1. **Historia de las redes neuronales artificiales**

1936 - Alan Turing. Fue el primero en estudiar el cerebro como una forma deber el mundo de la computación. Sin embargo, los primeros teóricos que concibieron los fundamentos de la computación neuronal fueron Warren McCulloch, un neurofisiólogo, y Walter Pitts, un matemático, quienes, en 1943, lanzaron una teoría acerca de la forma de trabajar de las neuronas (Un Cálculo Lógico de la Inminente Idea de la Actividad. Ellos modelaron una red neuronal simple mediante circuitos eléctricos.

1949 - Donald Hebb. Fue el primero en explicar los procesos del aprendizaje (que es el elemento básico de la inteligencia humana) desde un punto de vista psicológico, desarrollando una regla de como el aprendizaje ocurría. Aun hoy, este es el fundamento de la mayoría de las funciones de aprendizaje que pueden hallarse en una red neuronal. Su idea fue que el aprendizaje ocurría cuando ciertos cambios en una neurona eran activados. También intentó encontrar semejanzas entre el aprendizaje y la actividad nerviosa. Los trabajos de Hebb formaron las bases de la Teoría de las Redes Neuronales.

1950 - Karl Lashley. En sus series de ensayos, encontró que la información no era almacenada en forma centralizada en el cerebro sino que era distribuida encima de él. 1956 - Congreso de Dartmouth. Este Congreso frecuentemente se menciona para indicar el nacimiento de la inteligencia artificial.

1957 - Frank Rosenblatt. Comenzó el desarrollo del Perceptron. Esta es la red neuronal más antigua; utilizándose hoy en día para aplicación como identificador de patrones. Este modelo era capaz de generalizar, es decir, después de haber aprendido una serie de patrones podía reconocer otros similares, aunque no se le hubiesen presentado en el entrenamiento. Sin embargo, tenía una serie de limitaciones, por ejemplo, su incapacidad para resolver el problema de la función OR-exclusiva y, en general, era incapaz de clasificar clases no separables linealmente. 1959 - Frank Rosenblatt: Principios de Neurodinámica. En este libro confirmó que, bajo ciertas condiciones, el aprendizaje del Perceptron convergía hacia un estado.

1960 - Bernard Widroff/Marcian Hoff. Desarrollaron el modelo Adaline (ADAptative LINear Elements). Esta fue la primera red neuronal aplicada a un problema real (filtros adaptativos para eliminar ecos en las líneas telefónicas) que se ha utilizado comercialmente durante varias décadas. 1961 - Karl Steinbeck: Die Lernmatrix. Red neuronal para simples realizaciones técnicas (memoria asociativa). 1969 - Marvin Minsky/Seymour Papert. En este año casi se produjo la “muerte abrupta” de las Redes Neuronales; ya que Minsky y Papert probaron (matemáticamente) que el Perceptrons no era capaz de resolver problemas relativamente fáciles, tales comoel aprendizaje de una función no-lineal. Esto demostró que el Perceptron era muy débil, dado que las funciones no-lineales son extensamente empleadas en computación y en los problemas del mundo real.

1974 - Paul Werbos. Desarrolló la idea básica del algoritmo de aprendizaje de propagación hacia atrás (backpropagation); cuyo significado quedó definitivamente aclarado en 1985. 1977 - Stephen Grossberg: Teoría de Resonancia Adaptada (TRA). La Teoría de Resonancia Adaptada es una arquitectura de red que se diferencia de todas las demás previamente inventadas. La misma simula otras habilidades del cerebro: memoria a largo y corto plazo. 1985 - John Hopfield. Provocó el renacimiento de las redes neuronales con su libro: “Computación neuronal de decisiones en problemas de optimización.”

1986 - David Rumelhart/G. Hinton. Redescubrieron el algoritmo de aprendizaje de propagación hacia atrás (backpropagation). A partir de 1986, el panorama fue alentador con respecto a las investigaciones y el desarrollo de las redes neuronales. En la actualidad, son numerosos los trabajos que se realizan y publican cada año, las aplicaciones nuevas que surgen (sobretodo en el área de control) y las empresas que lanzan al mercado productos nuevos, tanto hardware como software.

1. **Ventajas y desventajas de una red neuronal**

**Ventajas de una red neuronal**

* **Aprendizaje Adaptativo:** Capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.
* **Auto-organización:** Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
* **Tolerancia a fallos:** La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.
* **Operación en tiempo real:** Los cómputos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.
* **Fácil inserción dentro de la tecnología existente:** Se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes.
* Las redes neuronales son sistemas dinámicos autoadaptativos. Son adaptables debido a la capacidad de autoajuste de los elementos procesales (neuronas) que componen el sistema. Son dinámicos, pues son capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones.
* las redes neuronales se usan para reconocer ciertas clases de patrones, ellas autoorganizan la información usada.

**Desventajas**

* La capacidad de las redes neuronales radica en su habilidad de procesar información en paralelo (esto es, procesar múltiples pedazos de datos simultáneamente). Desafortunadamente, las máquinas hoy en día son serie - sólo ejecutan una instrucción a la vez. Por ello, modelar procesos paralelos en máquinas serie puede ser un proceso que consuma mucho tiempo. Como todo en este día y época, el tiempo es esencial, lo que a menudo deja las redes neuronales fuera de las soluciones viables a un problema.
* las redes neuronales son la falta de reglas definitorias que ayuden a construir una red para un problema dado - hay muchos factors a tomar en cuanta: el algoritmo de aprendizaje, la arquitectura, el número de neuronas por capa, el número de capas, la representación de los datos y mucho más y otro de los problemas es la llamada caja negra el problema es que cuando modelamos estadísticamente somos capaces de ver que variables forman parte del modelo o cuales de las que finalmente se utilizaron para modelar fueron seleccionadas por los algoritmos para predecir o clasificar, podemos ver sus pesos y la ecuación final, cosa que no es posible en las redes neuronales.
* Existe una disyuntiva o discusión respecto a que es mejor si un modelo estadistico de probabilidad o una red neuronal, la respuesta a mi parecer es que depende del problema, hay problemas en los que funciona mejor una red pero hay otros en los que funciona mejor un modelo estadístico no hace daño probar las dos metodologías, la complejidad de modelamiento en ambos casos requiere de conocimientos sólidos y expertiz que permitan llegar a rsultados satisfactorios y válidos.

1. **Aplicaciones de las redes neuronales**

* Son capaces de determinar relaciones no lineales entre un conjunto de datos, asociando patrones de entrada o salidas correspondientes.
* Los tipos de aprendizaje disponibles pueden utilizarse para tareas de predicción y clasificación.
* Los modelos supervisados y no supervisados pueden ser aplicados para extraer y cancelar ruido de las señales.
* Una vez que la red ha sido entrenada y probada puede adaptarse por sí misma a los cambios.
* Una aproximación basada en redes neuronales artificiales puede aprender los modelos específicos de cada sistema de red y proporcionar aproximaciones aceptables de los sistemas.
* Reconocimiento de textos manuscritos.
* Reconocimiento del habla.
* Simulación de centrales de producción de energía.
* Detección de explosivos.
* Identificación de blancos de radares.

1. **Funciones de activación**

Funciones de activación (calculan la activación en función de la entrada total) y funciones de salida (calculan la salida en función de la activación).

La función de activación calcula la activación de la unidad en función de la entrada total y la activación previa, aunque en la mayor parte de los casos es simplemente una función no decreciente de la entrada total. Los tipos de función más empleados son: la función escalón, función lineal y la función sigmoidal.

La función de salida empleada usualmente es la función identidad y así la salida de la unidad de procesado es idéntica a su nivel de activación.

Las redes neuronales están formadas por un conjunto de neuronas artificiales interconectadas. Las neuronas de la red se encuentran distribuidas en diferentes capas de neuronas, de manera que las neuronas de una capa están conectadas con las neuronas de la capa siguiente, a las que pueden enviar información.

**Funciones de activación**

* Función de Activación Escalón
* Función de Activación Identidad
* Función de Activación Lineal -Mixta
* Función de Activación Sigmoidal
* Función de transferencia gaussiana
* Funciones de saturacion

**Función de activación Escalón:** La función escalón se asocia a neuronas binarias en las cuales cuando la suma de las entradas es mayor o igual que el umbral de la neurona, la activación es 1, si es menor, la activación es 0 (ó –1). Las redes formadas por este tipo de neuronas son fáciles de implementar en hardware, pero sus capacidades están limitadas.

**Función de activación identidad:** es adecuada cuando la función de activación que hemos utilizado para calcular la activación de la unidad es de tipo umbral.

**Función de activación Lineal – mixta:** La función escalón se asocia a neuronas binarias en las cuales cuando la suma de las entradas es mayor o igual que el umbral de la neurona, la activación es 1, si es menor, la activación es 0 (ó –1). Las redes formadas por este tipo de neuronas son fáciles de implementar en hardware, pero sus capacidades están limitadas.

**Función de activación Sigmoidal:** Cualquier función definida simplemente en un intervalo de posibles valores de entrada, con un incremento monotónico y que tengan ambos limites superiores e inferiores (por ejemplo las funciones sigmoidal y arco tangente), podrá realizar la función de activación o transferencia de forma satisfactoria

**Función de transferencia gaussiana:** Los centros y anchura de estas funciones pueden ser adaptados, lo cual las hace más adaptativas que las funciones sigmoidales.

**Función de saturación:** Son las funciones en las que los incrementos o disminuciones de la intensidad de la actividad de la unidad producen incrementos o disminuciones de los valores de salida hasta llegar a unos límites de salida máxima o mínima a partir de los cuales la salida se estabiliza y es la misma a pesar del incremento o disminución de la intensidad de actividad de la unidad.

1. **Historia de Perceptron**

La primera red neuronal conocida, fue desarrollada en 1943 por Warren McCulloch y Walter Pitts; esta consistía en una suma de las señales de entrada, multiplicadas por unos valores de pesos escogidos aleatoriamente. La entrada es comparada con un patrón preestablecido para determinar la salida de la red. Si en la comparación, la suma de las entradas multiplicadas por los pesos es mayor o igual que el patrón preestablecido la salida de la red es uno (1), en caso contrario la salida es cero (0). Al inicio del desarrollo de los sistemas de inteligencia artificial, se encontró gran similitud entre su comportamiento y el de los sistemas biológicos y en principio se creyó que este modelo podía computar cualquier función aritmética o lógica.

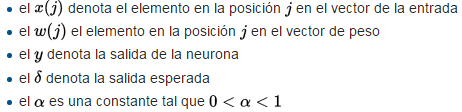
La red tipo Perceptrón fue inventada por el psicólogo Frank Rosenblatt en el año 1957. Su intención era ilustrar algunas propiedades fundamentales de los sistemas inteligentes en general, sin entrar en mayores detalles con respecto a condiciones específicas y desconocidas para organismos biológicos concretos.

Rosenblatt creía que la conectividad existente en las redes biológicas tiene un elevado porcentaje de aleatoriedad, por lo que se oponía al análisis de McCulloch Pitts en el cual se empleaba lógica simbólica para analizar estructuras bastante idealizadas. Rosenblatt opinaba que la herramienta de análisis más apropiada era la teoría de probabilidades, y esto lo llevó a una teoría de separabilidad estadística que utilizaba para caracterizar las propiedades más visibles de estas redes de interconexión ligeramente aleatorias.

**Fórmulas matemáticas**

En el perceptrón, existen dos tipos de aprendizaje, el primero utiliza una tasa de aprendizaje mientras que el segundo no la utiliza. Esta tasa de aprendizaje amortigua el cambio de los valores de los pesos.1

El algoritmo de aprendizaje es el mismo para todas las neuronas, todo lo que sigue se aplica a una sola neurona en el aislamiento. Se definen algunas variables primero:



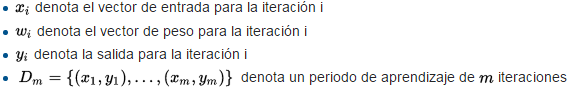
Los dos tipos de aprendizaje difieren en este paso. Para el primer tipo de aprendizaje, utilizando tasa de aprendizaje, utilizaremos la siguiente regla de actualización de los pesos:



Para el segundo tipo de aprendizaje, sin utilizar tasa de aprendizaje, la regla de actualización de los pesos será la siguiente:



Por lo cual, el aprendizaje es modelado como la actualización del vector de peso después de cada iteración, lo cual sólo tendrá lugar si la salida y difiere de la salida deseada. Para considerar una neurona al interactuar en múltiples iteraciones debemos definir algunas variables más:



En cada iteración el vector de peso es actualizado como sigue:



El periodo de aprendizaje  se dice que es separable linealmente si existe un valor positivo y un vector de peso tal que: 

Novikoff (1962) probo que el algoritmo de aprendizaje converge después de un número finito de iteraciones si los datos son separables linealmente y el número de errores está limitado a:



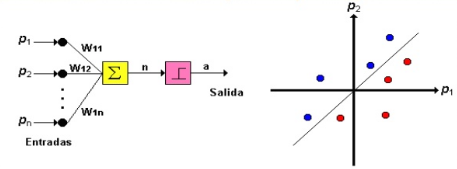
**Aprendizaje de supervisado:** entrada, salida, objeto.

**Aprendizaje no supervisado:** entrada y salida.

**Aprendizaje por reforzamiento:** recompensa/castigado.

**Estructura del Perceptron**

La neurona de salida realiza la suma ponderada de las entradas, suma el bias y pasa el resultado a una función de transferencia de tipo hardlim(s). La regla e decisión es responder +1 si el patrón presentado pertenece a la clase A, o 0(–1) si el patrón pertenece a la clase B.



El perceptrón simple sólo sirve para clasificar problemas linealmente separables, cosa que ya se podía hacer mediante métodos estadísticos, y de una forma mucho más eficiente. Únicamente para problemas linealmente separables y que sean de dos clases.

Las funciones de activación son de tipo sigmoide (puede ser lineal en la capa de salida).

1. **Historia de Adaline**

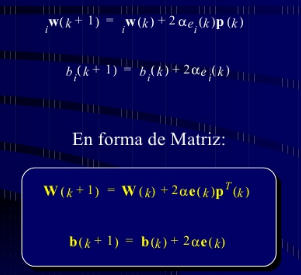
La Red Adaline fue desarrollada en el 1960 por Bernard Widrow y su estudiante Marcian Hoff de la universidad de Stanford.

ADALINE proviene de Adaptive Lineal Element (Elemento Lineal Adaptativo), pero antes de que se le diera este nombre esta red sufrió un cambio ya que primeramente se llamaba Adaptive Lineal Neuron (Neurona Linear Adaptiva), dicho cambio se dio por que la Red Adaline es un dispositivo que consta de un único elemento de procesamiento, como tal no es técnicamente considerada una red neuronal.

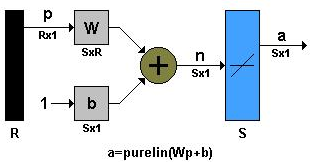
Adaline fue desarrollada para el reconocimiento de patrones binarios, por ejemplo predecir el siguiente bit en una línea telefónica.

Formada por un elemento denominado: Combinador Adaptativo Lineal (ALC). La salida lineal obtenida del ALC se aplica a un Conmutador Bipolar. El Umbral de la F. de T. se representa a través de una conexión ficticia de peso Wo (b)

**Ecuaciones**



**Estructura de adaline**



**En donde:**

p: Patrones de entrada

b: Umbrales de activación

a: Salida de la neurona

La salida de la red está dada por:



1. Que es la retropropagacion

La retropropagación consiste en propagar el error hacia atrás, es decir, de la capa de salida hacia la capa de entrada, pasando por las capas ocultas intermedias y ajustando los pesos de las conexiones con el fin de reducir dicho error. Hay distintas versiones o reglas del algoritmo de retropropagación y distintos arquitecturas conexionistas a las que pueden ser aplicados.

Durante mucho tiempo no se dispuso de algoritmos para entrenar redes multicapa, y como las redes de una capa estaban muy limitadas en cuanto a lo que eran capaces de representar, el campo de las redes neuronales artificiales estaba estancado. La invención y perfeccionamiento del algoritmo de retropropagación dio un gran impulso al desarrollo de este campo. Tiene un buen fundamento matemático y a pesar de sus limitaciones ha expandido enormemente el rango de problemas donde se aplican las redes neuronales artificiales.

El algoritmo emplea un ciclo propagación – adaptación de dos fases. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo las neuronas de la capa oculta solo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total.

La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento.

1. **Algoritmo backpropagation**

Paso 1. Inicializar los pesos de la red con valores pequeños aleatorios.

Paso 2. Presentar un patrón de entrada y especificar la salida deseada que debe generar la red.

Paso 3. Calcular la salida actual de la red. Para ello se presentan las entradas a la red y se calcula la salida de cada capa hasta llegar a la capa de salida, ésta será la salida de la red. Los pasos son los siguientes:

Se calculan las entradas netas para las neuronas ocultas procedentes de las neuronas de entrada.

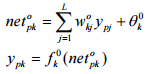
Para una neurona j oculta:



En donde el índice h se refiere a magnitudes de la capa oculta; el subíndice p, al p-ésimo vector de entrenamiento, y j a la j-ésima neurona oculta. El término θ puede ser opcional, pues actúa como una entrada más.

Se calculan las salidas de las neuronas ocultas

Se realizan los mismos cálculos para obtener las salidas de las neuronas de salida:



Paso 4 . Calcular los términos de error para todas las neuronas. Si la neurona k es una neurona de la capa de salida, el valor de la delta es:



La función f debe ser derivable. En general disponemos de dos formas de función de salida:

La función lineal:



La función sigmoidal: 

1. **Teorema de Kolmogorov**

Es un resultado de sistemas dinámicos sobre la persistencia de movimientos cuasiperiódicos. Este teorema resuelve parcialmente el problema de los divisores pequeños (que origina problemas de convergencia en sistemas con múltiples frecuencias). El teorema explica cómo se modifica el aspecto de las trayectorias de un sistema integrable bajo pequeñas perturbaciones.

Dada una sucesión de variables aleatorias independientes  con medias  y varianza estableciéndose



Se cumple que existe ley fuerte de los grandes números; así

