OPENCOURSEWARE
REDES DE NEURONAS ARTIFICIALES
Inés M. Galván – José M. Valls

Examen Final

Pregunta 1 (1 punto)

Responda brevemente a las siguientes preguntas:

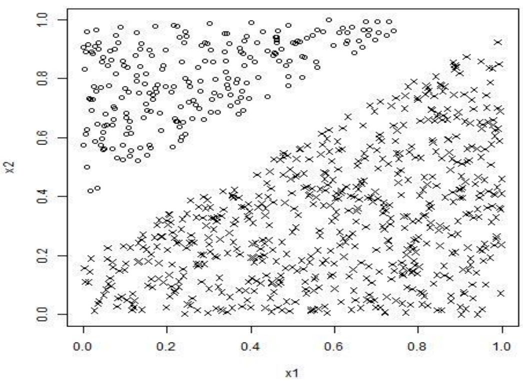
a) ¿Cuál es el objetivo en el aprendizaje del Perceptron Simple?
b) ¿Cuál es el objetivo en el aprendizaje del ADALINE?
c) ¿Cuál es el objetivo en el aprendizaje del Perceptron Multicapa?
d) ¿Cuál es el objetivo en el aprendizaje de las Redes de Base Radial?
e) ¿Cuál es el objetivo en el aprendizaje de los Mapas de Kohonen?

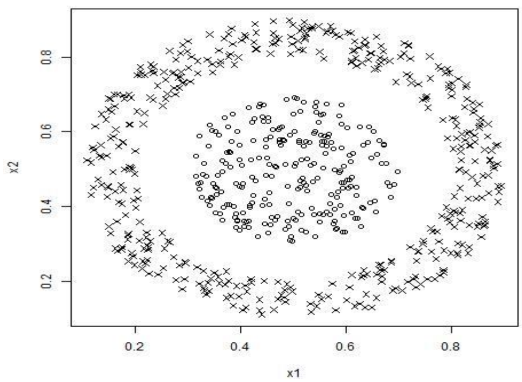
Respuesta

a) Construir un hiperplano que separe los patrones de entrada en dos clases de modo supervisado
b) Minimizar el error que mide la diferencia al cuadrado entre la salida deseada y la salida de la red para todos los patrones de entrenamiento a través de una aproximación lineal
c) Minimizar el error que mide la diferencia al cuadrado entre la salida deseada y la salida de la red para todos los patrones de entrenamiento través de una aproximación no lineal
d) Minimizar el error que mide la diferencia al cuadrado entre la salida deseada y la salida de la red para todos los patrones de entrenamiento través de una aproximación local y no lineal
e) Agrupar datos en el espacio de entrada con características similares

Pregunta 2 (1 punto)

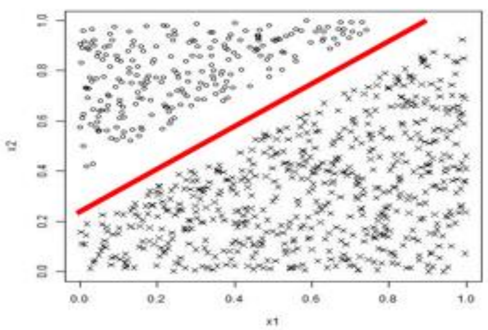
Dados los patrones que se muestran en las figuras (problema de clasificación), dibuje una solución que proporcionaría después del aprendizaje las siguientes redes: Perceptron Simple, Perceptron Multicapa y Mapas de Kohonen.

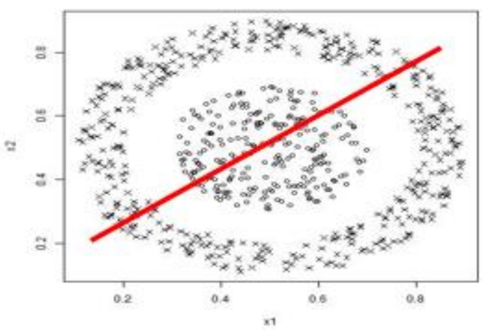




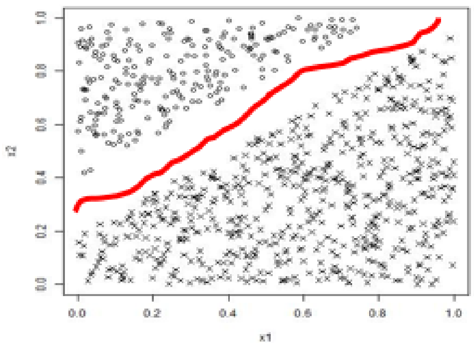
Respuesta

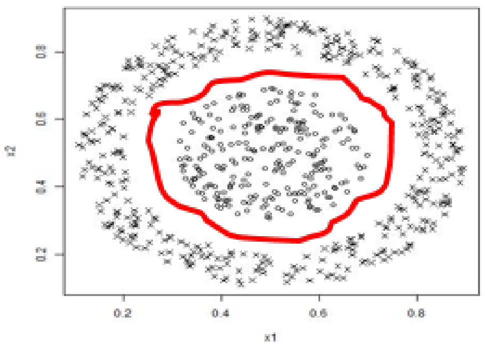
Perceptron Simple



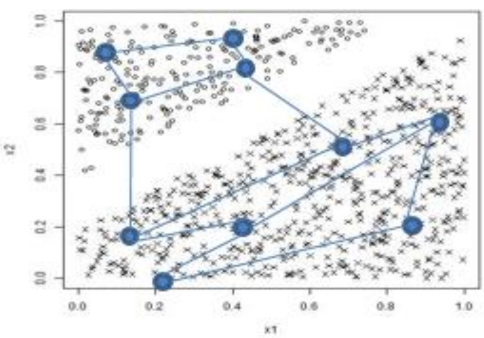


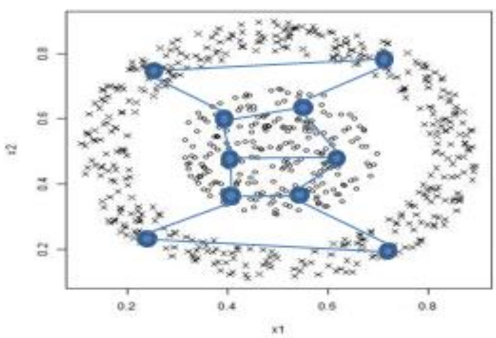
Perceptron Multicapa





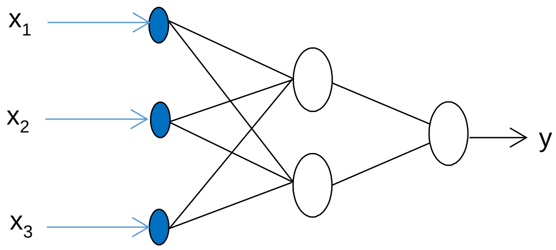
Mapas de Kohonen



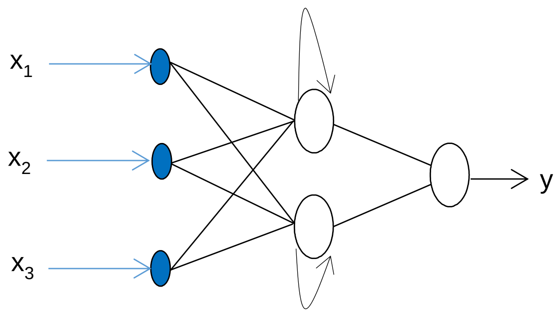


Pregunta 3 (2 puntos)

a) Dada la arquitectura de Perceptron Multicapa que se muestrea en la figura, con función de activación sigmoidal para las neuronas ocultas y la neurona de salida, escriba al expresión para calcular las activaciones de las neuronas de la red y la ley de aprendizaje para modificar los pesos.



b) Supóngase que la arquitectura anterior, se modifica introduciendo algunas conexiones recurrentes como se muestra en la siguiente figura. Indique las expresiones para calcular las activaciones de la red y explique razonadamente si la ley de aprendizaje del apartado anterior puede utilizarse para modificar los pesos de la red.



Respuesta

a) Sea $W^1=(w^1_{ij})$ (i=1,2,3; j=1,2) los pesos de la capa de entrada a la capa oculta y $W^2=(w^2_j)$ (j=1,2) los pesos de la capa oculta a la neurona de salida. Se $U=(u_j)$ los umbrales de las neuronas ocultas y v el umbral de la neurona de salida. Entonces las activaciones vienen dadas por:

Neuronas ocultas:

$$a_1 = f(w^1_{11}x_1 + w^1_{21}x_2 + w^1_{31}x_3 + u_1), \quad a_2 = f(w^1_{12}x_1 + w^1_{22}x_2 + w^1_{32}x_3 + u_2)$$

Neurona de salida:

$$y = f(w^2_1 a_1 + w^2_2 a_2 + v)$$

La ley de aprendizaje es:

$$w^2_j(n) = w^2_j(n-1) + \alpha\delta^3(n)a_j(n)$$

Siendo $\delta^3 = (s - y)y(1-y)$ con s la salida deseada

$$w^1_{ij}(n) = w^1_{ij}(n-1) + \alpha\delta^2_j(n)x_i(n)$$

Siendo $\delta^3_j = a_j(1-a_j)\delta^3 w^2_j$

b) Sea w^r_1 y w^r_2 los pesos de las conexiones recurrentes. Entonces las activaciones de las neuronas ocultas vienen dadas por:

$$a_1(t) = f(w^1_{11}x_1(t) + w^1_{21}x_2(t) + w^1_{31}x_3(t) + w^r_1 a_1(t-1) + u_1)$$

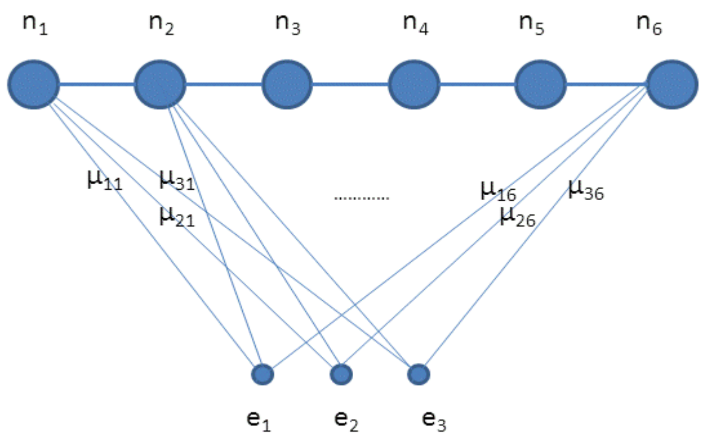
$$a_2(t) = f(w^1_{12}x_1 + w^1_{22}x_2 + w^1_{32}x_3 + w^r_2 a_2(t-1) + u_2)$$

La activación de la neurona de salida es igual que en el apartado anterior, introduciendo la variable tiempo: $y(t) = f(w_1^2 a_1(t) + w_2^2 a_2(t) + v)$

Los pesos de la capa oculta a la capa de salida, $W^2=(w^2_j)$, pueden adaptarse utilizando las expresiones del apartado anterior, pero los pesos $W^1=(w^1_{ij})$ y w^r_i no, debido a que cuando se calcula la derivada de la activación $a_i(t)$ respecto al peso w^1_{ij} o al peso w^r_i, es necesario también tener en cuenta la derivada de $a_i(t-1)$ respecto a dichos pesos. Para estos pesos sería necesario aplicar el algoritmo de aprendizaje en tiempo real o retropropagación a través del tiempo.

Pregunta 4 (2 puntos)

Considérese el mapa de Kohonen unidimensional que se muestra en la siguiente figura:



En cuanto a la extensión del vecindario, se considera que sólo contiene a las neuronas inmediatamente más próximas (neurona a la derecha y neurona a la izquierda), siendo la distancia de vecindario entre una neurona y sus vecinas igual a 2. Por ejemplo, el vecindario de la neurona n_2 sólo incluye a las neuronas n_3 y n_1 y la distancia de vecindario de n_2 con sus vecinas será $d(n_2,n_1) = d(n_2,n_3) = 2$. La capa de entrada tiene tres neuronas. Los pesos iniciales de cada una de las neuronas del mapa vienen indicados en la tabla 1. $\boldsymbol{\mu_{ij}}$ representa el peso de la conexión entre la entrada $\mathbf{e_i}$ y la neurona del mapa $\mathbf{n_j}$

	μ_{1j}	μ_{2j}	μ_{3j}
n_1	2	2	3
n_2	8	7.2	9.1
n_3	1	8	9
n_4	2	3	3
n_5	8	9	9
n_6	2.1	2.9	4

Tabla 1: Pesos de las conexiones entre la capa de entrada y la capa de competición

Dado el patrón de entrada **e=(3, 3, 3),** se pide calcular los nuevos pesos (o nuevas posiciones) de las neuronas de la red tras una iteración del proceso de aprendizaje del mapa. Construir una tabla similar a la tabla 1 donde se indicarán los nuevos pesos de las neuronas. Se utilizará una tasa de aprendizaje para esta iteración de $\alpha(t) = 0.5$.

Observación: no es necesario calcular explícitamente las distancias euclídeas.

Respuesta

Se puede ver que la neurona más cercana al patrón de entrada es n4, por lo tanto se actualizarán los pesos de esta neurona y las de su vecindario (n3 y n5).

$\Delta\mu_{14}=\alpha(t)\ (e_1 - \mu_{14}) = 0.5\ \ (3 - 2) = 0.5$; $\mu_{14} = 2 + 0.5 = 2.5$

$\Delta\mu_{24}=\alpha(t)\ (e_2 - \mu_{24}) = 0.5\ \ (3 - 3) = 0$; $\mu_{24} = 3$

$\Delta\mu_{34}=\alpha(t)\ (e_3 - \mu_{34}) = 0.5\ \ (3 - 3) = 0$; $\mu_{34} = 3$

Neuronas vecinas:

n_3

$\Delta\mu_{13}=\alpha(t)/2\ .\ (e_1 - \mu_{13}) = 0.25\ \ (3 - 1) = 0.5$; $\mu_{13} = 1 + 0.5 = 1.5$

$\Delta\mu_{23}=\alpha(t)/2\ .\ (e_2 - \mu_{23}) = 0.25\ \ (3 - 8) = -\ 1.25$; $\mu_{23} = 8 - 1.25 = 6.75$

$\Delta\mu_{33}=\alpha(t)/2\ .\ (e_3 - \mu_{33}) = 0.25\ \ (3 - 9) = -\ 1.5$; $\mu_{13} = 9 - 1.5 = 7.5$

n_5

$\Delta\mu_{15}=\alpha(t)/2\ .\ (e_1 - \mu_{15}) = 0.25\ \ (3 - 8) = -1.25$; $\mu_{15} = 8 - 1.25 = 6.75$

$\Delta\mu_{25}=\alpha(t)/2\ .\ (e_2 - \mu_{25}) = 0.25\ \ (3 - 9) = -\ 1.5$; $\mu_{25} = 9 - 1.5 = 7.5$

$\Delta\mu_{35}=\alpha(t)/2\ .\ (e_3 - \mu_{35}) = 0.25\ \ (3 - 9) = -\ 1.5$; $\mu_{15} = 9 - 1.5 = 7.5$

Tabla de pesos finales:

	µ1j	**µ2j**	**µ3j**
n1	2	2	3
n2	8	7.2	9.1
n3	***1.5***	***6.75***	***7.5***
n4	***2.5***	***3***	***3***
n5	***6.75***	***7.5***	***7.5***
n6	2.1	2.9	4

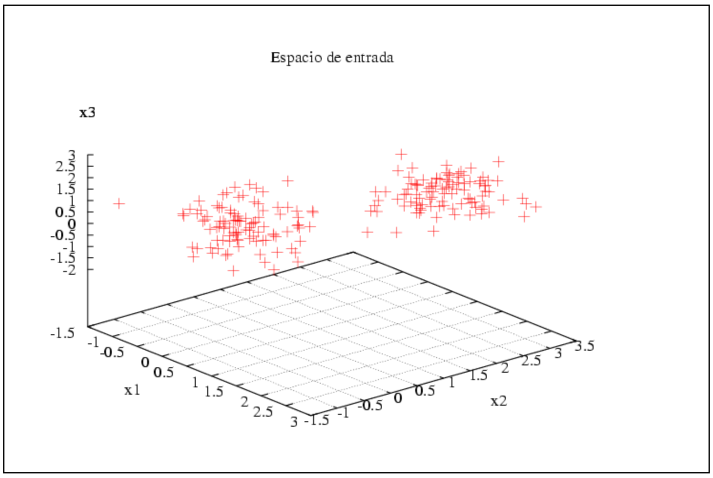
Pregunta 5 (2 puntos)

Se dispone de un conjunto de datos o patrones que corresponden a una función de tres dimensiones y=f(x1, x2, x3). A modo de ejemplo, se muestran algunos de estos patrones:

x1	x2	x3	y
1.56308784	2.618522336	2.133589992	3.58357
-0.242480382	-0.078638323	0.163898903	-4.56532
1.6465747	2.65788	0.456778	- 2.54535

………

En la siguiente figura se han representado estos puntos en el espacio de entrada (x1, x2, x3).



Se desea construir un modelo supervisado de red de neuronas para aproximar dicha función. Se pide:

a) ¿Qué modelo de red de neuronas podría utilizarse para abordar el problema? Elija un modelo de red para responder el resto de las preguntas.

b) ¿Sería conveniente realizar algún procesado de los datos? En caso afirmativo, indique cuál.

c) Para cada una de las posibles redes de neuronas que se puedan utilizar, indique una posible arquitectura de la red para abordar el problema

d) ¿Cómo mediría la capacidad de generalización de la red?

e) Supóngase que con los mismos datos de entrada se quiere resolver un problema de clasificación donde los datos de cada nube de puntos corresponden a clases diferentes. Proponga de manera razonada un modelo adecuado para abordar el problema, indicando si los patrones disponibles tendrían que sufrir alguna modificación.

Respuesta

a) Problema de regresión. Además es no lineal por los datos de ejemplo mostrados, ya que el primer y el tercer patrón son muy similares en la entrada pero tienen salidas completamente diferentes. Por tanto PM o RBR. Adaline no sería adecuado al ser no lineal.

b) Normalizar las variables de entrada y salida en el intervalo [0,1]. Para ello se elige el mínimo y el máximo de cada variable y se normaliza utilizando la siguiente expresión:

$\mathrm{VarNor}_i = (\mathrm{Var}_i - \mathrm{VMin}_i)/(\mathrm{VMax}_i - \mathrm{VMin}_i)$ i =, 1, 2, 3, 4

c) Siempre serán 3 entradas y 1 salida. Tanto si se elige el PM como las RBR, el número de neuronas ocultas habría que determinarlo experimentalmente. Para las RBR y debido al carácter local podrían utilizarse 2 ocultas, o bien 4 dos para cada agrupación de datos.
d) Extrayendo un conjunto de datos de test del total disponible (20%) y evaluando el error que comete la red sin modificar los pesos sobre este conjunto. Si es del orden del error de entrenamiento podemos decir que la red generaliza bien.
e) Las variables *x1, x2 y x3* serían la entrada al modelo. Dado que las clases son linealmente separables, se puede utilizar el PS y no es necesario utilizar el PM, ni las RBR.
La variable *y* que se muestra en la tabla no sería la salida deseada. En este caso, la variable de salida deseada se podría definir como:
Si (x1, x2, x3) pertenece a la primera nube de untos, entonces salida 1 y si no -1, que en este caso y según se observa en la figura sería: Si x2 < 2.5, salida 1, si no salida -1. Esta última expresión habría que comprobarla con el conjunto total de datos.
Por tanto, 3 entradas y 1 salida

Pregunta 6 (2 puntos)

Se dispone de un conjunto de datos sobre diferentes tipos de levaduras (hongos unicelulares), concretamente 10 tipos. El conjunto está compuesto por un total de 1486 instancias o patrones. Cada instancia está formada por 8 atributos numéricos de entrada (a_1, a_2,..., a_8), que representan las características de las levaduras, y 1 atributo de salida que indica el tipo de levadura (Tipo1, Tipo2, …., Tipo10). El número de ejemplos disponible de cada tipo de levadura es el siguiente: Tipo1: 463; Tipo2: 429; Tipo3: 244; Tipo4: 163; Tipo5: 51; Tipo6: 44; Tipo7: 37; Tipo8: 30; Tipo9: 20; Tipo10: 5

Se pretende construir un modelo para clasificar las levaduras a partir de sus características. Se pide:

a) Con qué tipos de redes y algoritmos conocidos puede abordarse este problema.
b) Explique el procedimiento a seguir para obtener los datos de entrenamiento y test.
c) Para cada tipo de red o algoritmo que pueda utilizarse, indique una posible arquitectura
d) Indique los parámetros que debería tener en cuenta para el aprendizaje de estas redes.

Respuesta

a) Se trata de un problema de clasificación (no línea, seguramente). Por tanto: el PM, las RBR, Mapas de kohonen, interpretándolos para clasificación supervisada (con la clase de cada patrón, se calibra el mapa, asignando a cada neurona una clase. Para clasificar un patrón, se calcula la neurona del mapa más cercana y se le asigna la clase de dicha neurona. También se puede utilizar LVQ
b) Al tratarse de un problema de clasificación con clases desbalanceadas (por ejemplo de la clase 1 se disponen de 463 patrones, mientras que para la clase 10, solo se dispone de 5 patrones), para separar los datos de entrenamiento y test es necesario mantener la proporción de cada clase con respecto al conjunto original. Así, si se extraen el 70% (por ejemplo) de los datos para entrenamiento, el procedimiento sería extraer el 70% de cada clase. Se cogen entonces 324, 300, 171, 114, 36, 31, 21, 14, 3 de las clases 1, 2, …, 10, respectivamente para entrenamiento y el resto para test. Se unen los datos para todas las clases y se aleatorizan los datos de entrenamiento (el test no es necesario)
c) Para todas las redes se utilizaran 8 entradas. Para el PM y las RBR, la mejor opción es utilizar 10 neuronas de salida, de modo que cada neurona de salida represente una clase. En este caso, la salida deseada se define como un vector de 10 coordenadas,

(x1,…,x10) siendo xi=1 si el patrón pertenece a la clase i y el resto de las coordenadas son 0. Para estas redes (PM y RBR) el número de neuronas ocultas se determina experimentalmente.

En el caso de los Mapas de Kohonen, el número de neuronas en la capa de competición también se determinará experimentalmente, pero al tener 10 clases, sería conveniente partir de un número de neuronas mayor que 10

En el caso de LVQ hay de decidir el número de prototipos por clase, que también se determina experimentalmente. Se podría partir de 1 prototipo por clase (en total 10) e ir incrementando en una unidad el número de prototipos pro clases, resultando 20, 30, etc. prototipos en total.

d) PM: razón de aprendizaje y número de ciclos

RBF: en la fase supervisada, al igual que el PM, razón de aprendizaje y número de ciclos. En la fase no supervisada ningún parámetro

Mapas de Kohonen: valor inicial de la razón de aprendizaje, tipo de vecindario, tamaño inicial del vecindario y razón para ir decreciendo el tamaño del vecindario

LVQ: Número de iteraciones