Estimation of Functions Representing Data Using Convolution Neural Network

Issei Koga
Graduate School / Faculty of Information Science and Electrical Engineering, Kyushu University
Nishi-ku, Fukuoka, Fukuoka, JAPAN
2IE17031W@s.kyushu-u.ac.jp

Kenji Ono
Research Institute for Information Technology, Kyushu University
Nishi-ku, Fukuoka, Fukuoka, JAPAN
keno@cc.kyushu-u.ac.jp

ABSTRACT

Symbolic regression can be useful for finding suitable formula, with reasonable form and behavior, from a given data for the already known physical systems. However, it can also be time consuming with inaccurate estimation results when dealing with complicated formulas and/or unknown systems. In order to reduce the computational cost and improve the estimation accuracy, it becomes highly desirable to prepare a short list of candidate functions that can compose the analytical formula. In this paper, we propose an estimation method to find the suitable functional forms by using the CNN framework with combined data value and its derivatives, from a given data without any previous knowledge.

CCS CONCEPTS

• **Computing methodologies** → *Classification and regression trees*;

KEYWORDS

Symbolic Regression, CNN, MLP, preconditioning

ACM Reference Format:
Issei Koga and Kenji Ono. 2018. Estimation of Functions Representing Data Using Convolution Neural Network. In *Proceedings of ACM conference (HPCAsia2018)*. ACM, Tokyo, JAPAN, Article 39, 3 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

There are various data analysis methods, such as statistical analysis, principal component analysis, and others. When a given data lack come necessary information, or suffers from the presence of strong noise, it can become difficult to select the most appropriate analysis method based on the previous knowledge or empirical rules. Therefore, it may cause inaccurate understanding for the underlying physical phenomena or model. To avoid from such issues, an automated analysis that provides precise analysis and interpretation, which does not depend on human knowledge and heuristics, becomes highly desirable. Symbolic regression can automatically search for a suitable formula that represents a given data by simultaneously searching the form and parameters of equations (analytical formulas). We can find reports in the literature [Martius and Lampert 2016; Schmidt and Lipson 2009] that it is possible to find suitable equation, with reasonable form and behavior, for the already known physical systems. Since some of the functions may not be utilized in certain approaches, it will be helpful if the functional form of the analytical formula is known in advance, and this will also be helpful for reducing the searching time. Therefore, it becomes highly desirable to prepare a short list of candidate functions that compose the analytical formula to reduce the computational time as well as to increase the estimation accuracy of the functional forms. In this paper, we propose an estimation method to find the suitable functional forms by using the Convolutional Neural Network (CNN). The CNN is used to find the suitable function for the analytical solution of a given data without any previous knowledge.

HPCAsia2018, Tokyo,JAPAN, March 2018

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/0000001.0000001

2 DATA LEARNING

2.1 Data Preparation

In this work, we did not use measured data from a real world system, and we used instead a set of artificially generated data (synthetic data). This data set was generated by using a set of eight functions $F(t)$, as shown in the Eq. 1, where the variable t indicates the time with a sampling interval of $\Delta t = 10^{-3}$. The values of the variable a have been randomly generated by selecting a value from the normal distribution with average 0 and variance 5, and the absolute value was used. The reason for that is, if a becomes a negative value thus it becomes impossible to use the logarithmic function. In the same manner, we opted to use the ε when $t = 0$, to avoid the function being infinite. When calculating the derivative values, by using the numerical differentiation for the second-order approximation, the error bounds can be estimated by the $O(\Delta t^2)$. Therefore, if we use $\varepsilon = 10^{-6}$ thus we can consider that the errors will not extrapolate the error bounds range. The generated synthetic data can use a maximum of two from the set of aforementioned functions, and the weight for the linear sum will be randomly selected from a normal distribution with an average of 0 and a variance of 5. The complexity of the measured data will be expressed by these coefficients (a and weight of linear sum).

$$\begin{gathered} log(at+\varepsilon),\ e^{at},\ \sin(at),\ \cos(at), \\ (at+\varepsilon)^{-2},\ (at+\varepsilon)^{-1},\ at,\ (at)^2 \end{gathered} \tag{1}$$

The number of combinations for the data being represented by the synthetic data set can be calculated as follows: $({}_8C_1)$ is the number of combinations for the data being generated by using each of the eight functions; $({}_8C_2)$ is the number of combinations for the data being generated by using simultaneously two functions. As a result, the total number of combinations becomes 36, that is $({}_8C_1 + {}_8C_2)$. We generated a set of 100 data by changing the coefficients for each of the combinations. Therefore, a total of 3,600 sets for both training data and test data has been generated.

2.1.1 Training Data Set. The output value of the network for each Eq. 1 is the probability composing of functions to express formula of the data. When the function is used for a given input data, then the ideal output of the network will be 1 (100 %). On the other hand, the ideal output for non-utilized functions is 0 (0 %). Therefore, we set each of the ideal outputs as the training data sets.

2.2 Data Pre-Processing

To be able to use the Convolution Neural Network (CNN) technique, it is necessary to convert the generated data into image data (matrix). In this section, we will explain the utilized numerical differentiation method, and will detail the conversion process to the image data.

2.2.1 Numerical Differentiation. We utilized the central difference scheme for the second-order derivatives to approximate the derivatives. During the approximation process, the data placed in the extremities cannot be approximated in the same way thus they were no used in the data learning. In this paper, we used up to the fourth-order derivatives.

2.2.2 Image Conversion. In addition to the function value, we used four derived functions from the first to the fourth-order derivatives, and as a result, we are able to use up to five different data sets. For each image conversion, we select two sets from them, and the respective values are assigned to the x and y axes of the image. Data with the lower order derivative is assigned to the x axis, and these vertical and horizontal axes are not replaced during the learning process. As the input data for CNN, a set of 10 (${}_5C_2$) images is generated. The function and derivative values assigned to the x and y axes indicate the pixel positions, which the points will be plotted. The distribution of the points within the image represents the data pattern. Since the input image size to CNN is fixed, different maximum or minimum values for the functions and derived functions are not allowed. Therefore, the assigned function and derivative values are separately normalized. The data is normalized within the minimum and maximum values to become [0, 1].

However, there may be cases where the data contains "outliers" that are largely out of the average value of the data. We used ε for functions where the output can become infinite, such as the logarithm function. However, there still can be abnormal values. When there is an abnormal value in the data, the value much smaller than the abnormal value becomes a nearly small constant value via data normalization, and as a result, this feature does not appear in the input image to the CNN. To avoid the influence of this small variation in the data learning process, the input data pass through the arctan filter before the data normalization in order to maintain the abnormal value in the same order.

3 EXPERIMENTS

This section will detail the data learning process, the utilized evaluation method, and the experimental setup. After that, we will discuss the experimental results. Experiments were performed by using Chainer 2.1.0 on Ubuntu 16.04 LTS.

3.1 Data Learning Process

The data learning was performed by using the learning data, described in the section 2.2.2, as the input data. We also used the training data set described in the section 2.1.1. The data learning was performed by using the mini-batch back propagation training method, and the number of mini-batch was set at 100. The error function uses the root mean square approach, and weight update method, proposed by Adam [Kingma and Ba 2014], was used.

3.2 Evaluation Method

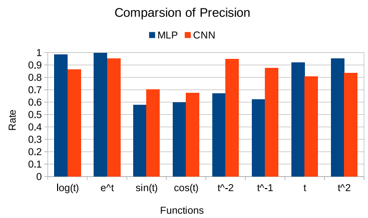
The experiments were performed by using the test data explained in the section 2.1 in order to confirm how accurate the proposed method can estimate the functions of analytic formula. When an output value is equal or larger than a given threshold value, it is judged that the function has been used. The threshold value is tested using the data used during the training, and the threshold value with the highest F-score (F-score will be described later) is selected. The precision ratio, the recall ratio, and the F-score are used for the performance evaluation. The precision ratio (positive predictive value) is the proportion of the data where the functions were estimated to be used among the data set where these functions were actually used. The recall ratio is the proportion of the estimated data and the data where the functions were used and the data where the functions were supposed to be used. From these aforementioned results, we calculate the harmonic mean between them as the F-score.

3.3 Experimental Setup

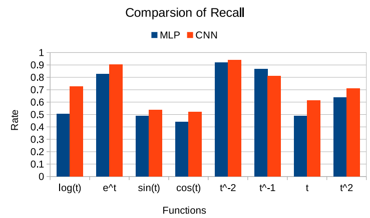
We executed two different kinds of experiments. In the first experiment, we compared the estimated results between the Multi-Layer Perceptron (MLP) and CNN, where the data that includes outliers has been used, in order to confirm the usefulness of the CNN in our proposed goal. In the second experiment, the data with and without outliers are compared by using the CNN, in order to confirm the estimation result when the data has no outliers. For the test data, the synthetic data described in the section 2.1 was used.

3.4 Experimental Results and Discussion

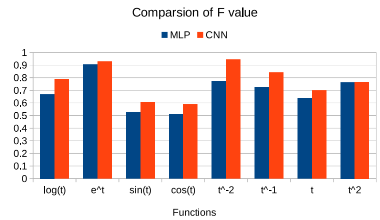
In this section, we will present and discuss the experimental results of experiments 1 and 2. Figure 1 shows the estimation results for the experiment 1. Comparing with the estimation results of CNN and MLP, we can verify that the result of CNN does not exceed that of MLP in all functions of regarding precision and recall ratios. However, for the case of F-score, we can verify that the CNN could obtain better results than MLP for all functions. In common with all graphs in Fig.1, the estimation results of trigonometric functions are lower than other functions. However, CNN could obtain good estimation results for the $\sin(t)$ and $\cos(t)$ functions regarding both precision and recall ratios. As a result, we can consider that the CNN is superior to the MLP. The experimental results for the second experiment is shown in Fig.2. In Fig. 2 (a), we can verify that the precision ratio when including the outliers is higher than that without including them for estimating the t^{-2} and t^{-1}, but not for estimating the $\sin(t)$ and $\cos(t)$ in Fig. 2 (b). We can verify that the F-score, without including the outliers, has higher overall results. Therefore, we can conclude from these experiments that the CNN,



(a) Precision

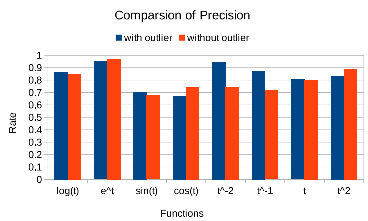


(b) Recall

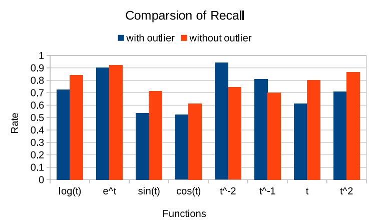


(c) F-Score

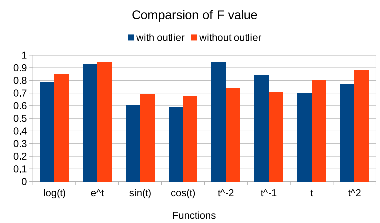
Figure 1: Difference of estimation results between MLP and CNN.



(a) Precision



(b) Recall



(c) F-Score

Figure 2: Comparison of estimation results between with and without outlier.

without including outliers, becomes preferable when employing an estimation method.

4 CONCLUSION

We proposed a method to estimate functions of analytical expressions by using data value and its derivatives, and we performed some experiments for the validation. The test data was initially transformed into image data, and then passed to CNN in order to estimate the function forms. By comparing with MLP, we confirmed that CNN is useful for function estimation. In addition, by eliminating the outliers via the arctan filter, it was also possible to further improve the accuracy of the function estimation.

REFERENCES

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). arXiv:1412.6980 http://arxiv.org/abs/1412.6980

Georg Martius and Christoph H. Lampert. 2016. Extrapolation and learning equations. *CoRR* abs/1610.02995 (2016). arXiv:1610.02995 http://arxiv.org/abs/1610.02995

Michael Schmidt and Hod Lipson. 2009. Distilling Free-Form Natural Laws from Experimental Data. *Science* 324, 5923 (2009), 81–85. https://doi.org/10.1126/science.1165893 arXiv:http://science.sciencemag.org/content/324/5923/81.full.pdf