

Inteligencia Computacional

Guía de trabajos prácticos 1

Perceptrón simple y perceptrón multicapa

1. Objetivos

- Aplicar diferentes arquitecturas de redes neuronales a la clasificación automática de datos reales.
- Verificar experimentalmente las limitaciones del método de separación por hiperplanos.
- Profundizar en los conceptos teóricos relacionados con la retro-propagación del error.
- Analizar y utilizar un algoritmo de entrenamiento completo para el perceptrón multicapa.
- Comprender la importancia de la validación cruzada y conocer las técnicas alternativas.

2. Trabajos prácticos

Ejercicio 1: Realice un programa que permita el entrenamiento y prueba de un perceptrón simple con una cantidad variable de entradas. El programa debe proveer las siguientes facilidades:

- lectura de los patrones de entrenamiento (entradas y salidas) desde un archivo en formato texto separado por comas,
- selección del criterio de finalización del entrenamiento,
- selección del número máximo de épocas de entrenamiento,
- selección de la tasa de aprendizaje (η),
- prueba del perceptrón entrenado mediante archivos de texto con el mismo formato separado por comas,

Si bien se trata de un programa muy sencillo, realice un buen diseño que pueda ser ampliado para implementar estructuras neuronales más complejas. Una vez obtenido dicho programa:

- a) Pruébalo en la resolución de los problemas OR y XOR, utilizando como patrones de entrenamiento y prueba los puntos (1,1), (1,-1), (-1,1) y (-1,-1) y un conjunto más amplio generado con pequeñas desviaciones aleatorias ($< 5\%$) en torno a éstos. Recuerde que para que la prueba tenga validez se deben utilizar patrones nunca presentados en el entrenamiento.
- b) Implemente una rutina de graficación que permita visualizar, para el caso de dos entradas, los patrones utilizados y la recta de separación que se va ajustando durante el entrenamiento del perceptrón simple. Utilice dicha rutina para visualizar el entrenamiento en los problemas OR y XOR.

Ejercicio 2: Realice un programa que permita generar un conjunto de particiones de entrenamiento y prueba a partir de un único archivo de datos en formato texto separado por comas. El programa debe permitir seleccionar la cantidad de particiones y el porcentaje de patrones de entrenamiento y prueba. Para probarlo:

- a) Genere una serie de datos a partir de los valores de la Tabla 2 y un conjunto más amplio generado con pequeñas desviaciones aleatorias ($< 10\%$) en torno a ellos. Realice la validación cruzada del perceptrón simple con 1000 datos y 5 particiones de entrenamiento y prueba con relación 80/20.
- b) Genere datos utilizando el ejemplo de la tabla anterior pero modificando el punto $\mathbf{x} = [-1 \quad +1 \quad -1] \rightarrow y_d = 1$. Realice la validación cruzada del perceptrón simple con 5000 datos y 50 particiones de entrenamiento y prueba con relación 80/20. Compare los resultados generando los datos con desviaciones aleatorias de 10, 50 y 70 % en torno a los valores originales.

Ejercicio 3: Implemente el algoritmo de retropropagación para un perceptrón multicapa de forma que se puedan elegir libremente las cantidades de capas de la red y de neuronas en cada capa.

- Para entrenar y probar el algoritmo utilice la base de datos `concent.csv`, que consiste en dos clases distribuidas en forma concéntrica como muestra la Figura 1. Represente

x_1	x_2	x_3	y_d
-1	-1	-1	1
-1	-1	1	1
-1	1	-1	-1
-1	1	1	1
1	-1	-1	-1
1	-1	1	-1
1	1	-1	1
1	1	1	-1

Tabla 1: Clases para el Ejercicio 2.

gráficamente, con diferentes colores, el resultado de la clasificación realizada por el perceptrón multicapa.

- Verifique experimentalmente la influencia de la incorporación del término de momento en la ecuación de adaptación de los pesos. Para esto, modifique su implementación para incluir el término de momento y compare la velocidad de convergencia del algoritmo.

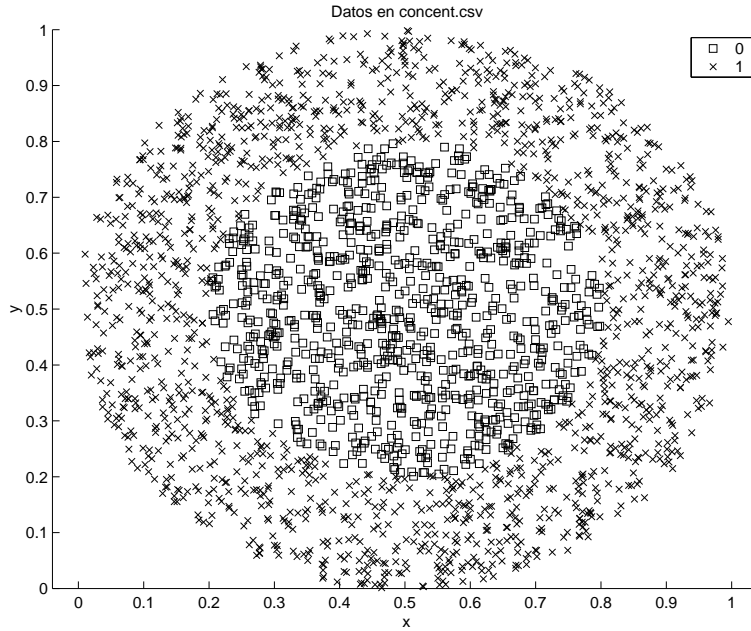


Figura 1: Distribución de clases para la base de datos `concent.csv`.

Ejercicio 4: *Iris* es el género una planta herbácea con flores que se utilizan en decoración. Dentro de este género existen muy diver-

sas especies entre las que se han estudiado la *Iris setosa*, la *Iris versicolor* y la *Iris virginica* (ver Figura 2).

Estas tres especies pueden distinguirse según las dimensiones de sus pétalos y sépalos. Un grupo de investigadores ha recopilado la información correspondiente a las longitudes y anchos de los pétalos y sépalos de 50 plantas de cada especie. En el archivo `iris.csv` se encuentran estas mediciones (en cm.) junto con un valor numérico que indica la especie reconocida por los investigadores (0=setosa; 1=versicolor; 2=virginica). Para la clasificación de una gran cantidad de estas plantas se desea crear un programa que aprenda de estos 150 patrones para luego realizar la tarea de forma automática.

Para la validación utilice los métodos *leave-k-out* y *leave-one-out* con un perceptrón multicapa como clasificador. Estime error de clasificación promedio y la desviación estándar según los dos métodos.



Figura 2: Muestra de la especie *Iris virginica*.

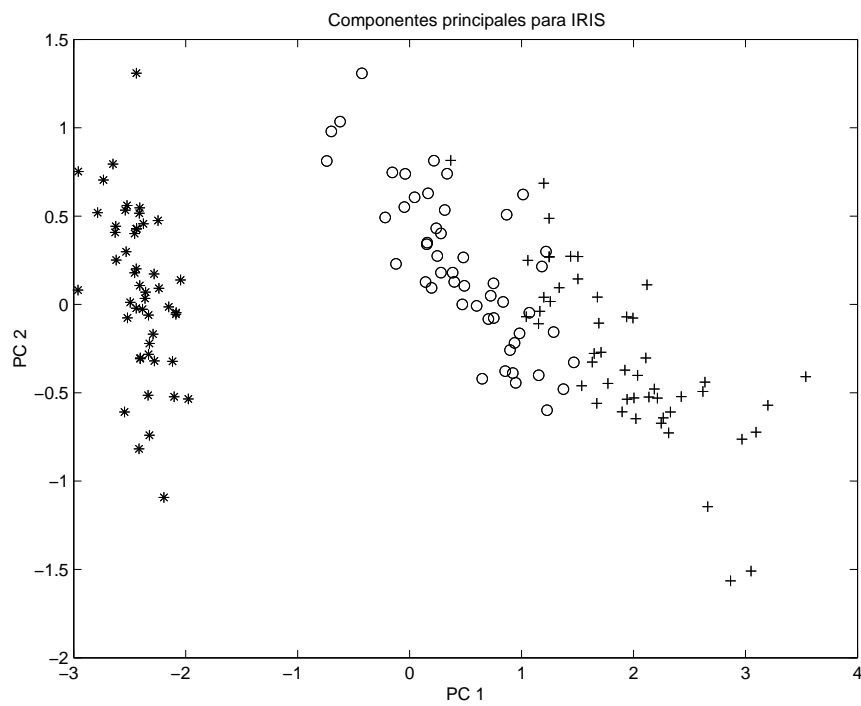


Figura 3: Proyección en \mathbb{R}^2 de la distribución de clases para la base de datos `iris.csv`.