Лекция
Online advertising 2

Владимир Гулин

10 апреля 2020 г.

План лекции

Kaggle ads click challenges

SOTA

Practical Lessons from industry companies

Kaggle ads click challenges

Kaggle ads click challenges

Criteo leaderboard

1	—	3 Idiots	0.44463	13	4y
2	—	Michael Jahrer and Jeong-Yoo...	0.44527	61	4y
3	—	beile	0.44610	67	4y

Avazu leaderboard

1	—	4 Idiots	0.3791384	273	3y
2	—	Owen	0.3803652	94	3y
3	—	Random Walker	0.3806351	242	3y

SOTA

Linear Prediction Models

$$\hat{y} = f(\boldsymbol{w}^T \boldsymbol{x})$$

- Pros
  - Highly efficient and scalable
  - Explore larger feature space and training data
- Cons
  - Modelling limit: feature independence assumption
  - Cannot capture feature interactions unless defining high order combination features
    - E.g., hour=10AM & city=London & browser=Chrome

Non-linear Models

- Factorization Machines
- Gradient Boosting Decision Trees
- Combined Models
- Deep Neural Networks

Factorization Machines

- Prediction based on feature embedding

$$y_{\mathrm{FM}}(\boldsymbol{x}) := \mathrm{sigmoid}\Big(\underbrace{w_0 + \sum_{i=1}^{N} w_i x_i}_{\text{Logistic Regression}} + \underbrace{\sum_{i=1}^{N}\sum_{j=i+1}^{N} \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle x_i x_j}_{\text{Feature Interactions}}\Big)$$

For x=[Weekday=Friday, Gender=Male, City=Shanghai]

$$y_{\mathrm{FM}}(\boldsymbol{x}) = \mathrm{sigmoid}\Big(w_0 + w_{\mathrm{Friday}} + w_{\mathrm{Male}} + w_{\mathrm{Shanghai}} + \langle \boldsymbol{v}_{\mathrm{Friday}}, \boldsymbol{v}_{\mathrm{Male}} \rangle + \langle \boldsymbol{v}_{\mathrm{Friday}}, \boldsymbol{v}_{\mathrm{Shanghai}} \rangle + \langle \boldsymbol{v}_{\mathrm{Male}}, \boldsymbol{v}_{\mathrm{Shanghai}} \rangle\Big)$$

[Rendle. Factorization machines. ICDM 2010.]

[Oentaryo et al. Predicting response in mobile advertising with hierarchical importance-aware factorization machine. WSDM 14]

Field-aware Factorization Machines

- Feature embedding for another field

$$y_{\mathrm{FFM}}(\boldsymbol{x}) = \mathrm{sigmoid}\left(w_0 + \sum_{i=1}^{N} w_i + \boxed{\sum_{i=1}^{N}\sum_{j=i+1}^{N} \langle \boldsymbol{v}_{i,\mathrm{field}(j)}, \boldsymbol{v}_{j,\mathrm{field(i)}} \rangle x_i x_j}\right)$$

Field-aware field embedding

For x=[Weekday=Friday, Gender=Male, City=Shanghai]

$$\begin{aligned} y_{\mathrm{FFM}}(\boldsymbol{x}) = \mathrm{sigmoid}\Big(& w_0 + w_{\mathrm{Friday}} + w_{\mathrm{Male}} + w_{\mathrm{Shanghai}} \\ & + \langle \boldsymbol{v}_{\mathrm{Friday,Gender}}, \boldsymbol{v}_{\mathrm{Male,Weekday}} \rangle + \langle \boldsymbol{v}_{\mathrm{Friday,City}}, \boldsymbol{v}_{\mathrm{Shanghai,Weekday}} \rangle \\ & + \langle \boldsymbol{v}_{\mathrm{Male,City}}, \boldsymbol{v}_{\mathrm{Shanghai,Gender}} \rangle \Big) \end{aligned}$$

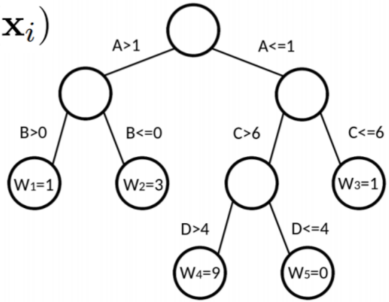
[Juan et al. Field-aware Factorization Machines for CTR Prediction. RecSys 2016.]

Gradient Boosting

- Additive decision trees for prediction

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F}$$

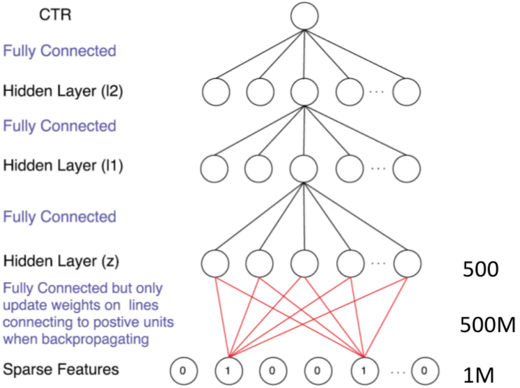
- Each decision tree $f_k(\mathbf{x}_i)$



[Chen and He. Higgs Boson Discovery with Boosted Trees . HEPML 2014.]

Neural Networks Models

- Difficulty: Impossible to directly deploy neural network models on such data



E.g., input features 1M, first layer 500, then 500M parameters for first layer

Review Factorization Machines

- Prediction based on feature embedding

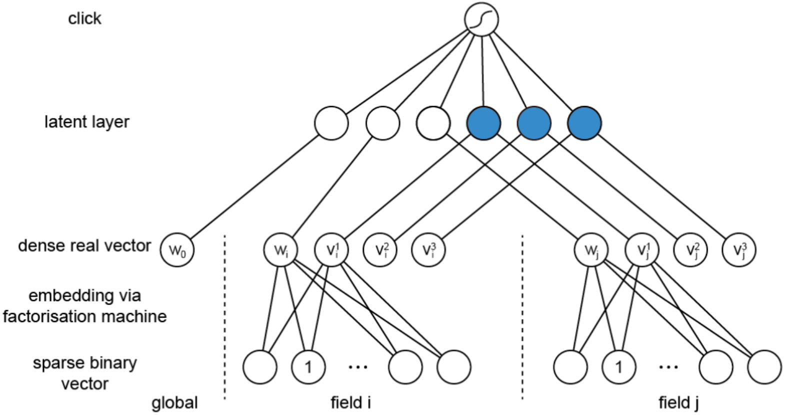
$$y_{\mathrm{FM}}(\boldsymbol{x}) := \mathrm{sigmoid}\Big(\underbrace{w_0 + \sum_{i=1}^{N} w_i x_i}_{\text{Logistic Regression}} + \underbrace{\sum_{i=1}^{N} \sum_{j=i+1}^{N} \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle x_i x_j}_{\text{Feature Interactions}}\Big)$$

  - Embed features into a k-dimensional latent space
  - Explore the feature interaction patterns using vector inner-product

[Rendle. Factorization machines. ICDM 2010.]

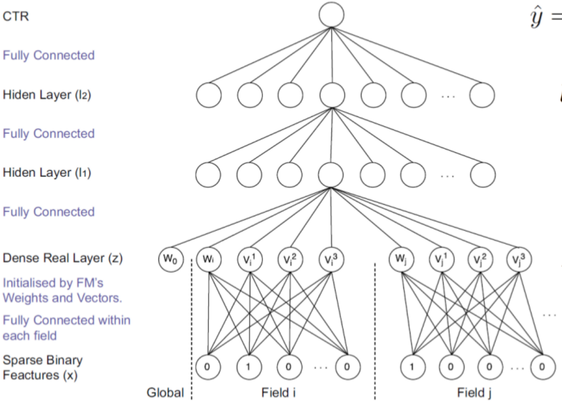
[Oentaryo et al. Predicting response in mobile advertising with hierarchical importance-aware factorization machine. WSDM 14]

Factorization Machines is a Neural Networks



$$y_{\mathrm{FM}}(\boldsymbol{x}) := \operatorname{sigmoid}\Big(w_0 + \sum_{i=1}^{N} w_i x_i + \sum_{i=1}^{N}\sum_{j=i+1}^{N} \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle x_i x_j\Big)$$

Factorization-Machine supported Neural Networks(FNN)



$$\hat{y} = \mathrm{sigmoid}(\boldsymbol{W}_3 \boldsymbol{l}_2 + b_3)$$

$$\boldsymbol{l}_2 = \tanh(\boldsymbol{W}_2 \boldsymbol{l}_1 + \boldsymbol{b}_2)$$

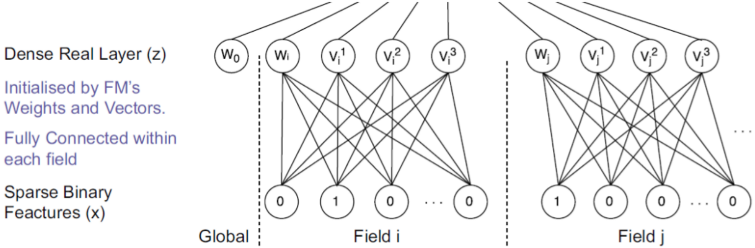
$$\boldsymbol{l}_1 = \tanh(\boldsymbol{W}_1 \boldsymbol{z} + \boldsymbol{b}_1)$$

$$\boldsymbol{z}_i = (w_i, v_i^1, v_i^2, \ldots, v_i^K)$$
$$= \boldsymbol{W}_0^i \cdot \boldsymbol{x}[\mathrm{start}_i : \mathrm{end}_i]$$

[Factorisation Machine Initialised]

[Zhang et al. Deep Learning over Multi-field Categorical Data – A Case Study on User Response Prediction. ECIR 16]

Factorization-Machine supported Neural Networks(FNN)



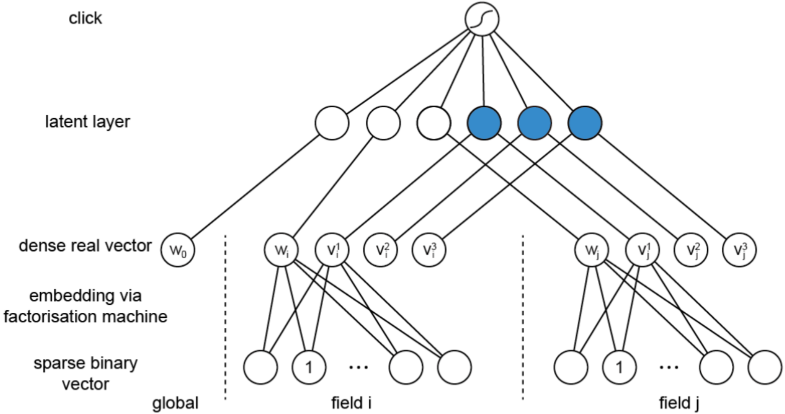
- Chain rule to update factorisation machine parameters

$$\frac{\partial L(y,\hat{y})}{\partial \boldsymbol{W}_0^i} = \frac{\partial L(y,\hat{y})}{\partial \boldsymbol{z}_i}\frac{\partial \boldsymbol{z}_i}{\partial \boldsymbol{W}_0^i} = \frac{\partial L(y,\hat{y})}{\partial \boldsymbol{z}_i}\boldsymbol{x}[\mathrm{start}_i : \mathrm{end}_i]$$

$$\boldsymbol{W}_0^i \leftarrow \boldsymbol{W}_0^i - \eta \cdot \frac{\partial L(y,\hat{y})}{\partial \boldsymbol{z}_i}\boldsymbol{x}[\mathrm{start}_i : \mathrm{end}_i].$$

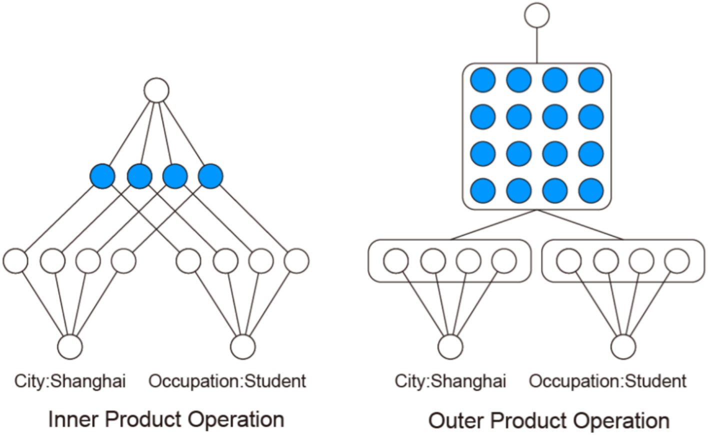
[Zhang et al. Deep Learning over Multi-field Categorical Data – A Case Study on User Response Prediction. ECIR 16]

Factorization-Machine different from Neural Networks



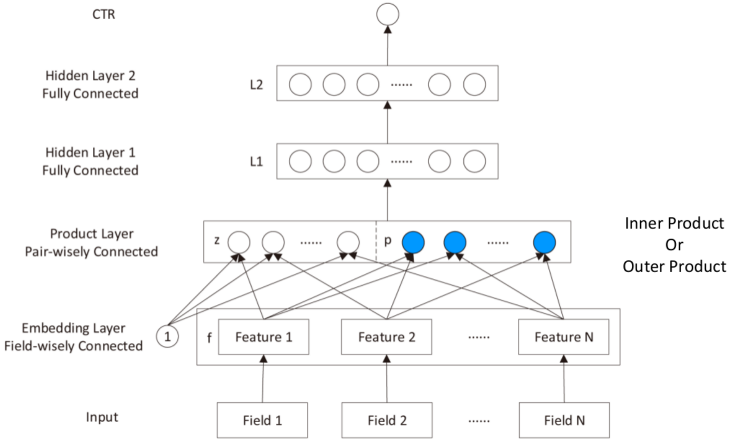
$$y_{\mathrm{FM}}(\boldsymbol{x}) := \mathrm{sigmoid}\Big(w_0 + \sum_{i=1}^{N} w_i x_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle x_i x_j\Big)$$

Product Operations as Feature Interactions



[Yanru Qu et al. Product-based Neural Networks for User Response Prediction. ICDM 2016]

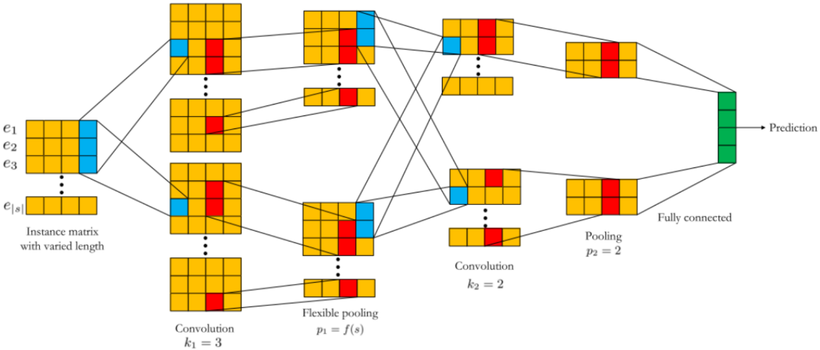
Product-based Neural Networks (PNN)



[Yanru Qu et al. Product-based Neural Networks for User Response Prediction. ICDM 2016]

Convolutional Click Prediction Model

- CNN to (partially) select good feature combinations



[Qiang Liu et al. A convolutional click prediction model. CIKM 2015]

Overall comparisons

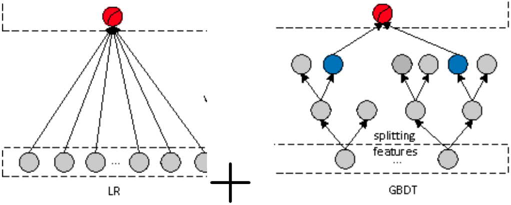
Model	AUC		Log Loss	
	Criteo	iPinYou	Criteo	iPinYou
LR	71.48%	73.43%	0.1334	5.581e-3
FM	72.20%	75.52%	0.1324	5.504e-3
FNN	75.66%	76.19%	0.1283	5.443e-3
CCPM	76.71%	76.38%	0.1269	5.522e-3
PNN-I	**77.79%**	79.14%	**0.1252**	5.195e-3
PNN-II	77.54%	**81.74%**	0.1257	5.211e-3
PNN-III	77.00%	76.61%	0.1270	**4.975e-3**

Model	RMSE		RIG	
	Criteo	iPinYou	Criteo	iPinYou
LR	9.362e-4	5.350e-07	6.680e-2	7.353e-2
FM	9.284e-4	5.343e-07	7.436e-2	8.635e-2
FNN	9.030e-4	5.285e-07	1.024e-1	9.635e-2
CCPM	8.938e-4	5.343e-07	1.124e-1	8.335e-2
PNN-I	**8.803e-4**	4.851e-07	**1.243e-1**	1.376e-1
PNN-II	8.846e-4	5.293e-07	1.211e-1	1.349e-1
PNN-III	8.988e-4	**4.819e-07**	1.118e-1	**1.740e-1**

Practical Lessons From Industry Companies

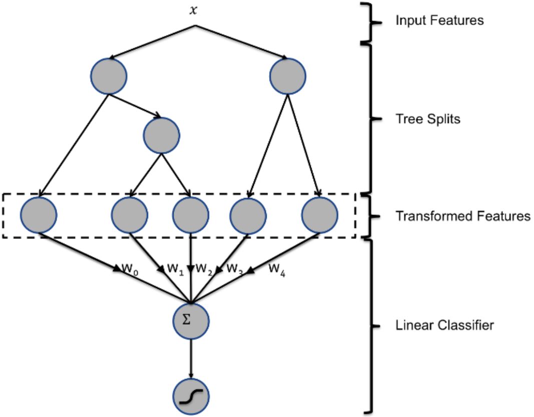
Yandex (2012)

Бустинг логрегрессии деревьями



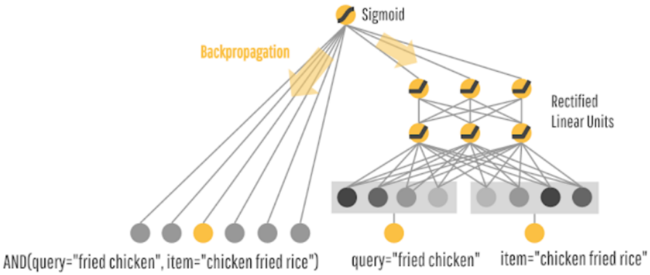
Facebook (2014)

Дообучаем деревья логрегрессией



Google (2016)

Совместно обучаем логрегрессию и нейронку



Microsoft Bing (2017)

Бустим нейронку деревьями

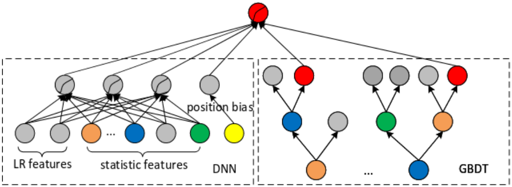


Figure 3: DNN+GBDT.

Microsoft Bing (2017)

Models	Position=ML1		Position=ALL		Description
	AUC Gain	RIG Gain	AUC Gain	RIG Gain	
NN	0.00%	0.00%	0.00%	0.00%	NN with 1 hidden layer and 30 hidden units (baseline model)
LR	-1.97%	-16.14%	-1.46%	-10.01%	LR with normalized position bias
LR V2	-1.81%	-10.68%	-0.91%	-5.13%	LR with inversed position bias
GBDT2LR	0.06%	-0.17%	0.05%	0.44%	Cascade leaf index in GBDT as categorical feature to LR (used in Facebook [
LR+GBDT	0.12%	-1.87%	-0.33%	-1.93%	Boost LR with GBDT (used in Yandex [25])
LR2GBDT V2	0.13%	-0.14%	0.03%	0.67%	Cascade LR with inversed position bias to GBDT
GBDT	0.14%	0.36%	0.03%	0.91%	GBDT initialized with inversed position bias
LR2GBDT	0.14%	-0.27%	0.01%	0.50%	Cascade LR with normalized position bias to GBDT
GBDT2NN	0.16%	1.29%	0.04%	1.32%	Cascade GBDT to NN
LR+GBDT V2	0.24%	1.36%	0.07%	1.04%	Boost LR (inversed position bias) with GBDT
NN2GBDT	0.25%	0.15%	0.08%	0.72%	Cascade NN to GBDT
GBDT+DNN	0.25%	1.33%	0.15%	1.52%	Average NN and GBDT
NN+GBDT	0.40%	2.81%	0.15%	1.30%	Boost NN with GBDT

Вопросы