Estimation of Functions Representing Data Using Convolution Neural Network

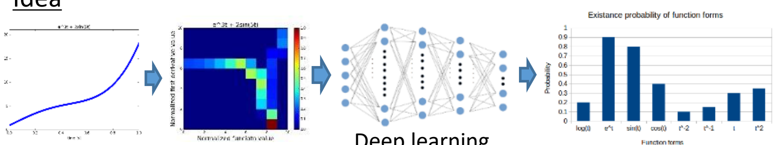
Issei Koga and Kenji Ono , Kyushu University , Contact Email : 2IE17031W@s.Kyushu-u.ac.jp

Motivation

There are several data analysis methods which do not need any previous knowledge regarding the data, such as statistical analysis and symbolic regression. However, the task to estimate a suitable model which best explains the data by applying, for instance, the symbolic regression can be a hard work. We focused on the fact that, if the function form of the terms in a given formula is possible to be known in advance, the estimation task may become easier than by using the standard symbolic regression. For example, in the formula "$e^{3t} + 2\sin 5t$", the function forms are "e^{t}" and "$\sin t$".

In this work, we propose an estimation method to find the suitable functional forms by using the Convolutional Neural Network (CNN).

Idea



Learning Data → image conversion → Deep learning → Existence probability of the function forms

Learning Data

In the experiments, we used a set of time series data, with sampling interval ($\triangle$t) of 10^{-3}, as the target data.

Data create method

The learning and test data were both generated by the linear sum of the function forms, used solely or in combination. The used forms are *log(at+ ε)*, e^{at}, *sin(at)*, *cos(at)*, $(at+\varepsilon)^{-2}$, $(at+\varepsilon)^{-1}$, at, and $(at)^2$, where ***a*** was set to |N(0,5)|, and **ε** to 10^{-6} to prevent infinite value.

Numerical differentiation

The second-order central difference approximation was used, and the order of the error is proportional to $\triangle t^{2}$ (= 10^{-6}) . We consider that there is no influence in adding O(10^{-6}) to the function form.

Architecture of Deep learning

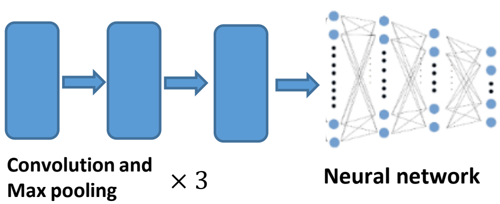


Image Conversion

We employed five kinds of data, i.e., the data value itself and its derivatives (from first to fourth derivatives), and obtained a set of 10 (${}_5\mathrm{C}_2$) images.

Graph generation (Fig.1 (a))

- Select two data -> x and y plot positions
- Apply *arctan* to eliminate outliers.
  Outliers: data averaging -> same order
- Data normalization [Max., Min.]
  -> [1 , 0] (because image size is fixed)

Image conversion (Fig.1 (b))

- The number of plots inside each of the grids in the graph
  -> converted as the pixel value

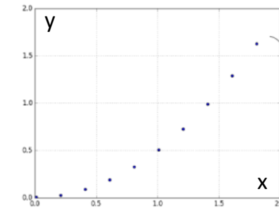


Fig.1 (a) Graph (data)

0	0	0	1
0	0	0	1
0	0	3	0
3	2	0	0

Fig.1 (b) Corresponding Image

Evaluation of the Estimation Accuracy

Method

We evaluated the output of CNN when using the test data (described in the Learning Data section). Experiment 1 shows a comparison with the Multi Layer Perceptron (MLP). Experiment 2 shows the results when including the outliers, and excluding the outliers (by applying the *arctan*) .

Evaluation Metrics (Tab. 1)

Precision[1], recall[2] and F-score (harmonic mean of Precision and recall)

Tab. 1 Output Evaluation

	Real existence	
Estimation	True	False
True	TP	FP
False	TN	FN

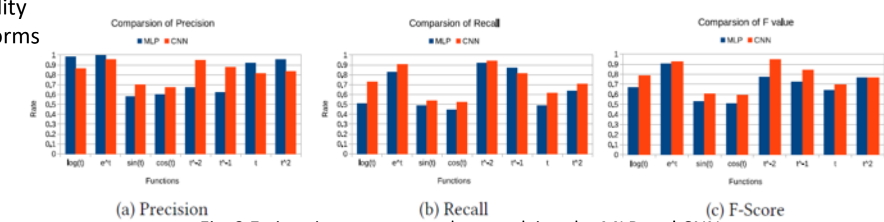
[1] Precision = $\frac{TP}{TP+FP}$

It is the truly correct rate among the results estimated as correct.

[2] Recall = $\frac{TP}{TP+TN}$

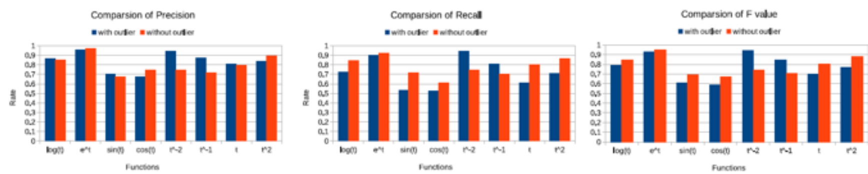
It is a rate that can be obtained within the set of correct results.

Results



(a) Precision (b) Recall (c) F-Score

Fig. 2 Estimation accruracy when applying the MLP and CNN.



(a) Precision (b) Recall (c) F-Score

Fig. 3 Estimation accuracy when including and excluding outliers.

Conclusions

- The estimation accuracy of the CNN was better than MLP (from the results of experiment 1).
- The estimation accuracy was improved when excluding the outliers (from the results of experiment 2).
- Through the experiments, we verified that the estimation accuracy of trigonometric functions was lower than other functions, thus the proposed **estimation method requires some improvements** to better estimate the most suitable functional forms.