OPENFL-2

2023, Group:

Agnieszka Szynalik Dmytro Kuzmin Jakub Machalica
Mariusz Kądziela

1 Introduction

Federated Learning is a machine learning approach that enables training of models on decentralized data without the need for data centralization. In Federated Learning, instead of sending data to a centralized server for training, data remains on edge devices such as smartphones, IoT devices or other distributed devices, and the training of the model takes place locally on the devices. The updated models are then sent back to a central server, where they are aggregated and combined to create a new, improved model. This way, Federated Learning allows for collaborative model training without exposing sensitive data to third parties.

A practical study of Federated Learning involves the implementation of the technology in a real-world scenario. In such a study, a team of researchers may start by identifying a particular use case where Federated Learning could be applied. They would then need to design and develop a system architecture that allows for the distributed training of machine learning models across a network of edge devices.

Next, they would need to develop the algorithms for the Federated Learning process, including how to select the devices for training, how to distribute the data among the devices, and how to aggregate the models returned by the devices. They would also need to develop security measures to protect the privacy and security of the data.

Once the system architecture and algorithms are developed, the team would need to test the system on a dataset to evaluate its performance. They would analyze metrics such as model accuracy, convergence speed, communication overhead, and power consumption. Based on the results of the evaluation, they would refine the system and algorithms to optimize performance.

Overall, a practical study of Federated Learning involves designing and developing a system that enables decentralized machine learning, implementing the algorithms for Federated Learning, evaluating the system's performance, and refining the system to optimize its performance. Such studies are essential for advancing the field of Federated Learning and its applications in real-world scenarios.

OpenFL is a cross-platform development tool that allows you to write code once and deploy it to multiple platforms. It uses the Haxe programming language, which is a high-level, strictly-typed language that is similar to ActionScript or JavaScript.

One of the key features of OpenFL is its ability to target multiple platforms, including iOS, Android, Windows, macOS, Linux, HTML5, and more. This allows developers to create applications that can run on a wide variety of devices and platforms.

OpenFL provides a wide range of features and functionality, including support for 2D and 3D graphics, animation, physics, sound, and user input. It also provides support for various file formats, including images, audio, and video.

In addition to its core features, OpenFL also has a number of extensions and plugins available, which can be used to add additional functionality to your applications. For example, there are extensions available for integrating with third-party services like Facebook or Google, or for adding support for specific game engines or frameworks.

OpenFL has an active community of developers and users, who contribute to its development, provide support, and share knowledge and resources. The community is very helpful and welcoming, and there are many resources available, such as tutorials, documentation, and forums.

Overall, OpenFL is a powerful and flexible tool for developing cross-platform applications and games, and it is well-suited for developers who want to create applications that can run on a wide variety of platforms and devices.

2 Theoretical background/technology stack

Federated learning is a privacy-preserving machine learning approach that ensures that sensitive data remains on devices and is not exposed to unauthorized access. OpenFL provides a range of tools to manage data privacy, including encryption and differential privacy.

It relies on distributed systems, which are networks of devices that communicate with each other to perform a task. Federated learning frameworks like OpenFL are designed to manage the complexity of such systems and ensure that they work together efficiently.

Moreover, OpenFL supports machine learning frameworks such as TensorFlow and PyTorch. These frameworks enable developers to build and deploy machine learning models using Python, which is a popular programming language for scientific computing and data analysis.

The infrastructure required to deploy a machine learning application depends on several factors, including the size of the data set, the complexity of the model, and the performance requirements of the application. AWS can be a great choice for running and hosting such application. It is a popular cloud computing platform that provides a wide range of services and tools. These services include Amazon Elastic Compute Cloud (EC2), Amazon S3, and Amazon Elastic Container Service (ECS).

Technolgy stack:

- Programming language: Python 3
  Python is a computer programming language often used to conduct data analysis. OpenFL supports Python 3.

- Machine learning framework: TensorFlow
  TensorFlow is an open-source machine learning framework developed by Google. It provides a wide range of tools for building and training machine learning models, including deep neural networks, convolutional neural networks, and recurrent neural networks.

- Federated learning framework: OpenFL
  OpenFL is an open-source federated learning framework that enables organizations to collaboratively train a model without sharing sensitive information. It allows developers to build and deploy machine learning models on a distributed network of devices. It provides tools for managing data privacy, version control, and model aggregation.

- Infrastructure: Amazon Web Services
  AWS is an online platform providing cost-effective, scalable cloud computing solutions. It offers a wide range of services and tools for building and deploying machine learning models.

- Virtualization: Docker
  Docker is a platform designed to help developers build, share, and run modern applications in a portable and scalable way. It is widely used in machine learning projects to manage dependencies and ensure reproducibility.

3 Case study concept description

4 Solution architecture

5 Environment configuration description

6 Installation method

7 How to reproduce

7.1 Infrastructure as Code approach

8 Demo deployment steps

8.1 Configuration set-up

8.2 Data preparation

8.3 Execution procedure

8.4 Results presentation

9 Summary – conclusions

References