**chapter**  **1:**  **overvIe**w**w**  **of**  **neural**  **networks**

* Comprensión de las redes neuronales biológicas
* Cómo se construye una red neuronal artificial
* Usos apropiados para redes neuronales

Las computadoras pueden realizar muchas operaciones considerablemente más rápido que un ser humano. Sin embargo, más rápido no siempre es mejor para la resolución de problemas. Hay muchas tareas para las que la computadora se queda considerablemente por debajo de su contraparte humana. Hay numerosos exámenes ples de esto. Por ejemplo, dadas dos fotos, un niño en edad preescolar puede decir la diferencia entre un gato y un perro. Sin embargo, esta misma tarea simple es extremadamente difícil para las computadoras de hoy en día.

El objetivo de este libro es enseñar al lector cómo construir redes neuronales, ya que el lenguaje de programación Java. Al igual que con cualquier tecnología, es tan importante saber cuándo no utilizar las redes neuronales como entender cuándo deben ser utilizadas. Este capítulo proporciona una introducción a los usos apropiados de las redes neuronales, y explica qué requisitos de programación son propicios para su uso.

Este capítulo comienza con una breve introducción a la estructura de las redes neuronales. Esta discusión proporciona una visión general de la arquitectura de red neuronal y explica cómo se construye una red neuronal típica. Después de esta introducción hay una discusión sobre cómo se entrena una red neuronal. Por último, se proporciona una visión general de la tarea definitiva, validando una red neuronal entrenada.

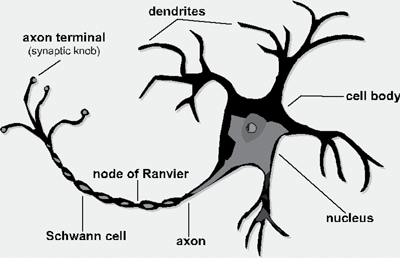
## ¿Cómo se construye una red neuronal biológica?

El término red neuronal, como se utiliza normalmente, es en realidad un nombre equivocado. Computadores intentan simular redes neuronales biológicas mediante la implementación de redes neuronales artificiales. Sin embargo, la mayoría de las publicaciones utilizan el término "red neuronal", en lugar de "*artificial neural network*" (ANN). Este libro sigue su ejemplo. A menos que el término "red neuronal" tenga el prefijo explícito con los términos "biológico" o "artificial", puede asumir que el término "red neuronal artificial" está destinado. Para explorar esta distinción, primero se le mostrará la estructura de una red neuronal biológica.

Para construir una computadora capaz de "pensamiento humano", los investigadores han utilizado el único modelo de trabajo que tienen disponible: el cerebro humano. Sin embargo, el cerebro humano en su conjunto es demasiado complejo para modelar. Más bien, se estudian las células individuales que componen el cerebro humano. En el nivel más básico, el cerebro humano se compone principalmente de células de neuronas. Son los bloques básicos del cerebro humano. Las redes neuronales artificiales intentan simular el comportamiento de estas células.

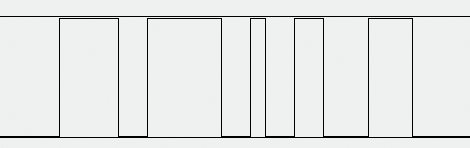
Una célula de neurona, como se ve en la Figura 1.1, acepta señales de dendritas. Cuando una neurona acepta una señal, esa neurona se puede disparar. Cuando una neurona se dispara, se transmite una señal sobre el axón de la neurona. En última instancia, la señal dejará la neurona mientras viaja a los terminales de axón. A continuación, la señal se transmite a otras neuronas o nervios.

### Figura 1.1: Una célula de neurona.



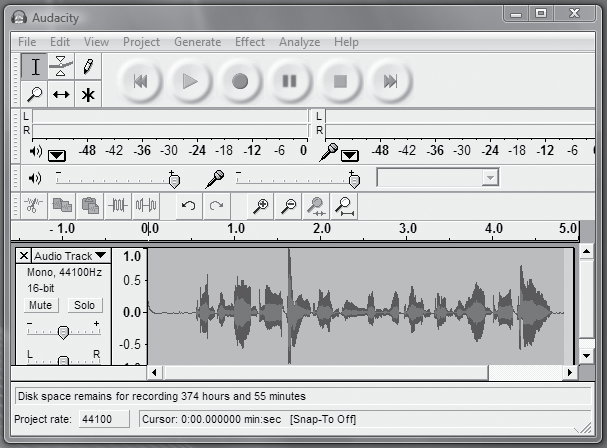
Esta señal, transmitida por la neurona, es una señal analógica. La mayoría de los computadores modernos son máquinas digitales, y por lo tanto, requieren una señal digital. Un ordenador digital procesa la información como desactivada o encendida, utilizando los dígitos binarios cero y uno, respectivamente. La presencia de una señal eléctrica se indica con un valor de uno, mientras que la ausencia de una señal eléctrica se indica con un valor de cero. La Figura 1.2 muestra una señal digital.

### Figura 1.2: Una señal digital.



Algunos de los primeros computers eran analógicos, más que digitales. Una computadora analógica utiliza una gama mucho más amplia de valores que cero y uno. Este rango más amplio se logra aumentando o disminuyendo la tensión de la señal. La Figura 1.3 muestra una señal analógica. Aunque los equipos analógicos son útiles para ciertas actividades de simulación, no son adecuados para procesar los grandes volúmenes de datos que los equipos digitales suelen volver a solicitar para procesar. Por lo tanto, casi todos los ordenadores en uso hoy en día son digitales.

### Figura 1.3: Grabadora de sonido que muestra un archivo analógico.



Las redes neuronales biológicas son analógicas. Como verás en la siguiente sección, simular redes neuronales analógicas en un ordenador digital puede presentar algunos desafíos. Las neuronas aceptan una señal analógica a través de las dendritas, como se ve en la Figura 1.1. Debido a que esta señal es analógica, el voltaje de cada señal variará. Si la tensión está dentro de un cierto rango, la neurona disparará. Cuando una neurona se dispara, se transfigura una nueva señal analógica de la neurona de disparo a otras neuronas. Esta señal se lleva a cabo sobre el axón de la neurona de disparo. Las regiones de entrada y salida se denominan sinapsis. Más adelante, en el capítulo 5, el ejemplo de red neuronal feedforward backpropagation demostrará que las sinapsis son la interfaz entre un programa y una red neuronal.

Una neurona toma una decisión disparando o no disparando. Las decisiones que se toman son decisiones de muy bajo nivel. Requiere un gran número de decisiones que deben tomar muchas neuronas sólo para leer esta frase. Las decisiones de alto nivel son el resultado de la entrada colectiva y la salida de muchas neuronas.

Las decisiones se pueden representar gráficamente trazando la entrada y salida de neuronas. La Figura 1.4 ilustra la entrada y salida de una neurona en particular. Como se mostrará en el capítulo 5, hay diferentes tipos de neuronas, todas las cuales tienen diferentesgráficos de salida en formade ly. Mirando el gráfico que se muestra en la Figura 1.4, se puede ver que la neurona en este ejemplo disparará a cualquier entrada superior a 0.5 voltios.

### Figura 1.4: Niveles de activación de una neurona.







Una neurona biológica es capaz de tomar decisiones básicas. Las redes neuronales artificiales se basan en este modelo. A continuación se muestra una explicación de cómo se simula este modelo mediante un ordenador digital.

# Resolver problemas con redes neuronales

Un objetivo significativo de este libro es mostrarte cómo construir redes neuronales y enseñarte cuándo usarlas. Como programador de redes neuronales, debes estar de pie qué problemas son adecuados para las soluciones de red neuronal y cuáles no. Un programador eficaz de la red neuronal también sabe qué estructura de red neuronal, si la hay, es más aplicable a un problema dado. Esta sección comienza centrándose primero en aquellos problemas que no son propicios para una solución de red neuronal.

## Problemas no adecuados para una solución de red neuronal

Los programas que se escriben fácilmente como diagramas de flujo son ejemplos de problemas para los que las redes neuronales no son apropiadas. Si su programa consta de pasos bien definidos, las técnicas de programación normales se mantendrán.

Otro criterio a tener en cuenta es si es probable que cambie la lógica del programa. Una de las principales características de las redes neuronales es su capacidad para aprender. Si el algoritmo utilizado para resolver el problema es una regla de negocio inmutable, no hay ninguna razón para usar una red neuronal. De hecho, podría ser perjudicial para la aplicación si la red neuronal intenta encontrar una mejor solución, y comienza a divergir del proceso deseado y produce resultados inesperados.

Por último, las redes neuronales a menudo no son adecuadas para problemas en los que debe saber exactamente cómo se derivó la solución. Una red neuronal puede ser muy útil para resolver el problema para el que fue entrenado, pero la red neuronal no puede explicar su razonamiento. La red neuronal sabe algo porque fue entrenado para saberlo. La red neuronal no puede explicar cómo siguió una serie de pasos para obtener la respuesta.

## Problemas adecuados para una red neuronal

Aunque hay muchos problemas para los que las redes neuronales no son adecuadas, también son muchos problemas para los que una solución de red neuronal es bastante útil. Además, las redes neuronales a menudo pueden resolver problemas con menos líneas de código que un algoritmo de programación tradicional. Es importante entender qué problemas requieren un enfoque de red neuronal.

Las redes neuronales son particularmente útiles para resolver problemas que no se pueden expresar como una serie de pasos, como reconocer patrones, clasificación, serie de predicciones y minería de datos.

*“Pattern* *recognition”* es quizás el uso más común para redes neuronales. Para este tipo de problema, la red neuronal se presenta un patrón. Esto podría ser una imagen, un sonido o cualquier otro dato. A continuación, la red neuronal intenta determinar si los datos de entrada coinciden con un patrón que se ha entrenado para reconocer. El capítulo 3, *“Using a Hop-field Neural Network”* proporciona un ejemplo de una red neuronal simple que reconoce los patrones de entrada.

La clasificación es un proceso que está estrechamente relacionado con el reconocimiento de patrones. Una red neuronal entrenada para la clasificación está diseñada para tomar muestras de entrada y clasificarlas en grupos. Estos grupos pueden ser borrosos, carentes de límites claramente definidos. Alternamente, estos grupos pueden tener límites bastante rígidos. El capítulo 12, *“OCR and the Self-Organizing Map”*, presenta un programa de ejemplo capaz de *“optical character recognition”* (OCR). Este programa toma muestras de escritura a mano y las clasifica por carta (por ejemplo, la letra "A" o “B").

## Formación de redes neuronales

Las neuronas individuales que componen una red neuronal están interconectadas a través de sus sinapsis. Estas conexiones permiten que las neuronas se señalen entre sí como se procesa la información. No todas las conexiones son iguales. A cada conexión se le asigna un peso de conexión. Si no hay conexión entre dos neuronas, entonces su peso de conexión es cero. Estos pesos son los que determinan la salida de la red neuronal; por lo tanto, se puede decir que los pesos de conexión forman la memoria de la red neuronal.

El entrenamiento es el proceso por el cual se asignan estos pesos de conexión. La mayoría de los algoritmos de entrenamiento comienzan asignando números aleatorios a una matriz de ponderaciones. A continuación, se examina la validez de la red neuronal. A continuación, los pesos se ajustan en función del rendimiento de la red neuronal y la validez de los resultados. Este process se repite hasta que el error de validación está dentro de un límite aceptable. Hay muchas maneras de entrenar redes neuronales. Los métodos de capacitación en redes neuronales generalmente se dividen en las categorías de enfoques supervisados, no supervisados y varios híbridos.

El entrenamiento supervisado se lleva a cabo dando a la red neuronal un conjunto de datos de muestra junto con las salidas previstas de cada una de estas muestras. El entrenamiento supervisado es la forma más común de entrenamiento de la red neuronal. Como procedimiento el entrenamiento supervisado, la red neural se toma a través de una serie de iteraciones, o épocas, hasta que la salida de la red neuronal coincide con la salida prevista, con una tasa razonablemente pequeña de error. Cada época es un paso a través de las muestras de entrenamiento.

La capacitación no supervisada es similar a la capacitación supervisada, excepto que no se proporcionan salidas anticipadas. El entrenamiento no supervisado generalmente ocurre cuando la red neuronal se utiliza para clasificar las entradas en varios grupos. La capacitación involucra muchas épocas, al igual que en la formación supervisada. A medida que avanza el entrenamiento, los grupos de clasificación son "descubiertos" por la red neuronal. La capacitación no supervisada está cubierta en el capítulo 11, Utilizando un mapa autoorganizado.

Hay varios métodos híbridos que combinan aspectos de la formación supervisada y no supervisada. Uno de estos métodos se llama entrenamiento de refuerzo. En este método, se proporciona una red neuronal con datos de muestra que no contienen out-puts anticipados, como se hace con el entrenamiento no supervisado. Sin embargo, para cada salida, se indica al trabajo de la red neuronal si la salida fue correcta o incorrecta dada la entrada.

Es muy importante entender cómo entrenar correctamente una red neuronal. Este libro explora varios métodos de entrenamiento de redes neuronales, incluyendo backpropagation, recocido simulado y algoritmos genéticos. Los capítulos 4 a 7 están dedicados a la formación de redes neuronales. Una vez que la red neuronal está entrenada, debe ser valiente- fechado para ver si está listo para su uso.

## Validación de redes neuronales

El paso final, validar una red neuronal, es muy importante porque le permite determinar si se requiere entrenamiento adicional. Para validar correctamente una red neuronal, los datos de validación deben reservarse que estén completamente separados de los datos de entrenamiento.

Como ejemplo, considere la posibilidad de una red de clasificación que debe agrupar elementos en tres grupos de clasificación diferentes. Se le proporcionan 10.000 elemen ts demuestra. Para estos datos de ejemplo, se conoce el grupo en el que se debe clasificar cada elemento. Para un sistema de este tipo, dividiría aleatoriamente los datos de muestra en dos grupos de 5.000 elementos cada uno. El primer grupo formaría el conjunto de entrenamiento. Una vez que la red fue entrenada correctamente, el segundo grupo de 5.000 elementos se utilizaría para validar la red neuronal.

Es muy importante que siempre se mantenga un grupo separado de datos para validación. En primer lugar, entrenar una red neuronal con un conjunto de muestras determinado y también usar este mismo conjunto para predecir el error anticipado de la red neuronal para un nuevo conjunto arbitrario seguramente conducirá a malos resultados. El error logrado con el conjunto de entrenamiento casi siempre será sustancialmente menor que el error en un nuevo conjunto dedatos sample. Siempre se debe mantener la integridad de los datos de validación.

Esto plantea una pregunta importante. ¿Qué sucede si la red neuronal que acaba de terminar de entrenar funciona mal en el conjunto de datos de validación? Si este es el caso, debe examinar las posibles causas. Podría significar que los pesos aleatorios iniciales no eran apropiados. Volver a ejecutar el entrenamiento con nuevos pesos iniciales podría corregir esto. Si bien un conjunto inadecuado de pesos aleatorios iniciales podría ser la causa, una posibilidad más, es que los datos de entrenamiento no fueron elegidos correctamente.

Si la validación está funcionando mal, es probable que haya datos presentes en el conjunto de validación que no estaban disponibles en los datos de entrenamiento. La forma en que se debe corregir esta situación es probar un enfoque aleatorio diferente para separar los datos en conjuntos de entrenamiento y validación. Si esto falla, debe combinar el entrenamiento y validación se establece en un gran conjunto de entrenamiento. A continuación, se deben adquirir nuevos datos para serve como los datosde validación.

En algunas situaciones puede ser imposible recopilar datos adicionales para usarlos como datos de entrenamiento o validación. Si este es el caso, entonces usted se queda sin otra opción que combinar todo o parte del conjunto de validación con el conjunto de entrenamiento. Si bien este enfoque renunciará a la seguridad de una buena validación, si no se pueden adquirir datos adicionales esta puede ser su única alternativa.

# Problemas comúnmente resueltos con redes neuronales

Hay muchos problemas diferentes que se pueden resolver con una red neuronal. Sin embargo, las redes neuronales se utilizan comúnmente para abordar tipos particulares de problemas. Los siguientes cuatro tipos de problemas se resuelven con frecuencia con redes neuronales:

* Clasificación
* Predicción
* Reconocimiento de patrones
* Optimización

Estos problemas se debatirán brevemente en las siguientes secciones. Muchos de los programas ex-amplios a lo largo de este libro abordarán uno de estos cuatro problemas.

## Clasificación

La clasificación es el proceso de clasificar entradas en grupos. Por ejemplo, una empresa seguro puede querer clasificar las aplicaciones de seguros en diferentes categorias de riesgo, o una organización en línea puede querer que su sistema de correo electrónico clasifique el correo entrante en grupos de mensajes de spam y no spam.

Decada diez, la red neuronal se entrena presentándole un grupo de muestra de datos e instrucciones sobre qué grupo pertenece cada elemento de datos. Esto permite a la red neuronal aprender las características que pueden indicar la pertenencia al grupo.

## Predicción

La predicción es otra aplicación común para las redes neuronales. Dada una serie basada en el tiempo de datos de entrada, una red neuronal predecirá los valores futuros. La precisión de la conjetura dependerá de muchos factores, como la cantidad y relevancia de los datos de entrada. Por ejemplo, las redes neuronales se aplican comúnmente a problemas que implican predecir movimientos en los mercados financieros.

Este libro mostrará varios ejemplos de predicción. El Capítulo 9, Redes Neuronales Predictivas, proporciona una explicación introductoria de cómo utilizar una red neuronal para hacer prediccioness. El Capítulo 10 muestra un enfoque neuronal básico para analizar el S&P 500.

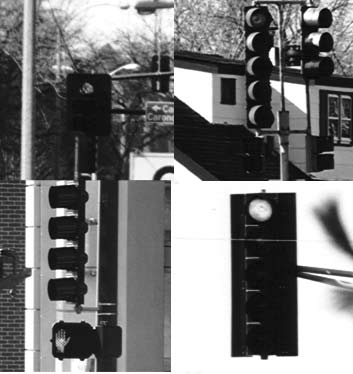
## Reconocimiento de patrones

El reconocimiento de patrones es uno de los usos más comunes para las redes neuronales. El reconocimiento de patrones es una forma de clasificación. El reconocimiento de patrones es simplemente la capacidad de reconocer un patrón. El patrón debe reconocerse incluso cuando está distorsionado. Considere el siguiente uso diario del reconocimiento de patrones.

Cada persona que tiene una licencia de conducir debe ser capaz de identificar con precisión un semáforo. Este esun procedimiento de reconocimiento de patrones extremadamente crítico llevado a cabo por innumerables conductores todos los días. Sin embargo, no todos los semáforos tienen el mismo aspecto, y la aparición de un semáforo en particular puede ser alterada dependiendo de la hora del día o la temporada. Además,existen muchas variaciones del semáforo. Aun así, reconocer un semáforo no es una tarea difícil para un conductor humano.

¿Qué tan difícil es escribir un programa informático que acepte una imagen y le diga si es un semáforo? Sin el uso de *“neural networks”*, esta podría ser una tarea muy compleja. La Figura 1.5 ilustra varios semáforos diferentes. La mayoría de los algoritmos de programación comunes se agotan rápidamente cuando se presentan con un problema complejo de reconocimiento de patrones.

### Figura 1.5: Semáforosdi fferent



Más adelante en este libro, se proporcionará un ejemplo de una red neuronal que lee la escritura a mano. Esta red neuronal realiza la tarea reconociendo patrones en las letras individuales dibujadas.

## Optimización

Otro uso común para las redes neuronales es la optimización. La optimización se puede aplicar a muchos problemas diferentes para los que se busca una solución óptima. Es posible que la red neuronal no siempre encuentre la solución óptima; más bien, busca encontrar una solución aceptable. Los problemas de optimización incluyen el ensamblaje de la placa de circuito, la asignación de recursos y muchos otros.

Tal vez uno de los problemas de optimización más conocidos es el problema de las ventas ambulantes (TSP). Un vendedor debe visitar un número establecido de ciudades. Le gustaría visitar todas las ciudades y viajar el menor número posible de millas. Con sólo unas pocas ciudades, este no es un problema complejo. Sin embargo, con un gran número de ciudades, los métodos de cálculo de fuerza bruta no funcionan casi tan bien como un enfoque de red neuronal.

# Uso de una red neuronal simple

A continuación se muestra un ejemplo de una red neuronal muy simple. Aunque la red es simple, incluye casi todos los elementos de las redes neuronales más complejas que se cubrirán más adelante en este libro.

En primer lugar, considere una neurona artificial, como se muestra en la Figura 1.6.

### Figura 1.6: Neurona artificial.

1.5



T=2.5

Hay dos atributos asociados con esta neurona: el umbral y el peso. El peso es 1.5 y el umbral es 2.5. Una señal entrante será amplificada, o desamplificada, por el peso a medida que cruza la sinapsis. Si la entrada ponderada supera el umbral, entonces la neurona se disparará.

Considere un valor de uno (**verdadero**) presentado como la entrada a la neurona. El valor de uno se multiplicará por el valor de peso de 1,5. Esto da como resultado un valor de 1,5. El valor de 1,5 está por debajo del umbral de 2,5, por lo que la neurona no se disparará. Esta neurona nunca disparará con valores booleanos. No todas las neuronas solo aceptan valores booleanos. Sin embargo, las neuronas de esta sección sólo aceptan los valores booleanos de uno (**verdadero**) y cero (**falso**).

**Una red neuronal para el operador And**

La neurona que se muestra en la Figura 1.6 no es terriblemente útil. Sin embargo, la mayoría de las neuronas no son terriblemente útiles, al menos no de forma independiente. Las neuronas se utilizan con otras neuronas para formar redes. Ahora veremos una red neuronal que actúa como una puerta **AND.**  La Tabla 1.1 muestra la tabla de la verdad para la operación lógica **AND.**

### Tabla 1.1: La operación lógica AND

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A Y B** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Se puede crear una red neuronal simple que reconozca la operación lógica **AND.** Habrá tres neuronas en total. Esta red contendrá dos entradas y una salida. Una red neuronal que reconoce la operación lógica **AND** se muestra en la Figura 1.7.

### Figura 1.7: Una red neuronal que reconoce el funcionamiento lógico AND.



1

1

T=1.5

Hay dos entradas a la red mostradas en el cuadro 1.7. Cada neurona tiene un peso de uno. El umbral es 1.5. Por lo tanto, una neurona sólo se disparará si ambas entradas son verdaderas. Si cualquiera de las entradas es **falsa,**la suma de las dos entradas no superará el umbral de 1,5.

Considere las entradas de **true**  y **false**. La entrada **verdadera** enviará un valor de uno a la neurona de salida. Esto está por debajo del umbral de 1,5. Del mismo modo, considere las aportaciones de  **true**  y **true**. Cada neurona de entrada enviará un valor de uno. Estas dos entradas se resumen por la neurona de salida, lo que resulta en dos. El valor de dos es mayor que 1,5, por lo tanto, la neurona se disparará.

## Una red neuronal para la operación or

También se pueden crear redes neuronales para reconocer otras operaciones lógicas. Con- sidere la operación lógica  **OR.** La tabla de la verdad para la operación lógica  **OR** se muestra en la Tabla 1.2. La **operación** lógica OR es  **true** si cualquiera de las entradas es  **true.**

### Tabla 1.2: La operación lógica OR

|  |  |  |
| --- | --- | --- |
| **Un** | **B** | **A O B** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

La red neuronal que reconocerá la operación **OR** se muestra en la Figura 1.8.

### Figura 1.8: Una red neuronal que reconoce la operación lógica OR.



1

1

T=0.9

La red neuronal  **OR** se parece mucho a la red neuronal  **AND.** La mayor diferencia es el valor umbral. Debido a que el umbral es menor, solo una de las entradas necesita tener un valor de **true** para que la neurona de salida se dispare.

## Una red neuronal para la operación XOR

A continuación, consideraremos una red neuronal para la operación exclusiva **(XOR)**lógica. La tabla de la verdad **XOR** se muestra en la Tabla 1.3.

### Tabla 1.3: La operación lógica XOR

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Un XOR B** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

La operación lógica **XOR**  requiere una red neuronal ligeramente más compleja que los operadores **AND**  y  **OR.** Las redes neuronales presentadas hasta ahora sólo han tenido dos capas: una capa de entrada y una capa de salida. Las redes neuronales más complejas también incluyen una o más capas ocultas. El operador **XOR**  requiere una capa oculta. Como resultado, la red neuronal **XOR**  a menudo se convierte en una especie de aplicación "Hello World" para redes neuronales. Verás al operador **XOR** de nuevo en este libro, ya que diferentes tipos de red neuronal son introduced y entrenados.

La Figura 1.9 muestra una red neuronal de tres capas que se puede utilizar para reconocer la

**Operador** XOR.

### Figura 1.9: Una red neuronal que reconoce la operación lógica XOR.



1

1

1

1

T=1.5

T=0.5

-1

1

T=0.5

Considere el caso en el que los valores de **true**  y **true** se presentan a esta red neuronal. Ambas neuronas de la capa oculta reciben el valor de dos. Esto está por encima de los umbrales de ambas neuronas de capa oculta, por lo que ambos dispararán. Sin embargo, la primera neurona oculta tiene un peso de -1, por lo que su contribución a la neurona de salida es -1. La segunda neurona tiene un peso de 1, por lo que su contribución a la neurona de salida es

1. La suma de 1 y -1 es cero. Cero está por debajo del umbral de la neurona de salida, por lo que la neurona de salida no se dispara. Esto es coherente con la operación **XOR,** ya que producirá false si ambas entradas son verdaderas.

Ahora considere si los valores de **false**  y **true** se presentan a la red neuronal. La entrada a la primera neurona de capa oculta será 1, desde la segunda neurona la entrada será 0. Esto es inferior al umbral de 1,5, por lo que no se disparará. La entrada a la segunda neurona de capa oculta también será 1, mas 0 de la segunda neurona de entrada. Esto es más que el 0.5 de umbral, por lo que se disparará. La entrada a la neurona de salida será cero desde la neurona oculta izquierda y 1 desde la neurona oculta derecha. Esto es mayor que 0.5, por lo que la neurona de salida se disparará. Esto es consistente con la operación **XOR,** porque producirá true si una de las neuronas de entrada es **verdadera** y la otra **falsa.**

Por supuesto, las redes neuronales mostradas en las secciones anteriores son muy simples. Sin embargo, ilustraron todos los puntos clave para redes neuronales más complejas. Los capítulos futuros introducirán tipos adicionales de redes neuronales; sin embargo, las obras de redes neuronales casi siempre contarán con pesos y umbrales.

# Resumen del capítulo

Las computadoras pueden procesar la información considerablemente más rápido que los seres humanos. Sin embargo, una computadora es incapaz de realizar muchas de las mismas tareas que un humano puede realizar fácilmente. Para procesos que no se pueden dividir fácilmente en un número finito de pasos, una red neuronal puede ser una solución ideal.

El término red neuronal normalmente se refiere a una red neuronal artificial. Una red neuronal artificial intenta simular las redes neuronales biológicas contenidas en los cerebros de todos los animales. Las redes neuronales artificiales se introdujeron por primera vez en la década de 1950 y a través de los años de su desarrollo han experimentado numerosos setbacks; todavía no han cumplido la promesa de simular el pensamiento humano.

Las redes neuronales están construidas de neuronas que forman capas. La entrada se presenta a las capas de las neuronas. Si la entrada a una neurona está dentro del rango que la neurona ha sido trained para, entonces la neurona se disparará. Cuando una neurona se dispara, se envía una señal a las capas de neuronas a las que está conectada la neurona de disparo. Las conexiones entre las neuronas se denominan sinapsis. Java se puede utilizar para construir una red de este tipo.

Las redes neuronales deben ser entrenadas y validadas. Por lo general, un conjunto de entrenamiento se divide a la mitad para proporcionar un conjunto de entrenamiento y validación. El entrenamiento de la red neuronal consiste en ejecutarla redneuronal a través de los datos de entrenamiento hasta que la red neuronal aprende a reconocer el conjunto de entrenamiento con una tasa de error suficientemente baja. La validación se produce cuando se comprueban los resultados de la red neuronal.

El hecho de que una red neuronal pueda procesar los datos de entrenamiento con una baja tasa de error, no significa que la red neuronal esté entrenada y lista para su uso. Antes de que la red neuroral se ponga en uso en producción, debe validarse. La validación implica presenseestablecer la validación en la red neuronal y comparar los resultados reales producidos por la red neuronal con los resultados previstos.

La red neuronal está lista para ser puesta en producción si, al final del proceso vali-dation, los resultados de la ejecución de validación cumplen un nivel de error satisfactorio. Si los resultados no son satisfactorios, entonces la red neuronal tendrá que ser reentrenada, ya que se puede colocar en producción.

Las redes neuronales están compuestas por muchas neuronas. Su umbral y peso se combinan en matrices de peso. Una matriz de peso se almacena en una matriz matemática regular. El capítulo 2 introduciráclases de Java severa l diseñadas para almacenar valores en una matriz y realizar matemáticas de matriz. Las redes neuronales de este libro se construirán sobre estas clases matriciales.

# Vocabulario

Activation Level

Analog Computer

Artificial Intelligence

Artificial Neura Network

Axon

Binary

Biological Neuronal Network

Dendrite

Fire

Digital Computer

Hidden Layer

Input Layer Layer

Matrix

Neural Network Neuron

Output Layer

Pattern Reconnition Preduction

Supervised Training Singal

Synapse

Trehshols

Training

Truth Table

Unsupervised Training

Weight Matrix

Validation

Xor

# Preguntas para revisión

* 1. **¿Qué tipos de problemas pueden abordar mejor las redes neuronales que las prácticas de programación tradicionales?**

Problemas que no se pueden expresar con una serie de pasos. En este tipo de problemas se pueden mencionar la clasificaciones, las predicciones.

* 1. **A continuación, se muestra una red neuronal simple para un operador. Escriba la tabla de la verdad para esta red neuronal. ¿Qué operador es éste?**

Corresponde a una compuerta logica AND



2

2

T=3.0

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **RESULTADO** |
| **0** | **0** | **0** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |

* 1. **Explicar el propósito de una red neuronal de clasificación. ¿Para qué tipo de problema del "mundo real" podría utilizarse una red neuronal de clasificación?**

Su proposito readica en tomar un grupo de datos y a partir de sus caracteristicas de importancia los reacrupa en grupos. Los campos agricolas pueden suponer un área de aplcación para este tipo de redes neuronales, en la seleccion de la materia prima.

* 1. **¿Cuál es el propósito de un valor umbral?**

El valor del umbral corresponde al valor que define cuando una neurona es disparada; para que suceda esto los valores de entrada deben ser igual o mayor al valor del umbral.

* 1. **Explicar la diferencia entre el entrenamiento supervisado y el entrenamiento no supervisado.**

En el entrenamiento supervisado temenos un resultado esperado para la red; es decir ya temenos los parametros que debe cubrir la red entrenada, mientras que en el no supervisano no temenos un resultado esperado.