



计算机工程
Computer Engineering
ISSN 1000-3428, CN 31-1289/TP

《计算机工程》网络首发论文

题目: 基于循环神经网络的推荐算法
作者: 高茂庭, 徐彬源
DOI: 10.19678/j.issn.1000-3428.0051294
网络首发日期: 2018-08-30
引用格式: 高茂庭, 徐彬源. 基于循环神经网络的推荐算法. 计算机工程.
<https://doi.org/10.19678/j.issn.1000-3428.0051294>



网络首发: 在编辑部工作流程中, 稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定, 且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式 (包括网络呈现版式) 排版后的稿件, 可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定; 学术研究成果具有创新性、科学性和先进性, 符合编辑部对刊文的录用要求, 不存在学术不端行为及其他侵权行为; 稿件内容应基本符合国家有关书刊编辑、出版的技术标准, 正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性, 录用定稿一经发布, 不得修改论文题目、作者、机构名称和学术内容, 只可基于编辑规范进行少量文字的修改。

出版确认: 纸质期刊编辑部通过与《中国学术期刊 (光盘版)》电子杂志社有限公司签约, 在《中国学术期刊 (网络版)》出版传播平台上创办与纸质期刊内容一致的网络版, 以单篇或整期出版形式, 在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊 (网络版)》是国家新闻出版广电总局批准的网络连续型出版物 (ISSN 2096-4188, CN 11-6037/Z), 所以签约期刊的网络版上网络首发论文视为正式出版。

基于循环神经网络的推荐算法

高茂庭, 徐彬源

(上海海事大学 信息工程学院, 上海 201306)

摘 要: 传统电影推荐算法大多基于用户和电影的静态属性进行推荐, 忽略了时间序列数据内在的时间和因果因素, 推荐质量有待提高。针对这个问题, 提出基于循环神经网络的推荐算法 (R-RNN), 利用循环神经网络 (RNN) 在处理时间序列上的优势, 用两个 RNN 分别挖掘用户和电影的时间状态, 使用 LSTM 单元实现更长距离的历史状态积累, 将用户状态和电影状态的内积作为最终评分。在三个公开数据集上的实验表明, 与基于概率矩阵分解、TimeSVD++、AutoRec 算法相比, 该算法能够有效降低均方根误差, 并提高预测评分的准确度。

关键词: 推荐算法; 循环神经网络; 长短期记忆; 时间动态; 潜在状态

Recommendation Algorithm Based on Recurrent Neural Network

GaoMaoting, XuBinyuan

(Collage of Information Engineering, Shanghai Maritime University, Shanghai 201306, China)

【Abstract】 Traditional recommendation algorithms is based on the static attributes of users and movies, and ignore the temporal and causal relationship inherent in the data. To solve this problem, a new recommendation algorithm based on recurrent neural network is proposed by using the advantages of recurrent neural network in processing time series. Two RNNs are used to deeply mine the potential status of users and films in time series, LSTM units are used to realize longer historical status accumulation, and the prediction rating is obtained by the product of the user status and the movie status. Experiments on three public datasets show that the new algorithm can effectively reduce the root mean square error and improve the accuracy of the rating prediction.

【Key words】 Recommender algorithm; recurrent neural network; long-short term memory; time dynamics; potential state
DOI:10.19678/j.issn.1000-3428.0051294

1 引言

电影领域中现有大多数推荐方法是围绕这样一个问题展开研究的^[1], 即给定一组由用户、电影、时间戳和评分组成的元组, 目标是为用户、电影、时间戳这三个属性的选择性组合预测评分, 然后通过预测评分与实际评分的偏差来衡量推荐质量。基于这些组合, 后来的研究者提出不同的推荐算法^[2~5], 如基于概率矩阵分解^[6], 最近邻^[7]和聚类^[8]方法等。

这些方法在一定程度上缓解了评分矩阵的稀疏性问题, 但传统方法认为用户和电影的属性是固定的, 没有考虑到时间序列数据中固有的时间和因果关系。时间序列数据是一种数据属性随时间变化的高维数据类型, 反映了用户状态 (如用户兴趣) 和电影状态 (如电影流行度) 的动态变化。例如在日常生活中, 用户的喜好会随思想的成熟或生活方式

的变化而变化, 对特定类型节目的兴趣也会有所减弱或加强。除了需要对时间动态建模之外, 还需要克服用后见之明的方式预测评分, 达到满足因果关系的预测。传统推荐算法没有明确区分用户评分的时间先后顺序, 甚至使用未来评分估计当前评分, 这在统计分析中违反了因果关系。由于受到训练和测试数据混合分布的阻碍, 模型在未来预测方面遇到了挑战。研究显示, 通过建立时序模型, 利用时间序列数据, 能够根据当前趋势更准确地预测未来行为。

文献[9]重点研究基于协同过滤的推荐系统中的时间效应建模问题, 分析时间特征如何影响协同过滤, 认为用户随时间改变他的评价习惯和偏好, 电影的流行度也在随时间而变化, 综合考虑这些时间效应, 提出了 TimeSVD++模型。但该模型不能试图预测未来的行为。

文献[10]建立一种基于动态时间的个性化推荐

基金项目: 国家自然科学基金 (61202022)

作者简介: 高茂庭 (1963—), 男, 江西九江人, 教授, 博士, 主要研究方向为智能信息处理、数据库与信息系统 (mtgao@163.com); 徐彬源 (1993—), 男, 河南信阳人, 硕士研究生, 主要研究方向为数据挖掘、数据分析

E-mail: 13849723913@163.com

模型,引入遗忘曲线来跟踪用户对资源偏好程度随时间变化情况,能获取准确的用户偏好信息,提高推荐质量。文献[11]在矩阵因子分解的技术上,考虑时间效应信息以更好地描述用户的行为,提出了基于推荐系统时间敏感的因子模型算法。上述算法利用时序数据在一定程度上提高了预测精度,但只分析了用户的时间状态,缺乏对物品时间状态的挖掘。

为了准确获取用户和电影的时间状态,使用循环神经网络(RNN)[12,13]。为了解决RNN中的梯度消失问题,在模型中引入了长短期记忆(Long-Short Term Memory)[14]单元。从而提出基于循环神经网络的推荐算法,考虑到时间序列数据和现实中的因果关系,严格按照评分发生的时间顺序划分数据集,然后用两个LSTM模型分别获取用户和电影的时间状态,根据用户和电影的历史状态预测下一时刻的评分行为,并在多个公开数据集上通过与传统模型的比较分析,验证该模型预测评分的有效性和准确度。

2 循环神经网络

2.1 简单的循环神经网络

循环神经网络特殊的结构使其特别适用于处理各种时序数据,通过网络中的循环结构记录时序数据的历史信息。与普通神经网络的区别在于其隐藏层之间是连接起来的,即当前时刻的输出不仅与当前输入有关,还依赖于上一时刻隐藏层的输出,RNN模型如图1所示。

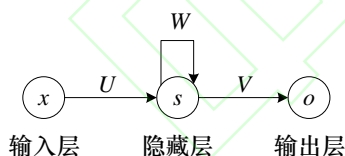


图1 RNN 结构图

图1中,向量 x 表示输入数据;向量 s 表示隐藏层的值;向量 o 表示输出层的值; U 是输入层到隐藏层的权重矩阵; V 是隐藏层到输出层的权重矩阵。循环神经网络隐藏层的值 s 不仅取决于当前这次的输入 x ,还依赖于上一次隐藏层的值 s 。权重矩阵 W 是隐藏层上一次的值作为这一次的输入的权重矩阵。

将循环神经网络按照时间线展开,如图2所示。

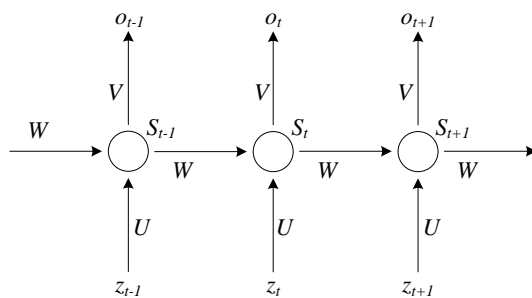


图2 RNN 时间线展开图

图2展开的网络在 t 时刻, z_t 是输入层的输入, f_1 是隐藏层激活函数, s_t 是隐藏层的输出, g_1 是输出层激活函数, o_t 是输出层的输出。关键一点是, s_t 的值不仅取决于 z_t ,还取决于 s_{t-1} 。循环神经网络的前向计算过程用公式表示如下:

$$o_t = g_1(V \cdot s_t) \quad (1)$$

$$s_t = f_1(U \cdot z_t + W \cdot s_{t-1}) \quad (2)$$

通过上述两个公式的循环迭代,有以下推导:

$$\begin{aligned} o_t &= g_1(Vs_t) \\ &= g_1(Vf_1(Uz_t + Ws_{t-1})) \\ &= g_1(Vf_1(Uz_t + Wf_1(Uz_{t-1} + Ws_{t-2}))) \\ &= g_1(Vf_1(Uz_t + Wf_1(Uz_{t-1} + Wf_1(Uz_{t-2} + \dots)))) \end{aligned} \quad (3)$$

可以看到,当前时刻包含了历史信息,这说明循环神经网络能够记忆历史信息。但传统的RNN总是用“覆写”的方式计算状态,即 $s_t = f_1(U \cdot z_t + W \cdot s_{t-1})$,根据求导的链式法则,这种形式直接导致梯度被表示为连乘积的形式,造成难以解决的梯度消失问题。直到1997年,Sepp Hochreiter和Jurgen Schmidhuber提出长短期记忆(LSTM)模型来解决循环神经网络中的梯度消失问题。

2.2 长短期记忆网络

本文采用长短期记忆(LSTM)模型[14]来解决稳定性和梯度消失的问题。在这个模型中,常规的神经元被存储单元代替,每个存储单元由输入门、输出门、自有状态组成。存储单元中管理向单元移除或添加的结构叫门限,有三种:遗忘门、输入门、输出门,门限由sigmoid激活函数和逐点乘法运算组成。前一个时间步长的隐藏状态被送到遗忘门、输入门和输出门。在前向计算过程中,输入门学习何时激活让当前输入传入存储单元,而输出门学习

何时激活让当前隐藏层状态传出存储单元。LSTM 结构如图 3 所示。

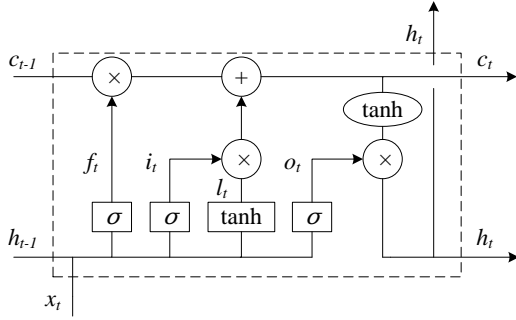


图 3 LSTM 结构图

状态更新满足以下：

$$f_t = \sigma(W^f [h_{t-1}, z_t] + z^f) \quad (4)$$

$$i_t = \sigma(W^i [h_{t-1}, z_t] + b^i) \quad (5)$$

$$o_t = \sigma(W^o [h_{t-1}, z_t] + b^o) \quad (6)$$

$$l_t = \tanh(W^l [h_{t-1}, z_t] + b^l) \quad (7)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot l_t \quad (8)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (9)$$

其中， f_t ， i_t 和 o_t 分别表示忘记门，输入门和输出门，控制着信息如何流经这个序列， l_t 表示当前输入 z_t 生成的候选数值， c_t 表示隐藏层神经元状态， $[h_{t-1}, z_t]$ 表示 $t-1$ 时刻隐藏层的输出和 t 时刻的输入的连接向量。 W^f 和 b^f 表示忘记门的权重矩阵和偏置， W^i 和 b^i 表示输入门的权重矩阵和偏置， W^o 和 b^o 表示输出门的权重矩阵和偏置， W^l 和 b^l 表示当前时刻输入数据生成当前单元状态值的权重矩阵和偏置。在下面的章节中，本文使用 $h_t = \text{LSTM}(h_{t-1}, z_t)$ 来表示这些操作。

3 基于循环神经网络的推荐模型

假设基于循环神经网络的推荐模型 (R-RNN) 分别解决用户和电影状态的时间演变。用户的状态演变取决于用户先前评价过的电影。同样，电影的状态演变依赖于在前一时间间隔中对其进行评分的用户。

本文使用 LSTM 来捕获用户和电影的时间动态。将用户对不同电影的评分按照时间戳进行划分作为用户向量输入，对电影也采用类似的做法。但

用户向量和电影向量都是高维稀疏的，不利于模型的训练。为了有效地处理高维稀疏的输入向量，使用自编码器的编码网络对序列数据进行预训练^[15]，以达到数据降维的目的，更利于模型的训练。通过这种方式，整合过去的观察数据并以综合的方式预测未来的轨迹，即将用户状态和电影状态做内积以实现评分预测。完整模型的结构如图 4 所示。

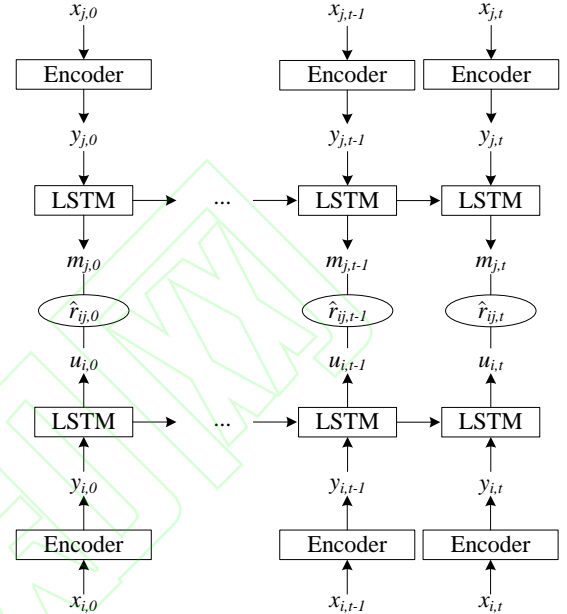


图 4 基于循环神经网络的推荐模型 (R-RNN)

图 4 中，为了处理时间动态，添加两个时间索引，即 $u_{i,t}$ 和 $m_{j,t}$ 分别表示用户 i 和电影 j 在时间步长 t 处的潜在状态。现在只需要定义更新函数，根据用户和电影历史状态更新最新状态，并预测评分：

$$\hat{r}_{ij,t} = f(u_{i,t}, m_{j,t}) \quad (10)$$

$$u_{i,t+1} = g(u_{i,t}, \{r_{ij,t}\}) \quad (11)$$

$$m_{j,t+1} = h(m_{j,t}, \{r_{ij,t}\}) \quad (12)$$

其中， $\hat{r}_{ij,t}$ 表示在时间步长 t 处用户 i 对电影 j 的预测评分， $r_{ij,t-1}$ 是在时间步长 $t-1$ 处的实际评分。学习函数 f ， g 和 h ，以便直接推断新用户的状态，而不需要进一步进行优化。本文解决优化问题来找到查找用户参数的函数，而不是解决优化问题来查找用户参数。这方面与[16]的自动编码器类似，学习过去评分的编码函数。创新之处在于，学习一个函数，它可以按顺序更新分数，并且能够一次对一组评分进行前向预测。

3.1 用户和电影状态

给定 M 个电影, N 个用户的数据集, $\mathbf{x}_{i,t} \in \mathbf{R}^M$ ($\mathbf{x}_{j,t} \in \mathbf{R}^N$) 表示给定用户 i (或电影 j) 在时间 t 的评分向量, $\mathbf{x}_{i,t}^j = k$ 表示给定用户 i 在时间步长 t 对电影 j 的评分为 k , $\mathbf{x}_{j,t}^i = k$ 表示给定电影 j 在时间步长 t 处被用户 i 的评分为 k , 否则对应元素值为 0, 则用户序列可以表示为 $\{\mathbf{x}_{i,0}, \mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,t-1}, \mathbf{x}_{i,t}\}$, 电影序列表示为 $\{\mathbf{x}_{j,0}, \mathbf{x}_{j,1}, \dots, \mathbf{x}_{j,t-1}, \mathbf{x}_{j,t}\}$, 但是组成这些序列的用户评分向量和电影评分向量是高维稀疏的, 直接用于 LSTM 网络的输入, 会导致模型根本无法训练, 需要对其进行数据降维处理。因此, 使用自编码器网络分别对用户向量和电影向量做预训练, 并分离出其中的编码网络以实现数据降维。

在自编码器对高维稀疏的用户向量和电影向量进行预训练的过程中, 分为编码网络和解码网络两个部分, 编码网络负责数据压缩, 解码网络负责数据恢复。解码网络的层数和编码网络的层数相同, 编码网络中各层的神经元个数逐层减少, 以捕捉高维输入数据中最重要的信息, 并实现数据降维。使用下列公式表示自编码器对用户向量和电影向量的编码过程:

$$\mathbf{y}_{i,t} = \mathbf{W}^{\text{uencoder}} [\mathbf{x}_{i,t}, \tau_t, \tau_{t-1}] \quad (13)$$

$$\mathbf{y}_{j,t} = \mathbf{W}^{\text{mencoder}} [\mathbf{x}_{j,t}, \tau_t, \tau_{t-1}] \quad (14)$$

其中, $\mathbf{W}^{\text{uencoder}}$ 和 $\mathbf{W}^{\text{mencoder}}$ 分别表示使用自编码器将用户和电影的高维稀疏向量向低维稠密向量的转换, τ_t 表示 wallclock。 $\mathbf{y}_{i,t}$ 和 $\mathbf{y}_{j,t}$ 是经过自编码器的编码过程后生成的低维稠密向量, 可以作为 LSTM 的直接输入, 以进一步得到用户时间状态 $\mathbf{u}_{i,t}$ 和电影时间状态 $\mathbf{m}_{j,t}$ 。于是得到以下模型:

$$\mathbf{u}_{i,t} = \text{LSTM}(\mathbf{u}_{i,t-1}, \mathbf{y}_{i,t}) \quad (15)$$

$$\mathbf{m}_{j,t} = \text{LSTM}(\mathbf{m}_{j,t-1}, \mathbf{y}_{j,t}) \quad (16)$$

其中, $\mathbf{u}_{i,t}$ 表示在时间步长 t 处用户 i 的状态, $\mathbf{m}_{j,t}$ 表示在时间步长 t 处电影 j 的状态。

3.2 评分预测

R-RNN 模型根据用户状态和电影状态的内积作为评分的估计, 计算公式如下所示:

$$\hat{r}_{ij,t} = \langle \mathbf{u}_{i,t}, \mathbf{m}_{j,t} \rangle \quad (17)$$

其中, $\langle \cdot \rangle$ 表示两个向量之间的内积。这种方法类似于潜在变量模型预测评分的方式, 如基于矩阵分解的推荐通过将原始评分矩阵分解成用户-特征矩阵和特征-项目矩阵, 得到用户和电影的潜在特征向量, 然后再利用两个矩阵还原原始评分矩阵, 并得到原始矩阵中的缺失评分。不同之处在于, R-RNN 模型严格按照评分时间戳划分数据集, 并克服后见之明的方式预测评分, 达到满足因果关系的预测, 结果更具说服力。

3.3 模型训练

不同于传统推荐算法, 在预测时间内, R-RNN 将最新的用户向量和电影向量作为输入, 更新状态并基于最新更新的状态进行预测。通过这种方式, 将过去评分带来的因果效应考虑在内。模型的优化目标是找到产生接近实际评分的预测评分的参数, 用 θ 表示模型中所有需要学习的参数, 即 $\mathbf{W}^{\text{uencoder}}$ 、 $\mathbf{W}^{\text{mencoder}}$ 、 \mathbf{W}^f 、 \mathbf{b}^f 、 \mathbf{W}^l 、 \mathbf{b}^l 、 \mathbf{W}^o 、 \mathbf{b}^o 、 \mathbf{W}^l 、 \mathbf{b}^l 。因此, R-RNN 的目标函数如下所示:

$$\min_{\theta} \sum_{(i,j,t) \in \mathcal{I}} \frac{1}{2} (r_{ij,t} - \hat{r}_{ij,t}(\theta))^2 + R(\theta) \quad (18)$$

其中, $\mathcal{I}_{\text{train}}$ 表示训练集中观察到的 (用户, 电影, 时间戳) 元组的集合, R 是关于参数的 L2 正则化项, 以防止模型过拟合的问题。模型主要求解步骤如下所示。

- 1) 使用 BP 算法[17]更新自编码器的权重参数 $\mathbf{W}^{\text{uencoder}}$ 、 $\mathbf{W}^{\text{mencoder}}$
 - a) 进行前馈传导计算, 得到自编码器各输出层的值。
 - b) 计算输出层以及各隐藏层的传播误差。
 - c) 计算最终各层权重参数的偏导数值。
 - d) 使用梯度下降法求得自编码器参数的最优解。
- 2) 使用 BPTT 算法更新 LSTM 单元中的权重参数 \mathbf{W}^f 、 \mathbf{b}^f 、 \mathbf{W}^l 、 \mathbf{b}^l 、 \mathbf{W}^o 、 \mathbf{b}^o 、 \mathbf{W}^l 、 \mathbf{b}^l
 - a) 依次前向计算各个控制门 f_t 、 i_t 、 o_t 的输出, 状态 c_t 的更新, 输出 h_t 的计算。
 - b) 分别计算输出门 o_t 、遗忘门 f_t 、输入门 i_t 以及细胞状态 c_t 的传播误差。

- c) 计算最终各层权重参数的梯度。
- d) 使用 Adam 算法求 LSTM 各参数最优解。

4 实验结果与分析

4.1 数据集描述

为了研究时间动态的有效性建模,在 Netflix 比赛和 IMDB 数据集上评估 R-RNN 模型。IMDB 数据集包含 1998 年 7 月至 2013 年 9 月期间收集的 1429600 个评分,完整 Netflix 数据集包含 1988 年 11 月至 2005 年 12 月期间收集的 100400000 个评分,其中 6 个月 Netflix 数据包含 2005 年 6 月至 2005 年 12 月期间收集的 15800000 个评分。每个数据点是一个时间戳粒度为 1 天的(用户 id,项目 id,时间戳,评分)元组。根据时间分割数据以模拟需要预测未来评分的实际情况。整个数据集都被分割成训练集和测试集。详细数据如表 1 所示。

表 1 实验数据集的详细数据

	IMDB	6 个月 Netflix	完整 Netflix
用户个数	440800	311300	477400
项目个数	114300	17700	17800
训练集评分个数	1400000	13700000	98100000
测试集评分个数	29600	2100000	2300000
训练集时间跨度	7/98-12/12	6/05-11/05	12/99-11/05
测试集时间跨度	1/13-9/13	12/05	12/05
稀疏性	0.0028%	0.28%	1.2%

4.2 实验设置

均方根误差 (root mean squared error, RMSE) 和平均绝对误差 (mean absolute error, MAE) 均是度量预测误差的指标,其中, RMSE 对大误差比较敏感, MAE 对小误差的积累比较敏感。由于 RMSE 加大了对预测不准的用户物品评分的惩罚,对系统的测评更加苛刻,实验中选择 RMSE 评估算法精度。

对于完整的 Netflix 和 IMDB 数据集,采用 2 个月的时间步长粒度,对于六个月的 Netflix 数据集,采用 7 天的时间步长粒度。每个时期之后,计算 RMSE。在验证集上给出最佳结果的模型上,计算在测试集上的 RMSE。

本文实验分为确定模型参数的实验和算法对比实验两种。实验 1 为自编码器编码网络的层数及其节点数的确定;实验 2 为隐藏层和细胞状态维度的确定;实验 3 为算法对比实验,通过对比不同方法

(包括 TimeSVD++、AutoRec、PMF) 预测评分的误差值 RMSE,验证本文提出算法的有效性,在预测精度上优于所有对比算法。上述实验是在 Tensorflow^[18]开发框架上完成的。

4.3 结果分析

实验 1 是利用自编码器对原始序列数据进行降维,实验中对不同层数不同神经元个数的自编码器网络进行测试。首先测试具有一层隐藏层的自编码器,节点数包括 200、400、600、800、1000,用 RMSE 测试自编码器的性能,表 2 列出了一层隐藏层自编码器不同数据集在不同隐藏层节点数下的 RMSE。

表 2 一层隐藏层自编码器的隐藏层节点数与 RMSE 的关系

	200	400	600	800	1000
IMDB	1.3526	1.3052	1.2690	1.2462	1.1995
Netflix full	1.1635	1.1180	1.0892	1.0590	1.0306
Netflix 6 months	1.2356	1.1875	1.1567	1.1249	1.0939

由表 2 可知,当隐藏层节点数为 1000 时, RMSE 值最小。可以看出,一层隐藏层自编码器的表现不是很好,接下来测试三层隐藏层的自编码器的性能。其中第一层和第三层隐藏层节点数设置为 1000,第二层节点数包括 100、150、200、250、300,表 3 列出了三层隐藏层自编码器不同数据集在不同隐藏层节点数下的 RMSE。

表 3 三层隐藏层自编码器的隐藏层节点数与 RMSE 的关系

	100	150	200	250	300
IMDB	0.9927	0.9613	0.9315	0.9101	0.8907
Netflix full	0.8802	0.8586	0.8329	0.8096	0.7963
Netflix 6 months	0.9225	0.8936	0.8673	0.8431	0.8215

由表 3 可知,当第二层隐藏层的节点数为 300 时, RMSE 值最小。接下来测试五层自编码器的性能,其中第一层和第五层隐藏层节点数设置为 1000,第二层和第四层隐藏层节点数设置为 300,第三层隐藏层节点数包括 10、20、30、40、50,表 4 列出了五层自编码器不同数据集在不同隐藏层节点数下的 RMSE。

表 4 五层隐藏层自编码器的隐藏层节点数与 RMSE 的关系

	10	20	30	40	50
IMDB	0.8629	0.8137	0.7836	0.7669	0.7691
Netflix full	0.7724	0.7358	0.7102	0.6813	0.6831
Netflix 6 months	0.8032	0.7580	0.7352	0.7132	0.7184

由表4可知,当第三层隐藏层的节点数为40时, RMSE 值最小。由上述实验确定了自编码器的层数及节点数,将其中的编码网络分离出来以实现数据降维,编码网络的层数为3层,每层节点数分别为1000,300,40。

实验2是LSTM单元中隐藏层和细胞状态节点数的确定,表5列出了不同数据集在不同隐藏层和细胞状态的节点数下的RMSE。

表5 LSTM单元隐藏层和细胞状态的节点数与RMSE关系

	10	20	30	40	50
IMDB	2.8124	1.9703	2.2647	2.7145	3.3817
Netflix full	1.6835	0.9224	1.1876	1.5783	2.1608
Netflix 6 months	1.8024	0.9427	1.2670	1.7152	2.3805

由表5可知,当节点数为20时,RMSE值最小。因此,LSTM单元隐藏层和细胞状态的节点数设置为20。

实验3是本章提出的基于LSTM网络的推荐算法与传统推荐算法的对比实验,包括基于概率矩阵分解(PMF)的推荐算法、基于TimeSVD++的推荐算法、基于自编码器的推荐算法(I-AutoRec和U-AutoRec),通过实验对比验证R-RNN模型在预测评分上的优越性。表6出了各模型在不同数据集上的对比实验结果。

表6 各模型在数据集上的RMSE

	PMF	I-AR	U-AR	TimeSVD++	R-RNN
IMDB	2.3913	2.0521	2.0290	2.0037	1.9703
Netflix full	0.9252	0.9364	0.9647	0.9275	0.9224
Netflix 6 months	0.9584	0.9778	0.9836	0.9589	0.9427

从表6以看出,R-RNN模型在所有比较方法中的所有数据集上实现了最佳的准确性。与PMF相比,R-RNN在IMDB上提高了1.7%,在6个月Netflix数据集上提高了1.6%。注意,表2中列出的RMSE高于Netflix比赛中的RMSE,因为本文测试了纯粹的未来评级。

虽然其他方法在不同数据集之间交换排名,但是R-RNN在所有数据集上的RMSE都是最小的,即推荐质量最好,说明该模型具有很好的健壮性,适用于不同的数据集。另外,PMF在14年IMDB数据集上的表现比TimeSVD++差太多,这再次强调了对时间动态建模的需求。

5 结束语

随着深度学习的快速发展,出现很多不同的神经网络模型,如在时间序列的处理和预测上有突破性进展的循环神经网络(RNN)。本文采用了RNN的一个变种LSTM,用随时间变化的用户和电影状态生成预测评分。实验证明,本文提出的R-RNN模型能够有效提高评分预测的准确度。

本文尝试将循环神经网络用于推荐,取得不错的效果。深度学习中还有很多其他的模型可用于推荐任务,如多层感知机、自编码器、卷积神经网络、深度语义相似性模型、生成对抗网络等。未来可以将不同的模型融合,整合不同的数据源以提高推荐精度。

参考文献:

- [1] J.Bennett and S.Lanning. The netflix prize. In Proceedings of KDD Cup and Workshop, volume 2007, page 35, 2007.
- [2] Zhao J, Li Y, Zhang Y. A Recommendation Algorithm via Developed Random Walk Takes User's Preference and Item's Genres[C]// International Conference on Information System and Data Mining. ACM, 2017:61-65.
- [3] Li M, Wang X J. An Ensemble Algorithm Used in Video Recommendation System[J]. In Proceedings of the 2017 International Conference on Big Data Research. ACM, 2017:67-71.
- [4] Karatzoglou A, Hidasi B. Deep Learning for Recommender Systems[C]// the Eleventh ACM Conference. ACM, 2017:396-397.
- [5] 李卫疆, 齐静, 余正涛,等. 融合信任传播和奇异值分解的社会化推荐算法[J]. 计算机工程, 2017, 43(08):236-242.
- [6] R.Salakhutdinov and A.Mnih. Probablistic matrix factorization. In NIPS, volume 20, 2008.
- [7] 王锦坤, 姜元春, 孙见山, 孙春华. 考虑用户活跃度和项目流行度的基于项目最近邻的协同过滤算法[J]. 计算机科学,2016.12:158-162.
- [8] A.Beutel, A.Ahmed, and A.J.Smola. ACCAMS: Additive co-clustering to approximate matrices succinctly. In WWW, pages 119-129. ACM, 2015
- [9] Y. Koren. Collaborative filtering with temporal dynamics. In Knowledge discovery and data mining KDD, pages 447-456, 2009.

- [10] 谭黎立, 聂瑞华, 梁军, 王进宏. 基于动态时间的个性化推荐模型[J]. 华南师范大学学报. 2017.03
- [11] 李忠武, 王辉, 魏再超. 基于推荐系统时间敏感的因子模型算法研究[J]. 电子商务. 2017.09
- [12] DU K, SWAMY M N S. Recurrent neural networks [M]. London: Springer, 2014: 337-353.
- [13] 俞骋超. 基于深度神经网络的用户会话推荐算法研究[D]. 浙江大学, 2016.
- [14] 陆泽楠, 商玉林. 基于 LSTM 神经网络模型的钢铁价格预测[J]. 科技视界, 2017.13:116-117
- [15] Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. Relation classification via convolutional deep neural network. In Proceedings of Coling, pages 2335–2344, 2014.
- [16] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie. Autorec: Autoencoders meet collaborative filtering. In WWW Companion, pages 111-112, 2015.
- [17] 罗频捷, 温荷. 基于改进 BP 神经网络的个性化推荐算法研究[J]. 四川理工学院学报(自然科学版), 2016, 29(1):39-43.
- [18] 章敏敏, 徐和平, 王晓洁, 等. 谷歌 TensorFlow 机器学习框架及应用[J]. 微型机与应用, 2017, 36(10):58-60.