18 序列模型

概要

- 序列模型
  - 马尔可夫（Markov）
  - 隐马尔可夫模型（HMM）

相关的随机变量

数据

- 至今 …
- 收集观察数据对 $(x_i, y_i) \sim p(x, y)$ 进行训练
- 对于看不见的$x' \sim p(x)$，估计 $y|x \sim p(y|x)$
- 例子：
  - 图像和目标对象
  - 回归问题
  - 房子和房价
- 数据的顺序无关紧要
  - 【其实是有关系的，空间和时间的顺序都很重要】

时序数据

- 实际中的很多数据有时序结构
- 电影的评价随时间变化而变化
  - 锚定（anchoring）效应：基于其他人的意见做出评价。拿奖后评分上升，直到奖项被忘记
  - 享乐适应（hedonic adaption）：人们迅速接受并且适应一种更好或者更坏的情况作为新的常态。看了很多好电影后, 人们的期望变高
  - 季节性：贺岁片、暑期档
  - 导演、演员的负面报道导致评分变低

序列数据

- 在使用应用程序时，许多用户都有很强的特定习惯
  - 在学生放学后社交媒体应用更受欢迎。在市场开放时股市交易软件更常用
- 预测明天的股价要比过去的股价更困难
  - 在统计学中，前者（对超出已知观测范围进行预测）称为外推法（extrapolation），而后者（在现有观测值之间进行估计）称为内插法（interpolation）
- 音乐、语音、文本和视频都是连续的。如果它们的序列被重排，那么会失去原有的意义
  - 文本标题“狗咬人”远没有“人咬狗”令人惊讶，尽管组成两句话的字完全相同
- 地震具有很强的相关性，即大地震发生后，很可能会有几次小余震
- 人类之间的互动是连续
  - 可以从微博上的争吵和辩论中看出

序列模型

统计工具

- 在时间 t 观察到 x_t, 那么得到 T 个不独立的随机变量

$$(x_1, \dots x_T) \sim p(\mathbf{x})$$

- 使用条件概率展开

$$p(a, b) = p(a)p(b \mid a) = p(b)p(a \mid b)$$

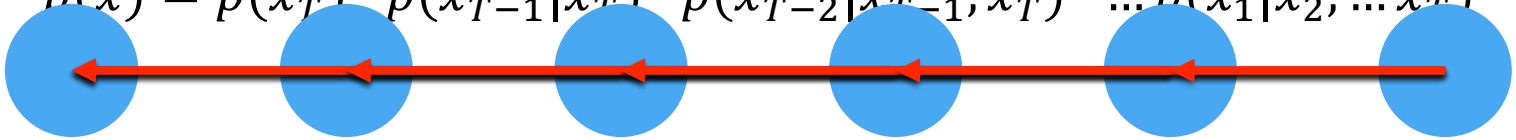
统计工具

- 前序

$$p(x) = p(x_1) \cdot p(x_2|x_1) \cdot p(x_3|x_1, x_2) \cdot \dots p(x_T|x_1, \dots x_{T-1})$$

- 反序

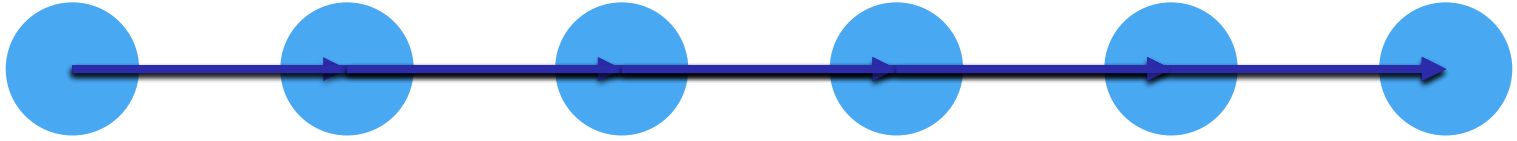
$$p(x) = p(x_T) \cdot p(x_{T-1}|x_T) \cdot p(x_{T-2}|x_{T-1}, x_T) \cdot \dots p(x_1|x_2, \dots x_T)$$



- 因果关系？

序列模型

- $p(x) = p(x_1) \cdot p(x_2|x_1) \cdot p(x_3|x_1, x_2) \cdot \dots p(x_T|x_1, \dots x_{T-1})$



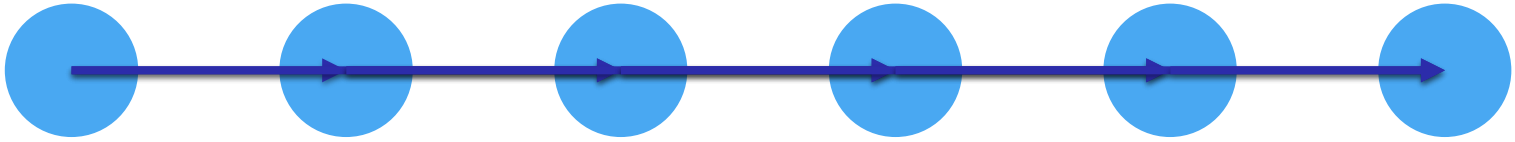
- 对条件概率建模

$$p(x_t|x_1, \dots x_{t-1}) = p(x_t|f(x_1, \dots x_{t-1}))$$

- 自回归模型
  - 对见过的数据建模

计划A - 马尔可夫（Markov）假设

- $p(x) = p(x_1) \cdot p(x_2|x_1) \cdot p(x_3|x_1, x_2) \cdot \dots p(x_T|x_{T-\tau}, \dots x_{T-1})$



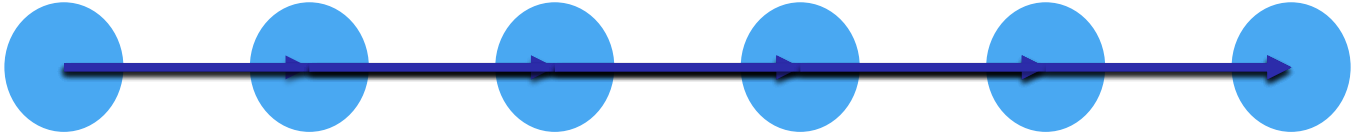
- 假设当前数据只和τ个过去数据相关

$$p(x_t \mid x_1, \dots x_{t-1}) = p(x_t \mid x_{t-\tau}, \dots x_{t-1}) = p\big(x_t \mid f(x_{t-\tau}, \dots x_{t-1})\big)$$

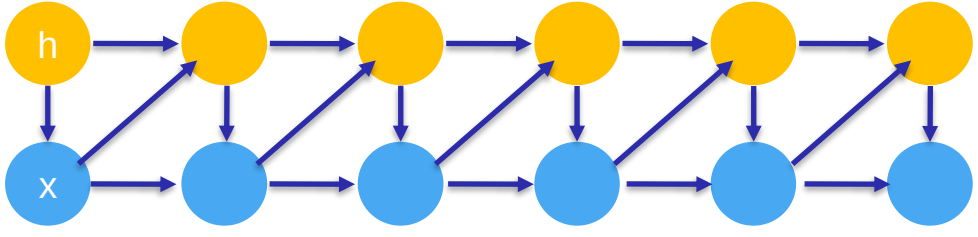
  - 可以在过去数据上训练一个MLP

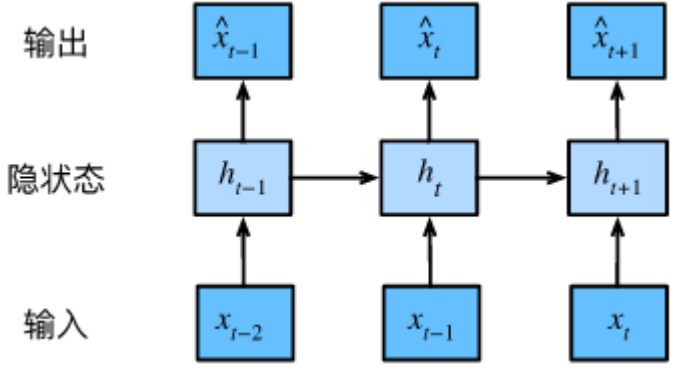
计划B – 潜变量模型

$$p(x) = p(x_1) \cdot p(x_2|x_1) \cdot p(x_3|x_1, x_2) \cdot \dots p(x_T|x_1, \dots x_{T-1})$$



- 引人潜变量 h_t 来表示过去信息 $h_t = f(x_1, \dots x_{t-1})$
  - 这样 $x_t = p(x_t \mid h_t)$





总结

- 时序模型中，当前数据跟之前观察到的数据相关
- 自回归模型使用自身过去数据来预测未来
- 马尔科夫模型假设当前只跟最近少数数据相关，从而简化模型
- 潜变量模型使用潜变量来概括历史信息

语言模型

语言模型

- 给定文本序列 $x_1, \ldots, x_T$，语言模型的目标是估计联合概率$P(x_1, x_2, \ldots, x_T)$
  - 常见计算方式：$P(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} P(x_t \mid x_1, \ldots, x_{t-1})$

- 它的应用包括
  - 做预训练模型 (eg BERT, GPT-3)
  - 生成本文，给定前面几个词，不断的使用$x_t \sim p(x_t \mid x_1, \ldots, x_{t-1})$ 来生成后续文本
  - 判断多个序列中哪个更常见
    - e.g. "to recognize speech” vs "to wreck a nice beach

使用计数来建模

- n是所有文本的总词数，$n(x), n(x,x')$是单个单词和连续单词对的出现次数
- 假设序列长度为1

$$p(x) = \frac{n(x)}{n}$$

- 假设序列长度为2，我们预测

$$p(x,x') = p(x)p(x' \mid x) = \frac{n(x)}{n}\frac{n(x,x')}{n(x)}$$

- 很容易拓展到长为 3 的情况

$$p(x,x',x'') = p(x)p(x' \mid x)p(x'' \mid x,x') = \frac{n(x)}{n}\frac{n(x,x')}{n(x)}\frac{n(x,x',x'')}{n(x,x')}$$

N元语法

- 当序列很长时, 文本量不够大, $n(x_1, \ldots, x_T) \leq 1$
- 使用马尔科夫假设缓解这个问题
  - 一元语法（马尔科夫假设的$\tau = 0$）

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2)p(x_3)p(x_4) = \frac{n(x_1)}{n}\frac{n(x_2)}{n}\frac{n(x_3)}{n}\frac{n(x_4)}{n}$$

$$= \frac{n(x_1)}{n}\frac{n(x_2)}{n}\frac{n(x_3)}{n}\frac{n(x_4)}{n}$$

  - 二元语法（马尔科夫假设的$\tau = 1$）

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2)p(x_4 \mid x_3)$$

$$= \frac{n(x_1)}{n}\frac{n(x_1, x_2)}{n(x_1)}\frac{n(x_2, x_3)}{n(x_2)}\frac{n(x_3, x_4)}{n(x_3)}$$

  - 三元语法（马尔科夫假设的$\tau = 2$）

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2)p(x_4 \mid x_2, x_3)$$

N-grams (更长的标记序列)

- 由于连续单词出现频率要低得多，估计这类单词正确的概率要困难得多
- 更平滑（长的 N-grams 序列很少）,执行某种形式的拉普拉斯平滑（Laplace smoothing）

$$\begin{aligned}
\hat{P}(x) &= \frac{n(x) + \epsilon_1/m}{n + \epsilon_1}, \\
\hat{P}(x' \mid x) &= \frac{n(x, x') + \epsilon_2 \hat{P}(x')}{n(x) + \epsilon_2}, \\
\hat{P}(x'' \mid x, x') &= \frac{n(x, x', x'') + \epsilon_3 \hat{P}(x'')}{n(x, x') + \epsilon_3}.
\end{aligned}$$

自然语言统计

- (最流行的词)无实际意义， 这些词通常可以被过滤掉，称为停用词（stop words）
- 词频衰减很快
  - 单词的频率满足齐普夫定律（Zipf‘s law），第 i 个最常用单词的频率 n_i 为:

$$n_i \propto \frac{1}{i^{\alpha}}$$

  - 等价于$\log n_i = -\alpha \log i + c$，其中 α 是刻画分布的指数， c 是常数

总结

- 语言模型估计文本序列的联合概率
- 使用统计方法时常采用n元语法

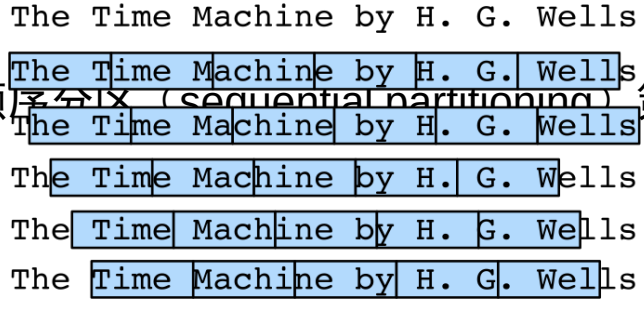
文本预处理

符号化

- 基本理念 - 将文本映射到 ID 序列
- 字符编码（每个字符都有一个 ID）
  - 小词汇量
  - 效果不好（DNN 需要学习拼写）
- 单词编码（每个单词有一个 ID）
  - 准确的拼写
  - 效果不好（巨大的词汇量 = 昂贵的多项式）
- 字节对编码（黄金区）
  - 频繁的子序列（如音节）

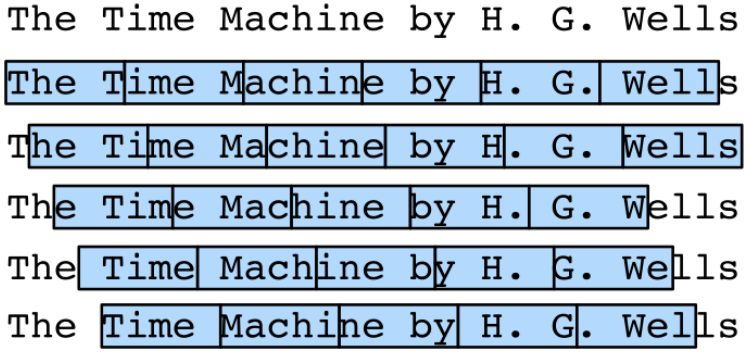
读取长序列数据

- 序列数据本质上是连续的。当序列变得太长而不能被模型一次性全部处理时， 我们可能希望拆分这样的序列方便模型读取
  - 假设模型中的网络一次处理具有预定义长度 的一个小批量序列。 现在的问题是如何随机生成一个小批量数据的特征和标签以供读取
- 从原始文本序列获得子序列的所有不同的方式。如何选择？
  - 事实上都一样好。但如只选择一个偏移量， 那么用于训练网络的子序列覆盖范围有限
  - 从随机偏移量开始划分序列， 以同时获得覆盖性（coverage）和随机性（randomness）
    - 随机采样（random sampling）和 顺序分区（sequential partitioning）策略



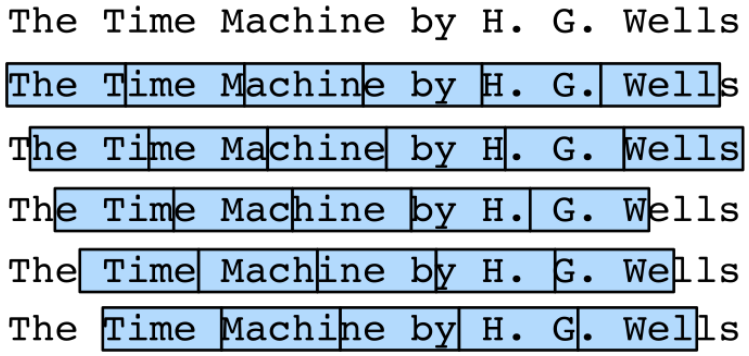
随机采样

- 随机分区
  - 选择随机偏移
  - 通过小批量随机分配序列
  - 样本独立性
  - 需要重置隐含状态



顺序分区

- 顺序分区
  - 选择随机偏移
  - 通过小批量按顺序分配序列
  - 相关样本
  - 保持小批量的隐含状态（更好）



代码 ...

总结

- 语言模型是自然语言处理的关键
- n元语法通过截断相关性，为处理长序列提供了一种实用的模型
- 长序列存在一个问题：它们很少出现或者从不出现
- 齐普夫定律支配着单词的分布，这个分布不仅适用于一元语法，还适用于其他n元语法
- 通过拉普拉斯平滑法可以有效地处理结构丰富而频率不足的低频词词组
- 读取长序列的主要方式是随机采样和顺序分区。在迭代过程中，后者可以保证来自两个相邻的小批量中的子序列在原始序列上也是相邻的