2018-12-25，**主要内容：**

深度学习初稿撰写要求：

1. 不能直接粘贴拷贝网上或者其他书籍内容，需按照自己的理解撰写。
2. 图表不能用现有图表，需重新制作，图用Vision绘图，表用Word制表。
3. 每章后需整理本章参考文献
4. 不局限于提纲内容，有修改或者新加内容随时和我讨论
5. 图文并茂，尽量通俗易懂 。

# 第七章 其他典型深度学习方法

**7.1 生成对抗网络**

## 7.1.1概述

生成对抗式网络（Generative Adversarial Networks，GANs）是一种以无监督方式学习目标分布的深度生成模型。这种目标分布的学习能广泛的应用于很多应用，包括：文字转化图形、图像生成、图像风格转换、图像重构与分类等。GANs最早在2014年提出，是一种半监督与无监督的学习方法，它实现通过隐性数据的高维分布。GANs是训练一对相互竞争的网络，基于博弈论的思想。这里我们做一个形象地比喻，可以将其中一个网络视为造假者（生成器），另一个看作专家。学习的过程就是造假者尽最大努力仿造出接近真实的物品，专家（判别器）的职责就是将仿造品与真品进行比较，鉴别真伪。重复以上过程，专家和伪造者同时训练并相互竞争，直到两者达到一个均衡和谐的状态后，最终伪造者能产出更为真实的的物品，专家则一个判断能力较强的分类器。前者可以用于机器创作（自动画出“猫”“狗”），而后者则可以用来机器分类（自动判断“猫”“狗”）。

## 7.1.2基本算法原理

7.1.2.1 基本结构

GANs基于博弈论的思想，其中生成器网络（generator network）与其对手判别器网络（discriminator network）相互竞争。 生成对抗网络是一种生成式的建模方法，GANs的结构和我们之前见到的神经网络略为不同。大体上来说，GANs由生成模型Generator和判别模型Discriminator组成，基本的结构图如下：



其中，生成模型我们可以将其看做一个神经网络模型，输入一个噪声/样本后，输出的将是一个图像。上面结构图中包括两个数据集一个真实数据集另一个合成数据集，即生成网络合成的猫图像数据。判别模型，也是一个简单的神经网络架构，输入是一副图像，分别来自真实训练数据和合成数据，通过判别网络后输出则是一个概率值，基于此来判断猫图像的真假程度。随着以上训练的进行，判别器‘被迫’增强自身的判别能力，而生成器‘被迫’生成越来越逼真的输出，以欺骗判别器。理论上，最终生成器和判别器会达到一种均衡“纳什均衡”，即生成网络合成的假样本数据进入判别网络后，判别网络输出的结果是近似接近0.5的值，不能区分真假样本。

7.1.2.2 形式化表示

GANs中生成器网络G采用以随机噪声/样本为输入的多层神经网络，判别器网络D也是一个多层神经网络。G可以表示成 ,其中z为随机噪声，服从分布。D输入一是个真实图像及合成（虚假）图像x, 输出结果可以表示为表示输入图像是真实和伪造图像。根据训练过程我们可以定义一个损失函数：

**（公式7.1）**

其中x、z分别表示来自训练集数据和随机噪声数据，优化目标是。

为便于理解，我们对上面公式做一些简单说明，是指让D的损失越大越好, 越大说明D越能发现G生成的数据是假的, 因为二者目的就是给对方制造麻烦的. 是指G希望D的损失越小越好, 因为越小说明G生成的数据越能骗过D。

是希望**(公式7.1)**最大, 当为1,为0时,说明D把所有真实的数据识别出来了,并把所有的假的数据也识别出来, 此时，我们便得到一个优秀的判别器，损失达到最大。相反，是希望**(公式7.1)**最小, 当时,判别器就把所有的假数据当作了真的, 此时，说明G非常成功，生成器就就具备了以假乱真的功能。

  总之，我们在更新D和G的参数时就是分别对D进行梯度上升更新,对G进行梯度下降更新。 到最后达到一个动态平衡, 也就是D的输出为, 即D已经不能区分真假了。

7.1.2.3 算法的优势和不足（https://blog.csdn.net/xg123321123/article/details/78034859）

1. 优势
2. 模型只用到了反向传播,而不需要马尔科夫链。
3. 理论上,只要是可微分函数都可以用于构建D和G,因为能够与深度神经网络(CNN、RNN等)结合做深度生成式模型。
4. G的参数更新不是直接来自数据样本,而是来自D的反向传播。
5. 训练时不需要对隐含变量做推断。
6. GANs是一种以半监督方式训练分类器的方法.在你没有很多带标签的训练集的时候,你可以不做任何修改的直接使用我们的代码,通常这是因为你没有太多标记样本。
7. 相比玻尔兹曼机,GANs生成实例的过程只需要模型运行一次,而不是以马尔科夫链的形式迭代很多次。
8. 不足
9. 可解释性差,生成模型的分布Pg(G)没有显式的表达。
10. 比较难训练,D与G之间需要很好的同步，例如D更新k次而G更新一次。GAN的学习过程可能发生崩溃问题，生成器开始退化，总是生成同样的样本点，无法继续学习。当生成模型崩溃时，判别模型也会对相似的样本点指向相似的方向，训练无法继续。
11. 网络难以收敛，目前所有的理论都认为GAN应该在纳什均衡上有很好的表现，但梯度下降只有在凸函数的情况下才能保证实现纳什均衡。
12. 对于生成离散数据比较困难,就像文本。

## 7.1.3应用与实例（https://www.jianshu.com/p/80bd4d4c2992）

GANs是一个生成模型，其最直接的应用就是真实数据的建模与生成，比如音频、图像等。同时，可在一定程度上解决传统机器学习训练数据不足的问题。总之，目前无论在学术界和还是在工业界，如推特、苹果公司等，GANs都有广泛的应用。

1. 文本到图像

实通过输入目标图像的相关属性的描述性文本后，GANs可输出与之对应的最为真的图片。

1. 图像超分辨率

图像超分辨率一直是一个很重要的研究课题，比较重要的是对天文图像和卫星图像做超分辨率，不管是在天文，军事还是其他方面，都有很重要的应用。在生活中，如果有标清的视频可以变为高清的视频

1. 图像翻译

指的是图像到图像的翻译，比如说将语义标注图、灰度图或边缘图作为GAN的输入，那么我们希望它输出能够和输入图一致的真实图像，例如这里的街景图和彩色图。

**参考文献**

⑴ Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio.2014. Generative Adversarial Networks.

(2) Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, Anil A Bharath.2017. Generative Adversarial Networks: An Overview.

7.6 深度森林

7.6.1概述

“深度森林”在2017年被提出，这是一种可以与深度神经网络相媲美的基于树的模型。近十年来，深层神经网络的发展在机器学习领域取得了显著进展。通过构建分层或深层结构，该模型能够在有监督或无监督的环境下从原始数据中学习良好的表征，这被认为是其成功的关键因素。成功的应用领域包括计算机视觉、语音识别、自然语言处理等。

目前，几乎所有的深层神经网络都使用具有随机梯度下降的反向传播作为训练过程中更新参数的幕后主力军。实际上，当模型由可微分量（例如，具有非线性激活函数的加权和）组成时，反向传播似乎仍是当前的最佳选择。其他一些方法如目标传播已经被作为训练神经网络的替代方法被提出，但其效果和普及还处于早期阶段。例如，研究表明，目标传播最多可达到和反向传播一样的效果，并且实际上常常需要额外的反向传播来进行微调。换句话说，老掉牙的反向传播仍然是训练神经网络等可微分学习系统的最好方法。

另一方面，探索使用非可微模块来构建多层或深度模型的可能性的需求不仅仅是学界的兴趣所在，其在现实应用上也有很大的潜力。例如，基于树的集成（例如随机森林或梯度提升决策树（GBDT）仍然是多个领域中建模离散或表格数据的主要方式，为此在这类数据上使用树集成来获得分层分布式表征是个很有趣的研究方向。在这样的案例中，由于不能使用链式法则来传播误差，反向传播不再可行。

这引发了两个基本的问题：首先，我们是否可以用非可微组件构建多层模型，从而中间层的输出可以被当作分布式表征？其次，如果是这样，如何在没有反向传播的帮助下，联合地训练这种模型？Deep Forest基于不可微的模块建立深度模块，这就是gcForest。本章内容就逐一讨论这些问题。

7.6.2 主要思想

7.6.2.1 多粒度串联森林

**gcForest**（muti-Grained Cascade Forest，多粒度串联森林），它是基于树的集成方法，通过对树组成的森林来集成并前后串联起来达到**表征学习**的效果。它的表征学习能力可以通过对高维输入数据的多粒度扫描而进行加强。串联的层数也可以通过自适应的决定从而使得模型复杂度不需要成为一个自定义的超参数，而是一个根据数据情况而自动设定的参数。gcForest方法也是延续DNN的对原始特征做逐层处理，其中的级联森林结构如下图所示：



每一级都是决策森林的集合，分别使用完全随机森林和随机森林，每个完全随机的森林包含完全随机树，通过随机选择一个特征在树的每个节点进行分割实现生长，直到叶子节点只包含相同类的实例。类似的，每个随机森林包含的树，通过随机选择sqrt(d)数量的特征作为候选（d是输入特征的数量），然后选择具有最佳gini值的特征作为分割。每个森林中的树的数值是一个超参数。

给定一个实例，每个森林会通过计算相关实例落入叶节点处的不同类的训练样本的百分比，然后对森林中所有树计算均值，以生成对类的分布的估计。

被估计的类分布形成类向量（class vector），该类向量接着与输入到级联的下一级的原始特征向量相连接。例如，假设有三个类，则四个森林每一个都将产生一个三维的类向量，因此，级联的下一级将接收12 = 3×4个增强特征（augmented feature）。

为了降低过拟合风险，每个森林产生的类向量由k折交叉验证（k-fold cross validation）产生。具体来说，每个实例都将被用作 k -1 次训练数据，产生 k -1 个类向量，然后对其取平均值以产生作为级联中下一级的增强特征的最终类向量。需要注意的是，在扩展一个新的级后，整个级联的性能将在验证集上进行估计，如果没有显着的性能增益，训练过程将终止；因此，级联中级的数量是自动确定的。与模型的复杂性固定的大多数深度神经网络相反，gcForest 能够适当地通过终止训练来决定其模型的复杂度（early stop）。这使得 gcForest 能够适用于不同规模的训练数据，而不局限于大规模训练数据。在这里，大家不妨将其和卷积神经网络的特征提取原理做一个比较。

接下来，gcForest该处理数据的特征关系了，也是受到DNN的启发，对序列数据有效，其中顺序是极为关键的，因此，在gcForest中，用多粒度扫描来增强级联森林。如下图所示：



上图表示了序列原始输入特征所用的方法，其大致思想是采用滑动窗口的方法，先生成若干个实例，然后通过实例生成两个森林，一个完全随机森林，一个随机森林，然后再把生成的两个森林生成对应的相同维度的“类向量”，最后把这两大类向量连接在一起，综上，就是通过某段原始特征生成部分深度森林的想法，但应对的一般来说会采用多个不同大小的窗口做扫描。

值得注意的是，gcForest会比DNN有更少的超参数，更好的一点在于gcForest对参数是有非常好的鲁棒性，哪怕用默认参数也可以获得很棒的结果。换句话来说，gcForest相对于DNN，不仅超参数更少，而且对超参数的依赖性也更低。因为这样，gcForest的训练更为便捷，理论分析也更为清晰，从超参数的数量来看，更少超参数意味着加入更少的主观因素干预。同时，因为gcForest在训练时可进行并行计算，所以，也意味着训练时间较短。

与深度神经网络的性能相比，深度森林还是具有一定的竞争力，尤其是在小规模的训练数据情况下。

7.6.2.2基于森林的自编码器（Auto Encoder Forest）

自编码器（Auto-encoder）是神经网络的一种，是一种重要的表示学习模型，是深度学习的关键要素之一。自编码器的基本结构是由一个编码器（encoder）和一个解码器(decoder)组成,其中encoder将输入映射到隐空间，encoder将隐空间的表示重构为原表示。

本方法是在随机森林的基础上构建出编码器与解码器。首先是前向编码过程，在一个有N个已训练决策树的森林中，前向编码过程接受输入数据并将其发送到集成方法中每棵树的根结点，一旦数据遍历（traverse）到所有树的叶结点，该过程将返回 T 维向量，每个元素 t 是树 t 中的叶结点的整数索引。如**算法1前向编码**的过程，注意该编码过程与如何分割树节点的特定学习规则是彼此独立的。例如，可以在随机森林的监督环境中学习决策规则，也可以在无监督的环境（例如完全随机树）中学习。

算法 7.1 前向编码

输入:输入数据x,其包含T棵树的森林F

输出：的编码

= . %初始化编码

**for** in range(T) **do**

= Forest.tree.

%返回树的叶子节点整数索引向量

**end**

return

在设计解码器的时候稍微有点技巧。利用决策树进行决策时需要将决策的路径记录下来，而每个决策路径对应了一个规则（rule）。在进行解码重构的过程中，利用森林中N个规则，运用最大相容规则Maximum Compatible Rule（MCR），获取N维编码所对应的更精确的规则，以此规则进行重构。对于枚举型属性，该规则中的值可直接作为属性值，对于连续变量，通常规则得到的是属性的一个范围，可以取其最大值，最小值，或者中值作为其最终属性。首先介绍MCR的算法原理：

算法 7.2 计算最大相容规则

输入:由森林定义的一个规则列表集合T

输出：最大相容规则

MCR = initialize\_list(). %初始化编码

**for** in range(T) **do**

path\_rule = rule\_list.

for node\_rule in path\_rule\_list **do**

j = node\_rule.attribute

MCR = intersect(MCR,node\_rule.bound).

end

**end**

return

后向解码过程。具体来说，给定一个已训练的森林和特定数据在 R^T 中前向编码的 x\_enc，后向解码将首先通过 x\_enc 中的每个元素定位独立的叶结点，然后获得对应决策路径的 T 个决策规则。随后通过计算 MCR（Maximal-Compatible Rule），我们能从 x\_enc 反推回输入空间中的 x\_dec，因此也就得到了重构。后向解码的算法原理如下所示：

算法 7.3 反向解码

输入:由T棵树组成的森林F的数据编码

输出：

rule\_list = list(). %初始化编码

**for** in range(T) **do**

path = forest.tree.get\_path().

path\_rule = calculate\_rules(path).

path\_rule = simplify(path\_rule).

rule\_list.append(path\_rule).

**end**

**MCR = calculate\_MCR(rule\_list).**

= sample(MCR)

return

eForest 能获得较低的重构误差（reconstruction error）和较快的训练速度。此外，模型本身还具有容损性（damage-tolerable）和可复用性（reusable）。

目前在实验中已证明 eForest 方法具有以下优势：

准确性：实验重构误差（reconstruction error）比基于 MLP 和 CNN 的自编码器更低。

高效性：eForest 在单块 KNL（多核 CPU）上训练的速度甚至比 CNN 自编码器在 Titan-X GPU 上还快。

容损性（Damage-tolerable）：训练的模型即使在部分损坏的情况下也能很好地工作。

可复用性（Reusable）：从一个数据集训练的模型可直接应用于同一领域下的其他数据集。

7.6.2.3 多层梯度提升决策树（Multi-Layered Gradient Boosting Decision Trees）

多层梯度提升决策树模型，它通过堆叠多个回归 GBDT 层作为构建块，并探索了其学习层级表征的能力。此外，与层级表征的神经网络不同，他们提出的方法并不要求每一层都是可微，也不需要使用反向传播更新参数。因此，多层分布式表征学习不仅可选择深度神经网络，也可选择决策树。

多层梯度提升决策树模型通过引入细粒度扫描和级联操作，该模型能够构建具有自适应模型复杂性的多层结构，并在一系列任务中实现竞争性能。树集成的优秀性能和层次分布式表示的表达能力（主要在神经网络中探索）。具体地说，多层结构，使用梯度提升决策树作为每层构建块，明确强调其表示学习能力，并且可以通过目标传播的变体联合优化训练过程。可以在监督和无监督环境下训练模型。~~这是我们确实可以使用树获得分层和分布式表示的第一个证明，这通常被认为仅可用于神经网络或一般的可微系统。理论正确性和实验结果表明了这种方法的有效性。~~

考虑具有M-1个中间层和一个最终输出层的多层前馈结构。表示，其中作为包括输入层和输出层的每个层的输出。对于特定输入数据x，每层的相应输出在中，其中。因此，学习任务是指学习每个层i>0的映射，使得最终输出在训练集上最小化经验损失L。添加正则化项的均方误差或交叉熵是寻找最小化损失L的一些常见选择。在无监督环境中，期望输出Y可以是训练数据本身，其导致自动编码器，并且损失函数是指输出和原始输入之间的重建误差。

当每个是参数化的且是可微的时，可以使用反向传播以高效的方式实现这种学习任务。基本程序是使用链式法则计算每层的每个参数的损失函数的梯度，然后执行参数更新的梯度下降。一旦完成训练，中间层的输出可以被视为模型学习的新表示。这种分层密集表示可以被解释为原始输入的多层抽象，并且被认为对于深度模型的成功是至关重要的。

然而，当是不可微的或甚至是非参数化的时，由于不可能计算损失函数相对于其参数的导数，因此反向传播不再适用。当是梯度提升决策树时，通过以下方法进行研究。

首先，在迭代t，假设给出从前一次迭代获得的，我们需要获得与每个配对的“伪逆”映射，使得。这可以通过最小化重建损失函数的期望值来实现：，其中损失可以是重建损失。像自动编码器一样，通常会建议随机噪声注入，也就是说，不是使用纯重建误差测量，而是将设置为：。通过这样做，在迫使逆映射学习如何将相邻训练数据映射到正确的流形意义上，使模型变得更鲁棒。此外，这种随机性注入还有助于通过将逆映射方向视为生成路径的方法来设计生成模型，该生成路径可被视为未来的探索工作。

其次，一旦我们更新了，我们就可以将其用作给定的并更新前一层的前向映射。这里的关键是为分配伪标签，其中*i∈{2，... M}*，并且每个层的伪标签被定义为。也就是说，在迭代t，对于所有中间层，每个层的伪标签可以“对齐”并从顶部到底部传播。然后，一旦计算出每个层的伪标签，每个就可以遵循朝向伪残差的梯度上升步骤,就像典型的回归GBDT一样。

唯一剩下的就是为最终层设置伪标签，以使整个结构准备好进行更新。 结果很简单，因为在M层，人们总是可以在定义顶层伪标签时使用真实标签y。 例如，很自然地将顶层的伪标签定义为： 。然后将设置为朝向伪残差。 换句话说，在迭代t，顶层计算其伪标签，然后通过反函数产生所有其他层的伪标签，然后可以相应地更新每个。一旦所有都得到更新，该过程就可以移动到下一次迭代以更新。在实践中，建议自下而上更新（在之前为i <j更新）并且每个可以朝向其当前的伪标签进行几轮叠加提升步骤。

在训练神经网络时，可以通过为每个参数分配随机高斯噪声来实现初始化，然后该过程可以继续进行下一阶段的参数更新。对于这里描述的树形结构模型，从所有可能的树配置的分布中绘制随机树结构并不是一项微不足道的任务，因此我们不是随机地初始化树结构，而是产生一些高斯噪声作为中间层的输出，并训练一些非常小的树以获得，其中索引0表示在该初始化阶段中获得的树结构。 然后，训练过程可以继续进行迭代更新前向映射和反向映射。整个过程可通过**算法1**进行形式化表示，并在**图1**中示出。



在分类任务中，可以将顶层中的前向映射设置为线性分类器。这样做有两个主要原因：首先，通过这样做，较低层将被迫学习一个尽可能线性可分的特征重新表示，这是一个有用的属性。其次，输出层和下面的层之间的维度的差异通常很大，因此，可能难以学习准确的逆映射。当使用线性分类器作为顶层的前向映射时，不需要计算该特定的对应逆映射，因为可以通过关于最后一个隐藏层的输出的全局损失的梯度来计算下面层的伪标签。

算法1：训练多层决策梯度提升树

**输入**：层数，每层维度,训练数据,损失函数，迭 代次数，噪声注入

**输出**：训练完成的多层决策梯度提升树

for t = 1 to E do

//利用真实标签来定义伪标签

for j = M to 2 do //从后向前即M到2进行传播

//更新逆映射

for k = 1 to do

Fit regression tree to ,i.e. using the training set

end

// 当更新完成后利用和对伪标签进行更新

end

for j =1 to M do

//当第 t 轮所有的伪标签更新完后，然后前向传播，用以计算正向反射函 //数 ，然后使用上一轮的 计算更新

for k = 1 to do

Fit regression tree to , i.e. using the training set

end

// 当第 t 轮所有的都更新完成之后,更新输出函数

end

end

return ,

目前已经提出了诸如目标传播的类似过程来使用层间反馈映射来训练神经网络。在某些条件下，前向映射参数的更新方向与反向传播训练时的更新方向之间的角度小于90度。然而，证据在很大程度上依赖于计算和的雅可比行列式，因此，它们的结果仅适用于神经网络。

以下定理证明，在某些条件下，中间层朝向其伪标签的更新有助于减少上述层的损失，从而有助于减少全局损失。这里的证据不依赖于和的可微性。

定理1. 表示将更新为将其输出从移动到，其中和在中并且表示的输入为，其在中。假设每个并保留其邻居的同构等距。现在假设这样的更新减少了它的局部损失，即，那么它有助于减少上面一层的损失，即以下情况：

证明如下：

在本节结束时，我们将讨论在设计多层模型时需要探索不可微模块的几个原因。首先，当前的对抗性攻击都是基于计算相对于输入的最终损失的导数。也就是说，无论训练过程如何，只要链式法则适用，就始终可以攻击系统。 诸如树之类的不可微模块可以自然地阻止这种计算，因此，执行恶意攻击会更加困难。其次，仍有许多感兴趣的数据集最适合用树形结构建模。提出可以将树集成的性能与具有多层表示的好处相结合的算法将具有很大的兴趣和潜力。

本节总结：

第一，基于决策树的模型更具有可解释性；第二，深度森林可以帮我们去理解很多在时序上和空间上都有规则的事情。

7.6.3应用与实例

利用深度森林在图象层级上的处理，如图像分类、人脸识别等，其实跟深度神经网络差不多，稍微能打平，甚至当数据集的量大的时候，深度神经网络还是会比深度森林好。但是在涉及到时间维度上，具有连续性的数据，或者从杂乱无章的文本中抽取情感。一般的，类似空间关系不是那么好的数据，其优势比较明显，比如通过机电信号去识别手势。说明它在处理这些连续有规则信号方面，非常的简洁高效。下面我们就以手势识别为例对深度森林做一个应用。

gcforest.py 整个框架的实现

fgnet.py 多粒度部分，FineGrained的实现

cascade/cascade\_classifier 级联分类器的实现

datasets/.... 包含一系列数据集的定义

estimator/... 包含决策树在进行评估用到的函数（多种分类器的预估）

layer/... 包含不同的层操作，如连接、池化、滑窗等

utils/.. 包含各种功能函数，譬如计算准确率、win\_vote、win\_avg、get\_windows等

数据集：

<http://archive.ics.uci.edu/ml/datasets/sEMG+for+Basic+Hand+moveme> nts#

数据引用：

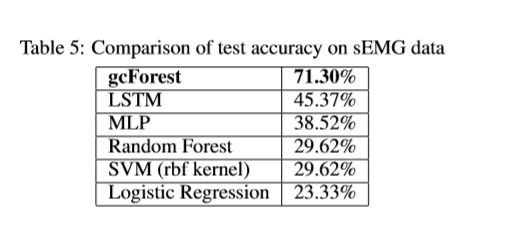
如果您发现这些数据库很有用，请引用以下内容：

对于数据库1），

C。Sapsanis，G。Georgoulas，A。Tzes，D。Lymberopoulos，“使用EMD改进基于EMG的基本手部运动分类”第35届IEEE医学与生物学学会年度国际会议13（EMBC 13），7月3日至7日，第5754-5757页，2013年。

对于数据库2）：

•S。Sapsanis。“使用肌电图识别基本的手部动作”。2013。



**参考文献**

[1]. Z.-H. Zhou and J. Feng. Deep forest: Towards an alternative to deep neural networks. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17), Melbourne, Australia, 2017.

[2]. J. Feng and Z.-H. Zhou. AutoEncoder by forest. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18) , New Orleans, LA, 2018.

[3]. J. Feng, Y. Yu, Z.-H. Zhou. Multi-layered gradient boosting decision trees. In: Advances in Neural Information Processing Systems 31 (NIPS'18), Montreal, Canada, 2018.

# 第十一章 深度学习在语音识别中的典型应用

**语音识别系统主要有四部分组成：信号处理、特征提取、声学模型、解码器**

11.1 声学模型（Acoustic Model, AM）

**11.1.1 概述**

声学模型的任务是给模型产生语音波形的概率。将声学和发音学的知识进行整合，以特征提取模块提取的特征为输入，生成声学模型得分。声学模型是语音识别系统的重要组成部分，它占据着语音识别大部分的计算开销，决定着语音识别系统的性能。传统的语音识别系统普遍采用的是基于GMM-HMM的声学模型，其中GMM用于对语音声学特征的分布进行建模，HMM则用于对语音信号的时序性进行建模。2006年深度学习兴起以后，深度神经网络（Deep Neural Networks，DNN）被应用于语音声学模型。2009年，Hinton及其学生将前馈全连接深度神经网络应用于语音识别声学建模，在TIMIT数据库上基于DNN-HMM的声学模型相比于传统的GMM-HMM声学模型可以获得显著的性能提升。DNN相比于GMM的优势在于：

1）DNN对语音声学特征的后验概率进行建模不需要对特征的分布进行去分布假设；

2）GMM要求对输入的特征进行去相关处理，而DNN可以采用各种形式的输入特征；

3）GMM只能采用单帧语音作为输入，而DNN则可以通过拼接相邻帧的方式利用上下文的有效信息。

2011年，DengLi等提出基于CD-DNN-HMM的声学模型，在大词汇量连续语音识别任务上取得成功，相比于传统的GMM-HMM系统可以获得超过20%的相对性能提升。基于DNN-HMM的语音声学模型开始取代GMM-HMM成为主流的声学模型。此后大量的研究人员投入到基于深度神经网络的语音声学建模研究中，语音识别取得了突破性的进展。

**11.1.2主流算法**

11.1.2.1 传统声学模型（GMM-HMM）

1. 混合高斯模型（Gaussian mixture model，GMM）

混合的高斯随机变量的分布来拟合训练数据（音频特征）时形成的模型。原始的音频数据经过短时傅里叶变换或者取倒谱后会变成特征序列，在忽略时序信息的条件下，这种序列非常适用于使用 GMM 进行建模。

如果一个连续随机变量X服从混合高斯分布，则它的概率密度函数为：

上式中分布概率是M个高斯分布的和，即所含成分，即单一高斯函数的数目。每个高斯分布有属于自己的和参数，以及对应的混合权重参数，权重值必须为正数，所有权重的和必须等于1，以确保公式给出数值是合理的概率密度值。表示第m个高斯成分的分布。

混合高斯模型分布最明显的性质是它的多模态，这使得混合高斯模型可以描述很多显示出多模态性质的物理数据，比如语音数据，而单高斯分布则不合适。数据中的多模态性质可能来自多种潜在因素，每一个因素决定分布中特定的混合成分。如果因素被识别出来，那么混合分布就可以被分解成有多个因素独立分布的集合。

在实际的 GMM 训练中，通常采用 EM（Expectation Maximization Algorithm，最大期望算法） 算法来进行迭代优化，以求取GMM中的加权系数及各个高斯函数的均值与方差等参数。

GMM 作为一种基于傅里叶频谱语音特征的统计模型，在传统语音识别系统的声学模型中发挥了重要的作用。其劣势在于不能考虑语音顺序信息，高斯混合分布也难以拟合非线性或近似非线性的数据特征。所以，当状态这个概念引入到声学模型的时候，就有了一种新的声学模型——隐马尔可夫模型（Hidden Markov model，HMM）。

1. 隐马尔科夫模型（Hidden Markov model，HMM）

 隐马尔可夫模型（Hidden Markov Model，HMM）是统计模型，用它处理的问题一般有两个特征：

第一：问题是基于序列的，比如时间序列，或者状态序列。

第二：问题中有两类数据，一类序列数据是可以观测到的，即观测序列；而另一类数据是不能观测到的，即隐藏状态序列，简称状态序列，该序列是马尔可夫链，由于该链不能直观观测，所以叫“隐”马尔可夫模型。

简单地说，状态序列前项能算出后项，但观测不到，观测序列前项算不出后项，但能观测到，观测序列可由状态序列算出。

 HMM模型的主要参数是λ=(A,B,)，如下图所示，数据的流程是通过初始状态生成第一个隐藏状态，结合生成矩阵B生成观测状态，根据转移矩阵A生成，和B再生成，以此类推，生成一系列的观测值。



HMM 的核心概念是状态，状态本身作为一个离散随机变量，马尔可夫链的每一个状态上都增加了不确定性或者统计分布使得 HMM 成为了一种双随机过程。隐马尔可夫模型能够描述语音信号中不平稳但有规律可学习的空间变量。具体的来说，隐马尔可夫模型具有顺序排列的马尔可夫状态，使得模型能够分段的处理短时平稳的语音特征，并以此来逼近全局非平稳的语音特征序列。

3）高斯混合-隐马尔科夫模型（GMM-HMM）

GMM-HMM网络在语音识别的发展过程中，曾是最为先进的语音识别系统。HMM模型对时序信息进行建模，在给定HMM的一个状态后，GMM对属于该状态的语音特征向量的概率分布进行建模。整个GMM-HMM网络其实主要是为了HMM网络服务的，比如把一系列梅尔频率倒谱系数（MFCC）特征正确的识别成对应HMM state 系列。这个过程涉及两个概率需要学习，一是把当前Frame的特征识别为这个state的概率，也就是通常HMM中说的Likelihood---这里指计算层面(也是GMM中的mean vector 和covariance matrix )，即GMM网络是为了获得当前state状态概率的，二是上个state转化为这个state的概率也就是状态转移概率Transition probabilities，这个过程是HMM中说的 Decoding---这里指计算层面。一个序列转化为另一个序列理论上有指数级种转化方式，所以每一个Frame只取概率最高的那个state，这样的路线选择方法被称为维特比算法方法。基本结构图如下所示：



最下面的Observation就是我们提取的特征。GMM-HMM就是把我们的特征用混合高斯模型区模拟，然后把均值和方差输入到HMM的模型里。我们获得Observation是语音数据，以下是一个词识别全过程：

1). 通过一定大小的移动窗口将语音数据切成等长Frames，对每个Frame提取特征（如：MFCC） 。

2).对每个Frame的特征跑GMM，得到每Frame属于每个状态的概率。

3). 根据每个单词的HMM状态转移概率a计算每个状态sequence生成该Frame的概率; 哪个词的HMM 序列跑出来概率最大，就判断这段语音属于该词。

11.1.2.2 深度神经网络-隐马尔科夫模型（DNN-HMM）

虽然GMM-HMM在以往取得了很多成功，但是随着深度学习的发展，DNN模型展现出了明显超越GMM模型的性能，于是就替代了GMM进行HMM状态建模。GMM模拟任意函数的功能取决于混合高斯函数的个数，所以具有一定的局限性，属于浅层模型。相反，深度神经网络可以模拟任意的函数，因而表达能力更强 。需要注意的是，替换GMM的深度神经网络模型要求是生成模型，如DBN、DBM等，因为在训练DNN-HMM网络的过程中会用到HMM的某个状态产生一个样本的概率。

需要注意的是，由于数字信号为连续型信号，DNN需要一定大小的输入，因此，DNN不能直接为语音进行建模。DNN具有强分类的能力，我们需要找到一种能够解决语音信号长度变化的方法，利用DNN的每个输出节点来估计连续密度HMM的某个状态的后验概率，即本部分所讲的DNN-HMM模型，包含了DNN很强的表现学习能力和HMM的序列化建模能力，在大多数的语音识别任务中，其表现能力都优于传统的混合高斯模型-HMM系统。除了DNN内在的鉴别属性，DNN-HMM还具有训练过程可以使用维特比算法和解码高效的特性。具体结构如下所示：



结构图上半部分是HMM，HMM的结构和转移概率是HMM-GMM模型训练的结果。中间部分是DNN结构，这个DNN模型来决定HMM的发射概率。通常情况下DNN模型的层数会小于5层，每层大概数千个神经元组成。下半部分是DNN模型的输入，其输入是多帧特征，而不是GMM模型中的一帧。在语音识别语音的过程中，一小段语音都会被提取成图中所对应的Observation,并根据HMM中的状态计算发射概率，选择发射概率最大路径作为最终结果。

**训练过程，**DNN-HMM包含三部分：深度神经网络DNN、隐马尔科夫模型HMM、状态先验概率分布prior。由于DNN-HMM与GMM-HMM模型共享因素绑定结构，训练DNN-HMM首先需要训练一个GMM-HMM系统。因为DNN训练标注是由GMM-HMM系统采用维特比算法产生，标注的质量会影响DNN的性能没所以训练一个优质的GMM-HMM模型作为初始模型是很重要的。

**11.1.3应用与实例**

**需有数据集和代码**

**参考文献**

1. [**https://blog.csdn.net/qq\_37572875/article/details/77899491**](https://blog.csdn.net/qq_37572875/article/details/77899491)
2. [**http://www.eeboard.com/news/yuyinshibie/**](http://www.eeboard.com/news/yuyinshibie/)
3. **http://www.doc88.com/p-2137495691629.html**
4. <https://www.jianshu.com/p/16fc3712fdf6>
5. https://blog.csdn.net/abcjennifer/article/details/27346787