[arxiv]

DADNN: Multi-Scene CTR Prediction via Domain-Aware Deep Neural Network

Junyou He[1,a],Guibao Mei[1,a],Feng Xing,Xiaorui Yang,Yongjun Bao,Weipeng Yan
Business Growth BU, JD
{hejunyou1,meiguibao,xingfeng7,lucky.yang,baoyongjun,Paul.yan}@jd.com

Contribution

1) To the best of our knowledge, this is a first study using a transfer learning (TL) model for multi-scene CTR prediction. The proposed method can support multiple scenes with a single model, thus greatly saving human labor and computation resources(offline training and online service). It is also scalable to continuously support new scenes.
2) In order to minimize the domain shift in different scenes, we propose a routing and domain layer where each scene has an individual domain-specific head similar to the multi-task learning (MTL) framework. Different from the traditional multi-task learning application, we solve the same task in different scenes which share the underlying representation.
3) Considering that the training of domain-specific heads is insufficient in scenes with limited samples and the knowledge of different scenes may be complementary under the same task, we adapt knowledge transfer (KT) among multiple scenes to enhance the opportunity of knowledge sharing by an internal KT process limited in a single model. We shift KT primary focus away from utilizing an extra teacher net to get compact models.
4) There is a certain difference among scenes. In the shared-bottom block, if the differences of scenes can be captured explicitly while considering the commonality, the CTR prediction performance can be further boosted. To this end, we introduce MMoE module. Each expert captures the commonness of samples from different standpoints, and the weights of gate allow for discriminative representations to be tailored for each individual scene. It is worth noting that the shared bottom network could be any popular models, not limited to MMoE.
5) DADNN has been successfully deployed in the online advertising system of JD, one of the largest B2C e-commerce platform in China. We have also conducted online A/B test to evaluate its performance in real-world CTR prediction tasks and obtained significant improvement in terms of CTR and CPM respectively.

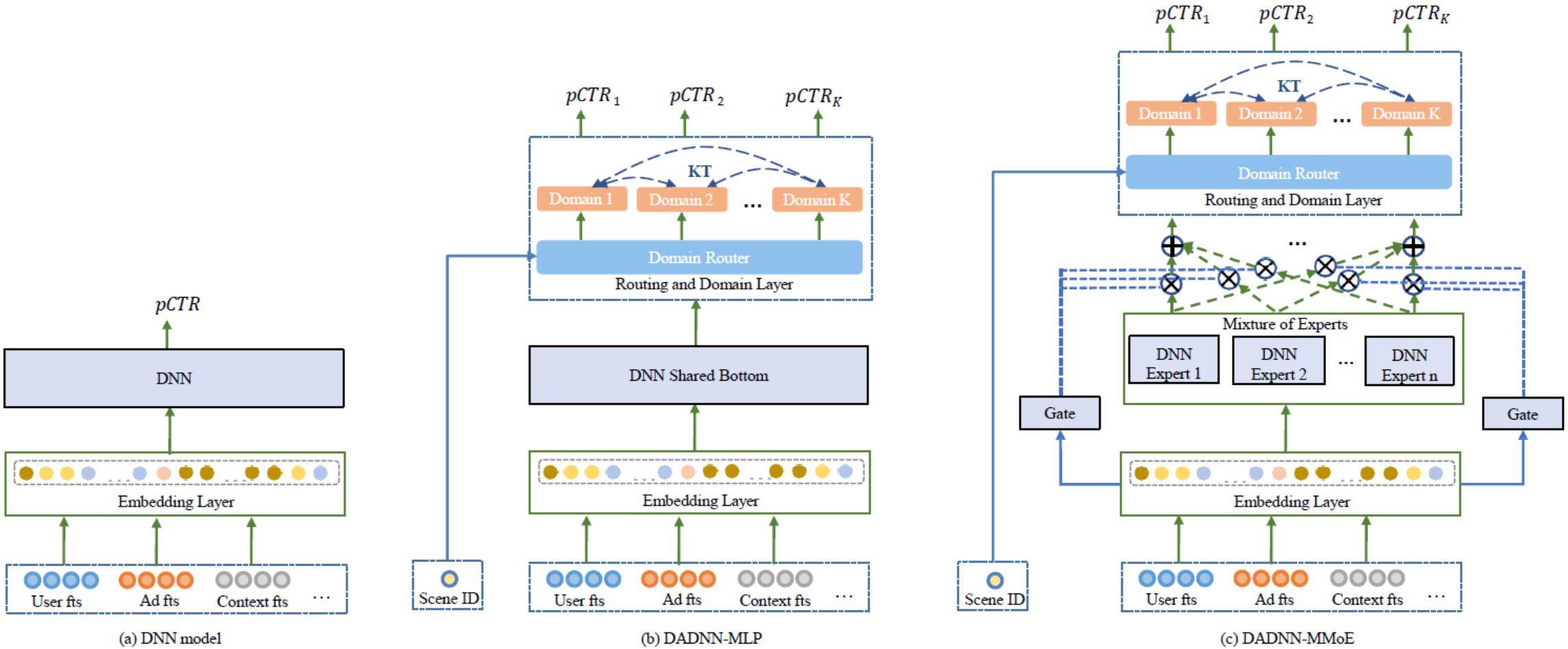

Fig. 2. Illustration of model architectures (fts-features). (a) DNN model, which considers only a single scene. (b) DADNN-MLP model, which further considers discriminative characteristics to be tailored for each individual scene. The routing layer uses a wide input of the scene id to distinguish the scene. KT represents internal knowledge transfer among multiple scenes. (c) DADNN-MMoE model, which introduces the multi-gate mixture-of-experts to substitute the hard shared-bottom block. The weights of gate allow for discriminative representations for each individual scene.

A. Sparse Input and Embedding Layer

to obtain the overall representation vector for the instance:$\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_i, \cdots, \mathbf{e}_f]$, where $\mathbf{e}_i$ is the embedding of i-th field and D is the embedding size. In practice, the embedding

B. Shared-Bottom Block

for each scene. The output y_k for k-th scene follows the corresponding domain-specific head h^k:

$$y_k = h^k(f(\mathbf{x})). \tag{1}$$

MMoE

hard shared-bottom model. We use a separate gating network g_k and n experts for each scene k. The output y_k of scene k is obtained as follows:

$$y_k = h^k(f^k(\mathbf{x})), \tag{2}$$

where

$$f^k(\mathbf{x}) = \sum_{j=1}^{n} g^k(\mathbf{x})_j f_j(\mathbf{x}). \tag{3}$$

The gating networks are simply linear transformations of the input with a softmax layer:

$$g^k(\mathbf{x}) = softmax(\mathbf{W}_{gk}\mathbf{x}), \tag{4}$$

where $\mathbf{W}_{gk}$ is trainable. Following the shared-bottom block, the routing layer splits all scene data into different domain layers.

MMoE

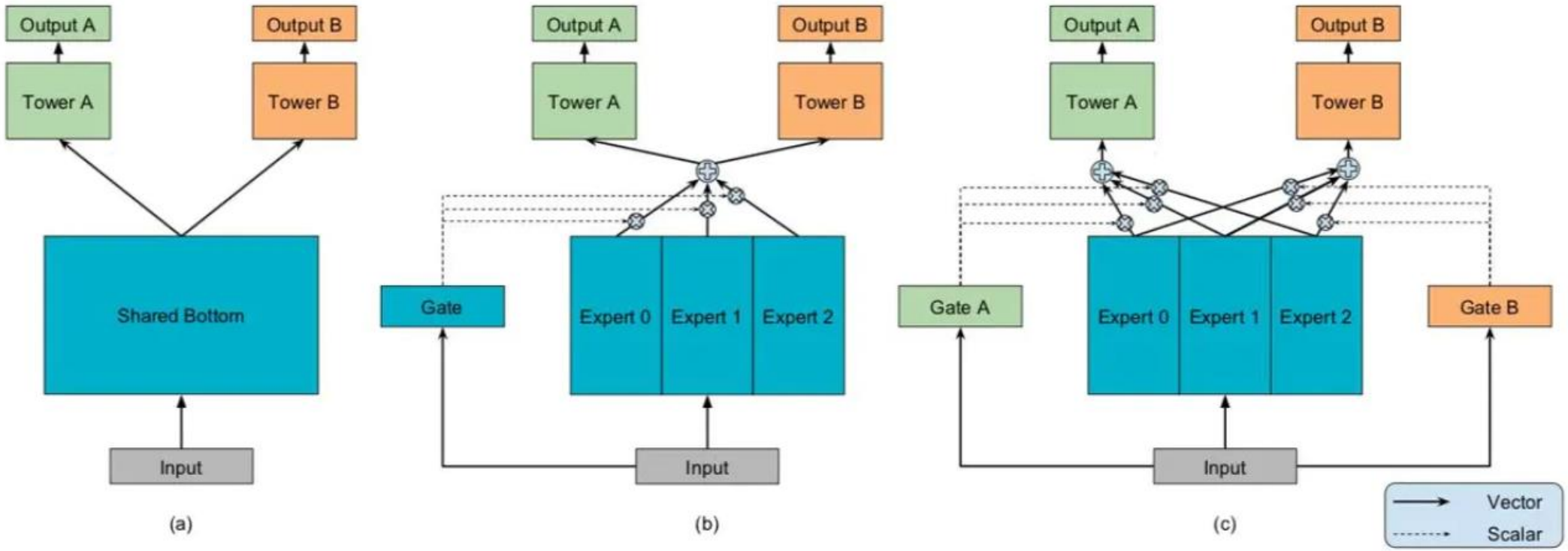

Figure 1: (a) Shared-Bottom model. (b) One-gate MoE model. (c) Multi-gate MoE model.

C. Routing and Domain Layer

introducing multiple data distributions. Given a dataset $D = \{(\mathbf{x}_i, y_i)|i = 1, 2, ..., N\}$, the objective function of our model is defined as follows:

$$\arg\min_{\mathbf{W}_d} L_d(\mathbf{W}_d; D), \tag{5}$$

where L_d is the total loss over training set. It can be formulated as:

$$L_d(\mathbf{W}_d; D) = \sum_{k=1}^{K} \alpha_k L_{d_k}, \tag{6}$$

where L_{d_k} is the loss for the k-th scene with α_k as the corresponding weight and K is the number of scenes. Through

Specifically, L_{d_k} is typically defined as the cross-entropy loss function:

$$L_{d_k} = -\frac{1}{N_k} \sum_{i=1}^{N_k} (y_i log p(\mathbf{x}_i) + (1 - y_i) log(1 - p(\mathbf{x}_i))), \tag{7}$$

where N_k is the size of k-th scene samples, y_i is the ground truth of i-th instance, and $p(\mathbf{x}_i)$ is the output of the k-th domain layer which represents the probability of sample $\mathbf{x}_i$ being clicked.

D. Knowledge Transfer

our proposed knowledge transfer method. Given a dataset $D = \{(\mathbf{x}_i, y_i)|i = 1, 2, ..., N\}$, the objective function of our model is defined as follows:

$$\arg\min_{\mathbf{W}_d, \mathbf{W}_{kt}} L_d(\mathbf{W}_d; D) + L_{kt}(\mathbf{W}_{kt}; D). \tag{8}$$

Specifically, L_{kt} is knowledge matching loss which represents the pairwise probabilistic prediction mimicking loss extended from [14], [15], which is defined as:

$$L_{kt} = \sum_{p=1}^{K} \sum_{\substack{q=1 \\ p \neq q}}^{K} u_{pq} L_{pq}, \tag{9}$$

$$L_{pq} = -\frac{1}{N_p} \sum_{i=1}^{N_p} (p(\mathbf{x}_i) log q(\mathbf{x}_i) + (1 - p(\mathbf{x}_i)) log(1 - q(\mathbf{x}_i))), \tag{10}$$

where $p(\mathbf{x})$ and $q(\mathbf{x})$ represent teacher network and student network respectively. u_{pq} is the corresponding weight of classifier p to q and N_p is the size of teacher samples. In our experiment, we set the u_{pq} to be 0.03. In particular, we

TABLE I
STATISTICS OF EXPERIMENTAL DATASET

Scene	User(M)	Ad(M)	Samples(M)	CTR*
1	1.3	1.6	29.9	3.63x
2	23.5	2.7	122.0	1.53x
3	3.5	1.4	40.5	1.93x
4	4.5	0.4	16.2	9.66x
5	18.5	2.4	77.5	x
6	24.1	6.2	184.1	2.34x
all	45.7	7.6		

* Because of commercial co
relative CTR values

is essential to the success of online advertising. It is the ratio of the average estimated CTR and empirical CTR:

$$calibration = pCTR/CTR. \tag{12}$$

The less the calibration differs from 1, the better the model is.

TABLE II
TEST AUC, CALIBRATION AND GAUC ON INDUSTRIAL DATASET

Method	Scene / Metric	1	2	3	4	5	6	GAUC
DNN(single)	AUC	0.69734	0.73114	0.74576	0.66172	0.78065	0.66585	0.71403
	calibration	1.08154	0.95360	0.97168	0.99172	**0.98647**	0.92132	
DNN	AUC	0.66112	0.73587	0.73289	0.67318	0.77273	0.66772	0.71225
	calibration	0.53774	0.98387	0.89748	0.97924	1.34186	0.94484	
Wide&Deep	AUC	0.65911	0.73539	0.73524	**0.67611**	0.77529	0.66775	0.71266
	calibration	0.56710	1.02160	0.94578	**1.00247**	1.39293	0.90866	
DCN	AUC	0.66202	0.73557	0.73710	0.67169	0.77528	0.66723	0.71279
	calibration	0.55615	0.98300	0.90693	1.05940	1.36822	0.94575	
DeepFM	AUC	0.65700	0.73562	0.73587	0.67219	0.77404	0.66916	0.71294
	calibration	0.58993	1.01282	0.95011	1.05969	1.37286	0.96842	
DADNN-MLP	AUC	0.70451	0.73657	0.74836	0.67146	0.78617	0.66766	0.71808
	calibration	**1.02730**	1.01409	**1.00267**	1.02859	0.89886	0.89494	
DADNN-MMoE	AUC	**0.70462**	**0.74020**	**0.75532**	0.67292	**0.78711**	**0.67472**	**0.72264**
	calibration	1.04104	**0.98831**	0.97281	0.98272	0.97660	**0.97541**	

TABLE III

THE PERFORMANCE OF DIFFERENT COMPONENTS IN DADNN.

Method	Scene / Metric	1	2	3	4	5	6	GAUC
BASE(DADNN-MLP)	AUC	0.70451	0.73657	0.74836	0.67146	0.78617	0.66766	0.71808
BASE With FT NO-KT	AUC	0.70978	0.73883	0.75554	0.67775	0.78592	0.67083	0.72099
MMoE With FT NO-KT	AUC	0.71126	0.74187	0.75490	0.67973	0.78933	0.67797	0.72508
BASE With FT	AUC	0.70471	0.74021	0.75346	0.67365	0.78887	0.67275	0.72204
MMoE With FT	AUC	0.71374	0.74558	0.76243	0.67926	0.79255	0.68071	0.72861