Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías
Seminario de Solución de Problemas de Inteligencia Artificial II
Dr. Diego Oliva
Depto. De Ciencias Computacionales
Practica 1
Perceptrón Simple y 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.
- Implementar un algoritmo de entrenamiento para el perceptrón multicapa y analizar su desempeño.
- Utilizar diferentes técnicas de validación cruzada y valorar su importancia.

2. Actividades

Ejercicio 1: Realice un programa que permita el entrenamiento y prueba de un perceptrón simple con una cantidad variable de entradas. El programa debe realizar lo siguiente:

- 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.
- Prueba del perceptrón entrenado en datos reales.

Una vez que se generó el programa, debe ser probado considerando lo siguiente:

1. Problema XOR. Los patrones para este problema son los puntos (1,1), (1,-1), (-1,1), (-1,-1). Además, deben considerarse alteraciones aleatorias (< 5%). Se debe generar un set de entrenamiento y otro de prueba. Utilizar los archivos XORtrn.csv y XORtst.csv
2. Mostrar gráficamente los patrones utilizado y la recta que los separa.

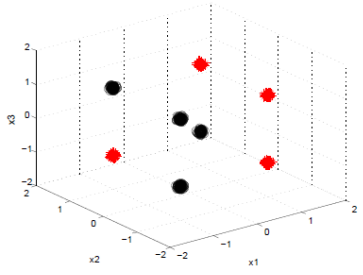
Ejercicio 2: Realizar un programa que permita generar un conjunto de particiones de entrenamiento considerando un dataset. El programa debe permitir seleccionar la cantidad e particiones y el porcentaje de patrones de entrenamiento y prueba. Para verificar su funcionamiento se debe realizar lo siguiente:

1. Usar el archivo spheres1d10.csv que contiene datos generados en base a la Tabla 1. Estos datos consideran alteraciones aleatorias (<10%), tal como se muestra en la Figura 1(a). Usando el perceptrón simple, crear cinco particiones de entrenamiento usando 80% de los datos y 20% para la generalización.

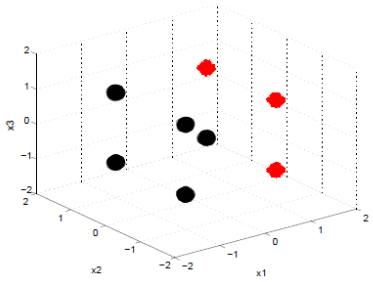
x_1	x_2	x_3	y_d
-1	-1	-1	1
-1	-1	1	1
-1	1	-1	-1
-1	1	1	1
1	-1	-1	-1
1	-1	1	-1
1	1	-1	1
1	1	1	-1

Tabla 1. Clases para el ejercicio 2.

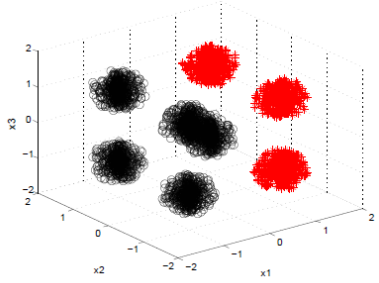
2. Considerando la Tabla 1, modificar el punto x=[-1, +1, -1] → y_d = 1. Con esto se genera un nuevo dataset. Los archivos spheres2d10.csv, spheres2d50.csv y spheres2d70.csv contienen los datos perturbados en un 10%, 50% y 70% y se presentan en las Figuras 1 (b), (c), (d). mediante el perceptrón simple realizar una clasificación con 10 particiones usando 80% de los datos y 20% para la generalización.



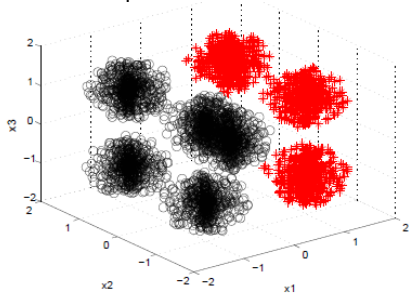
(a) Datos de la Tabla 1 original con perturbaciones <10%



(b) Datos de la Tabla 1 modificada con perturbaciones <10%



(c) Datos de la Tabla 1 modificada con perturbaciones <50%



(d) Datos de la Tabla 1 modificada con perturbaciones <70%

Figura 1. Distribución de clases para el ejercicio 2.

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

1. Para entrenar y probar el algoritmo se debe usar el dataset concentlite.csv, el cual contiene dos clases distribuidas de forma concéntrica (Figura 2). Debe representarse gráficamente con diferentes colores el resultado de la clasificación hecha por el perceptrón multicapa.

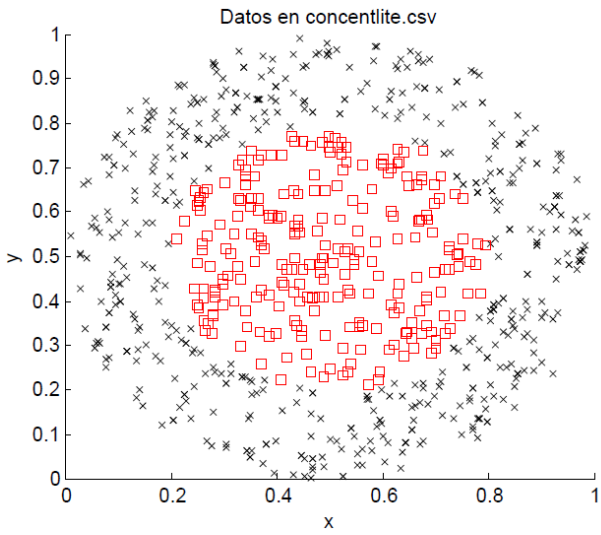


Figura 2. Distribución de clases para el dataset concentlite.

2. Probar otra regla se aprendizaje o alguna modificación a la retropropagación.

Ejercicio 4: Iris es el género de una planta herbácea con flores que se utilizan en decoración. Dentro de este género existen muy diversas especies entre las que se han estudiado la Iris setosa, la Iris versicolor y la Iris virginica (ver Figura 3).

Las tres especies se pueden diferenciar en base a las dimensiones de sus pétalos y sépalos. Se ha recopilado la información de 50 plantas de cada especie y se han almacenado en el archivo irisbin.csv. Dichas mediciones están en centímetros junto con un código binario que indica la especie a la que pertenece [-1, -1, 1] = setosa, [-1, 1, -1] = versicolor, [1, -1, 11] = virginica, la Figura 4 muestra la distribución de los datos contenidos en el archivo. Se debe crear un programa capaz de clasificar automáticamente los datos de 150 patrones usando un perceptrón multicapa. Es recomendable considerar 80% de los datos para entrenamiento y 20% para generalización.


Figura 3. Muestra de la especie Iris virginica.

Con la estructura optima de la red, se deben validar los resultados usando lo métodos *leave-k-out* y *leave-one-out* con un perceptrón multicapa como clasificador. Se debe estimar el error esperado de clasificación, el promedio y la desviación estándar de ambos métodos

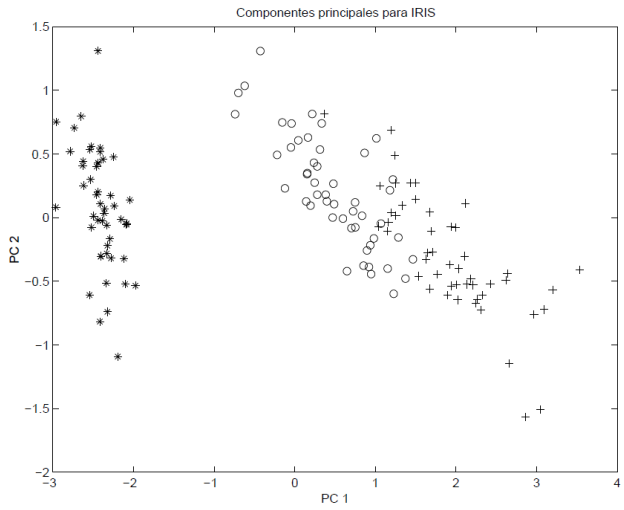


Figura 4. Proyección en dos dimensiones de la distribución de clases para el dataset Iris