

# TCN-Based Short Term Interest Personalized Sequence Recommendation

Liang Xu, Shaozhang Niu

College of Computer Science, Beijing University of Posts and Telecommunications, Beijing  
Email: 1357063398@qq.com, szniu@bupt.edu.cn

Received: Nov. 6<sup>th</sup>, 2019; accepted: Nov. 19<sup>th</sup>, 2019; published: Nov. 26<sup>th</sup>, 2019

---

## Abstract

With the rapid development of mobile Internet, personalized recommendation also faces new opportunities and challenges. Traditional recommendation algorithms pay more attention to static long-term feature recommendation, but in Mobile Taobao, Douying and other feed flow immersive applications, long-term fast changing browsing behavior requires faster calculation methods to capture frequent changes in interest. In this paper, a TCN-based interest sequence recommendation algorithm is proposed. By convolution in horizontal and vertical directions, the short-term dynamic interest information of users is captured, and then the long-term interest information of users is combined to complete the recommendation. The experimental results show that compared with other recommendation algorithms, R-TCN (Recommendation-TCN) has better performance in Top-N recommendation, faster training time, better ability to capture short-term interest changes, and is suitable for parallel processing industry recommendation environment.

## Keywords

Temporal Convolutional Network, Residual, Causal Dilated Convolution, Recommendation

---

## 基于TCN的短期兴趣个性化序列推荐

许 亮, 牛少彰

北京邮电大学计算机科学学院, 北京  
Email: 1357063398@qq.com, szniu@bupt.edu.cn

收稿日期: 2019年11月6日; 录用日期: 2019年11月19日; 发布日期: 2019年11月26日

---

## 摘 要

随着移动互联网蓬勃发展, 个性化推荐也迎来了新的机遇和挑战。传统推荐算法更注重静态长期特征的

推荐, 但手机淘宝、抖音等feed流沉浸式应用中, 长时间的快速变换的浏览行为, 需要更快的计算方法捕捉兴趣的频繁变化。本文提出一种基于TCN的兴趣序列推荐算法, 通过水平和垂直两个方向上卷积, 捕获用户的短期动态兴趣信息, 再结合用户长期兴趣信息, 完成推荐。实验结果表明, 与其他推荐算法相比, R-TCN (Rcommendation-TCN)的Top-N推荐效果更好, 训练时间更快, 更能捕获短期兴趣变化, 适合用于并行处理的工业推荐环境。

## 关键词

时间卷积网络, 残差, 因果空洞卷积, 推荐

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

在电商、互联网广告、移动互联网中, 推荐系统发挥着越来越大的价值。过去几年来, 一个明显的趋势是搜索、计算广告, 以及个性化推荐这三者在底层模型和技术工具上越来越趋于融合。相比于 PC 互联网, 用户使用场景和习惯的改变使得用户行为模型在推荐领域的重要性获得了空前提升。推荐不再单单局限于兴趣领域, 兴趣、关系、场景、行为模式, 这四者都会起到至关重要的作用。目前几乎所有的解决范式都是将用户理解为行为结构体, 这个结构体包含了兴趣、关系、场景、行为模式中的一种或者多种, 这些信息向量化之后成为大量 Feature。而推荐的目标就是预测用户的未来行为, 如点击行为(CTR 预估)、购买行为(CVR 预估)等等。但是大多数基于特征集合的方法更注重一般偏好, 更容易忽略短期动态变化的兴趣。一般偏好代表用户的长期和静态的行为, 更容易基于整体的特征矩阵运算得到。而用户的短期和动态行为, 则来自于在很近的时间物品之间的某种关系。例如在某电商平台的手机 app 平台, 在不断下拉的 feed 流环境下, 用户的兴趣动态变化频繁, 有些用户可能在购买小米手机后不久就购买手机配件, 尽管大多数用户一般不会购买手机配件。在这种情况下, 只考虑一般偏好的系统, 将错过在销售 iPhone 后推荐手机配件的机会, 因为购买手机配件不是一种长期的用户行为。

本文目标是通过结合用户的一般喜好和序列模式, 来推荐他\她未来最有可能需要的一系列物品。和以往各种深度学习机器学习的 Top-N 推荐算法不同的是, Top-N 序列推荐将用户行为建模为物品的序列, 而不是集合。

## 2. 相关研究

### 2.1. 基于特征集合

FM (Factorization Machine)是近年来在推荐、CTR 预估中常用的一种算法, 该算法在 LR 的基础上考虑交叉项, FM 在后半部分的交叉项中为每个特征都分配一个特征向量  $V$ , 这其实可以看作是一种 Embedding 的方法。Dr. Zhang [1]在文献中提出一种利用 FM 得到特征的 embedding 向量并将其组合成 dense real 层作为 DNN 的输入的模型——FNN [1], 将每个特征用其所属的 field 来表示, 导致原始输入大大减少, 但是 FM 参数需要预训练, 无法拟合低阶特征, 每个 field 只有一个非零值的强假设。Dr. Zhang 在 FNN 模型的基础上又提出了新模型 PNN [2]。但模型对于低阶特征的表达比较有限。

因此, Google 在 2016 年提出了大名鼎鼎的 Wide & Deep [3]结构来解决这样的问题。网络结构如

图 1 所示, Wide 部分是 LR 模型, Deep 部分是 DNN 模型。Wide and Deep 模型的核心思想是结合线性模型的记忆能力(memorization)和 DNN 模型的泛化能力(generalization), 在训练过程中同时优化 2 个模型的参数, 从而达到整体模型的预测能力最优。记忆(memorization)即从历史数据中发现 item 或者特征之间的相关性。泛化(generalization)即相关性的传递, 发现在历史数据中很少或者没有出现的新的特征组合。

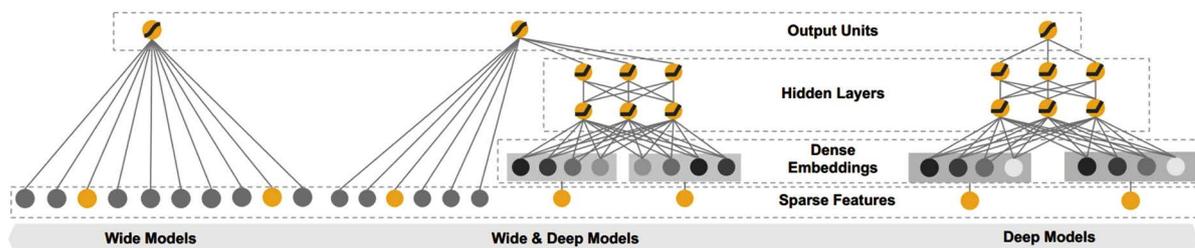


Figure 1. Wide & deep model structure

图 1. Wide & deep 模型结构

华为诺亚方舟团队将 Wide & Deep 部分的 LR 部分替换成 FM 来避免人工特征工程, 于是有了 DeepFM [4]。DeepFM 包含两部分: 神经网络部分 DNN 与因子分解机 FM 部分, 分别负责低阶特征的提取和高阶特征的提取。这两部分共享同样的输入。Google 在 2017 年提出了 Deep & Cross Network, 简称 DCN [5]的模型, 可以任意组合特征, 而且不增加网络参数。图 2 为 DCN 的结构。与此对照, 阿里在 2017 年提出的 Deep Interest Network, 简称 DIN [6]模型。与上面的 FNN, PNN 等引入低阶代数范式不同, DIN 的核心是基于数据的内在特点, 引入了更高阶的学习范式。用户的兴趣是多种多样的, 从数学的角度来看, 用户的兴趣在兴趣空间是一个多峰分布。在预测 CTR 时, 用户 embedding 表示的兴趣维度, 很多是和当前 item 是否点击无关的, 只和用户兴趣中的局部信息有关。因此, 受 attention 机制启发, DIN 在 embedding 层后做了一个 action unit 的操作, 对用户的兴趣分布进行学习后再输入到 DNN 中去, 网络结构如图 3 所示。

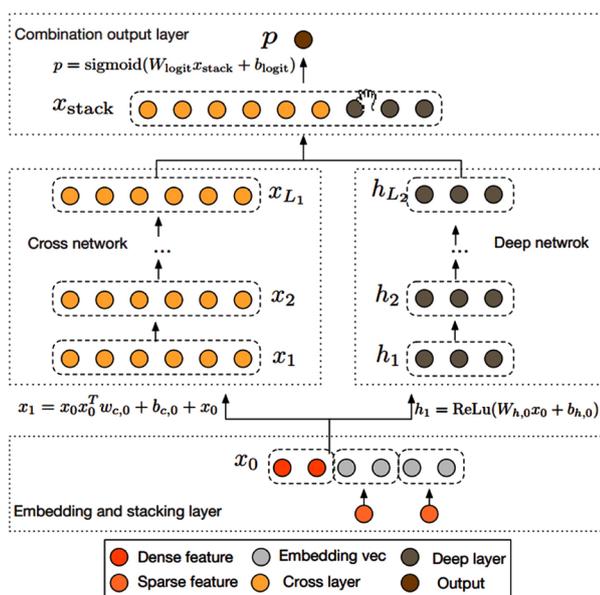


Figure 2. The Deep & Cross network

图 2. The Deep & Cross 网络

	预训练和人工特征	拟合低阶特征	拟合高阶特征	组成
FM		√	√	FM
DNN			√	MLP
FNN	√		√	FM+FLP
PNN			√	FM+PRODUCT+MLP
Wide & Deep		√	√	LR+EMBEDDING+MLP
DeepFM		√	√	FM+EMBEDDING+MLP
NFM		√	√	FM+EMBEDDING+MLP
AFM		√	√	FM+EMBEDDING+ATTENTION+MLP
DCN		√	√	EMBEDDING+CROSS+MLP+LR
DIN		√	√	EMBEDDING+ATTENTION+MLP

Figure 3. Summary comparison of various deep learning models on CTR

图 3. 在 CTR 上的各种深度学习模型汇总比较

以上介绍了深度学习在 CTR 场景下最新的网络结构, 总结起来如图 3 所示。各种 CTR 深度模型看似结构各异, 其实大多数可以用图 3 表示, 从图中可以 CTR 领域的深度学习过程可以用一个通用范式来表达。input->embedding: 把大规模的稀疏特征 ID 用 embedding 操作映射为低维稠密的 embedding 向量。embedding 层向量: concat, sum, average pooling 等操作, 大部分 CTR 模型在该层做改造。embedding->output: 通用的 DNN 全连接框架, 输入规模从  $n$  维降为  $k \times f$  维度甚至更低。其中, embedding vector 这层的融合是深度学习模型改造最多的地方, 该层是进入深度学习模型的输入层, embedding 融合的质量将影响 DNN 模型学习的好坏。

## 2.2. 基于特征序列

基于马尔可夫链[7] [8]的模型是 Top-N 序列推荐早期的一种方法,  $L$  阶马尔可夫链基于前  $L$  个行为来做推荐。一阶马尔可夫链是利用极大似然估计学习的一种物品到物品的转移矩阵。

Rendle 等人提出的分解个性化马尔可夫链(FPMC [9])及其变体通过将转移矩阵分解为两个潜在的和低秩的子矩阵来改进这种方法。基于物品相似模型的分解序列预测(Fossil [10])将该方法推广到高阶马尔可夫链, 利用加权和聚合方法对已有物品的潜在表示进行加权和聚合。然而, 现有的方法主要有两方面局限性: 1) 未能对联合级序列模式进行建模。2) 不允许跳过某些行为, 过去行为的影响不允许跳过几步对未来发生影响。例如, 游客在机场、酒店、餐厅、酒吧和景点中依次签到, 虽然机场和酒店的签到并不是在景点之前, 但它们与后者依然有很强的关联。在餐厅或酒吧签到对景点签到几乎没有影响, 因为去景点之前到餐厅或者酒吧不是需要的。为解决上述两种工作的局限性, 并结合已有方法的优点, 作者提出一种卷积序列嵌入推荐模型 R-TCN, 作为 Top-N 序列推荐的解决方案。

## 3. R-TCN

### 3.1. 模型结构

本文所提出的网络结构主要受 TCN [11]网络模型的启发。TCN 是将卷积网络设计中的最佳实践提炼而成的一个简单架构。其在不同任务和数据集上的性能优于典型的 RNN 网络。R-TCN 的主要特征是: 1) 架构中使用的卷积是因果关系, 保证没有从未来到过去的信息泄露, 这是适用于序列建模的前提; 2) 该架构直接对稀疏数据嵌入之后的矩阵进行卷积操作; 3) 通过使用非常深的网络(残差层搭建)和空洞卷积来构建非常长的有效历史规模, 即可以利用非常遥远的历史值做出预测。

R-TCN 由三个部分组成: embedding [12]层、卷积层和全连接层。

1) embedding 层: 为了训练 CNN, 本文从用户序列中为每个用户  $u$  提取  $L$  个连续物品作为输入, 并提取它们接下来的  $T$  个物品作为目标, 如图 4 所示, 这是通过在用户序列上滑动一个大小为  $L + T$  的窗口来完成的, 每个窗口都为  $u$  生成一个训练样本, 由一个三元组表示(用户  $u$ , 前  $L$  个物品, 后  $T$  个物品)。

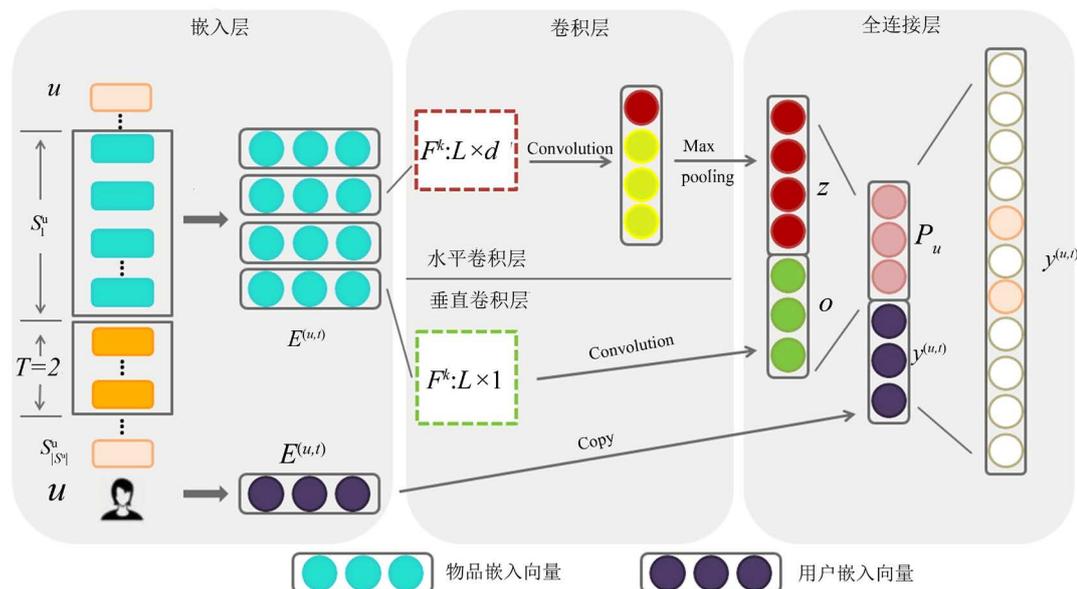


Figure 4. R-TCN model module diagram

图 4. R-TCN 模型模块图

2) 卷积层: R-TCN 使用因果空洞卷积作为卷积层, 而每两个这样的卷积层与恒等映射可以封装为一个残差模块。这样由残差模块堆叠起一个深度网络。网络结构如图 4 所示, R-TCN 由很多个残差块串联而成, 残差块的个数由具体任务的输入和输出的尺寸大小决定。

3) 全连接层: 将这两个卷积层的输出串联起来, 并将它们输入到一个全连接的神经网络中。

### 3.2. embedding 层

对于简单的公式, 可以直接以文本方式输入; 对于复杂的公式, 可以考虑使用公式编辑器, 或者将公式制作成图片后插入文中。编辑公式的过程中要特别注意减号与连字符的区别, 前者较长, 后者较短。NN 之所以在 CV 领域[13]大放异彩是由于其具有如下特性: 1) 参数共享通常一个特征检测子(如边缘检测)在图像某一部位有用也在其他部位生效。2) 稀疏连接每一层的输出只依赖于前一层一小部分的输入。在 NLP 任务中由于语句天然存在前后依赖关系, 所以使用 CNN 能获得一定的特征表达[14]。

在推荐任务中输入是空间无序的, 特征是高维稀疏的, 没有 NLP 和 CV 任务上空间上的强相关性和密集性, 造成了 CNN 在推荐领域使用甚少。

如图 4 部分, 将前  $L$  个物品在隐空间的表示连接起来, 作为用户  $u$  在当前时刻  $t$  的序列矩阵表示( $d$  是隐藏层的维数):

$$E^{(u,t)} \in R^{L \times d}$$

除了物品的表示之外, 本文还为用户  $u$  提供了一个用户表示:

$$P_u \in R^d$$

它表示潜在空间中的用户特性。这些表示在图 4 的左侧第一模块中, 用蓝色和紫色的实心圆表示。

### 3.3. 卷积层

#### 3.3.1. 总体介绍

如图 4 将前  $L$  个物品在隐空间的表示 Latent Space ( $L \times d$  的矩阵  $E(u, t)$ ) 看作一张“图像”，卷积层就是利用卷积核对其进行序列模式的搜索和学习。两个水平卷积核(用  $h \times d$  的矩阵表示，图中所示  $h = 2$ )通过在矩阵  $E$  上滑动来提取两种不同的联合级序列模式。其中第一个卷积核通过与“高铁”和“网红店”的 embedding 表示的交互，跳过“旅行箱”和“房卡”，提取出“(高铁, 网红店)→景点”的模式。同理，第二个卷积核提取了“(旅行箱, 房卡)→酒店”的序列模式。矩阵方块中颜色越深代表值越大。因此，水平卷积核可以被训练来提取具有多种模式。图 5 中间卷积部分的下半部分显示了垂直卷积层，在该层中，垂直卷积核(用  $L \times d$  的矩阵表示)在矩阵  $E$  上从左到右滑动，通过对前几项物品的潜在表示的加权来捕获点级序列模式。

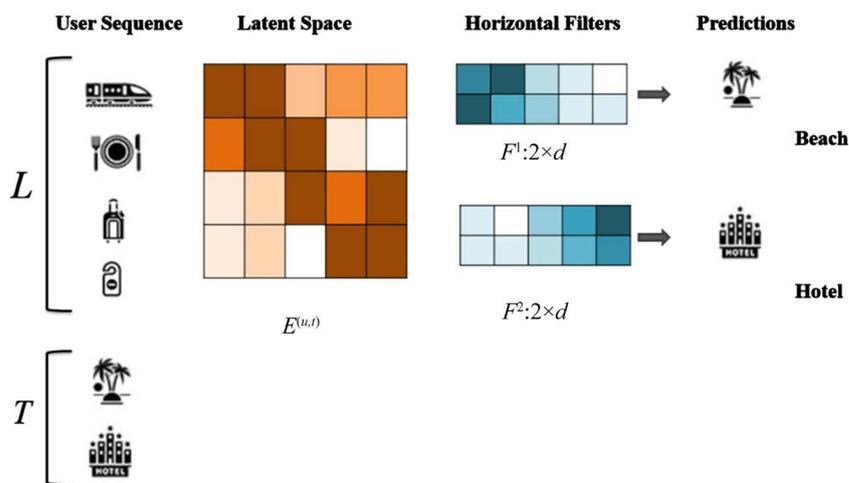


Figure 5. Sequence prediction diagram  
图 5. 序列预测示意图

#### 3.3.2. 残差

序列建模任务通常需要很长的感受野，而感受野依赖于网络深度以及卷积核大小等，为了使更大更深的人 R-TCN 网络更加稳定，采用通用残差模块。

残差网络(Residual Network, ResNet [15])通过加入 shortcut connections，变得更加容易被优化。包含一个 shortcut connection 的几层网络被称为一个残差块(residual block)，这已多次被证明有利于非常深的网络。一个深度网络中如果期望用一个非线性单元  $f(x, \theta)$  去逼近目标函数  $h(x)$ ，可以将目标函数拆分成恒等函数  $x$  和残差函数  $h(x) - x$ 。

$$h(x) = x + (h(x) - x)$$

根据通用近似定理，一个由神经网络构成的非线性单元有足够的力量来近似逼近原始目标函数或残差的学习会比前者简单。因此，原来的优化问题便转换为：让非线性单元  $f(x, \theta)$  去近似残差函数  $h(x) - x$ ，并用  $f(x, \theta) + x$  去逼近  $h(x)$ 。

R-TCN 的残差块结构如图 6 中所示。在 R-TCN 的残差模块内，有两层因果空洞卷积，根据非线性，使用了校正线性单元(relu, ReLU)。对于归一化，我们将权值归一化应用于卷积滤波器。此外，R-TCN 在残差模块内的每个因果空洞深度可分离卷积后都添加了 Dropout [16]以实现正则化，防止

过拟合。最终，由多个这样的残差模块堆叠起一个深度网络。图 6 给出了残差连接的示例。残差块里面有两层因果空洞卷积。实线代表滤波器，虚线代表恒等映射，在 R-TCN 中，输入与输出有不同的维度，因此需要使用额外的  $1 \times 1$  卷积来确保对应元素相加有相同的维度。 $k$  代表卷积核大小，这里卷积核尺寸为 3， $d$  表示空洞系数，这里为 1，也就是相当于普通卷积操作。

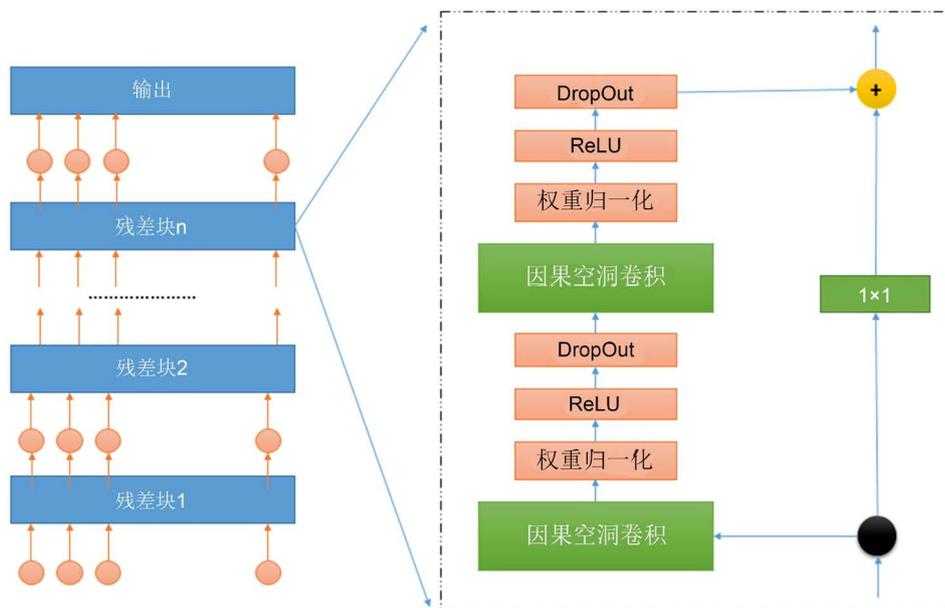


Figure 6. Residual block structure  
图 6. 残差块结构图

### 3.3.3. 因果空洞卷积层

序列问题强调的是先后顺序，RNN 的特性使其有助于解决时间序列预测问题，但它串行处理序列数据，训练速度慢。CNN 可以并行处理序列数据，速度比 RNN 快，但普通卷积网络会把序列中未来的数据和过去的数据同等地位对待，可能会导致未来数据到过去数据的泄露。因此普通卷积不能直接用于处理序列数据，要用因果卷积替代传统卷积。如图 7 所示，因果卷积[17]提供了处理时间流的适当工具，当

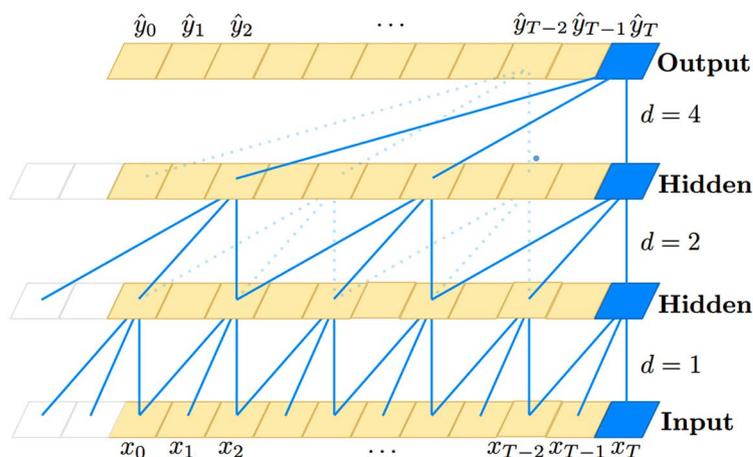


Figure 7. TCN causal convolution  
图 7. TCN 因果卷积

对输入数据的顺序很注重的时候, 因果卷积便可以发挥作用。一个简单的因果卷积只能用网络深度的线性大小来回顾历史, 而普通因果卷积网络的感受野很小, 多层卷积感受野增长太慢, 不能覆盖较长的序列。这使得将上述因果卷积应用于序列任务非常困难, 尤其是那些需要较长历史的任务。

根据 van den [18]等人的工作, 解决方案是使用空洞卷积, 不增加参数数量, 同时增加输出单元感受野, 使一个指数级大的感受野成为可能。如图 8 所示, 左图对应  $3 \times 3$  的空洞卷积, 空洞为 0, 和普通的卷积操作一样, 中间图对应  $3 \times 3$  的空洞卷积, 实际的卷积核还是  $3 \times 3$ , 空洞为 1, 但是该卷积核的感受野已经增大到了  $7 \times 7$ ; 右侧图对应空洞卷积实际的卷积核还是  $3 \times 3$ , 空洞为 3, 该卷积核的感受野已经增大到  $15 \times 15$ 。可以看出, 增加空洞后, 参数量不增加但是大大增加感受野范围。

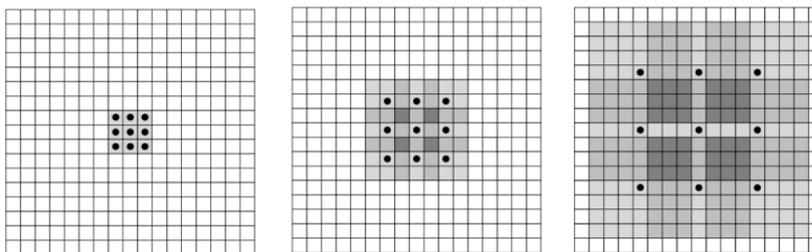


Figure 8. TCN cavity convolution

图 8. TCN 空洞卷积

### 3.4. 全连接层

在全连接层中, 如图 4 右部分所示, 作者将这两个卷积层的输出 `concat` 起来成为向量  $Z$ , 并将它们输入到一个全连接的神经网络中, 以获得更高层次和更抽象的特征。值得注意的是我们对  $Z$  和  $P_u$  进行了 `concat` 操作。如图 4 向量  $Z$  来自卷积操作, 向量  $P_u$  来自输入端的用户向量直连。向量  $Z$  表示用户的短期兴趣, 而向量  $P_u$  则代表了用户的长期兴趣。`Concat(Z, P_u)`将同时拥有长期和短期兴趣信息。同样的,  $P_u$  的生成方法是很多的, 我们采用 DeepFM 预训练之后形成的  $P_u$ , 因为对于一般兴趣的捕捉存在着包括机器学习(FM LR)和深度学习(DeepFM Wide & Deep)在内的众多方法。所以本文的模型具有结合 CNN、RNN 机器学习等多种模型共同发挥作用的潜力, 采集他们的优点, 同时序列预测又规避了他们的动态短期兴趣捕捉不足的弱点。

## 4. 实验准备

### 4.1. 数据集

Movielens [19]数据包含 138,493 个用户, 27,278 部电影, 21 个类别和 20,000,263 个样本。基于 `userid` 分割训练和测试集。在 138,493 名用户中, 随机抽取 10 万人形成训练集(大约 14,470,000 个样本), 其余 38,493 人形成测试集中(约 5,530,000 个样本)。特征包括 `movie_id`, `movie_cate_id` and `userrated movie_id_list`, `movie_cate_id_list`。

Amazon Dataset Amazon [20]数据集包含商品评论和来自 Amazon 的各种元数据, 将用作本课题的基准数据集。我们抽取 Electronics 这个类别的子集数据进行实验, 包含 192,403 个用户, 63,001 个商品, 801 个类别和 1,689,188 个样品。此数据集中的用户行为非常丰富, 每个用户和商品的评论超过 5 条。假设用户行为是  $(b_1, b_2, \dots, b_k, \dots, b_n)$ , 那么任务就是根据前  $k$  个浏览的商品预测第  $k+1$  个要浏览的商品。特征包括 `goods_id`, `cate_id`, `user reviewed goods_id_list` `cate_id_list`。统计两个数据集的数据细节, 如表 1 所示。

**Table 1.** Dataset statistics used in this article**表 1.** 本文使用的数据集统计

DataSet	Users	Goods	Categories	Samples
Amazon (Electro)	192,403	63,001	801	1,689,188
MovieLens	138,493	27,278	21	20,000,263

## 4.2. 对比模型

### 1) LR

Logistic regression (LR)是一种应用广泛的浅层回归方法。

### 2) FM

FM (Factorization Machine)是近年来在推荐、CTR 预估中常用的一种算法。

### 3) CF

协同过滤算法分为 UserCF (基于用户的算法)、ItemCF (基于物品的算法)两大类。

### 4) Wide & Deep

在实际工业应用中, Wide & Deep 模型已被广泛接受。根据上节所述, 包含 Wide (手工 LR)和 Deep (DNN)两部分。

### 5) DeepFM

根据上节所述, 利用 FM 替代 Wide & Deep 的 Wide 部分, 实现 End2End 训练。

## 4.3. 评价指标

我们在每个用户序列的 70%作为训练集, 并使用之后的 10%动作序列来搜索各种模型的最佳超参数。使用余下的 20%的作为模型表现的测试集。正如之前论文[21], 我们按 Precision@n、Recall@n 和 Mean Average Precision (MAP)评估一个模型。给定用户的 Top-N 预测项的列表, 表示为  $\hat{R}_{1:N}$ , 则

$$\text{Prec@N} = \frac{|R \cap \hat{R}_{1:N}|}{N}$$

$$\text{Recall@N} = \frac{|R \cap \hat{R}_{1:N}|}{|\hat{R}|}$$

为了计算所有用户的 Prec Recall 的平均值,  $N \in \{1, 5, 10\}$  AP 计算值如下:

$$AP = \frac{\sum_{N=1}^{|\hat{R}|} \text{Prec@N} \times \text{rel}(N)}{|\hat{R}|}$$

Precision (MAP)是所有用户的 AP 的平均值。

## 5. 实验结果

实验软件环境为 Windows 10 操作系统, 深度学习框架是 tensorflow1.12。实验硬件环境是 Intel Core i3-8100HQ 四核处理器, 4 GB 内存, GPU 为 NVIDIA(R) GeForce GTX 1060。

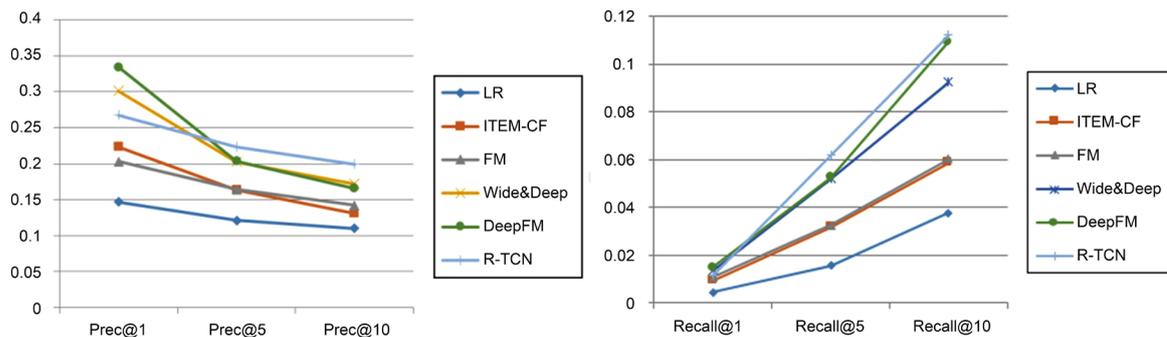
### 5.1. 推荐效果

表 2 显示了 Amazon 数据集和 MovieLens 数据集的结果。所有实验重复多次, 并汇报平均结果。显然, 所有的深度神经网络(Wide & Deep DeepFM)都显著地击败了机器学习模型(LR FM), 这确实证明了深度学习的力量。

**Table 2.** Comparison of statistical recommendation indicators  
**表 2.** 统计推荐效果指标比较

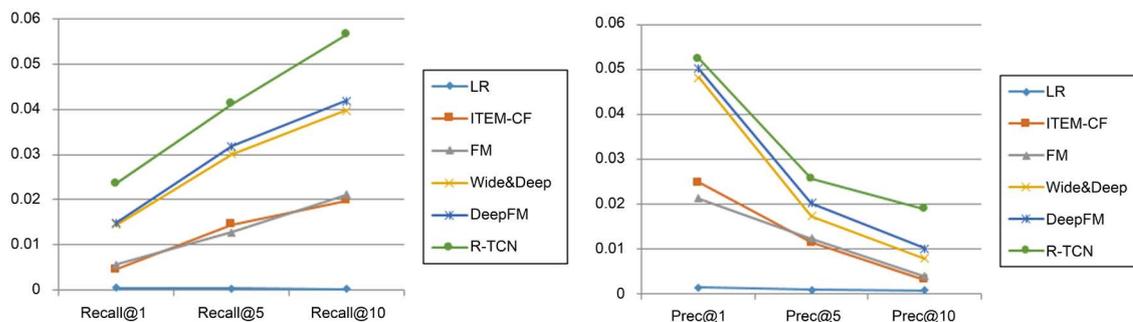
数据集	指标	LR	CF	FM	W&D	DeepFM	R-TCN
MovieLens	Prec@1	0.1467	0.2235	0.2027	0.3014	<b>0.3342</b>	0.2677
	Prec@5	0.1211	0.1635	0.1643	0.2024	0.2037	<b>0.2235</b>
	Prec@10	0.1101	0.1309	0.1423	0.1718	0.1656	<b>0.1991</b>
	Recall@1	0.0043	0.0091	0.0104	0.0133	<b>0.0147</b>	0.0112
	Recall@5	0.0156	0.0316	0.0323	0.0519	0.0523	<b>0.0616</b>
	Recall@10	0.0375	0.0587	0.0601	0.0923	0.1093	<b>0.1121</b>
Amazon	Prec@1	0.0014	0.0248	0.0212	0.0481	0.0501	<b>0.0523</b>
	Prec@5	0.0009	0.0114	0.0123	0.0173	0.0201	<b>0.0256</b>
	Prec@10	0.0007	0.0031	0.0039	0.0078	0.0101	<b>0.0189</b>
	Recall@1	0.0004	0.0045	0.0056	0.0145	0.0149	<b>0.0236</b>
	Recall@5	0.0003	0.0145	0.0128	0.0301	0.0318	<b>0.0412</b>
	Recall@10	0.0002	0.0198	0.0211	0.0398	0.0419	<b>0.0565</b>

如图 9 图 10 所示,在相对简单的 MovieLens 数据集上,Top1 推荐的任务与 CTR 任务很类似,DeepFM 表现是最好的,但是 Top5 Top10 的推荐任务, R-TCN 的指标都是第一, 好于其他任务。分析原因, 一是 Top-N 推荐任务中 R-TCN 模型通过 TCN 的宽阔视野和两个方向上的卷积捕捉到了更多更细腻的信息, 这一点其他模型无法做到, 二是结合其他的模型已有优点, 同时利用了长期静态的用户嵌入向量和短期动态向量的信息。



**Figure 9.** Movielens data set PREC/Recall (y-axis) vs. N (x-axis)

**图 9.** Movielens 数据集 PREC/Recall (y-axis) vs. N (x-axis)



**Figure 10.** Amazon dataset PREC/Recall (y-axis) vs. N (x-axis)

**图 10.** Amazon 数据集 PREC/Recall (y-axis) vs. N (x-axis)

在具有更丰富用户行为的 Amazon 数据集上, 大多数模型做 Top-N 推荐的准确率都大幅的下降了, 说明工业环境中个性化推荐确实是一个比较复杂的问题。在此数据集上, R-TCN 指标全部位列第一, 好于其他模型, 说明在更复杂的环境中, 能同时捕获长期和短期信息的模型, 扬长避短的模型更有优势。

## 5.2. 时间效率

以 LR 的时间为单位时间, times of LR 分别是各个模型训练时间对单位时间的比值。统计时间效率如表 3 (CPU)和表 4 (GPU)所示, 并将结果可视化对比, 如图 11 所示。

**Table 3.** CPU time efficiency comparison

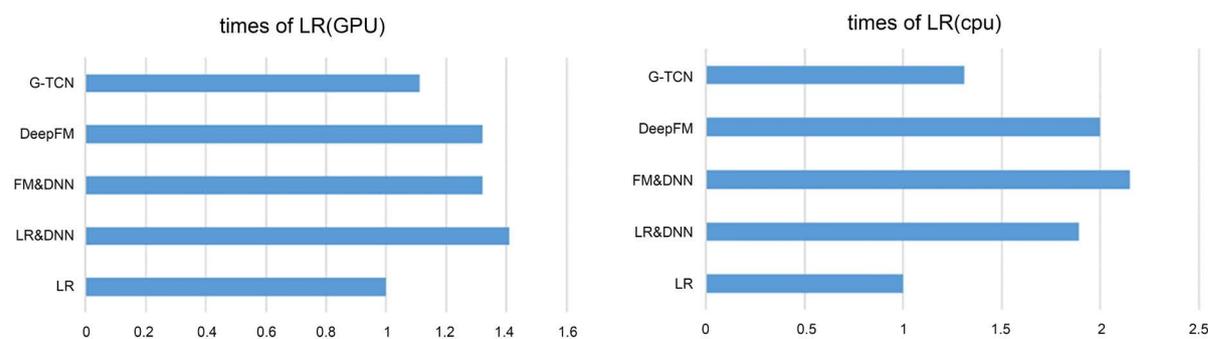
**表 3.** CPU 时间效率对比

model	times of LR (CPU based)
LR	1
LR & DNN	1.89
FM & DNN	2.15
DeepFM	2
R-TCN	1.61

**Table 4.** GPU time efficiency comparison

**表 4.** GPU 时间效率对比

model	times of LR (GPU based)
LR	1
LR & DNN	1.41
FM & DNN	1.32
DeepFM	1.32
R-TCN	1.2



**Figure 11.** Training time comparison

**图 11.** 训练时间比较

LR 由于工业上最广泛的机器学习方法, 得益于其时间效率极高。以 LR 训练时间为基准, 从时间效率指标上看, R-TCN 训练时间均是最短。在 CPU 环境上, R-TCN 领先优势更加明显。分析其原因, 得益于卷积操作的并行化处理效率。在生产环境, 借助于配套的大数据系统, 基于各方面的并行优化, 还有巨大的时间优化的潜力。缩短训练时间对于实现更加灵敏和快速的实时推荐有着重要的意义。

### 5.3. 结论

本文提出了一种新型的网络结构 R-TCN, 基于推荐效果和时间效率两方面的综合比较, 展现出一定的优势。在推荐效果方面, 高于 LR FM DeepFM 等工业上广泛使用的方法。在训练时间方面, 也领先上述相比较的模型, 并还有继续大幅优化的潜力。综合来看, R-TCN 未来具有一定的优化和使用潜力。

### 6. 结束语

针对用户推荐场景的序列预测问题, 本文提出了一种新的方法, R-TCN, 通过理论和实验证明了该方法在达优秀的推荐效果的同时, 还能大幅缩短训练时间, 并且理论上还有继续大幅优化时间效率的潜力。模型更加注重利用短时变化的兴趣信息, 更贴合各种沉浸式 APP 的算法推荐。训练时间缩短, 有助于提高并发瓶颈和实时计算。

### 参考文献

- [1] Zhang, W., Du, T. and Wang, J. (2016) Deep Learning over Multi-Field Categorical Data. In: Ferro, N., *et al.*, Eds., *Advances in Information Retrieval. ECIR 2016. Lecture Notes in Computer Science*, Springer, Cham, 45-57. [https://doi.org/10.1007/978-3-319-30671-1\\_4](https://doi.org/10.1007/978-3-319-30671-1_4)
- [2] Qu, Y., Fang, B., Zhang, W., *et al.* (2018) Product-Based Neural Networks for User Response Prediction over Multi-Field Categorical Data. *ACM Transactions on Information Systems*, **37**, Article No. 5. <https://doi.org/10.1145/3233770>
- [3] Cheng, H.T., Koc, L., Harmsen, J., *et al.* (2016) Wide & Deep Learning for Recommender Systems. *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, Boston, MA, 15 September 2016, 7-10. <https://doi.org/10.1145/2988450.2988454>
- [4] Guo, H., Tang, R., Ye, Y., *et al.* (2017) DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 19-25 August 2017, 1725-1731. <https://doi.org/10.24963/ijcai.2017/239>
- [5] Wang, R., Fu, B., Fu, G. and Wang, M.L. (2017) Deep & Cross Network for Ad Click Predictions. *Proceedings of the ADKDD'17*, Halifax, 14 August 2017, 1-7. <https://doi.org/10.1145/3124749.3124754>
- [6] Zhou, G., Song, C., Zhu, X., *et al.* (2017) Deep Interest Network for Click-Through Rate Prediction. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, United Kingdom, 19-23 August 2018, 1059-1068. <https://doi.org/10.1145/3219819.3219823>
- [7] Cheng, C., Yang, H.Q., Lyu, M.R. and King, I. (2013) Where You Like to Go Next: Successive Point-of-Interest Recommendation. *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, Beijing, China, 3-9 August 2013, 2605-2611.
- [8] He, R. and McAuley, J. (2016) Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. 2016 *IEEE 16th International Conference on Data Mining*, Barcelona, Spain, 12-15 December 2016, 191-200. <https://doi.org/10.1109/ICDM.2016.0030>
- [9] Rendle, S., Freudenthaler, C. and Schmidt-Thieme, L. (2010) Factorizing Personalized Markov Chains for Next-Basket Recommendation. In: *Proceedings of International Conference on World Wide Web*, ACM, New York, 811-820. <https://doi.org/10.1145/1772690.1772773>
- [10] Wang, P.F., Guo, J.F., Lan, Y.Y., Xu, J., Wan, S.X. and Cheng, X.Q. (2015) Learning Hierarchical Representation Model for Next Basket Recommendation. In: *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York, 403-412. <https://doi.org/10.1145/2766462.2767694>
- [11] Bai, S., Kolter, J.Z. and Koltun, V. (2018) An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.
- [12] Vo, N. and Hays, J. (2018) Generalization in Metric Learning: Should the Embedding Layer Be the Embedding Layer? 2019 *IEEE Winter Conference on Applications of Computer Vision*, Waikoloa Village, HI, 7-11 January 2019, 589-598. <https://doi.org/10.1109/WACV.2019.00068>
- [13] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Li, F.-F. (2014) Large-Scale Video Classification with Convolutional Neural Networks. 2014 *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 23-28 June 2014, 1725-1732. <https://doi.org/10.1109/CVPR.2014.223>
- [14] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) ImageNet Classification with Deep Convolutional Neural Net-

- 
- works. In: *Advances in Neural Information Processing Systems*, 1097-1105.
- [15] Kim, Y. (2014) Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, October 2014, 1756-1751.  
<https://doi.org/10.3115/v1/D14-1181>
- [16] Szegedy, C., Ioffe, S., Vanhoucke, V., *et al.* (2016) Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.
- [17] Brazil, T.J. (1995) Causal-Convolution-A New Method for the Transient Analysis of Linear Systems at Microwave Frequencies. *IEEE Transactions on Microwave Theory & Techniques*, **43**, 315-323.  
<https://doi.org/10.1109/22.348090>
- [18] Oord, A.V.D., Dieleman, S., Zen, H., *et al.* (2016) WaveNet: A Generative Model for Raw Audio.  
[19] <https://grouplens.org/datasets/movielens/20m/>
- [20] <http://jmcauley.ucsd.edu/data/amazon/>
- [21] Pan, R., Zhou, Y.H., Cao, B., Liu, N.N., Lukose, R., Scholz, M. and Yang, Q. (2008) One-Class Collaborative Filtering. 2008 *Eighth IEEE International Conference on Data Mining*, Pisa, Italy, 15-19 December 2008, 502-511.  
<https://doi.org/10.1109/ICDM.2008.16>