

# 基于会话的建议与 递归神经网络

**Bala'zs Hidasi**\*

重力研发公司

匈牙利, 布达佩斯

balazs.hidasi@gravityrd.com

**Alexandros Karatzoglou**

Telefonica Research 西班牙

巴塞罗那 alexk@tid.es

**Linus Baltrunas**<sup>†</sup>

网飞公司(Netflix)

美国加州Los Gatos市

lbaltrunas@netflix.com

**多蒙科斯-提克**

重力研发公司

匈牙利, 布达佩斯

domonkos.tikk@gravityrd.com

## 摘要

我们将循环神经网络(RNN)应用于一个新的领域,即推荐人系统。现实生活中的推荐系统经常面临这样的问题,即只能基于短期的会话数据进行推荐(例如一个小型的体育用品网站),而不是长期的用户历史(如Netflix的情况)。在这种情况下,经常被称赞的矩阵分解方法并不准确。这个问题在实践中通常是通过采用项目对项目的推荐来克服的,即推荐类似的项目。我们认为,通过对整个会话进行建模,可以提供更准确的建议。因此,我们提出了一种基于RNN的方法来进行基于会话的推荐。我们的方法也考虑到了任务的实际情况,并对经典的RNNs进行了一些修改,如一个排名损失函数,使其对这个特定的问题更加可行。在两个数据集上的实验结果显示,与广泛使用的方法相比,有明显的改进。

## 1 简介

在机器学习和推荐系统界,基于会话的推荐是一个相对不受重视的问题。许多电子商务推荐系统(尤其是小型零售商的推荐系统)和大多数新闻和媒体网站通常不会跟踪长期访问其网站的用户的用户ID。虽然cookies和浏览器指纹可以提供某种程度的用户可识别性,但这些技术往往不够可靠,而且还会引起隐私问题。即使跟踪是可能的,很多用户在一个较小的电子商务网站上只有一个或两个会话,而在某些领域(如分类网站),用户的行为往往显示出基于会话的特征。因此,同一用户的后续会话应该被独立处理。因此,大多数为电子商务部署的基于会话的推荐系统都是基于相对简单的方法,这些方法并不使用用户资料,例如项目与项目之间的相似性、共同出现或过渡概率。虽然有效,但这些方法往往只考虑到用户的最后一次点击或选择,忽略了过去点击信息。

推荐系统中最常用的方法是因子模型(Koren等人,2009;Weimer等人,2007;Hidasi和Tikk,2012)和邻域方法(Sarwar等人,2001;Ko-ren,2008)。因子模型的工作原理是将稀疏的用户-项目互动矩阵分解为一组 $d$ 维向量,数据集中的每个项目和用户都有一个向量。然后,推荐问题被视为一个矩阵完成/重建问题,潜伏因子向量被用来填补缺失的条目,例如,采取相应的用户-项目潜伏因子的点积。因子模型在基于会话的推荐中很难应用,因为没有

---

\*作者在研究本课题期间在Telefonica Research工作了3个月。

†这项工作是在作者是西班牙巴塞罗那Telefonica研究小组的成员时完成的。

的用户资料。另一方面，邻域方法，依赖于计算项目（或用户）之间的相似关系，是基于项目在会话（或用户资料）中的共同出现。邻域方法已被广泛用于基于会话的推荐中。

过去几年，深度神经网络在一些任务中取得了巨大的成功，比如图像和语音识别（Russakovsky等人，2014；Hinton等人，2012），其中非结构化数据通过几个卷积层和标准层（通常是整流线性）单元来处理。序列数据建模最近也吸引了很多人的注意，各种风味的RNNs是这种类型数据的首选模型。序列建模的应用范围包括从测试-翻译到对话建模到图像字幕。

虽然RNNs已经被应用于上述领域，并取得了显著的成功，但对推荐系统领域的关注却很少。在这项工作中，我们认为RNN可以应用于基于会话的推荐，并取得了显著的效果，我们处理了在对这种稀疏的连续数据进行建模时出现的问题，并通过引入适合训练这些模型的新排名损失函数，使RNN模型适应推荐者的设置。基于会话的推荐问题与一些NLP相关问题在建模方面有一些相似之处，因为它们都是处理序列的。在基于会话的推荐中，我们可以把用户进入网站时点击的第一个项目作为RNN的初始输入，然后我们想根据这个初始输入来查询模型的推荐。用户的每一次连续点击都会产生一个取决于之前所有点击的输出（推荐）。通常情况下，在推荐系统中可供选择的项目集可能有几万个甚至几十万个。除了项目集的巨大规模之外，另一个挑战是点击流数据集通常相当大，因此训练时间和可扩展性真的很重要。正如大多数信息检索和推荐设置一样，我们感兴趣将建模能力集中在用户可能感兴趣的顶级项目上，为此我们使用排名损失函数来训练RNN。

## 2 相关的工作

### 2.1 基于会话的推荐

推荐系统领域的大部分工作都集中在当用户标识符可用并且可以建立一个清晰的用户档案时的模型。在这种情况下，矩阵分解方法和邻域模型在文献中占主导地位，并且也被在线采用。在基于会话的推荐中采用的主要方法之一，以及对缺失的用户资料问题的自然解决方案是项目对项目的推荐方法（Sarwar等人，2001；Linden等人，2003），在这种情况下，项目对项目的相似性矩阵是从可用的会话数据中预先计算的，也就是说，在会话中经常一起点击的项目被认为是相似的。然后，这个相似性矩阵在会话中被简单地用来推荐与用户当前点击的项目最相似的项目。虽然简单，但这种方法已被证明是有效的，并被广泛采用。虽然有效，但这些方法只考虑到用户的最后一次点击，实际上是忽略了过去点击信息。

基于会话的推荐的一个有点不同的方法是马尔科夫决策过程（MDPs）（Shani等人，2002）。MDPs是顺序随机决策问题的模型。一个MDP被定义为一个四元组 $\langle S, A, Rwd, tr \rangle$ ，其中 $S$ 是一组状态， $A$ 是一组行动， $Rwd$ 是一个奖励函数， $tr$ 是状态转换函数。在推荐系统中，行动可以等同于推荐，最简单的MPD本质上是一阶马尔可夫链，其中下一个推荐可以根据项目之间的过渡概率简单计算。在基于会话的推荐中应用马尔可夫链的主要问题是，当试图包括所有可能的用户选择序列时，状态空间很快就变得无法管理。

通用因子化框架（GFF）的扩展版本（Hidasi & Tikk, 2015）可以使用会话数据进行推荐。它通过其事件的总和来模拟会话。它为项目使用了两种潜在的表示方法，一种是代表项目本身，另一种是代表项目作为会话的一部分。然后，会话被表示为部分会话项目表示的特征向量的平均值。然而，这种方法并不考虑会话中的任何排序。

## 2.2 推荐者中的深度学习

神经网络文献中最早的相关方法之一是使用受限波尔兹曼机（RBM）进行协同过滤（Salakhutdinov等人，2007）。在这项工作中，RBM被用来模拟用户与项目的交互，并进行推荐。这个模型已经被证明是性能最好的协同过滤模型之一。深度模型被用来从非结构化的内容中提取特征，如音乐或图像，然后与更传统的协同过滤模型一起使用。在Van den Oord等人（2013）的研究中，一个卷积深度网络被用来从音乐文件中提取特征，然后用于因子模型。最近，Wang等人（2015）引入了一种更通用的方法，即使用深度网络从任何类型的项目中提取通用的内容特征，然后将这些特征纳入标准的协同过滤模型以提高推荐性能。这种方法似乎在没有足够的用户-项目互动信息的情况下特别有用。

## 3 与护士的建议

递归神经网络已被设计用来对可变长度的序列数据进行建模。RNNs和传统的前馈深层模型的主要区别在于组成网络的单元中存在一个内部隐藏状态。标准的RNN使用以下更新函数来更新其隐藏状态 $h$ ：

$$h_t = g(W x_t + U h_{t-1}) \quad (1)$$

其中 $g$ 是一个平滑且有界的函数，如Logistic sigmoid函数 $x_t$ 是单元在时间 $t$ 的输入。RNN在给定其当前状态 $h$ 的情况下，对序列的下一个元素输出一个概率分布 $t$ 。

门控递归单元（GRU）（Cho等人，2014）是一个更精细的RNN单元模型，旨在处理梯度消失的问题。GRU门本质上是学习何时以及以何种程度来更新单元的隐藏状态。GRU的激活是以前的激活和候选激活 $\hat{h}_t$ 之间的线性插值：

$$h_t = (1 - z_t) h_{t-1} + z_t \hat{h}_t \quad (2)$$

其中，更新门由以下公式给出：

$$z_t = \sigma(W x_{zt} + U h_{zt-1}) \quad (3)$$

而候选激活函数 $\hat{h}_t$ 也是以类似的方式计算的：

$$\hat{h}_t = \tanh(W x_t + U (r_t \odot h_{t-1})) \quad (4)$$

最后，复位门 $r_t$ ，由以下公式给出：

$$r_t = \sigma(W x_{rt} + U h_{rt-1}) \quad (5)$$

### 3.1 定制格鲁模式

我们在基于会话的推荐模型中使用了基于GRU的RNN。网络的输入是会话的实际状态，而输出是会话中下一个事件的项目。会话的状态可以是实际事件的项目，也可以是迄今为止会话中的事件。在前一种情况下，使用1-of-N编码，即输入向量的长度等于项目的数量，并且只有对应于活动项目的坐标为1，其他为0。后一种设置使用这些表示的加权和，在这种情况下，如果事件发生得早，就会被打折扣。为了稳定起见，输入向量被归一化。我们希望这能有所帮助，因为它强化了记忆效应：强化了非常局部的排序约束，而这些约束并没有被RNN的较长记忆所很好地捕捉到。我们还试验了增加一个额外的嵌入层，但1-of-N编码总是表现得更好。

网络的核心是GRU层，在最后一层和输出之间可以添加额外的前馈层。输出是项目的预测

偏好，即每个项目在会议中成为下一个的可能性。当使用多个GRU层时，前一个层的隐藏状态是下一个层的输入。输入也可以选择性地连接

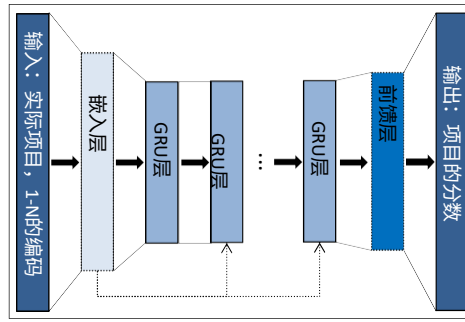


图1: 网络的一般结构。一次性处理一个事件的事件流。

到网络中更深的GRU层，因为我们发现这能提高性能。请看图1上的整个架构，它描述了一个事件的时间序列中单个事件的表现。

由于推荐系统不是递归神经网络的主要应用领域，我们修改了基础网络以更好地适应这一任务。我们还考虑了一些实际问题，以便我们的解决方案可以在实际环境中应用。

### 3.1.1 会话并行的小型批处理

用于自然语言处理任务的RNN通常使用序列中的小批量。例如，常见的是在句子的单词上使用滑动窗口，并将这些窗口片段放在彼此旁边，形成小批量。这并不适合我们的任务，因为（1）会话的长度可能非常不同，甚至比句子的长度还长：有些会话只包括两个事件，而其他的可能超过几百个；（2）我们的目标是捕捉一个会话是如何随时间演变的，所以分解成碎片没有意义。因此，我们使用会话并行的小批次。首先，我们为会话创建一个顺序。然后，我们使用前 $x$ 个会话的第一个事件来形成第一个小批处理的输入（期望的输出是我们活动会话的第二个事件）。第二个小批处理是由第二个事件形成的，以此类推。如果任何一个会话结束，下一个可用的会话将被放在其位置上。会话被假定为是独立的，因此当这种切换发生时，我们会重置适当的隐藏状态。更多细节见图2。

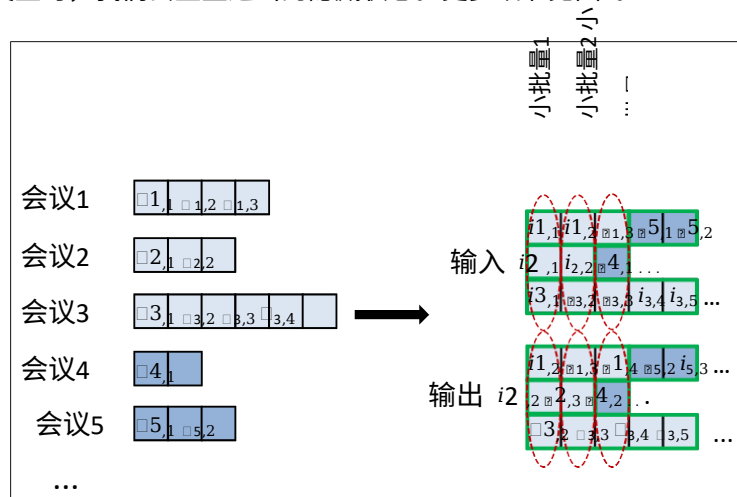


图2: 会话并行的小批量创建

### 3.1.2 对输出进行采样

当商品的数量很大时，推荐系统就特别有用。即使是 ， 对于一个中等规模的网店来说，这也是数以万计的，但在更大的网站上，有这样的情况并不罕见。

数十万个项目，甚至几百万个项目。在每个步骤中为每个项目计算一个分数将使算法与项目数量和事件数量的乘积成比例。这在实践中是无法使用的。因此，我们必须对输出进行采样，只计算一小部分项目的得分。这也就意味着只有部分权重会被更新。除了期望的输出，我们还需要计算一些负面的例子的分数，并修改权重，使期望的输出获得高排名。

对一个任意的缺失事件的自然解释是，用户不知道这个项目的存在，因此没有互动。然而，有一个很低的概率是，用户确实知道这个项目，但选择不互动，因为她不喜欢这个项目。项目越受欢迎，用户知道它的可能性就越大，因此缺失的事件更有可能是表达不喜欢。因此，我们应该根据项目的受欢迎程度按比例取样。我们没有为每个训练实例生成单独的样本，而是使用小批量的其他训练实例中的项目作为负面的例子。这种方法的好处是，我们可以通过跳过采样来进一步减少计算时间。此外，在实施方面也有好处，从使代码不那么复杂到加快矩阵操作。同时，这种方法也是一种基于流行度的抽样，因为一个项目出现在小批量的其他训练实例中的可能性与它的流行度成正比。

### 3.1.3 排名 亏损

推荐系统的核心是对项目进行基于相关性的排名。虽然这项任务也可以被解释为分类任务，但学习排名的方法（Rendle等人，2009；Shi等人，2012；Steck，2015）通常比其他方法更有优势。排名可以是点式、对式或列表式。点式排名是独立估计项目的得分或排名，损失的定义方式是相关项目的排名应该很低。成对排序比较了正反两方面项目的得分或排名，损失要求正方项目的排名应低于反方项目的排名。列表式排序使用所有项目的分数和等级，并将它们与完美排序进行比较。由于它包括排序，它通常在计算上比较昂贵，因此不常使用。另外，如果只有一个相关项目--如我们的案例--列表式排名可以通过成对排名来解决。

我们在解决方案中包括了几个点式和对式排名的损失。我们发现，在这个网络中，点式排名是不稳定的（更多评论见第4节）。另一方面，成对的排名损失则表现良好。我们使用了以下两种。

- **BPR**: Bayesian Personalized Ranking (Rendle等人，2009) 是一种使用成对排名损失的矩阵分解方法。它比较了一个正面项目和一个抽样的负面项目的得分。在这里

我们将正面项目的分数与几个抽样项目进行比较，并使用他们的平均数为损失。一次会议中某一点的损失定义为： $L_s = \frac{1}{N_s} \sum_{j=1}^{N_s} \log(\sigma(r_{s,i} - r_{s,j}))$ ，其中 $N_s$ 是样本量， $r_{s,k}$ 是项目 $k$ 的得分。

在会话的给定点， $i$ 是期望的项目（会话中的下一个项目）， $j$ 是负面样本。

- **TOP1**: 这个排名损失是我们为这个任务设计的。它是对相关项目的相对排名的正则化近似。相关项目的相对排名由以下公式给出

$$L_s = \frac{1}{N_s} \sum_{j=1}^{N_s} \log(\sigma(r_{s,j} - r_{s,i})) + \sigma r_{s,i}^2$$

## 4 实验



我们在两个数据集上对拟议的递归神经网络与流行的基线进行了评估。

第一个数据集是2015年RecSys挑战赛的数据集。<sup>1</sup>这个数据集包含一个电子商务网站的点击流，有时会以购买事件结束。我们使用该挑战赛的训练集，只保留点击事件。我们过滤掉长度为1的会话。网络是在~6个月的数据上训练的，包含7,966,257个会话，31,637,239次点击，37,483个项目。我们使用随后一天的会议进行测试。每个会话都被分配到训练集或测试集，我们不在会话中分割数据。由于协作过滤方法的性质，我们从测试集中过滤掉点击项目不在训练集中的点击。长度为1的会话也会从测试集中删除。在预处理之后，我们留下了15,324个会话和71,222个事件作为测试集。这个数据集将被称为RSC15。

第二个数据集是从一个类似Youtube的OTT视频服务平台收集的。至少在一定时间内观看视频的事件被收集。只有某些地区受到这种收集的影响，持续时间略短于2个月。在此期间，在屏幕左侧的每个视频之后，都会提供项目对项目的推荐。这些是由不同的算法提供的，并影响了用户的行为。预处理步骤与其他数据集类似，但要过滤掉很长的时段，因为它们可能是由机器人产生的。训练数据包括除上述期间最后一天以外的所有数据，有300万个会话，涉及33万个视频的1300万个观看事件。测试数据集包含收集期最后一天的会话，有3.7万个会话。~ 18万个观看事件。这个数据集将被称为视频。

评估是通过逐一提供一个会话的事件并检查下一个事件的项目的等级来完成的。一个会话结束后，GRU的隐藏状态被重置为零。项目按其得分从高到低排序，它们在这个列表中的位置就是它们的等级。在RSC15中，火车组的所有37483个项目都被排序。然而，由于项目数量众多，用VIDEO的话就不太现实了。在那里，我们将所需的项目与最受欢迎的30,000个项目进行排名。这对评价的影响可以忽略不计，因为很少被访问的项目经常得到低分。另外，基于人气的预过滤在实际的推荐系统中是很常见的。

由于推荐系统一次只能推荐几个项目，用户可能选择的实际项目应该在列表的前几个项目中。因此，我们的主要评估指标是召回率@20，即在所有测试案例中，所需项目在前20个项目中的比例。召回率并不考虑项目的实际排名，只要它在前N个项目中。这很好地模拟了某些实际情况，在这些情况下，没有突出推荐，绝对顺序并不重要。召回率通常也与重要的在线关键绩效指标相关，例如点击率（CTR）（Liu等人，2012；Hidasi & Tikk，2012）。实验中使用的第二个指标是MRR@20（平均互换排名）。这是所需项目的互惠排名的平均值。如果倒数排名高于20，则倒数排名被设置为零。MRR考虑到了项目的排名，这在推荐顺序很重要的情况下是很重要的（例如，排名较低的项目只有在滚动后才能看到）。

#### 4.1 基准线

我们将提议的网络与一组常用的基线进行比较。

- **POP**：人气预测器，总是推荐训练集中最受欢迎的项目。尽管它很简单，但在某些领域往往是一个强大的基线。
- **S-POP**：该基线推荐当前会议中最受欢迎的项目。在会议期间，随着项目获得更多的事件，推荐列表会发生变化。平局是用全局流行值来打破的。这个基线在具有高重复性的领域很强。
- **项目-KNN**：与实际项目相似的项目由该基线推荐，相似度定义为它们的会话向量之间的余弦相似度，即两个项目在会话中共同出现的次数除以单个项目出现的会话数量乘积的平方根。正则化也被包括在内，以避免很少访问的项目出现巧合的高相

似度。这个基线是实际系统中最常见的项目对项目的解决方案之一，它在 "其他浏览过这个项目的人也浏览过这些项目 "的情况下提供建议。尽管它很简单，但通常是一个强大的基线（Linden等人，2003；Davidson等人，2010）。

---

<sup>1</sup><http://2015.recsyschallenge.com/>

表1: 使用基线方法的召回率@20和MRR@20

基准线	RSC15		视频	
	召回@20	MRR@20	召回@20	记忆力@20
持续性	0.0050	0.0012	0.0499	0.0117
S-POP	0.2672	0.1775	0.1301	0.0863
项目-KNNN	0.5065	0.2048	0.5508	0.3381
BPR-MF	0.2574	0.0618	0.0692	0.0374

表2: 数据集/损失函数的最佳参数化

数据集	损失	小批量	辍学	学习率	势头
RSC15	TOP1	50	0.5	0.01	0
RSC15	BPR	50	0.2	0.05	0.2
RSC15	交叉熵	500	0	0.01	0
视频	TOP1	50	0.4	0.05	0
视频	BPR	50	0.3	0.1	0
视频	交叉熵	200	0.1	0.05	0.3

- **BPR-MF**: BPR-MF (Rendle等人, 2009) 是常用的矩阵分解方法之一。它通过SGD优化成对排名的目标函数 (见第3节)。矩阵分解法不能直接应用于基于会话的推荐, 因为新的会话没有预先计算的特征向量。然而, 我们可以通过使用迄今为止在会话中出现过的项目特征向量的平均值作为用户特征向量来克服这个问题。换句话说, 我们对可推荐的项目和迄今为止会话中的项目之间的特征向量的相似性进行平均。

表1显示了基线的结果。项目-KNN方法显然在其他方法中占优势。

#### 4.2 参数和结构优化

我们通过在每个数据集和损失函数的参数空间的随机选择点运行100个实验来优化超参数。通过单独优化每个参数, 进一步调整了最佳参数化。在所有情况下, 隐藏单元的数量被设置为100。然后将性能最好的参数用于不同大小的隐藏层。优化是在一个单独的验证集上完成的。然后在训练集和验证集上重新训练网络, 并在最后的测试集上进行评估。

表2中总结了性能最好的参数。权重矩阵由从 $[-x, x]$ 中均匀抽取的随机数初始化, 其中 $x$ 取决于矩阵的行和列的数量。我们用rmsprop (Dauphin等人, 2015) 和adagrad (Duchi等人, 2011) 进行了实验。我们发现adagrad的结果更好。

我们简单地试验了GRU以外的其他单元。我们发现经典的RNN单元和LSTM的表现都比较差。

我们尝试了几种损失函数。基于点式排名的损失, 如交叉熵和MRR优化 (如Steck (2015)) 通常是不稳定的, 即使有正规化。例如, 交叉熵在RSC15和VIDEO的100次随机运行中分别只产生了10和6个数值稳定的网络。我们假设这是由于独立地试图为所需的项目取得高分, 而且对于负面的样本来说, 负面的推动力是很小的。另一方面, 基于配对排名的损失表现良好。我们发现第3节中介绍的那些 (BPR和TOP1) 表现最好。

我们研究了几种结构, 发现单层的GRU单元是表现最好的。增加层数总是导致在测试集上

测量的训练损失、召回率和MRR方面的最差表现。我们认为这是由于一般情况下短的

表3：不同类型的单层GRU的召回率@20和MRR@20，与最佳基线（项目-KNN）相比。每个数据集的最佳结果被突出显示。

损失/#单位	RSC15		视频	
	召回@20	MRR@20	召回@20	MRR@20
TOP1 100	0.5853 (+15.55%)	0.2305 (+12.58%)	0.6141 (+11.50%)	0.3511 (+3.84%)
BPR 100	0.6069 (+19.82%)	0.2407 (+17.54%)	0.5999 (+8.92%)	0.3260 (-3.56%)
交叉熵 100	0.6074 (+19.91%)	0.2430 (+18.65%)	0.6372 (+15.69%)	0.3720 (+10.04%)
TOP1 1000	0.6206 (+22.53%)	<b>0.2693 (+31.49%)</b>	<b>0.6624 (+20.27%)</b>	<b>0.3891 (+15.08%)</b>
BPR 1000	<b>0.6322 (+24.82%)</b>	0.2467 (+20.47%)	0.6311 (+14.58%)	0.3136 (-7.23%)
交叉熵 1000	0.5777 (+14.06%)	0.2153 (+5.16%)	-	-

会议的寿命不需要不同分辨率的多个时间尺度来正确表示。然而这其中的确切原因目前还不清楚，需要进一步研究。使用项目的嵌入，结果稍差，因此我们保留了1-of-N编码。另外，把所有以前的事件放在输入端而不是前面的事件上，并没有导致额外的准确性提高；这并不奇怪，因为GRU--像LSTM--有长期和短期的记忆。在GRU层之后添加额外的前馈层也没有帮助。然而，增加GRU层的大小却提高了性能。我们还发现，使用tanh作为输出层的激活函数是有益的。

#### 4.3 结果

表3显示了表现最好的网络的结果。有1000个隐藏单元的VIDEO数据的交叉熵在数值上是不稳定的，因此我们没有介绍这种情况下的结果。这些结果与最佳基线（item-KNN）进行了比较。我们展示了100和1000个隐藏单元的结果。运行时间取决于参数和数据集。一般来说，在GeForce GTX Titan X GPU上，较小的变体和较大的变体之间的运行时间差异不大，网络的训练可以在几个小时内完成。<sup>2</sup>在CPU上，较小的网络可以在一个实际可接受的时间范围内进行训练。对于推荐系统来说，频繁的再训练通常是可取的，因为新的用户和项目经常被引入。

在两个数据集上，基于GRU的方法在两个评价指标上都比item-KNN有很大的优势，即使单元数为100。<sup>3</sup>增加单元数可以进一步改善成对损失的结果，但是交叉熵的准确度却下降了。即使交叉熵在100个隐藏单元的情况下给出了更好的结果，但随着单元数的增加，配对损失的变体超过了这些结果。尽管增加单元数会增加训练时间，但我们发现在GPU上从100个单元增加到1000个单元的成本并不高。另外，基于交叉熵的损失被发现在数值上是不稳定的，因为网络单独试图增加目标项目的得分，而对其他项目的负推力相对较小。因此，我们建议使用这两个配对损失中的任何一个。TOP1损失在这两个数据集上的表现稍好，导致比表现最好的基线高出20-30%的准确性。

## 5 结论和未来工作

在本文中，我们将一种现代循环神经网络（GRU）应用于新的应用：推荐系统。我们选择了基于会话的推荐任务，因为它是一个实际的重要领域，但没有得到很好的研究。为了更好地适应这一任务，我们对基本的GRU进行了修改，引入了会话并行的小批，基于小批的输出采样和排名损失函数。我们表明，我们的方法可以大大超过用于该任务的流行基线。我们认为，我们的工作可以成为深度学习在推荐系统中的应用和基于会话的一般推荐的基

础。

---

<sup>2</sup>在GPU上使用Theano，并对副张量运算符进行了修正。

<sup>3</sup>除了在VIDEO数据上使用BPR损失，并对MRR进行评估。

我们近期的工作将集中在对拟议的网络进行更彻底的检查。我们还计划在自动提取的项目表征上训练网络，这种表征是建立在项目本身的内容（如缩略图、视频、文本）而不是当前的输入上。

鸣谢

导致这些结果的工作得到了欧盟第七框架工作计划（FP7/2007-2013）的资助，根据 CrowdRec 资助协议 n° 610594。

## 参考文献

- Cho, Kyunghyun, van Merriënboer, Bart, Bahdanau, Dzmitry, and Bengio, Yoshua. 论神经机器翻译的恰当性: *arXiv preprint arXiv:1409.1259*, 2014.
- Dauphin, Yann N, de Vries, Harm, Chung, Junyoung, and Bengio, Yoshua. *arXiv preprint arXiv:1502.04390*, 2015.
- Davidson, James, Liebald, Benjamin, Liu, Junning, et al. The YouTube video recommendation system. In *Recsys'10: ACM Conf. on Recommender Systems*, pp.293-296, 2010. ISBN 978-1-60558-906-0.
- Duchi, John, Hazan, Elad, and Singer, Yoram. 用于在线学习和随机优化的自适应子梯度方法. *The Journal of Machine Learning Research*, 12:2121-2159, 2011.
- Hidasi, B. and Tikk, D. Fast ALS-based tensor factorization for context-aware recommendation from implicit feedback. In *ECML-PKDD'12, Part II*, number 7524 in LNCS, pp.67-82. Springer, 2012.
- Hidasi, Bala'zs and Tikk, Domonkos. 语境感知推荐的一般因子化框架. *数据挖掘和知识发现*, 第 1-30 页, 2015 年。ISSN 1384-5810. doi: 10.1007/s10618-015-0417-y. URL <http://dx.doi.org/10.1007/s10618-015-0417-y>.
- Hinton, Geoffrey, Deng, Li, Yu, Dong, Dahl, George E, Mohamed, Abdel-rahman, Jaitly, Navdeep, Senior, Andrew, Vanhoucke, Vincent, Nguyen, Patrick, Sainath, Tara N, 等. Deep neural networks for acoustic modeling in speech recognition: 四个研究小组的共同观点. *信号处理杂志, IEEE*, 29 (6) : 82-97, 2012.
- Koren, Y. Factorization meets the neighborhood: a multi-faceted collaborative filtering model. In *SIGKDD'08: ACM Int. Conf. on Knowledge Discovery and Data Mining*, pp.426-434, 2008.
- Koren, Yehuda, Bell, Robert, and Volinsky, Chris. 推荐系统的矩阵分解技术. *计算机*, 42 (8) : 30-37, 2009.
- Linden, G., Smith, B., and York, J. Amazon. com 推荐: Item-to-item collaborative filtering. *互联网计算, IEEE*, 7 (1) : 76-80, 2003.
- 刘启文, 陈天健, 蔡静, 于殿海. Enlister: 百度对最大的中文问答网站的推荐系统. In *RecSys-12: Proc. of the 6th ACM Conf. on Recommender Systems*, pp.285-288, 2012.
- Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. BPR: 来自隐性反馈的贝叶斯个性化排名. In *UAI'09: 25th Conf. on Uncertainty in Artificial Intelligence*, pp.452-461, 2009. ISBN 978-0-9749039-5-8.



Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael S., Berg, Alexander C. and Li, Fei-Fei. Imagenet 大规模视觉识别挑战。 *CoRR*, abs/1409.0575, 2014. URL <http://arxiv.org/abs/1409.0575>.

- Salakhutdinov, Ruslan, Mnih, Andriy, and Hinton, Geoffrey.用于协同过滤的受限波尔兹曼机。在 *第24届机器学习国际会议论文集*中, pp.791-798.ACM, 2007.
- Sarwar, Badrul, Karypis, George, Konstan, Joseph, and Riedl, John.基于项目的协同过滤推荐算法。在 *第十届万维网国际会议论文集*中, 第285-295页。ACM, 2001.
- Shani, Guy, Brafman, Ronen I, and Heckerman, David.一个基于mdp的推荐系统。在 *第十八次人工智能不确定性会议论文集*中, 第453-460页。Morgan Kaufmann Publishers Inc., 2002.
- Shi, Yue, Karatzoglou, Alexandros, Baltrunas, Linas, Larson, Martha, Oliver, Nuria, and Hanjalic, Alan.Climf: 学习用协作式的少即是多过滤来最大化互惠等级。In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pp.139-146, New York, NY, USA, 2012.ACM.ISBN 978-1-4503-1270-7. doi: 10.1145/2365952.2365981. URL <http://doi.acm.org/10.1145/2365952.2365981>.
- Steck, Harald.通过矩阵因子化进行高斯排名。In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pp.115-122, New York, NY, USA, 2015.ACM.ISBN 978-1-4503-3692-5. doi: 10.1145/2792838.2800185.url <http://doi.acm.org/10.1145/2792838.2800185>.
- Van den Oord, Aaron, Dieleman, Sander, and Schrauwen, Benjamin.基于内容的深度音乐推荐。In *Advances in Neural Information Processing Systems*, pp.2643-2651, 2013.
- Wang, Hao, Wang, Naiyan, and Yeung, Dit-Yan.用于推荐系统的协作式深度学习。In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pp.1235-1244, New York, NY, USA, 2015.ACM.
- Weimer, Markus, Karatzoglou, Alexandros, Le, Quoc Viet, and Smola, Alex.Max margin matrix factorization for collaborative ranking. *神经信息处理系统的进展*, 2007.