第十二章 应用

- 大规模深度学习
- 计算机视觉
- 语音识别
- 自然语言处理
- 其他应用
  - 推荐系统
  - 知识表示、推理和问答

大规模深度学习

联结主义

- 单个特征不是智能的
- 大量的神经元或者特征作用在一起往往能够表现出智能

快速的CPU 实现

- 通过设计一些特定的CPU 上的操作可以大大提升效率
- 通过优化数据结构避免高速缓存缺失、使用向量指令等

GPU实现

- 显卡设计为拥有高度并行特性以及很高的内存带宽
- 通用GPU 可以执行任意的代码
- GPU 代码是天生多线程的，不同线程之间必须仔细协调好
- GPU 另一个常见的设定是使一个组中的所有线程都同时执行同一指令

大规模深度学习

大规模的分布式实现

- 数据并行：
  - 每一个输入的样本都可以在单独的机器上运行
- 模型并行：
  - 多个机器共同运行一个数据点，每一个机器负责模型的一个部分
- 异步随机梯度下降：
  - 几个处理器的核共用存有参数的内存

模型压缩

- 减少推断所需开销
- 基本思想是用一个更小的模型取代原始耗时的模型
- 拥有最小泛化误差的模型往往是多个独立训练而成的模型的集成

大规模深度学习

动态结构

- 加速数据处理系统
- 神经网络中的动态结构有时被称为**条件计算**：
  - 只计算那些需要的特征
- 级联：
  - 使级联中靠后的成员单独具有高容量
  - 每个单独的模型具有低容量，但是由于许多小型模型的组合，整个系统具有高容量
- 选通器
  - 硬专家混合体
- 开关：注意力机制

专用硬件

- 专用集成电路
- 现场可编程门阵列

计算机视觉

预处理

- 将图像格式化为具有相同的比例严格上说是唯一一种必要的预处理
- 对比度归一化
  - 全局对比度归一化（GCN）
    - $$X'_{i,j,k} = s\frac{X_{i,j,k} - \bar{X}}{\max\{\epsilon, \sqrt{\lambda + \frac{1}{3rc}\sum_{i=1}^{r}\sum_{j=1}^{c}\sum_{k=1}^{3}(X_{i,j,k} - \bar{X})^2}\}}$$
      - 全局对比度归一化常常不能突出我们想要突出的图像特征
  - 局部对比度归一化（LCN）
    - 通过减去邻近像素的平均值并除以邻近像素的标准差来修改每个像素
    - 常需要正则化

计算机视觉

预处理

图 12.2: 全局对比度归一化和局部对比度归一化的比较。直观上说，全局对比度归一化的效果很巧妙。它使得所有的图片的尺度都差不多，这减轻了学习算法处理多个尺度的负担。局部对比度归一化更多地改变了图像，丢弃了所有相同强度的区域。这使得模型能够只关注于边缘。较好的纹理区域，如第二行的屋子，可能会由于归一化核的过高带宽而丢失一些细节。

计算机视觉

数据集增强

- 通过增加训练集的额外副本来增加训练集的大小，进而改进分类器的泛化能力
- 类别信息对于许多变换是不变的
- 几何变换
  - 随即转换
  - 旋转
  - 翻转
  - 颜色的随机波动
  - 非线性几何变换

语音识别

自动语音识别（ASR）

- $f^*_{\mathrm{ASR}}(\boldsymbol{X}) = \arg\max_{\boldsymbol{y}} P^*(\mathbf{y} \mid \mathbf{X} = \boldsymbol{X})$
- 隐马尔科夫模型（HMM）
  - 对音素序列建模
- 高斯混合模型（GMM）
  - 对声学特征和音素之间的关系建模
- GMM-HMM 模型：
  - HMM 生成音素的序列和离散的子音素状态
  - GMM 把每一个离散的状态转化为一个简短的声音信号
- 受限玻尔兹曼机
  - 无监督的预训练被用来构造一个深度前馈网络
  - 输入是从一个固定规格的输入窗（以当前帧为中心）的谱声学表示抽取
  - 预测当前帧所对应的HMM 状态的条件概率

自然语言处理

基于n-gram 的模型

- 给定前$n-1$个标记后的第n个标记的条件概率
- $$P(x_1,\ldots,x_\tau) = P(x_1,\ldots,x_{n-1})\prod_{t=n}^{\tau} P(x_t \mid x_{t-n+1},\ldots,x_{t-1})$$
- 通常同时训练n-gram 模型和n-1 gram 模型
- $$P(x_t \mid x_{t-n+1},\ldots,x_{t-1}) = \frac{P_n(x_{t-n+1},\ldots,x_t)}{P_{n-1}(x_{t-n+1},\ldots,x_{t-1})}$$
- 限制：
  - 在许多情况下从训练集计数估计得到的P_n很可能为零
  - 维数灾难
- 基于类的语言模型

自然语言处理

神经语言模型

- 使用词的分布式表示对自然语言序列建模
- 神经语言模型共享一个词（及其上下文）和其他类似词（和上下文之间）的统计强度
- 模型为每个词学习的分布式表示，允许模型处理具有类似共同特征的词来实现这种共享

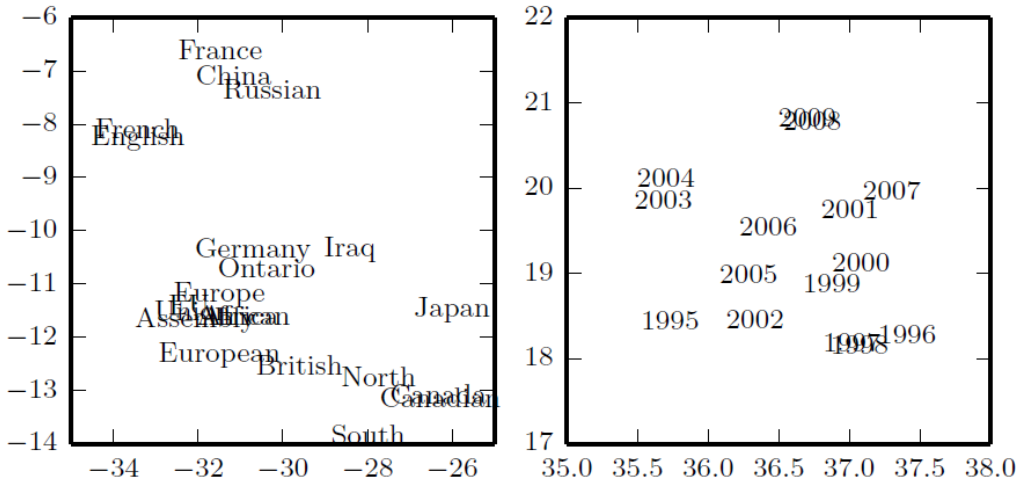


图 12.3: 从神经机器翻译模型获得的词嵌入的二维可视化 (Bahdanau *et al.*, 2015)。此图在语义相关词的特定区域放大，它们具有彼此接近的嵌入向量。国家在左图，数字在右图。注意，这些嵌入是为了可视化才表示为 2 维。在实际应用中，嵌入通常具有更高的维度并且可以同时捕获词之间多种相似性。

自然语言处理

高维输出

- 模型输出词，而不是符号
- $a_i = b_i + \sum_j W_{ij} h_j \quad \forall i \in \{1, \ldots, |\mathbb{V}|\}$
- $\hat{y}_i = \frac{e^{a_i}}{\sum_{i'=1}^{|\mathbb{V}|} e^{a_{i'}}}.$
- 如果h包含n_h个元素，则上述操作复杂度是$O(|V|n_h)$
- 降低成本：
  - 使用短列表
  - 分层softmax
  - 重要采样
  - 排名损失

自然语言处理

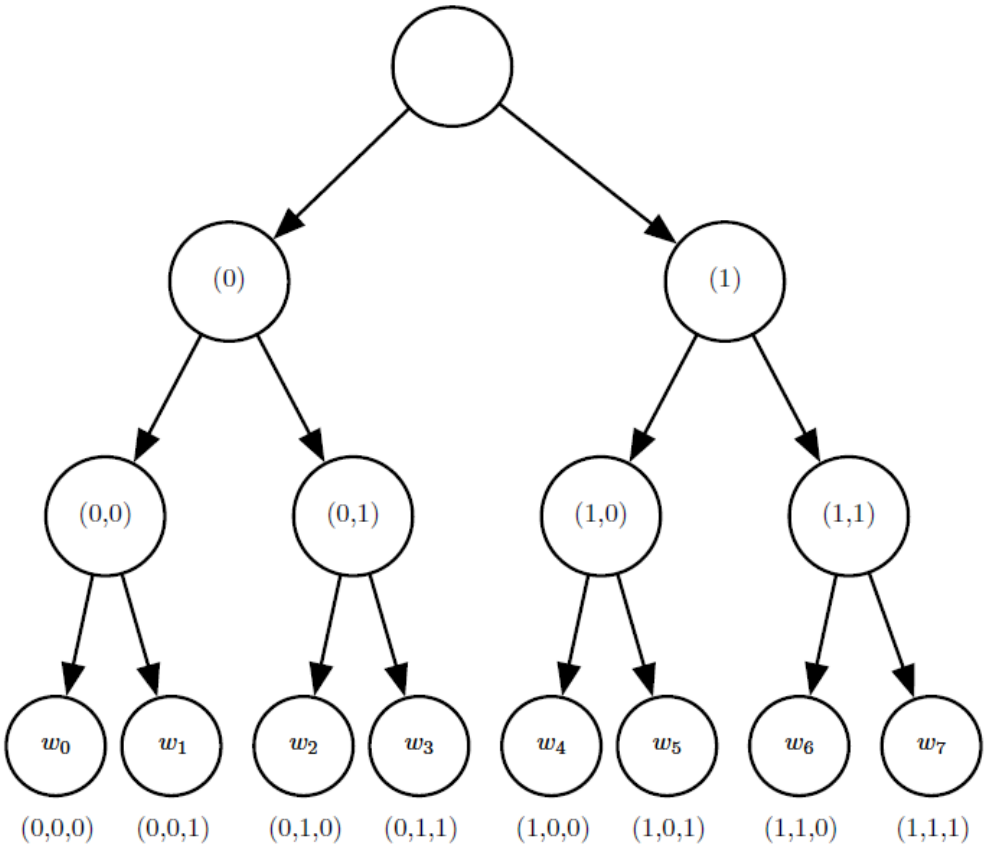
使用短列表

- 将词汇表$\mathbb{V}$分为最常见词汇（由神经网络处理）的短列表（shortlist）$\mathbb{L}$和较稀有词汇的尾列表$\mathbb{T} = \mathbb{V} \backslash \mathbb{L}$（由n-gram模型处理）
- 组合两个列表
- 添加额外的输出单元计算估计$P(i \in \mathbb{T}|C)$
- 计算$P(y = i|C) = 1_{i \in \mathbb{L}} P(y = i|C, i \in \mathbb{L})\big(1 - P(i \in \mathbb{T}|C)\big)$
  $+1_{i \in \mathbb{T}} P(y = i|C, i \in \mathbb{T}) P(i \in \mathbb{T}|C)$
- 缺点：
  - 神经语言模型的潜在泛化优势仅限于最常用的词

自然语言处理

分层softmax

- 将因子分解方法引入神经语言模型
  - 先建立词的类别，然后是词类别的类别 .....
  - 嵌套类别构成一棵树，其叶子为词



自然语言处理

分层softmax

- 将因子分解方法引入神经语言模型
  - 先建立词的类别，然后是词类别的类别 .....
  - 嵌套类别构成一棵树，其叶子为词
- 选择一个词的概率是由路径（从树根到包含该词叶子的路径）上的每个节点通向该词分支概率的乘积给出
- 使用逻辑回归计算每一个节点的条件概率
- 挑战：
  - 如何最好地定义这些词
- 缺点：
  - 树结构不能解决**在给定上下文中选择最可能的词**这个问题
  - 分层softmax 倾向于更差的测试结果（可能因为词类选择不好）
  - 计算所有的词概率的成本仍是很高的

自然语言处理

重要采样

- 以仅采样词的子集
- $$\begin{aligned}\frac{\partial \log P(y \mid C)}{\partial \theta} &= \frac{\partial \log \operatorname{softmax}_y(\boldsymbol{a})}{\partial \theta} \\ &= \frac{\partial}{\partial \theta} \log \frac{e^{a_y}}{\sum_i e^{a_i}} \\ &= \frac{\partial}{\partial \theta}(a_y - \log \sum_i e^{a_i}) \\ &= \frac{\partial a_y}{\partial \theta} - \sum_i P(y=i \mid C) \frac{\partial a_i}{\partial \theta}\end{aligned}$$
- 从模型中采样需要对词汇表中所有的i计算$P(i|C)$
- 可以从提议分布中取样，通过适当的权重校正从错误分布采样引入的偏差
- $$w_i = \frac{p_{n_i}/q_{n_i}}{\sum_{j=1}^{N} p_{n_j}/q_{n_j}}$$
- $$\sum_{i=1}^{|\mathbb{V}|} P(i \mid C) \frac{\partial a_i}{\partial \theta} \approx \frac{1}{m} \sum_{i=1}^{m} w_i \frac{\partial a_{n_i}}{\partial \theta}$$

自然语言处理

排名损失

- 正确词的得分a_y比其他词a_i排名更高
- $$L = \sum_i \max(0, 1 - a_y + a_i)$$
- 缺点：
  - 不提供条件概率

自然语言处理

神经机器翻译

- 以一种自然语言读取句子并产生等同含义的另一种语言的句子
  - 一个组件通常会提出许多候选翻译
  - 第二个组成部分（语言模型）评估提议的翻译
- 基于MLP 方法
  - 需要将序列预处理为固定长度
- RNN
  - 读取上下文C 并且生成目标语言的句子
- 注意力机制
  - 读取器读取原始数据，转化为分布式表示
  - 存储器存储读取器输出的特征向量列表
  - 利用存储器的内容顺序地执行任务，生成翻译语句

自然语言处理

神经机器翻译

- 注意力机制

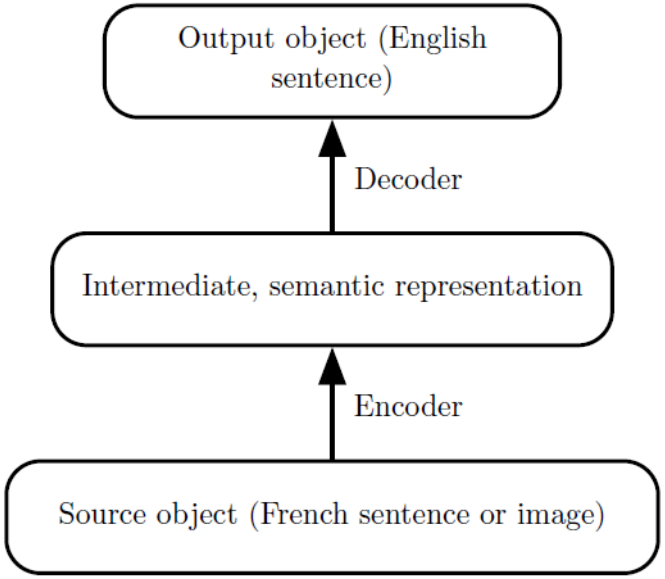


图 12.5: 编码器-解码器架构在直观表示（例如词序列或图像）和语义表示之间来回映射。使用来自一种模态数据的编码器输出（例如从法语句子到捕获句子含义的隐藏表示的编码器映射）作为用于另一模态的解码器输入（如解码器将捕获句子含义的隐藏表示映射到英语），我们可以训练将一种模态转换到另一种模态的系统。这个想法已经成功应用于很多领域，不仅仅是机器翻译，还包括为图像生成标题。

推荐系统

协同过滤推荐系统

- 泛化方式依赖于不同用户或不同项目的目标变量值之间的模式相似性
- 参数方法：目标变量的双线性预测
  - 通过用户嵌入和项目嵌入之间的点积获得预测
- 限制
  - 冷启动推荐问题

基于内容的推荐系统

- 引入单个用户和项目的额外信息解决冷启动
- 通过深度学习架构来学习从丰富的用户特征或项目特征集到嵌入的映射可以

推荐系统

探索与利用

- 利用
  - 从目前学到的最好策略采取动作，也就是我们所知的将获得高奖励的动作
- 探索
  - 采取行动以获得更多的训练数据
- 强化学习需要权衡探索与利用，最突出的权衡因素是**时间尺度**
  - 如果代理只有短暂的时间积累奖励，那么我们喜欢更多的利用。
  - 如果代理有很长时间积累奖励，那么我们开始更多的探索，以便使用更多的知识更有效地规划未来的动作
- 问题：
  - 难以评估和比较不同的策略

知识表示、推理和回答

知识库

- 存储数据集中实体之间的关系的信息
- 关系型数据库

应用：

链接预测

- 预测知识图谱中缺失的弧
- 很难评估链接预测任务上模型的性能

词义消歧

- 决定在某些语境中哪个词的意义是恰当

问答系统

- 能处理输入信息并记住重要的事实，并以之后能检索和推理的方式组织