

Recommendation system using Deep Autoencoder for Tensor data

Jina Park*, Hwan-Seung Yong**

Abstract

These days, as interest in the recommendation system with deep learning is increasing, a number of related studies to develop a performance for collaborative filtering through autoencoder, a state-of-the-art deep learning neural network architecture has advanced considerably. The purpose of this study is to propose autoencoder which is used by the recommendation system to predict ratings, and we added more hidden layers to the original architecture of autoencoder so that we implemented deep autoencoder with 3 to 5 hidden layers for much deeper architecture. In this paper, therefore we make a comparison between the performance of them. In this research, we use 2-dimensional arrays and 3-dimensional tensor as the input dataset. As a result, we found a correlation between matrix entry of the 3-dimensional dataset such as item-time and user-time and also figured out that deep autoencoder with extra hidden layers generalized even better performance than autoencoder.

▶ Keyword: Deep Autoencoder, Recommendation system, Collaborative filtering, Tensor data

I. Introduction

사용자의 과거 구매 이력 등을 기반으로 개인의 취향을 분석해 상품이나 콘텐츠를 추천하는 추천 시스템 방법은 다양한 분야에서 사용되고 있다[1]. 추천 시스템을 활용하면 사용자들의 만족도를 높일 수 있을 뿐만 아니라 상품이나 콘텐츠를 제공하는 기업도 이익을 얻을 수 있기에 추천 시스템을 잘 구축하여 추천 성능을 높이는 것이 중요하다.

추천 시스템 방법은 크게 두 가지로 내용 기반 추천 방법(context-based) 방법과 개인화(personalized) 추천 방법으로 나눌 수 있다[2]. 시간이나 장소와 같은 요소들을 통해 상품을 추천하는 내용 기반 추천 방법과 달리 개인화 방법은 협업 필터링(collaborative filtering) 방법을 통해 사용자들에게 상품이나 콘텐츠를 추천한다. 이 방법을 통해 상품에 대한 사용자의 관심사를 예측할 수 있는데, 그 사용자와 비슷한 관심사를 가진 다른 사용자들의 선호도 등을 고려하여 사용자에게 상품을 추천한다. 이 방법에서는 취향이 비슷한 사용자 그룹 내에서는 상품에 관한 의견이 비슷할 것이라는 이론을 가정하고 추천한다.

최근 딥 러닝(deep learning) 기술이 발전하고 딥 러닝에 대한 관심이 증가하며 추천 성능을 향상시키기 위해 협업 필터링 방법과 접목한 연구가 수행되고 있다. 본 논문에서는 개인화 추천 방법을 바탕으로 텐서플로우 GPU 환경에서 협업 필터링 기반의 딥 러닝 모델인 오토 인코더와 은닉 레이어를 3개 이상 추가한 딥 오토 인코더를 구현하여 추천 시스템을 제안한다. 또한 2차원 행렬에 시간 항목이 추가된 3차원 텐서 데이터를 이용하였다. 따라서 사용자와 아이템 기반으로 사용자의 평점 예측뿐만 아니라 시간과 각 항목 간의 연관성을 분석하는 실험을 진행하였다.

결과적으로 본 연구에서 구현한 모델인 딥 오토 인코더와 기존에 수행되었던 추천 시스템 방법인 행렬 분해(matrix factorization)[3] 방법과의 실험 결과에서 오차율을 비교하여 추천 시스템의 성능에 영향을 줄 수 있음을 확인하고 3차원 텐서 데이터를 이용한 실험을 통해 데이터 항목 간의 관계를 비교하고 분석하였다.

• First Author: Jina Park, Corresponding Author: Hwan-Seung Yong

*Jina Park (gina9647@ewhain.net), Dept. of Computer Science & Engineering, Graduate School, Ewha Womans University

**Hwan-Seung Yong (hsyong@ewha.ac.kr), Dept. of Computer Science & Engineering, Ewha Womans University

• Received: 2019. 06. 18, Revised: 2019. 08. 06, Accepted: 2019. 08. 06.

• This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2016R1D1A1B03931529)

논문의 구성은 다음과 같다. 2장에서는 추천 시스템에서의 딥 러닝과 딥 러닝을 이용한 데이터 분석에 대한 관련 연구와 동향에 대하여 알아본다. 3장에서 본 연구를 통해 구현한 딥 러닝 모델인 오토 인코더와 딥 오토 인코더에 대하여 설명하고 실험 결과에 대하여 논의한 후 4장에서 논문을 마무리 짓는다.

II. Preliminaries

1. Related works

이 장에서는 딥 러닝을 이용한 추천 시스템 관련 연구에 대하여 기술하고 빅 데이터와 고차원 데이터 등과 같이 크기가 크고 복잡한 데이터를 딥 러닝을 사용해 분석한 사례에 관하여 서술한다.

딥 러닝은 현재 강화 학습과 패턴 인식, 이미지 인식 등의 연구 분야에서 활발하게 사용되고 있다[4]. 딥 러닝 기술을 활용한 추천 시스템 방법에는 RBM(Restricted Boltzmann Machine) 방법이 있는데 RBM 구조를 통해 영화 평점 데이터인 넷플릭스 데이터를 추천 시스템에 적용하여 기존 연구인 SVD 방법과 비교해 더 높은 성능의 결과를 보여주었다[5]. 추천 시스템이 전자 상거래 서비스에서 중요한 역할을 하고 있는 만큼 딥 러닝을 적용하여 효과적인 추천 시스템을 구현하기 위한 연구가 진행되고 있다. 넷플릭스나 온라인 사이트인 아마존 등에서도 추천 시스템을 활용하여 소비자들에게 영화와 상품을 추천하고 있다. 특히 넷플릭스의 경우, 추천 알고리즘 개선 대회(Netflix Prize Contest)[6]를 개최하여 영화 추천 성능을 높이고자 하였는데, 넷플릭스 내에서 시청된 영화의 약 80%가 추천 시스템에 의해서 시청될 만큼 추천 시스템을 밀접하게 활용하고 있다[7]. 동영상 스트리밍 사이트인 유튜브에서도 사용자들의 과거 동영상 시청 기록을 바탕으로 심층 신경망(deep neural network)을 통해 동영상 추천 시스템에 적용하였다. 매일 생성되는 많은 양의 데이터를 정교한 추천 시스템 연구에 활용하기 위해 심층 신경망을 사용하였으며 기존의 추천 시스템 연구에서 사용했던 방법인 행렬 분해[3] 방법과 비교했을 때, 보다 더 많은 양의 데이터를 연구에 수행하는 데 효과적이었다고 설명하였다[8].

현재 기계 학습, 영상 및 음성 인식, 자연어 처리 등의 딥 러닝 연구 분야에서 빅 데이터(big data)와 고차원 데이터 등 사용되는 데이터의 양이 많아지고 크기가 커지는 만큼 이 데이터를 분석하여 의미를 추출하는 데 딥 러닝의 역할이 중요해지고 있으며 데이터를 효율적으로 다루는 방법에 대한 선행 연구가 수행되고 있다. 또한 오늘날 기계 학습을 통한 연구 등에서 데이터를 학습하여 연구에 활용하는 부분이 중요한 만큼 기존의 데이터 학습 방법들과 달리 딥 러닝을 통한 지도학습(supervised learning) 과 비지도 학습(unsupervised learning) 등을 활용하며 보다 효율적인 학습이 가능하다. 구

글, 페이스북, 마이크로소프트 등과 같은 기업들 역시 사용자에 의해 매일 생성되는 거대한 양의 데이터를 수집하여 딥 러닝 모델을 활용해 분석하는 연구를 진행하고 있다. 특히 구글의 경우, 웹 사이트 내 거리 뷰, 이미지 검색, 번역기 등에서 딥 러닝을 통해 복잡한 데이터를 처리하여 분석한다[9].

본 연구에서는 기존 딥 러닝 추천 시스템 연구에서 사용되었던 2차원 행렬 데이터뿐만 아니라 3차원 텐서 데이터를 이용한 실험을 통해 추천 시스템을 구현하고 각 데이터 항목 간의 연관성을 분석하여 제안하고자 한다.

III. The Proposed Scheme

본 장에서는 3차원 텐서 데이터를 이용한 추천 시스템을 위한 딥 오토 인코더의 구현에 대하여 설명하고 시스템 환경과 데이터, 실험 결과에 대하여 서술한다. 오토 인코더는 협업 필터링 분야의 심층 신경망 구조의 하나로서, 본 연구에서 구현한 오토 인코더의 기본 구조는 Fig. 1과 같다.

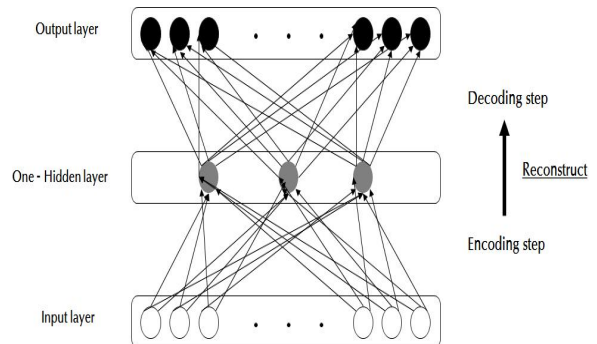


Fig. 1. Architecture of Autoencoder

Fig. 1에서 오토 인코더의 기본 구조는 입력 레이어, 하나의 은닉 레이어, 출력 레이어로 구성된다. 입력 레이어와 출력 레이어의 뉴런의 수는 동일해야 하며 이러한 구조로 인해 라벨이 있는 정답 데이터 없이 입력 데이터만으로 비지도 학습이 가능하다. 또한 입력 레이어의 뉴런의 수보다 은닉 레이어의 뉴런 수를 작게 해 입력 레이어의 데이터를 압축하여 데이터의 특징을 분석할 수 있도록 한다. 이 과정은 입력 레이어에서 은닉 레이어로 가는 과정인 인코딩 단계와, 은닉 레이어에서 출력 레이어로 가는 과정인 디코딩 단계로 나눌 수 있다. 딥 오토 인코더는 하나의 은닉 레이어로 구성된 오토 인코더에 은닉 레이어를 추가하여 쌓은 구조이며 딥 오토 인코더를 구현할 때는 과적합 문제의 발생을 피하기 위해 데이터에 맞게 은닉 레이어의 개수와 뉴런의 수를 조절하여야 한다.

Fig. 2와 Fig. 3은 오토 인코더 실험에서 각각 2차원 행렬과 3차원 무비렌즈(MovieLens) 데이터를 사용하기 위한 오토 인코더의 구조를 표현한 것이다. 오토 인코더 실험은 아이템 기반

(item-based) 방법과 사용자 기반(user-based) 방법으로 구현하였는데, Fig. 2와 Fig. 3은 아이템 기반의 구현 모델이기에 입력 레이어와 출력 레이어의 뉴런의 수는 각각의 데이터 내 아이템 수와 동일하게 구성하였다.

구현한 오토 인코더와 딥 오토 인코더 모델을 통해 출력 레이어를 거쳐 재구성된 예측 벡터와 실제 입력 데이터 벡터의 오차율인 RMSE(Root Mean Squared Error)를 구하여 성능을 평가한다. 모델의 가중치(Weight)는 평균 0.0, 표준편차는 0.07 표준 정규 분포로 가중치를 초기화하였다. 편파성(Bias)은 0으로 초기화하였다. 실험 결과에서 오차율이 낮을수록 예측 벡터(prediction vector)와 실제 입력 벡터의 차이가 작을 것이므로 추천 가능성이 높을 것으로 판단하였다.

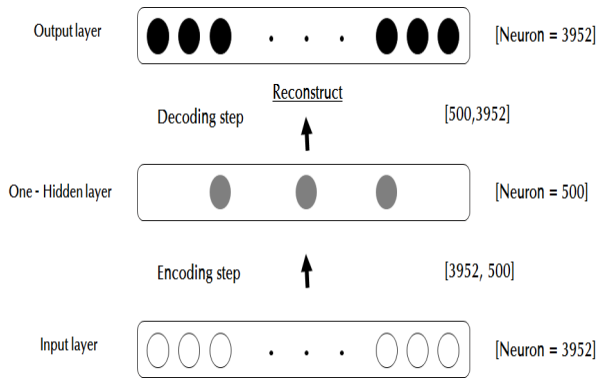


Fig. 2. Autoencoder used in 2-dimensional Movielens data experiment

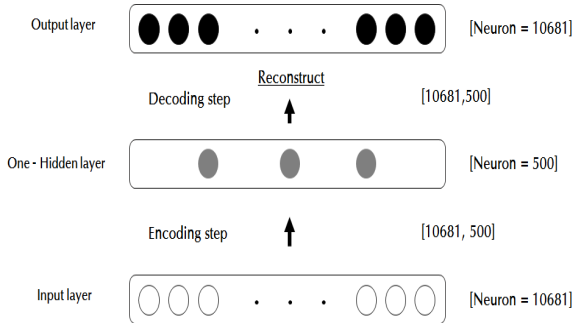


Fig. 3. Autoencoder used in 3-dimensional Movielens tensor data experiment

다음으로는 딥 오토 인코더 구현 과정에 대하여 살펴본다. 오토 인코더와 마찬가지로 아이템 기반(item-based) 방법과 사용자 기반(user-based) 방법으로 2차원 행렬 데이터와 3차원 텐서 데이터를 이용해 딥 오토 인코더 모델을 구현하였다. Fig. 4는 아이템 기반 방법의 2차원 무비렌즈 데이터를 입력 데이터로 사용하였으며, 은닉 레이어가 3개인 딥 오토 인코더 모델이다.

Fig. 4는 Fig. 2와 마찬가지로 2차원 행렬 데이터에서 사용한 실험 모델로, 첫 번째와 세 번째 은닉 레이어의 뉴런의 개수는 1,000개이고 가운데의 은닉 레이어의 뉴런의 수는 500개로 대칭 형태로 이루어져 있다. Fig. 5는 3차원 텐서 데이터인 무

비렌즈 데이터를 통해 아이템 기반 방법으로 실험한 구조로 은닉 레이어가 5개인 딥 오토 인코더를 표현한 것이다. Fig. 5에서 입력 레이어와 출력 레이어의 뉴런의 개수는 3차원 무비렌즈 데이터의 아이템의 수와 동일하다. Fig. 2와 Fig. 4는 2차원 행렬 데이터 실험 모델을 표현한 것이고 Fig. 3과 Fig. 5는 3차원 텐서 데이터에서 사용한 실험 모델의 형태이다. 딥 오토 인코더를 구현할 때는 은닉 레이어의 개수와 뉴런의 수를 유동적으로 조정하여 실험하였으며 가장 오차율이 낮은 경우를 구하는 것을 실험 목표로 하였다.

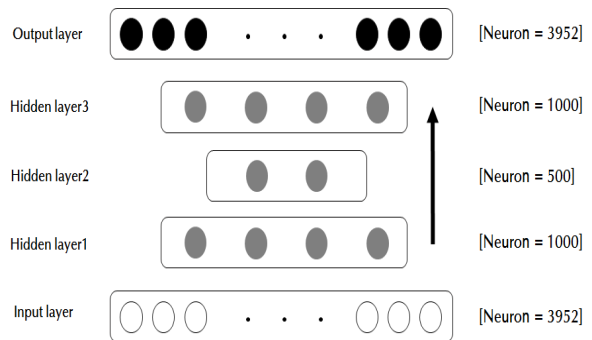


Fig. 4. Deep Autoencoder used in 2-dimensional Movielens data experiment

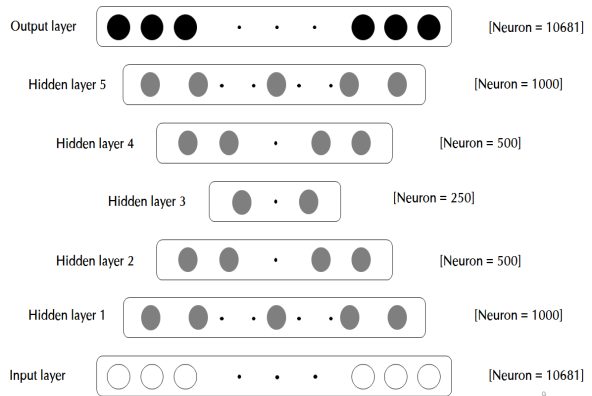


Fig. 5. Deep Autoencoder used in 3-dimensional Movielens data experiment

시스템 환경은 Ubuntu 16.04 64bit, Tensorflow 1.8.0 환경에서 실행하였으며 수행한 시스템 구현 환경은 Table 1을 통해 확인할 수 있다.

Table 1. Experimental system environment

Operating System	Ubuntu 16.04 64bit
CPU	Intel core i7-6700k CPU
Graphic Card	Nvidia GeForce GTX 1070
Language	Python3, Tensorflow 1.8.0

3.1 Dataset

실험 데이터는 사용자, 아이템, 평점으로 구성된 2차원 행렬 데이터와 사용자, 아이템, 시간, 평점 항목으로 구성된 3차원 텐서 데이터이며 2차원 데이터로 영화 데이터인 무비렌즈 데이터 [10], 3차원 데이터로는 텐서 데이터 [11]인 무비렌즈 데이터와 레스토랑 데이터인 엘프(Yelp) 데이터 [12]를 이용하였다.

실험에 사용한 2차원 행렬 데이터인 무비렌즈 데이터는 6,040명의 사용자들이 약 3,950개의 영화에 대해 평가한 데이터로, 사용자들이 시청한 영화에 대해서만 점수가 기록되어 있다. 점수는 1점부터 5점까지의 숫자로 평가되었으며 이를 0이 아닌 데이터(non-zero data)라고 한다. 또한 사용자들이 시청하지 않은 영화에 대해서는 평점이 0으로 기록되는데 이 데이터는 제로 데이터(zero data)라고 한다. Fig. 6은 실제로 실험에 사용한 2차원 영화 평점 입력 데이터를 나타낸 것이다. Fig. 6에서 한 행의 첫 번째 열은 사용자이며 두 번째 열은 영화를 나타내며 세 번째 열은 사용자가 시청한 영화에 대하여 평가한 평점 점수이다. 마지막 열은 타임 스탬프(time_stamp) 데이터로 본 실험에서는 사용하지 않는 값이다.

user::movie::rating::time_stamp
1::1193::5::978300760
1::661::3::978302109
1::914::3::978301968
1::3408::4::978300275
1::2355::5::978824291
1::1197::3::978302268
1::1287::5::978302039
1::2804::5::978300719
1::594::4::978302268
...
6040::162::4::956704953
6040::2725::4::997454180
6040::2728::5::957717123
6040::3388::1::956716407
6040::1784::3::997454464
6040::2739::2::956715942
6040::2745::3::956716157
6040::2746::2::956716157
6040::2750::2::956715865

Fig. 6. Actual 2-dimensional Movielens input dataset

실험에 사용된 3차원 텐서 데이터 [11]는 영화 평점 데이터인 무비렌즈 데이터 [10]와 레스토랑 평점 데이터인 엘프 데이터 [12]를 사용하였다. Fig. 7은 3차원 무비렌즈 데이터와 3차원 엘프 데이터를 나타내며 Fig. 7에서 확인할 수 있듯이 3차원 데이터는 2차원 행렬에 시간(Year-Month) 열이 추가된 데이터이다. 무비렌즈 데이터의 시간 열에서 나타내는 정수 값은 1부터 157까지이며 1995년 1월부터 2009년 1월까지의 시간을 나타낸다. 엘프 데이터 내 시간 열의 정수 값은 1부터 108까지로, 2005년 2월부터 2014년 1월까지의 시간이 포함되어 있다. 정수 값은 연도와 월을 내포하고 있는 데이터이다.

User	movie	time	rating
1	121	1	5.0
1	184	1	5.0
1	229	1	5.0
1	290	1	5.0
1	314	1	5.0
1	326	1	5.0
1	352	1	5.0
1	353	1	5.0
1	359	1	5.0
...			
71567	1753	22	3.0
71567	1793	22	3.0
71567	1826	22	2.0
71567	1834	22	4.0
71567	1837	22	4.0
71567	1899	22	1.0
71567	1900	22	1.0
71567	1901	22	1.0
71567	1902	22	1.0

(a) actual Movielens input dataset

User	restaurant	time	rating
1	1	1	3
2	2	2	5
3	3	3	3
4	4	4	4
5	5	5	1
6	6	6	4
7	7	7	2
8	8	8	3
9	9	9	5
...			
41056	401	60	3
70814	2402	70	5
70815	5578	13	1
14	1489	29	2
7695	10507	16	4
806	316	67	4
70816	42	60	1
70817	10644	36	1
37372	9181	61	5

(b) actual Yelp input dataset

Fig. 7. Actual 3-dimensional Movielens input dataset. (a) actual Movielens input dataset, (b) actual Yelp input dataset

본 연구에서는 시간 항목이 포함된 3차원 텐서 데이터를 이용하여 실험을 통해 시간 데이터와 각 데이터 항목들 간의 관계와 평점에 미치는 영향에 대하여 분석한다. 입력 데이터의 90%는 학습 데이터로, 나머지 10%는 테스트 데이터로 나누어 구성하였다. 학습 단계에서 인코더는 입력 데이터의 특징을 매핑한다. 인코딩 단계를 거치고 필요에 따라 활성화 함수를 취한 후 디코딩 단계를 시작한다. 본 연구에서는 Sigmoid 함수를 활성화 함수로 사용하였다. 그리고 실험에서의 추천 성능을 평가하기 위해 재구성된 결과와 실제 입력 데이터의 오차율(RMSE)을 계산해 실험 성능을 측정 및 비교하였다.

3.2 Experiments

실험에서 딥 오토 인코더는 은닉 레이어가 3개인 경우와 5개인 경우로 구현했으며 가장 오차율이 낮은 케이스를 찾는 것을 목표로 하였다. Table 2는 2차원 데이터에서 기존의 행렬 분해 방법과 딥 오토 인코더 실험의 오차율(RMSE)을 비교한

것이다. 딥 오토 인코더를 이용한 실험에서 오차율이 더 낮아진 것을 확인할 수 있었다. Table 2를 통해 3개의 은닉 층을 가진 딥 오토 인코더가 2차원 영화 평점 데이터인 무비렌즈 데이터에서 행렬 분해 방법보다 오차율이 줄어들었음을 알 수 있다.

Table 2. Comparison of experimental results by methods in 2-dimensional data

Data	Model	Test loss(RMSE)
2-dimensional MovieLens	Matrix Factorization	0.93
	Autoencoder	0.97
	Deep Autoencoder (3 Hidden layers)	0.92

2차원 행렬 데이터와 2차원 데이터에 시간 축이 포함된 3차원 텐서 데이터를 통해 구현한 오토 인코더와 딥 오토 인코더의 실험 결과는 Table 3을 통해 확인 가능하다. Table 3에서 레이어(Layer)는 입력 레이어와 출력 레이어를 제외한 은닉 레이어로, 만약 3(1000, 500, 1000)으로 표현되어 있다면 총 3개의 은닉 레이어를 사용하였으며, 첫 번째 은닉 레이어의 유닛 개수는 1,000개, 두 번째 은닉 레이어의 유닛 개수는 500개, 마지막 레이어의 유닛 개수는 1,000개임을 의미한다. 본 연구에서는 3차원 텐서 데이터인 영화 평점 데이터 무비렌즈와 레스토랑 평점 데이터인 옐프 데이터를 이용해 사용자가 평점을 주었을 당시의 시간(Year-Month) 항목과 다른 데이터 항목들과의 평점에 대한 연관성을 파악하였다. Table 3에서 데이터별

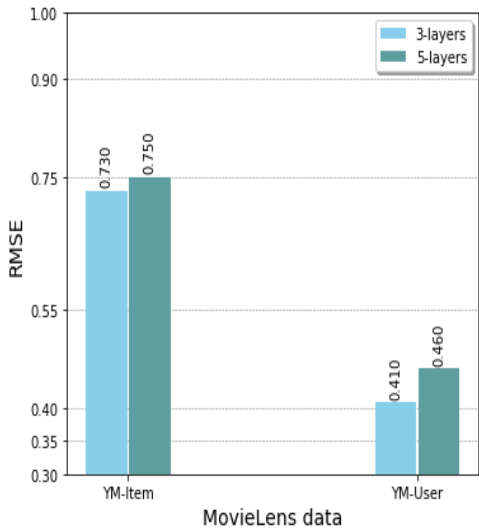
로 오차율(RMSE) 이 가장 낮은 값을 실험 결과 내에서의 최적의 결과라고 판단하였으며, 동일한 데이터에서 오토 인코더만을 가지고 실험했을 때의 오차율보다 딥 오토 인코더에서의 실험 오차율이 더 낮았으므로 3차원 텐서 데이터의 추천 시스템 연구에서 딥 오토 인코더가 더 효과적일 수 있으며 필요하다는 것을 확인하였다.

또한 Fig. 8을 통해 3차원 데이터에서의 실험 결과를 그래프로 나타내었다. 그래프를 통해 은닉 레이어가 3개일 때와 5개일 때의 딥 오토 인코더를 이용한 3차원 무비렌즈 데이터와 옐프 데이터에서의 시간(Year-Month)과 아이템(Item), 시간(Year-Month)과 사용자(User)와의 실험 결과를 비교할 수 있다. 무비렌즈 데이터에서는 시간과 사용자 데이터를 사용해 실험한 결과(0.41)가 시간과 아이템 데이터를 사용해 실험한 결과(0.73)보다 오차율이 낮게 측정되었고 5개의 은닉 레이어를 사용했을 때보다 3개의 은닉 레이어를 사용했을 때 더 오차율이 낮게 측정되었다. 반면 옐프 데이터에서는 시간과 사용자 데이터를 이용한 실험한 결과(1.6)보다 시간과 아이템 데이터를 이용한 실험 결과(1.17)가 더 오차율(RMSE) 이 낮았다.

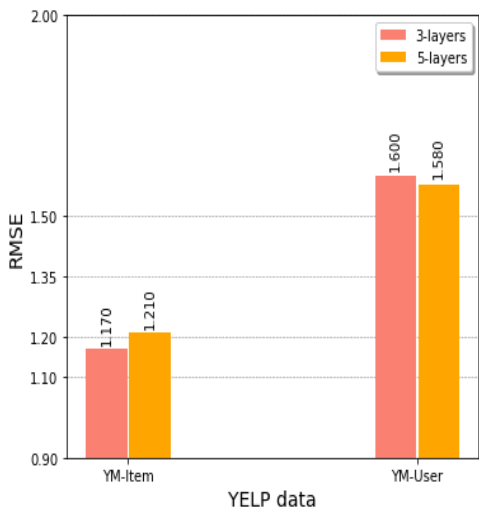
이를 통해 추천 시스템을 구현할 때, 실험 데이터에서의 시간과 아이템의 관계와 시간과 사용자의 관계는 데이터마다 상이할 수 있다는 것을 확인할 수 있었다. 또한 은닉 레이어의 개수와 뉴런 수도 각 데이터마다 다른 구성에서 최적의 결과를 보여주었다.

Table 3. Results of total Autoencoder and Deep Autoencoder experiment

Data	Model	Layer(Unit)	Method	Test loss (RMSE)
2-dimensional MovieLens	Autoencoder	1(500)	Item-based (User-Item)	0.97
		3(1000,500,1000)		0.96
	Deep Autoencoder	3(500,500,1000)		0.94
		3(500,800,1000)		0.92
		5(1000,500,250,500,1000)		0.99
		5(128,250,500,800,1000)		1.16
		3-dimensional MovieLens		Autoencoder
3-dimensional MovieLens	Deep Autoencoder	3(512, 256, 512)	User-based (YM-User)	0.44
		3(256,128,256)	Item-based (YM-Item)	0.73
		5(1000,500,250,500,1000)	User-based (YM-User)	0.41
		5(1000,500,250,500,1000)	Item-based (YM-Item)	0.75
		5(1000,500,250,500,1000)	User-based (YM-User)	0.46
3-dimensional YELP	Deep Autoencoder	3(500,800,1000)	Item-based (YM-Item)	1.17
		3(1024,512,1024)	User-based (YM-User)	1.6
		5(1000,500,250,500,1000)	Item-based (YM-Item)	1.21
		5(1000,500,250,500,1000)	User-based (YM-User)	1.58



(a) MovieLens data



(b) YELP data

Fig. 8. Comparison of results of Deep Autoencoder experiments considering time column in 3-dimensional tensor data. (a) MovieLens data, (b) YELP data

IV. Conclusions

본 연구에서는 기존의 추천 시스템 연구에서 사용되었던 데이터인 2차원 행렬 데이터와, 2차원 행렬에 시간 열이 포함된 3차원 텐서 데이터를 이용하여 오토 인코더 실험을 수행하였으며 오토 인코더의 은닉 레이어를 겹겹이 쌓아 더 깊어진 형태의 딥 오토 인코더를 설계하여 추천 시스템을 구현하였다.

실험 데이터는 영화 평점 데이터인 무비렌즈 데이터와 레스토랑 평점 데이터인 옐프 데이터를 사용하여 2차원 행렬 데이터와 3차원 텐서 데이터를 비교하였으며 3차원 텐서 데이터에

서 사용자의 예상 평점에 대한 데이터 항목들 간의 관계를 분석하고자 하였다. 또한 오토 인코더와 딥 오토 인코더의 인코딩 및 디코딩 단계를 거친 입력 데이터는 예측 벡터로 재구성되는데, 이 예측 벡터를 통해 아이টে에 대한 사용자의 미래 평점을 예측한다. 이를 통해 추천 시스템에 적용함으로써 활용할 수 있다. 실험을 통해 2차원 행렬 데이터와 3차원 텐서 데이터 모두 은닉 레이어의 구성에 따라 딥 오토 인코더를 이용한 실험 결과가 오토 인코더만으로 실험한 결과보다 오차율(RMSE)이 낮아 추천 성능에 영향을 줄 수 있다고 판단하였다. 그리고 기존의 전통적인 선행 연구 방법인 행렬 분해 방법과 동일한 데이터로 비교하였을 때도 딥 오토 인코더의 실험 결과에서 오차율(RMSE)이 더 낮았음을 확인하였다. 또한 실험에서 사용한 3차원 텐서 데이터를 통해 시간 데이터를 포함한 각 데이터 항목 간의 연관성을 파악할 수 있었다. 실험을 통해 추천 시스템 연구를 위한 딥 오토 인코더 모델을 구현할 때는 은닉 레이어의 양을 무조건 늘린다고 해서 오차율이 낮아지는 것이 아니며 입력 데이터들의 특성에 맞게 각 항목들의 관계를 분석하여 학습 변수 및 은닉 레이어의 뉴런 수 등을 재구성해야 한다는 것을 확인하였다. 본 실험은 GPU 텐서플로우 환경에서 수행되었으며 트레인 이폭(train epoch)을 1,000번 수행할 때 약 20분 내의 시간이 소요되었다.

본 연구를 통해 2차원 행렬 데이터에서 시간 열이 추가된 3차원 텐서 데이터를 이용하여 사용자와 아이টে, 시간을 고려한 영화와 레스토랑 추천 시스템을 제안하였으며 수행한 내용을 확장하여 향후 3차원 이상의 고차원 데이터를, 학습 전 전처리 과정 없이 연구에 활용할 수 있도록 진행할 계획이다. 따라서 3차원 이상의 고차원 데이터를 딥 오토 인코더 학습에 적용할 때 데이터를 나누기 위한 전처리과정 없이 순수한 데이터 그대로를 입력받아 학습시킬 수 있을 것으로 기대된다. 추후 연구로 딥 오토 인코더 학습 알고리즘 내 사용한 알고리즘 계산방식과 활성화 함수, 은닉 레이어의 뉴런의 개수 등을 수정하여 실험 결과를 보완하는 것을 연구목표로 한다.

REFERENCES

- [1] J. B. Schafer, D. Frankowski, J. Herlocker, & S. Sen. "Collaborative filtering recommender systems." In The adaptive web. Springer, Berlin, Heidelberg, pp. 291-324, 2007.
- [2] G. Adomavicius, & A. Tuzhilin, "Context-aware recommender systems". In Recommender systems handbook, Springer, Boston, MA, pp. 217-253, 2011.
- [3] Y. Koren, R. Bell, & C. Volinsky, "Matrix factorization techniques for recommender systems". Computer, (8), pp. 30-37, 2009.

- [4] Y. LeCun, Y. Bengio, & G. Hinton, “Deep learning”. *nature*, 521(7553), 436. 2015.
- [5] R. Salakhutdinov, A. Mnih, & G. Hinton, “Restricted Boltzmann machines for collaborative filtering”. In *Proceedings of the 24th international conference on Machine learning* pp. 791–798, ACM, 2007.
- [6] J. Bennett, & S. Lanning, “The netflix prize”. In *Proceedings of KDD cup and workshop*, pp. 35, 2007.
- [7] C. A. Gomez-Urbe, & N. Hunt, “The netflix recommender system: Algorithms, business value, and innovation”. *ACM Transactions on Management Information Systems (TMIS)*, 6(4), 13. 2016.
- [8] P. Covington, J. Adams, & E. Sargin, “Deep neural networks for youtube recommendations”. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 191–198. ACM. 2016.
- [9] N. Jones. “Computer science: The learning machines”. *Nature News*, 505(7482), 146, 2014.
- [10] MovieLens. <http://grouplens.org/datasets/movielens/> (accessed Jan., 08, 2019).
- [11] BigTensor, <https://datalab.snu.ac.kr/bigtensor/> (accessed Jan., 08, 2019).
- [12] Yelp, http://www.yelp.com/dataset_challenge/ (accessed Jan., 08, 2019).

Authors



Jina Park received the B.S. degrees in Computer Science and Engineering from Kookmin University, Korea, in 2017, and M.S. degrees in Computer Science and Engineering from Ewha Womans University, Korea, in 2019, respectively

Jina Park is interested in Deep learning, Database, and Recommendation system.



Hwan-Seung Yong received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Seoul National University, Korea, in 1983, 1985 and 1994, respectively Hwan-Seung Yong is currently a Professor in the Department of

Computer Science, Ewha Womans University. He is interested in Database, Data mining, OLAP, Multimedia Database and Ubiquitous computing.