Análisis con BackPropagation

1. Primeras pruebas

En primer lugar, se elabora el script con el que se realizarán las pruebas, en él se leen los datos de la base de datos ya preprocesada y se elaboran los datasets de entrenamiento, validación y test, para posteriormente construir la red neuronal, entrenarla, validarla y calcular los datos necesarios para mostrar porcentajes de acierto.

```
#Imports
from __future__ import division
from pybrain.datasets import SupervisedDataSet
from pybrain.supervised.trainers import BackpropTrainer
from pybrain.tools.shortcuts import buildNetwork
from pybrain.utilities import percentError
import numpy as np
import pylab as pl
import math as ma

#Leer las bases de datos
patternTrain = np.loadtxt("BreastCancerPreprocessedTrain.csv", dtype=float, delimiter=',')
patternValid = np.loadtxt("BreastCancerPreprocessedValid.csv", dtype=float, delimiter=',')
patternTest = np.loadtxt("BreastCancerPreprocessedTest.csv", dtype=float, delimiter=',')

#Conseguir el numero de filas y columnas
numPatTrain, numColsTrain = patternTrain.shape
numPatValid, numColsValid = patternValid.shape
numPatTest, numColsTest = patternTest.shape

#Generar el input
patternTrainInput = patternTrain[:, 1:numColsTrain]
patternValidInput = patternValid[:, 1:numColsValid]
patternTestInput = patternTest[:, 1:numColsTest]

#Generar salidas deseadas
patternTrainTarget = np.zeros([numPatTrain, 2])
patternValidTarget = np.zeros([numPatValid, 2])
patternTestTarget = np.zeros([numPatTest, 2])


#Crear los dataset supervisados
trainDS = SupervisedDataSet(numColsTrain-1, 2)
for i in range(numPatTrain):
        patternTrainTarget[i, patternTrain[i, 0]] = 1.0
        trainDS.addSample(patternTrainInput[i], patternTrainTarget[i])

validDS = SupervisedDataSet(numColsValid-1, 2)
for i in range(numPatValid):
        patternValidTarget[i, patternValid[i, 0]] = 1.0
        validDS.addSample(patternValidInput[i], patternValidTarget[i])


testDS = SupervisedDataSet(numColsTest-1, 2)
for i in range(numPatTest):
        patternTestTarget[i, patternTest[i, 0]] = 1.0
        testDS.addSample(patternTestInput[i], patternTestTarget[i])

#Crear red con una capa oculta
numHiddenNodes = 25
myLearningRate = 0.0001
myMomentum = 0.1
```

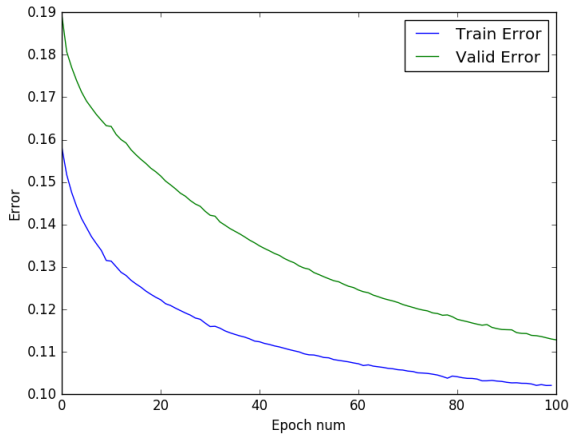
```
#Crear el trainer y hacer enternar el DS
trainer = BackpropTrainer(net, trainDS, learningrate=myLearningRate, momentum=myMomentum)
trainError = trainer.trainUntilConvergence(verbose=True, trainingData=trainDS, validationData=validDS)

#Crear la gráfica con los errores de validación y entrenamiento
pl.plot(trainError[0], label='Train Error')
pl.plot(trainError[1], label='Valid Error')
pl.xlabel('Epoch num')
pl.ylabel('Error')
pl.legend(loc='upper right')
pl.show()

#Obtener porcentajes
results = net.activateOnDataset(validDS)

patResult = -1
positivo = 0
negativo = 0
falsoPositivo = 0
falsoNegativo = 0

for i in range(numPatValid):
	if max(results[i]) == results[i, 0]:
		patResult = 0
	else:
		patResult = 1

	if (patternValid[i, 0] == 1 and patternValid[i, 0] == patResult):
		positivo = positivo + 1
	elif (patternValid[i, 0] == 0 and patternValid[i, 0] == patResult):
		negativo = negativo + 1
	elif (patternValid[i, 0] == 1 and patternValid[i, 0] != patResult):
		falsoNegativo = falsoNegativo + 1
	elif (patternValid[i, 0] == 0 and patternValid[i, 0] != patResult):
		falsoPositivo = falsoPositivo + 1

print("Positivo: %d" % positivo)
print("Negativo: %d" % negativo)
print("Falso Positivo: %d" % falsoPositivo)
print("Falso Negativo: %d" % falsoNegativo)
print("\n")

positivoTotal = positivo + falsoNegativo
negativoTotal = negativo + falsoPositivo

percentPositivo = positivo / positivoTotal * 100
percentNegativo = negativo / negativoTotal * 100
percentFalsoPositivo = falsoPositivo / negativoTotal * 100
percentFalsoNegativo = falsoNegativo / positivoTotal * 100
accuracy = ((positivo + negativo) / numPatValid) * 100
recall = (positivo / positivoTotal) * 100
precision = (positivo / (positivo + falsoPositivo)) * 100

print("Porcentaje de aciertos positivos: %3.2f%%" % percentPositivo)
print("Porcentaje de falsos negativos: %3.2f%%" % percentFalsoNegativo)
print("Porcentaje de aciertos negativos: %3.2f%%" % percentNegativo)
print("Porcentaje de falsos positivos: %3.2f%%" % percentFalsoPositivo)
print("\n")

print("Accuracy: %3.2f%%" % accuracy)
print("Recall: %3.2f%%" % recall)
print("Precision: %3.2f%%" % precision)
```

Para las pruebas iniciales se usarán 25 neuronas ocultas, 0,0001 de Learning Rate y 0,1 de Momentum, llegando a un máximo de 100 iteraciones de entrenamiento. Estas pruebas servirán como trampolín a análisis posteriores, teniendo una referencia de partida que intentar mejorar con dichos análisis. Los porcentajes obtenidos serán calculados con el dataset de validación.

1.1. Cáncer de mama

El dataset de validación para el cáncer de mama contiene un 53,41% de casos positivos y un 46,59% de casos negativos.

Los resultados fueron los siguientes:

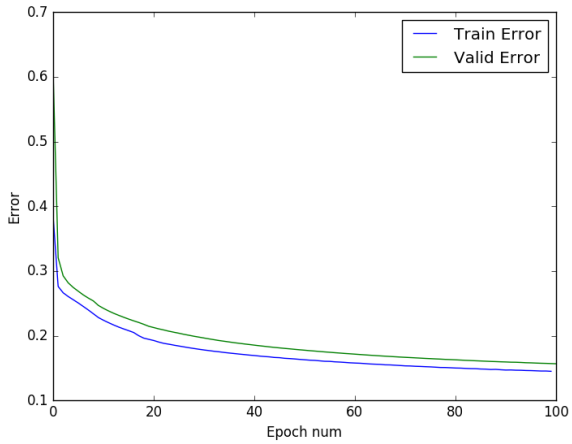



```
Positivo: 74
Negativo: 56
Falso Positivo: 26
Falso Negativo: 20


Porcentaje de aciertos positivos: 78.72%
Porcentaje de falsos negativos: 21.28%
Porcentaje de aciertos negativos: 68.29%
Porcentaje de falsos positivos: 31.71%


Accuracy: 73.86%
Recall: 78.72%
Precision: 74.00%
```

1.2. Melanoma

La base de datos de validación para este tipo de cáncer contiene un 42,92% de pacientes con cáncer y un 57,08% de pacientes sin cáncer.

Se han obtenido los siguientes resultados:

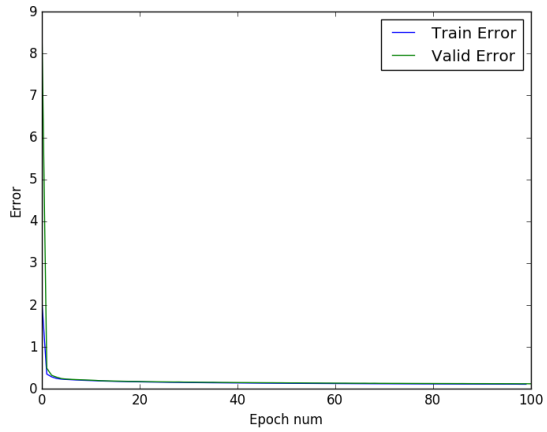



```
Positivo: 86
Negativo: 41
Falso Positivo: 88
Falso Negativo: 11


Porcentaje de aciertos positivos: 88.66%
Porcentaje de falsos negativos: 11.34%
Porcentaje de aciertos negativos: 31.78%
Porcentaje de falsos positivos: 68.22%


Accuracy: 56.19%
Recall: 88.66%
Precision: 49.43%
```

1.3. Cáncer de pulmón

La base de datos para el cáncer de pulmón es la más extensa de las tres por lo que se esperan los mejores resultados de todos los casos. El conjunto de validación con el que se llevarán a cabo los análisis tiene un 40.78% de pacientes con cáncer y un 59.22% de pacientes sin cáncer.

Los resultados fueron los siguientes:




```
Positivo: 209
Negativo: 197
Falso Positivo: 153
Falso Negativo: 32


Porcentaje de aciertos positivos: 86.72%
Porcentaje de falsos negativos: 13.28%
Porcentaje de aciertos negativos: 56.29%
Porcentaje de falsos positivos: 43.71%


Accuracy: 68.70%
Recall: 86.72%
Precision: 57.73%
```

En general los resultados son satisfactorios para ser pruebas iniciales, si bien es cierto que en el caso del cáncer de pulmón aunque tenga porcentajes de accuracy menores a los del cáncer de mama se espera que durante el análisis la mejora sea mayor que la experimentada por este y obtenga mejores resultados, debido a la gran cantidad de datos disponibles.

Ahora el objetivo será mejorar estos porcentajes y mejorar la gráfica de aprendizaje intentando conseguir menores errores en menor tiempo modificando para ello las variables necesarias del algoritmo de aprendizaje y la red neuronal.

2. Análisis de neuronas ocultas

A continuación, se procederá a hacer pruebas con varios números distintos de neuronas ocultas, para observar en cuál de ellos la red neuronal tiene un mejor resultado. Las pruebas se realizarán modificando el número de neuronas ocultas de 5 a 100 con pasos de 5 neuronas (5, 10, 15,…,100), pero dejando el resto de variables con el mismo valor que en la prueba inicial, es decir, un Learning Rate de 0.0001 y un Momentum de 0.1, ya que estas variables influyen en el aprendizaje de la red y no en la red en sí misma.

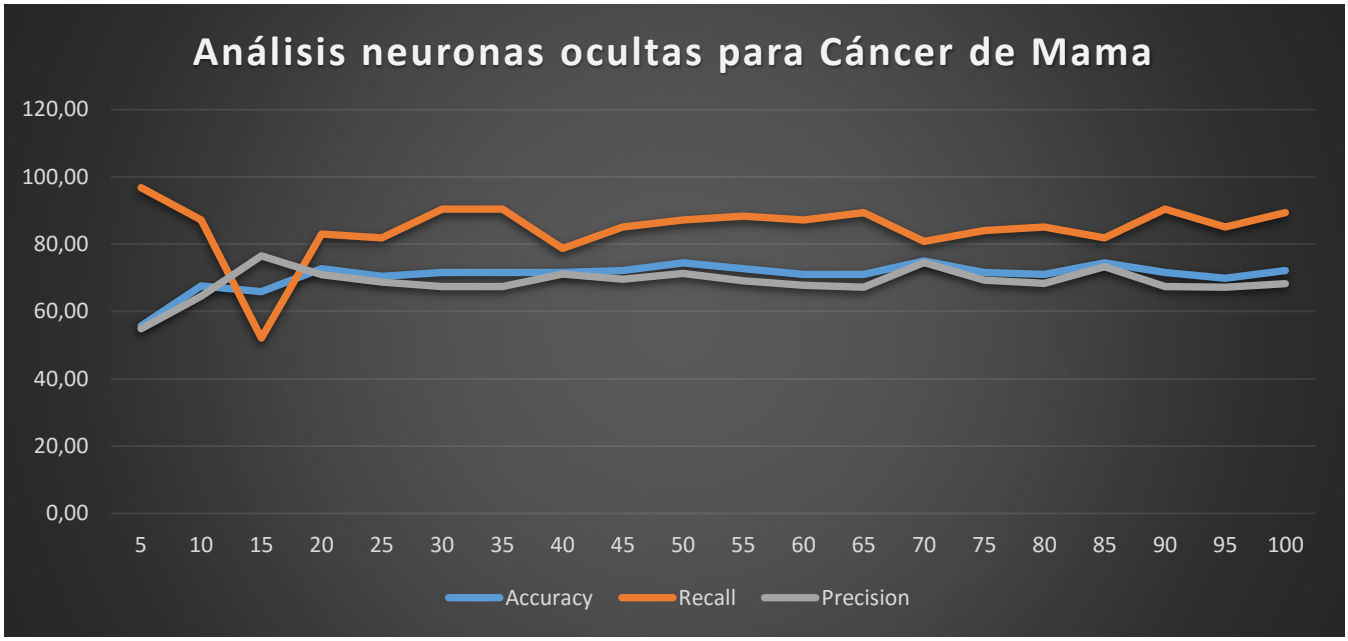
Para cada número de neuronas se harán 5 pruebas obteniendo el mejor de los resultados como representante de esa instancia.

Sobre todo se deben observar las columnas para Accuracy y Recall. El Accuracy es el porcentaje de aciertos totales, incluidos positivos y negativos, mientras que el Recall es el porcentaje de aciertos positivos dentro de todos los valores positivos que contenía la base de datos.

Interesa conseguir un Accuracy lo más alto posible siempre y cuando el Recall no sea demasiado bajo, ya que es más grave diagnosticar negativo a un paciente con cáncer que diagnosticar positivo a un paciente sin cáncer.

2.1. Cáncer de mama

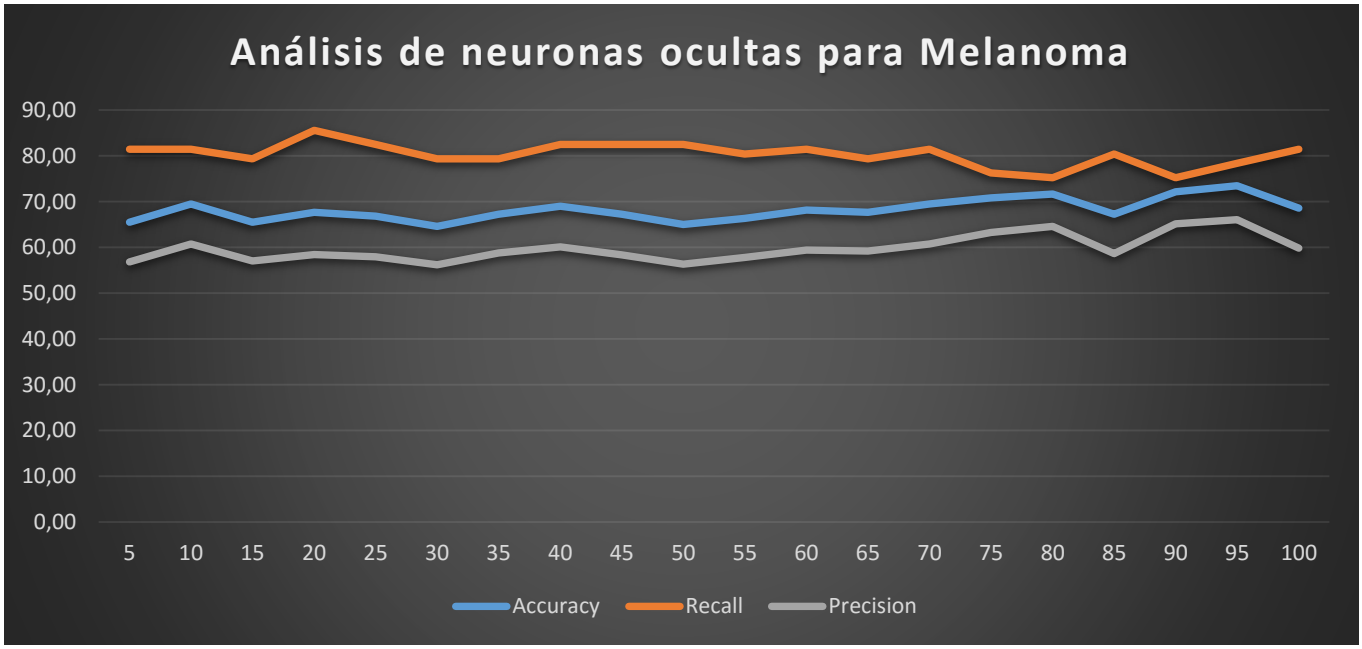
Neuronas ocultas	Accuracy	Recall	Precision
5	55,68	96,81	54,82
10	67,61	87,23	64,57
15	65,91	52,13	76,56
20	72,73	82,98	70,91
25	70,45	81,91	68,75
30	71,59	90,43	67,46
35	71,59	90,43	67,46
40	71,59	78,72	71,15
45	72,16	85,11	69,57
50	74,43	87,23	71,30
55	72,73	88,30	69,17
60	71,02	87,23	67,77
65	71,02	89,36	67,20
70	75,00	80,85	74,51
75	71,59	84,04	69,30
80	71,02	85,11	68,38
85	74,43	81,91	73,33
90	71,59	90,43	67,46
95	69,89	85,11	67,23
100	72,16	89,36	68,29



Se ha decidido **usar 50 neuronas ocultas**. A pesar de no tener el accuracy más alto, el recall para esta instancia mejora en mucho los valores para instancias cercanas en accuracy.

2.2. Melanoma

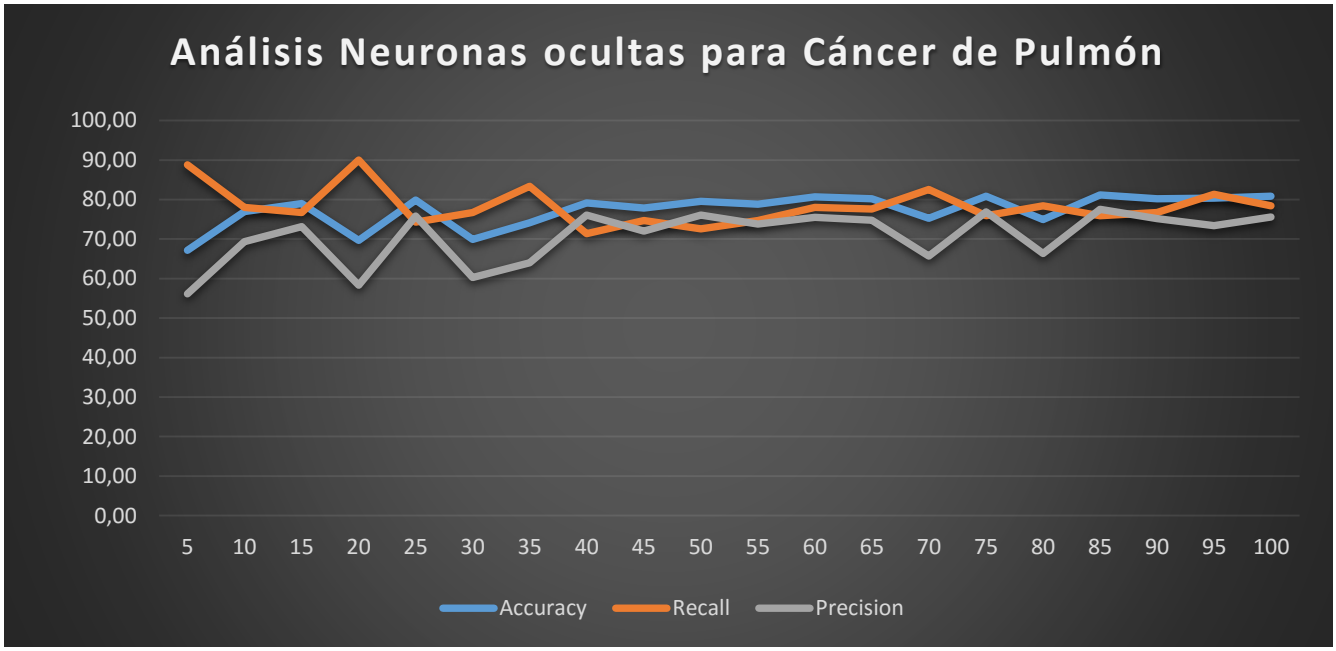
Neuronas ocultas	Accuracy	Recall	Precision
5	65,49	81,44	56,83
10	69,47	81,44	60,77
15	65,49	79,38	57,04
20	67,70	85,57	58,45
25	66,81	82,47	57,97
30	64,60	79,38	56,20
35	67,26	79,38	58,78
40	69,03	82,47	60,15
45	67,26	82,47	58,39
50	65,04	82,47	56,34
55	66,37	80,41	57,78
60	68,14	81,44	59,40
65	67,70	79,38	59,23
70	69,47	81,44	60,77
75	70,80	76,29	63,25
80	71,68	75,26	64,60
85	67,26	80,41	58,65
90	72,12	75,26	65,18
95	73,45	78,35	66,09
100	68,58	81,44	59,85



Los resultados en el caso del melanoma son peores que el cáncer de mama, siendo peores en general el accuracy, el recall y el precision. Esto puede deberse a que los datos de entrenamiento son menos significativos. Se elegirá **usar 95 neuronas ocultas**, esta instancia obtuvo el mejor accuracy (73,45%) y un recall aceptable en relación (78,35%), ya que para accuracy menores, el recall no es mucho mayor.

2.3. Cáncer de pulmón

Neuronas ocultas	Accuracy	Recall	Precision
5	67,17	88,80	56,17
10	76,99	78,01	69,37
15	79,02	76,76	73,12
20	69,71	90,04	58,33
25	79,86	74,27	75,85
30	69,88	76,76	60,26
35	74,11	83,40	64,01
40	79,19	71,37	76,11
45	77,83	74,69	72,00
50	79,53	72,61	76,09
55	78,85	74,69	73,77
60	80,71	78,01	75,50
65	80,20	77,59	74,80
70	75,30	82,57	65,68
75	80,88	75,93	76,89
80	74,96	78,42	66,32
85	81,22	75,93	77,54
90	80,20	76,76	75,20
95	80,37	81,33	73,41
100	80,88	78,42	75,60



Obviamente por tener más casos con los que entrenar la red neuronal se demuestra que los resultados son mejores, traspasando la barrera del 75 % de accuracy en la mayoría de los casos. Se ha elegido **usar 100 neuronas ocultas**, ya que a pesar de no tener el mejor accuracy, la relación de recall es mucho mejor, ya que solo perdiendo 0,4 puntos de accuracy comparado con el mejor resultado gana 3 puntos en recall, lo cual es bastante aceptable.

3. Análisis de Learning Rate

A continuación se realizará el análisis de la variable Learning Rate. Hasta ahora ha tenido un valor de 0.0001, pero se procederá a realizar pruebas para obtener el mejor resultado posible. Las pruebas consistirán en hallar con qué valor de Learning Rate se alcanza un error determinado en menos tiempo.

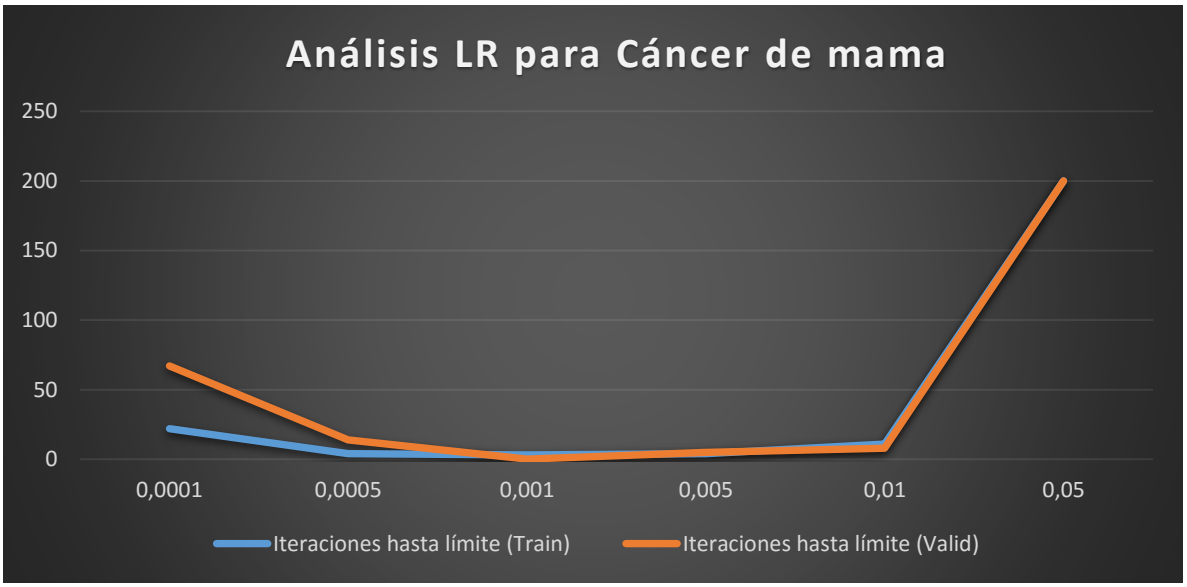
En nuestro caso, observando valores de error en las pruebas previas se ha decidido que dicha cota sea '0.13', por lo cual el script calculará la iteración en la que el error baja de dicho valor. Existen casos en los que el error no llega a bajar de ese valor durante las iteraciones que tiene determinadas el entrenamiento, por lo cual se harán 5 pruebas por cada valor de Learning Rate usando el mejor de ellos como representante de esa instancia intentando ignorar los casos en los que no se llegue a ese límite en el caso del error de entrenamiento como en el error de validación.

Se ha decidido limitar el entrenamiento a 200 iteraciones, ya que aproximadamente a esa altura el valor de error será continuo. Se analizará el comportamiento de la red neuronal con los siguientes valores de Learning Rate:

0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05 (Se pretendía realizar pruebas con valores mayores, pero la red neuronal provocaba un overflow).

3.1. Cáncer de Mama

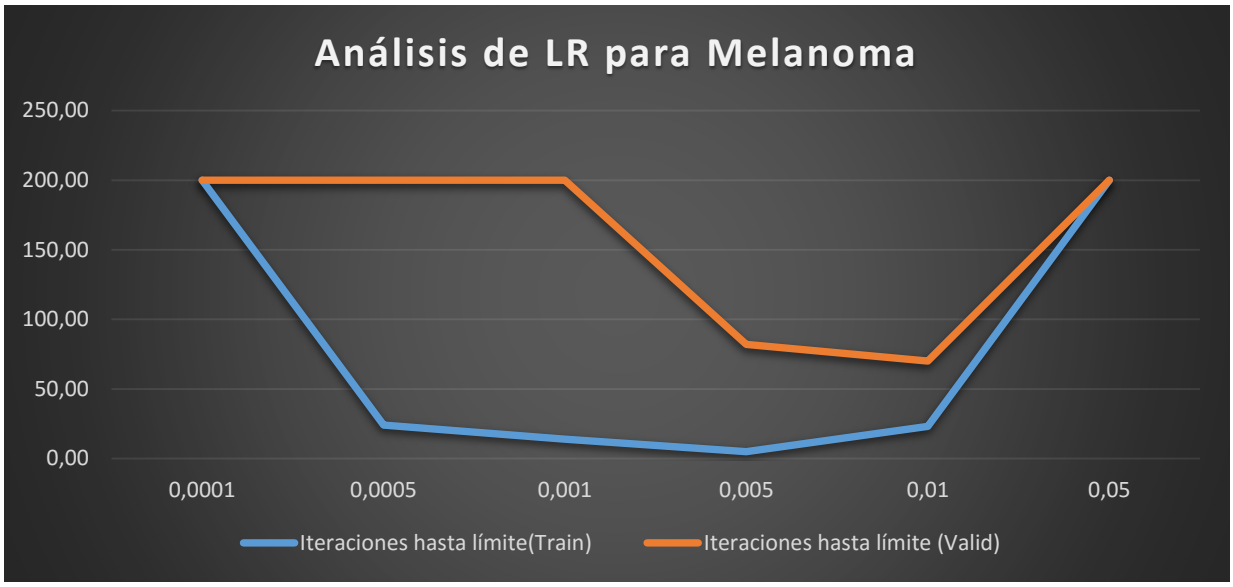
Learning Rate	Iteraciones hasta límite (Train)	Iteraciones hasta límite (Valid)
0,0001	22	67
0,0005	4	14
0,001	3	8
0,005	4	5
0,01	11	8
0,05	200	200



Se utilizará 0.005 de Learning Rate. Con este valor se obtuvo el menor número de iteraciones para el error de validación (5) y el segundo mejor para el error de entrenamiento (4).

3.2. Melanoma

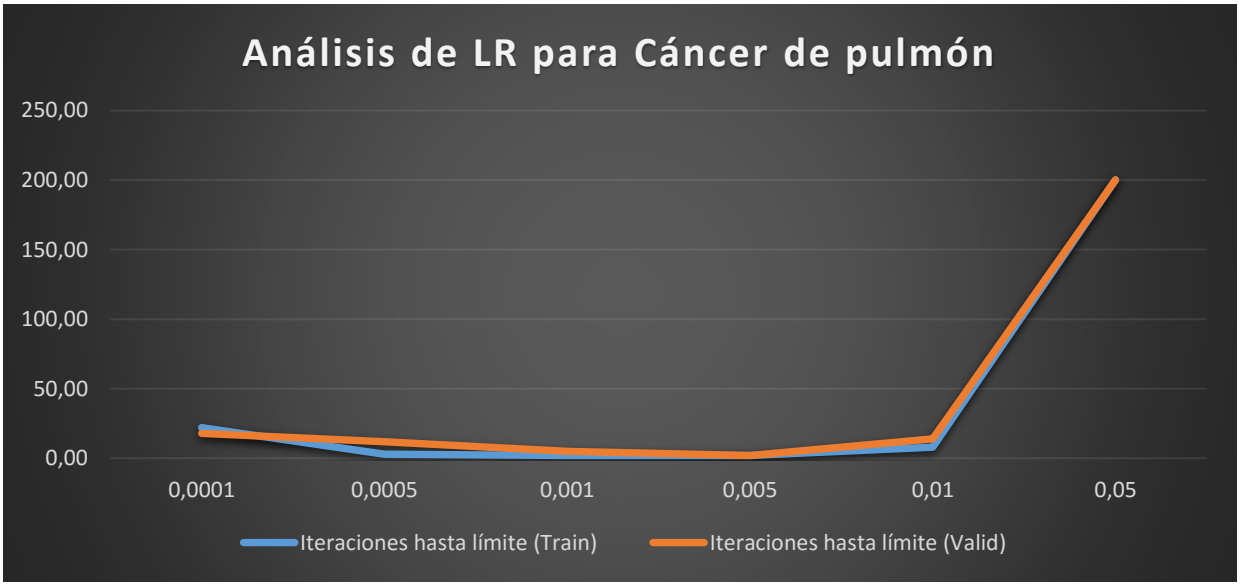
Learning rate	Iteraciones hasta límite(Train)	Iteraciones hasta límite (Valid)
0,0001	200,00	200,00
0,0005	24,00	200,00
0,001	14,00	200,00
0,005	5,00	82,00
0,01	23,00	70,00
0,05	200,00	200,00



En este caso se ven más dificultades para alcanzar la cota de error de 0,13 que con el cáncer de mama. Se ha decidido seleccionar **0,005 de Learning Rate**, al igual que en caso anterior, ya que, a pesar de no tener el mínimo de iteraciones para el error de validación, sí que tiene el mínimo de iteraciones para el error de entrenamiento y la relación es mejor.

3.3. Cáncer de pulmón

Learning Rate	Iteraciones hasta límite (Train)	Iteraciones hasta límite (Valid)
0,0001	22,00	18,00
0,0005	3,00	12,00
0,001	2,00	5,00
0,005	2,00	2,00
0,01	8,00	14,00
0,05	200,00	200,00



De nuevo para el este cáncer se elegirá usar 0.005 de Learning Rate, ebteniendo los mejores resultados en general, tal y como se esperaba con 2 iteraciones para alcanzar la cota, tanto en error de entrenamiento como en error de validación.

En conclusión, era de esperar que los tres tipos de cáncer tuvieran el mismo o parecido Learning Rate, ya que es un valor que suele depender del número de entradas y salidas de la red neuronal, por lo que al ser las mismas para los tres tipos, debían coincidir.