

SESSION-BASED RECOMMENDATIONS WITH RECURRENT NEURAL NETWORKS

ABSTRACT

우리는 새로운 도메인, 즉 추천 시스템에 반복 신경망(RNN)을 적용한다. 실제 추천 시스템은 종종 (사례와 같이) 긴 사용자 이력 대신 짧은 세션 기반 데이터(예: 작은 스포츠웨어 웹 사이트)에만 추천을 적용해야 하는 문제에 직면한다. 이 상황에서는 자주 칭찬받는 행렬 인수 분해 접근법이 정확하지 않다. 이 문제는 대개 항목별 권장 사항, 즉 유사한 항목을 권장함으로써 실제로 극복된다. 우리는 전체 세션을 모델링함으로써 보다 정확한 추천을 제공할 수 있다. 따라서 세션 기반 추천에 대한 RNN 기반 접근 방식을 제안한다. 우리의 접근 방식은 또한 작업의 실질적인 측면을 고려하고 이 특정 문제에 대해 더 실행 가능하게 하는 순위 손실 함수와 같은 고전적인 RNN에 대한 몇 가지 수정을 도입한다. 두 데이터 세트에 대한 실험 결과는 널리 사용되는 접근법에 비해 현저한 개선을 보여준다.

1. INTRODUCTION

세션 기반 추천은 머신 러닝 및 추천 시스템 커뮤니티에서 상대적으로 인식되지 않는 문제이다. 많은 전자상거래 추천 시스템(특히 소규모 소매업체의 추천 시스템)과 대부분의 뉴스 및 미디어 사이트는 일반적으로 장기간에 걸쳐 사이트를 방문하는 사용자의 사용자 ID를 추적하지 않는다. 쿠키와 브라우저 지문이 어느 정도 사용자 인식 수준을 제공할 수 있지만, 기술은 종종 충분히 신뢰할 수 없고 더 나아가 사생활에 대한 우려를 제기한다. 추적이 가능하더라도 많은 사용자는 더 작은 전자상거래 사이트에서 한 두 개의 세션만 가지고 있으며, 특정 도메인(예: 기밀 사이트)에서는 사용자의 행동이 세션 기반 특성을 나타내는 경우가 많다. 따라서 동일한 사용자의 후속 세션은 독립적으로 처리되어야 한다. 결과적으로, 전자 상거래를 위해 배치된 대부분의 세션 기반 추천 시스템은 항목 간 유사성, 동시 발생 또는 전환 확률과 같은 사용자 프로필을 사용하지 않는 비교적 간단한 방법에 기초한다. 이러한 방법은 효과적이지만, 과거 클릭의 정보를 무시한 사용자의 마지막 클릭 또는 선택만 고려하는 경우가 많다.

추천 시스템에 사용되는 가장 일반적인 방법은 요인 모델(Korean et al., 2009; Weimer et al., 2007; Hidasi & Tikk, 2012)과 근린 방법(Sarwar et al., 2001; Korean, 2008)이다. 요인 모델은 희소 사용자 항목 상호 작용 행렬을 데이터 세트의 각 항목 및 사용자에 대해 하나씩 d차원 벡터 집합으로 분해하여 작동한다. 그런 다음 추천 문제는 매트릭스 완료/재구성 문제로 처리되며, 잠재 인자 벡터는 예를 들어 해당 사용자 항목 잠재 인자의 도트 곱을 취함으로써 누락된 항목을 채우는 데 사용된다. 사용자 프로필이 없기 때문에 세션 기반 권장 사항에 요인 모델을 적용하기가 어렵다. 반면에 항목(또는 사용자) 간의 유사성 계산에 의존하는 근린 방법은 세션(또는 사용자 프로필)의 항목 동시 발생에 기초한다. 근린 방법은 세션 기반 권장 사항에 광범위하게 사용되어 왔다.

지난 몇 년 동안 이미지 및 음성 인식과 같은 많은 작업(Russakovsky et al., 2014; Hinton et al., 2012)에서 심층 신경망의 엄청난 성공을 보아왔다. 이 작업에서는 구조화되지 않은 데

이터가 여러 컨볼루션 및 표준 계층(일반적으로 수정된 선형)을 통해 처리된다. 순차적 데이터 모델링도 최근 많은 관심을 끌고 있으며, 다양한 RNN이 이러한 유형의 데이터에 대한 선택 모델이 되고 있다. 시퀀스 모델링의 적용 범위는 테스트 번역부터 대화 모델링, 이미지 캡션까지 다양하다.

RNN은 앞서 언급한 도메인에 적용되었지만, 추천 시스템 영역에 거의 주의를 기울이지 않았다. 이 연구에서 우리는 RNN이 놀라운 결과를 가진 세션 기반 추천에 적용될 수 있다고 주장한다. 우리는 이러한 희소한 순차 데이터를 모델링할 때 발생하는 문제를 처리하고 적합한 새로운 순위 손실 기능을 도입하여 추천자 설정과 이러한 모델을 훈련시키는 작업에 RNN 모델을 적용한다. 세션 기반 추천 문제는 둘 다 시퀀스를 다루는 한 모델링 측면에서 일부 NLP 관련 문제와 몇 가지 유사점을 공유한다. 세션 기반 추천 사항에서 사용자가 웹 사이트를 입력할 때 클릭하는 첫 번째 항목을 RNN의 초기 입력으로 고려할 수 있으며, 그런 다음 추천 사항을 위해 이 초기 입력을 기반으로 모델을 쿼리하려고 한다. 그런 다음 사용자가 연속적으로 클릭할 때마다 이전 클릭에 따라 달라지는 출력(추천)이 생성된다. 일반적으로 추천 시스템에서 선택할 수 있는 항목 집합은 수만 또는 수십만 개일 수 있다. 항목 세트의 큰 크기와 별개로, 클릭 스트림 데이터 세트는 일반적으로 상당히 크기 때문에 교육 시간과 확장성이 매우 중요하다는 것이 또 다른 과제이다. 대부분의 정보 검색 및 추천 설정과 마찬가지로, 우리는 사용자가 관심을 가질 수 있는 상위 항목에 모델링 능력을 집중하는 데 관심이 있으며, 이를 위해 순위 손실 함수를 사용하여 RNN을 훈련시킨다.

2. RELATED WORK

2.1 SESSION-BASED RECOMMENDATION

추천 시스템 영역의 많은 작업은 사용자 식별자를 사용할 수 있고 명확한 사용자 프로필을 구축할 수 있을 때 작동하는 모델에 초점을 맞추었다. 이 설정에서 행렬 분해 방법과 이웃 모델은 문헌을 지배했으며 온라인에서도 사용된다. 세션 기반 추천과 누락된 사용자 프로필 문제에 대한 자연적 해결책에 채택된 주요 접근법 중 하나는 item-to-item 접근법(Sarwar et al., 2001; Linden et al., 2003)이다. 항목 유사성 매트릭스에 대한 이 설정은 사용 가능한 세션 데이터에서 미리 계산된다. 즉, 세션에서 자주 함께 클릭되는 항목은 유사한 것으로 간주된다. 이 유사성 매트릭스는 세션 중에 사용자가 현재 클릭한 것과 가장 유사한 항목을 추천하는 데 사용된다. 이 방법은 간단하지만 효과가 입증되었으며 널리 사용된다. 효과적이기는 하지만 이러한 방법은 사용자의 마지막 클릭만 고려하며 사실상 과거 클릭 정보는 무시한다.

세션 기반 추천에 대한 다소 다른 접근 방식은 마르코프 의사 결정 과정(MDP)이다(Shani 등, 2002). MDP는 순차적 확률적 의사결정 문제의 모델이다. MDP는 4개의 튜플 S, A, R, γ 로 정의된다. 여기서 S 는 상태 집합이고, A 는 동작 집합 R 는 보상 함수이고 γ 는 상태 전이 함수이다. 추천 시스템에서 동작은 추천과 동일시될 수 있으며 가장 간단한 MDP는 본질적으로 1차 마르코프 체인으로, 다음 권장 사항은 항목 간의 전환 확률을 기반으로 간단히 계산할 수 있다. 세션 기반 추천에 마르코프 체인을 적용하는 경우의 주요 문제는 가능한 모든 사용자 선택 시퀀스를 포함하려고 할 때 상태 공간이 빠르게 관리할 수 없게 된다는 것이다.

일반 인수 분해 프레임워크(GFF)의 확장 버전(Hidasi & Tikk, 2015)은 추천에 세션 데이터를 사용할 수 있다. 이벤트 합계를 기준으로 세션을 모델링한다. 항목에는 두 가지 종류의 잠재 표현을 사용한다. 하나는 **항목 자체를 나타내고 다른 하나는 항목을 세션의 일부로 나타내기 위한 것이다.** 그런 다음 세션은 세션 부분 항목 표현의 특징 벡터의 평균으로 표시된다. 그러나 이 접근법은 세션 내의 어떤 순서도 고려하지 않는다.

2.2 DEEP LEARNING IN RECOMMENDERS

협업 필터링을 위해 제한된 볼츠만 기계(RBM)를 사용하는 신경망 문헌의 첫 번째 관련 방법 중 하나이다(Salakhutdinov 등, 2007). 이 작업에서 RBM은 사용자 항목 상호 작용을 모델링하고 추천을 수행하는 데 사용된다. 이 모델은 가장 성능이 우수한 협업 필터링 모델 중 하나로 나타났다. 딥 모델은 음악 또는 이미지와 같은 구조화되지 않은 콘텐츠에서 기능을 추출하는 데 사용되었으며, 이는 더 전통적인 협업 필터링 모델과 함께 사용된다. Van den Oord 등(2013)에서는 컨볼루션 심층 네트워크를 사용하여 음악 파일에서 특징을 추출한 다음 요소 모델에 사용된다. 보다 최근에 Wang 등(2015)은 심층 네트워크를 사용하여 모든 유형의 항목에서 일반 콘텐츠 기능을 추출하는 보다 일반적인 접근 방식을 도입했으며, 이러한 기능은 표준 협업 필터링 모델에 통합되어 추천 성능을 향상시킨다. 이 접근법은 사용자 항목 상호 작용 정보가 충분하지 않은 환경에서 특히 유용한 것으로 보인다.

3. RECOMMENDATIONS WITH RNNs

순환 신경망은 가변 길이 시퀀스 데이터를 모델링하기 위해 고안되었다. RNN과 기존 피드포워드 심층 모델 사이의 주요 차이점은 네트워크를 구성하는 장치에 내부 은닉 상태가 존재한다는 것이다. 표준 RNN은 다음 업데이트 기능을 사용하여 은닉 상태 h 를 업데이트한다.

$$\mathbf{h}_t = g(W\mathbf{x}_t + U\mathbf{h}_{t-1})$$

여기서 g 는 로지스틱 시그모이드 함수 \mathbf{x}_t 와 같이 매끄럽고 제한된 함수이다. RNN은 현재 상태 h 가 주어졌을 때 시퀀스의 다음 요소에 대한 확률 분포를 출력한다.

게이트 순환 장치(GRU)(Cho 등, 2014)는 사라지는 그레이디언트 문제를 다루는 것을 목표로 하는 RNN 장치의 보다 정교한 모델이다. GRU 게이트는 기본적으로 장치의 숨겨진 상태를 언제 얼마나 업데이트해야 하는지 학습한다. GRU의 활성화는 이전의 활성화와 후보 활성화 \hat{h}_t 사이의 선형 보간이다.

$$\mathbf{h}_t = (1 - \mathbf{z}_t)\mathbf{h}_{t-1} + \mathbf{z}_t\hat{\mathbf{h}}_t$$

여기서 업데이트 게이트는 다음을 통해 제공된다.

$$\mathbf{z}_t = \sigma(W_z\mathbf{x}_t + U_z\mathbf{h}_{t-1})$$

후보 활성화 함수 \hat{h}_t 는 유사한 방법으로 계산된다.

$$\hat{\mathbf{h}}_t = \tanh(W\mathbf{x}_t + U(\mathbf{r}_t \odot \mathbf{h}_{t-1}))$$

그리고 마지막으로 리셋 게이트 \mathbf{r}_t 는 다음과 같이 주어진다.

$$\mathbf{r}_t = \sigma(W_r\mathbf{x}_t + U_r\mathbf{h}_{t-1})$$

3.1 CUSTOMIZING THE GRU MODEL

우리는 세션 기반 추천을 위해 모델에서 GRU 기반 RNN을 사용했다. 네트워크의 입력은 세션의 실제 상태이고 출력은 세션의 다음 이벤트 항목이다. 세션 상태는 실제 이벤트의 항목이거나 지금까지 세션의 이벤트일 수 있다. 전자의 경우 1-of-N 인코딩이 사용된다. 즉, 입력 벡터의 길이는 항목 수와 같고 활성 항목에 해당하는 좌표만 1이고 나머지는 0이다. 후자의 설정은 이벤트가 더 일찍 발생한 경우 할인되는 이러한 표현의 가중 합계를 사용한다. 안정성의 관점에서 입력 벡터는 정규화된다. RNN의 더 긴 메모리에 의해 잘 포착되지 않는 매우 로컬 순서 제약 조건의 강화라는 메모리 효과를 강화하기 때문에 이것이 도움이 될 것으로 기대한다. 우리는 또한 추가 임베딩 레이어를 추가하는 실험을 했지만 1-of-N 인코딩이 항상 더 나은 성능을 보였다.

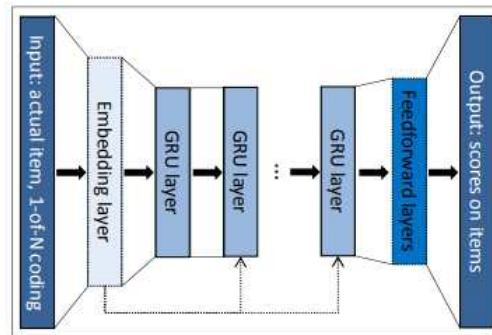


Figure 1: General architecture of the network. Processing of one event of the event stream at once.

네트워크의 핵심은 GRU 계층이며 마지막 계층과 출력 사이에 추가 피드포워드 계층을 추가할 수 있다. 출력은 항목의 예측된 선호도, 즉 각 항목의 세션에서 다음이 될 가능성이다. GRU 레이어가 여러 개 사용될 때, 이전 레이어의 숨겨진 상태는 다음 레이어의 입력이다. 입력은 또한 선택적으로 네트워크의 더 깊은 GRU 계층에 연결될 수 있는데, 이는 우리가 성능을 향상시킨다는 것을 발견했기 때문이다. 이벤트의 시계열 내에서 단일 이벤트의 표현을 묘사하는 그림 1의 전체 아키텍처를 참조하셈.

추천 시스템은 반복 신경망의 주요 적용 영역이 아니기 때문에 작업에 더 적합하도록 기본 네트워크를 수정했다. 또한 우리의 솔루션이 실제 환경에서 적용될 수 있도록 실용적인 사항도 고려했다.

3.1.1 SESSION-PARALLEL MINI-BATCHES

자연어 처리 작업을 위한 RNN은 일반적으로 순차적 미니 배치를 사용한다. 예를 들어, 문장의 단어 위에 sliding window를 사용하고 이 창문 조각들을 서로 옆에 놓아 mini-batches를 만드는 것이 일반적이다. 이것은 우리의 임무에 맞지 않는다. 왜냐하면 (1) 세션의 길이는 문장과는 매우 다를 수 있기 때문이다. 일부 세션은 2개의 이벤트로 구성되는 반면, 다른 세션은 수백 개가 넘을 수 있다. (2) 우리의 목표는 시간이 지남에 따라 세션이 어떻게 진화하는지 포착하는 것이므로 조각으로 분해하는 것은 말이 되지 않을 것이다. 따라서 세션 병렬 미니 배치를 사용한다. 먼저 세션의 순서를 만든다. 그런 다음 첫 번째 X 세션의 첫 번째 이벤

트를 사용하여 첫 번째 미니 배치의 입력을 형성한다(원하는 출력은 활성 세션의 두 번째 이벤트). 두 번째 미니 배치는 두 번째 이벤트 등으로 구성된다. 세션 중 하나라도 종료되면 사용 가능한 다음 세션이 해당 위치에 배치된다. 세션은 독립적이라고 가정하므로 이 스위치가 발생할 때 적절한 hidden state를 재설정한다. 자세한 내용은 그림 2를 참조하셈.

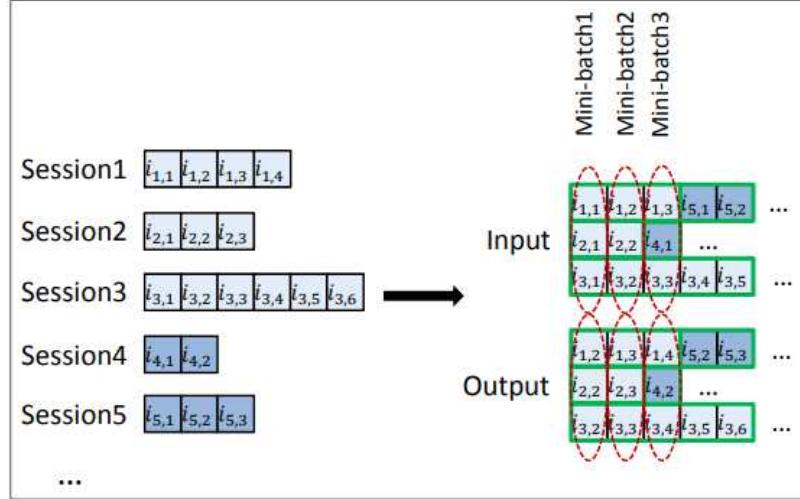


Figure 2: Session-parallel mini-batch creation

3.1.2 SAMPLING ON THE OUTPUT

추천 시스템은 항목 수가 많을 때 특히 유용하다. 중간 규모의 웹샵에서도 이것은 수 만개의 범위에 속하지만, 더 큰 사이트에서는 수십만개의 아이템을 가지고 있거나 심지어 수백만개의 아이템을 가지고 있는 경우가 드물지 않다. 각 단계에서 각 항목에 대한 점수를 계산하면 알고리즘이 항목 수와 사건 수의 곱에 따라 확장된다. 이것은 실제로 사용할 수 없을 것이다. 따라서 우리는 출력을 샘플링하고 항목의 작은 부분집합에 대한 점수만 계산해야 한다. 이것은 또한 일부 가중치만 업데이트됨을 의미한다. 원하는 출력 외에 일부 부정적인 예에 대한 점수를 계산하고 원하는 출력이 높은 순위가 되도록 가중치를 수정해야 한다.

임의의 누락 사건에 대한 자연스러운 해석은 사용자가 항목의 존재를 몰랐기 때문에 상호 작용이 없었다는 것이다. 그러나 사용자가 항목을 싫어하기 때문에 항목에 대해 알고 있고 상호 작용하지 않기로 선택했을 가능성은 낮다. 항목이 인기가 많을수록 사용자가 항목에 대해 알고 있을 가능성이 높기 때문에 누락된 이벤트가 싫어함을 나타낼 가능성이 더 높다. 따라서 우리는 인기에 비례하여 품목을 표본으로 추출해야 한다. 각 훈련 예제에 대해 별도의 샘플을 생성하는 대신 미니 배치의 다른 훈련 예제의 항목을 부정적인 예로 사용한다. 이 접근 방식의 이점은 샘플링을 건너뛰어 계산 시간을 추가로 줄일 수 있다는 것이다. 또한 코드를 덜 복잡하게 만들어 더 빠른 매트릭스 작업으로 구현하는 측면에서도 이점이 있다. 한편, 이 접근 방식은 인기도 기반 샘플링이기도 하다. 왜냐하면 항목이 미니 배치의 다른 교육 예제에 포함될 가능성은 인기도에 비례하기 때문이다.

3.1.3 RANKING LOSS

추천 시스템의 핵심은 항목의 관련성 기반 순위이다. 작업을 분류 작업으로 해석할 수도 있지

만 순위를 매기는 방식(Rendle et al., 2009; Shi et al., 2012; Steck, 2015)은 일반적으로 다른 접근 방식을 능가한다. 순위는 pointwise, pairwise 또는 listwise일 수 있다. Pointwise Ranking은 점수나 항목의 순위를 서로 독립적으로 추정하며 손실은 해당 항목의 순위가 낮도록 정의된다. pairwise 순위는 긍정적인 항목과 부정적인 항목의 쌍의 점수 또는 순위를 비교하고 손실은 긍정적인 항목의 순위가 부정적인 항목보다 낮아야 한다. Listwise 순위는 모든 항목의 점수와 순위를 사용하여 완벽한 순서와 비교한다. 정렬을 포함하므로 일반적으로 계산 비용이 더 많이 들고 자주 사용되지 않는다. 또한 우리의 경우와 같이 관련 항목이 하나만 있는 경우 페어별 순위를 통해 목록별 순위를 해결할 수 있다.

우리는 솔루션에 몇 가지 pointwise 및 pairwise 순위 손실을 포함했다. 우리는 이 네트워크에서 pointwise 순위가 불안정하다는 것을 발견했다(자세한 설명은 섹션 4 참조). 반면에 페어와이즈 순위 손실은 잘 수행되었다. 우리는 다음 두 가지를 사용한다.

- BPR : Bayesian Personalized Ranking(Rendle et al., 2009)은 페어별 순위 손실을 사용하는 행렬 분해 방법이다. 긍정적인 항목과 샘플링된 부정적인 항목의 점수를 비교한다. 여기에서 긍정적인 항목의 점수를 여러 샘플 항목과 비교하고 다음을 사용한다. The loss

at a given point in one session is defined as:
$$L_s = -\frac{1}{N_s} * \sum_{j=1}^{N_s} \log(\sigma(\hat{r}_{s,i} - \hat{r}_{s,j})),$$

where N_s is the 표본 크기, $\hat{r}_{s,k}$ is the 항목에 대한 점수 k 는 세션의 주어진 지점, i is 원하는 항목(세션의 다음 항목) and j are the 부정적인 샘플.

- TOP1 : 이 순위 손실은 우리가 이 과제를 위해 고안한 것이다. 관련 항목의 상대적 순위에 대한 정규화된 근사치이다. 관련 항목의 상대적 순위는 다음과 같다.

$$\frac{1}{N_s} * \sum_{j=1}^{N_s} I\{\hat{r}_{s,j} > \hat{r}_{s,i}\}.$$
 우리는 $I\{\cdot\}$ 에 Sigmoid를 근사한다. 이를 최적화하면 i 의 점수가 높

도록 매개변수가 수정된다. 그러나 특정 긍정적인 항목도 부정적인 예로 작용하여 점수가 점점 더 높아지는 경향이 있기 때문에 이는 불안정하다. 이를 피하기 위해, 우리는 부정적인 예제의 점수가 약 0점이 되도록 하고 싶다. 이것은 부정적인 항목의 점수에 대한 자연스러운 기대이다. 그래서 우리는 손실에 정규화 항을 추가했다. 이 용어가 상대적인 순위와 같은 범위에 있고 이와 유사하게 작동하는 것이 중요하다. 최종 손실 함수는 다음과 같

다:
$$L_s = \frac{1}{N_s} * \sum_{j=1}^{N_s} \sigma(\hat{r}_{s,j} - \hat{r}_{s,i}) + \sigma(\hat{r}_{s,j}^2)$$

4. EXPERIMENTS

제안된 재귀 신경망을 두 데이터 세트의 인기 있는 베이스라인에 대해 평가한다.

첫 번째 데이터 세트는 RecSys Challenge 2015의 데이터 세트이다. 이 데이터 세트에는 구매 이벤트로 끝나는 e커머스 사이트의 클릭 스트림이 포함되어 있다. 우리는 챌린지의 훈련 세트로 작업하고 클릭 이벤트만 유지한다. 길이가 1인 세션을 필터링한다. 네트워크는 37,483 개 항목에 대한 31,637,239회의 클릭에 대한 7,966,257개의 세션을 포함하는 최대 6개월의 데이터에 대해 훈련된다. 테스트를 위해 다음 날의 세션을 사용한다. 각 세션은 훈련 또는 테

스트 세트에 할당되며 세션 중간에 데이터를 분할하지 않는다. 협업 필터링 방법의 특성 때문에 클릭한 항목이 트레인 세트에 없는 테스트 세트에서 클릭을 필터링한다. 길이가 1인 세션도 테스트 세트에서 제거된다. 전처리 후 테스트 세트에 대한 71,222개 이벤트의 15,324개 세션이 남는다. 이 데이터 세트를 **RSC15**라고 한다.

두 번째 데이터 세트는 Youtube와 같은 OTT 비디오 서비스 플랫폼에서 수집된다. 일정 시간 이상 동영상상을 시청한 이벤트를 수집했다. 특정 지역만 2개월 미만의 짧은 기간 동안 이 수집의 대상이 되었다. 이 시간 동안 화면 왼쪽에서 각 비디오 후에 항목 간 권장 사항이 제공되었다. 이들은 다양한 알고리즘을 선택하여 제공되었으며 사용자의 행동에 영향을 미쳤다. 전처리 단계는 봇에 의해 생성되었을 가능성이 있는 매우 긴 세션을 필터링하는 기능이 추가된 다른 데이터 세트의 단계와 유사하다. 교육 데이터는 앞서 언급한 기간의 마지막 날을 제외한 모든 날로 구성되며 33만 개의 동영상에 대해 최대 300만 세션의 1300만 개의 시청 이벤트가 있다. 테스트 세트에는 수집 기간의 마지막 날의 세션이 포함되어 있으며 최대 37,000개의 세션과 최대 180,000개의 시청 이벤트가 있다. 이 데이터 세트를 **VIDEO**라고 한다.

평가는 세션의 이벤트를 하나씩 제공하고 다음 이벤트의 항목 순위를 확인하는 방식으로 이루어진다. 세션이 완료되면 GRU의 hidden state가 0으로 재설정된다. 항목은 점수에 따라 내림차순으로 정렬되며 이 목록에서 해당 위치는 순위이다. RSC 15에서는 트레인의 37,483개 품목이 모두 순위에 올랐다. 그러나, 이것은 많은 수의 아이템 때문에 VIDEO에서는 비실용적이었다. 거기서 우리는 가장 인기 있는 30,000개의 품목에 대해 원하는 품목의 순위를 매겼다. 이는 거의 방문하지 않은 항목들이 낮은 점수를 받는 경우가 많기 때문에 평가에 거의 영향을 미치지 않는다. 또한 인기 기반 사전 필터링은 실제 추천 시스템에서 일반적이다.

추천 시스템은 한 번에 몇 개의 항목만 추천할 수 있으므로 사용자가 선택할 수 있는 실제 항목은 목록의 처음 몇 개의 항목 중 하나가 되어야 한다. 따라서 우리의 1차 평가 지표는 모든 테스트 사례에서 상위 20개 항목 중 원하는 항목을 가진 경우의 비율인 $recall@20$ 이다. 리콜은 상위 N개 항목에 포함된 항목의 실제 순위를 고려하지 않는다. 이는 추천을 강조하지 않고 절대 순서가 중요하지 않은 특정 실제 시나리오를 잘 모델링한다. 또한 리콜은 클릭률(CTR)과 같은 중요한 온라인 KPI와도 잘 상관된다(Liu et al., 2012; Hidasi & Tikk, 2012). 실험에 사용된 두 번째 메트릭은 $MRR@20$ (평균 역수 순위)이다. 그것은 원하는 항목의 역수 순위의 평균이다. 순위가 20 이상이면 역순위는 0으로 설정된다. MRR은 항목의 순위를 고려하며, 이는 추천 순서가 중요한 경우(예: 하위 순위는 스크롤한 후에만 표시됨)에 중요하다.

4.1 BASELINES

제안된 네트워크를 일반적으로 사용되는 베이스라인 세트와 비교한다.

- **POP** : 교육 세트의 가장 인기 있는 항목을 항상 추천하는 인기 예측 변수. 단순함에도 불구하고 특정 도메인에서 강력한 기준선이 되는 경우가 많다.
- **S-POP** : 이 베이스라인은 현재 세션의 가장 인기 있는 항목을 추천한다. 항목이 더 많은 이벤트를 얻으면 세션 동안 추천 목록이 변경된다. 타이는 글로벌 인기도 값을 사용하여 분할된다. 이 기준선은 반복성이 높은 영역에서 강력하다.
- **Item-KNN** : 실제 항목과 유사한 항목은 이 베이스라인에 의해 권장되며 유사성은 세션 벡터 간의 코사인 유사성으로 정의된다. 즉, 세션에서 두 항목의 공존 횟수를 개별 항목이

발생한 세션 수의 곱의 제곱근으로 나눈 값이다. 자주 방문하지 않는 품목의 높은 유사성을 동시에 피하기 위해 정규화도 포함된다. 이 기준선은 실제 시스템에서 가장 일반적인 항목 간 솔루션 중 하나로, "이 항목을 본 다른 사람들도 이러한 항목을 보았다" 설정에 대한 권장 사항을 제공한다. 단순함에도 불구하고 일반적으로 강력한 기준선이다(Linden 등, 2003, Davidson 등, 2010).

- **BPR-MF** : BPR-MF(Rendle et al., 2009)는 일반적으로 사용되는 행렬 분해 방법 중 하나이다. SGD를 통해 페어별 순위 목적 함수(섹션 3 참조)를 최적화한다. 새 세션에는 미리 계산된 기능 벡터가 없기 때문에 행렬 분해를 세션 기반 권장 사항에 직접 적용할 수 없다. 그러나 지금까지 세션에서 발생한 항목의 항목 특성 벡터의 평균을 사용자 특성 벡터로 사용하여 이를 극복할 수 있다. 즉, 추천 항목과 지금까지 세션 항목 간의 특징 벡터 유사도를 평균화한다.

Table 1: Recall@20 and MRR@20 using the baseline methods

Baseline	RSC15		VIDEO	
	Recall@20	MRR@20	Recall@20	MRR@20
POP	0.0050	0.0012	0.0499	0.0117
S-POP	0.2672	0.1775	0.1301	0.0863
Item-KNN	0.5065	0.2048	0.5508	0.3381
BPR-MF	0.2574	0.0618	0.0692	0.0374

Table 2: Best parametrizations for datasets/loss functions

Dataset	Loss	Mini-batch	Dropout	Learning rate	Momentum
RSC15	TOP1	50	0.5	0.01	0
RSC15	BPR	50	0.2	0.05	0.2
RSC15	Cross-entropy	500	0	0.01	0
VIDEO	TOP1	50	0.4	0.05	0
VIDEO	BPR	50	0.3	0.1	0
VIDEO	Cross-entropy	200	0.1	0.05	0.3

표 1은 베이스라인의 결과를 보여준다. Item-KNN 접근법이 다른 방법을 확실히 지배한다.

4.2 PARAMETER & STRUCTURE OPTIMIZATION

각 데이터 세트와 손실 함수에 대해 매개 변수 공간의 무작위로 선택된 지점에서 100개의 실험을 실행하여 하이퍼 파라미터를 최적화했다. 각 파라미터를 개별적으로 최적화하여 최적의 파라미터화를 추가로 조정했다. hidden units의 수는 모든 경우에 100으로 설정되었다. 그런 다음 크기가 다른 hidden layers와 함께 가장 성능이 좋은 파라미터를 사용했다. 최적화는 별도의 validation 검사 세트에서 수행되었다. 그런 다음 네트워크는 훈련과 검증 세트에서 재교육되고 최종 테스트 세트에서 평가되었다.

가장 잘 수행되는 파라미터화는 표 2에 요약되어 있다. 가중치 행렬은 $[-x, x]$ 에서 균일하게 추출된 난수로 초기화되었으며, 여기서 x 는 행렬의 행과 열 수에 따라 다르다. 우리는 rmsprop(Dauphin et al., 2015)과 adagrad(Duchi et al., 2011)를 모두 실험했다. 우리는 adagrad가 더 나은 결과를 제공한다는 것을 발견했다.

우리는 GRU 이외의 다른 단위로 간략하게 실험했다. 우리는 고전적인 RNN 유닛과 LSTM이 더 나쁜 성능을 보인다는 것을 발견했다.

우리는 몇 가지 손실 함수를 시도했다. 교차 엔트로피 및 MRR 최적화(Steck., 2015)와 같은

포인트별 순위 기반 손실은 정규화에도 불구하고 대개 불안정했다. 예를 들어 교차 엔트로피는 RSC15와 VIDEO에 대한 100개의 무작위 실행 중 각각 10개와 6개의 수치적으로 안정적인 네트워크를 생성했다. 우리는 이것이 원하는 항목에 대해 독립적으로 높은 점수를 얻으려고 하기 때문이며 부정적인 샘플에 대해서는 부정적인 푸시(negative push)가 작기 때문이라고 가정한다. 반면 페어별 순위 기반 손실은 좋은 성과를 거두었다. 섹션 3(BPR 및 TOP1)에서 소개한 것이 가장 우수한 성능을 발휘하는 것으로 나타났다.

여러 아키텍처가 조사되었으며 GRU 장치의 단일 계층이 가장 성능이 좋은 것으로 밝혀졌다. 레이어를 추가하면 항상 훈련 손실과 리콜 및 테스트 세트에서 측정된 MRR에서 최악의 성능이 초래된다. 이는 일반적으로 세션의 수명이 짧기 때문에 서로 다른 해상도의 여러 시간 척도가 적절하게 표현되지 않아도 되기 때문이라고 가정한다. 그러나 이것의 정확한 이유는 아직 알려지지 않았으며 더 많은 연구가 필요하다. 항목 임베딩을 사용하면 약간 더 나쁜 결과가 나타나므로 1-of-N 인코딩을 유지했다. 또한 세션의 모든 이전 이벤트를 이전 이벤트 대신 입력에 배치해도 추가적인 정확도가 향상되지 않았다. 이는 LSTM과 같은 GRU가 장단기 메모리를 모두 가지고 있기 때문에 놀라운 일이 아니다. GRU 계층 뒤에 피드포워드 계층을 추가하는 것도 도움이 되지 않았다. 그러나 GRU 계층의 크기를 늘리면 성능이 향상되었다. 또한 출력층의 활성화 함수로 tanh를 사용하는 것이 유익하다는 것을 발견했다.

Table 3: Recall@20 and MRR@20 for different types of a single layer of GRU, compared to the best baseline (item-KNN). Best results per dataset are highlighted.

Loss / #Units	RSC15		VIDEO	
	Recall@20	MRR@20	Recall@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%)
Cross-entropy 100	0.6074 (+19.91%)	0.2430 (+18.65%)	0.6372 (+15.69%)	0.3720 (+10.04%)
TOP1 1000	0.6206 (+22.53%)	0.2693 (+31.49%)	0.6624 (+20.27%)	0.3891 (+15.08%)
BPR 1000	0.6322 (+24.82%)	0.2467 (+20.47%)	0.6311 (+14.58%)	0.3136 (-7.23%)
Cross-entropy 1000	0.5777 (+14.06%)	0.2153 (+5.16%)	—	—

4.3 RESULTS

표 3은 가장 성능이 좋은 네트워크의 결과를 보여준다. 1000개의 hidden units가 있는 VIDEO 데이터에 대한 교차 엔트로피는 수치적으로 불안정했기 때문에 우리는 그 시나리오에 대한 결과를 제시하지 않는다. 결과는 최상의 베이스라인(Item-KNN)과 비교된다. 우리는 100과 1000개의 hidden units로 결과를 보여준다. 실행 시간은 매개 변수와 데이터 세트에 따라 다르다. 일반적으로 GeForce GTX Titan X GPU에서 더 작은 변형과 더 큰 변형 사이의 런타임 차이는 그리 높지 않으며 네트워크의 훈련은 몇 시간 내에 수행될 수 있다. CPU에서 더 작은 네트워크는 실질적으로 허용되는 시간 내에 훈련될 수 있다. 새로운 사용자와 항목이 자주 소개되기 때문에 자주 재교육이 추천 시스템에 바람직하다.

GRU 기반 접근 방식은 단위 수가 100개라도 두 데이터 세트의 두 평가 지표 모두에서 항목 KNN에 비해 상당한 이득을 가진다. 단위 수를 증가시키면 페어별 손실의 결과는 더욱 향상되지만 교차 엔트로피의 경우 정확도는 감소한다. 교차 엔트로피는 100개의 숨겨진 유닛에서 더 나은 결과를 제공하지만, 페어별 손실 변형은 유닛 수가 증가함에 따라 이러한 결과를 능가한다. 유닛 수를 늘리면 훈련 시간이 늘어나지만, GPU에서 100 유닛에서 1000 유닛으로 이동하는 것이 그리 비싸지 않다는 것을 알게 되었다. 또한 크로스 엔트로피 기반 손실은 네트워크

크가 개별적으로 대상 항목에 대해 점수를 높이려고 시도한 결과 수치적으로 불안정한 반면, 다른 항목에 대해서는 부정적인 푸시가 상대적으로 작은 것으로 나타났다. 따라서 두 쌍의 손실 중 하나를 사용할 것을 추천한다. TOP1 손실은 이 두 데이터 세트에서 약간 더 잘 수행되므로 최상의 성능 베이스라인에 비해 정확도가 최대 20-30% 향상된다.

5 CONCLUSION & FUTURE WORK

본 논문에서는 새로운 애플리케이션 도메인인 추천 시스템에 일종의 현대 반복 신경망(GRU)을 적용하였다. 세션 기반 추천 작업은 실질적으로 중요한 영역이지만 잘 연구되지 않았기 때문에 선택했다. 세션 병렬 미니 배치, 미니 배치 기반 출력 샘플링 및 순위 손실 함수를 도입하여 작업에 더 잘 맞도록 기본 GRU를 수정했다. 우리는 우리의 방법이 이 작업에 사용되는 인기 있는 베이스라인을 크게 능가할 수 있다는 것을 보여주었다. 우리는 우리의 연구가 추천 시스템과 일반적으로 세션 기반 추천 모두에서 딥 러닝 응용 프로그램의 기초가 될 수 있다고 생각한다.

우리의 즉각적인 향후 작업은 제안된 네트워크의 보다 철저한 검사에 초점을 맞출 것이다. 또한 현재 입력된 내용 대신 항목 자체의 내용(예: 썸네일, 비디오, 텍스트)을 기반으로 자동으로 추출된 항목 표현에 대해 네트워크를 훈련시킬 계획이다.

ACKNOWLEDGMENTS

이러한 결과로 이어지는 작업은 CrowdRec Grant Agreement n^o 610594에 따라 유럽 연합의 7차 프레임워크 프로그램(FP7/2007-2013)으로부터 자금 지원을 받았셈.

REFERENCES

Cho, Kyunghyun, van Merriënboer, Bart, Bahdanau, Dzmitry, and Bengio, Yoshua. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014.

Dauphin, Yann N, de Vries, Harm, Chung, Junyoung, and Bengio, Yoshua. Rmsprop and equilibrated adaptive learning rates for non-convex optimization. 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. Adaptive subgradient methods for online learning and stochastic optimization. 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, Balazs and Tikk, Domonkos. General factorization framework for context-aware recommendations. Data Mining and Knowledge Discovery, pp. 1-30, 2015. 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, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. Signal Processing Magazine, IEEE, 29(6):82-97, 2012.

Koren, Y. Factorization meets the neighborhood: a multifaceted 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. Matrix factorization techniques for recommender systems. Computer, 42(8):30-37, 2009.

Linden, G., Smith, B., and York, J. Amazon. com recommendations: Item-to-item collaborative filtering. Internet Computing, IEEE, 7(1):76-80, 2003.

Liu, Qiwen, Chen, Tianjian, Cai, Jing, and Yu, Dianhai. Enlister: Baidu's recommender system for the biggest Chinese Q&A website. 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: Bayesian personalized ranking from implicit feedback. 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 large scale visual recognition challenge. CoRR, abs/1409.0575, 2014.
URL <http://arxiv.org/abs/1409.0575>.

Salakhutdinov, Ruslan, Mnih, Andriy, and Hinton, Geoffrey. Restricted boltzmann machines for collaborative filtering. In Proceedings of the 24th international conference on Machine learning, pp. 791-798. ACM, 2007.

Sarwar, Badrul, Karypis, George, Konstan, Joseph, and Riedl, John. Item-based

collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web, pp. 285-295. ACM, 2001.

Shani, Guy, Brafman, Ronen I, and Heckerman, David. An mdp-based recommender system. In Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence, pp. 453-460. Morgan Kaufmann Publishers Inc., 2002.

Shi, Yue, Karatzoglou, Alexandros, Baltrunas, Linas, Larson, Martha, Oliver, Nuria, and Hanjalic, Alan. Climf: Learning to maximize reciprocal rank with collaborative less-is-more filtering. 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. Gaussian ranking by matrix factorization. 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. Deep content-based music recommendation. In Advances in Neural Information Processing Systems, pp. 2643-2651, 2013.

Wang, Hao, Wang, Naiyan, and Yeung, Dit-Yan. Collaborative deep learning for recommender systems. 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. Maximum margin matrix factorization for collaborative ranking. Advances in neural information processing systems, 2007.