MAGNA

INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL

MAGNA

INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL

APLICACIÓN DE REDES NEURONALES EN MANTENIMIENTO PREDICTIVO

Ing. Breynner Chávez

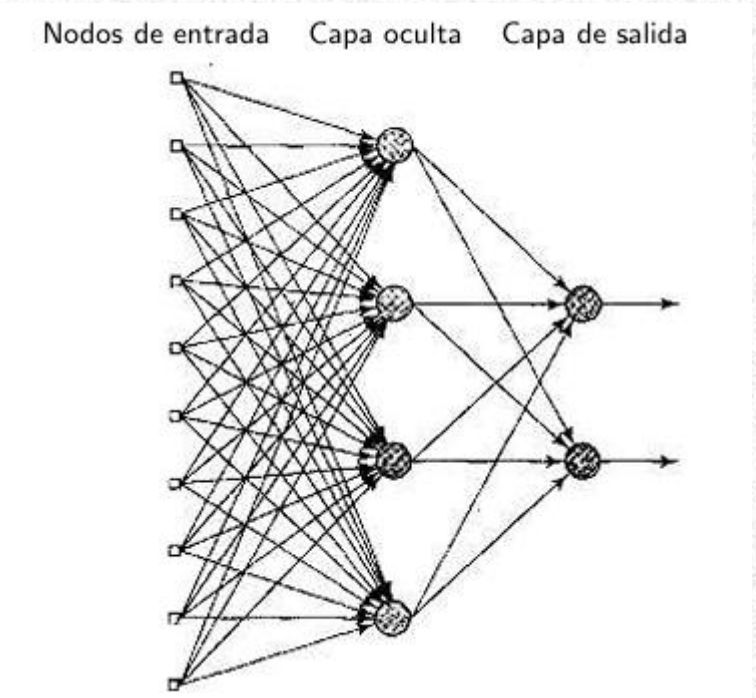
Fundamentos de redes neuronales

¿Qué son las Redes Neuronales?

Las redes neuronales (RNA) son programas de computadora diseñados para aprender y resolver problemas, inspirándose en cómo funciona nuestro cerebro.

Funciona igual que nuestras neuronas, se comunican para tomar decisiones (como mover un brazo o resolver un problema), las redes neuronales procesan información para llegar a una conclusión o hacer una predicción.

Están diseñadas para aprender y reconocer patrones complejos en grandes volúmenes de datos.



Esquema de una RNA de dos capas.

Fundamentos de redes neuronales

¿Qué son las Redes Neuronales?

La industria moderna enfrenta grandes desafíos para garantizar la continuidad de las operaciones.
Siendo una gran reto minimizar fallas inesperadas y mejorar la eficiencia de los equipos.

Las redes neuronales, como una rama avanzada de la inteligencia artificial, ofrecen un enfoque potente para abordar el reto de analizar datos históricos y en tiempo real, provenientes de los sensores industriales.

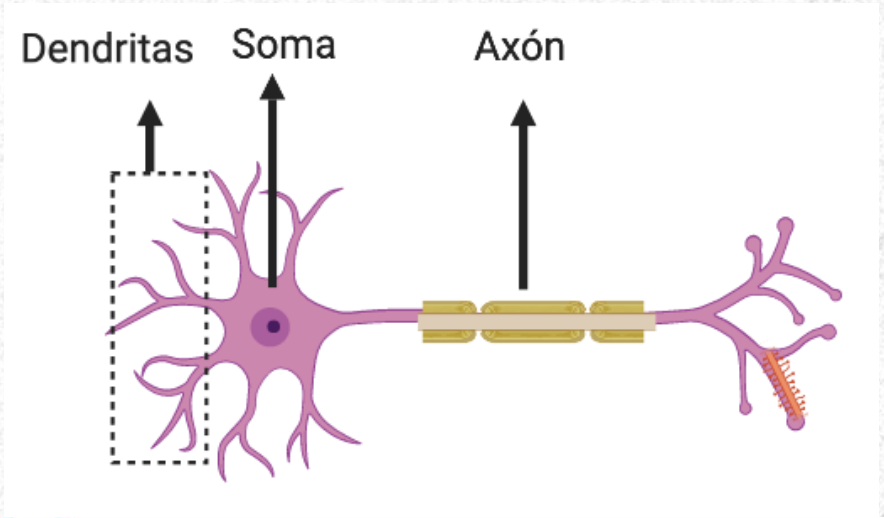
Con el objetivo de prever fallos antes de que ocurran, reduciendo costos y mejorando la seguridad.

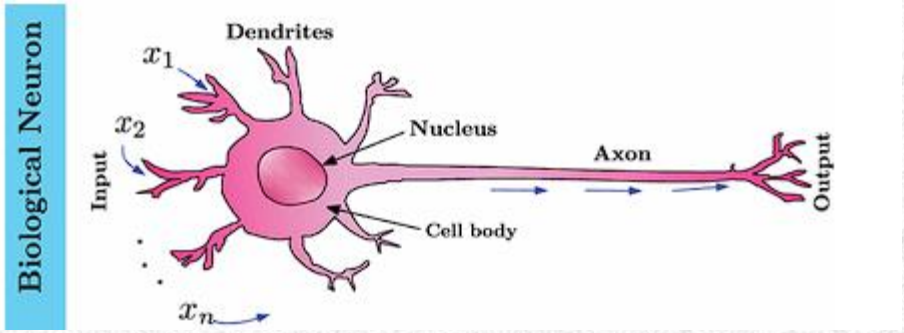
Fundamentos de redes neuronales

Analogía biológica: Cómo se relaciona con el cerebro

Neurona biológica: En el cerebro, cada neurona recibe señales de otras neuronas a través de conexiones llamadas sinapsis.

Cuando las señales son lo suficientemente fuertes, la neurona dispara una respuesta que se transmite a otras neuronas.



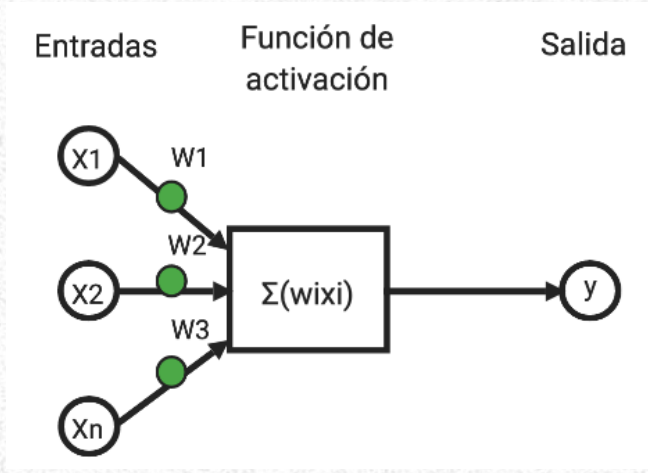


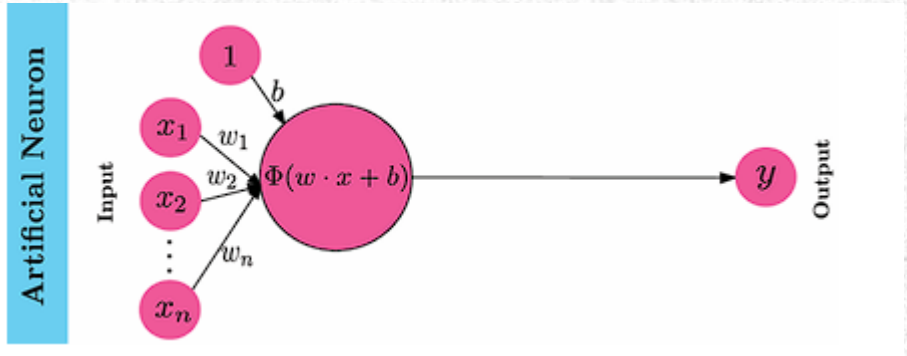
Fundamentos de redes neuronales

Analogía biológica: Cómo se relaciona con el cerebro

Neurona artificial: En una red neuronal artificial, cada "neurona" es una unidad matemática que: Recibe información (números) de otras neuronas.

Combina esa información usando "pesos" (que determinan la importancia de cada señal).Decide si la señal es lo suficientemente fuerte aplicando una fórmula matemática llamada función de activación.





Fundamentos de redes neuronales

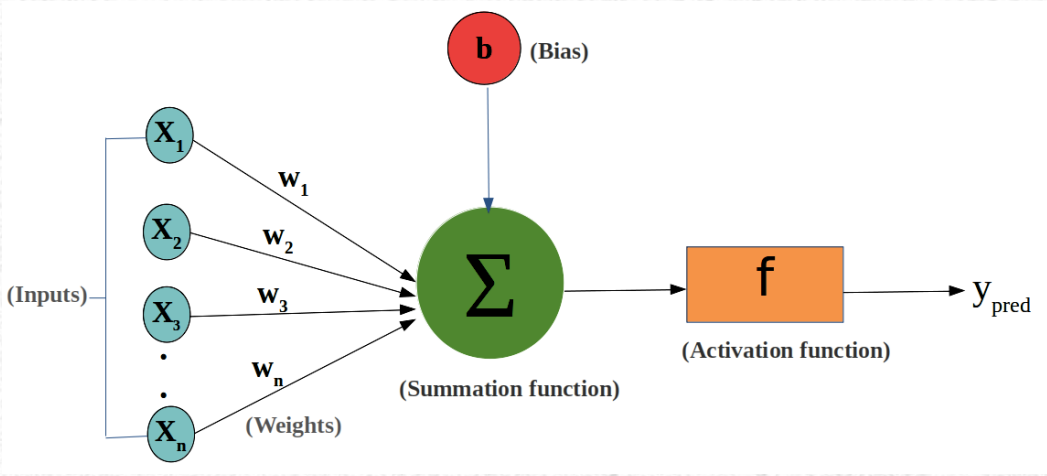
Pesos y sesgos:

Son parámetros clave que se ajustan durante el proceso de entrenamiento para que la red aprenda a resolver problemas específicos.

Pesos (Weights)

Los pesos son valores numéricos que conectan una neurona a otra. Determinan la importancia de la entrada de una neurona para la siguiente.

Cada conexión en una red tiene un peso asociado. Este peso amplifica o reduce la contribución de un valor de entrada en la salida de la neurona.



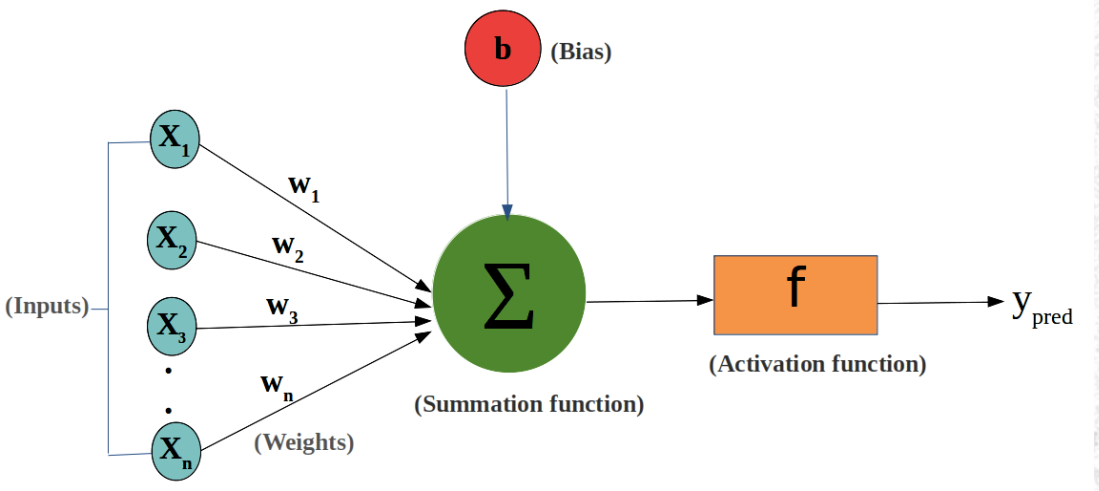
Fundamentos de redes neuronales

Pesos y sesgos:

Matemáticamente:

Si una neurona recibe entradas $x1,x2,\ldots,xn$ con pesos asociados $w1,w2,\ldots,wn$w 1, el cálculo interno es:
$z=w1x1+w2x2+\cdots+wnxn+b$

Donde b es el sesgo.

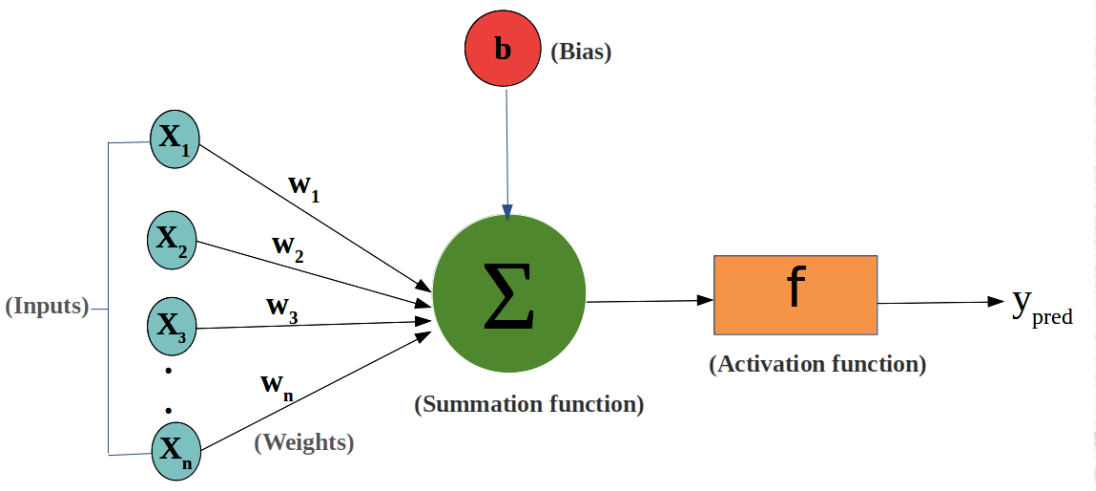


Fundamentos de redes neuronales

Pesos y sesgos:

El sesgo es un valor adicional que se suma al cálculo interno de cada neurona. Actúa como una constante que permite a la red aprender patrones que no necesariamente pasan por el origen (cuando todas las entradas son cero).

Sin un sesgo, una neurona puede tener limitaciones para ajustarse correctamente a ciertos datos.



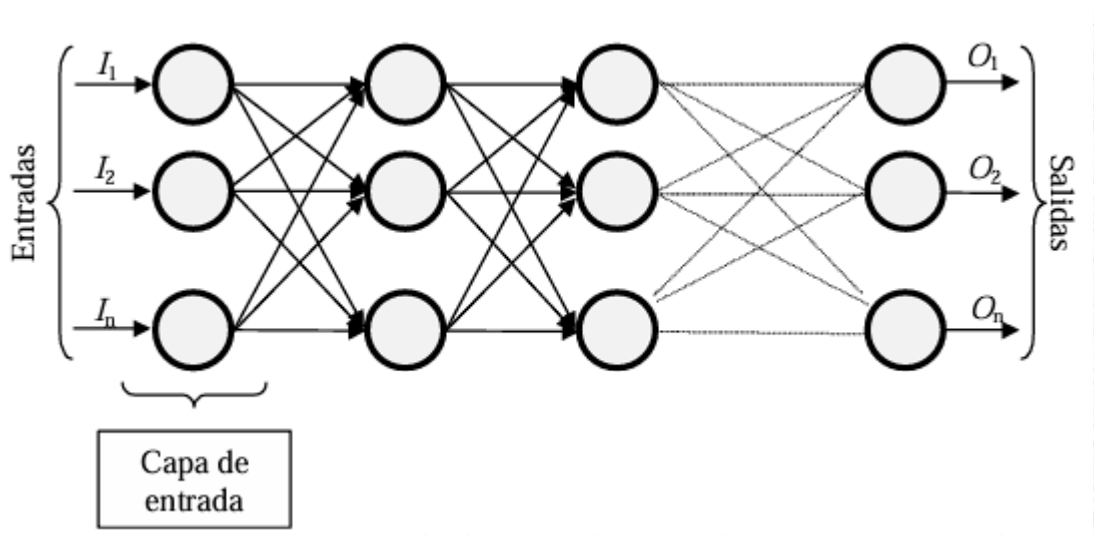
Fundamentos de redes neuronales

Arquitectura Básica de una Red Neuronal

Las redes neuronales están compuestas por tres partes principales: capa de entrada, capas ocultas, y capa de salida. Cada una de estas cumple una función específica en el procesamiento de datos para obtener predicciones.

Capa de entrada:

Es el punto inicial donde ingresan los datos al sistema.
Representa las características (también llamadas variables o "features") del problema que queremos analizar.



Fundamentos de redes neuronales

Arquitectura Básica de una Red Neuronal

Características:

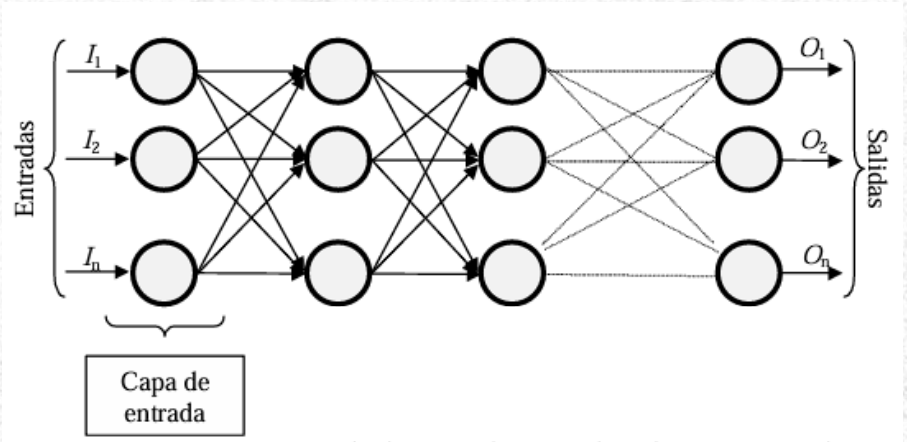
Propósito: Transportar la información del mundo real a la red neuronal.

Número de neuronas: Corresponde al número de variables que se están utilizando como entrada.

Si queremos predecir fallos en una máquina industrial, las entradas podrían ser:

- **Temperatura.**
- **Nivel de vibración.**
- **Presión del aceite.**
- **Velocidad del motor.**

Si tenemos cuatro variables de entrada, entonces la capa de entrada tendrá cuatro neuronas.

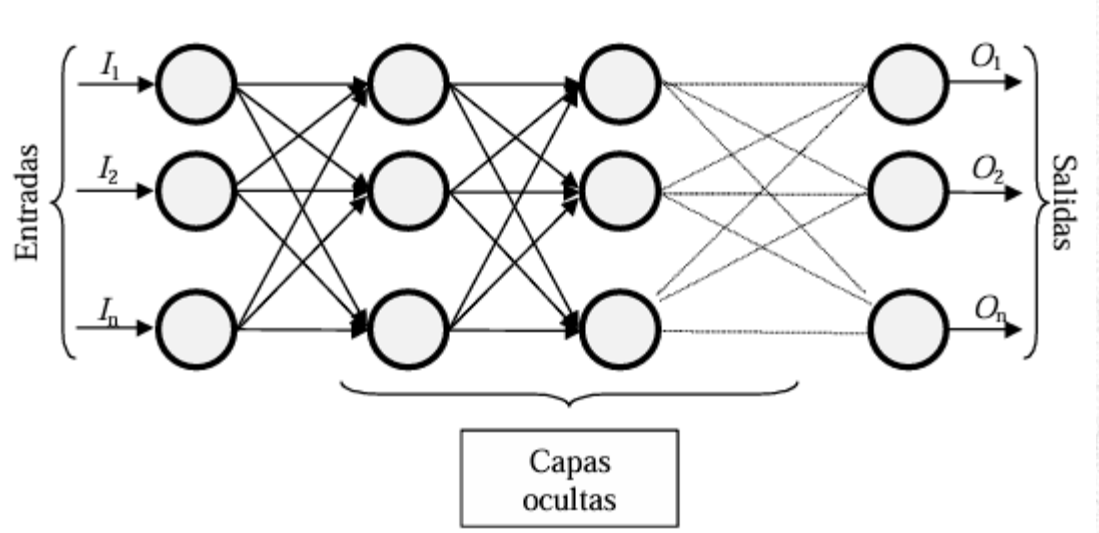


Fundamentos de redes neuronales

Arquitectura Básica de una Red Neuronal

Capas ocultas:

Las capas ocultas son el "motor" de la red neuronal. Aquí se procesan y transforman los datos para extraer patrones complejos que no son evidentes a simple vista.



Características de las Capas Ocultas:

Neurotransmisión: Cada neurona recibe datos de todas las neuronas de la capa anterior, realiza un cálculo matemático y transmite el resultado a las neuronas de la siguiente capa.

Fundamentos de redes neuronales

MAGNA
INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL

Arquitectura Básica de una Red Neuronal

Cálculo interno: Se realiza una combinación lineal de los datos de entrada, multiplicados por sus pesos, y se le agrega un sesgo. Luego, el resultado pasa por una **función de activación** que introduce no linealidad.

Cálculo matemático: $z=w1x1+w2x2+\cdots+wnxn$

Donde:

- xi: Las entradas de la capa anterior.
- wi: Pesos asociados a cada conexión.
- b: Sesgo (valor adicional).
- z: El valor de salida intermedia (antes de la activación).

Fundamentos de redes neuronales

MAGNA
INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL

Arquitectura Básica de una Red Neuronal

Patrones complejos: A medida que los datos pasan por varias capas ocultas, la red "aprende" a identificar patrones más abstractos:

- En una red con una capa oculta, puede identificar relaciones simples.
- Con más capas ocultas, puede aprender relaciones más complejas, como interacciones entre múltiples variables.

Profundidad de la Red:

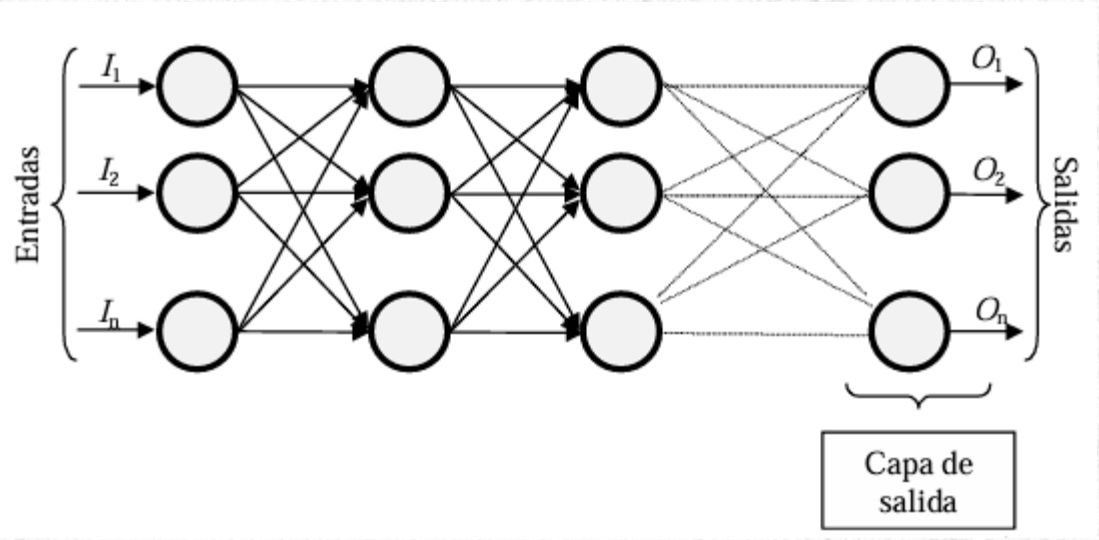
•Si una red tiene muchas capas ocultas, se denomina **red profunda** (deep learning). Esto es útil para problemas muy complejos, como el reconocimiento de imágenes o el procesamiento de grandes volúmenes de datos.

Fundamentos de redes neuronales

Arquitectura Básica de una Red Neuronal

Capa de salida:

- Es el resultado final de todos los cálculos.
- Si el objetivo es predecir un fallo, la salida podría ser:
  - ❑ "0" = No habrá fallo.
  - ❑ "1" = Habrá un fallo.
  - ❑ O un número como "80%" indicando la probabilidad de un fallo.



Fundamentos de redes neuronales

Aplicaciones en Mantenimiento Predictivo

Las redes neuronales son muy útiles en la industria para prevenir problemas y mejorar la eficiencia. Aquí algunos ejemplos:

1. **Predicción de fallos en equipos:**
   - ✓ Las redes analizan datos históricos (como la temperatura, la presión o la vibración de una máquina) para detectar patrones que preceden a un fallo.
   - ✓ Ejemplo: Si una turbina genera más vibración de lo habitual durante varias semanas, la red puede predecir que es probable que falle pronto.

Fundamentos de redes neuronales

Aplicaciones en Mantenimiento Predictivo

2. **Detección de anomalías:**
   - ✓ La red aprende el "comportamiento normal" de un equipo y puede identificar cuándo algo no está funcionando como debería, incluso si no lleva al fallo inmediato.
   - ✓ Ejemplo: Si un motor comienza a calentarse de forma irregular, la red detectará este comportamiento extraño.

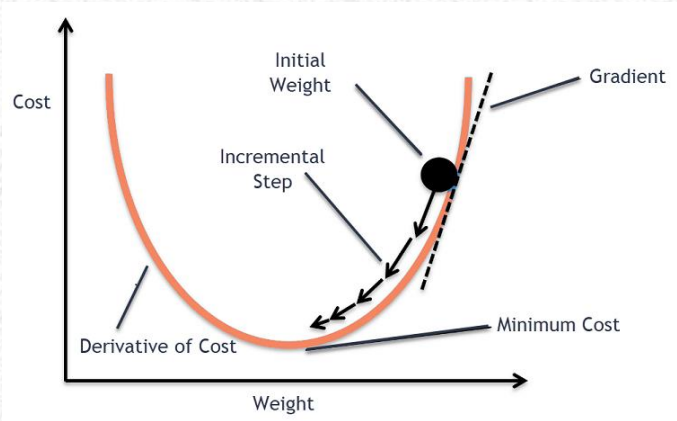
3. **Optimización del tiempo de mantenimiento:**
   - ✓ En lugar de realizar mantenimientos regulares (que pueden ser innecesarios) o reactivos (esperar a que algo falle), las redes ayudan a planificar mantenimientos en el momento exacto que realmente se necesita.
   - ✓ Esto ahorra dinero y tiempo para la empresa.

Gradient Descent (Descenso por Gradiente)

MAGNA
INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL

¿Qué es Gradient Descent?

Es un algoritmo de optimización ampliamente utilizado en aprendizaje automático y redes neuronales para minimizar la función de pérdida.
La función de pérdida mide qué tan lejos están las predicciones de los valores reales. Gradient Descent ajusta los parámetros de la red neuronal (pesos y sesgos) para hacer que las predicciones sean más precisas.
Imaginemos que estamos en la cima de una montaña y necesitamos descender al valle más bajo (el punto mínimo).
La pendiente (o gradiente) indica en qué dirección debemos movernos para descender. Si seguimos moviéndonos en esa dirección paso a paso, eventualmente llegaremos al fondo.



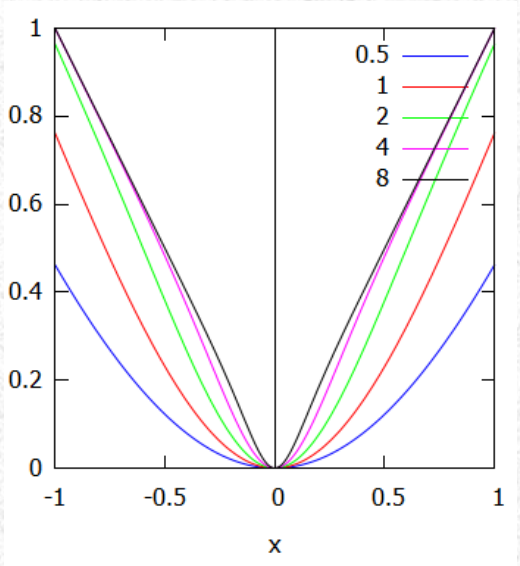
Gradient Descent (Descenso por Gradiente)

¿Qué es Gradient Descent?

Matemáticas detrás de Gradient Descent

1. **Función de pérdida (L):** Es la medida de error entre las predicciones de la red neuronal y los valores reales. Una de las funciones más comunes es el Error Cuadrático Medio (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$



- y_i : Valor real.
- $\hat{y}_i$: Valor predicho.
- n : Número de muestras en el conjunto de datos.

El objetivo de Gradient Descent es minimizar esta función.

Gradient Descent (Descenso por Gradiente)

MAGNA
INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL

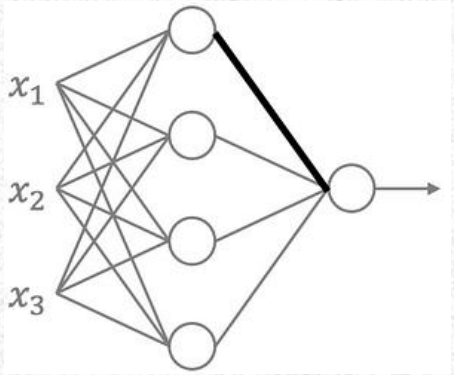
¿Qué es Gradient Descent?

Matemáticas detrás de Gradient Descent

2. **Desplazamiento descendente: Cómo encuentra el mínimo:** El algoritmo de gradient descent calcula el gradiente de la función de pérdida con respecto a los pesos. Este gradiente indica la dirección en la que aumenta la función de pérdida. El algoritmo ajusta los pesos en la dirección opuesta al gradiente para minimizar el error.

$$w_{nuevo} = w_{viejo} - n \cdot \frac{\partial L}{\partial w}$$

- w_{nuevo} : Nuevo valor del peso.
- w_{viejo}: Peso actual.
- n: Tasa de aprendizaje, que controla el tamaño del paso que tomamos en cada iteración.
- $\frac{\partial L}{\partial w}$: Gradiente de la función de pérdida respecto al peso.



Gradient Descent (Descenso por Gradiente)

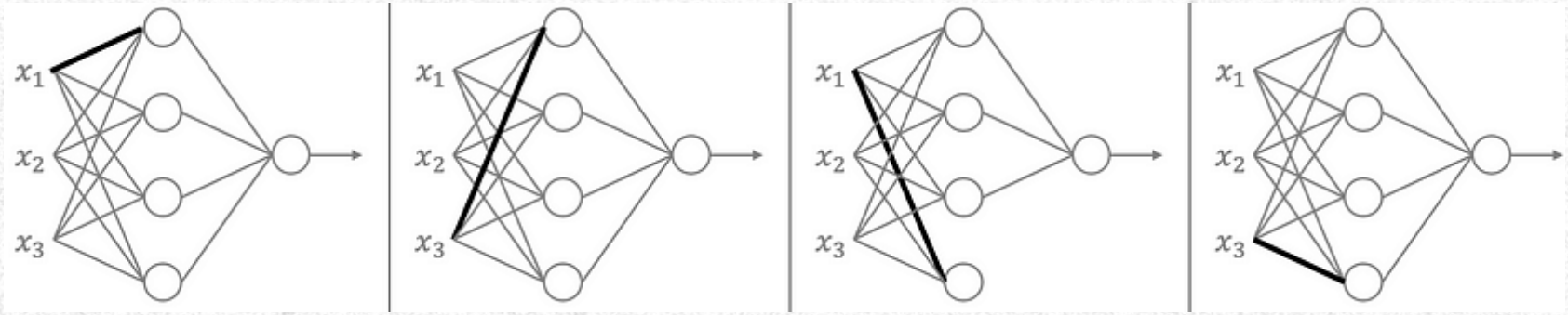
¿Qué es Gradient Descent?

La Tasa de Aprendizaje (n)

- **Actualización de pesos (regla de actualización):**
  - Tasa pequeña (n = 0.01): El descenso será lento, pero seguro.
  - Tasa grande (n = 0.5): El descenso será rápido, pero puede saltarse el punto más bajo (mínimo global), causando inestabilidad.

Iteraciones y convergencia:

El descenso de gradiente repite este proceso varias veces (llamadas épocas) hasta que el error es lo suficientemente bajo o deja de mejorar.

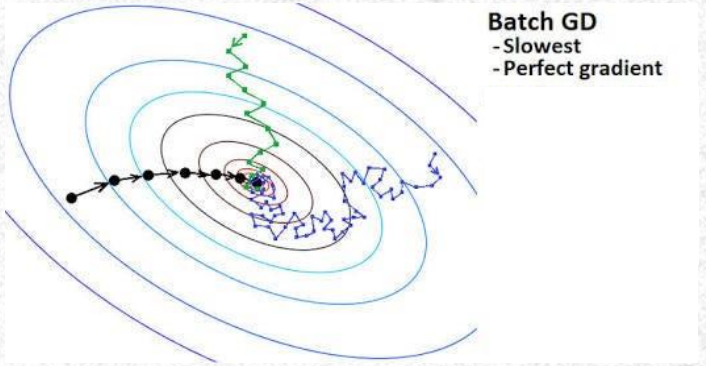


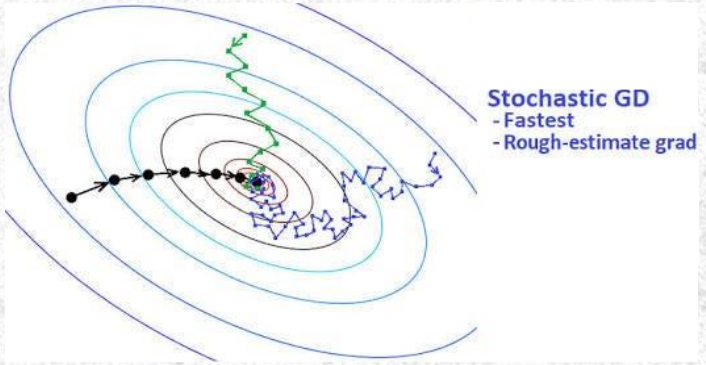
Gradient Descent (Descenso por Gradiente)

MAGNA
INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL

Tipos de Gradient Descent

1. **Batch Gradient Descent (Descenso por Gradiente en Lote):**
   - Usa todo el conjunto de datos para calcular el gradiente en cada iteración.
   - Ventaja: Es estable y preciso.
   - Desventaja: Computacionalmente costoso para conjuntos de datos grandes.
2. **Stochastic Gradient Descent (SGD):**
   - Actualiza los pesos después de calcular el gradiente de una única muestra de datos.
   - Ventaja: Es rápido y funciona bien en conjuntos de datos grandes.
   - Desventaja: Puede ser menos estable, ya que los gradientes varían mucho con cada muestra.

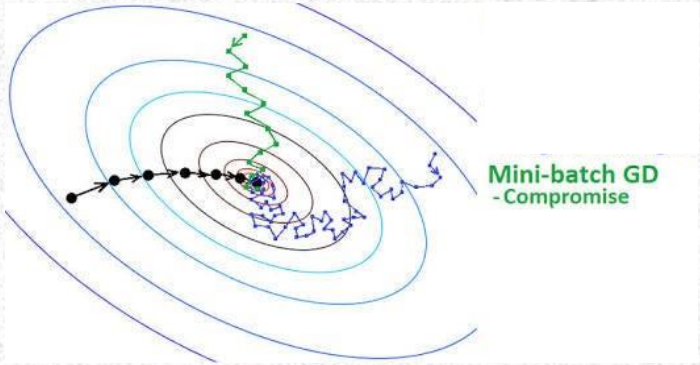


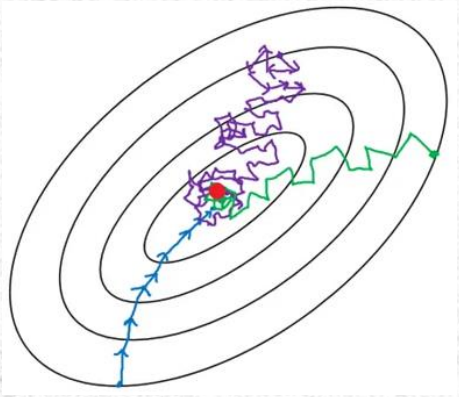


Gradient Descent (Descenso por Gradiente)

Tipos de Gradient Descent

3. **Mini-Batch Gradient Descent:**
   - Calcula el gradiente usando un subconjunto de datos (mini-lote).
   - Ventaja: Combina la velocidad de SGD con la estabilidad de Batch Gradient Descent.





Batch Gradient Descent: La función de pérdida desciende suavemente hacia el mínimo. Se define el tamaño de batch_size igual al número total de muestras en el conjunto de entrenamiento.
Stochastic Gradient Descent: El descenso tiene saltos irregulares debido a la inestabilidad. Se define el tamaño de batch_size en 1 (una muestra por actualización).
Mini-Batch Gradient Descent: Un compromiso entre ambos: descenso relativamente estable y rápido. Se define un tamaño de batch_size mayor que 1, pero menor que el número total de muestras en el conjunto de entrenamiento (32, 64, 128, etc.)

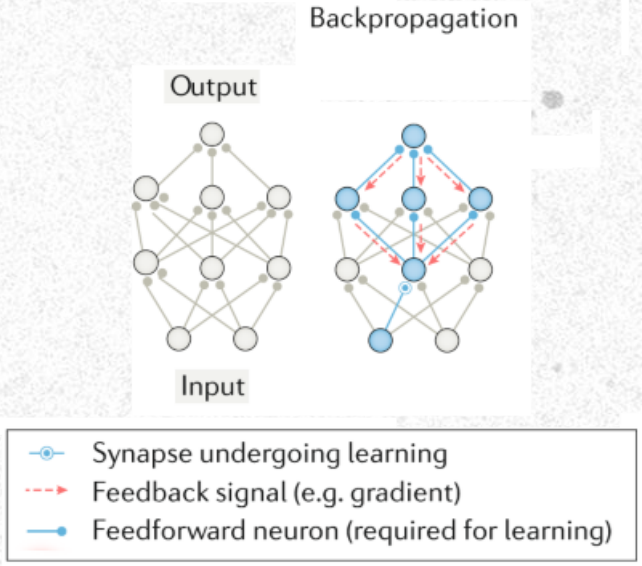
Backpropagation (Retropropagación)

¿Qué es Backpropagation?

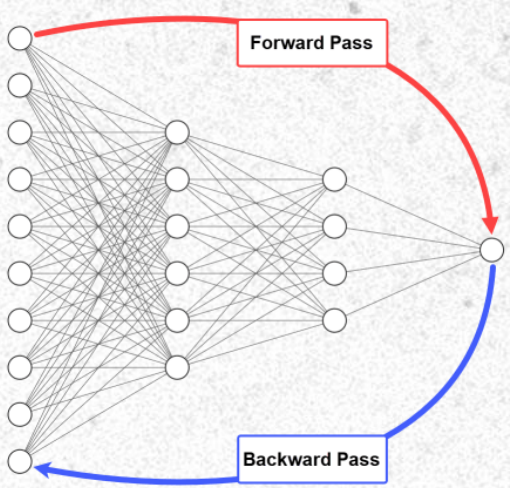
La retropropagación (o Backpropagation) es un algoritmo que se utiliza en redes neuronales para ajustar los pesos y minimizar el error. Este algoritmo es esencial para que las redes neuronales aprendan a hacer predicciones más precisas.

En pocas palabras:

1. Calcula cuánto nos equivocamos en la predicción (error).
2. Propaga ese error hacia atrás en la red, desde la salida hacia las capas iniciales.
3. Ajusta los pesos de la red para reducir el error en futuras predicciones.



Backpropagation (Retropropagación)



¿Cómo funciona Backpropagation?

El algoritmo se divide en dos fases principales:

1. **Forward Pass (Propagación hacia adelante):**

   En esta fase, los datos "viajan" desde la capa de entrada hasta la capa de salida.

   Pasos:

   1. Entrada: Los datos iniciales (por ejemplo, imágenes, números) entran a la red.
   2. Cálculo de salidas: En cada capa, los datos se transforman mediante operaciones matemáticas:
      - Se multiplican por los pesos.
      - Se suma un sesgo (bias).
      - Se aplica una función de activación.
   3. Predicción: Finalmente, la red produce una salida predicha ($\hat{y}$).

      Ejemplo simplificado: Supongamos que queremos predecir si un motor necesita mantenimiento inmediato (y=1) o si puede seguir operando (y=0), la red genera una salida como: $\hat{y}$ =0.85

      Esto significa que la red estima un 85% de probabilidad de que el motor requiera mantenimiento inmediato

Backpropagation (Retropropagación)

¿Cómo funciona Backpropagation?

2. **Cálculo del error**

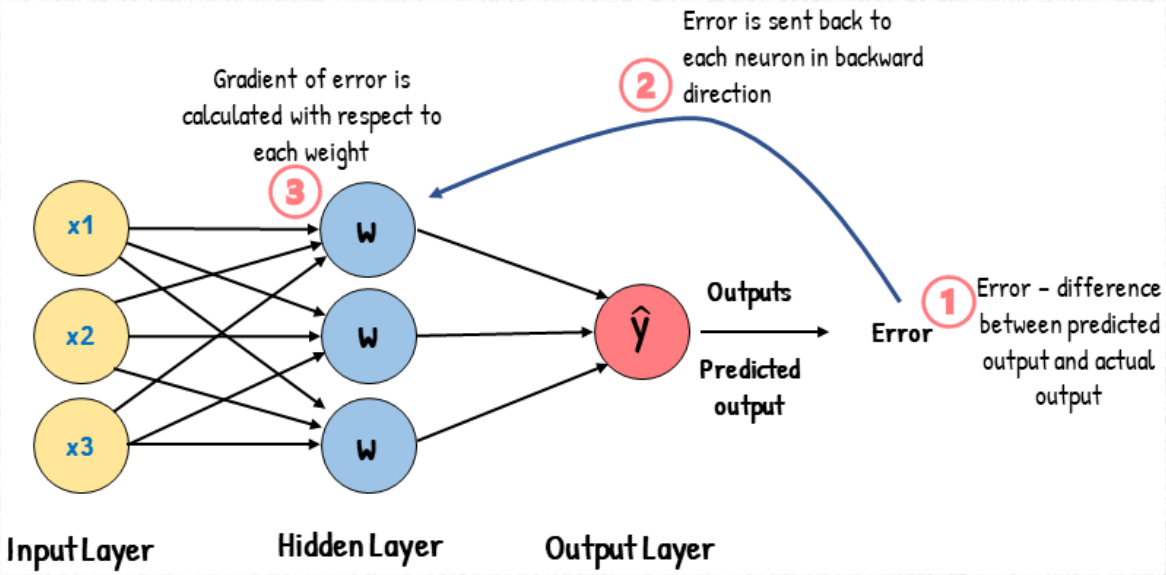
   Aquí comparamos lo que la red predijo ($\hat{y}$) con el valor real (y) usando una función de error. Fórmula general del error:

   $$E = y - \hat{y}$$

   Ejemplo: Si la red predijo $\hat{y}$ =0.8 y la etiqueta real es y=1, el error será:

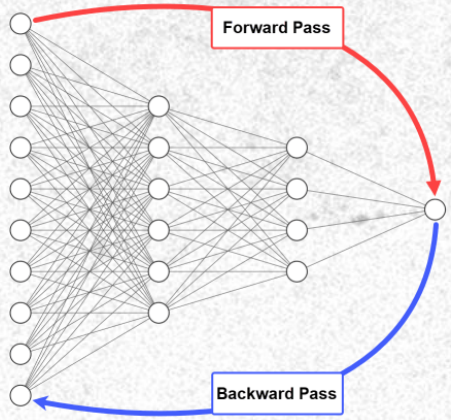
   $$E = 1 - 0.8 = 0.2$$

   El objetivo es reducir este error tanto como sea posible.



Backpropagation (Retropropagación)

MAGNA
INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL



¿Cómo funciona Backpropagation?

3. **Backward Pass (Propagación hacia atrás)**
   En esta fase, la red ajusta sus pesos para aprender de sus errores.
   ¿Cómo se hace esto?
   1. Regla de la cadena: El error de la capa de salida se descompone y se "propaga" hacia atrás a través de la red. Esto se hace calculando cómo cada peso contribuyó al error.
   2. Cálculo de gradientes: Para cada peso (w) de la red, se calcula un gradiente ($\frac{\partial E}{\partial w}$), que indica cuánto debe cambiar ese peso para reducir el error.
   3. Ajuste de pesos: Una vez calculado el gradiente, los pesos se ajustan ligeramente en la dirección que reduce el error, utilizando una fórmula basada en Gradient Descent:

$$w_{nuevo} = w_{viejo} - n \cdot \frac{\partial E}{\partial w}$$

Backpropagation (Retropropagación)

Ejemplo

En Supongamos que trabajamos en una planta industrial y queremos predecir la vibración anómala en equipos mecánicos.
Los datos recopilados incluyen:
Entradas (Features):

- ✓ Velocidad del motor.
- ✓ Carga aplicada al sistema.
- ✓ Temperatura ambiental.
- ✓ Historial de vibraciones.

Salida (Target):

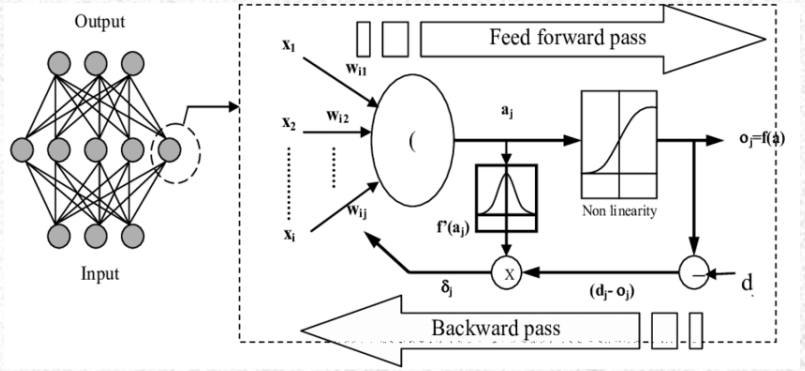
- ✓ Nivel de vibración (valor continuo).

Backpropagation (Retropropagación)

Ejemplo

1. **Cálculo del Error:**
   La red neuronal predice el nivel de vibración basado en las características del equipo. Por ejemplo, predice que la vibración es de 1.2 mm/s, pero el valor real medido es 1.5 mm/s.
   El error se calcula como:

$$error = \frac{1}{2}(1.5 - 1.2)^2 = 0.045$$



Backpropagation (Retropropagación)

MAGNA
INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL

Ejemplo

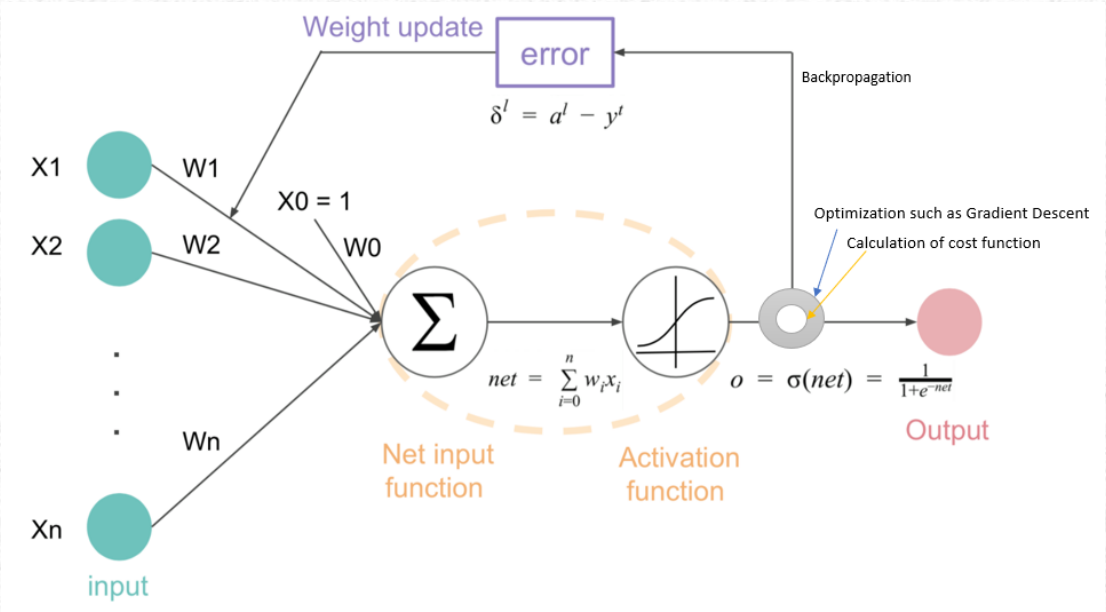
2. **Propagación hacia Atrás:**
   El error se propaga hacia las capas intermedias, ajustando los pesos de las conexiones para reflejar mejor la relación entre las características y la vibración.
   Por ejemplo:
   - Si se determina que el peso asociado a la velocidad del motor es demasiado bajo para reflejar su verdadera influencia, el algoritmo lo incrementará.
   - Si la red descubre que la carga aplicada tiene un efecto insignificante en la vibración, el peso correspondiente se reducirá.
3. **Actualización de Pesos:**
   Usando la fórmula de ajuste de pesos, se incrementan o reducen los pesos de cada conexión según el gradiente calculado. En el siguiente ciclo, la red ajustada hará predicciones más precisas.

Backpropagation (Retropropagación)

Relación entre Backpropagation y Gradient Descent

- **Backpropagation** calcula los gradientes ($\frac{\partial E}{\partial w}$) para cada peso en la red.
- **Gradient Descent** utiliza esos gradientes para actualizar los pesos.

En resumen, Backpropagation es el cerebro matemático detrás del proceso de aprendizaje.



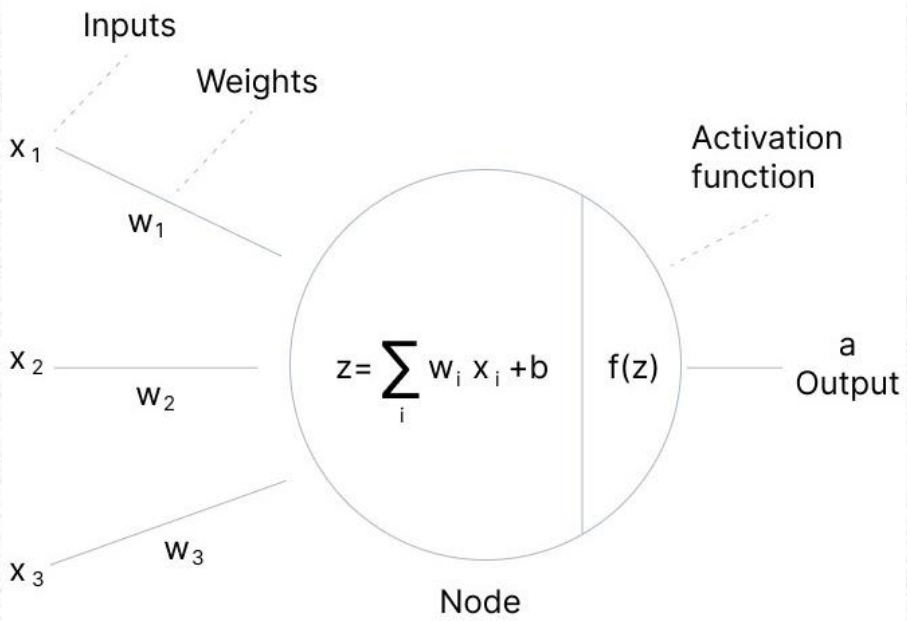
Funciones de Activación

¿Qué son las funciones de activación y cuál es su rol en las redes neuronales?

Las funciones de activación son componentes clave en las redes neuronales que introducen no linealidad en los modelos. Sin ellas, las redes serían equivalentes a un modelo lineal, sin capacidad de aprender patrones complejos en los datos.

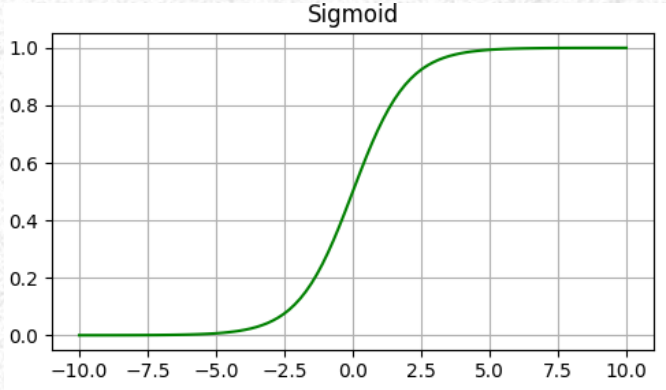
En términos sencillos:

- Actúan como "interruptores" que deciden si cierta información debe pasar o no.
- Permiten a las redes aprender relaciones más sofisticadas entre entrada y salida.



Funciones de Activación

Funciones de Activación Comunes

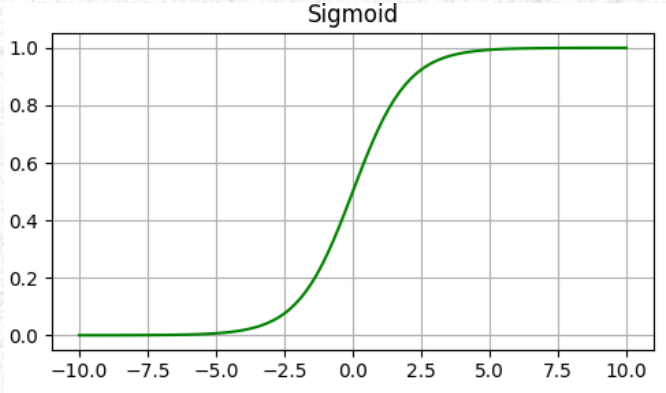


1. **Sigmoid (Logística)**
   - ✓ Fórmula: $\sigma(x) = \frac{1}{1+e^{-x}}$
   - ✓ Rango de salida: Entre 0 y 1.
   - ✓ Rol principal: Convierte valores de entrada en probabilidades, haciendo que sean más fáciles de interpretar. Muy útil para modelos que predicen probabilidades (como clasificación binaria).
   - ✓ Características: La función Sigmoid mapea cualquier valor de entrada entre 0 y 1, lo que la convierte en una opción ideal para modelar probabilidades. Por ejemplo, en un sistema de predicción de fallas, se puede usar en la capa de salida para indicar la probabilidad de que un equipo falle.

Funciones de Activación

Funciones de Activación Comunes

1. Sigmoid (Logística)

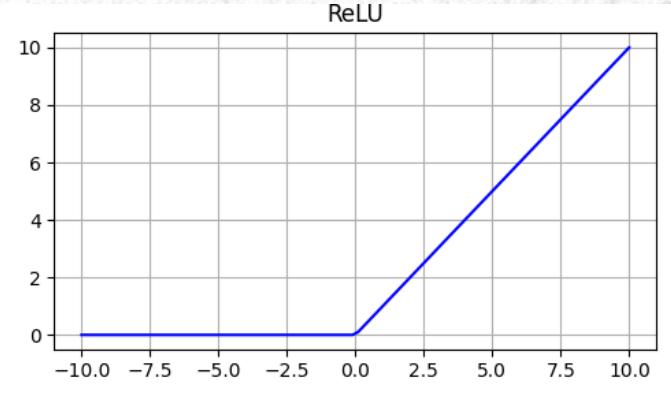


Ventajas	Desventajas
➢ Simple y fácil de entender. ➢ Produce salidas suaves y acotadas.	➢ Problema de desvanecimiento del gradiente: En valores muy grandes o pequeños, el gradiente se hace casi cero, dificultando el entrenamiento de redes profundas. ➢ Salidas no centradas en cero, lo que puede ralentizar el aprendizaje.

✓ Ejemplo de uso: Predicción de si un motor “fallará" o "no fallará ".

Funciones de Activación

Funciones de Activación Comunes



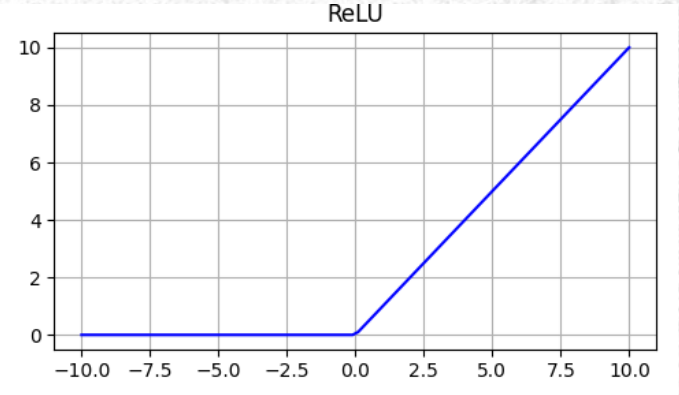
2. ReLU (Rectified Linear Unit)

- ✓ Fórmula: $f(x) = max(0, x)$
- ✓ Rango de salida: 0 a infinito.
- ✓ Rol principal: Introduce no linealidad, manteniendo la simplicidad computacional. Resuelve el problema del desvanecimiento del gradiente de funciones como sigmoid.
- ✓ Características: ReLU es rápida de calcular y ayuda a superar el problema de desvanecimiento del gradiente (un problema que puede hacer que las redes profundas no aprendan adecuadamente). Sin embargo, puede tener el inconveniente de que algunos de los nodos de la red se "mueran" (es decir, dejan de aprender), aunque este problema se ha mitigado con variantes de ReLU, como Leaky ReLU.

Funciones de Activación

Funciones de Activación Comunes

2. ReLU (Rectified Linear Unit)



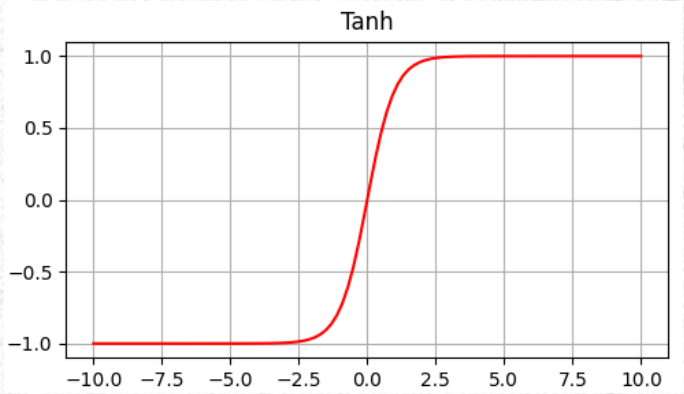
Ventajas	Desventajas
➢ Fácil de calcular. ➢ Acelera la convergencia del modelo en redes profundas.	➢ Problema de neuronas muertas: Si una neurona recibe valores negativos constantemente, su salida será siempre 0 y dejará de contribuir al aprendizaje.

✓ Ejemplo de uso: Estimación de tiempo hasta el fallo de un equipo basado en datos operativos, donde se resalta la importancia de cambios no lineales en los patrones de comportamiento.

Funciones de Activación

Funciones de Activación Comunes

3. Tanh (Tangente Hiperbólica)

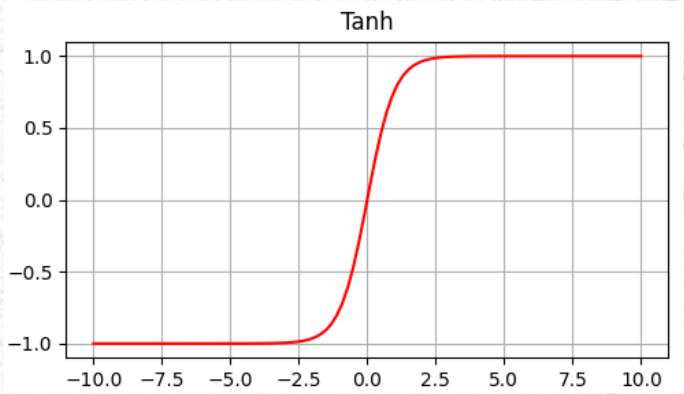


- ✓ Fórmula: $tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$
- ✓ Rango de salida: Entre -1 y 1.
- ✓ Rol principal: Similar a sigmoid, pero mejor para datos centrados en cero (cuando las entradas tienen valores positivos y negativos).
- ✓ Características: Al estar centrada en cero, la función Tanh es más eficiente en algunos casos porque las activaciones negativas también pueden transmitir información útil durante el proceso de aprendizaje.

Funciones de Activación

Funciones de Activación Comunes

3. Tanh (Tangente Hiperbólica)



Ventajas	Desventajas
➢ Salida centrada en cero, lo que facilita el aprendizaje en algunos casos. ➢ Útil para modelos donde los datos incluyen valores negativos.	➢ También sufre de desvanecimiento del gradiente, aunque en menor medida que sigmoid.

✓ Ejemplo de uso: Modelado de series temporales de datos de sensores (ej. predicción de temperatura o vibración en equipos).

Funciones de Activación

MAGNA
INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL

Comparación entre funciones

A continuación, presentamos una tabla para destacar sus diferencias clave:

Función	Rango	Ventaja clave	Desventaja clave	Usos comunes
Sigmoid	0 a 1	Intuitiva, salida acotada	Desvanecimiento del gradiente	Clasificación binaria
ReLU	0 a ∞	Simple, eficiente	Problema de neuronas muertas	Redes profundas (visión, texto)
Tanh	-1 a 1	Salida centrada en cero	Desvanecimiento del gradiente	Series temporales

Conexión de estos conceptos con Mantenimiento Predictivo

Gradient Descent	Backpropagation	Funciones de activación
Se usa para ajustar los pesos de una red que predice el tiempo de vida de una máquina.	Permite que la red aprenda de los datos históricos de sensores.	Aseguran que la red pueda aprender relaciones no lineales entre las variables, como las vibraciones y la temperatura.

Funciones de Activación

MAGNA
INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL

Ejemplo Práctico: Predicción de Fallas de una Máquina

Las redes neuronales se aplican en el mantenimiento predictivo porque pueden analizar grandes volúmenes de datos no estructurados (como lecturas de sensores), aprender patrones y hacer predicciones precisas. Las funciones de activación juegan un papel crucial en este proceso.

1. Entrada de Datos: Supongamos que tenemos sensores en una máquina que miden la temperatura, la vibración, la presión y la velocidad. Los datos de estos sensores se procesan para predecir si la máquina fallará en el futuro cercano.
2. Red Neuronal: La red neuronal se entrena con estos datos históricos. Las funciones de activación ayudan a la red a aprender relaciones no lineales entre los diferentes parámetros (como cómo la temperatura y la vibración interaccionan entre sí para afectar la probabilidad de un fallo).

Funciones de Activación

MAGNA
INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL

Ejemplo Práctico: Predicción de Fallas de una Máquina

3. Capa Oculta con ReLU: En las capas ocultas de la red, se utiliza la función ReLU para procesar grandes cantidades de datos de sensores. ReLU ayuda a que los modelos sean eficientes y aprenda de manera más efectiva, incluso cuando los datos contienen patrones complejos y no lineales. Al aplicar ReLU, la red se enfoca solo en los valores positivos que son relevantes para el aprendizaje (valores negativos se descartan).
4. Capa de Salida con Sigmoid: Finalmente, en la capa de salida, utilizamos una función de activación Sigmoid, que mapea los resultados de la red neuronal entre 0 y 1. Esto nos da una probabilidad de fallo, donde un valor cercano a 1 indica que el fallo es muy probable y un valor cercano a 0 indica que el fallo es poco probable.

Funciones de Activación

MAGNA
INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL

Ejemplo Práctico: Predicción de Fallas de una Máquina

¿Cómo las Funciones de Activación Mejoran la Predicción?

- ✓ ReLU en Capas Ocultas: ReLU permite a la red neuronal capturar relaciones complejas entre las entradas, ya que transforma la información no lineal de manera eficiente. Es útil cuando las características de los sensores no siguen patrones lineales, lo que es común en sistemas industriales.
- ✓ Sigmoid en la Capa de Salida: La salida de la red es un valor entre 0 y 1, lo que lo hace interpretable como una probabilidad. Si la probabilidad es mayor que un umbral específico (por ejemplo, 0.8), la red puede predecir que un fallo es probable y sugerir una intervención antes de que ocurra el problema.
- ✓ Tanh como Alternativa: En algunos casos, la función Tanh puede ser utilizada en lugar de la Sigmoid o ReLU, especialmente cuando los datos de entrada están centrados en cero. Esto puede mejorar la eficiencia del aprendizaje al eliminar sesgos en las activaciones.

MAGNA

INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL