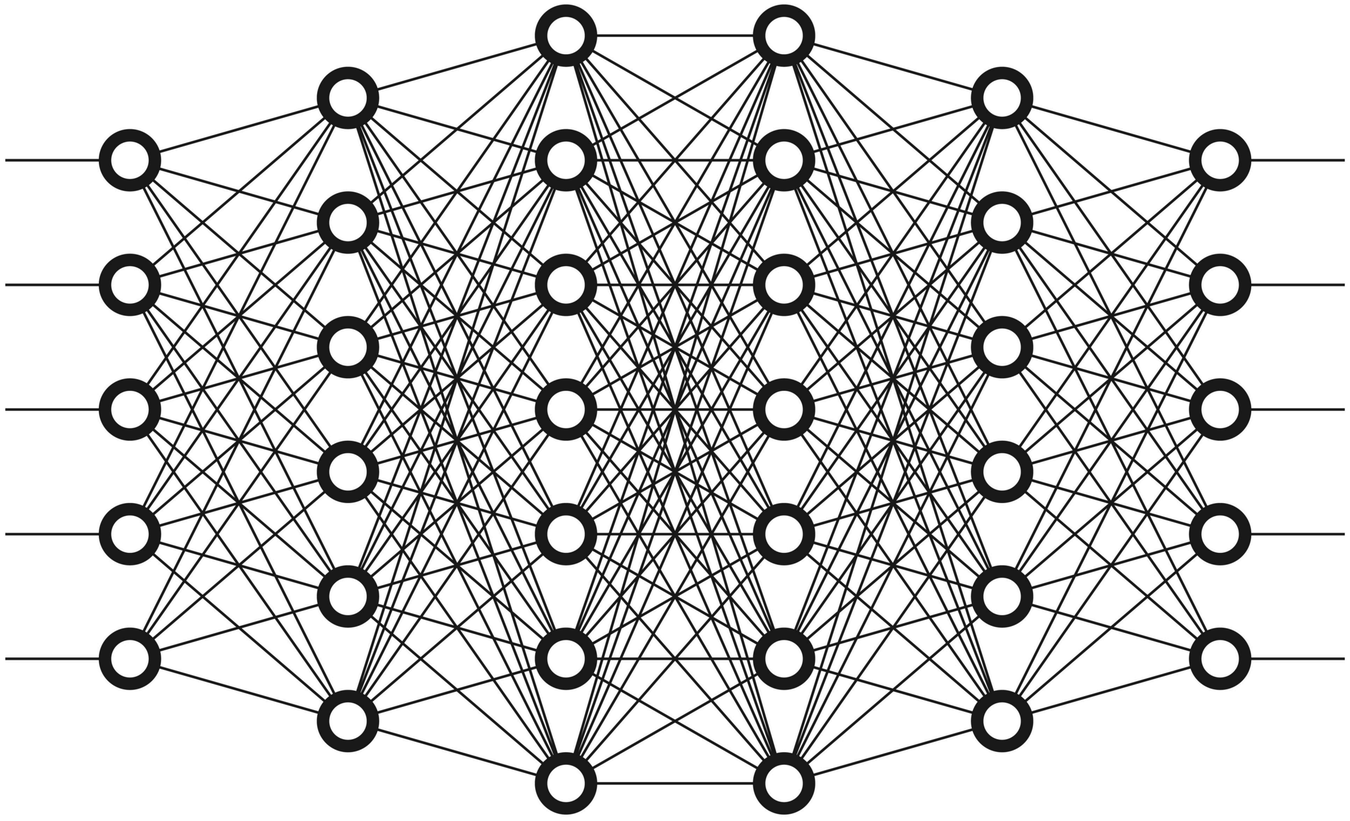
Master on Foundations of Data Science

Recommender Systems

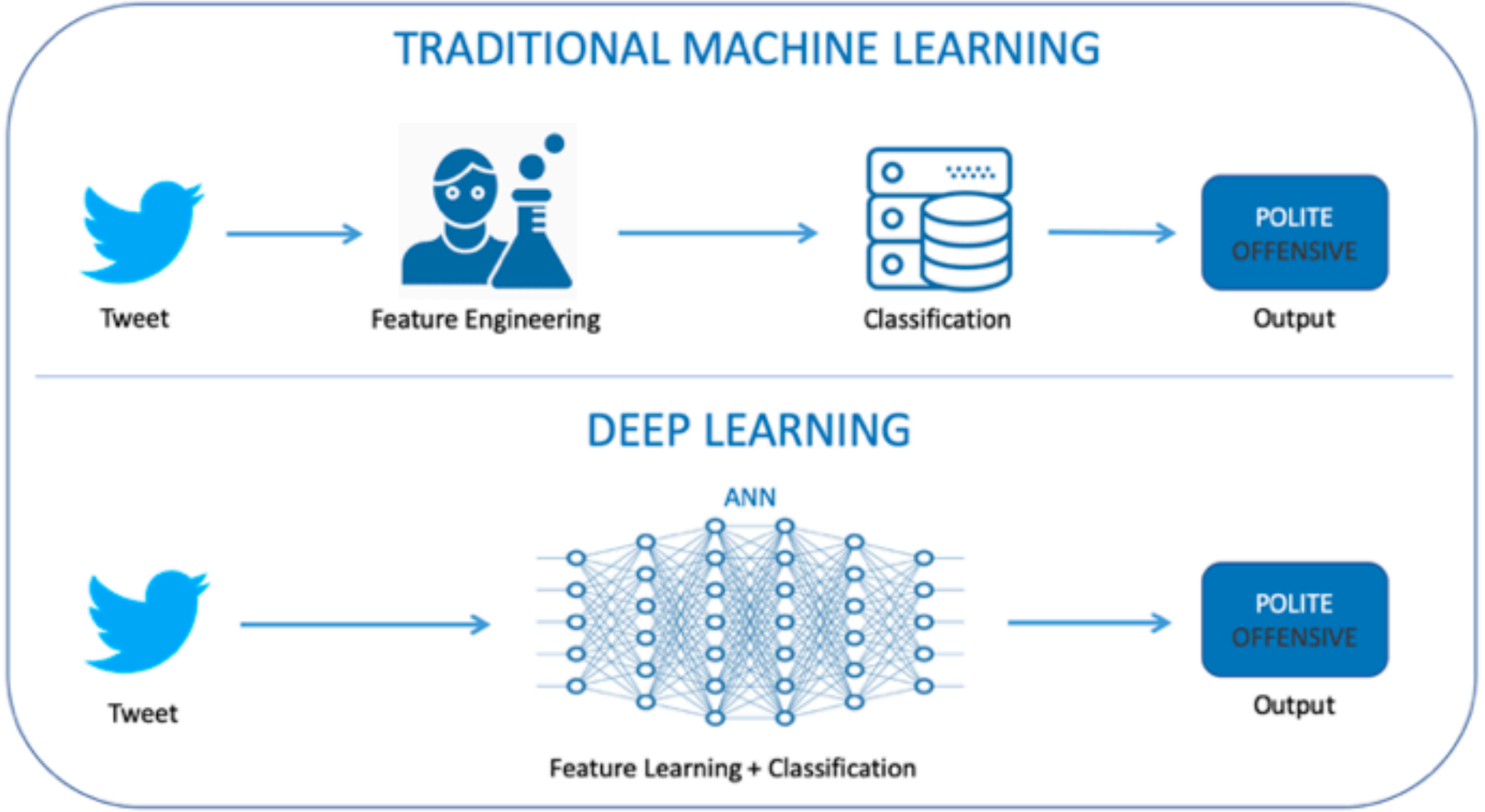
Deep Learning

Santi Seguí | **2021-2022**

What is deep Learning?



An example


TRADITIONAL MACHINE LEARNING
Tweet
Feature Engineering
Classification
POLITE
OFFENSIVE
Output
DEEP LEARNING
ANN
Tweet
Feature Learning + Classification
POLITE
OFFENSIVE
Output

From Linear to Non-Linear Models

Why Deep Learning in RecSys

- Deep learning can **model the non-linear interactions** in the data;
- **Reduce the effort** to define hand-crafted features;
- Enables recommendation models to include **heterogeneous** information such as text, images, audio,..
- Deep learning is powerful for **sequential modeling tasks;**
- Deep learning possesses **high flexibility** (easy to use complex new loss functions).

Drawbacks

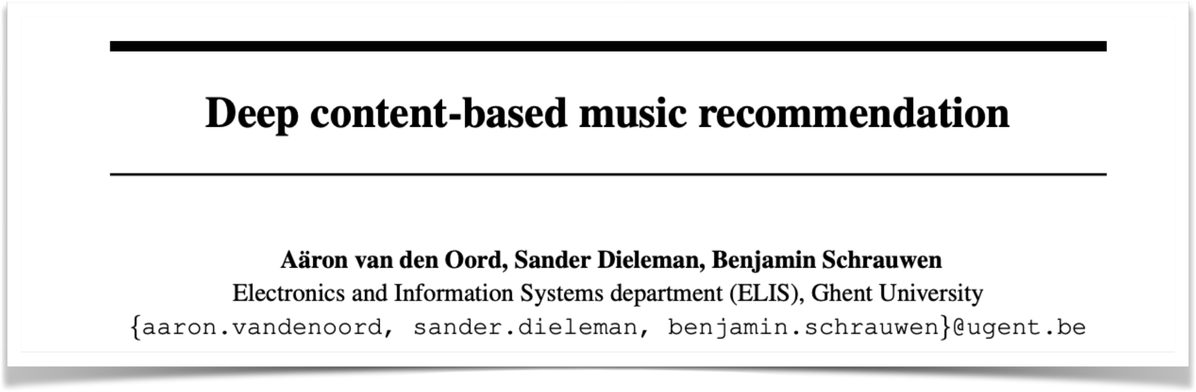
- Lack of **interpretability**
- Needs of **Big Data**
- Hyperparameter **tuning**

Content Based information

- Interesting solutions:
  - Deep content-based music recommendation
  - Content2Vec [Nedelect et.al. 2017]

MUSIC RECOMMENDATION

Problem: what happen if a new item is released?

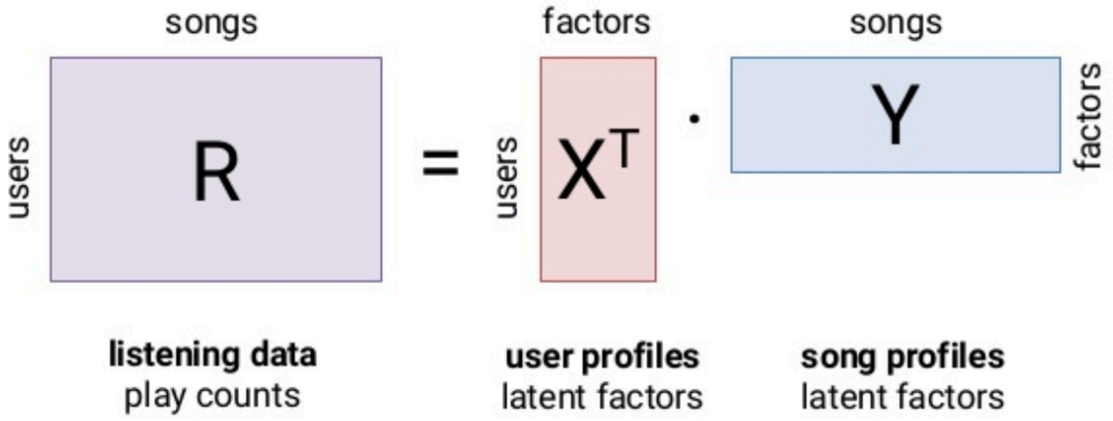


Deep content-based music recommendation

Aäron van den Oord, Sander Dieleman, Benjamin Schrauwen
Electronics and Information Systems department (ELIS), Ghent University
{aaron.vandenoord, sander.dieleman, benjamin.schrauwen}@ugent.be

Deep content-based music recommendation

Aäron van den Oord, Sander Dieleman, Benjamin Schrauwen
Electronics and Information Systems department (ELIS), Ghent University
{aaron.vandenoord, sander.dieleman, benjamin.schrauwen}@ugent.be



15

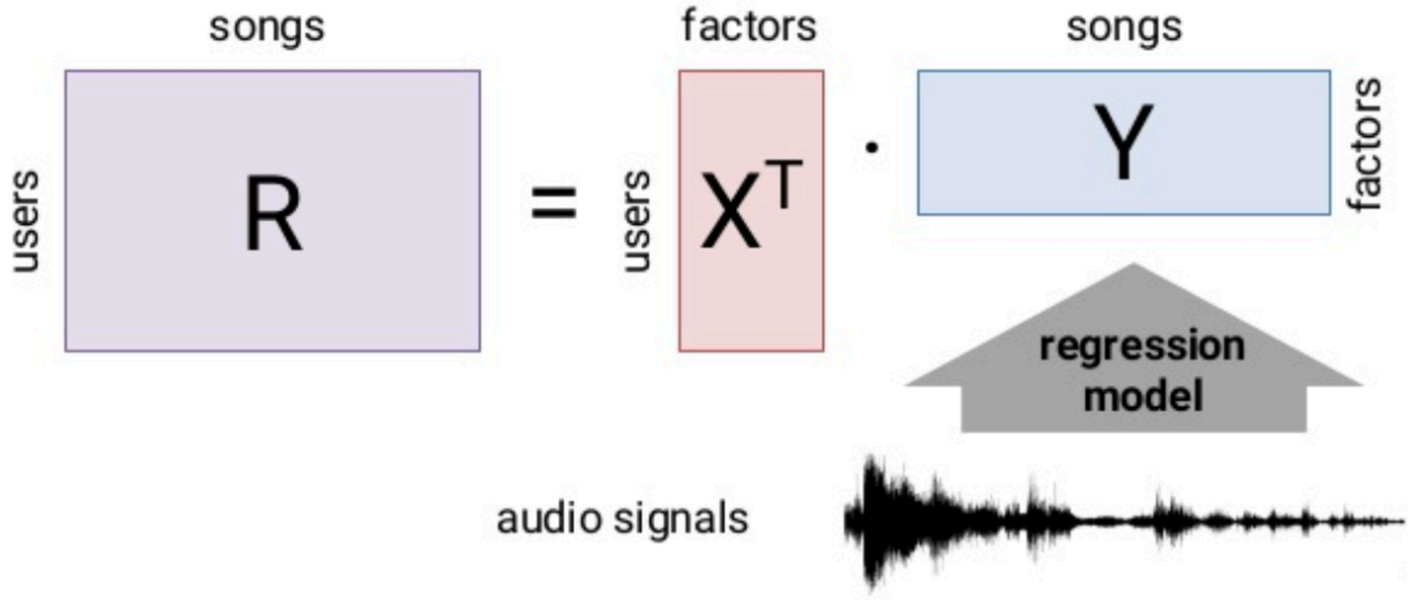
$$
\begin{aligned}
p_{ui} &= I(r_{ui} > 0), \\
c_{ui} &= 1 + \alpha \log(1 + \epsilon^{-1} r_{ui}).
\end{aligned}
$$

$$
\min_{x_\star, y_\star} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u ||x_u||^2 + \sum_i ||y_i||^2 \right)
$$

Deep content-based music recommendation

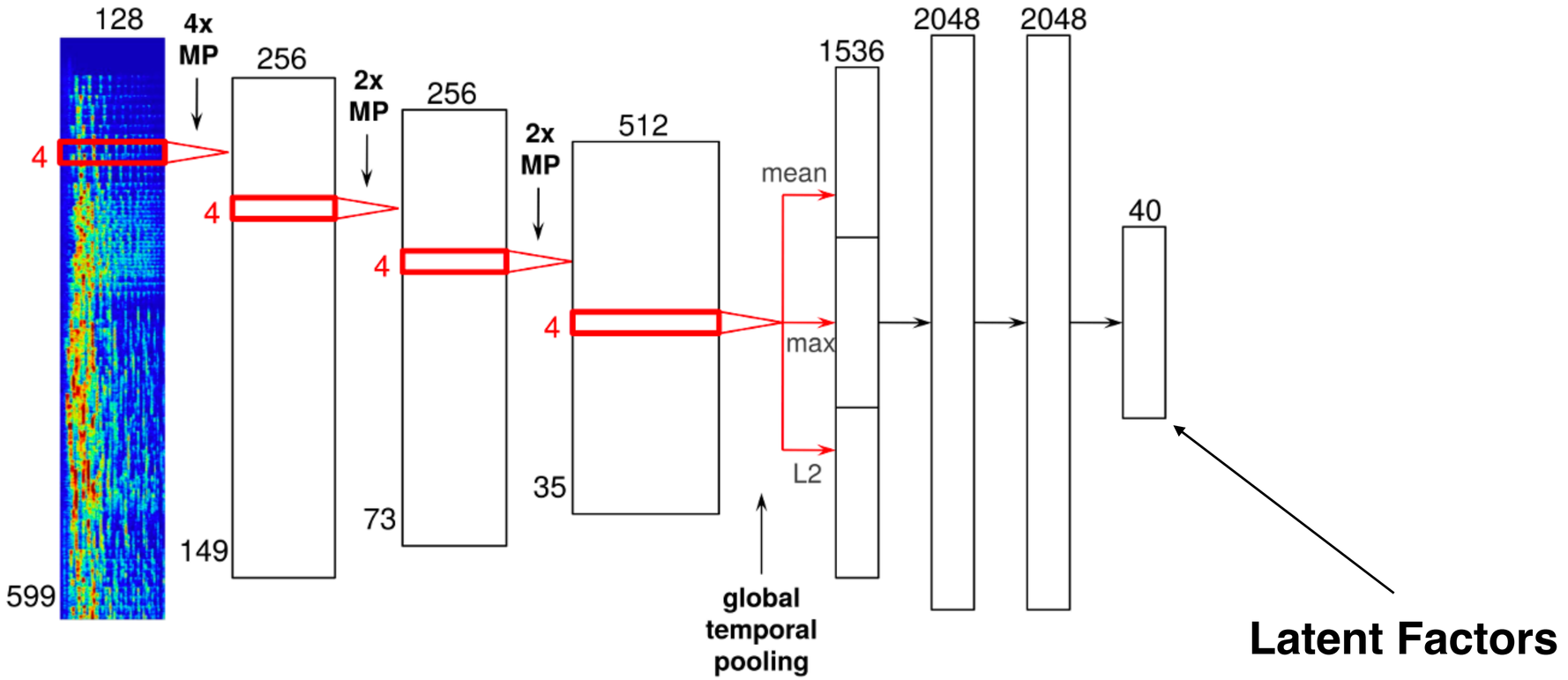
Aäron van den Oord, Sander Dieleman, Benjamin Schrauwen
Electronics and Information Systems department (ELIS), Ghent University
{aaron.vandenoord, sander.dieleman, benjamin.schrauwen}@ugent.be

Solution: Predict the latent factors for those new samples using the audio signal



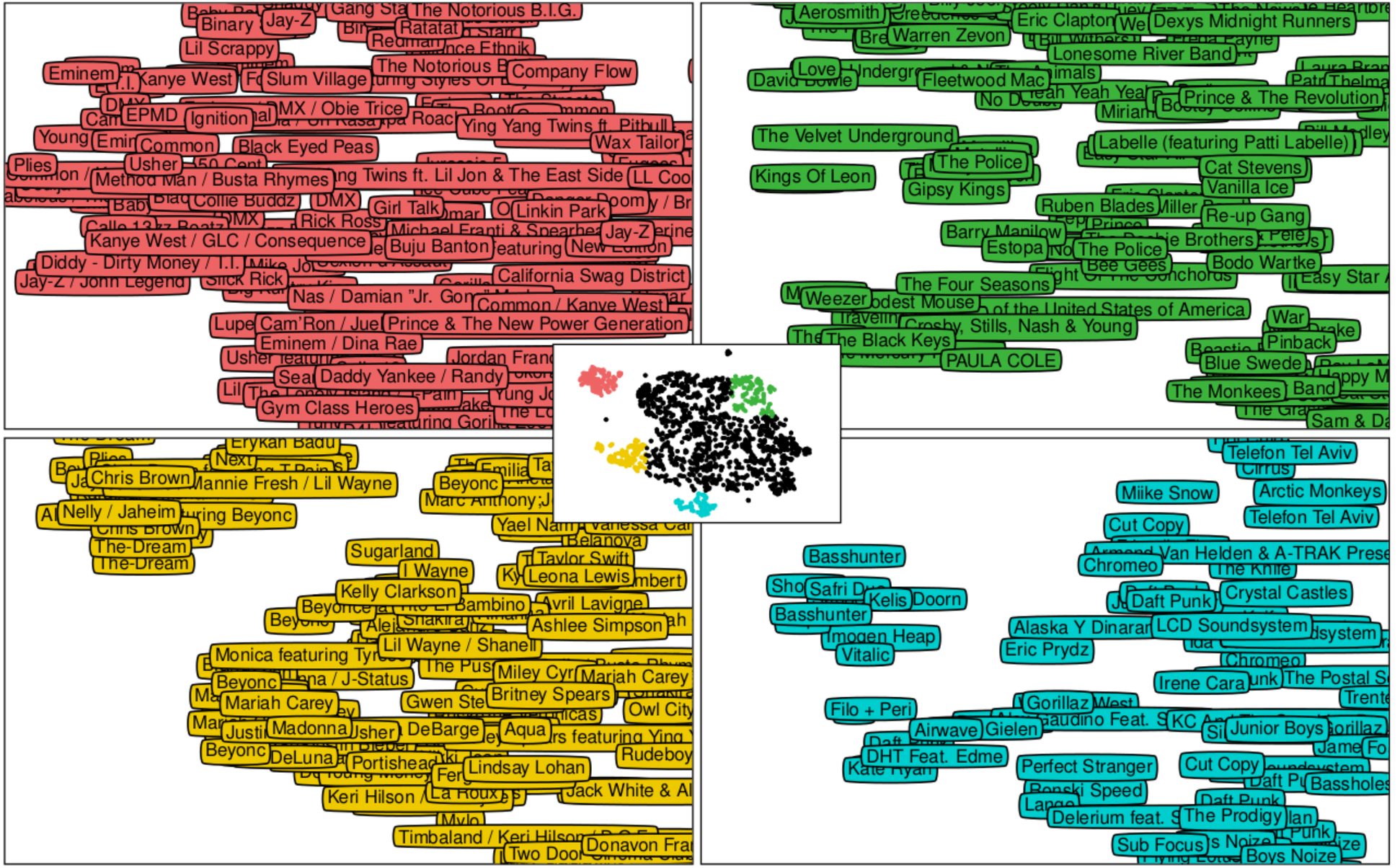
Deep content-based music recommendation

Aäron van den Oord, Sander Dieleman, Benjamin Schrauwen
Electronics and Information Systems department (ELIS), Ghent University
{aaron.vandenoord, sander.dieleman, benjamin.schrauwen}@ugent.be



Latent Factors

Deep content-based music recommendation 356 2013
A Van den Oord, S Dieleman, B Schrauwen
Advances in neural information processing systems, 2643-2651



Deep content-based music recommendation 356 2013
A Van den Oord, S Dieleman, B Schrauwen
Advances in neural information processing systems, 2643-2651

http://benanne.github.io/2014/08/05/spotify-cnns.html

This summer, I'm interning at Spotify in New York City, where I'm working on content-based music recommendation using convolutional neural networks. In this post, I'll explain my approach and show some preliminary results.

Overview

This is going to be a long post, so here's an overview of the different sections. If you want to skip ahead, just click the section title to go there.

CONTENT2VEC: SPECIALIZING JOINT REPRESENTATIONS OF PRODUCT IMAGES AND TEXT FOR THE TASK OF PRODUCT RECOMMENDATION

Thomas Nedelec, Elena Smirnova & Flavian Vasile
Criteo Research
Paris, 32 Blanche, France
{t.nedelec,e.smirnova,f.vasile}@criteo.com

Goal: Create the best product representation

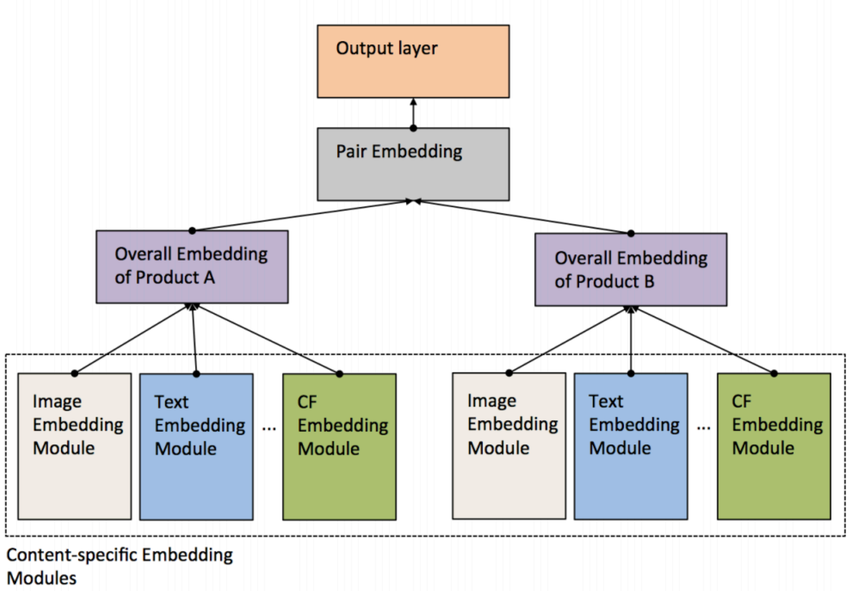


Figure 2: Content2Vec architecture combines content-specific modules with residual vector to produce embedding vector for each product, then uses these vectors to compute similarities between products.

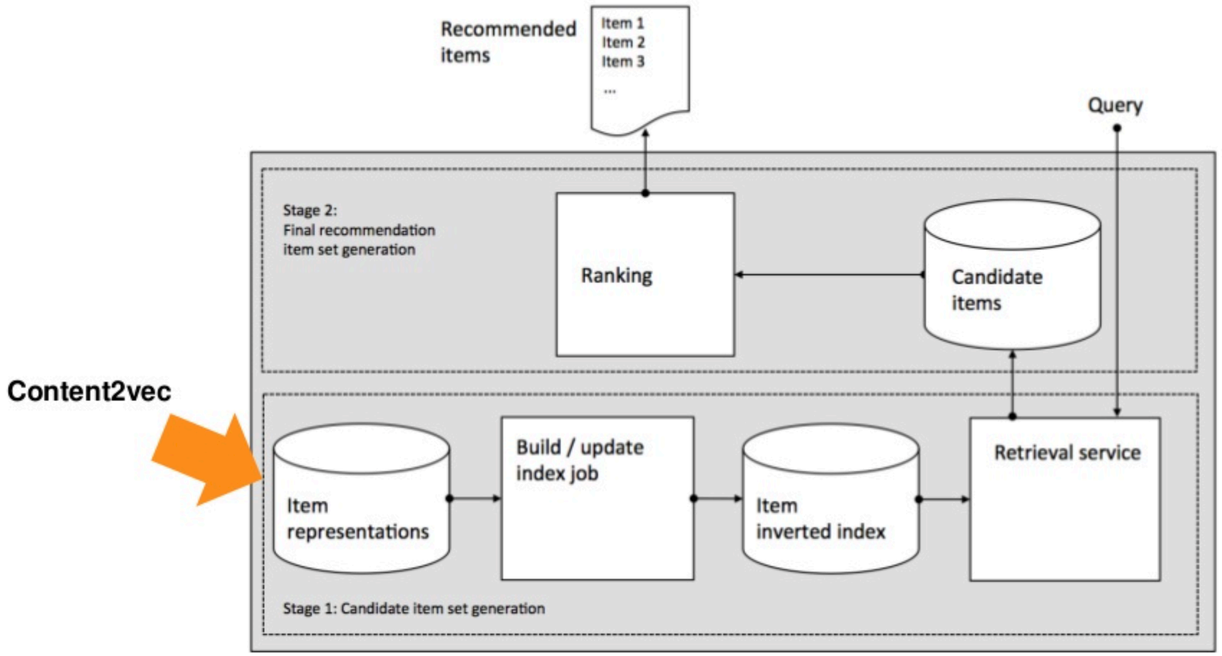
Takes into account all product signals (images, Text,…)

Defines the product-2-product similarity based on the co-occurrence of the products

Siamese loss function

CONTENT2VEC: SPECIALIZING JOINT REPRESENTATIONS OF PRODUCT IMAGES AND TEXT FOR THE TASK OF PRODUCT RECOMMENDATION

Thomas Nedelec, Elena Smirnova & Flavian Vasile
Criteo Research
Paris, 32 Blanche, France
{t.nedelec,e.smirnova,f.vasile}@criteo.com



Deep Neural Networks for YouTube Recommendations

Paul Covington, Jay Adams, Emre Sargin
Google
Mountain View, CA
{pcovington, jka, msargin}@google.com

ABSTRACT

YouTube represents one of the largest scale and most sophisticated industrial recommendation systems in existence. In this paper, we describe the system at a high level and focus on the dramatic performance improvements brought by deep learning. The paper is split according to the classic two-stage information retrieval dichotomy: first, we detail a deep candidate generation model and then describe a separate deep ranking model. We also provide practical lessons and insights derived from designing, iterating and maintaining a massive recommendation system with enormous user-facing impact.

Keywords

recommender system; deep learning; scalability

1. INTRODUCTION

YouTube is the world's largest platform for creating, sharing and discovering video content. YouTube recommendations are responsible for helping more than a billion users discover personalized content from an ever-growing corpus of videos. In this paper we will focus on the immense impact deep learning has recently had on the YouTube video recommendations system. Figure 1 illustrates the recommendations on the YouTube mobile app home.

Recommending YouTube videos is extremely challenging from three major perspectives:

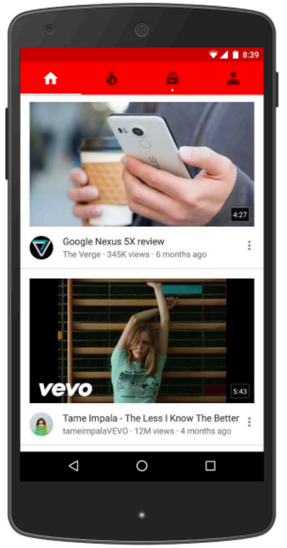


Figure 1: Recommendations displayed on YouTube mobile app home.

with well-established videos can be understood from an exploration/exploitation perspective.

Proceedings of the 10th ACM Conference on Recommender Systems, ACM, New York, NY, USA (2016)

Deep Neural Networks for YouTube Recommendations

Paul Covington, Jay Adams, Emre Sargin
Google
Mountain View, CA
{pcovington, jka, msargin}@google.com

- Divides the recommendation task into **2 stages**: candidate generation and candidate ranking:
  - The **candidate generation network** retrieves a subset from all video corpus.
    - From millions of videos to hundreds
  - The **ranking network generates** a top-n list based on the nearest neighbors' scores from the candidates.

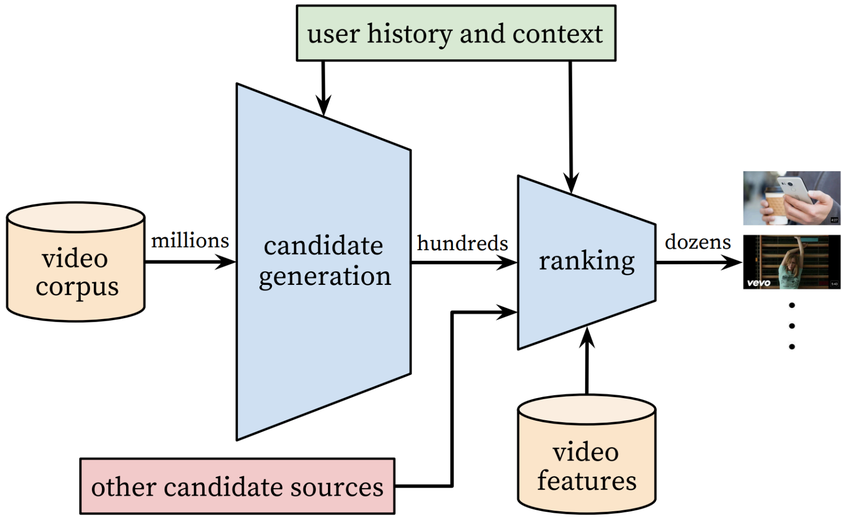


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.

Deep Neural Networks for YouTube Recommendations

Paul Covington, Jay Adams, Emre Sargin
Google
Mountain View, CA
{pcovington, jka, msargin}@google.com

Candidate Generation

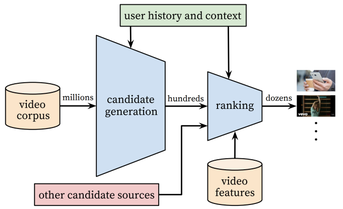


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.

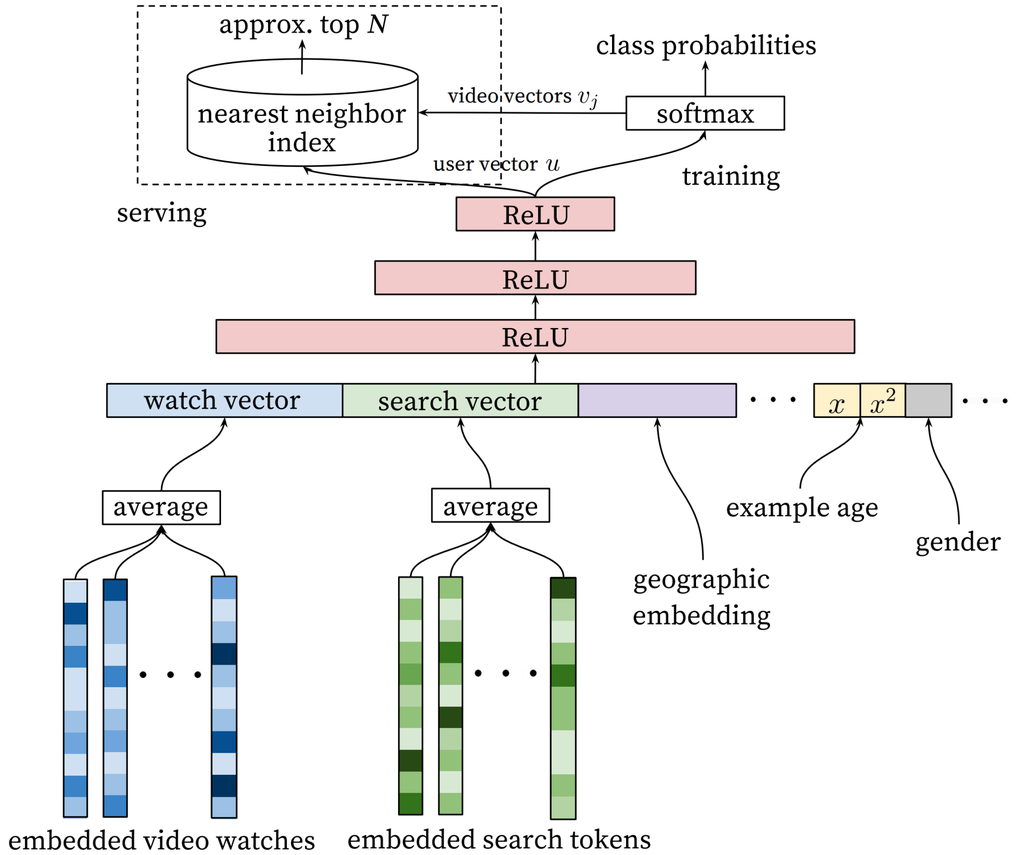


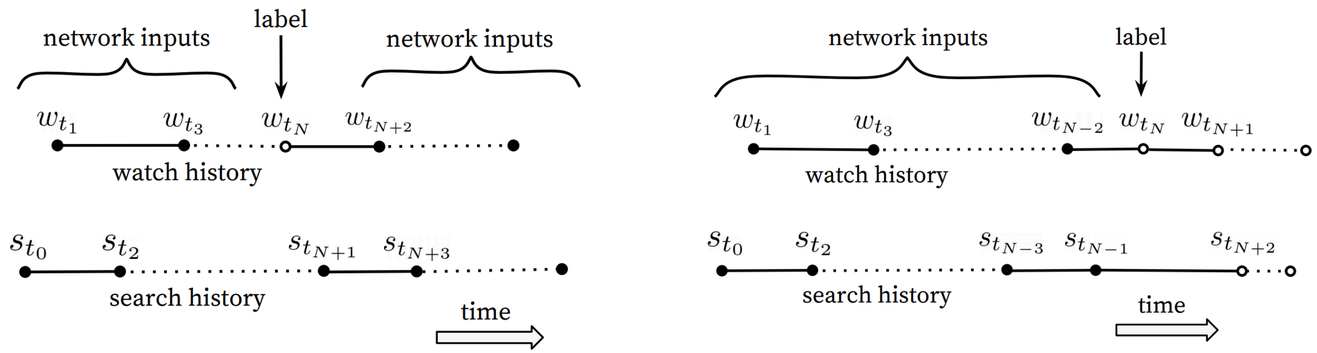
Figure 3: Deep candidate generation model architecture showing embedded sparse features concatenated with dense features. Embeddings are averaged before concatenation to transform variable sized bags of sparse IDs into fixed-width vectors suitable for input to the hidden layers. All hidden layers are fully connected. In training, a cross-entropy loss is minimized with gradient descent on the output of the sampled softmax. At serving, an approximate nearest neighbor lookup is performed to generate hundreds of candidate video recommendations.

Deep Neural Networks for YouTube Recommendations

Paul Covington, Jay Adams, Emre Sargin
Google
Mountain View, CA
{pcovington, jka, msargin}@google.com

Candidate Generation

Training Data Generation



(a) Predicting held-out watch

(b) Predicting future watch

Figure 5: Choosing labels and input context to the model is challenging to evaluate offline but has a large impact on live performance. Here, solid events • are input features to the network while hollow events ∘ are excluded. We found predicting a future watch (5b) performed better in A/B testing. In (5b), the example age is expressed as $t_{\max} - t_N$ where $t_{\max}$ is the maximum observed time in the training data.

Deep Neural Networks for YouTube Recommendations

Paul Covington, Jay Adams, Emre Sargin
Google
Mountain View, CA
{pcovington, jka, msargin}@google.com

Ranking Generation

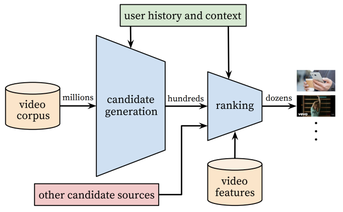


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.

Figure 7: Deep ranking network architecture depicting embedded categorical features (both univalent and multivalent) with shared embeddings and powers of normalized continuous features. All layers are fully connected. In practice, hundreds of features are fed into the network.

Deep Neural Networks for YouTube Recommendations

Paul Covington, Jay Adams, Emre Sargin
Google
Mountain View, CA
{pcovington, jka, msargin}@google.com

Ranking Generation

- Use impression data to specialize and calibrate candidate predictions for the particular user interface.
  - e.g. A user may watch a given video with hight probability but is unlikely to click on the specific homepage impression due to the chosen thumbnail image
- Deep neural network with similar architecture as candidate generation to assign an **independent score** to each video impression **using logistic regression**
- The model is trained with logistic regression under cross-entropy loss. However, the positive (clicked) impressions are weighted by the observed watch time on the video

DNN + FM (Hybrid) approach

Neural Collaborative Filtering·

Xiangnan He
National University of Singapore, Singapore
xiangnanhe@gmail.com

Lizi Liao
National University of Singapore, Singapore
liaolizi.llz@gmail.com

Hanwang Zhang
Columbia University
USA
hanwangzhang@gmail.com

Liqiang Nie
Shandong University
China
nieliqiang@gmail.com

Xia Hu
Texas A&M University
USA
hu@cse.tamu.edu

Tat-Seng Chua
National University of Singapore, Singapore
dcscts@nus.edu.sg

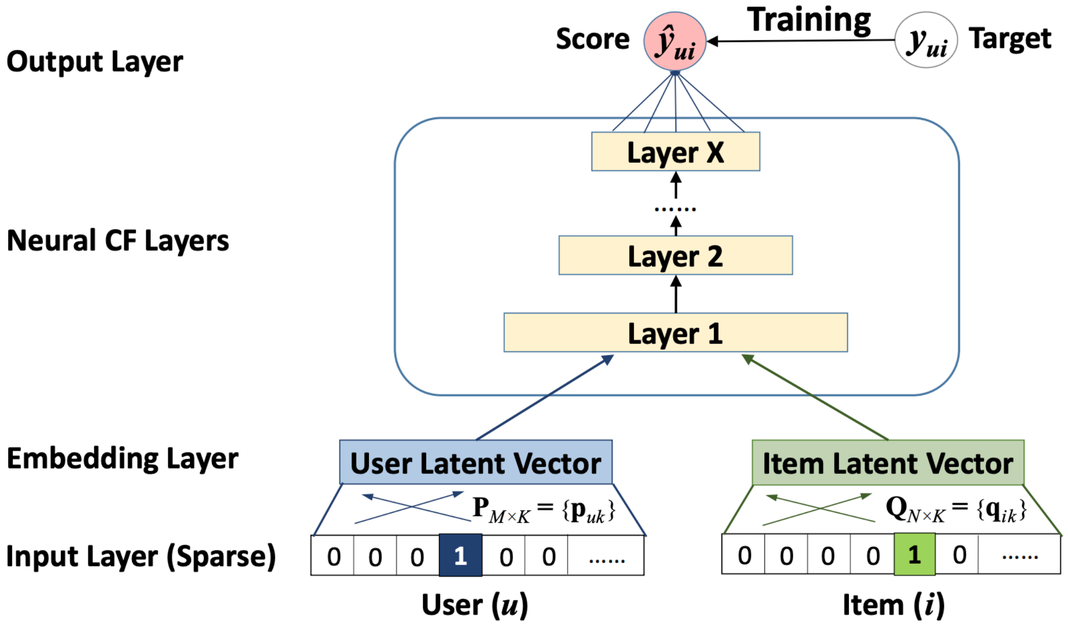


Figure 2: Neural collaborative filtering framework

Neural Collaborative Filtering·

Xiangnan He
National University of Singapore, Singapore
xiangnanhe@gmail.com

Lizi Liao
National University of Singapore, Singapore
liaolizi.llz@gmail.com

Hanwang Zhang
Columbia University
USA
hanwangzhang@gmail.com

Liqiang Nie
Shandong University
China
nieliqiang@gmail.com

Xia Hu
Texas A&M University
USA
hu@cse.tamu.edu

Tat-Seng Chua
National University of Singapore, Singapore
dcscts@nus.edu.sg

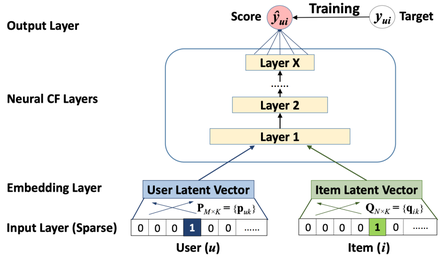


Figure 2: Neural collaborative filtering framework

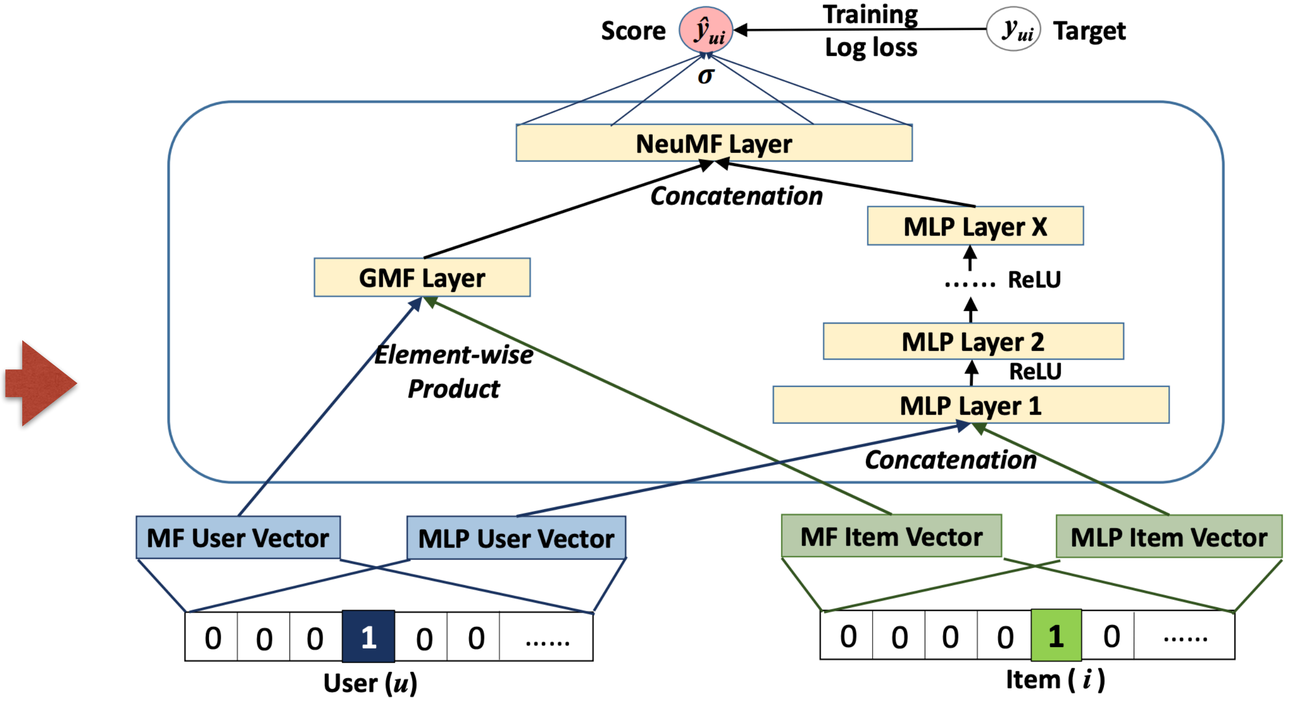


Figure 3: Neural matrix factorization model

Neural Collaborative Filtering

Xiangnan He
National University of Singapore, Singapore
xiangnanhe@gmail.com

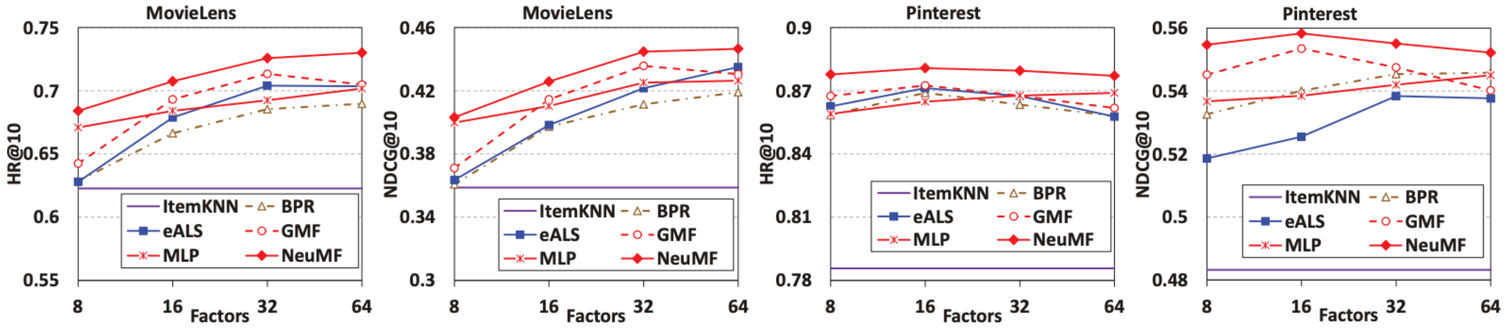
Lizi Liao
National University of Singapore, Singapore
liaolizi.llz@gmail.com

Hanwang Zhang
Columbia University
USA
hanwangzhang@gmail.com

Liqiang Nie
Shandong University
China
nieliqiang@gmail.com

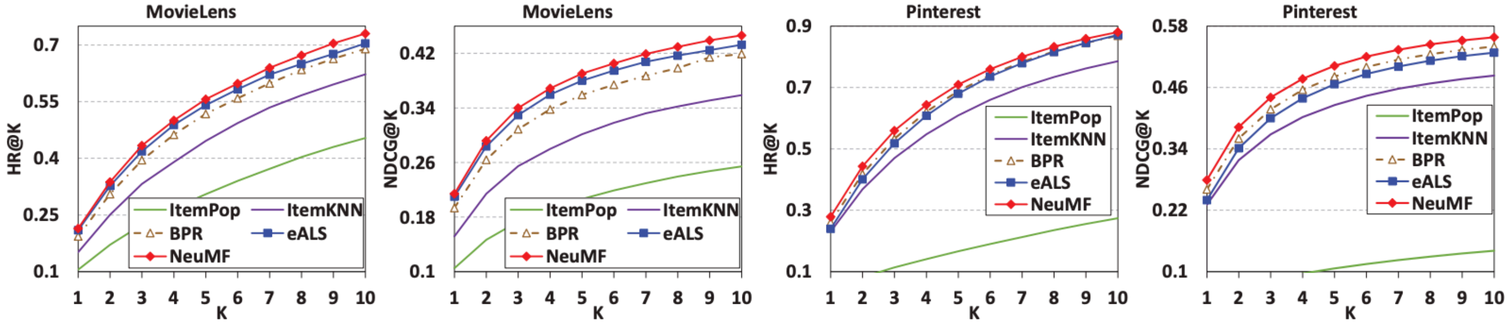
Xia Hu
Texas A&M University
USA
hu@cse.tamu.edu

Tat-Seng Chua
National University of Singapore, Singapore
dcscts@nus.edu.sg



(a) MovieLens — HR@10 (b) MovieLens — NDCG@10 (c) Pinterest — HR@10 (d) Pinterest — NDCG@10

Figure 4: Performance of HR@10 and NDCG@10 *w.r.t.* the number of predictive factors on the two datasets.



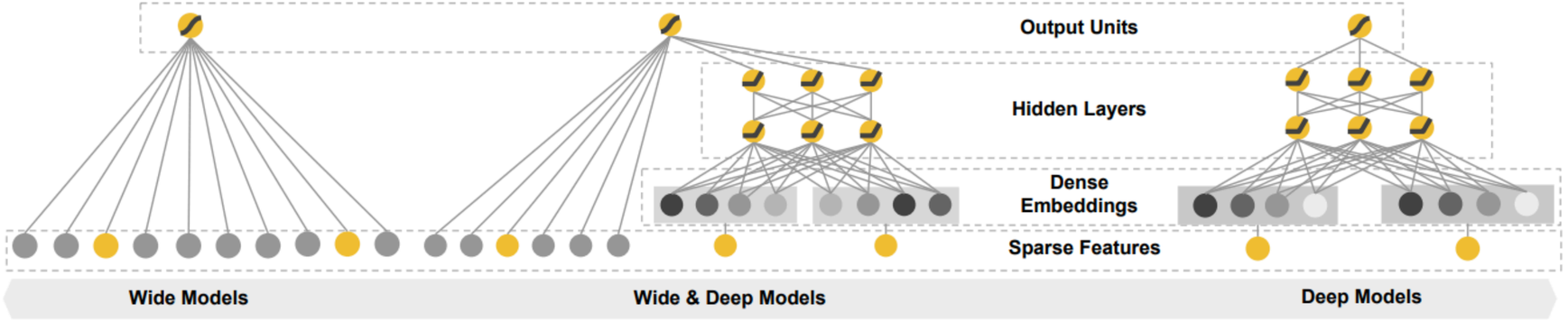
(a) MovieLens — HR@K (b) MovieLens — NDCG@K (c) Pinterest — HR@K (d) Pinterest — NDCG@K

Figure 5: Evaluation of Top-K item recommendation where K ranges from 1 to 10 on the two datasets.

IBPR - Bayesian Personalized Ranking
eALS - *state-of-the-art* MF
MLP - Multi-Layer Perceptron
GMP - Generalized Matrix Factorization
NeufMF - Neural Matrix Factorization

Wide & Deep Learning for Recommender Systems

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, Hemal Shah
Google Inc.*



Memorisation vs **Generalisation**

Memorisation can be loosely defined as learning the frequent co-occurrence of items or features and exploiting the correlation available in the historical data. **Generalisation**, on the other hand, is based on transitivity of correlation and explores new feature combinations that have never or rarely occurred in the past.

Wide & Deep Learning for Recommender Systems

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, Hemal Shah

Google Inc.*

Trains a **wide linear model** and a **deep neural network**

Memorization

Recommendations based on memorisation are usually more topical and directly relevant to the items on which users have already performed actions.

Generalization

Generalisation tends to improve the diversity of the recommended items. Generalisation can be added by using features that are less granular , but manual feature engineering is often required.

Wide & Deep Learning for Recommender Systems

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, Hemal Shah
Google Inc.*

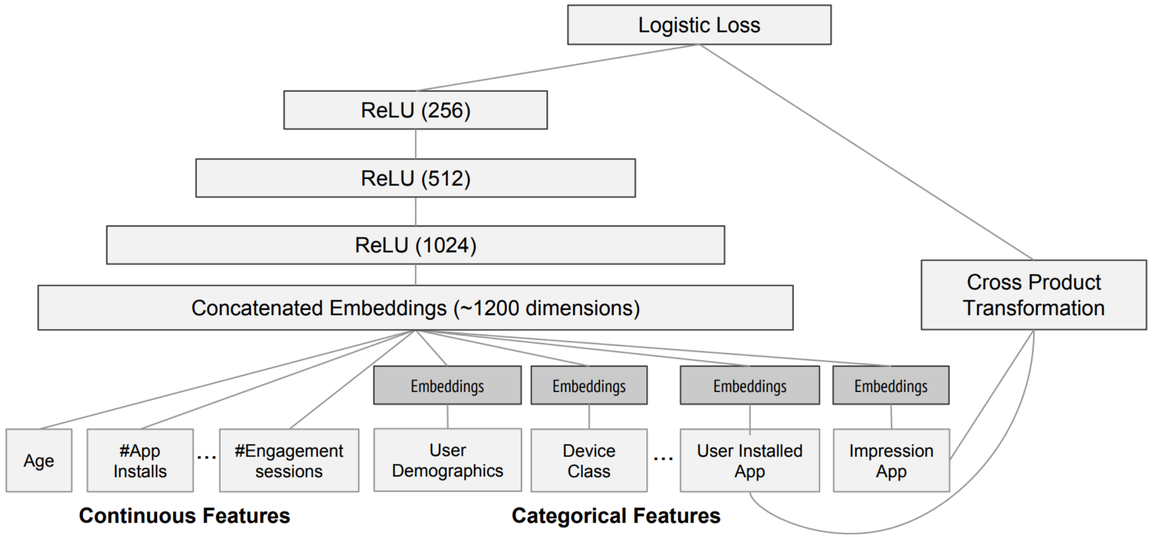


Figure 4: Wide & Deep model structure for apps recommendation.

Wide & Deep Learning for Recommender Systems

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, Hemal Shah
Google Inc.*

- Training:
  - 500 billions of examples

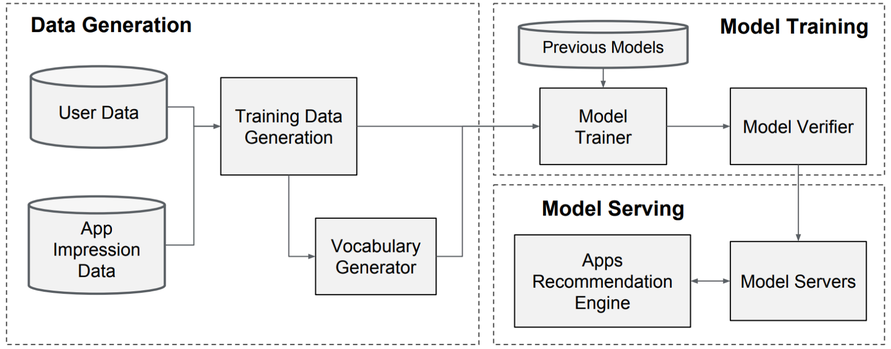
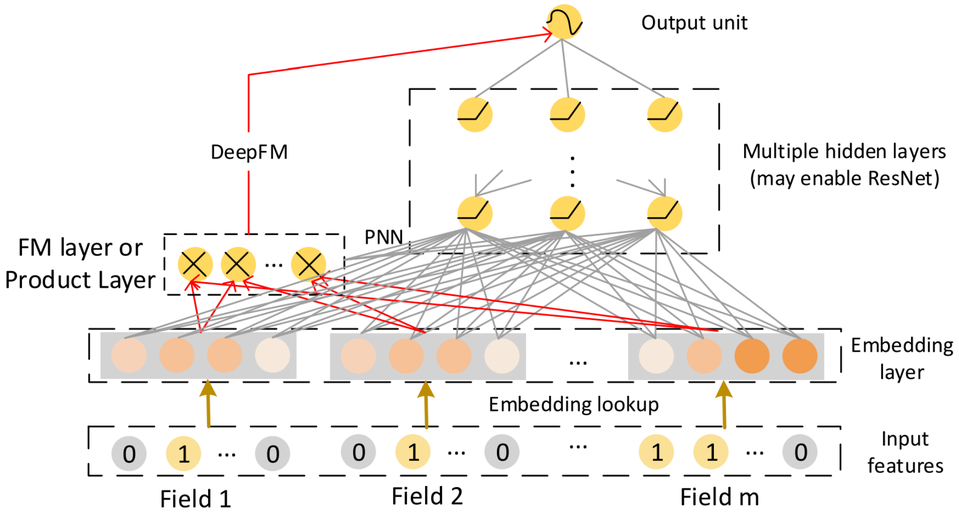


Figure 3: Apps recommendation pipeline overview.

DeepFM: A Factorization-Machine based Neural Network for CTR Prediction

Huifeng Guo[*1] **, Ruiming Tang**[2]**, Yunming Ye**[†1]**, Zhenguo Li**[2]**, Xiuqiang He**[2]
[1]Shenzhen Graduate School, Harbin Institute of Technology, China
[2]Noah's Ark Research Lab, Huawei, China
[1]huifengguo@yeah.net, yeyunming@hit.edu.cn
[2]{tangruiming, li.zhenguo, hexiuqiang}@huawei.com



As an extension of the Wide and Deep Learning approach and that integrates **Factorization Machine** (the wide component) and **Multi-Layer Perceptron** (the deep component).
DeepFM does not require tedious **feature engineering.**

xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems

Jianxun Lian
University of Science and Technology of China
jianxun.lian@outlook.com

Xiaohuan Zhou
Beijing University of Posts and Telecommunications
maggione@bupt.edu.cn

Fuzheng Zhang
Microsoft Research
fuzzhang@microsoft.com

Zhongxia Chen
University of Science and Technology of China
czx87@mail.ustc.edu.cn

Xing Xie
Microsoft Research
xingx@microsoft.com

Guangzhong Sun
University of Science and Technology of China
gzsun@ustc.edu.cn

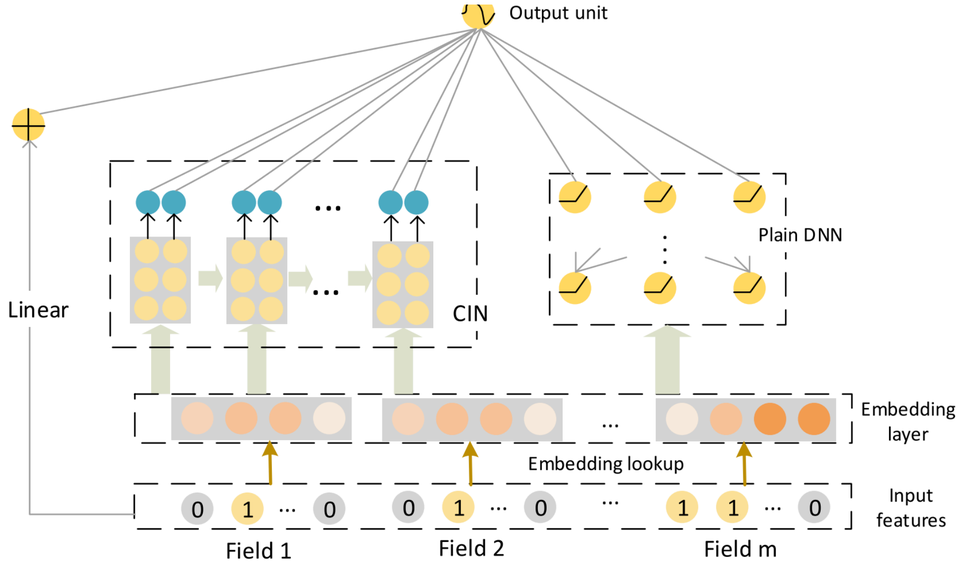


Figure 5: The architecture of xDeepFM.

- It learns feature interactions at a vector-wise level.

https://towardsdatascience.com/extreme-deep-factorization-machine-xdeepfm-1ba180a6de78

Neural Factorization Machines for Sparse Predictive Analytics*

Xiangnan He
School of Computing
National University of Singapore
Singapore 117417
dcshex@nus.edu.sg

Tat-Seng Chua
School of Computing
National University of Singapore
Singapore 117417
dcscts@nus.edu.sg

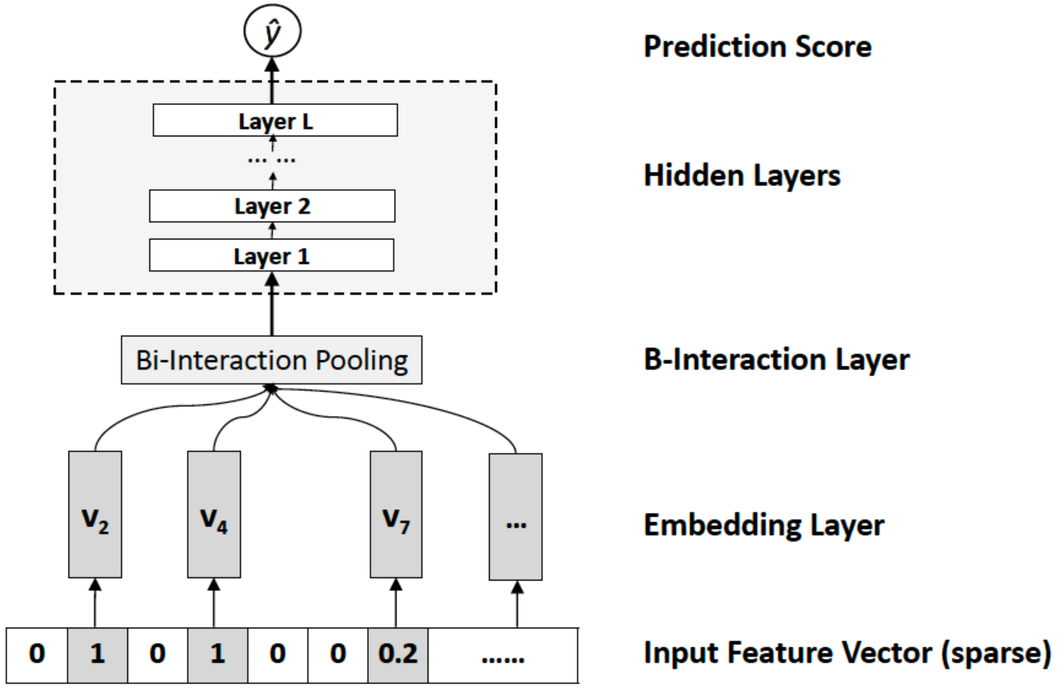


Figure 2: Neural Factorization Machines model (the first-order linear regression part is not shown for clarity).

Autoencoder based

Deep Collaborative Filtering via Marginalized Denoising Auto-encoder

Sheng Li*
Northeastern University
Boston, MA, USA
shengli@ece.neu.edu

Jaya Kawale
Adobe Research
San Jose, CA, USA
kawale@adobe.com

Yun Fu
Northeastern University
Boston, MA, USA
yunfu@ece.neu.edu

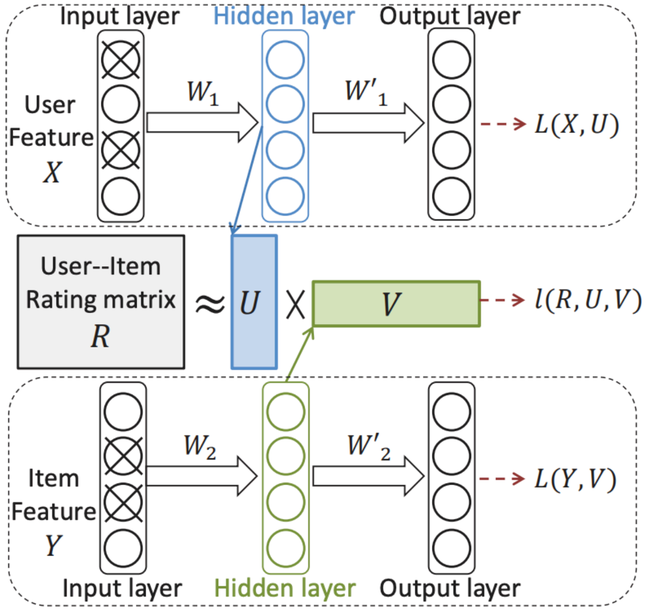


Figure 1: Illustration of DCF framework. The inputs are user-item rating matrix R, the user feature set X and the item feature set Y. Our approach jointly decomposes R and learns latent factors (i.e., U, V) from ratings and side information (i.e., X and Y). In particular, the latent factors are extracted from the hidden layer of deep networks.

Restricted Boltzmann Machines for Collaborative Filtering

Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Andriy Mnih AMNIH@CS.TORONTO.EDU
Geoffrey Hinton HINTON@CS.TORONTO.EDU
University of Toronto, 6 King's College Rd., Toronto, Ontario M5S 3G4, Canada

Abstract

Most of the existing approaches to collaborative filtering cannot handle very large data sets. In this paper we show how a class of two-layer undirected graphical models, called Restricted Boltzmann Machines (RBM's), can be used to model tabular data, such as user's ratings of movies. We present efficient learning and inference procedures for this class of models and demonstrate that RBM's can be successfully applied to the Netflix data set, containing over 100 million user/movie ratings. We also show that RBM's slightly outperform carefully-tuned SVD models. When the predictions of multiple RBM models and multiple SVD models are linearly combined, we achieve an error rate that is well over 6% better than the score of Netflix's own system.

Low-rank approximations based on minimizing the sum-squared distance can be found using Singular Value Decomposition (SVD). In the collaborative filtering domain, however, most of the data sets are sparse, and as shown by Srebro and Jaakkola (2003), this creates a difficult non-convex problem, so a naive solution is not going work.[1]

In this paper we describe a class of two-layer undirected graphical models that generalize Restricted Boltzmann Machines to modeling tabular or count data (Welling et al., 2005). Maximum likelihood learning is intractable in these models, but we show that learning can be performed efficiently by following an approximation to the gradient of a different objective function called "Contrastive Divergence" (Hinton, 2002).

2. Restricted Boltzmann Machines (RBM's)

SESSION-BASED RECOMMENDATIONS WITH RECURRENT NEURAL NETWORKS

Balázs Hidasi *
Gravity R&D Inc.
Budapest, Hungary
balazs.hidasi@gravityrd.com

Alexandros Karatzoglou
Telefonica Research
Barcelona, Spain
alexk@tid.es

Linas Baltrunas †
Netflix
Los Gatos, CA, USA
lbaltrunas@netflix.com

Domonkos Tikk
Gravity R&D Inc.
Budapest, Hungary
domonkos.tikk@gravityrd.com