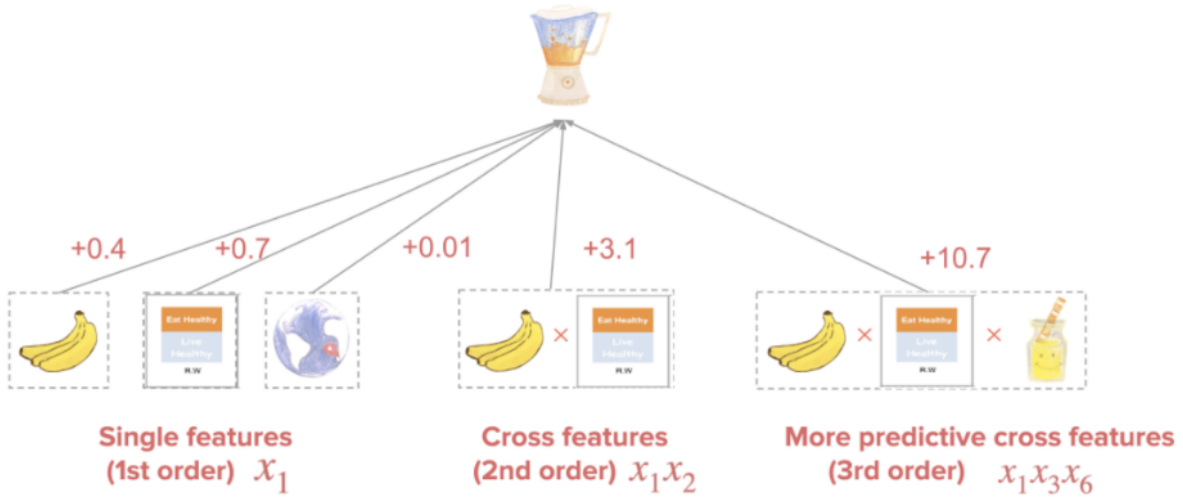
DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems

임도연

1. Introduction

Introduction

- Learning to rank(LTR)은 데이터와 정보가 흘러 넘치는 현대 정보검색 분야에서 추천시스템, 검색 분야, 광고 분야에서 머신러닝과 딥러닝과 결합되면서 더욱 광범위하게 발전하고 있는 분야가 됨.
- LTR 모델의 중요한 요소 중 하나는 효과적인 feature crosses를 통해 수학적으로 실무적으로 모델의 성능을 큰 기여를 할 수 있음.
- 효과적인 feature crosses는 개별 feature가 모델에 전달하는 영향 이외의 추가적인 상호작용 정보가 포함되어 있기 때문에 모델의 높은 성능에 있어 중요함.



Introduction

- 대부분의 데이터가 categorical로서 web-scale applications에 있는 large하고 sparse한 combinatorial search space를 포함함.
- 고차원 벡터를 저차원의 벡터로 투영하는 Embedding techniques가 발전되며 다양한 분야에서 사용됨.
- LTR 모델은 linear model과 FM 기반 모델에서 deep neural networks로 이동하고 있음.
- 또한 최근 연구를 통해 단순한 딥러닝 모델로는 2nd, 3rd feature cross를 추정하는 것에는 비효율적이라고 밝혀짐.
- Feature crosses를 찾기 위해 wider하고 deeper한 network를 통해 model capacity를 증가시킴.
- 기존 DCN의 경우 wider하고 deeper한 network를 기반으로 feature cross를 적용하는 것에 있어서는 효과적이였지만 large-scale 환경에서의 production에서는 어려움을 겪음 -> DCN-V2가 나타나게 된 계기

DCN-V2의 핵심은 cross layer를 통한 explicit feature interaction을 학습하고

deep layer를 통해implicit interaction을 학습한다.

2. Related Work

Related Work

최근 feature interaction learning work의 핵심 아이디어는 explicit과 implicit feature crosses를 활용하는 것

1. Parallel Structure

- Wide and deep model로부터 영감을 얻음

  wide component : raw feature들의 crosses를 input으로 받음

  deep component : DNN 모델

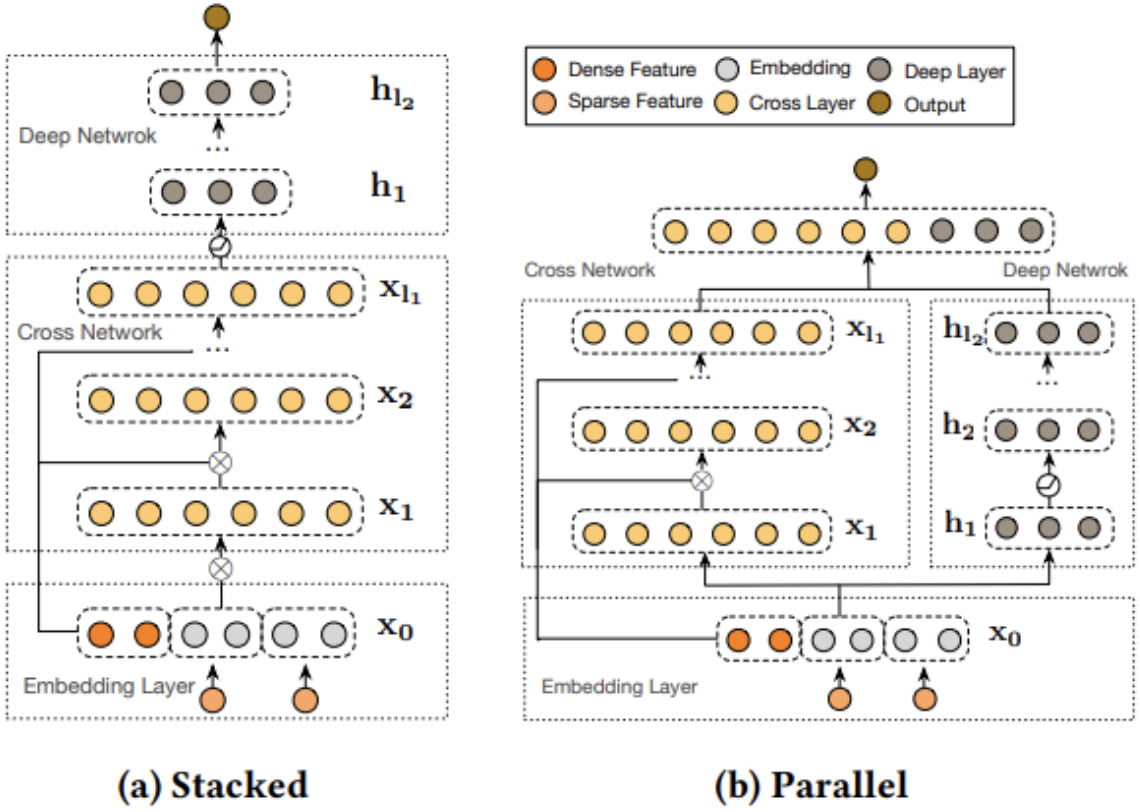
- Wide component를 위한 cross feature를 선별하는 과정에서 feature engineering 문제에 직면함.
- 하지만 wide component를 향상시키기 위해 이 모델을 많이 적용함.

2. Stacked Structure

- embedding layer와 DNN 모델 중간에서 explicit feature crosses를 만들어내는 interaction layer
- interaction layer은 초반에 feature interaction을 뽑아내고 다음 hidden layer의 학습을 용이하게 함.

3. Proposed Architecture: DCN-V2

Proposed Architecture: DCN-V2



(a) Stacked (b) Parallel

- embedding layer를 시작으로 explicit feature interacitons을 뽑아내는 multiple cross layer를 포함하는 cross layer 그리고 implicit feature interactions를 뽑아내는 deep network로 구성되어 있음.
- cross network와 deep network를 결합하는 방식의 차이로 stacked와 parallel 2가지 Structure이 존재

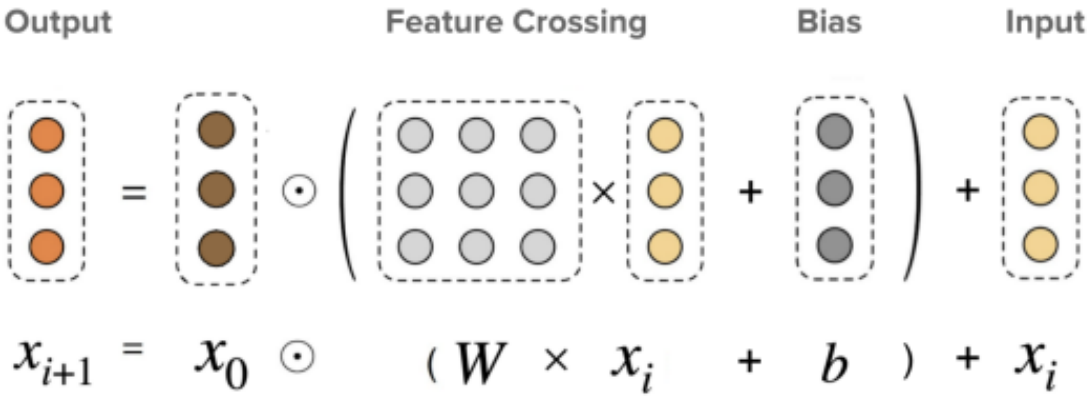
Proposed Architecture: DCN-V2

1. Embedding Layer

- Categorical과 dense feature이 input
- Embedded vector는 categorical feature의 임베딩과 dense feature의 정규화된 값이 concat되어 출력

2. Cross Network

- DCN-V2의 핵심은 explicit feature crosses를 뽑아내는 cross layer에 있음.
- 각 layer에서 발생하는 cross layer function을 시각화하여 표현한 그림(아래)



- 첫번째 cross layer의 계산의 경우 $x0$가 linear 계산을 통과한 결과와 $x0$가 element-wise product되면서 update되는 weight가 기존 feature간의 interaction 정보를 담고 있음

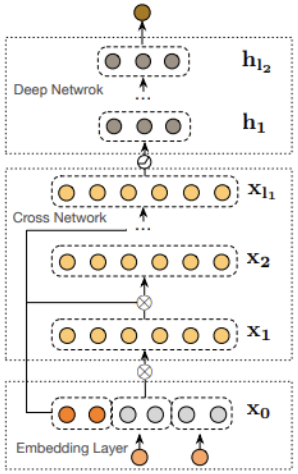
Proposed Architecture: DCN-V2

3. Deep Network

- Deep Network는 전형적인 feed-forward neural network로 linear 계산과 activation fuction으로 이루어져 있음

4. Deep and Cross Combination

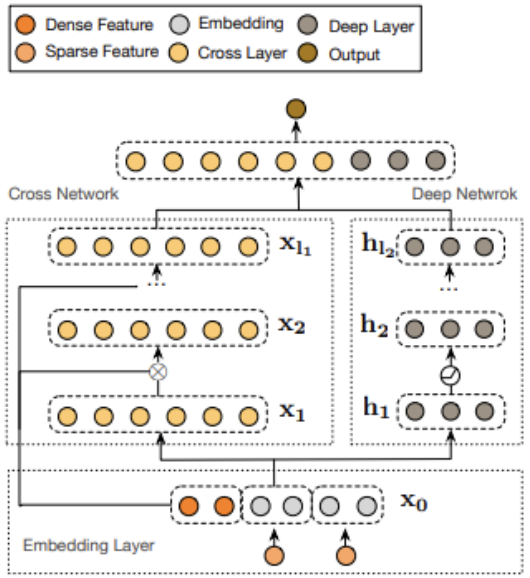
- Stacked Structure과 Parallel Structure 사이 data dependent에 따라 구조의 성능 차이가 발생



(a) Stacked

- x0가 cross network를 통과한 후 deep network를 통과하는 구조
- Cross network의 출력 값은 deep network의 입력 값에 해당

Proposed Architecture: DCN-V2



(b) Parallel

- X0이 cross network와 deep network에 동시에 입력되는 구조
- Cross network 출력과 deep network 출력 값이 concatenate으로 최종 output layer를 통과

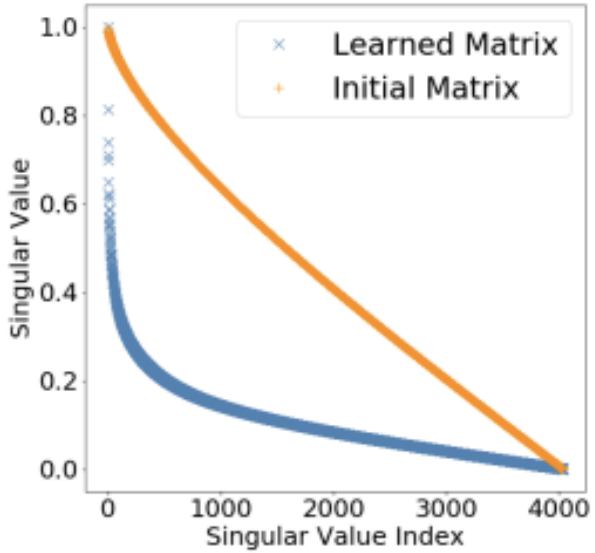
- Loss function으로는 binary label의 learning to rank system에서 주로 사용되는 Log Loss 사용

$$\text{loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) + \lambda \sum_{l} \|W_l\|_2^2,$$

Proposed Architecture: DCN-V2

5. Cost-Effective Mixture of Low-Rank DCN

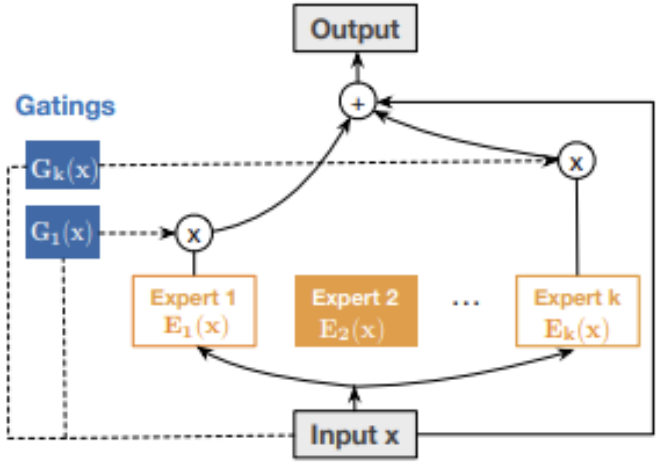
- 정확도를 유지하면서 cost를 줄일 수 있는 방법을 추구함.
- Low-rank techniques는 computational cost를 줄이기 위해 사용됨.



(a) Singular Values

- Learned matrix W의 singular decay pattern으로 initial matrix 와 비교해 더 빠른 spectrum decay pattern을 보여줌.

Proposed Architecture: DCN-V2



(b) Mixture of Low-rank Experts

- Mixture-of-Experts(MoE)에서 아이디어를 얻어옴.
- Experts와 gating 2가지 component으로 구성되어 있음.
- Single expert에 의존하는 것 대신 multiple experts를 사용
- Input x에 의존한 gating mechanism을 사용해 learned crosses를 결합함.

6. Complexity Analysis

- time and space complexity 관점에서 cross networ와 DCN-Mix를 비교했을 때 Rk << d일 경우 더 효율적임.

4. Model Analysis

Model Analysis

1. Polynomial Approximation

- 2가지 관점에서 DCN-V2 모델 분석
    1) Element xi를 unit으로, elements 사이 interaction 분석 -> bitwise
    2) Feature embedding을 xi unit으로, feature-wise interactions 분석 -> feature-wise

-> DCN과 DCN-V2를 비교했을 때 같은 polynomial class에서 더 많은 parameter를 가지고 있는 것이 더 expressive함.

- DCN은 bitwise만 적용한 반면, DCN-V2는 두가지 경우 모두 적용해 더 expressive함.

Model Analysis

2. Connections to Related Work

- DCN-V2와 다른 SOTA feature interaction 사이의 connection에 대해 공부함.
- 각 모델의 feature interaction component에 집중하고 DNN component는 무시함.

1) DCN
2) DLRM and DeepFM
3) xDeepFM
4) AutoInt
5) PNN

5. Research Questions

Research Questions

DCN-V2를 연구하며 아래 reaserach question을 따라 답을 구하려고 노력했다.

Q1. DNN 기반 ReLU 모델 보다 더 효율적인 feature interaction learning model이 될 수 있는가?

Q2. DNN과의 통합 대신 각 baseline의 feature interaction component이 어떻게 수행하는가?

Q3. baseline과 비교해 제안된 Mdcn은 어떻게 접근하는가?

모델의 정확도와 cost 사이에서 trade-off를 얻을 수 있을까?

Q4. mDCN에서의 settings는 어떻게 모델의 quality에 영향을 주는가?

Q5. mDCN은 중요한 feature crosses를 뽑아내는가?

이는 모델에 좋은 understandability를 제공하는가?

6. Empirical Understanding of Feature Crossing Techniques

Empirical Understanding of Feature Crossing Techniques

- 최근 많은 연구는 전통적인 neural networks에서는 효율적으로 학습되지 않은 model explicit feature crosses를 제안함.

1) 어떤 경우 전통적인 neural network가 비효율적이 되는지
2) DCN-V2의 cross network의 각 요소들의 역할

Table 1: RMSE and Model Size (# Parameters) for Polynomial Fitting of Increasing Difficulty.

	DCN (1Layer)		DCN-V2 (1Layer)		DNN (1Layer)		DNN (large)	
	RMSE	Size	RMSE	Size	RMSE	Size	RMSE	Size
f_1	8.9E-13	12	**5.1E-13**	24	2.7E-2	24	4.7E-3	41K
f_2	1.0E-01	9	**4.5E-15**	15	3.0E-2	15	1.4E-3	41K
f_3	2.6E+00	300	**6.7E-07**	10K	2.7E-1	10K	7.8E-2	758K

- Cross patterns이 단순한 f1에서는 DCN-V2, DCN 모두 효율적이나 f3과 같이 복잡해지면 DCN-V2는 정확성을 유지하나 DCN은 떨어짐.
- DNN의 성능은 wider and deep structure에서 좋지 않음.

Empirical Understanding of Feature Crossing Techniques

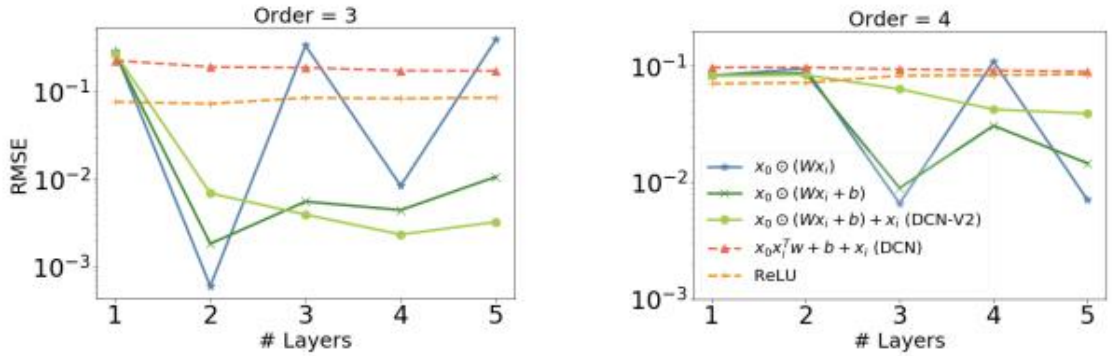


Figure 4: Homogeneous polynomial fitting of order 3 and 4. x-axis represents the number of layers used; y-axis represents RMSE (the lower the better). In the legend, the top 3 models are DCN-V2 with different component(s) included.

- Layer depth에 따른 mean RMSE 변화를 나타냄.
- Order-3 polynomial이 layer2에서 가장 좋은 성능을 보임.

Table 2: Combined-order (1 - 4) Polynomial Fitting.

#Layers	1	2	3	4	5
DCN-V2	1.43E-01	2.89E-02	**9.82E-03**	9.87E-03	9.92E-03
DNN	1.32E-01	1.03E-01	1.03E-01	1.09E-01	1.05E-01

7. Experimental Results

Experimental Results

3개의 dataset과 2가지의 platform을 통해 feature interaction learning에 있어 DCN-V2의 effectiveness를 증명했다.

Datasets

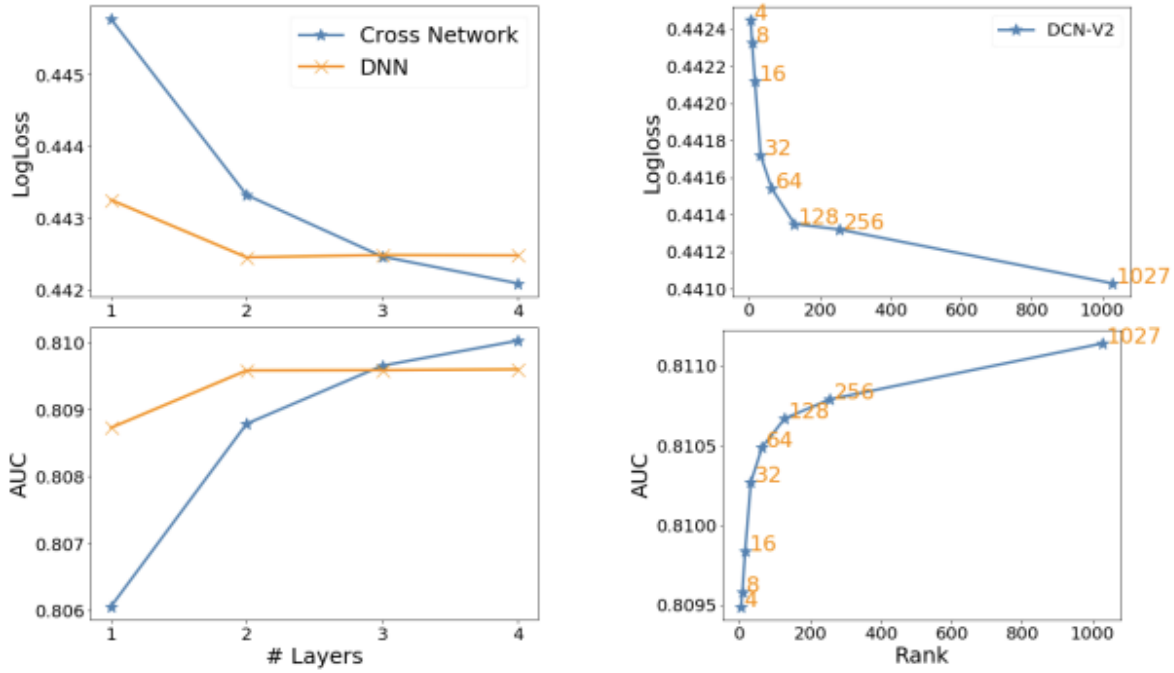
Table 3: Datasets.

Data	# Examples	# Features	Vocab Size
Criteo	45M	39	2.3M
MovieLen-1M	740k	7	3.5k
Production	> 100B	NA	NA

Table 5: LogLoss (test) of feature interaction component of each model (no DNN). Only categorical features were used. In the 'Setting' column, l stands for number of layers.

	Model	LogLoss	Best Setting
2nd	PNN [35]	0.4715 ± 4.430e-04	OPNN, kernel=matrix
	FM	0.4736 ± 3.04E-04	–
>2	CIN [26]	0.4719 ± 9.41E-04	l=3, cinLayerSize=100
	AutoInt [46]	0.4711 ± 1.62E-04	l=2, head=3, attEmbed=40
	DNN	0.4704 ± 1.57E-04	l=2, size=1024
	CrossNet	0.4702 ± 3.80E-04	l=2
	CrossNet-Mix	**0.4694 ± 4.35E-04**	l=5, expert=4, gate=$\frac{1}{1+e^{-x}}$

Experimental Results



(a) Layer depth (b) Matrix rank

Figure 5: Logloss and AUC (test) v.s. depth & matrix rank.

Layer가 2보다 작은 경우 DNN은 cross network에서 좋은 성능을 보였으나 layer가 많아질수록 cross network는 성능 차이를 줄여나감

Experimental Results

Table 6: LogLoss and AUC (test) on Criteo and Movielen-1M. The metrics were averaged over 5 independent runs with their stddev in the parenthesis. In the 'Best Setting' column, the left reports DNN setting and the right reports model-specific setting. l denotes layer depth; n denotes CIN layer size; h and e, respectively, denotes #heads and att-embed-size; K denotes #experts and r denotes total rank.

Baseline	Criteo						MovieLens-1M			
	Logloss	AUC	Params	FLOPS	Best Setting		Logloss	AUC	Params	FLOPS
PNN	0.4421 (5.8E-4)	0.8099 (6.1E-4)	3.1M	6.1M	(3, 1024)	OPNN	0.3182 (1.4E-3)	0.8955 (3.3E-4)	54K	110K
DeepFm	0.4420 (1.4E-4)	0.8099 (1.5E-4)	1.4M	2.8M	(2, 768)	–	0.3202 (1.0E-3)	0.8932 (7.7E-4)	46K	93K
DLRM	0.4427 (3.1E-4)	0.8092 (3.1E-4)	1.1M	2.2M	(2, 768)	[512,256,64]	0.3245 (1.1E-3)	0.8890 (1.1E-3)	7.7K	16K
xDeepFm	0.4421 (1.6E-4)	0.8099 (1.8E-4)	3.7M	32M	(3, 1024)	l=2, n=100	0.3251 (4.3E-3)	0.8923 (8.6E-4)	160K	990K
AutoInt+	0.4420 (5.7E-5)	0.8101 (2.6E-5)	4.2M	8.7M	(4, 1024)	l=2, h=2, e=40	0.3204 (4.4E-4)	0.8928 (3.9E-4)	260K	500K
DCN	0.4420 (1.6E-4)	0.8099 (1.7E-4)	2.1M	4.2M	(2, 1024)	l=4	0.3197 (1.9E-4)	0.8935 (2.1E-4)	110K	220K
DNN	0.4421 (6.5E-5)	0.8098 (5.9E-5)	3.2M	6.3M	(3, 1024)	–	0.3201 (4.1E-4)	0.8929 (2.3E-4)	46K	92K
Ours										
DCN-V2	**0.4406 (6.2E-5)**	**0.8115 (7.1E-5)**	3.5M	7.0M	(2, 768)	l=2	0.3170 (3.6E-4)	0.8950 (2.7E-4)	110K	220K
DCN-Mix	0.4408 (1.0E-4)	0.8112 (9.8E-5)	2.4M	4.8M	(2, 512)	l=3, K=4, r=258	**0.3160 (4.9E-4)**	**0.8964 (2.9E-4)**	110K	210K
CrossNet	0.4413 (2.5E-4)	0.8107 (2.4E-4)	2.1M	4.2M	–	l=4, K=4, r=258	0.3185 (3.0E-4)	0.8937 (2.7E-4)	65K	130K

Table 7: Logloss and AUC (test) with a fixed memory budget.

#Params		7.9E+05	1.3E+06	2.1E+06	2.6E+06
LogLoss	CrossNet	**0.4424**	**0.4417**	**0.4416**	**0.4415**
	DNN	0.4427	0.4426	0.4423	0.4423
AUC	CrossNet	**0.8096**	**0.8104**	**0.8105**	**0.8106**
	DNN	0.8091	0.8094	0.8096	0.80961

8. Productionizing DCN-V2 at Google

Productionizing DCN-V2 at Google

Table 8: Relative AUCLoss of DCN-V2 v.s. same-sized ReLUs

1layer ReLU	2layer ReLU	1layer DCN-V2	2layer DCN-V2
0%	-0.15%	-0.19%	-0.45%

- DCN-V2는 AUCLoss에서 상당한 향상을 보임.
- 같은 사이즈의 ReLU layer를 DCN-V2로 대체함으로써 성능 개선이 이루어짐

9. Conclusions and Future Work

Conclusions and Future Work

- Explicit crosses를 하기 위해 새로운 모델 DCN-V2를 제안함.
- Model performance와 latency 사이에서 trade-off를 성취하기 위해 mixture of low-rank DCN를 제안함.
- 이를 통해 web-scale learning에서 성공적으로 배포했고 offline model accuracy와 online business metric gains에 상당함을 보였음.

Q&A