Collaborative Text Filtering

Peter J. Bakke (s183268), Daniel Horvath (s172185), Christian Hansen (s146498) and Thomas Brenner (s181857)

DTU Compute, Technical University of Denmark

Introduction

Collaborative text filtering is one of the most popular and effective approaches for recommender systems. Recommender systems are based on the idea, that given previously collected data about users and their interactions with items, you can predict whether a given user wants to have an interaction with a given item. This is widely used for platforms like Netflix, Amazon, Youtube and news websites. These platforms can increase their profits by being able to predict their consumers interests and showing content relevant for the user.

The purpose of this project is to match two text descriptions of varied lengths. More concretely we propose a model to recommend articles to users based on the abstracts from other articles a user has indicated as read.

General Methodology

In general we try to maximize the probability of a match between a user and an item given some features:

$$\max p(m|K;\theta)$$

where m denotes the binary variable on whether there is match, K denotes the features, and θ denotes the parameters in the model.

To predict a users preferences we use

$$p(m) = \sigma(f(\text{UserId}, \text{MovieId}))$$

where $\sigma(\cdot)$ denotes the sigmoid function and $f(\cdot)$ is some function of users and items. In the matrix factorization $f(\cdot)$ could be given by the inner product of embeddings of users and items, e.g.:

$$f(\cdot) = u \cdot m^T$$

where

$$u = \text{Embedding}(x_u)$$
$$m = \text{Embedding}(x_m)$$

When we turn to the more advanced models $f(\cdot)$ is e.g. a neural net or LSTM net with user/item embeddings as input features instead of a simple inner product between the two, [2, 9, 6].

Data

- We use the publicly available **MovieLens** dataset from `https://grouplens.org/datasets/movielens/` for the first part of our project
- We use the publicly available **CiteULike** from `http://www.citeulike.org/faq/data.adp` for the second part of our project

Key points

- We construct a baseline model using **Matrix Factorization** on the MovieLens and CiteUlike datasets
- We construct a **Collaborative Text Filtering** model on the same data using
  - Feed Forward Networks, [1, 7]
  - LSTM Networks, [3]

  And compare the results to the baseline model, [4, 8, 5]
- We implement the models using the **Pytorch** deep learning framework and use **TorchText** for creating word embeddings and batch iterators
- We train the models using virtual machines on the **Google Colab** GPU cloud

MovieLens

The MoveieLens dataset consist of 20 million ratings on a scale from 0.5 to 5, of 27,000 different movies by 138,000 users. Taking outset in the MovieLens dataset the objective is to predict how a specific user will rate a specific movie. Figure 1 display the architecture we use in the simple collaborative filtering neural network for the MovieLens dataset. Table 1 show the root mean squared error (RMSE) for the matrix factorization and the simple collaborative filtering neural network on the MovieLens dataset. For comparison we have included the RMSE from FastAI that use a similar model on the same dataset.

The Figures 2 and 3 show how the matrix factorization and simple collaborative filtering neural network trains on the MovieLens dataset.

Table 1: Results

Model	RMSE
Matrix Factorization	0.895
Neural Network - FastAI	0.889
Neural Network - our result	0.881

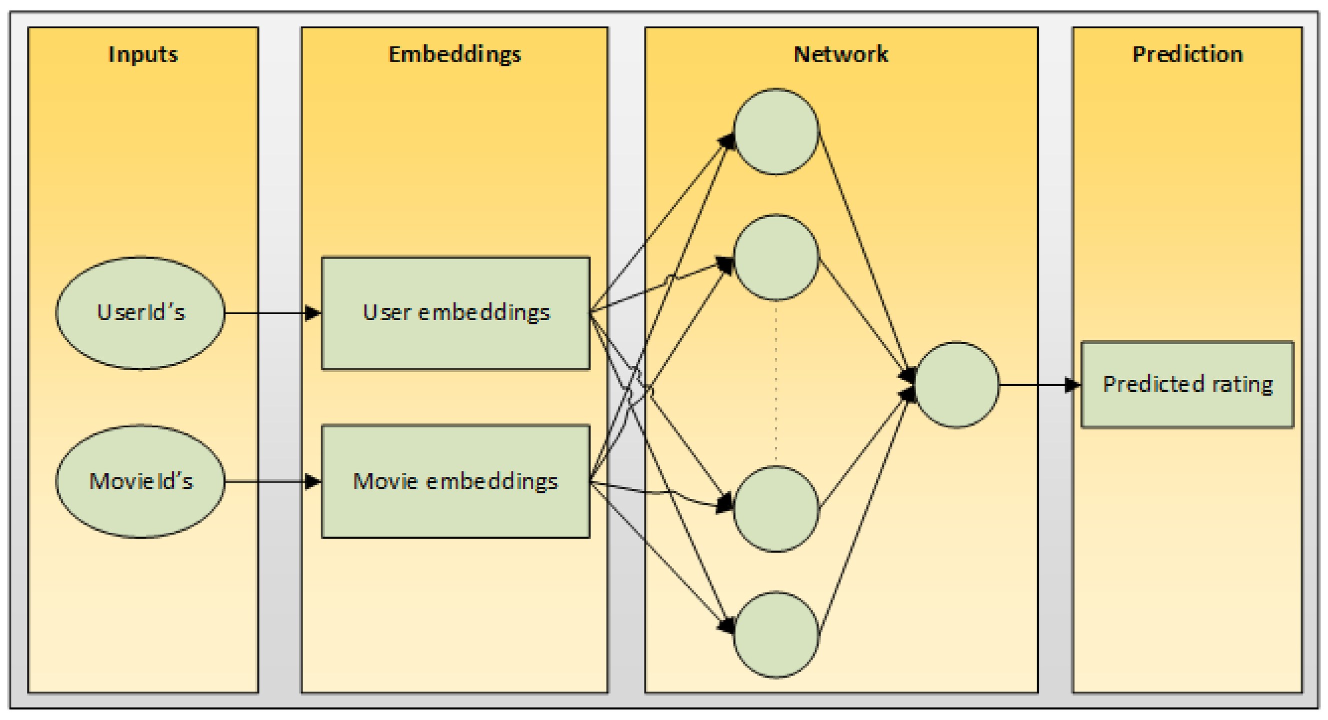

Figure 1: MovieLens neural net representation

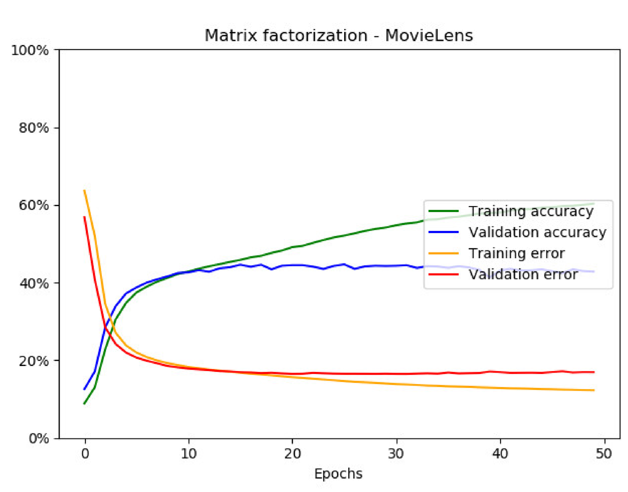

Figure 2: Training and Validation loss and Accuracy for the Matrix Factorization

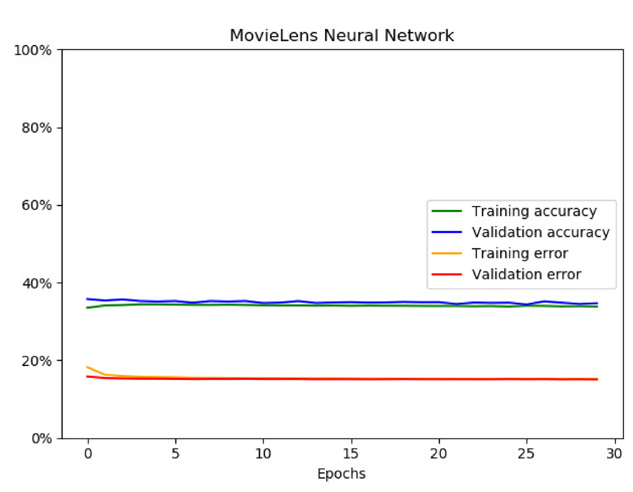

Figure 3: Training and Validation loss and Accuracy for the Neural Network

CiteULike

The CiteULike dataset consist of users represented by an Id and articles represented by an Id, the article Title and the article Abstract. The user-article interaction is represented as a binary attribute on whether a specific userId has put an ArticleId in his/her "basket". The modelling aim is to build a model that can recommend articles to users based on what they have previously read.

Table 2: Results

Model	Val. Accuracy	#Epoch
MF	74%	20
FNN	58%	3
LSTM	84%	15

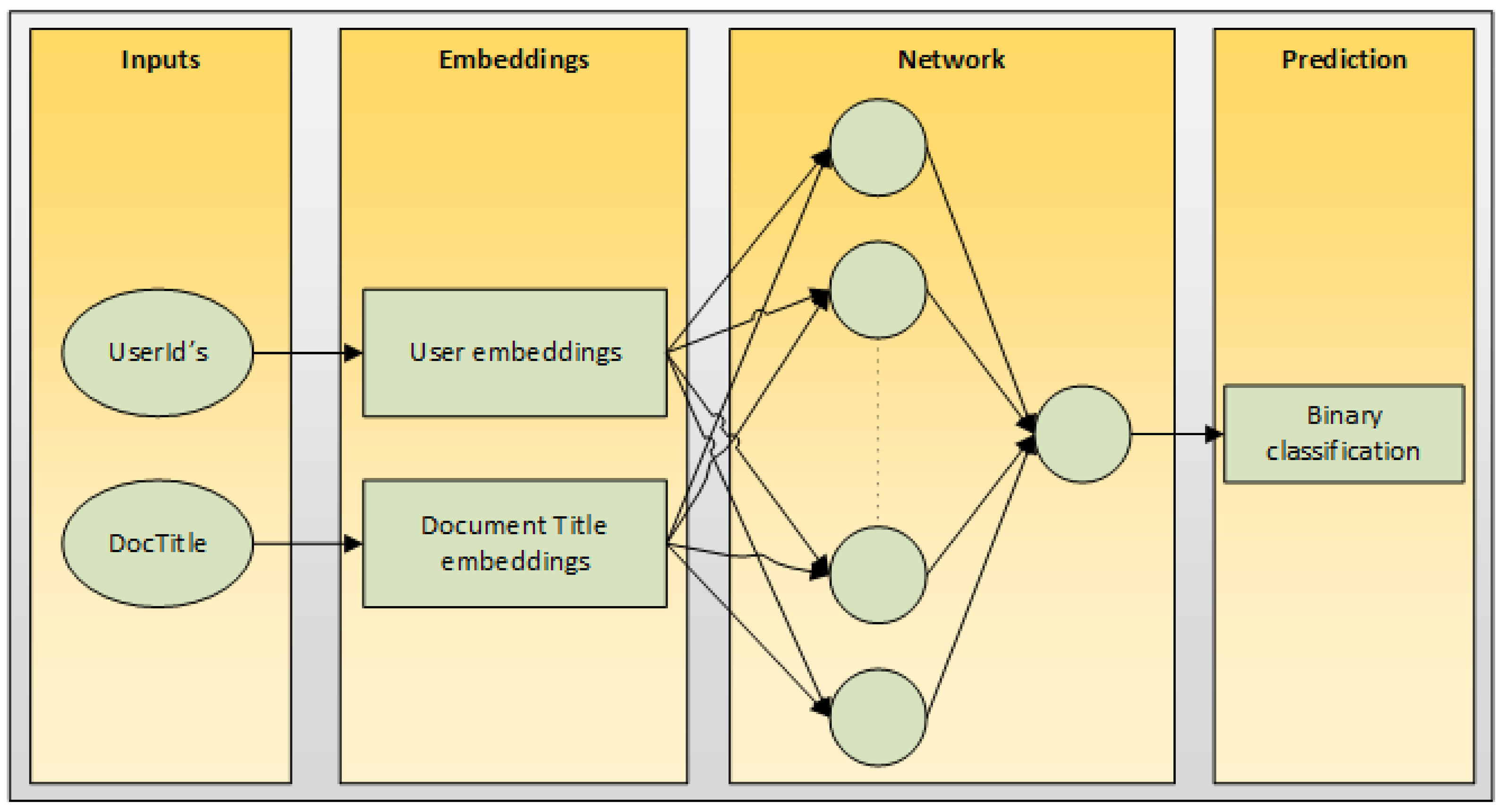

Figure 4: CiteULike Neural net representation

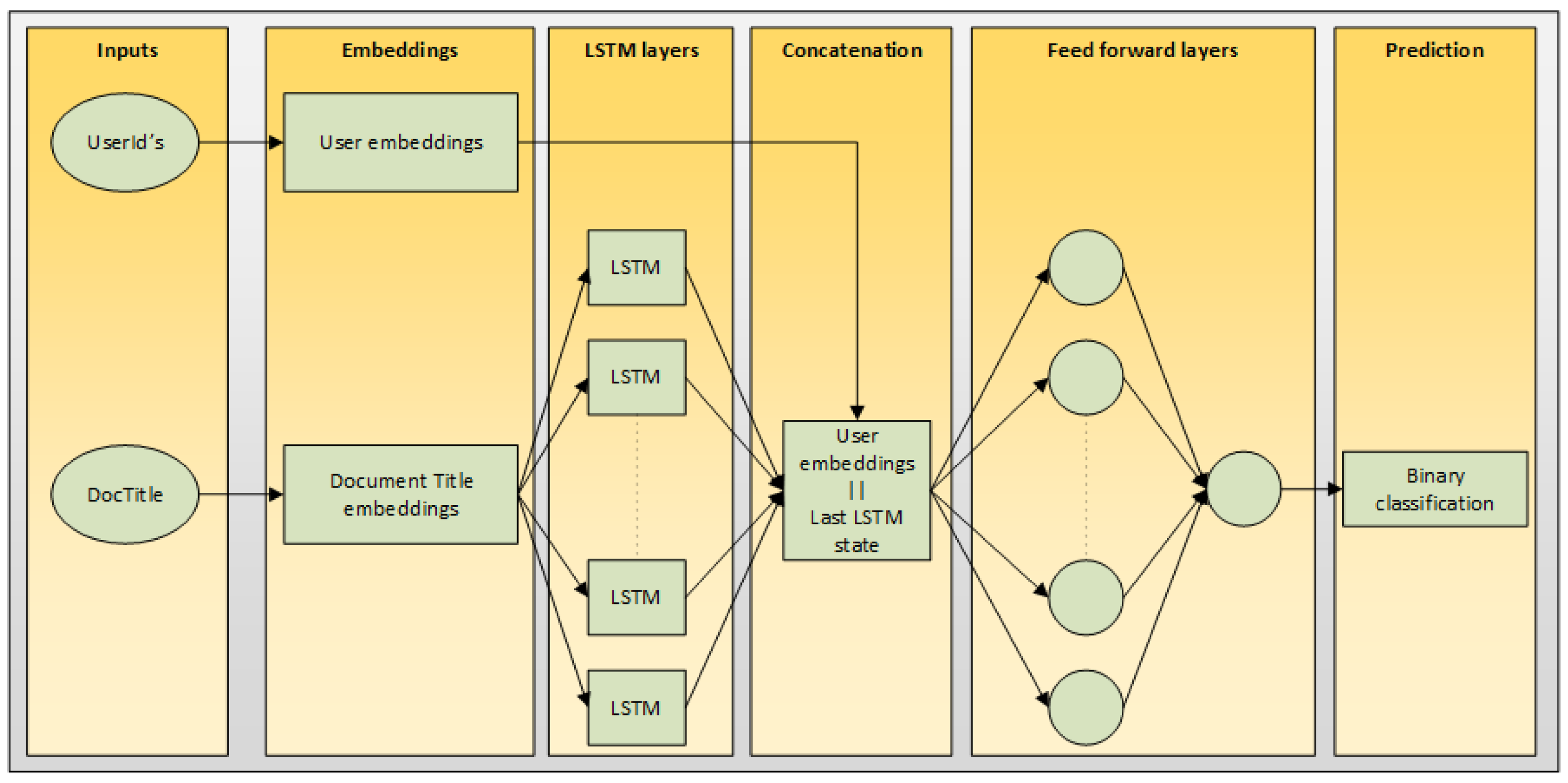

Figure 5: CiteULike LSTM net representation. For clarity the temporal structure of the LSTM blocks have not been shown.

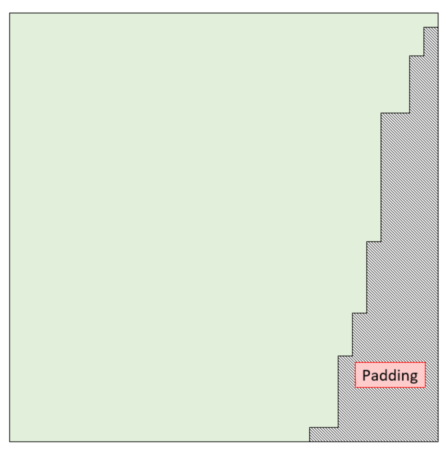

Figure 6: Batch sorting

Figure 4 show the simple collaborative filtering neural network architecture used on the CiteULike dataset. Figure 5 show the architecture of the collaborative filtering network enhanced with a number of LSTM layers and blocks. When training the model we need to take into account that the sequences are of varying length in order to not waste too much computation on calculating on the padding. Figure 6 visualize how the amount of padding (the grey area) is minimized in the batch by sorting the documents on the length of the sentences.

Table 2 show the best achieved accuracy on the validation set in the three different models together with how many Epochs it took to reach that result.

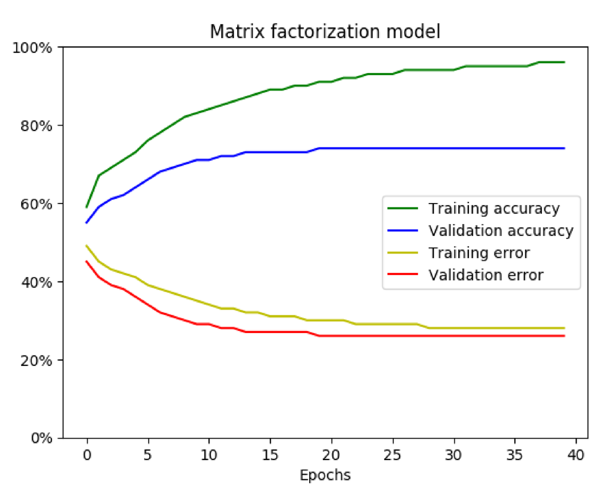

Figure 7: Training and Validation loss and Accuracy for the Matrix Factorization on the CiteULike data

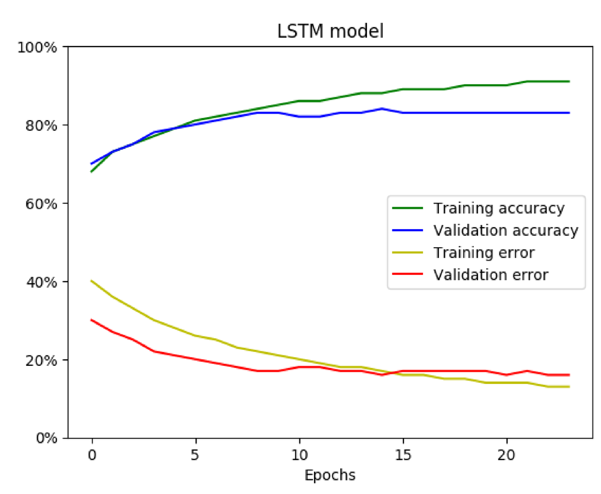

Figure 8: Training and Validation loss and Accuracy for the LSTM net on the CiteULike data

Acknowledgements

The authors wish to thank Alexander R. Johansen and Jose Juan Almagro Armenteros from DTU Lyngby for their constructive feedback and fruitful discussions during the process of the project.

References

[1] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua. Neural collaborative filtering. *CoRR*, abs/1708.05031, 2017. URL `http://arxiv.org/abs/1708.05031`.

[2] F. Hill, K. Cho, and A. Korhonen. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*, 2016.

[3] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

[4] G. Hu, Y. Zhang, and Q. Yang. LCMR: local and centralized memories for collaborative filtering with unstructured text. *CoRR*, abs/1804.06201, 2018. URL `http://arxiv.org/abs/1804.06201`.

[5] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, 2008.

[6] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

[7] M. A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015. URL `http://neuralnetworksanddeeplearning.com/`.

[8] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM. ISBN 1-58113-348-0. doi: 10.1145/371920.372071. URL `http://doi.acm.org/10.1145/371920.372071`.

[9] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 353–362, New York, NY, USA,

s183268@student.dtu.dk