J. Chosun Natural Sci.
Vol. 13, No. 4 (2020) pp. 147 – 151
https://doi.org/10.13160/ricns.2020.13.4.147

Concept Drift Based on CNN Probability Vector in Data Stream Environment

Tae Yeun Kim[1] and Sang Hyun Bae[2†]

Abstract

In this paper, we propose a method to detect concept drift by applying Convolutional Neural Network (CNN) in a data stream environment. Since the conventional method compares only the final output value of the CNN and detects it as a concept drift if there is a difference, there is a problem in that the actual input value of the data stream reacts sensitively even if there is no significant difference and is incorrectly detected as a concept drift. Therefore, in this paper, in order to reduce such errors, not only the output value of CNN but also the probability vector are used. First, the data entered into the data stream is patterned to learn from the neural network model, and the difference between the output value and probability vector of the current data and the historical data of these learned neural network models is compared to detect the concept drift. The proposed method confirmed that only CNN output values could be used to reduce detection errors compared to how concept drift were detected.

Keywords: Convolution Neural Network Algorithm (CNN), Concept Drift, Data Stream, Probability Vector

1. Introduction

With the distribution of smart device and the development of sensor computing environment including Internet of Things (IoT), many of the data stream environments have been established[1]. Data entered in data stream environment have concept drift as their features to have changing trend as time passes by[2,3]. Most of the applications such as mechanical learning, sampling, or filtering in data stream environment operate as data with fixed probability distribution or static data are assumed. In case of concept drift, applications such as sampling or filtering are required to newly operate perform learning according to changes or change the operation to prevent quality degradation in predictive functions or algorithm results. Therefore, there is a need to accurately detect concept drift occurring in data stream environment.

In order to detect concept drift in data stream environment, a technique using convolutional neural network has been suggested[4,5]. This method compares output labels from Convolution Neural Network Algorithm (CNN) about the current and previous inputs and detect concept drift. However, there is an issue in this method for how there might be an error in deriving concept drift in case of label vibration for releasing different labels. Such an error in detecting concept drift causes unnecessary drift in data stream applications and also functional and quality degradation. Therefore, there is a need to come up with a method to accurately detect concept drift.

In this study, a technique using probability vector is suggested to enhance the accuracy of concept drift detecting technique in the use of CNN in data stream environment[6].

2. Related Research

In the data streaming environment where massive data are continuously entered, it is difficult to save all the entered data. Therefore, they are processed in window unit. Assuming the data probability distribution as P_i, P_{i+1} from two windows i, i+ to contain data in two adjacent periods, concept drift are defined to occur in i+1 period in case of P_i, P_{i+1}[7,8].

[1]National Program of Excellence in Software center, Chosun University, Gwangju
[2]Department of Computer Science & Statistics, Chosun University, Gwangju

[†]Corresponding author : shbae@chosun.ac.kr
(Received : October 2, 2020, Revised : November 11, 2020, Accepted : November 28, 2020)

CNN is a neural network combining convolutional filter and neural technology to be used for recognizing an image[9,10]. CNN proceeds classifying data by using internal Feedforward Neural Network (FNN) that extracts characteristics of data by using convolutional filter and releasing label with an input of extracted characteristics. If entering data in the size of one window to learned neural network, it is feasible to obtain label of entered data. According to studies conducted to deal with concept drift detecting technique by using convolutional neural network in data stream environment, CNN was distributed to detect concept drift, while entering data to neural network in two different periods for once, respectively[11,12]. Neural network compares two labels generated in two different periods. If these two labels are different, it is judged that concept drift occur. If they are equal, it is judged that no concept drift occur.

3. Concept Drift Detection Method Using Neural Network

In data stream environment where data are rapidly entered, it is not possible to save all the data and process them. Therefore, data are processed in window unit. In this study, simple data stream where 0 or 1 is to be entered as shown in the Fig. 1 is assumed. Fig. 1 represents a list of entered data from the left side expressing stream with one line. Window groups data in certain size and is regarded to be applied to a certain period depending on the nth data being entered. Afterwards, every time when new data are entered, the most past data are removed, while including new data and configuring a window in the next period.

Under these circumstances, the distribution with 1 as an input is assumed to be important, detecting significant changes in the distribution of 1 as a conceptual drift. In other words, concept drift are regarded to occur when the distribution of 1 has significantly changed between two windows. Suggested method uses CNN to measure the density of 1. In order to apply CNN designed for classification of video data to general data stream environment, data stream input is converted to be treat as video data. Window data are converted to black and white video data with the height of 1 where the width of data is the same as window size. At this time, light intensity value of each pixel of converted data become data value.

CNN is learned by using converted window and label as learning data. Label means a representative value of windows with similar distributions of 1. In this study, label is expressed as an integer from 0 to 5 by classifying the density of 1 in each window in the unit of 20% (0%=0, 20%=1,…, 100% = 5). Learned CNN is entered with window generating the label on entered window.

CNN calculates probability vector as shown in the Fig. 2 prior to deciding the label to be generated when window is entered at a random period. Probability vector is saved with probability for how input window is included in each label. The number of elements in probability vector is the same as the number of label, and each location of vector elements corresponds with one label. Since the total of probability is 1, the total of all the elements in probability vector is 1, and each element value is between 0 and 1. Label corresponding to the location of the highest value of probability vector calculated by CNN becomes the output label in input window. In the Fig. 2, 1 as a location of 0.6 as the highest value in probability vector is generated as a label.

In this study, both label and probability vector are used to detect concept drift. In order to use probability vector generated by CNN to detect concept drift, CNN is distributed to the stream as shown in the Fig. 3. Since 2 probability vectors are required for one attempt in detecting concept drift, CNN was expressed for two times in the Fig. 3. Window at time i and window at time i+1 are entered once, respectively, in CNN.

CNN generates each probability vector PV_i and P_i, PV_{i+1} on entered windows. Every time when new data

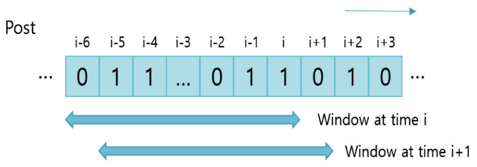

Fig. 1. Data stream environment.

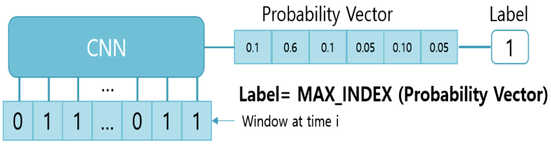

Fig. 2. Convolution Neural Network algorithm (CNN).

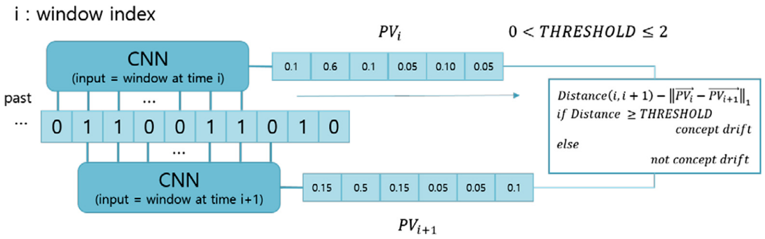

Fig. 3. CNN placement for concept drift detection.

are entered in stream, neural network generates probability vector on input window.

If there is a significant change when neural network compares two probability vectors generated in two different window periods, it is judged that concept drift occur in i+1 period. If there is no significant change, it is judged that no concept drift occur. In this study, probability vector obtained in two different window periods, (PV_i, PV_{i+1}), was used, utilizing the Equation. (1) as a criterion to detect concept drift.

$$\text{Distance (i, i + 1)} = \|PV_i - PV_{i+1}\|_1 \tag{1}$$

Distance is the difference between element values located in the same position for all the positions of elements in two probability vectors, while calculating the difference of absolute values of two probability vectors by adding all the differences. Since distance is a result calculated by adding the differences of two probability vectors, it is in a range from 0 to 2. The smaller the distance value is, the more likely it means how similar probability vectors have been generated. The close the value is to 2, the more likely it means for two windows to generate different probability vectors.

The smaller the distance value is, the more likely it means how similar probability vectors have been generated. The close the value is to 2, the more likely it means for two windows to generate different probability vectors.

In order to detect concept drift by using distance, threshold value is setup as a criterion for detecting concept drift. The value same as distance in the range from 0 to 2 is setup as a threshold value. If distance value represents a value higher than the threshold value, it is judged that concept drift occur in the i+1 period. Other than this case, it is judged that no concept drift have occurred.

4. Experiment and Evaluation

4.1. CNN and Learning Data

CNN to be used for the experiment in the use of TensorFlow is established generating a pair of simple window and label for CNN learning[13]. One window is generated to save total 1000 values, each one from 0 to 1. Label is a randomly granted number depending on the density of 1 included in each window, and density of 1 in each label is shown in the Table 1.

At this time, if the number of data in each label is different in learning data, there might be an issue of class imbalance. In this study, the number of learning data is generated by 10,000 in each label to resolve aforementioned issue. In addition, the number of 1 in each window was controlled, generating random position of 1 to prevent a significant different of density of 1 among data with label.

If entering only the window except for labels of all the learning data to CNN learned with simple data, the proportion of generating the same label from learning data was 99.9%. Information of organized neural network is shown in the Table 2.

Table 1. Density of 1 by label for learning data

Label	0	1	2	3	4	5
Density	0%	20%	40%	60%	80%	100%

Table 2. CNN information used in the experiment

Category	Value
Window Size	1000
Number of learning	20000 Times (Batch =200)
CNN Filter	1×5×56(First Floor), 1×5×480(Second Floor)
FNN Scale	1024 (First Floor)
Hit rate	99.9%

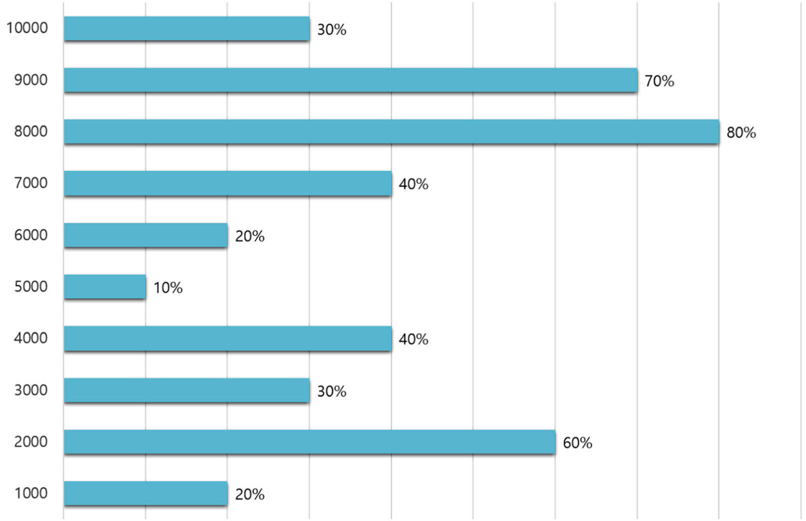

Fig. 4. Density of 1 point in time for evaluation data.

4.2. Experiment Data

Experiment data were generated identically with learning data, but the number of it was total 10,000 without label. At the same time, concept drift were represented by showing changes in the density of 1 in each window. Threshold value was set to be 1.0.

Density of 1 in data window in each period is shown in the Fig. 4.

Horizontal axis means window period, and vertical axis means the density of 1 in each period. It is expected to detect concept drift in 1000, 2000, 4000, 6000, 7000, and 9000 as a period when there is a difference in density of 1 to be higher than 20%.

4.3. Experiment Results

According to the results of detecting concept drift on experiment data with CNN, horizontal axis meant window period, and vertical axis meant the value of probability vector in each period. Threshold value was shown to be 1.0 on the horizontal axis. When distance value turns out to be higher than this horizontal axis, concept drift are detected.

Distance value turned out to be higher than 1.0 in the periods of 1000, 2000, 4000, 6000, 7000, 8000, and 9000 as a period when density of 1 was higher than 20%. Therefore, concept drift were detected. In the period of 8000 when only changes of 10% (less than 20%) were randomly applied, concept drift were confirmed. In the interval where it was difficult to accurately identify the label with density in two adjacent labels, relatively high distance value was confirmed.

However, since there was no distance value to be higher than 1.0 as setup to be threshold value in this interval, concept drift were not detected. When detecting concept drift by simply comparing label values, there is an error in detection from label vibration in the interval. If detecting concept drift by using probability vector, changes in label not exceeding threshold are ignored even if there is label vibration making it feasible to reduce the error in detection.

5. Conclusion

In this study, data entered to data stream were made to be a pattern and learned to CNN model, while comparing the difference of probability vector between previous data in learned CNN model and current output of data to detect concept drift. It was confirmed to be able to reduce error in detection through the suggested method compared to when detecting concept drift by using only CNN output values.

Acknowledgments

This study was supported by research funds from Chosun University, 2020.

References

[1] K. J. Kim, S. Y. Oh and M. S. Lee, “Pattern Classification for IoT Stream Data using Convolutional Neural Networks”, Journal of KIISE, Vol. 35, No. 2, pp. 106-115, 2019.

[2] A. Haque, L. Khan and M. Baron, “Semi supervised adaptive framework for clasifying evolving data stream”, Advances in Knowledge Discovery and Data Mining, volume 9078 of Lecture Notes in Computer Science, Springer International Publishing, pp. 383-394, 2015.

[3] T. Y. Kim, S. H. Bae and Y. E. An. “Design of Smart Home Implementation Within IoT Natural Language Interface”, IEEE Access, Vol. 8 pp. 84929-84949, 2020.

[4] E. J. Lee, S. Y. Oh and M. S. Lee, “Pattern Classification based on Attention Mechanism and CNN for Sensor Stream Data including Missing Values”, Journal of KIISE, Vol. 36, No. 2, pp. 56-68, 2020.

[5] S. Cheng and G. Zhou, “Multi-stream CNN for facial expression recognition in limited training data”, Multimedia Tools and Applications, Vol. 78, No. 16, pp. 22861-22882, 2019.

[6] E. S. Lee and E. R. Jeong, “Deep Learning based Frame Synchronization Using Convolutional Neural Network”, Journal of the Korea Institute of Information and Communication Engineering, Vol. 24, No. 4, pp. 501-507, 2020.

[7] A. S. Iwashita and J. P. Papa, “An overview on concept drift learning”, IEEE Access, Vol. 7, pp. 1532-1547, 2018.

[8] J. Lu, A. Liu, Y. Song and G. Zhang, “Data-driven decision support under concept drift in streamed big data”, Complex & Intelligent Systems, Vol. 6, No. 1, pp. 157-163, 2020.

[9] S. B. Yang and S. J. Lee, “Improved CNN Algorithm for Object Detection in Large Images”, Journal of The Korea Society of Computer and Information, Vol. 25, No. 1, pp. 45-53, 2020.

[10] J. H. Choi, “Binary CNN Operation Algorithm using Bit-plane Image”, Journal of Korea Institute of Information, Electronics, and Communication Technology, Vol. 12, No. 6, pp. 567-572, 2019.

[11] B. Krawczyk and M. Woźniak, “One-class classifiers with incremental learning and forgetting for data streams with concept drift”, Soft Computing, Vol. 19, No. 12, pp. 3387-3400, 2015.

[12] S. Wang, L. L. Minku and X. Yao, “A systematic study of online class imbalance learning with concept drift”, IEEE transactions on neural networks and learning systems, Vol. 29, No. 10, pp. 4802-4821, 2018.

[13] Z. J. Gao, N. Pansare and C. Jermaine, “Declarative Parameterizations of User-Defined Functions for Large-Scale Machine Learning and Optimization”, IEEE Transactions on Knowledge and Data Engineering, Vol. 31, No. 11, pp. 2079-2092. 2018.