



(RESEARCH ARTICLE)



A novel approach for federated machine learning using Raspberry Pi

Barida Baah ^{1,*}, Onate Egerton Taylor ² and Chioma Lizzy Nwagbo ³

¹ Department of Computer Science, Ebonyi State University, Abakaliki- Nigeria.

² Department of Computer Science, Rivers State University, Port Harcourt-Nigeria.

³ Department of Computer Science and Robotics Education, College of Education, Nsugbe, Anambra State, Nigeria.

Global Journal of Engineering and Technology Advances, 2021, 06(03), 063–068

Publication history: Received on 01 February 2021; revised on 05 March 2021; accepted on 08 March 2021

Article DOI: <https://doi.org/10.30574/gjeta.2021.6.3.0042>

Abstract

The problems of privacy and security is becoming a major challenge when it comes to the distributed systems, federated machine learning system especially when data are been transmitted or learned on a network , this necessitated the reasons for this research work which is all about wireless federated machine learning process using a Raspberry Pi. The Raspberry Pi 4 is a single hardware board with built in Linux operating system. We used data set of names from nine (9) different languages and then develop a training model using recurrent neural network to train this names compare to the names in the existing language like French, Scottish to predict if the names are from any of this language, this is done wirelessly with the Wi-Fi network in a federated machine learning environment for experimental setup with PySft's that is installed in the python environment. The system was able to predict that name from which the language it originate from, the methodology that is implore in the research work is the Rapid Application Development (RAD). The benefits of this system are to ensure privacy, reduces the computing power, ensure real time learning and most importantly it is cost effective.

Keyword: Wireless; Federated Machine Learning; Raspberry Pi

1. Introduction

Applied learning strategies include integrating training data into a specified database or database. For example, if the first e-commerce company wants to set up its customer information module for purchasing its products, it would run the information on the data collected from the application or website. The data may be related to the time spent on the actual product page Products that have been searched but not purchased and products that have been purchased. The data should be sent and received through an encoder or data source [1].

Although the comparisons could be quite simple, computer history could substitute for what Federated Learning is all about. There has been a huge gap in the early days of information technology, making the most advanced computer simulations. Finally, the researcher switched to a remote system where computers were distributed between client computers, clients, and internal servers [2].

The Federated Learning structure uses the same model. Machine learning machines are distributed over the systems of computer equipment, rather than on large, centralized systems. This computer model, while initially thought of, would not have been hand-made, as the mobile computer would be too slow to run any Machine Learning module.

Consequently, learning improved in the near-to-late 2018s. Since billions of downloads, equipped with AI chips and high-capacity computers, starting with the Samsung S9, or Apple X series, some of the machine learning(ML) models should run in that and focus on such phones, delivered in the next 3-5 years.

* Corresponding author: Barida Baah
Department of Computer Science, Ebonyi State University, Abakaliki- Nigeria.

2. Review of Related Works

Heyam H. Al-Baity et al [3] proposed an effective web service discovery using feature selection, it aims is to employ feature selection in order to search for subset characteristics that can increase the accuracy classification and also enhanced the pace of the traditional classifiers using machine learning.

Laachemi et al [4] apply the use of support vector machine with the classification that improved on its accuracy level based on a stochastic local search (SLS), at the initial stage, the web service quality dataset attributes utilizes the support vector machine classifier to find solutions that is optimal as they have been scaled and its values resized. SLS-SVM approach has an accuracy which was 84.86%, that is far better than the SVM with similar classifier.

Liu et al. [5] proposed the integrated of SVM classifier and the latent Dirichlet allocation (LDA) based models which is used big-scale services to classify and reduced the cost of labeling manual services for training. Algorithm of LDA was used to remove high-level topics from services. The base classifier for this technique is the SVM was used due to the facts that its does well on the classification text. Web services classifier is an important aspect based on its descriptions. Furthermore, with the introduction of a database-based learning strategy that is active was used to reduced manual labeling services cost that is needed to build and trained the set. Manually labeling of services was done in the preprocessing stage, with descriptions information based on its capabilities. Their research takes a looked at the groups with high numbers of services and then select 10 groups. These resulted to increase in the service number of the LDA-SVM which provides the results that are more accurate compare to the SVM. The efficacy of classification frame work in its active learning service was clearly shown in their experimental result that was obtained.

Mustafa et al. [6] opined multi-layer perception neural network (MLPNN) novel classification model which applies optimization through search tabu. It utilizes multi-layer perception neural network model to inspire the neuroscience which was utilize in the prediction of process. MLPs were utilized back propagation in conjunction with Levenberg-Marquardt algorithm that trained the multi-layer perception classifier and TS in the optimization of the classifier. The quality of web-service dataset is like the dataset used [4]. The MLP-TS demonstrated better results of experiment.

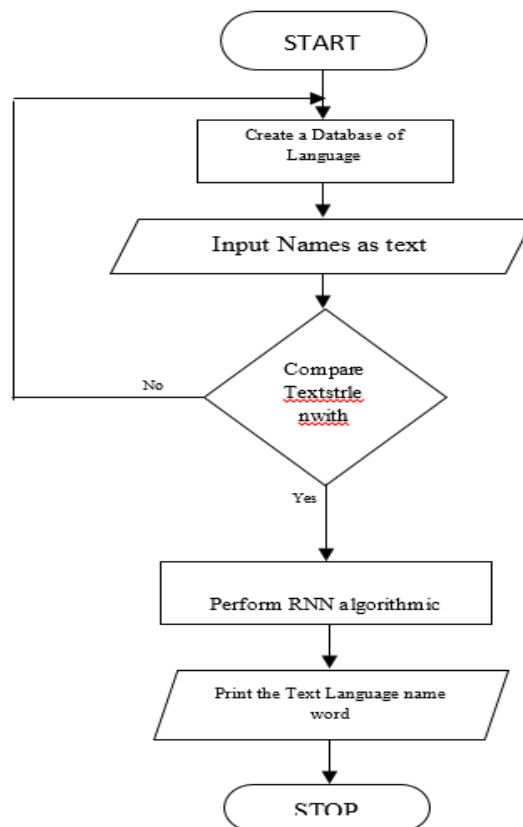


Figure 1 A Proposed Flowchart for the federated learning of text classification

3. System Design

In this research paper work we will define all the technical and functional requirements. All the software and hardware specifications that are used must be setup in this phase. In starting this design phase, we get the initial overview of the processes of using the software. For this project, Raspbian will be used as operating system for Raspberry Pi and coding will be done in python.

3.1. Experiment Setup

In the experiment set up section, we present the setting of the experiment process for hardware and software process on collected datasets. The Federated Learning was first implemented on the laptop (coordinating server) before using the Raspberry pi. A Raspberry PI 4 running Raspbian 20.04 connected to the internet via SSH and Wi-Fi network with each PI having is configured with a static IP address configured on them for convenience of addressing. The static IP address are 192.168.0.77 and 192.168.0.78 respectively. Figure 2 show a laptop running on Ubuntu Linux connected to the same LAN with the Raspberry Pi via Wi-Fi



Figure 2 An experiment setup of the system

Figure 2 shows the successful experiment setup for the two Raspberry Pi 4 installation process

```

pi@raspb01: ~
File Edit View Search Terminal Help
pi@raspb01:~$ python3
Python 3.6.7 (default, Apr 17 2019, 17:04:02)
[GCC 6.3.0 20170516] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> import torch
>>> print(torch.__version__)
1.0.0a0+db5d313
>>>

```

Figure 3 Successfully imported Pytorch in Python

Figure 3 above shows the successful installation of imported Pytorch in Python environment

3.2. Training the Recurrent Neural Network on Raspberry Pi

The Recurrent Neural Network will be trained for classification of a person's surname to its most likely language of origin in a federated way, using workers sever running on the two Raspberry PIs that are already equipped with python3.6, PySyft, and Pytorch. A character-level Recurrent Neural Network treats words as a series of characters, outputting a prediction and “hidden state” for every character, feeding its previous state into each next step. We then

take the final prediction, which is the output, i.e. which class the word belongs to. Hence the training process continuous sequentially in a character-by-character manner through the different hidden layers.

4. Result and Discussion

We plot the results using Matplotlib's pyplot. The plot will show us the learning rate of our recurrent neural network. The plot will show us the learning rate of our network.

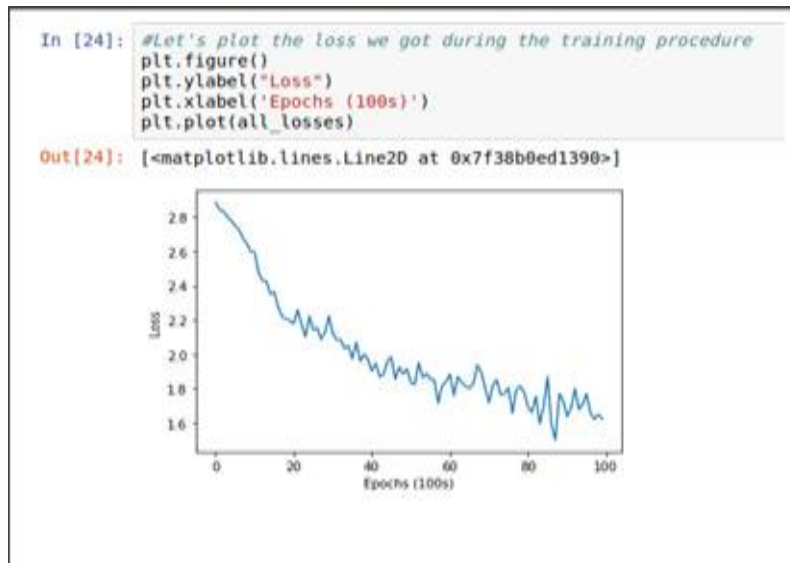


Figure 4 A Screenshot showing learning rate of the network

4.1. Testing the Trained RNN (Predicting names)

We can do inference locally by getting model back from Raspberry Pi and we can test using different models from each worker or we could also find average all the model parameter from different worker, figure 5 shows testing on two (2) separate model from two (2) Raspberry Pi ("bob", "alice"). We defined a function that take in a name and return the likely languages base on our database list created

```
In [27]: predict(model_alice.copy(), "Qing", alice)
predict(model_alice.copy(), "Daniele", alice)

predict(model_bob.copy(), "Qing", alice)
predict(model_bob.copy(), "Daniele", alice)

> Qing
(-0.93) Korean
(-0.94) Chinese
(-3.33) Dutch

> Daniele
(-1.22) Dutch
(-2.05) Czech
(-2.18) Spanish

> Qing
(-0.48) Vietnamese
(-2.33) Arabic
(-2.62) French

> Daniele
(-0.99) French
(-1.56) Russian
(-1.89) Italian
```

Figure 5 Screenshot returning likely languages of names

4.2. Evaluation Performance

We created a confusion matrix to see how our Recurrent Neural Network has performed on different categories. The bright spots off the main axis indicate that languages the guesses is incorrect while the other spots on the main axis guesses correctly as shown in the figure 6 below:

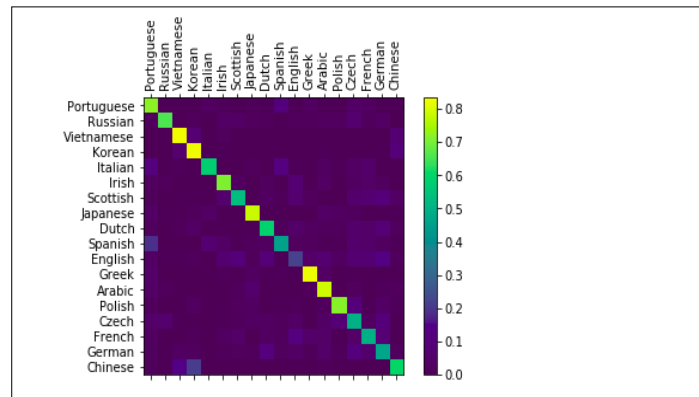


Figure 6 A screenshot of graph guess evaluation performance

5. Conclusion

In conclusion, it is usually tasks demanding to perform classification of text using Recurrent Neural Network(RNN) algorithm with long-Short Time Memory Network(LSTM) in python due to the fact that the sequence of input text or name over spaces of time is sometime vary in length with a very large vocabulary of input symbols as a result its usually require a complex model to learn the long-term dependencies or context between symbols in its input sequences. Our develop system was able to achieve the following:

A platform for text name classification to predict name in language

A platform that integrate the Raspberry Pi 4 with the preloaded Ubuntu Linux Operating System to ensure privacy of the user training model

Compliance with ethical standards

Acknowledgments

We authors are immensely grateful to the editorial board and management of global journal of engineering and technology advances (GJETA) for their speedy and rigorous process of review and publication of articles.

Disclosure of conflict of interest

There is no conflict of interest among all authors.

References

- [1] F Rottenberg, T Choi, P Luo, CJ Zhang, AF Molisch. IEEE Transactions on Wireless Communications 2020; 19 (4):2728-2741
- [2] Romano Fantacci, Benedetta Picano Federated learning framework for mobile edge computing networks. CAAI Trans. Intell. Technol. 2020; 5(1):15-21
- [3] Heyam H Al-Baity, Norah I Al Showiman. Towards Effective Service Discovery using Feature Selection and Supervised Learning Algorithms, *International Journal of Advanced Computer Science and Applications (IJACSA)* 2019; Vol. 10 No. 5:pp. 191-200

- [4] A Laachemi and D Boughaci, “A stochastic local search combined with support vector machine for Web services classification in International Conference on Advanced Aspects of Software Engineering (ICAASE), Constantine, 2016; pp. 9–16.
- [5] J Liu, Z Tian, P Liu, J Jiang, and Z Li “An Approach of Semantic Web Service Classification Based on Naive Bayes,” in IEEE International Conference on Services Computing (SCC), San Francisco, CA, 2016; pp. 356–362.
- [6] A Syed Mustafa and Y Kumara Swamy, “Web Service classification using Multi-Layer Perceptron optimized with Tabu search,” in IEEE International Advance Computing Conference (IACC), Bangalore, 2015; pp. 290–294.