creative commons
COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공 학 석 사 학 위 논 문

마이크로소프트 추천기를 이용한 와이드 앤 딥 모델 기반의 도서 추천 시스템

2023년 2월

부 경 대 학 교 대 학 원

I T 융 합 응 용 공 학 과

전 유 현

공 학 석 사 학 위 논 문

마이크로소프트 추천기를 이용한 와이드 앤 딥 모델 기반의 도서 추천 시스템

지도교수 권 기 룡

이 논문을 공학석사 학위논문으로 제출함.

2023년 2월

부 경 대 학 교 대 학 원

I T 융 합 응 용 공 학 과

전 유 현

전유현의 공학석사 학위논문을 인준함

2023년 02월 17일

위 원 장 공학박사 김 영 봉 (인)

위 원 공학박사 송 하 주 (인)

위 원 공학박사 권 기 룡 (인)

목 차

Ⅰ. 서론 ····· 1

1.1 연구 배경 ····· 1

1.2 연구 목적 및 내용 ····· 2

Ⅱ. 관련연구 ····· 5

2.1 인공지능 ····· 5

2.2 머신러닝 ····· 6

2.2.1 지도학습 ····· 6

2.2.2 비지도학습 ····· 7

2.2.3 강화학습 ····· 7

2.3 딥러닝 ····· 8

2.4 추천 시스템 ····· 9

2.4.1 콘텐츠 기반 필터링 ····· 10

2.4.2 협업 필터링 ····· 11

2.4.3 하이브리드 필터링 ····· 12

2.4.4 딥러닝 기반 추천 시스템 ····· 13

Ⅲ. 제안하는 도서 추천 시스템 ····· 15

3.1 와이드 앤 딥 모델 ····· 15

3.1.1 와이드 모델 ····· 16

3.1.2 딥 모델 ····· 17

3.1.3 와이드 앤 딥 모델의 결합 ····· 19

3.2 마이크로소프트 추천기 ····· 20

Ⅵ. 실험 및 결과 ····· 22
4.1 실험 환경 ····· 22
4.2 실험 데이터 ····· 22
4.3 성능 평가 지표 ····· 25
4.4 실험 결과 및 고찰 ····· 29

Ⅴ. 결론 ····· 34

참고문헌 ····· 36

PUKYONG NATIONAL UNIVERSITY
부경대학교

표 차 례

[표 1] 추천 시스템 예시 ····· 10
[표 2] 알고리즘별 추천 시스템 논문 ····· 14
[표 3] 마이크로소프트 추천기 알고리즘 리스트 ····· 21
[표 4] 실험 환경 ····· 22
[표 5] GoodBooks-10k 데이터셋 ····· 24
[표 6] 4,000번 훈련한 결과 ····· 29
[표 7] 20,000번 훈련한 결과 ····· 29
[표 8] 은닉층의 노드 수 변화에 따른 결과 ····· 32
[표 9] 학습률 변화에 따른 결과 ····· 32
[표 10] 와이드 앤 딥 모델과 NMF의 성능 비교 ····· 33

그 림 차 례

[그림 1] 머신러닝의 학습 종류 ····· 6
[그림 2] 콘텐츠 기반 필터링의 추천 과정 ····· 11
[그림 3] 협업 필터링의 추천 과정 ····· 12
[그림 4] 와이드 모델 ····· 17
[그림 5] 딥 모델 ····· 18
[그림 6] 와이드 앤 딥 모델 ····· 19
[그림 7] GoodBooks-10k 데이터셋 ····· 23
[그림 8] 추출한 GoodBooks-10k 데이터셋 ····· 24
[그림 9] 장르 데이터를 변환한 GoodBooks-10k 데이터셋 ····· 25
[그림 10] 추출한 사용자 아이디, 도서 아이디와 장르 데이터 ····· 26
[그림 11] 추출한 데이터의 평점 예측 ····· 27
[그림 12] 4,000번 훈련한 결과 그래프 ····· 30
[그림 13] 20,000번 훈련한 결과 그래프 ····· 31
[그림 14] 딥 모델과 와이드 모델에서 각각 학습률을 0.1과 0.001로 지정하였을 때의 결과 그래프 ····· 33

마이크로소프트 추천기를 이용한
와이드 앤 딥 모델 기반의 도서 추천 시스템

전 유 현

부 경 대 학 교 대 학 원 IT 융 합 응 용 공 학 과

요 약

빅데이터 시대로 다양하고 많은 정보가 쏟아지고 있다. 이러한 과잉 정보 시대에 개인화된 추천 시스템(Recommender System)은 사용자가 상품을 고르고 찾아보는 데 드는 시간을 줄여주고 고객의 취향에 맞는 상품을 제안하여 구매율이 늘어난다. 대표적인 추천 알고리즘으로는 콘텐츠 기반 필터링(Content-Based Filtering), 협업 필터링(Collaborative Filtering) 및 하이브리드 필터링(Hybrid Filtering)이 있지만 성능 개선을 위해 딥러닝(Deep Learning) 기반의 추천 시스템이 연구되고 있다. 하지만 딥러닝 기반의 추천 시스템은 구조가 복잡하여 구현이 어렵다는 단점이 있다. 이를 개선하기 위해 본 논문에서는 효율적이고 생산적인 구축이 가능한 마이크로소프트 추천기(Microsoft Recommenders)를 사용하여 와이드 앤 딥 모델(Wide and Deep Model) 기반의 도서 추천 시스템을 제안한다. 마이크로소프트 추천기는 마이크로소프트에서 제공하는 오픈 소스 리포지터리(repository)로 최적화된 추천 시스템 개발을 위해 다양한 알고리즘과 기능을 제공하고 있다. 제안한 방법을 통해 실험한 결과 안정적인 성능의 딥러닝 모델 기반의 도서 추천 시스템을 구축할 수 있었다.

A Book Recommender System based on Wide and Deep Model using Microsoft Recommenders

Yoo-Hyun Jun

Department of IT Convergence and Application Engineering of Pukyong National University

Abstract

In the era of big data, a lot of diverse information is pouring in. In this age of excessive information, personalized recommendations will reduce the time required for users to select and search for products, and will increase the purchase rate by suggesting products that suit the customer's taste. Representative recommendation algorithms include content-based filtering, collaborative filtering, and hybrid filtering, but recommendation systems based on deep learning are being studied to improve performance. However, recommender systems based on deep learning have a complex structure. So, It is difficult to implement. To improve this, this paper proposes a book recommendation system based on wide and deep model using the Microsoft Recommenders, which can be built efficiently and productively. The Microsoft Recommenders is an open source repository provided by Microsoft and provides various algorithms and functions for developing an optimized recommender system. As a result of experimentation through the proposed method, it was possible to build a book recommendation system based on deep learning model with stable performance.

Ⅰ. 서 론

1.1 연구 배경

인터넷의 발달과 스마트폰의 보급으로 인해 방대한 양의 정보들이 생성되고 다양한 형태의 대규모 데이터가 쌓이는 빅데이터 시대가 도래하였다. 이러한 과잉 정보 시대에 사용자 취향에 맞는 개인화된 추천은 중요한 과제로 대두되었다. 전자상거래에서 사용자의 특성을 고려한 맞춤형 추천을 제공하여 고객이 상품을 찾아보고 선택하는 데 소요되는 시간을 줄일 수 있고, 회사에서는 고객의 취향에 맞는 상품을 예측하고 제안하여 구매율이 늘어 매출을 증가시킬 수 있다.

현재 넷플릭스(Netflix), 유튜브(YouTube), 페이스북(Facebook) 그리고 구글(Google)과 같은 많은 회사에서 사용자의 정보를 사용하여 추천 리스트를 만들어 제공하고 있다[1-4]. 전자상거래 기반의 종합 쇼핑몰인 아마존(Amazon)은 사용자의 구매 정보와 검색 정보를 기반으로 사용자에게 맞춤 추천 상품을 제공하기 위해 다양한 추천 알고리즘을 사용하고 있다[5].

기존의 추천 알고리즘보다 추천의 정확도를 높이기 위해 다양한 분야에서 좋은 성과를 보여주고 있는 딥러닝을 이용한 추천 시스템 연구가 이루어지고 있다. 딥러닝 기반의 추천 모델을 사용하여 기사, 음악, 영화 등 다양한 분야에서 연구가 진행되고 있다[6-8].

딥러닝 기반의 추천 시스템 모델인 와이드 앤 딥 모델은 구글에서 개발한 추천 시스템 모델로 구글 앱 스토어에서 앱 추천 시스템에 사용하고 있는 모델이다[9]. 대학 도서관 추천 시스템에 와이드 앤 딥 모델을 적용한 연구가 진행되었다[10]. 하지만 딥러닝 기반의 추천 시스템은 구조가 복잡

하여 구현하기가 어렵다는 단점이 있다.

텐서플로우(Tensor Flow)는 딥러닝과 머신러닝 등에 사용되는 대표적인 오픈소스 소프트웨어 중 하나이다[11]. 텐서플로우는 구글에서 개발한 프레임 워크로 딥러닝과 머신러닝 개발에 필요한 다양한 기능을 제공하며 확장성이 뛰어나다는 장점이 있다. 안정적인 도서 추천 시스템의 구축을 위해 텐서플로우를 사용한 연구가 진행되었다[12]. 하지만 텐서플로우 프레임 워크에 대한 이해를 위해 많은 시간을 들여야 하며 딥러닝 모델을 만들기 위해 기초 작업부터 직접 구축해야 하므로 초보자가 쉽게 접근하기 어렵다는 단점이 있다.

1.2 연구 목적 및 내용

본 논문에서는 구조가 복잡하여 구현이 어려운 딥러닝 추천 시스템의 단점을 개선하기 위해 마이크로소프트에서 제공하는 추천 시스템 개발을 위한 오픈 소스 리포지터리인 마이크로소프트 추천기[13]를 사용하여 생산적인 구축이 가능한 딥러닝 기반의 도서 추천 시스템을 제안한다.

추천 시스템은 주로 사용자로부터 정보를 분석하고 고객으로부터 받은 데이터를 기반으로 다른 제품에 대한 사용자의 관심을 예측하는 데 사용된다. 따라서 이러한 사용자 정보를 이용하여 사용자의 관심사를 파악하고 적절한 추천 리스트를 생성하여 이를 제공하는 것이 중요하다. 추천 시스템은 음악, 비디오 및 이미지 등 다양한 분야에서 연구되고 있으며 사용자의 특징을 파악하여 사용자가 관심 가질 수 있는 추천 리스트를 만들어 제공한다[14-16]. 식당, 패션 등 다양한 분야의 서비스에 추천 시스템이 사용되고 있으며 계속해서 개발 및 업데이트되고 있다[17,18].

사용자를 위한 기존의 추천 알고리즘으로는 콘텐츠 기반 필터링, 협업

필터링과 하이브리드 필터링이 주로 사용되었다. 콘텐츠 기반 필터링은 사용자가 과거에 사용한 콘텐츠를 기반으로 유사한 콘텐츠를 추천한다[19]. 협업 필터링은 과거 경험을 기반으로 하며 취향이 비슷한 사용자 간의 정보를 비교하여 추천한다[20]. 하이브리드 필터링은 콘텐츠 기반 필터링과 협업 필터링을 혼합한 방식으로 좋은 성능을 보이지만 구현 부분이 까다로워 적용이 어렵다는 단점이 있다[21]. 현재 추천 시스템은 기존의 추천 시스템 필터링을 서로 다른 조합으로 구성하여 제공하고자 하는 아이템에 맞는 추천 시스템 모델을 구축하여 사용하고 있다.

최근에는 딥러닝의 빠른 성장으로 인해 다양한 분야에 적용되어 좋은 성능을 보여주고 있으며, 추천 시스템에서도 딥러닝 기반의 추천 시스템 모델이 연구되고 있다. 하지만 딥러닝 기반의 추천 시스템 구축에는 높은 기술력과 많은 시간이 든다는 단점이 있다.

이를 개선하고자 본 논문에서는 마이크로소프트 추천기를 이용한 와이드 앤 딥 모델 기반의 도서 추천 시스템을 제안한다. 최적화된 도서 추천 시스템을 구현하기 위해 와이드 앤 딥 모델에 은닉층(hidden layer)의 노드(node) 수와 학습률(learning rate)을 조절하여 실험을 진행하였다. 각 은닉층의 노드 수를 16, 32, 128로 지정했을 때 평균 절대 오차(Mean Absolute Error, MAE)는 0.8110, 평균 제곱근 오차(Root Mean Squared Error, RMSE)는 0.9948로 가장 좋은 성능을 보여주는 것을 확인하였다. 가장 좋은 성능을 보여준 각 은닉층의 노드 수에서 학습률의 값을 조절하여 실험을 진행한 결과, 딥 모델에서 학습률 값을 0.1로 와이드 모델에서 학습률을 0.001로 지정하였을 때 평균 절대 오차는 0.7967, 평균 제곱근 오차는 0.9874로 가장 좋은 성능을 확인할 수 있었다.

본 논문은 다음과 같이 구성된다. 2장에서 딥러닝 기반의 추천 시스템에 대한 연구를 설명하고, 3장에서는 와이드 앤 딥 모델 기반의 추천 시스템

과 마이크로소프트 추천기를 설명하여 본 논문에서 제안하는 연구 방법에 대한 이해를 돕고자 한다. 4장에서는 딥러닝 기반의 도서 추천 시스템의 실험을 통해 실험 결과에 대한 분석 및 고찰한다. 마지막 5장에서는 논문에서의 최종 결론을 서술한다.

PUKYONG NATIONAL UNIVERSITY
부경대학교

Ⅱ. 관련연구

본 장에서는 인공지능(Artificial Intelligence, AI), 머신러닝(Machine Learning), 딥러닝을 설명하고 기존의 추천 시스템과 딥러닝 기반의 추천 시스템을 살펴보고 개발 도구인 텐서플로우와 마이크로소프트 추천기에 대하여 알아본다.

2.1 인공지능

인공지능은 학습, 문제 해결, 패턴 인식 등과 같이 인간의 지능과 연결된 문제를 해결하기 위한 컴퓨터 시스템으로 인간의 지능을 기계 등에 인공적으로 구현한 것이다.

인공지능이라는 용어는 1956년 미국 다트머스 학술회의에서 존 매카시(John McCarthy)가 처음 사용하였다. 인공지능이 해결하는 문제에 따라 약한 인공지능과 강한 인공지능 2가지로 구분한다[22].

약 인공지능은 기존에 인간은 쉽게 해결할 수 있으나 컴퓨터로 처리하기에는 어려웠던 사진에서 물체를 인식하거나 소리를 듣고 상황을 파악하는 것과 같은 인식 분야 문제를 컴퓨터로 수행하는 것에 중점을 두고 있다. 인간과 유사한 지능을 목표로 하기보다는 현실적이고 실용적인 목표를 가지고 개발되고 있는 인공지능이라고 할 수 있으며 특정한 문제를 해결하는 도구로 활용되고 있다.

강 인공지능은 일반적으로 모든 상황에 대해 스스로 학습과 행동이 가능하며, 그 수준이 최소한 인간의 지적 수준과 유사한 경우를 의미한다. 강 인공지능은 현실적으로 구현의 어려움으로 인해 현재까지는 큰 성과가 없

는 상태이다[22].

2.2 머신러닝

머신러닝은 인공지능의 알고리즘으로 전자 상거래에서 추천부터 이미지 인식, 카메라 및 스마트폰과 같은 제품에 결합하여 사용되고 있다[23]. 머신러닝은 데이터를 기반으로 학습하고 이를 통해 이미지를 식별하며 음성을 텍스트로 변환하고, 사용자의 관심사를 학습하여 뉴스, 콘텐츠, 상품을 제공하여 선택하는 데 도움을 주고 있다. 머신러닝의 학습 종류는 지도학습(Supervised Learning), 비지도학습(Unsupervised Learning), 강화학습(Reinforcement Learning)이 있으며 그림 1과 같다.

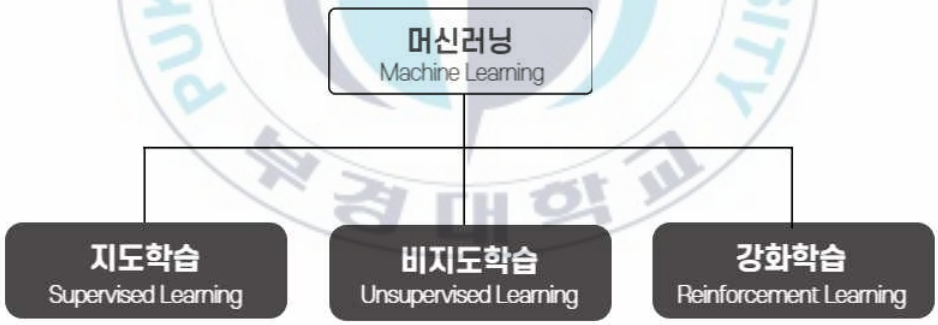

그림 1. 머신러닝의 학습 종류

2.2.1 지도학습

지도학습에서는 데이터가 입력값과 출력값이 함께 주어진다. 입력값에 대한 출력값을 알려준다는 의미에서 지도학습이라는 명칭을 사용하며 지도

학습으로 해결할 수 있는 대표적인 문제에는 회귀(regression)와 분류(classification)가 있다. 회귀는 출력이 연속된 실수로 주어지고 분류는 몇 가지 부류로 주어진다[23]. 최근 딥러닝이 보편화되면서 다양한 문제에 머신러닝을 적용하여 응용 범위를 점점 확대하는 추세이다.

2.2.2 비지도학습

비지도학습에서는 입력값만 주어진다. 특징 벡터만 주어지고 출력값은 제공하지 않아 지도하지 않는다는 의미에서 비지도학습이라고 한다[23]. 비지도학습은 출력값 없이 인공지능이 입력 데이터에서 패턴과 상관관계를 찾아내도록 하는 머신러닝의 학습 방법이다. 비지도학습은 데이터 양이 많지 않거나 훈련 데이터를 수집하는 데 드는 비용이 높을 경우에 주로 사용된다. 비지도학습은 머신러닝이 스스로 문제를 풀도록 내버려 두고 스스로 학습하도록 한다. 대표적인 비지도학습으로 군집화(clustering)와 차원 축소(dimentionality reduction)가 있다.

2.2.3 강화학습

강화학습은 행동 심리학에서 나온 이론으로 어떤 환경에서 주체(agent)가 현재의 상태(state)에서 선택할 수 있는 행동(action) 중에서 최대의 보상(reward)을 주는 행동을 학습하는 것이다. 강화학습도 지도학습처럼 출력값을 주어 지도하는데, 출력값 형태의 차이가 있다. 장기 게임을 예로 들면, 장기에서는 두 사람이 번갈아 수를 놓는데 각각의 수를 입력값으로 볼 수 있다. 게임이 다 끝난 후 승패를 통해 얻은 점수를 출력값으로 준다. 입

력값마다 출력값을 주는 지도학습과 달리 강화학습은 연속된 입력값의 열에 출력값 하나만 주는 방식이다. 따라서 각각의 입력값에 출력값을 배분하는 알고리즘이 추가로 필요하다[23].

2.3 딥러닝

다층 퍼셉트론(Multi Layer Perceptron, MLP)에 은닉층을 여러 개 추가하면 심층 신경망이 되며, 심층 신경망을 학습시키는 알고리즘을 딥러닝이라고 한다. 딥러닝은 복잡하여 연구 성과가 없던 영상 인식 문제, 성능이 낮아 실용화하기 힘들었던 음성 인식과 언어 번역과 같은 분야에서 머신러닝 알고리즘을 개선하여 획기적으로 높은 성능을 보여주었다.

이전의 음성 인식 분야는 더딘 성능 향상으로 연구가 지체되고 있었다. 음성 데이터베이스를 추가하고 머신러닝 알고리즘을 개선하며, 매개변수도 조절해가며 여러 변화를 주어도 성능이 미세하게만 향상되었다. 성능 향상에 침체기였던 음성 인식에 딥러닝을 적용하자 순식간에 25%의 오류율 저하가 이루어졌다. 이는 10년 치의 성능 향상에 해당하는 수치였다[20]. 이러한 획기적인 성능 개선으로 인해 딥러닝에 대한 관심이 증가하게 되었다.

컨볼루션 신경망(Convolutional Neural Network, CNN)은 대표적인 딥러닝 모델로 작은 크기의 컨볼루션 마스크를 사용하기 때문에 완전 연결 구조인 다층 퍼셉트론보다 매개변수가 훨씬 적다. 또한 모든 노드가 가중치를 공유하는 가중치 공유 기법을 사용하여 성능 개선이 이루어지게 되었다. 컨볼루션 신경망의 등장으로 성능 개선에 더딘 영상 인식과 음성 인식 분야에서 성능 개선이 가능해지며 딥러닝에 대한 관심이 증가하게 되었다[23].

2.4 추천 시스템

스마트폰의 보급으로 방대한 데이터가 축적되고 있다. 예전에는 사람들이 정보를 찾는 것이 중요한 과제였다면 다양한 매체에서 정보가 쏟아지는 빅데이터 시대에는 방대한 정보 속에서 개인의 취향을 파악해서 사용자 맞춤형 콘텐츠를 제공하는 것이 중요해지게 되었다. 그러므로 개인화된 추천 시스템에 대한 연구가 활발히 진행되고 있다. 현재 추천 시스템은 표 1에서와 같이 넷플릭스, 아마존, 유튜브, 구글 등 다국적 기업에서 영상, 상품, 앱 등을 추천 시스템을 적용하여 서비스하고 있으며 각 기업의 특색에 맞는 상품에 대한 추천 시스템을 자체적으로 연구하고 계속해서 업데이트하여 좋은 성능의 추천 시스템 개발을 위해 노력하고 있다. 추천 시스템은 개인화된 추천으로 인해 고객이 상품을 찾는 시간을 줄여줄 수 있으며 상품 구매까지 연결되어 매출 상승의 효과까지 기대할 수 있다.

넷플릭스에서는 영화와 시리즈를 추천해주고, 구글 뉴스(Google News)에서는 뉴스를, 페이스북에서는 사람과 광고를 추천해주며 유튜브에서는 온라인 비디오 등 다양한 분야에서 추천이 이루어지고 있다[1-4].

기업에서는 추천하고자 하는 분야에 따라 적합한 입력 데이터를 사용하여 추천 리스트를 제공하고 있다. 예를 들어, 구글은 사용자의 검색 정보를 입력 데이터로 사용한다. 유튜브와 같은 다른 구글 제품을 통해 수집된 데이터를 기반으로 사용자에게 관련성 있는 추천 리스트를 제공하고 있다.

표 1. 추천 시스템 예시

Platform	Contents
넷플릭스[1]	시리즈, 영화
유튜브[2]	온라인 비디오
페이스북[3]	친구, 광고
구글 뉴스[4]	뉴스 기사

기존의 추천 시스템 모델로는 콘텐츠 필터링, 협업 필터링과 이 둘을 혼합한 하이브리드 필터링이 주로 사용되었다. 최근 딥러닝의 발전으로 인해 딥러닝을 적용한 추천 시스템 연구가 진행되고 있다.

2.4.1 콘텐츠 기반 필터링

콘텐츠 기반 필터링은 아이템에 대한 프로필 정보와 사용자의 선호도 프로필을 기반으로 사용자에게 아이템을 추천하는 방식이다[19]. 사용자가 과거 선호한 아이템과 유사한 상품을 추천해주며 아이템 간의 유사도에 기초하여 추천 리스트를 선정한다.

콘텐츠 기반 필터링의 추천 과정은 그림 2와 같다. 사용자 1이 A라는 영화를 선호했을 때, A 영화의 영화감독인 가의 다른 작품인 영화 B를 추천해준다. 추천하기 위해 사용되는 아이템의 정보는 감독뿐만 아니라 리뷰, 평점, 출연 배우 등 다양한 정보를 사용할 수 있다.

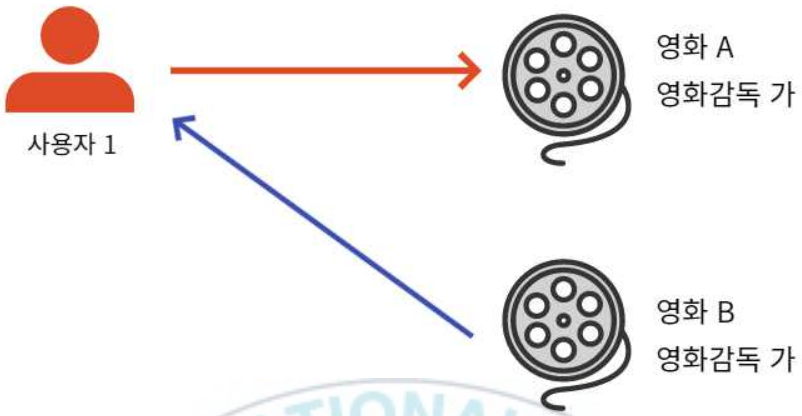

그림 2. 콘텐츠 기반 필터링의 추천 과정

콘텐츠 기반 필터링은 구조가 단순하여 구현이 쉬워 초기 추천 시스템에서 많이 사용되었다. 웹 페이지, 뉴스 기사, 음식점, TV 프로그램 및 판매 상품 추천에 이르기까지 다양한 영역에서 사용될 수 있다.

2.4.2 협업 필터링

협업 필터링은 여러 사용자가 제공한 평가 정보를 사용하여 추천 리스트를 만든다[20]. 유사도를 계산하여 추천하고자 하는 사용자와 선호도가 유사한 다른 사용자가 선택한 아이템을 추천한다. 같은 아이템을 선택한 사용자들은 선호도가 유사할 것이라는 가정을 기반으로 한 모델이다.

협업 필터링의 추천 과정은 그림 3과 같다. 사용자 1과 사용자 3이 사과를 선택했다고 가정했을 때, 사용자 1과 사용자 3은 유사한 취향을 가졌다고 보고 사용자 1이 선택했던 아이템 중 사용자 3이 선택하지 않았던 체리

를 추천하는 방식이다.

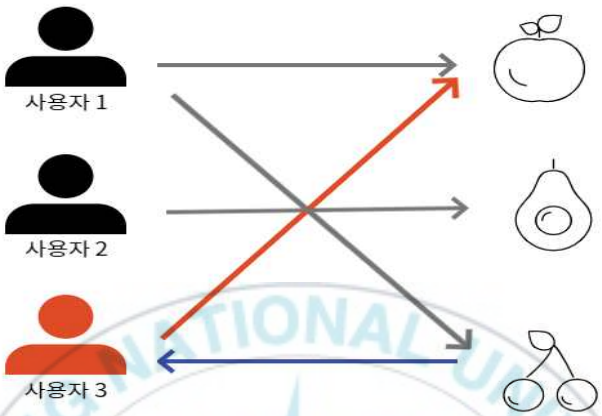

그림 3. 협업 필터링의 추천 과정

대표적인 협업 필터링 알고리즘으로 특이값 분해(Singular Value Decomposition, SVD)와 음수 미포함 행렬 분해(Non-negative Matrix Factorization, NMF)가 있다.

2.4.3 하이브리드 필터링

추천 시스템에 많이 사용되는 협업 필터링은 초기 사용자의 데이터가 부족할 경우 적절한 아이템 추천을 하지 못하는 콜드 스타트(cold start) 문제가 있다. 이를 개선하기 위해 데이터가 적은 초기에는 콘텐츠 기반 필터링으로 추천하고 데이터가 쌓이면 협업 필터링을 사용하는 하이브리드 필터링을 사용한다. 하이브리드 필터링은 콘텐츠 기반 필터링과 협업 필터링을 결합하여 각각의 단점을 보완하고, 장점을 결합하여 더 높은 성능의 추

천 결과를 달성하는 알고리즘이다. 하이브리드 필터링은 좋은 성능을 보여주지만 구조가 복잡하여 구현이 어렵다는 단점이 있다[21].

2.4.4 딥러닝 기반 추천 시스템

심층 신경망은 음성 인식, 컴퓨터 비전 및 자연어 처리에서 엄청난 성과를 내고 있다. 심층 신경망의 뛰어난 유연성 덕분에 강력하고 성능 좋은 많은 모델을 구축할 수 있게 되었다[24]. 추천 시스템 분야에서도 딥러닝을 사용한 연구가 다양한 분야에서 진행되고 있다. 딥러닝 모델을 활용하여 가능한 방법이 여러 가지이지만 연구 분야와 데이터에 적합한 모델을 신중하게 선정하여 설계되어야 한다. 여기서는 일부 응용 분야에서 효과가 입증된 딥러닝 기반의 추천 시스템 연구를 알아보고자 한다.

심층 오토 인코더(deep auto encoder) 네트워크를 사용한 딥러닝 기반의 추천 모델을 이용하여 사용자가 동일한 필름 및 다른 영화를 보고 분류한 다른 사용자의 취향에 따라 사용자가 필름에 할당하는 평점을 예측하거나 온라인 소셜 미디어에서의 추천 시스템이 연구되었다[25,26]. 이외에도 심층 오토 인코더를 활용한 추천 시스템 연구에서 좋은 성능을 보여주었다[27-29].

다층 퍼셉트론을 사용한 딥러닝 기반의 추천 시스템 모델은 신경망의 유연성, 복잡성과 비선형성을 사용하여 추천 시스템 매트릭스를 만든다[30,31]. 다층 퍼셉트론을 이용하여 음악과 친구 추천 연구가 진행되었다[32,33].

CNN 기반의 상품, 영화, 음악과 지역 추천 시스템[34-37], RNN 기반의 추천 시스템[38-40], DNN 기반의 친구 추천 시스템[41,42], 데이터에 따라 여러 조합으로 구성한 딥러닝 기반의 하이브리드 추천 시스템[43-45]에 대

한 연구가 표 2에서와 같이 진행되었다.

표 2. 알고리즘별 추천 시스템 논문

Algorithm	Publications
Auto Encoder	[25,26,27,28,29]
MLP	[30,31,32,33]
CNN	[34,35,36,37]
RNN	[38,39,40]
DNN	[41,42]
Hybrid Models	[43,44,45]

위에서 살펴본 모델은 추천 시스템을 더욱 개인화, 전문화하여 더 나은 성능을 보여주었다. 도서 추천 분야에서는 로지스틱 회귀(Logistic Regression, LR)와 심층 신경망(Deep Neural Network, DNN)을 결합한 하이브리드 기반의 딥러닝 추천 시스템 모델인 와이드 앤 딥 모델을 사용하여 도서관 대출 추천 시스템에 적용한 연구가 있었다[10]. 하지만 로지스틱 회귀와 심층 신경망을 결합하여 구축해야 하므로 모델 구축에 많은 시간이 들고 추천 시스템과 딥러닝에 대한 높은 수준의 지식과 개발 기술이 요구되어 구현이 어렵다는 단점이 있다.

Ⅲ. 제안하는 도서 추천 시스템

제안하는 와이드 앤 딥 모델 기반의 도서 추천 시스템[46,47]에서 사용한 와이드 앤 딥 모델과 마이크로소프트 추천기에 대하여 알아보고자 한다.

3.1 와이드 앤 딥 모델

와이드 앤 딥 모델은 구글에서 2016년 발표한 딥러닝 추천 시스템 모델로 구글 앱 스토어에서 앱 추천에 사용되고 있다[9]. 사용자의 검색 쿼리를 입력 데이터로 사용하여 선호도를 파악한 후 추천 매트릭스를 생성하여 상위 10여 개의 어플을 추천해준다. 다양한 분야에서 적용하여 연구가 진행되고 있는 대표적인 딥러닝 추천 모델 중 하나이다.

와이드 앤 딥 모델은 기억(memorization)에 강점이 있는 선형 모델과 일반화(generalization)에 강점이 있는 심층 신경망 모델의 장점을 결합한 모델로 뻔하거나 중복되는 추천을 하는 선형 모델의 단점과 사용자의 선호도와 상반된 추천을 할 수 있는 심층 모델의 단점을 상호 보완하여 성능이 향상된 모델이다.

회귀나 분류 문제를 풀 때, 주로 선형 모델을 사용한다. 선형 모델은 데이터 특징 간의 상호작용을 기억하는 데 효과적이다. 하지만 지나친 일반화로 뻔한 추천만 할 수 있다는 단점이 있다. 반면, 임베딩을 이용한 심층 신경망 모델은 이전에 발생하지 않은 새로운 특징 간의 조합을 학습하여 다양한 추천을 가능하게 한다. 하지만 데이터의 특징을 자세히 기억하지 못하는 문제가 있다. 선형 모델과 심층 신경망 모델의 문제를 개선하기 위하여 기억에 특화된 선형 모델과 일반화에 특화된 심층 신경망 모델을 결

합한 것이 와이드 앤 딥 모델이다.

와이드 앤 딥 모델은 구글플레이 스토어에서 앱을 추천하는 데 사용되는 모델이며 사용자가 검색한 값과 아이템, 사용자의 설치 기록 및 열어본 기록을 전처리 후 모델 학습 데이터로 사용하여 검색한 쿼리 값에 대한 수백 개의 랭킹을 만들고 그중에서 상윗값을 추천해준다. 도서 분야에서 와이드 앤 딥 모델을 적용하여 도서관 대출 추천 시스템 연구가 진행되었다[10].

3.1.1 와이드 모델

와이드 앤 딥 모델의 와이드 모델은 선형 모델로 입력 데이터에서 사용자와 아이템 사이의 직접적인 연결 관계를 학습한다. 식은 다음과 같다.

$$y = w^T x + b \tag{1}$$

여기서 y는 최종 예측의 확률, x는 특징 벡터, w는 최종적으로 훈련된 각 특징 벡터의 가중치인 모델의 매개변수이다. 그리고 b는 실제 결과와 예측 결과 사이의 편향(bias)이다.

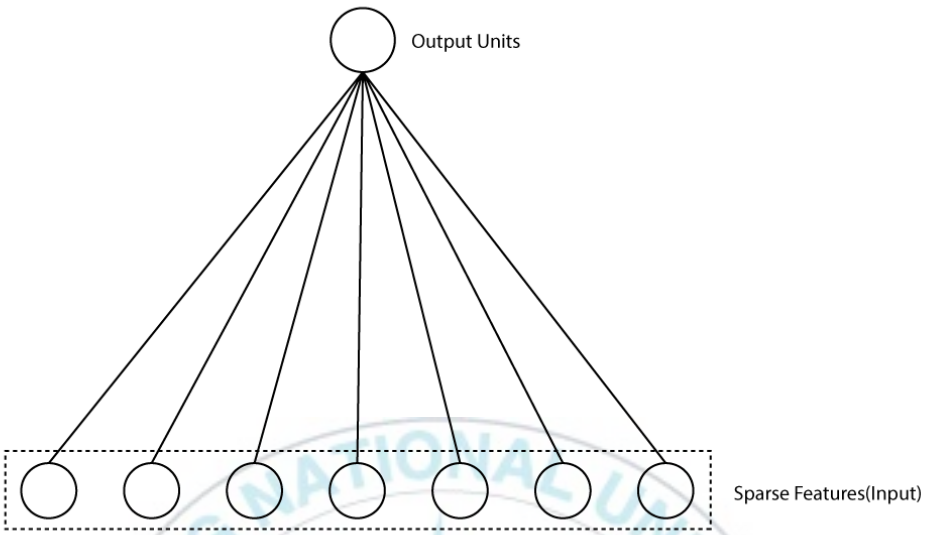

그림 4. 와이드 모델

와이드 모델은 그림 4와 같다. 입력 데이터로 처리하기 위해 원 핫 인코딩(One Hot Encoding)을 사용한다. 희소 특징(Sparse Features)에 대해 교차곱 변환을 적용한다. 교차 계산을 통해 비선형 관계를 학습한다. 이때, 값은 0 아니면 1로 나타난다. 이러한 방법은 1이 되는 모든 경우의 수를 학습하기 때문에 기억에 강하며 사용자의 취향이 반영된 특징을 학습하는데 좋은 성능을 보여주지만 0이 되는 값은 학습이 안 된다는 단점이 있다.

3.1.2 딥 모델

와이드 앤 딥 모델의 딥 모델은 심층 신경망 모델로 구조는 그림 5와 같다. 데이터 흐름 방향의 관점에서, 데이터 흐름은 입력 데이터가 네트워크에 들어간 후에는 두 계층 사이에 피드백이 없는 단방향이다[9]. 즉, 입력값은 이전 레이어로 반환되지 않는다. 딥 모델의 입력값은 일반적으로 문

자열 유형이며, 이러한 문자열 값을 벡터로 변환하여 사용한다. 변환된 벡터는 일반적으로 임의의 방법으로 초기화되며, 균일하게 또는 무작위로 분포할 수 있다. 무작위로 분포하는 목적은 벡터의 크기를 무작위로 초기화하고 훈련 프로세스에서 손실 함수를 최소화하여 모델을 최적화하기 위해서이다. 임베딩 벡터는 신경망의 은닉층에서 과거 데이터에 존재하지 않는 특징을 학습한다.

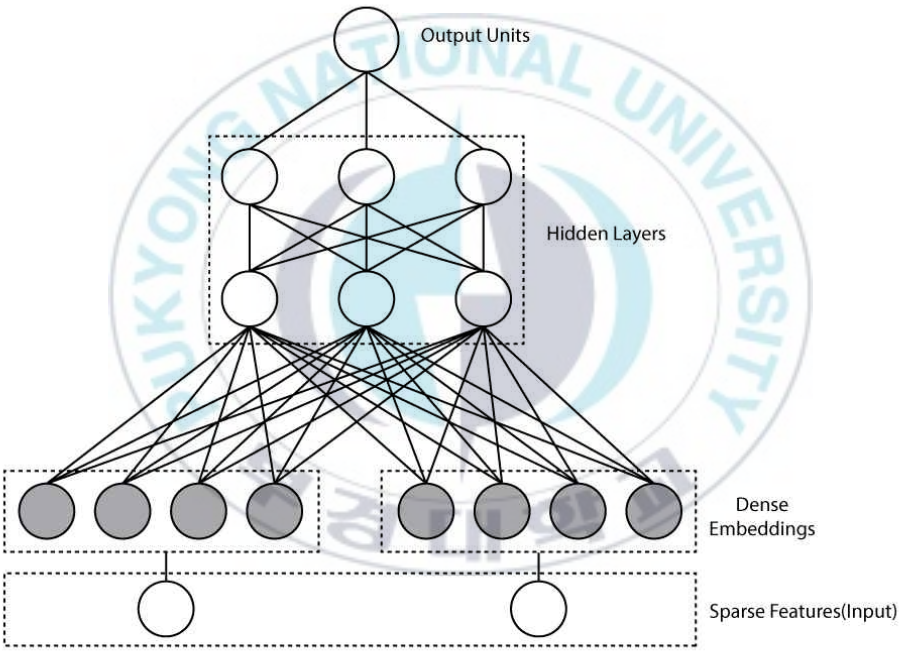

그림 5. 딥 모델

딥 모델은 임베딩된 카테고리 특징의 결과를 입력 데이터로 사용한다. 다음 식과 같은 계산을 수행한다.

$$a^{(l+1)} = f(W^{(l)}a^{(l)} + b^{(l)}) \quad (2)$$

여기서 l은 레이어 수를 의미하며 f는 활성화 함수로 일반적으로 ReLU를 사용한다. $W^{(l)}$, $a^{(l)}$, $b^{(l)}$는 l번째 층의 모델 가중치, 활성화 함수와 편향을 나타낸다.

3.1.3 와이드 앤 딥 모델의 결합

와이드 모델과 딥 모델의 결합 학습(joint training)은 그림 6과 같다. 와이드 앤 딥 모델은 와이드 모델과 딥 모델을 동시에 학습하고 각 모델의 약점을 보완하여 성능이 좋다. 결합 학습은 와이드 모델과 딥 모델을 각각 학습하면서 동시에 최적화한다.

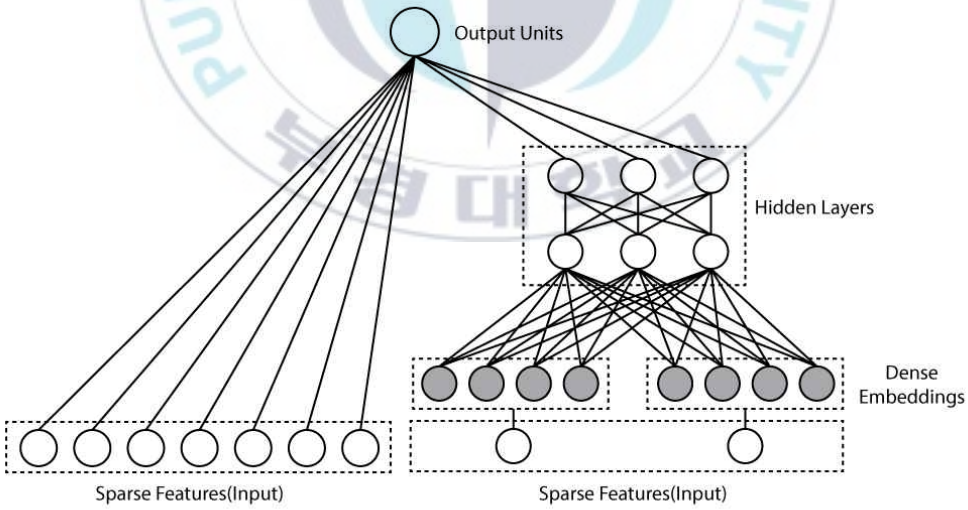

그림 6. 와이드 앤 딥 모델

각 모델에서 나온 결과를 다음과 같은 계산을 통해 최종 출력값이 나오게 된다.

$$P(Y=1|x) = \sigma(w_{wide}^{T}[x, \varnothing(x)] + w_{deep}^{T} a^{(l_f)} + b) \quad (3)$$

여기서 Y는 이진 클래스 라벨이고 $\sigma(\bullet)$는 시그모이드 함수이고 $\varnothing(x)$는 초기 x의 변환 값, b는 최종 활성화의 가중치를 나타내는 편향이다. w_{wide}는 와이드 모델 가중치 벡터이고 w_{deep}는 $a^{(l_f)}$에 적용된 가중치이다.

3.2 마이크로소프트 추천기

마이크로소프트 추천기 리포지터리는 추천 시스템의 설계, 평가 및 배포 프로세스를 효율적이고 생산적으로 구현할 수 있도록 도와주는 파이썬(Python) 유틸리티이며, 주피터 노트북(Jupyter Notebook)의 오픈 소스 컬렉션이다[48]. 마이크로소프트의 데이터 과학자들이 다양한 분야의 추천 시스템 작업 개발을 위해 적절한 사용 예시를 통합하기 위해 구현하였다. 마이크로소프트 추천기의 목표는 각 분야에서 주어진 데이터에 대한 최적의 추천 시스템을 효과적으로 구축하고 비교하여 최종 모델을 찾아 배포하는 것이다.

마이크로소프트 추천기 리포지터리는 새로운 추천 시스템 알고리즘과 코드 예제가 계속해서 업데이트되고 있으며, 계속해서 개선 및 확장되고 있어 최신의 추천 모델 적용이 가능하며 신뢰도와 안정성이 높다. 다양한 추

천 알고리즘을 제공하고 있어 하나의 알고리즘을 선택하여 모델을 훈련할 수 있다. 제공되고 있는 추천 시스템 알고리즘은 표 3과 같다.

표 3. 마이크로소프트 추천기 알고리즘 리스트

Algorithm	Type
LightGBM	콘텐츠 기반 필터링
Deep Knowledge Aware Network	협업 필터링
Neural Collaborative Filtering	협업 필터링
Restricted Boltzman Machines	협업 필터링
Surprise(Non-negative Matrix Factorization)	협업 필터링
Extreme Deep Factorization Machine	하이브리드 필터링
Vowpal Wabbit	하이브리드 필터링
Wide and Deep	하이브리드 필터링

마이크로소프트 추천기의 프로세스는 다음과 같다. 우선 개발 환경을 설정한다. 아나콘다(Anaconda) 또는 도커(Docker)를 사용하여 개발을 위해 원하는 특정 환경을 설정하여 관리할 수 있다. 데이터 세트 유틸리티는 데이터를 가져오고 분할하는 기능을 제공한다.

사용자는 훈련에 사용할 매개변수를 설정하고 모델의 성능을 평가할 수 있다. 추천 시스템 알고리즘을 선택한 후 매개변수를 조정하고 성능을 평가하는 과정을 여러 번 반복하여 성능을 비교한 다음 추천하고자 하는 데이터 세트와 해당 분야에 가장 적합한 알고리즘을 찾을 수 있다. 실험을 통해 최적의 모델을 찾았다면 최종 선택한 모델을 배포할 수도 있다.

Ⅳ. 실험 및 결과

4.1 실험 환경

본 논문에서 수행하는 실험 환경은 표 4와 같다. 구글 코랩 프로(Colab pro)를 사용하여 실험을 진행하였다. 코랩은 웹 브라우저에서 파이썬 코드를 작성하고 실행할 수 있는 클라우드 기반의 주피터 노트북 개발 환경이다.

표 4. 실험 환경

Experiment Environment	
OS	Windows 10 64bit
Language	Python 3.7
Development Environment	Colab pro
Browser	Google Chrome

4.2 실험 데이터

GoodBooks-10k 도서 데이터를 사용하여 딥러닝 추천 시스템 모델 중 하나인 와이드 앤 딥 모델을 적용하여 연구하였다[49]. GoodBooks-10k 도서 데이터는 Goodreads 웹사이트에서 제공하는 도서 데이터를 크롤링하여 모아놓은 데이터 세트로 도서, 평점(rating), 도서 태그(tag) 등 도서에 대한 정보를 그림 7과 같이 제공하고 있다.

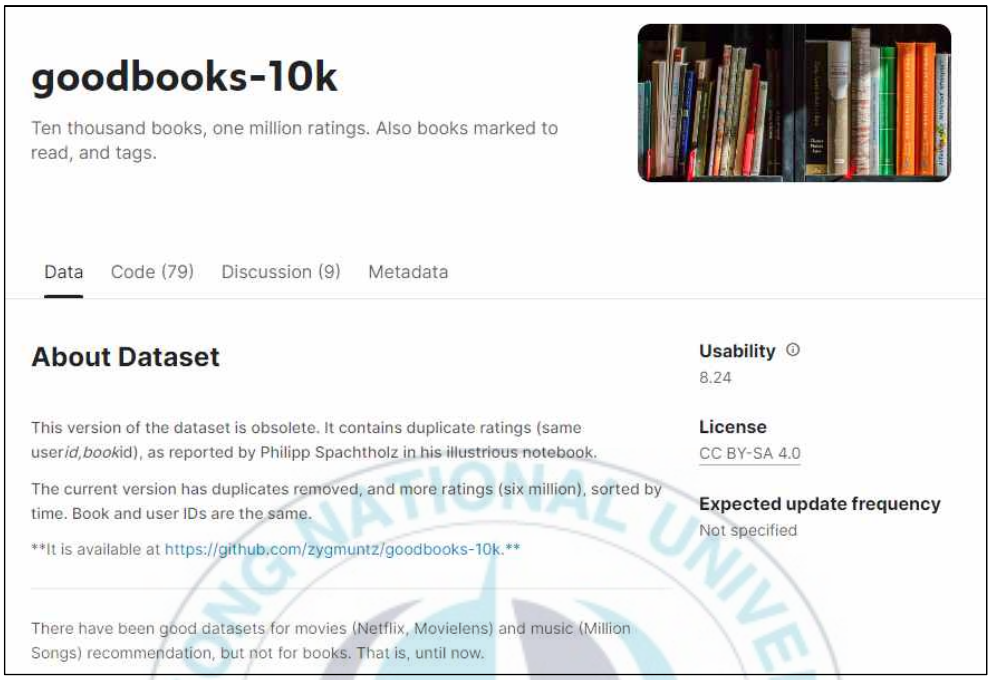

그림 7. GoodBooks-10k 데이터셋

도서 파일은 책의 메타정보를 가지고 있는 데이터 파일이다. 도서 태그 파일은 책과 태그의 정보가 담긴 파일이다. 평점 파일은 사용자가 책에 대해 점수를 준 평점 정보가 담긴 파일이다. 태그 파일은 태그의 정보가 담긴 파일이다. 실험에서 이 파일 중에서 도서 파일과 평점 파일을 사용하였다. 도서 아이디, 사용자 아이디, 평점, 제목 변수를 추출하여 사용하였다. 추가로 장르 변수를 입력 데이터로 사용할 예정이었으나 기존의 데이터 세트에는 없어서 확장된 데이터 세트에서 가져와 병합하여 사용하였다.

표 5. GoodBooks-10k 데이터셋

데이터셋	사용자	아이템	평점	장르
GoodBooks-10k	1-53,424	1-10,000	1-5	1-39

GoodBooks-10k 데이터셋은 표 5와 같이 각 도서에 사용자가 부여한 1부터 5까지의 평점 정보를 제공하고 있다. 장르는 39개의 장르로 구별되어 있다. 장르 데이터는 하나의 도서에 여러 장르로 구성된 다중 레이블 형식이므로 입력 데이터로 사용하기 위해 사이킷-런(scikit-learn)의 다중 라벨 이진화(multi label binarizer)를 사용하여 변환하였다. 예를 들어, 해당 도서의 장르가 판타지와 픽션이라면 [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, ...]와 같이 변환되게 된다. 추출한 데이터셋은 그림 8과 같고, 장르 데이터를 변환한 데이터셋은 그림 9와 같다.

	book_id	user_id	rating	title	genres
0	1	314	5	The Hunger Games (The Hunger Games, #1)	[young-adult, fiction, fantasy, science-fictio...
1	1	439	3	The Hunger Games (The Hunger Games, #1)	[young-adult, fiction, fantasy, science-fictio...
2	1	588	5	The Hunger Games (The Hunger Games, #1)	[young-adult, fiction, fantasy, science-fictio...
3	1	1169	4	The Hunger Games (The Hunger Games, #1)	[young-adult, fiction, fantasy, science-fictio...
4	1	1185	4	The Hunger Games (The Hunger Games, #1)	[young-adult, fiction, fantasy, science-fictio...

그림 8. 추출한 GoodBooks-10k 데이터셋

	book_id	user_id	rating	title	genres
0	1	314	5	The Hunger Games (The Hunger Games, #1)	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, ...
1	1	439	3	The Hunger Games (The Hunger Games, #1)	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, ...
2	1	588	5	The Hunger Games (The Hunger Games, #1)	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, ...
3	1	1169	4	The Hunger Games (The Hunger Games, #1)	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, ...
4	1	1185	4	The Hunger Games (The Hunger Games, #1)	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, ...

그림 9. 장르 데이터를 변환한 GoodBooks-10k 데이터셋

본 실험에는 도서와 사용자의 평점 데이터와 도서 속성인 다중 라벨 이진화로 변환한 장르 데이터를 입력값으로 사용하여 실험을 진행하였다.

4.3 성능 평가 지표

마이크로소프트 추천기에서는 실험이 잘 진행되었는지 확인하기 위해서 다양한 성능 평가 지표를 제공하고 있다. 본 논문에서는 평균 절대 오차와 평균 제곱근 오차, NDCG@K(Normalized Discounted Cumulative Gain), Precision@K를 사용했다.

평균 절대 오차와 평균 제곱근 오차는 대표적인 평점 예측 기반 성능 지표로, 예측 점수와 실제 점수의 차이를 비교하는 것이다. 도서에 대한 평점 예측은 다음과 같은 과정으로 수행된다. 우선 도서 아이디와 해당 도서의 평점을 준 사용자 아이디를 추출한다. 추출한 데이터는 그림 10과 같다.

	user_id	book_id	genres
0	46111	4887	[0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, ...
1	37322	1707	[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, ...
2	12914	1855	[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
3	19045	2395	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, ...
4	41714	2404	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, ...
5	26668	5101	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
6	22171	386	[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, ...
7	49739	6798	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, ...
8	17394	2550	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, ...
9	3893	5071	[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, ...

그림 10. 추출한 사용자 아이디, 도서 아이디와 장르 데이터

추출한 데이터에서 도서의 평점 예측이 이루어진다. 평점 예측값은 그림 11과 같이 나오게 된다. 해당 도서의 실제 평점과 예측 평점의 유사도를 구한다.

	user_id	book_id	genres	prediction
0	46111	4887	[0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, ...	3.540402
1	37322	1707	[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, ...	4.083021
2	12914	1855	[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	4.080554
3	19045	2395	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, ...	4.293251
4	41714	2404	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, ...	3.699023
5	26668	5101	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	3.756789
6	22171	386	[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, ...	3.776874
7	49739	6798	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, ...	3.673519
8	17394	2550	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, ...	3.752723
9	3893	5071	[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, ...	3.796551

그림 11. 추출한 데이터의 평점 예측

평균 절대 오차는 예측값과 실제 값의 차이를 나타낸다. 식은 다음과 같다[50].

$$MAE = \frac{\sum |p_{i,j} - r_{i,j}|}{N} \quad (4)$$

N은 전체 평가 데이터 수이며 $p_{i,j}$는 예측 점수이고 $r_{i,j}$는 실제 점수이다. 예측 점수와 실제 점수의 차이에 절댓값을 한 뒤, 이들의 합을 전체 데이터 수로 나눈 값이다.

평균 제곱근 오차는 오차의 제곱을 평균으로 나눈 평균 제곱 오차(Mean Squared Error, MSE)에 루트를 한 것이다. 각 성능 평가 지표의 식은 아

래와 같다[50].

$$MSE = \frac{1}{N}\sum(p_{i,j} - r_{i,j})^2 \tag{5}$$

$$RMSE = \sqrt{\frac{1}{N}\sum(p_{i,j} - r_{i,j})^2} \tag{6}$$

여기서, $p_{i,j}$는 예측 점수이고 $r_{i,j}$는 실제 점수이며, N은 총 데이터의 개수이다. 평균 제곱 오차는 오차에 제곱하여 오차가 큰 값에 가중치를 부여하는 평가 방법이다. 평균 제곱근 오차는 평균 제곱 오차 값에 제곱근을 하여 예측 오차가 큰 관측치에 상대적으로 적은 가중치를 부여한다.

NDCG@K와 Precision@K는 랭킹 기반 성능 평가 지표로, 여기서 K는 추천 아이템의 수를 의미한다. 정확도(precision)는 예측한 평점과 실제 평점이 일치한 비율을 의미하며 높을수록 추천 정확도가 높다. 정확도의 식은 다음과 같다.

$$Precision = \frac{a}{a+c} \tag{7}$$

a는 예측 평점이 실제 평점과 일치한 개수이고, $a+c$은 예측한 평점의 개수이다. 모두 일치할 경우 1이 되고, 점수가 높을수록 추천 성능이 좋다.

Precision@K는 사용자가 관심 있을 것으로 예상하여 추천한 아이템 K개 중에 실제 사용자가 관심 있는 아이템의 비율을 계산한 것으로 1에 가까울수록 좋다. 예를 들어, 추천한 아이템이 10개이고 추천한 아이템 중 사

용자가 선호한 아이템이 5개이면 Precision@10 = 0.5가 된다.

NDCG@K는 K개의 아이템을 추천하는 경우, 추천 순서에 가중치를 두어 평가한다. 값은 1에 가까울수록 좋다. NDCG@K는 추천 순서별로 관련도 값(relevance)을 다르게 적용하여 계산한다.

4.4 실험 결과 및 고찰

와이드 앤 딥 모델 성능을 비교하기 위해 와이드 모델과 딥 모델 그리고 와이드 앤 딥 모델 각각에서 실험을 진행하였다. 스텝(step) 수를 다르게 하여 실험을 진행하였다. 표 6 및 7은 각각 스텝 수를 4,000번, 20,000번으로 훈련을 진행한 결과이다.

표 6. 4,000번 훈련한 결과

MODEL	RMSE	MAE	NDCG@K	Precision@K
와이드 모델	1.0567	0.8348	0.0013	0.0663
딥 모델	0.9965	0.7733	0.0	0.0
와이드 앤 딥 모델	1.0000	0.7777	0.0	0.0

표 7. 20,000번 훈련한 결과

MODEL	RMSE	MAE	NDCG@K	Precision@K
와이드 모델	0.9893	0.7977	0.0014	0.0004
딥 모델	0.9807	0.8077	0.0021	0.0004
와이드 앤 딥 모델	1.0025	0.8379	0.0	0.0

와이드 모델과 딥 모델에서 4,000번 훈련한 결과보다 20,000번 훈련한 결과가 더 좋은 성능을 보였지만, 와이드 앤 딥 모델의 경우 4,000번 훈련한 결과가 성능이 더 좋게 나왔다.

와이드 앤 딥 모델 성능 평가를 그래프로 확인해보면 그림 12 및 13과 같다. 그 이후로는 성능 차이가 크지 않는 것을 보아 4,000번 수행이 적절한 것으로 추론할 수 있다.

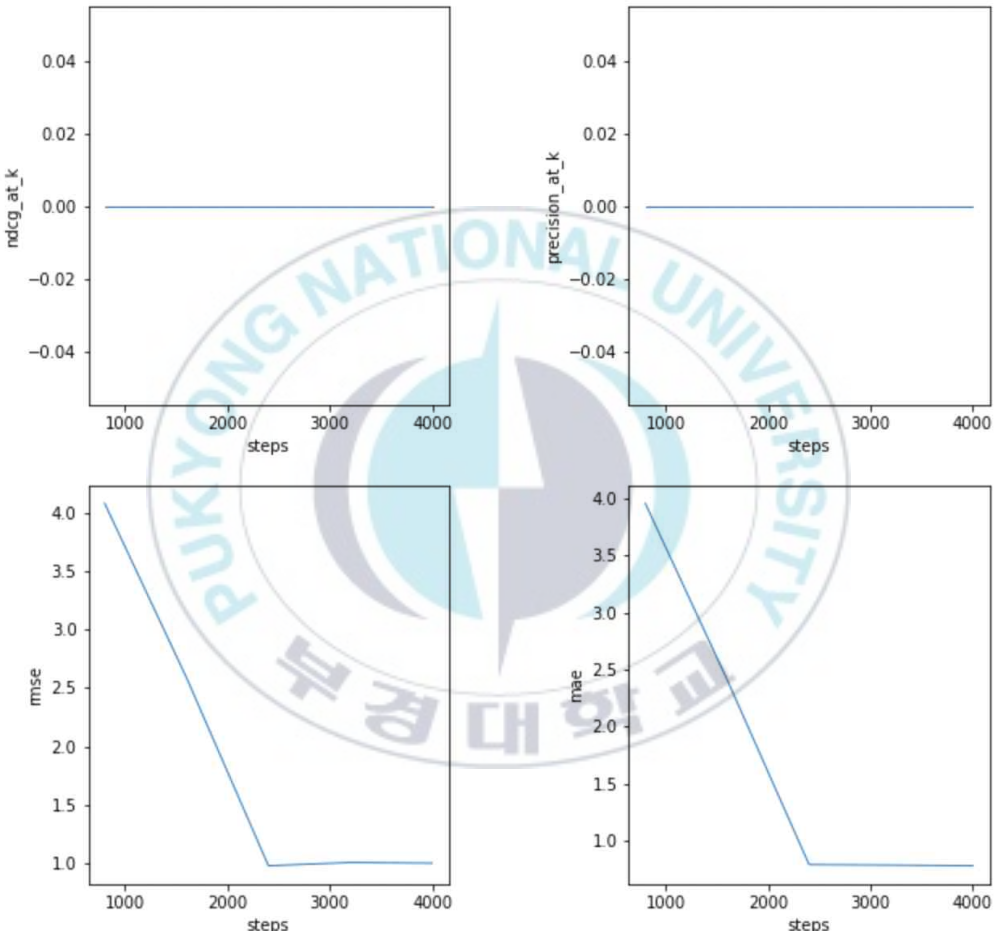

그림 12. 4,000번 훈련한 결과 그래프

PUKYONG NATIONAL UNIVERSITY 부경대학교

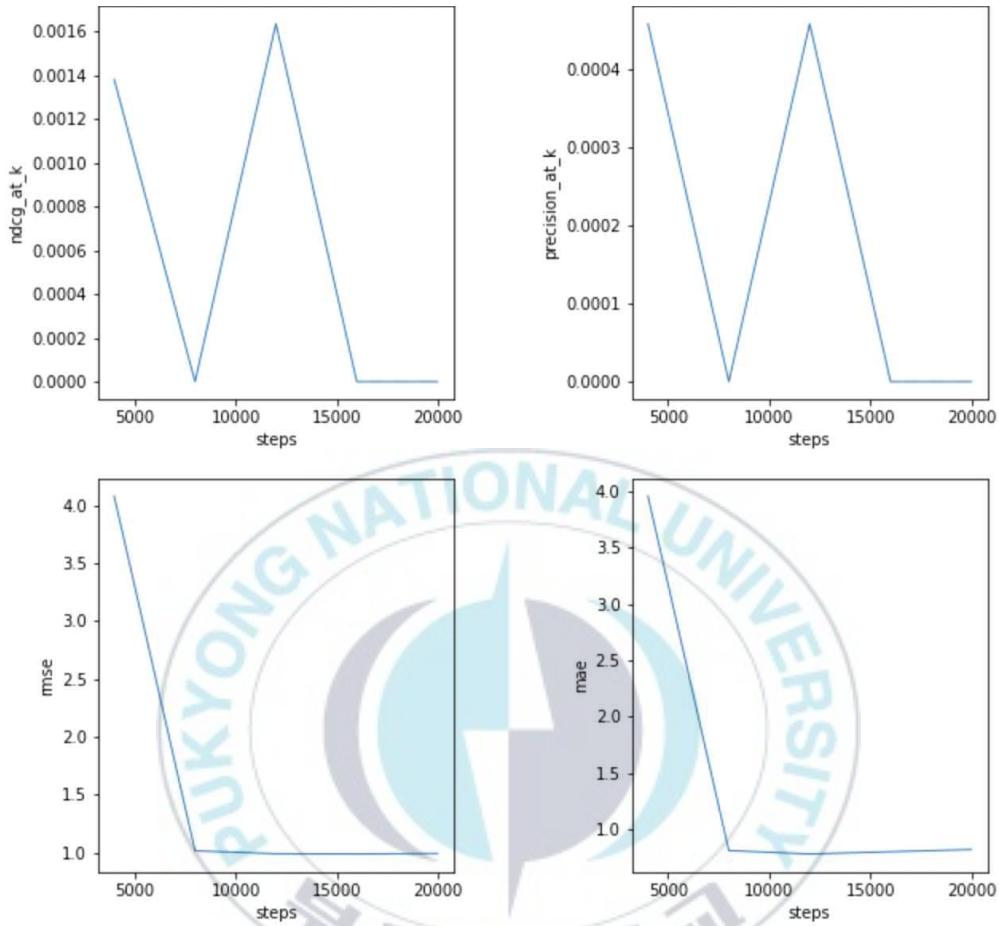

그림 13. 20,000번 훈련한 결과 그래프

다음으로는 와이드 앤 딥 모델에 은닉층의 노드 수와 학습률을 조절하여 실험을 진행하였다. 은닉층의 노드 수는 각각 지정하여 실험하였다. 결과는 표 8과 같다. 은닉층의 노드를 16, 32, 128로 지정했을 때 평균 절대 오차는 0.8110, 평균 제곱근 오차는 0.9948로 가장 좋은 성능을 보여주었다.

표 8. 은닉층의 노드 수 변화에 따른 결과

은닉층의 노드 수	MAE	RMSE
8|16|64	0.8248	1.0055
16|32|128	0.8110	0.9948
32|64|256	0.8889	1.0517

가장 좋은 성능을 보여준 은닉층의 노드 수에서 학습률의 값을 조절하여 실험을 진행하였다. 실험 결과는 다음 표 9와 같다. 딥 모델에서 학습률 값을 0.1로 와이드 모델에서 학습률을 0.001로 지정하였을 때, 평균 절대 오차는 0.7967, 평균 제곱근 오차는 0.9874로 가장 좋은 성능을 보여주었다.

표 9. 학습률 변화에 따른 결과

딥 모델 학습률	와이드 모델 학습률	MAE	RMSE
0.1	0.1	0.8110	0.9948
0.1	0.01	0.8223	1.0268
0.1	0.001	0.7967	0.9874
0.01	0.1	0.8282	1.0083
0.001	0.1	0.8353	1.0157

딥 모델에서 학습률 값을 0.1로 와이드 모델에서 학습률을 0.001로 지정하였을 때의 결과 그래프는 그림 14와 같다. 성능 평가 점수가 현저하게 줄어드는 것을 확인할 수 있었다.

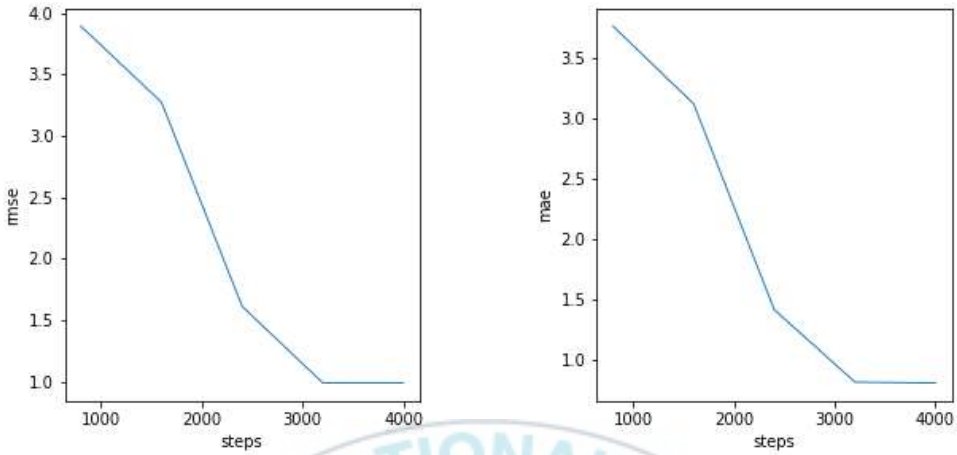

그림 14. 딥 모델과 와이드 모델에서 각각 학습률을 0.1과 0.001로 지정하였을 때의 결과 그래프

표 10은 와이드 앤 딥 모델과 협업필터링 기반의 음수 미포함 행렬 분해의 성능 평가를 비교한 것이다. 와이드 앤 딥 모델의 성능이 우수하다는 것을 확인할 수 있다.

표 10. 와이드 앤 딥 모델과 NMF의 성능 비교

MODEL	MAE	RMSE
와이드 앤 딥	0.7967	0.9874
NMF	0.8232	1.0564

Ⅴ. 결 론

인터넷의 발달과 스마트폰의 보급으로 인해 매일 다양한 정보들이 생성되는 빅데이터 시대가 되었으며 사용자 취향에 맞는 개인화된 추천은 중요한 과제로 대두되었다. 현재 많은 회사에서 추천 시스템을 적용하여 사용자의 취향을 예측하고 상품을 제공하고 있다.

추천 성능을 높이고자 딥러닝을 이용한 추천 시스템에 대한 연구가 활발히 이루어지고 있다. 하지만 딥러닝은 구조가 복잡하고 높은 기술력을 요구하여 구현이 어렵다는 단점이 있다. 와이드 앤 딥 모델은 기존의 딥러닝 기반의 도서 추천 시스템으로 연구되어 좋은 성능을 보여주었지만, 구조가 복잡하여 구현하는 데 많은 시간과 기술이 필요하다는 단점이 있어 개인이나 소규모 추천 시스템에 적용하기 어려웠다.

본 논문에서는 도서 데이터에 대한 효율적인 추천 시스템 구축을 위해 마이크로소프트 추천기를 사용한 딥러닝 추천 시스템 모델인 와이드 앤 딥 모델 기반의 도서 추천 시스템을 제안하였다. 마이크로소프트 추천기는 최적화된 추천 시스템 구축을 간편하게 할 수 있도록 오픈 소스를 제공하고 있어 이를 바탕으로 연구를 진행하였다. 기존의 추천 시스템 구축에 있어 드는 시간과 비용을 획기적으로 개선하였으며 안정적인 성능을 확인하였다.

GoodBooks-10k의 도서 데이터 중에서 평점과 장르 데이터를 인풋 데이터로 사용하여 도서 추천 시스템을 연구를 진행하였고, 비교를 위해 와이드 모델, 딥 모델과 와이드 앤 딥 모델에서 실험을 진행하였고, 평균 절대 오차, 평균 제곱근 오차, NDCG@K, Precision@K를 사용하여 성능을 평가하였다. 은닉층의 노드 수와 학습률을 조정하여 최적의 추천 시스템을 구축하였고, 마이크로소프트 추천기를 사용하여 적은 시간으로 안정적인 성

능의 추천 시스템이 실행되는 것을 확인하였다. 도서 추천 시스템 분야에 있어 본 논문에서 제시한 생산적인 딥러닝 기반의 도서 추천 시스템 구축에 도움이 될 것으로 판단된다.

향후에는 마이크로소프트 추천기에서 와이드 앤 딥 모델 이외에도 콘텐츠 기반의 필터링, 협업 필터링, 하이브리드 필터링 기반의 다양한 딥러닝 추천 시스템 모델을 적용하여 기존보다 성능이 뛰어난 딥러닝 기반의 도서 추천 시스템을 구축하고자 한다.

PUKYONG NATIONAL UNIVERSITY
부경대학교

참고문헌

[1] https://www.netflix.com/

[2] www.youtube.com

[3] www.facebook.com

[4] https://news.google.com/

[5] https://www.amazon.com/

[6] C. Chen, X. Meng, Z. Xu, and T. Lukasiewicz, "Location-Aware Personalized News Recommendation with Deep Semantic Analysis," *IEEE Access*, vol. 5, pp. 1624-1638, 2017.

[7] M. Schedl, "Deep Learning in Music Recommendation Systems," *Frontiers in Applied Mathematics and Statistics*, vol. 44, 2019.

[8] B. Yi, X. Shen, Z. Zhang, J. Shu, and H. Liu, "Expanded Autoencoder Recommendation Framework and Its Application in Movie Recommendation," *2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*, pp. 298-303, Dec. 2016.

[9] H. T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & Deep Learning for Recommender Systems," *Proc. of the 1st Workshop on Deep Learning for Recommender Systems*, pp. 7-10. 2016.

[10] Y. Ma, J. Jiang, S. Dong, C. Li, and X. Yan, "Book

Recommendation Model Based on Wide and Deep Model," *2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID)*, pp. 247-254, 2021.

[11] https://www.tensorflow.org/

[12] A. Anandaraj, P. Yeshwanth Ram, K. Sri Ram Kumar, M. Revanth, and R. Praveen, "Book Recommendation System with TensorFlow," *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS),* pp. 1665-1669, 2021.

[13] https://github.com/microsoft/recommenders

[14] Van den Oord, Aaron, Sander Dieleman, and Benjamin Schrauwen. "Deep Content-Based Music Recommendation," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[15] J. Lee, S. Abu-El-Haija, B. Varadarajan, and A. Natsev, "Collaborative Deep Metric Learning for Video Understanding," *Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* pp. 481-490, 2018.

[16] C. Lei, D. Liu, W. Li, Z. -J. Zha, and H. Li, "Comparative Deep Learning of Hybrid Representations for Image Recommendations," *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 2545-2553, 2016.

[17] W. T. Chu and Y. L. Tsai, "A Hybrid Recommendation System Considering Visual Information for Predicting Favorite Restaurants," *World Wide Web*, vol. 20, pp. 1313-1331, 2017.

[18] R. He and J. McAuley, "Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering," *Proc. of the 25th International Conference on World*

Wide Web, pp. 507-517, April 2016.

[19] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," *In The Adaptive Web*, pp. 325-341, 2007.

[20] Charu C. Aggarwal, *Recommender Systems,* vol. 1. Cham: Springer International Publishing, 2016.

[21] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, pp. 331-370, 2002.

[22] Wikipedia, "인공지능," https://ko.wikipedia.org/wiki/인공지능, Retrieved Oct 2022.

[23] 오일석, *기계 학습*, 한빛아카데미, 2017

[24] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep Learning Based Recommender System: A Survey and New Perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, Issue 1, pp. 1-38, 2019.

[25] O. Kuchaiev and B. Ginsburg, "Training Deep Auto Encoders for Collaborative Filtering," *arXiv:1708.01715*, 2017.

[26] Y. Pan, F. He, and H. Yu, "Learning Social Representations with Deep Autoencoder for Recommender System," *World Wide Web*, vol. 23, pp. 2259 - 2279, 2020.

[27] S. Li, J. Kawale, and Y. Fu, "Deep Collaborative Filtering via Marginalized Denoising Auto-encoder," *Proc. of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 811-820, Oct. 2015.

[28] X. Li and J. She, "Collaborative Variational Autoencoder for Recommender Systems," *Proc. of the 23rd ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining, pp. 305-314, Aug. 2017.

[29] W. Yan, D. Wang, M. Cao, and J. Liu, "Deep Auto Encoder Model With Convolutional Text Networks for Video Recommendation," *in IEEE Access*, vol. 7, pp. 40333-40346, 2019.

[30] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural Collaborative Filtering," *Proc. of the 26th International Conference on World Wide Web*, pp. 173-182, April 2017.

[31] Q. Sun, L. L. Shi, L. Liu, Z. X. Han, L. Jiang, Y. Lu, and J. Panneerselvam, "A Dynamic Collaborative Filtering Algorithm based on Convolutional Neural Networks and Multi-layer Perceptron," *2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)*, pp. 459-466, Dec. 2021.

[32] S. Fan and M. Fu, "Music Genre Recommendation Based on MLP & Random Forest," *2022 IEEE 5th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, pp. 331-334, 2022.

[33] K. Patil and N. Jadhav, "Multi-Layer Perceptron Classifier and Paillier Encryption Scheme for Friend Recommendation System," *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pp. 1-5, 2017.

[34] H. W. J. Jayasanka, B. T. G. S. Kumara, and B. Kuhaneswaran, "CNN Based Costume Categorization Approach to Bootstrap the Costume Recommendations," *2021 International Conference on*

Decision Aid Sciences and Application (DASA), pp. 741-744, 2021.

[35] W. Wang, C. Ye, P. Yang, and Z. Miao, "Research on Movie Recommendation Model Based on LSTM and CNN," *2020 5th International Conference on Computational Intelligence and Applications (ICCIA)*, pp. 28-32, 2020.

[36] S. Joshi, T. Jain, and N. Nair, "Emotion Based Music Recommendation System Using LSTM-CNN Architecture," *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 01-06, 2021.

[37] C. Yan and Y. Shi, "A Personalized Location Recommendation based on Convolutional Neural Network," *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pp. 1516-1519, 2020.

[38] F. Ç. Sarıoğlu and Y. Yaslan, "Item Prediction with RNN Using Different Types of User-Item Interactions," *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pp. 1-4, 2019.

[39] Z. Qu, S. Zheng, X. Wang, X. Song, B. Li, and X. Song, "Converged Recommendation System Based on RNN and BP Neural Networks," *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 308-313, 2018.

[40] M. Jiang, Z. Yang, and C. Zhao, "What To Play Next? A RNN-Based Music Recommendation System," *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 356-358, 2017.

[41] S. Singh, R. Rajan, S. Nandini, D. Ramesh, and C. P. Prathibhamol,

"Friend Recommendation System in a Social Network based on Link Prediction Framework using Deep Neural Network," *2022 2nd International Conference on Intelligent Technologies (CONIT)*, pp. 1-7, 2022.

[42] J. Gong J. Gong, Y. Zhao, S. Chen, H. Wang, L. Du, S. Wang, M. Z. A. Bhuiyan, H. Peng, and B. Du, "Hybrid Deep Neural Networks for Friend Recommendations in Edge Computing Environment," *in IEEE Access*, vol. 8, pp. 10693-10706, 2020.

[43] Z. Huang, C. Yu, J. Ni, H. Liu, C. Zeng, and Y. Tang, "An Efficient Hybrid Recommendation Model With Deep Neural Networks," *in IEEE Access*, vol. 7, pp. 137900-137912, 2019.

[44] M. R. I. Sarker and A. Matin, "A Hybrid Collaborative Recommendation System Based On Matrix Factorization And Deep Neural Network," *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, pp. 371-374, 2021.

[45] W. Xiong and C. He, "Personalized Movie Hybrid Recommendation Model Based on GRU," *2021 4th International Conference on Robotics, Control and Automation Engineering (RCAE)*, pp. 161-164, 2021.

[46] Y. H. Jun, S. J. Lim, and K. R. Kwon, "Design of Book Recommendation System based on Deep Learning using BX-Books Data Set," *Proc. of Korea Multimedia Society, Spring Conference*, vol. 25, no. 1, pp. 51-52, May 2022.

[47] Y. H. Jun, S. J. Lim, and K. R. Kwon, "User-Customized Book

Recommendation System using Wide and Deep Model," *Proc. of Korea Multimedia Society, Autumn Conference*, vol. 25, no. 2, pp. 12-14, Nov. 2022.

[48] S. Graham, J.-K. M, and T. Wu, "Microsoft Recommenders: Tools to Accelerate Developing Recommender Systems," *Proc. of the 13th ACM Conference on Recommender Systems*, 2019.

[49] https://www.kaggle.com/zygmunt/goodbooks-10k

[50] J. Son, S. B. Kim, H. Kim, and S. Cho, "Review and Analysis of Recommender Systems," *Journal of Korean institute of industrial engineers*, pp. 41, no. 2, pp. 185-208, 2015.

PUKYONG NATIONAL UNIVERSITY 부경대학교