❤

[1주차] Deep Neural Networks for YouTube Recommendations

0. Abstract

- YouTube의 추천 시스템에 대해 설명하며, Deep Learning에 의해 이루어진 성능 향상에 초점을 맞춤
    - deep candidate generation model
    - deep ranking model
    - 대규모 추천 시스템을 설계, 반복, 유지관리하면서 얻은 실질적인 교훈과 통찰을 제공

1. Introduction

유튜브의 추천 시스템에는 다음 세 가지 주요 도전 과제가 주어짐

- 스케일
    - 작은 문제에서는 잘 작동하더라도 유튜브와 같이 규모가 큰 문제에서는 잘 작동하지 않을 수 있음
    - 유튜브의 방대한 사용자와 콘텐츠를 처리하기 위해서는 전문화된 분산 학습 알고리즘과 효율적인 서비스 시스템이 필요함
- 신선도
    - 유튜브에는 매초마다 수많은 비디오들이 업로드되는 동적인 콘텐츠이므로, 추천 시스템은 새로 업로드된 콘텐츠뿐만 아니라 이에 따른 사용자의 행동을 바로 모델링할 수 있을 만큼 반응이 빨라야함
    - 새로 올라오는 콘텐츠와 이미 자리잡은 비디오들 사이의 균형을 탐색 및 활용 관점에서 이해할 수 있음
- 노이즈
    - 사용자의 사용 히스토리는 관찰되지 않은 외부 요인들로 인해 예측하기 어려움

- 사용자 만족도의 직접적인 원인을 거의 얻지 못하는 대신, 시청 시간이나 클릭 등을 통해 간접적으로 추론해야 함
- 따라서 추천 시스템 알고리즘은 사용자의 행동이나 선호도를 예측하는데 사용되는 데이터가 가진 독특한 특징과 노이즈들을 잘 처리하고 이해할 수 있어야 함

유튜브는 딥러닝을 활용하여 큰 변화를 겪음

- TensorFlow 플랫폼을 사용하여 대규모 데이터로 다양한 심층 신경망을 실험하고 학습시킴
- Youtube는 TensorFlow 기술을 사용하여 약 10억 개의 매개변수를 학습하고, 수백억 개의 예시로 훈련시켜 추천 시스템을 개선함

논문 구성

- 시스템 개요 소개
- candidate generation model(후보 생성 모델)
- 클릭 확률이 아닌 예상 시청 시간을 예측하는 랭킹 모델
- 실험 결과 및 은닉층의 깊이가 모델 성능 향상에 기여하는 방법
- 결론 및 교훈

2. System Overview

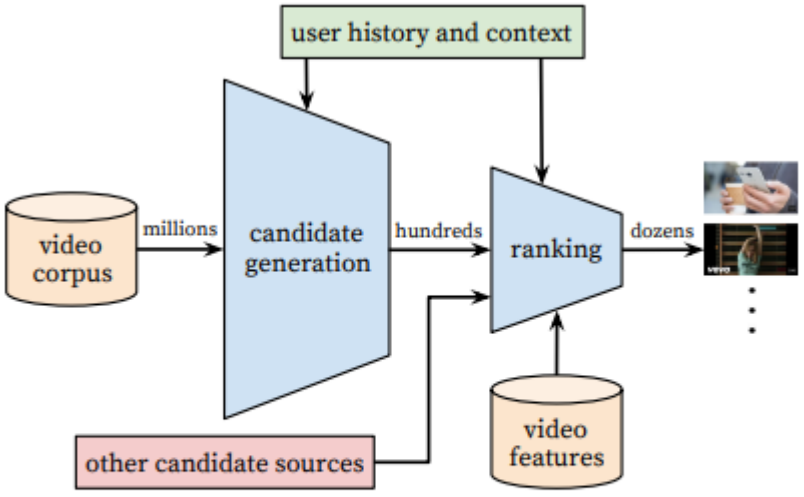


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.

유튜브 추천 시스템은 크게 두 가지 주요 신경망으로 구성됨

- 후보 생성 네트워크
  - 사용자의 유튜브 활동 이력을 바탕으로 대량의 비디오 중에서 소수의 비디오(수백 개)를 선별해 내는 역할
  - 사용자들 간의 유사성을 고려하여 개인화된 추천을 제공, 이때 유사성은 사용자들의 시청 기록, 검색 쿼리, 인구통계학적 정보 등을 통해 파악됨
- 순위 매기기 네트워크
  - 선별된 후보 비디오들 사이에서 어떤 비디오를 사용자에게 어떤 순서로 보여줄지 결정
  - 각 비디오에 대한 사용자의 관심도를 예측하기 위해 비디오와 사용자의 다양한 특성을 분석하여 점수를 매기고, 이 점수가 높은 순으로 사용자에게 비디오를 추천함

이러한 두 단계 접근법을 통해 유튜브는 수백만 개의 비디오 중에서도 사용자에게 개인화되고 흥미로운 비디오를 효과적으로 추천할 수 있으며, 다양한 출처에서 온 추천 후보들을 통합할 수 있는 능력을 가지고 있음

시스템의 성능을 향상시키기 위해 개발자들은 정밀도와 재현율 같은 여러 메트릭을 사용하여 지속적으로 시스템을 개선함. 그리고 최종적으로는 라이브 실험을 통한 A/B 테스팅을 진행하여 클릭률, 시청 시간 등의 실제 사용자 반응을 측정함으로써 알고리즘의 효과를 검증함

➡️**오프라인에서의 실험 결과가 항상 실제 사용자 경험과 일치하지 않을 수 있기 때문에 검증 과정은 매우 중요**

3. Candidate Generation

- 후보 생성
  - 유튜브의 방대한 콘텐츠 중에서 사용자에게 관련될 수 있는 수백 개의 비디오를 선별해내는 과정
  - 이 추천 시스템이 등장하기 전에는, 사용자의 아이템에 대한 선호도 순위를 예측하는 것을 목표로 가지고 행렬 분해 방식으로 구현된 시스템이 사용되었음
  - 신경망 모델의 초기 버전은 사용자의 이전 시청 목록만을 임베딩하는 얕은 네트워크로, 이 행렬 분해 행위를 모방
  - 위 접근 방식은 행렬 분해 기법의 비선형 일반화로 볼 수 있으며, 이로써 기존 방식보다 더 복잡하고 정교한 방식으로 사용자의 선호도를 예측할 수 있게 됨

3.1 Recommendation as Classfication

- 특정 비디오를 사용자에게 추천하는 과정을 매우 많은 선택지(비디오) 중에서 하나를 고르는 문제로 다룸
- 사용자와 그 상황을 고려하여, 각 비디오가 사용자에게 얼마나 적합한지를 계산함. 이때 사용자와 상황, 그리고 비디오 정보는 모두 고차원의 벡터(임베딩)로 변환되어 계산에 사용됨
    - 사용자 U와 상황 C를 기반으로, 많은 수의 비디오 i(클래스)가 포함된 코퍼스 V(비디오 전체 집합)에서 특정 시간 t에서의 비디오 시청 wt를 정확하게 분류하는 것

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

    - (P(w_t = i|U, C))는 사용자 (U)와 상황 (C)가 주어졌을 때, 특정 시간 (t)에 비디오 (i)를 시청할 확률을 나타냄
    - (u)는 사용자와 상황의 정보를 합친 것을 고차원 공간에서 나타낸 벡터, 즉 '임베딩'을 의미. 이 임베딩은 사용자와 상황의 다양한 특성들을 수치화하여 표현한 것
    - (v_j)는 각 비디오의 특성을 나타내는 임베딩이며, 이 역시 고차원 벡터로 표현됨
    - (\frac{e^{v_{iu}}}{\sum_{j \in V}e^{v_{ju}}})는 사용자와 상황에 대응하는 임베딩 (u)와 각 비디오 임베딩 (v_j) 간의 관계를 이용해, 특정 비디오 (i)가 선택될 확률을 계산하는 부분
        - 분자: 사용자와 해당 비디오와의 적합도
        - 분모: 모든 가능한 비디오들에 대한 적합도의 합
    - 사용자와 그 상황을 대표하는 벡터(u)와 각 비디오를 대표하는 벡터(v_j)를 기반으로 특정 비디오가 사용자에게 추천될 확률(P)을 계산하며, 이 확률은 모든 가능한 비디오에 대한 계산값을 기반으로 결정됨
- 사용자가 비디오를 끝까지 시청한 것과 같은 암시적인 반응을 사용하여 학습함
    - 좋아요 혹은 싫어요와 같은 직접적인 피드백에 비해 훨씬 많은 데이터를 제공하기에 추천 시스템의 정확도를 높이는 데 도움이 됨

Efficient Extreme Multiclass

큰 규모의 분류 문제를 효율적으로 해결하기 위한 기술적 접근 방법

- 음성 클래스 샘플링

    - '후보 샘플링(candidate sampling)'이라고 불리며, 배경 분포에서 음성 클래스(즉, 대상 레이블이 아닌 클래스)를 샘플링하여 모델 훈련 과정에서 사용함
    - 이후 중요도 가중치(importance weighting)를 통해 이 샘플링을 보정하며, 실제 레이블과 샘플링된 음성 클래스에 대해 교차 엔트로피 손실을 최소화함으로써 모델을 훈련함
        - 샘플링으로 인해 과대표되거나 과소표된 클래스의 샘플들이 실제 모델 학습에 미치는 영향을 조정
        - 실제 레이블과 모델이 예측한 레이블 간의 차이를 나타내는 교차 엔트로피 손실을 최소화하는 방향으로 학습이 진행되는 것
        - 중요도 가중치는 각 샘플이 손실에 기여하는 정도를 조정하여, 모든 클래스가 공평하게 학습에 기여할 수 있도록 도움
- 효율성
    - 실제로 수천 개의 음성 클래스를 샘플링하여 사용하는데, 이는 전통적인 소프트맥스 방법에 비해 100배 이상 속도를 향상시킬 수 있음
- 대안적 접근 방법 - 계층적 소프트맥스
    - 트리를 통해 각 노드를 탐색하면서 클래스 집합 간의 구분을 시도하지만, 종종 관련 없는 클래스 집합을 구별해야 하기 때문에 분류 문제가 더 어려워지고 성능이 저하될 수 있음
- 제공 시 고려사항
    - 사용자에게 제시할 상위 N개의 클래스(비디오)를 계산할 때, 수백만 개의 아이템을 몇십 밀리초의 엄격한 서빙 지연 시간 내에 점수를 매기기 위해서는 클래스의 수에 비례하지 않는(sublinear, 비디오의 총 수가 늘어남에 따라 계산 시간이 비디오 수에 정비례해서 늘어나지 않아야 함) 근사 점수 계산 방식이 필요
        - 근사 점수 계산 방식: 완벽하게 정확한 점수를 계산하는 대신, 실제 점수에 매우 가까운 근사값을 훨씬 빠른 시간 내에 계산
        - 알고리즘의 복잡성을 크게 줄여주며, 엄격한 시간 제약 내에서도 사용자에게 상위 N개의 가장 관련성 높은 비디오를 추천할 수 있도록 함
    - YouTube의 이전 시스템은 해싱 기법에 의존했었고, 여기서 설명하는 분류기도 비슷한 접근 방식을 사용
    - 서빙 시간에는 소프트맥스 출력층에서의 보정된 가능성(calibrated likelihoods)이 필요하지 않기 때문에, 점수 문제는 점곱 공간에서의 최근접 이웃 탐색 문제로 간소화됨

        - 각 클래스(비디오)에 속할 확률값을 정확히 계산할 필요 X, 비디오의 상대적 순위를 얻으면 됨(어떤 비디오가 다른 비디오보다 더 관련성이 높은지)
        - 사용자와 가장 관련성이 높은(가장 가까운) 몇몇 비디오를 찾으면 되기 때문에 KNN 문제로 간주
    - 이에 대해 일반적인 목적의 라이브러리를 사용할 수 있으며, A/B 테스트 결과는 선택된 최근접 이웃 탐색 알고리즘에 특별히 민감하지 않았다는 것을 발견함
        - 어떤 특정 알고리즘을 선택하더라도 결과에 크게 영향 X

3.2 Model Architecture

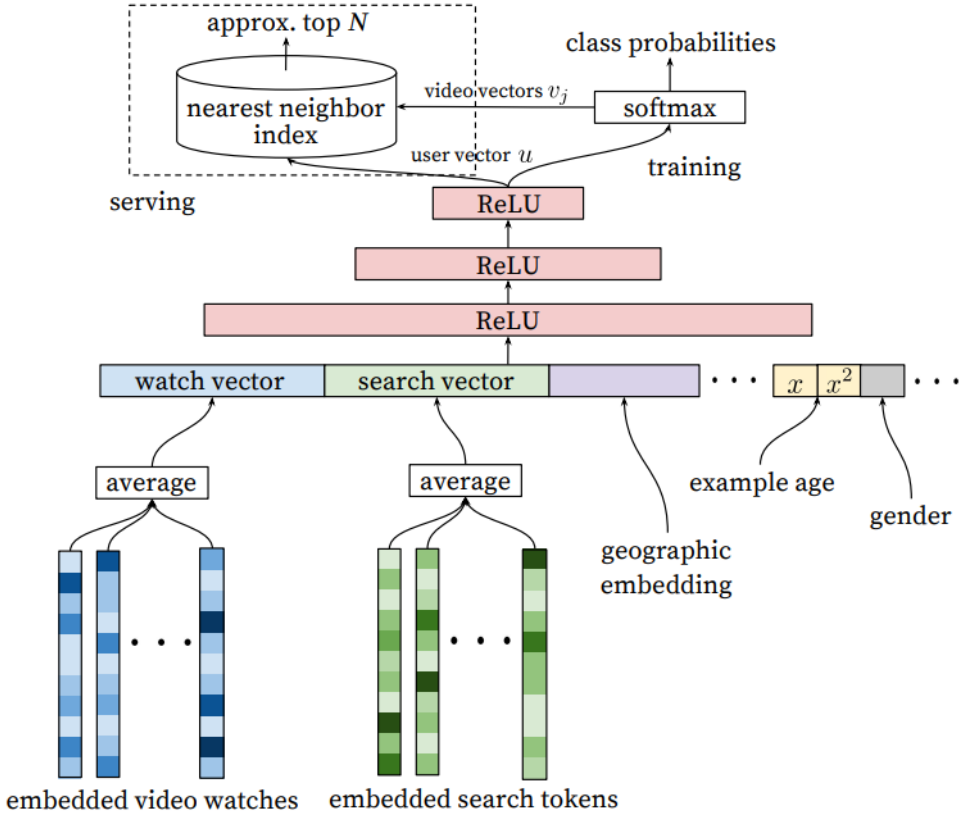


Figure 3: Deep candidate generation model architecture showing embedded sparse features concatenated with dense features. Embeddings are averaged before concatenation to transform variable sized bags of sparse IDs into fixed-width vectors suitable for input to the hidden layers. All hidden layers are fully connected. In training, a cross-entropy loss is minimized with gradient descent on the output of the sampled softmax. At serving, an approximate nearest neighbor lookup is performed to generate hundreds of candidate video recommendations.

- CBOW(Continuous Bag of Words)
    - 자연어 처리에서 사용되는 모델로, 주변 단어들을 통해 특정 단어를 예측하는 방법
    - 비디오 추천 시스템에 CBOW 모델의 아이디어를 적용

- 고차원 임베딩 학습
  - 각 비디오에 대해 고차원의 임베딩(비디오를 대표하는 벡터)을 학습, 고정된 어휘 사전에 맞춰서 생성함(일정한 규칙이나 범위 내에서 생성되도록 조정)
  - 생성된 비디오 임베딩은 전방향 피드 포워드 신경망(입력된 벡터를 바탕으로 비디오가 사용자에게 얼마나 관련이 있는지를 판단)에 입력됨
- 사용자의 시청 이력 표현
  - 비디오 ID의 가변 길이 시퀀스로 나타내며, 이 시퀀스는 임베딩을 통해 밀집된 벡터 표현으로 매핑됨
- 임베딩 평균화
  - 신경망은 고정된 크기의 밀집 입력(많은 정보를 담고 있는 고정된 크기의 벡터)을 필요로 함
  - 여러 전략(합, 요소별 최대값 등) 중에서 임베딩을 단순히 평균내는 것이 가장 좋은 성능을 보였음
- 임베딩과 모델 파라미터의 공동 학습
  - 임베딩들은 모델의 다른 모든 파라미터들과 함께 일반적인 경사 하강법 역전파 업데이트를 통해 함께 학습됨
- feature들의 연결과 ReLU층
  - feature들은 넓은 첫 번째 레이어로 연결되고, 그 다음에는 여러 층의 완전 연결된 선형 유닛(ReLU)이 이어짐
  - 추가적으로 비디오 외적인 시청 feature들도 모델에 포함됨

3.3 Heterogeneous Signals

- 심층 신경망을 이용하는 것의 큰 장점은 다양한 유형의 정보를 쉽게 모델에 추가할 수 있다는 점
  - 사용자의 검색 기록이나 시청 기록 같은 데이터는 작은 단위로 쪼개져서 모델에 포함됨
  - 이렇게 처리된 정보는 사용자의 검색 습관을 잘 나타내는 요약된 데이터로 변환됨
- 나이, 성별, 지리적 위치, 사용하는 기기 등의 인구 통계학적 특성
  - 새로운 사용자에게도 적절한 추천을 제공하기 위해 사용됨
  - 모델에 직접 입력되어 사용자 맞춤형 추천을 가능하게 함

“Example Age” Feature

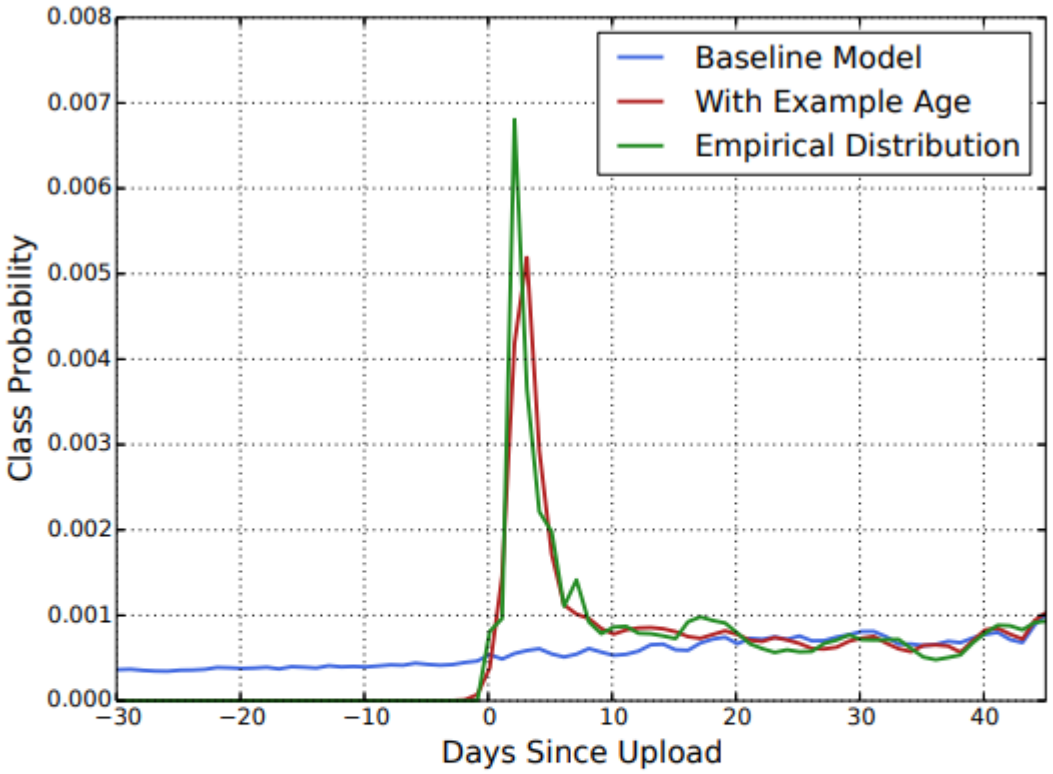
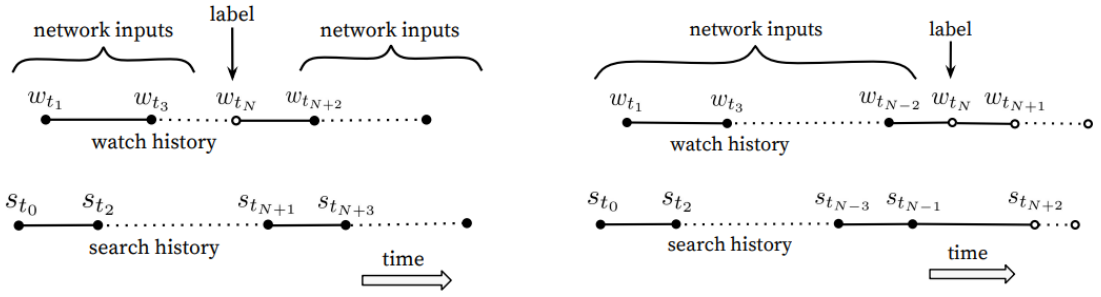


Figure 4: For a given video [26], the model trained with example age as a feature is able to accurately represent the upload time and time-dependant popularity observed in the data. Without the feature, the model would predict approximately the average likelihood over the training window.

- 유튜브에서는 새로운 동영상을 추천함으로써 사용자의 관심을 끌 뿐만 아니라, 인기 있는 콘텐츠가 더 널리 퍼지도록 돕는 것도 중요함
- 기계 학습 시스템은 과거 데이터를 바탕으로 미래를 예측하려고 하기 때문에 과거에 집중하는 경향이 있음
    - 동영상의 인기가 시간에 따라 계속 변한다는 점을 고려하지 못할 수 있음(역주행 등)
    - 이를 해결하기 위해 유튜브는 동영상이 얼마나 최신인지를 모델이 학습할 수 있도록 정보를 제공함
    - 실제로 동영상을 추천할 때는 이 정보를 최신 상태로 조정하여, 새로운 콘텐츠에 대한 추천이 더 잘 이루어지도록 함

3.4 Label and Context Selection



(a) Predicting held-out watch (b) Predicting future watch

Figure 5: Choosing labels and input context to the model is challenging to evaluate offline but has a large impact on live performance. Here, solid events • are input features to the network while hollow events ○ are excluded. We found predicting a future watch (5b) performed better in A/B testing. In (5b), the example age is expressed as $t_{\max} - t_N$ where $t_{\max}$ is the maximum observed time in the training data.

- 사용자에게 맞춤형 콘텐츠를 제공할 때, 기계 학습을 사용하여 사용자의 취향과 관심사에 가장 잘 맞는 콘텐츠를 찾아내는 것이 핵심
  - 단순히 인기 콘텐츠를 추천하는 것보다 훨씬 복잡함
  - 사용자의 과거 시청 기록, 검색 기록, 상호작용 데이터 등을 분석하여 개인화된 추천을 제공해야 하며, 사용자가 아직 발견하지 못한 새로운 콘텐츠를 제안해야 함
- 모든 유튜브 시청 기록을 사용하여 추천 시스템을 훈련시킴
  - 추천 시스템은 사용자의 다음 시스템을 예측하기 위해 개인의 취향과 관심사를 반영해야 하며, 이를 위해 복잡한 데이터 분석과 처리가 필요함
  - 사용자가 추천 외의 다른 방법으로 동영상을 발견하는 경우, 그 발견을 협업 필터링을 통해 빠르게 다른 사용자에게 전파함
  - 사용자 당 훈련 예제의 고정된 수를 생성하여, 손실 함수에서 모든 사용자를 동등하게 가중하는 것이 실시간 지표를 개선하는 데 중요한 역할을 함 >> 활동이 매우 활발한 소수의 사용자가 손실을 지배하는 것을 방지
- 사용자의 프라이버시 보호와 데이터 보안 또한 중요한 고려사항
  - 과적합 방지
    - 다음으로 볼 영상을 추천할 때, 단순히 검색어가 나타난 페이지의 내용만 복사해서 추천하는 것 X(단지 검색 결과 페이지의 구조를 학습하는 것 X)
    - 검색어 안에 담긴 여러 단어들을 개별적으로 인식하도록 하여, 사용자가 정말로 관심 있어 할 만한 내용을 추천해주는 방법을 사용함
  - 사용자의 데이터를 안전하게 처리하고, 사용자의 동의 하에 데이터를 활용해야 함
  - 추천 알고리즘의 투명성과 공정성을 확보하는 것도 중요한 과제

- 사용자에게 왜 특정 콘텐츠가 추천되었는지 설명을 제공하고, 편향 없는 추천을 위해 알고리즘을 지속적으로 개선해야 함

3.5 Experiments with Features and Depth

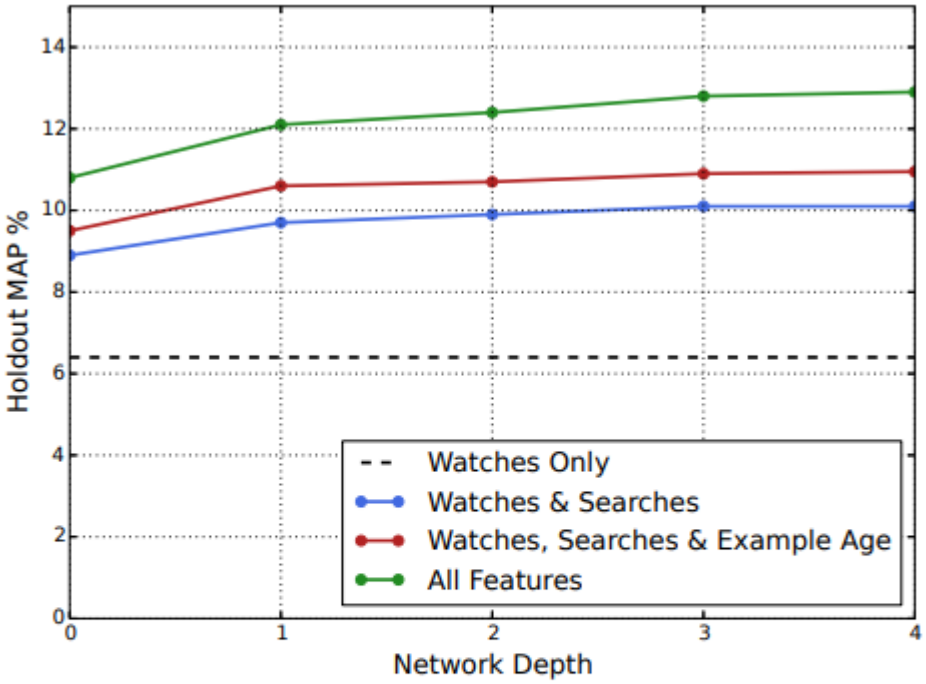


Figure 6: Features beyond video embeddings improve holdout Mean Average Precision (MAP) and layers of depth add expressiveness so that the model can effectively use these additional features by modeling their interaction.

- 모델 구성
  - 1백만 개의 동영상과 1백만 개의 검색 토큰이 각각 256개의 부동 소수점 숫자로 임베딩됨
  - 최대 50개의 최근 시청 기록과 50개의 최근 검색 기록을 포함하는 가방(bag) 크기를 기반으로 함
  - 소프트맥스층은 동일한 1백만 개의 동영상 클래스에 대한 다항 분포를 256 차원으로 출력함(별도의 출력 동영상 임베딩)
- epoch

- 모든 YouTube 사용자에 대해 데이터에 대한 여러 에폭(epoch) 동안 수렴될 때까지 훈련됨
- 네트워크 구조는 네트워크 하단이 가장 넓고 각 후속 은닉층이 단위 수를 절반으로 줄이는 일반적인 "타워" 패턴을 따름
  - Depth 0: 기본적으로 선형 분해 방식으로, 이전 시스템과 매우 유사하게 수행됨
  - Depth 1: 단순히 연결된 층을 256 차원의 소프트맥스와 일치하도록 변환하는 선형 층이 있음
  - Depth 2: 256개의 ReLU 활성화 함수를 갖는 층이 추가됨
  - Depth 3: 512개의 ReLU로 이어지는 1024 ReLU와 256 ReLU가 추가됨
  - Depth 4: 2048 ReLU, 1024 ReLU, 512 ReLU, 마지막으로 256 ReLU가 차례로 추가됨
- 네트워크의 깊이와 너비가 증가함에 따라 모델의 성능이 향상되었지만, 이익이 감소하고 수렴이 어려워질 때까지 추가됨(>>차원의 저주, 과적합 발생)
- 더 많은 특징과 깊이를 추가하면 보류 데이터(holdout data)에서의 정밀도가 크게 향상됨을 보여줌
  - 보류 데이터: 모델이 완전히 훈련되고 나서, 마지막으로 모델의 성능을 평가하기 위해 사용되는 데이터. 훈련 과정에서 전혀 사용되지 않으며, 모델이 얼마나 잘 일반화되었는지를 평가하는 데 사용됨

4. Ranking

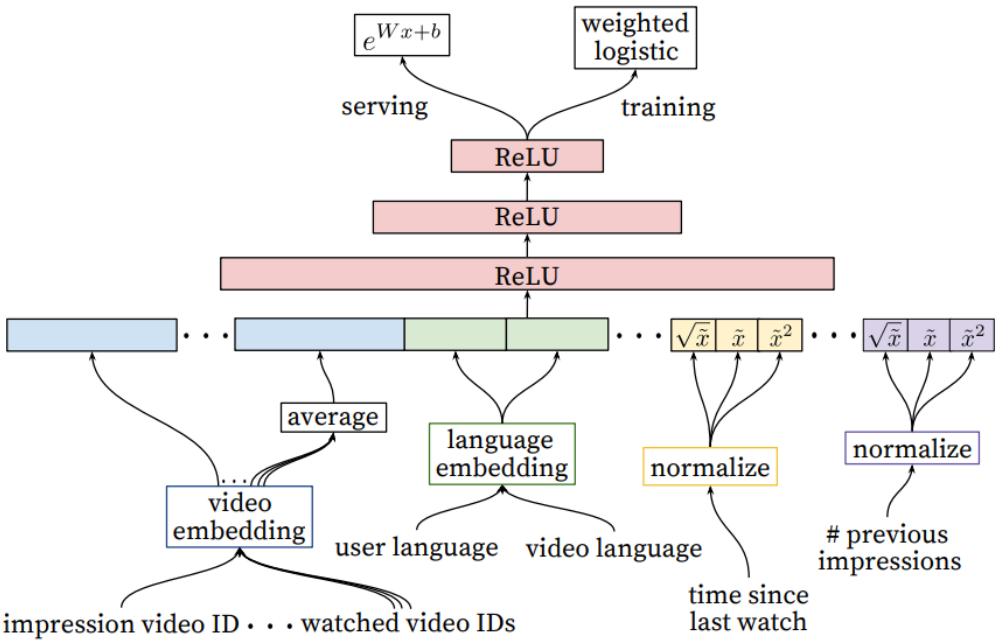


Figure 7: Deep ranking network architecture depicting embedded categorical features (both univalent and multivalent) with shared embeddings and powers of normalized continuous features. All layers are fully connected. In practice, hundreds of features are fed into the network.

- 랭킹 시스템의 핵심 목적은 사용자에게 맞춤화된 비디오 추천을 제공하는 것
  - 후보 생성 단계보다 훨씬 적은 수의 비디오를 대상으로 함
  - 각 비디오에 대한 독립적인 점수를 매기고, 이를 기반으로 비디오들을 정렬해 사용자에게 제시함
  - 로지스틱 회귀를 사용하는 깊은 신경망을 통해 구현되며, 최종적으로 사용자에게 제공되는 비디오 목록은 실시간 A/B 테스팅을 통해 지속적으로 조정됨
  - 단순히 클릭을 유도하는 클릭베이트 비디오보다는, 사용자가 실제로 관심을 가지고 시청할 내용을 우선시하는 방식

4.1 Feature Representaton

- 범주형 feature, 연속형 feature, order형 feature로 나뉨
  - 카디널리티(가능한 값의 범위)가 매우 다양
  - 일부는 이진값을 가짐(예: 사용자가 로그인 되어 있는지 여부)
  - 또는 수백만 개의 가능한 값을 가질 수 있음(예: 사용자의 마지막 검색 쿼리)
- 단일값만을 제공하는지, 아니면 값의 집합을 제공하는지에 따라 나뉠 수 있음
  - 단일값 범주형 특징의 예로는 평가되고 있는 인상의 비디오 ID가 있음
  - 다중값 특징의 예로는 사용자가 시청한 마지막 N개의 비디오 ID들의 집합이 있음

- 아이템의 속성("인상")과 사용자/맥락의 속성("쿼리")에 따라 분류
    - 쿼리 특징은 요청당 한 번 계산되는 반면, 인상 특징은 평가되는 각 아이템에 대해 계산됨

Feature Engineering

- Ranking 모델은 수백 가지의 다양한 feature들을 사용함
    - 범주형
    - 연속형
- 이론적으로 딥러닝은 이러한 특징들을 자동으로 처리할 수 있는 잠재력을 가지고 있음에도 불구하고, 실제로는 원시 데이터를 그대로 딥러닝 모델에 넣기가 어렵다는 문제가 있음
  ➡️
  사용자와 비디오 데이터를 분석하고 유용한 특징으로 변환하는 작업을 해야 함
    - 사용자의 행동과 그 행동이 어떻게 비디오 평가와 관련되는지를 시간적 순서에 따라 정확하게 나타내는 것이 중요한 과제
- 사용자가 특정 비디오 혹은 그와 유사한 비디오와 어떻게 상호작용했는지에 대한 정보가 매우 중요함
    - 사용자가 평가되고 있는 비디오를 업로드한 채널에서 과거에 시청한 비디오의 수
    - 사용자가 이 주제에 관한 비디오를 마지막으로 본 시간
    - 비디오 추천을 받았지만 시청하지 않은 경우 >> 추천 목록에서 덜 중요하게 다루게 됨

Embedding Categorical Features

- 범주형 데이터를 신경망이 잘 이해하고 처리할 수 있도록 하기 위해 이 데이터를 신경망이 처리하기 쉬운 형태로 바꿔주어야 함 >> **임베딩**
    - 임베딩: 범주형 데이터를 신경망이 이해할 수 있는 밀집된 벡터 형태로 변환하는 것
        - 비디오 ID라는 범주형 데이터가 있다고 하면 각 비디오 ID는 임베딩을 통해 일정한 크기의 벡터로 변환됨
        - 만약 32차원의 벡터를 사용한다면 각 비디오 ID는 32개의 숫자로 이루어진 벡터로 표현됨
    - 매우 많은 범주형 값(수백만 개의 비디오 ID)을 처리할 때, 모든 값을 직접 다루는 것은 비효율적일 수 있음
        - 가장 자주 나타나는 값을 선택하고, 나머지는 "기타" 범주로 취급하여 처리

- 같은 종류의 데이터(예를 들어, 비디오 ID)에 대해서는 같은 임베딩을 공유
  - 신경망이 데이터를 더 잘 이해하고, 학습 속도를 빠르게 하며, 필요한 메모리 양을 줄일 수 있음
  - 이로 인해 신경망은 각각의 범주형 데이터를 특화된 방식으로 잘 처리할 수 있게 됨
- 임베딩 공간은 모델의 매개변수 대부분을 차지함
  - 만약 100만 개의 비디오 ID를 32차원 벡터로 임베딩한다면, 이는 신경망의 다른 부분보다 훨씬 더 많은 매개변수를 필요로 함

Normalizing Continuous Features

- 신경망은 입력 데이터의 크기나 어떻게 분포되어 있는지에 매우 민감함
- 여러 개의 결정 트리를 함께 사용하는 방법인 앙상블은 각각의 데이터 특성이 얼마나 커지든 상관없이 잘 작동함
- 데이터를 신경망에 넣기 전에, 특히 연속적으로 변하는 데이터(온도나 길이와 같은 것들)를 잘 조정하는 것이 중요
  - 정규화를 통해 스케일을 0~1 사이로 조정
  - 데이터의 제곱, 제곱근도 함께 사용

4.2 Modeling Expected Watch Time

- 목표: 훈련 예시가 긍정적인 경우(비디오 인상이 클릭됨) 또는 부정적인 경우(인상이 클릭되지 않음)인지에 따라 예상 시청 시간을 예측하는 것
  - 긍정적인 예시는 사용자가 비디오를 시청한 시간의 양으로 주석이 달림
  - 예상 시청 시간을 예측하기 위해 가중 로지스틱 회귀 기법을 사용
  - 교차 엔트로피 손실 하에서 로지스틱 회귀로 훈련됨
    - 긍정적인(클릭된) 인상의 비디오에서는 관찰된 시청 시간으로 가중치가 부여됨
    - 부정적인(클릭되지 않은) 인상은 모두 단위 가중치를 받음
  - 로지스틱 회귀에 의해 학습된 확률은 훈련 예시의 수인 N, 긍정적인 인상의 수인 k, 그리고 i번째 인상의 시청 시간인 Ti일 때 P = **Ti / (N-k)**가 됨
    - 긍정적인 인상의 비율이 작다고 가정하면(우리 경우에 해당), 학습된 확률은 대략적으로 E[T](1 + P)
    - P는 클릭 확률이고 E[T]는 동영상의 예상 시청 시간임. P가 작으므로 이 곱은 E[T]에 가까워짐

- 최종 활성화 함수로 지수 함수를 사용하여 확률을 생성 >> 예상 시청 시간을 밀접하게 추정

4.3 Experiments with Hidden Layers

Hidden layers	weighted, per-user loss
None	41.6%
256 ReLU	36.9%
512 ReLU	36.7%
1024 ReLU	35.8%
512 ReLU → 256 ReLU	35.2%
1024 ReLU → 512 ReLU	34.7%
1024 ReLU → 512 ReLU → 256 ReLU	34.6%

Table 1: Effects of wider and deeper hidden ReLU layers on watch time-weighted pairwise loss computed on next-day holdout data.

- 위 Table1에는 다양한 은닉 계층 구성을 사용하여 다음 날 보류 데이터에서 얻은 결과를 보여줌
  - 각 구성에 대해 표시된 값("가중치, 사용자별 손실")은 한 페이지에서 사용자에게 표시된 긍정적(클릭된) 및 부정적(클릭되지 않은) 인상을 모두 고려하여 얻은 것
  - 먼저 이 두 인상(긍정, 부정)을 모델로 점수를 매기며, 부정적 인상이 긍정적 인상보다 더 높은 점수를 받으면 긍정적 인상의 시청 시간을 잘못 예측된 시청 시간으로 간주
  - 가중치, 사용자별 손실은 보류된 인상 쌍에 대한 전체 시청 시간의 비율로서 잘못 예측된 시청 시간의 총량임
- 실험 결과, 은닉 계층의 너비를 늘리거나 깊이를 늘림으로써 예측의 정확도를 높일 수 있다는 것을 발견
  - 특히 1024-너비의 ReLU 함수, 그 다음으로 512-너비, 그리고 256-너비 ReLU 함수를 사용하는 구성이 최고의 성능을 보임
  - 하지만 이는 더 많은 서버 CPU 시간을 요구한다는 단점이 존재
  - 정규화된 연속형 변수만을 입력으로 사용했을 때 손실이 약간 증가했고, 긍정적 및 부정적 예시에 같은 가중치를 두어 훈련했을 때는 시청 시간에 따른 손실이 크게 증가했음

5. Conclusions

- 유튜브에서 비디오를 추천하기 위해 사용하는 DNN 알고리즘

    - 후보 생성 단계: 수많은 비디오 중에서 사용자가 관심 있어 할 만한 비디오를 후보로 선별
    - 순위 결정 단계: 후보로 선정된 비디오들 사이에서 어떤 비디오를 사용자에게 먼저 보여줄지 결정
- deep collaborative filtering model
    - 사용자의 과거 시청 기록, 비디오의 인기도 같은 다양한 정보를 분석하여, 사용자가 좋아할 만한 비디오를 찾아냄
    - 기존에 사용되던 방식보다 훨씬 정교하게 사용자의 취향을 파악할 수 있음
- 비디오가 업로드된 시간
    - 최근에 올라온 비디오일수록 사용자에게 보여주는 것이 더 좋은 반응을 얻었다는 걸 발견
    - 모델이 오래된 비디오에 대한 편향 없이 현재 인기 있는 트렌드를 반영할 수 있게 해줌
- 순위 결정 단계
    - 비디오가 사용자에게 얼마나 오래 시청될지를 예측하는 문제에 초점을 맞춤
    - 로지스틱 회귀라는 방법 사용하여 각 비디오가 얼마나 시청될지 예측, 사용자가 실제로 클릭할 확률만을 고려하는 것보다 더 정확한 결과를 보여줌
    - 단순히 클릭할 비디오를 추천하는 것이 아니라 사용자가 실제로 관심 있어 하고 시청할 만한 비디오를 추천하고자 함

논문에 대한 의견 및 의문점(꼭지)

➡️DNN에게는 '블랙 박스'라는 별명이 붙어있다. DNN은 입력(사용자의 시청 기록 등)에서 복잡한 패턴을 학습하여 출력(추천 콘텐츠 등)을 생성하는 데 매우 효과적인 기계 학습 모델이며 이 과정은 여러 층을 통해 비선형 변환을 반복적으로 적용하는 방식으로 이루어진다. 그 결과 DNN은 매우 복잡한 함수를 근사할 수 있으며, 이는 높은 예측 성능으로 이어진다. 그러나 이 복잡성이 바로 DNN을 '블랙 박스'로 만드는 원인으로, DNN 내부에서 일어나는 수많은 계산 과정과 파라미터의 상호작용은 인간이 쉽게 이해하거나 해석하기 어렵기 때문에 붙여진 별명이다. 특히 깊은 층을 가진 네트워크에서는 어떤 특정 입력이 최종 출력에 어떻게 영향을 미치는지를 정확히 파악하기가 매우 어렵다. 유튜브의 경우 개발자 뿐만이 아니라 전세계 남녀노소 할 것 없이 무수히 많은 사람들이 사용하고 있는 플랫폼인데, 유튜브의 추천 시스템은 이러한 많은 사용자들의 개인적인 데이터들을 바탕으로 훈련을 한다. 하지만 모델의 내부가 어떻게 작동하는지 제대로 알지도 못하는 상황에서 개개인의 데이터를 가져다 모델 훈련에 사용하는 것은 개인 정보 보호 윤리에 위배될 여지가 다분히 있다고 본다. 따

라서 '블랙 박스'적인 특징을 최소화한 DNN 모델을 사용하여 신뢰도를 높이거나, 주기적으로 개인정보가 어떻게 훈련에 활용되고 있는지 사용자들에게 안내를 해야 할 필요가 있다고 생각한다.