Algoritmo de Retropropagación

Introducción

La Figura 1 presenta un ejemplo de una red neuronal típica, conocida en la literatura como perceptrón multi-capas.. La red consta de neuronas de procesamiento (círculos) y canales de flujo de información entre las neuronas llamadas interconexiones. Los rectángulos son neuronas que simplemente almacenan entradas a la red. Cada neurona de procesamiento posee una cantidad limitada de memoria y realiza un cálculo local que transforma las entradas en la salida. Este cálculo se denomina la función de activación o función transferencial de la neurona. Las funciones transferenciales pueden ser lineales o no y consisten en ecuaciones algebraicas o diferenciales.

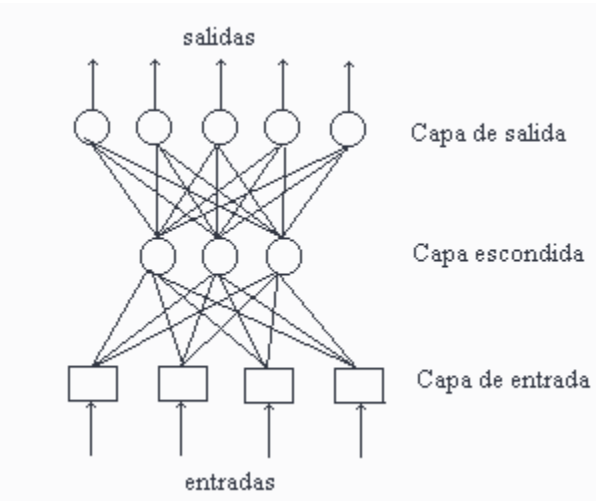

Figura 1. Ejemplo de un perceptrón con 3 capas de neuronas

En la red mostrada en la figura 1, hay 3 capas de neuronas: La capa de entrada, la escondida y la capa de salida. En algunas publicaciones la capa de entrada no se cuenta y desde este punto de vista la red de la figura 1 podrá considerarse de 2 capas. No obstante, nosotros adoptaremos la convención de que la capa de entrada se cuente como una más.

Retropropagación.

De acuerdo con un estudio reciente, existen más de 50 tipos de redes neuronales que han sido estudiadas y/o usadas en una gran diversidad de aplicaciones. Sin embargo, quizás la más extendida y utilizada para la identificación de procesos dinámicos y su control es la llamada red de retropropagación, que es la que expondremos a continuación.

Una red de retropropagación típica puede esquematizarse tal como se muestra en la figura 1. El concepto de retropropagación se refiere a un método de entrenamiento de la red y por extensión se le llama "red de retropropagación" a aquella que se entrena mediante dicho método. Es conveniente aclarar que el método o algoritmo de retropropagación no es el único existente en la actualidad, no obstante, puede afirmarse que este método y sus modificaciones han sido los más exitosos en la solución de una gran diversidad de problemas prácticos. Entre las aplicaciones de este tipo de red podemos citar:

- Reconocimiento y síntesis de voz (Elman y Zipser, 1987)
- Reconocimiento de patrones visuales o imágenes (Rumelhart y McClelland, 1986)

- Análisis de señales de sonar (Gorman y Sejnowsky, 1988)
- Aplicaciones en la defensa (Castelaz, P., 1988)
- Diagnóstico médico (Bounds, et al, 1988)
- Control de columnas de destilación y otros procesos (Birky y McAvoy, 1989)

La red de retropropagación es un tipo de red neuronal que es capaz de desarrollar una aproximación tan fina como se quiera de cualquier función no-lineal y = f(x), a partir de un conjunto de pares de ejemplo x, y. Este método es una generalización de la ley de aprendizaje por el descenso más rápido presentada por Widrow y Hoff (1960) que analizaron una red neuronal de dos capas llamada ADALINE y desarrollaron un método de descenso local según la dirección del gradiente.

El método propuesto por Widrow y Hoff fue generalizado al caso de capas múltiples con el nombre de "algoritmo de retropropagación" (Werbos, 1974; Rumelhart y McClelland, 1986). La cuestión del número de capas en una red es muy importante tal como demuestran Minsky y Papert (1972) en su texto clásico "Perceptrons". Ellos mostraron que una red de dos capas del tipo estudiado hasta entonces estaba limitada en cuanto al espectro de problemas que podría resolver y especularon sobre que el estudio de redes multicapas sería un área estéril. No obstante, la experiencia ha demostrado que esa especulación era incorrecta.

En efecto, las redes multicapas pueden dar resultados que son imposibles de alcanzar con las redes de dos capas. Como ha sido discutido por Rumelhart y McClelland (1986), la adición de capas escondidas permite al algoritmo de retropropagación desarrollar una "representación interna" del problema que puede ser vital para su solución. Parece ser que la presencia de una o más capas escondidas le da a la retropropagación nuevas dimensiones en términos de su habilidad para "aprender" la representación de una función arbitraria f(x).

Algoritmos de Cálculo.

Consideremos la red representada en forma simplificada en la figura 2 y supongamos que se dispone de un conjunto de pares de datos, x^p, y^p, p=1. 2....., N.

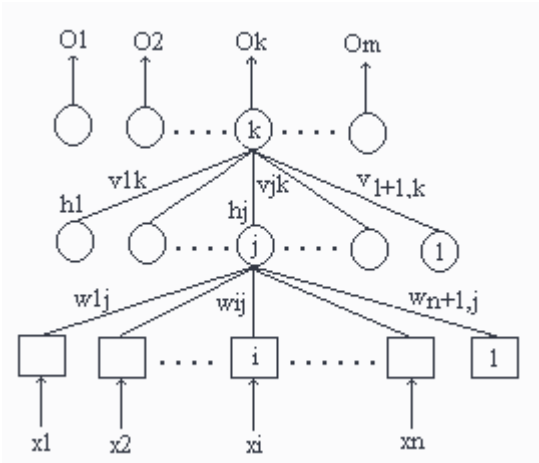

Figura 2. Esquema de una red neuronal de 3 capas

Para la neurona j de la capa escondida, definimos la función de excitación, correspondiente a la muestra p de los datos:

$$S_j^p = \sum_{i=1}^{n} w_{ij}^p x_i^p + w_{n+1,j}^p \qquad (1)$$

Es conveniente aclarar en este punto que los datos de entrada y de salida x_i, y_i. deben estar normalizados en el intervalo 0-1. Los coeficientes w_{ij}, v_{jk}, se denominan "pesos" y a la entrada n+1, cuyo valor es fijado en el valor 1, se le conoce como "bias". Este último elemento permite que la red pueda modelar sistemas en que las salidas pueden tener valores distintos de cero cuando todas las entradas son cero.

La salida de la neurona j de la capa escondida se calcula generalmente como una función no-lineal de la excitación, aunque en algunos casos particulares puede ser también una función lineal. Por el momento vamos a definir la función no lineal de salida o función de activación, como la función sigma, tanto para las neuronas de la capa escondida como para las de la capa de salida. Esta función presenta considerables ventajas dada su simplicidad, aunque es posible utilizar otras funciones como la tangente hiperbólica, la función de Gauss, etc. La forma de la función sigma o sigmoide, se representa en la figura 3.

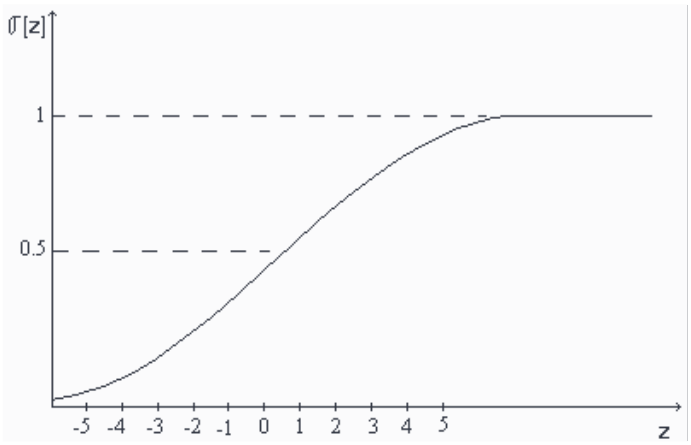

Figura 3. Forma típica de la función sigma

Entonces la salida de la neurona j de la capa escondida, se expresa:

$$h_j^p = f(S_j^p) = \frac{1}{1+\exp(-S_j^p)} \qquad (2)$$

La excitación de la neurona k de la capa de salida, se calcula en forma análoga, mediante la expresión:

$$r_k^p = \sum_{j=1}^{l} v_{jk}^p h_j^p + v_{l+1,k}^p \qquad (3)$$

y por último, la salida de la neurona k de la capa de salida se expresa como:

$$O_k^p = f(r_k^p) = \frac{1}{1+\exp(-r_k^p)} \qquad (4)$$

Definimos ahora al error en la salida k, para la muestra p como:

$$e_k{}^p = y_k{}^p - O_k{}^p \tag{5}$$

donde $y_k{}^p$ es la muestra p de la salida k. El criterio a minimizar en la muestra p se define:

$$E_p = \frac{1}{2}\sum_{k=1}^{m}(e_k{}^p)^2 = \frac{1}{2}\sum_{k=1}^{m}(y_k{}^p - O_k{}^p)^2 \tag{6}$$

o también:

$$E_p = \frac{1}{2}\sum_{k=1}^{m}(y_k{}^p - f(r_k{}^p))^2 \tag{7}$$

El proceso de entrenamiento de la red consiste entonces en presentar secuencialmente las entradas $x_i{}^p$ (i =1,2,.....,n), (p=1,.....,M), calcular las salidas de la red $O_k{}^p$ (k=1,2,.....,m) (p=1,2,.....,M), los errores $e_k{}^p$ y el criterio E_p y aplicar algún procedimiento de minimización de la función E_p con respecto a los coeficientes de peso w_{ij} y v_{jk} de manera que estos se vayan aproximando paulatinamente a los valores que garantizan un error mínimo entre las salidas de la red O_k y los datos de salida y_k.

Este procedimiento se repite tantas veces como sea necesario, es decir, los vectores de datos x^p e y^p se utilizan reiteradamente hasta tanto el error en cada salida y el criterio E_p, para p = 1,2,....N, se encuentren por debajo del límite prefijado. Una vez lograda esta condición, se dice que la red está entrenada, lo que significa que ella será capaz de reproducir la función y = f(x) con suficiente exactitud si los datos de entrenamiento han sido bien seleccionados y suficientes. A cada ciclo de uso de los datos de entrenamiento suele denominársele "época" en el argot de las redes neuronales.

Para minimizar la función E_p con respecto a los coeficientes de peso w_{ij} y v_{jk} vamos a utilizar el método del "descenso más rápido" (steepest descent) que consiste en moverse siempre en la dirección del negativo del gradiente de la función E_p con respecto a los coeficientes w_{ij} y v_{jk}.

El gradiente de la función E_p es un vector multidimensional cuyas componentes son las derivadas parciales $\frac{\partial E_p}{\partial v_{jk}{}^p}$ y $\frac{\partial E_p}{\partial w_{ij}{}^p}$ o sea:

$$\nabla E_p = \begin{bmatrix} \dfrac{\partial E_p}{\partial v_{jk}{}^p} \\ \dfrac{\partial E_p}{\partial w_{ij}{}^p} \end{bmatrix} \tag{8}$$

Vamos a calcular en primer lugar las derivadas parciales con respecto a las neuronas de la capa de salida, lo que resulta más simple:

$$\frac{\partial E_p}{\partial v_{jk}{}^p} = \frac{\partial E_p}{\partial O_k{}^p}\frac{\partial O_k{}^p}{\partial r_k{}^p}\frac{\partial r_k{}^p}{\partial v_{jk}{}^p} \tag{9}$$

$$\frac{\partial E_p}{\partial O_k{}^p} = \frac{\partial\left(\frac{1}{2}\sum_{k=1}^{m}(y_k{}^p - O_k{}^p)^2\right)}{\partial O_k{}^p} = -(y_k{}^p - O_k{}^p) \tag{10}$$

$$\frac{\partial O_k{}^p}{\partial r_k{}^p} = \frac{\partial\left(\frac{1}{1+e^{-r_k{}^p}}\right)}{\partial r_k{}^p} = \frac{e^{-r_k{}^p}}{(1+e^{-r_k{}^p})^2} = O_k{}^p(1-O_k{}^p) \tag{11}$$

$$\frac{\partial r_k{}^p}{\partial v_{jk}{}^p} = \frac{\partial\left(\sum_{j=1}^{l} v_{jk}{}^p h_j{}^p + v_{l+1,k}{}^p\right)}{\partial v_{jk}{}^p} = h_j^p \tag{12}$$

Sustituyendo (10), (11) y (12) en (9), tenemos que:

$$\frac{\partial E_p}{\partial v_{jk}{}^p} = -(y_k{}^p - O_k{}^p)\,O_k{}^p(1-O_k{}^p)\,h_j{}^p \tag{13}$$

Definimos ahora:

$$\delta_k{}^p = (y_k{}^p - O_k{}^p)O_k{}^p(1-O_k{}^p) \tag{14}$$

y sustituyendo (14) en (13), tenemos entonces:

$$\frac{\partial E_p}{\partial v_{jk}{}^p} = -\delta_k{}^p h_j{}^p \tag{15}$$

La segunda parte del gradiente de E_p requiere del cálculo de sus derivadas parciales con respecto a los coeficientes de peso w_{ij} que conectan la capa de entrada con la capa escondida. A continuación procedemos a realizar dicho cálculo.

En primer lugar, retomemos la expresión que define al criterio E_p:

$$E_p = \frac{1}{2}\sum_{k=1}^{m}(y_k{}^p - f(r_k{}^p))^2 \tag{16}$$

y teniendo en cuenta la definición de $r_k{}^p$ según (3), podemos escribir:

$$E_p = \frac{1}{2}\sum_{k=1}^{m}(y_k{}^p - f(\sum_{j=1}^{l} v_{jk}{}^p h_j{}^p + v_{l+1,k}{}^p))^2 \tag{17}$$

Sabemos además, que de acuerdo con (1) y (2):

$$h_j^p = f(\sum_{i=1}^{n} w_{ij}^p x_i^p + w_{n+1,j}^p) \tag{18}$$

Aplicando ahora la regla de la cadena de la derivación y teniendo en cuenta las relaciones (16)-(18), se llega a:

$$\frac{\partial E_p}{\partial w_{ij}^p} = \left(\sum_{k=1}^{m} \left(\frac{\partial E_p}{\partial O_k^p} \frac{\partial O_k^p}{\partial r_k^p} \frac{\partial r_k^p}{\partial h_j^p} \right) \right) \frac{\partial h_j^p}{\partial S_j^p} \frac{\partial S_j^p}{\partial w_{ij}^p} \tag{19}$$

$$= -\left(\sum_{k=1}^{m} (y_k^p - O_k^p) O_k^p (1 - O_k^p) v_{jk}^p \right) h_j^p (1 - h_j^p) x_i^p \tag{20}$$

$$\frac{\partial E_p}{\partial w_{ij}^p} = -\left(\sum_{k=1}^{m} \delta_k^p v_{jk}^p \right) h_j^p (1 - h_j^p) x_i^p \tag{21}$$

Definimos:

$$\Delta_j^p = \left(\sum_{k=1}^{m} \delta_k^p v_{jk}^p \right) h_j^p (1 - h_j^p) \tag{22}$$

y sustituyendo (22) en (21), tenemos que:

$$\frac{\partial E_p}{\partial w_{ij}^p} = -\Delta_j^p x_i^p \tag{23}$$

Teniendo en cuenta las expresiones (15) y (23), resulta:

$$\nabla E_p = -\begin{bmatrix} \delta_k^p h_j^p \\ \Delta_j^p x_i^p \end{bmatrix} \tag{24}$$

El algoritmo de retropropagación aplicado al entrenamiento de las neuronas de una red de 3 capas, consiste en aplicar el método del descenso más rápido, o en otras palabras, moverse en la dirección del negativo del gradiente, dando un paso de actualización de los valores de los coeficientes de peso v_{jk}^p y w_{ij}^p con cada par de vectores de entrada y salida x_p, y_p, o sea:

$$\begin{bmatrix} v_{jk}^p \\ w_{ij}^p \end{bmatrix} = \begin{bmatrix} v_{jk}^{p-1} \\ w_{ij}^{p-1} \end{bmatrix} - \eta \nabla E_p \tag{25}$$

y entonces:

$$\begin{bmatrix} v_{jk}^p \\ w_{ij}^p \end{bmatrix} = \begin{bmatrix} v_{jk}^{p-1} \\ w_{ij}^{p-1} \end{bmatrix} + \eta \begin{bmatrix} \delta_k^p h_j^p \\ \Delta_j^p x_i^p \end{bmatrix} \tag{26}$$

De donde son evidentes las relaciones:

$$v_{jk}^{p} = v_{jk}^{p-1} + \eta \delta_k^{p} h_j^{p} \tag{27}$$

$$w_{ij}^{p} = w_{ij}^{p-1} + \eta \Delta_j^{p} x_i^{p} \tag{28}$$

en las que η es una constante que se denomina **coeficiente de aprendizaje**.

Se ha demostrado experimentalmente que una forma de acelerar la convergencia en el proceso de aprendizaje es incluir un término proporcional al cambio realizado en los coeficientes de peso en la muestra anterior. Entonces definimos:

$$I_{v_{jk}}^{p} = \eta \delta_k^{p-1} h_j^{p-1} + \alpha I_{v_{jk}}^{p-1} \tag{29}$$

$$I_{w_{ij}}^{p} = \eta \Delta_j^{p-1} x_i^{p-1} + \alpha I_{w_{ij}}^{p-1} \tag{30}$$

y:

$$v_{jk}^{p} = v_{jk}^{p-1} + \eta \delta_k^{p} h_j^{p} + \alpha I_{v_{jk}}^{p} \tag{31}$$

$$w_{ij}^{p} = w_{ij}^{p-1} + \eta \Delta_j^{p} x_i^{p} + \alpha I_{w_{ij}}^{p} \tag{32}$$

donde el coeficiente α se denomina **coeficiente de momentum** y sus valores son generalmente menores que los de η.

La utilización del coeficiente de momentum en las ecuaciones de actualización de los coeficientes de peso tiene también el efecto benéfico de disminuir considerablemente la probabilidad de que el algoritmo de minimización quede atrapado en un mínimo local.

Los valores específicos de los coeficientes η y α deben escogerse experimentalmente; como valores típicos puede partirse de los valores $\eta = 0.5$ y $\alpha = 0.1$. Hemos comprobado experimentalmente que valores mayores de η aunque pueden resultar en una convergencia más rápida, pueden también conducir a una aproximación oscilatoria en la convergencia de los coeficientes v_{jk} y w_{ij}. Se ha intentado también, con relativo éxito, el uso de coeficientes de aprendizaje y de momentum variables, partiendo de valores menores hasta lograr la aproximación a un entorno de los valores estacionarios de v_{jk} y w_{ij} y después ir sucesivamente aumentando a η y α.

Es importante destacar que la estrategia de entrenamiento de una red neuronal mediante el algoritmo de retropropagación explicado hasta aquí, tiene algunas particularidades que lo distinguen de un problema convencional de optimización. Si se dispone de una muestra de M pares de valores x^p, y^p de entrada y salida, estos datos serán presentados secuencialmente a la red, calculándose una actualización de los coeficientes de peso w_{ij}, v_{jk} para cada par de datos, hasta agotarlos todos, es decir, para p = 1, 2,...., M. Es muy frecuente que un solo ciclo o época no sea suficiente para lograr un valor aceptablemente pequeño del criterio E_p para todo p y entonces es necesario reciclar los datos, o en otras palabras, comenzar una nueva época. Dependiendo del tamaño de la muestra de entrenamiento, se necesitarán más o menos épocas, pudiendo ser su número, desde unas decenas, hasta cientos o miles.

El método de la retropropagación basado en el algoritmo del descenso más rápido (Fletcher y Powell, 1963), tiene la virtud de su simplicidad, no obstante, como es conocido, ese algoritmo puede resultar en algunos casos, demasiado lento y además, no siempre garantiza la convergencia a un mínimo absoluto, es decir, puede empantanarse en un mínimo local.

Existen otros métodos, como los llamados del tipo Quasi-Newton (Powell, 1970) que garantizan la convergencia a un mínimo absoluto y además convergen más rápido, siendo característico de ellos que en cada etapa se calcula un tamaño del paso óptimo, para lo que se requiere el cálculo o estimación de la matriz Hessiana, formada por las segundas derivadas del criterio E_p con respecto a w_{ij} y v_{jk}. No obstante, para redes de dimensiones grandes, el número de coeficientes de peso puede ser hasta de varios cientos o miles, lo cual hace que la dimensión del problema de optimización crezca intolerablemente.

Adaptado de: A Aguado, Temas de Identificación y Control Adaptable. ICIMAF, CUBA.
ISBN 959-7056-11-9