**실무에 바로 쓰는 AI예측/추천 4주차 수업노트**

22.5.19

4주차 배울 내용

**추천 시스템**

* 추천시스템 이해
* Collative Filtering, Content-Based Filtering
* 하이브리드 모형
* 딥러닝의 개입

학습 내용

02. 추천시스템에 대한 이해

03. Collative Filtering 모형

04. CF 실습

05. Latent Factor Model(MF)

06. Latent Factor Model(MF) 실습

07. Deep Learning for RecSys

08. Neural Collaborative Filtering(NCF)

09. You Tube 딥러닝 추천시스템 이해하기

10. Wide & Deep 모델 이해하기

**02. 추천시스템에 대한 이해**

**추천시스템**



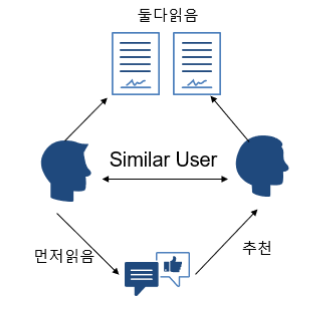
**추천 시스템의 종류(추천유형)**

* 사용자 수동 큐레이션 ⇒ Favorite폴더 등
* 단순 통계형 ⇒ Top 100, 최신음악, 요즘유행하는
* **개인화 ⇒ 개인에 맞춤형 아이템을 추천**
  + 모든 개인이 다른 추천 결과물을 갖게 됨

**추천 시스템의 종류(콘텐츠)**

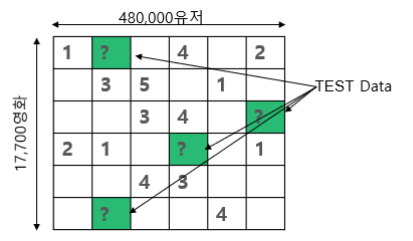
* 영화/음악 추천 ⇒ Spotify, netflix, Youtube
* 상품추천 ⇒ 쿠팡, 아마존, Ably
* 뉴스추천 ⇒네이버 뉴스 추천, 페이스북 뉴스 피드
* 다른 다양한 추천들 ⇒ 앱 추천, 해시태그 추천, 레스토랑 추천(구글맵), 소셜추천(링크드인)

**03. Collative Filtering 모형**

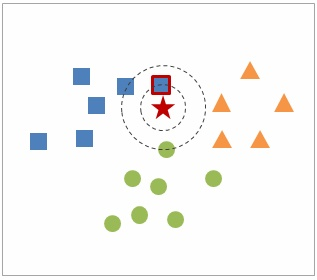


* user based Filtering
* Item Based Filtering

**넷플릭스 Prize**



**KNN? (k-nearest neighbors)**



**장점**

* 영화, 드라마, 도서 등등 추천 대상에 제한이 없다 (평가 정보만 있으면 됨)
* 간단하고 연산이 빠름

**단점**

* Cold Start: 충분한 데이터 확보가 전제
* Sparsity: 평가 데이터(User-Item Matrix)에 빈 곳이 많다
* First rater: 한 번도 평가 되지 않은 드라마는 절대 추천되지 않는다
* Popularity Bias: 독특한 취향의 User 추천이 어렵거나 인기 있는 드라마가 추천

**04. CF 실습**

**샘플데이터**

ML-100K: <https://grouplens.org/datasets/movielens/>

**ml-latest-small.zip**

movies.csv, ratings.csv

**1. dataset 불러오기**

**2. rating 을 movie feature처럼 만들기**

**3. 유사도 구하기**

* **knn사용하기**
* 결과 조회

**4. Normalize 하기**

**5. 각 스코어별로 보기**

**6. 50개이상 평가된 movie만 필터하기**

**7. 각 유저별로 Rating이 몇개씩 있는지 확인해보기**

**8. rating이 50개 이상이고 유효한 유저들로만 필터걸기**

**9. Movie-User Matrix만들기**

**10. Knn모델만들기 (User-Item 필터링 이후에 다시 Knn모델 적용)**

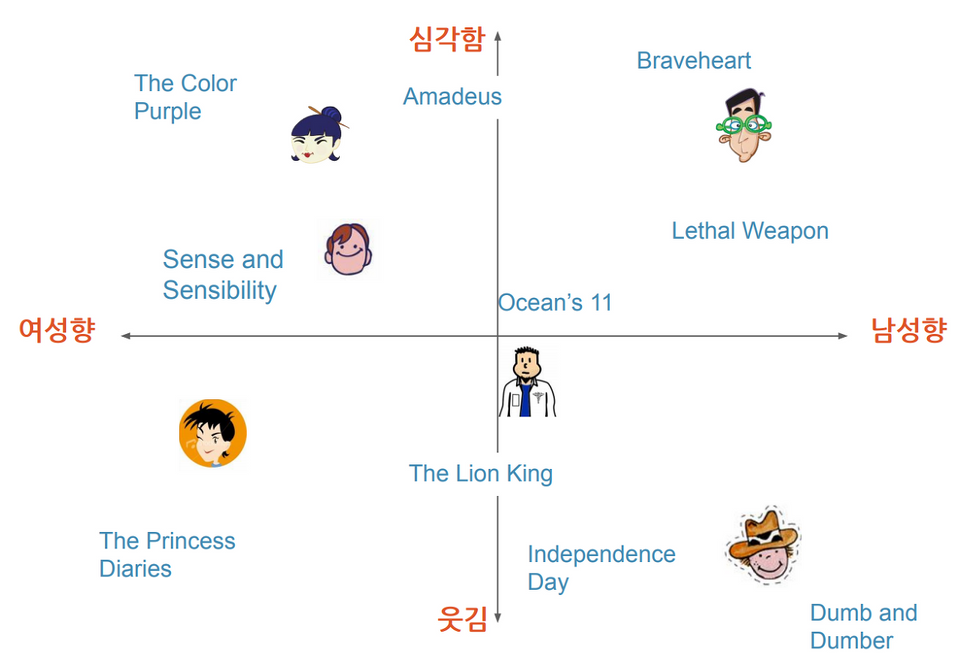
**11. 시청항목과 추천 항목 출력 함수 정의**

**12. 영화제목 입력 후 직접예측해보기**

| **You have input movie: Iron Man**  **Found possible matches in our database: ['Iron Man (2008)']**  **Recommendation system start to make inference**  **......**  **Recommendations for Iron Man:**  **1: Batman Begins (2005), with distance of 0.3474416136741638**  **2: Sherlock Holmes (2009), with distance of 0.34635400772094727**  **3: Kung Fu Panda (2008), with distance of 0.3432350754737854**  **4: Inception (2010), with distance of 0.3307400345802307**  **5: District 9 (2009), with distance of 0.31877219676971436**  **6: Up (2009), with distance of 0.31706738471984863**  **7: WALL·E (2008), with distance of 0.27033132314682007**  **8: Avengers, The (2012), with distance of 0.26102906465530396**  **9: Avatar (2009), with distance of 0.25990235805511475**  **10: Dark Knight, The (2008), with distance of 0.24018973112106323** |
| --- |

**05. Latent Factor Model(MF)**

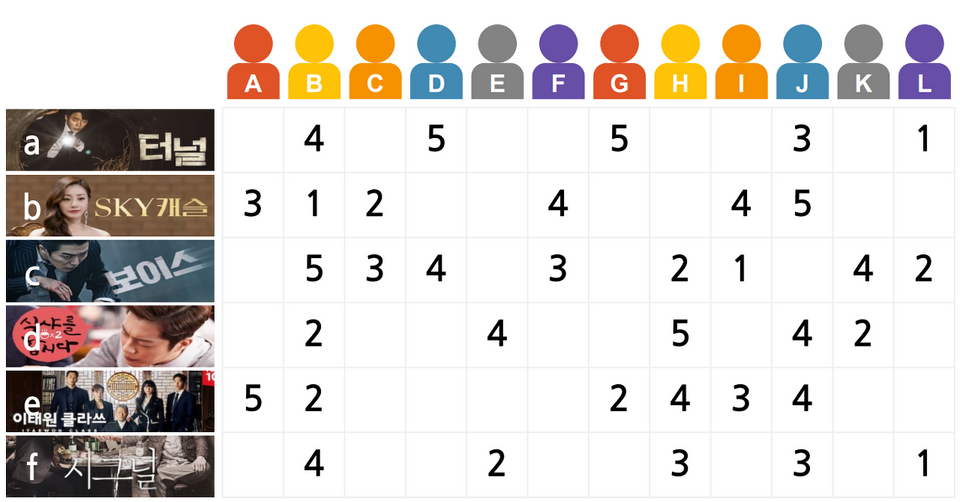
Latent Factor Model(MF)은 사용자와 아이템을, 잠재적인 차원(Factor)들을 사용해 나타낼 수 있다고 보는 모델



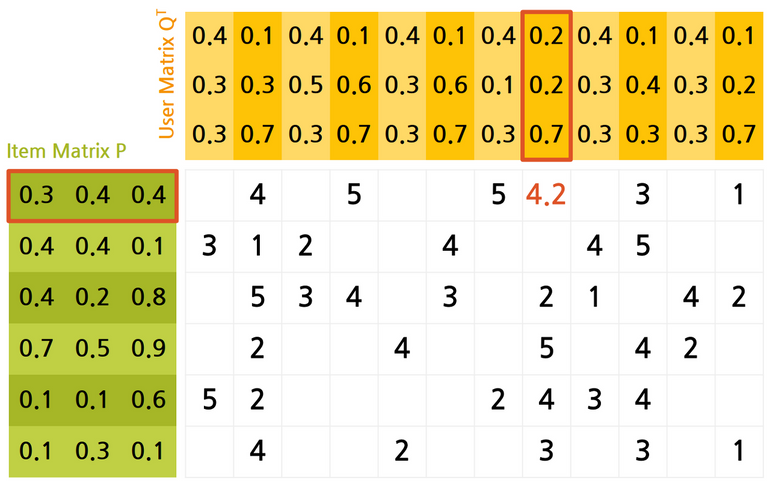
* Latent Factor : 실제로 차원의 의미는 알수 없음/여러개일수 있음
* 여성향/남성향, 웃김/심각함은 Latent Factor 즉, 분석을 통해 추출을 해야 함

**Matrix Factorization**

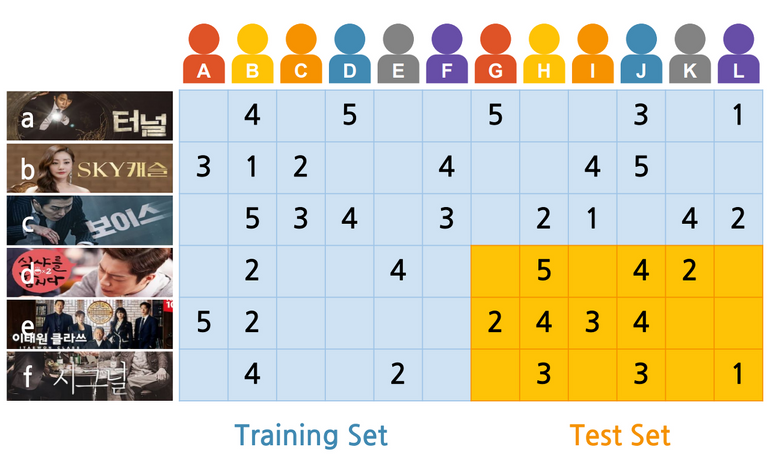
실제 영화별 사용자 평점이 다음과 같이 주어졌다고 할 때



* User Matrix와 Item Matrix 각각 3개씩 두고,
* Matrix에 임의의 소수값들을 부여한 후(처음엔 random값으로 아무렇게나 채워넣을 수 있음) ,
* 존재하는 평점을 활용하여 학습시킴  
  - 사용자 벡터 X 아이템 벡터 = 평점 (예상평점)  
  - 예상평점이 실제평점과 유사하게끔 학습시킴  
  - (사용자벡터)X(아이템벡터) = 4.8, 실제평점=5.0 ⇒ 오차:0.2
* 실제 평점과 Matrix 값이 가장 유사한 Matrix가 완성이 되면,
* **공란 즉, 아직 보지 않은 사용자의 평점도 구할 수 있다는 이론**



학습 데이터와 검증데이터 분리 결과(Result Matrix)



**06. Latent Factor Model(MF) 실습**

**샘플데이터**

ML-100K: <https://grouplens.org/datasets/movielens/>

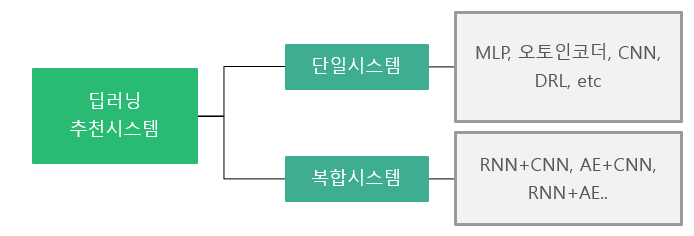
**ml-100k.zip**

movies.csv, ratings.csv

**07. Deep Learning for 추천시스템**

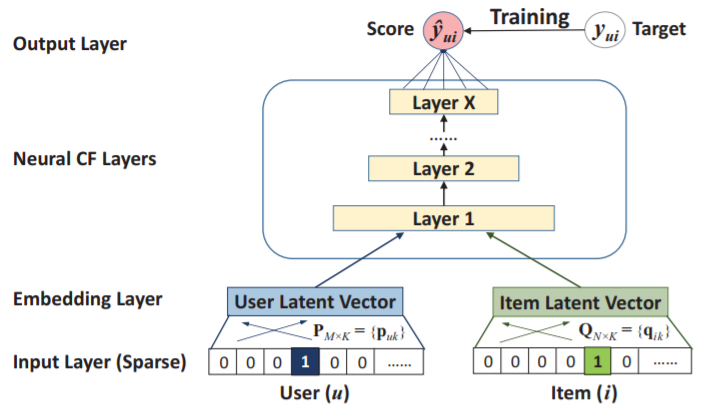
**Why Deep learning?**

* 다양한 Feature(User, Item) 간의 non-linear하고 복잡한 관계를 학습 할 수 있음  
  - 기존의 MF방법은 User와 Item간의 선형적 관계에 집중했음
* 모듈확장 및 교체가 용이   
  - 다양한 자료의 형태에 따라서 딥러닝 모형을 조합하여 거대한 추천 시스템을 구축할 수 있음(예: 이미지는 CNN, 시계열데이터는 RNN등)

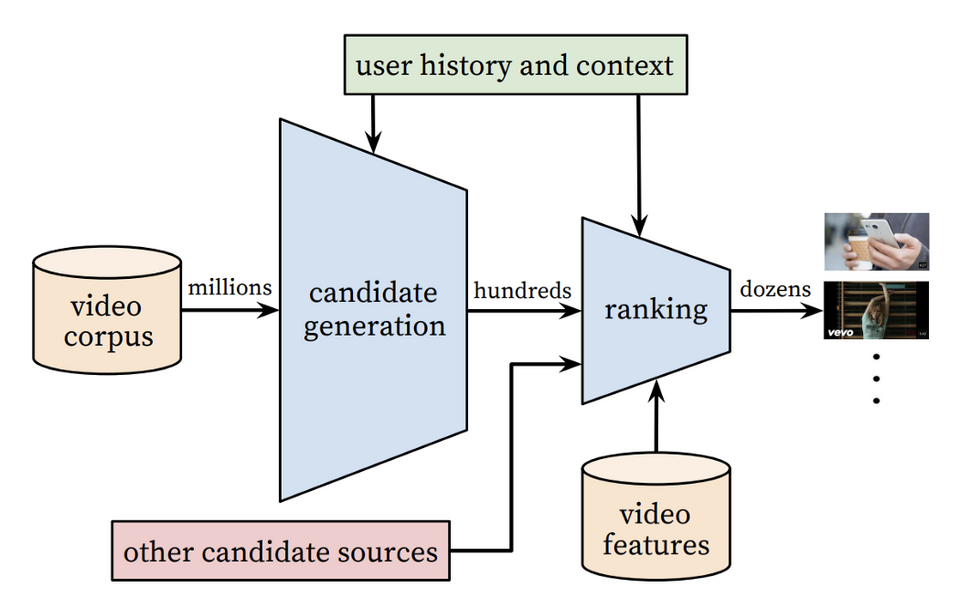


* 기존모델은 사용자 정보를 ID의 단일 값이 아닌 다양한 정보를 활용하기 유연하지 않음(나이, 성별 등 사용자 속성 반영이 어려움)
* 기존모델은 학습 데이터가 많아지면 메모리 사용량이 급격히 늘어나거나 학습 속도가 느려짐

**08. Neural Collaborative Filtering(NCF)**



**09. You Tube 딥러닝 추천시스템 이해하기**



* 두단계로 나누어져있음

**후보모델**: target 사용자에게 추천할 후보군을 뽑는 1단계 후보모델 (몇백개단위)

높은 재현율(recall)의 모델은 사용자가 선호하는 동영상은 최대한 많이 찾아내야함

**랭킹모델**: 그 후보들이 얼마나 관심있을지 순위를 매김

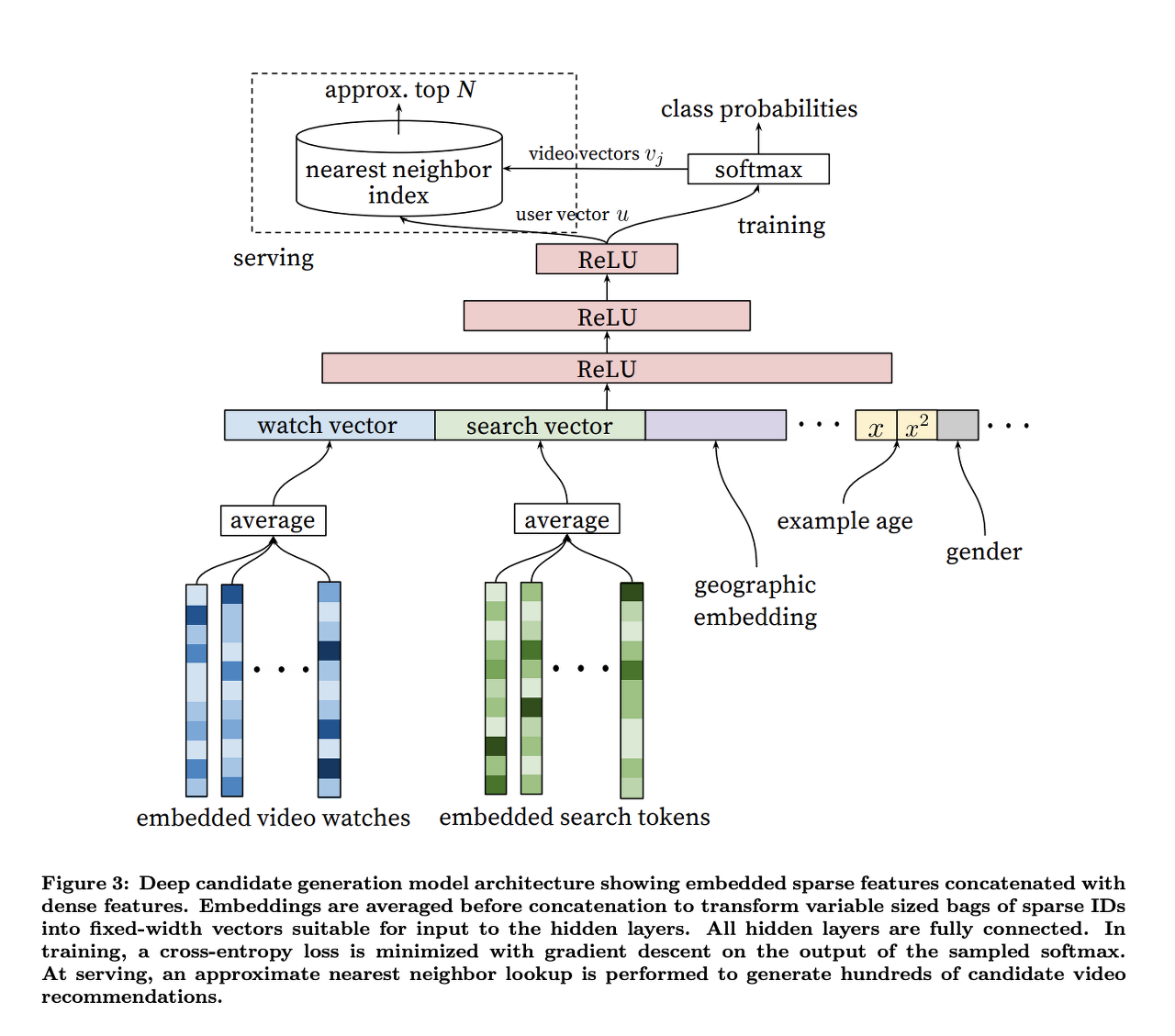
* 추천할 대상이 너무 많아 모든 경우의 수를 계산하기에는 너무 오래걸리기때문에 두단계로 하는것이 전형적

**Scale, Freshness, Noise 문제를 해결하는 모델!**

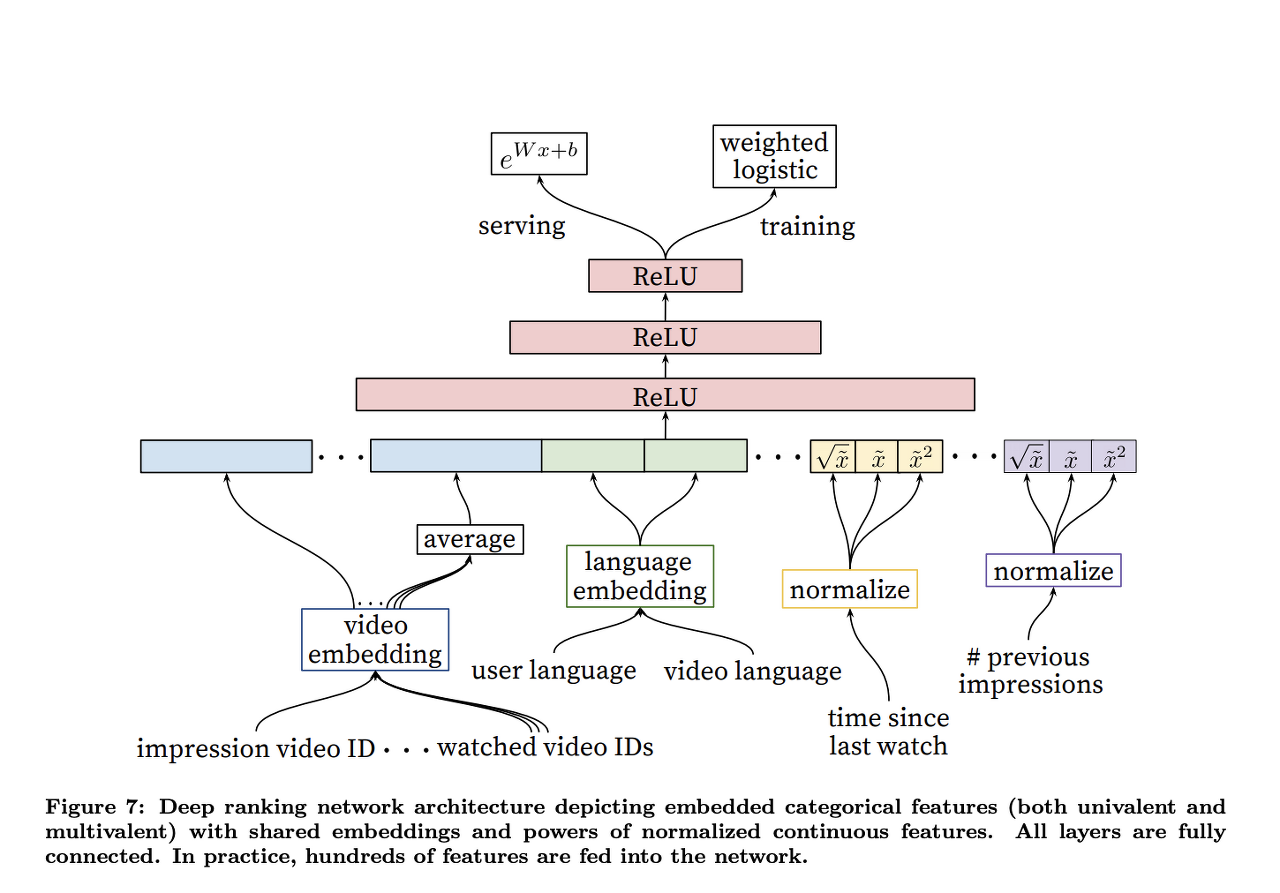
* Scale: 데이터규모
* Freshness: 데이터의 생성 주기
* Noise: 데이터의 질

**후보모델 (Candidate Generation Network)**

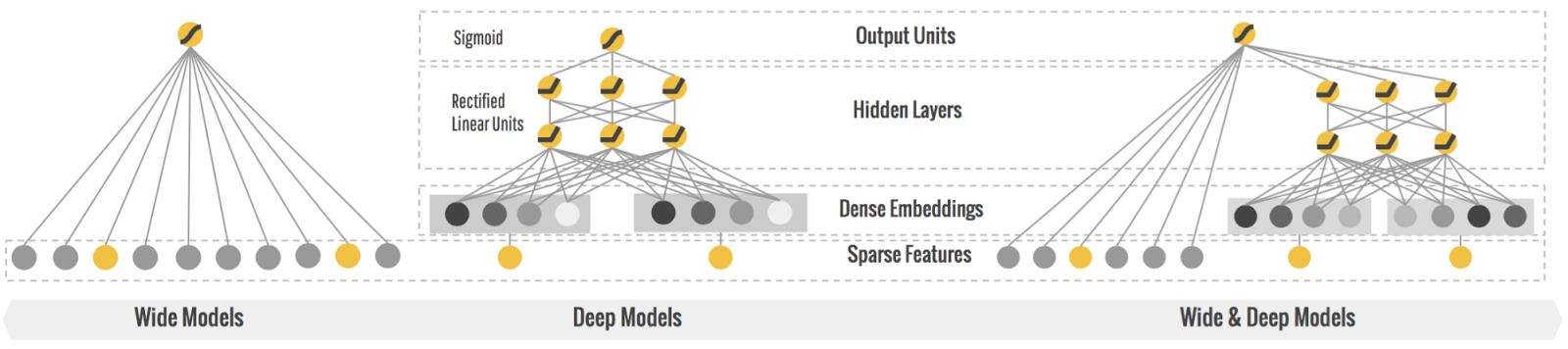
User, Context 정보가 주어지면 특정 시간에 이 비디오를 볼 확률을 구하는과정



**랭킹모델**



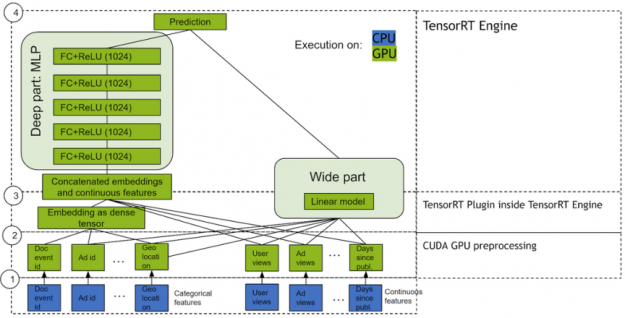
**10. Wide & Deep 모델 이해하기**



너무 구체화(overfit)되거나 너무 일반화(underfit)되는 것을 적절히 안배하기위해 고안

* 뜨거운 아메리카노를 검색했는데 "커피"라는 일반화 때문에 아이스라떼를 추천 → 따뜻한 음료를 원하는데 차가운 음료를 추천

**구글 앱스토어**

****