딥러닝 논문 요약 및 구현 스터디

발표자

조홍철, 이현준

Deep Neural Networks for youtube recommendations

0.Abstract

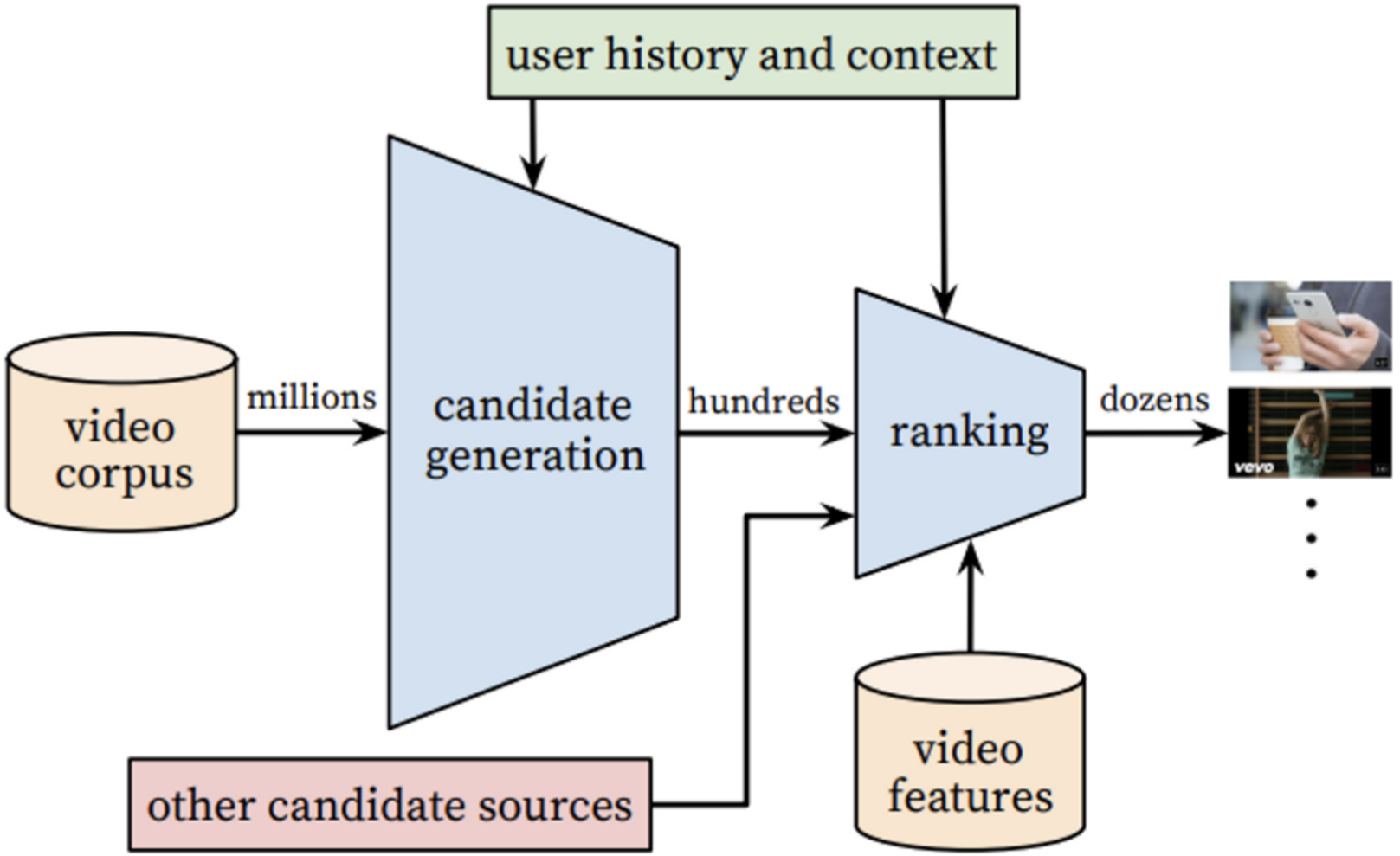
- deep learning을 통해 엄청난 성능 개선
- deep candidate generation model과 deep ranking model에 대해 설명
- 대규모 추천 시스템 구축 및 유지를 경험하며 생긴 실용적인 교훈과 통찰력을 제공

1.Introduction

- YouTube 추천은 10억 명 이상의 사용자가 계속 증가하는 동영상 컨텐츠에 대해 맞춤 컨텐츠를 추천하기 위한 미션이 있다
- YouTube 추천은 다음 세 가지 측면에서 극도로 첼린징하다
  - Scale
    - 작은 문제에 효과적으로 작동하는 많은 기존 추천 알고리즘들이 YouTube 규모에서 동작하지 않는다
    - YouTube의 막대한 사용자 규모와 동영상을 처리하기 위해서는 전문화된 분산 학습 알고리즘과 효율적인 서빙 시스템이 반드시 필요하다
  - Freshness
    - Youtube에는 매순간 많은 시간 분량의 동영상이 업로드 되며, 새로운 영상 콘텐츠에 따른 사용자의 반응도 실시간으로 발생한다
    - 추천 시스템에서는 이러한 실시간 반응 요소들을 적절하게 적용 및 반영할 수 있는 방법을 찾아야한다
    - 새로운 콘텐츠와 잘 알려진 동영상들을 균형 있게 다루는 것은 탐색(Exploration)과 활용(Exploitation) 관점에서 이해할 수 있다

    - Noise
        - 희소성과 다양한 관측 불가능한 외부 요인들 때문에 예측하기가 기본적으로 어렵다
            - 희소성 : 영상에 대해 직접적으로 평가하는 경우가 드물기 때문에
        - 노이즈가 섞인 암시적 피드백 신호를 모델링
            - 동영상을 시청했다는 implict feedback을 주로 사용
        - 콘텐츠와 관련된 메타데이터는 잘 정의된 체계가 없어 구조화가 잘 되지 않는다
- Matrix Factorization에 비해 딥러닝을 이용한 추천에 대한 연구가 상대적으로 적었다
- 논문의 구성
    - Section 2: 시스템에 대한 간단한 개요
    - Section 3: candidate generation model에 대한 구체적인 설명
    - Section 4: ranking model에 대한 구체적인 설명
    - Section 5: 결론 및 시사점

2. System Overview



- 크게 candidate generation과 ranking으로 나뉜다
- candidate generation
  - user history and context를 input으로 한다
  - 유저가 관심있을 법한 수백 개의 동영양 컨텐츠를 output으로 한다
  - 협업 필터링을 사용하며, 사용자들 간의 유사성은 동영상 시청 기록의 ID, 검색어 토큰 및 인구 통계 정보와 같은 고수준 특징들을 기반으로한다
- ranking
  - 비디오와 사용자를 설명하는 다양한 feature들을 사용하여 각 비디오에 점수를 할당
  - 가장 높은 점수를 받은 비디오들이 사용자에게 제시된다
- 개발 과정에서는 offline metrci(precision, recall, ranking loss 등)을 활용하여 시스템을 반복적으로 개선
- 알고리즘 또는 모델의 효과를 최종적으로 결정하기 위해서는 라이브 실험을 통한 A/B 테스트에 의존한다
- 라이브 실험에서는 CTR, 시청 시간 및 사용자 참여를 측정하는 여러 다른 메트릭들의 변화를 측정할 수 있다
- 이는 라이브 A/B 결과가 오프라인 실험과 항상 상관관계가 있는 것은 아니기 때문에 중요

3. Candidate Generation

- 엄청난 양의 컨텐츠 중에서 유저와 관련성이 있는 수백 개의 컨텐츠로 범위를 좁힘
- 이전에는 rank loss를 기반으로 matrix factorization이 사용
- Shallow Network를 통해 Factorization 진행
- Factorization 기술을 non-linear으로 일반화했다고 볼 수 있음

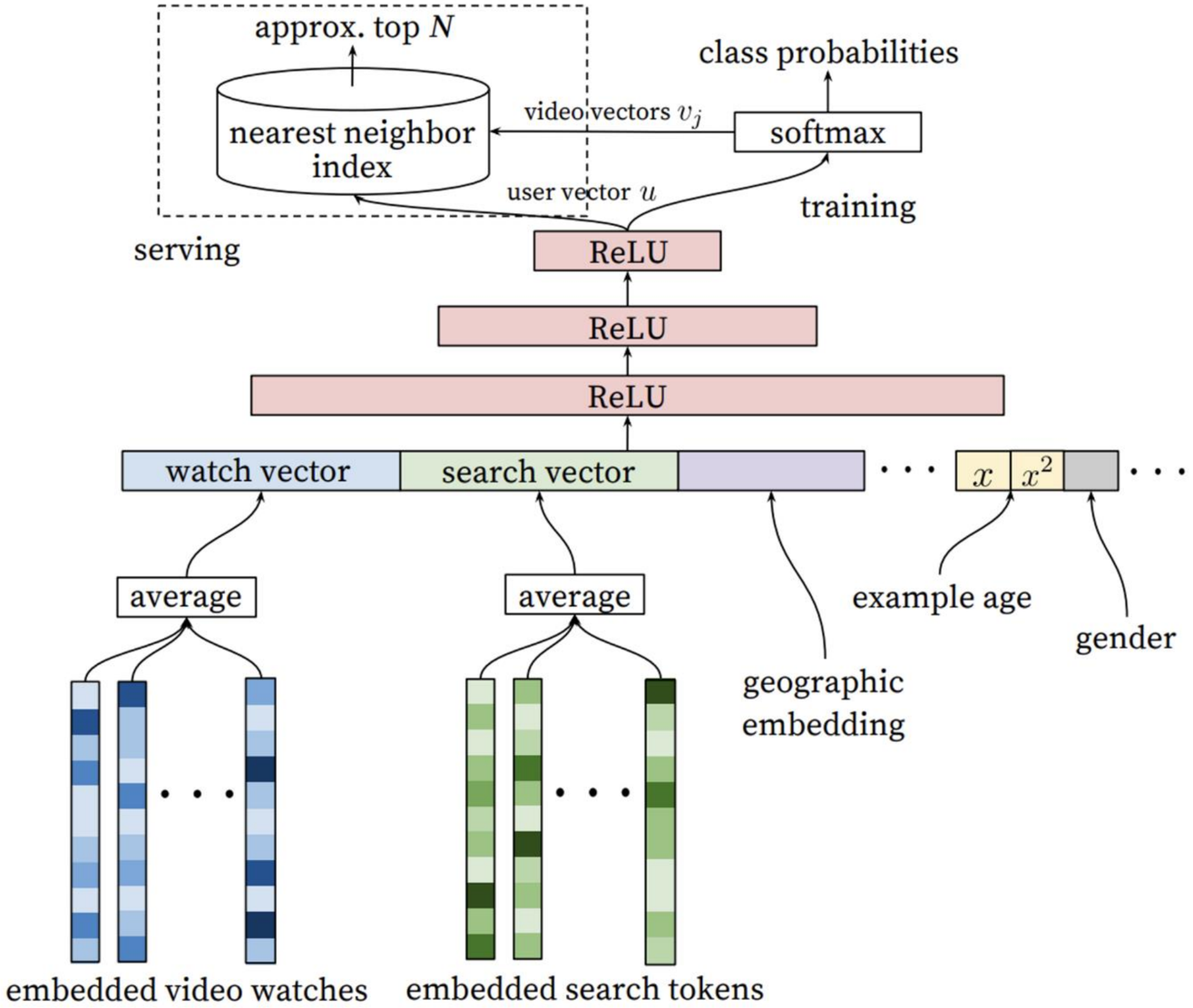
3-1. Recommendation as Classification

$$P(w_t = i|U, C) = \frac{e^{v_i u}}{\sum_{j \in V} e^{v_j u}}$$

- 사용자(U)와 Context(C)를 기반으로 특정 시간(t)에서 수백만개의 아이템(V) 중 각 아이템(i)의 시청 wt를 예측
  - u는 user-context pair의 고차원 임베딩
  - vj 는 각각의 후보 동영상의 임베딩
  - 임베딩 벡터들은 단순히 희소 entities 를 매핑하여 얻어진 dense 한 벡터
  - deep neural networks는 사용자의 히스토리와 맥락을 기반으로 사용자 임베딩 u를 학습
    - 사용자 임베딩을 활용하여 소프트맥스 분류기를 통해 다양한 동영상들을 분류하는데 사용됩니다.

- explict feedback(thumbs up/down)이 존재하지만, 사용자가 시청 완료한 영상 콘텐츠를 positive로 분류하여 implicit feedback을 학습함
    - implict feedback이 explicit feedback보다 훨씬 더 많이 존재
    - implict feedback을 활용하면 explicit feedback이 부족한 tail부분의 정보를 보완하여 더 다양하고 정확한 추천을 생성할 수 있게 된다
- Efficient Extreme Multiclass
    - Softmax classification 에서 클래스의 갯수가 늘어날때 계산량이 기하급수적으로 증가
    - negative sampling을 통해 기존 softmax 보다 약 100배 가량 속도를 향상했다

3-2. Model Architecture



- sparse한 ID들로 이루어진 가변 길이의 특징들을 처리하기 위해 임베딩(embedding)을 사용
  - 사용자의 시청 기록은 비디오 ID들로 구성된 가변 길이의 시퀀스로 표현되며, 이러한 비디오 ID들은 임베딩을 통해 밀집 벡터 표현으로 매핑
- 임베딩들을 평균화하여 고정된 길이의 벡터로 변환한 뒤 concatenate. 이렇게 함으로써, 모델은 가변 크기의 입력을 고정된 크기의 벡터로 변환하여 히든 레이어에 입력으로 사용할 수 있다
  - 여러 전략중에서 임베딩의 평균을 사용하는 것이 가장 좋은 결과를 냈다
- Hidden layers: Deep candidate generation model은 fully connected된 여러 개의 hidden 레이어를 가진다.
- 훈련 단계에서는 cross-entropy loss가 사용되며, 샘플링된 softmax(output of the sampled softmax)에 대한 gradient descent를 이용하여 loss를 최소화한다
- Serving: 실제 서비스에서의 운영 시, approximate nearest neighbor lookup을 수행하여 수백 개의 후보 동영상 추천을 생성한다