

Research Literature

2019-2020 Masters Thesis Development

Riley Miller (rileymiller@mymail.mines.edu)
Colorado School of Mines

November 1, 2019

References

- [1] F. Chollet, *Deep Learning With Python*. Shelter Island, NY: Manning, 2018.

Notes on the book, key takeaways and deep learning fundamentals

- [2] H. S. H. 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, “Wide & deep learning for recommender systems,” in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, September 2016.

Cheng et al. implemented a combination of wide and deep learning to surface relevant content to users on the Google Play store. They used the two different techniques to combat some of the model fallacies that can appear in recommender systems. The team of researchers tackled the two primary tradeoffs for recommender systems by using wide and deep learning to optimize for memorization and generalization. Memorization in recommender systems “can be defined as learning the frequent concurrence of items and features and exploiting the correlation in historical data”. On the other hand, generalization is based on transitivity of correlation and explores new and rare feature combination, this approach leads to greater diversity of recommendations. On a practical basis, the researchers utilized wide learning to help with

the memorization aspect, many recommender systems utilize simple logistic regression algorithms to surface relevant content. However, this approach falls short when trying to query item feature pairs that haven't appeared in the training data. On the other hand, many researchers have tried to use deep learning neural networks to generate recommendations since they're able to generate feature pairs that haven't appeared in the training set, however, these deep learning approaches can over generalize and make less relevant recommendations. These researchers made use of joint training which simultaneously optimizes the parameters of the wide and deep models at training time. The teams data pipeline consisted of three stages: data generation, model training, and model serving. During training their model input layer took in training data and vocabulary and generated sparse and dense features with labels. The wide learning component consisted of the cross product transformation of user installed apps and impression apps. The deep learning model consisted of a 32 dimension embedding vector which is learned for each categorical feature. These embeddings are then concatenated together with dense features, resulting in a dense vector of approximately 1200 dimensions. These models were trained on over 500 billion examples. To optimize for performance, the researchers optimized the performance by using multithreading parallelism by running smaller batches in parallel instead of running all of the candidate inferences through the model at the same time. The recommender servers recommend over 10 million apps a second. <https://dl.acm.org/citation.cfm?id=2988454>

- [3] H. W. N. Wang and D.-Y. Yeung, "Collaborative deep learning for recommender systems," *Proc. KDD*, pp. 1235–1244, 2015.

These researchers introduced a concept they called collaborative deep learning (CDL) which is a hierarchical Bayesian model which combines deep representation learning for the content information and collaborative filtering for the ratings. The paper introduces three main approaches for recommender systems: content based models, collaborative filtering models, and hybrid methods. The content based methods make use of user profiles and product descriptions to make recommen-

dations. CF approaches use history such as past activities or preferences without using user or product info. While hybrid methods utilize a mixture of both content and CF approaches. Limitations for content models are user privacy while CF approaches stumble when ratings on products are sparse. The researchers highlight that deep learning models are state of the art in other fields such as computer vision and NLP and that they are able to learn features automatically, yet they are inferior to shallow models such as CF when it comes to learning similarity and relationships between items. To construct their CDL model, the researchers utilized Stacked Denoising Autoencoders (SDAE) which is a feedforward neural network for learning representations.