Análisis con BackPropagation

1. Breast Cancer

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 y validarla.

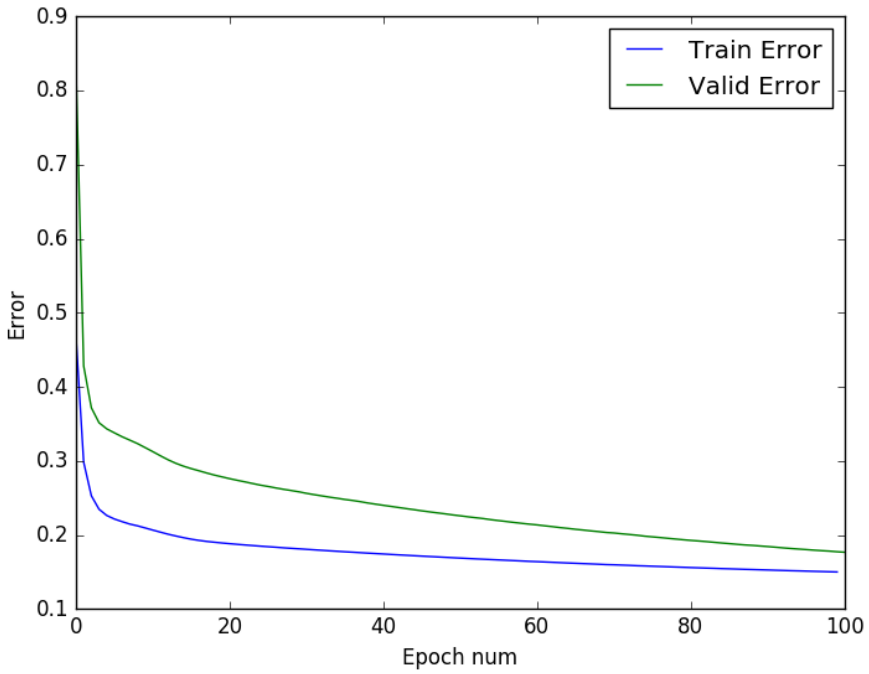
```
#Imports
from __future__ import division
from pybrain.datasets import SupervisedDataSet
from pybrain.supervised.trainers import BackpropTrainer
from pybrain.tools.shortcuts import buildNetwork
from pybrain.utilities import percentError
import numpy as np
import pylab as pl
import math as ma

#Read training sets
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 = 100
myLearningRate = 0.0001
myMomentum = 0.1
net = buildNetwork(numColsTrain-1, numHiddenNodes, 2, bias=True)

#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,
maxEpochs=100)

#Plot training and validation errors
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()

#Obtención de los 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

positivoTotal = positivo + falsoNegativo
negativoTotal = negativo + falsoPositivo

percentPositivo = positivo / positivoTotal * 100
percentNegativo = negativo / negativoTotal * 100
percentFalsoPositivo = falsoPositivo / negativoTotal * 100
percentFalsoNegativo = falsoNegativo / positivoTotal * 100
percentTotal = ((positivo + negativo) / numPatValid) * 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("Porcentaje total de acierto: %3.2f%%" % percentTotal)
```

1.1. Primera prueba

Para la primera prueba se ha usado una red neuronal con 25 neuronas ocultas, 0.0001 de Learning Rate y 0.1 de Momentum. Estos porcentajes se obtienen analizando el Dataset de validación, el cual tiene un porcentaje de pacientes con cáncer de un 53.41% y un porcentaje de pacientes sin cáncer de 46.59%.

Los resultados fueron los siguientes:




```
Porcentaje de aciertos positivos: 44.68%
Porcentaje de falsos negativos: 55.32%
Porcentaje de aciertos negativos: 89.02%
Porcentaje de falsos positivos: 10.98%
Porcentaje total de acierto: 65.34%
```

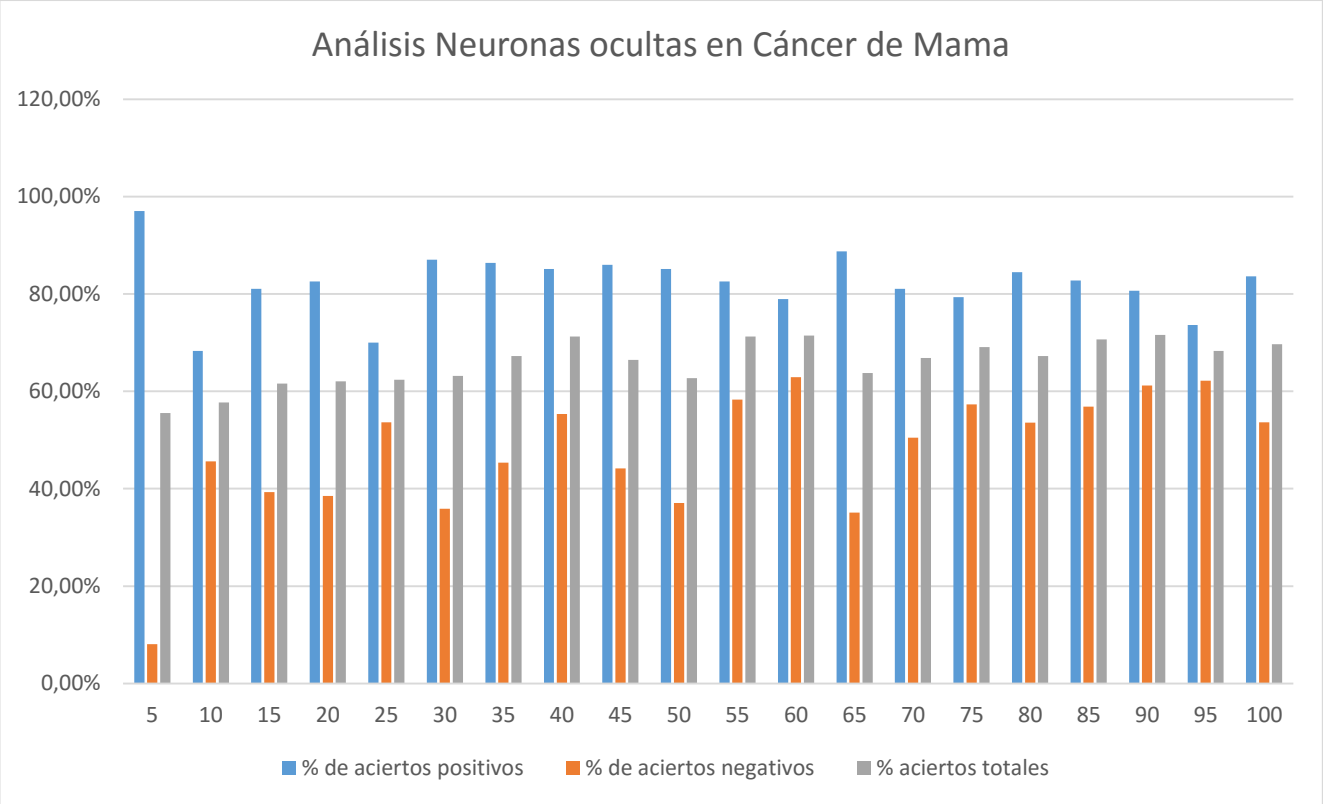
Ahora el objetivo será mejorar estos porcentajes y mejorar la gráfica de aprendizaje intentando conseguir menores errores en menor tiempo modificando para ello las variables necesarias del algoritmo de aprendizaje y la red neuronal.

1.2. Modificando el número de neuronas

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 para estos datos en concreto. 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 la media de acierto de todas ellas como representante de esa instancia.

Nº Neuronas Ocultas	% de aciertos positivos	% de aciertos negativos	% aciertos totales
5	96,99%	8,05%	55,57%
10	68,30%	45,58%	57,73%
15	81,06%	39,27%	61,59%
20	82,55%	38,54%	62,04%
25	70,00%	53,66%	62,39%
30	87,02%	35,85%	63,18%
35	86,38%	45,36%	67,27%
40	85,11%	55,37%	71,25%
45	85,96%	44,15%	66,48%
50	85,11%	37,07%	62,73%
55	82,54%	58,29%	71,25%
60	78,94%	62,93%	71,48%
65	88,72%	35,12%	63,75%
70	81,06%	50,49%	66,82%
75	79,36%	57,32%	69,09%
80	84,47%	53,56%	67,27%
85	82,76%	56,83%	70,68%
90	80,64%	61,22%	71,59%
95	73,62%	62,20%	68,29%
100	83,62%	53,66%	69,66%



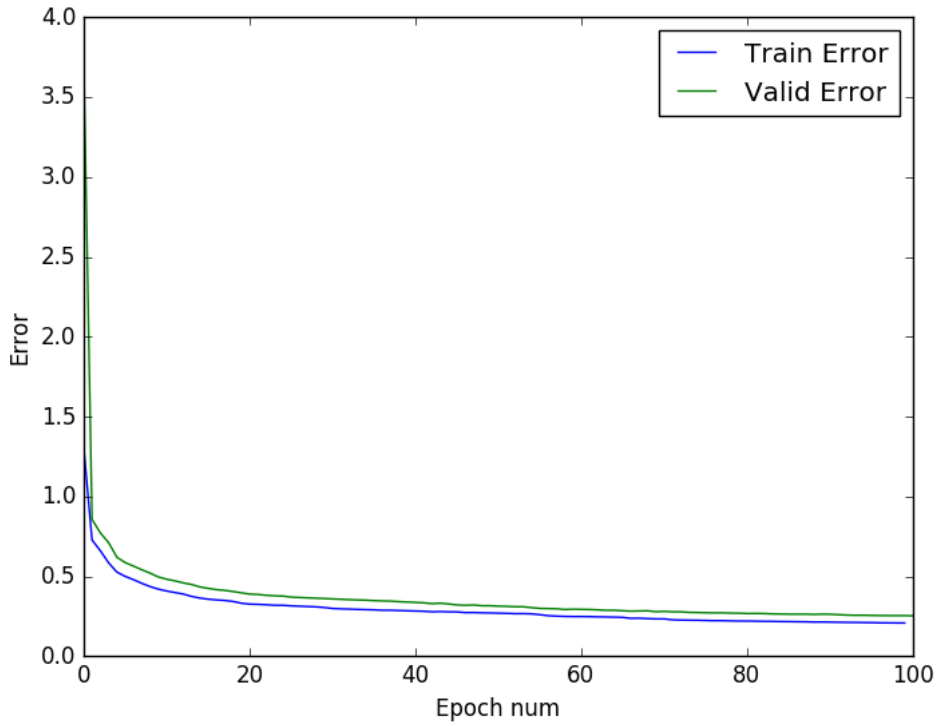
Los estudios con 40, 60 o 90 neuronas ocultas arrojaron resultados muy similares, pero varios factores inclinaron la balanza por **usar 90 neuronas ocultas**. Con este número se obtuvo el mejor resultado de media (71,59%) y el mejor resultado individual (73,86%) y el tiempo de procesamiento no es exageradamente superior como para considerar el uso de 40 o 60 neuronas.

2. Melanoma

En este caso no se mostrará el script para la red neuronal, ya que es prácticamente idéntico al que corresponde al cáncer de mama, excepto que se usan las bases de datos relacionadas con el melanoma.

2.1. Primera prueba

Para esta primera prueba del melanoma se han usado los mismos datos que para la primera prueba del cáncer de mama (25 neuronas ocultas, 0.0001 de Learning Rate y 0.1 de Momentum) y se han obtenido los siguientes resultados:



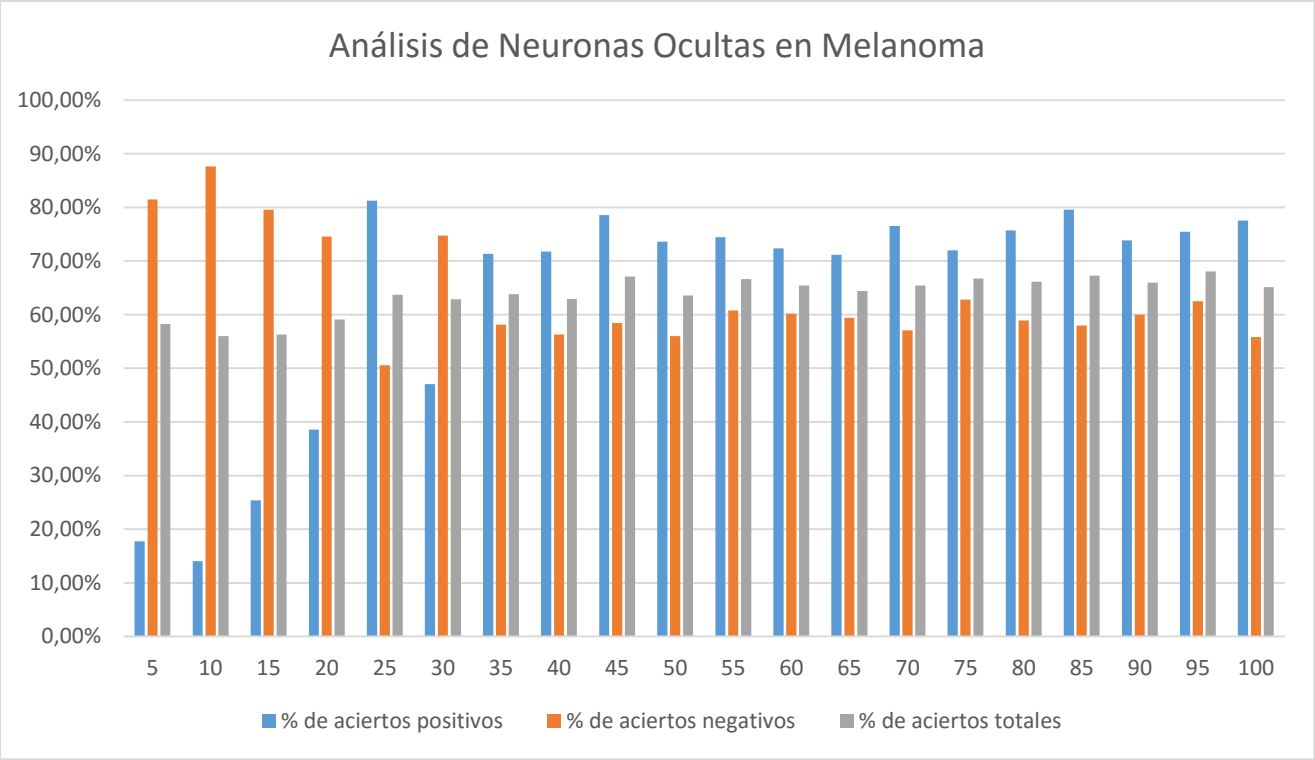

```
Porcentaje de aciertos positivos: 36.08%
Porcentaje de falsos negativos: 63.92%
Porcentaje de aciertos negativos: 75.19%
Porcentaje de falsos positivos: 24.81%
Porcentaje total de acierto: 58.41%
```

En este caso, comparándolo con la primera prueba del cáncer de mama, se obtienen peores porcentajes, tanto para los aciertos positivos y negativos, como para los aciertos totales, hay que ver si esta tendencia permanece durante los análisis para obtener el número óptimo de neuronas ocultas o mejora el nivel de lo visto en el análisis del cáncer de mama. La base de datos de validación usada para el melanoma tiene un 42.92% de pacientes con cáncer y un 57.08 de pacientes sin cáncer.

2.2. Modificando el número de neuronas

Repetimos, idénticamente al análisis del cáncer de mama, se estudiarán los resultados para un número de neuronas ocultas entre 5 y 100 avanzando de cinco en cinco.

Nº Neuronas ocultas	% de aciertos positivos	% de aciertos negativos	% de aciertos totales
5	17,73%	81,49%	58,23%
10	14,02%	87,60%	56,02%
15	25,36%	79,54%	56,28%
20	38,55%	74,57%	59,11%
25	81,24%	50,54%	63,72%
30	47,01%	74,73%	62,83%
35	71,34%	58,14%	63,81%
40	71,75%	56,28%	62,92%
45	78,56%	58,45%	67,08%
50	73,61%	55,97%	63,54%
55	74,43%	60,78%	66,64%
60	72,37%	60,16%	65,40%
65	71,13%	59,38%	64,42%
70	76,49%	57,06%	65,40%
75	71,96%	62,79%	66,73%
80	75,67%	58,91%	66,11%
85	79,59%	57,98%	67,25%
90	73,81%	60,00%	65,93%
95	75,46%	62,48%	68,05%
100	77,53%	55,82%	65,13%



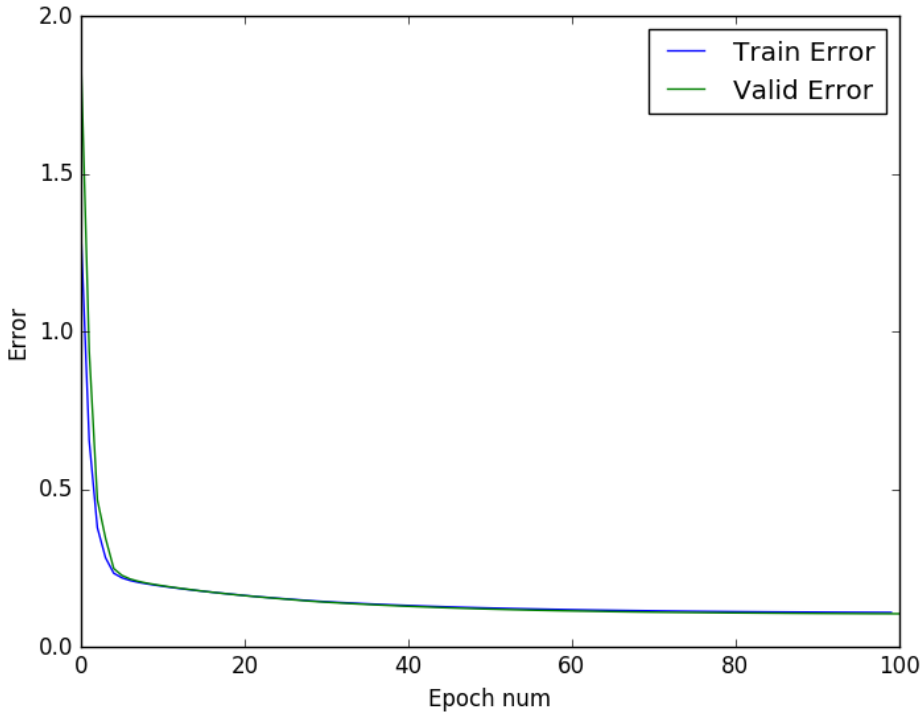
Los resultados en el melanoma son más bajos que en el caso del cáncer de mama, no llegando a superar los resultados totales el 70% de acierto, se puede suponer que se debe a que los datos de entrenamiento son menos significativos. **Se elegirá usar 95 neuronas ocultas**, ya que con esta configuración se obtuvieron los mejores resultados de media (68,05), así como el segundo mejor resultado individual (71,28). Aun así, siempre se busca, si es posible, que el porcentaje de aciertos de pacientes con cáncer sea el más alto posible, ya que es más grave decirle a un paciente con cáncer que no lo tiene a decirle a un paciente sin cáncer que lo tiene.

3. Cáncer de pulmón

Al igual que en el caso anterior, el script es prácticamente idéntico al usado en el primer análisis, pero cambiando las bases de datos.

3.1. Primera prueba

De nuevo se realizará la primera prueba con 25 neuronas ocultas, 0.0001 de learning rate y 0.1 de Momentum. 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.



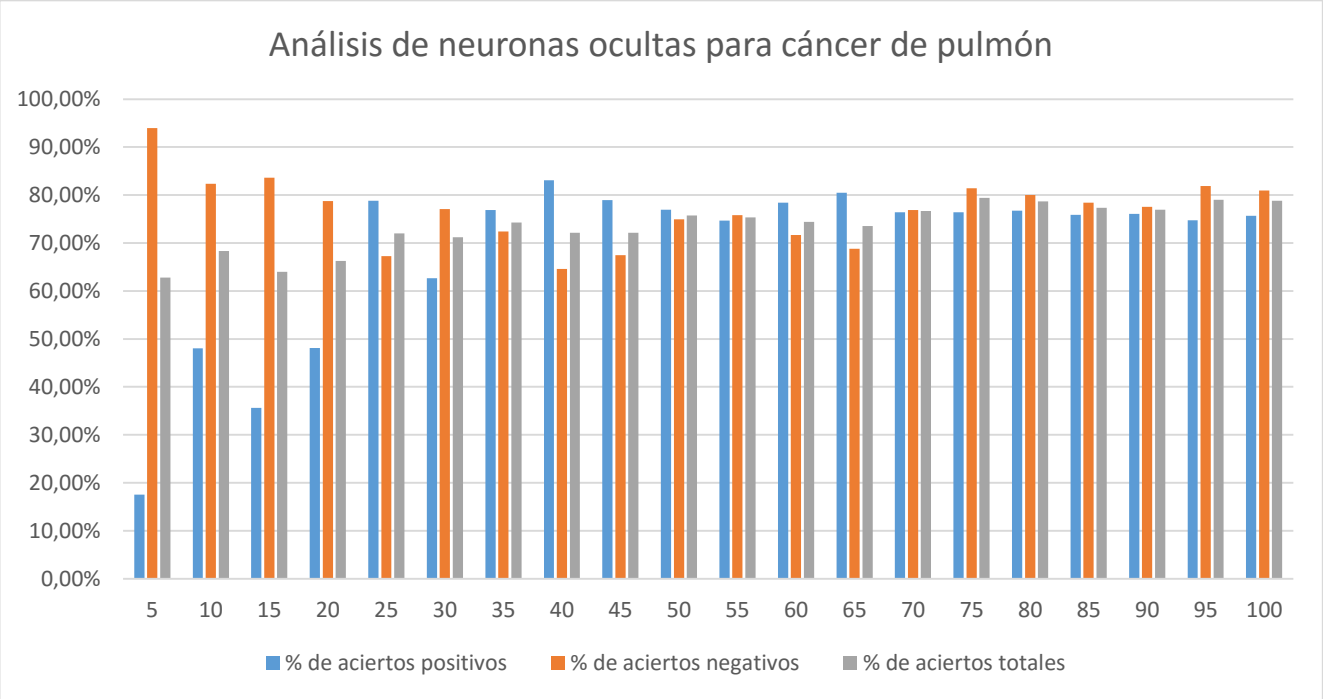

```
Porcentaje de aciertos positivos: 87.14%
Porcentaje de falsos negativos: 12.86%
Porcentaje de aciertos negativos: 54.57%
Porcentaje de falsos positivos: 45.43%
Porcentaje total de acierto: 67.85%
```

Cabe destacar el alto porcentaje de aciertos positivos y el mayor porcentaje de aciertos totales de las tres pruebas iniciales, por lo que la teoría de que, al tener un mayor número de instancias, los resultados serían mejores se va cumpliendo.

3.2. Modificando el número de neuronas

De nuevo, se estudiarán los resultados para un número de neuronas ocultas entre 5 y 100 avanzando de cinco en cinco.

Nº Neuronas ocultas	% de aciertos positivos	% de aciertos negativos	% de aciertos totales
5	17,51%	94,00%	62,81%
10	48,05%	82,34%	68,36%
15	35,60%	83,60%	64,03%
20	48,13%	78,74%	66,26%
25	78,84%	67,26%	71,98%
30	62,65%	77,08%	71,20%
35	76,89%	72,40%	74,25%
40	83,07%	64,57%	72,12%
45	78,92%	67,48%	72,15%
50	76,93%	74,97%	75,77%
55	74,69%	75,83%	75,36%
60	78,42%	71,66%	74,42%
65	80,50%	68,80%	73,57%
70	76,43%	76,86%	76,68%
75	76,43%	81,43%	79,39%
80	76,76%	80,00%	78,68%
85	75,85%	78,40%	77,36%
90	76,10%	77,54%	76,96%
95	74,77%	81,89%	78,98%
100	75,69%	80,97%	78,82%



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 aciertos en la mayoría de los casos. Se ha elegido **usar 75 neuronas ocultas**, ya que obtiene el mejor resultado de media (79.39%) y uno de los mejores individualmente (80.71%). A parte es de las instancias con mejor relación 'acierto positivo – acierto negativo', teniendo un alto porcentaje de positivos sin dejar caer demasiado el porcentaje de negativos.