Prediction of preferences on M-Learning based on VARK score using DNN to classify multi-label and single-label data

Yiyi Shao

Research School of Computer Science, Australian National University
u6554479@anu.edu.au

Abstract. Mobile learning, also known as M-Learning, is a popular way to use digital technology to learning multiple contexts in the learning and teaching process. This paper will illustrate how to use a Deep Neural Network (DNN) to solve a classification problem with a symbolic dataset by using Pytorch/Python. Specifically, this DNN model is trained to predict four different preferences of mobile learners, which are audio, PowerPoint, video, and e-book base on VARK (visual, aural, read/write, kinesthetic) scores. Modern techniques are applied to preprocess data, construct a Multi-layer perceptron (MLP) model, and evaluate its performance. Meanwhile, the dataset and one modern technique are relatively from two published research paper in order to make comparisons and discuss the results of multi-label classifier and single-label classifier.

Keywords: Mobile learning, M-learning preferences, DNN, MLP, K-fold cross-validation, symbolic data

1 Introduction

M-learning is a current mainstream of digital learning to support continuous access to the learning process. With the help of mobile devices, mobile learners are able to learn through a variety of virtual media, such as audio, PowerPoint, video, and e-book. To predict the preferences on M-learning is a significant way to personalize mobile learners' usage and target potential users for the virtual media applications.

This paper introduces a Deep Neural Network (DNN) classification model to predict four preferences of mobile learners, including audio, PowerPoint, video, and e-book, by using the symbolic dataset from the study of AI-Ismail, Gedeon, and Yamin (2017). The data was from a survey which collected 345 students from Australia and Saudi Arabia. In order to extend the research and modify the purpose of the use of the dataset, this paper is intended to train an DNN model to classify the preferences of different mobile learners by analyzing the VARK (visual, aural, read/write, kinesthetic) scores with multi-label data and single-label data, instead of statistically analyzing the effects of personality traits and preferences on M-learning [1].

There are basically three stages in the process of implementation, which are preprocessing, training, testing. In the preprocess stage, by applying the encoding technique from the study of Bustos and Gedeon (1995), the development of this classification model became simpler. Decrypting neural network data is an effective approach to deal with large amount of erratically reliable data and produce meaningful prediction [2]. The pre-encoding approaches for the input and output simplified the structure of the dataset. Furthermore, normalization is performed to keep data contrast in one level. After the implementation of DNN, the k-fold cross validation was used to optimize the training of the model. Moreover, this paper also compares and discusses the results of two experiments to see the difference between multi-label classifier and single-label classifier.

2 Method

This section will illustrate the technique implemented to develop the model and test to evaluate the accuracy of the classification model, which includes data analysis, data preprocess, DNN implementation, k-fold cross-validation.

2.1 Data analysis

There are five sheets in the original data, but only Sheet 3 was used as the raw data in this research. The small dataset originally has 345 records and 188 columns. Based on the first 140 columns, the task is to predict the value for the last 48 columns, which are in groups of 4 preferences (audio, ppt, video, reading) in 12 contexts. Instead of having 48 outputs, the classifier will be trained to only predict the preferences among audio, ppt, video, reading. To achieve this, the raw data will be expanded and add one column to represent the last 48 columns, which also provides the labels for the training data. As Fig. 1 shown, this symbolic dataset needs preprocess method to encode the data into numerical format in order to build the neural network model in the later step. Moreover, it was also shown that some columns represent the question answers collected from the survey, while certain column are the analysis data extracted from the question answers. Thus, it is essential to remove useless columns to reduce duplicate information and simplify the dataset. After the exploratory data analysis, five features are decided to use as the input data (Fig. 2).

Age	Gender	Current Status	Nationality	listened education	Watched education	Read Education
18-24	f	married	Saudi	y	y	y
18-24	f	single	Saudi	y	y	y
18-24	f	single	Saudi	n	y	y
18-24	f	single	Saudi	y	y	y
18-24	f	single	Saudi	y	y	y
18-24	f	single	Saudi	y	y	y
18-24	f	single	Saudi	y	y	y
18-24	f	single	Saudi	y	y	y
18-24	f	single	Saudi	y	y	y
18-24	f	single	Saudi	y	y	y
18-24	f	single	Saudi	y	y	y

Fig. 1. For some columns in the dataset, the contexts consist of intervals, words or letters. It would be difficult to feed the network without encoding the raw data.

# A score	K score	R score	V score	preference
78.21748656	35.46917585	10.22628259	26.54687532	0
37.69705841	35.46917585	42.01454322	36.27202918	0
57.95727249	64.67190011	52.6106301	45.99718305	0
57.95727249	74.40614153	52.6106301	45.99718305	0
27.56695137	45.20341727	20.82236947	55.72233691	0
37.69705841	45.20341727	63.20671698	16.82172145	0
17.43684433	25.73493444	42.01454322	65.44749078	0

Fig. 2. Data set with 345 rows and five columns. The first 4 columns selected from the original data, while the last column extracted from the last 48 columns of the original data and was encoded into integer.

2.2 Data Preprocessing

Real world data is always incomplete and may cause errors when it is sent to the model. Data preprocessing is a data mining technique that transforms raw data in order to make them understandable for a model. Several methods used in the data preprocessing stage for the purpose of encoding, normalization, filtering noise.

Input encoding. A suitable pattern representation is significant for good learning [2]. Several symbolic columns had been applied integer encoding, such as gender, current status, nationality. In addition, to convert each unique symbolic label to numerical format, integer encoding was used to map each label to an integer, as shown in Table 1.

Table 1. Label data, which represents user preference, are mapped to integers

Preference raw data	Numerical format
audio	0
ppt	1
video	2
e-book	3

There are several approaches to encode data. First, an encoded representation of feature maps can be stored and decode them for use during backward [3]. This is a useful way for Deep Neural network which basically depends on GPU to run a complex deep network. Second, Convolutional Neural Network also needs efficient data encoding to reduce the required area for data storage [4].

2.3 Deep Neural Network implementation

To solve the classification problem, a Multi-layer perceptron (MLP) model is constructed with 1 hidden layer.

Activation function. In a neural network, the responsibility of activation function is to transfer the summed weighted input from the node into the activation of the node or output for that input. Rectified Linear Unit (ReLU) is one of the most widely used activation functions [5]. The result of ReLU is the maximum value of zero and the input, while the sigmoid squishes the input values between 0 and 1. When it is training on a reasonable sized batch, there will be data points giving positive values to given node and make the average derivative rarely close to zero. This allows the gradient descent to keep progressing.

$$\mathrm{ReLu}(x) = \max(0, x) \tag{1}$$

Loss function. Cross-Entropy loss is applied to measure the performance of this classification model whose output is a probability value between 0 and 1. As shown in the Fig. 3, a perfect model would have a log loss of 0.

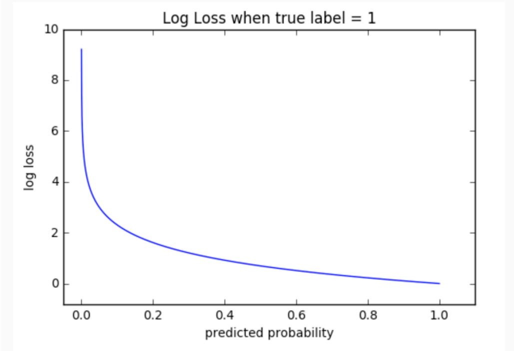

Fig. 3. The range of possible loss values given a true observation.

Optimizer. Adam is an adaptive learning rate method for deep learning, which computes individual learning rates for different parameters. In order to estimate the first and second moments of gradient, the learning rate is adapted for each weight in the neural network.

K-fold cross validation. K-fold cross validation is an effective procedure of evaluating the classification model. K equivalent subsets were randomly selected from the modeling set, then fit the model using the k-1 folds and validate using the left one and repeat until every fold has been the test set [6]. It is helpful to reduce computation time and bias, and the variance of the result estimation is reduced due to the increase of k.

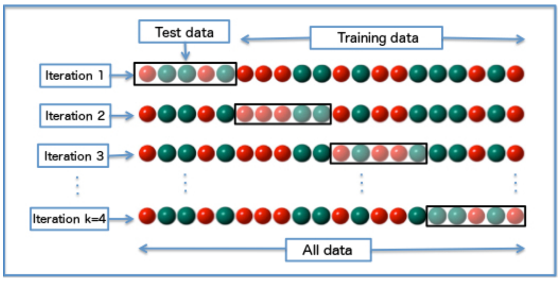

Fig. 4. The k-fold cross validation follows the steps shown.

3 Results and discussion

Two experiments have been conducted to figure out which kind of dataset is more suitable to solve this classification problem. Two datasets, which are multi-labeled and single-labeled respectively. As shown in Fig. 5, in the multi-label case, one sample might be assigned more than one class, while there are more than 2 classes in total in the multi-class case. In this section, the results of two experiments will be compared and discussed.

multiclass

	0	1	2	3	4
0	1.000	0.000	0.000	0.000	0.000
1	0.000	1.000	0.000	0.000	0.000
2	0.000	1.000	0.000	0.000	0.000
3	0.000	0.000	0.000	1.000	0.000
4	0.000	1.000	0.000	0.000	0.000
5	1.000	0.000	0.000	0.000	0.000
6	0.000	0.000	1.000	0.000	0.000

Format Resize Background color OK Cancel

(a) Labels of multi-class data

multilabel

	0	1	2	3	4
0	1.000	0.000	1.000	0.000	1.000
1	0.000	1.000	0.000	0.000	0.000
2	0.000	1.000	0.000	0.000	0.000
3	0.000	0.000	0.000	1.000	0.000
4	0.000	1.000	1.000	0.000	0.000
5	1.000	0.000	0.000	1.000	0.000

Format Resize Background color OK Cancel

(b) Labels of Multi-label data

Fig. 5. The difference between multi-class and multi-label classification.

Multi-label dataset. There are 48 columns of labels to represent the learning performance of mobile users. For each sample, there might be more than one mobile preference for each mobile learner.

Single-label dataset. There is only one column as the label column, which present 4 different mobile learning preference (audio, ppt, video, reading), which represent each mobile learning can only have one preference.

From the description of the dataset, it is obvious that multi-label dataset is more representative to feed into the network in order to achieve user customization. However, having 48 outputs seems difficult to train the model, compared with 1 output.

3.1 Experiment 1: The single-label, multi-class classification

This experiment using a single-label, 4-class dataset to train the MLP model. The accuracy of 94% was achieved, with a significant problem: the data is seriously unbalanced. As shown in the Table 2. Most machine learning classification algorithms are sensitive to unbalanced data of the predict classes. The model that has been trained and tested on such a dataset could now predict “audio” for all samples and still gain a very high accuracy, because an unbalanced dataset will bias the model to predict the more common class. In this case, the model only learned how the audio preference data look like.

There are many approaches to balance data for modelling. For example, the training set can be resampled by the methods of under-sampling and over-sampling.

Under-sampling. It balances the dataset by reducing the size of the abundant class. This method is used when quantity of data is sufficient. In the process of k- fold validation, it randomly selects an equal number of samples to create a balanced new dataset for each epoch. However, in our case, the number of other three classes are insufficient to do under-sampling.

Oversampling. It is used when the quantity of data is insufficient. It tries to balance dataset by increasing the size of rare samples. Instead of getting rid of abundant samples, new rare samples can be repeated in each fold of cross validation. which is a good way to resolve the imbalance problem.

Table 2. The total number of each class.

Preference raw data	Total
audio	326
ppt	8
video	5
e-book	6

3.2 Experiment 2: DNN multi-label classification

This experiment using a multi-label dataset, which has 48 columns to represent the preference of mobile leaners. The accuracy of 76% was achieved. Multi-label is more natural for many real problems, such as predicting purchase and object detection. However, more extra labels cannot make sure higher accuracy and better performance. In this case, the small dataset is insufficient to provide enough samples for each class and some extreme class often unbalanced.

There are many approaches to improve the performance of the multi-label classifier. For example, a simple technique called Binary Relevance can be used to treat each label as a separate single class classification problem. Assume that the data set is like Fig. 6 (a), that X is the independent feature and Y is the target variable. This multi-label problem can be broken in to 4 single-class problem as shown in Fig.6 (b). It is a straightforward method but arguable because some correlations between labels may be ignored and make it a potentially weak performance.

X	Y_1	Y_2	Y_3	Y_4
$x^{(1)}$	0	1	1	0
$x^{(2)}$	1	0	0	0
$x^{(3)}$	0	1	0	0
$x^{(4)}$	1	0	0	1
$x^{(5)}$	0	0	0	1

(a) Assumed dataset

X	Y_1
$x^{(1)}$	0
$x^{(2)}$	1
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	0

X	Y_2
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	1
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_3
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_4
$x^{(1)}$	0
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	1

(b) In binary relevance

Fig. 6. Binary relevance to break the multi-label problem into single class classification

4 Conclusion and future work

This paper investigates how to use pre-encoded data to train a classification model for the prediction of mobile learners' preferences based on VARK scores. The motivation of choosing this dataset is related to the popularity of mobile learning. A DNN model was constructed to solve the classification problem. Methods used are introduced to provide in an understandable way. A discussion of whether output encoding should be performed during the data processing was provided as well. In this paper, the techniques mentioned in the study of Bustos and Gedeon are applied in the DNN model. Bustos and Gedeon discuss many approaches for data encoding to consider. According to the actual condition, the symbolic parts of the raw data have to be encoded into integer representations in order to provide convenience for the training. Although this paper analyzed the results of the pretrained model, some effort was required to improve the performance in the future. For example, algorithms related to feature selection and feature extraction can be applied to make the prediction more meaningful. In addition, more approaches can be attempted to better resolve the unbalanced multi-label classification problem in the future, such as Classifier Chains and Label Powerset.

References

1. M. Al-Ismail, T. Gedeon, and M. Yamin, "Effects of personality traits and preferences on M-learning," International Journal of Information Technology, vol. 9, no. 1, pp. 77–86, 2017.
2. R. A. Bustos and T. D. Gedeon, "Decrypting Neural Network Data: A Gis Case Study," Artificial Neural Nets and Genetic Algorithms, pp. 231–234, 1995.
3. A. Jain, A. Phanishayee, J. Mars, L. Tang, and G. Pekhimenko, "Gist: Efficient Data Encoding for Deep Neural Network Training," 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA), 2018.
4. H.-P. Trinh, M. Duranton, and M. Paindavoine, "Efficient Data Encoding for Convolutional Neural Network application," ACM Transactions on Architecture and Code Optimization, vol. 11, no. 4, pp. 1–21, Sep. 2015.
5. W. Wang, M. T. Kim, A. Sedykh, and H. Zhu, "Developing Enhanced Blood–Brain Barrier Permeability Models: Integrating External Bio-Assay Data in QSAR Modeling," Pharmaceutical Research, vol. 32, no. 9, pp. 3055–3065, Nov. 2015.
6. D. Yarotsky, "Error bounds for approximations with deep ReLU networks," Neural Networks, vol. 94, pp. 103–114, 2017.