



FLEXible, una librería para la simulación de experimentos de Aprendizaje Federado

Alberto Argente del Castillo Garrido

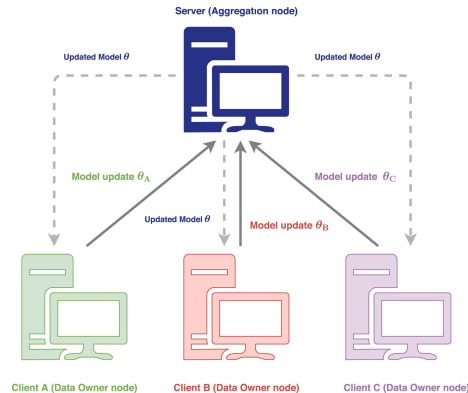


Índice

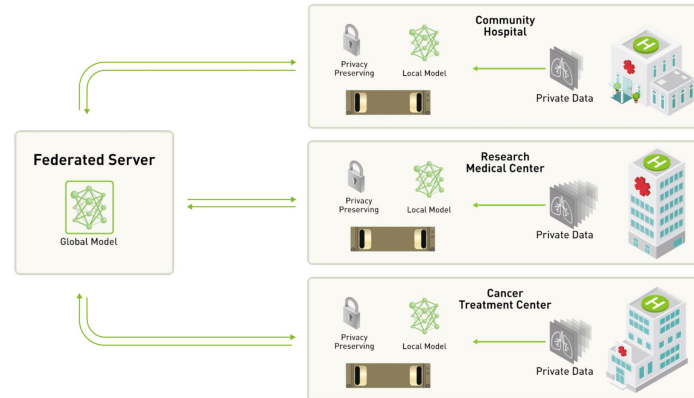
- ¿Qué es el aprendizaje federado?
- ¿Por qué crear FLEXible?
- FLEXible
- Verticales de FLEXible

¿Qué es el Aprendizaje Federado?

El Aprendizaje Federado (AF) es un paradigma de aprendizaje automático que permite entrenar de forma distribuida modelos de aprendizaje sobre múltiples dispositivos electrónicos sin necesidad de compartir los datos.



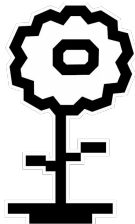
Arquitectura cliente-servidor



Motivación tras FLEXible

Las librerías existentes presentan diversos inconvenientes:

- Experimentación limitada
- Compatibilidad con otras librerías limitada
- Dificultad de ampliación a nuevas funcionalidades
- Dificultad de uso



PySift

FLEXible → pip install flexible-fl



- Librería para la federación de modelos
- Compatible con múltiples librerías de Deep Learning y Machine Learning
- Dispone de múltiples datasets federados
- Federación de datos centralizados customizable
- Facilidad de adaptación de código centralizado a federado
- Módulos disponibles:
 - Datos
 - Actores
 - Modelos
 - Pool → Se encarga de las comunicaciones entre los actores
- “Comunicaciones” de bajo nivel a más alto nivel, para usuarios expertos/principiantes:
 - Uso de *primitivas*
 - Uso de decoradores
 - Realizar comunicaciones a mano



FLEXible (ejemplo carga de datos)



Cargar dataset directamente federado:

```
from flex.datasets import load
from torchvision import transforms

flex_dataset, test_data = load("federated_emnist", return_test=True, split="digits")
```

Dataset federado con FLEXible:

```
config = FedDatasetConfig(seed=0)
config.replacement = False
config.n_nodes = 100

flex_dataset = FedDataDistribution.from_config(
    centralized_data=Dataset.from_torchvision_dataset(train_data), config=config
)
```

FLEXible (ejemplo agregador)



Usando las primitivas de FLEXible:

```
from flex.pool import fed_avg  
  
aggregators.map(fed_avg)
```

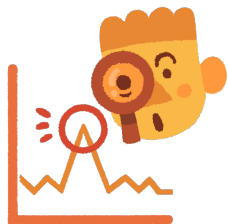
Usando los decoradores de FLEXible:

```
from flex.pool import aggregate_weights  
import tensorly as tl  
  
tl.set_backend("pytorch")  
  
@aggregate_weights  
def aggregate_with_fedavg(list_of_weights: list):  
    agg_weights = []  
    for layer_index in range(len(list_of_weights[0])):  
        weights_per_layer = [weights[layer_index] for weights in list_of_weights]  
        weights_per_layer = tl.stack(weights_per_layer)  
        agg_layer = tl.mean(weights_per_layer, axis=0)  
        agg_weights.append(agg_layer)  
    return agg_weights  
  
# Aggregate weights  
aggregators.map(aggregate_with_fedavg)
```

Verticales de FLEXible



Los verticales son librerías adyacentes a FLEXible que complementan su funcionalidad.



Flex-Anomalies



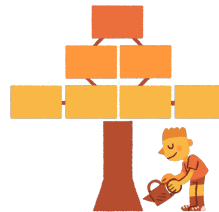
Flex-block



Flex-Clash



Flex-NLP



Flex-Trees

Desarrolladores



<https://github.com/FLEXible-FL/FLEXible>



<https://flexible.readthedocs.io/en/latest/index.html>



Alberto Argente
@AlArgente



Daniel Jiménez
@Xehartnort

FIN

¡Muchas gracias por su atención!