Purchase and Its Sign Analysis from Customer Behaviors Using Deep Convolutional Neural Networks

Shintaro Saito[1(✉)], Kohei Otake[2], and Takashi Namatame[1]

[1] Department of Industrial and System Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan
a14.s8tm@g.chuo-u.ac.jp, nama@indsys.chuo-u.ac.jp
[2] Department of Management Systems Engineering, Tokai University, 2-3-23, Takanawa, Minato-ku, Tokyo 108-0074, Japan
otake@tsc.u-tokai.ac.jp

Abstract. In this paper, we predict purchase from access log data and consider customer behavior about purchase sign on E-commerce site. In addition, applying Convolutional neural networks to this study, we discuss probability of purchase or not purchase and data preprocessing. By extracting hidden layer of model, we consider customer behaviors about purchase and not purchase. Furthermore, we discuss the way of transforming no image datasets to image-like data. We think about probability of using its networks for no images data through this study.

Keywords: Access log data · Purchase sign · Convolutional neural networks

1 Introduction

Recently, we become able to access detail customer data on electronic-commerce (EC) site and can analyze customer behavior more than before. In particular, using access log data, it extends the possibility of analyzing customer behavior and it become express customer information searching behavior. In this paper, mainly using web access log data, we predict purchasing and related behaviors in EC.

Nowadays, neural network or its extended method is applied to explain various business objectives. There are many types of research of applying deep learning to various tasks based on the improvement of computer performance. Especially, convolutional neural networks (CNN) which is often used in image recognition is applied to the churn analyzing [1]. AlphaZero might be the most famous among them [2]. For classification or predicting tasks, most of the deep learning models have a better performance than the conventional machine learning approaches (e.g. Support Vector Machine (SVM), logistic regression model), while there is a problem about incapable of explaining how the models decide for these tasks. In this study, we attempt to find features of time-series customer access log data by convolutional neural networks and consider purchasing sign from hidden layer state. Furthermore, we discuss transforming datasets of no images to image-like data.

2 Datasets

In this study, we use purchase record and web access log data on a golf EC site. Aggregating session log data by the users for days in a month, we make datasets of 30 rows and 11 columns for users. 11 columns are types of access pages which are new item pages, old item pages, news pages, and so on. Also, we label purchasing or not purchasing in the next month as supervised learning. We use various datasets, locate features in a certain order, in order of high correlation with purchasing or rearrangement placing features in some rules. These are enumerated in Tables 1 and 2 (Fig. 1).

Table 1. Locating features samples

Feature samples	Locate feature
1	In order of high correlation
2	High correlation form side to side

Table 2. Correlations of features about purchase

Feature name	Correlation with purchase
New item pages	0.023
Old item pages	0.027
Outlet pages	0.120
Sale pages	0.011
Lesson pages	0.019
News pages	0.120
Event pages	0.083
Gear pages	0.029
Reserve pages	0.067
Reserve cv pages	0.087
Purchase done pages	0.051

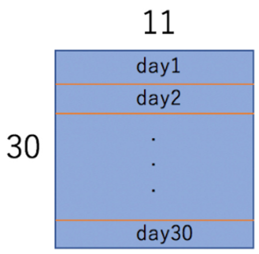

Fig. 1. The image of input data.

3 Methods

3.1 Convolutional Neural Networks

Convolutional neural networks (CNN), which is one of Deep Learning method have some convolutional and pooling layers. In convolutional layers, it is filtered small images with numbers to extract features, and it transform a certain area of images into a rough feature. In pooling layers, features which are mapped in convolutional layers are additionally summarized in maximum or average feature. After convolutional layers, we can grasp features through activation layers. Activation layers put them into a certain state. Then, we get output values from networks, then do back propagation which calculates errors of every unit in layers. These layers can make feature maps from images. This algorithm is as follows.

$$\begin{cases} u_{m,n,k}^{l+1} = \sum_{p,q,k} w_{p,q,k,k'}^{l+1} y_{m+p,n+q,k}^{l} + b_{k'}^{l+1} \\ z_{m,n,k'}^{l} = h\left(u_{m,n,k}^{l+1}\right) \end{cases} \tag{1}$$

$$where \begin{cases} k : chanel \\ k' : kernel \\ m,n : length\,of\,row\,and\,column \\ p,q : size\,of\,filta \\ p = \{1,2,\ldots,P^{l+1}\} \\ q = \{1,2,\ldots,Q^{l+1}\} \\ k = \{1,2,\ldots,K^{l}\} \\ k' = \{1,2,\ldots,K^{l+1}\} \end{cases}$$

$$\begin{cases} u_{m,n,k}^{l+1} : output\,of\,(m,n)\,pixel\,at\,(l+1)layer\,in\,k'\,chanel \\ w_{p,q,k,k'}^{l+1} : weight\,of\,(p,q)\,pixel\,at\,(l+1)layer\,and\,k\,chanel\,in\,k'\,kernel \\ z_{m,n,k'}^{l} : output\,of\,(m+p,n+q)\,pixel\,at\,l\,layer\,in\,k\,chanel \\ b_{k'}^{l+1} : bias\,every\,chanel \\ h(x) : activation\,function \end{cases}$$

Every filter of convolution layer is determined in some rules. In this study, we determined the weights of filters from He normal [5]. When n is the number of filter pixel, the weights are generated from normal distribution that mean is 0, and variance is $2/n$.

There are some activation functions, for example Relu, sigmoid, Leakly Relu. They are as shown below in Fig. 2. The architecture of CNN is shown like in Fig. 3.

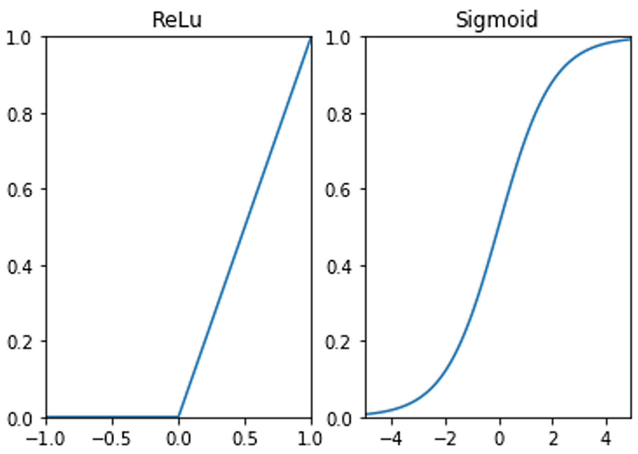

Fig. 2. ReLu and Sigmoid.

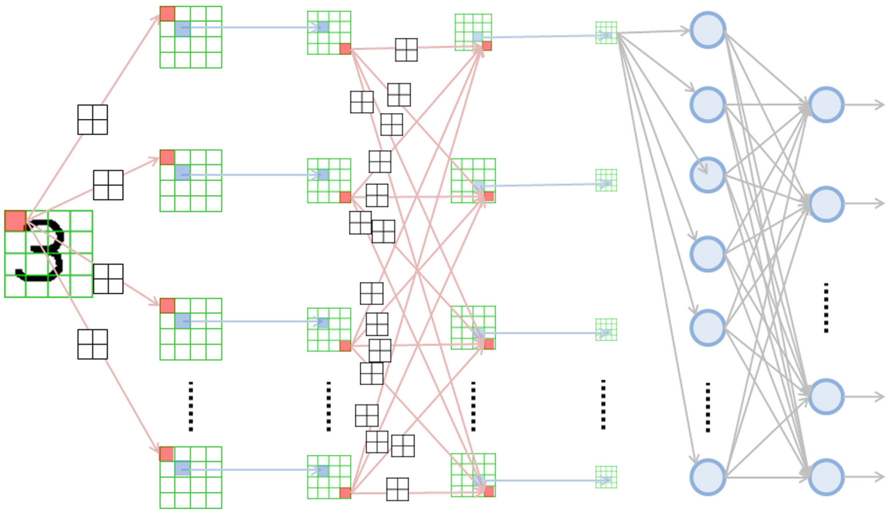

Fig. 3. The Architecture of convolutional neural networks [4].

The back propagation is as follows.

$$w_t = w_{t-1} - \eta \frac{dE}{dw} \tag{2}$$

$$\frac{dE}{dw} = \frac{dE}{dy}\frac{dy}{dt}\frac{dt}{dw} \tag{3}$$

$$\text{Where}\begin{cases} E = -\sum_{n=1} \log\left(y^{[n]}_{correct\ label\ at\ n\ sanple}\right) \\ \frac{dy}{dt} = f'(t) \\ \frac{dt}{dw} = u \\ f(t) : activation\ function \\ u : output\ of\ hidden\ layer \end{cases}$$

We get new weight when this calculate is repeated form output layer to input layer.

3.2 Residual Network (Resnet)

There are famous model structures of CNN, Resnet that won first place in ILSVRC 2015 [3] is one of them. It is very important that the depth of convolution layers for CNN. The more deep layers are, the more accurately CNN is because it can extract feature maps from convolution layers. However, it is said that the accuracy of many convolution layers model is more worse than not it. In fact, 20 convolution layers model is more accurate than 56 convolution layers model [3]. Resnet solved its problem. Residual learning uses previous input for every residual block. It is shown like in Fig. 4.

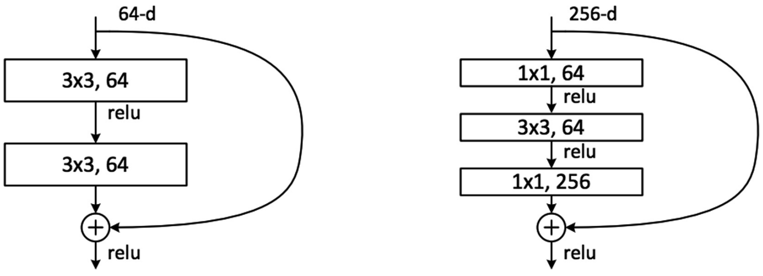

Fig. 4. The examples of residual block [3].

Resnet has some residual blocks in model, then Resnet can become more a accurate model. Using residual blocks, Resnet helps CNN can have many convolution layers, it is shown like in Fig. 5 [3].

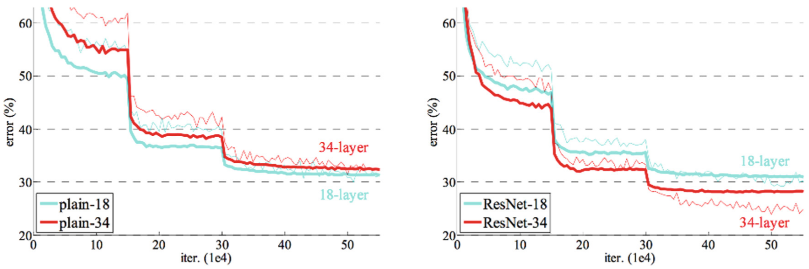

Fig. 5. The results of Resnet about ImageNet [3].

We use Resnet architecture in this study. This paper constructed a convolution layer in first convolution layer and final it, and 9 residual blocks like left on Fig. 3 every residual block have batch normalization layers after convolution layers. The filter size of every residual blocks is 3 × 3, channels are 64, training batch size is 64, and so on. To summarize, the CNN architecture is as follows Table 3.

Table 3. The architecture of CNN

The architecture	Contents
Filter size (first convolution layer)	5
Filter size (the others)	3
The number of filters	64
Optimization	Adam
Activate function	RRelu
Batch size	64

4 Data Preprocessing

Access log data is very sparse data, so in order to discriminate purchase data, we transform every pixel value into new it of subtracting it from 256. In addition, we produced impulse noise and median filter data from purchase and no purchase data to increase patterns of train data. In the end, we standardized all data. Lastly, we separate all data into train and test data, and make validation data from train data.

5 Modeling

We get the result from fitting the train and test data. These results are shown like in Table 4.

Table 4. Score of model

	Datasets 1	Datasets 2	Datasets 2′	Datasets 3
Accuracy	0.68	0.60	0.61	0.57
Precision	0.46	0.41	0.40	0.48
Recall	0.068	0.51	0.441	0.58
F-measure	**0.12**	**0.45**	**0.42**	**0.52**

- Datasets 1: not adding noise to train data in order of high correlation
- Datasets 2: adding impulse noise and median filter to train data in order of high correlation
- Datasets 2′: adding impulse noise and median filter to train data in order of high correlation from side to side
- Datasets 3: adding impulse noise to train data in order of high correlation

6 Discussion

First, we discuss the datasets. As you can see from datasets 2 and datasets 2′ of Table 4, the best way of locating feature is that every feature locates in high correlation. Furthermore, adding noise to train data is good way for modeling. It is based on datasets 1 of Table 4. Also, adding median filter is not so good preprocessing to increase the data patterns. However, impulse noise is good preprocessing for the image-like data of customer behaviors. We can say that it is good for the discrimination. In short, for the discrimination of image-like data of customer behaviors, impulse noise is should be added to the data which is sparse data.

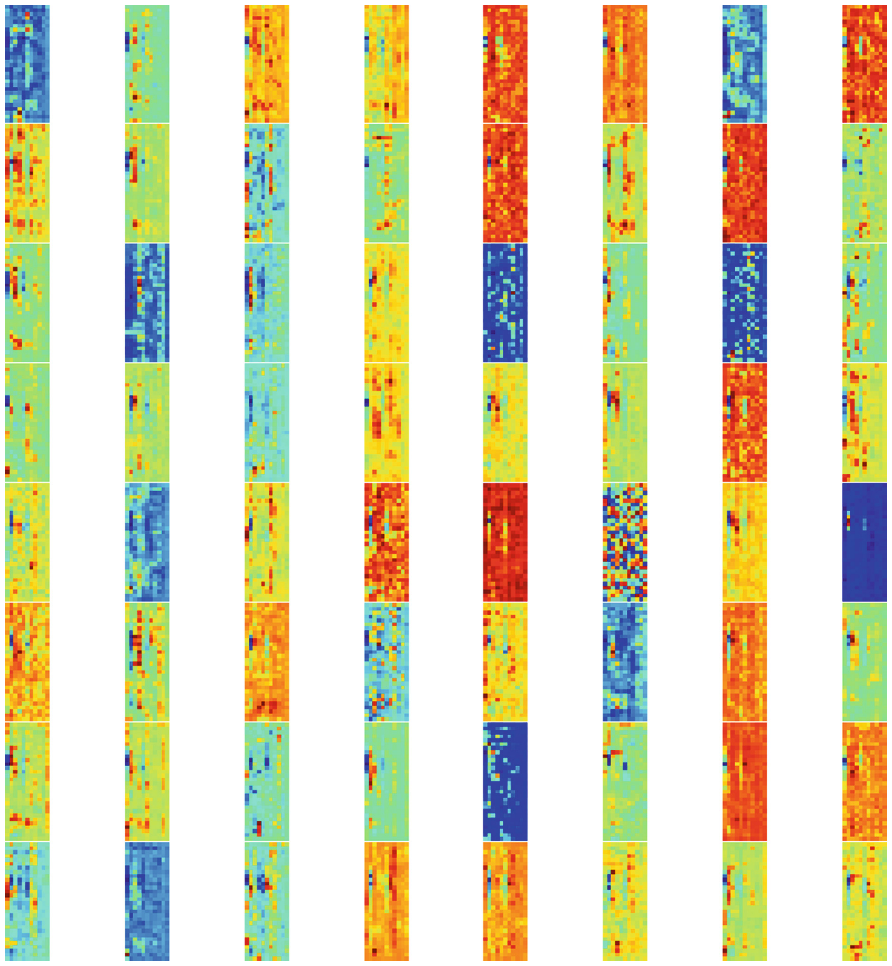

Fig. 6. Feature maps of maximum probability purchasing from last convolutional layer. (Color figure online)

Next, we discuss purchase behaviors. From here on, the data sets are regarded as Data sets 3. We extracted feature mapping from hidden layers of maximum probability purchasing or not purchasing and minimum probability purchasing or not purchasing. They are shown like in Figs. 6, 7, 8 and 9.

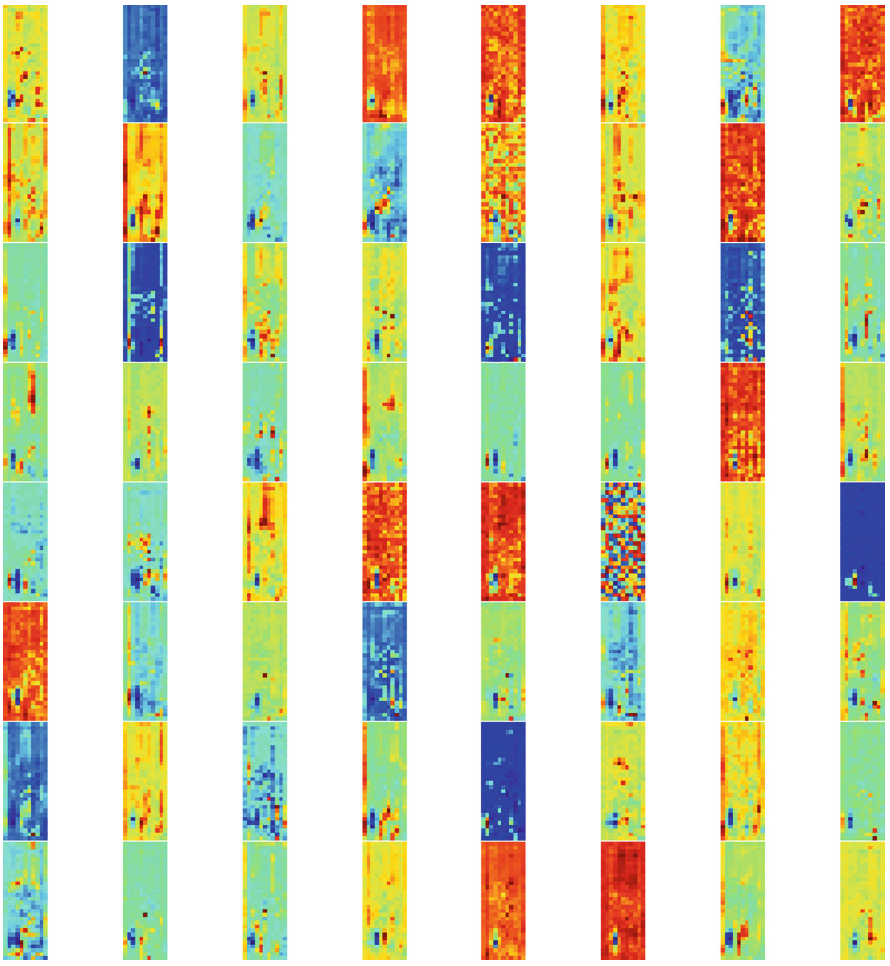

Fig. 7. Feature maps of minimum probability purchasing from last convolutional layer. (Color figure online)

It shows how networks extract features from inputted customer behaviors log data. Using this figure, we can monitor the purchasing or not purchasing trend. We use CNN for mapping time-series customer behaviors in a month, and grasp purchase or not purchase sign. Each pixel expresses the value when the color is near blue, the active value is high and red is low. As you can see Figs. 6 and 7, customer behaviors of

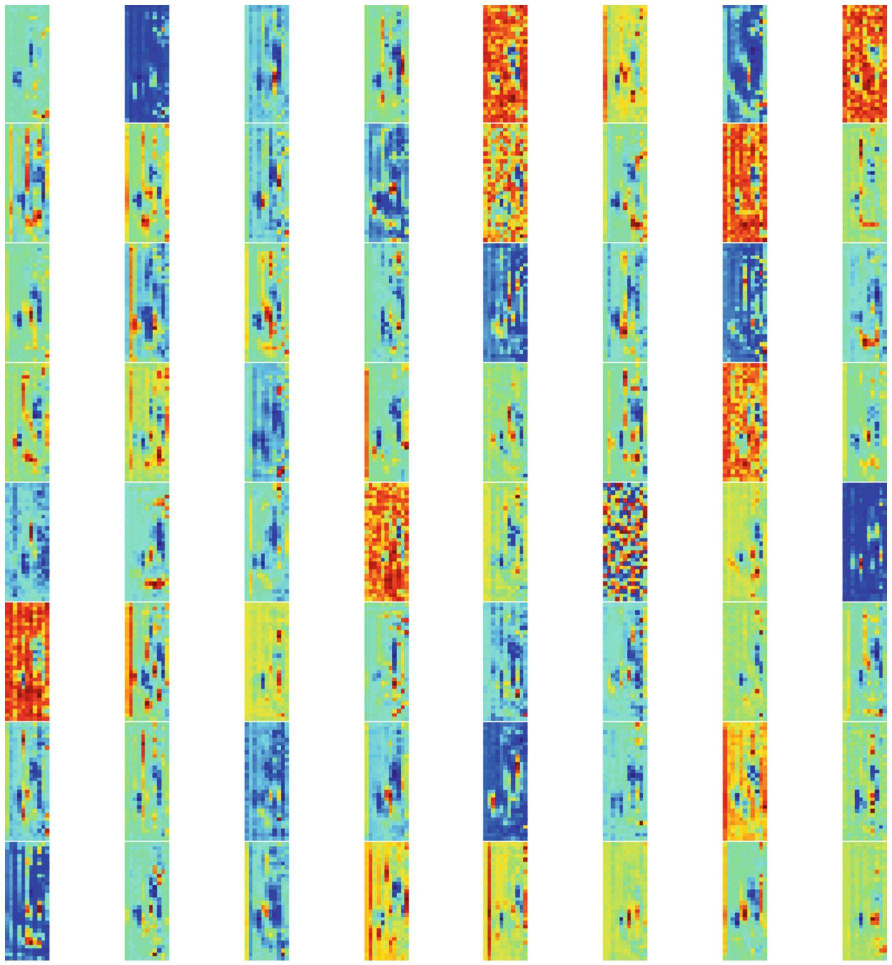

Fig. 8. Feature maps of maximum probability not purchasing from last convolutional layer. (Color figure online)

purchase class are time-series features in left edge. Namely, when time line customer behaviors are active about news pages outlet pages, gear pages, and old item pages, the possibility of purchase is high. On the other hands, there are active behaviors somewhere in a month, the possibility of purchase is low. Considering kinds of pages about max possibility of purchase, the time-series activity of customers about news pages or outlet pages is thought to be purchase sign. And, customers behaviors are active at the end of month, possibility of purchase is low. In terms of not purchasing class, when customer behaviors are active about such new item pages or sale pages, the possibility is low. They may use this E-commerce site for viewing products. Customers for minimum probability of not purchase are active about center of data. Thus, they are active users for reserving golf course and do not use this site not E-commerce site as reserving site.

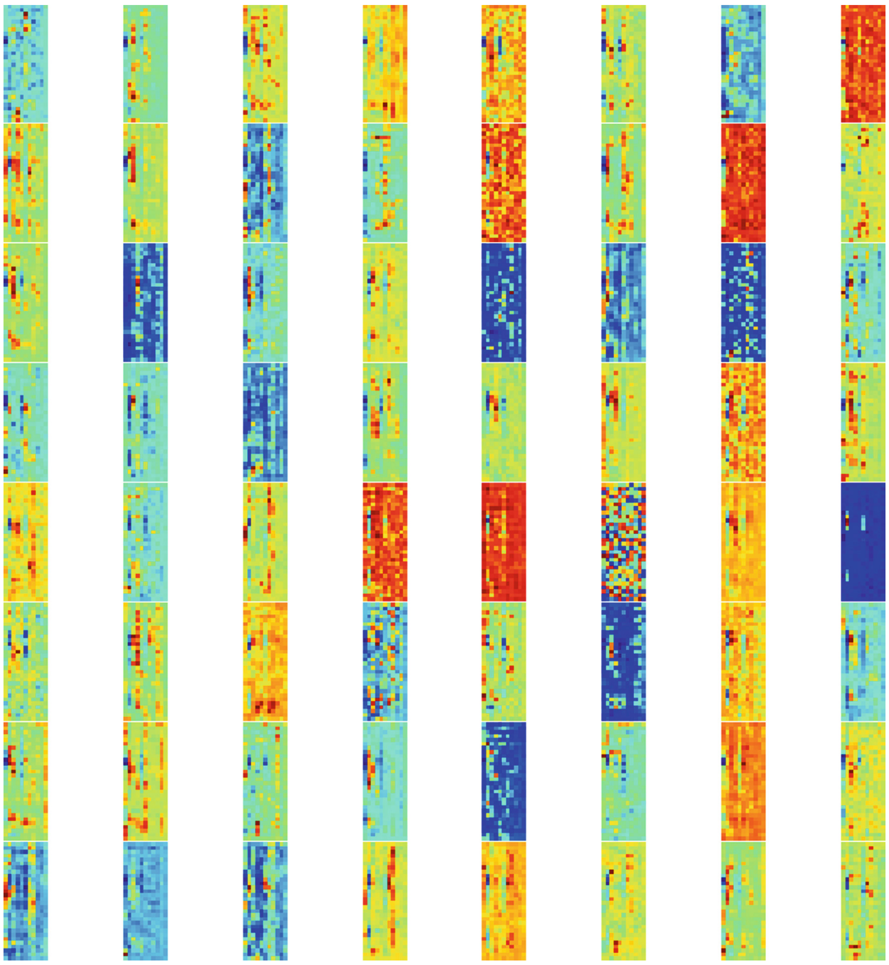

Fig. 9. Feature maps of minimum probability not purchasing from last convolutional layer. (Color figure online)

7 Conclusion

In this study, we proposed a CNN model to predict purchase and grasp its sign. Especially, to visualize hidden layers, we could grasp customer behavior from time-series its access log data by using CNN. So, this study expanded ability of applying CNN to no images discrimination. However, some studies are remaining, for example, more interpreting the characteristics of hidden layers or improving model accuracy. These are our future works.

References

1. Wangperawong, A., Brun, C., Laudy, O., Pavasuthipaisit, R.: Churn analysis using deep convolutional neural networks and autoencoders. Cornell University (2016). arXiv:1604.05377
2. Silver, D., et al.: Mastering the game of Go with deep neural networks and tree search. Proc. Nat. **529**, 484–489 (2016)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
4. Neural Network - Deep Learning image source for presentation or seminar. http://nkdkccmbr.hateblo.jp/entry/2016/10/06/222245. 31 Jan 2019
5. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: ICCV, pp. 1026–1034 (2015)