uc3m | Universidad Carlos III de Madrid

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

Preguntas y Ejercicios para Evaluación: Tema 3

1. Señale las afirmaciones correctas:
   a) El Perceptron multicapa con funciones de activación lineal en todas sus neuronas puede resolver un problema de regresión no lineal
   b) El Perceptron Multicapa con funciones de activación lineal en todas sus neuronas es equivalente a un Adaline
   c) El Perceptron multicapa con funciones de activación sigmoidal en las neuronas ocultas y función de activación lineal en las neuronas de salida puede resolver un problema de regresión no lineal
   d) El Perceptron multicapa con funciones de activación sigmoidal en las neuronas ocultas y función de activación umbral en las neuronas de salida puede resolver un problema de regresión no lineal
   e) En el Perceptron Multicapa las neuronas ocultas compiten entre ellas para conseguir un buen nivel de activación.
   f) En el Perceptron Multicapa el uso de función de activación sigmoidal impide resolver problemas lineales

   Respuesta
   b), c)

2. ¿Cuál o cuáles de las siguientes afirmaciones es cierta en el algoritmo de retropropagación?

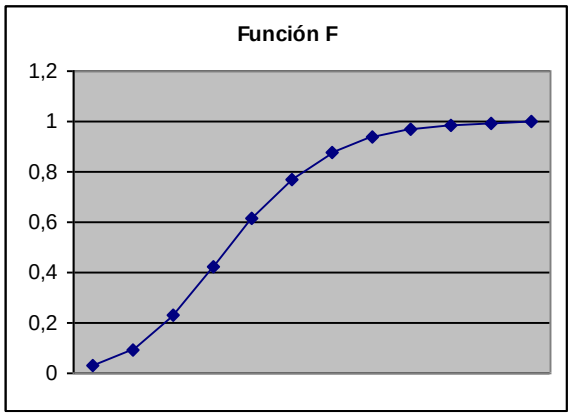
   a) El ajuste de un peso de la última capa se realiza utilizando el delta de la neurona de destino y la activación de la neurona de origen.
   b) El ajuste de un peso de la última capa se realiza utilizando únicamente el error de la neurona de destino y la activación de la neurona de origen.
   c) El ajuste de un peso de la capa de entrada a la primera capa oculta se realiza utilizando únicamente el error medido en la salida y la activación de la neurona oculta.
   d) El ajuste de un peso de la capa de entrada a la primera capa oculta se realiza utilizando únicamente el delta de la neurona oculta y todas las activaciones de las neuronas de entrada.

   Respuesta
   a)

3. Supóngase que se desea aproximar la función $F(x_1,x_2,x_3)$ a partir del conjunto de ejemplos que se muestran en la siguiente tabla, siendo x_1,x_2,x_3 las variables de entrada e $y = F(x_1,x_2,x_3)$ los valores deseados de la función para las variables de entrada. Dicha variable presenta la

forma que se muestra en la figura (en el eje Y se representa el valor de la función para cada uno de los puntos; el eje X representa los patrones).

x_1	x_2	x_3	$y=F(x_1,x_2,x_3)$
0,05	-3	-1	0,03309
0,13	-2,2	-0,6	0,09154
0,21	-1,4	-0,2	0,22882
0,29	-0,8	0,2	0,42678
0,37	-0,4	0,6	0,61420
0,45	0	1	0,77294
0,53	0,4	1,4	0,87921
0,61	0,8	1,8	0,93963
0,69	1,2	2,2	0,97083
0,77	1,6	2,6	0,98614
0,85	2	3	0,99347
0,93	2,4	3,4	0,99694



¿Se puede aproximar con éxito dicha función con el ADALINE? En caso afirmativo, indique los pasos que tendrían que realizarse para conseguir aproximar dichos puntos. En caso negativo, indique las modificaciones que deben realizarse al ADALINE, así como a su ley de aprendizaje, para resolver el problema con éxito.

Respuesta

NO, ya que el ADALINE define funciones lineales y la función que se pretende aproximar es no lineal, se trata de una función sigmoidal.

Para aproximar dicha función, bastaría considerar en la neurona de salida del ADALINE la función de activación sigmoidal entre 0 y 1. Y la ley para modificar los pesos, siguiendo la regla delta generalizada sería:

$w_i\,(t+1) = w_i\,(t) + \alpha(d-y)y(1-y) \cdot x_i$

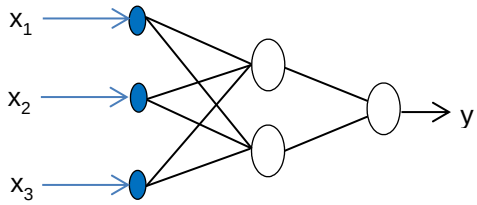
siendo y(1-y) la derivada de la salida de la red (función sigmoidal)

4. ¿Por qué el aprendizaje del Perceptron Multicapa puede detenerse en un mínimo local? ¿Cómo distinguiría una situación de mínimo local con una situación de saturación o parálisis? Razone su respuesta

Respuesta

Los pesos del PM se modifican utilizando la derivada del error, la cual es cero es un mínimo local, por tanto, en un mínimo local los pesos no cambian de una iteración a otra (método del descenso del gradiente), y el error de entrenamiento se estabiliza. En el caso de la parálisis, el error de entrenamiento también se estabiliza, pero en este caso es debido a que las activaciones están en las asíntotas (0 ó 1) (la modificación de los pesos depende de y(1-y)). Para distinguirlo, se pueden analizar los pesos de la arquitectura. Si éstos toman valores grandes (positivos o negativos), seguramente la(s) neurona(s) está(n) saturada(s). A veces, en estado de saturación, si se deja continuar el aprendizaje durante un número de ciclos, el error de entrenamiento acaba disminuyendo

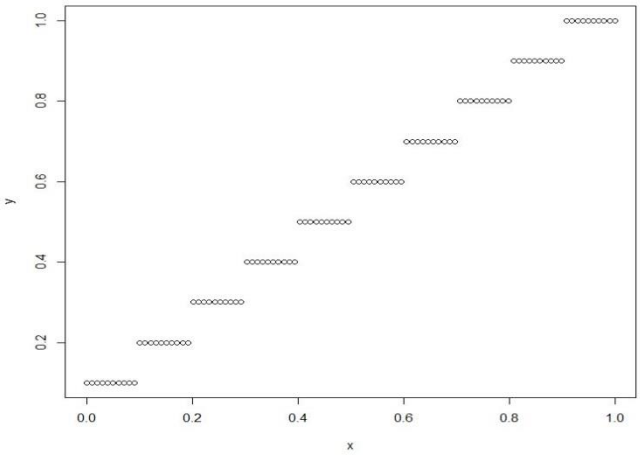
5. Dada la arquitectura de red que se muestra en la figura, con funciones de activación lineal (f(x)=x) para todas las neuronas de la red, ¿qué tipo de problemas podrían ser resueltos con éxito por dicha red? Razone su respuesta.



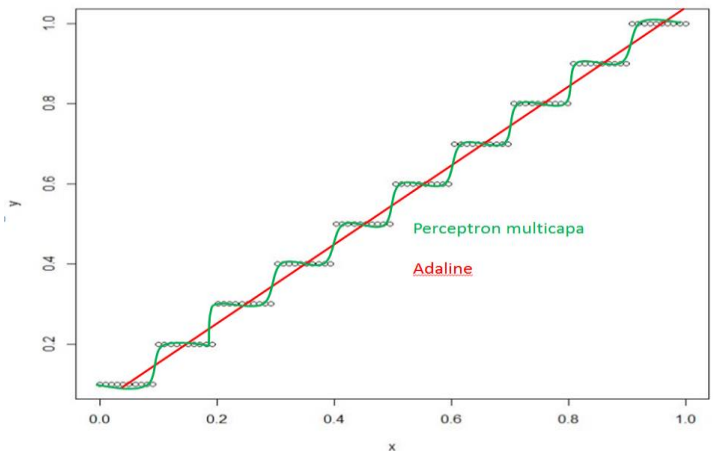
Respuesta

Al utilizar funciones de activación lineal en la capa oculta y capa de salida, la arquitectura es equivalente a un Adaline, por lo que se podrían resolver problemas de regresión lineal. También problemas de clasificación lineal, aunque para ello la salida de la red, que sería un número real, tendría que ser interpretada para clasificar en tantas clases como tenga el problema

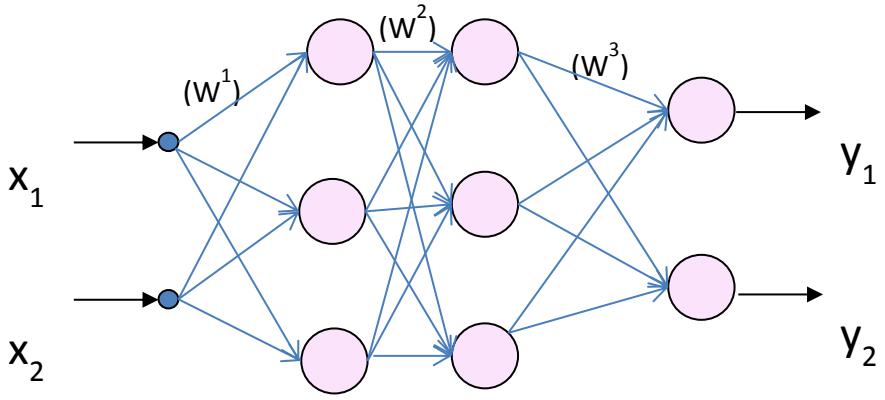
6. Disponemos de un conjunto de datos unidimensionales representados en la siguiente gráfica, donde "x" es el atributo de entrada e "y" la salida. Dibujar sobre la misma gráfica las funciones generadas por un Adaline y por un Perceptron Multicapa, cuando se entrenan con estos datos.



Respuesta



7. Dada la siguiente arquitectura del PM, escriba las ecuaciones para calcular las activaciones de la red y la ley de aprendizaje para todos los pesos de la red teniendo en cuenta que todas las neuronas tienen función de activación sigmoidal. ¿Las leyes de aprendizaje obtenidas son válidas si la función de activación en las neuronas de salida es la función lineal f(x)=x?



Respuesta:
Sea la siguiente notación:

$$W^1 = \begin{pmatrix} w_{11}^1 & w_{12}^1 & w_{13}^1 \\ w_{21}^1 & w_{22}^1 & w_{23}^1 \end{pmatrix} \quad W^2 = \begin{pmatrix} w_{11}^2 & w_{12}^2 & w_{13}^2 \\ w_{21}^2 & w_{22}^2 & w_{23}^2 \\ w_{31}^2 & w_{32}^2 & w_{33}^2 \end{pmatrix} \quad W^3 = \begin{pmatrix} w_{11}^3 & w_{12}^3 \\ w_{21}^3 & w_{22}^3 \\ w_{31}^3 & w_{32}^3 \end{pmatrix}$$

Las activaciones se obtienen aplicando la función de activación a la suma del producto de los pesos por las activaciones de las neuronas de la capa anterior más el bias de la neurona:

$$a_1^2 = f(\sum_{j=1}^{2} w_{j1}^1 a_j^1 + u_1^2) \quad a_2^2 = f(\sum_{j=1}^{2} w_{j2}^1 a_j^1 + u_2^2) \quad a_3^2 = f(\sum_{j=1}^{2} w_{j3}^1 a_j^1 + u_3^2)$$

$$a_1^3 = f(\sum_{j=1}^{3} w_{j1}^2 a_j^2 + u_1^3) \quad a_2^3 = f(\sum_{j=1}^{3} w_{j2}^2 a_j^2 + u_2^3) \quad a_3^3 = f(\sum_{j=1}^{3} w_{j3}^2 a_j^2 + u_3^3)$$

$$y_1 = f(\sum_{j=1}^{3} w_{j1}^3 a_j^3 + u_1^4) \qquad y_2 = f(\sum_{j=1}^{3} w_{j2}^3 a_j^3 + u_2^4)$$

Las leyes para la modificación de los pesos son:
Última capa (delta de la neurona a la que llega el peso por activación de la neurona de la parte el peso):

$$w_{ji}^3(n) = w_{ji}^3(n-1) + \alpha \delta_i^4(n) a_j^3(n)$$

El delta se calcula teniendo en cuenta el error medido en la salida por la derivada de la función de activación de la neurona de salida a la que llega el peso, es decir:

$$\delta_i^4 = (s_i - y_i) y_i (1 - y_i) \quad \text{i=1,2} \quad \text{j=1,2,3}$$

De la primera capa oculta a la segunda capa oculta (delta de la neurona a la que llega el peso por activación de la neurona de la parte el peso):

$$w_{kj}^2(n) = w_{kj}^2(n-1) + \alpha\delta_j^3(n)a_k^2(n)$$

En este caso, el delta se calcula teniendo en cuenta la suma de los deltas de las neuronas de la segunda capa oculta multiplicados por el peso correspondiente y la derivada de la función de activación de la neurona oculta a la que llega el peso, es decir:

$$\delta_j^3 = a_j^3(1-a_j^3)\sum_{i=1}^{2}\delta_i^4 w_{ji}^3 \qquad \text{k=1,2,3} \quad \text{j=1,2,3}$$

De la capa de entrada a la primera capa oculta (delta de la neurona a la que llega el peso por activación de la neurona de la parte el peso):

$$w_{kj}^1(n) = w_{kj}^1(n-1) + \alpha\delta_j^2(n)a_k^1(n)$$

Al igual que en el caso anterior, el delta se calcula teniendo en cuenta la suma de los deltas de las neuronas de la primera capa oculta multiplicados por el peso correspondiente y la derivada de la función de activación de la neurona oculta a la que llega el peso, es decir:

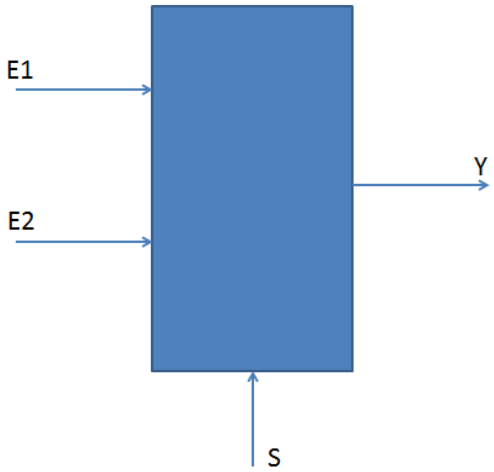
$$\delta_j^2 = a_j^2(1-a_j^2)\sum_{i=1}^{3}\delta_i^3 w_{ji}^2 \qquad \text{k=1,2} \quad \text{j=1,2,3}$$

Si la función de activación en la salida es la función lineal, las ecuaciones para modificar los pesos de la red son las mismas, con la excepción de los pesos de la segunda capa oculta a la capa de salida, para los cuales el delta es diferente, debido a que la derivada de f(x)=x es 1, por tanto:

$$w_{ji}^3(n) = w_{ji}^3(n-1) + \alpha\delta_i^4(n)a_j^3(n)$$

$$\delta_i^4 = (s_i - y_i)$$

8. En la siguiente figura se representa multiplexor donde las entradas E1, E2 pueden tomar cualquier valor real en el intervalo [-10, 10]. S es la entrada de control o selección y puede tomar los valores -1 o +1. Y es la salida del multiplexor. El multiplexor funciona de la siguiente manera: la salida Y coincide con E1 cuando el valor de S es +1 y con E2 cuando S= -1 .



Se pretende utilizar un PM simular el comportamiento del multiplexor. Se pide:

a) ¿De qué clase de problema se trata? (clasificación, regresión …)
b) Elija una posible arquitectura para resolver el problema
c) Explique el mecanismo para la obtención de los patrones de entrenamiento, validación y test
d) Indique los pasos para realizar el aprendizaje de la red
e) Indique los parámetros que intervienen en el aprendizaje

Respuesta

a) Se trata de un problema de aproximación o regresión, ya que se dispondría de un conjunto de datos cuya salida es continua y conocida. Habría que aproximarla con el menor error posible. Realmente no se dispone de los datos, pero pueden ser generados artificialmente a partir de las especificaciones del problema. El PM es adecuado para abordar el problema

b) Debido a que la salida del multiplexor Y depende de E1, E2 y de S, en la capa de entrada se deben tener tres neuronas (E1, E2, S). Capa de salida con 1 neurona para Y. Y en principio una capa oculta sería suficiente, pero el número de neuronas ocultas debe determinarse experimentalmente, realizando pruebas con diferente número de neuronas.

   Como la salida puede tomar cualquier valor en [-10,10], la función de activación de la neurona de salida sería la función lineal. Las neuronas ocultas tendrían función de activación sigmoidal.

c) Habría que generar un conjunto de datos teniendo en cuenta el funcionamiento del multiplexor. Se generan dos secuencias de números aleatorios reales entre -10 y 10, que van a representar E1 y E2. El tamaño de estas secuencias podría ser, por ejemplo de 1000 datos. Se genera una secuencia aleatoria de un número binario (-1 o 1) para S. A continuación se calcula la salida deseada Y teniendo en cuenta el funcionamiento del multiplexor: si S =1, Y=E1 y si no, Y=E2.

   Una vez obtenido el conjunto, sería conveniente normalizar las entradas en el intervalo [0,1], aleatorizarlos y extraer los tres conjuntos: entrenamiento, validación y test, con una proporción pro ejemplo de 60%, 20% y 20%, respectivamente

d) Una vez fijada la arquitectura, los pasos para el aprendizaje son:

   Paso 1: Se inicializan los pesos y umbrales (bias) de la red (valores aleatorios próximos a 0)

   Paso 2: Se presenta un patrón n de entrenamiento y se calcula la salida de la red

   Paso 3: Se evalúa el error cuadrático, e(n), cometido por la red para el patrón n.

   Paso 4: Se modifican pesos y umbrales utilizando la regla delta generalizada. Para ello se calcula en primer lugar los valores δ para todas las neuronas de la capa de salida, y a continuación los valores δ para el resto de las neuronas de la red, empezando desde la última capa oculta y retropropagando dichos valores hacia la capa de entrada

   Paso 5: Se repiten los pasos 2, 3 y 4 para todos los patrones de entrenamiento, completando así un ciclo de aprendizaje

   Paso 6: Se evalúa el error total cometido por la red. Es el error de entrenamiento.

   Paso 7: Se presentan los patrones de test (o validación), calculando únicamente la salida de la red (no se modifican los pesos) y se evalúa el error total en el conjunto de test (o validación).

   Paso 8: Se repiten los pasos 2, 3, 4, 5, 6 y 7 hasta alcanzar un mínimo del error de entrenamiento, para lo cual se realizan m ciclos de aprendizaje.

e) Razón de aprendizaje y el número de ciclos. Ambos se determinan experimentalmente y se eligen los mejores valores en base al error en conjunto de validación.

9. Las moléculas de ADN consisten en una secuencia construida a partir de cuatro elementos básicos, llamados nucleótidos o bases (Adenina, Guanina, Timina y Citosina). Una serie de complejos mecanismos bioquímicos realizan una traducción a partir del ADN para obtener como resultado las proteínas que intervienen en el metabolismo celular. Una cuestión importante a la hora de la creación de proteínas es el *tipo de transición* que contiene la secuencia de ADN. Estas transiciones puedes ser del tipo denominado Intrón/Exón o Exón/Intrón. Pudiera ocurrir también que la secuencia de ADN no contuviera ninguna transición. Las encimas encargadas de realizar este proceso detectan el tipo de transición de forma automática, pero se desconoce el criterio en base al cual se puede determinar en qué secuencias de nucleótidos existen transiciones del tipo Intrón/Exón o Exón/Intrón.

   Se pretende construir una red de neuronas para intentar identificar de forma fiable el tipo de transición que contienen las secuencias de ADN. Para ello se dispone de un conjunto de 3100 fragmentos de ADN, para los cuales es conocido si contienen transiciones Intrón/Exón, Exón/Intrón o ninguna de ellas. Cada uno de los fragmentos contiene una secuencia de 60 bases. Un ejemplo extraído de dicho conjunto es:

ACTCATCGAACTCTGCTGATAGCCAATGAGGTAATTTTCTTTATGATTCCTACAGTCTGT, Exon/Intron
TGACCTGATCTTTGCTCTCCCCCTGGCCAGTTGAGGAGGAGAACCCGGACTTCTGGAACC,
Intron/Exon
AGAACCACCTACTTCTTGGGGAGGTAGGTCTGCTTCCCTTCAACTCAGGATACAACTGCT, Ninguna
GTGGGCTGAAGCCTGGCTCTGTCCCTGCAGGGTGCCTGGTATGTGTGGAACCGCACTGAG,
Intron/Exon
AAGCACACCACGGATCTAGATGCCAGTAAAGTGAGTTCAAATATCCCACTTCTGATTTGC, Exon/Intron
AGGAAAAAGAGGAAGGGAGGGAGGGAGGAAGGAAGGAAAGAAGGAAGGAAGGGAGAGAGA,
Ninguna
siendo A=Adenina, G=Guanina, T=Timina y C=Citosina.

Se pide:

a) ¿Qué tipo de modelos de redes de neuronas de las estudiadas hasta ahora se podrían utilizar para abordar el problema?
b) Indicar el número de neuronas de entrada y salida para la red.
c) Explicar el mecanismo para la obtención de los patrones de entrenamiento y test
d) Explique un criterio para determinar a partir de la salida de la red, de qué tipo de transición se trata.
e) Suponiendo que se dispone de cierta información que dice que el tipo de transición, que en el caso de existir, está localizada entre las bases de las posiciones 30 y 31, ¿realizaría algún cambio en su planteamiento anterior? Razone su respuesta.

Respuesta

a) Problema de clasificación en principio no lineal, por tanto de las redes estudiadas hasta ahora se puede utilizar el PM. RBR también
b) Las neuronas de entrada reciben las 60 bases del fragmento, por tanto tendremos 60 neuronas de entradas. Las bases tiene que codificarse con valores numéricos, por ejemplo A=0, G=0.3, T=0.7 y C=1 (pueden elegirse otras codificaciones)
   Como se trata de un problema de clasificación en 3 clases, podemos considerar 3 salidas. En este caso, las salidas deseadas para la red son:
   Intrón/Exón =(1,0,0)

Exón/Intrón =(0,1,0)
Ninguna=(0,0,1)

c) Se extraen aleatoriamente el 70% (por ejemplo) de los datos. En este caso, al tratarse de un problema de clasificación, es necesario mantener la proporción de cada clase con respecto al conjunto original. Por tanto el mecanismo sería:
   70% de la clase Intrón/Exón
   70% de la clase Exón/Intrón
   70% de la clase Ninguna.
   Se unen y se aleatorizan. En este caso y de acuerdo a la codificación empleada no hace falta normalizar. El resto de los datos se unen y se pueden utilizar para el test.
d) Se calcula el índice para el cual se obtiene el máximo de (y_1, y_2, y_3). Si el índice 1, entonces Intrón/Exón, si es 2 Exón/Intrón y si es 3, entonces la clase será Ninguna
e) Se utilizaría solo dos entradas a la red, justo las bases de las posicione 30 y 31. El resto de las bases del fragmento sería información irrelevante.