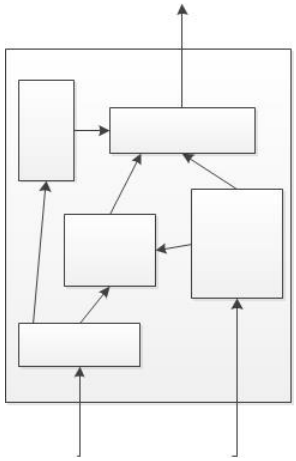

图 1.6 计算图可以表示任意一种神经网络

1.2 分布式计算

在神经网络中，分布式计算在硬件资源上需要更多的GPU；在软件设计方面主要涉及到基于模型和基于数据的并行计算，同步和异步的随机梯度下降。基于模型的并行计算是每个可计算资源中使用相同的数据，然后运行切分后的模型；基于数据的并行计算是每个可计算资源中使用相同的模型，然后运行不同的数据。同步的随机梯度下降是中心节点同步所有可计算资源的参数更新，而异步的随机梯度下降是每个可计算资源负责自己的参数更新并在最后阶段向中心节点传回最终的值。

1.3 卷积神经网络（CNN）

卷积神经网络是具有为处理网格数据而设计的特殊结构的神经网络，它的结构是受到生物视觉感知神经系统提启发而提出的。它的权重共享网络结构类似于生物神经网络，降低了网络模型的复杂度，减少了权值得数量。该优点在网络的输入是多维图像时表现的更为明显，使图像可以直接作为网络的输入，避免了传统识别算法中复杂的特征提取和数据重建过程。这种网络结构对平移、比例、倾斜或者其他形式的变形具有高度不变性。

卷积神经网络的基本结构包含输入层、卷积层和池化层。如下图1.7所示，它主要包含了三种思想：局部感受野、共享权重、池化，对基本结构和这三种思想做一一介绍。

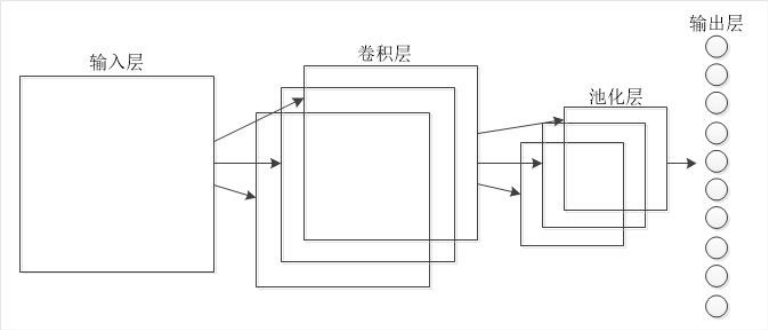

图 1.7 卷积神经网络的基本结构

1.3.1 局部感觉野（稀疏链接）

在卷积神经网络中，每个隐层的单元并不是连接到所有的输入层的单元上，比如，在输入层为28*28像素一张图片上，每一个隐层的单元连接到输入层5*5像素的小区域上，如图1.8所示。这个区域在输入层被称为对应隐层单元的局部感受野。每一个局部感受野都有一组权重和一个偏执值。我们可以这样认为每一个隐层单元可以学到对应局部感受野的特征。

我们可以滑动这个局部感受野在28*28输入层，按照从左到右，自上而下的方向，以一个步长的幅度，从而形成一个第一个24*24隐层，如下图1.8所示。我们可以按照具体的不同任务设置不同的步长。

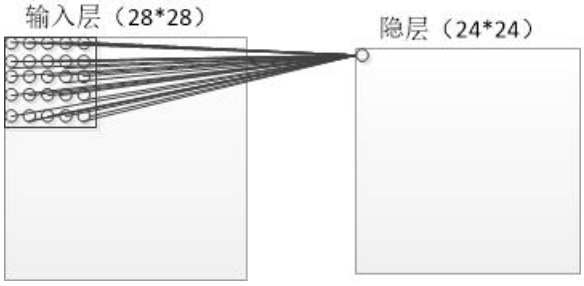

图 1.8 局部感觉野

1.3.2 共享权重

从上面可知，每一个隐层单元和对应的局部感受野有一个偏执值和一组5*5的权重矩阵。在建立第一个隐层的过程中，我们可以使用相同的权重和偏执值。也就是说，对于第j 行第k列的隐层单元来说，其公式如下：

$$h_{l,k} = \sigma(b + \sum_{l=0}^{4}\sum_{m=0}^{4} w_{l,m} a_{j+l,k+m}) \tag{1.2}$$

这里，σ是神经网络激活函数，b是共享的偏执值，$w_{l,m}$是5*5共享的权重矩阵，a为输入矩阵。

这意味着所有单元在第一个隐层学到的是相对于输入层不同位置的同类型的特征。这个权重矩阵代表图片的物理边缘信息，第一个隐层就可以学到整个图片的边缘信息。我们可以设置不同的共享权重和偏执值，得到分别代表图片不同特征的多个隐层。这个过程被称为特征提取，隐层被称为特征映射，不同的共享权重和偏执值被称为核函数或者过滤器，由不同核函数生成的一组特征映射被称为卷积层。如下图1.9所示。

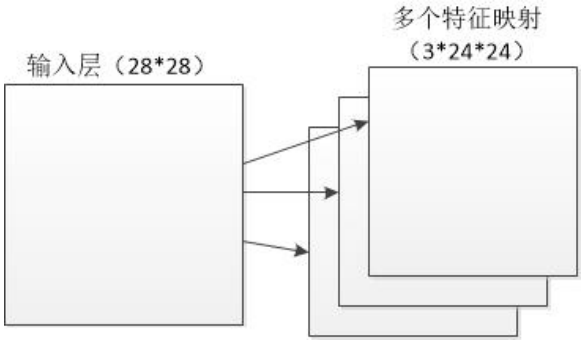

图 1.9 由不同的核函数得到不同的特征映射

1.3.3 池化

在卷积神经网络中，卷积层后面会连接一个池化层。池化层的主要作用是降低卷积层的分辨率。这种操作具有使特征映射的输出对平移和其他形式的敏感度下降的作用。

每一个单元在池化层是通过对卷积层的相对应的小区域做一些操作（比如，取最大值或者平均值）得到的。例如，我们设置用于池化的卷积层上的小区域为2*2像素，然后对24*24做池化，可以得到一个12*12的池化层，如下图1.10所示。

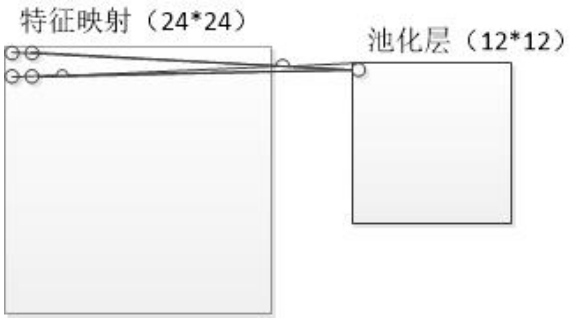

图 1.10 池华层

最后。我们可以设置一个输出层，不过输出层和池化层之间是全连接的，如图2.1所示。卷积神经网络中的卷积和池化是一种很强的先验知识，卷积负责抽取不同的特征，池化负责降低分辨率，使在加入噪声的输入提取的

特征具有不变性。卷积神经网络使用BP进去训练，由于其参数共享的特点，使它的训练时间远远小于传统的向前神经网络。

1.4 循环神经网络（RNN）

循环神经网络是一种专门为处理序列数据而设计的特殊结构的神经网络。传统的神经网络假设所有的数据之间都是互相独立的，而现实生活中我们要处理的数据往往具有依赖关系，比如预测一个句子的某一个词，如果我们知道这个词之前的信息，应该会有更好的效果。

循环神经网络的最大特点就是参数共享，这一点与卷积神经网络有点相似，不过又有很大的不同。卷积神经网络在处理一维数据的时候通过共享参数学习到数据间短暂的时间依赖关系。因为卷积操作的输出是对应输入数据相邻的局部单元的函数。循环神经网络的每一步都是用相同的参数，使其能够学到比较长的序列依赖关系，甚至具有逻辑推理的能力。

1.4.1 循环神经网络结构

循环神经网路，即一个序列当前的输出与前面的输出也有关。具体的表现形式为网络会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层之间的节点不再无连接而是有连接的，并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出。理论上，RNNs能够对任何长度的序列数据进行处理。

根据不同的任务，其结构细节会有不同。如果想把一个序列数据滚成一个固定的向量，我们只需要把它最后一步的隐层状态作为输出，不需要输出层。如果想建立一个生成模型，我们就需要把下一个单词作为输出层，如下图1.11所示。

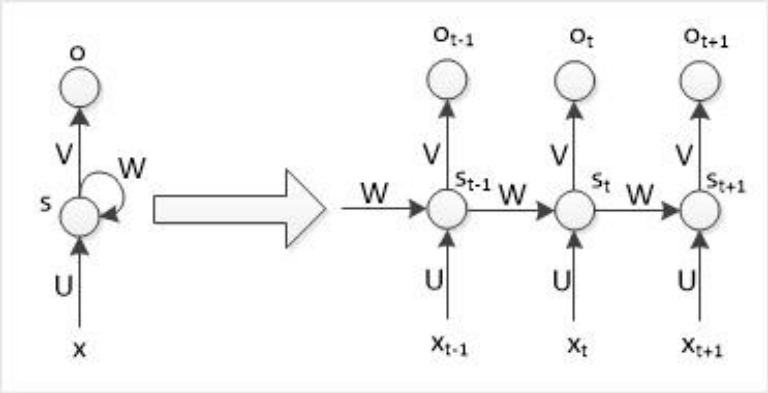

图 1.11 循环神经网络

上面右部分显示了循环神经网络展开之后变成一个前向神经网络。这就意味着我们可以根据输入数据的长短把它展开为拥有不同层的前向神经网络。比如，一个序列的长度为6，它可以被展开为6层的前向神经网络。循环神经网络作为生成模型工作的训练步骤如下：

- x_t是在第t步的输入，假设输入x为整个词表的one-host向量。
- s_{t-1}是第t-1步的隐层状态。它是这个网络的记忆单元。st根根当前的输入和上一步隐层的状态：$s_t = f(Ux_t + Ws_{t-1})$，f为激活函数，当t为1时，一般把上步的隐层状态值设为0。
- o_t是第t步的输出。因为我们假设上述模型为生成模型，输出就是在一个单词的概率，o_t为一个由所有单词概率组成的向量。
- 我们可以用损失函数计算每一步的输出和下一个输入之间的残差，然后通过BPTT进行参数的更新。

现在人们为了增强循环神经网络的表现力，发明了不同变种，这些改进比较一致的地方是增加网络的深度。

1.4.2 梯度问题及解决方法

循环神经网络梯度实际上通过雅可比矩阵相乘计算出来的，公式如下所示：

$$L = L(s_T(s_{T-1}(...s_{t+1}(s_t,...)))) \tag{1.3}$$

$$\frac{\partial L}{\partial s_t} = \frac{\partial L}{\partial s_T}\frac{\partial s_T}{\partial s_{T-1}}\cdots\frac{\partial s_{t+1}}{\partial s_t} \tag{1.4}$$

当雅克比矩阵的奇异值都大于1时，梯度就出现暴涨；当雅克比矩阵的奇异值都小于1时，梯度会出现消失的现象；当雅克比矩阵的奇异值是随机的是，方差就会暴增。

为了解决这个问题，一般有以下几个方法可以使用：

- 裁剪梯度：解决了梯度爆炸
- 选择式传递梯度：传播长时依赖，如LSTM
- 动量方法：模拟便宜的二阶导
- 参数初始化：在一个比较正确的位置以避免梯度爆炸或者消失
- 稀疏梯度：打破均衡
- 梯度传播正则化：避免梯度的消失
- LSTM的自循环：避免梯度消失

1.4.3 长短时间记忆网络（*LSTM*）

由于传统的循环神经网络会出现梯度消失的问题，我们在实际工作中并不会用它，而是使用增加了各种门的变体。现在比较成功的变体就是长短时间记忆网络。它的核心思想是通过受限的自循环去产生一条梯度能长距离传播的路径，受限的自循环是由输入门、遗忘门和输出门控制，如下图31.12所示。这种网络使雅克比矩阵的特征值都略小于1，很好的解决了梯度消失的问题。

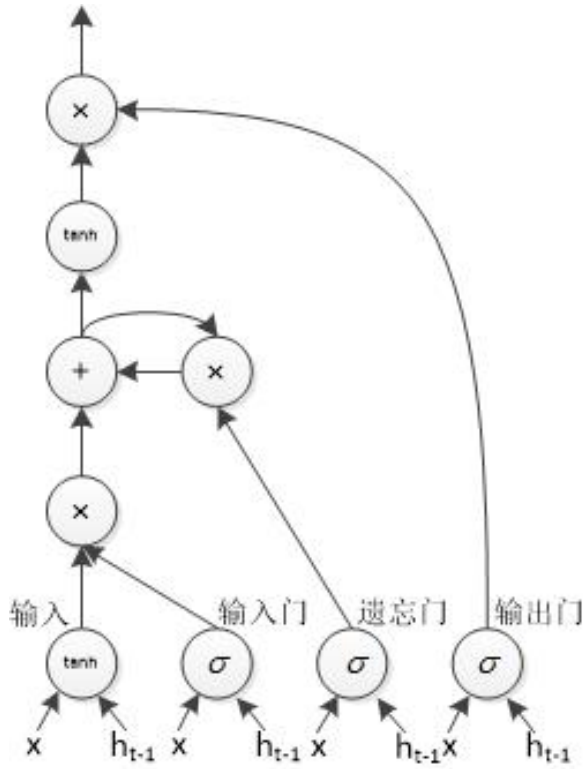

图 1.12 长短时间记忆网络基本结构

1.5 正则化

机器学习的中心问题是如何在训练集和测试集上都有比较好的表现。过高的方差将会导致过拟合，复杂的海森矩阵、局部最小点、马鞍点和过慢的收敛都会导致欠拟合。深度神经网络有非常灵活的结构，这也导致了容易过拟合和欠拟合。正则化是使学习算法不仅仅降低在训练集上的错误率也降低在测试集上的错误率，从而防止过拟合和欠拟合。正则化的方法是引入根据先验知识设计出的限制项和惩罚项，下面介绍几种常用的正则化。

1.5.1 参数范式惩罚项

对参数进去正则化背后的思路就是如果我们的参数值对应一个较小的值，那么往往我们会得到一个形式更简单的假设。实际上，这些参数的值越小，通常对应于越光滑的函数，也就是更加简单的函数。因此就不易发生过拟合的问题。对参数进去正则化一般在目标函数中加入一个参数范式惩罚项，公式如下所示：

$$\tilde{J}(\theta;X;y) = J(\theta;X,y) + \alpha\Omega(\theta) \tag{1.5}$$

这里，$\alpha \in [0,\infty)$是超参数，Ω是范式惩罚项，限制参数范围相对于标准的目标函数。Ω可以是L1或者L2,也可以是其他任意的限制函数，公式如下所示：

$$\tilde{J}(w;X;y) = \frac{\alpha}{2}w^T w + J(w;X,y) \tag{1.6}$$

$$\tilde{J}(w;X;y) = \alpha\|w\|_1 + J(w;X,y) \tag{1.7}$$

L1会引起参数的稀疏性，因为标准目标函数和惩罚项函数更容易在L1的夹角相遇，如下图1.13所示。

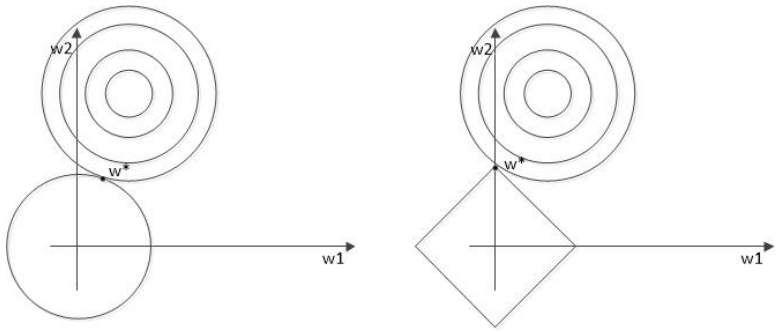

图 1.13 左边为加入L2惩罚项，右边为加入L1惩罚项

1.5.2 数据扩增

使模型泛化能力更强大的办法是增加训练数据，不过在现实中，我们拥有的数据往往很有限。有一种方法可以解决这个问题，我们可以伪造一些数据集，把它们添加到训练集里。对于一些机器学习的任务，这种方法比较合理。

这种方法最适合分类。因为分类就是把一个复杂，高维的输入数据x映射到单一的分类y。比如，图片识别，训练集很少，我们可以通过对训练集数据进行变形，生成一些新的数据，从而增加训练集。

数据扩增的方法还有一下几种：

- 输入数据加入噪声：在输入中加入一些随机的噪声，可以提高模型的鲁棒性
- 在隐层加入噪声（drop out）：在训练时，随机把隐层节点置0
- 在目标中加入噪声：在分类问题中，如果有多个类别需要预测，我们可以为这些类别设置一个概率值，而不是0或1，使得预测平滑

1.5.3 共享参数

共享参数参照人类的先验知识而设计出的一种正则化方法，实现共享参数的典型神经网络结构有如下几种：

- CNN：通过卷积操作进行参数共享
- RNN：通过自循环实现参数共享
- NADE：输入与之前以一定步长的输入共享参数

1.5.4 相关联训练

相关联训练就是用一个模型训练的结果去影响和指导另一个模型的训练，模型之间就形成了一个正则化。主要有以下几个方式：

- 半监督学习:有两种方式，第一种：利用标记过的数据，指导没有标记过的数据进行训练；第二种：有标记的任务和无标记的任务共享一部分模型

- 多任务学习：利用一个模型，去训练不同的任务。不同的任务之间会对模型参数形成相互牵制的作用。
- 协作学习：一个模型的训练结果作为一个模型的

1.5.5 稀疏性表示

稀疏性是指使输出大部分为0，一般被认为可以把不相关的参数给剔除，可以提高神经网络的效果。使神经网络参数稀疏性也是一种正则化，主要有以下几种方式产生稀疏性：

- L0 and L1范式
- 在隐层单元增加惩罚项（L0和L1）
- 预训练导致稀疏性
- 稀疏的受限玻尔兹曼机，稀疏的深度自信网络，抗噪声的自编码网络
- 在卷积操作时，只保留最大值，其他的都置为0

1.5.6 复向训练

复向训练的思路是找到被分错的样本，在这些样本中加入噪声之后再进行训练。加入噪声的原因是一般认为样本在输入空间的位置变化很小的话，在输出空间的变化也应该很小。加入噪声之后训练模型可以提高网络的泛化能力。

1.5.7 流形正则化

任何事物都可以在低维空间表示，流形正则化就用了有监督和无监督样本共同来来控制样本分布的几何形状进而找出分布特征。它的原理就是先用半监督的方法找到数据的分布信息，然后给这些学到的分布信息贴上少量有监督的标签。???

1.5.8 知识转移

知识转移的基本原理就是训练一个模型作为老师，让这个老师指导其他模型的训练。它常见的用法是一个复杂的模型训练一个简单的模型，或者一个简单的模型训练一个复杂的模型。