TRABAJO FIN DE GRAO
GRAO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN COMPUTACIÓN

Desarrollo de una librería para el aprendizaje federado bajo una arquitectura peer-to-peer

Estudante: Pedro Guijas Bravo
Dirección: Daniel Rivero Cebrián
Enrique Fernández Blanco

A Coruña, octubre de 2022

Resumen

En la última década, la evolución del *Machine Learning* ha sido muy próspera, necesitando los modelos más fructíferos ser nutridos por grandes volúmenes de datos. A menudo, la obtención y gestión de estos datos es complicada, siendo generalmente escasos y sujetos a medidas de privacidad. El *Federated Learning* implica un cambio de paradigma en el entrenamiento de modelos de *Machine Learning*. Este nuevo enfoque permite realizar el proceso de aprendizaje sobre datos distribuidos entre una gran cantidad de clientes.

A pesar de que esta novedosa técnica trae consigo numerosas ventajas, una de sus grandes limitaciones es la necesidad de un servidor que orqueste todo el proceso de aprendizaje, suponiendo un único punto de falla. Así mismo, se necesitará disponer de una gran infraestructura para hacer escalables estos sistemas. Para tratar de solventar estas desventajas, surgirán nuevas aproximaciones denominadas *Decentralized Federated Learning*, siendo una de las soluciones más prometedoras el uso de redes *peer-to-peer*.

Ante la inexistencia de alguna librería de soporte al *Decentralized Federated Learning*, en este proyecto, se propone el desarrollo de una librería de propósito general que permita el ***Federated Learning* sobre redes *peer-to-peer*, empleando el protocolo *Gossip***. El uso del protocolo *Gossip* garantizará la tolerancia a fallos en la red *peer-to-peer*, creando un ecosistema descentralizado, escalable y robusto.

La librería busca dar soporte a toda clase de dispositivos, haciendo especial hincapié en su facilidad de uso y ampliación futura. Además de permitir el despliegue, ésta, posibilita la ejecución de simulaciones, haciendo posible la realización de pruebas en entornos controlados.

Para el desarrollo, se ha hecho uso de la metodología ágil *SCRUM*. Las iteraciones centrales se han destinado a la implementación del sistema, mientras que la inicial y final se han dedicado a la preparación del proyecto y realización de pruebas respectivamente. En las diversas pruebas realizadas, se han empleado los *datasets* de *MNIST* y *FEMNIST*, obteniendo resultados realmente similares a ejecuciones equivalentes con entrenamientos clásicos.

Índice

1. Introducción

En la actualidad, la Inteligencia Artificial se ha convertido en uno de los campos más prometedores de la informática. Una de sus principales ramas, el *Machine Learning* (ML), es cada vez más popular y empleada, siendo sus impresionantes avances fruto de modelos más y más complejos, nutridos por grandes cantidades de datos.

La forma convencional de trabajar en ML consiste en la centralización de información y cómputo en un mismo lugar, implicando limitaciones. Una limitación sería relativa a la infraestructura, puesto que el mantenimiento de la información y los diferentes procesos de aprendizaje requieren de grandes recursos. Otra de las grandes limitaciones se centra en la naturaleza de la información. Los datos provienen de una gran variedad de fuentes, estando sujetos usualmente a medidas de privacidad que incluso pueden llegar a impedir su difusión. De acuerdo con estas medidas, se está generando un potencial riesgo de privacidad al usarse este enfoque centralizado para procesos de aprendizaje.

De este modo, en 2016, Google propone el *Federated Learning* (FL) [1], en español, Aprendizaje Federado. Este nuevo enfoque pretende solucionar los problemas expuestos anteriormente gracias a la creación de modelos sobre datos distribuidos. Su funcionamiento es simple, consiste en un proceso iterativo orquestado por un servidor central. En cada iteración, diferentes nodos contribuirán a un modelo global por medio de la agregación de modelos locales. Es decir, para obtener los modelos locales, los nodos realizarán paralelamente entrenamientos empleando sus propios datos y recursos.

A pesar de las evidentes ventajas que otorga no compartir explícitamente los datos, el servidor orquestador también implica una serie de limitaciones. Estas consisten en la existencia de un punto único de falla y un posible cuello de botella, limitando considerablemente la escalabilidad. Para solucionar estos problemas, se busca que los nodos consigan comunicarse entre ellos, prescindiendo del servidor. A este conjunto de aproximaciones se las denomina Aprendizaje Federado Descentralizado [2].

El uso de redes *peer-to-peer* (p2p) y protocolos *Gossip* [3] es una de las aproximaciones más prometedoras al Aprendizaje Federado Descentralizado. La arquitectura p2p, permitirá la creación de una infraestructura donde los dispositivos o nodos se comporten de forma colaborativa entre iguales. En cuanto al protocolo *Gossip*, permitirá crear un mecanismo robusto para el intercambio de mensajes, aprovechando la redundancia de las redes p2p. De esta forma, se obtendrá un ecosistema descentralizado, que garantiza tanto la escalabilidad como la robustez. Destáquese que la tolerancia a fallos resultará crucial en entornos reales, sobre todo si se desea dar soporte a dispositivos *edge*, los cuales pese a disponer de cada vez más poder de cómputo, suelen verse afectados por problemas de disponibilidad relacionados con el uso de baterías y redes de telefonía. Lamentablemente, la mayoría de trabajo relacionado con esta aproximación está enfocado a la investigación, no existiendo librerías que permitan la fácil creación de dichos sistemas.

En consecuencia, motivado por la inexistencia de una librería de propósito general para el Aprendizaje Federado Descentralizado, en este proyecto se llevará a cabo el desarrollo una implementación de este tipo. Al ser una librería de carácter general, tratará de darse soporte a toda clase de dispositivos, buscando la ligereza del sistema centrada en el soporte a dispositivos de bajas prestaciones. Así mismo, no se establecerá una topología en específico, tratando que el usuario decida la más adecuada para su siste-

ma. En lo tocante a las comunicaciones, se tratarán de minimizar, incluyendo métodos criptográficos para garantizar la confidencialidad y confiabilidad en las mismas.

Buscando la comodidad y flexibilidad, la librería estará orientada al usuario, siendo completamente agnóstica a los *frameworks* de ML existentes, procurando la fácil inclusión de estos. Concretamente, este proyecto se centrará en las Redes de Neuronas Artificiales (RR.NN.AA), puesto que el Aprendizaje Federado apenas se ha explorado con otra clase de modelos.

Asimismo, para poder validar y probar los sistemas creados, la librería no solo estará orientada al despliegue, sino que también se contemplarán los entornos simulados. La simulación facilitará el desarrollo y uso de la librería, permitiendo una sencilla transición hacia el despliegue.

Finalmente, se llevará a cabo un ejemplo práctico real en donde se pueda validar y demostrar el funcionamiento y cualidades de la librería.

1.1. Objetivos

El objetivo de este proyecto será la **creación de una librería que dé soporte al Aprendizaje Federado sobre redes p2p, haciendo uso del protocolo *Gossip***. Para poder cumplir con este objetivo, se han identificado los siguientes sub-objetivos concretos que se deben abordar para poder llevar a cabo el primero:

- **Estudio:** Estudio de Aprendizaje Federado, redes p2p y protocolos *Gossip*.
- **Simplicidad y Facilidad:** Se tratará de llevar estos dos principios, tanto al desarrollo de la librería, como al uso de la misma por parte del usuario final.
- **Modularidad y Abstracción:** Conceptos que junto con los anteriores facilitarán la ampliación de funcionalidades, tanto para los desarrolladores iniciales como futuros.

  Un diseño basado en estos dos conceptos permitirá entre otras cosas, la fácil incorporación de nuevos *framework* de ML y algoritmos de agregación federados.
- **Robustez:** Se ha de tener en cuenta la posibilidad de fallos en los nodos, controlando su ocurrencia sin perjudicar al sistema.
- **Privacidad:** No será suficiente la privacidad por diseño del FL, puesto que se podría realizar ingeniería inversa. Por lo tanto, la librería debe proporcionar una capa de privacidad adicional.
- **Investigación y Despliegue:** Se pretenderá que la librería sea apta para estos dos tipos de usuarios, permitiendo una sencilla migración hacia el despliegue.
- **Pruebas:** El sistema desarrollado deberá ser cautelosamente testeado y probado, tratando de acercarse a entornos reales.

1.2. Viabilidad del proyecto

Como en cualquier proyecto, será necesario realizar un análisis que ayude a determinar la decisión de si llevarlo a cabo, o no. El uso del análisis Fortalezas Oportunidades

Debilidades Amenazas (FODA) [4], permitirá buscar y analizar, de forma proactiva y sistemática todas las variables que intervienen en el proyecto, con el fin de determinar la viabilidad del mismo.

De este modo, la figura 1 muestra la matriz FODA analizando la situación del proyecto. A grosso modo, las fortalezas y oportunidades estarán asociadas a lo novedoso y útil que puede resultar el sistema. La utilidad del sistema se ve potenciada teniendo en cuenta el auge de la Inteligencia Artificial en todos los sectores de la sociedad. En cambio, las debilidades y amenazas están sujetas nuevamente al riesgo que implica la realización de un producto novedoso, como pudiera ser la falta de información y la incertidumbre ante la acogida de los posibles clientes y comunidad la *open source*.

En consecuencia, se ha concluido que el proyecto es apto para ser realizado.

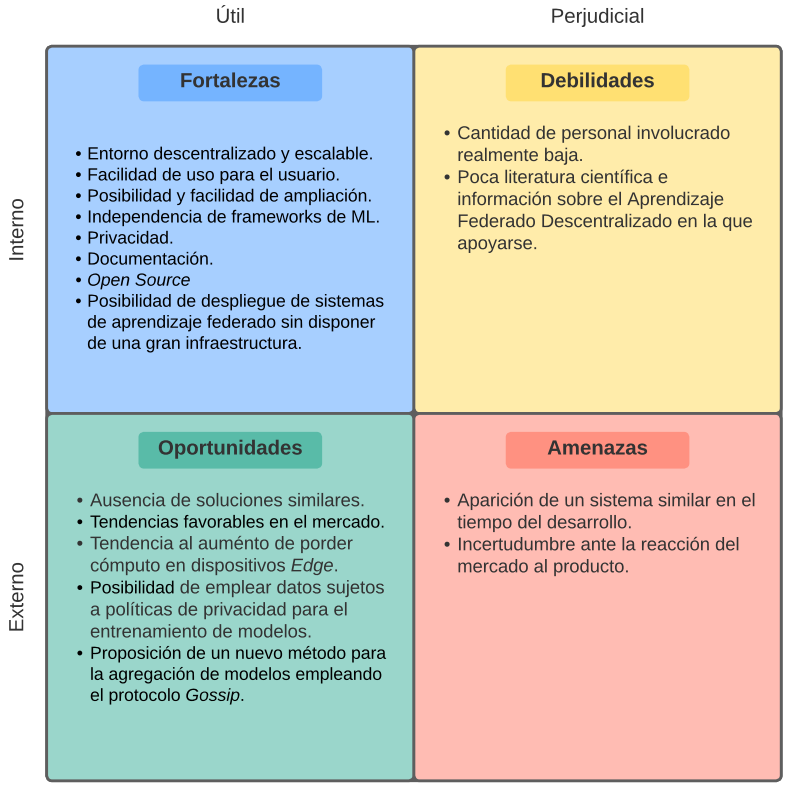

Figura 1: Matriz resultante del análisis FODA del proyecto

2. Software libre

Esta sección se centrará en la importancia del *software* libre en este proyecto. Abordando la justificación del mismo, el tipo de licencia empleada y las herramientas *open-source* usadas.

2.1. Justificación

El código ha sido publicado bajo una licencia de código abierto esencialmente por los siguientes motivos:

- **Motivación personal**: El código abierto es algo de lo que todo el mundo se puede nutrir, tanto empleando el *software* en sí mismo, como de forma educativa, leyéndolo y comprendiéndolo. De este modo, el autor del proyecto, además de estar de acuerdo con los ideales *open source*, se siente en deuda con la comunidad. Por consiguiente, la librería buscará ayudar a un sector de la Inteligencia Artificial apenas explotado.
- **Servicio comunidad**: Permitir a la comunidad científica, de manera libre, la inclusión de modificaciones sobre la librería con el fin de aumentar sus funcionalidades. Esto, es especialmente importante ya que al tratarse de una tecnología tan novedosa, esta se encontrará en un continuo desarrollo.

  Así mismo, el resto de comunidad también podrá contribuir de otras múltiples formas, como la detección de *bugs* o la aportación de ideas y características deseadas. El método para la contribución será detallado en la sección 4.1.
- **Transparencia**: Uno de los aspectos más importantes, pues la librería seguramente trate datos sensibles, siendo posible para cualquiera validar de forma directa el tratamiento de los mismos.

2.2. Herramientas y librerías empleadas

Se ha empleado el lenguaje de programación *python* [5], con las librerías que se muestran en la tabla 1. Como se puede apreciar, la librería desarrollada se enmarca dentro de un entorno donde el software libre tiene una gran importancia.

Librería	Licencia
Threading y *Sockets* (propias de *python*)	*Python Software Foundation License (PSFL)* [6]
Pytest	*MIT License* [7]
Sphinx	*BSD 2-Clause license* [**?**]
Pycryptodome	*BSD 2-Clause license*
Pytorch	*BSD-style license*

Cuadro 1: Licencias presentes en las librerías empleadas.

2.3. Licencia

Se han planteado múltiples licencias como *Berkeley Software Distribution (BSD)*, *Massachusetts Institute of Technology (MIT)* o *Apache* [8], no obstante la licencia empleada en este proyecto ha sido ***GNU General Public License v3 (GPLv3)*** [9]. El principal motivo de elección de *GPLv3* ha sido que es algo más restrictiva que las demás candidatas. Así mismo, uno de sus puntos más importantes es que se trata de una licencia *copyleft*, es decir, se exige que se preserven las mismas libertades sobre copias y derivados.

Admás, debe destacarse la compatibilidad de *GPLv3* con las licencias de las librerías empleadas, las cuales han sido previamente mencionadas.

3. Desarrollo del TFG

En esta sección se tratará de forma realmente breve el desarrollo del proyecto, así como los resultados del mismo y las lineas de trabajo futuro.

3.1. Desarrollo

Como se ha mencionado, en este sub-apartado se tratará el proceso de desarrollo de forma rápida y sin centrarse en aspectos de bajo nivel. Para un análisis más detallado consúltese la documentación del proyecto o la memoria original del Trabajo de Fin de Grado (TFG) [1].

A continuación se explicará el trabajo realizado de forma muy similar a como se ha estructurado con la metodología empleada (*SCRUM* [10]), permitiendo una comprensión más simple y natural del trabajo realizado:

- **Creación de la red p2p**: En primer lugar, se ha creado un sistema p2p base sobre el que ejecutar el aprendizaje. Debe destacarse por lo tanto, el robusto diseño del sistema así como el uso de patrones para garantizar su mantenimiento y fácil ampliación. En lo tocante a tecnologías debe destacarse el uso del protocolo *TCP* [11] así como la concurrencia.
- **Inclusión de Aprendizaje Federado en la red p2p**: Con la red creada, se ha ampliado la librería para permitir la ejecución del Aprendizaje Federado en la misma. Se ha empleado el *framework open-source* de *pytorch*, no obstante, no está nada acoplado a la librería, permitiendo sencillas modificaciones, así como la inclusión de nuevos *frameworks*. En general, debe destacarse el esfuerzo en la tolerancia a fallos así como, nuevamente, la capacidad de ampliación y fácil uso de la librería.
- **Inclusión del protocolo *Gossip***: Para dotar a la red de una gran robustez ante la ocurrencia de fallos, así como permitir la comunicación entre nodos no directamente conectados, se incluirá el protocolo *Gossip*. Éste, se empleará para la difusión de mensajes y modelos, además de la agregación de modelos. Adicionalmente, en este proyecto se ha propuesto una **nueva optimización en la agregación de modelos** empleando este protocolo. Dicha propuesta supondrá un importante ahorro en el coste de las comunicaciones.
- **Cifrado en las comunicaciones**: Finalmente, para dotar al sistema con una capa de privacidad adicional, se han encriptado las comunicaciones con el cifrado simétrico *RSA* [12].

[1] Disponible en https://github.com/pguijas/federated_learning_p2p/blob/main/other/memoria.pdf

3.2. Resultados

Como resultado del desarrollo, se ha creado una librería innovadora que permite la creación de entornos de aprendizaje federado escalables y robustos, así como su simulación. Dicha librería, presenta un diseño que garantiza una gran usabilidad y extensibilidad.

Para validar y estudiar el funcionamiento de los sistemas que permite crear la librería, se han realizado una serie de experimentos. Estas pruebas, han demostrado que la evolución de los modelos es prácticamente idéntica a la obtenida con entrenamientos clásicos equivalentes [2].

A continuación, se mostrarán brevemente los resultados de uno de los experimentos realizados, pudiendo consultar el resto en la memoria oficial del TFG. En el experimento a tratar, se buscará entrenar un *MultiLayer Perceptron (MLP)* [13] con el dataset *MNIST* [14]. En cuanto a la configuración del experimento federado, se han empleado 6 nodos y ejecutado 10 rondas de 1 *epoch* [3] por nodo.

La figura 2 refleja la evolución de la precisión en el conjunto test [4], apreciándose como la precisión final y su evolución es realmente similar para ambos métodos de entrenamiento.

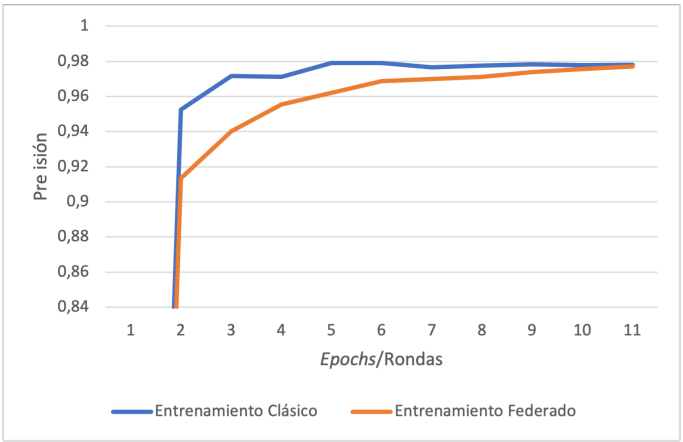

Figura 2: Comparativa de la precisión sobre el conjunto test en un entrenamiento clásico y un entrenamiento federado. Las métricas han sido obtenidas al finalizar cada ronda en un entrenamiento federado o cada *epoch* en un entrenamiento clásico. La cantidad de cómputo entre muestras será idéntica para ambas ejecuciones.

3.3. Trabajo futuro

En primer lugar, debe recalcarse que el TFG realizado únicamente han abordado los primeros pasos de lo que en un futuro será una librería más completa y compleja. Dado que es un proyecto con muy poco recorrido, puede suponerse que las líneas de trabajo futuro serán muy abundantes y diversas.

Motivado por los buenos resultados de la primera versión de la librería, se ha crea-

[2] Entrenamientos con los datos y cómputo centralizados en una misma ubicación.

[3] un *epoch* es una iteración del algoritmo de aprendizaje automático

[4] El conjunto test es un conjunto de muestras no empleadas en el entrenamiento de los modelos. Este conjunto servirá para medir la capacidad de inferencia del modelo, es decir, su rendimiento.

do un pequeño *roadmap* que detalla de forma ordenada el próximo trabajo a realizar. Destáquese que no se han especificado los periodos temporales debido a la disponibilidad realmente escasa del pequeño equipo de desarrollo recién formado. Obviamente, la inyección de liquidez al proyecto por cualquier método de financiación, así como la aparición de colaboradores derivados de la publicación del código bajo una licencia *open source*, acelerará de forma radical las primeras fases de desarrollo.

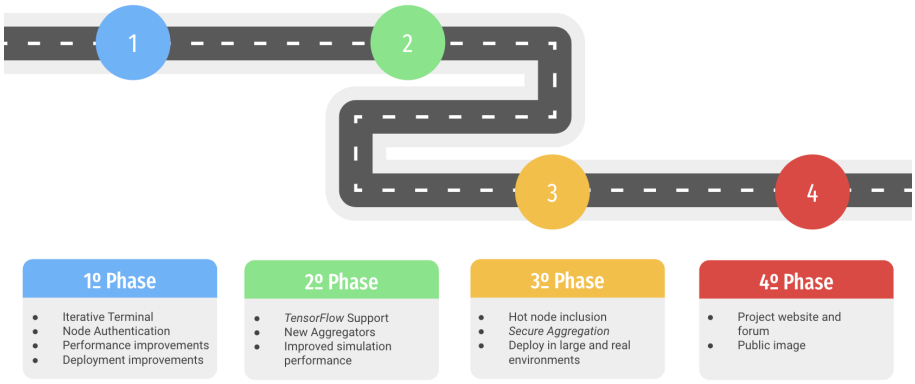

Figura 3: *Roadmap* o hoja de ruta con los próximos pasos del proyecto.

4. Actividad Pública

Como se verá en los próximos sub-apartados, la presencia pública aún no se ha ejecutado al completo. Esto es debido a que se han priorizado levemente la inclusión de una serie de funcionalidades a la librería. La decisión está motivada para hacer más eficaz la fase destinada a la actividad pública, haciendo más evidente la notoria la utilidad y funcionalidad de la librería.

4.1. Repositorio

El código fuente se encuentra en un repositorio público de *GitHub* [5], siendo el uso de *pull requests* e *issues* la forma de contribuir al proyecto. Estos, serán revisados con frecuencia, buscando tener el proyecto actualizado y mantenido. Adicionalmente, se han publicado una serie de buenas prácticas y consejos para contribuir al proyecto, buscando tener un código consistente, uniforme y estructurado.

También se ha reestructurado la forma de trabajar en el repositorio, pues se hasta el momento se ha utilizado el mismo de forma privada y unipersonal. Se emplearán ramas o *branches*, empleando la rama principal únicamente versiones estables y funcionales de la librería. La forma de actualizar esta rama principal será nuevamente empleando los ya mencionados *pull requests*.

Por último, debe mencionarse que se tratará de mantener una lista actualizada con *issues* de funcionalidades a implementar, para facilitar la colaboración por parte de la comunidad. , para incentivar la colaboración por parte de terceros.

[5] https://github.com/pguijas/federated_learning_p2p

4.2. Gestor de paquetes

Para facilitar la instalación y uso de la librería, esta se encuentra empaquetada y subida al gestor de paquetes oficial de *Python*, *PIP* [6]. De este modo, con un simple comando podrá instalarse y ejecutarse:

pip install p2pfl

4.3. Página Web

En el momento de redacción de este documento, únicamente se encuentra disponible una página web con la documentación del proyecto. Esta trata de introducir y explicar todo lo relativo a la librería, desde un inicio rápido, hasta detalles de implementación o como contribuir al proyecto.

Como se detalla en el *roadmap*, en próximas fases se creará una página web que sea la principal imagen del producto. Entre otras cosas, integrará la documentación existente, se publicarán las noticias más relevantes del proyecto (difundiéndolas también por vía *mail*) y se habilitará un pequeño foro para dudas de la comunidad. Hasta la creación de dicha web, las herramientas que proporciona *GitHub* serán más que suficientes.

Dado que los potenciales usuarios de la librería son un nicho realmente escaso, se trabajará especialmente en el *SEO*, tratando de absorber gran parte del público desde buscadores.

4.4. Listas de correo

Debido a que aún no se dispone de la web, se ha creado una lista de correo con *Google Groups* [7] para comunicar de forma directa y rápida noticias relacionadas con la librerías.

Adicionalmente, con el crecimiento de la comunidad se planteará el uso de sistemas de comunicación más modernos y completos, como la creación de un canal oficial en *Discord* y/o *Slack*. Estos se emplearán para la resolución directa de dudas, la coordinación de la comunidad y la publicación de nuevas.

5. Conclusiones

En primer lugar, el Aprendizaje Federado es una tecnología realmente novedosa y prometedora, siendo las librerías y *frameworks* existentes, por lo general, escasos e inmaduros. En lo tocante al Aprendizaje Federado Descentralizado, las librerías son directamente inexistentes. De este modo, una librería de código abierto, como la desarrollada, implica una base sólida para futuros desarrollos, ya bien sea orientados a investigación o a la creación de sistemas de Aprendizaje Federado Descentralizado.

Para impulsar la presencia pública de la librería, se ha destinado la mayoría del esfuerzo en *GitHub*. En esta plataforma se encuentran la mayoría de potenciales usuarios, que podrán emplear o extender la librería. Centrándose en el Aprendizaje Federado, es un enfoque muy novedoso que se investiga de manera activa, por lo tanto, se buscará

[6] https://pypi.org/project/p2pfl/
[7] https://groups.google.com/g/p2pfl

atraer a esta comunidad científica, la cual tiende a trabajar con herramientas libres para favorecer la reutilización de código y trabajo previo.

En cuanto al rendimiento y funcionamiento de la librería ha sido realmente satisfactorio, cumpliendo los objetivos propuestos. No obstante, existirán multitud de lineas de trabajo que aún deben ser abordadas.

Para finalizar, mencionar nuevamente que en la memoria original del proyecto, accesible desde el repositorio de *GitHub*, podrá encontrarse una explicación más detallada y completa de la mayoría de elementos tratados en el actual documento.

6. Bibliografía

[1] B. McMahan and D. Ramage, "Federated learning: Collaborative machine learning without centralized training data," 2017. [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[2] R. M. Parizi, A. Dehghantanha, Q. Zhang, and K. Franke, "Decentralized federated learning: An introduction and the road ahead," in *Annual Hawaii International Conference on System Sciences*. IEEE Computer Society, 2021.

[3] M. Jelasity, "Gossip," in *Self-organising software*. Springer, 2011, pp. 139–162.

[4] S. Ghazinoory, M. Abdi, and M. Azadegan-Mehr, "Swot methodology: a state-of-the-art review for the past, a framework for the future," *Journal of business economics and management*, vol. 12, no. 1, pp. 24–48, 2011.

[5] G. Van Rossum and F. L. Drake Jr, *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[6] "Python software foundation license." [Online]. Available: https://docs.python.org/3/license.html

[7] "The mit license, line by line." [Online]. Available: https://writing.kemitchell.com/2016/09/21/MIT-License-Line-by-Line.html

[8] "Apache license, version 2.0." [Online]. Available: https://www.apache.org/licenses/LICENSE-2.0

[9] "Gnu general public license," Free Software Foundation. [Online]. Available: http://www.gnu.org/licenses/gpl.html

[10] J. P. Alexander Menzinsky, Gertrudis López, "Scrum manager," *Iubaris Info 4 Media SL*, 2016.

[11] L. M. Crespo Martínez and F. A. Candelas-Herías, *Introducción a TCP/IP: sistemas de transporte de datos*. Universidad de Alicante, 1998.

[12] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, p. 120–126, feb 1978. [Online]. Available: https://doi.org/10.1145/359340.359342

[13] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[14] Y. LeCun and C. Cortes, “MNIST handwritten digit database,” 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/