Нейронные сети в рекомендациях

Введение

В этом уроке поговорим об использовании нейронных сетей для рекомендательных систем.

План

1. Понятие эмбеддинг
2. Neural Collaborative Filtering
3. General Matrix Factorization
4. Базовая архитектура нейронной сети

Понятие эмбеддинг

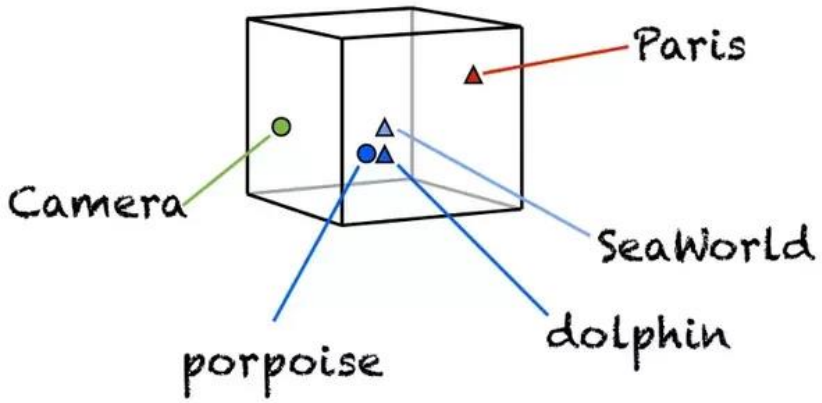
Векторные представления товаров

Embedding (эмбеддинг) – это способ представлять объекты, когда каждый объект превращается в вектор фиксированной длины, и близким объектам соответствуют близкие векторы. Практически всем известным моделям требуется, чтобы данные на входе были фиксированной длины, и набор векторов — простой способ привести их к такому виду.

Эмбеддинг можно получать из чего угодно:
Из текстов — **word2vec.**
Фото – последний слой нейронной сети для задачи классификации.
Из характеристик – например **PCA** (метод главных компонент).

Embedding (векторное представление)

title	dim-0	dim-1	dim-2	dim-3	dim-4	...	dim-45	dim-46	dim-47	dim-48	dim-49
War and Peace	-0.279165	-0.107367	0.114153	0.143709	-0.141921	...	-0.067178	0.230711	-0.230550	0.199285	-0.099167
Anna Karenina	-0.248443	-0.000578	0.150472	0.151845	0.000908	...	-0.141615	0.178011	-0.230794	0.042102	-0.189196
The Hitchhiker's Guide to the Galaxy (novel)	-0.190761	-0.060406	0.115548	-0.249868	-0.120824	...	-0.038944	0.084992	-0.047035	-0.054157	-0.209883



Neural Collaborative Filtering

Neural Collaborative Filtering

В 2017-м группа исследователей опубликовала работу о **Нейронной Коллаборативной Фильтрации (NCF)**. Она содержит обобщенный фреймворк для изучения нейронных зависимостей, моделируемых факторизацией матриц при коллаборативной фильтрации с помощью нейронной сети. Авторы также объяснили, как получить зависимости высшего порядка (MF имеет порядок всего лишь 2), и как объединить оба эти подхода.

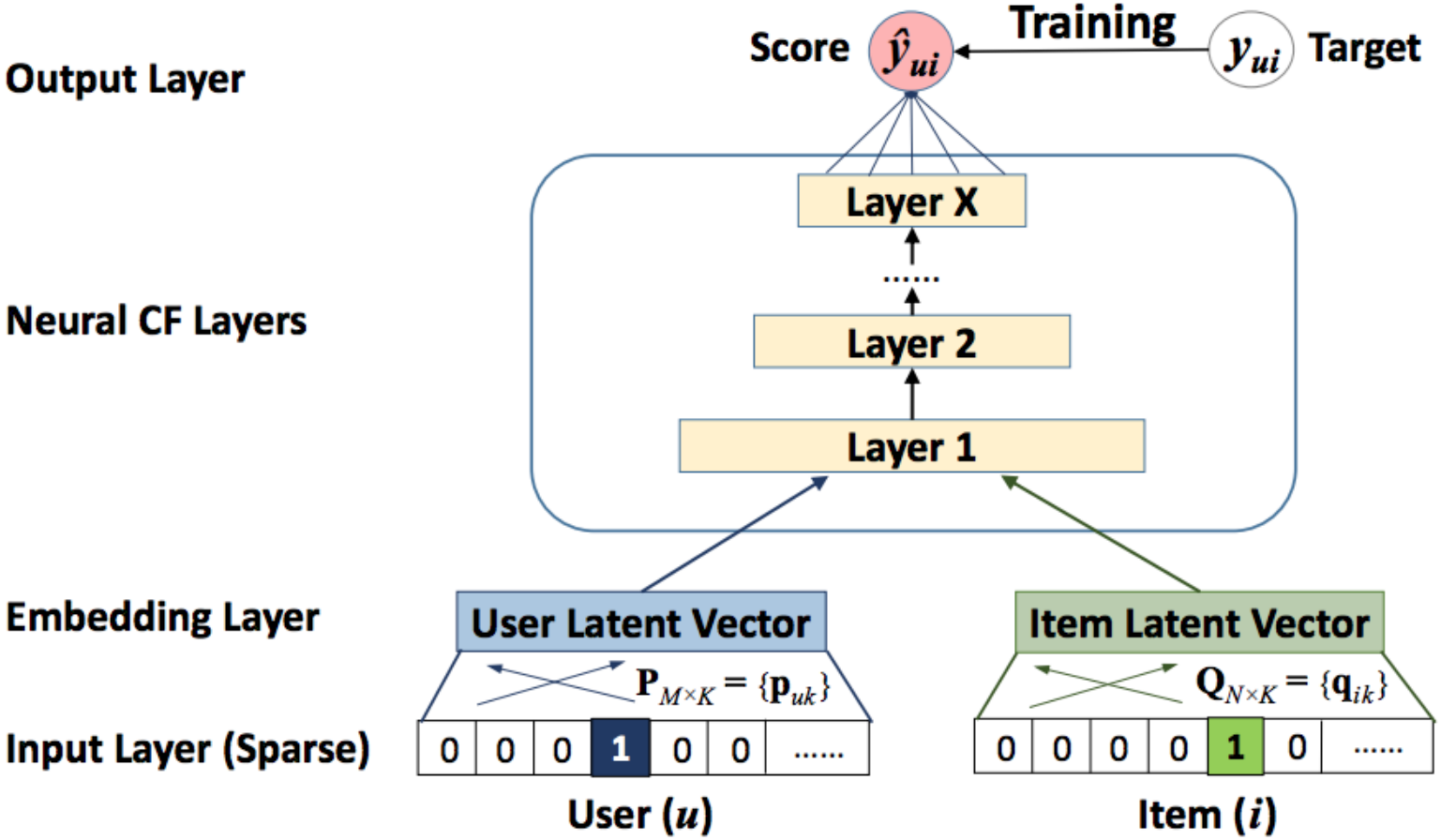
https://arxiv.org/pdf/1708.05031.pdf

Neural Collaborative Filtering

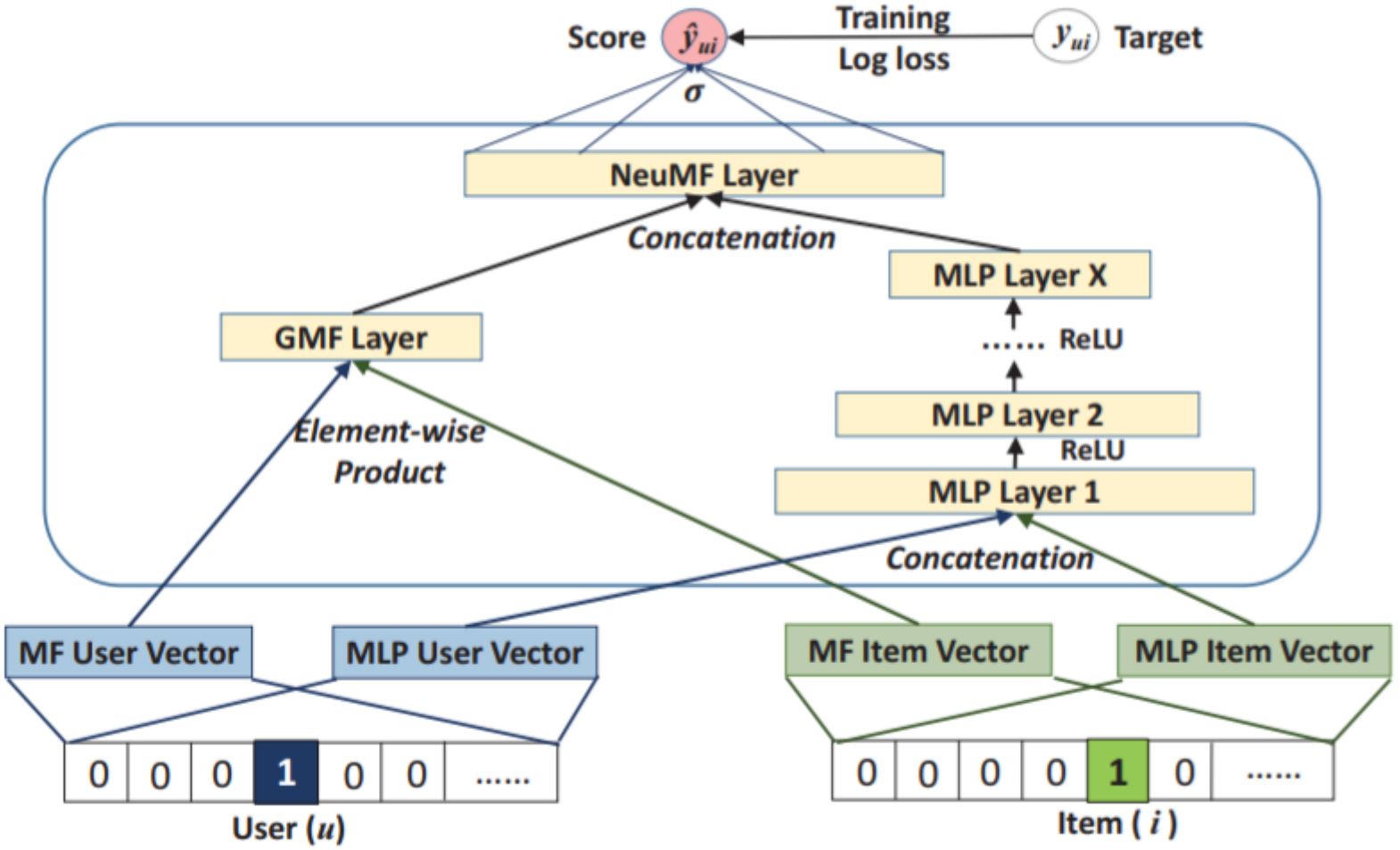
Нейронная сеть может (теоретически) усвоить любую функциональную зависимость. Это значит, что зависимость, которую модель коллаборативной фильтрации выражает матричной факторизацией, может быть усвоена нейронной сетью.

Идея: NCF предлагает простой слой представления сразу для пользователей и объектов, за которым следует простая нейронная сеть вроде многослойного перцептрона, которая должна усвоить зависимость между представлениями пользователя и объекта, аналогичную произведению факторизованных матриц.

Neural Collaborative Filtering



Neural Collaborative Filtering



Neural Collaborative Filtering

Преимущество подхода – нелинейность многослойного перцептрона. Простое произведение матриц, используемое при матричной факторизации, всегда будет ограничивать модель взаимодействиями 2-й степени, тогда как нейронная сеть с X слоями в теории может усвоить взаимодействия гораздо более высоких степеней.

Generalized Matrix Factorization

Generalized Matrix Factorization

Первый нейронный CF-слой:

$$\phi_1(p_u, q_i) = p_u \times q_i$$

Generalized Matrix Factorization

Поскольку NCF использует два пути для моделирования пользователей и элементов, интуитивно понятно объединить функции двух путей путем их конкатенации. Однако простая конкатенация векторов не учитывает никаких взаимодействий между скрытыми функциями пользователя и элемента, что недостаточно для моделирования эффекта совместной фильтрации. Чтобы решить эту проблему, предлагается добавить скрытые слои в объединенный вектор, используя стандартный MLP для изучения взаимодействия между пользователем и скрытыми функциями элемента.

Полносвязная нейронная сеть

$$z_1 = \phi_1(p_u, q_i) = \begin{bmatrix} p_u \\ q_i \end{bmatrix},$$

$$\phi_2\,(z_1) = a_2(W_2^T z_1 + b_2),$$

… …

$$\phi_L(z_L - 1) = a_L(W_L^T z_{L-1} + b_L)$$

$$\hat{y}_{ui} = \sigma(h^T \phi_L(z_{L-1}))$$

Generalized Matrix Factorization

Модель для объединения GMF с однослойным MLP:

$$\hat{y}_{ui} = \sigma\left(h^T a\left(p_u \times q_i + W\begin{bmatrix} p_u \\ q_i \end{bmatrix} + b\right)\right)$$

Generalized Matrix Factorization

Объединение GMF и MLP последним скрытым слоем

$$\phi^{GMF} = p_u^G \times q_i^G$$

$$\phi^{MLP} = a_L\left(W_L^T\left(a_{L-1}\left(\ldots a_2\left(W_2^T\begin{bmatrix}p_u^M\\ q_i^M\end{bmatrix} + b_2\right)\ldots\right)\right) + b_L\right)$$

$$\hat{y}_{ui} = \sigma\left(h^T\begin{bmatrix}\phi^{GMF}\\ \phi^{MLP}\end{bmatrix}\right)$$

Метрика HR@k

Hit Ratio или HR или HR@K.

Для каждого пользователя HR@K соответствует тому, относится ли тестовый элемент к top-K элементам этого пользователя.

$$\mathbf{HR@K} = \begin{cases} \mathbf{1}, & \textbf{если тестовый элемент находится в top-K} \\ \mathbf{0}, & \text{иначе} \end{cases}$$

Производительность разных моделей на 3-х датасетах

Ciao	HR@50	HR@100	HR@200	NDCG@50	NDCG@100	NDCG@200	RI
MP	0.1047	0.1384	0.1776	0.0396	0.0452	0.0506	+67.96%
ItemKNN	0.1453	0.1884	0.2468	0.0497	0.0581	0.0668	+26.14%
BPR	0.1531	0.1930	0.2558	0.0517	0.0598	0.0685	+21.91%
WMF	0.1587	0.2011	0.2608	0.0562	0.0631	0.0714	+16.40%
ExpoMF	0.1602	0.1994	0.2613	0.0569	0.0626	0.0709	+16.41%
GMF	0.1668	0.2103	0.2674	0.0633	0.0687	0.0752	+9.36%
NCF	0.1651	0.2108	0.2712	0.0629	0.0695	0.0764	+8.84%
ConvNCF	0.1682	0.2237	0.2741	0.0641	0.0714	0.0787	+5.90%
ENMF-U	0.1750**	0.2296**	0.2945**	0.0651**	0.0741**	0.0830**	–
ENMF-I	0.1749**	0.2311**	0.2946**	0.0643*	0.0734**	0.0823**	–
ENMF-A	0.1757**	0.2331**	0.3015**	0.0662**	0.0753**	0.0850**	–

Производительность разных моделей на 3-х датасетах

Epinion	HR@50	HR@100	HR@200	NDCG@50	NDCG@100	NDCG@200	RI
MP	0.0661	0.1068	0.1659	0.0234	0.0299	0.0382	+153.96%
ItemKNN	0.1312	0.2082	0.2929	0.0455	0.0563	0.0682	+34.41%
BPR	0.1708	0.2338	0.3007	0.0548	0.0646	0.0747	+17.04%
WMF	0.1765	0.2384	0.3158	0.0605	0.0685	0.0789	+11.07%
ExpoMF	0.1784	0.2368	0.3064	0.0602	0.0691	0.0781	+11.70%
GMF	0.1811	0.2513	0.3388	0.0613	0.0739	0.0845	+5.52%
NCF	0.1816	0.2534	0.3442	0.0621	0.0750	0.0869	+4.08%
ConvNCF	0.1833	0.2510	0.3418	0.0617	0.0742	0.0851	+4.87%
ENMF-U	0.1893**	0.2647**	0.3523**	0.0639**	0.0761**	0.0883**	–
ENMF-I	0.1888**	0.2667**	0.3534**	0.0634**	0.0759**	0.0884**	–
ENMF-A	0.1911**	0.2688**	0.3546**	0.0648**	0.0773**	0.0893**	–

Производительность разных моделей на 3-х датасетах

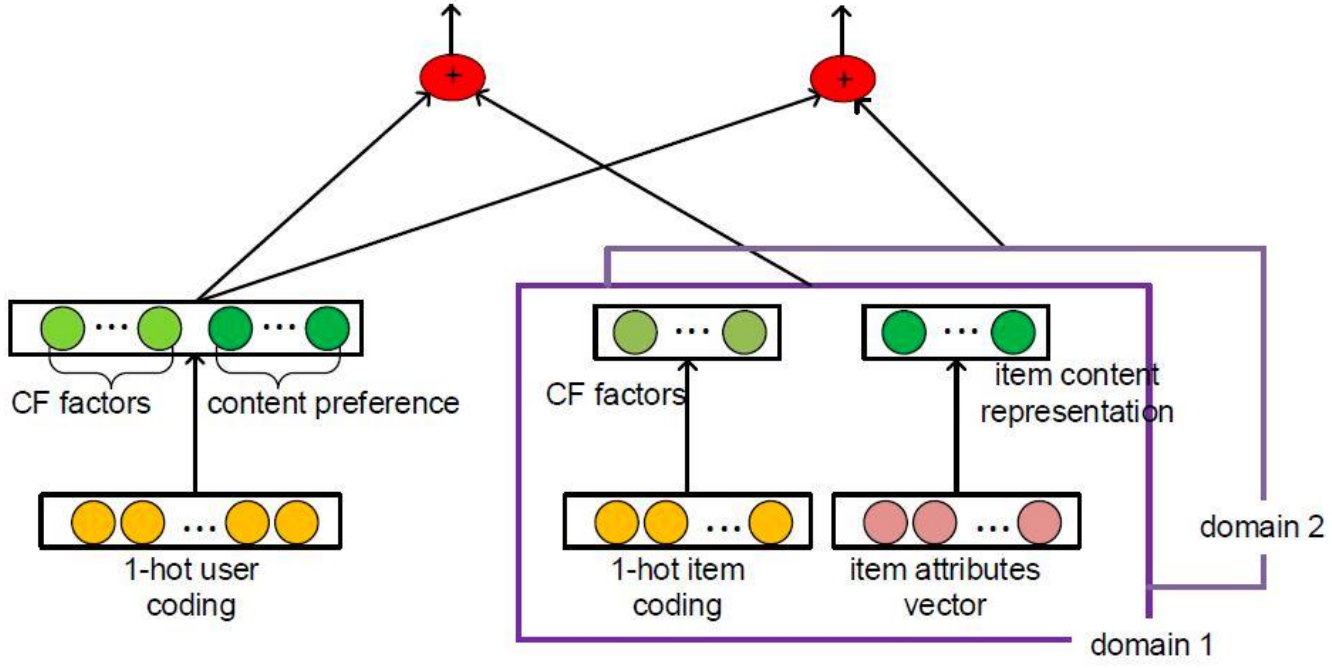
Movielens	HR@50	HR@100	HR@200	NDCG@50	NDCG@100	NDCG@200	RI
MP	0.1842	0.2099	0.3382	0.0441	0.0481	0.0659	+109.01%
ItemKNN	0.2101	0.2889	0.3918	0.0598	0.0724	0.0867	+59.18%
BPR	0.2637	0.4048	0.5710	0.0757	0.0986	0.1217	+17.59%
WMF	0.2924	0.4378	0.6040	0.0909	0.1073	0.1324	+6.47%
ExpoMF	0.2904	0.4368	0.5927	0.0865	0.1100	0.1346	+7.11%
GMF	0.2847	0.4226	0.5847	0.0821	0.1086	0.1289	+10.30%
NCF	0.2902	0.4316	0.6023	0.0837	0.1097	0.1324	+8.02%
ConvNCF	0.2943	0.4403	0.6017	0.0872	0.1112	0.1333	+6.30%
ENMF-U	0.3117**	0.4574**	0.6092**	0.0962**	0.1198**	0.1410**	–
ENMF-I	0.3105**	0.4576**	0.6107**	0.0956**	0.1194**	0.1398**	–
ENMF-A	0.3124**	0.4581**	0.6139**	0.0968**	0.1202**	0.1419**	–

Базовая архитектура нейронной сети

Cross-domain Content-boosted CF

Cross-domain Content-boosted CF

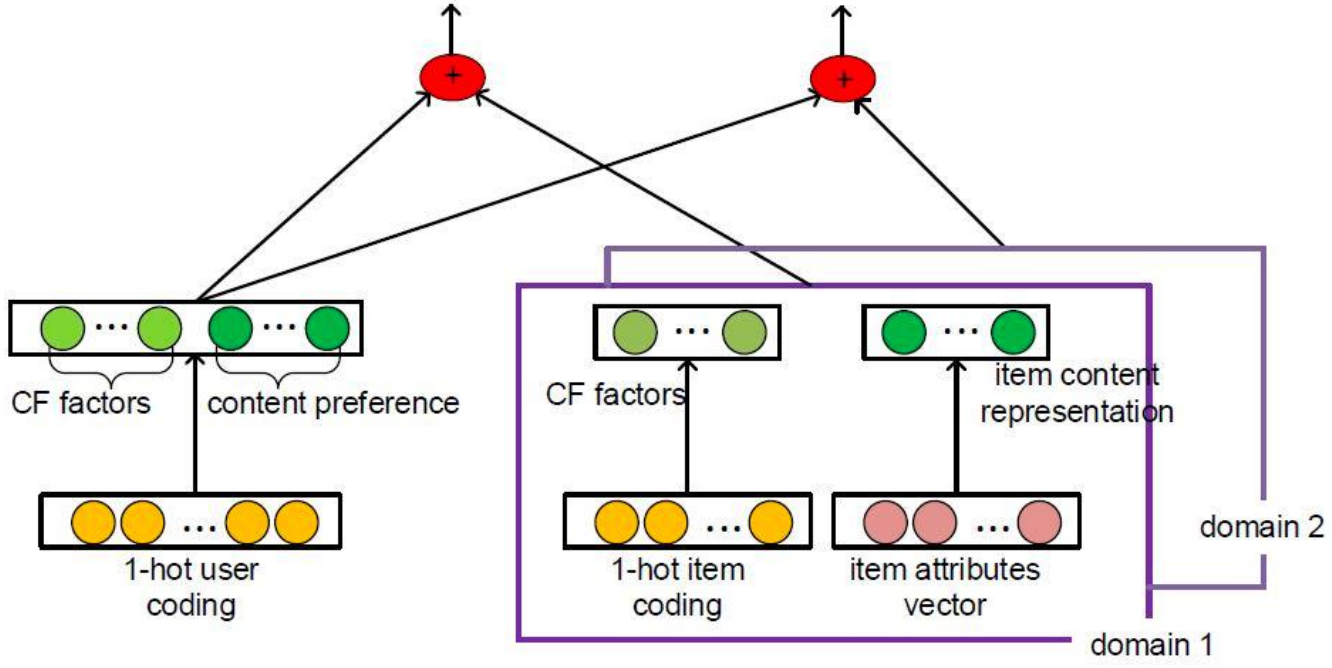
Чтобы преодолеть проблему разреженности данных, предлагается систему междоменных рекомендаций под названием CCCFNet, которая может сочетать совместную фильтрацию и фильтрацию на основе содержимого в единой структуре. Сначала мы представляем структуру факторизации, чтобы связать CF и контентную фильтрацию.



Cross-domain Content-boosted CF

Cross-domain Content-boosted CF

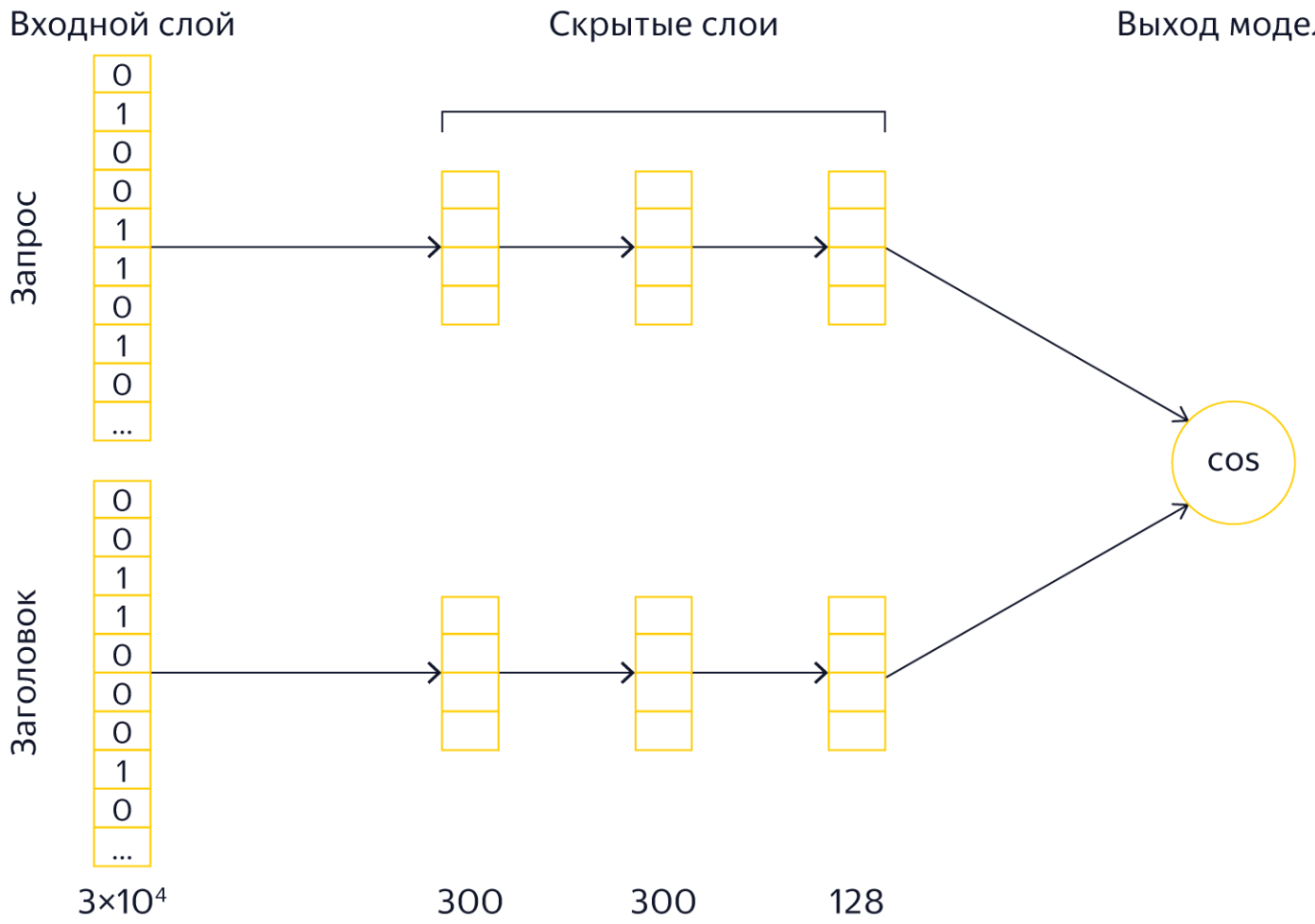
Затем мы обнаруживаем, что оценка MAP этой структуры может быть встроена в нейронную сеть с несколькими представлениями. Благодаря встраиванию этой нейронной сети фреймворк может быть дополнительно расширен за счет передовых методов глубокого обучения.



Разные архитектуры нейронных сетей

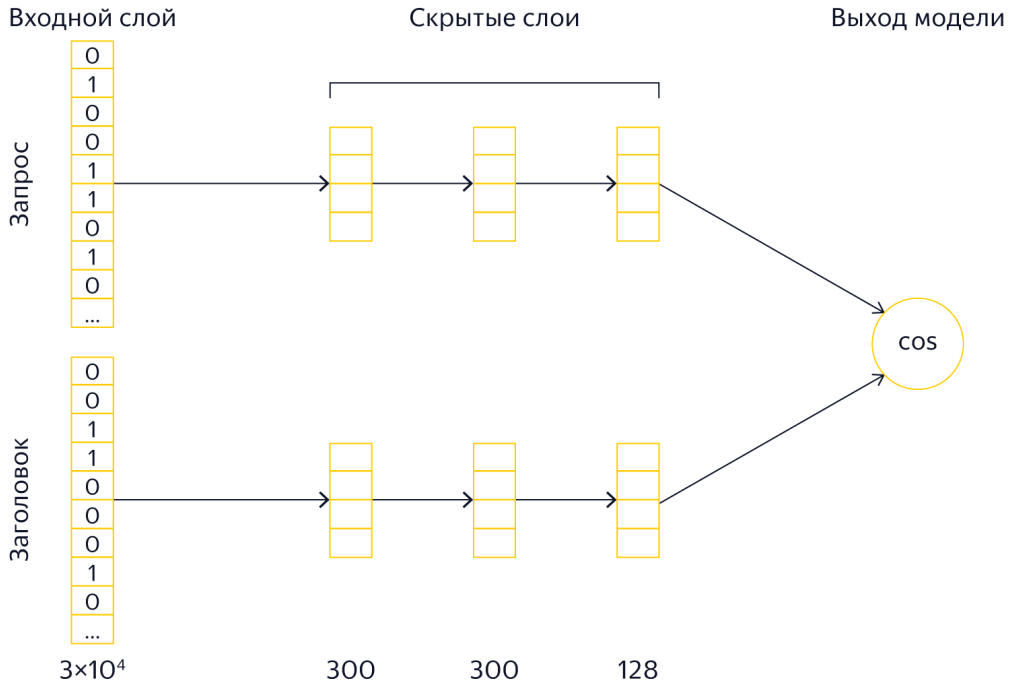
DSSM

В 2013 году исследователи из Microsoft Research описали свой подход, который получил название Deep Structured Semantic Model.



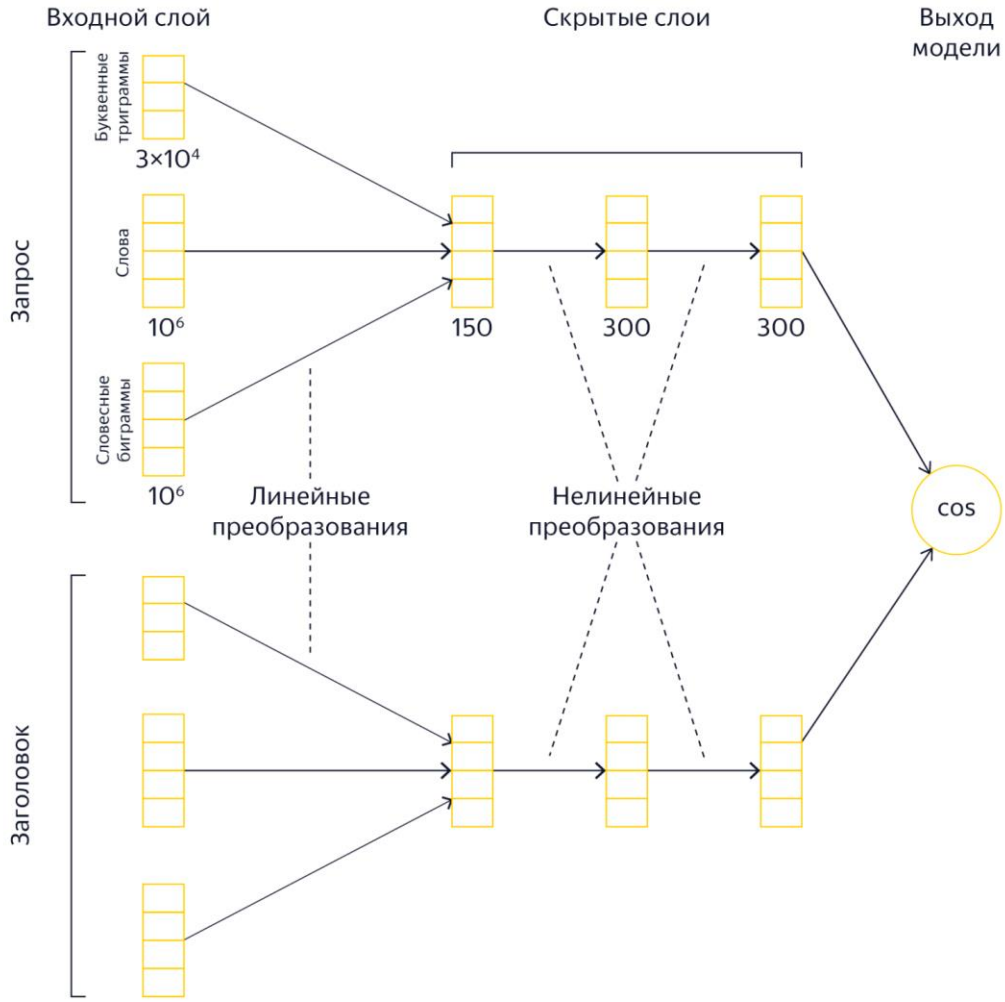
DSSM

На вход модели подаются тексты запросов и заголовков. Для уменьшения размеров модели, над ними производится операция, которую авторы называют word hashing. К тексту добавляются маркеры начала и конца, после чего он разбивается на буквенные триграммы. По сути, мы отмечаем таким образом вхождение триграмм из текста в словарь, состоящий из всех известных триграмм. Если сравнить такие вектора, то можно узнать только о наличии одинаковых триграмм в запросе и заголовке, что не представляет особого интереса. Поэтому теперь их надо преобразовать в другие вектора, которые уже будут иметь нужные нам свойства семантической близости.



DSSM. Большой входной слой

В оригинальной модели DSSM входной слой представляет собой множество буквенных триграмм. Его размер равен 30 000.



Пример DSSM. Алгоритм «Палех» от Яндекс

В ходе обучения DSSM был разработан поисковый алгоритм «Палех» использует нейронные сети для того, чтобы находить документы не по словам, которые используются в запросе и в самом документе, а по смыслу запроса и заголовка.

- Клюв – это высокочастотные запросы
- Туловище – среднечастотные
- Хвост – низкочастотные

Алгоритм «Палех» от Яндекс

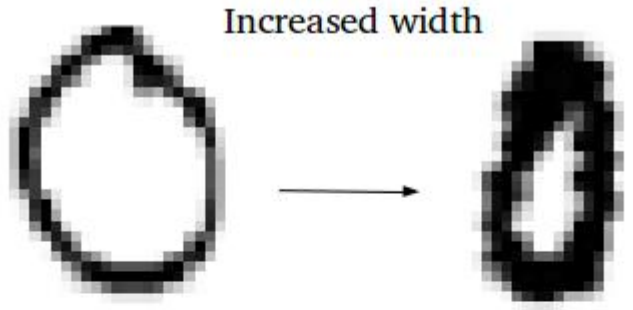
Как происходит обучение нейросети?
Каждый пример — это пара «запрос — заголовок».

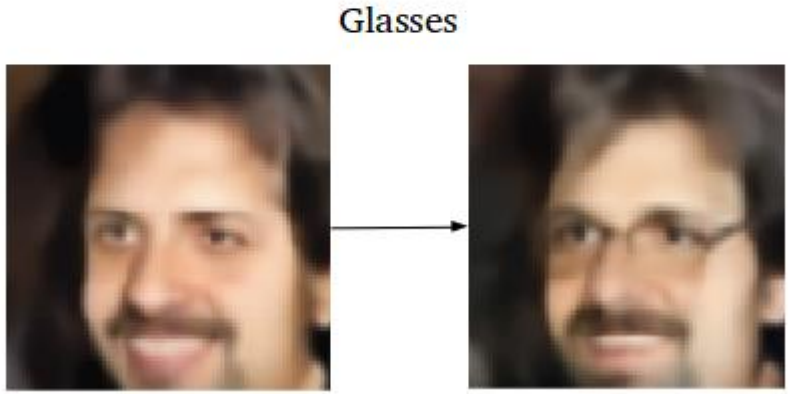
Яндекс научил нейронную сеть переводить миллиарды известных Яндексу заголовков веб-страниц в числа — а точнее, в группы из трёхсот чисел каждая. В результате все документы из базы данных Яндекса получили координаты в трёхсотмерном пространстве.

Autoencoder VAE

Вариационный автоэнкодер (Variational Autoencoder – VAE) — генеративная модель, которая находит применение во многих областях исследований: от генерации новых человеческих лиц до создания полностью искусственной музыки.

Генеративные модели используют для того, чтобы производить случайные выходные данные, которые выглядят схоже с тренировочным набором данных, и тоже самое вы можете делать с помощью VAEs.

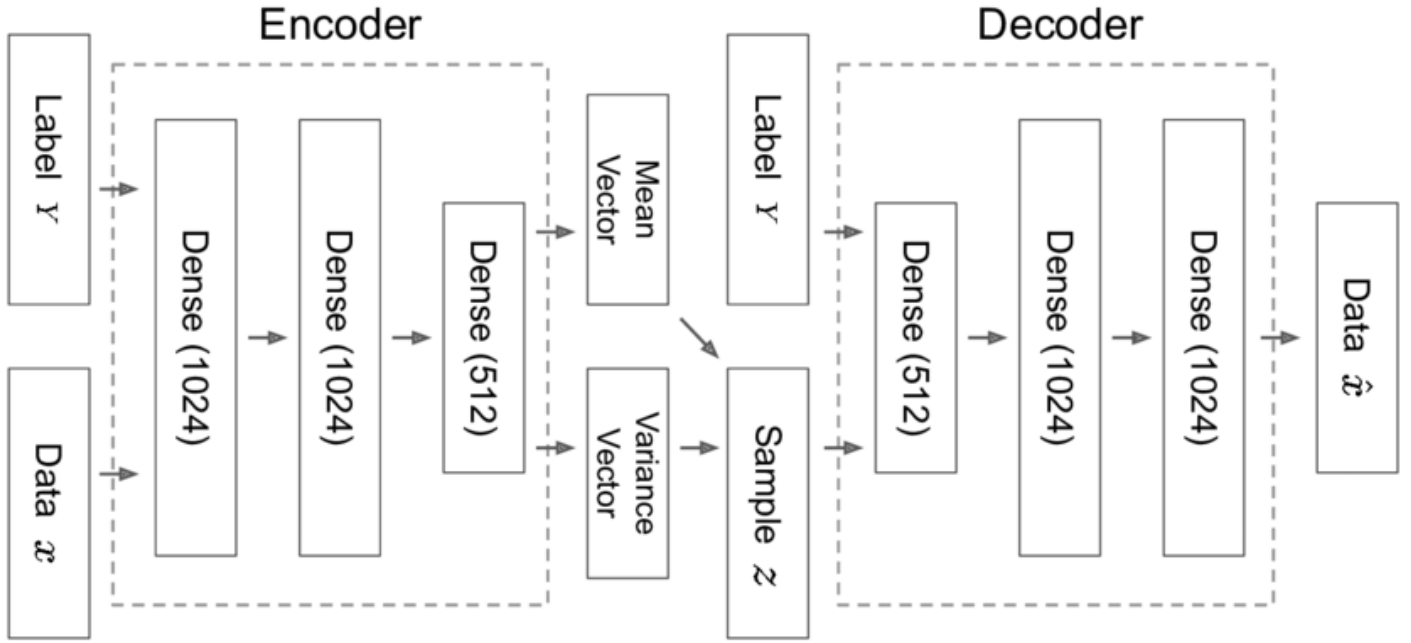




Autoencoder VAE

Нейросеть автоэнкодера является парой из двух соединенных нейросетей – **энкодера и декодера**.

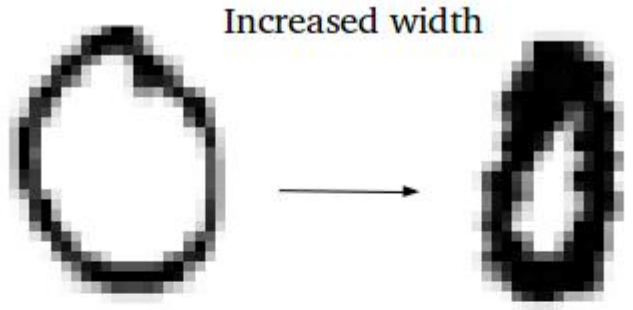
Энкодер принимает входные данные и преобразует их, делая представление более компактным и сжатым. В свою очередь, **декодер** использует преобразованные данные для трансформации их обратно в оригинальное состояние.

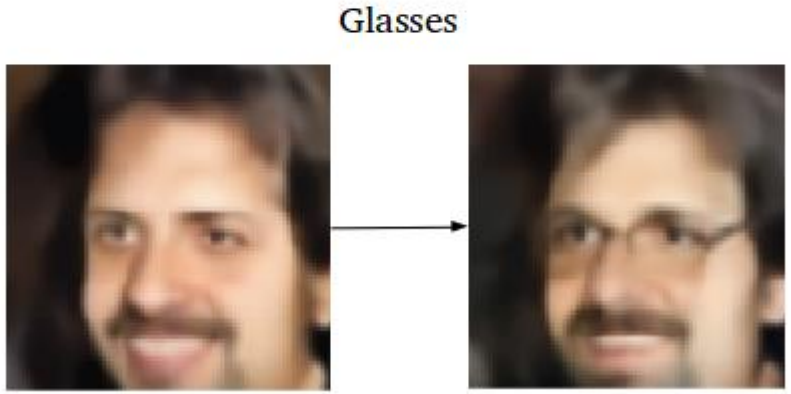


Autoencoder VAE

Вариационный автоэнкодер (Variational Autoencoder – VAE) — генеративная модель, которая находит применение во многих областях исследований: от генерации новых человеческих лиц до создания полностью искусственной музыки.

Генеративные модели используют для того, чтобы производить случайные выходные данные, которые выглядят схоже с тренировочным набором данных, и тоже самое вы можете делать с помощью VAEs.

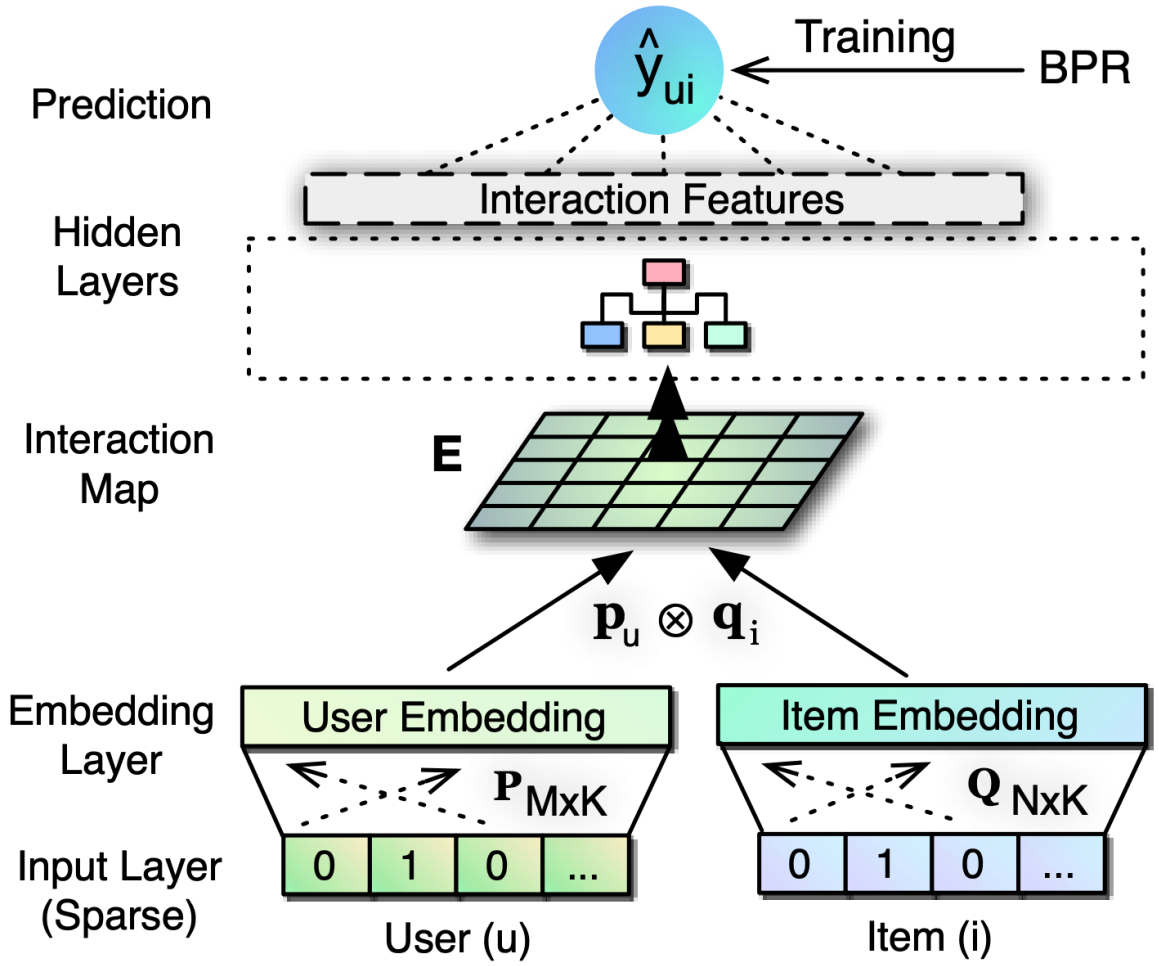




Convolution ONCF

Outer Product-based Neural Collaborative Filtering

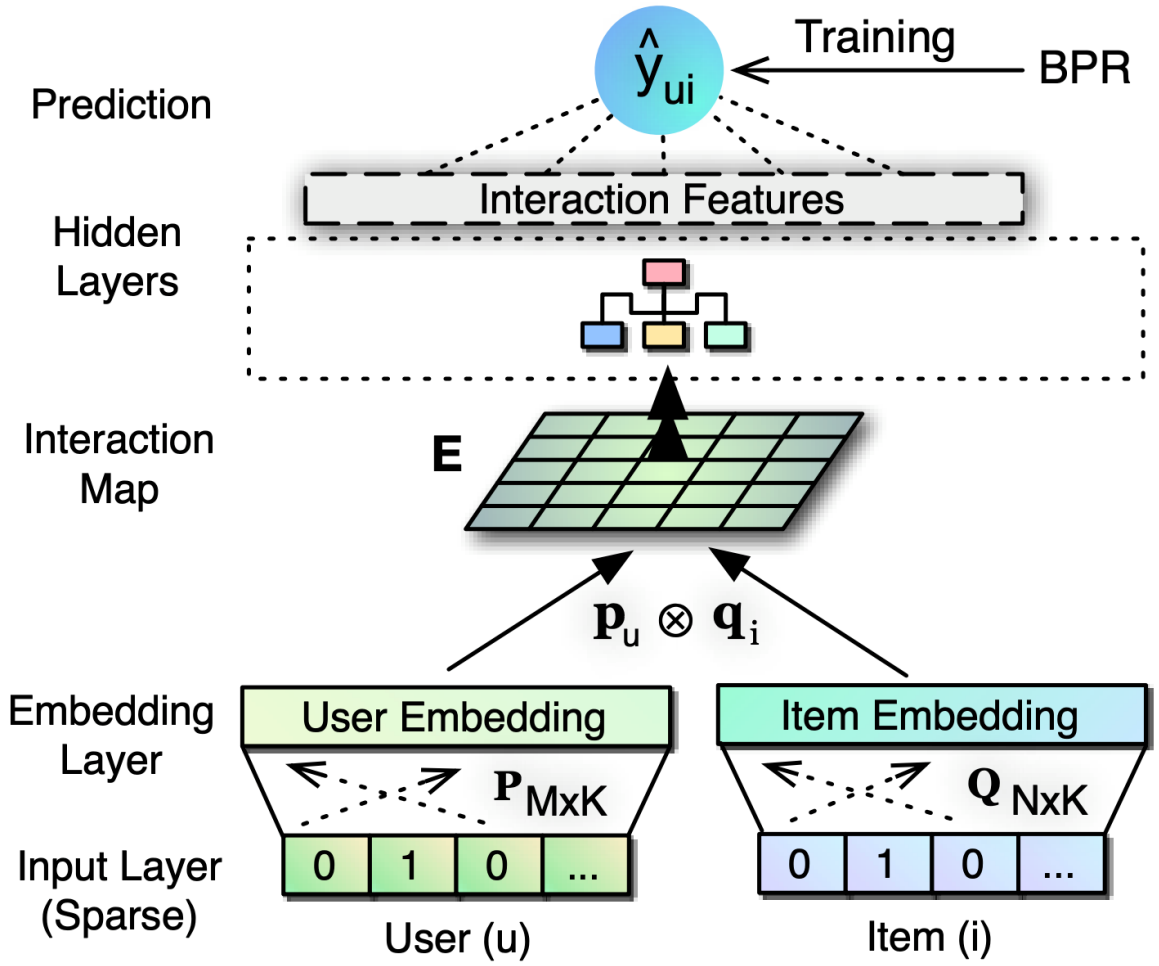
Целью моделирования является оценка соответствия между пользователем u и элементом i, то есть $\widehat{y_{ui}}$; а затем мы можем создать персонализированный список рекомендаций для пользователя на основе оценок.



Convolution ONCF

Outer Product-based Neural Collaborative Filtering

Целью моделирования является оценка соответствия между пользователем u и элементом i, то есть $\widehat{y_{ui}}$; а затем мы можем создать персонализированный список рекомендаций для пользователя на основе оценок.



Convolution ONCF

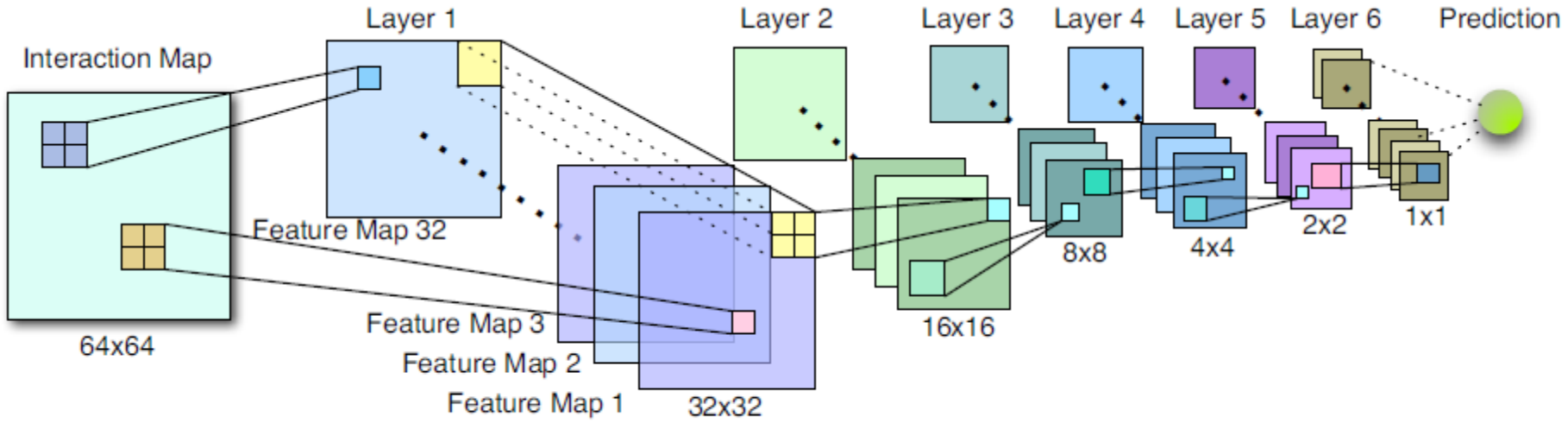
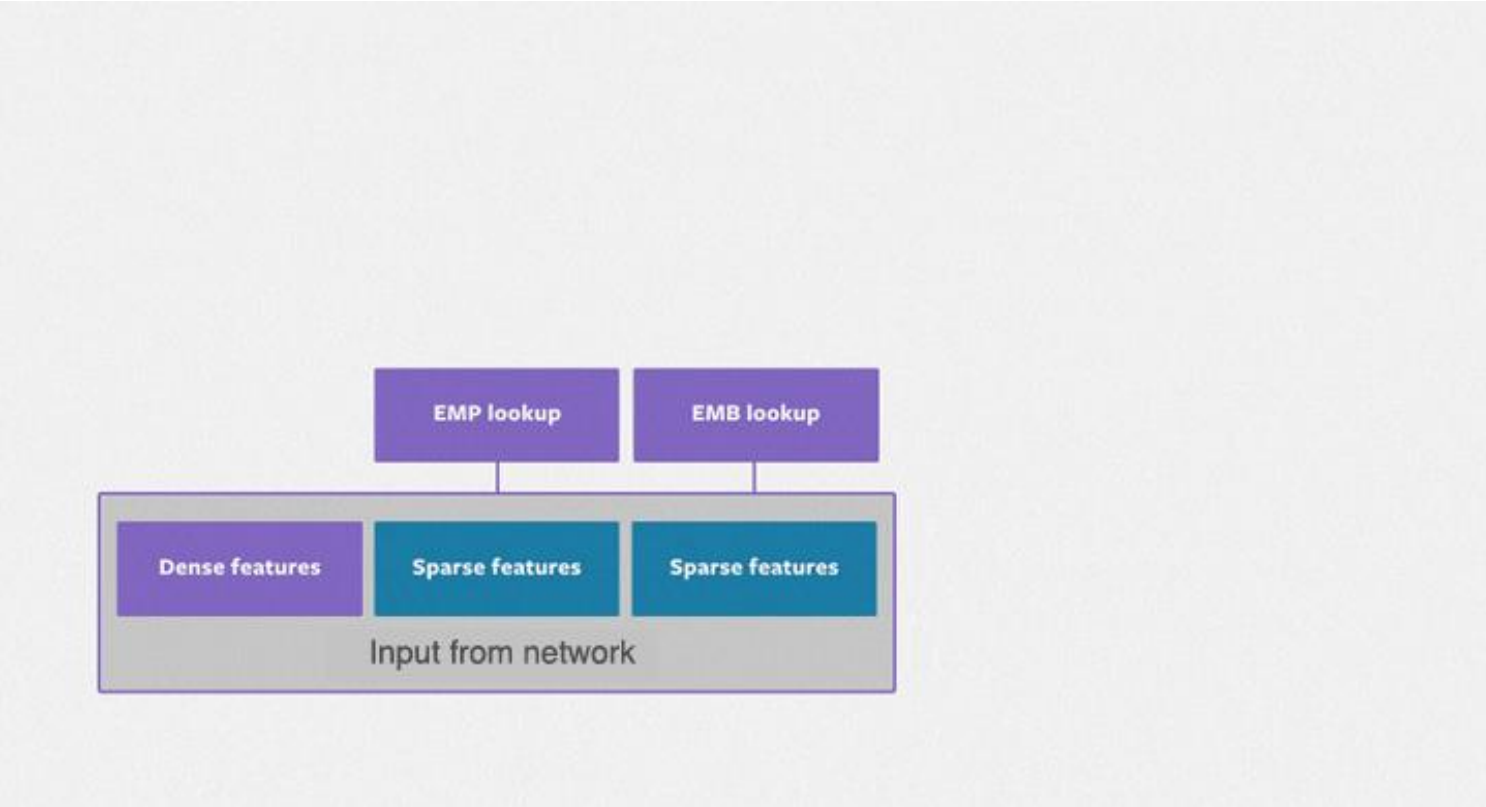


Figure 2: An example of the architecture of our ConvNCF model that has 6 convolution layers with embedding size 64.

https://www.ijcai.org/proceedings/2018/0308.pdf

Библиотека DLRM от Facebook

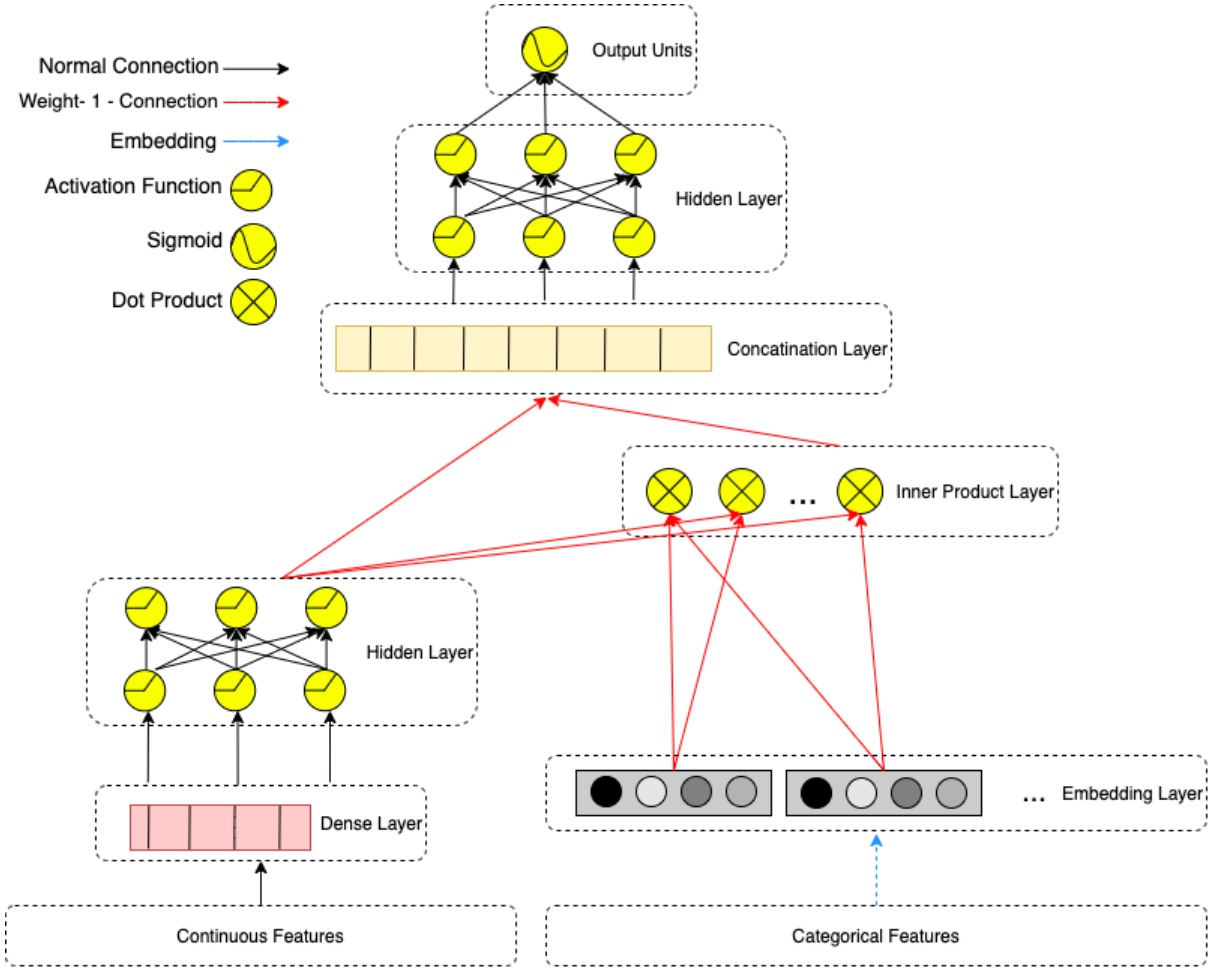
Библиотека DLRM на python для построение рекомендательных систем.



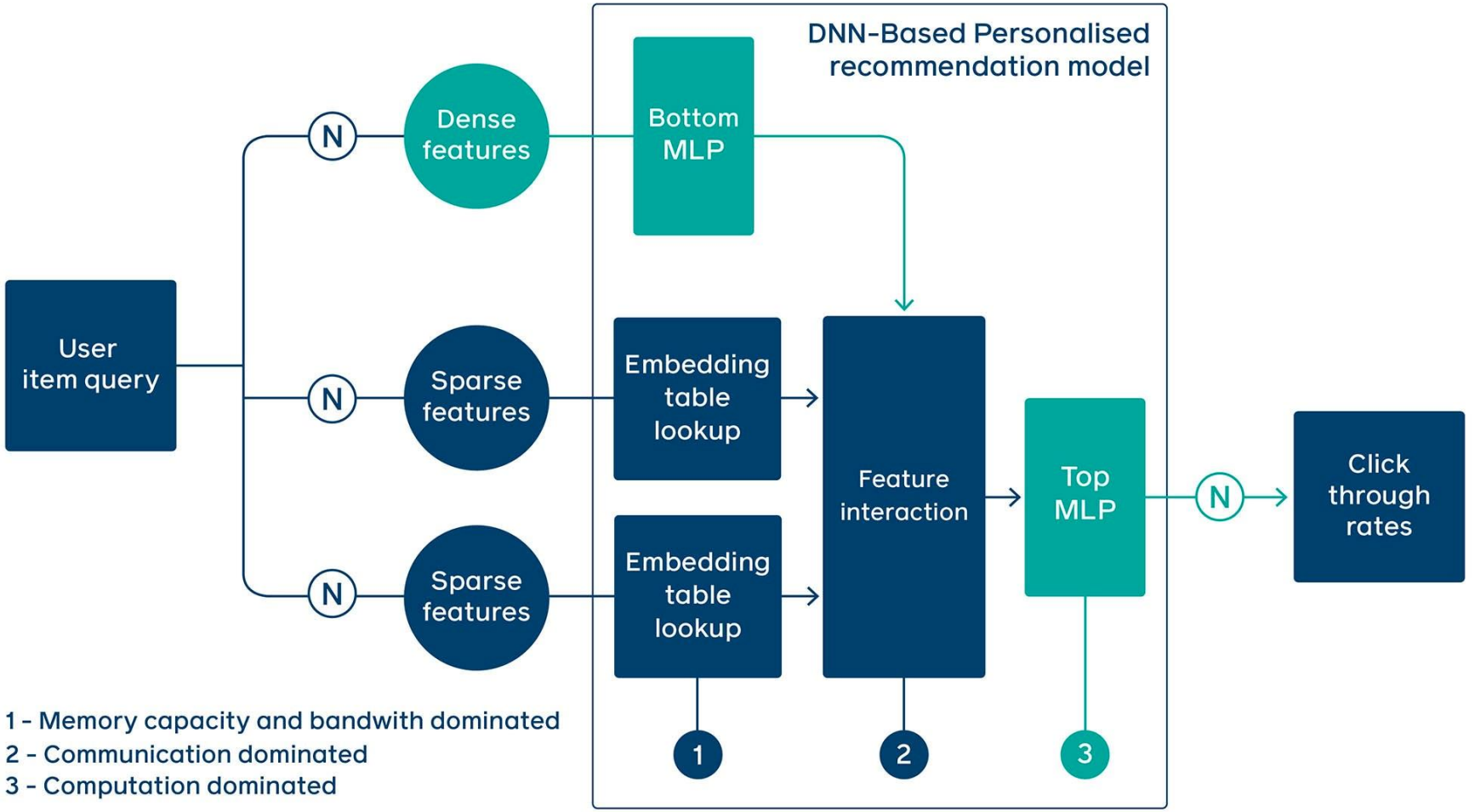
https://github.com/facebookresearch/dlrm

Библиотека DLRM от Facebook

Каждый категориальный признак представлен вектором представления, а постоянные признаки обрабатываются многослойным перцептроном таким образом, чтобы на выходе получались векторы такого же размера, как и векторы представления. На второй стадии рассчитываются взаимные произведения всех векторов представления и выходных векторов перцептрона. После этого произведения соединяются вместе и передаются в другой многослойный перцептрон, а в конце концов – в функцию сигмоиды, выдающую вероятность.



Библиотека DLRM от Facebook



Are We Really Making Much Progress?

Обратная сторона

Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches

Maurizio Ferrari Dacrema
Politecnico di Milano, Italy
maurizio.ferrari@polimi.it

Paolo Cremonesi
Politecnico di Milano, Italy
paolo.cremonesi@polimi.it

Dietmar Jannach
University of Klagenfurt, Austria
dietmar.jannach@aau.at

https://github.com/MaurizioFD/RecSys2019_DeepLearning_Evaluation

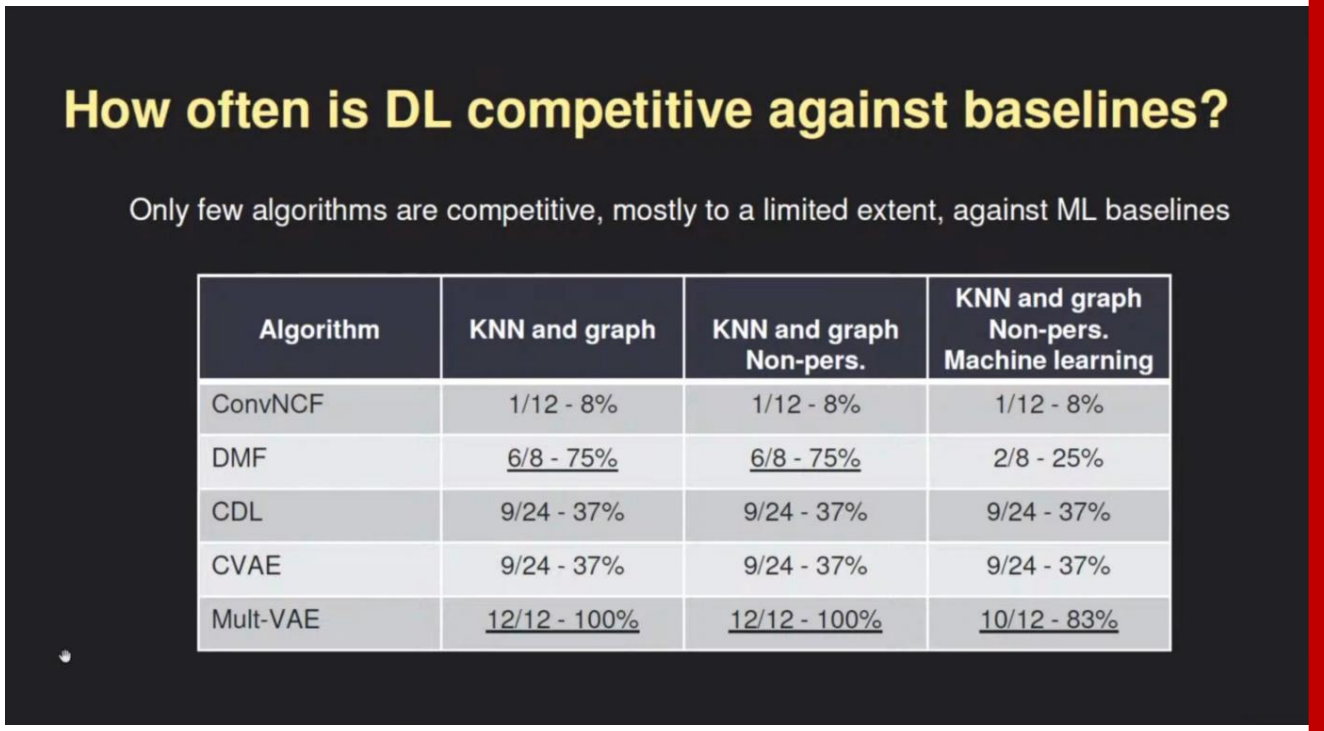
https://arxiv.org/pdf/1907.06902.pdf

https://www.youtube.com/watch?v=JlLHIrzrmi4

Are We Really Making Much Progress?

Выявленные проблемы

- **Воспроизводимость:** можно воспроизвести менее половины лучших работ (7/18 в исходной статье, 12/26 в расширенной версии) в этой области
- **Прогресс:** можно показать, что только *1/7* (в исходной статье) результатов стабильно превосходит хорошо настроенные базовые показатели



Algorithm	KNN and graph	KNN and graph Non-pers.	KNN and graph Non-pers. Machine learning
ConvNCF	1/12 - 8%	1/12 - 8%	1/12 - 8%
DMF	6/8 - 75%	6/8 - 75%	2/8 - 25%
CDL	9/24 - 37%	9/24 - 37%	9/24 - 37%
CVAE	9/24 - 37%	9/24 - 37%	9/24 - 37%
Mult-VAE	12/12 - 100%	12/12 - 100%	10/12 - 83%

Воспроизводимость

Воспроизводимость – числовые результаты, представленные в исследовании, могут быть точно воспроизведены с использованием исходных артефактов (то есть данных)

Воспроизводимые работы по алгоритмам глубокого обучения среди лучших рекомендаций по серии конференций с 2015 по 2018 г.

Conference	Rep. ratio	Reproducible
KDD	3/4 (75%)	[17], [23], [48]
RecSys	1/7 (14%)	[53]
SIGIR	1/3 (30%)	[10]
WWW	2/4 (50%)	[14], [24]
Total	7/18 (39%)	

Non-reproducible: KDD: [43], RecSys: [41], [6], [38], [44], [21], [45], SIGIR: [32], [7], WWW: [42], [11]

Воспроизводимость

Экспериментальные результаты для метода CMN с использованием показателей и пороговых значений, представленных в исходной статье. Числа напечатаны жирным шрифтом, если они соответствуют наилучшему результату или когда базовый уровень превосходит CMN.

Итог: воспроизводимость очень низкая

	CiteULike-a			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.1803	0.1220	0.2783	0.1535
UserKNN	**0.8213**	**0.7033**	**0.8935**	**0.7268**
ItemKNN	**0.8116**	**0.6939**	0.8878	**0.7187**
$P^3\alpha$	**0.8202**	**0.7061**	0.8901	**0.7289**
$RP^3\beta$	**0.8226**	**0.7114**	**0.8941**	**0.7347**
CMN	0.8069	0.6666	0.8910	0.6942
	Pinterest			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.1668	0.1066	0.2745	0.1411
UserKNN	**0.6886**	**0.4936**	0.8527	**0.5470**
ItemKNN	**0.6966**	**0.4994**	**0.8647**	**0.5542**
$P^3\alpha$	0.6871	**0.4935**	0.8449	**0.5450**
$RP^3\beta$	**0.7018**	**0.5041**	**0.8644**	**0.5571**
CMN	0.6872	0.4883	0.8549	0.5430
	Epinions			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	**0.5429**	**0.4153**	**0.6644**	**0.4547**
UserKNN	0.3506	0.2983	0.3922	0.3117
ItemKNN	0.3821	0.3165	0.4372	0.3343
$P^3\alpha$	0.3510	0.2989	0.3891	0.3112
$RP^3\beta$	0.3511	0.2980	0.3892	0.3103
CMN	0.4195	0.3346	0.4953	0.3592

Воспроизводимость

Итог: воспроизводимость очень низкая

Проблемы:

- артефакты не доступны или не работы
- контакт с авторами часто бесполезен
- частота методологических проблем
- произвольный экспериментальный дизайн

	CiteULike-a			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.1803	0.1220	0.2783	0.1535
UserKNN	**0.8213**	**0.7033**	**0.8935**	**0.7268**
ItemKNN	**0.8116**	**0.6939**	0.8878	**0.7187**
$P^3\alpha$	**0.8202**	**0.7061**	0.8901	**0.7289**
$RP^3\beta$	**0.8226**	**0.7114**	**0.8941**	**0.7347**
CMN	0.8069	0.6666	0.8910	0.6942
	Pinterest			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	0.1668	0.1066	0.2745	0.1411
UserKNN	**0.6886**	**0.4936**	0.8527	**0.5470**
ItemKNN	**0.6966**	**0.4994**	**0.8647**	**0.5542**
$P^3\alpha$	0.6871	**0.4935**	0.8449	**0.5450**
$RP^3\beta$	**0.7018**	**0.5041**	**0.8644**	**0.5571**
CMN	0.6872	0.4883	0.8549	0.5430
	Epinions			
	HR@5	NDCG@5	HR@10	NDCG@10
TopPopular	**0.5429**	**0.4153**	**0.6644**	**0.4547**
UserKNN	0.3506	0.2983	0.3922	0.3117
ItemKNN	0.3821	0.3165	0.4372	0.3343
$P^3\alpha$	0.3510	0.2989	0.3891	0.3112
$RP^3\beta$	0.3511	0.2980	0.3892	0.3103
CMN	0.4195	0.3346	0.4953	0.3592

Обзор базовых методов

Table 1. Overview of Baseline Methods

Family	*Method*	*Description*
Non-personalized	TopPopular	Recommends the most popular items to everyone [10]
Nearest-Neighbor	UserKNN	User-based k-nearest neighbors [26]
	ItemKNN	Item-based k-nearest neighbors [27]
Graph-based	$P^3\alpha$	A graph-based method based on random walks [9]
	$RP^3\beta$	An extension of $P^3\alpha$ [23]
Content-Based and Hybrid	ItemKNN-CBF	ItemKNN with content-based similarity [20]
	ItemKNN-CFCBF	A simple item-based hybrid CBF/CF approach [21]
	UserKNN-CBF	UserKNN with content-based similarity
	UserKNN-CFCBF	A simple user-based hybrid CBF/CF approach
Non-Neural Machine Learning	iALS	Matrix factorization for implicit feedback data [15]
	PureSVD	A basic matrix factorization method [10]
	NFM	A basic non-negative matrix factorization method [8]
	FunkSVD	Matrix factorization for rating prediction [17]
	MF BPR	Matrix factorization optimized for ranking [25]
	SLIM ElasticNet	A scalable linear model [18, 22]
	SLIM BPR	A variation of SLIM optimizing ranking [4]
	$EASE^R$	A recent linear model, similar to auto-encoders [28]

Выбор и распространение слабых базовых линий

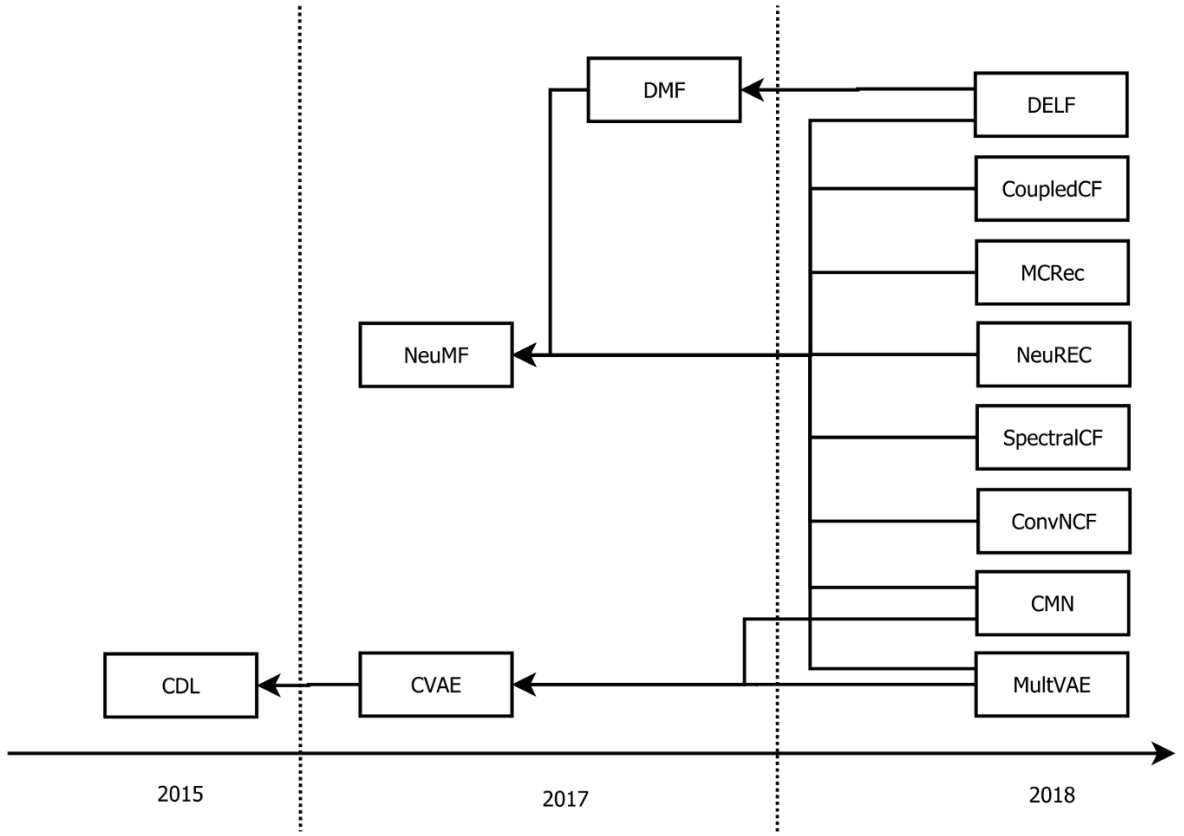


Fig. 1. Overview of Neural Methods, arrows indicate when a newer method used another one as baseline in the experiments.

Ошибки и недостаток информации

- предоставленное разделение данных не соответствует описанию
- выбрано лучшее значение показателя независимо от периода времени
- ошибка в вычислении метрики

Ошибки и недостаток информации

- предоставленное разделение данных не соответствует описанию
- выбрано лучшее значение показателя независимо от периода времени
- ошибка в вычислении метрики

Однако проблема не в deep-learning

Что делать дальше?

Советы для практиков

- Не пользуйтесь модным алгоритмом глубокого обучения с первой попытки. Это может стоить вам драгоценного времени
- Вместо этого начните с изучения вашего набора данных и настройки применимых базовых моделей
- Помните, что получателями рекомендаций являются настоящие люди

Вопросы

Семинар: Построение нейронной сети для рекомендательной системы