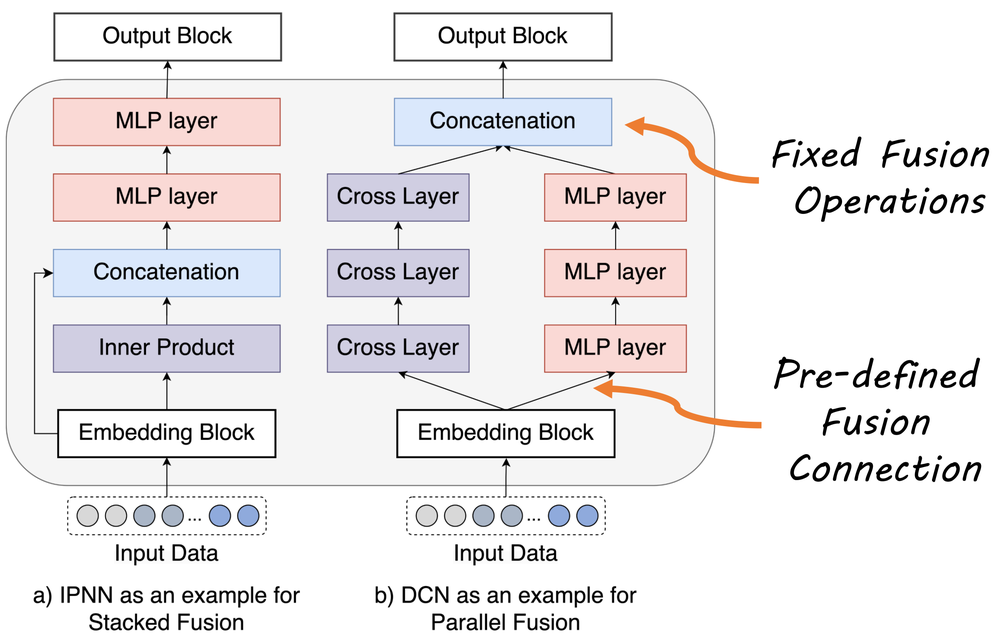
Fusion Matters: Learning Fusion in Deep Click-through Rate Prediction Models

Kexin Zhang*, Fuyuan Lyu*, Xing Tang, Dugang Liu, Chen Ma, Kaize Ding, Xiuqiang He and Xue Liu

Fusion Design in CTR



Stacked or Parallel?

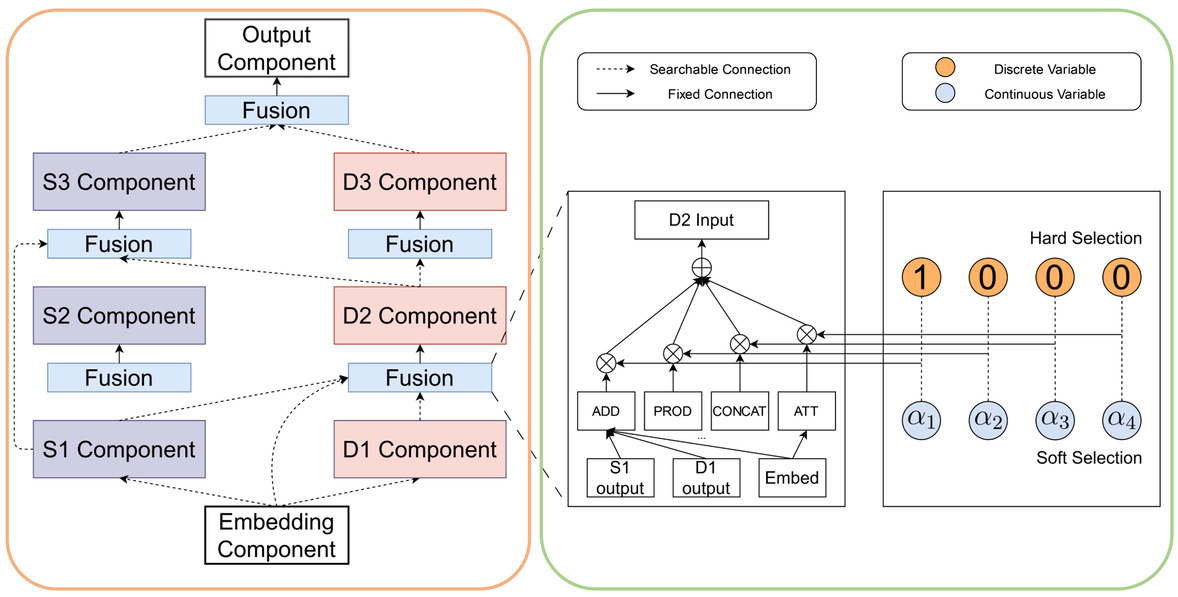
Experiment

Method	Criteo		Avazu		KDD12	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
FNN	0.8037	0.4473	0.7860	0.3766	0.7978	0.1534
PNN	0.8048	0.4463	0.7886	0.3752	0.8011	0.1527
DCNv2s	0.8088	0.4427	0.7877	0.3753	0.8030	0.1536
DeepFM	0.8038	0.4486	0.7856	0.3806	0.7963	0.1532
DCN	0.8063	0.4450	0.7875	0.3779	0.7968	0.1537
xDeepFM	0.8067	0.4453	0.7860	0.3773	0.7966	0.1542
DCNv2p	0.8085	0.4451	0.7894	0.3759	0.8012	0.1531
EDCN	0.8102	0.4419	0.7917	0.3727	0.8122	0.1498
AutoCTR	0.8082	0.4436	0.7883	0.3761	0.7949	0.1533
NASRec	0.8090	0.4435	0.7893	0.3752	0.7958	0.1530
OptFu.-H	0.8108*	0.4413*	0.7935*	0.3717*	0.8129	0.1496
OptFu.-S	**0.8113***	**0.4408***	**0.7938***	**0.3715***	**0.8158***	**0.1489***
Impr	0.0011	0.0023	0.0021	0.0012	0.0036	0.0009

Observations:

- Fusion Learning is important yet overlooked
  OptFusion yields best performance
- Fusion Learning is efficient
  OptFusion yields mid-level efficiency

OptFusion



Fusion Connection Learning:

- Binary Formulation: $c(G_i, G_j) = \begin{cases} 1, & connected \\ 0, & disconnected \end{cases}$
- Level-constraint: $c(G_i, G_j) = 1 \xrightarrow{\text{yields}} L(i) < L(j), \forall (G_i, G_j)$
- Solved as discrete optimization:

$$c(G_i, G_j) \to \alpha_{ij} = \begin{cases} > 0, & c(G_i, G_j) = 1 \\ \leq 0, & c(G_i, G_j) = 0 \end{cases}$$

Fusion Operation Selection:

- Aggregation: $\hat{\boldsymbol{e}}_j = o_j(\{c(G_i, G_j) \cdot \boldsymbol{e}_i\}),\ o_j \in \mathcal{O}$
- Solved as NAS: $\hat{\boldsymbol{e}}_j = \sum_{o \in \mathcal{O}} p_j^o \cdot o\left(\{\mathbb{I}_{\alpha_{ij}>0} \cdot \boldsymbol{e}_i\}\right)$

Ablation Study

Operation	Criteo		Avazu		KDD12	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
Add	0.8111	0.4422	0.7872	0.3970	0.7924	0.1585
Product	0.8077	0.4443	0.7860	0.3784	0.7938	0.1584
Concatenate	0.8075	0.4445	0.7837	0.3814	0.7926	0.1546
Attention	0.8073	0.4442	0.7843	0.3794	0.7883	0.1597
Hard	0.8108	0.4413	0.7935	0.3717	0.8129	0.1496
Soft	**0.8113**	**0.4408**	**0.7938**	**0.3715**	**0.8158**	**0.1489**

On Fusion Operation

Number	Criteo			Avazu		
	AUC	Logloss	Time (h)	AUC	Logloss	Time (h)
n=2	0.8112	0.4408	3.38h	0.7937	0.3716	1.47h
n=3	0.8113	0.4408	4.70h	0.7938	0.3715	1.98h
n=4	0.8115	0.4406	5.76h	0.7939	0.3717	2.67h

On #Components

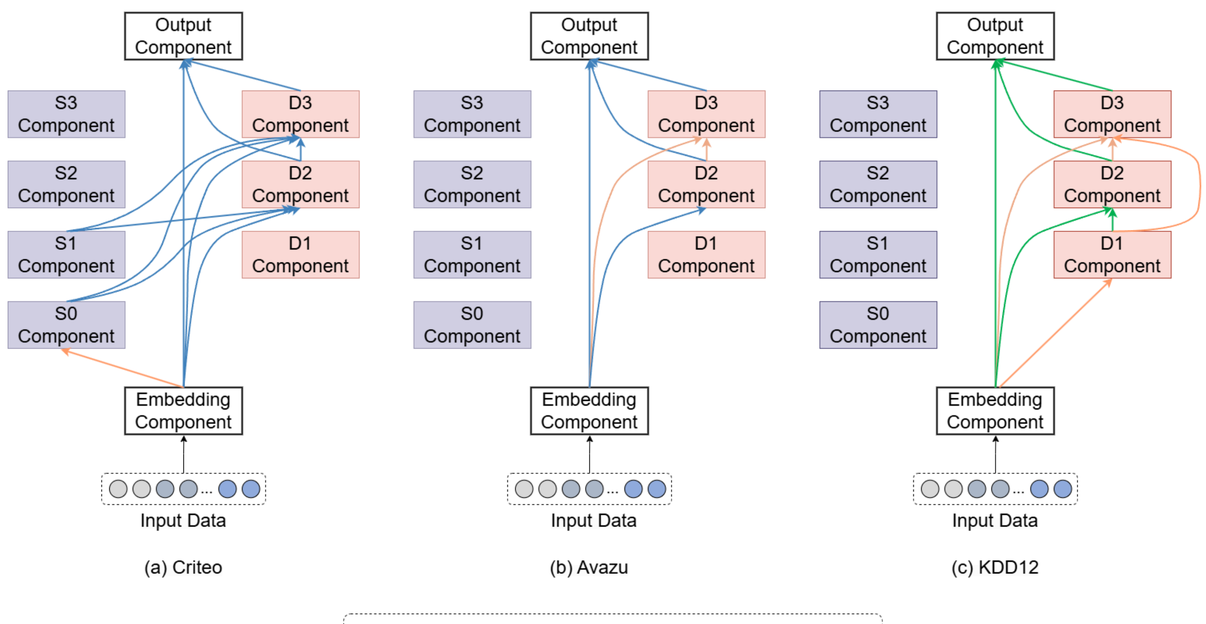
Methods	Criteo		Avazu	
	AUC	Logloss	AUC	Logloss
One-shot	**0.8113**	**0.4408**	**0.7938**	**0.3715**
Sequential	0.8109	0.4411	0.7934	0.3717

On Selection Alg.

Observations:

- Selecting fusion operation is necessary
- Slightly increase with #Components
- One-shot selection is necessary

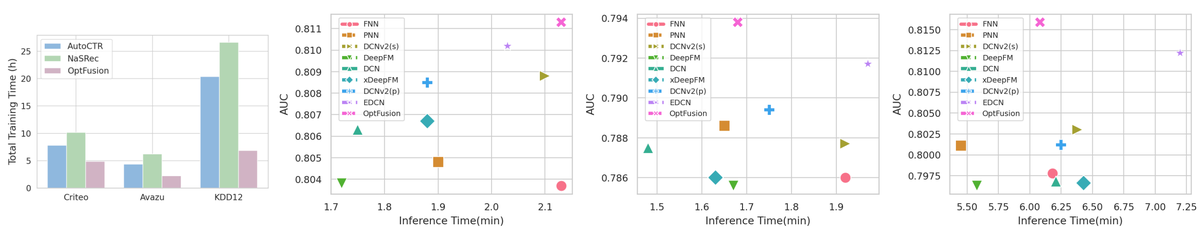
Case Study



Interesting Observation:

- Residual Connection among deep components

Experiment



Summary & Future Work

- Fusion Learning instead of component
- Fusion Connection Learning & Fusion Operation Selection
- Residual Connection among deep components

Paper

Code