Research on CTR Prediction for Contextual Advertising Based on Deep Architecture Model ⋆

Zilong Jiang [*], **Shu Gao** [*], **Wei Dai** [**]

[*] *School of Computer Science and Technology, Wuhan University of Technology, 430063, Wuhan, China (e-mail: wuhanjzl@163.com)*
[**] *School of Economics and Management, Hubei Polytechnic University, 435003, Huangshi, China (e-mail: 3297947@qq.com)*

Abstract: Click-Through Rate (CTR) prediction is an important step in internet advertising system because it affects web publisher's profits and advertiser's payment. With the traditional machine learning models having surface architecture, the satisfying results can not be obtained by many prediction methods. This paper proposes a deep architecture model (DBNLR) that integrates deep belief network (DBN) with logistical regression (LR) to deal with the problem of CTR prediction for contextual advertising. In this model, DBN is used for automatically getting abstract and complicated features from original data that consists of contents of advertisements, users' information, click logs and pages information without any artificial intervention and prior knowledge, and then a regression model is adopted to calculate the probability value of CTR prediction. Many experiments on relative datasets show that the DBNLR model, compared with another deep architecture model SAELR, has better value of Area Under Curve (AUC) and improves the effect of CTR prediction for contextual advertising which will produce great economic benefit in the area of internet advertising.

Keywords: CTR prediction; contextual advertising; deep architecture

1. INTRODUCTION

With the development of internet, the internet advertising, as a new advertising style, usually generates a great deal of incomes for websites. Being different from traditional advertising, internet advertising has the following advantages: 1) the enormous number of netizens; 2) abundant forms, such as video advertising, contextual advertising, display advertising, sponsored search advertising, etc.; 3) precise assessment.

Contextual advertising is one of important branches of internet advertising (Maryam K. et al., 2011). Specifically speaking, it is a textual advertising usually displayed on third party web pages. The system analyses the pages in real time and extracts the feature words once a user visits the web page, and then delivers corresponding advertisements with high matching degree to the current page.

In order to maximize incomes, it is necessary for web publisher to predict the expected incomes for each advertisement under current circumstances and choose a certain amount of advertisements from optional advertisements to deliver. The CTR prediction is defined to estimate the ratio of clicks times to impressions of advertisements that will be displayed in the internet (Thore G. et al., 2010). In the common cost-per-click model, the advertiser pays the web publisher only when a user clicks their advertisements and visits the advertiser's site (Thore G. et al., 2010). Nowadays, CTR prediction for internet advertising has attracted widespread attention of researchers of industry and academia for its importance in advertisement selection.

Trofimov and his colleagues from Yandex company solved the click-through rate prediction problem in sponsored search by means of MatrixNet, and studied different related problems such as evaluating and tuning the MatrixNet algorithm, feature importance, performance, accuracy and training dataset size (Ilya T. et al., 2012). Konig and Christian from Microsoft company proposed a supervised model based on Multiple Additive Regression Trees that offered accurate prediction of news click-through rates and satisfies the requirement of adapting quickly to emerging news events (Konig A. C. et al., 2009). Yukihiro Tagami, et al. from Yahoo Japan Corporation introduced a click-through rate prediction algorithm based on the learning-to-rank approach, which defined a ranking model by using partial click logs and then used a regression model on it (Yukihiro T. et al., 2013). Rohit Kumar, et al. from B V Bhoomaraddi College of Engineering and Technology adopted logistic regression to predict the CTR of search engine advertising and achieved around 90% accuracy on a dataset of around 25 GB (Kumar R. et al., 2015). Afroze I. Baqapuri and Ilya Trofimov proposed a novel architecture to solve the CTR prediction for sponsored search advertising by combining artificial neural networks (ANN) with boosted trees (Afroze I. Baqapuri et al., 2015).

Research on CTR prediction mainly concentrates on industry according to the above literatures, traditional models are widely adopted due to their convenience of realization and utility. Classical statistics models for CTR

⋆ This project is supported by Humanities and Social Science Youth Foundation of Ministry of Education of China, No.13YJCZH028.

prediction such as logistic regression (Haibin C. et al., 2012), boosted trees (Afroze I. Baqapuri et al., 2015; Ilya T. et al., 2012; Dave K. S. et al., 2010), regression trees, decision trees and probit regression (Thore G. et al., 2010), etc, are widely used, with their precision greatly relying on the design of features. However, the complex mapping relation, especially the data with abundant meaning can not be efficiently expressed by these surface models and traditional artificial neural networks (Bengio Y., 2009). Therefore, surface models for CTR prediction have the deficiency of feature extraction and prediction precision.

2. DEEP LEARNING AND ITS APPLICATION

Deep learning, proposed by Geoffrey Hinton, et al. (Hinton G. E., 2006), is a kind of neural network (NN) consisting of a large numbers of simple neural nodes and the real objects are abstractly expressed through simulating the learning mechanism of human brain in terms of this neural network. In the deep learning, the neural nodes of every layer receive the input of lower layers. Through the non-liner mapping relation between input and output, the sample data are mapped to features in the different layers from bottom to top which shows the strong ability of learning essential features of dataset from the minority sample datasets (Bengio Y. and Delalleau O., 2011; Bengio Y. and Lecun Y., 2007).

As for the deep belief network (DBN), Hinton put forward the training algorithm step by step in unsupervised way to solve the optimizing problem of deep architecture (Hinton G. E., 2012). Then, he proposed the multi-layer auto encoder (AE) deep architecture (Wei Luo et al., 2015). Lecun Y., et al. put forward convolutional neural networks (CNN), which is the first real learning algorithm with multi-layer structure (Krizhevsky A. et al., 2012). Moreover, many researchers explore the deformation structure of deep learning, such as denoising auto encoders (Vincent P. et al., 2011), deep convex net (DCN) (Yu Dong and Deng Li, 2011), deep stacking networks (DSN)(Li Deng et al., 2013), and so on.

AE and DBN are widely used in the areas of speech recognition, natural language process (NLP), image data processing whereas CNN is usually used in the area of image data processing. Xinting Gao, et al. from Institute for Infocomm Research of Singapore proposed a deep learning system to automatically learn features for grading the severity of nuclear cataracts from slit-lamp images. Local filters learned from image patches are fed into a convolutional neural network to further extract higher-order features. Based on these features, support vector regression model is adopted to calculate the cataract grade (Xinting Gao et al., 2014). Kuniaki Noda, et al. from Waseda University used a deep denoising autoencoder to obtain noise-robust audio features and used a convolutional neural network to extract visual features from raw mouth area images in audio-visual speech recognition system (Kuniaki Noda et al., 2015). Feng Shen, et al. from University of Science and Technology Beijing proposed a deep learning model that consists of auto encoder and support vector machine to solve the problem of classification and dimension reduction of Chinese web texts (Feng Shen et al., 2013). Allan Campbell, et al. adopted deep belief network to discover features of evolved abstract art images. They trained a deep belief network with 10 layers and used the output activities at each layer as training data for traditional classifiers (decision tree and random forest) (Allan Campbell et al., 2015). E. M. Albornoz, et al. from Universidad Nacional del Litoral took advantage of restricted boltzmann machines and deep belief networks to classify emotions in speech (E. M. Albornoz et al., 2014). Li Deng, et al. from Microsoft Research of Redmond successfully applied deep stacking networks (DSN) to information retrieval (IR) task. The DSN-based system outperformed the LambdaRank-based system which represented a recent state-of-the-art for IR in normalized discounted cumulative gain measures (Li Deng et al., 2013).

Currently, there are few relative literatures of deep learning applying to the area of contextual advertising. This paper puts forward a deep architecture model DBNLR for dealing with the problem of CTR prediction for contextual advertising.

This paper introduces the background knowledge and relative theories of deep learning in section 3 and describes the design steps of deep architecture model DBNLR for CTR prediction in section 4. In section 5, the details about training and tuning of DBNLR model and another deep architecture model SAELR for CTR prediction are given and the performance between DBNLR model and SAELR model in the same datasets are compared. This paper comes to the conclusion and describes the future work in last section.

3. RELATIVE THEORY

Because this paper adopts the deep learning approach DBN, the introduction of relative theory will be focused on the component of DBN, that is restricted boltzmann machine (RBM).

3.1 RBM

Restricted boltzmann machine is a two-layer boltzmann machine, in which there are connections between all nodes but no connections in the same layer (Rumelhart D. et al., 1986). In other words, RBM does not exist in visible-visible or hidden-hidden connections. As shown in figure 1, the first layer is the set of visible nodes (input layer) and the second one is the set of hidden nodes. V denotes the set of visible nodes (or visible vector) whereas H denotes the set of hidden nodes (or hidden vector). The state of all these nodes is binary random variables +1 or -1 (Hinton G. E., 2006).

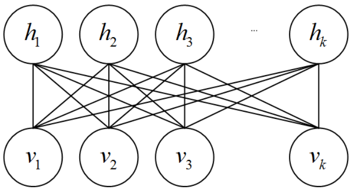

Fig. 1. The architecture of RBM

The energy field of RBM can be represented by energy function $E(v,h|\Theta)$:

$$E(v,h|\Theta) = -\sum_{i=1}^{I} a_i v_i - \sum_{j=1}^{J} b_j h_j - \sum_{i=1}^{I}\sum_{j=1}^{J} w_{ji} v_i h_j \quad (1)$$

In the formula, the variable a_i denotes the bias of the visible node i, the variable b_j denotes the bias of the hidden node j and the variable w_{ji} denotes the weight of connection edge between the visible node i and the hidden node j. The RBM assigns a probability for each state of configure (v, h), using the energy function given by formula (2):

$$p(v,h) = \frac{exp(-E(v,h|\Theta))}{Z} \tag{2}$$

Z denotes the normalizing constant given by formula (3):

$$Z = \sum_v \sum_h exp(-E(v,h|\Theta)) \tag{3}$$

This probability $p(v, h)$ (formula (2)) is measured through the ratio of the energy of current joint configuration state over both visible and hidden nodes / the normalizing constant. The normalizing constant represents the sum of energy of all the joint configurations state over both visible and hidden nodes.

The probability $p(v)$ of the visible nodes set V is calculated as follows:

$$p(v) = \sum_h p(v,h) = \frac{\sum_h exp(-E(v,h))}{Z} \tag{4}$$

RBM possesses the properties that if all the hidden nodes are given, each visible node of visible nodes set is interindependent and can be obtained independently, and vice versa. Then the conditional probability of hidden nodes h and the visible vector v can be expressed in the following formulas:

$$p(h|v) = \prod_j p(h_j|v) \tag{5}$$

$$p(v|h) = \prod_i p(v_i|h) \tag{6}$$

They can be sampled from the the conditional probability $p(v_i|h)$ and $p(h_j|v), i \in \{1, \ldots, I\}, j \in \{1, \ldots, J\}$, the activation function of a node is the logistic function in formula (12).

With regard to a given random input vector v includes the state of all visible nodes, and the state of the hidden node j is set to 1 in probability:

$$p(h_j = 1|v) = \sigma(b_j + \sum_{i=1}^{I} v_i w_{ji}) \tag{7}$$

In the same way, a random hidden vector h is given, the state of the visible node i can be set to 1 in probability:

$$p(v_i = 1|h) = \sigma(a_i + \sum_{j=1}^{J} h_j w_{ji}) \tag{8}$$

In the right side of formula (7) and (8), $\sigma(x) = \frac{1}{1+e^{-x}}$, the variable a_i is the bias of the visible node i, and the variable b_j is the bias of the hidden node j.

Therefore, the probability of the visible vector v, is calculated by the formula: $p(v) = \sum_h p(v, h) = \sum_h p(v|h)p(h)$, and the probability value can be increased by adjusting the weights and the biases of RBM network.

3.2 Learning Algorithm of RBM

In order to update the weights and the biases of RBM network, Hinton put forward the Contrastive Divergence (CD-1) algorithm (Hinton G. E., 2012). The algorithm, as the most common and efficient training approach for RBM with depending on the approximation to run the sampler only for a single Gibbs iteration instead of many iterations until the chain converges in the end, presents an approximation of the gradient of another objective function. The following rules can be applied to update the parameters (weights and bias) of network:

$$\Delta w_{ij} = \varepsilon(< v_i h_j >_0 - < v_i h_j >_1)$$
$$\Delta b_j = \varepsilon(< h_j >_0 - < h_j >_1)$$
$$\Delta a_i = \varepsilon(< v_i >_0 - < v_i >_1)$$

There are two phases (positive and negative) for training the network in the process of using one-step Gibbs sample before updating the weights and biases. During the positive phase, the conditional probability of hidden node $p(h_j|v)_0$ is calculated based on the input vector which is given by formula (7). Based on this probability, the state of hidden node $< h_j >_0$ is sampled. The symbol $<>_0$ denotes the state of the node before reconstruction, whereas the symbol $<>_1$ denotes the state of the node after a one-step reconstruction. During the negative phase, the nodes of hidden layer are used for reconstructing the nodes of visible layer by sampling according to formula (8), and then the hidden layer is subsequently recalculated from the visible layer by means of formula (7). ε denotes the learning rate. The overall training process is summarized below:

The steps of CD-1 Algorithm:
Step-1: $< v_i >_0 \leftarrow D$; D is all input data; epoch = 0;
Step-2: calculate $p(h_j|v)_0$ from formula (7);
Step-3: $< h_j >_0 \sim p(h_j|v)_0$; sample $< h_j >_0$ from $p(h_j|v)_0$;
Step-4: calculate $< v_i >_1$ by sampling according to formula (8);
Step-5: calculate $< h_j >_1$ by sampling according to formula (7);
Step-6: update weights and biases

$$w'_{ij} = w_{ij} + \varepsilon(< v_i h_j >_0 - < v_i h_j >_1) \tag{9}$$

$$b'_j = b_j + \varepsilon(< h_j >_0 - < h_j >_1) \tag{10}$$

$$a'_i = a_i + \varepsilon(< v_i >_0 - < v_i >_1) \tag{11}$$

Step-7: If the terminal condition appears, the process will be terminated. Otherwise, the epoch is assigned to a new value (epoch plus one), and the process will return to step 2.

4. CTR PREDICTION APPROACH FOR CONTEXTUAL ADVERTISING BASED ON DEEP ARCHITECTURE MODEL

4.1 Data Pre-processing

This paper chooses the contents of advertisements, users' information, pages information and click logs as main features source.

First of all, the Chinese text contents of web pages are split into words by IKAnalyzer tool in Lucene (IKAnalyzer, 2015; Pirro G. et al., 2009). All stop words are removed referring to the Chinese corpus of People's Daily, and each web page with remaining words is regarded as a document of page. Meanwhile, the corresponding URL of web page provides a unique pageID with the way of encryption for this document of page. All information above is saved as a record in corresponding tables of database to form the dataset of current website pages.

Secondly, users' information can be obtained by means of users' registration information or partially from click logs. These basic users' characteristics, for example, the visiting person's age, userID, career, residence, etc., are collected. Each user's information is saved as a record in corresponding tables of database to form the dataset of users' information.

Thirdly, a contextual advertisement has the characteristics of both information of contents and registration. This research need to cut the contents of contextual advertisement to get partial features through IKAnalyzer, and collect advertisement's properties of registration for getting partial features. Each set of advertisement characteristics is saved as a record in corresponding tables of database to form the dataset of advertisement information.

Fourthly, the fields of userID, pageID, advertisementID, click information of advertisement, etc. are extracted from every record of click logs. According to the dataset of current website pages, dataset of users' information dataset and dataset of advertisement information, userID, current URL and advertisementID can find their respective information in corresponding datasets. All corresponding information contained in above three kinds of datasets are jointed to form a new record containing users' information, information of page words, information of advertisement features and click information and these records are regarded as an new input records dataset CP3.

Fifthly, scanning all records of dataset of current website pages, dataset of users' information, dataset of advertisement information can generate an overall vector space made up of all the present words in lexicographical order.

Sixthly, every record in new input records set CP3 corresponds to a vector of vector space. The count which every word appears in a record is calculated and is mapped into a value between 0 and 1 by means of the sigmoid function $\sigma(x)$. Every element of a vector consists of the mapping values of corresponding word elements in this record, and these vectors will be regarded as the input of deep architecture model.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{12}$$

4.2 Features Extraction Based on DBN

Deep belief network is a generative model which is composed of a stack of restricted boltzmann machines and is trained in a greedy layer-wise unsupervised manner, relying on the training algorithm of restricted boltzmann machines to initialize the parameters of a deep belief network (Hinton G. E. and Salakhutdinov R. R., 2006). The architecture of DBN is shown in the figure 2. The input values of visible layer of the first RBM are directly derived from the original data. After training the parameters of the first RBM with CD-1 algorithm, the obtained nodes of hidden layer can be considered as another compact expression of the input vector. Then, the parameters (weight, bias) of the first RBM are fixed, the hidden nodes (h1) of second layer are regarded as input vector to get hidden nodes (h2) of the output layer of another RBM, and so on. Each new layer is stacked on the top of the current DBN. Lower layer is used for detecting simple features and feeding into upper layer, which gradually refines previous learned information and further finds more complex features, thereby more abstract information representing underlying the regularities of input data are generated (Ranzato M. et al., 2007; Roux N. L. et al., 2008). The training process is done in an unsupervised manner, without relying on manual interventions (Krizhevsky A. et al., 2012; Lee H. et al., 2007; Bengio Y., 2009).

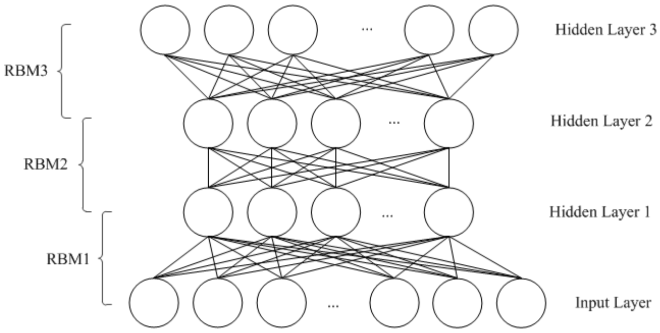

Fig. 2. The architecture of DBN

Because only a RBM is trained each time, the learned parameters (weight and bias) of current RBM just makes sure that the mapping relation to the input of current RBM is locally optimal instead of the whole DBN is optimal. Then, a classifier will be added onto the DBN. Finally, the data with label is used for fine-tuning the parameters of the whole network in a top-down and supervised way (Goodfellow I. J. et al., 2009). The process of fine-tuning will be presented in the following experiment stage.

4.3 The Calculation of CTR Prediction Based on the Logical Regression Model

When input data passed the whole DBN, learned features on the top layer of the DBN are the most representative features for modeling the original input data (Hinton G. E., 2012; Ranzato M. et al., 2007), and are regarded as the input vector of regression layer.

Click-Through Rate (CTR) denotes the probability of an advertisement is clicked by users. It is a classic regression problem, this paper adopts logistic regression to calculate the value of CTR prediction for this case.

The input vector x of logistic regression model is a set of abstracted features that contain the contents of advertisements, current users' information, and current pages information. The value of Y will be assigned to 1 only when advertisement r is clicked. The expression $P(Y = 1|x)$ represents the probability that advertisement r is clicked when input vector x is given. As shown in the following formula,

$$\begin{aligned} CTR^{(r)} &= P(Y=1|x) \\ &= \frac{exp(w \cdot x + b)}{1 + exp(w \cdot x + b)} \\ &= \frac{1}{1 + exp(-w \cdot x + b)} \end{aligned} \tag{13}$$

In this formula, b denotes a definite parameter, x is the input vector consisting of set of all abstracted features from DBN, and w is the learned weight vector for these features in logistic regression model.

In another way, this paper extends input vector and weight vector, $w = (w^{(1)}, \ldots, w^{(n)}, b)^T$, $x = (x^{(1)}, \ldots, x^{(n)}, 1)^T$, then logistic regression model can be expressed as succinctly:

$$CTR^{(r)} = P(Y = 1|x) \quad = \frac{1}{1 + exp(-w \cdot x)} \qquad (14)$$

To obtain the unknown parameter value w, this paper adopts the maximum likelihood estimation approach to learn the value of w while training datasets is given. $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}, x_i \in \mathbf{R}^n, y_i \in \{0, 1\}$.

Supposed: $P(Y = 1|x) = \pi(x), P(Y = 0|x) = 1 - \pi(x)$. Likelihood function is shown in the following formula: $\prod_{i=1}^{N}[\pi(x_i)]^{y_i}[1-\pi(x_i)]^{1-y_i}$. Whereas Log likelihood function is shown in the following formula:

$$\begin{aligned} L(w) &= \sum_{i=1}^{N}[y_i \log \pi(x_i) + (1 - y_i) \log(1 - \pi(x_i))] \\ &= \sum_{i=1}^{N}[y_i \log \frac{\pi(x_i)}{1 - \pi(x_i)} + \log(1 - \pi(x_i))] \qquad (15) \\ &= \sum_{i=1}^{N}[y_i(w_i \cdot x_i) - \log(1 + exp(w_i \cdot x_i))] \end{aligned}$$

w_i denotes the weight vector corresponding to the input vector x_i. Estimated value of w can be given if maximum of $L(w)$ is obtained. Gradient descent approach (Burges C. et al., 2005) is adopted to calculate the maximum of $L(w)$, maximum likelihood estimation $\hat{w}$ of w can be obtained.

Finally, the click probability of advertisement r can be calculated by using the logistic regression model, as follows:

$$CTR^{(r)} = P(Y = 1|x) = \frac{1}{1 + exp(-\hat{w} \cdot x)} \qquad (16)$$

The probability produced in the regression layer is the result of CTR prediction for advertisement r.

4.4 The Architecture and Algorithm Idea of DBNLR Model

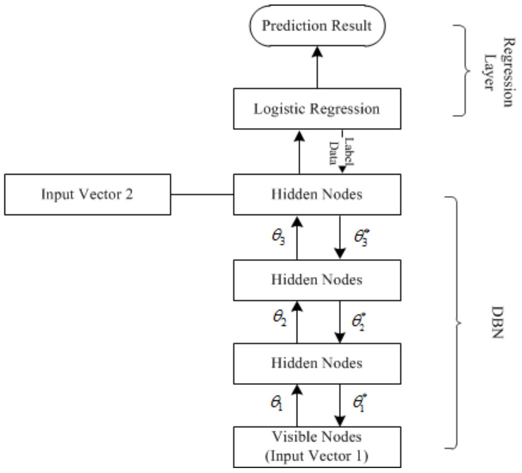

Fig. 3. The architecture of DBNLR model

The architecture of DBNLR model can be shown in figure 3. The idea of DBNLR model can be summarized as follows.

Firstly, this paper pre-processes data including advertisements information, users' information, pages information and click logs according to available format. Secondly, DBN is applied to extract features from the inputs through multi-layers of nonlinear features transformation to give expression on the complicated mapping relation between inputs and outputs in the contextual advertising system. After passing the DBN, the abstract and representative features with low dimensionality and high cohesion will be obtained from original data with high dimensionality. Lastly, a logical regression layer is used for generating the result of CTR prediction based on these abstract features learned by DBN.

5. DESIGN OF EXPERIMENTS AND ANALYSIS OF RESULTS

5.1 Source Data and Dataset of Training and Testing

This paper collects the data that includes all pages information, and users' information of "a regional portal website", all advertisements records of advertisement lib "OKadlib" and all users' click logs of this website. After data pre-processing, 143750 pieces of joint records are produced. All joint records are divided into training dataset and testing dataset according to 3:1 ratio. In the following training process of models, 10-fold cross-validation (Arlot S. and Celisse A., 2010; Bengio Y. and Grandvalet Y., 2004) will be used.

5.2 Hardware Equipments

The workstations with high performance of a large steel group company are used in the experiments. The hardware configuration of workstations is shown as follows.

Table 1. Hardware equipments

Name of equipment	Type and parameters
CPU	Intel XEON E5-2687W*2
Memory	32GB DDR3 ECC REGS 1600MHz*32
GPU	Tesla GPU K20*2
Hard Disk	1T SATA

5.3 Software

Python 2.7 is used for developing programmes and the Theano python library is used for building and training the deep architecture.

5.4 Metric of Performance Evaluation of Models

This paper adopts the Area Under Curve (AUC) (Fawcett T., 2006) to measure the result of prediction. Traditional metric, such as precision, can not properly reflect the performance of classifiers because the samples in different classifications are unbalanced. Receiver Operating Characteristic (ROC) (Fawcett T., 2006) can be introduced as a new metric of performance evaluation. ROC focuses on the two metrics of confusion matrix—True Positive Ratio (T-PR)and False Positive Ratio (FPR), in the space of ROC as shown in table 2, the abscissa of every point represents the FPR and the ordinate of every point represents the TPR, which describes the weigh of classifier between TP and FP. $TPR = \frac{TP}{TP+FN}$, $FPR = \frac{FP}{FP+TN}$.

The value of CTR prediction in this paper, $CTR^{(r)} =$

Table 2. Confusion matrix

	Positive	Negative
True	True Positive(TP)	True Negative(TN)
False	False Positive(FP)	False Negative(FN)

$P(Y=1|x^{(r)})$, is between 0 and 1, meanwhile, a threshold (such as 0.002) is set according to the records of sample. If value of CTR prediction is bigger than the threshold, the classification of this sample is assigned to 1. If value of CTR prediction is smaller than the threshold, the classification of this sample is assigned to 0. According to the prediction values in testing dataset and the results of classification, the corresponding TPR and FPR can be calculated. The TPR and FPR can conceive a point in ROC space. Different points are jointed to a curve in ROC space. AUC is the value of area under the curve in ROC space. Therefore, the AUC value can be used for evaluating whether the performance of classifier is good or not. In the performance evaluation, high AUC value represents the good performance.

5.5 The Process and Analysis of Experiments

In the process of training the deep architecture DBNLR, each restricted boltzmann machines (RBM) is trained in an unsupervised learning manner using contrastive divergence -1 algorithm. The trained restricted boltzmann machines (RBM) are stacked to form a DBN. Then, a logical regression layer is added onto DBN to generate the DBNLR model. Finally, in the fine-tuning step, backpropagation in a supervised manner using data with class labels (Ian J. Goodfellow et al., 2009) and gradient descent method are applied to fine-tune the weight and bias of the whole deep architecture model.

5.5.1. The Training and Optimizing of DBNLR Model

Before the training of DBNLR model, when the stop words are removed from the dataset containing 143750 pieces of records, the dimensionality of all present words is 5369. Therefore, the number of nodes of visible layer (input layer) is 5369. According to the experience, when the number of nodes of visible layer is big, the nodes of hidden layer need the process "dimensionality reduction - dimensionality addition - dimensionality reduction". However, when the number of nodes of visible layer is small, the nodes of hidden layer just need the process "dimensionality addition - dimensionality reduction". In general, the number of hidden nodes about increases or decreases with multiple. Currently, there is no method of quickly setting the number of hidden layers and the number of hidden nodes of every layer. Therefore, only many experiments and experience are made to search for the relative optimal structure, which is also the future research direction. In the process of training the DBNLR model, the initial control parameters of precision can be presented in the following table 3.

Table 3. The initial control parameters of precision of DBNLR model

Name of parameters	Value
Unsupervised learning rate	0.01
Supervised learning rate	0.1
Number of unsupervised epochs	10
Rate of fine-tuning	0.1
Number of fine-tuning	1000

To avoid the under-fitting and over-fitting of model, 10-fold cross-validation (Arlot S. and Celisse A., 2010; Bengio Y. and Grandvalet Y., 2004) is used in the training dataset. However, the computation of generalization errors in 10-fold cross-validation is replaced by the computation of AUC value.

After 20 experiments with different configuration of hidden layers and hidden nodes, the highest AUC value is obtained when the layers and nodes are shown as follows: the number of hidden layers is 3; the nodes of the first hidden layer are 50; the nodes of the second hidden layer are 500; the nodes of the third hidden layer are 100. This paper chooses this configuration as the final structure. The data of partial experimental results is shown in the table 4. The AUC value in the table 4 is the average in 10-fold cross-validation.

5.5.2. The Training and Optimizing of SAELR Model

The CTR prediction for sponsored search advertising is anaylzed by means of integrating artificial neural networks (ANN) with boosted trees in the literature (Afroze I. Baqapuri et al., 2015). In this analysis, ANN is used for extracting the features, however, boosted trees for anaylzing the sponsored search advertising in this literature is not suitable for current contextual advertising and ANN can not overcome the problems of the rate of convergence and local minimum point. Therefore, this paper adopts another deep learning approach—sparse auto encoder (SAE) in the literature (Feng Shen et al., 2013) to construct another deep architecture model SAELR for anaylzing contextual advertising.

Auto encoder is a kind of neural network which regenerates input signals as far as possible. It assumes that the input and output is equal and adopts the unlabelled data to train and adjust the parameters of network. The specific process is shown as follows.

(1) A code can be obtained when an unlabeled data is inputted into an encoder, and then in terms of the decoder this code is decoded into some information, which conceives an error with the inputing information. In order to make the error of reconstruction get the smallest value, the parameters of encoder and decoder need to be adjusted. Finally, the code becomes an expression of inputting information.

(2) The features (code) which are generated by encoder are regarded as the input of next layer. According to the step 1, every layer is trained and a network with multi-layers is obtained.

This paper realizes the CTR prediction approach for contextual advertisement based on SAELR model and trains it by means of the 10-fold cross-validation in all datasets. In the process of training the SAELR model, the initial control parameters of precision can be presented in the following table 5.

The nodes of hidden layer in SAELR model are the feature representation of input data. The number of nodes of hidden layer has a large impact on the precision of prediction result. If the number of nodes of hidden layer is too small, the feature representation will be dense. On

Table 4. AUC values of DBNLR model with different layers and nodes

AUC	Number of layers	Nodes of h1	Nodes of h2	Nodes of h3	Nodes of h4	Nodes of h5
0.7427	3	40	400	0	0	0
0.7982	4	40	400	80	0	0
0.7854	5	40	400	200	100	0
0.8403	4	50	500	100	0	0
0.8316	5	50	500	250	50	0
0.8331	4	60	600	120	0	0
0.8206	6	60	600	300	150	75
0.8182	6	70	700	350	140	70
0.8229	4	70	700	140	0	0
0.8130	6	100	1000	500	250	125

Table 5. The initial control parameters of precision of SAELR model

Name of parameters	Value
Learning rate	0.01
Decline rate	0.3
Epochs	100

the contrary, if the number of nodes of hidden layer is too big, the feature representation will be sparse, the data can not be efficiently described in above two cases.

The results of many experiments are showed in the figure 4. As for the nodes of input layer with 5369 dimensionality, the result is optimal when the nodes of hidden layer are 400. More nodes of hidden layer can just increase the training time instead of improving the precision. This paper chooses this configuration with 400 nodes of hidden layer as the final structure of SAELR model.

In the figure 4, the AUC values of SAELR model with different nodes of hidden layers in training process are described. The AUC value in the figure 4 is the average in 10-fold cross-validation.

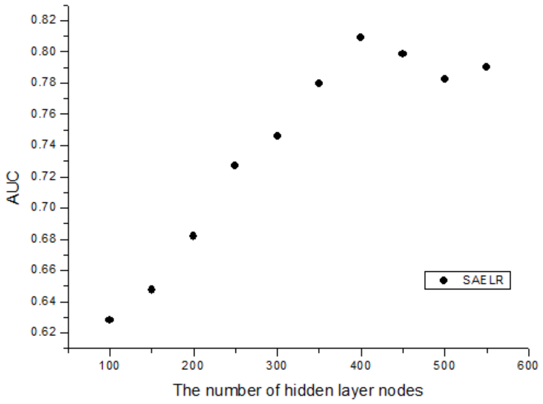

Fig. 4. The AUC values of SAELR model with different nodes of hidden layers

5.5.3 The Comparison of Performance between DBNLR Model and SAELR Model

In order to make a comparison between two models, this paper distributes the whole testing dataset into equal 10 portions. Testset1 contains all data of portion 1 and testset2 contains all data of portion1 and portion 2. By this analogy, testset10 contains all data of 10 portions. These two models are simultaneously operated in different scale of testing datasets. The running results of these two models are shown in the figure 5. According to the figure 5, when the scale of testing datasets is small, the AUC values of SAELR model are bigger than that of DBNLR model. However, with the increasement of scale of testing datasets, the AUC values of DBNLR model are bigger than that of SAELR model. In the testset8, the difference of AUC values of these two models is nearly 6.2 percent. It demonstrates that the ability of extracting abstract features of DBNLR model is stronger than that of SAELR model in current application.

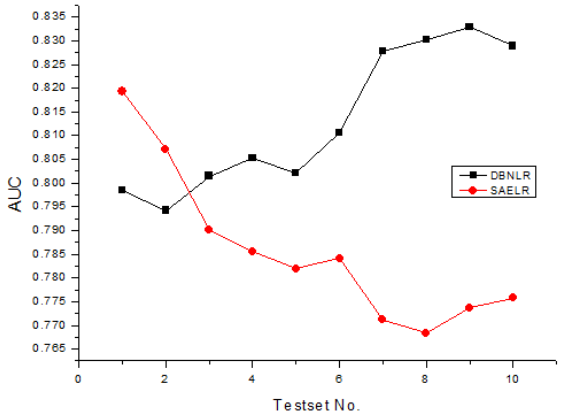

Fig. 5. The AUC values of two models in testing datasets with different scale

As for the running time, it can be shown in the figure 6. According to the figure 6, the running time of DBNLR model is longer than that of SAELR model in every testing dataset. In the testset10, the difference of running time is 44.7 seconds. However, it is acceptable for the difference of time for the sake of high precision.

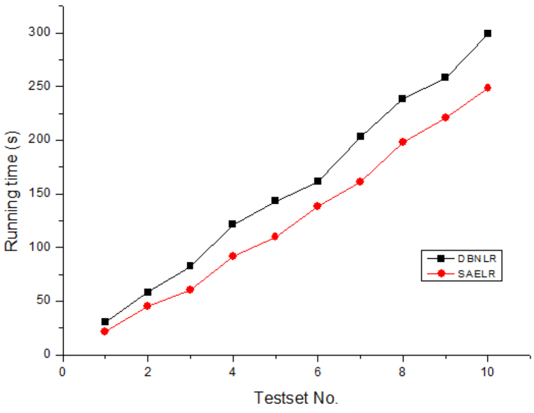

Fig. 6. The running time of two models in testing datasets with different scale

6. CONCLUSION AND FUTURE WORK

This paper proposes a deep architecture model DBNLR to deal with the problem of CTR prediction for contextual advertising. In this model, the features and abstract relation of web pages, advertisements and users' information are adequately learned, which is useful for efficiently improving the precision of CTR prediction for contextual advertisement.

In addition, this paper makes a comparison of performance between DBNLR model and another deep architecture model SAELR in different scale of testing datasets and finds that although the time of training and operation of DBNLR model is longer than that of SAELR model, the AUC value of DBNLR model is higher than that of SAELR model in large scale of testing datasets.

Meanwhile, our future work will concentrate on the following researches. Owing to the long time of training DBNLR model, the first future work will focus on the parallelization of training method based on multithread, multi-cores or GPU (Gabriel Munteanu et al., 2015). Because there is no method of quickly setting the number of hidden layers and the number of hidden nodes of every layer, the second future work will focus on finding the objective method to deal with this case with fewer experiments. The third future work will focus on exploring more deep architecture models to improve the CTR prediction for contextual advertising.

REFERENCES

Afroze I. Baqapuri, Ilya Trofimov. (2015). Using neural networks for click prediction of sponsored search. *http://arxiv.org/abs/1412.6601.*

Allan Campbell, Vic Ciesielksi, and A. K. Qin. (2015). Feature discovery by deep learning for aesthetic analysis of evolved abstract images. *The 4th International Conference on Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMUSART 2015)*, Copenhagen, Denmark, pp. 27-38.

Arlot S., Celisse A..(2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, vol.4, pp. 40-79.

Bengio Y.. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, vol. 2(1), pp. 1-12.

Bengio Y., Delalleau O.. (2011). On the expressive power of deep architectures. *Algorithmic Learning Theory*, Berlin, Heidelberg, pp. 18-36.

Bengio Y., Grandvalet Y.. (2004). No unbiased estimator of the variance of K-fold cross-validation. *Journal of Machine Learning Research*, vol.5, pp. 1089-1105.

Bengio Y., Lecun Y.. (2007). Scaling learning algorithms towards AI. *Large-Scale Kernel Machines*, Massachusetts: MIT Press, pp. 1-34.

Burges C., Shaked T., Renshaw T., et al..(2005). Learning to rank using gradient descent. *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, Bonn, Germany, pp. 89-96.

Dave K. S., Varma, V.. (2010). Learning the click-through rate for rare/new ads from similar ads. *Proc. of the 33rd Annual International Conference on Research and Development in Information Retrieval (ACM SIGIR 2010)*, Geneva, Switzerland, pp. 897-898.

E. M. Albornoz, M. Sanchez Gutierrez, F. Martinez Licona, H. L. Rufiner, and J. Goddard.. (2014). Spoken emotion recognition using deep learning. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications Lecture Notes in Computer Science*, vol. 8827, pp. 104-111.

Fawcett T..(2006). An introduction to ROC analysis. *Pattern Recognition Letters*, vol. 27(8), pp. 861-874.

Feng Shen, Xiong Luo, Yi Chen. (2013). Text classification dimension reduction algorithm for Chinese web page based on deep learning. *International Conference on Cyberspace Technology (CCT 2013)*, Beijing, China, pp. 451-456.

Gabriel Munteanu, Stefan Mocanu, Daniela Saru.(2015). GPGPU optimized parallel implementation of AES using C++ AMP. *Control Engineering and Applied Informatics*, vol. 17(2), pp. 73-81.

Goodfellow I. J., Le Q. V., Saxe A. M., and Ng Andrew Y.(2009). Measuring invariances in deep networks. *The 23rd Annual Conference on Neural Information Processing Systems (NIPS 2009)*, Vancouver, BC, Canada, pp. 646-654.

Haibin C., Roelof Z. V., Eren M. J., et al. (2012). Multimedia features for click prediction of new ads in display advertising. *Proc. of the 18th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD 2012)*, Beijing, China, pp. 777-785.

Hinton G. E.. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, vol. 18(7), pp. 1527-1554.

Hinton G. E..(2012). A practical guide to training restricted boltzmann machines. *Neural Networks: Tricks of the Trade*, vol. 16(7), pp. 599-619.

Hinton G. E., Salakhutdinov R. R.. (2006). Reducing the dimensionality of data with neural networks. *Scinece*, vol.313, pp.5786:504.

Ian J. Goodfellow, Quoc V. Le, Andrew M. Saxe, Honglak Lee and Andrew Y. Ng . (2009). Measuring invariances in deep networks. *23rd Annual Conference on Neural Information Processing Systems (NIPS 2009)*, Vancouver, BC, Canada, pp. 646-654.

IKAnalyzer.(2015). *http://code.google.com/p/ik-analyzer/downloads/list.*

Ilya T., Anna K., Valery T.. (2012). Using boosted trees for click-through rate prediction for sponsored search. *Proc. of the 6th International Workshop on Data Mining for Online Advertising and Internet Economy*, Beijing, China, pp. 131-136.

Konig A. C., Gamon, M., Qiang W.. (2009). Click-through prediction for news queries. *Proc. of the 32nd Annual International Conference on Research and Development in Information Retrieval (ACM SIGIR 2009)*, Boston, MA, United States, pp. 347-354.

Krizhevsky A., Sutskever I., and Hinton G. E..(2012). Imagenet classification with deep convolutional neural networks. *Proc. of the 26th Annual Conference on Neural Information Processing Systems (NIPS 2012)*, Lake Tahoe, NV, United States, pp. 1097-1105.

Kumar R., Naik S. M., Naik V. D., Shiralli S., Sunil V. G., Husain M..(2015). Predicting clicks: CTR estimation of advertisements using logistic regression classifier. *IEEE International Advance Computing Conference (IACC 2015)*, Banglore, pp. 1134-1138.

Kuniaki Noda, Yuki Yamaguchi, Kazuhiro Nakadai, Hiroshi G. Okuno and Tetsuya Ogata .(2015). Audio-visual speech recognition using deep learning. *Applied Intelligence*, vol.42, pp. 722-737.

Lee H., Ekanadham C., and Andrew N. Y.. (2007). Sparse deep belief net model for visual area v2. *Proc. of the 21st Annual Conference on Neural Information Processing Systems (NIPS 2007)*, Vancouver, BC, Canada, pp. 873-880.

Li Deng, Xiaodong He, and Jianfeng Gao. (2013). Deep stacking networks for information retrieval. *The 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, Vancouver, Canada, pp. 3153-3157.

Maryam K., Wei L., Ruofei Z., and Jianchang M..(2011). A stochastic learning-to-rank algorithm and its application to contextual advertising. *Proc. of the 20th International Conference on World Wide Web*, Hyderabad, India, pp. 377-386.

Pirro G., Talia D..(2009). An approach to Ontology mapping based on the Lucene search engine library. *Proc. of the 18th International Workshop on Database and Expert Systems Applications*, Regensburg, Germany, pp. 407-411.

Ranzato M., Boureau Y., and Lecun Y..(2007). Sparse feature learning for deep belief networks. *Proc. of the 21st Annual Conference on Neural Information Processing Systems (NIPS 2007)*, Vancouver, BC, Canada, pp. 237-251.

Roux N. L., Bengio Y..(2008). Representational power of restricted boltzman machines and deep belief networks. *Neural Computation*, vol. 20(6), pp. 1631-1649.

Rumelhart D., McClelland J..(1986). Information processing in dynamical systems: Foundations of harmony theory. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1 (1), pp. 194-281.

Thore G., Candela J. Q., Thomas B., and Ralf H..(2010). Web-scale bayesian click-through rate prediction for sponsored search advertising in microsofts bing search engine. *Proc. of the 27th International Conference on Machine Learning (ICML 2010)*, Haifa, Israel, pp. 13-20.

Vincent P., Larochelle H., Lajoie I., et al.. (2011). Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, vol.11(12) , pp. 3371-3408.

Wei Luo, Jian Yang, Wei Xu, and Tao Fu. (2015). Locality-constrained sparse auto-encoder for image classification. *IEEE Signal Processing Letters*, vol. 22(8), pp. 1070-1073.

Xinting Gao, Stephen Lin, Tien Yin Wong. (2014). Automatic feature learning to grade nuclear cataracts based on deep learning. *The 12th Asian Conference on Computer Vision (ACCV 2014)*, Singapore, pp. 632-642.

Yu Dong, Deng Li. (2011). Deep convex net: a scalable architecture for speech pattern classification. *Proc of the 12th Annual Conference of International Speech Communication Association*, Florence, Italy, pp. 2285-2288.

Yukihiro T., Shingo O., Koji Y.. (2013). CTR prediction for contextual advertising: learning-to-rank approach. *Proceedings of the 7th International Workshop on Data Mining for Online Advertising*, Chicago, IL, United States, pp. 421-435.