循环神经网络

一、 循环神经网络

1.1 序列模型

1.1.1 自回归模型

利用前期若干时刻的随机变量的线性组合来描述以后某时刻随机变量的线性回归模型。 可以表示为:

$$y_t = \varphi_0 + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + e_t \tag{1 - 1}$$

φ_0 ———— 常数项

φ_1~φ_p ———— 模型参数

e_t ———— 噪声

因为随着 x 与时间 t 不存在函数关系，所以相比于线性回归是用 x 预测 y，自回归模型是使用 x 预测后面的 x。

自然语言处理的输入输出基本上都是序列，序列问题是自然语言处理最本质的问题。

整个序列的估计值:

$$P(x_1, ..., x_\tau) = \prod_{t=1}^{\tau} P(x_t | x_{t-1}, ..., x_1) \tag{1 - 2}$$

1.1.2 马尔可夫模型

在已知目前状态（现在）的条件下，它未来的演变（将来）不依赖于它以往的演变 (过去)。例如森林中动物头数的变化构成——马尔可夫过程。在现实世界中，有很多过程都是马尔可夫过程,如液体中微粒所作的布朗运动、传染病受感染的人数、车站的候车人数等，都可视为马尔可夫过程。(这里虽然我也不清楚这些现象到底是不是，姑且就认为是吧!)

在马尔可夫性的定义中，"现在"是指固定的时刻，但实际问题中常需把马尔可夫性中的“现在”这个时刻概念推广为停时（见随机过程）。

在马尔科夫过程中，在给定当前知识或信息的情况下，过去（即当前以前的历史状态）对于预测将来（即当前以后的未来状态）是无关的。这种性质叫做无后效性。简单地说就是将来与过去无关，值与现在有关，不断向前形成这样一个过程。

马尔可夫模型可以写为：

$$P(x_1, ..., x_\tau) = \prod_{t=1}^{\tau} P(x_t \mid x_{t-1}) 当 P(x_1 \mid x_0) = P(x_1) \tag{1 - 3}$$

时间和状态都是离散的马尔可夫过程称为马尔可夫链,简记为Xn=X(n),n=0,1,2...马尔可夫链是随机变量X1,X2,X3...的一个数列。

这种离散的情况其实草是我们所讨论的重点，很多时候我们就直接说这样的离散情况就是一个马尔科夫模型。

1.1.3 因果关系

将$P(x_1, ..., x_\tau)$倒叙展开，基于条件概率公式：

$$P(x_1, ..., x_\tau) = \prod_{x_t}^{1} x_{t+1}, ..., x_\tau \tag{1 - 4}$$

1.2 文本预处理

1. 将文本作为字符串加载到内存中。
2. 将字符串拆分为词元（如单词和字符）。
3. 建立一个词表，将拆分的词元映射到数字索引。
4. 将文本转换为数字索引序列，方便模型操作。

将文本数据映射为词元，以及将这些词元可以视为一系列离散的观测，例如单词或字符。

1.3 语言模型和数据集

1.3.1 学习语言模型

基本概率规则：见前面章节

$$P(x_1, x_2, ..., x_\tau) = \prod_{t=1}^{\tau} P(x_t | x_1, ..., x_{t-1}) \tag{1 - 5}$$

这样一个有四个单词的文本序列就是：

$$\begin{gathered} P(\text{deep}, \text{learning}, \text{is}, \text{fun}) = P(\text{deep}) P(\text{learning} | \text{deep}) \\ P(\text{is} \mid \text{deep}, \text{learning}) \mid P(\text{fun} \mid \text{deep}, \text{learning}, \text{is}) \end{gathered} \tag{1 - 6}$$

可以写出：

$$\hat{P}(\text{learning} \mid \text{deep}) = \frac{n(\text{deep}, \text{learning})}{n(\text{deep})} \tag{1 - 7}$$

因为单词组合不一定会出现，所以 n 可能为 0，可以改为：

$$\hat{P}(x) = \frac{n(x) + \frac{\varepsilon_1}{m}}{n + \varepsilon_1} \tag{1 - 8}$$

$$\widehat{x' | x} = \frac{n(x, x') + \varepsilon_2 \hat{P}(x')}{n(x) + \varepsilon_2} \tag{1 - 9}$$

$$\hat{P}(x'' | x, x') = \frac{n(x, x', x'') + \varepsilon_3 \hat{P}(x'')}{n(x, x') + \varepsilon_3} \tag{1 - 10}$$

然而，这样的模型很容易变得无效，原因如下：首先，我们需要存储所有的计数；其次，这完全忽略了单词的意思。例如，“猫”（cat）和“猫科动物”（feline）可能出现在相关的上下文中，但是想根据上下文调整这类模型其实是相当困难的。

1.3.2 马尔可夫模型与 n 元语法

用于序列建模的近似公式：

$$P(x_1, x_2, x_3, x_4) = P(x_1) P(x_2) P(x_3) P(x_4) \tag{1 - 11}$$

$$P(x_1, x_2, x_3, x_4) = P(x_1) P(x_2 \mid x_1) P(x_3 \mid x_2) P(x_4 \mid x_3) \tag{1 - 12}$$

$$P(x_1, x_2, x_3, x_4) = P(x_1) P(x_2 \mid x_1) P(x_3 \mid x_1, x_2) P(x_4 \mid x_2, x_3) \tag{1 - 13}$$

1.4 循环神经网络

隐变量模型：

$$P(x_t | x_t - 1, ..., x_1) \approx P(x_t | h_{t-1}) \tag{1 - 14}$$

h_{t-1} ———— 隐状态

使用当前输入x_t和先前隐状态h_{t-1}来计算时间步t处的任何时间的隐状态：

$$h_t = f(x_t, h_{t-1}) \tag{1 - 15}$$

1.4.1 无隐状态的神经网络

可以表示为：

$$H = \varphi(\mathrm{X}\mathrm{W}_{\mathrm{xh}} + b_h) \tag{1 - 16}$$

$$O = \mathrm{H}\mathrm{W}_{\mathrm{hq}} + b_q \tag{1 - 17}$$

1.4.2 有隐状态的循环神经网络

当前时间步隐藏变量由当前时间步的输入 与前一个时间步的隐藏变量一起计算得出：

$$H_t = \varphi(X_t W_{\mathrm{xh}} + H_{t-1} W_{\mathrm{hh}} + b_h) \tag{1 - 18}$$

对于时间步 t，输出层的输出类似于多层感知机中的计算：

$$O_t = H_t W_{\mathrm{hq}} + b_q \tag{1 - 19}$$

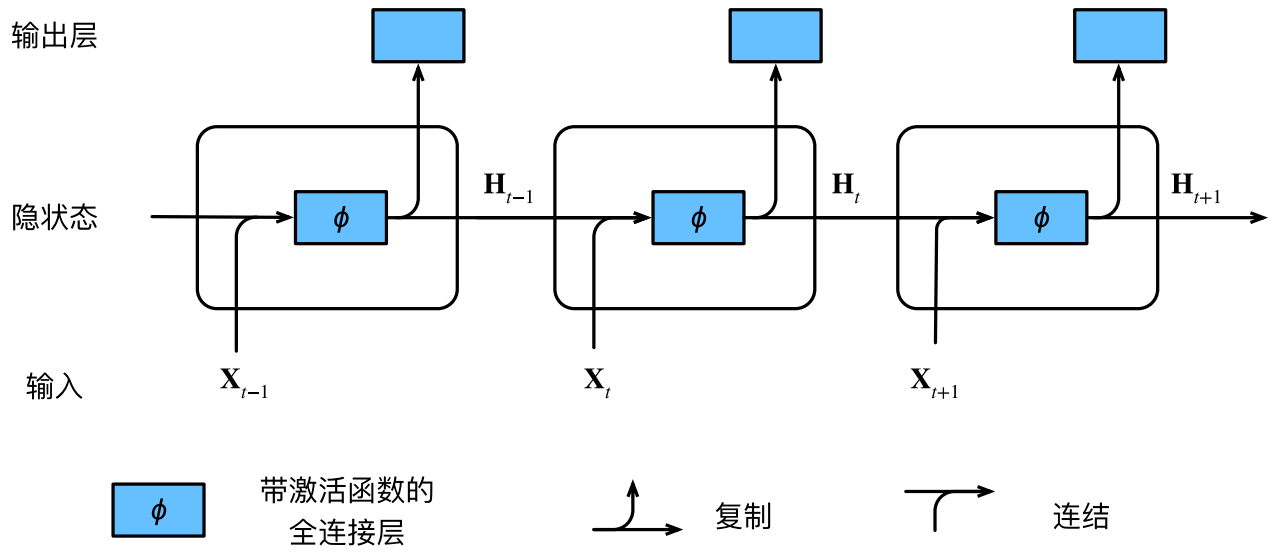

Figure 1 - 1: ResNet 结构

1.4.3 基于循环神经网络的字符级语言模型

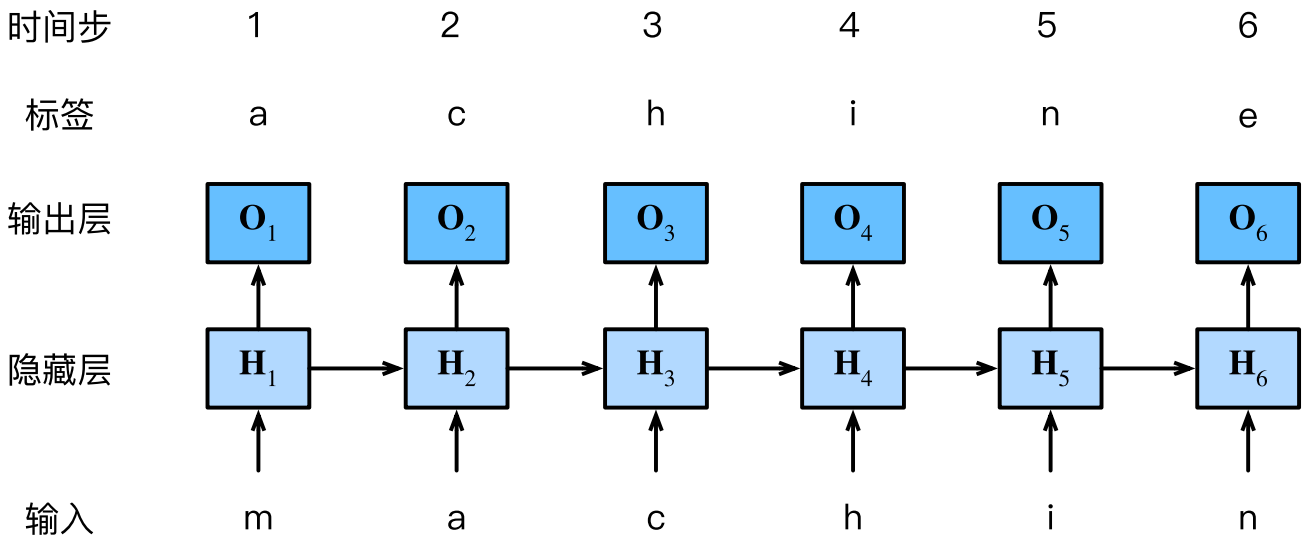

Figure 1 - 2: ResNet 结构

困惑度的最好的理解是“下一个词元的实际选择数的调和平均数”。

1.5 梯度剪裁

二、 引用

1. 《8. 循环神经网络 — 动手学深度学习 2.0.0 documentation》. 见于 2024 年 7 月 15 日. https://zh.d2l.ai/chapter_recurrent-neural-networks/index.html.
2. 《马尔科夫模型系列文章（一）——马尔科夫模型-CSDN 博客》. 见于 2024 年 7 月 15 日. https://blog.csdn.net/qq_27825451/article/details/100117715.
3. 《自回归模型（AR Model）-CSDN 博客》. 见于 2024 年 7 月 15 日. https://blog.csdn.net/shigangzwy/article/details/69525576.