



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Informe PTI Lab 5: Machine learning-based application

Arnau Garcia Gonzalez

PTI Quadrimestre tardor Curs 2025-26

Índex

1. Introducció.....	3
2. Part 1: Federated Learning Network.....	4
2.1 Preparació de l'entorn.....	4
2.2 Entrenament del model Machine Learning.....	4
2.2.1 Dataset.....	4
2.2.2 Executar la implementació.....	5
2.3 Tasca a: Sessions d'entrenament.....	6
2.4 Tasca b: Guardar el model entrenat.....	6
3. Part 2: Desplegament d'un Model amb Flask.....	7
3.1 Preparació de l'entorn.....	7
3.2 Executar el prototip de l'aplicació "medical doctor".....	7
3.3 Desplegament de l'aplicació amb Docker i Minikube.....	8

1. Introducció

En aquest laboratori es farà una prova d'un prototip d'una aplicació basada en machine learning que ajudarà un metge a avaluar una imatge de radiografia per determinar si un pacient té pneumonia o no.

El laboratori es dividirà en dues parts:

- Es farà l'entrenament d'un model de machine learning utilitzant federated learning. S'utilitzarà una implementació existent que permet entrenar un model profund amb dades distribuïdes.
- Es farà l'execució d'una aplicació prototip que utilitzarà el model entrenat en la primera part per a realitzar la classificació de radiografies. Com a extensió d'aquesta part es desplegarà l'aplicació amb docker i Minikube.

La connexió entre ambdues parts serà que el model entrenat en la primera part s'utilitzarà en la segona. Tot i això, per facilitar la realització del laboratori, la segona part inclou models per defecte, de manera que ambdues parts es podran fer de manera independent.

El resultat final del laboratori serà l'adquisició d'una experiència inicial sobre com es pot entrenar un model de machine learning mitjançant una aplicació distribuïda i com s'integra aquest model en una aplicació prototip.

2. Part 1: Federated Learning Network

En aquesta part es farà l'entrenament d'un model de machine learning mitjançant la tècnica de federated learning. Aquesta tècnica permet entrenar un model de forma distribuïda entre diversos nodes clients que comparteixen els seus pesos amb un node central o servidor, sense necessitat d'intercanviar les dades originals. D'aquesta manera, s'aconsegueix millorar el rendiment del model tot preservant la privacitat de les dades mèdiques utilitzades per a l'entrenament.

2.1 Preparació de l'entorn

Per aquesta pràctica és necessària una versió de python superior a la 3.8. Per tant, primer que tot s'ha comprovat la versió de python. Posteriorment, s'ha creat un entorn virtual i s'ha activat. També s'han instal·lat totes les dependències amb pip3.

```
Shell
python3 --version          # Comprova la versió de python
python3 -m venv practfl    # Crea l'entorn virtual
source practfl/bin/activate # Activa l'entorn virtual
pip3 install -r requirements.txt # Instal·la les dependències amb pip3
```

2.2 Entrenament del model Machine Learning

Aquesta part del laboratori consisteix a entrenar un model de *deep learning* utilitzant la tècnica de *federated learning*, que permet distribuir el procés d'entrenament entre diversos dispositius (clients) sense compartir directament les dades originals.

Hi ha dos tipus de nodes: el servidor, que coordina el procés d'entrenament i agrega els resultats dels clients, i els clients que entrena el model amb les seves dades locals i envia els pesos al servidor. La comunicació entre el servidor i els clients es fa mitjançant peticions HTTP i cadascun d'ells executa un petit servidor Flask.

2.2.1 Dataset

Per a aquest laboratori s'ha utilitzat el conjunt de dades Chest X-Ray, que conté imatges de radiografies de tòrax normals i amb pneumònia. S'ha descarregat aquest últim i s'ha comprovat que es tenia la següent estructura de directoris:

```
Shell
federated-learning-network-main/client/database/chest_xray/
├── train/
│   ├── NORMAL/
│   └── PNEUMONIA/
└── test/
```

```
|— NORMAL /  
|— PNEUMONIA /
```

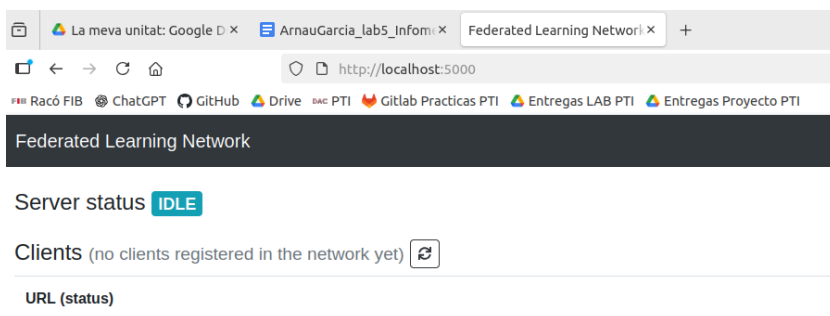
2.2.2 Executar la implementació

Una vegada l'estructura de directoris ha estat correcta s'ha procedit amb l'execució. S'han obert quatre terminals diferents (en totes elles s'ha activat l'entorn prèviament creat).

En la primera terminal s'ha creat el node central, el server.

```
Shell  
cd server      #Accedeix al directori del server  
flask run      #El comença a córrer
```

Això activa el node server a <http://localhost:5000>. Visitant des del navegador es veu el dashboard de la xarxa.



Després d'això s'han iniciat els tres clients. En un dels clients s'han seguit les següents passes:

```
Shell  
source practfl/bin/activate  
cd client  
#En els altres dos clients com a port s'ha posat 5002 i 5003  
export CLIENT_URL='http://localhost:5001'  
flask run --port 5001
```

I en els altres dos s'ha fet el mateix però canviant el port pels 5002 i 5003.

Després de fer això podem veure en el server que els clients s'han registrat correctament.

```

I'm client 3
2025-10-12 00:40:36.807526: W tensorflow/stream_executor/platform/default/dso_loader.cc:64]
Could not load dynamic library 'libcudart.so.11.0'; dLError: libcudart.so.11.0: cannot open
shared object file: No such file or directory; LD_LIBRARY_PATH: /home/alumne/catkin_ws/devel
/lib:/opt/ros/noetic/lib
2025-10-12 00:40:36.807635: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above
CUDA dLError if you do not have a GPU set up on your machine.
Initializing ChestXRayModelTrainer...
Warning: SERVER_URL environment variable is not defined, using DEFAULT_SERVER_URL: http://12
7.0.0.1:5000
Registering in central node: http://127.0.0.1:5000
Doing request http://127.0.0.1:5000/client
Response received from registration: <Response [201]>
Client registered successfully
* Serving Flask app 'app.py'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a produ
ction WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5003
* Running on http://192.168.206.131:5003
Press CTRL+C to quit

```

La meva unitat: Google D X ArnauGarcia_lab5_inform X Feder

← → ↻ http://localhost:5000

Racó FIB ChatGPT GitHub Drive PTI Gitlab Practicas PTI

Federated Learning Network

Server status **IDLE**

Clients (3 clients are registered in the network)

URL (status)
http://localhost:5001 IDLE
http://localhost:5002 IDLE
http://localhost:5003 IDLE

2.3 Tasca a: Sessions d'entrenament

Després d'haver fet la part prèvia (al final de l'apartat anterior es poden veure les imatges demanades a 2.3.1 i 2.3.2), s'ha continuat amb la primera tasca, fer sessions d'entrenament.

Per això s'ha accedit al dashbord s'ha premut Launch training i s'ha triat l'opció CHEST_X_RAY. Després de fer això a la consola es pot veure el següent resultat per cadascun dels clients.

```

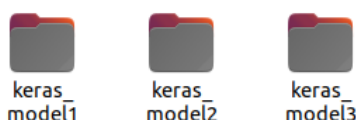
10/10 - 23s - loss: 16.9712 - accuracy: 0.7000 - val_loss: 58.0693 - val_accuracy: 0.4400 -
23s/epoch - 2s/step
20/20 [=====] - 3s 159ms/step - loss: 51.2416 - accuracy: 0.5000
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_co
nvolution_op while saving (showing 2 of 2). These functions will not be directly callable af
ter loading.
Deleting content of temporary folder /home/alumne/Documents/PTI-FIB/Practicas_lab/Lab5/ml_ap
plication/federated-learning-network-main/client/tmp1/chest_xray/tmpyicqo5ug
Deleting temporary dataset folder /home/alumne/Documents/PTI-FIB/Practicas_lab/Lab5/ml_appli
cation/federated-learning-network-main/client/tmp1/chest_xray/
model params SHAPE: (3, 3, 3, 32)
model params SHAPE: (32,)
model params SHAPE: (3, 3, 32, 64)
model params SHAPE: (64,)
model params SHAPE: (200704, 2)
model params SHAPE: (2,)
Sending calculated model weights to central node
Response received from updating central model params: <Response [200]>
Model params updated on central successfully
Training finished...
127.0.0.1 - - [12/Oct/2025 00:57:30] "POST /training HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [12/Oct/2025 00:57:30] "POST /training HTTP/1.1" 200 -

```

Després s'han fet algun entrenament més per tal de millorar la precisió. Per això s'ha hagut de posar la variable USE_TRAINED_MODEL en **config.py** com a True.

2.4 Tasca b: Guardar el model entrenat

Aquest apartat té com a objectiu verificar que el lloc on es guarda el model entrenat és correcte. També s'ha posat la variable USE_TRAINED_MODEL en true per poder entrenar més el model, sobre el model ja entrenat. Per a això s'ha mirat a **config.py** la variable INITIAL_MODEL_PATH. S'ha vist que els directoris són els següents:



3. Part 2: Desplegament d'un Model amb Flask

En aquesta segona part del laboratori, es posa en pràctica el model de machine learning entrenat a la part anterior. L'objectiu és integrar aquest model en una aplicació web que permeti a un metge carregar radiografies de tòrax i obtenir una classificació automàtica: si la imatge és NORMAL o mostra signes de PNEUMÒNIA.

Aquesta aplicació es desenvolupa amb Flask (Python) i ofereix una interfície web senzilla on es poden provar tant models entrenats com no entrenats per comparar els resultats.

3.1 Preparació de l'entorn

La preparació de l'entorn es la mateixa que en la primera part de la pràctica però ara des del directori `keras-flask-deploy-webapp-class_pneu`:

```
Shell
python3 --version          # Comprova la versió de python
python3 -m venv practfl    # Crea l'entorn viretual
source practfl/bin/activate # Activa l'entorn virtual
pip3 install -r requirements.txt # Instala les dependències amb pip3
```

3.2 Executar el prototip de l'aplicació “medical doctor”

Una vegada fet això s'ha modificat el fitxer `app_pneu.py` per modificar la variable `MODEL_PATH` i així poder escollir entre els models entrenats i els no entrenats. També s'ha configurat per utilitzar els models entrenats en la part 1.

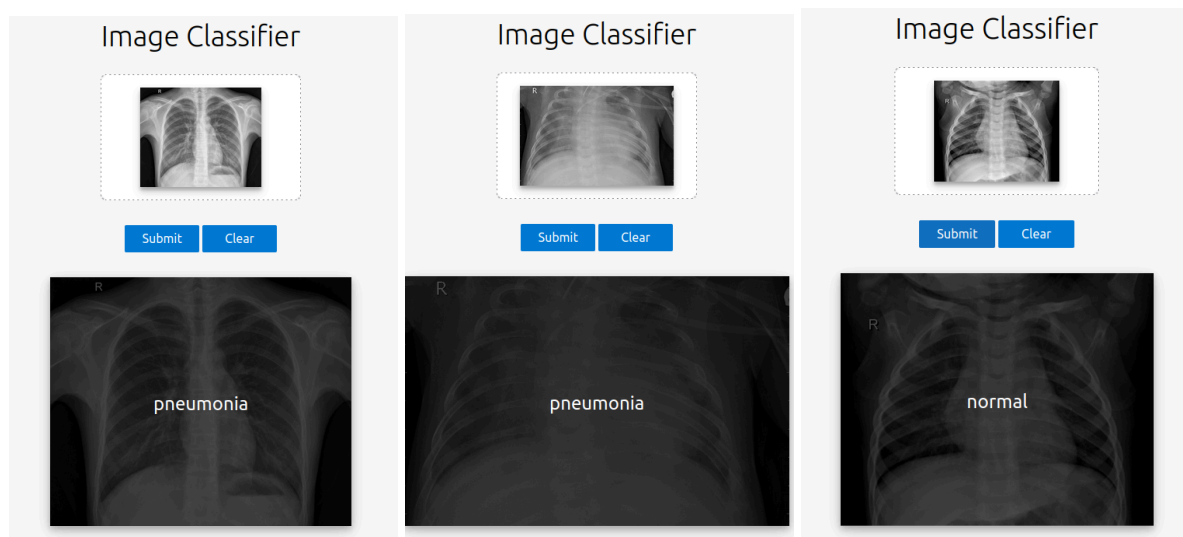
```
Shell
#Model entrenat
MODEL_PATH = './keras_model_trained'

#Model sense entrenar
MODEL_PATH = './keras_model_untrained'

#Model part 1 (també s'ha canviat pels altres dos enrenats)
MODEL_PATH =
'/home/alumne/Documents/PTI-FIB/Practicas_lab/Lab5/ml_application/federated-
learning-network-main/client/keras_model1'
```

Després s'ha iniciat el programa i s'ha vist el següent a `http://localhost:5000`.

S'han obtingut tant resultats correctes com erronis en tots els casos:



3.3 Desplegament de l'aplicació amb Docker i Minikube

Per Dockeritzar 'aplicació s'ha comprovat primer de tot que el Dockerfile fos correcte, i s'han executat les següents comandes:

```
Shell
docker build -t pneumonia-app .
docker run -p 5000:5000 pneumonia-app
```

per finalment desplegar a minikube s'ha fet el següent:

```
Shell
#Iniciem minikube
minikube start

#Activem el entorn de minikube per construir el contenidor
eval $(minikube docker-env)
docker build -t pneumonia-app:latest .

#Comprovem que s'ha creat correctemnt
docker images
```

Després s'ha creat el següent pneumonia-app-deployment.yaml.

```
Shell
apiVersion: apps/v1
kind: Deployment
metadata:
```



```

    name: pneumonia-app-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: pneumonia-app
  template:
    metadata:
      labels:
        app: pneumonia-app
    spec:
      containers:
        - name: pneumonia-app
          image: pneumonia-app:latest
          imagePullPolicy: IfNotPresent
          ports:
            - containerPort: 5000
---
apiVersion: v1
kind: Service
metadata:
  name: pneumonia-app-service
spec:
  type: LoadBalancer
  selector:
    app: pneumonia-app
  ports:
    - port: 5000
      targetPort: 5000

```

Una vegada creat s'ha aplicat el desplegament:

```

Shell
kubectl apply -f pneumonia-app-deployment.yaml

#Per veure els pods i comprovar l'estat
kubectl get pods
kubectl get svc

```

```

teras-flask-deploy-webapp-class_pneu$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
carrental-754d85f666-2kqsf          1/1     Running   1 (11m ago)  7d11h
carrental-754d85f666-v7x4n          1/1     Running   1 (11m ago)  7d11h
pneumonia-app-deployment-bb6f6cd8c-ckpkp 1/1     Running   0           8s
(practfl) alumne@nipigon:~/Documents/PTI-FIB/Practicas_lab/Lab5/ml_application/k

```

Finalment, s'ha accedit a l'URL <http://192.168.49.2:31436> per comprovar que l'aplicació estava desplegada.

```
pneumonia-app-deployment-bb6f6cd8c-ckp 1/1 Running 0 8s
(practfl) alumne@nipigon:~/Documents/PTI-FIB/Practicas_lab/Lab5/ml_application/k
eras-flask-deploy-webapp-class_pneu$ minikube service pneumonia-app-service --ur
l
http://192.168.49.2:31436
(practfl) alumne@nipigon:~/Documents/PTI-FIB/Practicas_lab/Lab5/ml_application/k
```

