**CAPÍTULO 2**

**2. PRESENTACIÓN DE LA TEORÍA PARA LAS REDES NEURONALES**

**2.1. Las Redes Neuronales y la Modelación**

El estudio de las Redes Neuronales comenzó en la década de 1960, generando primero un gran interés que sin embargo decayó al poco tiempo y que dio lugar luego al escepticismo en la comunidad científica. Su lento despegue se debió principalmente al incipiente desarrollo de la Computación en esa época, rama fuertemente ligada a las Redes Neuronales; y a la lenta aparición de modelos matemáticos precisos.

En los tiempos actuales, con el creciente desarrollo de las computadoras, las Redes Neuronales han tenido un desarrollo muy importante, así como también han encontrado aplicaciones en diferentes campos del conocimiento humano, entre los que podemos citar:

El Aprendizaje artificial.-

* Auto programación de una computadora para realizar tareas.- se tiene un conjunto de datos experimentales para la red, y mediante la retroalimentación, la red se ajusta a sí misma para la realización de determinada tarea.
* Optimización.- en el área de la investigación de operaciones, dado un conjunto de restricciones y una función objetivo, encontrar la solución óptima. Como ejemplo tenemos el problema del agente viajero.
* El problema de clasificación.- asignar a los elementos de un conjunto un grupo. Como ejemplo podemos citar los análisis de conglomerados y el reconocimiento de letras escritas a mano.

En las ciencias cognoscitivas.-

* Modelamiento de niveles altos de razonamiento
* Modelamiento de niveles bajos de razonamiento

Neurobiología.-

* En la modelación del cerebro humano para entender su funcionamiento.

Matemáticas.-

* En la estadística no paramétrica
* En los modelos de regresión y de predicción.

Filosofía./Psicología.-

* Resolviendo problemas filosóficos profundos como por ejemplo: ¿Puede el comportamiento humano ser explicado únicamente a través de símbolos o se necesita de un modelo de bajo nivel como el de las Redes Neuronales?

**2.1.1. Áreas específicas de aplicación**

Entre las áreas específicas en donde se están utilizando con éxito las redes neuronales tenemos:

* El Procesamiento de señales.- reducción de ruidos, eliminación de resonancias y otras aplicaciones
* La Robótica.- para el reconocimiento visual. De hecho esta fue una de las principales aplicaciones prácticas que se le dio a las redes neuronales. El experimento consistía en un automóvil con una cámara incorporada, y con foquitos a los lados de una carretera. El automóvil debía ser capaz de auto-conducirse sin salirse de la vía.
* La Reproducción del Habla
* El Reconocimiento del Habla
* La Visión 3D, en el reconocimiento de caras y fronteras.
* Los negocios.- en la concesión de créditos, y en tasas de seguros. En la construcción de programas de decisión que tengan un alto grado de concordancia con las decisiones de los profesionales (sistemas expertos).
* Las finanzas.- en la predicción del mercado bursátil.
* Para la compresión de datos.

 **2.1.2. Modelamiento del cerebro humano en una red neuronal.-**

El cerebro humano esta formado por alrededor de un billón de células nerviosas que se conocen como neuronas, éstas a su vez se conectan con otras neuronas a través de lo que conocemos como sinápsis. Así los elementos principales de toda neurona son: el cuerpo de la célula, las dendritas (o entradas), y los axones (o salidas). Partiendo de esta forma física y funcional de las células que componen al cerebro humano creamos un modelo matemático que las imite.

El marco que define a esta red está conformado por los siguientes elementos.

- Un conjunto de unidades de proceso (los nodos propiamente dichos).

- Un conjunto de “estados”  de activación para cada unidad .

- Una conexión entre las unidades, que denominaremos , que representa al peso o conexión entre la unidad  y la unidad . Esta determina la magnitud del efecto entre unidades, y es primordial en el proceso de ajuste.

- Una regla de propagación, la cual determina la entrada efectiva  de una unidad de entre sus entradas externas.

- Una función de activación , que depende de la entrada efectiva  y del estado de activación actual . A esta función la llamaremos la renovación.

- Una entrada externa  para cada unidad.

- Un método para reunir información (una regla de aprendizaje).

- Un ambiente o entorno dentro del cual el sistema debe operar. El cual le proporcionará las señales de entrada y si es posible las señales de error.

Este es el marco general de las redes neuronales, ahora veamos como ajustamos esto a nuestro interés, es decir a pronosticar los valores en una serie temporal.

**2.2. Redes de una sola capa**

Las redes neuronales pueden ser caracterizadas de acuerdo al número de capas que tienen. Las unidades de una red se dividen en tres tipos, las unidades de entrada, las unidades de salida y las unidades escondidas o internas. Todas las unidades, a excepción de las de entrada, reciben ingresos de una o más unidades de un nivel anterior; así mismo todas las unidades, a excepción de las de salida, son ingresos de una o más unidades de un nivel posterior. Así, según el número de niveles de una red, se la denomina como de una, dos, o más capas. Por convención no se cuenta al nivel conformado por unidades de entrada, es decir, a una red conformada por dos o más unidades de entradas y una o más unidades de salida la clasificaremos como de una sola capa. En la figura 2.1 podemos observar un esquema sencillo de una neurona.

Figura 2.1

Esquema sencillo de la modelación de una neurona



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos.

Elaboración: Andrés Abad

Veremos primero el caso más sencillo de este tipo de redes, llamado el perceptrón. El perceptrón está conformado por 2 unidades de entrada y una de salida.

La información que ingresa a la unidad de salida será una suma ponderada de las unidades de ingresos , con los pesos  para . Además incluimos un término constante , al que por simplicidad lo consideraremos como el peso de una variable ficticia . Con la notación utilizada, el ingreso  a la unidad de salida  será igual a . Por el momento no especificaremos una función de activación  de la unidad de salida.

Una vez definida la estructura de esta red nuestro interés se centrará en la determinación de los pesos . Esto es lo que se llama el aprendizaje de la red, y se logra utilizando una regla de aprendizaje, y utilizando los datos de entrenamiento (es decir los , donde  es el conjunto de datos de aprendizaje).

**2.3. Entrenamiento de redes neuronales artificiales**

Una red neuronal debe ser configurada de tal manera que con la aplicación de un conjunto de datos de entrada se produzca un deseado conjunto de salidas. Existen varios métodos para definir los valores iniciales del conjunto de pesos de la red. Una forma de definir estos valores es obtener un conocimiento *a priori*. Otra forma es “entrenar” a la red alimentándola con datos de aprendizaje y permitiéndole modificar sus pesos de acuerdo a lo que llamamos una regla de aprendizaje.

 **2.3.1.** **Tipos de aprendizaje**

Podemos categorizar a los tipos de aprendizaje en dos grupos. Estos son:

**2.3.1.1.** **Aprendizaje Supervisado o asociativo.-** es cuando la red es entrenada a través del empleo de entradas junto con sus respectivas salidas. Estos pares pueden ser provistos por un “juez externo”, o por algún sistema contenido en la propia red, caso en el que se la denomina auto-supervisada.

* + - 1. **Aprendizaje no Supervisado o auto-organizado.-** este se tiene cuando una unidad de salida es entrenada para responder a conglomerados o clusters, dentro del conjunto de datos de entrada. En este tipo de aprendizaje el sistema está listo a descubrir estadísticamente algunas características inherentes en la población de entrada. A diferencia del método supervisado de aprendizaje, aquí no existe un conocimiento *a priori*, sino que el sistema debe desarrollar sus propios criterios para la clasificación de sus entradas.

En la actualidad existen algunas propuestas para esta regla de aprendizaje, algunas muy exóticas. Como introducción veremos dos, la regla de aprendizaje simple, o del perceptrón, y la regla de los mínimos errores cuadráticos, MEC; también llamada la regla delta.

Ambas reglas parten de una misma idea, la de ajustes iterativos de los pesos, es decir . Lo que se hace es incrementar (o disminuir), en cada entrenamiento una cantidad  al valor del peso anterior. La diferencia entre los dos métodos que estudiaremos radica en la cantidad .

**2.3.2.** **Regla de aprendizaje del perceptrón**

Esta regla utiliza como función de activación a la función sgn, es decir a la función signo, es decir  si  y  si . Por lo tanto el estado  de la unidad de salida después de recibir los ingresos de las unidades de entrada será 1 ó -1, y estará dado por . El  que propone esta regla de aprendizaje es sencillamente , donde  es el valor deseado cuando se ingresan los valores de entrenamiento. Dado que  toma únicamente los valores de 1 o -1, lo que estamos definiendo es simplemente un incremento o decrecimiento directamente proporcional al valor de la entrada . Si el signo es positivo estaremos hablando de un “estímulo”, y si es negativo diremos que se trata de una “inhibición”. El algoritmo para esta regla de aprendizaje es:

1.- Asignar valores aleatorios a los pesos de la red. Generalmente valores pequeños.

2.- Mientras no se alcance un nivel deseado de predicción repetir:

- Tomar un dato de entrenamiento  y obtener la salida .

- Si entonces .

3.- FIN

Debe notarse que este tipo de algoritmos son muy buenos en funciones de clasificación, en los cuales si la salida es 1 pertenece a cierta clase y si es -1 a otra clase, como en las funciones booleanas. Veremos ahora la regla delta, cuya aplicación se halla un poco menos restringida, ya que no es necesario utilizar la función signo como la función de activación.

**2.3.3. Regla de aprendizaje MEC o delta.-**

Como ya dijimos esta segunda regla de aprendizaje utiliza el mismo principio que la anterior, la diferencia se halla en el término , y en que no se necesita que la salida sea booleana. Utilizaremos una medida de error, la que definiremos como , donde es el error cuando se utiliza el dato de entrenamiento ,  es el resultado deseado, y  es el resultado obtenido. La medida que buscamos será , y como es de esperarse lo que busca la regla delta es encontrar los  que minimicen este error cuadrático.

El  para esta regla de aprendizaje es , donde  representa una tasa de aprendizaje (usualmente pequeña, por ejemplo ). La idea es definir un vector gradiente , y encontrar la dirección en la que este vector hace menor al error . Para calcular estas derivadas parciales hacemos lo siguiente: , por la regla de la cadena. Dada la linealidad de  tenemos que , ahora tenemos que , finalmente tenemos que , donde . El algoritmo es exactamente el mismo, la única diferencia es el cálculo de los .

Existe una generalización del método de aprendizaje delta, para redes neuronales de dos o más capas. Esta técnica es conocida como Backpropagation.

**2.4. Redes de dos Capas**

**2.4.1. Método de Backpropagation.-**

Una red de una sola capa tiene severas restricciones en cuanto al conjunto de funciones que puede ajustar, como por ejemplo el problema del O excluyente (XOR). En general, todos aquellos problemas en donde, graficados en el plano, los datos que pertenecen a dos grupos distintos no puedan ser separados por un línea recta.

En 1969 Minsky y Papert, mostraron que una red de dos capas con alimentación hacia delante puede sobreponerse a muchas restricciones, pero no presentaron una solución al problema de cómo ajustar los pesos de entrada de las unidades escondidas. Una solución a esta dificultad fue presentada por Rumelhart, Hinton y Williams en 1986. La idea central detrás de esta solución es que los errores de las unidades escondidas (1era capa), son determinados por propagación hacia atrás de los errores de las unidades de salida. Por esta razón el método es generalmente llamado regla de aprendizaje Back-propagation. Esta regla también puede ser considerada como una generalización de la regla delta para funciones de activación no lineales, y redes de multicapas.

 **2.4.2. Redes de Multicapas con alimentación hacia delante**

Una red de alimentación hacia delante tiene una estructura de capas. Cada capa consiste en unidades que reciben sus entradas de unidades directamente detrás de éstas y envían sus salidas a unidas en capas directamente adelante. No hay conexiones dentro de una capa. Las unidades  de entrada alimentan a la primera capa de  unidades escondidas. Las unidades de entrada son simplemente unidades de “ventilación”; ningún proceso se lleva a cabo en estas unidades. La activación de las unidades escondidas es una función  de las entradas con sus respectivos pesos y una constante, según la ecuación:

****

Las salidas de las unidades escondidas son distribuidas hacia la siguiente capa de  unidades escondidas, y así sucesivamente hasta la última capa de unidades de salidas. En la figura 2.2, a continuación podemos ver un esquema de una red de multicapas.

Figura 2.2

Diseño de la red multicapa



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaborado: Andrés Abad Robalino

Aunque el método back-propagation puede ser aplicado a redes con cualquier número de capas, se ha demostrado que una sola capa de unidades escondidas es suficiente para aproximar con una precisión arbitraria, cualquier función que tenga un número finito de discontinuidades. Siempre que la red tenga funciones de activación no lineales en las capas escondidas. Este resultado es conocido como el teorema de aproximación universal. En la mayoría de las aplicaciones de redes con alimentación hacia delante y con una capa escondida simple, se utiliza una función de activación sigmoide para las unidades.

**2.4.3.** **La regla delta generalizada**

Para utilizar unidades que tienen una función de activación no lineal, debemos generalizar la regla delta, que presentamos en este mismo capítulo. Ya no usaremos funciones de activación lineales sino un conjunto de funciones de activación no lineales. Esta función de activación será una función diferenciable para el total de las entradas, dada por

 (1)

en donde

. (2)

Para plantear la correcta generalización de la función delta, como fue presentada en la sección previa, debemos fijar

 (3)

La medida de error  es definida como el error cuadrático total para el conjunto p de entradas y de sus correspondientes salidas:

 (4)

Donde es la salida deseada para la unidad cuando el conjunto p es ingresado. Luego fijaremos  como la suma de errores cuadráticos. Podemos escribir

. (5)

Por la ecuación 2 vemos que el segundo factor es

 (6)

Si definimos

 (7)

Conseguiríamos una regla actualizada que es equivalente a la regla delta descrita previamente, resultando esto en un descenso por el gradiente sobre la superficie de error. Si hacemos que los pesos se reajusten de acuerdo a:

 (8)

Aquí interpretaremos lo que  debería ser para cada unidad k en la red. El resultado interesante, que conseguiremos ahora, es que existe un cálculo recursivo simple para estos  que puede ser interpretado como la propagación de las señales de error hacia atrás a lo largo de la red.

 Para calcular  aplicaremos la regla de la cadena. Escribiremos esta derivada parcial como el producto de dos factores, un factor que refleje los cambios en el error como una función de las salidas de las unidades y otro que refleje el cambio en las salidas como una función de los cambios de las entradas. Tendremos

 (9)

Calculando el segundo factor, por la ecuación 1, vemos que

 (10)

es simplemente la derivada de la función de alisamiento F, para la ésima unidad, evaluada en las entradas . Para calcular el primer factor de la ecuación 9, consideraremos dos casos. Primero, asumiremos que la unidad  es una unidad de salida de la red. En este caso, se sigue de la definición de que

 (11)

El cuál es el mismo resultado que obtuvimos con la función delta estándar. Substituyendo esto y la ecuación 10 en la ecuación 9, obtenemos

 (12)

para cualquier unidad  de salida. Segundo, si es que  no es una unidad de salida, sino una unidad escondida , realmente conocemos todavía la contribución de la unidad al error de salida de la red. Sin embargo, la medida del error puede ser escrita como una función de las entradas de las capas escondidas hacia las capas de salida;  y utilizamos la regla de la cadena para escribir

 (13)

 Substituyendo esto en la ecuación 9 obtenemos

 (14)

Las ecuaciones 12 y 14 proporcionan un procedimiento recursivo para calcular el  para todas las unidades de la red, que son usadas después para calcular los cambios de los pesos de acuerdo a la ecuación 8. Este procedimiento constituye la generalización de la regla delta para una red multicapas con alimentación hacia adelante de unidades no lineales.

**2.4.3.1. Explicación de Back-propagation**

La ecuación que acabamos de obtener podrá ser matemáticamente correcta, pero, ¿Qué es lo que realmente significa? ¿Hay alguna otra forma de comprender el método back-propagation que recitando las ecuaciones necesarias?

 La respuesta es, por supuesto, sí. De hecho, todo el proceso de back-propagation es intuitivamente muy claro. Cuando un grupo de aprendizaje es utilizado, los valores de activación son propagados a las unidades de salida, y las salidas producidas son comparadas con las deseadas, generalmente terminamos con un error en cada una de las unidades de salida. Llamaremos a este error para una unidad de salida particular . Debemos llevar hacia cero.

El método más simple para hacer esto es cambiando los pesos. Lo cual se hace de tal manera, que la próxima vez, el error será cero para este grupo en particular. Sabemos de la regla delta, que para poder reducir el error a cero, debemos adaptar los pesos respectivos de acuerdo a

 (15)

Ese es el paso uno, pero esto no es suficiente. Cuando solo aplicamos esta regla, los pesos de las unidades de entrada a las escondidas nunca son cambiados, y no alcanzamos a abarcar en su cabalidad el poder de representación de la red, como es requerido por el Teorema de Aproximación Universal. Con el fin de adaptar los pesos de unidades de entrada a las unidades escondidas, de nuevo aplicaremos la regla delta. En este caso, no tenemos un valor para de las unidades escondidas. Esto es resuelto por la regla de la cadena, la cual distribuye el error de una unidad de salida a todas las unidades escondidas que se conectan a ella. De manera diferente, una unidad escondida recibe un de cada unidad de salida igual al delta de esa unidad de salida, multiplicada por el peso de la conexión entre esas dos unidades. En símbolos: . Solo falta la función de activación de las unidades escondidas; tiene que ser aplicada al , antes que el proceso de back-propagation pueda continuar.

**2.4.3.2. Trabajando con Back-Propagation**

La aplicación de la regla delta generalizada involucra dos fases: durante la primera fase, la entrada es presentada y propagada hacia delante, a través de la red para poder calcular los valores de salida  para cada unidad de salida. Esta salida es comparada con su respectiva salida deseada , resultando en una señal de error  para cada unidad de salida. La segunda fase comprende un recorrido por la red hacia atrás durante el cual, la señal de error es pasada a cada unidad de la red y luego son calculados los cambios apropiados en los pesos.

 **2.4.3.3. Ajuste de pesos con una función sigmoide de activación.-**

 Los resultados previos pueden ser resumidos en tres ecuaciones:

* El peso de una conexión es ajustada por una cantidad proporcional al producto de una señal de error , en la unidad k que recibe la entrada y la salida de la unidad j, enviando esta señal por la conexión:

 (16)

* Si la unidad es una unidad de salida, la señal de error está dada por:

 (17)

 Si tomamos como función de activación F, a la función sigmoide, definida así:

 (18)

En este caso, la derivada es igual a







 (19)

 La señal de error para una unidad de salida pueda ser escrita como

 (20)

* La señal de error para una unidad escondida es determinada recursivamente en términos de las señales de error de las unidades a las que se conecta directamente, y de los pesos de esas conexiones. Para el caso particular de la función sigmoide de activación, tendremos:

 (21)

**2.4.3.4. Aprendizaje por datos de entrenamiento.-**

El método de back-propagation ejecuta un descenso por el gradiente en la superficie del error total siempre que los pesos sean ajustados después de que se haya utilizado el conjunto completo de datos de entrenamiento. Lo más común es que la regla de aprendizaje sea aplicada a cada dato de entrenamiento por separado, es decir, se utiliza un dato p,  es calculado, y los pesos son adaptados (p=1,2,…,P). Existen resultados empíricos que indican que esto produce una convergencia más rápida. Es de notar que se deberá tener cuidado con el orden en el cuál los datos de entrenamiento son utilizados. Por ejemplo, cuando se usa la misma secuencia una y otra vez, la red se puede enfocar más en los primeros datos. Este problema se puede superar fácilmente permutando el conjunto de datos de entrenamiento.