Ejercicio 2

October 30, 2022

1 2 (30 puntos) Perceptrón multi-capa para detección de glaucoma en imágenes de fondo de ojo

1. Modifique el perceptrón multi-capa y la red convolucional implementadas en el codigo base provisto. Explique las modificaciones necesarias para implementar que ambos modelos funcionen para el objetivo especificado.

1.1 Imports

[ ]:
```
from __future__ import print_function
import argparse
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torchvision import datasets, transforms
from torch.optim.lr_scheduler import StepLR
import numpy as np
import torch
import torchvision
import matplotlib.pyplot as plt
from time import time
from torchvision import datasets, transforms
from torchvision.io import read_image
from torch import nn, optim
from matplotlib import cm
from matplotlib import style
from locale import normalize
from matplotlib.pyplot import imshow

%matplotlib widget
style.use('default')
```

1.2 Parámetros

[ ]:
```
lr = 0.01 # Learning rate empleado para la función de optimización SGD
epochs = 15 # Cantidad de iteraciones que se ejecutaran por cada intento
attempts = 10 # Cantidad de intentos totales que se ejecutaran (en cada uno de␣
 ↪los cuales se creará el modelo, entrenará y obtendrán datos de efectividad)
img_size = 227 # Tamaño asignado para el width y height de las imágenes para␣
 ↪entrenar y probar.
batch_size = 32 # Cantidad de imágenes que serán procesadas por lote.
criterion = nn.NLLLoss() # Función de perdida que se calculará individualmente␣
 ↪para cada iteración y que se suma al total del intento
ruta_base = r'.\\DatosDePrueba\\' # Ruta de la computadora a partir de la cual␣
 ↪se accede a los datos de prueba
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu') #␣
 ↪Dispositivo en el que se ejecutará
transform = transforms.Compose([transforms.ToTensor(), transforms.
 ↪Resize((img_size, img_size))]) # Transformaciones aplicadas a los sets de␣
 ↪entrenamiento y pruebas (todas las imágenes se ajustan al cuadrado img_size)
```

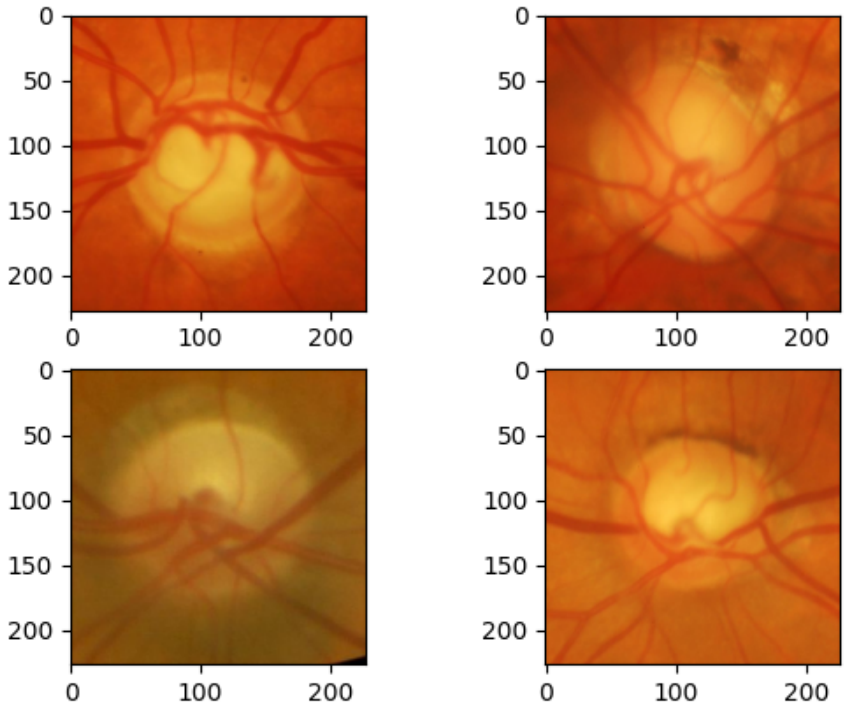
1.3 Cargado de datos de entrenamiento y prueba

[ ]:
```
def load_data(nom_carpeta): # Carga imágenes desde el folder según el nombre␣
 ↪que se indique
    set = torchvision.datasets.ImageFolder(ruta_base + nom_carpeta, transform =␣
 ↪transform)
    loader = torch.utils.data.DataLoader(set, batch_size = batch_size, shuffle␣
 ↪= True)
    return loader

def load_test_train_data(): # Carga imágenes de entrenamiento y pruebas
    trainloader = load_data('train')
    testloader = load_data('test')
    return trainloader, testloader

trainloader, testloader = load_test_train_data()
dataiter = iter(trainloader)
images, labels = dataiter.next()


print('[Batch_size, channels, img_size, img_size] =,', images.mT.shape)

fig, axs = plt.subplots(2,2)
axs[0,0].imshow(images[0].T.numpy().squeeze(), cmap='gray_r')
axs[0,1].imshow(images[1].T.numpy().squeeze(), cmap='gray_r')
axs[1,0].imshow(images[2].T.numpy().squeeze(), cmap='gray_r')
axs[1,1].imshow(images[3].T.numpy().squeeze(), cmap='gray_r')
```

```
[Batch_size, channels, img_size, img_size] =, torch.Size([32, 3, 227, 227])
```

```
C:\Users\XPC\AppData\Local\Temp\ipykernel_46556\2524731925.py:19: UserWarning:
The use of `x.T` on tensors of dimension other than 2 to reverse their shape is
deprecated and it will throw an error in a future release. Consider `x.mT` to
transpose batches of matricesor `x.permute(*torch.arange(x.ndim - 1, -1, -1))`
to reverse the dimensions of a tensor. (Triggered internally at
..\aten\src\ATen\native\TensorShape.cpp:2985.)
  axs[0,0].imshow(images[0].T.numpy().squeeze(), cmap='gray_r')
```

[ ]:
```
<matplotlib.image.AxesImage at 0x20747706e60>
```



1.4 Creando el modelo del perceptrón multicapa

[ ]:
```
def create_MLP_model(verbose = True): # Creación del modelo
  model = nn.Sequential(
    nn.Linear(img_size*img_size*3,128), # Empleando como entrada las␣
↪dimensiones de la imagen (w * h) multiplicado por los 3 canales de color que␣
↪se están utilizando => 154587 = 227 * 227 * 3
    nn.Sigmoid(),
    nn.Linear(128,64),
    nn.Sigmoid(),
    nn.Linear(64,2),
```

```
    nn.LogSoftmax(dim=1))

  if (verbose): print("Running on device: ", device)
  model.to(device)
  return model

mlp_model = create_MLP_model()
print("MLP model")
print(mlp_model)
```

```
Running on device:  cuda
MLP model
Sequential(
  (0): Linear(in_features=154587, out_features=128, bias=True)
  (1): Sigmoid()
  (2): Linear(in_features=128, out_features=64, bias=True)
  (3): Sigmoid()
  (4): Linear(in_features=64, out_features=2, bias=True)
  (5): LogSoftmax(dim=1)
)
```

1.5 Entrenando el modelo

[ ]:
```
def train_model(model, criterion, epochs, lr, verbose = True):
    start = time()
    epochs_list = []
    running_loss_list= []
    optimizer = optim.SGD(model.parameters(), lr= lr, momentum=0.9)

    for e in range(epochs):
        running_loss = 0

        for images, labels in trainloader: # Ejecución de la iteración por lotes
            images = images.to(device) # Se preparan las imágenes del lote en␣
 ↪el device usado
            labels = labels.to(device)
            images = images.view(images.shape[0], -1)

            optimizer.zero_grad()
            output = model(images) # Aplicando el modelo para cada batch
            loss = criterion(output, labels) # Y calculando el error asociado
            loss.backward() # Aprendizaje por backpropagation
            optimizer.step() # weights optimization
            running_loss += loss.item() # Se suma la pérdida para el total de␣
 ↪la iteración

        else:
```

```
            if (verbose): print("Epoch {} - Training loss: {}".format(e,␣
 ↪running_loss/len(trainloader)))

    if (verbose): print("\nTraining Time (in minutes) =",(time()-start)/60)
    return model

print("Training MLP model")
print("Device", next(mlp_model.parameters()).is_cuda)

mlp_model = train_model(mlp_model, criterion, epochs, lr)
```

```
Training MLP model
Device True
Epoch 0 - Training loss: 0.6811960376799107
Epoch 1 - Training loss: 0.6621619202196598
Epoch 2 - Training loss: 0.6019255146384239
Epoch 3 - Training loss: 0.5343579966574907
Epoch 4 - Training loss: 0.45502882823348045
Epoch 5 - Training loss: 0.3947829157114029
Epoch 6 - Training loss: 0.47907054238021374
Epoch 7 - Training loss: 0.378512155264616
Epoch 8 - Training loss: 0.334895103238523
Epoch 9 - Training loss: 0.296166880056262
Epoch 10 - Training loss: 0.2786378934979439
Epoch 11 - Training loss: 0.30313558131456375
Epoch 12 - Training loss: 0.35534085519611835
Epoch 13 - Training loss: 0.2813700754195452
Epoch 14 - Training loss: 0.2712722350843251

Training Time (in minutes) = 0.6175908843676249
```

1.6 Probando el modelo

[ ]:
```
def test_model_mlp(testloader, model, verbose = True):
    correct_rate, false_negative_rate, all_count = 0, 0, 0

    for images,labels in testloader:
      images = images.to(device)  # Se preparan las imágenes del lote en el␣
 ↪device usado
      labels = labels.to(device)

      for i in range(len(labels)): # se itera sobre los índices de targets
        img = images[i].view(1, img_size*img_size*3)
        with torch.no_grad():
            logps = model(img)
        ps = torch.exp(logps)
        probab = list(ps.cpu().numpy()[0])
```

```
        pred_label = probab.index(max(probab)) # Se obtiene el target predicho␣
↪de la iteración actual
        true_label = labels.cpu().numpy()[i] # Se obtiene el target correcto de␣
↪la iteración actual

        if (true_label == pred_label): correct_rate += 1 # Predicción correcta␣
↪si igual a target
        else:
          if (pred_label == 0): false_negative_rate += 1 # Falso negativo si␣
↪predicción es 0 pero correcto 1

        all_count += 1

    if (verbose):
      print("Images Tested =", all_count)
      print("Correct Tests =", correct_rate)
      print("False Positive Tests =", (all_count - correct_rate) -␣
↪false_negative_rate)
      print("False Negative Tests =", false_negative_rate)
      print("\nModel Accuracy (Average) =", np.round((correct_rate/
↪all_count)*100,4),"%")

    return correct_rate, false_negative_rate, all_count

print("Testing MLP model")
res = test_model_mlp(testloader, mlp_model)
```

```
Testing MLP model
Images Tested = 210
Correct Tests = 120
False Positive Tests = 90
False Negative Tests = 0

Model Accuracy (Average) = 57.1429 %
```

1.7 Ejecuciones consecutivas de entrenamiento y pruebas

a) Entrene el perceptrón multi-capa usando el conjunto de datos completo para entrenamiento y validación. Calibre los hiper-parámetros necesarios para obtener los mejores resultados posibles y reportelos. Ejecute el entrenamiento 10 veces por 15 épocas por corrida, y reporte la tasa de aciertos tasa de aciertos, falsos positivos y falsos negativos promedio y su desviación estándar para esas 10 corridas.

[ ]:
```
print('{} training and testing attempts:\n'.format(attempts))
aciertos = []

for i in range(attempts):
```

```
    mlp_model = create_MLP_model(False)
    mlp_model = train_model(mlp_model, criterion, epochs, lr, False)
    correct_rate, false_negative_rate, all_count = test_model_mlp(testloader,␣
 ↪mlp_model, False)
    aciertos.append(correct_rate)

    print('Results of attempt #{}:'.format(i))
    print('\tCorrect: {}\tFalse Positive Tests: {}\tFalse Negative Tests:␣
 ↪{}\tModel Accuracy: {}\n'.format(correct_rate, (all_count - correct_rate) -␣
 ↪false_negative_rate, false_negative_rate, correct_rate/all_count))

resultados = torch.tensor(aciertos)
print('Average correct tests: {}'.format(torch.mean(resultados.double()).
 ↪item()))
print('Standard deviation: {}\n'.format(torch.std(resultados.double()).item()))
```

```
10 training and testing attempts:

Results of attempt #0:
        Correct: 189    False Positive Tests: 13        False Negative Tests: 8
Model Accuracy: 0.9

Results of attempt #1:
        Correct: 180    False Positive Tests: 25        False Negative Tests: 5
Model Accuracy: 0.8571428571428571

Results of attempt #2:
        Correct: 187    False Positive Tests: 15        False Negative Tests: 8
Model Accuracy: 0.8904761904761904

Results of attempt #3:
        Correct: 187    False Positive Tests: 4 False Negative Tests: 19
Model Accuracy: 0.8904761904761904

Results of attempt #4:
        Correct: 165    False Positive Tests: 42        False Negative Tests: 3
Model Accuracy: 0.7857142857142857

Results of attempt #5:
        Correct: 180    False Positive Tests: 24        False Negative Tests: 6
Model Accuracy: 0.8571428571428571

Results of attempt #6:
        Correct: 185    False Positive Tests: 11        False Negative Tests: 14
Model Accuracy: 0.8809523809523809

Results of attempt #7:
```

```
        Correct: 161    False Positive Tests: 46        False Negative Tests: 3
Model Accuracy: 0.7666666666666667

Results of attempt #8:
        Correct: 188    False Positive Tests: 4 False Negative Tests: 18
Model Accuracy: 0.8952380952380953

Results of attempt #9:
        Correct: 153    False Positive Tests: 57        False Negative Tests: 0
Model Accuracy: 0.7285714285714285

Average correct tests: 177.5
Standard deviation: 12.99786307223016
```

2 Ejecución de corridas sobre el modelo MLP empleando distintos LR cada dos corridas

2.1 LR 0.1

2.1.1 Intento 1

Training MLP model

- Epoch 0 - Training loss: 0.7369163818657398
- Epoch 1 - Training loss: 0.6828029863536358
- Epoch 2 - Training loss: 0.6817193254828453
- Epoch 3 - Training loss: 0.6662973240017891
- Epoch 4 - Training loss: 0.6442956253886223
- Epoch 5 - Training loss: 0.7077872566878796
- Epoch 6 - Training loss: 0.6701737195253372
- Epoch 7 - Training loss: 0.6801279932260513
- Epoch 8 - Training loss: 0.6780841983854771
- Epoch 9 - Training loss: 0.6868789345026016
- Epoch 10 - Training loss: 0.6634330414235592
- Epoch 11 - Training loss: 0.6495210751891136
- Epoch 12 - Training loss: 0.6918781399726868
- Epoch 13 - Training loss: 0.7025960497558117
- Epoch 14 - Training loss: 0.6821348257362843

Training Time (in minutes) = 0.6270199020703634

Testing MLP model

- Images Tested = 210
- Correct Tests = 105
- False Positive Tests = 105
- False Negative Tests = 0

Model Accuracy (Average) = 50.0 %

2.1.2 Intento 2

Training MLP model

- Epoch 0 - Training loss: 0.6815511770546436
- Epoch 1 - Training loss: 0.7036696895956993
- Epoch 2 - Training loss: 0.6777684912085533
- Epoch 3 - Training loss: 0.5605829171836376
- Epoch 4 - Training loss: 0.6949820667505264
- Epoch 5 - Training loss: 0.6932078413665295
- Epoch 6 - Training loss: 0.6773754023015499
- Epoch 7 - Training loss: 0.7096707485616207
- Epoch 8 - Training loss: 0.6872112117707729
- Epoch 9 - Training loss: 0.6869790367782116
- Epoch 10 - Training loss: 0.6942282542586327
- Epoch 11 - Training loss: 0.7144105024635792
- Epoch 12 - Training loss: 0.6969177834689617
- Epoch 13 - Training loss: 0.722785871475935
- Epoch 14 - Training loss: 0.7197191417217255

Training Time (in minutes) = 0.6199134389559428

Testing MLP model

- Images Tested = 210
- Correct Tests = 105
- False Positive Tests = 105
- False Negative Tests = 0

Model Accuracy (Average) = 50.0 %

2.2 LR 0.01

2.2.1 Intento 1

Training MLP model

- Epoch 0 - Training loss: 0.6827975213527679
- Epoch 1 - Training loss: 0.6605164743959904
- Epoch 2 - Training loss: 0.6082503236830235
- Epoch 3 - Training loss: 0.5294231809675694
- Epoch 4 - Training loss: 0.4830891713500023
- Epoch 5 - Training loss: 0.4102689679712057
- Epoch 6 - Training loss: 0.37855312693864107
- Epoch 7 - Training loss: 0.3209534380584955
- Epoch 8 - Training loss: 0.31282537151128054
- Epoch 9 - Training loss: 0.28880161326378584
- Epoch 10 - Training loss: 0.2564556635916233
- Epoch 11 - Training loss: 0.2673725155182183
- Epoch 12 - Training loss: 0.260069502517581
- Epoch 13 - Training loss: 0.2765366407111287

- Epoch 14 - Training loss: 0.29861110355705023

Training Time (in minutes) = 0.61580970287323

Testing MLP model

- Images Tested = 210
- Correct Tests = 185
- False Positive Tests = 22
- False Negative Tests = 3

Model Accuracy (Average) = 88.0952 %

2.2.2 Intento 2

Training MLP model

- Epoch 0 - Training loss: 0.7057710886001587
- Epoch 1 - Training loss: 0.6802574507892132
- Epoch 2 - Training loss: 0.6727069728076458
- Epoch 3 - Training loss: 0.6631590500473976
- Epoch 4 - Training loss: 0.6375782452523708
- Epoch 5 - Training loss: 0.5831635408103466
- Epoch 6 - Training loss: 0.5207257680594921
- Epoch 7 - Training loss: 0.4437313415110111
- Epoch 8 - Training loss: 0.36804841458797455
- Epoch 9 - Training loss: 0.3788630049675703
- Epoch 10 - Training loss: 0.3542727045714855
- Epoch 11 - Training loss: 0.34089745953679085
- Epoch 12 - Training loss: 0.3315946366637945
- Epoch 13 - Training loss: 0.28785437252372503
- Epoch 14 - Training loss: 0.2747528455220163

Training Time (in minutes) = 0.6155428210894267

Testing MLP model

- Images Tested = 210
- Correct Tests = 167
- False Positive Tests = 0
- False Negative Tests = 43

Model Accuracy (Average) = 79.5238 %

2.3 LR 0.03

2.3.1 Intento 1

Training MLP model

- Epoch 0 - Training loss: 0.7078320719301701
- Epoch 1 - Training loss: 0.6655924059450626
- Epoch 2 - Training loss: 0.5964057557284832

- Epoch 3 - Training loss: 0.5449526645243168
- Epoch 4 - Training loss: 0.5642602499574423
- Epoch 5 - Training loss: 0.4499690402299166
- Epoch 6 - Training loss: 0.4313625395298004
- Epoch 7 - Training loss: 0.41177461203187704
- Epoch 8 - Training loss: 0.3985733538866043
- Epoch 9 - Training loss: 0.45222795754671097
- Epoch 10 - Training loss: 0.4886812958866358
- Epoch 11 - Training loss: 0.63541179895401
- Epoch 12 - Training loss: 0.6751604080200195
- Epoch 13 - Training loss: 0.6382348835468292
- Epoch 14 - Training loss: 0.6463134028017521

Training Time (in minutes) = 0.5967423796653748

Testing MLP model

- Images Tested = 210
- Correct Tests = 126
- False Positive Tests = 84
- False Negative Tests = 0

Model Accuracy (Average) = 60.0 %

2.3.2 Intento 2

Training MLP model

- Epoch 0 - Training loss: 0.6665529124438763
- Epoch 1 - Training loss: 0.6589692607522011
- Epoch 2 - Training loss: 0.6091506369411945
- Epoch 3 - Training loss: 0.4977915268391371
- Epoch 4 - Training loss: 0.4174402579665184
- Epoch 5 - Training loss: 0.3881151517853141
- Epoch 6 - Training loss: 0.4580058716237545
- Epoch 7 - Training loss: 0.4472315367311239
- Epoch 8 - Training loss: 0.39315566048026085
- Epoch 9 - Training loss: 0.5275244060903788
- Epoch 10 - Training loss: 0.44969186559319496
- Epoch 11 - Training loss: 0.5123887434601784
- Epoch 12 - Training loss: 0.4593819808214903
- Epoch 13 - Training loss: 0.6415568217635155
- Epoch 14 - Training loss: 0.6992569081485271

Training Time (in minutes) = 0.6200714071591695

Testing MLP model

- Images Tested = 210
- Correct Tests = 117
- False Positive Tests = 93

- False Negative Tests = 0

Model Accuracy (Average) = 55.7143 %

2.4 LR 0.001

2.4.1 Intento 1

Training MLP model

- Epoch 0 - Training loss: 0.6840277686715126
- Epoch 1 - Training loss: 0.6728365309536457
- Epoch 2 - Training loss: 0.6657264605164528
- Epoch 3 - Training loss: 0.661868829280138
- Epoch 4 - Training loss: 0.655780304223299
- Epoch 5 - Training loss: 0.6464584358036518
- Epoch 6 - Training loss: 0.6451276578009129
- Epoch 7 - Training loss: 0.6388522908091545
- Epoch 8 - Training loss: 0.6317578107118607
- Epoch 9 - Training loss: 0.622311670333147
- Epoch 10 - Training loss: 0.6149414479732513
- Epoch 11 - Training loss: 0.6064675077795982
- Epoch 12 - Training loss: 0.5985792241990566
- Epoch 13 - Training loss: 0.5939729511737823
- Epoch 14 - Training loss: 0.586756456643343

Training Time (in minutes) = 0.6210965474446615

Testing MLP model

- Images Tested = 210
- Correct Tests = 129
- False Positive Tests = 79
- False Negative Tests = 2

Model Accuracy (Average) = 61.4286 %

2.4.2 Intento 2

Training MLP model

- Epoch 0 - Training loss: 0.6771115027368069
- Epoch 1 - Training loss: 0.6740257628262043
- Epoch 2 - Training loss: 0.6744579784572124
- Epoch 3 - Training loss: 0.6677091605961323
- Epoch 4 - Training loss: 0.6612282618880272
- Epoch 5 - Training loss: 0.6544647552073002
- Epoch 6 - Training loss: 0.6521985307335854
- Epoch 7 - Training loss: 0.6469268091022968
- Epoch 8 - Training loss: 0.6415448822081089
- Epoch 9 - Training loss: 0.637713972479105
- Epoch 10 - Training loss: 0.6335229761898518

- Epoch 11 - Training loss: 0.6256400793790817
- Epoch 12 - Training loss: 0.6197583377361298
- Epoch 13 - Training loss: 0.6133669465780258
- Epoch 14 - Training loss: 0.6117302030324936

Training Time (in minutes) = 0.6143083612124125

Testing MLP model

- Images Tested = 210
- Correct Tests = 113
- False Positive Tests = 97
- False Negative Tests = 0

Model Accuracy (Average) = 53.8095 %

2.5 LR 0.003

2.5.1 Intento 1

Training MLP model

- Epoch 0 - Training loss: 0.6874632090330124
- Epoch 1 - Training loss: 0.6731723919510841
- Epoch 2 - Training loss: 0.6625744327902794
- Epoch 3 - Training loss: 0.6509069390594959
- Epoch 4 - Training loss: 0.6418970599770546
- Epoch 5 - Training loss: 0.626440305262804
- Epoch 6 - Training loss: 0.6111289225518703
- Epoch 7 - Training loss: 0.5899031274020672
- Epoch 8 - Training loss: 0.5702399350702763
- Epoch 9 - Training loss: 0.5409582275897264
- Epoch 10 - Training loss: 0.5159615613520145
- Epoch 11 - Training loss: 0.48713919520378113
- Epoch 12 - Training loss: 0.45356206968426704
- Epoch 13 - Training loss: 0.4310105964541435
- Epoch 14 - Training loss: 0.4148829597979784

Training Time (in minutes) = 0.6126735369364421

Testing MLP model

- Images Tested = 210
- Correct Tests = 155
- False Positive Tests = 52
- False Negative Tests = 3

Model Accuracy (Average) = 73.8095 %

2.5.2 Intento 2

Training MLP model

- Epoch 0 - Training loss: 0.6764406636357307
- Epoch 1 - Training loss: 0.6625492610037327
- Epoch 2 - Training loss: 0.6434317827224731
- Epoch 3 - Training loss: 0.6279043667018414
- Epoch 4 - Training loss: 0.6049855165183544
- Epoch 5 - Training loss: 0.583499364554882
- Epoch 6 - Training loss: 0.5568369124084711
- Epoch 7 - Training loss: 0.5259818211197853
- Epoch 8 - Training loss: 0.4958261363208294
- Epoch 9 - Training loss: 0.4689131695777178
- Epoch 10 - Training loss: 0.43894270434975624
- Epoch 11 - Training loss: 0.43525825813412666
- Epoch 12 - Training loss: 0.398719334974885
- Epoch 13 - Training loss: 0.3785032369196415
- Epoch 14 - Training loss: 0.3820067588239908

Training Time (in minutes) = 0.6345779856046041

Testing MLP model

- Images Tested = 210
- Correct Tests = 179
- False Positive Tests = 13
- False Negative Tests = 18

Model Accuracy (Average) = 85.2381 %

3 Ejecución de corridas sobre el modelo MLP empleando los mismos LR en todas las corridas (LR = 0.01)

3.1 Ejecución #1

3.1.1 Training MLP model

- Epoch 0 - Training loss: 0.6755483783781528
- Epoch 1 - Training loss: 0.6295309439301491
- Epoch 2 - Training loss: 0.5783903300762177
- Epoch 3 - Training loss: 0.5343762598931789
- Epoch 4 - Training loss: 0.4437193498015404
- Epoch 5 - Training loss: 0.39533296041190624
- Epoch 6 - Training loss: 0.41546942479908466
- Epoch 7 - Training loss: 0.3631477542221546
- Epoch 8 - Training loss: 0.36750431172549725
- Epoch 9 - Training loss: 0.3276412822306156
- Epoch 10 - Training loss: 0.3217639746144414
- Epoch 11 - Training loss: 0.31924678198993206
- Epoch 12 - Training loss: 0.3261508094146848
- Epoch 13 - Training loss: 0.26522853458300233
- Epoch 14 - Training loss: 0.27376265823841095

Training Time (in minutes) = 0.6461522181828817

3.1.2 Testing MLP model

- Images Tested = 210
- Correct Tests = 184
- False Positive Tests = 20
- False Negative Tests = 6

Model Accuracy (Average) = 87.619 %

3.2 Ejecución #2

3.2.1 Training MLP model

- Epoch 0 - Training loss: 0.6789938472211361
- Epoch 1 - Training loss: 0.6559620313346386
- Epoch 2 - Training loss: 0.5979436337947845
- Epoch 3 - Training loss: 0.5186452232301235
- Epoch 4 - Training loss: 0.4913574047386646
- Epoch 5 - Training loss: 0.4588260278105736
- Epoch 6 - Training loss: 0.42552139796316624
- Epoch 7 - Training loss: 0.38591246493160725
- Epoch 8 - Training loss: 0.3318267595022917
- Epoch 9 - Training loss: 0.3677603965625167
- Epoch 10 - Training loss: 0.3285478949546814
- Epoch 11 - Training loss: 0.30944768711924553
- Epoch 12 - Training loss: 0.31867357436567545
- Epoch 13 - Training loss: 0.2513004643842578
- Epoch 14 - Training loss: 0.32883313158527017

Training Time (in minutes) = 0.6374959429105123

3.2.2 Testing MLP model

- Images Tested = 210
- Correct Tests = 184
- False Positive Tests = 20
- False Negative Tests = 6

Model Accuracy (Average) = 87.619 %

3.3 Ejecución #3

3.3.1 Training MLP model

- Epoch 0 - Training loss: 0.669500358402729
- Epoch 1 - Training loss: 0.6638954095542431
- Epoch 2 - Training loss: 0.6090108752250671
- Epoch 3 - Training loss: 0.5524321272969246
- Epoch 4 - Training loss: 0.46861656941473484

- Epoch 5 - Training loss: 0.4160280302166939
- Epoch 6 - Training loss: 0.43497278820723295
- Epoch 7 - Training loss: 0.3823196701705456
- Epoch 8 - Training loss: 0.33440618123859167
- Epoch 9 - Training loss: 0.29762012232095003
- Epoch 10 - Training loss: 0.4215814843773842
- Epoch 11 - Training loss: 0.35535851027816534
- Epoch 12 - Training loss: 0.38565926626324654
- Epoch 13 - Training loss: 0.3213418032974005
- Epoch 14 - Training loss: 0.28836515452712774

Training Time (in minutes) = 0.6408170580863952

3.3.2 Testing MLP model

- Images Tested = 210
- Correct Tests = 177
- False Positive Tests = 28
- False Negative Tests = 5

Model Accuracy (Average) = 84.2857 %

3.4 Ejecución #4

3.4.1 Training MLP model

- Epoch 0 - Training loss: 0.6817897818982601
- Epoch 1 - Training loss: 0.6486177258193493
- Epoch 2 - Training loss: 0.6065827794373035
- Epoch 3 - Training loss: 0.5542502943426371
- Epoch 4 - Training loss: 0.4871502108871937
- Epoch 5 - Training loss: 0.4139289911836386
- Epoch 6 - Training loss: 0.43559644743800163
- Epoch 7 - Training loss: 0.3611491918563843
- Epoch 8 - Training loss: 0.3386910930275917
- Epoch 9 - Training loss: 0.33063480723649263
- Epoch 10 - Training loss: 0.28561106510460377
- Epoch 11 - Training loss: 0.35864088498055935
- Epoch 12 - Training loss: 0.3085365677252412
- Epoch 13 - Training loss: 0.28754513058811426
- Epoch 14 - Training loss: 0.27902613021433353

Training Time (in minutes) = 0.6235944430033366

3.4.2 Testing MLP model

- Images Tested = 210
- Correct Tests = 188
- False Positive Tests = 6
- False Negative Tests = 16

Model Accuracy (Average) = 89.5238 %

3.5 Ejecución #5

3.5.1 Training MLP model

- Epoch 0 - Training loss: 0.6733688861131668
- Epoch 1 - Training loss: 0.6364881433546543
- Epoch 2 - Training loss: 0.5911890938878059
- Epoch 3 - Training loss: 0.5003748796880245
- Epoch 4 - Training loss: 0.4436192847788334
- Epoch 5 - Training loss: 0.4071769081056118
- Epoch 6 - Training loss: 0.36832852475345135
- Epoch 7 - Training loss: 0.31542853731662035
- Epoch 8 - Training loss: 0.31395438965409994
- Epoch 9 - Training loss: 0.3593072723597288
- Epoch 10 - Training loss: 0.3414111491292715
- Epoch 11 - Training loss: 0.366707980632782
- Epoch 12 - Training loss: 0.3180706249549985
- Epoch 13 - Training loss: 0.29311301838606596
- Epoch 14 - Training loss: 0.3368922136723995

Training Time (in minutes) = 0.6206583380699158

3.5.2 Testing MLP model

- Images Tested = 210
- Correct Tests = 156
- False Positive Tests = 53
- False Negative Tests = 1

Model Accuracy (Average) = 74.2857 %

3.6 Ejecución #6

3.6.1 Training MLP model

- Epoch 0 - Training loss: 0.6783614419400692
- Epoch 1 - Training loss: 0.6365665383636951
- Epoch 2 - Training loss: 0.5871519818902016
- Epoch 3 - Training loss: 0.5118771959096193
- Epoch 4 - Training loss: 0.44583754427731037
- Epoch 5 - Training loss: 0.3815526831895113
- Epoch 6 - Training loss: 0.3625989994034171
- Epoch 7 - Training loss: 0.42644291929900646
- Epoch 8 - Training loss: 0.3171516880393028
- Epoch 9 - Training loss: 0.2905137366615236
- Epoch 10 - Training loss: 0.3491507964208722
- Epoch 11 - Training loss: 0.4224322587251663
- Epoch 12 - Training loss: 0.31965601071715355
- Epoch 13 - Training loss: 0.28112538903951645

- Epoch 14 - Training loss: 0.2759061469696462

Training Time (in minutes) = 0.6020392537117004

3.6.2 Testing MLP model

- Images Tested = 210
- Correct Tests = 164
- False Positive Tests = 45
- False Negative Tests = 1

Model Accuracy (Average) = 78.0952 %

3.7 Ejecución #7

3.7.1 Training MLP model

- Epoch 0 - Training loss: 0.6865210831165314
- Epoch 1 - Training loss: 0.6606931053102016
- Epoch 2 - Training loss: 0.6039097309112549
- Epoch 3 - Training loss: 0.5431239902973175
- Epoch 4 - Training loss: 0.5155778210610151
- Epoch 5 - Training loss: 0.5216354019939899
- Epoch 6 - Training loss: 0.3976256269961595 5
- Epoch 7 - Training loss: 0.380113935098052
- Epoch 8 - Training loss: 0.33804269321262836
- Epoch 9 - Training loss: 0.3554385444149375
- Epoch 10 - Training loss: 0.3892012722790241
- Epoch 11 - Training loss: 0.35064855590462685
- Epoch 12 - Training loss: 0.3019036818295717
- Epoch 13 - Training loss: 0.318946554325521
- Epoch 14 - Training loss: 0.2749772281385958

Training Time (in minutes) = 0.6188109715779623

3.7.2 Testing MLP model

- Images Tested = 210
- Correct Tests = 163
- False Positive Tests = 45
- False Negative Tests = 2

Model Accuracy (Average) = 77.619 %

3.8 Ejecución #8

3.8.1 Training MLP model

- Epoch 0 - Training loss: 0.7105191797018051
- Epoch 1 - Training loss: 0.6804268285632133
- Epoch 2 - Training loss: 0.6639746502041817
- Epoch 3 - Training loss: 0.6529141366481781

- Epoch 4 - Training loss: 0.6171780154109001
- Epoch 5 - Training loss: 0.557680094614625
- Epoch 6 - Training loss: 0.5138429421931505
- Epoch 7 - Training loss: 0.43288153782486916
- Epoch 8 - Training loss: 0.37753714621067047
- Epoch 9 - Training loss: 0.3594362251460552
- Epoch 10 - Training loss: 0.40580929070711136
- Epoch 11 - Training loss: 0.3357595829293132
- Epoch 12 - Training loss: 0.4155731126666069
- Epoch 13 - Training loss: 0.3880708534270525
- Epoch 14 - Training loss: 0.3696711454540491

Training Time (in minutes) = 0.6401947577794392

3.8.2 Testing MLP model

- Images Tested = 210
- Correct Tests = 133
- False Positive Tests = 77
- False Negative Tests = 0

Model Accuracy (Average) = 63.3333 %

3.9 Ejecución #9

3.9.1 Training MLP model

- Epoch 0 - Training loss: 0.6786065921187401
- Epoch 1 - Training loss: 0.6465523429214954
- Epoch 2 - Training loss: 0.5971381552517414
- Epoch 3 - Training loss: 0.5270331036299467
- Epoch 4 - Training loss: 0.4790476132184267
- Epoch 5 - Training loss: 0.39014657586812973
- Epoch 6 - Training loss: 0.41562835685908794
- Epoch 7 - Training loss: 0.343689332716167
- Epoch 8 - Training loss: 0.3210268598049879
- Epoch 9 - Training loss: 0.3434343272820115
- Epoch 10 - Training loss: 0.2874893303960562
- Epoch 11 - Training loss: 0.3127787606790662
- Epoch 12 - Training loss: 0.2764781713485718
- Epoch 13 - Training loss: 0.288813638035208
- Epoch 14 - Training loss: 0.2772562876343727

Training Time (in minutes) = 0.6265958666801452

3.9.2 Testing MLP model

- Images Tested = 210
- Correct Tests = 175
- False Positive Tests = 0
- False Negative Tests = 35

Model Accuracy (Average) = 83.3333 %

3.10 Ejecución #10

3.10.1 Training MLP model

- Epoch 0 - Training loss: 0.6788728050887585
- Epoch 1 - Training loss: 0.644656702876091
- Epoch 2 - Training loss: 0.6026467494666576
- Epoch 3 - Training loss: 0.5293423216789961
- Epoch 4 - Training loss: 0.43763526156544685
- Epoch 5 - Training loss: 0.3918335437774658
- Epoch 6 - Training loss: 0.36272476986050606
- Epoch 7 - Training loss: 0.33194719441235065
- Epoch 8 - Training loss: 0.39527758676558733
- Epoch 9 - Training loss: 0.31878712866455317
- Epoch 10 - Training loss: 0.32153183966875076
- Epoch 11 - Training loss: 0.33105915877968073
- Epoch 12 - Training loss: 0.2790641039609909
- Epoch 13 - Training loss: 0.28298106137663126
- Epoch 14 - Training loss: 0.27733859745785594

Training Time (in minutes) = 0.6084364891052246

3.10.2 Testing MLP model

- Images Tested = 210
- Correct Tests = 188
- False Positive Tests = 3
- False Negative Tests = 19

Model Accuracy (Average) = 89.5238 %