

PTI pràctica 5: MI_applicaction

Joan Llonch Majó

Luis Jesús Valverde Zavaleta

Entorn y configuración de la práctica

L'entorn que hem utilitzat per a la realització de la pràctica és una imatge proporcionada la UPC que és la Ubuntu22v3r1 i que l'hem corregut en la màquina virtual VirtualBox.

Per a configurar la màquina i tenir-la preparada per a la realització de la pràctica, seguim els comandos específicats en l'enunciat de l'assignatura. Les eines que hem hagut d'usar per a aquesta pràctica bàsicament han estat python3, l'entorn virtual anomenat practfl.

- sudo apt-get install -y python3
- python3 -m venv practfl

Un cop instal·lat l'entorn virtual al nostre PC l'hem activat utilitzant la comanda:

- source practfl/bin/activate

Una vegada instal·lat i activat l'entorn virtual comprovem a la mateixa terminal que surti practfl a la línia de comandes per a veure que s'ha executat amb èxit. Per a continuar treballant ens han indicat que necessitem una sèrie de paquets i, per tant, els hem d'instal·lar amb la comanda

- pip3 install -r requirements.txt

Aquesta el que fa és instal·lar-nos un framework de machine learning per a poder posteriorment treballar amb ell. Cal afegir que també hem hagut d'instal·lar pip3, ja que no venia instal·lada per defecte usant la comanda:

- sudo apt install python3-pip

Familiarització amb les eines de la pràctica

En aquest apartat volem estendre l'explicació sobre l'entorn utilitzat en concret l'entorn virtual i el framework Tensorflow. El framework Tensorflow és de codi obert desenvolupat per Google per realitzar machine learning la qual ens proporciona una plataforma flexible i fàcil d'usar per a desenvolupar, construir i entrenar models de machine learning de forma automàtica i eficient. Algunes de les seves característiques poden ser la flexibilitat per als desenvolupadors per a desenvolupar xarxes neuronals senzilles fins a models recurrents per a processar el llenguatge natural, l'escalabilitat és un altre punt fort a estacar al per escalar des de dispositius mòbils fins a dispositius IoT per a grans infraestructures, també ens proporciona interoperabilitat entre diversos llenguatges com ara poden ser Python i C++ per a facilitar-nos la integració i per últim les eines de desplegament que ofereix com ara API's per a la gran varietat de plataformes com apps web, dispositius mòbils i servidors.

Quan parlem sobre l'entorn virtual "practfl" no deixa de ser una eina usada per crear un ambient aïllat el qual podem instal·lar paquets i dependències (en el nostre cas de python). Això ens pot permetre tenir diferents versions per a diferents entorns i no

interferir amb el SO. En el nostre cas l'hem utilitzar perquè les dependències del projecte siguin consistents. A més a més, aquest entorn virtual ha de ser creat i gestionat per a una eina com venv en el nostre cas la qual és una eina de Python per a crear entorns virtuals, ja que com hem mencionat anteriorment ens facilita la gestió de les dependències.

Realització de la primera part de la pràctica

Un cop realitzada la preparació de la pràctica amb l'entorn virtual i totes les dependències hem realitzat la pràctica, aquesta consta de dues parts la primera la qual consisteix a demostrar amb captures de pantalla que hem completat la pràctica correctament. Aquesta primera part es subdivideix en dues parts les quals poden ser independents l'una de l'altra, ja que a la primera part consisteix a entrenar un model i la segona part a utilitzar aquest model, però, a la segona part ja ens proporcionen models entrenats, per tant, les dues subparts poden ser independents. La segona part de la pràctica consisteix a dur a terme una extensió d'alguna de les dues subparts de la primera part les quals poden ser a nivell teòric o pràctic. En el nostre cas serà teòrica. En el nostre cas volem indagar sobre com els models porten a cap aquests entrenaments i són capaços de millorar.

Nosaltres en el nostre cas vam decidir començar i dur a terme la pràctica amb l'ordre marcat, és a dir, fer primer part el primer subapartat de la carpeta federated-learning-network-main.

El model que ens proporcionen a entrenar consisteix en una xarxa federada d'aprenentatge que consisteix en diversos nodes distribuïts que es comuniquen entre ells per enviar missatges relacionats amb l'entrenament de machine learning dels processos. A nivell d'aplicació hi ha dos tipus de nodes el node servidor on orquestra l'entrenament dels processos (només n'hi ha un) i el segon tipus que són clients d'aquest primer o treballadors els quals usaran el model de machine learning per a entrenar en l'àmbit local. Aquests nodes es comuniquen entre ells a través de http. En el nostre cas aquesta implementació treballa amb un conjunt de dades que són radiografies del pit on ens les proporcionen a través d'aquest link https://data.mendeley.com/public-files/datasets/rscbjbr9sj/files/f12eaf6d-6023-432f-a_cc9-80c9d7393433/file_downloaded

Aquest ens proporciona radiografies d'una persona normal (sana) i d'una persona amb pneumònia (malata).

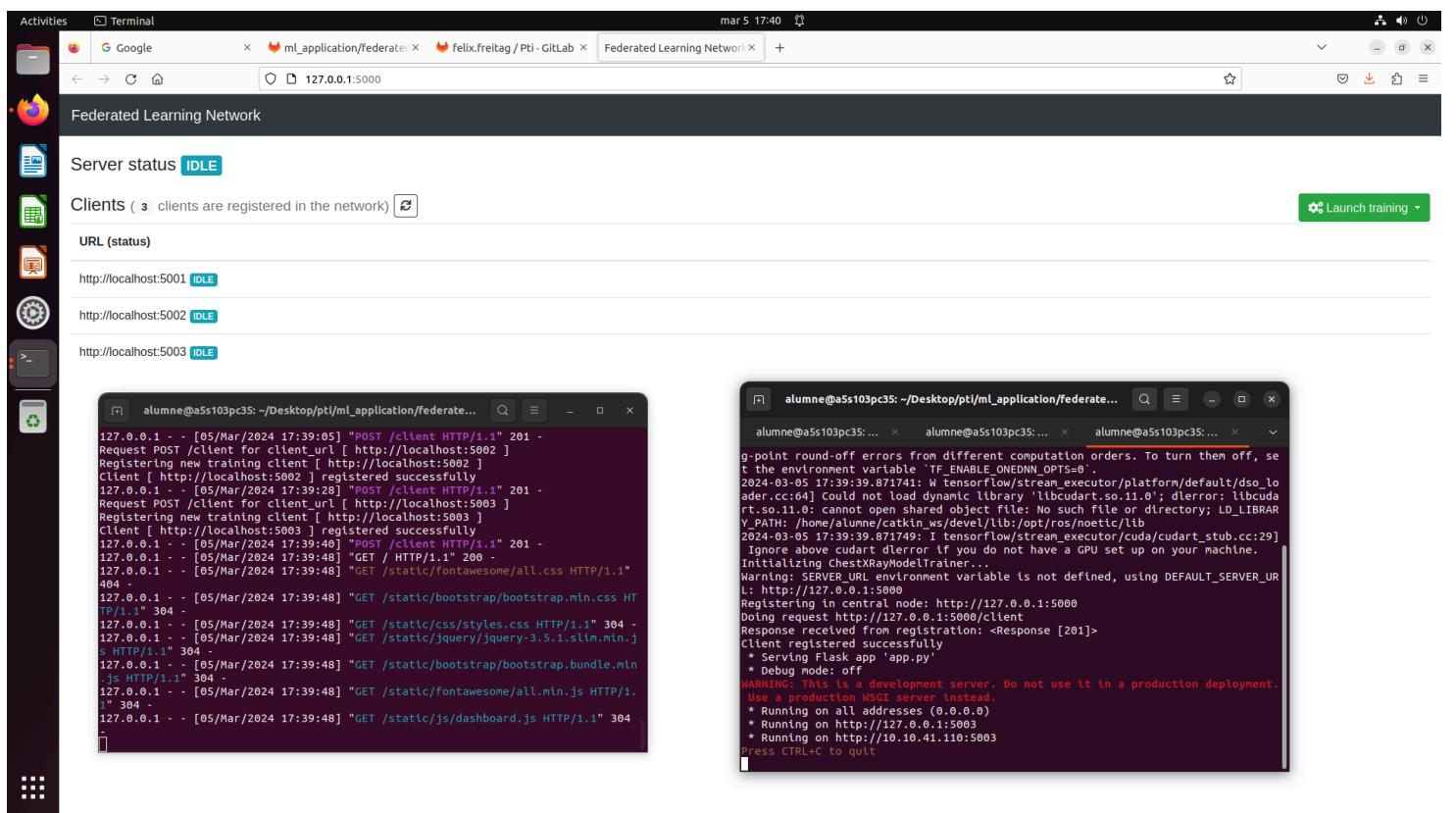
Per a córrer la nostra aplicació hem d'activar l'entorn virtual a cadascuna de les terminals en el nostre cas en seran quatre tres clients del tipus dos i un servidor de tipus 1. Per a activar usarem la comanda `source practfl/bin/activate`.

Un cop realitzada la comanda obrirem el servidor amb el node del tipus 1, per a fer-ho, accedirem a la carpeta del servidor on la trobarem al path `$HOME/federated-learning-network/server` i executarem `flask run` el

qual ens inciara un node central a l'adreça <http://localhost:5000>. Després iniciarem els tres clients dirigint-nos a les carpetes dels clients `$HOME/federated-learning-network/client` i executarem les commandes `export CLIENT_URL='http://localhost:5001'`

`flask run --port 5001` on la primera serveix per a establir una variable d'entorn que és l'URL d'un dels clients i la segona inicia el servidor al port 5001 on el port detectarà l'arxiu `app.py` i l'executarà.

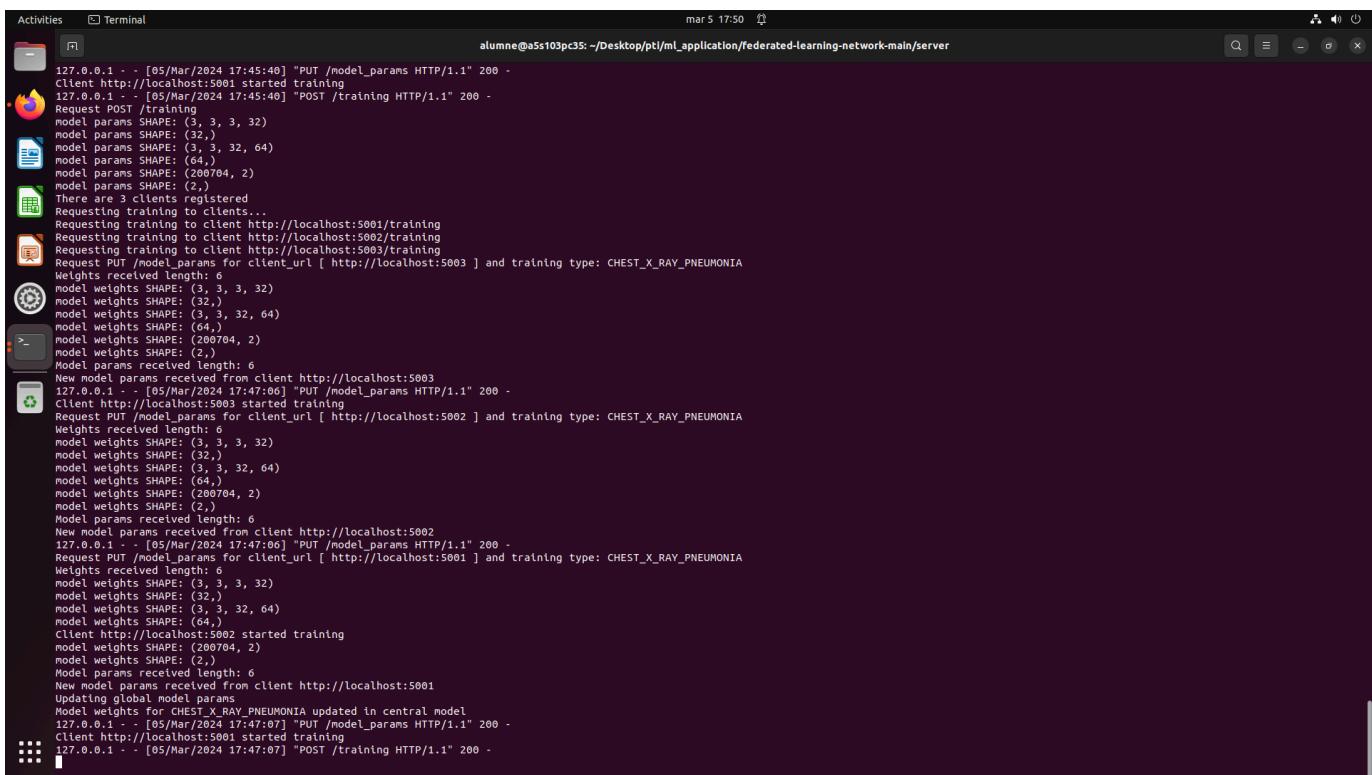
Això ho farem igual per als dos clients restants el que simplement els hi canviarem el port 5001 per a 5002 i 5003 respectivament. Adjunto aquí la foto corresponent 1 i 2.



En aquesta podem trobar la primera foto que se'n demana la qual és la consola una vegada s'hagin registrat els tres clients, però encara no han estat entrenats, la qual és la que tenim mirant a la pantalla a l'esquerra. La segona consola tenim la foto d'un dels clients després de ser registrat pel servidor i abans que hagi estat entrenat. En aquesta podem veure el missatge a mesura que el client ha estat registrat de forma correcta i de fons de pantalla tenim els tres clients registrats al servidor.

Entrenament

Aquesta part de la pràctica consisteix en l'entrenament d'aquests models, un cop han estat registrats al node central accedim a l'adreça <http://localhost:5000> (fons de pantalla de l'imatge 1) i cliquem sobre lunch training i seleccionem que siguin entrenats amb les radiografies de pit perquè facin una ronda d'entrenament. Ajunto imatges 3(imatge de la consola del servidor després d'haver realitzat una ronda d'entrenament i la imatge 4 el qual és la consola del client després de realitzar una ronda d'entrenament.



```
Activities Terminal mar 5 17:50
alumne@a5s103pc35: ~/Desktop/ptl/ml_application/federated-learning-network-main/server
127.0.0.1 - - [05/Mar/2024 17:45:40] "PUT /model_params HTTP/1.1" 200 -
Client http://localhost:5001 started training
127.0.0.1 - - [05/Mar/2024 17:45:40] "POST /training HTTP/1.1" 200 -
Request POST /training
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,)

There are 3 clients registered
Requesting training to clients...
Requesting training to client http://localhost:5001/training
Requesting training to client http://localhost:5002/training
Requesting training to client http://localhost:5003/training
Request PUT /model_params for client_url [ http://localhost:5003 ] and training type: CHEST_X_RAY_PNEUMONIA
Weights received length: 6
model weights SHAPE: (3, 3, 3, 32)
model weights SHAPE: (32,)
model weights SHAPE: (3, 3, 32, 64)
model weights SHAPE: (64,)
model weights SHAPE: (200704, 2)
model weights SHAPE: (2,)

Model params received length: 6
New model params received from client http://localhost:5003
127.0.0.1 - - [05/Mar/2024 17:47:06] "PUT /model_params HTTP/1.1" 200 -
Client http://localhost:5003 started training
Request PUT /model_params for client_url [ http://localhost:5002 ] and training type: CHEST_X_RAY_PNEUMONIA
Weights received length: 6
model weights SHAPE: (3, 3, 3, 32)
model weights SHAPE: (32,)
model weights SHAPE: (3, 3, 32, 64)
model weights SHAPE: (64,)
model weights SHAPE: (200704, 2)
model weights SHAPE: (2,)

Model params received length: 6
New model params received from client http://localhost:5002
127.0.0.1 - - [05/Mar/2024 17:47:06] "PUT /model_params HTTP/1.1" 200 -
Request PUT /model_params for client_url [ http://localhost:5001 ] and training type: CHEST_X_RAY_PNEUMONIA
Weights received length: 6
model weights SHAPE: (3, 3, 3, 32)
model weights SHAPE: (32,)
model weights SHAPE: (3, 3, 32, 64)
model weights SHAPE: (64,)
Client http://localhost:5002 started training
model weights SHAPE: (200704, 2)
model weights SHAPE: (2,)

Model params received length: 6
New model params received from client http://localhost:5001
Updating global model params
Model weights for CHEST_X_RAY_PNEUMONIA updated in central model
127.0.0.1 - - [05/Mar/2024 17:47:07] "PUT /model_params HTTP/1.1" 200 -
Client http://localhost:5001 started training
127.0.0.1 - - [05/Mar/2024 17:47:07] "POST /training HTTP/1.1" 200 -
```

Imatge 3.

```

Activities Terminal mar 5 17:44
alumne@as5103pc35: ~/Desktop/ptl/ml_application/federated-learning-network-main/client
alumne@as5103pc35: ~/Desktop/ptl/ml_application/federated-learning-network-main/client
alumne@as5103pc35: ~/Desktop/ptl/ml_application/federated-learning-network-main/client

conv2d (Conv2D)      (None, 224, 224, 32)    896
max_pooling2d (MaxPooling2D) (None, 112, 112, 32)    0
)
conv2d_1 (Conv2D)      (None, 112, 112, 64)    18496
max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64)    0
)
flatten (Flatten)      (None, 200704)    0
dense (Dense)      (None, 2)    401410
=====
Total params: 426,802
Trainable params: 420,802
Non-trainable params: 0
Using default model weights
Create temp dataset...
Temporary dataset folder /home/alumne/Desktop/ptl/ml_application/federated-learning-network-main/client/tmp1/chest_xray/ doesn't exist, creating it
Temporary folder for training: /home/alumne/Desktop/ptl/ml_application/federated-learning-network-main/client/tmp1/chest_xray/tmpfhnjvcjm
Loading CHEST X-RAY IMAGES dataset...
Found 200 images belonging to 2 classes.
Found 100 images belonging to 2 classes.
2
Epoch 1/2
10/10 [=====] - loss: 21.4632 - accuracy: 0.6200 - val_loss: 11.2282 - val_accuracy: 0.6400 - 2s/epoch - 17ms/step
Epoch 2/2
10/10 [=====] - loss: 19.6337 - accuracy: 0.6000 - val_loss: 4.9777 - val_accuracy: 0.8400 - 1s/epoch - 139ms/step
1/20 [>.....] - ETA: 0s - loss: 9.6319e-06 - accuracy: 3/20 [=====>.....] - ETA: 0s - loss: 6.3639 - accuracy: 0.86 4/20 [=====>.....] - ETA: 0s - loss: 6.9980 - accuracy: 0.77 9/20 [=====>.....] - ETA: 0s - loss: 7.9495 - accuracy: 0.76 7/20 [=====>.....] - ETA: 0s - loss: 6.9980 - accuracy: 0.77 9/20 [=====>.....] - ETA: 0s - loss: 5.8114 - accuracy: 0.86 10/20 [=====>.....] - ETA: 0s - loss: 5.8114 - accuracy: 0.86 10/20 [=====>.....] - ETA: 0s - loss: 5.2234 - accuracy: 0.7816/20 [=====>.....] - ETA: 0s - loss: 6.3933 - accuracy: 0.7520/20 [=====>.....] - ETA: 0s - loss: 4.7864 - accuracy: 0.8017/20 [=====>.....] - ETA: 0s - loss: 6.2798 - accuracy: 0.7618/20 [=====>.....] - ETA: 0s - loss: 6.3933 - accuracy: 0.7520/20 [=====>.....] - ETA: 0s - loss: 7.5719 - accuracy: 0.7420/20 [=====>.....] - 1s 49ms/step - loss: 7.5719 - accuracy: 0.7400
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing z of 2). These functions will not be directly callable after loading.
Deleting content of temporary folder /home/alumne/Desktop/ptl/ml_application/federated-learning-network-main/client/tmp1/chest_xray/tmpfhnjvcjm
Deleting temporary dataset folder /home/alumne/Desktop/ptl/ml_application/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 - - [05/Mar/2024 17:42:53] "POST /training HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [05/Mar/2024 17:42:53] "POST /trainning HTTP/1.1" 200 -

```

imatge 4, primera ronda d'entrenament.

```

Activities Terminal mar 5 17:45
alumne@as5103pc35: ~/Desktop/ptl/ml_application/federated-learning-network-main/client
alumne@as5103pc35: ~/Desktop/ptl/ml_application/federated-learning-network-main/client
alumne@as5103pc35: ~/Desktop/ptl/ml_application/federated-learning-network-main/client

Training started...
Model: "sequential"
=====
Layer (type)      Output Shape     Param #
=====
conv2d (Conv2D)      (None, 224, 224, 32)    896
max_pooling2d (MaxPooling2D) (None, 112, 112, 32)    0
)
conv2d_1 (Conv2D)      (None, 112, 112, 64)    18496
max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64)    0
)
flatten (Flatten)      (None, 200704)    0
dense (Dense)      (None, 2)    401410
=====
Total params: 426,802
Trainable params: 420,802
Non-trainable params: 0
Using model weights from central node
Create temp dataset...
Temporary dataset folder /home/alumne/Desktop/ptl/ml_application/federated-learning-network-main/client/tmp1/chest_xray/ doesn't exist, creating it
Temporary folder for training: /home/alumne/Desktop/ptl/ml_application/federated-learning-network-main/client/tmp1/chest_xray/tmpqn5m2pq
Loading CHEST X-RAY IMAGES dataset...
Found 200 images belonging to 2 classes.
Found 100 images belonging to 2 classes.
2
Epoch 1/2
10/10 - loss: 6.2240 - accuracy: 0.6000 - val_loss: 0.8046 - val_accuracy: 0.8400 - 2s/epoch - 179ms/step
Epoch 2/2
10/10 - loss: 2.5548 - accuracy: 0.7400 - val_loss: 3.3036 - val_accuracy: 0.7600 - 1s/epoch - 115ms/step
20/20 [=====] - ETA: 0s - loss: 4.9324 - accuracy: 0.6300
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing z of 2). These functions will not be directly callable after loading.
Deleting content of temporary folder /home/alumne/Desktop/ptl/ml_application/federated-learning-network-main/client/tmp1/chest_xray/tmpqn5m2pq
Deleting temporary dataset folder /home/alumne/Desktop/ptl/ml_application/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 - - [05/Mar/2024 17:45:40] "POST /training HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [05/Mar/2024 17:45:40] "POST /trainning HTTP/1.1" 200 -

```

imatge 4, segonaronda d'entrenament.

```

Activities Terminal mar 5 17:47
alumne@as5103pc35:~/Desktop/ptl/ml_application/federated-learning-network-main/client$ python3 app_pneu.py
Training started...
Model: "sequential"
Layer (Type)          Output Shape       Param #
=====
conv2d (Conv2D)      (None, 224, 224, 32) 896
max_pooling2d (MaxPooling2D) (None, 112, 112, 32) 0
conv2d_1 (Conv2D)     (None, 112, 112, 64) 18496
max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64) 0
flatten (Flatten)    (None, 208704) 0
dense (Dense)        (None, 2) 401410
=====
Total params: 426,802
Trainable params: 426,802
Non-trainable params: 0
Using model weights from central node
Create temp dataset...
Temporary dataset folder /home/alumne/Desktop/ptl/ml_application/federated-learning-network-main/client/tmp/chest_xray/ doesn't exist, creating it
Temporary folder for training: /home/alumne/Desktop/ptl/ml_application/federated-learning-network-main/client/tmp/chest_xray/tmphwcs8r03
Loading CHEST X-RAY IMAGES dataset...
Found 2000 images belonging to 2 classes.
Found 1000 images belonging to 2 classes.
2
Epoch 1/2
10/10 [=====] - loss: 6.3628 - accuracy: 0.7000 - val_loss: 1.5086 - val_accuracy: 0.7200 - 2s/epoch - 179ms/step
Epoch 2/2
10/10 [=====] - loss: 1.3757 - accuracy: 0.8200 - val_loss: 3.8419 - val_accuracy: 0.6000 - 1s/epoch - 119ms/step
2/20 [=====] - loss: 3.6659 - accuracy: 0.6400
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 2 of 2). These functions will not be directly callable after loading.
Deleting contents of temporary folder /home/alumne/Desktop/ptl/ml_application/federated-learning-network-main/client/tmp/chest_xray/tmphwcs8r03
Delete temporary dataset folder /home/alumne/Desktop/ptl/ml_application/federated-learning-network-main/client/tmp/chest_xray/tmphwcs8r03
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: (208704, 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 - [05/Mar/2024 17:47:07] "POST /training HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [05/Mar/2024 17:47:07] "POST /training HTTP/1.1" 200 -

```

Imatge 4, tercera ronda d'entrenament.

Un cop realitzat l'entrenament d'aquest model en varies rondes ens dirigim al subapartat 2 per a continuar la pràctica.

La primera part d'aquesta pràctica (preparing enviroment) l'obviem, ja que ha estat explicada anteriorment en detall. Per tant, ens dirigim a l'apartat per a córrer l'aplicació medica. En aquest primer tros de la pràctica ens hem de dirigir al fitxer app_pneu.py on hi ha tota la configuració del model, en aquest cas hem de tocar les línies on ens diu d'on volem seleccionar el model el qual pot ser un model entrenat (ja ve donat) MODEL_PATH = './keras_model_trained', un model desentrenat MODEL_PATH ='./keras_model_untrained' o per últim afegint el path on conte el nostre model entrenat de l'apartat 1. Un cop dutes a terme aquestes modificacions correm l'aplicació amb la comanda `python3 app_pneu.py`.

Cal afegir que nosaltres ho hem volgut testejar de les tres maneres per a veure els resultats que ens donava i per a preguntar-nos i entendre les diferències que hi ha entre aquests.

Aquestes primeres tres imatges corresponen al model entrenat.

Activities Firefox Web Browser mar 5 18:29

```
alumne@asus10pc35:~/Desktop/plt/ml_application/keras-flask-deploy-webapp-class_p... mar 5 18:29
 3 protobuf-3.20.3 protobuf-5.9.8 tensorflow-2.15.2 tensorflow-data-server-0.7.2 tensorflow-cpu-2.15.0
  .posti tensorflow-estimator-2.15.0 tensorflow-metadata-1.14.0 tensorflow-datasets-4.9.4 tensorflow-2.15.0
  ope.event-5.4. zope.interface-6.2
  [practical] alumne@asus10pc35:~/Desktop/plt/ml_application/keras-flask-deploy-webapp-class_pneu$ python
  3 -m flask run
  2024-03-05 18:27:34.913025: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You
  may see slightly different numerical results due to floating-point round-off errors from different co
  mputation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
  2024-03-05 18:27:34.913025: W tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow bin
  ary is optimized to use available CPU instructions in performance-critical operations.
  To enable the following instructions: AVX2 AVX32F AVX32_VNNI FMA, in other operations, rebuild Tens
  orflow with the appropriate compiler flags.
  /home/alumne/Desktop/plt/ml_application/Federated-learning-network/main/practical/lib/python3.10/site-p
  ackages/scipy/integrate.py:155: UserWarning: A NumPy version >=1.18.5 and <1.25.0 is required for this
  warning. Scipy's default version is 1.26.4
  warnings.warn(f"A NumPy version >={np.__version__} and <{np._minversion}" Model loaded. Start serving..
  127.0.0.1 - - [2024-03-05 18:27:43] "GET / HTTP/1.1" 200 3071 0.003129
  127.0.0.1 - - [2024-03-05 18:27:43] "GET /static/main.css HTTP/1.1" 200 3205 0.003053
  127.0.0.1 - - [2024-03-05 18:27:43] "GET /static/main.js HTTP/1.1" 200 4242 0.000508
  Input image shape:
  (960, 1184, 1)
  Image saved
  Model: "sequential"
  Layer (Type) Output Shape Param #
  =====
  conv2d (Conv2D) (None, 224, 224, 32) 896
  max_pooling2d (MaxPooling2D) (None, 112, 112, 32) 0
  conv2d_1 (Conv2D) (None, 112, 112, 64) 18496
  max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64) 0
  flatten (Flatten) (None, 200764) 0
  dense (Dense) (None, 2) 401410
  =====
  Total params: 420882 (1.61 MB)
  Trainable params: 420882 (1.61 MB)
  Non-trainable params: 0 (0.00 Byte)
  1/1 - 0s - 72ms/epoch - 72ms/step
  [[0.5438527 0.4561473]]
  0.5438527
  0.4561473
  max probability 0.544
  Classification result: pneumonia
  127.0.0.1 - - [2024-03-05 18:29:25] "POST /predict HTTP/1.1" 200 153 0.238059
```

ml_application/keras-flask-deploy-webapp-class_pneu Demo localhost:5000

Image Classifier

Submit Clear

R pneumonia

Aquesta primera imatge correspon a una pneumònia degut a una bacteria la qual ens la classifica de forma correcta.

Activities Firefox Web Browser mar 5 18:30

```
alumne@asus10pc35:~/Desktop/plt/ml_application/keras-flask-deploy-webapp-class_p... mar 5 18:30
  conv2d_1 (Conv2D) (None, 112, 112, 64) 18496
  max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64) 0
  flatten (Flatten) (None, 200764) 0
  dense (Dense) (None, 2) 401410
  =====
  Total params: 420882 (1.61 MB)
  Trainable params: 420882 (1.61 MB)
  Non-trainable params: 0 (0.00 Byte)
  1/1 - 0s - 72ms/epoch - 72ms/step
  [[0.5438527 0.4561473]]
  0.5438527
  0.4561473
  max. probability 0.544
  Classification result: pneumonia
  127.0.0.1 - - [2024-03-05 18:29:25] "POST /predict HTTP/1.1" 200 153 0.238059
  Input image shape:
  (768, 960, 1)
  Image saved
  Model: "sequential"
  Layer (Type) Output Shape Param #
  =====
  conv2d (Conv2D) (None, 224, 224, 32) 896
  max_pooling2d (MaxPooling2D) (None, 112, 112, 32) 0
  conv2d_1 (Conv2D) (None, 112, 112, 64) 18496
  max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64) 0
  flatten (Flatten) (None, 200764) 0
  dense (Dense) (None, 2) 401410
  =====
  Total params: 420882 (1.61 MB)
  Trainable params: 420882 (1.61 MB)
  Non-trainable params: 0 (0.00 Byte)
  1/1 - 0s - 72ms/epoch - 72ms/step
  [[0.469505074 0.53049123]]
  0.469505074
  0.53049123
  max. probability 0.530
  Classification result: normal
  127.0.0.1 - - [2024-03-05 18:30:23] "POST /predict HTTP/1.1" 200 150 0.100319
```

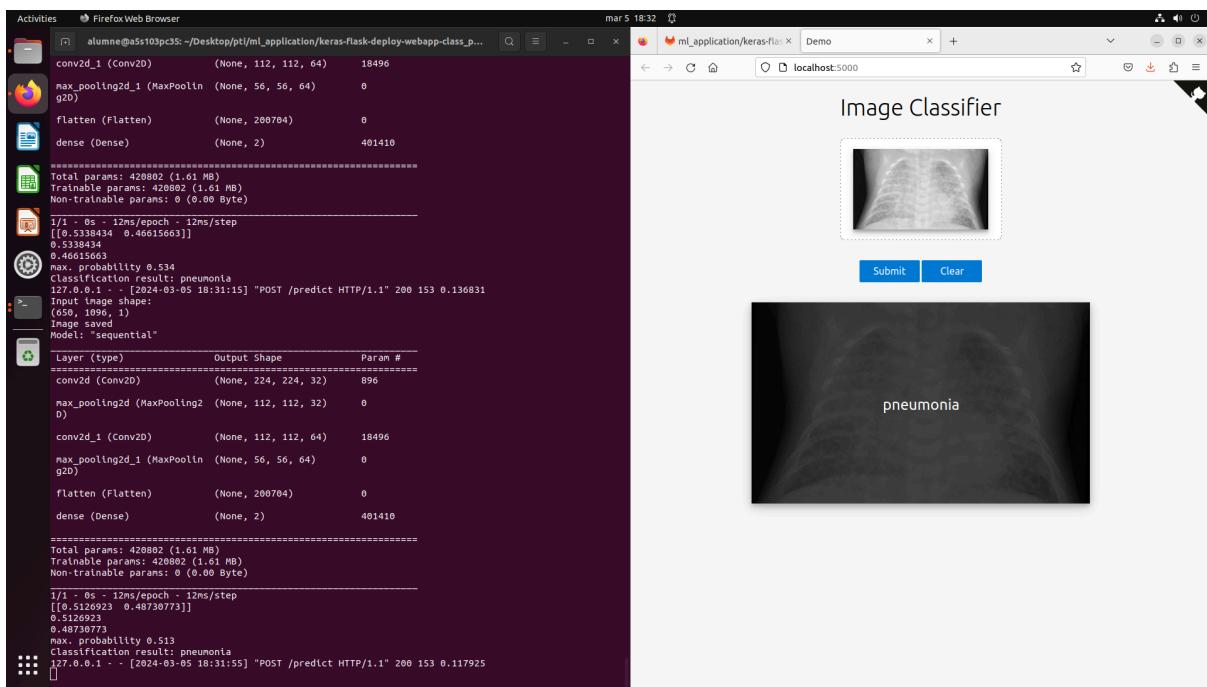
ml_application/keras-flask-deploy-webapp-class_pneu Demo localhost:5000

Image Classifier

Submit Clear

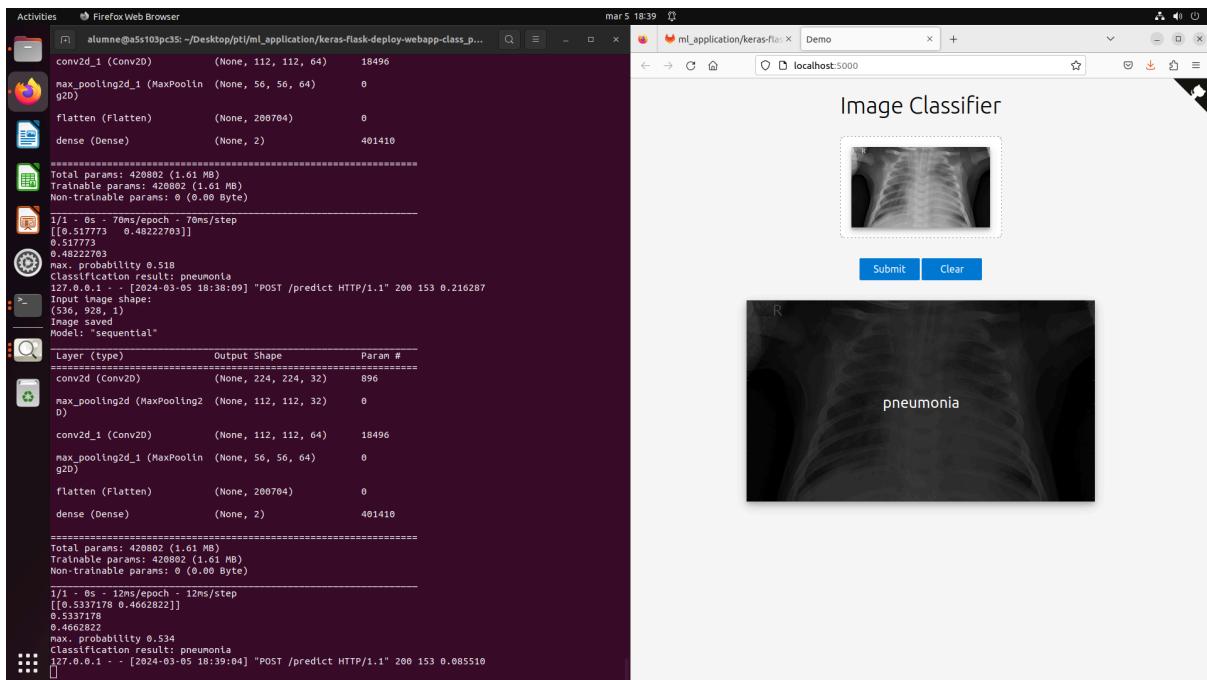
R normal

Aquesta segona imatge correspon a una foto d'una persona sana i també la classifica de forma correcta.



Aquesta última foto és d'una persona amb pneumònia la qual ha estat infectada amb un virus i també la classifica de forma correcta.

Les següents tres imatges són imatges testejades amb el model desentrenat per a fer-ho simplement hem descomentat la línia `MODEL_PATH = './keras_model_untrained'` i hem comentat la línia `MODEL_PATH = './keras_model_trained'`.



Aquesta primera imatge correspon a una pneumònia degut a una bacteria la qual ens la classifica de forma correcta.

Activities Firefox Web Browser

```
alumne@as5103pc35:~/Desktop/pt/ml_application/keras-flask-deploy-webapp-class_p... mar 5 18:39
conv2d_1 (Conv2D)           (None, 112, 112, 64)    18496
max_pooling2d_1 (MaxPoolin (None, 56, 56, 64)      0
flatten (Flatten)          (None, 280764)         0
dense (Dense)              (None, 2)                401410
=====
Total params: 420882 (1.61 MB)
Trainable params: 420882 (1.61 MB)
Non-trainable params: 0 (0.00 Byte)
[1/1 - 0s - 12ms/epoch - 12ms/step
[[0.5337178 0.4662822]]]
0.5337178
0.4662822
max. probability 0.534
Classification result: pneumonia
127.0.0.1 - - [2024-03-05 18:39:04] "POST /predict HTTP/1.1" 200 153 0.085510
Input image shape:
(1953, 2170, 1)
Image saved
Model: "sequential"
Layer (Type)          Output Shape        Param #
=====
conv2d (Conv2D)        (None, 224, 224, 32)   896
max_pooling2d (MaxPooling2 (None, 112, 112, 32) 0
conv2d_1 (Conv2D)       (None, 112, 112, 64)    18496
max_pooling2d_1 (MaxPoolin (None, 56, 56, 64)      0
g2d)
flatten (Flatten)       (None, 280764)         0
dense (Dense)          (None, 2)                401410
=====
Total params: 420882 (1.61 MB)
Trainable params: 420882 (1.61 MB)
Non-trainable params: 0 (0.00 Byte)
[1/1 - 0s - 12ms/epoch - 12ms/step
[[0.5405819 0.4594181]]]
0.5405819
0.4594181
max. probability 0.541
Classification result: pneumonia
127.0.0.1 - - [2024-03-05 18:39:51] "POST /predict HTTP/1.1" 200 153 0.473585
```

localhost:5000

Image Classifier

Submit Clear

pneumonia

Aquesta segona imatge correspon a una foto d'una persona sana i la classifica de forma incorrecta això és degut al fet que aquest model no ha estat entrenat com l'anterior i, per tant, comet força errors.

Activities Firefox Web Browser

```
alumne@as5103pc35:~/Desktop/pt/ml_application/keras-flask-deploy-webapp-class_p... mar 5 18:38
DATASET_EXTRACT keras_model_untrained README.md requirements.txt uploads
kerasfile LICENSE README.md util.py
(practfl) alumne@as5103pc35:~/Desktop/pt/ml_application/keras-flask-deploy-webapp-class_pneu$ vim app_pneu.py
(practfl) alumne@as5103pc35:~/Desktop/pt/ml_application/keras-flask-deploy-webapp-class_pneu$ python 3 app_pneu.py
[2024-03-05 18:36:53.436569]: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
[2024-03-05 18:36:53.457199]: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
/home/alumne/Desktop/pt/ml_application/federated-learning-network/main/practfl/lib/python3.10/site-p
ackages/scipy/_int_.py:155: UserWarning: A NumPy version >=1.18.5 and <1.25.0 is required for this
version of SciPy (detected version 1.26.4
warnings.warn("NumPy version %s-%s is not available. AVX512_VNNI FMA, in other operations, rebuild TensorFlow or Flow with the appropriate compiler flags.
[2024-03-05 18:36:58] "GET / HTTP/1.1" 200 3071 0.002723
[2024-03-05 18:36:58] "GET /static/main.css HTTP/1.1" 304 183 0.002826
[2024-03-05 18:36:58] "GET /static/main.js HTTP/1.1" 304 182 0.000472
Input image shape:
(752, 700, 1)
Image saved
Model: "sequential"
Layer (Type)          Output Shape        Param #
=====
conv2d (Conv2D)        (None, 224, 224, 32)   896
max_pooling2d (MaxPooling2 (None, 112, 112, 32) 0
conv2d_1 (Conv2D)       (None, 112, 112, 64)    18496
max_pooling2d_1 (MaxPoolin (None, 56, 56, 64)      0
g2d)
flatten (Flatten)       (None, 280764)         0
dense (Dense)          (None, 2)                401410
=====
Total params: 420882 (1.61 MB)
Trainable params: 420882 (1.61 MB)
Non-trainable params: 0 (0.00 Byte)
[1/1 - 0s - 70ms/epoch - 70ms/step
[[0.517773 0.48222703]]]
0.517773
0.48222703
max. probability 0.518
Classification result: pneumonia
127.0.0.1 - - [2024-03-05 18:38:09] "POST /predict HTTP/1.1" 200 153 0.216287
```

localhost:5000

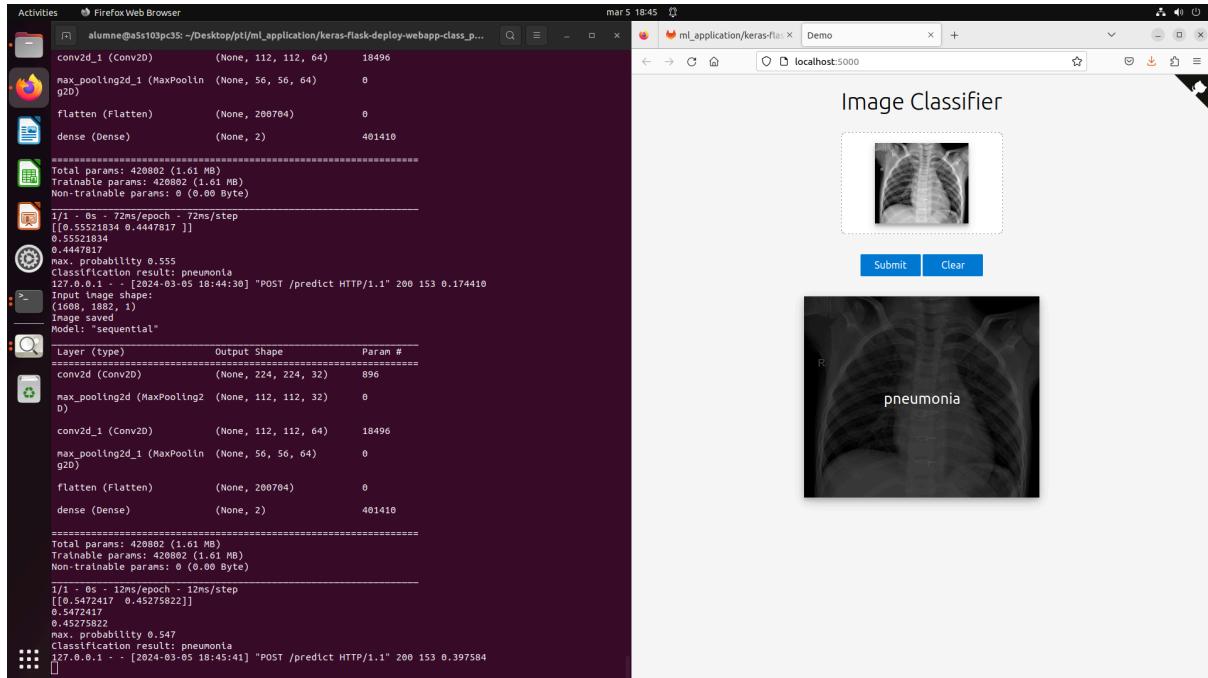
Image Classifier

Submit Clear

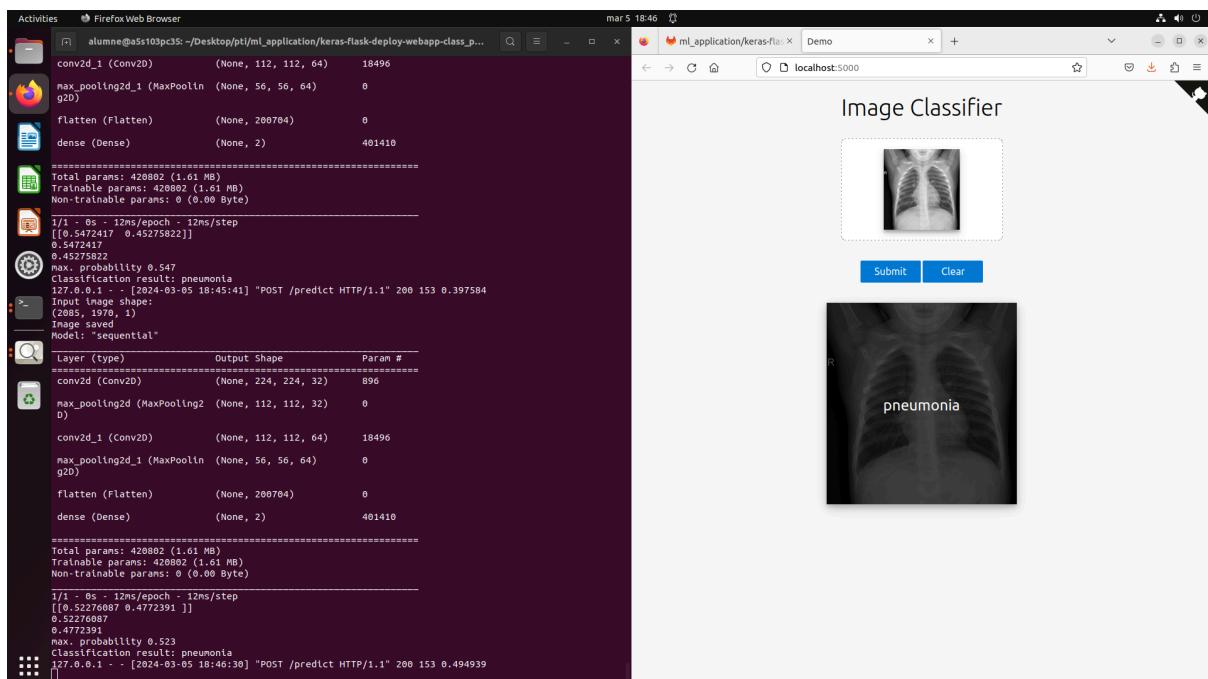
pneumonia

Aquesta última foto és d'una persona amb pneumònia la qual ha estat infectada amb un virus i també la classifica de forma correcta.

Les següents tres imatges són imatges testejades amb el model entrenat en la part 1 de la pràctica per a fer-ho simplemènt hem comentat la línia `MODEL_PATH = './keras_model_untrained` i hem comentat la línia `MODEL_PATH = '<path de la carpeta>`.



Aquesta primera imatge correspon a una pneumònia degut a una bacteria la qual ens la classifica de forma correcta.



Aquesta segona imatge correspon a una foto d'una persona sana i la classifica de forma incorrecta.

Activities Firefox Web Browser

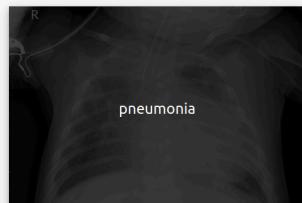
```
alumne@as5103pc35:~/Desktop/pti/ml_application/keras-flask-deploy-webapp-class_p... mar 5 18:44
DATASET EXTRACT keras_model_untrained apache requirements.txt uploads
Dockerfile LICENSE README.md static util.py
(practfl) alumne@as5103pc35:~/Desktop/pti/ml_application/keras-flask-deploy-webapp-class_pneu$ vim ap
(practfl) alumne@as5103pc35:~/Desktop/pti/ml_application/keras-flask-deploy-webapp-class_pneu$ python
3 app_pneu.py
2024-03-05 18:43:24.980889: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You
may see slightly different numerical results due to floating-point round-off errors from different co
mputation orders. To change the behavior, set the environment variable TF_ENABLE_ONEDNN_OPTS=1.
2024-03-05 18:43:25.081603: I tensorflow/core/framework/op_kernel.cc:182] This TensorFlow bina
ry is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild Tens
orflow with the appropriate compiler flags.
/home/alumne/Desktop/pti/ml_application/federated-learning-network-main/practfl/lib/python3.10/site-p
ackages/tensorflow/python/_pywrap_tensorflow_internal.so: undefined symbol: __cudf_feature_guard_cc1181] This TensorFlow bina
ry is optimized to use available CPU instructions in performance-critical operations.
Version of SciPy (detected version 1.26.4)
warnings.warn("A NumPy version >={np._inversion} and <{np._maxverstion}"
Model Loaded. Start serving...
127.0.0.1 - - [2024-03-05 18:43:39] "GET / HTTP/1.1" 200 3071 0.003146
127.0.0.1 - - [2024-03-05 18:43:39] "GET /static/main.css HTTP/1.1" 304 183 0.002494
127.0.0.1 - - [2024-03-05 18:43:39] "GET /static/main.js HTTP/1.1" 304 182 0.000394
Input image shape:
(640, 944, 1)
Image saved
Model: "sequential"
Layer (type) Output Shape Param #
=====
conv2d (Conv2D) (None, 224, 224, 32) 896
max_pooling2d (MaxPooling2D) (None, 112, 112, 32) 0
conv2d_1 (Conv2D) (None, 112, 112, 64) 18496
max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64) 0
flatten (Flatten) (None, 200704) 0
dense (Dense) (None, 2) 401410
=====
Total params: 420802 (1.61 MB)
Trainable params: 420802 (1.61 MB)
Non-trainable params: 0 (0.00 Byte)
[1/1 - 0s - 72ms/epoch - 72ms/step
[[0.55521034 0.4447817 ]]
0.55521034
0.4447817
max. probability 0.555
Classification result: pneumonia
127.0.0.1 - - [2024-03-05 18:44:30] "POST /predict HTTP/1.1" 200 153 0.174410
```

localhost:5000

Image Classifier



Submit Clear



R

pneumonia

Aquesta foto és d'una persona amb pneumònia la qual ha estat infectada amb un virus i també la classifica de forma correcta.

Activities Firefox Web Browser

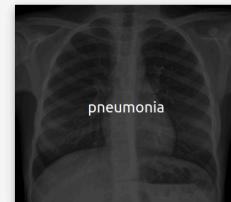
```
alumne@as5103pc35:~/Desktop/pti/ml_application/keras-flask-deploy-webapp-class_p... mar 5 18:47
conv2d_1 (Conv2D) (None, 112, 112, 64) 18496
max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64) 0
flatten (Flatten) (None, 200704) 0
dense (Dense) (None, 2) 401410
=====
Total params: 420802 (1.61 MB)
Trainable params: 420802 (1.61 MB)
Non-trainable params: 0 (0.00 Byte)
[1/1 - 0s - 12ms/epoch - 12ms/step
[[0.52276087 0.4772391 ]]
0.52276087
0.4772391
max. probability 0.523
Classification result: pneumonia
127.0.0.1 - - [2024-03-05 18:46:30] "POST /predict HTTP/1.1" 200 153 0.494939
127.0.0.1 - - [2024-03-05 18:46:30] "POST /predict HTTP/1.1" 200 153 0.494939
Input image shape:
(635, 944, 1)
Image saved
Model: "sequential"
Layer (type) Output Shape Param #
=====
conv2d (Conv2D) (None, 224, 224, 32) 896
max_pooling2d (MaxPooling2D) (None, 112, 112, 32) 0
conv2d_1 (Conv2D) (None, 112, 112, 64) 18496
max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64) 0
flatten (Flatten) (None, 200704) 0
dense (Dense) (None, 2) 401410
=====
Total params: 420802 (1.61 MB)
Trainable params: 420802 (1.61 MB)
Non-trainable params: 0 (0.00 Byte)
[1/1 - 0s - 41ms/epoch - 41ms/step
[[0.54189116 0.4581089 ]]
0.54189116
0.4581089
max. probability 0.542
Classification result: pneumonia
127.0.0.1 - - [2024-03-05 18:47:05] "POST /predict HTTP/1.1" 200 153 0.469049
```

localhost:5000

Image Classifier



Submit Clear



pneumonia

Aquesta quarta imatge correspon a una foto d'una persona sana i la classifica de forma incorrecta.

```

Activities Firefox Web Browser
alumne@as105pc35: ~/Desktop/plt/ml_application/keras-flask-deploy-webapp-class_p...
conv2d_1 (Conv2D)           (None, 112, 112, 64)    18496
max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64)    0
flatten (Flatten)          (None, 280764)        0
dense (Dense)              (None, 2)             401410
=====
Total params: 420802 (1.61 MB)
Trainable params: 420802 (1.61 MB)
Non-trainable params: 0 (0.00 Byte)
[0.54189116 0.4581089]
0.54189116
0.4581089
max. probability 0.542
Classification result: pneumonia
127.0.0.1 - [2024-03-05 18:47:05] "POST /predict HTTP/1.1" 200 153 0.469049
Input image shape:
(1706, 1741, 1)
Image saved
Model: "sequential"
Layer (Type)          Output Shape      Param #
conv2d (Conv2D)        (None, 224, 224, 32)   896
max_pooling2d (MaxPooling2D) (None, 112, 112, 32)   0
conv2d_1 (Conv2D)       (None, 112, 112, 64)    18496
max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64)    0
flatten (Flatten)      (None, 280764)        0
dense (Dense)          (None, 2)             401410
=====
Total params: 420802 (1.61 MB)
Trainable params: 420802 (1.61 MB)
Non-trainable params: 0 (0.00 Byte)
[0.5174016 0.48259845]
0.5174016
0.48259845
max. probability 0.517
Classification result: pneumonia
127.0.0.1 - [2024-03-05 18:47:58] "POST /predict HTTP/1.1" 200 153 0.301324

```

The terminal shows the model architecture and two predictions. The first prediction is for an X-ray image showing pneumonia, with a probability of 0.542. The second prediction is also for pneumonia, with a probability of 0.517. The web application interface shows a normal X-ray image with the label 'pneumonia'.

Aquesta cinquena imatge correspon a una foto d'una persona sana i la classifica de forma incorrecta.

```

Activities Firefox Web Browser
alumne@as105pc35: ~/Desktop/plt/ml_application/keras-flask-deploy-webapp-class_p...
conv2d_1 (Conv2D)           (None, 112, 112, 64)    18496
max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64)    0
flatten (Flatten)          (None, 280764)        0
dense (Dense)              (None, 2)             401410
=====
Total params: 420802 (1.61 MB)
Trainable params: 420802 (1.61 MB)
Non-trainable params: 0 (0.00 Byte)
[0.5391213 0.46087867]
0.5391213
0.46087867
max. probability 0.539
Classification result: pneumonia
127.0.0.1 - [2024-03-05 18:50:01] "POST /predict HTTP/1.1" 200 153 0.434051
Input image shape:
(1706, 1740, 1)
Image saved
Model: "sequential"
Layer (Type)          Output Shape      Param #
conv2d (Conv2D)        (None, 224, 224, 32)   896
max_pooling2d (MaxPooling2D) (None, 112, 112, 32)   0
conv2d_1 (Conv2D)       (None, 112, 112, 64)    18496
max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64)    0
flatten (Flatten)      (None, 280764)        0
dense (Dense)          (None, 2)             401410
=====
Total params: 420802 (1.61 MB)
Trainable params: 420802 (1.61 MB)
Non-trainable params: 0 (0.00 Byte)
[0.49920914 0.5017919]
0.49920914
0.5017919
max. probability 0.502
Classification result: normal
127.0.0.1 - [2024-03-05 18:50:56] "POST /predict HTTP/1.1" 200 150 0.270934

```

The terminal shows the model architecture and two predictions. The first prediction is for an X-ray image showing pneumonia, with a probability of 0.539. The second prediction is for a normal X-ray, with a probability of 0.502. The web application interface shows a normal X-ray image with the label 'normal'.

Aquesta sisena imatge correspon a una foto d'una persona sana i la classifica de forma incorrecta.

Per tant, podem concloure que el model entrenat a l'apartat anterior li caldríen més rondes d'entrenament per a ser un model fiable el qual els doctors hi poguessin confiar això és deu a què com hem mencionat anteriorment no ha estat prou entrenat i comet alguns errors sobretot quan la persona està sana. A feines de visió per als computadors com ara la classificació d'imatges.

Part 2: Entrenament dels models.

En aquesta pràctica a nosaltres ens va fascinar molt com els models eren capaços de discriminar quines radiografies tenien pulmonia i quines estaven sanes de forma correcta gràcies a l'entrenament rebut per això hem volgut indagar en aquest tema, és a dir, sobre com s'entretenen aquests models per poder trobar la solució correcta. Això, pot comportar grans avenços en un sector on hi ha escassetat d'especialistes i en conseqüència els resultats i diagnòstic pot tardar setmanes o inclús mesos. Amb l'ús d'aquest tipus de models podem ser capaços de detectar aquestes malalties de forma primerenca. En aquest cas nosaltres hem utilitzat un model anomenat Keras el qual consisteix en una xarxa neuronal API per a python que està integrada a Tensorflow, aquesta s'usa per a construir models de machine learning com podria ser el nostre cas. La principal diferència entre Keras i Tensorflow és que el segon consisteix en un conjunt de llibreries open-source per a crear i treballar amb xarxes neuronals, en canvi, Keras és una API d'alt nivell que còrrer per sobre de Tensorflow, ens simplifica la implementació de xarxes neuronals complexes.

Quant a la recopilació de dades per al seu entrenament el primer que fem serà recopilar dades d'un conjunt d'imatges de radiografies toràciques proporcionades a través

del

link

[https://data.mendeley.com/public-files/datasets/rscbjbr9sj/files/f12eaf6d-6023-432f-a_cc9-80c9d7393433/file downloaded](https://data.mendeley.com/public-files/datasets/rscbjbr9sj/files/f12eaf6d-6023-432f-a_cc9-80c9d7393433/file_downloaded), i en cada una d'elles estan etiquetats si és o no és una pneumònia en la carpeta d'entrenament. Aquestes imatges es dividiran en conjunts d'entrenament, validació i prova. El primer conjunt s'utilitza per ajustar els pesos del model, el conjunt d'entrenament s'usa per a ajustar els hiperparàmetres del model que són les configuracions que no depenen directament del conjunt de dades durant l'entrenament sinó que s'estableixen en començar aquest. Aquests hiperparàmetres poden afectar el comportament i rendiment del model durant l'entrenament, a la prediccio i a l'hora d'influir a la capacitat del model per aprendre i generalitzar correctament a partir de les dades. Quan construïm el model hem de partir de què Keras és un model de xarxa neuronal convolucional, que és una arquitectura comunament usada per a les feines de visió per als computadors com la detecció d'objectes en imatges. Aquest model es compon d'aquestes capes convolucionals, cases d'agrupació i capes totalment connectades, aquestes capes es combinen per a extreure les característiques, més importants de les imatges i realitzar la classificació. Per a entendre, però com funciona aquesta classificació hem d'anar un pas més enllà i entendre cadascuna de les capes i el seu funcionament dins el model. Les primeres capes convolucionals són fonamentals, ja que s'encarreguen que cada neurona en una capa convolucional estigui connectada només a una petita regió local (camp receptiu) en lloc d'estar connectades totes a l'entrada. Això ens permet que la xarxa aprengui patrons locals en les imatges. Durnat l'entrenament, aquestes capes apliquen una sèrie de filtres a l'entrada per a detectar característiques com ara les textures i formes en les diferents imatges. Cada filtre il·lisca sobre la imatge d'entrada i efectua operacions de convolució,

aquesta operació consisteix a superposar un filtre en diferents ubicacions de la imatge per produir un nou valor a la matriu de sortida (mapa de característiques). Cada posició es multiplica per cada element del filtre pels píxels corresponents a la regió de la imatge coberta pel filtre, aquests productes es sumen i el resultat es col·loca en la posició corresponent de la matriu. Aquest procés es repeteix per a totes les posicions de la imatge, generant un mapa de característiques que ressalta la presència de la característica específica a ressaltar.

La capa d'agrupació s'usa per a la dimensionalitat de les característiques i fer que la seva representació sigui més mal·leable. També s'fa ús de l'operació d'agrupació com l'agrupació màxima o l'agrupació mitjana, que redueix la mida dels mapes conservant les seves característiques més importants. A més a més, l'agrupació ajuda al fet que el model sigui més invariants petites translacions, rotacions i deformacions de les característiques extrems.

Finalment, tenim les capes totalment connectades aquestes estan compostes de neurones que connecten a totes les neurones de la capa anterior. Aquestes s'utilitzen per a combinar les característiques obtingudes de les dues capes anteriors en una representació final per a la classificació. Durant l'entrenament, aquesta capa ajusta els pesos de les connexions entre neurones per aprendre patrons més complexes i realitzar la classificació final. En conjunt aquestes capes permeten que una CNN aprengui representacions jeràrquiques de les imatges començant des de característiques molt simples com la textura fins a més complexes i abstractes en capes més profundes. Això fa que la CNN siguin tan efectives per a feines de visió per als computadors com ara la classificació d'imatges.