

# 基于改进自编码器的在线课程推荐模型<sup>①</sup>



宋晓丽<sup>1</sup>, 贺龙威<sup>2</sup>

<sup>1</sup>(中国石油大学(华东), 青岛 266580)

<sup>2</sup>(海洋石油工程(青岛)有限公司, 青岛 266520)

通信作者: 宋晓丽, E-mail: songxiaoli@upc.edu.cn

**摘要:** 随着互联网技术的发展以及 2020 年新冠疫情的爆发, 越来越多的学生选择在线教育. 然而在线课程数量庞大, 往往无法及时找到合适的课程, 个性化智能推荐系统是解决这一问题的有效方案. 本文根据用户在线学习具有明显时序性的特点, 提出一种基于改进自编码器的在线课程推荐模型. 首先, 利用长短期记忆网络改进自编码器, 使得模型可以提取数据的时序性特征; 然后, 利用 Softmax 函数进行课程的推荐. 实验结果表明, 所提方法与协同过滤算法和基于传统自编码器的推荐模型相比, 具有更高的推荐准确率.

**关键词:** 深度学习; 数据挖掘; 自编码器; 个性化推荐

引用格式: 宋晓丽, 贺龙威. 基于改进自编码器的在线课程推荐模型. 计算机系统应用, 2022, 31(3): 288-293. <http://www.c-s-a.org.cn/1003-3254/8356.html>

## Online Course Recommendation Model Based on Enhanced Auto-encoder

SONG Xiao-Li<sup>1</sup>, HE Long-Wei<sup>2</sup>

<sup>1</sup>(China University of Petroleum, Qingdao 266580, China)

<sup>2</sup>(Offshore Oil Engineering (Qingdao) Co. Ltd., Qingdao 266520, China)

**Abstract:** With the development of Internet technology and the outbreak of COVID-19 in 2020, more and more students have chosen online education. However, due to the large number of online courses, students are often unable to find suitable courses in time. A personalized intelligent recommendation system is an effective solution to this problem. Considering the obvious sequential characteristics of users for online learning, an online course recommendation model based on enhanced auto-encoders is proposed. First, the auto-encoder is enhanced with the long short-term memory network, so the model can extract the sequential characteristics of data. Then, the Softmax function is used to recommend online courses. Experimental results show that the proposed method has higher recommendation accuracy than the collaborative filtering algorithm and the recommendation model based on traditional auto-encoders.

**Key words:** deep learning; data mining; auto-encoder; personalized recommendation

## 1 引言

随着互联网技术的发展, 在线教育以其打破课堂时空限制等优势越来越受到人们的关注, 尤其是新冠疫情爆发以来, 人们对在线教育的需求大大增加. 中国互联网络信息中心报告显示: 2020 年 3 月, 我国在线教育用户规模达 4.23 亿, 较 2018 年底增长 110.2%, 占网

民整体的 46.8%. 然而, 随着在线教育的发展, 信息过载现象愈发严重, 如何使用户在短时间内找到适合自己的学习资源便成了一项巨大的挑战. 为了解决信息过载的问题, 科研人员提出了推荐系统的概念, 通过对信息的个性化推荐, 提高了人们的用户体验<sup>[1]</sup>.

有关推荐系统算法的研究较多, 主要分为 3 类: 基

<sup>①</sup> 基金项目: 中国石油大学(华东) 教学研究与改革项目(QT-202005)

收稿时间: 2021-05-07; 修改时间: 2021-06-08; 采用时间: 2021-06-21; csa 在线出版时间: 2022-01-24

于内容的推荐算法、协同过滤的推荐算法和混合的推荐算法. 许多在线教育网站采用传统的协同过滤的推荐算法<sup>[2]</sup>, 其基本思想是利用已有的用户选课信息来构建相似用户群体, 进而进行课程的推荐, 但传统的协同过滤算法存在冷启动和数据稀疏性等问题<sup>[3]</sup>.

由于深度学习具有强大的非线性映射能力, 在处理高维数据时, 可以有效地将其映射到低维空间从而提取高层次的特征, 随着近几年深度学习的发展, 利用深度学习进行个性化推荐也得到了广泛的应用<sup>[4-6]</sup>. 由于自编码器具有强大的隐特征学习能力, 能够有效处理数据的稀疏性问题<sup>[7]</sup>. 另外, 由于用户在进行选课时往往会考虑到之前学过的课程, 用户的选课序列具有明显的时序特点, LSTM (long short-term memory, 长短期记忆网络) 具有强大的时序建模能力, 能够有效处理时间序列数据. 根据上述特性, 本文利用 LSTM 改进自编码器, 将其应用于在线教育的课程推荐领域, 根据用户的课程学习历史来推荐之后的学习课程.

## 2 国内外研究现状

### 2.1 协同过滤推荐算法

协同过滤算法是根据用户的历史数据划分用户群体, 根据群体内其他用户的偏好来为其进行推荐, 分为两大类: 基于用户的协同过滤推荐和基于物品的协同过滤推荐.

刘国丽等人<sup>[8]</sup>提出了一种融合专家信任的协同过滤推荐算法, 在专家信任度计算公式中引入了项目平衡因子, 提高了算法的推荐准确率. Li 等人<sup>[9]</sup>设计了一个应用协同过滤算法的个性化在线教育平台, 实验验证了平台的性能和功能的有效性. 为了解决在物联网情境下的推荐问题, Cui 等人<sup>[10]</sup>提出了一种结合时间相关系数和改进的 K-means 聚类方法的推荐模型, 实验结果表明了其在物联网情境下的优越性能.

### 2.2 基于深度学习的推荐算法

随着深度学习的发展, 越来越多的研究人员将深度学习方法应用到推荐系统上. 利用深度学习的非线性映射能力, 对协同过滤算法中的用户矩阵和物品矩阵进行建模, 使模型具有非线性表达能力.

王素琴等人<sup>[11]</sup>提出了一种基于 LSTM 和课程关联分类的推荐模型, 首先利用 GSP (generalized sequential pattern mining algorithm) 算法和谱聚类算法对课程进行分组, 然后利用 LSTM 进行课程的时间序列建模, 最

后利用 Softmax 函数对课程进行分类, 实验结果表明与协同过滤等算法相比, 该算法具有更高的推荐准确性. 卜祥鹏<sup>[12]</sup>提出了一种基于 GRU (gated recurrent unit, 门控循环单元) 和课程关联关系的推荐模型, 结合课程之间的关联关系, 利用 GRU 和 Softmax 函数进行课程推荐, 实验结果表明与协同过滤或仅考虑课程关联的推荐算法相比, 推荐的准确率更高. Sedhain 等人<sup>[13]</sup>利用自编码器将数据映射到非线性隐空间, 实验证明与 MF (matrix factorization, 矩阵分解) 和 CF (collaborative filtering, 协同过滤) 相比, 具有更低的均方误差. Strub<sup>[3]</sup>结合评价信息和诸如用户属性和物品描述等辅助信息, 利用自编码器进行矩阵的分解, 通过实验验证了所提方法的有效性.

## 3 相关概念

### 3.1 符号表示及问题定义

假设在当前的在线教育系统中, 共有  $m$  位用户,  $n$  门课程, 则用  $U = \{u_1, u_2, \dots, u_m\}$  表示所有用户的集合,  $I = \{I_1, I_2, \dots, I_n\}$  表示所有课程的集合. 对于用户  $u \in U$ , 其学习课程的序列可以表示为  $X = \{I_{t_2}^u, I_{t_2}^u, \dots, I_{t_i}^u\}$ . 课程推荐的目的是已知  $t_1 - t_i$  时间段内用户已经学习的课程, 向其推荐  $t_{i+1}$  时刻用户可能感兴趣的课程.

### 3.2 LSTM

长短期记忆网络是一类特殊的循环神经网络, 最早由 Hochreiter 和 Schmidhuber 为了解决传统循环神经网络梯度消失的问题而提出<sup>[14]</sup>. 不同于前馈神经网络跟普通的循环神经网络, 长短期记忆网络不仅可以处理点数据 (例如图片), 还可以处理间隔和延迟非常长的序列数据 (例如语音跟视频). 长短期记忆网络在诸如连笔字识别<sup>[15]</sup>、语音识别<sup>[16]</sup> 以及时间序列数据异常检测<sup>[17]</sup> 等任务中取得了优秀的成果. 其结构如图 1.

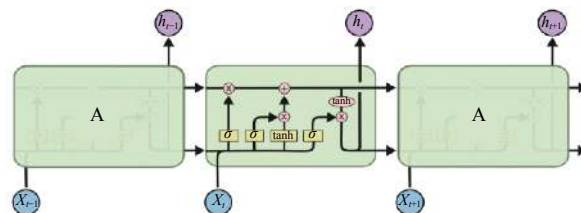


图 1 LSTM 网络结构<sup>[18]</sup>

长短期记忆网络主要由 3 个称为门的特殊结构来控制神经元的状态, 这 3 个门分别称为遗忘门、输入

门跟输出门。

长短期记忆网络的工作由3个阶段完成:

(1) 遗忘阶段. 该阶段来决定神经元需要遗忘哪些信息, 对输入的信息进行选择性的抛弃。

(2) 选择记忆阶段. 该阶段来决定给神经元添加哪些信息。

(3) 输出阶段. 该阶段来决定神经元的哪些信息会被输出。

### 3.3 自编码器

自编码器是一种以无监督学习的方法来学习数据编码的人工神经网络, 其特点是学习目标跟输入数据相同, 一般用途是通过训练神经网络来忽略无关信息以实现数据的降维。

自编码器由两部分组成: 对输入数据进行压缩编码的编码器和将编码重建为输入数据的解码器。其基本结构如图2所示。

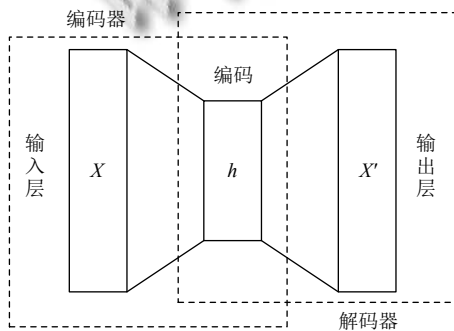


图2 自编码器网络结构

对于编码器映射  $\varphi$  和解码器映射  $\psi$  有:

$$\varphi: X \rightarrow F \tag{1}$$

$$\psi: F \rightarrow X \tag{2}$$

$$\varphi, \psi = \underset{\varphi, \psi}{\operatorname{argmin}} \|x - \psi[\varphi(x)]\|^2 \tag{3}$$

自编码器在编码阶段, 将输入  $x \in \mathbb{R}^d = X$  映射到隐藏层  $h \in \mathbb{R}^p = F$ ,  $h$  常被称为编码、隐变量或隐表示, 对于  $h$  有:

$$h = f(Wx + b) \tag{4}$$

$f$  为激活函数, 常用的有 Sigmoid 函数和 ReLU 函数,  $W$  是输入  $x$  的权重矩阵,  $b$  为偏置单元。在训练过程中通过反向传播算法来更新  $W$  和  $b$ 。

在解码阶段, 将  $h$  映射到输出,  $x'$  和  $x$  结构相同:

$$x' = f'(W'h + b') \tag{5}$$

通过最小化重构误差 (例如均方误差), 来训练自编码器:

$$\begin{aligned} L(x, x') &= \|x - x'\|^2 \\ &= \|x - f'(W'(f(Wx + b)) + b')\|^2 \end{aligned} \tag{6}$$

## 4 基于改进自编码器的在线课程推荐模型

### 4.1 时序特征的提取

为了对用户选课数据的时序性建模, 利用 LSTM 替换自编码器中的前馈神经网络, 类似于 RNN Encoder-Decoder<sup>[19]</sup> 和 EncDec-AD<sup>[20]</sup>, 将二者相结合。

所提时序特征提取方法如图3所示。

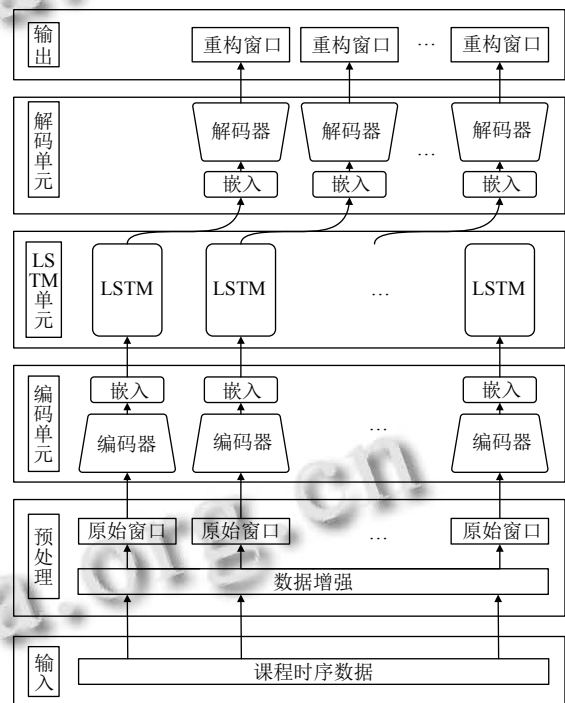


图3 时序特征提取方法

本文提出的时序特征提取方法包括输入层、编码层、LSTM层、解码层和输出层5部分。首先对输入数据进行预处理, 包括数据增强和按窗口划分序列, 随后将每一个序列送入编码器得到特征向量, 再将此特征向量作为 LSTM 网络的输入, 最后将 LSTM 的输出送入解码器, 得到对原始数据窗口下一个窗口的重建。通过最小化重构误差进行训练, 训练完成后, 提取 LSTM 网络输出作为提取的时序特征。

(1) 对原始的时序数据进行预处理, 将处理好后的数据输入到输入层。

预处理过程中采用了数据增强 (data augmentation) 技术, 通过添加高斯噪音来提高模型的准确率.

$$X' = X + \gamma * \sigma^2 * randn(m) \quad (7)$$

其中,  $X'$  为增强后的数据,  $X$  为原始数据,  $\gamma$  为噪音的比例,  $m$  为原始数据的长度,  $\sigma^2$  表示原始数据的方差,  $randn(m)$  可以产生  $m$  条服从标准正态分布的数据.

然后, 将添加噪音的数据以长度为  $p$  的窗口进行序列划分.

$$w_t = [x_{t-p+1}, x_{t-p+2}, \dots, x_t] \quad (8)$$

其中,  $x_i \in X'$ ,  $w_t$  表示结束时刻为  $t$  的数据序列.

(2) 将  $k$  个不重叠的窗口进行序列划分, 经编码后得到的特征表示  $E_t$  作为 LSTM 网络的输入.

$$W_t = [w_{t-(k-1) \times p}, w_{t-(k-2) \times p}, \dots, w_t] \quad (9)$$

$$e_t^i = \text{Encoder}(w_t^i) \quad \text{for } i = 1, 2, \dots, k \quad (10)$$

$$E_t = [e_t^1, e_t^2, \dots, e_t^k] \quad (11)$$

$W_t$  表示结束时刻为  $t$  的窗口序列,  $e_t^i$  表示  $W_t$  中的第  $i$  个窗口  $w_t^i$  的特征表示,  $E_t$  表示与窗口序列  $W_t$  相对应的特征表示.

(3) 训练 LSTM 网络, 用一个序列  $E_t$  中的前  $k-1$  个窗口作为输入数据, 来预测接下来的后  $k-1$  个窗口.

$$[\hat{e}_t^2, \hat{e}_t^3, \dots, \hat{e}_t^k] = \text{LSTM}([e_t^1, e_t^2, \dots, e_t^{k-1}]) \quad (12)$$

LSTM 的计算可以按照以下公式展开:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (13)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (14)$$

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (15)$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (16)$$

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (17)$$

$$h_t = O_t * \tanh(C_t) \quad (18)$$

其中,  $W_f, W_i, W_c, W_o$  与  $b_f, b_i, b_c, b_o$  分别为  $t$  时刻 LSTM 单元中遗忘门、输入门、细胞状态、输出门的权重矩阵和偏置向量,  $\sigma$  为激活函数,  $*$  表示矩阵按元素相乘,  $h_{t-1}$  表示  $t-1$  时刻隐藏层的状态.

应用梯度裁剪 (gradient clipping)<sup>[21]</sup> 来解决 LSTM 中的梯度爆炸问题. 通过将超过阈值的梯度设定为固定值来使梯度值保持在合理水平. 更确切地说, 在梯度  $g$  更新前, 裁剪  $g$  的范数  $\|g\|$ :

$$\text{if } \|g\| > \nu \quad (19)$$

$$g \leftarrow \frac{g\nu}{\|g\|} \quad (20)$$

其中,  $\nu$  是阈值的范数,  $g$  是要用来更新的参数.

采用真实值跟 LSTM 预测值之间的均方误差 (mean square error, MSE) 作为损失函数, MSE 计算公式如式 (21) 所示:

$$MSE = \frac{1}{k-1} \sum_{i=1}^{k-1} (e_t^i - \hat{e}_t^{i+1})^2 \quad (21)$$

其中  $e_t^i$  为真实数据值,  $\hat{e}_t^{i+1}$  为预测数据值.

(4) 利用解码器将 LSTM 的输出重构为时间序列窗口.

$$\hat{w}_{t-(k-i) \times p} = \text{Decoder}(\hat{e}_t^i) \quad \text{for } i = 2, 3, \dots, k \quad (22)$$

#### 4.2 在线课程的推荐

经过上一步提取到数据的时序特征后, 将 LSTM 的输出作为输入, 经过 Softmax 函数, 得到模型最后的推荐结果. 如图 4 所示.

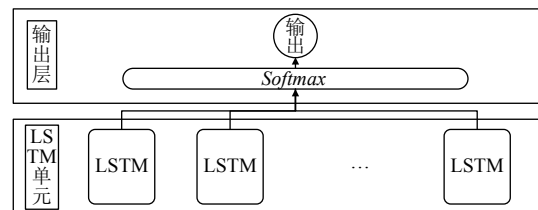


图4 进行课程推荐

$$y_u = \text{Softmax}(W_u \cdot h + b_u) \quad (23)$$

## 5 实验结果及分析

### 5.1 实验数据集

采用 MOOC 真实数据集<sup>[22]</sup>, 共 458 454 条选课记录, 包含 82 535 位用户和 1 302 门课程. 每条记录包含用户 id、选课时间、课程 id、课程名称、课程类别等属性.

### 5.2 数据预处理

(1) 首先, 将原始数据数据简化为三元组 (用户 id, 选课时间, 课程 id).

(2) 其次, 从 82 535 位用户中, 剔除选修课程数量少于 10 门的用户, 剩余 130 812 条数据, 8 268 位用户.

(3) 然后, 再按照用户分组, 共分为 8 268 个组, 在每个组内按照选课时间升序进行排序.

(4) 最后, 按照 (用户 id, 课程 id 时间序列) 合并组内数据, 得到最终数据集, 共计 8 268 条数据.

### 5.3 评价指标

本文采用 Top- $N$  推荐系统中常用的查准率 (precision,  $P$ )、查全率 (recall,  $R$ )、 $F_1$ -分数 ( $F_1$ -Score,  $F_1$ ) 3 个评价指标来衡量所提方法的推荐性能, 表达式如下.

$$P = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \quad (24)$$

$$R = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (25)$$

$$F_1 = \frac{2 \times P \times R}{P + R} \quad (26)$$

其中,  $R(u)$  表示根据用户在训练数据集上课程学习数据给用户做出的课程推荐列表,  $T(u)$  表示用户在测试数据集上的课程学习列表.

### 5.4 实验对比

对于 Top- $N$  课程推荐问题, 本文取  $N \in \{1, 3, 10\}$ , 设置 3 组对比实验, 每组对比实验中, 与协同过滤算法 (CF) 和基于传统的自编码器的推荐算法 (AE) 进行对比.

实验结果如表 1、表 2、表 3 所示.

表 1  $N=1$  时实验对比

评价指标	CF	AE	本文方法
$P$	0.136	0.152	<b>0.167</b>
$R$	0.122	0.143	<b>0.154</b>
$F_1$	0.129	0.147	<b>0.160</b>

表 2  $N=3$  时实验对比

评价指标	CF	AE	本文方法
$P$	0.264	0.281	<b>0.305</b>
$R$	0.259	0.272	<b>0.292</b>
$F_1$	0.261	0.276	<b>0.298</b>

表 3  $N=10$  时实验对比

评价指标	CF	AE	本文方法
$P$	0.525	0.577	<b>0.595</b>
$R$	0.513	0.571	<b>0.586</b>
$F_1$	0.519	0.574	<b>0.590</b>

由表 1、表 2、表 3 可得以下结论:

(1) 在同一组对比实验内, 所提方法在查准率、查全率、 $F_1$ -Score 3 个指标上的表现均要优于协同过滤算法和基于传统的自编码器的推荐算法. 当  $N=1$  时, 与 CF 算法相比, 准确率提高 22.8%, 召回率提高 26.2%,  $F_1$ -Score 提高 24.6%; 与 AE 算法相比, 准确率提高

9.9%, 召回率提高 7.7%,  $F_1$ -Score 提高 8.7%. 当  $N=3$  时, 与 CF 算法相比, 准确率提高 15.5%, 召回率提高 12.7%,  $F_1$ -Score 提高 14.1%; 与 AE 算法相比, 准确率提高 8.5%, 召回率提高 7.4%,  $F_1$ -Score 提高 7.9%. 当  $N=10$  时, 与 CF 算法相比, 准确率提高 13.3%, 召回率提高 14.2%,  $F_1$ -Score 提高 13.8%; 与 AE 算法相比, 准确率提高 3.1%, 召回率提高 2.6%,  $F_1$ -Score 提高 2.9%. 实验结果表明, 基于深度学习方法的推荐算法性能表现要优于传统的算法, 这是因为深度学习算法可以提取更深层次的数据特征. 本文所提方法由于采用了 LSTM 对数据的时序特征进行建模, 因此效果也要优于基于传统的自编码器的推荐算法.

(2) 不同对比实验组间, 随着  $N$  的增大, 模型的 3 个性能评价指标也越来越高, 这是因为随着推荐课程的数量增加, 包含用户喜欢的课程的概率也越来越大.

## 6 结论与展望

本文针对在线教育的课程推荐问题, 提出了一种基于改进自编码器的在线课程推荐模型. 首先, 利用长短期记忆网络改进自编码器, 使得模型可以提取数据的时序性特征; 然后, 利用提取到的数据特征进行在线教育的课程推荐; 最后, 在真实的 MOOC 数据集上进行实验验证, 实验结果表明所提方法与协同过滤算法和基于传统的自编码器的推荐算法相比, 具有更高的准确性.

但是, 本文工作仍存在以下不足: 在 LSTM 网络中, 有众多的超参数需要手动设置, 仅凭个人经验难以找到最合适的数值, 接下来将研究如何利用神经网络搜索, 去自动寻找最优的超参数.

### 参考文献

- Zhang S, Yao LN, Sun AX, *et al.* Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, 2019, 52(1): 5.
- Yang B, Lei Y, Liu JM, *et al.* Social collaborative filtering by trust. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39(8): 1633–1647.
- Strub F, Gaudel R, Mary J. Hybrid recommender system based on autoencoders. *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. Boston: ACM, 2016. 11–16.
- 陈征, 宋轩, 杜先瑞, 等. 基于多图神经网络的个性化推荐

- 模型. 中国传媒大学学报(自然科学版), 2020, 27(6): 14–18.
- 5 郭旦怀, 张鸣珂, 贾楠, 等. 融合深度学习技术的用户兴趣点推荐研究综述. 武汉大学学报(信息科学版), 2020, 45(12): 1890–1902.
- 6 徐亚军, 郭俭. K12 学习平台个性化学习资源推荐. 计算机系统应用, 2020, 29(7): 217–221. [doi: [10.15888/j.cnki.csa.007510](https://doi.org/10.15888/j.cnki.csa.007510)]
- 7 Strub F, Mary J. Collaborative filtering with stacked denoising autoencoders and sparse inputs. NIPS Workshop on Machine Learning for eCommerce. Montreal, 2015.
- 8 刘国丽, 廉孟杰, 于丽梅, 等. 融合专家信任的协同过滤推荐算法. 计算机系统应用, 2021, 30(4): 160–167. [doi: [10.15888/j.cnki.csa.007919](https://doi.org/10.15888/j.cnki.csa.007919)]
- 9 Li J, Ye Z. Course recommendations in online education based on collaborative filtering recommendation algorithm. Complexity, 2020, 2020: 6619249.
- 10 Cui ZH, Xu XH, Xue F, *et al.* Personalized recommendation system based on collaborative filtering for IoT scenarios. IEEE Transactions on Services Computing, 2020, 13(4): 685–695.
- 11 王素琴, 吴子锐. 利用 LSTM 网络和课程关联分类的推荐模型. 计算机科学与探索, 2019, 13(8): 1380–1389.
- 12 卜祥鹏. 基于 GRU 和课程关联关系的推荐模型. 软件, 2020, 41(6): 137–142.
- 13 Sedhain S, Menon AK, Sanner S, *et al.* Autorec: Autoencoders meet collaborative filtering. Proceedings of the 24th International Conference on World Wide Web. Florence: ACM, 2015. 111–112.
- 14 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735–1780.
- 15 Graves A, Liwicki M, Fernández S, *et al.* A novel connectionist system for unconstrained handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(5): 855–868.
- 16 Sak H, Senior A, Beaufays F. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv: 1402.1128, 2014.
- 17 Malhotra P, Vig L, Shroff G, *et al.* Long short term memory networks for anomaly detection in time series. 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges: ESANN, 2015. 89–94.
- 18 Staudemeyer RC, Morris ER. Understanding LSTM—a tutorial into long short-term memory recurrent neural networks. arXiv: 1909.09586, 2019.
- 19 Cho K, Van Merriënboer B, Gulcehre C, *et al.* Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv: 1406.1078, 2014.
- 20 Malhotra P, Ramakrishnan A, Anand G, *et al.* LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv: 1607.00148, 2016.
- 21 Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. Proceedings of the 30th International Conference on International Conference on Machine Learning. Atlanta: JMLR, 2013. III-1310–III-1318.
- 22 Zhang J, Hao BW, Chen B, *et al.* Hierarchical reinforcement learning for course recommendation in MOOCs. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33(1): 435–442.