Huawei | Noah's Ark Lab

Memorize, Factorize, or be Naïve: Learning Optimal Feature Interaction Methods for CTR Prediction

Fuyuan Lyu[1,2*], Xing Tang[2*], Huifeng Guo[2], Ruiming Tang[2], Xiuqiang He[2], Rui Zhang[3] and Xue Liu[1]

[1]School of Computer Science, McGill University, Montreal, Canada, [2]Huawei Noah's Ark Lab, [3]www.ruizhang.info

Presenter: Fuyuan Lyu

Outline

- Background
- Method
- Result
- Discussion
- Conclusion

Background

CTR prediction

Features as input

- Click-through rate(CTR) prediction can be viewed as a binary classification problem given different features as input[1-4]

Feature Interaction

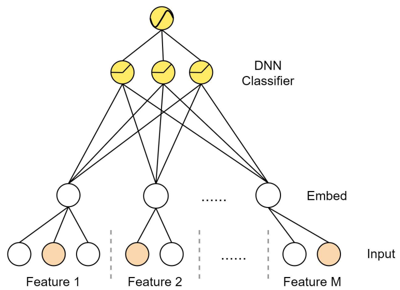
- Feature interaction is the most important question in CTR modeling [1-4]
- Modeling feature interaction correctly can significantly improve performance

[1] Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." *Proceedings of the 1st workshop on deep learning for recommender systems.* 2016.
[2] Guo, Huifeng, et al. "DeepFM: a factorization-machine based neural network for CTR prediction." *arXiv preprint arXiv:1703.04247* (2017).
[3] Qu, Yanru, et al. "Product-based neural networks for user response prediction." *2016 IEEE 16th International Conference on Data Mining (ICDM).* IEEE, 2016.
[4] Zhang, Weinan, Tianming Du, and Jun Wang. "Deep learning over multi-field categorical data." *European conference on information retrieval.* Springer, Cham, 2016.

Background

Feature Interaction

Naive | Memorize | Factorize



E.g: FNN[1], LR

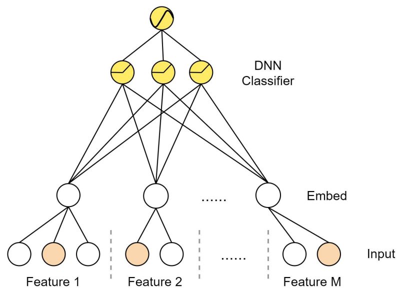
- Does not explicitly model feature interaction
- Rely on the capability of DNN to model feature interaction

[1] Zhang, Weinan, Tianming Du, and Jun Wang. "Deep learning over multi-field categorical data." *European conference on information retrieval.* Springer, Cham, 2016.

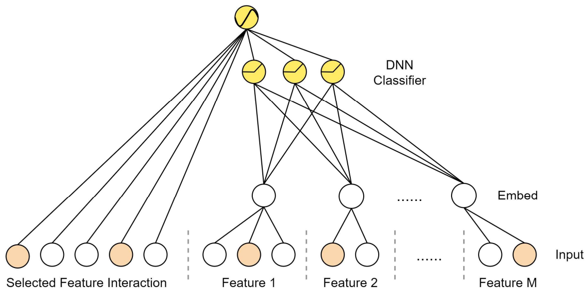
Background

Feature Interaction

Naive | Memorize | Factorize



E.g: FNN[1], LR



E.g: Wide & Deep[2], Poly2[3]

- Does not explicitly model feature interaction
- Rely on the capability of DNN to model feature interaction

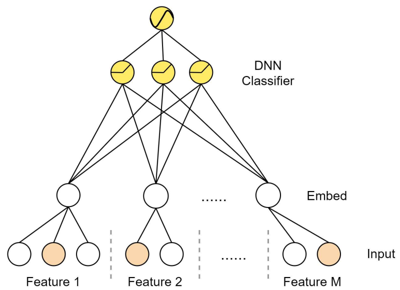
- Use cross-product transformation to memorize part or all feature interaction

[1] Zhang, Weinan, Tianming Du, and Jun Wang. "Deep learning over multi-field categorical data." *European conference on information retrieval.* Springer, Cham, 2016.
[2] Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." *Proceedings of the 1st workshop on deep learning for recommender systems.* 2016.
[3] Chang, Yin-Wen, et al. "Training and testing low-degree polynomial data mappings via linear SVM." *Journal of Machine Learning Research* 11.4 (2010).

Background

Feature Interaction

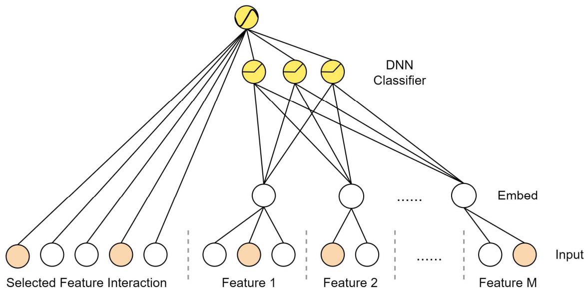
Naive



E.g: FNN[1], LR

- Does not explicitly model feature interaction
- Rely on the capability of DNN to model feature interaction

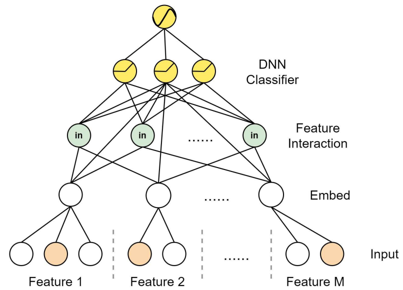
Memorize



E.g: Wide & Deep[2], Poly2[3]

- Use cross-product transformation to memorize part or all feature interaction

Factorize



E.g: FM[4], PNN[5], DeepFM[6]

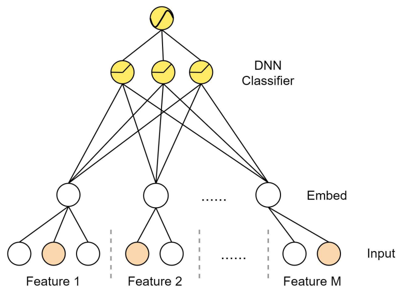
- Learn latent vector of original features to model feature interaction implicitly

[1] Zhang, Weinan, Tianming Du, and Jun Wang. "Deep learning over multi-field categorical data." *European conference on information retrieval.* Springer, Cham, 2016.
[2] Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." *Proceedings of the 1st workshop on deep learning for recommender systems.* 2016.
[3] Chang, Yin-Wen, et al. "Training and testing low-degree polynomial data mappings via linear SVM." *Journal of Machine Learning Research* 11.4 (2010).
[4] Rendle, Steffen. "Factorization machines." 2010 IEEE International conference on data mining. IEEE, 2010.
[5] Qu, Yanru, et al. "Product-based neural networks for user response prediction." *2016 IEEE 16th International Conference on Data Mining (ICDM).* IEEE, 2016.
[6] Guo, Huifeng, et al. "DeepFM: a factorization-machine based neural network for CTR prediction." *arXiv preprint arXiv:1703.04247* (2017).

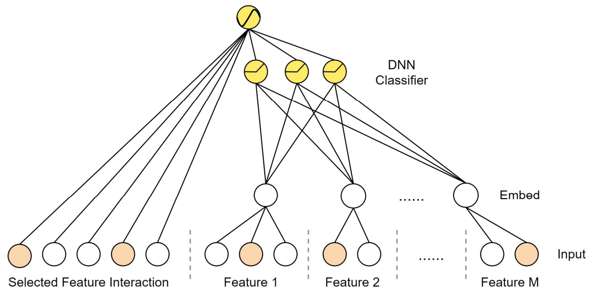
Background

Feature Interaction

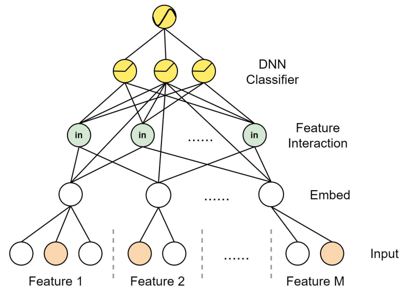
Naive	Memorize	Factorize



E.g: FNN, LR



E.g: Wide & Deep, Poly2



E.g: FM, PNN, DeepFM, CAN

- Does not explicitly model feature interaction
- Rely on the capability of DNN to model feature interaction

Limited model capability

- Use cross-product transformation to memorize part or all feature interaction

Need feature engineering or easy to overfit

- Learn latent vector of original features to model feature interaction implicitly

Different feature may influence each other

Background

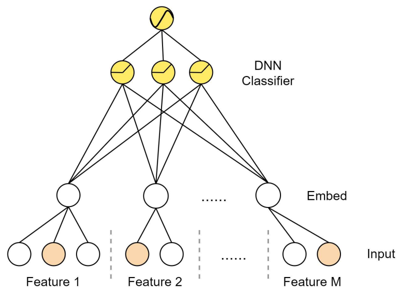
OptInter: searching for optimal feature interaction methods

Feature Interaction

Naive

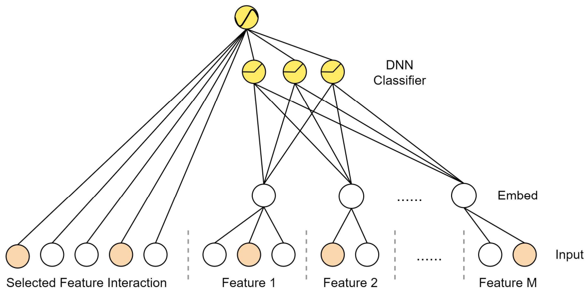
Memorize

Factorize



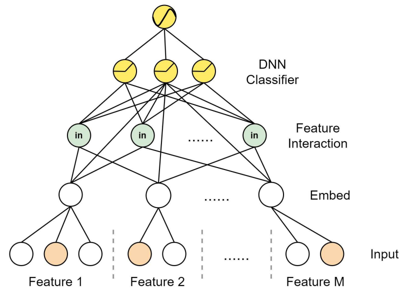
E.g: FNN, LR

Limited model capability



E.g: Wide & Deep, Poly2

Need feature engineering or easy to overfit



E.g: FM, PNN, DeepFM, CAN

Different feature may influence each other

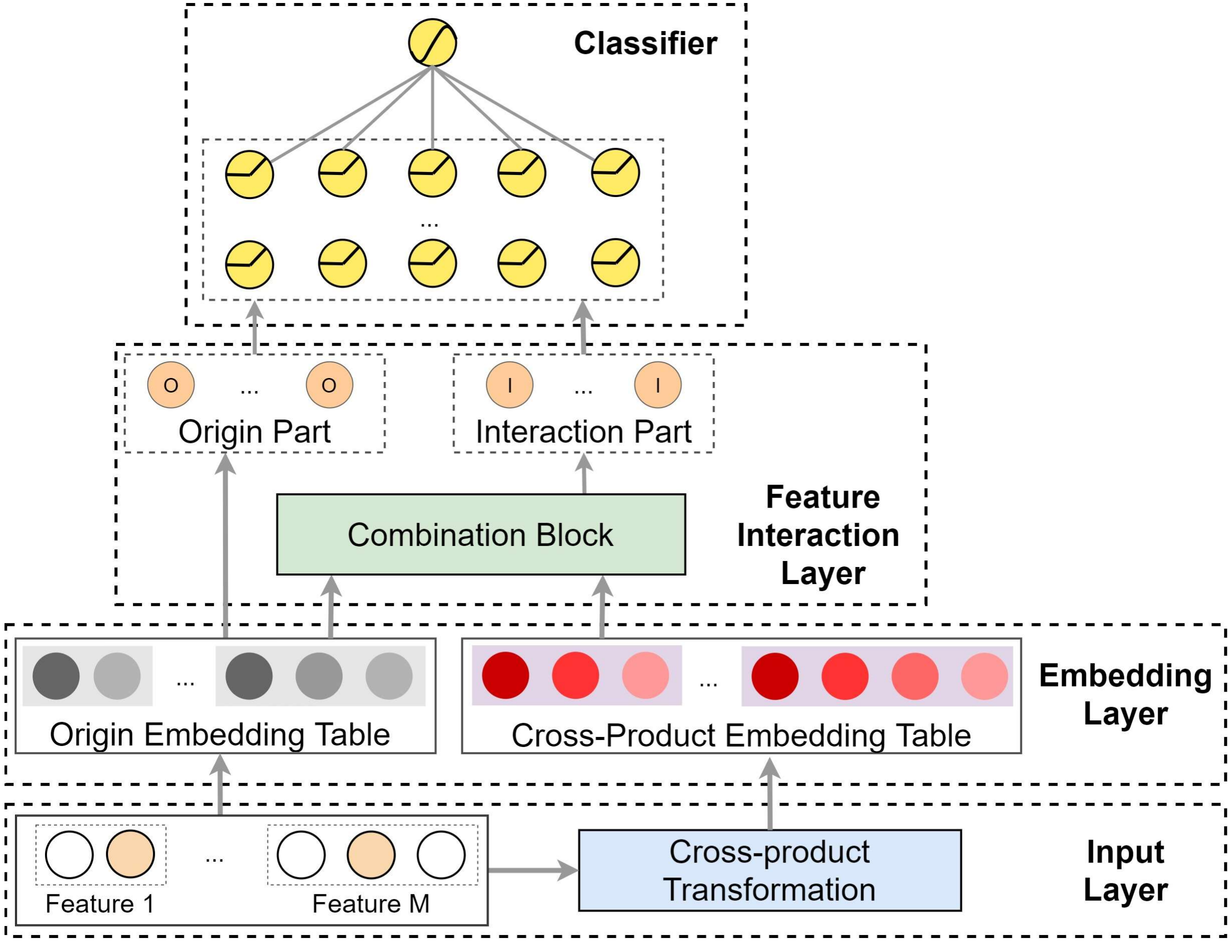
Different methods perform differently on various dataset

Method

A data-driven approach to search for optimal feature interaction methods

Only consider 2-nd order because:

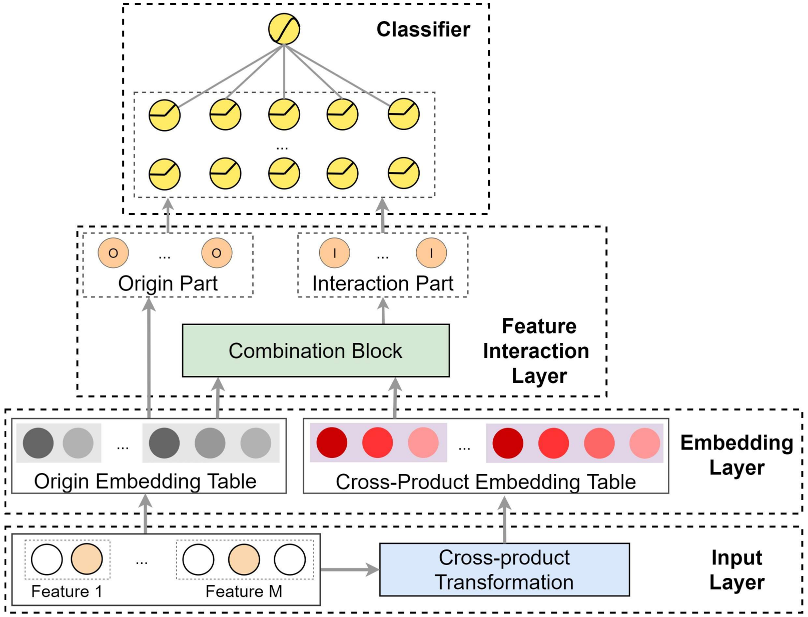
1. Empirical experience shows that 2-nd order is good enough for most cases[1-4]
2. Combination explosion for higher-order



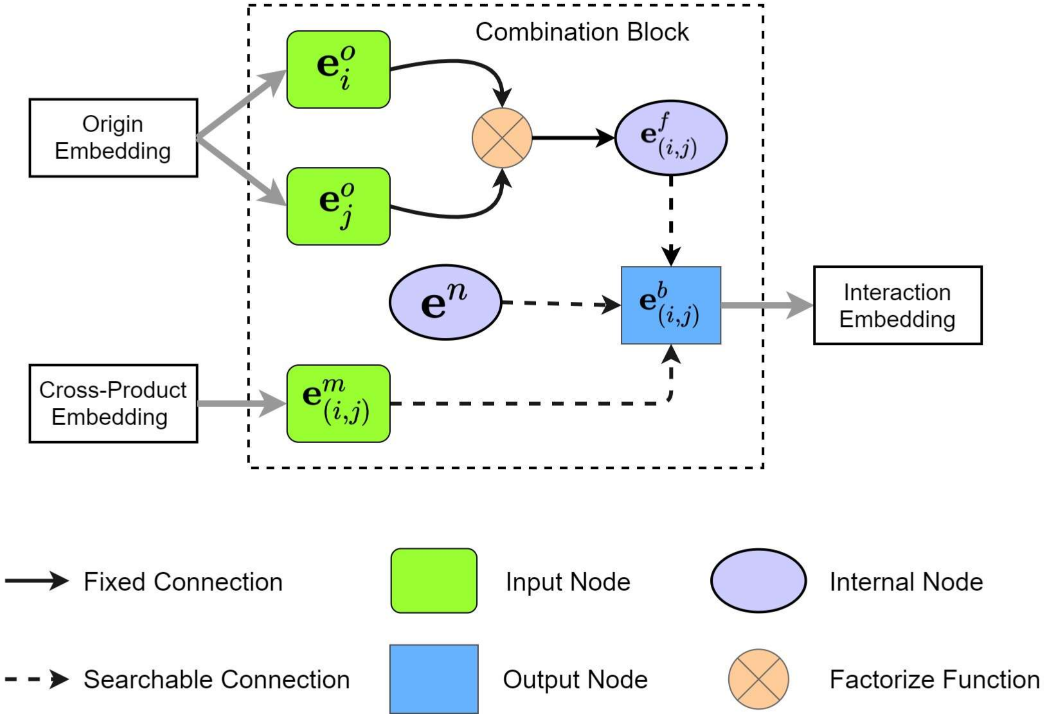
[1] Chang, Yin-Wen, et al. "Training and testing low-degree polynomial data mappings via linear SVM." *Journal of Machine Learning Research* 11.4 (2010).
[2] Rendle, Steffen. "Factorization machines." 2010 IEEE International conference on data mining. IEEE, 2010.
[3] Qu, Yanru, et al. "Product-based neural networks for user response prediction." *2016 IEEE 16th International Conference on Data Mining (ICDM).* IEEE, 2016.
[4] Guo, Huifeng, et al. "DeepFM: a factorization-machine based neural network for CTR prediction." *arXiv preprint arXiv:1703.04247* (2017).

Method

A data-driven approach to search for optimal feature interaction methods



Relax to discrete optimization to continuous optimization



Method

Using Gumbel-softmax[1] trick to approximate categorical distribution

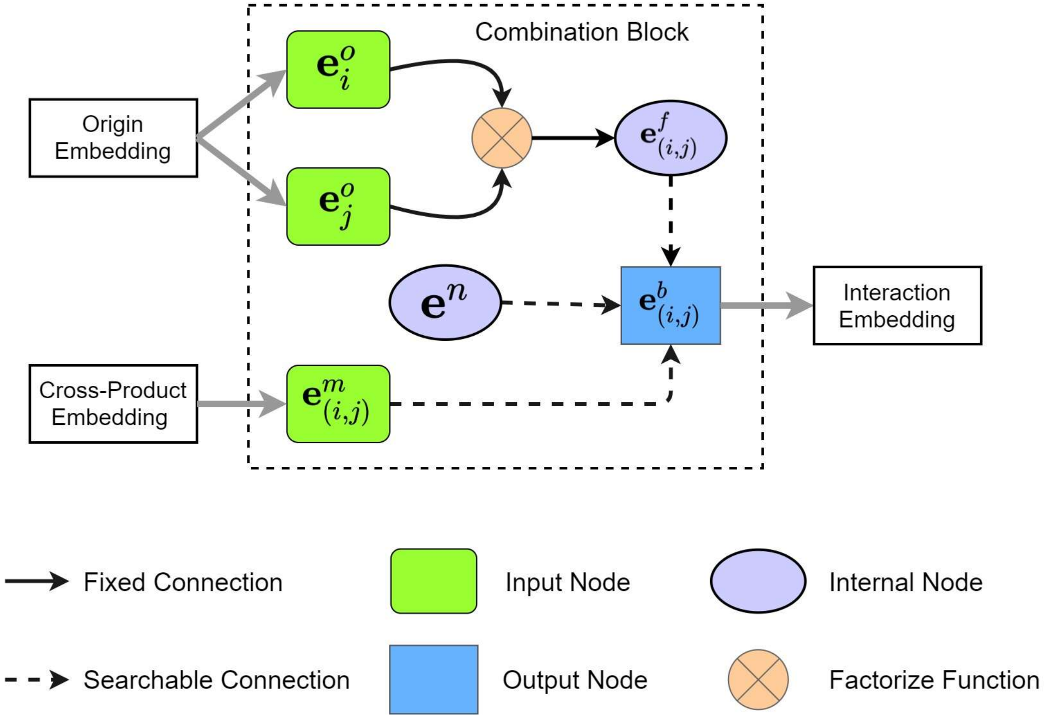
(a) searching

$$\mathbf{e}^b_{(i,j)} = \sum_{k\in\mathcal{K}} p^k_{(i,j)} \cdot \mathbf{e}^k_{(i,j)}$$
$$= p^m_{(i,j)} \cdot \mathbf{e}^m_{(i,j)} + p^f_{(i,j)} \cdot \mathbf{e}^f_{(i,j)} + p^n_{(i,j)} \cdot \mathbf{e}^n$$

(b) re-training

$$\mathbf{e}^b_{(i,j)} = \mathbf{e}^{k^*}_{(i,j)},$$
$$\text{s.t. } k^* = \text{argmax}_{k\in\mathcal{K}} p^k_{(i,j)}.$$

Relax to discrete optimization to continuous optimization



[1] Jang, Eric, Shixiang Gu, and Ben Poole. "Categorical reparameterization with gumbel-softmax." arXiv preprint arXiv:1611.01144 (2016).

Method

OptInter as a framework

Category	Model	Feature Interaction Layer: Method	Feature Interaction Layer: Func.	Classifier
naïve	LR [24]	$\{n\}$	-	Shallow
	FNN [5]	$\{n\}$	-	Deep
memorized	Poly2 [8]	$\{m\}$	-	Shallow
	Wide&Deep [1]	$\{m\}$	-	S&D
factorized	FM [9]	$\{f\}$	$\langle \mathbf{e}_i^o, \mathbf{e}_j^o \rangle$	Shallow
	FwFM [11]	$\{f\}$	$\langle \mathbf{e}_i^o, \mathbf{e}_j^o \rangle w_{(i,j)}$	Shallow
	FmFM [12]	$\{f\}$	$\mathbf{e}_i^o W_{(i,j)} \mathbf{e}_j^{oT}$	Shallow
	IPNN [3]	$\{f\}$	$\langle \mathbf{e}_i^o, \mathbf{e}_j^o \rangle$	Deep
	OPNN [3]	$\{f\}$	$\langle \mathbf{e}_i^o, \mathbf{e}_j^o \rangle_\phi$	Deep
	DeepFM [2]	$\{f\}$	$\langle \mathbf{e}_i^o, \mathbf{e}_j^o \rangle$	Deep
	PIN [4]	$\{f\}$	$\text{net}(\mathbf{e}_i^o, \mathbf{e}_j^o)$	Deep
hybrid	AutoFIS [15]	$\{n, f\}$	flexible	Deep
	OptInter	$\{n, m, f\}$	flexible	Deep

f: factorize
m: memorize
n: naive

Result

Dataset	#samples	#cont	#cate	#cross	#orig value	#cross value	pos ratio
Criteo	4.6×10^7	13	26	325	5.1×10^5	3.7×10^7	0.23
Avazu	4.0×10^7	0	24	276	1.2×10^6	2.4×10^8	0.17
iPinYou	1.9×10^7	0	16	120	9.4×10^5	6.8×10^7	0.0008
Private	8.0×10^8	0	9	36	4.0×10^5	7.1×10^7	0.17

Dataset	Criteo			Avazu			iPinYou			Private		
Metric	AUC	Log loss	Param.	AUC	Log loss	Param.	AUC	Log loss	Param.	AUC	log Loss	Param.
LR	0.7785	0.4708	0.5M	0.7685	0.3862	1.2M	0.7565	0.005685	0.9M	0.7690	0.3836	0.4M
FNN	0.7996	0.4512	13M	0.7858	0.3761	51M	0.7779	0.005627	19M	0.8348	0.3353	32M
FM	0.7845	0.4681	10M	0.7826	0.3790	49M	0.7779	0.005574	19M	0.8304	0.3406	32M
IPNN	0.8005	0.4504	13M	0.7887	0.3745	51M	0.7786	0.005644	19M	0.8410	0.3303	32M
DeepFM	0.7997	0.4512	13M	0.7860	0.3760	51M	0.7789	0.005636	19M	0.8383	0.3325	32M
PIN	0.8016	0.4508	17M	0.7825	0.3789	52M	0.7779	0.005574	20M	0.8331	0.3365	33M
OptInter-F	0.8003	0.4507	21M	0.7860	0.3761	56M	0.7762	0.005688	23M	0.8380	0.3325	37M
Poly2	0.7827	0.4751	22M	0.7860	0.3795	241M	0.7743	0.005578	69M	0.8307	0.3390	71M
OptInter-M	$\boxed{0.8094}$	0.4423	225M	$\boxed{0.8061}$	0.3638	1012M	$\boxed{0.7800}$	0.005640	296M	$\boxed{0.8415}$	0.3265	738M
AutoFIS	0.8014	0.4514	22M	0.7861	0.3758	51M	0.7792	0.005620	19M	0.8413	0.3299	32M
OptInter	**0.8101***	**0.4417***	100M	**0.8062***	**0.3637***	827M	**0.7825***	**0.005606***	26M	**0.8425***	**0.3256***	302M

- Memorize is very effective
- OptInter can automatically select optimal modeling methods for each feature interaction on different datasets

Result

Dataset	Criteo					Avazu				
Model	FM	FNN	IPNN	DeepFM	OptInter	FNN	FM	IPNN	DeepFM	OptInter
AUC	0.7543	0.7990	0.8014	0.7678	**0.8101***	0.7677	0.7848	0.7923	0.7691	**0.8062***
log loss	0.5192	0.4516	0.4495	0.5075	**0.4417***	0.3947	0.3768	0.3723	0.3934	**0.3637***
Orig.E.	200	200	200	200	20	700	700	700	700	40
Cross.E.	0	0	0	0	10	0	0	0	0	4
Param.	103M	109M	109M	109M	100M	860M	953M	954M	953M	827M

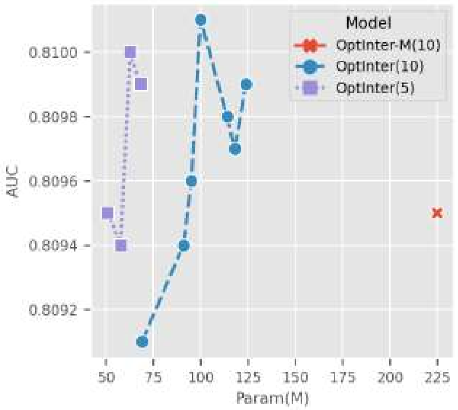
Orig. E. : Original embedding length
Cross. E. : Cross-product embedding length

- The increase of param size does not necessary leads to the increase of model performance
- OptInter can better utilize params than other SOTA methods

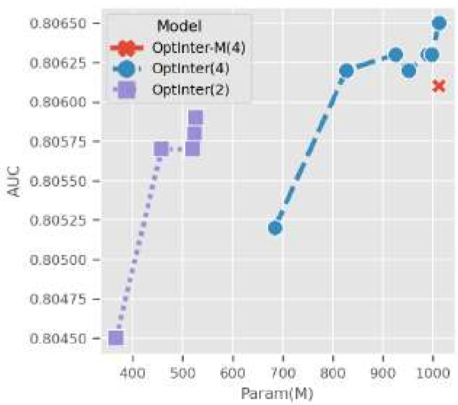
Result

Dataset	Model	AUC	log loss	Arch	Param.
Criteo	Random	0.8089	0.3764	-	84M
	Bi-level	0.8099	0.3741	[114,109,104]	95M
	OptInter	**0.8101**	**0.3760**	[117,98,110]	100M
Avazu	Random	0.8030	0.3658	-	418M
	Bi-level	Out of Memory			
	OptInter	**0.8062**	**0.3637**	[107,73,96]	827M
iPinYou	Random	0.7781	0.005734	[36,38,46]	108M
	Bi-level	0.7796	0.005620	[34,16,70]	31M
	OptInter	**0.7825**	**0.005606**	[25,12,83]	26M

- Our search algorithm is better than bi-level optimization



(a) Criteo



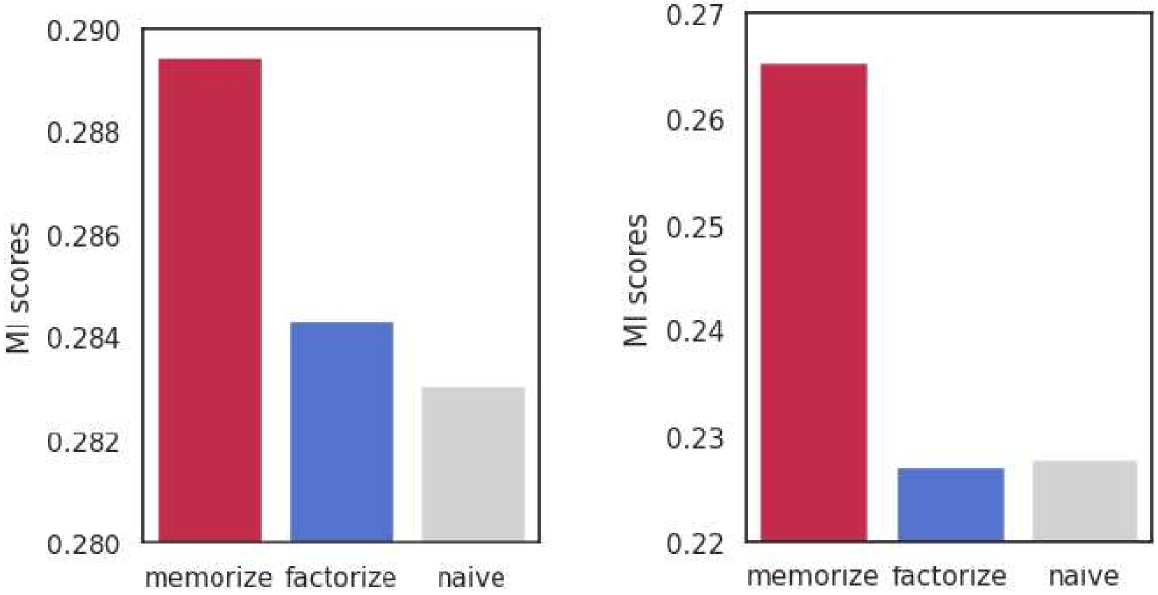
(b) Avazu

- Performance degrades dramatically when the model shrink below certain threshold
- Reducing embedding size is more effective when resource is constrained

Dataset	Criteo		Avazu	
Metric	w.	w.o.	w.	w.o.
AUC	0.8101	0.7953	0.8062	0.7772
log loss	0.3760	0.4558	0.3637	0.3829

- Retraining is necessary

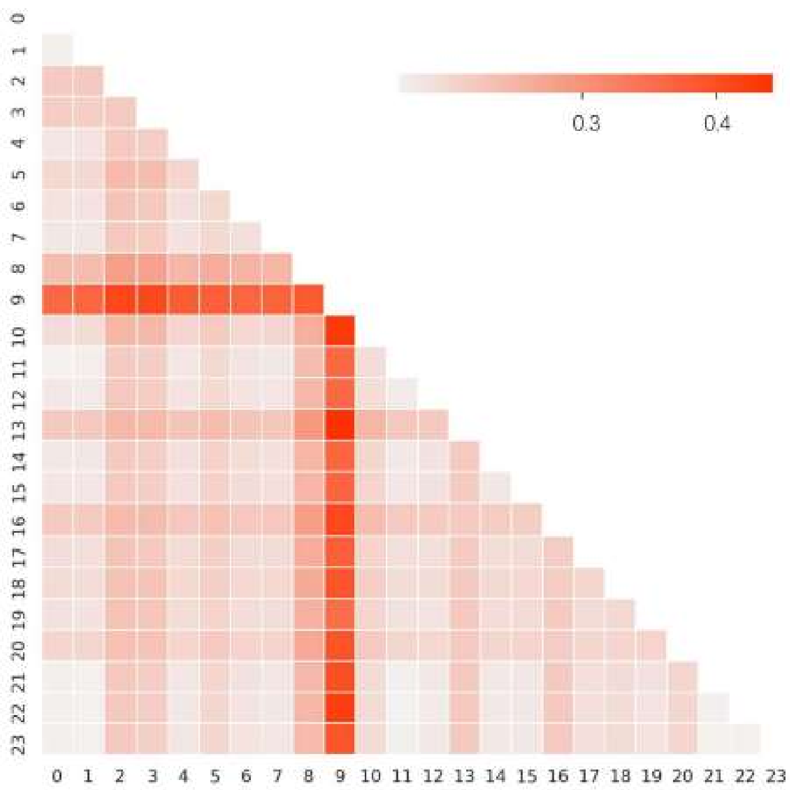
Discussion



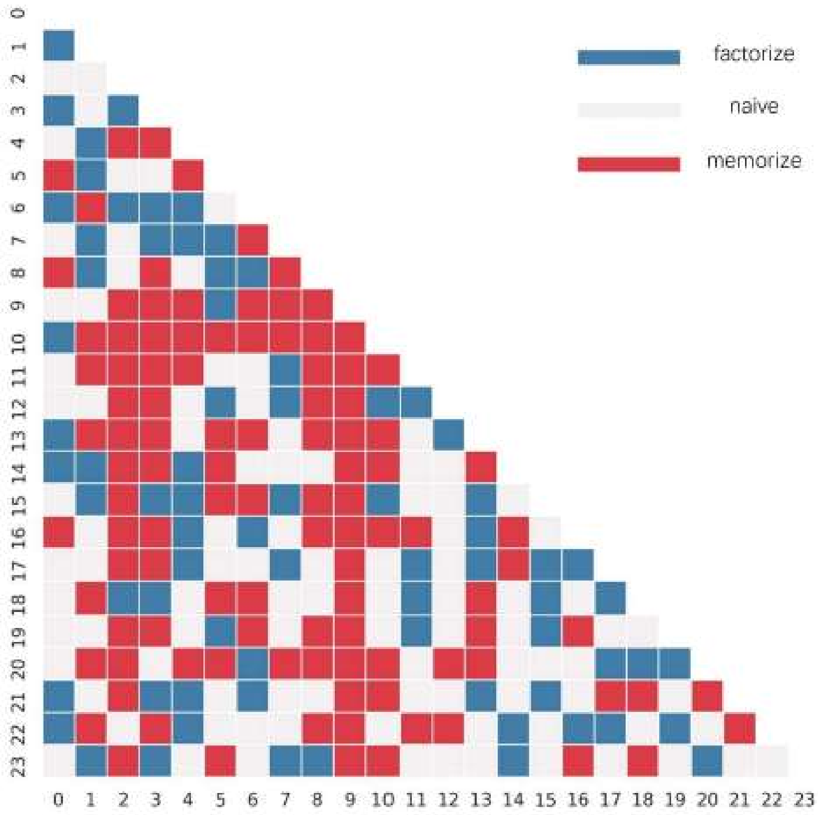
(a) Criteo (b) Avazu

- memorize feature interaction with **rich** information
- ignore feature interaction with **poor** information
- factorize different feature interaction **given dataset**

Discussion-Avazu Case Study



(a) MI scores between feature interactions and labels



(b) Obtained optimal methods for feature interactions

Conclusion

1. A data-driven framework OptInter including naïve, memorize, factorize feature interaction methods
2. A two-stage learning algorithm to select optimal modeling method for each feature interaction
3. Conduct comprehensive experiments on 3 public and 1 private datasets to demonstrate effectiveness and provide interpretability

Thanks for listening!