



UNIVERSIDAD DE BURGOS

COMPUTACIÓN NEURONAL Y EVOLUTIVA

---

## P1: Thyroid

---

Diseñar y entrenar una red neuronal para clasificar pacientes en tres grupos clínicos dependiendo de la patología en la glándula tiroides (pacientes sanos, con hipertiroidismo o con hipotiroidismo).

Estudiantes:

DAVID MIGUEL LOZANO  
JAVIER MARTÍNEZ RIBERAS

Profesor de la asignatura:

ÁLVARO HERRERO COSÍO

1º semestre 2016

# Índice

A. Introduction	2
B. Motivación	2
C. Descripción del conjunto de datos	2
D. Descripción del procedimiento	3
E. Estudio	4
F. Mejor resultado obtenido	7
G. Explotación de la red	7

## A. Introduction

El objetivo de la práctica es diseñar y entrenar una red neuronal para la clasificación de pacientes en tres grupos dependiendo de la patología en la glándula tiroides. Se trata de un problema de reconocimiento de patrones en el que dados unos datos de entrada la red neuronal será capaz de asignarlos a una de las tres clases definidas (pacientes sanos, con hipertiroidismo o con hipotiroidismo). Haremos uso del dataset de ejemplo *Thyroid* que provee Matlab.

Realizaremos un estudio sobre el ajuste en el número de neuronas, ejecutando varios entrenamientos con la red y modificando el valor de este parámetro. Y discutiremos qué configuración nos proporciona los mejores resultados. Por último, explotaremos la red y analizaremos las salidas devueltas.

## B. Motivación

La motivación que nos llevó a elegir el tema del trabajo fue el que ambos poseíamos algún familiar que sufría alguna alteración en la glándula tiroides. Además, el gran desarrollo que están teniendo las redes neuronales aplicadas a la medicina en los últimos años nos desató interés para adentrarnos en este tema.

## C. Descripción del conjunto de datos

El conjunto de datos utilizados los hemos obtenido del dataset de ejemplo *Thyroid* que provee Matlab. Los datos provienen del *UCI Machine Learning Repository* [1] y fueron donados por la Universidad de California.

El dataset contiene datos de 7200 pacientes agrupados en dos matrices:

- *thyroidInputs*: matriz de 21x7200 con los datos de los 7200 pacientes caracterizados por 15 atributos binarios y 6 atributos continuos.
- *thyroidTargets*: matriz de 3x7200 en donde se asocia un vector de tres clases a cada paciente. En este vector se define a cuál de las tres clases pertenece el paciente.

Las tres clases que contiene el dataset son:

1. Paciente sano.
2. Paciente con hipertiroidismo.
3. Paciente con hipotiroidismo.

La descripción precisa del significado real de cada atributo y los valores que puede tomar la podemos encontrar en la página web del repositorio [1]. Para nuestra tarea no nos es relevante.

Nuestro objetivo es construir una red neuronal y entrenarla con los datos reales propocionados para que posteriormente sea capaz de clasificar a nueveos pacientes.

Dividiremos aleatoriamente los datos que contiene el data set en tres grupos:

- *Training*(70 %): lo utilizaremos para el entrenamiento supervisado.
- *Validation*(15 %): lo utilizaremos para comprobar cuándo la red está generalizando y detener el entranamiento para evitar el sobreajuste (*overfitting*).
- *Test*(15 %): lo utilizaremos de forma totalmente independiente al entrenamiento para comprobar la generalización de la red.

## D. Descripción del procedimiento

Para resolver la tarea planteada creamos una red neuronal con una topología *patternet*. Esta topología posee dos capas *feedforward*, con una función de activación sigmoideal en la capa oculta y una función *softmax* en la capa de salida. Se caracteriza por poder clasificar vectores con una precisión arbitrariamente buena si posee suficientes neuronas en su capa oculta.

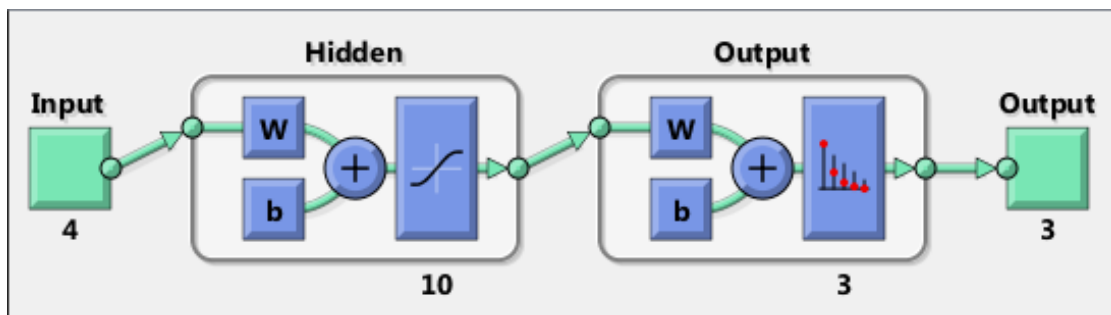


Figura 1: Pattern recognition network.

En nuestro estudio vamos a variar el número de neuronas de la capa oculta para intentar conseguir los mejores resultados (tanto en error, como en tiempo de

entrenamiento). La capa de salida siempre va a tener tres neuronas, cada una se encargará de evaluar la pertenencia a una de las tres clases.

Otro elemento que puede afectar al rendimiento de nuestra red es la función de entrenamiento utilizada. Probaremos ocho funciones distintas que implementa Matlab para comparar resultados.

Para realizar el estudio lo más automatizado posible, hemos creado una función (*autoTrain*) que se encarga de entrenar la red de acuerdo con los parámetros pasados. Por otro lado tenemos un script (*trainAndShow.m*) que llama a esta función variando el número de neuronas de la capa oculta (de 1 a 100 con paso 1) y los algoritmos de entrenamiento (*trainbfg*, *trainrp*, *trainscg*, *traincgb*, *traincgf*, *traincgp*, *trainoss* y *trainidx*). Además, repite el entrenamiento de cada caso 10 veces para disminuir la varianza y la aleatoriedad.

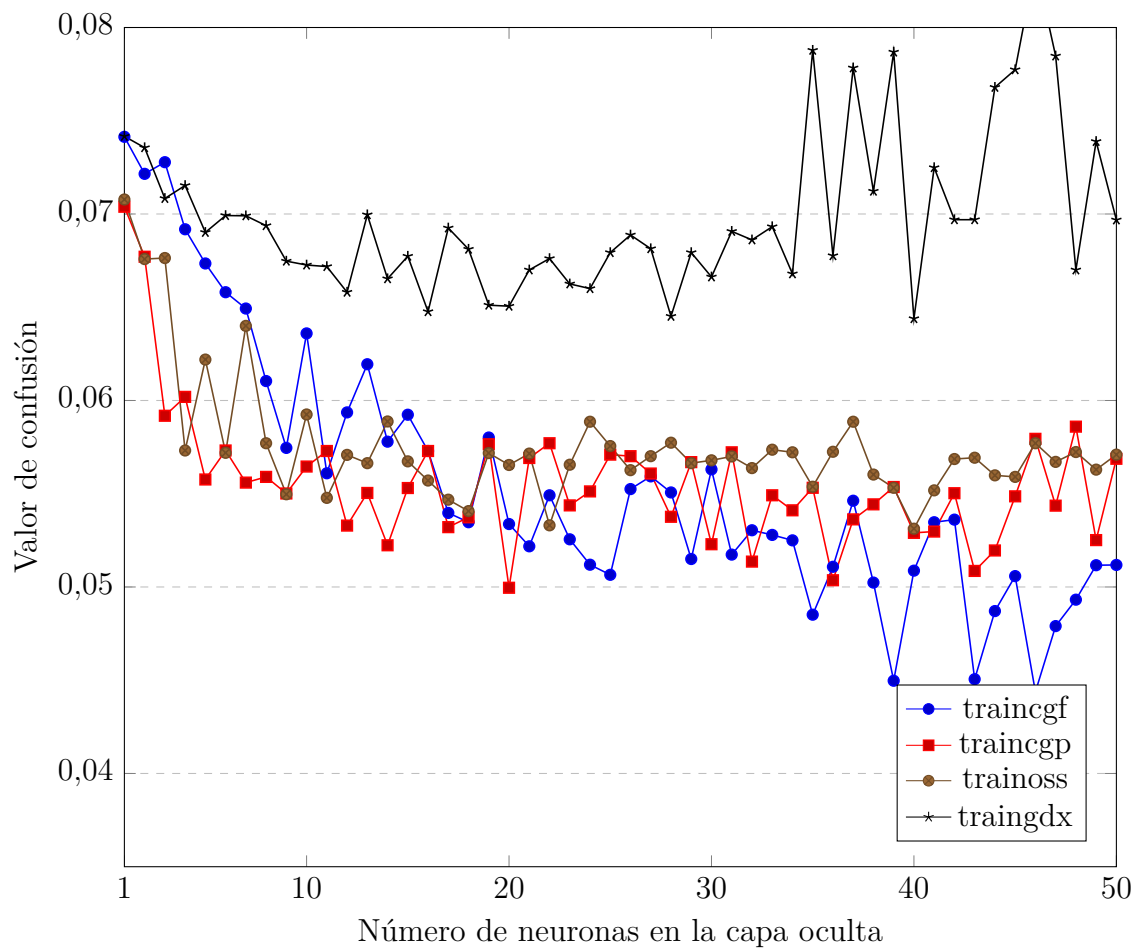
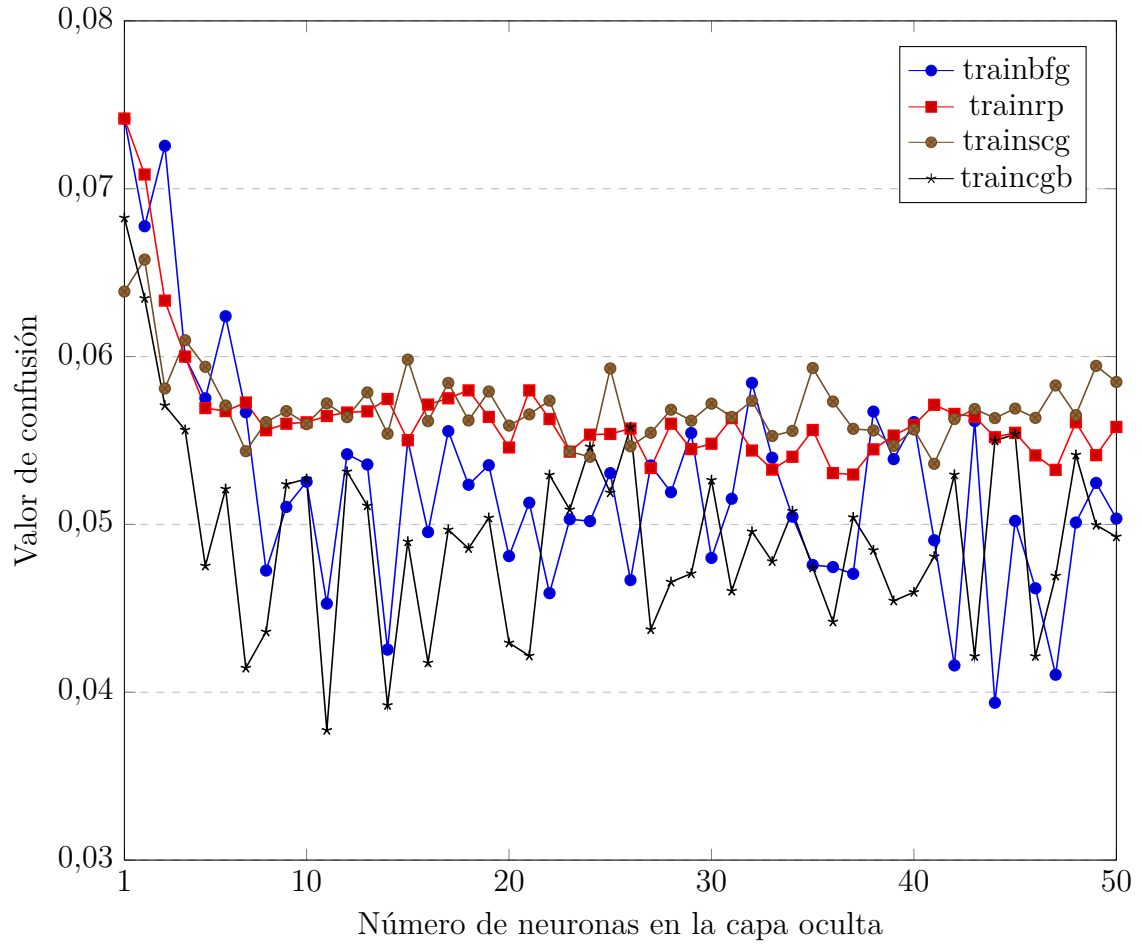
El script proporciona como salida el valor de confusión medio en cada caso, la matriz de confusión completa y el tiempo medio de entrenamiento. Con tres formatos de salida, un primer formato legible para humanos, otro formateado para la librería de gráficos *pgfplots* y un tercer formato como CSV.

*\*Todo el código generado en esta práctica se encuentra disponible en el siguiente repositorio: [P1\\_Thyroid](#).*

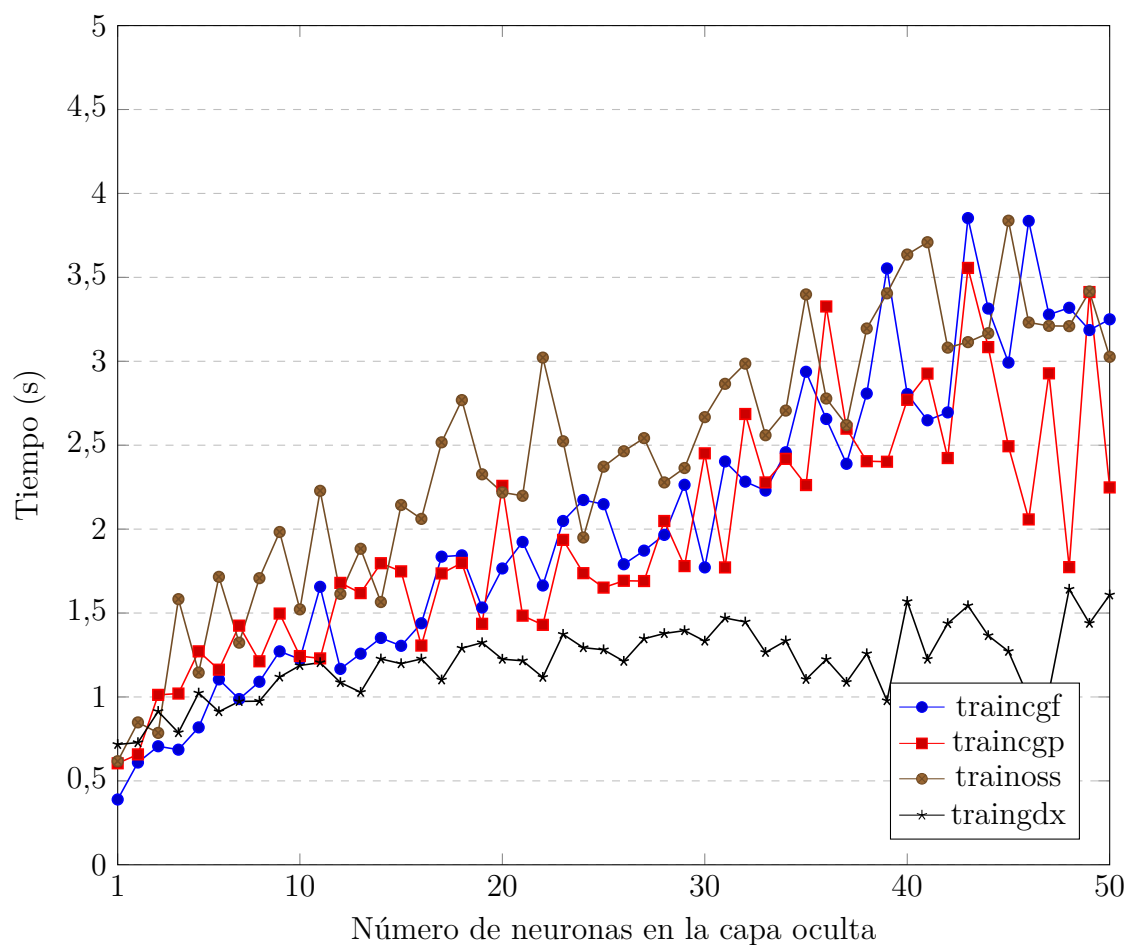
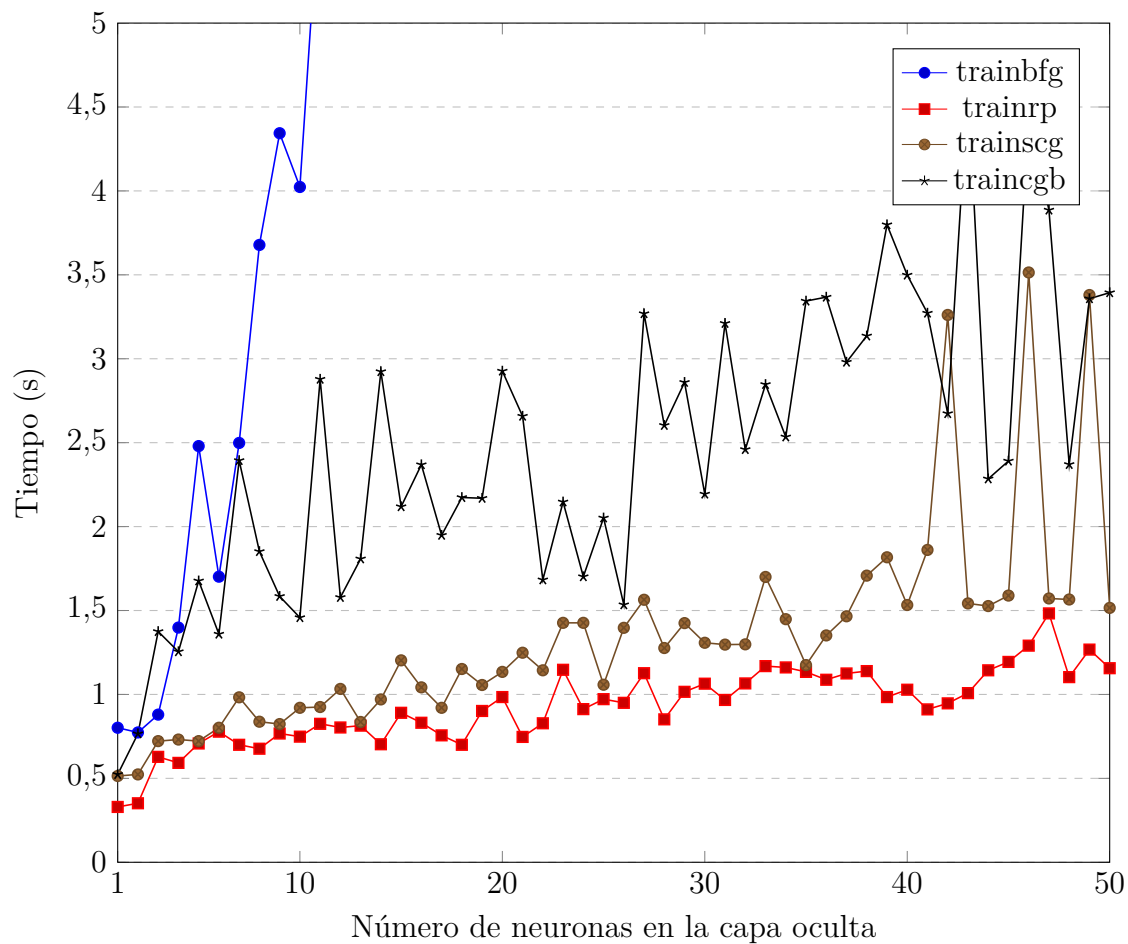
## E. Estudio

asdfasdf

Evolución del valor de confusión en la red neuronal



Evolución del tiempo de entrenamiento



## **F. Mejor resultado obtenido**

asdfasdf

## **G. Explotación de la red**

asdfasdf



## Referencias

- [1] D.J. Newman A. Asuncion. UCI machine learning repository, 2007. URL <https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease>.