Kỷ yếu Hội nghị KHCN Quốc gia lần thứ XI về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR); Hà Nội, ngày 09-10/8/2018
DOI: 10.15625/vap.2018.0002

AN EFFICIENT DEEP LEARNING METHOD FOR CUSTOMER BEHAVIOUR PREDICTION USING MOUSE CLICK EVENTS

Khang Nguyen [1], Anh V. Nguyen[2], Lan N. Vu[3], Nga Mai[4], Binh P. Nguyen[5]

[1] Software Group, IBM Vietnam, VN
[2]Institute of Information Technology, Vietnam Academy of Science and Technology, VN
[3]Institute of Information Technology, Vietnam Academy of Science and Technology, VN
[4]Faculaty of Mathematics and Information Technology, Thang Long University, VN
[5]School of Mathematics and Statistics, Victoria University of Wellington, NZ

khangnt@vn.ibm.com, anhnv@ioit.ac.vn, vnlan@ioit.ac.vn, ngamt@thanglong.edu.vn, phubinh@ieee.org

ABSTRACT: *This paper represents an efficient method of predicting customer purchase behavior based on mouse click event using deep learning. The data stream of mouse click event about customer interactions is fundamental and popular information that are recorded for different purposes, such as to elaborate the customer online purchase behavior. These information is now accelerating on internet with diverse features that are required to be analyzed by a powerful machine learning method... Wide and Deep Learning Network is selected with enhancements for predictive modeling to leverage the best capabilities of both Wide network and Deep network as joint learning neural network. The Wide and Deep neural network outperforms other same types of deep learning models with the ability to learn the behavior from both low and high-order interactions of features, while leveraging the memorization capability of linear models and generalization capability of deep neural network. Experimental results on real life 33 million click dataset shows that the Wide and Deep model performs better predictions with accuracy up to 78.26%, compared to other models such as deep neural network, product-based neural network, and factorization-machine support neural network.*

Keywords: *Deep Learning, Deep and Wide Neural Network, Customer Purchase Behavior, Mouse Click Events.*

I. INTRODUCTION

In the e-commerce industry, it is crucial to make the appropriate marketing decision to maximize the business benefits. The decision making is normally based on customer profile like age, sex, education, income, etc. and their history purchases for each customer segment. However, for small e-commerce businesses and most new e-commerce sites, it is not practical to build a complex data management system with the capability of collecting and managing customer profiles over time, due to constraints of time and resources. In addition, many anonymous users, without any profile information, make only few interactions within one or two sessions on small websites so features are not explicit to analyze and predict their decision. There are recently other types of information, which can be used to analyze and predict customer purchase behavior, are their interactions with businesses through websites such as purchase frequency, complaints, date and time, value of purchased products.... This information is called *customer behavioral features*.

Customer behavior prediction is not simply to derive from each mouse click event, it requires to evaluate the sequential clicks as a session to predict the behavior through the analysis of customer's hidden interactions beyond the these click events. Typically, hidden interactions are very complex to interpret, thus requires powerful modern machine learning to detect the potential patterns for learning, especially in the case of huge datasets with multiple features.

The recommender systems for small ecommerce websites currently use simple approaches in which user information is unavailable, e.g., suggestions based on similarity on the product features, the probability of simultaneous occurrence or the product conversion rate, etc., common methods are often based on linear models, such as *Follow-The-Regularized-Leader* (FTRL) (McMahan et al., 2013). Although bringing along some specific results, these models are not able to capture hidden interactions of features, particularly high-order interactions that did not appear obviously in the training dataset. Alternative approach uses the *Factorization Machine model* (FM) [Rendle, 2010], which evaluates the interaction between a pair of features by calculating their vector scalar product. Theoretically, the FM model can simulate high-order interactions, however, in practical applications, the model is only applied to 2nd order interactions due to high complexity

With the ability to effectively learn, capture and simulate features, deep neural networks are increasingly used in the analysis of hidden interactions among features. Several recent studies have implemented *Convolutional Neural Network* (CNN) and *Recurrent Neural Network* (RNN) to analyze and predict customer click events [Liu et al., 2015; Zhang et al., 2014]. However, the accuracy of the CNN depends much on interactions between contiguous features, while the RNN is only used to process sequential data. [Zhang et al., 2016] investigated and proposed the use of *Factorization-machine supported Neural Network* (FNN), in which the author conducts training in an FM network and then uses the output to feed into deep neural networks; therefore, the output of this method is limited by capability of the FM model. [Qu et al., 2016] considered interactions between features by adding a product layer between the

embedding layer and the first hidden layer of deep neural networks, aiming to develop a *Product-based Neural Network* (PNN). However, in the study by [Cheng et al., 2016] indicates that PNN, FNN, and many other deep learning networks limit their capacity of remembering low-order interactions; while considering both high and low-order interactions will increase the accuracy of the model.

For the above-mentioned reasons, the study presents an approach to predict customer purchase behavior based on behavioral features from the whole session in the clickstream data, which is applied for the context where customer profile is not accurate and incomplete or not available. By processing and analyzing the customer's sequential click events in the whole session, the hidden patterns associated with the customer purchase behavior will be identified and extracted as input for the deep learning model, which predicts the customer purchase decision. Deep and Wide network [Cheng et al., 2016] with the ability to learn both low- and high-order interactions of features will be used for predictive modeling.

II. THE PROPOSED METHOD

II.1 Problem Statement

The study assumes that each click session is about an independent user. The customer purchase behavior is hidden within the characteristics of each session (number of times, frequency, click time, etc.). The goal of this paper is to predict customer purchase behavior based on their click events in a session on a specific e-commerce site, with two answers:

(1) predict whether the user make a purchase in this session;
(2) if any, forecast which items will be purchased.

For the above purpose, the problem is given as a binary classification, in which the classifier returns the probability of the "purchase" event.

Assume that the training data set consists of n samples (χ, y), whereas χ is the sequence of data recorded by m features relating to customer and product, and $y \in \{0, 1\}$ denotes respectively the purchase behavior of the customer ($y = 1$ if the customer buys the product, $y = 0$ otherwise).

Features of χ can be categories (e.g., product category...) or contiguous (e.g. number of clicks ...). Each category feature is represented by a coefficient vector, and contiguous features are represented by their own values.

Thus, each sample in the dataset is represented by a point (x, y) where $x = [x_1, x_2, ..., x_j, ..., x_m]$, is a m–dimension vector, and x_j is the vector that represents the *j-th* field in χ. Usually, x has multi-dimensions and very sparse density. The main objective of this paper is to build a predictive model $y \approx \hat{y} = f(x)$ to evaluate the probability of a customer purchase decision for specific products.

II.2 Feature Extraction

We assume that the mouse click events contains some very basic data that any system can generate in production, with this assumption, we generalize the raw data as

- The *click data* consists of product and its category, the click time of that session
- The *buy data* contains of the product that is purchased with the quantity and the price, linked to the session above. A product is clicked but not purchased then it is not recorded in the buy data

The raw data is being engineered and extracted to features that are predictably impacted to the customer purchase behavior. After the intensive engineering stage, we propose to use the factor *"Purchase/Click"* as the core to analyze purchase possibility over other factors, also considered as extracted features. Below are some key factor analyses that provides the valuable insights for the purchase possibility

- *Clicks per Session*: The more clicks are occurred the higher purchase possibility is (Figure 1a).
- *The most clicked products*: The more clicks on the product the higher possibility that product will be purchased. However, the purchase ratio seems not consistent and varies on different products (Figure 1b)
- *Session Hour*: The purchase possibility also depends on the hour of the day, the data shows that the possibility is quite high from 3pm to 7pm, and rather low around 11pm to 4am next day (Figure 1c)
- *Session Date*: Data shows that the date of the week could reflect the purchase possibility with the peak on Saturday and lowest on Tuesday (Figure 1d).
- *Session Duration*: This seems to be the critical extracted feature to predict the purchase per click ratio. The duration of 15 to 20 minutes shows the highest possibility and drops it the session takes longer (Figure 1e).

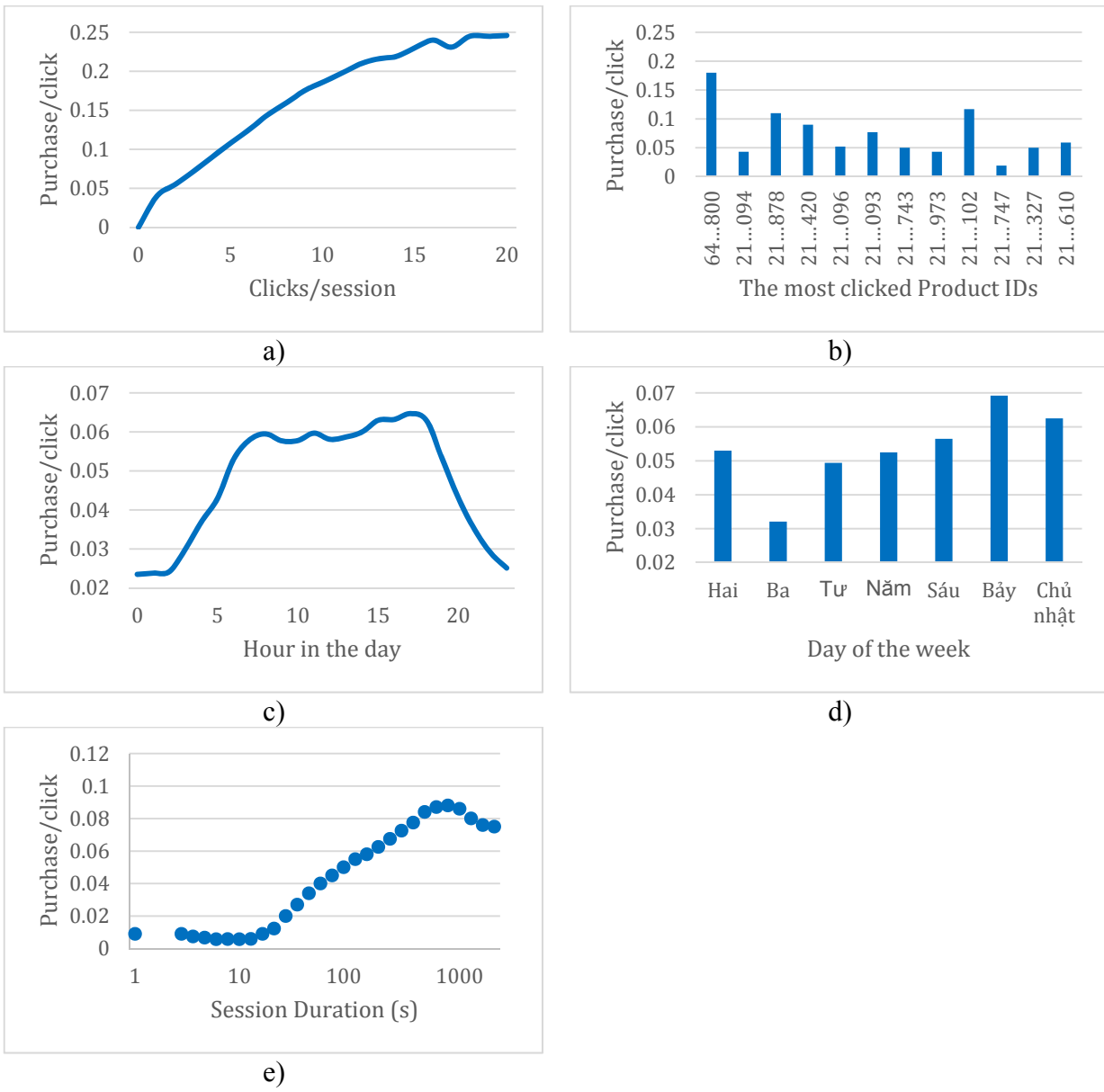

Figure 1. The purchase/click trends of extracted features

In addition, the analysis shows that the purchase possibility could be predicted by other features such as the *product displayed duration*, *number of clicked products in the session*, *the previous and the next product that are being viewed*… Although these features are not strongly impacted to the purchase decision they are still used to train the model with the purpose to improve the prediction accuracy

Based on the above feature engineering and extraction process, there are 26 extracted features as the input layer for the neural network of the predictive model, they are grouped into different classes: item features, session features, and time features.

Item Features (2 features)		
Product ID	Categorical feature	The ID of the product
Cat ID	Categorical feature	The ID of the category that the product belongs to
Session Features *(11 features)*		
The First Product	Categorical feature	The first product ID in the session
The Pre Product	Categorical feature	The previous product ID in the session
Session Duration	Contiguous feature	The duration of the session
Current Duration	Contiguous feature	The duration from the session starts
#Click per Session	Contiguous feature	Number of clicks per session
#Product per Session	Contiguous feature	Number of products per session
#Click So Far	Contiguous feature	Number of clicks so far in the session
#Product So Far	Contiguous feature	Number of products clicked so far in the session

#View of Product	Contiguous feature	Numbers of views for the product in the session
#Product of the same Cat	Contiguous feature	Numbers of products in the same category
#Cat	Contiguous feature	Number of categories containing the selected product in the session
Time Features (9 features)		
Session Start	Categorical feature	Session start timestamp (3 features: hour, minute, second)
The first time that product is clicked	Categorical feature	The timestamp when the product is clicked at first (3 features: hour, minute, second)
Current Time	Categorical feature	The current timestamp (3 features: hour, minute, second)
Boolean Features (4 features)		
The most clicked product	Boolean feature	The most clicked product in the session
The most viewed product	Boolean feature	The most viewed product in the session
The first clicked product	Boolean feature	The first clicked product in the session
The most viewed category	Boolean feature	The most viewed category in the session

Given the above features, the research also analyses the hidden interactions among them using the cross-product transformation. With this approach, all categorical features are converted into vectors with the same dimension before calculating the cross-product by pairs, resulting the new vector with the corresponding dimension that can interpret the combined interactions of each individual feature to the prediction. The pair is selected using the trial and failure technique, each cross-product vector is fed into the model for training and will be selected if it increases the model accuracy. This trial loop results the following pairs

- The current product ID x the first product ID in the session
- The current product ID x the previous product ID in the session
- The current product ID x the category ID of that product in the session
- The longest viewed product ID x the viewing duration in the session.

Finally, there are 26 extracted features and 4 cross-product features, used to construct the predictive model

II.3 Wide and Deep Learning Model

In this paper, the predictive model is expected to learn and capture both low- and high- order interactions of features. Thus, Deep and Wide Network [Cheng et al., 2016] is chosen with enhancements to solve the problem.

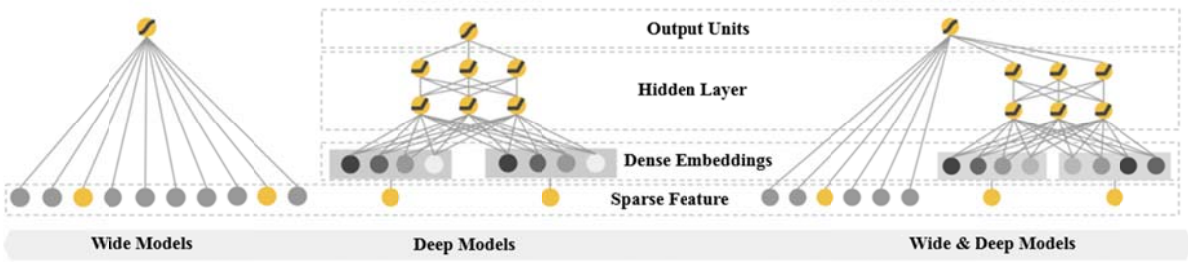

Figure 2. The Wide, Deep, and the joint models.

Wide and Deep learning model is a joint model between Wide model (linear) and Deep model (neural network), including two following components:

The Wide Component

This component is a generalized linear model as below:

$$y = W^T x + b$$

In which y is the prediction, $x = [x_1, x_2, ..., x_m]$ is a vector of m features, $W = [w_1, w_2, ..., w_m]$ are parameters of the model and b is the bias. The feature set includes raw input features and transformed features. The transformation used is the *cross-product transformation*:

$$\varphi_k(x) = \prod_{i=1}^{d} x_i^{c_{ki}} \qquad c_{ki} \in \{0,1\}$$

where c_{ki} is a boolean variable that is 1 if the i-th feature is a part of the k-th transformation φ_k, and 0 otherwise. This transformation captures the interactions between the features, and adds nonlinearity to the generalized linear

model. The wide component can effectively memorize sparse interactions between the features, but it lacks the capability to generalize latent feature combinations which will be handled in the *deep component*.

The Deep Component

This component is a feed-forward neural network combined with *embeddings* layer in the first layer. Compared with deep neural networks that have the input of image or audio which are continuous sequences of real numbers, the raw input in the click-prediction problem contains multi features and sparse. Hence, embedding layers are used to transform these features into vectors with less dimensions, called *dense embedding vectors*.

To go into detail, outputs of embedding layer are $a^{(0)} = [e_1, e_2, ..., e_i, ..., e_m]$, where e_i is an embedding vector of i-th feature, and m is the number of features. These embedding vectors, together with contiguous features, are then passed into the hidden layers of the neural network:

$$a^{(l+1)} = \sigma(W^{(l)}a^{(l)} + b^{(l)})$$

where σ is the activation function, e.g. rectified linear units ReLUs as $f(x) = x^+ = \max(0, x)$; $W^{(l)}$, $a^{(l)}$ and $b^{(l)}$ are the model weights, activations and bias at l-th layer.

The learning process of the joint network happens simultaneously at both components to produce the final result of the joint predicative model:

$$\hat{y} = Sigmoid(y_R + y_S) = \frac{1}{1 + e^{-(y_R + y_S)}}$$

where $\hat{y} \in (0, 1)$ is the predicted value of the probability of purchase decision, y_R is the output of the wide component while y_S is the output of the deep component.

With this approach, the Deep and Wide network is capable of learning both low and high-order interactions of features, while leveraging the memorization capability of linear models and generalization capability of deep neural networks. This network model is especially optimized for large input dataset with huge number of features.

II.4 Comparison with other Models

Given the results of successfully applying deep learning into various areas, several models have been developed to solve the click event prediction requirement. This section compares the Deep and Wide model versus other different models to provide an overview on deep learning methods in predicting customer purchase behavior based on the mouse click event.

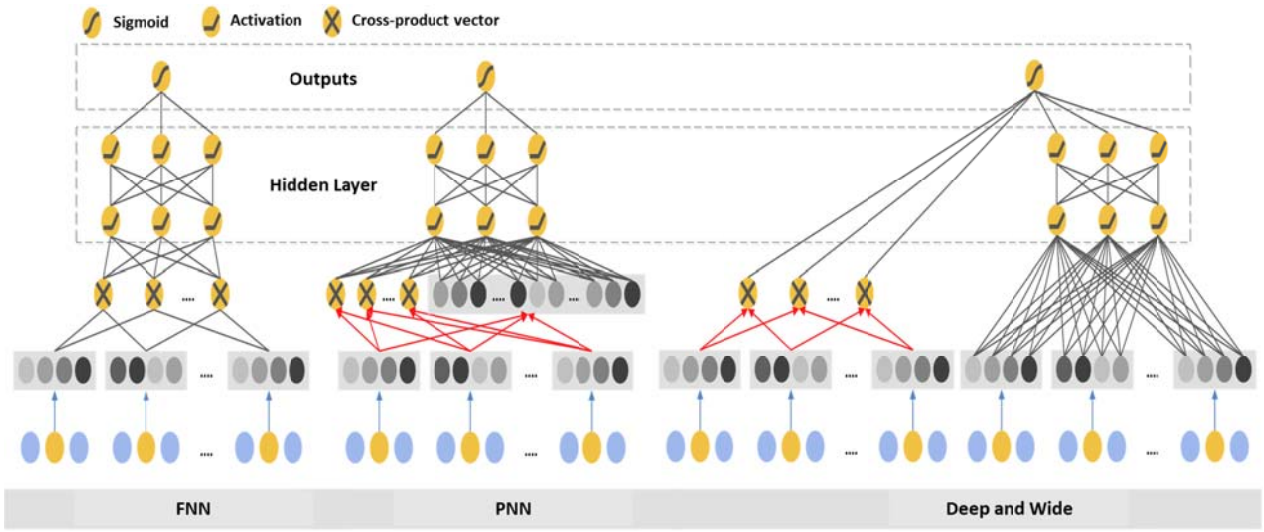

Figure 3. Wide and Deep model versus Others

FNN

FNN is a feed-forward neural network initialized by the output of factorization machine FM network [Zhang et al., 2016]. The training of the FM model prior to the application of the deep network has led to two major limitations of this method: (1) the parameters of the embedded layer are mainly influenced by the FM network; and (2) the performance of the network is reduced due to the loss generated by FM preprocessing before being fed into the DNN network. In addition, FNN can only learn the higher-order interactions of features. In contrast, Deep and Wide network does not require pre-training and are capable of learning both low-order and high-order interaction of features.

PNN

The PNN is a feed-forward neural network that adds a vectorized transformation layer prior to the first hidden layer, enabling the network to capture the high-order interactions of features [Qu et al., 2016]. Respectively to the vectorized transformation, PNN is divided into 3 variants: IPNN (*Inner Product-based Neural Network*), OPNN (*Outer Product-based Neural Network*) and PNN *, where IPNN is based on scalar product, OPNN uses the cross product of the vectors, and PNN * combines both the scalar product and the cross product. Like FNN, all variants of PNN does not capture the lower-order interactions of features.

Table 1. A comparison table of deep learning models used in click event prediction

NN Type	Pre-training not required	High-order interactions	Low-order interactions
FNN	×	✓	×
PNN	✓	✓	×
Wide and Deep	✓	✓	✓

III. EXPERIMENT RESULTS

III.1 Dataset

We use in this paper the dataset created by YOOCHOOSE GmbH to support participants in the RecSys Challenge 2015 (http://2015.recsyschallenge.com)

This dataset provides a collection of sequences of click events; click sessions from a large Europe e-commerce business, where each session encapsulates the click events that the user performed in the session. The data was collected for 6 months in 2014 from April to September. To protect end user privacy, all identities have been masked in the dataset. The training data is comprised of two different files:

Click dataset (yoochoose-clicks.dat): represents the clicks of the users over the items. Each record/line in the file has the following features/format: (1) *Session ID* – the id of the session, in one session there are one or many clicks. (2) *Timestamp* - the time when the click occurred. (3) *Item ID* - the unique identifier of the item that has been clicked. (4) *Category* – the context of the click.

Buy dataset (yoochoose-buys.dat): represents the buy events of the users over the items. Each record/line in the file has the following features: (1) *Session ID* - the id of the session. (2) *Timestamp* - the time when the purchase occurred. (3) *Item ID* – the unique identifier of item that has been bought. (4) *Price* – the price of the item, (5) *Quantity* – the quantity in this buying.

Each Session ID in *yoochoose-buys.dat* always appears in *yoochoose-clicks.dat* - the data with same Session ID will associate and represent the click event of a specific customer during a session. The Session Duration can be very short (minutes) or very long (several hours), which may include one or many clicks or buying events, depending on customer interactions.

Table 2. Dataset Statistics

	yoochoose-clicks.dat	yoochoose-buys.dat
Number of records	33,003,944	1,150,753
Number of items	52,739	19,949
Number of sessions	9,249,729	509,696

The entire dataset is randomly divided into 3 sets of data:

60% training dataset	used to train the predictive model
20% validation dataset	used to evaluate efficiency and select optimal network structure
20% testing dataset	used to compare the candidate models

III.2 Optimize the Network Structure for Deep Component

This paper assesses the impact of different network structures for the deep component on the predicting model using *trial and error method*. The optimized factors of structure evaluation are: (1) the number of hidden layers; (2) deep neural network shape; and (3) the number of neurons in the hidden layer.

To evaluate the most effective model, this study makes an analysis and comparison between candidate models based on the three measures *Accuracy*, *AUC* (Area Under ROC), and *Logloss* (cross entropy) obtained from the model using the validation dataset

Number of hidden layers

Assume network parameters remain unchanged, increasing in the number of hidden layers will increase the network complexity. As shown in Figure 4, increasing the number of hidden layers from 1 to 3 will improve the learning capability of the model. However, as the number of hidden layers increases, the model is less efficient because the complex network often leads to "over-fitting".

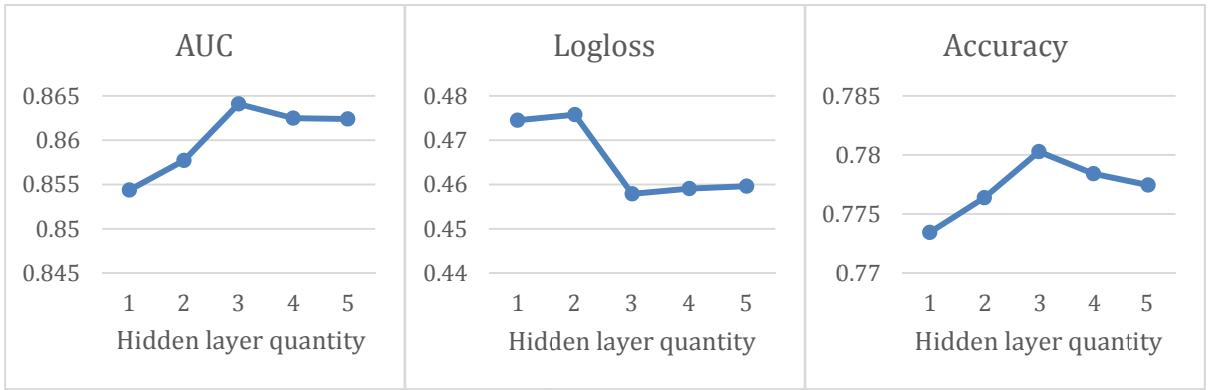

Figure 4. Comparison on candidate models when changing the hidden layer quantity

Based on the results, this study will use a 3 - hidden layer deep learning network.

Deep neural network shape

This research assesses the candidate models with different deep learning network shapes: constant network, increasing network, decreasing network and diamond network. While the network shape is evaluated, the total number of neurons and the number of hidden layers is kept unchanged

For the implementation, we assume and allocate total 1200 hidden neurons for the 3 hidden layers, the number of neurons in the hidden layers of the four networks is: *constant network*: 400-400-400, *increasing network*: 300-400-500, *decreasing network*: 500-400-300, *diamond network*: 350-500-350 respectively.

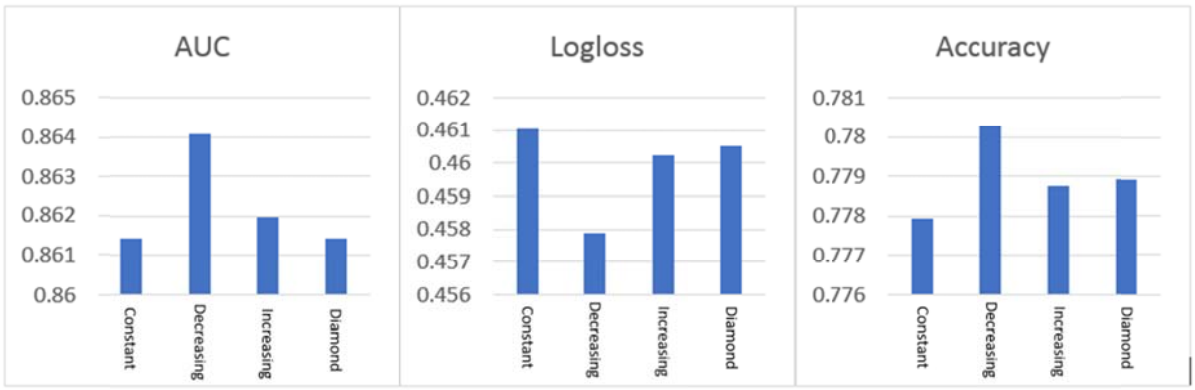

Figure 5. Comparison on candidate models when changing the shape of neural networks

As shown in this feature, the *decreasing network* outperforms the rest, so it will be used for the evaluation.

Quantity of neurons in hidden layers

Similar to the number of hidden layers, increasing in the number of neurons will enhance the complexity of the network. As shown in Figure 6, when the average number of neurons in each hidden layer increases from 200 to 400, the learning ability of the model is consequently increased. However, as the number of neurons increases from 400 to 700, the efficiency of the model tends to decrease. Thus, it can be inferred that increasing the number of neurons does not necessarily increase the efficiency of the model, as the exceeding number of neurons can lead to over-fitting network.

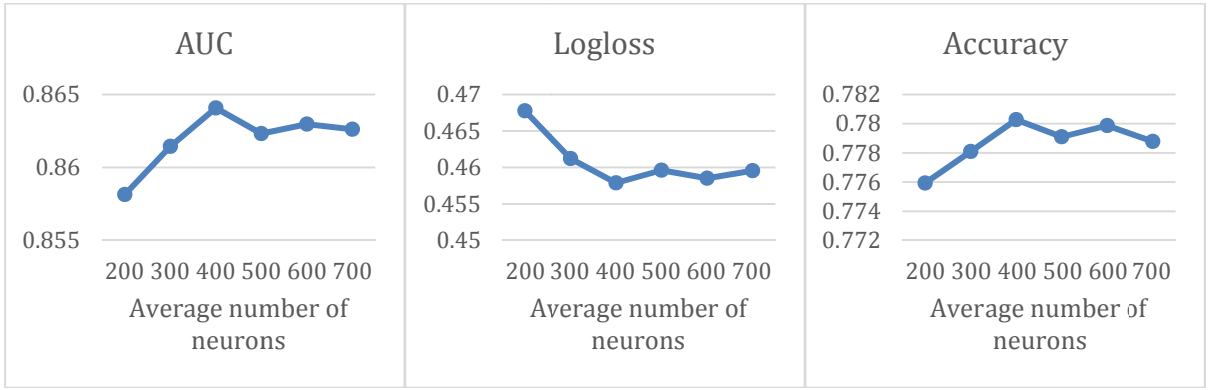

Figure 6. Comparison on candidate models when changing the average number of neurons in each hidden layer

From the experimental results, this research will use a deep learning network with 3 hidden layers, which respectively have 500-400-300 neurons for prediction.

III.3 The Proposed Predicting Model

This paper refers to [Cheng et al., 2016] to build predicting model with enhancements in feature extraction and selection for wide and deep components, and the structure proposal for deep neuron network component. The detail proposal is as below

The Wide component: composed of 2 feed-forward layers, in which output layer contains 1 neuron and input layer has the number of neurons: $N = N_T + N_C$, where N_T is the number of categorical fields, N_C is the pairs of cross interactions among those categorical fields.

The Deep component: composed of 6 feed-forward layers, in which 1 output layer with 1 neuron only. 1 input layer having the number of neurons equal to the number of features, 1 embedding layer, 3 hidden layers with the number of neurons 400 – 400 – 400 respectively. The hidden neurons are using the activation function *ReLU*, and the output neurons are using the activation function *sigmoid*

Other parameters are: *Learning algorithm: Adagrad, learning rate: 0.1, batch size: 64.*

III.4 Results and Analyses

Based on the neural network selection presented above, this study run a prediction using the dataset. The model structure used is shown in Figure 7.

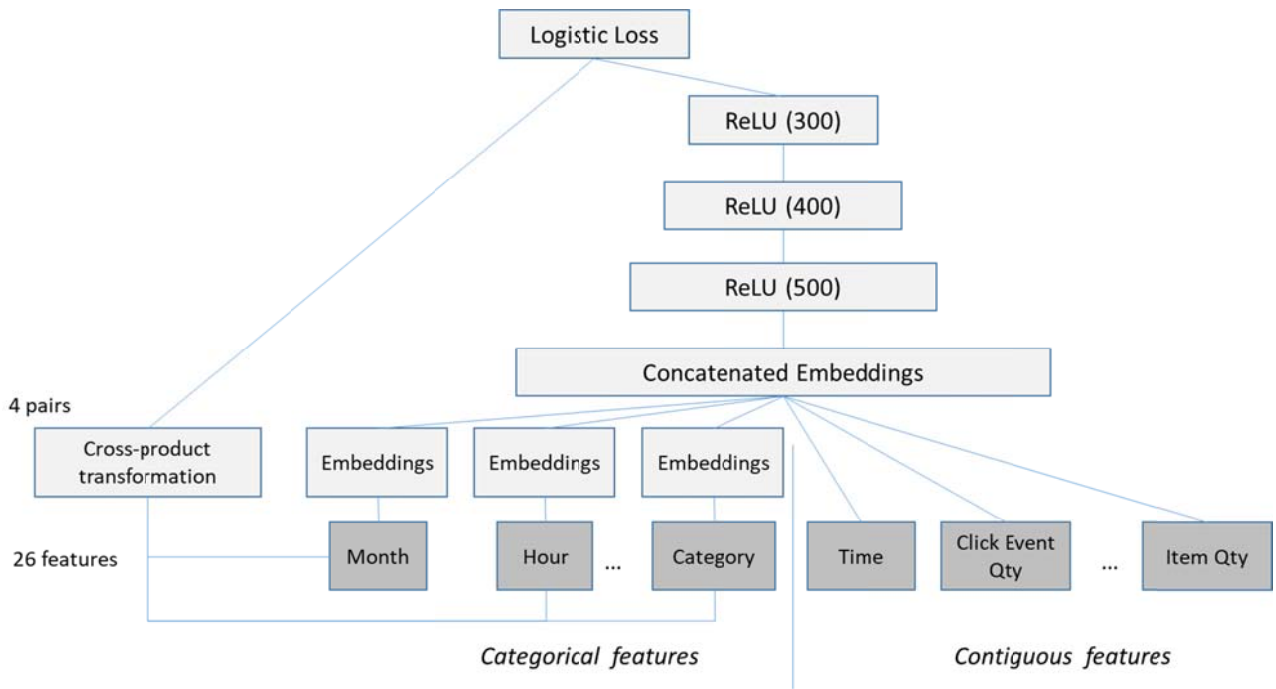

Figure 7. Wide and Deep model structure for click event prediction

This paper compares the effectiveness of the Deep and Wide model with different models: Linear Model (LR), original Deep Neural Network (DNN), Product-based Neural network (PNN) model and Factorization machine support Neural Network (FNN) model.

The results on the testing dataset are shown in Table 3

Table 3. Comparison on model efficiency in predicting click event

Model	Accuracy	AUC	Logloss
LR	69.67%	76.04%	58.42%
DNN	77.89%	85.21%	61.45%
PNN	78.08%	85.96%	53.32%
FNN	78.14%	86.20%	50.61%
Wide and Deep	**78.26%**	**86.70%**	**45.19%**

In conclusion, the Deep and Wide model produces better results than the other models, the final accuracy is 78.26%, 0.12% higher than the other models., with AUC exceeding 0.5%

IV. CONCLUSION

In this paper, the Wide and Deep learning network is proposed and enhanced to solve the problem of predicting the customer purchase behavior based on mouse click event. The Wide and Deep learning model outperforms similar deep learning models for the following reasons: (1) no pre-training required, (2) ability to learn both low- and high-order interactions of features, (3) integrating the memorization capability of the linear model and the generalization ability of deep neural network into the joint model. The experiment result on real life dataset proves that the Wide and Deep learning model achieves better results than other similar models in predicting customer click events.

The proposed method shows that the model is able to predict customer behavior with high accuracy based on the analysis of customer click events in a session, without considering user identity information. Therefore, this method can be applied to any businesses that are unable to collect customer information accurately and completely.

V. ACKNOWLEDGEMENT

The authors gratefully acknowledge the support for this paper from IoIT (VAST) research project CS18.12.

VI. REFERENCES

[1] [Chen et al., 2016] Junxuan Chen, Baigui Sun, Hao Li, Hongtao Lu, and Xian-Sheng Hua. Deep CTR prediction in display advertising. In MM, 2016.

[2] [Cheng et al., 2016] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & deep learning for recommender systems. CoRR, abs/1606.07792, 2016.

[3] [Juan et al., 2016] Yu-Chin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. Field-aware factorization machines for CTR prediction. In RecSys, pages 43-50, 2016.

[4] [Larochelle et al., 2009] Hugo Larochelle, Yoshua Bengio, J´erˆome Louradour, and Pascal Lamblin. Exploring strategies for training deep neural networks. JMLR, 10:1-40, 2009.

[5] [Liu et al., 2015] Qiang Liu, Feng Yu, Shu Wu, and Liang Wang. A convolutional click prediction model. In CIKM, 2015.

[6] [McMahan et al., 2013] H. Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Mar Hrafnkelsson, Tom Boulos, and Jeremy Kubica. Ad click prediction: a view from the trenches. In KDD, 2013.

[7] [Qu et al., 2016] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. Product-based neural networks for user response prediction. CoRR, abs/1611.00144, 2016.

[8] [Rendle and Schmidt-Thieme, 2010] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In WSDM, pages 81–90, 2010.

[9] [Rendle, 2010] Steffen Rendle. Factorization machines. In ICDM, 2010.

[10] [Zhang et al., 2014] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie Yan Liu. Sequential click prediction for sponsored search with recurrent neural networks. In AAAI, 2014.

[11] [Zhang et al., 2016] Weinan Zhang, Tianming Du, and Jun Wang. Deep learning over multi-field categorical data - A case study on user response prediction. In ECIR, 2016.

ỨNG DỤNG MẠNG HỌC SÂU VÀ RỘNG TRONG DỰ ĐOÁN HÀNH VI MUA CỦA KHÁCH HÀNG DỰA TRÊN CƠ SỞ DỮ LIỆU NHẤP CHUỘT

Nguyễn Tuấn Khang, Nguyễn Việt Anh, Vũ Như Lân, Mai Thúy Nga, Nguyễn Phú Bình

TÓM TẮT: *Trong thương mại điện tử, dữ liệu nhấp chuột của khách hàng là loại dữ liệu thông dụng và cơ bản nhất được ghi nhận, trong đó có chứa nhiều thông tin ẩn liên quan tới hành vi mua sắm của khách hàng. Thông thường, các thông tin này cùng với tương tác ẩn của chúng rất phức tạp, và chỉ có thể được phát hiện nhờ kỹ thuật học máy hiện đại. Với khả năng học và nắm bắt và mô phỏng thuộc tính hiệu quả, mạng nơ-ron sâu ngày càng được sử dụng phổ biến trong phân tích đánh giá tương tác ẩn của các trường thuộc tính. Bài báo trình bày phương pháp dự báo hành vi mua của khách hàng dựa trên cơ sở dữ liệu nhấp chuột sử dụng mạng học Sâu và Rộng để xây dựng mô hình dự báo. So với các mô hình học sâu cùng loại, mạng Sâu và Rộng vượt trội hơn do có khả năng học được tương tác bậc thấp lẫn bậc cao của các trường thuộc tính, đồng thời tận dụng được khả năng ghi nhớ của mô hình tuyến tính và khả năng tổng quát hóa của mạng nơ-ron sâu vào trong cùng một mô hình. Kết quả thử nghiệm trên dữ liệu chuỗi nhấp chuột thực tế với hơn 33 triệu phiên làm việc cho thấy, so với các mô hình cùng loại khác như mạng nơ-ron sâu (DNN), mạng nơ-ron tích chập (PNN) hay mạng nơ-ron phân tích nhân tử (FNN), mô hình Sâu và Rộng cho dự báo tốt hơn với độ chính xác lên đến 78.26%.*