Categorical Features in Large Scale ML

Шугаепов Ильнур
VK.com
ilnur.shug@gmail.com

февраль 2020 г.

Click-through rate prediction
Problem formulation

- Dataset: $X^N = \{z_i\}_{i=1}^N$, где $z_i = (\mathbf{x}_i, y_i) \sim P(z), y_i \in \{0, 1\}$

Click-through rate prediction

Problem formulation

- Dataset: $X^N = \{z_i\}_{i=1}^N$, где $z_i = (\mathbf{x}_i, y_i) \sim P(z), y_i \in \{0, 1\}$
- Prediction:

$$\hat{y}_i = f_{\mathbf{w}}(\mathbf{x}_i) = \mathbb{P}\{y = 1 \mid \mathbf{x}_i\}$$

Click-through rate prediction
Problem formulation

- Dataset: $X^N = \{z_i\}_{i=1}^N$, где $z_i = (\mathbf{x}_i, y_i) \sim P(z), y_i \in \{0, 1\}$
- Prediction:
$$\hat{y}_i = f_{\mathbf{w}}(\mathbf{x}_i) = \mathbb{P}\{y = 1 \mid \mathbf{x}_i\}$$
- Loss function (Binary Cross-Entropy):

$$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 - \frac{1}{N}\sum_{i=1}^{N} y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)$$

Simple Solution
Linear Model

- $\mathbf{x}_i \in \mathbb{R}^n$
- $f_{\mathbf{w}}(\mathbf{x}) = \sigma(\phi_{LM}(\mathbf{w}, \mathbf{x}))$, where $\phi_{LM}(\mathbf{w}, \mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$

Types of features

$$\mathbf{x} = (x_1, \dots, x_d)$$

Types of features:

- $x_k \in \{0, 1\}$ — binary
- $x_k \in \mathbb{R}$ — real-valued
- $x_k \in \mathcal{C}_k, |\mathcal{C}_k| < \infty$ — **categorical**
- $x_k \in \mathcal{C}_k, |\mathcal{C}_k| < \infty, \mathcal{C}_k$ - ordered — ordinal

Categorical Features

$$\mathcal{C}_k = \{c_{k,1}, \ldots, c_{k,m}\}$$

Examples:

- Ids: `UserId`, `ItemId`, etc
- Category (product, ad, client), usually comes from hierarchy
- City, color, etc

Cross Features

Определение (Cross Features aka Combinatorial Features)

Дано: индивидуальные фичи $X_i \in \mathcal{C}_i$ *и* $X_j \in \mathcal{C}_j$*, тогда cross feature* $X_{i,j} \in \mathcal{C}_i \times \mathcal{C}_j$*. Новая фича может иметь как sparse, так и dense представления.*

Cross Features

Определение (Cross Features aka Combinatorial Features)

Дано: индивидуальные фичи $X_i \in \mathcal{C}_i$ и $X_j \in \mathcal{C}_j$, тогда cross feature $X_{i,j} \in \mathcal{C}_i \times \mathcal{C}_j$. Новая фича может иметь как sparse, так и dense представления.

Пример:

- `UserSexId` $\in \mathcal{C}$, и `AdDomainId` $\in \mathcal{C}' : |\mathcal{C}| = 3, |\mathcal{C}'| = 20000$ - категориальные фичи, `AdDomainId` $\times$ `UserSexId` - новая категориальная фича,

Cross Features

Определение (Cross Features aka Combinatorial Features)

Дано: индивидуальные фичи $X_i \in \mathcal{C}_i$ и $X_j \in \mathcal{C}_j$, тогда cross feature $X_{i,j} \in \mathcal{C}_i \times \mathcal{C}_j$. Новая фича может иметь как sparse, так и dense представления.

Пример:

- `UserSexId` $\in \mathcal{C}$, и `AdDomainId` $\in \mathcal{C}' : |\mathcal{C}| = 3, |\mathcal{C}'| = 20000$ - категориальные фичи, `AdDomainId` $\times$ `UserSexId` - новая категориальная фича,
- sparse представление - просто one-hot encoding,

Cross Features

Определение (Cross Features aka Combinatorial Features)

Дано: индивидуальные фичи $X_i \in \mathcal{C}_i$ *и* $X_j \in \mathcal{C}_j$*, тогда cross feature* $X_{i,j} \in \mathcal{C}_i \times \mathcal{C}_j$*. Новая фича может иметь как sparse, так и dense представления.*

Пример:

- `UserSexId` $\in \mathcal{C}$, и `AdDomainId` $\in \mathcal{C}' : |\mathcal{C}| = 3, |\mathcal{C}'| = 20000$ - категориальные фичи, `AdDomainId` $\times$ `UserSexId` - новая категориальная фича,
- sparse представление - просто one-hot encoding,
- dense представление - target mean encoding.

Проблемы:

1. cross features часто несут полезный сигнал, но ручной поиск и добавление таких фичей в модель довольно трудоемкий процесс

Проблемы:

1. cross features часто несут полезный сигнал, но ручной поиск и добавление таких фичей в модель довольно трудоемкий процесс
2. при использовании cross features размерность пространства признаков быстро растет

Методы работы с категориальными признаками

1. Dummy variables, one-hot encoding
2. Embeddings
3. Naive Bayes, counters
4. CatBoost
5. Hashing

One-Hot encoding

1. Каждая переменная $x_k \in \mathcal{C}_k$ заменяется на $|\mathcal{C}_k|$ бинарных $z_{k,t}, t = 1, \dots, |\mathcal{C}_k|$
2. Если $x_k = c_{k,t}$, то $\mathbf{z} = (0, \dots, 1, \dots, 0)$, единица на t-ой позиции

FIELD-AWARE FACTORIZATION MACHINES

Poly2

$$\phi_{Poly2}(\mathbf{w}, \mathbf{x}) = \underbrace{\phi_{LM}(\mathbf{w}, \mathbf{x})}_{\text{linear part}} + \underbrace{\sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} w_{h(j_1,j_2)} x_{j_1} x_{j_2}}_{\text{pair-wise feature interactions}},$$

where $h(j_1, j_2)$ is a function encoding j_1 and j_2 into a natural number

Poly2

$$\phi_{Poly2}(\mathbf{w}, \mathbf{x}) = \underbrace{\phi_{LM}(\mathbf{w}, \mathbf{x})}_{\text{linear part}} + \underbrace{\sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} w_{h(j_1, j_2)} x_{j_1} x_{j_2}}_{\text{pair-wise feature interactions}},$$

where $h(j_1, j_2)$ is a function encoding j_1 and j_2 into a natural number

Problems:

- The complexity is $\mathcal{O}(\overline{n}^2)$, where $\overline{n}$ is the average number of non-zero elements per instance

Poly2

$$\phi_{Poly2}(\mathbf{w}, \mathbf{x}) = \underbrace{\phi_{LM}(\mathbf{w}, \mathbf{x})}_{\text{linear part}} + \underbrace{\sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} w_{h(j_1, j_2)} x_{j_1} x_{j_2}}_{\text{pair-wise feature interactions}},$$

where $h(j_1, j_2)$ is a function encoding j_1 and j_2 into a natural number

Problems:

- The complexity is $\mathcal{O}(\overline{n}^2)$, where $\overline{n}$ is the average number of non-zero elements per instance
- Overfitting of $w_{h(j_1, j_2)}$ for rare occurrences of combination x_{j_1}, x_{j_2}

Factorization Machines [4, 5]

$$\phi_{FM}(\mathbf{w}, \mathbf{x}) = \phi_{LM}(\mathbf{w}, \mathbf{x}) + \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (\mathbf{w}_{j_1} \cdot \mathbf{w}_{j_2}) x_{j_1} x_{j_2},$$

where $\mathbf{w}_j \in \mathbb{R}^k$ - latent vector,

[1] https://en.wikipedia.org/wiki/Definiteness_of_a_matrix

Factorization Machines [4, 5]

$$\phi_{FM}(\mathbf{w}, \mathbf{x}) = \phi_{LM}(\mathbf{w}, \mathbf{x}) + \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (\mathbf{w}_{j_1} \cdot \mathbf{w}_{j_2}) x_{j_1} x_{j_2},$$

where $\mathbf{w}_j \in \mathbb{R}^k$ - latent vector,

- *Parameter Estimation Under Sparsity*: Data for one interaction helps also to estimate the parameters for related interactions
- *Expressiveness*: It is well known that for any positive definite matrix[1] X, there exists a matrix W such that $X = W \cdot W^T$ provided that k is sufficiently large.

[1] https://en.wikipedia.org/wiki/Definiteness_of_a_matrix

Factorization Machines [4, 5]

$$\phi_{FM}(\mathbf{w}, \mathbf{x}) = \phi_{LM}(\mathbf{w}, \mathbf{x}) + \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (\mathbf{w}_{j_1} \cdot \mathbf{w}_{j_2}) x_{j_1} x_{j_2},$$

- The complexity is $\mathcal{O}(\overline{n}^2 k)$

Factorization Machines [4, 5]

$$\phi_{FM}(\mathbf{w},\mathbf{x}) = \phi_{LM}(\mathbf{w},\mathbf{x}) + \sum_{j_1=1}^{n}\sum_{j_2=j_1+1}^{n}(\mathbf{w}_{j_1}\cdot\mathbf{w}_{j_2})x_{j_1}x_{j_2},$$

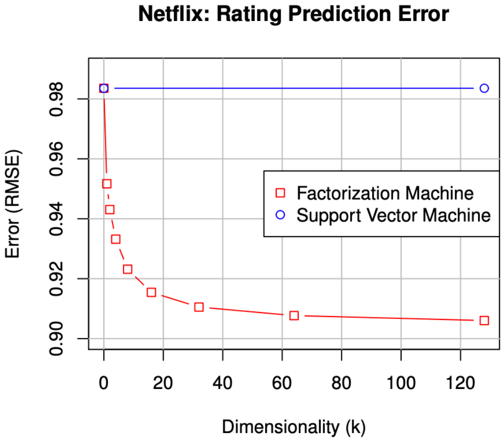
- The complexity is $\mathcal{O}(\overline{n}^2 k)$
- Complexity can be reduced to $\mathcal{O}(\overline{n}k)$ if we re-write $\phi_{FM}(\mathbf{w},\mathbf{x})$ as

$$\phi_{FM}(\mathbf{w},\mathbf{x}) = \frac{1}{2}\sum_{j=1}^{n}(\mathbf{s}-\mathbf{w}_j x_j)\cdot\mathbf{w}_j x_j,$$

where $\mathbf{s} = \sum_{j'=1}^{n}\mathbf{w}_{j'}x_{j'}$

Factorization Machines [4, 5]

Performance



Field-aware Factorization Machines [2, 3]

Motivation

Example

Clicked	Publisher (P)	Advertiser (A)	Gender (G)
Yes	VK	Nike	Male

Field-aware Factorization Machines [2, 3]

Motivation

Example

Clicked	Publisher (P)	Advertiser (A)	Gender (G)
Yes	VK	Nike	Male

Recall that for FMs, $\phi_{FM}(\mathbf{w}, \mathbf{x})$ is

$$\mathbf{w}_{\mathrm{VK}} \cdot \mathbf{w}_{\mathrm{Nike}} + \mathbf{w}_{\mathrm{VK}} \cdot \mathbf{w}_{\mathrm{Male}} + \mathbf{w}_{\mathrm{Nike}} \cdot \mathbf{w}_{\mathrm{Male}}$$

Field-aware Factorization Machines [2, 3]

Motivation

Example

Clicked	Publisher (P)	Advertiser (A)	Gender (G)
Yes	VK	Nike	Male

Recall that for FMs, $\phi_{FM}(\mathbf{w}, \mathbf{x})$ is

$$\mathbf{w}_{\mathrm{VK}} \cdot \mathbf{w}_{\mathrm{Nike}} + \mathbf{w}_{\mathrm{VK}} \cdot \mathbf{w}_{\mathrm{Male}} + \mathbf{w}_{\mathrm{Nike}} \cdot \mathbf{w}_{\mathrm{Male}}$$

In FFMs, each feature has several latent vectors. $\phi_{FFM}(\mathbf{w}, \mathbf{x})$ is

$$\mathbf{w}_{\mathrm{VK,A}} \cdot \mathbf{w}_{\mathrm{Nike,P}} + \mathbf{w}_{\mathrm{VK,G}} \cdot \mathbf{w}_{\mathrm{Male,P}} + \mathbf{w}_{\mathrm{Nike,G}} \cdot \mathbf{w}_{\mathrm{Male,A}}$$

Usually $k_{\mathrm{FFM}} \ll k_{\mathrm{FM}}$

FFM
Definition

- FM

$$\phi_{FM}(\mathbf{w}, \mathbf{x}) = \phi_{LM}(\mathbf{w}, \mathbf{x}) + \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (\mathbf{w}_{j_1} \cdot \mathbf{w}_{j_2}) x_{j_1} x_{j_2},$$

  where $\mathbf{w}_j \in \mathbb{R}^k$,

- FFM - all features can be grouped into fileds (`Client, Group, UserSex, etc`)

$$\phi_{FFM}(\mathbf{w}, \mathbf{x}) = \phi_{LM}(\mathbf{w}, \mathbf{x}) + \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (\mathbf{w}_{j_1, f_2} \cdot \mathbf{w}_{j_2, f_1}) x_{j_1} x_{j_2},$$

  where f_1 and f_2 are respectively the fields of j_1 and j_2.

Learning

- SGD
- AdaGrad

Evaluation

Model and implementation	parameters	training time (seconds)	public set logloss	rank
LM-SG	$\eta = 0.2, \lambda = 0, t = 13$	527	0.46262	93
LM-LIBLINEAR-CD	$s = 7, c = 2$	1,417	0.46239	91
LM-LIBLINEAR-Newton	$s = 0, c = 2$	7,164	0.46602	225
Poly2-SG	$\eta = 0.2, \lambda = 0, B = 10^7, t = 10$	12,064	0.44973	14
Poly2-LIBLINEAR-Hash-CD	$s = 7, c = 2$	24,771	0.44893	13
FM	$\eta = 0.05, \lambda = 2 \times 10^{-5}, k = 40, t = 8$	2,022	0.44930	14
FM	$\eta = 0.05, \lambda = 2 \times 10^{-5}, k = 100, t = 9$	4,020	0.44867	11
LIBFM	$\lambda = 40, k = 40, t = 20$	23,700	0.45012	14
LIBFM	$\lambda = 40, k = 40, t = 50$	131,000	0.44904	14
LIBFM	$\lambda = 40, k = 100, t = 20$	54,320	0.44853	11
LIBFM	$\lambda = 40, k = 100, t = 50$	398,800	0.44794	9
FFM	$\eta = 0.2, \lambda = 2 \times 10^{-5}, k = 4, t = 9$	6,587	0.44612	3

Рис.: Criteo Dataset[2]. t - number of epochs

[2] http://www.kaggle.com/c/criteo-display-ad-challenge

Resume

Inference complexity:

	# variables	complexity
LM	n	$\mathcal{O}(\bar{n})$
Poly2	B	$\mathcal{O}\left(\bar{n}^2\right)$
FM	nk	$\mathcal{O}(\bar{n}k)$
FFM	nfk	$\mathcal{O}\left(\bar{n}^2 k\right)$

Resume

Inference complexity:

	# variables	complexity
LM	n	$\mathcal{O}(\bar{n})$
Poly2	B	$\mathcal{O}\left(\bar{n}^2\right)$
FM	nk	$\mathcal{O}(\bar{n}k)$
FFM	nfk	$\mathcal{O}\left(\bar{n}^2 k\right)$

Drawbacks:

- Degree-2 interactions only,
- Slow to train.

DEEP CROSSING

Deep Crossing [6]
Architecture

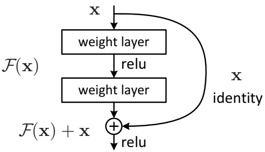


Рис.: Residual Unit

- Embedding and Stacking

$$\mathbf{x}_{\text{emb},i} = max(\mathbf{0}, W_{\text{embed},i}\mathbf{x}_i + \mathbf{b}_i),$$

$$\mathbf{x}_0 = \left[\mathbf{x}_{\text{emb},1}^T, \ldots, \mathbf{x}_{\text{emb},k}^T, \mathbf{x}_{\text{dense}}^T\right]^T,$$

- Residual Units (5 Layers),
- Scoring

$$p = \sigma\left(\mathbf{w}_{\text{logits}}\mathbf{h}_L\right).$$

Deep Crossing
Architecture

- Embeddings dimensionality - 256,
- Features with dimensionality lower than 256 are stacked without embedding.
- Number of nodes inside the Residual Unit - [64-512].

DEEP & CROSS

Deep & Cross [7]
Main Contributions

- Deep & Cross — network that explicitly applies feature crossing at each layer, efficiently learns predictive cross features of bounded degrees, and requires no manual feature engineering or exhaustive searching.
- Experimental results have demonstrated that with a cross network, DCN has lower logloss than a DNN with nearly an order of magnitude fewer number of parameters.

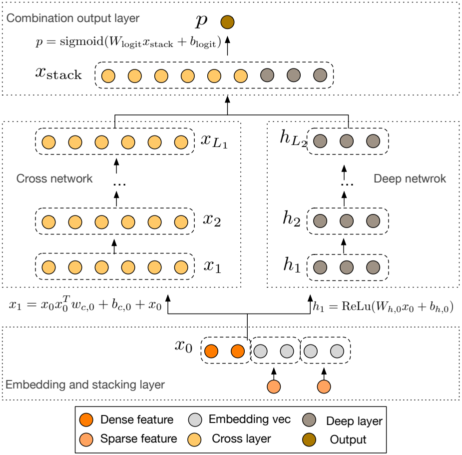


Рис.: The Deep & Cross Network

- Embedding and Stacking

$$\mathbf{x}_{\text{emb},i} = W_{\text{embed},i} \mathbf{x}_i,$$

$$\mathbf{x}_0 = \left[\mathbf{x}_{\text{emb},1}^T, \ldots, \mathbf{x}_{\text{emb},k}^T, \mathbf{x}_{\text{dense}}^T\right]^T,$$

- Cross Network

$$\mathbf{x}_{l+1} = \mathbf{x}_0 \mathbf{x}_l^T \mathbf{w}_l + \mathbf{b}_l + \mathbf{x}_l,$$

- Deep Network

$$\mathbf{h}_{l+1} = ReLU(W_l \mathbf{h}_l + \mathbf{b}_l),$$

- Combination Layer

$$p = \sigma\left([\mathbf{x}_{L_1}^T, \mathbf{h}_{L_2}^T] \mathbf{w}_{\text{logits}}\right),$$

- $\mathbf{x}_i \in \mathbb{R}^{n_v \times 1}$, $W_{\text{embed},i} \in \mathbb{R}^{n_e \times n_v}$
- $\mathbf{w}_l, \mathbf{b}_l \in \mathbb{R}^{d \times 1}$
- Then, the number of parameters involved in the cross network is

$$d \times L_c \times 2.$$

- Therefore, a cross network introduces negligible complexity compared to its deep counterpart, keeping the overall complexity for DCN at the same level as that of a traditional DNN.

Evaluation
CTR-prediction

Dataset: The Criteo Display Ads[3] — 13 integer features and 26 categorical features where each category has a high cardinality. For categorical features, embed the features in dense vectors of dimension $6 \cdot (\texttt{category cardinality})^{1/4}$. Concatenating all embeddings results in a vector of dimension 1026.

Таблица: Best test logloss from different models. "DC" is deep crossing, "DNN" is DCN with no cross layer, "FM" is Factorization Machine based model, "LR" is logistic regression.

Model	DCN	DC	DNN	FM	LR
Logloss	**0.4419**	0.4425	0.4428	0.4464	0.4474

[3] https://www.kaggle.com/c/criteo-display-ad-challenge

Evaluation
Number of parameters

Таблица: #parameters needed to achieve a desired logloss.

Logloss	0.4430	0.4460	0.4470	0.4480
DNN	3.2×10^6	1.5×10^5	1.5×10^5	7.8×10^4
DCN	$\mathbf{7.9} \times \mathbf{10^5}$	$\mathbf{7.3} \times \mathbf{10^4}$	$\mathbf{3.7} \times \mathbf{10^4}$	$\mathbf{3.7} \times \mathbf{10^4}$

Evaluation
Number of cross-layers

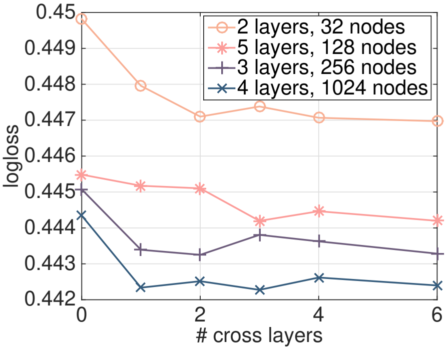


Рис.: Improvement in the validation logloss with the growth of cross layer depth. The case with 0 cross layers is equivalent to a single DNN model. In the legend, “layers” is hidden layers, “nodes” is hidden nodes. Different symbols represent different hyperparameters for the deep network.

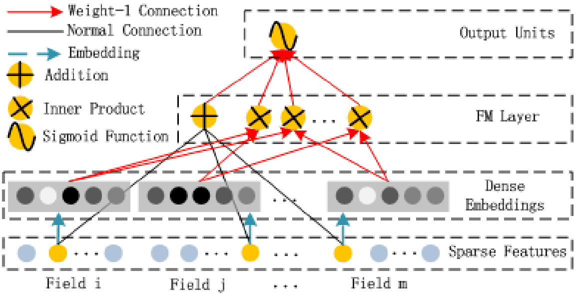
DeepFM

DeepFM [1]

- $$\mathbf{x} = [x_{field_1}, x_{field_2}, \ldots, x_{filed_j}, \ldots, x_{field_m}]$$
  categorical field is represented as a vector of one-hot encoding, and each continuous field is represented as the value itself, or a vector of one-hot encoding after discretization
- $$f_{\mathbf{w}}(\mathbf{x}) = \sigma(\phi_{FM} + \phi_{DNN})$$

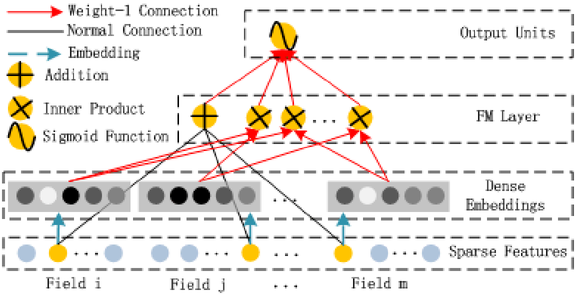
FM component

ϕ_{FM}



FM component

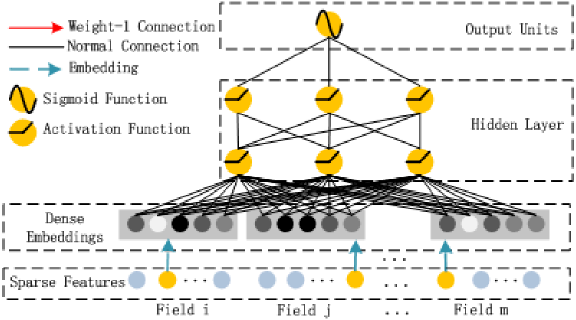
ϕ_{FM}



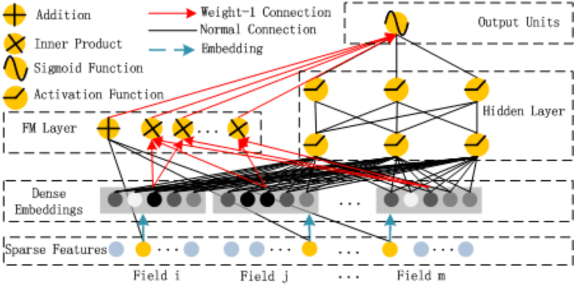
$$\phi_{FM}(\mathbf{w}, \mathbf{x}) = \phi_{LM}(\mathbf{w}, \mathbf{x}) + \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (\mathbf{w}_{j_1} \cdot \mathbf{w}_{j_2}) x_{j_1} x_{j_2},$$

Deep component

ϕ_{DNN}



Model


Addition
Inner Product
Sigmoid Function
Activation Function
Weight-1 Connection
Normal Connection
Embedding
Output Units
Hidden Layer
FM Layer
Dense Embeddings
Sparse Features
Field i
Field j
Field m

Evaluation

Таблица: Performance on CTR prediction. Criteo[4]

	Criteo	
	AUC	LogLoss
LR	0.7686	0.47762
FM	0.7892	0.46077
LR & DNN	0.7981	0.46772
FM & DNN	0.7850	0.45382
DeepFM	**0.8007**	**0.45083**

LR & DNN, FM & DNN - separate embeddings in LR/FM and DNN parts

[4] `http://www.kaggle.com/c/criteo-display-ad-challenge`

RESUME

Brief Resume

- Познакомились с задачей CTR-prediction
- Узнали как работать с категориальными признаками высокой размерности (one-hot encoding, embeddings)
- Узнали про cross-features
- Рассмотрели несколько примеров (FM, FFM, Deep & Cross, DeepCrossing, DeepFM)

References I

[1] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[2] Y. Juan, D. Lefortier, and O. Chapelle. Field-aware factorization machines in a real-world online advertising system. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 680–688. International World Wide Web Conferences Steering Committee, 2017.

[3] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin. Field-aware factorization machines for ctr prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 43–50. ACM, 2016.

[4] S. Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

[5] S. Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):1–22, 2012.

References II

[6] Y. Shan, T. R. Hoens, J. Jiao, H. Wang, D. Yu, and J. Mao. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 255–262. ACM, 2016.

[7] R. Wang, B. Fu, G. Fu, and M. Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, page 12. ACM, 2017.

[8] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617*, 2017.