Attention Convolutional Neural Network for Advertiser-level Click-through Rate Forecasting

Hongchang Gao^{†*}, Deguang Kong^{*}, Miao Lu^{*}, Xiao Bai^{*} and Jian Yang^{*}

^{*}Yahoo Research, Oath, 701 1st Avenue, Sunnyvale, CA

[†]University of Pittsburgh, P.O. Box 1212, Pittsburgh, PA

{hongchanggao, doogkong}@gmail.com, {mlu, xbai, jianyang}@oath.com

ABSTRACT

Click-through rate (CTR) is a critical problem in online advertising. Most existing researches only focus on the user-level CTR prediction. However, advertiser-level CTR forecasting also plays a very important role because advertisers typically decide how much they would like to bid for advertisements to achieve the maximum clicks given their budget based on CTR forecasting. Over-forecasting will make the advertiser to pay more than necessary but get less return on investment (ROI). Under-forecasting will make the advertiser to spend less money on campaigns but they cannot achieve the desired ROI goals.

In this paper, we focus on the advertiser-level CTR forecasting and formulate it as a time series forecasting problem based on the historical CTR record. This is a very challenging problem due to the heavy fluctuation and highly non-linearity of time series. Furthermore, advertisers usually provide useful contextual information for their campaigns, such as text descriptions, targeting locations and devices, which has high correlation with CTR but has not yet been used for CTR forecasting. Thus, we propose a novel context-aware attention convolutional neural network (CACNN), which can capture the high non-linearity and local information of the time series, as well as the underlying correlation between the time series of CTR and the contextual information. To the best of our knowledge, this is the first work employing convolutional neural network and incorporating heterogeneous information to perform CTR forecasting at advertiser level. We implement the system on Yahoo TensorFlowOnSpark platform which enables distributed deep learning on a cluster of GPU and CPU servers, and achieves faster learning speed and data access on HDFS when available. The effectiveness of CACNN model has been demonstrated in real-world Yahoo advertising dataset, and therefore deployed in production with daily rolling of the model.

CCS CONCEPTS

• Computing methodologies → Neural networks;

KEYWORDS

Advertising; Forecasting; Click-Through Rate; Time Series; Attention; Convolutional Neural Network; Sponsored Search

This paper is published under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW 2018, April 23-27, 2018, Lyon, France

© 2018 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC BY 4.0 License.

ACM ISBN 978-1-4503-5639-8/18/04.

<https://doi.org/10.1145/3178876.3186184>

ACM Reference Format:

Hongchang Gao[†], Deguang Kong^{*}, Miao Lu^{*}, Xiao Bai^{*} and Jian Yang^{*}. 2018. Attention Convolutional Neural Network for Advertiser-level Click-through Rate Forecasting. In *WWW 2018: The 2018 Web Conference, April 23-27, 2018, Lyon, France*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3178876.3186184>

1 INTRODUCTION

In ads marketplace, click-through rate (CTR) is the ratio of clicks on a specific link to the number of times an advertisement was served (i.e., impression), which is widely used in measuring the success of online advertising campaigns. For advertisers, forecasting CTR is very important to estimate the performance of ads campaigns for different targeting in order to achieve the maximum return on investment (ROI). On one hand, over-forecasting of CTR will result in over-forecasting of clicks given the same bid price, and actually advertisers tend to bid less and therefore lose the opportunity. On the other hand, under-forecasting of CTR will make the advertiser to pay more than necessary but get less ROI. For these reasons, there is an increasing demand for estimating advertiser-level CTR accurately. With accurate forecasting of CTR, advertisers can get informed decisions to plan, adjust and optimize their campaign performance.

There exist many works of predicting CTR, most of which, however, focus on user-level CTR forecasting using rich user-level features (such as user clicks, ads format, etc) to optimize Ad revenue per thousand impressions (RPM) while improving users' satisfaction. Different from these works, this paper presents an advertiser-level CTR forecasting to guide advertisers to achieve their desired traffic goals by keeping the organic traffic growth. Considering the usefulness of forecasting CTR for advertisers, it becomes crucial to accurately forecast them. However, we need to address several technical challenges that are discussed below.

- C1: How to accurately forecast CTR since CTR can be different even on different days with similar targeting and advertiser profiles? What is the cause of variability? Is it random noise or due to the seasonality and trend?
- C2: How to accurately forecast CTR given the advertisers' targeting information? Given the correlations between CTR and advertisers' targeting, they may carry useful information to estimate CTR.

⁰The work was done during an internship of the first author at Yahoo Research. The correspondence should be addressed to doogkong@gmail.com. Thanks for the support of machine learning infrastructure team (Lee Yang and Andy Feng) who helps us deploy the experiments and production code using TensorflowOnSpark infrastructure.

- C3: How to combine both historical CTR information and advertisers' targeting information for CTR forecasting? Given these heterogeneous and orthogonal information, the question is how to forecast CTR to capture their connections.

However, to the best of our knowledge, existing works have not effectively addressed these challenges mentioned above. Inspired by the huge success of deep neural networks, we propose a novel context-aware attention convolutional neural network (CACNN) to model CTR by capturing their inherit property such as temporal¹ and targeting features. Extensive experiments performed on CTR forecasting tasks demonstrate that the proposed method consistently outperforms the baselines. To summarize, this paper makes the following contributions.

- To address the challenge C1, we propose CNN based approach to automatically learn kernel weights by performing convolution operations between the temporal CTR and weights in the kernels, such that the temporal features like seasonality and trend can be captured by these feature maps in CNN.
- To address the challenge C2, this paper develops a well-designed multi-layer perceptron (MLP) model to capture the underlying useful information from targeting features to help CTR forecasting.
- To address the challenge C3, the context-aware attention CNN is proposed to combine the CNN module and the MLP module such that it can capture the correlation between the historical CTR and the contextual information for better CTR forecasting.
- With the combination of the above techniques, our proposed method has shown the constantly better performance on the real-world CTR forecasting.

2 RELATED WORK

In this work, we propose to leverage contextual information to make advertiser-level click-through-rate forecasting. We review in this section two domains closely related to this work, i.e., CTR prediction and time series forecasting, and discuss how our work is different from them.

2.1 CTR prediction

Predicting click-through-rate is a critical task for online advertising. There have been a number of successful CTR prediction models for sponsored search advertising [7, 18, 27], contextual advertising [19, 21], display advertising [3, 24], and native advertising [9, 17]. A common objective of these models is to accurately predict the probability of an ad being clicked once it is shown to end users. Differently, our work aims at predicting click-through-rate at advertiser level, i.e., predicting the fraction of ads of the same advertiser being clicked during a given period of time. To the best of our knowledge, there has not been any published research on advertiser-level CTR prediction, so that ad-level CTR prediction is the most relevant to our work.

¹In the paper, CTR historical data is also called CTR time series data. Temporal features are learned by training attention CNN given CTR history input. Temporal feature refers to the learned feature maps using CTR history data.

Ad-level CTR prediction is typically achieved by feeding a machine learning model with well-designed features [9, 18]. There have been a lot of efforts on building features, such as textual features [20], click features [7], contextual features [18], and psychology features [22]. He et al. [9] pointed out that among the different kinds of features, historical features like the click-through-rate of an ad in the last week or the average click-through-rate of a user, provide considerably more explanatory power than contextual features for predicting CTR. Therefore, our work also relies on historical click features to make advertiser-level CTR prediction. The main difference between our work and the existing ones is that the existing approaches usually feed the historical features directly to the model, while our approach considers them as time series and first relies on convolution neural networks to learn their correlations. Edizel et al. [6] proposes to use convolutional neural networks to predict ad-level CTR for sponsored search. In this work, the prediction is made solely based on the characters in the query and ad texts. Zhang et al. [26] relies on user' sequential ad browsing behavior to train a recurrent neural network for CTR prediction. Different from these works that only use one kind of features, our CTR prediction model leverages historical click data and contextual information together to make more accurate predictions. The estimated CTR is used for predicting the performance of advertising campaigns, so as to provide more accurate bid optimizations [14] and recommendations [15], [13].

2.2 Time Series Forecasting

The time series forecasting is the traditional way to generate future values based on historical data. Autoregressive Integrated Moving Average (ARIMA) [2, 5, 10, 25] is the most popular and frequently used time series model. The time series is assumed to be linear and follows a known distribution, e.g., Gaussian distribution. It has a series of variations, such as the Autoregressive (AR) [2, 10], Moving Average (MA) [2, 10] or Exponential Moving Average (EMA), and Autoregressive Moving Average (ARMA) [2, 5, 10]. The popularity of the ARIMA model is mainly due to its flexibility and simplicity to describe the inherent structure of the series. Another widely used approach is the Exponential smoothing method. The simplest of the exponentially smoothing methods is naturally called "simple exponential smoothing" (SES), which is a weighted average of previous observations. "Naive" approach is a special case, which employs the last observed value of the series as the forecast. Since SES is more suitable for forecasting without trend and seasonal pattern, Holt (1957) [11] proposed the linear trend model, including a forecast equation and two smoothing equations for level and trend. To capture seasonality, the Holt-Winters seasonal method [23] introduces one more smoothing equations for the seasonal component. There are two variations of this method that differ in the nature of the seasonal component: additive and multiplicative. The additive method is preferred when the seasonal variations are roughly constant through the series, while the multiplicative method is preferred when the seasonal variations are changing proportionally to the level of the series.

As far as we know, the CTR time series forecasting is very limited. Agarwal et al. [1] proposed a dynamic Gamma-Poisson model

Figure 1: The overview architecture of proposed CACNN model, where Attention CNN model is used to learn features from CTR history, and geo-targeting, device targeting information are processed by an MLP model. These two parts are followed by the concatenation layer and fully connected layers.

to estimate article click-through-rate in the context of content recommendation, but is neither classic time series model nor in the advertising field. Thus, to the best of our knowledge, it is the first paper including time series feature in the advertiser-level CTR forecasting, and the traditional time series models are included as baselines.

3 PROBLEM DEFINITION

The advertiser-level CTR forecasting plays a critical role because, given their budgets, advertisers typically decide how much they would like to bid for their advertisements to achieve the maximum clicks based on CTR forecasting. Given the forecasted CTR, an advertiser can have estimations of how many clicks he may get and how much ads cost he needs to pay, *i.e.*,

$$\begin{aligned} \#Click &= \#Impression \times CTR \\ Spend &= \#Click \times CPC_cost \end{aligned} \quad (1)$$

In practice, #Impression is usually determined by the supply side, and bid price can have impacts on winning rate of the impressions and CPC_cost. Back to our scenario, if an advertiser over-estimates CTR, the number of clicks and spend will be over-estimated, in order to maximize the clicks within the budget, the advertiser tends to bid less than the actual bid. Therefore, the advertiser will lose opportunities to win some impressions. On the other hand, if an advertiser under-estimates CTR, then the number of clicks and spend will be under-estimated, in order to maximize the clicks within budget limit, the advertiser tends to bid more than the actual bid. Due to the high bid price, the advertiser will win more impressions than his expectation. However, his money could be wasted and he probably gets a lower return on investment (ROI).

Most of the existing approaches (especially at user-level) formulate CTR forecasting as a classification problem, *i.e.*, whether the ad is clicked or not. However, at advertiser-level, they are more interested in the overall performance of CTR because many key performance indicators (KPI) (such as Revenue per mille (RPM), Click Yield) depend on accurate numbers of CTR. At advertiser level, CTR estimation can help advertisers to plan their budget

and bid, and make informed decisions for achieving their targeting goals. Therefore, our argument is at advertiser level, CTR prediction (or called forecasting)² is at least as important as (if not more important than) click prediction. In this paper, we formulate it as a context-aware CTR forecasting problem, where the historical CTR over a period is viewed as “*historical information or time series information*”, while the targeting (such as geo-location, device type) is viewed as the “*contextual information*”. Notice that long-term CTR forecasting is even harder and much more infeasible in practice due to the high varieties of advertisers’ behaviors, in this paper, we focus on short-term CTR forecasting. In particular, our goal is to make accurate short-term CTR forecasting for advertisers by reducing the discrepancy between the actual CTR and the predicted CTR. Any long-term CTR forecasting can be done in a roll-up way as long as the short-term CTR forecasting is accurate.

However, it is well known that the time series are usually highly non-linear, and therefore it is tough to forecast. Most of existing methods fail to capture the inherent non-linear structures of time series. For instance, the auto-regressive method employs a linear model to perform forecasting based on the historical record, which is far from satisfactory. Additionally, most existing time series forecasting methods cannot incorporate the related feature information easily. Taking the auto-regressive method as an example, it can only employ the historical information to perform forecasting, having no way to incorporate the extra information (such as the contextual information).

To address problems mentioned above, we propose a novel deep neural network based method to perform CTR forecasting. Specifically, a novel context-aware attention convolutional neural network is proposed to handle the time series, which can capture the highly non-linearity and local information. Additionally, to incorporate the contextual information such as the targeting information, we employ the multi-layer perceptual (MLP) model for each kind of information to learn the correlation between them and CTR. The

²Throughout the paper, CTR prediction and CTR forecasting are used interchangeably. We prefer “CTR forecasting” which makes a difference with standard binary click prediction.

final prediction is made by combining these parts together. The architecture of our proposed approach is shown in Figure 1. As far as we know, this is the *first* work employing convolutional neural network and incorporating heterogeneous information to perform CTR forecasting at advertiser level.

Figure 2: Each advertiser has the historical CTR record, targeting country and device type. The shown native ad is served on Yahoo website.

4 METHODOLOGY

In this section, we will present the detail of the proposed novel *Context-Aware Convolutional Neural Network (CACNN)* method for CTR forecasting at advertiser level.

In ads marketplace, each advertiser can initiate multiple campaigns and each campaign may contain multiple ads. Therefore, “campaign” can be viewed as the container for different ads that target at different people interest, age, gender, geo-location, device type, etc. For example, for an advertiser who is willing to sell CDs, there could be multiple campaigns such as “rock campaign”, “classic campaign”. Within each campaign, there are multiple ads such as “pearl jam ads”, “green day ads”, etc. In this paper, the advertiser-level CTR modeling is actually performed at campaign-level by capturing the fine-grained contextual information and historical information related to this campaign. In the following, when we refer “advertiser-level” CTR, actually it denotes “campaign level” CTR for advertisers.

Given the historical CTR record $X = \{x_1, x_2, \dots, x_n\}$ where $x_i \in \mathcal{R}^T$ is the historical record of the i -th campaign in the past T days and n is the number of campaigns, $Y = \{y_1, y_2, \dots, y_n\}$ where $y_i \in \mathcal{R}$ is the record of the i -th campaign in the $T+1$ day. Note that, here y_i is a continuous value that denotes the clicked probability, not a binary decision (click vs. non-click) as existing methods. In other words, we formulate the advertiser-level CTR forecasting problem as a regression problem, rather than a classification problem as existing methods.

Additionally, in order to make more accurate forecasting, we also incorporate the contextual information, including the targeting country $S = \{s_1, s_2, \dots, s_n\} \in \mathcal{R}^{C \times n}$ where C is the number of countries and the targeting device type $G = \{g_1, g_2, \dots, g_n\} \in \mathcal{R}^{M \times n}$ where M is the number of device types. The reason for incorporating these information is that intuitively, CTR is strongly connected with these contextual information (including targeting country and targeting device). By capturing their correlations, we can have more accurate forecasting.

Based on these terminologies, the task of CTR forecasting is to forecast Y as accurately as possible based on the historical record X and the contextual information S, G . Formally, the task is to learn a map f for all advertisers such that

$$\hat{Y} = f(X, S, G), \quad (2)$$

where the differences between the forecasted $\hat{Y}$ and the ground-truth Y are minimized for each campaign.

As mentioned before, in this paper, we formulate Eq.(2) as a regression problem. In particular, to incorporate the historical CTR information (a.k.a CTR time series), we propose a novel attention convolutional neural network to capture the local information and the highly non-linear structure of the historical record. Furthermore, to incorporate the contextual information, which is the targeting country and device type in this paper, a well-designed multilayer perceptron model is proposed. Consequently, the context-aware convolutional neural network can incorporate both historical information and contextual information for CTR forecasting. Moreover, attention CNN is built by jointly learning a unified model by collecting features (in the form of CTR history and targeting) from all campaigns, instead of one model per campaign, which enables CNN model to capture a variety of relations between input features and output CTR value.

4.1 Attention CNN for Historical CTR Learning

In this paper, the historical CTR record $X \in \mathcal{R}^{n \times T}$ is viewed as the time series. It is well known that such kind of historical record is usually highly non-linear and has a lot of local information, such as seasonality. If these information can be successfully captured, the forecasting will be improved significantly.

Inspired by the huge success of CNN in classification and object detection [16] where CNN can capture local and non-linear information of an image, we employ 1-D CNN for time series to capture the temporal information for CTR forecasting. In particular, a kernel is employed to perform convolutional operation on the time series, followed by a non-linear activation operation, just as shown in the following:

$$\begin{aligned} h_{ij} &= k_j * x_i \\ z_{ij} &= g(h_{ij}) \end{aligned} \quad (3)$$

where $*$ denotes convolutional operation, $x_i \in \mathcal{R}^T$ is the time series of the i -th advertiser, $k_j \in \mathcal{R}^d$ is the j -th convolutional kernel and d is its length, $g(\cdot)$ is an element-wise non-linear activation function, z_{ij} is the learned feature map. An illustration is shown in Figure 3. In Figure 3, there are two kernels whose size is 2 and 3 separately. Thus, there are two feature maps from these two kernels.

Since the convolutional kernel only connects with a local part of x_i , it can capture the local information of the corresponding time series, such as the seasonality. Additionally, by employing different size of kernels, different scales of local information can be easily captured. More specifically, kernels with a small size will capture the short-term local information of the time series while kernels with a large size will capture the long-term ones. After obtaining these different scales of local information, the non-linear activation function following the convolutional layer can help to capture the non-linear structure of the time series. As a result, our method

can capture different scales of local information and non-linear structure from historical CTR record.

Figure 3: The 1-D convolutional operation on time series. There are 2 kernels whose length is 2 and 3 separately. Feature Map 1 is learned by kernel 1, and Feature Map 2 is learned by kernel 2.

Furthermore, in our approach, we use multiple kernels with the same size to capture different aspects of each scale of local information, just as shown in Figure 4. Different branches in Figure 4 correspond to the different size of kernels and there are multiple kernels with the same size in each branch. Thus, there are multiple learned feature maps with the same size, describing different aspects of the same scale of local information. However, how to combine them together is critical for further forecasting. A direct way is to use the mean of these feature maps as their combination, *i.e.*,

$$x_i^{mean} = \frac{1}{m} \sum_{j=1}^m z_{ij} \quad (4)$$

where x_i^{mean} is the obtained representation for the i -th time series, z_{ij} is the j -th feature map of the i -th time series and m is the number of feature maps. However, this approach will lead to the equal contribution from different feature maps. Actually, different feature maps describe different aspects of the time series. Some are more informative while some are less. Thus, it is necessary to discriminate their difference.

To this end, we propose a novel *attention convolutional neural network* that combines different feature maps with different weights as follows:

$$x_i^{att} = \sum_{j=1}^m \alpha_j z_{ij} \quad (5)$$

where x_i^{att} is the obtained representation for the i -th time series, α_j is the attention weight, z_{ij} is the j -th feature map of the i -th time series. An illustration is shown in Figure 5. In Figure 5, there are four feature maps, which are combined together according to Eq. (5) to obtain x_i^{att} .

Mathematically, α_j is defined as follows:

$$\alpha_j = \frac{\exp s(z_{ij}, \theta)}{\sum_{j=1}^m \exp s(z_{ij}, \theta)} \quad (6)$$

where $s(z_{ij}, \theta)$ denotes the auxiliary neural network whose parameters are θ . This auxiliary neural network learns a map from z_{ij} to a scalar such that the informative feature map will be assigned

Figure 4: The architecture of attention convolutional neural network for historical information. The three branches correspond to the different size of kernels. In each branch, there are multiple feature maps with the same size to capture different aspects of the same scale of local features.

with a large weight and vice versa. In this paper, the auxiliary neural network is a one-layer multilayer perceptron model, which is defined as follows:

$$s(z_{ij}, \theta = \{u, v, b\}) = v^T \tanh(u^T z_{ij} + b) \quad (7)$$

The model parameters $\theta = \{u, v, b\}$ will be jointly learned with the CNN model.

Figure 5: The illustration of the attention mechanism on feature maps.

After obtaining the combined feature by attention operation, x_i^{att} is followed by fully connected layers to capture the highly non-linear structure. Finally, these branches are connected by a concatenation layer to get the final representation from the historical CTR record. To summarize, Fig.4 gives an overview of the attention CNN used for historical CTR learning.

4.2 Contextual Information Learning

Contextual information, such as the targeting country and device type, usually provides useful information for CTR forecasting. For example, advertisements targeting to different countries usually behave quite differently due to the effect of the culture and habit of residents. Similarly, advertisements shown on different devices generally have different performance due to different user preferences.

To validate this argument, we sample 10% of advertisements in our dataset and show the median value of CTR in different countries in Figure 6, as well as that in different device types in Figure 7. As a reminder, we hide the values of CTR, the names of countries and devices for confidentiality reasons.

From these two figures, we can clearly see that there is a significant difference among different countries as well as different devices. Thus, it is necessary to capture the underlying correlation between the contextual information and CTR for better forecasting.

Figure 6: The median of CTR in different countries. x-axis: county id, y-axis: median of CTR in the particular country.

Figure 7: The median of CTR in different devices. x-axis: device type; y-axis: median of CTR in the particular device type.

In this paper, we consider two kinds of contextual information, they are the targeting country and device type. However, how to combine them together is challenging, because they are from heterogeneous sources. If we concatenate these two types of information directly, they will disturb each other. One may dominate the other one because these two kinds of information are from different spaces mathematically. Therefore, simply concatenating them is not a good choice.

To address this issue, we propose a deep learning based method as shown in Figure 8. The intuition is, although these two kinds of targeting information are heterogeneous in the low-level layer, they should be homogeneous in the high-level layer since they are depicting the same advertisement. Mathematically, these two types of targeting information are dependent on the same latent subspace. Thus, they can be concatenated in the high-level layer.

In detail, these two kinds of targeting information are categorical features. Thus, we employ the one-hot representation in this paper. Specifically, for the i -th advertisement, it has the country representation as $s_i \in \mathbb{R}^C$ where C is the total number of countries, and s_i has only one non-zero value. Similarly, the i -th advertisement has the device type representation $g_i \in \mathbb{R}^M$ where g_i is also

a one-hot vector and M is the total number of device types. The input is fed into an embedding layer which projects the sparse high-dimensional one-hot feature into a low-dimensional dense one, and a fully connected layer, which learns the high-level feature from the low-level one, follows it. After that, the two branches are connected by a concatenation layer, which is further followed by fully connected layers to get the final representation from the contextual information. To summarize, Fig. 8 gives an overview of the architecture used for contextual information learning.

Figure 8: The architecture of the contextual information modeling.

4.3 Wrap up: A Unified Framework by Combining Context and CTR History

After obtaining the high-level representation from the historical and contextual feature, we combine them together as shown in Figure 1, to make CTR forecasting. The intuition is same as the combination of two kinds of targeting information. In particular, the historical information and the contextual one describe the same advertiser, which means that they depend on the same latent space, such that they can be concatenated in the high-level layer. This gives the proposed context-aware attention convolutional neural network (CACNN). Essentially, it solves an optimization problem as follows:

$$\min_{W, \theta} L(W, \theta) = \sum_{i=1}^n \ell(y_i, \hat{y}_i(x_i, s_i, g_i; W, \theta)) + \lambda(\|W\|_F^2 + \|\theta\|_F^2), \quad (8)$$

where $\hat{y}_i(x_i, s_i, g_i; W, \theta)$ is the predicted CTR value of the i -th advertisement, y_i is the true CTR value, W denotes model parameters of CACNN, θ denotes model parameters of the auxiliary neural network. $\ell(\cdot, \cdot)$ is the loss function, which is ℓ_2 loss in this paper. This model can be trained in an end-to-end way efficiently using stochastic gradient descent algorithm.

5 EXPERIMENT

In this section, extensive experiments are conducted to verify the effectiveness of our proposed model.

Table 1: Descriptions of the CTR dataset.

Data	#Samples	#Features
X	247,777	30
S	247,777	219
G	247,777	4

5.1 Dataset Descriptions

In this paper, we collect ads campaign data from a major advertising platform of an internet company. In particular, the CTR dataset $\{X, S, G\}$ consists of 247,777 campaigns, where each ads campaign includes: campaign ID, CTR record in 30 days (denoted as X), targeting country (denoted as S), and targeting device type (denoted as G). We summarize the details of this dataset in Table 1. As shown in Table 1, there are totally 219 countries and 4 device types.

It is a bit tricky to build the model since we only have 1-month data and the CTR data is highly correlated with the date. In particular, to predict the CTR in the 30-th day, we would like to use the historical data as much as possible and therefore, we use the first 29 days' CTR as the training set. In the training set, in order to utilize the historical data as much as possible for an accurate model, the first 28 days' CTR is viewed as the historical information, i.e., $X_{train} \in \mathbb{R}^{n \times 28}$, the CTR in the 29-th day is viewed as the response: $Y_{train} \in \mathbb{R}^n$. When training the model, 10% of the training set is randomly selected as the validation set. After that, the CTR from the 2-nd day to the 29-th day is set as $X_{test} \in \mathbb{R}^{n \times 28}$, and the CTR in the 30-th day is viewed as the ground-truth for testing data, i.e., $Y_{test} \in \mathbb{R}^n$. The illustration about the construction of the training and testing set is shown in Figure 9.

Figure 9: Split of training vs. testing

Furthermore, since there are missing values at some days in CTR data, we use the mean value of all the other samples in the same day to fill the corresponding missing ones. In addition, the CTR value is generally very small at advertiser-level, saying 0.1%. Therefore, in order to facilitate the modeling, all CTR values are first normalized to $[0, 1]$ before training, and all the normalized values are mapped back to its original scales after prediction. Finally, to get the stationary CTR time series, we difference each time series by one-step as the final input to our model, and then we map them back after prediction. Mathematically, the difference is defined as follows:

$$\hat{x}_{i,j} = x_{i,j+1} - x_{i,j}, \quad (9)$$

where $x_{i,j}$ is the j -th element of the time series $x_i \in \mathbb{R}^T$. Therefore, we have $\hat{x}_i \in \mathbb{R}^{27}$.

In production, the model is updated every day by looking back the history data in the certain period (e.g., 1 month) with augmented new feeds every day. Therefore, the trained model always captures the latest trends, seasonality and patterns implicitly described in the advertising dataset.

5.2 Experiment Settings

Network Architecture. For the historical information (i.e., CTR time series) part, we employ three kinds of kernels. Their length are set as 3, 5, and 7 respectively. The number of each kind of kernels is set as 50. Following the convolutional layer, there are one attention layer and three fully connected layers. We summarize the dimension of each layer in Table 2. For the contextual information (i.e., targeting information) part, the embedding layer is followed by one fully connected layer and one concatenation layer. After that, there are two fully connected layers. We summarize the dimension of each layer in Table 3.

After combining the feature learned from the historical information and contextual information together, there are two fully connected layers whose dimensions are 1×5 and 1×1 respectively. Inspired by ResNet [8], in our design, there is a short connection from the input historical time series to the concatenation layer to avoid the gradient vanishing/exploding problem. Here, the historical time series is directly mapped to a scalar which is concatenated with the other branches such that the dimension of the concatenation layer is 1×5 .

Implementation using Tensorflow on Spark We learn the network parameters in CACNN by solving the optimization problem in an end-to-end way using stochastic gradient descent and implement the system using TensorFlowOnSpark³ infrastructure which provides the benefits of faster learning using TensorFlow and access of HDFS by Spark for easily deployment on GPU and CPU clusters. Fig.10 gives an illustration of infrastructure of training the CACNN model.

Figure 10: The architecture of large scale training framework using TensorflowOnSpark

Baseline Methods To verify the performance of our proposed method, we compare it against both widely used statistical methods and deep learning methods, which are illustrated as follows.

³<https://github.com/yahoo/TensorFlowOnSpark>

Table 2: The dimension of each layer in the model for the historical information.

Input	Conv	Att	FC	FC	FC
1 × 27	50 × 25	1 × 25	1 × 25	1 × 25	1 × 1
	50 × 23	1 × 23	1 × 23	1 × 23	1 × 1
	50 × 21	1 × 21	1 × 21	1 × 21	1 × 1

Table 3: The dimension of each layer in the model for the contextual information.

Input	Emb	FC	Con	FC	FC
1 × 4	1 × 4	1 × 4	1 × 36	1 × 16	1 × 1
1 × 158	1 × 64	1 × 32			

• Exponential Moving Average (EMA) model [12] is a widely used method for time series forecasting, which is defined as follows:

$$\hat{x}_t = \begin{cases} x_1, & t = 1 \\ \alpha x_t + (1 - \alpha)\hat{x}_{t-1}, & t > 1 \end{cases} \quad (10)$$

where x_t is the value at the t -th time step, $\alpha \in (0, 1)$ is the weight, and $\hat{x}_t$ is the predicted value of the t -th time step.

• Holts-Linear (HL) model [4] can handle the time series with a trend. Mathematically, it is defined as follows:

$$\begin{aligned} \hat{x}_{t+\Delta_t} &= l_t + \Delta_t b_t \\ l_t &= \alpha x_t + (1 - \alpha)(l_{t-1} + b_{t-1}) \\ b_t &= \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \end{aligned} \quad (11)$$

which corresponds to the forecasting equation, the level equation and the trend equation respectively. Here, $\alpha \in (0, 1)$ and $\beta \in (0, 1)$ are the smoothing parameters, $\hat{x}_{t+\Delta_t}$ is the predicted value at the $(t + \Delta_t)$ -th time step based on x_t .

• Holts-Winter Additive (HWA) model [4] is the extension of HL model to capture the seasonality, which is defined as follows:

$$\begin{aligned} \hat{x}_{t+\Delta_t} &= l_t + \Delta_t b_t + s_{t-m+(\Delta_t)_m^+} \\ l_t &= \alpha(x_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}) \\ b_t &= \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \\ s_t &= \gamma(x_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m} \end{aligned} \quad (12)$$

where s_t denotes the seasonal information, $(\Delta_t)_m^+$ is the remainder of $\Delta_t - 1$ over m . The other variables are same as HL.

• Holts-Winter Multiplicative (HWM) model [4] is another model which can capture the seasonality. Formally, it is defined as follows:

$$\begin{aligned} \hat{x}_{t+\Delta_t} &= (l_t + \Delta_t b_t)s_{t-m+(\Delta_t)_m^+} \\ l_t &= \alpha(x_t/s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}) \\ b_t &= \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \\ s_t &= \gamma(x_t/(l_{t-1} + b_{t-1})) + (1 - \gamma)s_{t-m} \end{aligned} \quad (13)$$

The variables are same as HWA.

• Autoregressive Integrated Moving Average (ARIMA) model [4] is also widely used for time series forecasting, which combines the autoregressive model and moving average model together.

$$\begin{aligned} \hat{x}_t &= \mu + \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} \\ &\quad - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q}. \end{aligned} \quad (14)$$

where $\phi_p x_{t-p}$ is the autoregressive terms with order p and $\theta_q e_{t-q}$ is the lagged forecasted errors with order q .

• Naive Method basically assumes there exist cyclic patterns (with cycle T) in time series and therefore, the forecasting for $(t + T)$ -th day is slightly fluctuation of the t -th day observation. The fluctuation (denoted as δ_t) is depicted using the average trend within the cycle. Formally,

$$\begin{aligned} \hat{x}_t &= x_{t-T} + \delta_t, \\ \delta_t &= \frac{1}{T} \left(\sum_{i=1}^T x_{t-i} - \sum_{j=T+1}^{2T} x_{t-j} \right). \end{aligned} \quad (15)$$

• Multilayer Perceptron Model (MLP) is a popular feedforward neural network method consisting of fully connected layers. Here we concatenate raw feature of the historical information with the contextual information together and employ a 4-layer MLP model to perform CTR prediction. The dimension of each layer is [250, 64, 32, 16, 1].

• CNN without attention (CNN) only employs the historical information and there is no attention mechanism. Specifically, Eq. (5) is employed to combine different feature maps learned from the historical time series.

• CNN with attention (ACNN) also only employs the historical time series feature, but with attention mechanism as shown in Eq. (6).

• Context-aware CNN (CCNN) uses both the historical time series and contextual targeting information, but Eq. (5) is employed to combine different feature maps learned from the historical time series.

Evaluation Metrics. To evaluate the performance of different methods, we employ three kinds of evaluation metric. We summarize them as follows.

• Absolute Percentage Error (APE) is to measure the predicted error for each CTR, which is defined as $APE = \frac{|y_i^t - \hat{y}_i^t|}{y_i^t}$, where y_i^t is the ground-truth value of the i -th CTR at the t -th day and $\hat{y}_i^t$ is the predicted value at t -th day.

• Absolute Error (AE) is to measure absolute value of predicted error for each CTR, i.e., $AE = |y_i^t - \hat{y}_i^t|$.

• Root-Mean-Square Error (RMSE) is to measure the performance globally, which is defined as: $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i^t - \hat{y}_i^t)^2}$.

For all these three metrics, smaller value denotes better result.

Figure 11: MedianAPE of forecasted CTR using CACNN. x-axis: the number of clicks, y-axis: MedianAPE at different number of clicks.

Table 4: The forecasting performance of different methods. CACNN refers to the proposed model that combines contextual and historical information with attention scenario.

Method	MedianAPE	MedianAE	RMSE
Naive	0.3070	0.1599	0.2996
EMA	0.2557	0.1302	0.2308
HL	0.2343	0.1221	0.2431
HWA	0.2446	0.1279	0.2431
HWM	0.2468	0.1299	0.2505
ARIMA	0.2417	0.1250	0.2639
MLP	0.3257	0.1818	0.2438
CNN	0.1856	0.0956	0.1901
ACNN	0.1837	0.0981	0.1869
CCNN	0.1805	0.0941	0.1893
CACNN	0.1774	0.0916	0.1848

5.3 Results Analysis

In Table 4, we present median of APE, median of AE, and RMSE. Compared with baseline methods, our proposed method CACNN can achieve the best performance. Specifically, we have the following observations from Table 4.

- Although MLP is a deep learning based method, its performance is far from satisfactory. The reason is two-fold. On one hand, it concatenates heterogeneous features directly which causes them to disturb each other. On the other hand, MLP has no local connection as CNN such that it fails to capture the local information.
- As for the CNN-based methods, all of them can achieve significant improvement over the other state-of-the-art methods. In particular, CACNN takes both the historical and contextual information to utilize the correlation between them, achieving a better result than ACNN which only takes the historical information.
- The methods with attention mechanism can achieve the better result in terms of most of metrics compared with those without attention mechanism.

Additionally, Fig. 11 illustrates the Median APE of CTR forecasting at different number of clicks, which means that our proposed method is robust to the number of clicks.

Figure 12: The feature maps learned by three kernels

Convolutional Feature Map To further show the effect of the convolutional operation, we randomly select one time series from the dataset and show its feature maps learned by the three kernels in Figure 12. Note that the feature maps shown in Figure 12 are the

combined feature map such that there is only one feature map for each size. These feature maps are very interesting in capturing the inherent structures “hidden” in time domain observations, which, however, are viewed as important factors for forecasting future behaviors.

(a) Attention weight

(b) First three feature maps

Figure 13: The attention weight and the feature map for the kernel with size being 3.

Attention Weight To show the effect of the attention mechanism, we plot the attention weight of 50 kernels whose length is 3 in Figure 13(a). To verify the correctness of the attention weight, we also present the first three feature maps in Figure 13(b), which correspond to the first three weights in Figure 13(a). From Figure 13(a), we can find that feature map 1 and feature map 3 has similar attention weight. The feature maps in Figure 13(b) further justify this since both of them have very similar patterns and magnitudes. In the meantime, although feature map 2 has very similar patterns to the other two, it has the very different magnitude which is far from the raw time series such that it may disturb the prediction. Thus, feature map 2 is assigned with a small attention weight compared to feature map 1 and 3. All of these observations demonstrate the reasonability of our method.

6 CONCLUSION

In this paper, we propose a novel context-aware convolutional neural network for advertiser-level CTR forecasting, which is able to well capture the correlation between the historical CTR record and the contextual information, obtaining more accurate forecasting. The model is deployed using TensorflowOnSpark framework and is updated daily in practice. We are working towards using generative model (e.g., Generative Adversarial Networks) to further improve the performance of CTR forecasting.

REFERENCES

- [1] Deepak Agarwal, Bee-Chung Chen, and Pradheep Elango. 2009. Spatio-temporal models for estimating click-through rate. In *Proceedings of the 18th international conference on World wide web*. ACM, 21–30.
- [2] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.
- [3] Olivier Chapelle, Eren Manavoglu, and Romer Rosales. 2014. Simple and Scalable Response Prediction for Display Advertising. *ACM Trans. Intell. Syst. Technol.* 5, 4, Article 61 (Dec. 2014), 34 pages. <https://doi.org/10.1145/2532128>
- [4] Chris Chatfield. 2000. *Time-series forecasting*. CRC Press.
- [5] John H Cochrane. 2005. Time series for macroeconomics and finance. *Manuscript, University of Chicago* (2005).
- [6] Bora Edizel, Amin Mantrach, and Xiao Bai. 2017. Deep Character-Level Click-Through Rate Prediction for Sponsored Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. ACM, New York, NY, USA, 305–314. <https://doi.org/10.1145/3077136.3080811>
- [7] Thore Graepel, Joaquin Quiñero Candela, Thomas Borchert, and Ralf Herbrich. 2010. Web-scale Bayesian Click-through Rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML '10)*. Omnipress, USA, 13–20. <http://dl.acm.org/citation.cfm?id=3104322.3104326>
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). <http://arxiv.org/abs/1512.03385>
- [9] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, and Joaquin Quiñero Candela. 2014. Practical Lessons from Predicting Clicks on Ads at Facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising (ADKDD'14)*. ACM, New York, NY, USA, Article 5, 9 pages. <https://doi.org/10.1145/2648584.2648589>
- [10] Keith W Hipel and A Ian McLeod. 1994. *Time series modelling of water resources and environmental systems*. Vol. 45. Elsevier.
- [11] Charles C Holt. 2004. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting* 20, 1 (2004), 5–10.
- [12] Philippe Jorion. 2006. *Value at Risk, 3rd Ed*. McGraw Hill Professional.
- [13] Deguang Kong, Konstantin Shmakov, Xiannian Fan, and Jian Yang. 2018. A Portfolio Optimization Approach for Bid Recommendation. In *WWW'2018*. ACM, to appear.
- [14] Deguang Kong, Konstantin Shmakov, and Jian Yang. 2018. A CPA Goal Approach for Bid Optimization. In *WWW'2018*. ACM, to appear.
- [15] Deguang Kong, Konstantin Shmakov, and Jian Yang. 2018. An Inflection Point Model for Bid Recommendation. In *WWW'2018*. ACM, to appear.
- [16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.
- [17] Cheng Li, Yue Lu, Qiaozhu Mei, Dong Wang, and Sandeep Pandey. 2015. Click-through Prediction for Advertising in Twitter Timeline. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. ACM, New York, NY, USA, 1959–1968. <https://doi.org/10.1145/2783258.2788582>
- [18] H. Brendan McMahan, Gary Holt, D. 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. 2013. Ad Click Prediction: A View from the Trenches. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*. ACM, New York, NY, USA, 1222–1230. <https://doi.org/10.1145/2487575.2488200>
- [19] Berthier Ribeiro-Neto, Marco Cristo, Paulo B. Golgher, and Edleno Silva de Moura. 2005. Impedance Coupling in Content-targeted Advertising. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05)*. ACM, New York, NY, USA, 496–503. <https://doi.org/10.1145/1076034.1076119>
- [20] Benyah Shaparenko, Özgür Çetin, and Rukmini Iyer. 2009. Data-driven Text Features for Sponsored Search Click Prediction. In *Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising (ADKDD '09)*. ACM, New York, NY, USA, 46–54. <https://doi.org/10.1145/1592748.1592755>
- [21] Yukihiro Tagami, Shingo Ono, Koji Yamamoto, Koji Tsukamoto, and Akira Tajima. 2013. CTR Prediction for Contextual Advertising: Learning-to-rank Approach. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising (ADKDD '13)*. ACM, New York, NY, USA, Article 4, 8 pages. <https://doi.org/10.1145/2501040.2501978>
- [22] Taifeng Wang, Jiang Bian, Shusen Liu, Yuyu Zhang, and Tie-Yan Liu. 2013. Psychological Advertising: Exploring User Psychology for Click Prediction in Sponsored Search. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*. ACM, New York, NY, USA, 563–571. <https://doi.org/10.1145/2487575.2487699>
- [23] Peter R Winters. 1960. Forecasting sales by exponentially weighted moving averages. *Management science* 6, 3 (1960), 324–342.
- [24] Ling Yan and Wu-Jun Li. 2014. Coupled Group Lasso for Web-scale CTR Prediction in Display Advertising. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32 (ICML '14)*. JMLR.org, II–802–II–810. <http://dl.acm.org/citation.cfm?id=3044805.3044982>
- [25] G Peter Zhang. 2003. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* 50 (2003), 159–175.
- [26] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. 2014. Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI'14)*. AAAI Press, 1369–1375. <http://dl.acm.org/citation.cfm?id=2893873.2894086>
- [27] Zeyuan Allen Zhu, Weizhu Chen, Tom Minka, Chenguang Zhu, and Zheng Chen. 2010. A Novel Click Model and Its Applications to Online Advertising. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM '10)*. ACM, New York, NY, USA, 321–330. <https://doi.org/10.1145/1718487.1718528>