Array 15 (2022) 100205

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array

Learning to rank method combining multi-head self-attention with conditional generative adversarial nets

Jinzhong Li [a], Huan Zeng [a,*], Lei Peng [a], Jingwen Zhu [b], Zhihong Liu [c]

[a] Department of Computer Science and Technology, College of Electronic and Information Engineering, Jinggangshan University, Ji'an, China
[b] Department of Mathematics, College of Mathematics and Physics, Jinggangshan University, Ji'an, China
[c] Network Information Center, Jinggangshan University, Ji'an, China

ARTICLE INFO

MSC:
41A05
41A10
65D05
65D17

Keywords:
Learning to rank
Information retrieval generative adversarial networks
Conditional generative adversarial nets
Multi-head self-attention mechanism
Ranking model

ABSTRACT

The existing methods of learning to rank often ignore the relationship between ranking features. If the relationship between them can be fully utilized, the performance of learning to rank methods can be improved. Aiming at this problem, an approach of learning to rank that combines a multi-head self-attention mechanism with Conditional Generative Adversarial Nets (CGAN) is proposed in this paper, named *GAN-LTR. The proposed approach improves some design ideas of Information Retrieval Generative Adversarial Networks (IRGAN) framework applied to web search, and a new network model is constructed by integrating convolution layer, multi-head self-attention layer, residual layer, fully connected layer, batch normalization, and dropout technologies into the generator and discriminator of Conditional Generative Adversarial Nets (CGAN). The convolutional neural network is used to extract the ranking feature representation of the hidden layer and capture the internal correlation and interactive information between features. The multi-head self-attention mechanism is used to fuse feature information in multiple vector subspaces and capture the attention weight of features, so as to assign appropriate weights to different features. The experimental results on the MQ2008-semi learning to rank dataset show that compared with IRGAN, our proposed learning to rank method *GAN-LTR has certain performance advantages in various performance indicators on the whole.

1. Introduction

Search and recommendation are the most dominant ways to access information in the Internet era, and learning to rank is one of the key techniques. Learning to rank [1], which uses machine learning methods to train ranking models to solve ranking problems, is a research hot spot of information retrieval and machine learning, which plays an important position in practical applications such as search engines and recommendation systems. More and more learning to rank methods are widely used in these scenarios.

In recent years, with the explosive development of deep learning techniques, it has become a mainstream approach in academia and industry to design learning to rank methods to solve the ranking problem in information retrieval using deep learning algorithms or models such as generative adversarial networks [2], recurrent neural networks [3], convolution neural networks [4], deep neural networks [5], and deep Q-networks [6]. Literature [7] deeply investigated the problems associated with applying shallow or deep neural networks to train ranking models in information retrieval. Google's TF-Ranking [8] is an open-source library for training large-scale learning to rank models using deep learning algorithms in the TensorFlow framework, which contains a variety of deep neural network-based learning to rank methods.

Current learning to rank approaches only consider the correlation between individual features and the ranking results, and it do not consider the correlation between the local combinations among the features and the ranking results. To address this problem, this paper proposes a method of learning to rank combining a multi-head self-attention mechanism and conditional generative adversarial nets to further improve the ranking performance, which considers the relationship between ranking features, mines the potential features and assigns corresponding weights for training the models of learning to rank.

The main contributions of this paper are as follows.

(1) Combining multi-head self-attention with Conditional Generative Adversarial Nets (CGAN), we propose a method of learning to rank, named *GAN-LTR, which is a novel and improved method based on Information Retrieval Generative Adversarial Networks (IRGAN). To the best of our knowledge, *GAN-LTR is the state-of-the-art one.

(2) Our *GAN-LTR approach constructs a new network model of the generator and discriminator, which integrates the convolutional layer,

* Corresponding author.
E-mail address: zenghuan0123@163.com (H. Zeng).

https://doi.org/10.1016/j.array.2022.100205
Received 2 March 2022; Received in revised form 31 May 2022; Accepted 3 June 2022
Available online 16 June 2022

multi-head self-attention layer, residual layer, fully connected layer, batch normalization, dropout technology and other improvements into the generator and discriminator of the CGAN framework.

(3) A set of experiments are performed and analyzed. Compared with the frontier method of current learning to rank method IRGAN, the experimental results indicate that our *GAN-LTR obtains some better results in multiple performance indicators, especially, our method has a faster convergence speed.

2. Related work

This paper investigates learning to rank methods that combine multi-head self-attention mechanisms and conditional generative adversarial nets. The main works closely related to this paper contains: learning to rank methods based on generative adversarial networks, learning to rank methods based on attention mechanisms, and generative adversarial networks based on attention mechanisms.

2.1. Learning to rank methods based on generative adversarial networks

The first pioneering work on solving information retrieval problems using Generative Adversarial Networks (GAN) is the Information Retrieval Generative Adversarial Network (IRGAN) proposed by Wang et al. [2], which implements a unified description of two schools of thinking of generative retrieval models and discriminative retrieval models in information retrieval modeling. IRGAN uses the idea of confrontation between generator and discriminator in GAN, and adopts a minimax algorithm in game theory to integrate the generative retrieval model and discriminative retrieval model into a unified framework in a way of confrontation training, so that the two models can improve each other, and finally make the retrieved documents more accurate. The full score paper won the nomination award for the best paper of SIGIR2017, which is both very innovative and very practical. Just like GAN in other fields, it has brought a change in the research paradigm of information retrieval [9].

At present, a small amount of work has improved or extended IRGAN. Deshpande et al. [10] proposed two models influenced by self-contrastive estimation and co-training to improve the performance of IRGAN. Jain et al. [11] improved the convergence of IRGAN through a novel optimization objective based on Proximal Policy Optimization (PPO) and a sampling technique based on Gumbel-Softmax. Lu et al. [12] proposed a personalized adversarial training framework PSGAN for personalized search with limited and noisy click data to alleviate the problems of scarcity of high-quality user data and noisy data in personalized search. The PSGAN framework is extended on the basis of IRGAN, the generated model is enhanced to train the attention of those hard-to-distinguish training data through adversarial training, the discriminator is used to evaluate the personalized relevance of documents, and the generator is used to learn the distribution of relevant documents. Two models are proposed in this framework: a model based on the document selection and a model based on the query generation to effectively improve the quality of personalized search. Park et al. [13] proposed an adversarial sampling and training framework different from IRGAN to learn Ad-hoc retrieval models with implicit feedback, and applied adversarial training to the pairwise learning to rank framework.

Although there have been some improved methods for IRGAN, its performance still has some room for improvement due to the instability of IRGAN training and its simple network model.

2.2. Learning to rank methods based on attention mechanisms

Applying attention mechanisms to the learning to rank task, which has been proven that it can successfully focus on different aspects of the input [14]. Wang et al. [14] applied the attention mechanism to the list problem of learning to rank and proposed a new attention-based deep neural network for the learning to rank problem, which applied the attention mechanism to merge different embeddings of query and search results, and used the list approach to sort the search results. Jiang et al. [15] proposed a learning framework for explicit result diversification using the attention mechanism, recurrent neural networks and max pooling technology, which used the attention mechanism to capture the subtopics to be concerned while selecting the next document. Zhang et al. [16] proposed an attention-based learning-to-rank model for structured map search, which was a novel deep neural network architecture of learning-to-rank. Qin et al. [17] proposed a method of learning to rank based on self-attention network for search result diversification task. This method used self-attention to model the interactions between all candidate documents and subtopics, which can comprehensively measure the relationships between the whole candidate documents and the coverage degrees of candidate documents to different subtopics. Sun et al. [18] explored modeling the interactions between documents for learning to rank by employing regularized self-attention. Pobrotyn et al. [19] studied context-aware learning to rank with self-attention mechanism, and proposed a learnable, context-aware, and scoring function based on self-attention, which allows for modeling of inter-item dependencies not only at the loss level but also in the computation of items' scores.

The above research shows that integrating attention mechanism into learning to rank method helps to improve the performance of learning to rank method.

2.3. Generative adversarial networks based on attention mechanisms

At present, due to some excellent characteristics of attention mechanism, the study of applying attention mechanism to generative adversarial networks has become a hot spot pursued by some researchers, and some good research results have been gradually formed. Zhang et al. [20] proposed Self-Attention Generative Adversarial Networks (SAGANs), which integrated the self-attention mechanism into the convolutional GANs framework. The self-attention module is complementary to the convolutional structure so that each pixel was associated with other pixels, and helps to model the long-range, multi-level dependencies across different image regions. Xu et al. [21] added the attention mechanism to GANs and proposed the attentional generative adversarial networks model, named AttnGAN, for synthesizing images from text descriptions. Emami et al. [22] proposed a novel spatial attention GAN model, i.e. SPA-GAN, which introduced the attention mechanism into the generative adversarial network architecture to help the generator focus more on the most discriminative regions between the source and target domains, resulting in more realistic output images. Jiang et al. [23] proposed a super-resolution magnetic resonance image reconstruction method by using self-attention based generative adversarial networks, which integrated the self-attention mechanism into the super-resolution GAN framework to calculate the weight parameters of the input features.

The above research shows that incorporating the attention mechanism into GAN helps GAN to focus on some important and critical information, thereby improving the performance of GAN.

Therefore, based on these research foundations described above, and inspired by their ideas, this paper intends to integrate the attention mechanism and the generative adversarial networks to design the learning to rank method, that is, to propose a method of learning to rank, named *GAN-LTR, which combines multi-head self-attention with Conditional Generative Adversarial Nets (CGAN) [24]. *GAN-LTR method will improve some design ideas of IRGAN framework applied to web search, that is to say, this method integrates the convolutional layer, multi-head self-attention layer, residual layer, fully connected layer, batch normalization and dropout technology into the generator and discriminator of the CGAN framework. Moreover, it uses the softsign activation function to replace the tanh activation function in IRGAN to construct a new network model, so as to further improve the performance of the learning to rank method.

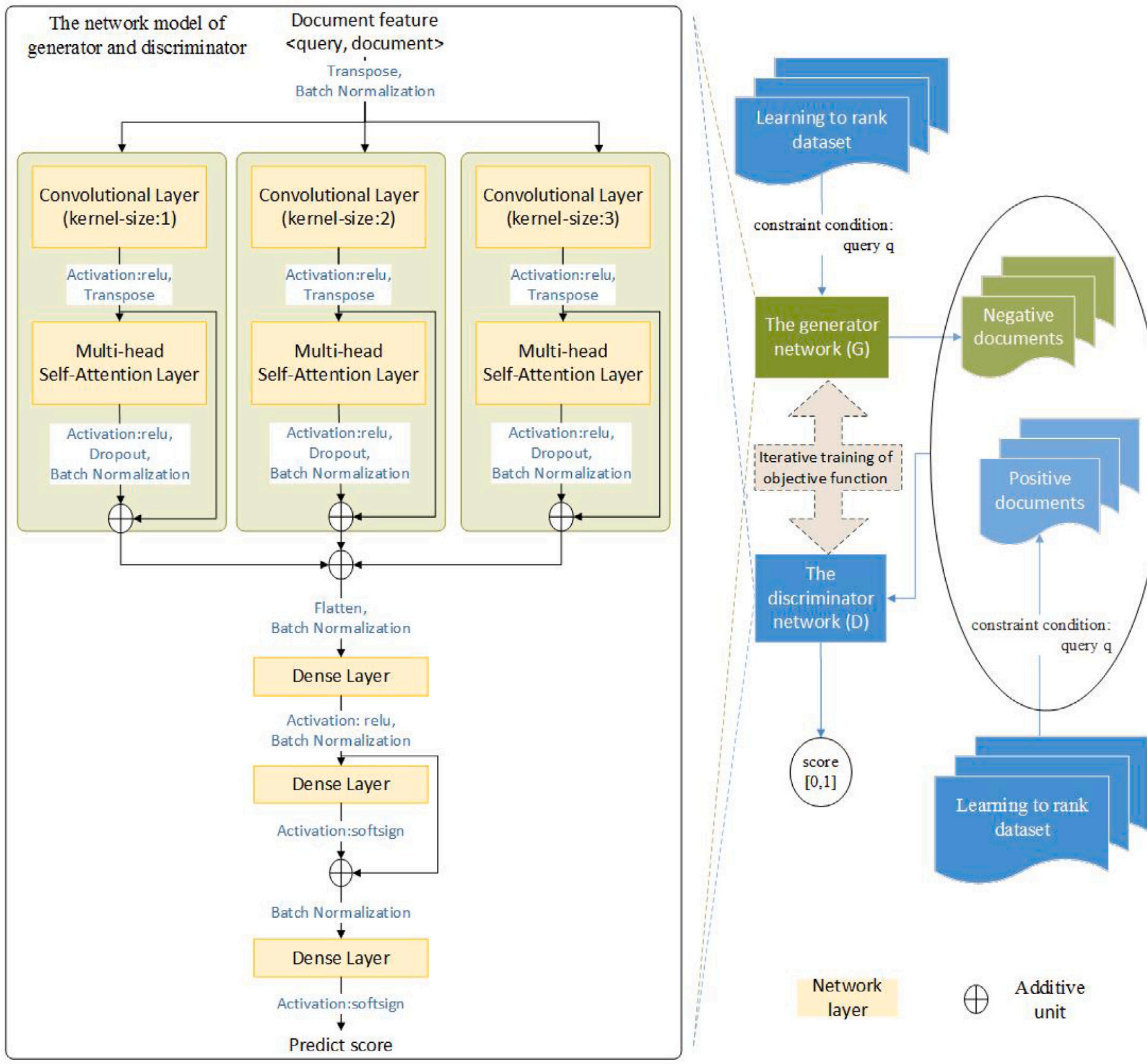

Fig. 1. The overall framework of *GAN-LTR.

3. A learning to rank approach combining multi-head self-attention mechanisms with conditional generative adversarial nets

The overall framework of learning to tank method *GAN-LTR, which integrates multi-head self-attention mechanisms and conditional generative adversarial nets (CGAN), is shown in Fig. 1. In this figure, the right side is the framework of CGAN, which controls the generated documents by setting query q as the constraint of CGAN; the left side is a new network model reconstructed for the generator and discriminator in CGAN, and the generator and discriminator use the same network model. In the new network model, Transpose is the transpose operation of the input matrix, and Batch Normalization (BN) is the batch normalization operation, which is used several times in the model to alleviate the problem of gradient vanishing, so as to improve the stability of the network. Flatten is a flattening operation for the input matrix to transform the matrix into a one-dimensional vector. Activation represents the activation function, and relu and softsign are used as the activation function, where relu is used as the activation function for the output of the previous layers of the network model, and softsign is used as an alternative to tanh activation function in the last two layers to better alleviate the problem of gradient vanishing. Dropout represents that hidden neurons are deactivated randomly, and L2 regularization is used to prevent over-fitting of the model. The network model includes convolutional layer, multi-head self-attention layer, residual layer and fully connected layer, etc, where the addition operation ⊕ represents the residual layer, and Dense Layer represents the fully connected layer.

The generator network and the discriminator network use the same network model, and the main process of its network model is as follows: Firstly, several convolution kernels of size $1 \times k$, $2 \times k$, $3 \times k$ are respectively used to convolve the input features in order to achieve the local feature extraction of the input features; Secondly, the multi-head self-attention layer and the residual layer are used to achieve the global feature extraction for the different convolution results; Thirdly, the output of the multi-headed self-attention layer is spread into a one-dimensional vector, and the results after batch normalization and residual layer accumulation are input into the full connection layer to obtain new features with the same dimensionality as the ranking features; Then, the output of the fully connected layer is performed a fully connected layer and residual layer operations; Finally, the output documents belong to the prediction scores of positive and negative examples respectively after a fully connected layer.

The main layers of the network model of the generator and discriminator in *GAN-LTR are described in detail below: convolutional layer, multi-head self-attention layer, residual layer, and fully connected layer.

3.1. Convolutional layer

The convolutional network has the characteristics of local correlation and weight sharing. The local correlation feature can avoid the excessive amount of parameters of the fully connected network, and can extract the correlation between local features, a single convolutional kernel for weight sharing on feature extraction also avoids the disadvantage of the excessive number of parameters.

Kim [25] explores the convolutional neural network for sentence classification. Inspired by the idea of its convolutional layer, we apply it to the network model of *GAN-LTR. Suppose that $f_i \in R^k$ is the ith feature vector of document d, where k is the feature vector dimension of each feature in the document, then the document vector with length n can be represented as $f_{i:n} = concat([f_1, f_2, \dots, f_n])$, where the function $concat()$ is the splicing function. More generally, let $f_{i:i+j}$ be as the splicing feature vector of the document $f_{i,i+1,\dots,i+j}$. The convolution operation is to apply a convolutional kernel $w \in R^{xk}$ with the window size of x to the document features $f_{i:i+x-1}$ to produce a new feature c_i, i.e. $c_i = S(wf_{i:i+x-1} + b)$, where the bias value $b \in R$, and S is a linear function. The kernel function acts on all possible windows in the document features $\{f_{1:x}, f_{2:x+1}, \dots, f_{n-x+1:n}\}$ to generate a new feature mapping $c = [c_1, c_2, \dots, c_{n-x+1}]$ of dimension R^{n-x+1}, where padding is valid. When padding is the same, the blank area will be added so that the final features are mapped $c = [c_1, c_2, \dots, c_n]$.

For the convolutional layer in *GAN-LTR, we first transpose the document features into column vectors and perform a batch normalization operation, and then convolve the processed document features using m convolutional kernels of size $1 \times k$, $2 \times k$, $3 \times k$ respectively, where the convolution step is 1 and the padding is the same, to obtain three feature matrices with $m \times n$, where m denotes the number of convolutional kernels, n denotes the dimension of the document feature vector, and k is the dimension of the feature vector. After each feature matrix is activated and transposed, it is used as the input of the multi-head self-attention layer.

3.2. Multi-head self-attention layer

The multi-head self-attention mechanism originates from the Transformer model proposed by Google [26]. It is a special internal attention mechanism for modeling the dependencies between elements, which can directly calculate the attention on a sequence itself and reassign the weights at each position to obtain a more reasonable feature representation.

The multi-head self-attention layer is composed of multiple self-attention layers which can be computed in parallel. The queries, keys, and values matrices are denoted by Q, K, and V, respectively, and Q, K, and V are all the same. Suppose that there are h heads, and each head transforms Q, K, and V into a subspace through a linear transformation during computation. The parameters of the linear transformation of each head are different and learnable, so as to ensure that the model learns relevant features from different representation subspaces [26]. The proposed method *GAN-LTR in this paper incorporates the multi-head self-attention layer to model the internal dependencies among the ranking features and aggregate these features to obtain higher-level features.

For the multi-head self-attention layer in *GAN-LTR, its core idea is to calculate the relationships between each ranking feature and all other ranking features, and these feature-to-feature relationships reflect the correlation and importance degree between different features to a certain extent. Its input is composed of three matrices of query Q, key K and value V, and its output is the multi-head self-attention matrix after multi-head splicing and linear transformation.

According to the idea of multi-head attention in the literature [26], we divide the calculation process of multi-head self-attention into three steps as follows.

Step 1: Linear transformation. After the output of the convolutional layer is activated and transposed, matrices Q, K and V required to calculate the self-attention scores are obtained and the parameter matrices of them are set to W^Q, W^K and W^V respectively. Then, perform linear transformation on three matrices Q, K and V with the same dimensions but different initial parameters, that is, perform h different projections on Q, K, and V.

Step 2: Self-attention calculation. For each head, the attention scores of Q and K are calculated by the scaled dot product method, and the attention scores are normalized by the Softmax function, then the attention scores and its corresponding matrix V are weighted and summed to obtain the self-attention result of the ith head, as shown in Eqs. (1) and (2).

$$SelfAttention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

$$head_i = SelfAttention(QW_i^Q, KW_i^K, VW_i^V) \tag{2}$$

Step 3: Splicing and linear transformation. The self-attention results of the h heads are spliced and then linearly transformed to obtain the final multi-head self-attention output, as shown in Eq. (3).

$$MultiHeadSelfAttention(Q, K, V) = Concat(head_1, \dots, head_k)W^0 \tag{3}$$

In each of the above steps, the scaling factor d_k represents the dimensionality of the key K, and $W_i^Q \in R^{d_{model} \times d_k}$, $W_i^K \in R^{d_{model} \times d_k}$, $W_i^V \in R^{d_{model} \times d_v}$ and $W^0 \in R^{hd_v \times d_{model}}$ are the parameter matrixes respectively, d_{model} denotes the dimensionality of the element vector.

3.3. Residual layer

A residual network simply adds a skip connection between the input and output, and does not add a new type of network layer. In the residual network, the input x is transformed through the network layer F to obtain the output value $F(x)$, and then the output value $F(x)$ is summed with the input x to get the final output $H(x)$, i.e. $H(x) = x + F(x)$. By stacking the residual modules, the whole network training can be stabilized with the deepening of network depth.

Xie et al. [27] explores the aggregated residual transformation of deep neural networks. Inspired by its idea of aggregated residual transformation, we apply it to the network model of *GAN-LTR. In order to take into account the properties of the residual layer and consider the overall impact of each part of the residuals on the final result, we add influence factors a and b to the residual network such that $H(x) = ax + bF(x)$, where a and b are trainable hyper-parameters with gradient descent, and $0 \le a \le 1$ and $0 \le b \le 1$. when $a = 1$ and $b = 1$, $H(x)$ will degenerate into a general residual network, otherwise it will become a weighted residual network. More generally, if there are multiple networks with outputs of the same dimensional size, then $H(x) = \sum_{i=1}^{n} w_i H_i(x)$, where n denotes the number of networks, w_i denotes the weight of the ith network $H_i(x)$, and $0 \le w_i \le 1$.

For the residual layer in *GAN-LTR, it is used in the network model of the generator and discriminator of CGAN respectively in the following three cases, i.e., the additive operations $\oplus$: Firstly, the output of the convolution layer and the output of the multi-head self-attention layer are connected to improve the stability of downstream tasks; Secondly, the outputs of three multi-head self-attention layers are connected to linearly superimpose different outputs; Thirdly, the fully connected layers with the same dimensional outputs are concatenated to further encode useful upstream features and downstream features.

3.4. Fully connected layer

The fully connected layer, realizes the mutual mapping of different linear spaces through linear functions. The activation function is added to enhance the classification effect of the fully connected layer and avoid the gradient vanishing.

For the fully connected layer in *GAN-LTR, that is, the Dense layer in Fig. 1 is used in three spaces in the network model of the generator and discriminator of CGAN, which are: (1) The results of flattening and batch normalization for the output of the second case in the residual layer are input to a fully connected layer, and they are mapped to a linear space with the same dimensional size as the input feature's dimension for mapping from the high-dimensional space to the low-dimensional space. Here, the relu function is used as the activation function. (2) The last two layers of the network model are two fully connected layers, which correspond to the two fully connected layers in IRGAN. But here, the softsign function is used as the activation function, the activation function of tanh hyperbolic tangent in IRGAN is modified to a softsign activation function with a smoother curve to avoid gradient vanishing.

4. Experimental comparison and analysis

To verify the performance of our proposed learning to rank method *GAN-LTR that combines the multi-head self-attention mechanism and conditional generative adversarial nets, we re-code the IRGAN experimental code[1] and perform experiments on the same dataset of learning to rank MQ2008-semi applied to the web search experiments of IRGAN, then compare and analyze the experimental results with IRGAN applied to the dataset in web search. For other representative methods of learning to rank, i.e., RankNet, LambdaRank and LambdaMART, since IRGAN has been compared with them in the literature [2] in terms of performance, their experimental results show that IRGAN method has brought significant performance improvements in all metrics (including $NDCG@N$, $Precision@N$, MAP and MRR). Therefore, *GAN-LTR is only compared with IRGAN, and the comparison with these methods will not be repeated in this paper. We use the same evaluation metrics in IRGAN [2] to compare the performance of *GAN-LTR and IRGAN, namely $Precision@N$, Mean Average Precision (MAP), Normalized Discounted Cumulative Gain ($NDCG@N$) and Mean Reciprocal Ranking (MRR), which are the generally used standard ranking performance indicators for learning to rank in information retrieval field. Therefore, for the calculation methods and formulas of these evaluation indicators, please refer to the relevant literature [28], which are ignored here.

4.1. Experimental setup

Following some experimental settings of IRGAN applied to Web search, we preprocess the MQ2008-semi dataset of learning to rank, that is, all query-document pairs with relevance values greater than 0 are regarded as positive samples, and all query-document pairs with relevance values of 0 or −1 are regarded as unlabeled samples. Meanwhile, the MQ2008-semi dataset is randomly split into training set and test set according to 4:1, and the sampling method and training process of IRGAN are also adopted, etc.

The experimental results of IRGAN applied to Web search found that IRGAN-Pointwise training was in a stable state when epoch > 100, so the value of epoch was set to 100 to reduce the training time, while other parameters remain consistent with IRGAN' parameters in the original paper. For the training of IRGAN-Pairwise, we set the learning rate of the generator of GAN to 0.0003 and the number of training epochs of discriminator and generator to be 30 and 25 respectively by parameter tuning. On the basis of this parameter setting, the experimental results of *GAN-LTR and IRGAN are compared and analyzed.

[1] https://github.com/geek-ai/irgan

Table 1
Ranking performance comparison of *GAN-LTR and IRGAN on the MQ2008-semi dataset.

Learning to rank methods	Precision@3	Precision@5	Precision@10	MAP
IRGAN-pointwise	0.1873	0.1619	0.1190	0.1801
*GAN-LTR-pointwise	0.1968	0.1771	0.1314	0.1936
IRGAN-pairwise	0.1968	0.1657	0.1248	0.1923
*GAN-LTR-pairwise	0.2000	0.1752	0.1276	0.2036
Learning to rank methods	NDCG@3	NDCG@5	NDCG@10	MRR
IRGAN-pointwise	0.2087	0.2163	0.2377	0.3487
*GAN-LTR-pointwise	0.2192	0.2304	0.2484	0.3407
IRGAN-pairwise	0.2204	0.2208	0.2440	0.3497
*GAN-LTR-pairwise	0.2274	0.2348	0.2565	0.3573

4.2. Results and discussion

In order to facilitate the performance comparison with IRGAN, *GAN-LTR also follows the same settings on some performance metrics and learning curves of IRGAN applied to web search on the MQ2008-semi dataset of learning to rank. Table 1 shows the comparison of the ranking performance metrics of pointwise and pairwise learning to rank methods of *GAN-LTR and IRGAN on the MQ2008-semi dataset for $Precision@N$, $NDCG@N$, MAP, and MRR. As the results are shown in Table 1, it can be seen that in the vast majority of cases, *GAN-LTR is better than IRGAN in each corresponding evaluation metrics. The reasons are mainly due to the integration of convolutional neural networks and multi-head self-attention mechanisms, which can effectively extract the implicit ranking features of <query q, document d>, obtain the correlation between the ranking features, and assign different weights to different ranking features. Due to the existence of the multi-head self-attention mechanism, the weight assignments within multiple vector subspaces can be obtained, and the weights of ranking feature within $\langle q, d\rangle$ can be assigned more reasonably, and the contribution of important ranking features is enhanced, thereby improving the performance of the learning to rank method *GAN-LTR.

Fig. 2 shows the changes of the non-adversarial learning curve of the pre-trained models of *GAN-LTR and IRGAN before adversarial training on the MQ2008-semi dataset, and Figs. 3 and 4 show the changes of learning curve of pointwise and pairwise learning to rank methods of *GAN-LTR and IRGAN during adversarial training on the MQ2008-semi dataset, respectively. They are all the results obtained under the same hyper-parameters (including the number of training epochs, learning rate, batch size, etc.). Specifically, the results shown in Fig. 2 reflect the performance's changes of the non-adversarial learning curves of the pre-trained models of *GAN-LTR and IRGAN before adversarial training on the evaluation criteria $NDCG@5$ and $Precision@5$, which is iteratively evolved with the number of rounds. It can be seen from this figure that the pre-training model of *GAN-LTR performs better than the pre-training model of IRGAN for the evaluation metrics $Precision@5$ and $NDCG@5$. Furthermore, we can also see that the convergence of our proposed method *GAN-LTR is faster compared with IRGAN. So our proposed method *GAN-LTR is better than IRGAN in evaluation metrics and convergence. The main reason is that the multi-head self-attention layer is integrated into the pre-training model of *GAN-LTR, which can capture the attention weights of ranking features and assign higher weights to important ranking features, thereby improving the performance of the *GAN-LTR method as a whole. The results shown in Fig. 3 reflect the performance's changes of the adversarial learning curves of the respective generators in the pointwise learning to rank methods of *GAN-LTR and IRGAN on the evaluation criteria $Precision@5$ and $NDCG@5$, which are iteratively evolved with the number of epochs. The results shown in Fig. 4 reflect the evolution of the adversarial learning curves of the respective generators and discriminators in the pairwise learning to rank methods of IRGAN and *GAN-LTR on the evaluation criteria

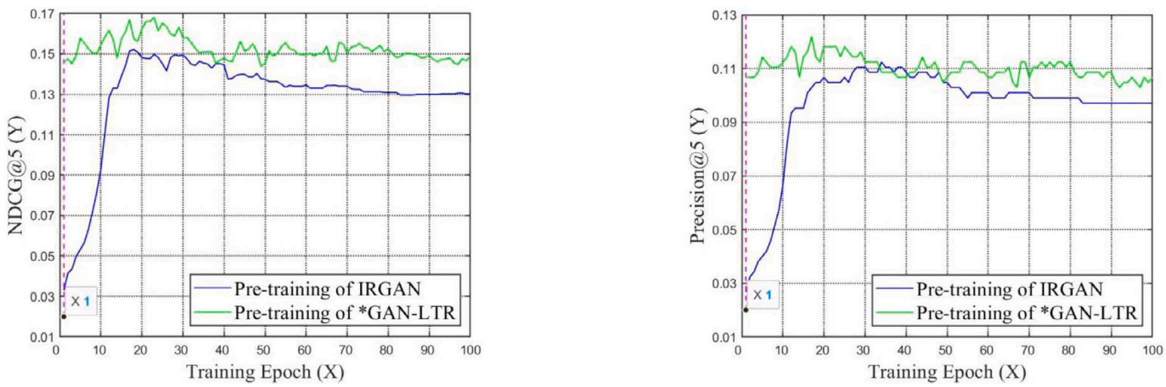

Fig. 2. Learning curves of the pre-trained models of *GAN-LTR and IRGAN before the adversarial training on the MQ2008-semi dataset.

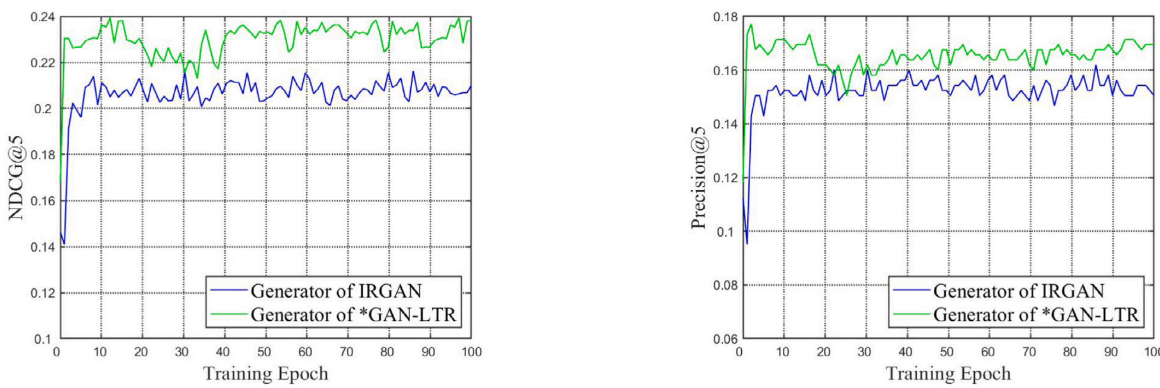

Fig. 3. Pointwise learning curves of *GAN-LTR and IRGAN on the MQ2008-semi dataset.

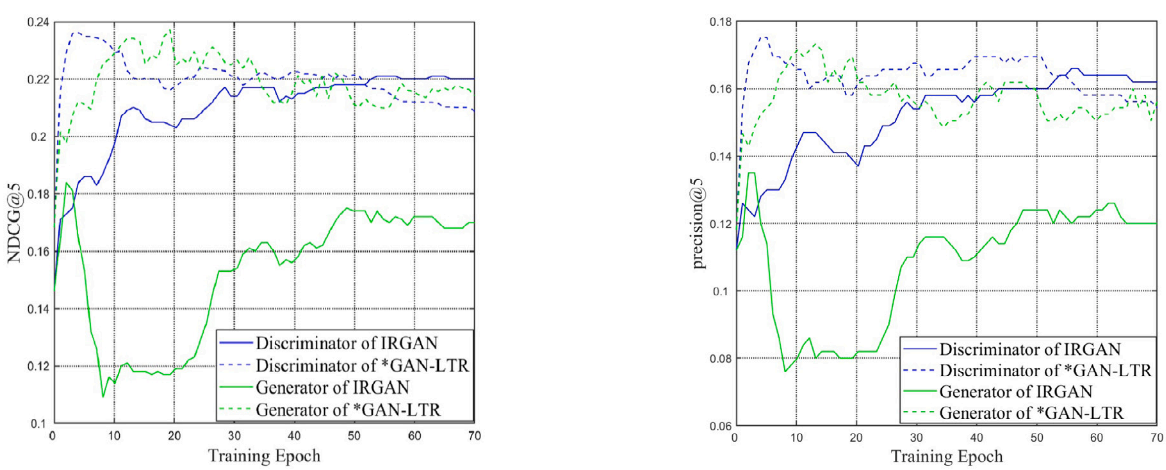

Fig. 4. Pairwise learning curves of *GAN-LTR and IRGAN on the MQ2008-semi dataset.

$Precision@5$ and $NDCG@5$ with the number of iterations. It can be seen from the results in these two figures that in the early stage of iterative training, the values of the evaluation criteria $NDCG@5$ and $Precision@5$ obtained by the respective pointwise and pairwise learning to rank methods of *GAN-LTR and IRGAN show a increasing trend with the iterative training. The main reason is that the generator network and the discriminator network in *GAN-LTR and IRGAN methods lead to the improvement of the generative ability and the discriminative ability in the continuous adversarial game, thus increasing the value of each evaluation metric. In the later stage of iterative training, the value of each evaluation metric tends to be stable and fluctuate up and down, indicating that the adversarial training of each method has reached a stable state. Furthermore, from the overall point of view, Figs. 3 and 4 also show that the curves of various performance indicators of *GAN-LTR are roughly above those of IRGAN, indicating that the corresponding performance indicators obtained by *GAN-LTR training have obtained better results, that is, *GAN-LTR has better performance than IRGAN. In the same way, the main reason for such results is still due to the incorporation of technologies such as convolutional neural networks and multi-head self-attention mechanisms in CGAN. Finally, compared with the results of the performance criteria $NDCG@5$ and $Precision@5$ of the pre-trained model shown in Fig. 2, the results of the performance criteria shown in Figs. 3 and 4 show superior performance, indicating that the iterative adversarial training of CGAN can improve the performance of pre-trained models (here, non-adversarially trained models) to a certain extent.

5. Conclusion

Aiming at the fact that the existing learning to rank methods often ignore the relationship between ranking features, this paper proposes a new method of learning to rank, named *GAN-LTR, that combines a multi-head self-attention mechanism with Conditional Generative Adversarial Nets (CGAN). This method adopts the sampling method and training process of IRGAN, and improves some design ideas of applying the IRGAN framework to web search. *GAN-LTR has made the following improvements on the basis of IRGAN: In the network model of CGAN, the convolutional layer and multi-head self-attention layer are added to extract the local features and global combined features of the documents and assign appropriate weights to different ranking features, the residual layer is added to avoid the gradient vanishing and degradation problems caused by the network being too deep, batch normalization is added to enhance the stability of the network, the dropout technique is added to randomly deactivate the hidden units to avoid over-fitting, and the activation function hyperbolic tangent tanh function is modified to a softsign activation function with a smoother curve to avoid gradient vanishing. The experimental results on the MQ2008-semi learning to rank dataset show that: compared with the experimental results of IRGAN applied to web search, the learning to rank method *GAN-LTR proposed in this paper obtains better results in various performance indicators. As a whole, *GAN-LTR has certain performance advantages.

CRediT authorship contribution statement

Jinzhong Li: Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Writing – original draft, Writing – review & editing, Project administration, Funding acquisition. **Huan Zeng:** Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Lei Peng:** Writing – review & editing, Supervision. **Jingwen Zhu:** Investigation, Supervision, Project administration, Funding acquisition. **Zhihong Liu:** Validation, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded by the National Natural Science Foundation of China, grant number 62141203 and 61762052; the Natural Science Foundation of Jiangxi Province of China, grant number 20212BAB202021; the Science and Technology Program of Jiangxi Provincial Department of Education, China, grant number GJJ180574 and GJJ201015; the Guiding Science and Technology Program of Ji'an City, grant number 2021[8]No. Base 11. All authors have read and agreed to the published version of the manuscript.

References

[1] Li H. Learning to rank for information retrieval and natural language processing. Synth Lect Hum Lang Technol 2014;7(3):1–121.
[2] Wang J, Yu L, Zhang W, Gong Y, Xu Y, Wang B, Zhang P, Zhang D. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval. 2017, p. 515–24.
[3] Zhu X, D. K. Listwise learning to rank by exploring unique ratings. In: Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM'20). 2020, p. 3–7.
[4] Liu X, Van De Weijer J, Bagdanov AD. Exploiting unlabeled data in cnns by self-supervised learning to rank. IEEE Trans Pattern Anal Mach Intell 2019;41(8):1862–78.
[5] Ai Q, Wang X, Bruch S, Golbandi N, Bendersky M, Najork M. Learning groupwise multivariate scoring functions using deep neural networks. In: Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR'19). 2019, p. 85–92.
[6] Sharma A. Listwise learning to rank with deep Q-networks. 2020, arXiv preprint arXiv:2002.07651.
[7] Guo J, Fan Y, Pang L, Yang L, Ai Q, Zamani H, Wu C, Croft WB, Cheng X. A deep look into neural ranking models for information retrieval. Inf Process Manage 2020;57(6):102067.
[8] Pasumarthi RK, Bruch S, Wang X, Li C, Bendersky M, Najork M, Pfeifer J, Golbandi N, Anil R, Wolf S. Tf-ranking: Scalable tensorflow library for learning-to-rank. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019, p. 2970–8.
[9] Zhang W. Generative adversarial nets for information retrieval: Fundamentals and advances. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018, p. 1375–8.
[10] Deshpande A, Khapra MM. Evaluating a generative adversarial framework for information retrieval. 2020, arXiv preprint arXiv:2010.00722.
[11] Jain M, Kamath SS. Improving convergence in IRGAN with PPO. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD. 2020, p. 328–9.
[12] Lu S, Dou Z, Jun X, Nie J-Y, Wen J-R. Psgan: A minimax game for personalized search with limited and noisy click data. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019, p. 555–64.
[13] Park DH, Chang Y. Adversarial sampling and training for semi-supervised information retrieval. In: The World Wide Web Conference (WWW'19). 2019, p. 1443–53.
[14] Wang B, Klabjan D. An attention-based deep net for learning to rank. 2017, arXiv preprint arXiv:1702.06106.
[15] Jiang Z, Dou Z, Zhao WX, Nie J-Y, Yue M, Wen J-R. Supervised search result diversification via subtopic attention. IEEE Trans Knowl Data Eng 2018;30(10):1971–84.
[16] Zhang C, Evans MR, Lepikhin M, Yankov D. Fast attention-based learning-to-rank model for structured map search. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021, p. 942–51.
[17] Qin X, Dou Z, Wen J-R. Diversifying search results using self-attention network. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020, p. 1265–74.
[18] Sun S, Duh K. Modeling document interactions for learning to rank with regularized self-attention. 2020, arXiv preprint arXiv:2005.03932.
[19] Pobrotyn P, Bartczak T, Synowiec M, Białobrzeski R, Bojar J. Context-aware learning to rank with self-attention. 2020, arXiv preprint arXiv:2005.10084.
[20] Zhang H, Goodfellow I, Metaxas D, Odena A. Self-attention generative adversarial networks. In: Proceedings of the 36th International Conference on Machine Learning. 2019, p. 7354–63.
[21] Xu T, Zhang P, Huang Q, Zhang H, Gan Z, Huang X, He X. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018, p. 1316–24.
[22] Emami H, Aliabadi MM, Dong M, Chinnam RB. Spa-gan: Spatial attention gan for image-to-image translation. IEEE Trans Multimed 2020;23:391–401.
[23] Jiang M, Zhi M, Li Y, Li T, Zhang J. Super-resolution reconstruction of MR image with self-attention based generate adversarial network algorithm. Science in China (Information Sciences) 2021;51(6):959–70.
[24] Mirza M, Osindero S. Conditional generative adversarial nets. 2014, arXiv preprint arXiv:1411.1784.
[25] Kim Y. Convolutional neural network for sentence classification. 2014, arXiv preprint arXiv:1408.5882.
[26] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017). 2017, p. 5998–6008.
[27] Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17). 2017, p. 5987–95.
[28] Chakrabarti S, Khanna R, Sawant U, Bhattacharyya C. Structured learning for non-smooth ranking losses. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2008, p. 88–96.