Recommendation Systems Approaches on the Netflix Prize Data Set

Cody Blakeney
Email: cjb92@txstate.edu

Samuel Teich
Email: st1289@txstate.edu

***Abstract*—Recommendation systems play an increasingly vital role in the ability to make informed decisions in modern societies, given the ever growing deluge of data which might inform those decisions. Without the ability to quickly choose from among a few well selected options ordinary activities from applying to jobs to searching the internet, buying household goods, or reading the latest news might become difficult, deceptive, or even dangerous. Given the many relatively recent major advances in machine learning techniques, especially in relation to deep learning, it is important to understand how recommendation systems have changed and improved. In this paper we investigate approaches to recommendation systems using deep learning.**

1. Introduction

Data characterizes the modern world. Corresponding to exponential growth in computing power over time, and the related proliferation of sensors and interactions with computing systems, the volume of recorded data has exploded. It is estimated that in 2010 there was more than 1 Zettabyte of recorded unstructured data, and that as of 2018 there exists more than 2 Zettabytes structured data and 20 Zettabytes structured data [2][6]. While much of this deluge is comprised of highly specific and narrowly specialized data, not useful for making decisions at a personal level, there still is a wealth of data - far more than is human interpretable - which could be used to inform such decisions. Recommendation systems are predictive tools designed specifically for this use case, helping pare from a glut of options a select few choices based on available data.

The Netflix Prize competition [1], started in 2006 and finalized in 2009, is, along with the publishing of the ImageNet dataset in 2009 [3], credited as one of the events that encouraged the perception of a revolution in the field of machine learning both among professional audiences and the public at large [5]. In the near decade since team BellKor's Pragmatic Chaos won the Netflix Prize competition [1], rapid advances, especially in Deep Learning, have continued to push boundaries in machine learning and recomendation systems. The Netflix dataset remains an important benchmark dataset for recomendation systems.

In this paper we will discuss recent improvements in the field of recomendation systems and attempt to benchmark some of those improvments on the Netflix dataset. Section 2 will cover some important backround material on the field of recomendation systems. Section 3 will discuss some recent and related results. Section 4 will talk about our experimental setup, while Section 5 will include the specific experiments carried out. Section 6 discusses our results, conclusions, and will cover our expectations for future work.

2. Background

Some basic concepts

2.1. Recommendation Systems

test

2.2. Deep Learning

test

2.3. Embedding Layers

test

3. Related

test

3.1. Takeaway From Related Work

test

4. Approach

Our approach to investigating the architecture for deep learning recommendation systems, was to first decide on a baseline with which we could compare the effectiveness of our techniques. After finding a benchmark we reviewed the literature we had assembled to decide which design was most applicable to our dataset, and our ability to implement.

We knew from the Netflix competition that the winners had an RMSE of 0.8567. We also wanted a way to compare our engineered solution, to one that simple took all the data Netflix provided and used it as inputs on a naive feed-forward neural net.

The theme of all the papers we read covering the various methodologies for applying deep learning to recommendation systems, was creating more meaningful representations

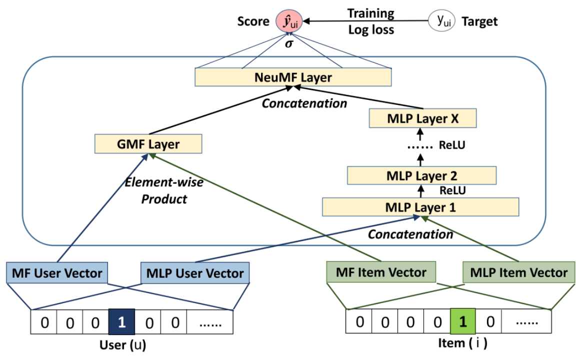

Figure 1. NCF diagram

of the users, the items they interacted with, and context around the item. Many of the models we read about involved many complex hierarchical layers. While many of these exotic arrangements made sense, we were not sure we had the technical expertise to implement them in the allotted time. We settled on a technique called Neural Collaborative Filtering [4]. Neural Collaborative Filtering or NCF is a process that involves creating a hybrid of traditional recommendation system and deep learning. As you can see from figure 1 it combines user vectors and item vectors from matrix factorization as well as user and item embedding layers as inputs to the neural network. In order to try to replicate this architecture we would need to need to find appropriate Matrix Factorization representations for our users and our movies, as well as create embedding layers for for them.

5. Setup

test

5.1. The Netflix Dataset

test

5.2. The IMDB Dataset

test

5.3. Experimental Setup

test

6. Experiments

6.1. Characteristics of the Netflix Dataset

6.2. Naive Feed-forward Deep Neural Networks

What we call our Naive Feed-forward network is the benchmark we made to test the efficacy of the models we were training. We tested several versions before deciding on a final candidate Naive Model. We had a categorical classifier, a regression model, and another categorical model with a better layer design.

6.2.1. Naive Categorical Model. Our first model had inputs of one hots for movies and users, mean, median and variance of the current movie calculated from the training set, and time represented as a month input, a day input, and a year since beginning input. This model had two hidden layers with 128 nodes each using relu activation and a softmax output to guess categorically what the rating was for that user. With this configuration the model converged to about 38% accuracy on our validation set. Because of the distribution of ratings in our data set this is about as good as always guessing the rating 4 for every movie. Not great, but better than randomly chance.

6.2.2. Naive Regression Model. For training the problem as a regression model we learned we had to be very careful with the layer selection. Too few layers and inadequate funneling down of the layers towards the output layer resulted in a model that was hopelessly lost at trying to find a gradient. We finally had success with a model using a hidden layers of size 512, 64, 8, and a linear output layer. After training we were surprised because it had a very good RMSE score. Closer inspection showed that the model was predicting a floating point value that was always between 3.4 and 3.6, very close to the mean value of the data set. When we rounded the value and calculated the accuracy and RMSE it was only 21% accurate with and RMSE of 1.203.

6.2.3. Naive Categorical better layer design.

6.3. Neural Network Embeddings on the Netflix Dataset

6.4. Neural Network Embeddings on the IMDB Dataset

6.5. SVD Embeddings on the Netflix Dataset

When considering the methods for representing

6.6. Deep Feed-forward Neural Networks with Embedding Layers

6.7. Hierarchical Architectures

7. Conclusions

7.1. Experimental Discussion

7.2. Future Work

References

[1] [Online]. Available: https://www.netflixprize.com/index.html

[2] J. R. David Reinsel, John Gantz, "Data age 2025: The evolution of data to life-critical," 04 2017. [Online]. Available: https://www.seagate.com/files/www-content/our-story/trends/files/Seagate-WP-DataAge2025-March-2017.pdf

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: a large-scale hierarchical image database," pp. 248–255, 06 2009.

[4] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural Collaborative Filtering," 2017. [Online]. Available: http://arxiv.org/abs/1708.05031

[5] H. Quentin, "Reasons to believe the a.i. boom is real," 03 2018. [Online]. Available: https://www.nytimes.com/2016/07/19/technology/reasons-to-believe-the-ai-boom-is-real.html

[6] L. Rizzatti, "Digital data storage is undergoing mind-boggling growth," 09 2016. [Online]. Available: https://www.eetimes.com/author.asp?section_id=36&doc_id=1330462