

현재 PyScript의 실제 응용 가능성

은 진^o, Scott Uk-Jin Lee^{*}

^o한양대학교 컴퓨터공학과,

^{*}한양대학교 컴퓨터공학과

e-mail: thefrozensnow@naver.com^o, scottlee@hanyang.ac.kr^{*}

Analyze the possibility of current PyScript in practical application

Zhen Yin^o, Scott Uk-Jin Lee^{*}

^oDept. of Computer Science and Engineering, Hanyang University,

^{*}Dept. of Computer Science and Engineering, Hanyang University

● 요약 ●

With the advent of PyScript, there are more and more topics in practice. However, the official guideline has not provided a detailed description. This paper investigates the practical application about PyScript. The research results show that the current functions in PyScript cannot be well handled in web pages, and even cannot use the advantage model of Python, which is not suitable for practical application in the short term. However, it may be widely used in web development in the future since writing functions through Wasm can improve the efficiency of program execution.

키워드: PyScript, WebAssembly

I. Introduction

PyScript is a javascript based framework released by Anaconda in 2022, which allows developers to execute Python code in browsers. PyScript aims to utilize more benefits from the powerful functions of Python and various third-party libraries [1]. However, Pyscript can only call libraries that exist in Pyodide at moment [2]. It means PyScript cannot use large amount libraries like python, which limits the scope of practical application of PyScript. Since it is still in the exploratory stage, this paper investigates the value of PyScript in the current practical application and discusses relevant suggestions.

II. Related works

Hanley believes that writing WebAssembly (Wasm) templates and integrating them into PyScript can improve efficiency in browsers. This is very important for the development of PyScript, because the execution speed of PyScript in browser is too slow at present [3]. Ma. et al studied and analyzed the efficiency

of deep learning frameworks written in Python and Javascript. The results show that the TensorFlow framework has the highest efficiency in browsers [4]. However, PyScript cannot implement Tensorflow framework in the web like JS, that means PyScript will not be used to implement deep learning projects in the web at moment.

III. The Proposed Scheme

First of all, we carefully reviewed the announcements, test discussions and related application demonstrations about PyScript on anaconda and Github, and summarized the following points:

1. Currently, Pyodide does not support the popular functions in Python, such as crawler and game making.
2. PyScript can only use scikit-learn machine learning library at present, and popular deep learning frameworks, such as Tensorflow, cannot be executed in the browser through PyScript at present.

3. PyScript takes a long time to execute in the browser, which is not suitable for the development of practical applications.

Then we further analyze second and third points. Because PyScript is a browser dependent language, we compare it with JavaScript (JS).

We accroding to the execution logic of the PyScript official demo to implement with JS, and compare them. We found that even the slowest mind framework in JS framework [4] is faster than PyScript in language recognition and image recognition. In addition, when the page was loaded successfully and the response operation was carried out in the page, the browser would stop running when the CPU utilization of the content loaded by PyScript was too high, which fundamentally led to the fact that PyScript could not be used in practical applications at present. To improve the execution speed of PyScript, according to John Hanley's suggestion [3], currently ,among the languages supported by Wasm, C, C++, and Java are the languages with high utilization rate, so we use them to write a Wasm file to calculate Markov Model, that object allows Python access to the Wasm exports The running time after execution on the web is shown in the Fig. 1.

Through experiments, we found with the increase of observation sequence, the Wasm function written in C has the highest execution efficiency in browser. Because deep learing needs to deal with a huge amount of data, we believe that if PyScript want to improve the application ability of AI in the browser, developers should use C language to write Wasm and integrate it into PyScript.

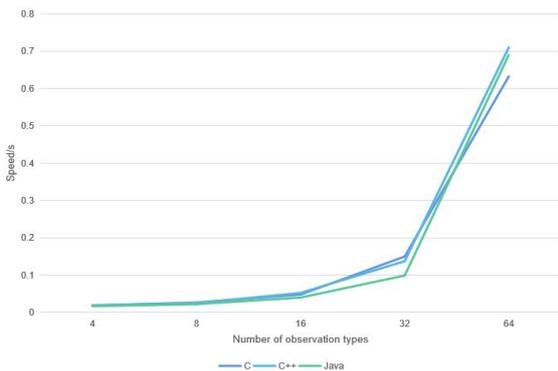


Fig. 1. Execution speed of wasm written in different languages

IV. Conclusions

This paper analyzes the current practical application value of PyScript and provides some suggestions for developers to improve PyScript, and provides suggestions for programmers who want to use PyScript for web development.

ACKNOWLEDGEMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-01343, Artificial Intelligence Convergence Research Center (Hanyang University ERICA)).

REFERENCES

[1] <https://pyscript.net/>
 [2] <https://pyodide.org/en/stable/>
 [3] <https://www.jhanley.com/pyscript-interfacing-with-wasm/>
 [4] Ma, Y., Xiang, D., Zheng, S., Tian, D., & Liu, X. Moving deep learning into web browser: How far can we go?. In *The World Wide Web Conference. May, 2019.* pp. 1234-1244.