Deep Learning

Abril de 2022

Germán Alonso Lascurain

germanalonso@opendeusto.es

Índice de contenidos

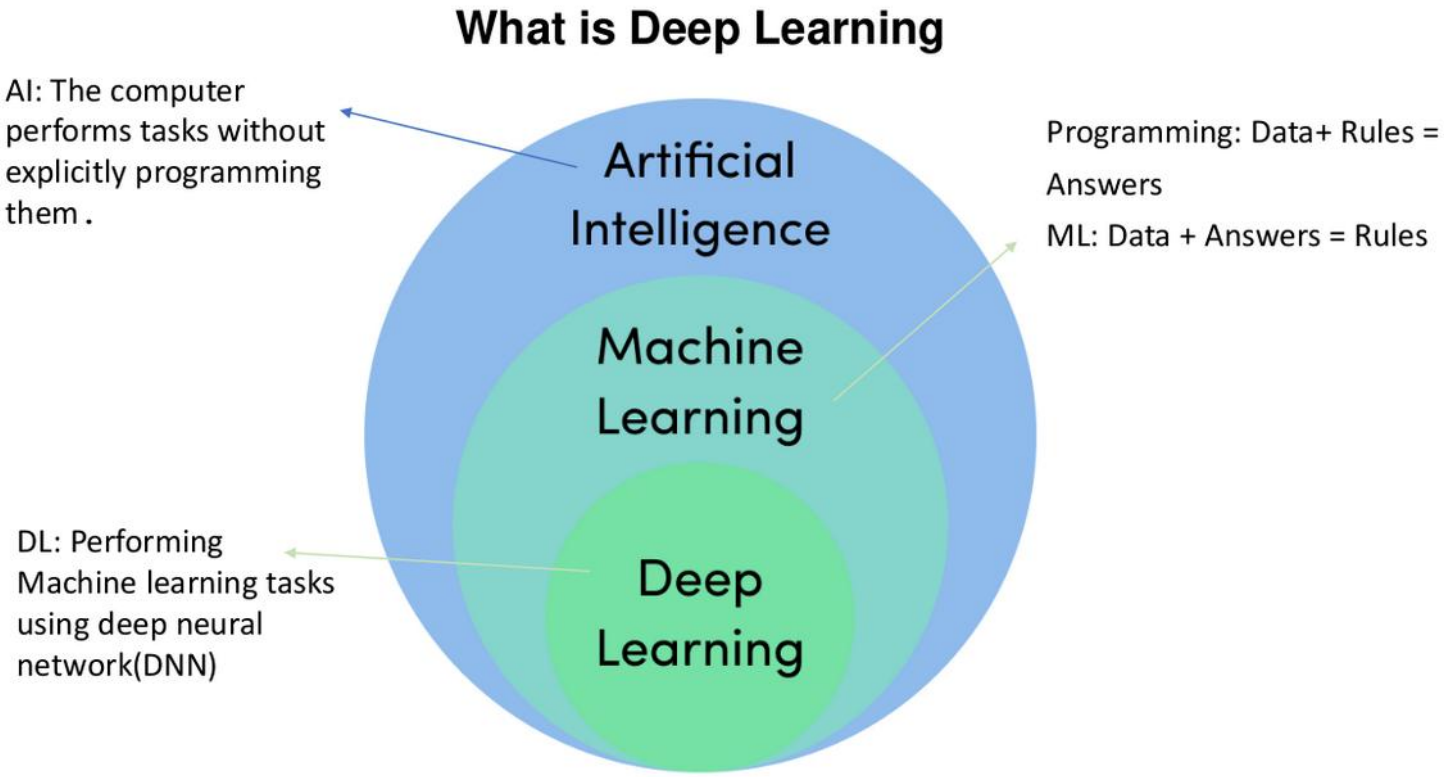
1. Introducción

Durante décadas los científicos han perseguido la construcción de algoritmos capaces de procesar la información de la misma manera que el cerebro.

Tras años de desarrollo se han desarrollado diferentes estructuras de redes neuronales artificiales.

Las redes neuronales, poseen propiedades que les permiten aprender a partir de ejemplos.

1. Introducción



1. Introducción

La historia de las redes neuronales puede resumirse así:

- 1936 - Alan Turing. Fue el primero en estudiar el cerebro como una forma de ver el mundo de la computación.

- Los primeros teóricos que concibieron los fundamentos de la computación neuronal fueron Warren McCulloch, un neurofisiólogo, 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 Nerviosa - Boletín de Matemática Biofísica 5: 115-133). 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, **éste 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.

1. Introducción

- 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 finito (Teorema de Convergencia del Perceptron).

- 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.

1. Introducción

- 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 como el 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.”

1. Introducción

- 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 (sobre todo para simulaciiones).

1. Introducción

El cerebro humano, por naturaleza

- Es capaz de reconocer patrones
- Puede hacer asociaciones entre conceptos
- Puede manejar información de gran complejidad
- Es tolerante al ruido

Un ordenador

- Tiene una gran capacidad de cálculo
- Puede hacer cálculos extremadamente precisos
- Puede implementar lógica

1. Introducción

Los seres humanos y los ordenadores son intrínsecamente adecuados para **diferentes** tipos de tareas.

Por ejemplo, el cálculo de la raíz cúbica de un número grande es muy fácil para un ordenador, pero es extremadamente dificil para los humanos. En cambio, una tarea como el reconocimiento de los objetos de una imagen es un asunto sencillo para un humano, pero tradicionalmente ha sido muy difícil para un algoritmo de algoritmo de aprendizaje automatizado.

Solo en los últimos años el aprendizaje profundo (deep learning) ha mostrado una precisión en algunas de estas tareas que supera la de un humano.

El rendimiento superior de las redes neuronales profundas refleja el hecho de que las redes neuronales biológicas obtienen gran parte de su potencia de la **profundidad.**

Las redes biológicas están conectadas de maneras que no comprendemos del todo. En los pocos casos en los que la estructura biológica se entiende a cierto nivel, se han logrado avances significativos diseñando redes neuronales artificiales siguiendo de esas líneas.

1. Introducción

Un ejemplo clásico de este tipo de arquitectura es el uso de la red neuronal convolucional para el reconocimiento de imágenes. Esta arquitectura se inspiró en los experimentos de Hubel y Wiesel experimentos de 1959 sobre la organización de las neuronas en la corteza visual del gato.

2. Motivación, aplicaciones y ventajas de una RNA

- **Objetivo**: Usar los principios de organización del cerebro para construir sistemas inteligentes.
  - RNA (redes neuronales artificiales) ➔ Emulación (modelo matemático) del funcionamiento del cerebro a bajo nivel.

- **Cerebro/RNAs**
  - Sistemas masivamente paralelos formados por un gran número de elementos simples (neuronas) interconectados

2. Motivación, aplicaciones y ventajas de una RNA

- **Aplicaciones**:
  Las RNA se usan principalmente en dominios difíciles con grandes necesidades de aprendizaje:
  - Biología:
    - Aprender más acerca del cerebro y otros sistemas.
    - Obtención de modelos de la retina.

  - Empresa:
    - Evaluación de probabilidad de formaciones geológicas y petrolíferas.
    - Identificación de candidatos para posiciones específicas.
    - Explotación de bases de datos.
    - Optimización de plazas y horarios en líneas de vuelo.
    - Optimización del flujo del tránsito controlando convenientemente la temporización de los semáforos.
    - Reconocimiento de caracteres escritos.
    - Modelado de sistemas para automatización y control.

2. Motivación, aplicaciones y ventajas de una RNA

- Medio ambiente:
  - Analizar tendencias y patrones.
  - Previsión del tiempo.

- Finanzas:
  - Previsión de la evolución de los precios.
  - Valoración del riesgo de los créditos.
  - Identificación de falsificaciones.
  - Interpretación de firmas.

2. Motivación, aplicaciones y ventajas de una RNA

- Manufacturación:
  - Robots automatizados y sistemas de control (visión artificial y sensores de presión, temperatura, gas, etc.).
  - Control de producción en líneas de procesos.
  - Inspección de la calidad.

- Medicina:
  - Analizadores del habla para ayudar en la audición de sordos profundos.
  - Diagnóstico y tratamiento a partir de síntomas y/o de datos analíticos (electrocardiograma, encefalogramas, análisis sanguíneo, etc.).
  - Monitorización en cirugías.
  - Predicción de reacciones adversas en los medicamentos.
  - Entendimiento de la causa de los ataques cardíacos.

2. Motivación, aplicaciones y ventajas de una RNA

- Militares:
  - Clasificación de las señales de radar.
  - Creación de armas inteligentes.
  - Optimización del uso de recursos escasos.
  - Reconocimiento y seguimiento en el tiro al blanco.

- La mayoría de estas aplicaciones consisten en realizar un reconocimiento de patrones, como ser: buscar un patrón en una serie de ejemplos, clasificar patrones, completar una señal a partir de valores parciales o reconstruir el patrón correcto partiendo de uno distorsionado.

- Sin embargo, está creciendo el uso de redes neuronales en distintos tipos de sistemas de control.
- Desde el punto de vista de los casos de aplicación, la ventaja de las redes neuronales reside en el procesado paralelo, adaptativo y no lineal.

- El dominio de aplicación de las redes neuronales también se lo puede clasificar de la siguiente forma: asociación y clasificación, regeneración de patrones, regresión y generalización, y optimización.

2. Motivación, aplicaciones y ventajas de una RNA

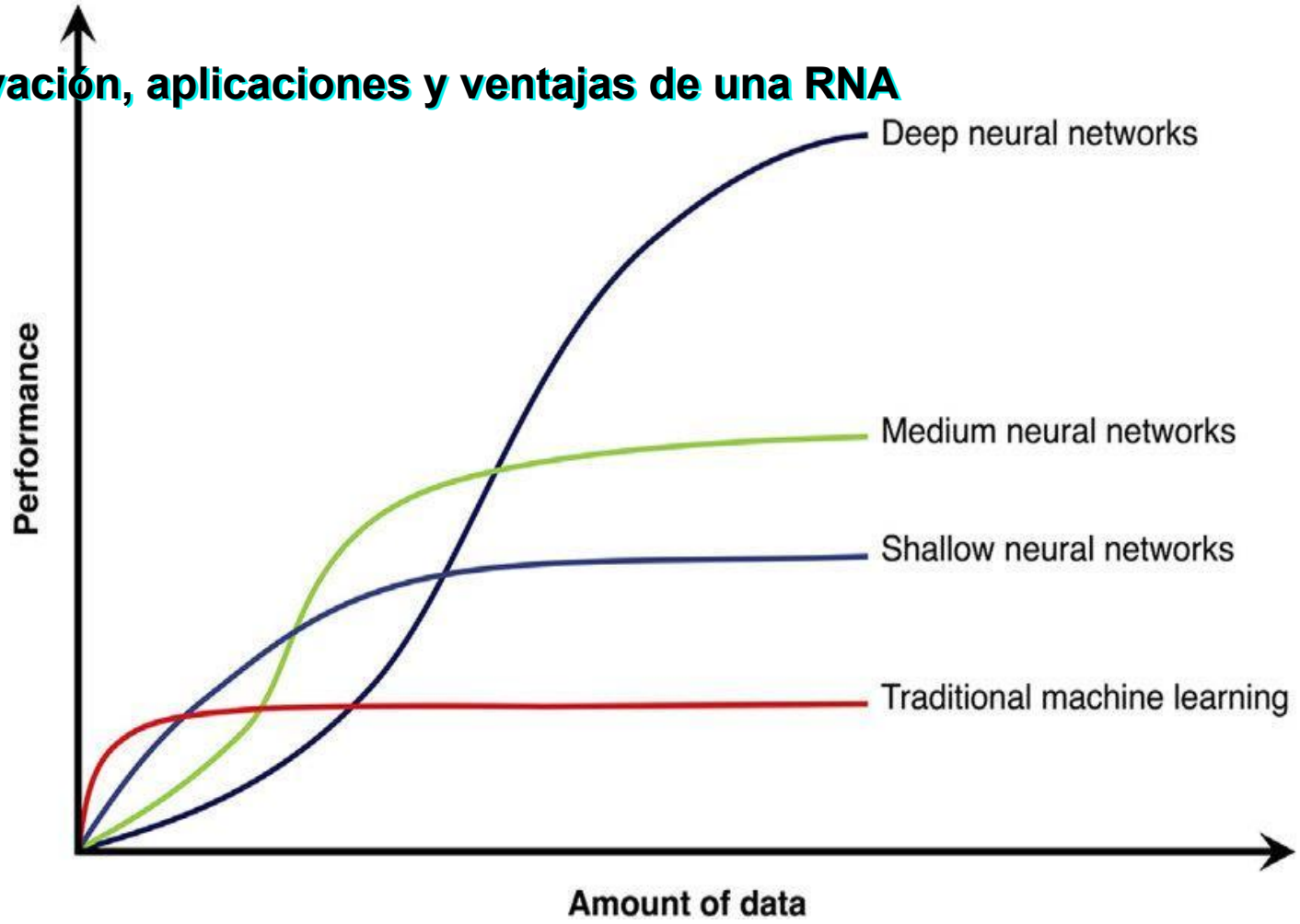
- Ventajas:

  Las RNA presentan un gran número de características semejantes a las del cerebro.

- Son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc.

  - **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

2. Motivación, aplicaciones y ventajas de una RNA

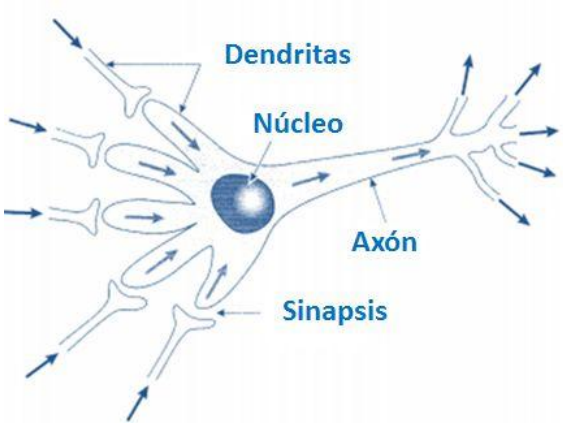


3. Neurona

- Una **neurona** es una célula especializada del tejido nervioso que asegura la conducción y la transmisión del influjo nervioso.

    - El ser humano tiene entre 20.000 millones y 200.000 millones
    - Cada una con hasta 30.000 conexiones con otras

- Es la **base del procesamiento del cerebro humano.**



3. Neurona

- Funcionamiento
  - Neurona ”acumula” todos los potenciales positivos y negativos que recibe en sus entradas (dendritas)
  - Si la suma de esos impulsos es ”suficiente”, cambia su potencia y genera su salida en el axón que se propagará.

- Aprendizaje
  - La intensidad de las sinapsis (conexiones) puede modificarse.
  - Conexiones más o menos fuertes.
  - Permite modificar comportamiento de la neurona para adaptarse a nuevas situaciones (aprendizaje).
  - También es posible modificar el comportamiento mediante creación/destrucción de sinapsis y muerte de neuronas.

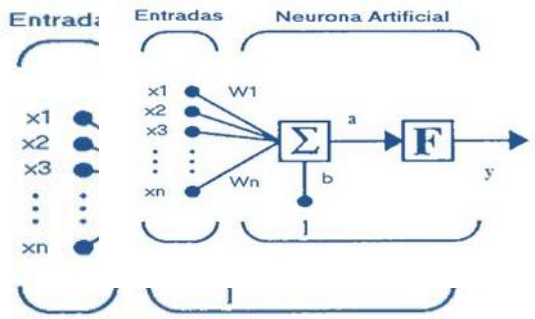
3. Neurona

- Nacemos con una estructura neuronal que:
  - Es flexible y permite crear y destruir neuronas.
  - Durante la etapa de crecimiento, el aprendizaje permite desarrollar nuevas uniones y destruir otras.
  - La estructura neuronal cambia a lo largo de la vida.
  - Estos cambios consisten en **reforzamiento o debilitamiento de conexiones.**

- El de ser capaz de memorizar la cara de una persona que nos presentan, consiste en alterar “varias” sinapsis de nuestra red neuronal.

- El cerebro humano es muy flexible y plástico permitiendo la alteración de dichas sinapsis en casos de daño grave.

4. Perceptrón (neurona artificial)



- Funcionamiento
  - Cada neurona recibe diferentes entradas (x1 , x2 , x3 ,...)
  - Cada señal se multiplica por un valor, denominado peso, que da una idea de la fuerza de esa conexión (w1 , w2 , w3 ,...)
  - Se calcula una salida, por una función de Transferencia (F)
    - **Función de activación:**

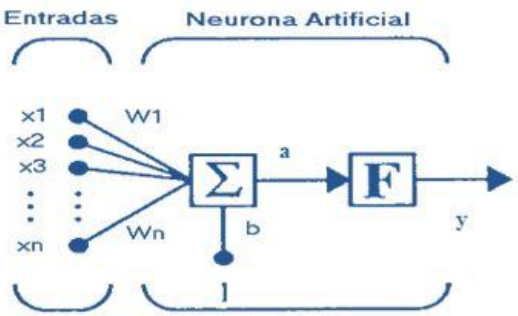
      a = x1*w1 + x2*w2 + ... + xn*wn + b

    - **Función de transferencia**

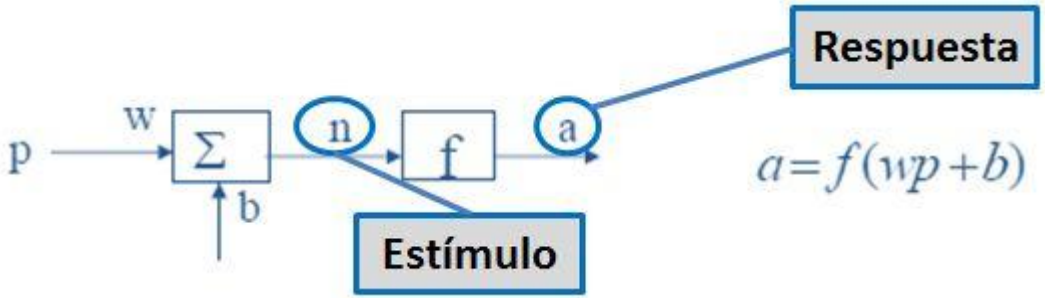
      y = F(a) = F(x1*w1 + x2*w2 + ... + xn*wn + b)

    - **Bias o polarización**: entrada constante de magnitud 1, y peso b que se introduce en el sumador
- ¿Os recuerda a algo?

4. Perceptrón (neurona artificial)



- La entrada (p) se multiplica por el peso (w)
- El resultado se multiplica por un sesgo (b)
- La salida del sumatorio (n) se conoce como estímulo
- El estímulo va a la función de transferencia (f), la cual produce una salida (a) de la neurona



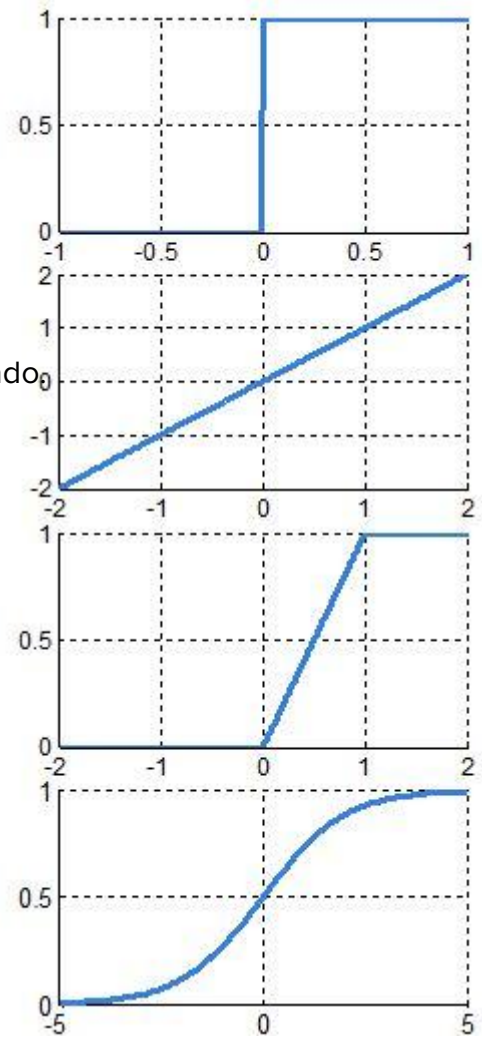
5. Funciones de transferencia

- Una neurona biológica puede estar activa (excitada) o inactiva (no excitada); es decir, que tiene un “ estado de activación ”. Las neuronas artificiales también tienen diferentes estados de activación; algunas de ellas solamente dos, al igual que las biológicas, pero otras pueden tomar cualquier valor dentro de un conjunto determinado.

- Existen multitud de funciones de transferencia.
- Cada función de transferencia sirve para un propósito o tipo de aprendizaje.

- Las más comunes:

  - Umbral: $a = \begin{cases} 0, si\ n < 0 \\ 1, si\ n \geq 0 \end{cases}$
  - Lineal: $a = n$
  - Lineal saturada: $a = \begin{cases} 0, si\ n < 0 \\ n, si\ 0 \leq n < 1 \\ 1, si\ n \geq 1 \end{cases}$
  - Sigmoide: $a = \frac{1}{1+e^{-n}}$



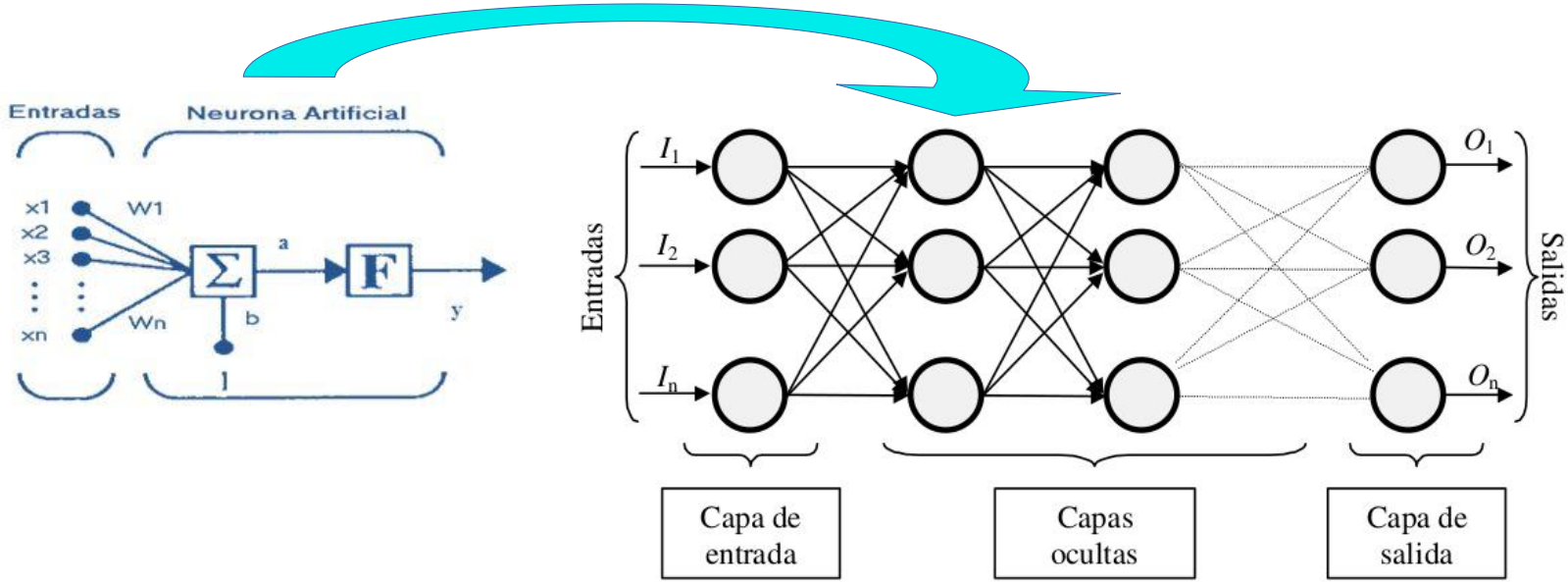
5. Funciones de transferencia

- Otras funciones de transferencia

FUNCTION	RANGE	FUNCTION OF NET INPUT g
Identity	$(-\infty, +\infty)$	g
Exponential	$(0, \infty)$	$\exp(g)$
Reciprocal	$(0, \infty)$	$1/g$
Square	$[0, +\infty)$	g^2
Logistic	$(0, 1)$	$\frac{1}{1+\exp(-g)}$
Softmax	$(0, 1)$	$\frac{\exp(g)}{\sum exponentials}$
Gauss	$(0, 1]$	$\exp(-g^2)$
Sine	$[-1, 1]$	$\sin(g)$
Cosine	$[-1, 1]$	$\cos(g)$
Elliott	$(0, 1)$	$\frac{g}{1+\lvert g\rvert}$
Tanh	$(-1, 1)$	$\tanh(g) = 1 - \frac{2}{1+\exp(2g)}$
Arctan	$(-1, 1)$	$\frac{2}{\pi}\arctan(g)$

6. Elementos básicos de una RNA

- Una red neuronal no es más que un conjunto de neuronas (normalmente) organizadas en capas
  - **Capa de entrada**: recibe los atributos de los datos.
  - **Capas ocultas**: son capas intermedias que permiten a la red aprender las relaciones entre los datos.
  - **Capa de salida**: es la capa final, que **genera la salida del sistema a partir de la información recibida.**



6. Elementos básicos de una RNA

- Capa de salida

- La capa de salida es la que nos devuelve la respuesta del sistema, a partir de la información recibida.

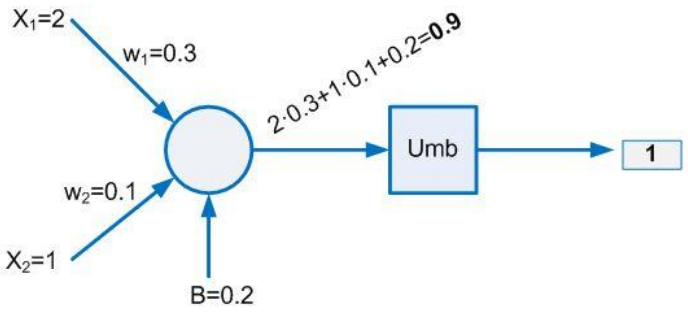
- Para problemas de **regresión**, usaremos neuronas con funciones lineales o sigmoidales.

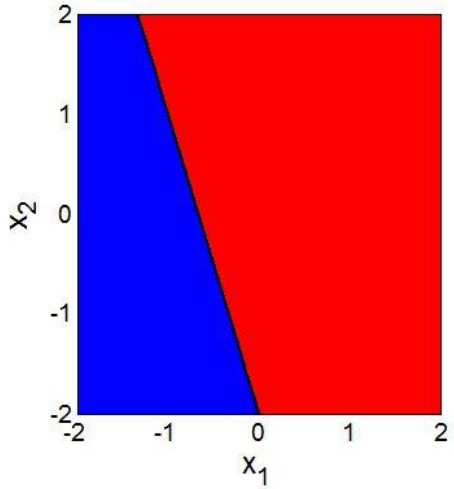
- Para problemas de **clasificación binarios**, podemos usar **una neurona** con función de activación umbral, o función de activación “contínua” (probabilidad).

- Para problemas de **clasificación multiclase**, podemos usar **tantas neuronas como clases.**

7. Funcionamiento de una neurona (perceptrón)

- Usaremos un modelo con una neurona y con una función de activación umbral que multiplica las entradas por los pesos.





- Evaluación

$$- \ a = Umb(b + \sum w \cdot x) = \begin{cases} 1, si\ (0{,}2 + 0{,}3 \cdot x_1 + 0{,}1 \cdot x_2) \geq 0 \\ -1, si\ (0{,}2 + 0{,}3 \cdot x_1 + 0{,}1 \cdot x_2) < 0 \end{cases}$$

8. Clasificadores lineales básicos (AND, OR, NOT)

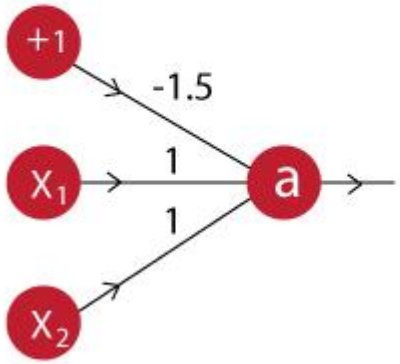
Con un único perceptrón, podemos implementar clasificadores lineáles básicos del tipo AND, OR, NOT

- AND

La salida de la neurona en este caso es a = f(-1.5 + x1 + x2)

X1	X2	X1 AND X2	(-1.5+X1+X2)	a
0	0	0	-1.5	0
0	1	0	-0.5	0
1	0	0	-0.5	0
1	1	1	0.5	1

El Bias es de 1,5. Vemos como la salida a coincide con la operación lógica X1 and X2

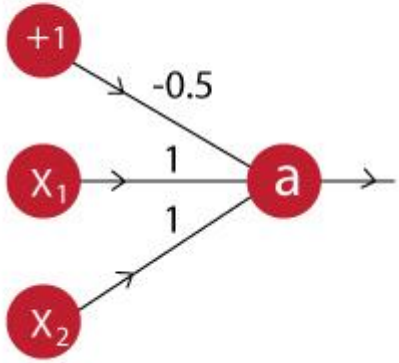


8. Clasificadores lineales básicos (AND, OR, NOT)

- OR

La salida de la neurona en este caso es a = f(-0.5 + x1 + x2)

X1	X2	X1 OR X2	(-0.5+X1+X2)	a
0	0	0	-0.5	0
0	1	1	0.5	1
1	0	1	0.5	1
1	1	1	1.5	1



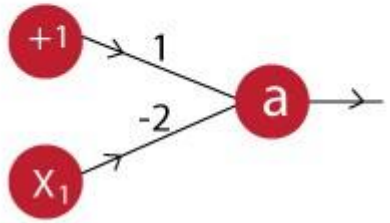
El Bias es de 0,5. Vemos como la salida a coincide con la operación lógica X1 OR X2, sólo hemos modificado el valor del Bias.

8. Clasificadores lineales básicos (AND, OR, NOT)

- NOT

La salida de la neurona en este caso es a = f(1 – 2 * x1)

X1	NOT X1	(1-2*X1)	a
0	1	1	1
1	0	-1	0



Generalmente para estos casos se usa una función de activación logística, puesto que es diferenciable y nos permite el uso del método backpropagation.

8. Clasificadores lineales básicos (AND, OR, NOT)

- https://www.analyticsvidhya.com/blog/2016/03/introduction-deep-learning-fundamentals-neural-networks.../

9. Mecanismos de aprendizaje

- Biológicamente se acepta que la información memorizada en el cerebro se relaciona con los valores de las conexiones.

- En las RNA:
  - El objetivo es obtener una salida a partir de unos inputs.
  - Se considera que el **conocimiento** se encuentra representado en los **pesos de las conexiones**.
  - El **proceso de aprendizaje** se basa en **cambios en estos pesos**.
  - Durante el proceso de aprendizaje, la topología de la red y funciones de cada neurona (entrada, activación y salida), no cambia, pero SÍ los pesos.

- Los cambios en el proceso de aprendizaje se reducen a **destrucción, modificación y creación de conexiones entre las neuronas.**
  - La creación de una conexión implica que el peso de la misma pasa a tener un valor distinto de cero.
  - Una conexión se destruye cuando su valor pasa a ser cero.

9. Mecanismos de aprendizaje

- 2 métodos de aprendizaje:
  1) Aprendizaje supervisado.
     i. Aprendizaje por corrección de error.

        Consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los obtenidos a la salida de la red, es decir, en función del error cometido en la salida. La generalización de este método de aprendizaje es el backpropagation.

     ii. Apredizaje por refuerzo.

        Se trata de un aprendizaje supervisado, más lento que el anterior, que se basa en la idea de no disponer de un ejemplo completo del comportamiento deseado, es decir, de no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada. Aquí la función del supervisor se reduce a indicar si la salida obtenda se ajusta a la deseada (éxito = 1, fracaso = -1) y luego se ajustan los pesos basándose en un mecanismo de probabilidades.

     iii. Aprendizaje estocástico.

        Consiste básicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

9. Mecanismos de aprendizaje

2) Aprendizaje no supervisado

   Las redes con aprendizaje no supervisado (también conocido como autosupervisado) no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta.

10. Aprendizaje del perceptrón

- Supongamos una entrada con valor positivo:
  - Si la neurona responde -1, y debiera responder con 1 ➔ Debemos incrementar el peso correspondiente
  - Si la neurona responde con 1, y debiera responder con -1 ➔ Se debe decrementar el peso correspondiente

- Si la entrada es negativa, razonamiento inverso
  - Si la neurona responde -1, y debiera responder con 1 ➔ Debemos decrementar el peso correspondiente
  - Si la neurona responde con 1, y debiera responder con -1 ➔ Se debe incrementar el peso correspondiente

10. Aprendizaje del perceptrón

- La respuesta de la red neuronal se compara con la respuesta deseada y el error cometido se utiliza para modificar los pesos

$$w_i = w_i + \Delta w_i$$
$$\Delta w_i = \alpha \cdot x_i (d_k - y_k)$$

donde:

- α es una constante positiva, usualmente pequeña (0,1) llamada factor de aprendizaje que modera las actualizaciones de los pesos.
  - En cada iteración, si d=1 y y=0, entonces (d−y>0), y por tanto los w_i correspondientes a x_i positivos aumentarán (y disminuirán los correspondientes a x_i negativos)
  - Análogamente ocurre si es y=1 e d = 0
- Cuando y = d, los w_i no se modifican

10. Aprendizaje del perceptrón

- Teorema:
  - El algoritmo anterior converge en un número finito de pasos a un vector de pesos que clasifica correctamente todos los ejemplos de entrenamiento, siempre que éstos sean **linealmente separables** y suficientemente pequeño (Minsky and Papert, 1969).

- Por tanto,
  - En el caso de conjuntos de entrenamiento linealmente separables, el resultado será perfecto, con un perceptrón.
  - No hay manera “buena” de comprobar si un problema es linealmente separable.

10. Aprendizaje del perceptrón

- Teorema:
  - El algoritmo anterior converge en un número finito de pasos a un vector de pesos que clasifica correctamente todos los ejemplos de entrenamiento, siempre que éstos sean **linealmente separables** y suficientemente pequeño (Minsky and Papert, 1969).

- Por tanto,
  - En el caso de conjuntos de entrenamiento linealmente separables, el resultado será perfecto, con un perceptrón.
  - No hay manera “buena” de comprobar si un problema es linealmente separable.

11. Modelos multicapa

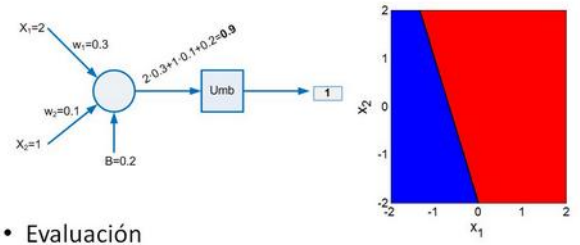
- ¿Porqué deberíamos usar modelso con más de una capa?
  ➜ El perceptrón nos permite resolver sólo problemas lineales y ésto es un problema porque en la vida real y en los problemas reales, no sucede casi nunca.
  ➜ Básicamente corta los datos con una línea recta.

  - El algoritmo anterior converge en un número finito de pasos a un vector de pesos que clasifica correctamente todos los ejemplos de entrenamiento, siempre que éstos sean **linealmente separables** y suficientemente pequeño (Minsky and Papert, 1969).

- Por tanto,
  - En el caso de conjuntos de entrenamiento linealmente separables, el resultado será perfecto, con un perceptrón.
  - No hay manera “buena” de comprobar si un problema es linealmente separable.

11. Modelos multicapa

- ¿Porqué deberíamos usar modelso con más de una capa?
  ➜ El perceptrón nos permite resolver sólo problemas lineales y ésto es un problema porque en la vida real y en los problemas reales, no sucede casi nunca.
  ➜ Básicamente corta los datos con una línea recta.



- Evaluación

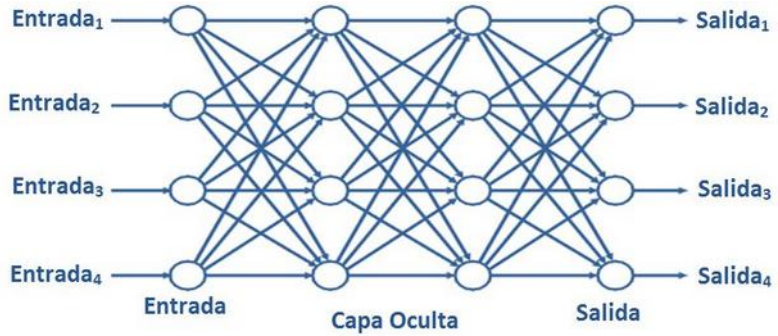
$$- \ a = Umb(b + \sum w \cdot x) = \begin{cases} 1, si\ (0{,}2 + 0{,}3 \cdot x_1 + 0{,}1 \cdot x_2) \geq 0 \\ -1, si\ (0{,}2 + 0{,}3 \cdot x_1 + 0{,}1 \cdot x_2) < 0 \end{cases}$$

11. Modelos multicapa

Las redes multicapas son aquellas que disponen de un conjunto de neuronas agrupadas en varios (2, 3, etc.) niveles o capas.

En estos casos, una forma para distinguir la capa a la que pertenece una neurona, consistiría en fijarse en el origen de las señales que recibe a la entrada y el destino de la señal de salida.

Normalmente, todas las neuronas de una capa reciben señales de entrada desde otra capa anterior (la cual está más cerca a la entrada de la red), y envían señales de salida a una capa posterior (que está más cerca a la salida de la red). **A estas conexiones se las denomina conexiones hacia adelante o feedforward.**



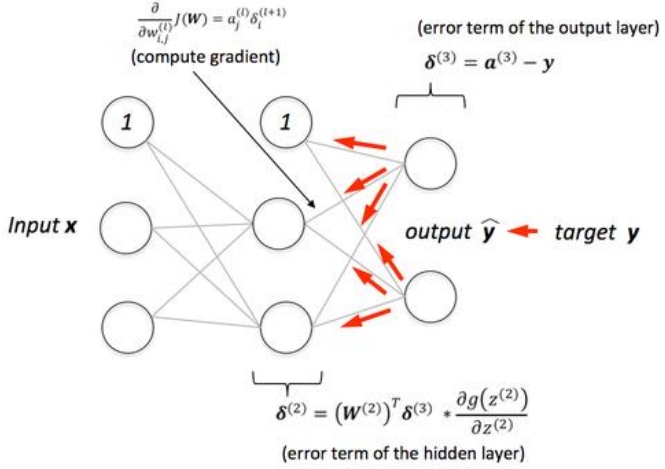
11. Modelos multicapa

La activación se propaga a través de los pesos desde la capa de entrada hacia las intermedias.

- Aprendizaje, se actualizan 2 conjuntos de pesos:
  - Aquellos entre la capa intermedia y la de salida, y
  - Aquellos entre la capa de entrada y la capa intermedia.

- El error debido al primer conjunto de pesos se calcula empleando el método anteriormente descrito.

- Entonces se propaga hacia atrás la parte del error debido a los errores que tienen lugar en el segundo conjunto de pesos y se asigna el error proporcional a los pesos que lo causan.



11. Modelos multicapa

- Podemos utilizar cualquier número de capas ocultas

- El método es bastante general
  - El tiempo de entrenamiento puede ser excesivo para arquitecturas con muchas capas (Deep Learning)
  - El perceptrón multicapa es el modelo más utilizado

- En las capas ocultas se suelent utilizar funciones de transferencia sigmoideas, normalmente.

12. Aprendizaje de redes de varias capas

- Backpropagation:
  - La red empieza fijando pesos de forma aleatoria.
  - Proceso iterativo hasta alcanzar un criterio de parada.

- Ciclos (epochs) de dos procesos:
  - Fase forward: calcular la salida de la red.
    - Neuronas activadas en secuencia desde la capa de inputs a la outputs.
    - Aplicar a cada neurona sus pesos y función de activación.
    - Al llegar a la capa final se produce una señal de output.

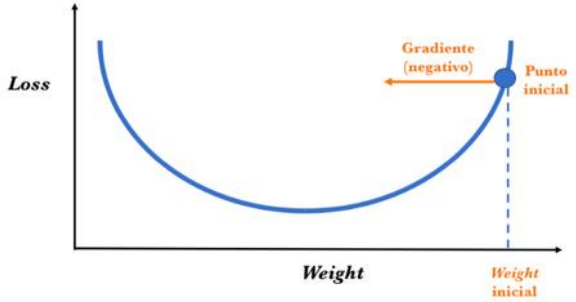
  - Fase backward: calcular el error y reajustar pesos
    - Comparar la diferencia entre la señal de output y los valores reales.
    - Se propaga el error hacia atrás para recalcular los pesos entre las neuronas.

13. Descenso del gradiente

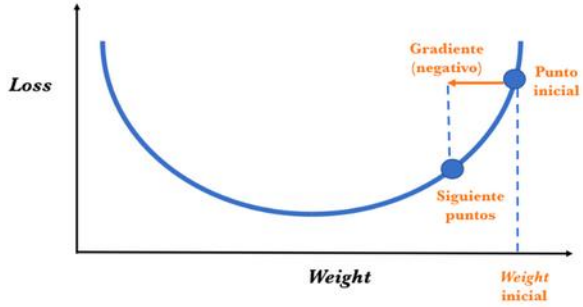
- Back propagation se basa en el descenceso del gradiente para alcanzar el valor óptimo de los pesos de una RNA
- Es un proceso de optimización que busca minimizar la pérdida en nuestra RNA
- *Gradient descent* usa la primera derivada (gradiente) de la función de loss cuando realiza la actualización en los parámetros.
- Por ello, es necesario que las funciones de activación sean “derivables”
- El gradiente nos da la pendiente de una función en un punto. El gradiente, siempre apunta en el sentido en el que se incrementa el valor de la función loss.
- Si tomamos el negativo del gradiente, estaremos buscando aquel punto que minimiza el *loss*

13. Descenso del gradiente

- Supongamos que la línea representa los valores que toma la función loss para cada posible parámetro y el punto (inicial) indicado muestra el negativo del gradiente.

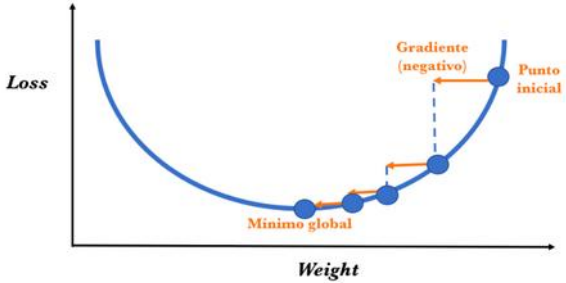


- Para determinar el siguiente valor para el parámetro (para simplificar la explicación, consideremos solo el peso/weight), el algoritmo de gradient descent modifica el valor del peso inicial para ir en sentido contrario al del gradiente (ya que este apunta en el sentido en que crece la loss y queremos reducirla), añadiendo una cantidad proporcional a este. La magnitud de este cambio está determinado por el valor del gradiente y por un hiperparámetro *learning rate*. Por lo tanto, conceptualmente es como si siguiéramos la dirección de la pendiente cuesta abajo hasta que alcanzamos un mínimo local.



13. Descenso del gradiente

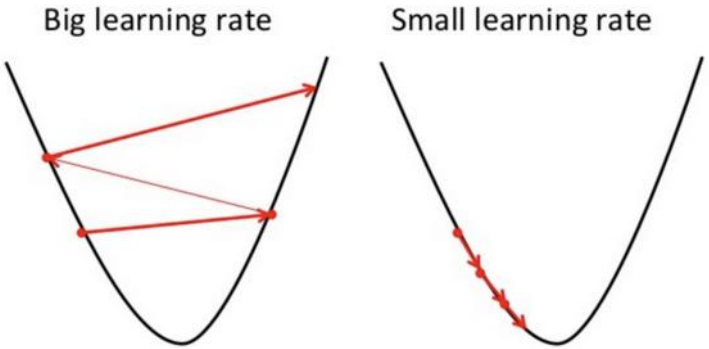
- El algoritmo gradient descent repite este proceso acercándose cada vez más al mínimo hasta que el valor del parámetro llega a un punto más allá del cual no puede disminuir la función de loss:



13. Descenso del gradiente

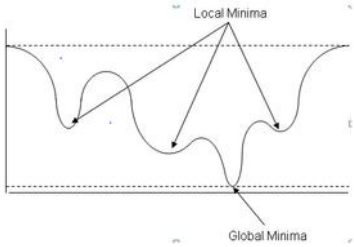
- Uno de los problemas del algoritmo es que si utilizamos un ratio de aprendizaje muy grande, podemos “saltarnos” el mínimo local.

- Si el ratio es muy pequeño, pero no iteramos lo sufiente, podemos no llegar al mínimo.



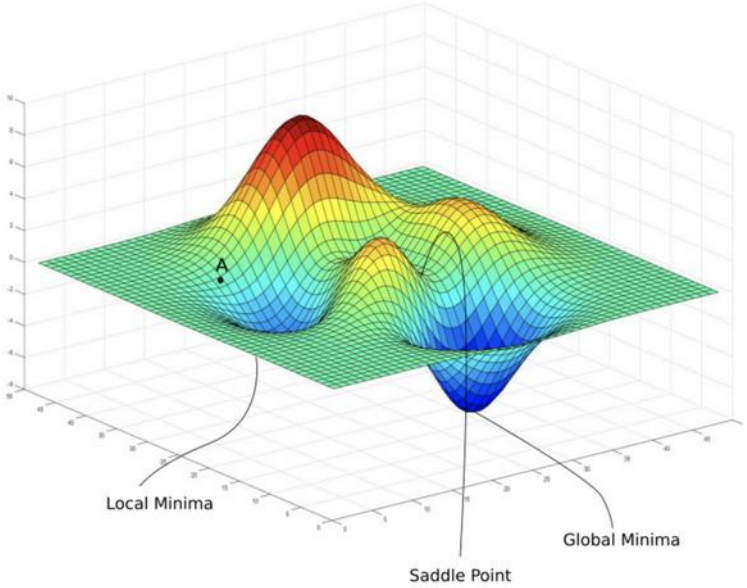
13. Descenso del gradiente

- También puede ocurrir (y ocurre), que caigamos en un mínimo local que no es el global.



- Solución (parcial):

1. Regularizar nuestros datos, bien con L1 (Lasso) o L2 (Ridge).
2. Añadir un “momentum” (inercia) que consiste básicamente en añadir una fracción de los pesos obtenidos previamente a los nuevos. Es un parámetro existente en la librería.



14. Número de capas ocultas

- Para muchos conjuntos de datos, una sola capa puede ser suficiente.
- No hay una teoría que nos permita optimizar el número de capas ocultas necesarias.
- Teorema

  “It has been proven that a neural network with at least one hidden layer of sufficient neurons is a universal function approximator. This means that neural networks can be used to approximate any continuous function to an arbitrary precision over a finite interval”

15. Número de neuronas por capa

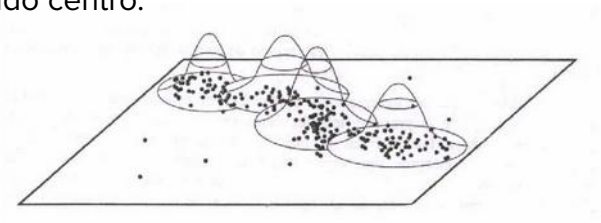
- Entrenar varias redes con distinto número de neuronas ocultas y estimar el error generalizado de cada red.
  - Procedimiento simple: empezar sin neuronas ocultas y añadir una neurona cada vez.
  - Calcular el error de generalización de cada red.
  - Dejar de añadir neuronas si aumenta la generalización del error (overfitting)

- Para una red con 2 capas ocultas
  - ¿Funcionará mejor con 1 neurona en la 1ª y 1000 en la 2ª?
  - ¿Funcionará mejor con 1000 neurona en la 1ª y 1 en la 2ª?

16. Otros modelos de redes neuronales

- Radial Basis Neural Network

  Similares a las RNA vistas, pero con la particularidad de que la función de activación usada es radial. Por tanto calculan la salida en función de la distancia a un punto denominado centro.



  - Características:
    - ➔ Simétricas respecto de x=0
    - ➔ Se definen con al menos 2 parámetros:

      Centro ➔ Punto donde la función posee un extremo

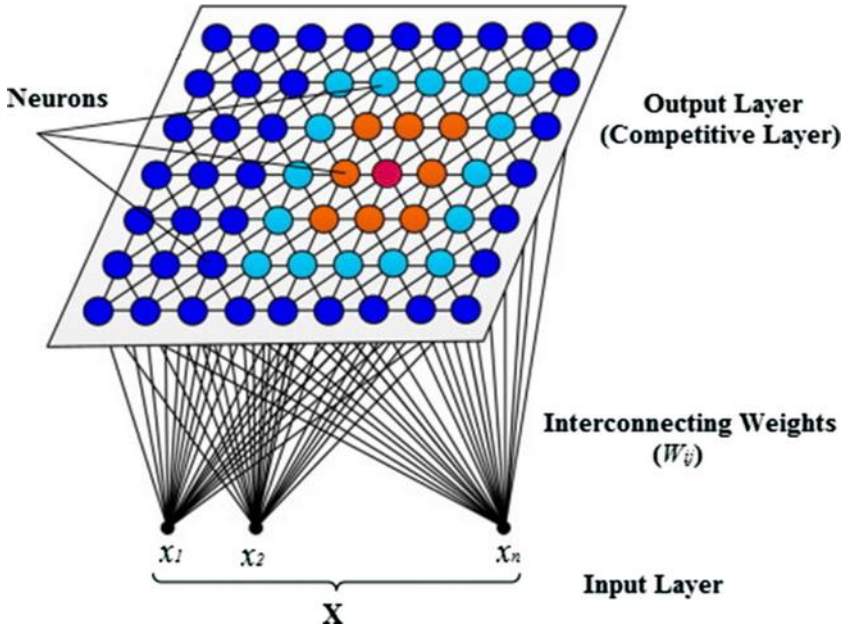
      Anchura ➔ Magnitud de la variación de la función según se aleja del centro.

Las salidas de las redes de neuronas de base radial son, por tanto una combinación lineal de gausianas, cada una de las cuales se activa para una determinada porción del espacio definido por los patrones de entrada.

- Se usan principalmente en problemas de clustering.

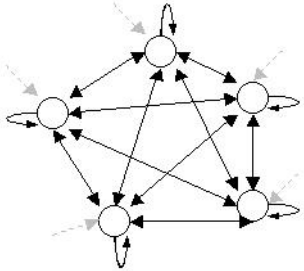
16. Otros modelos de redes neuronales

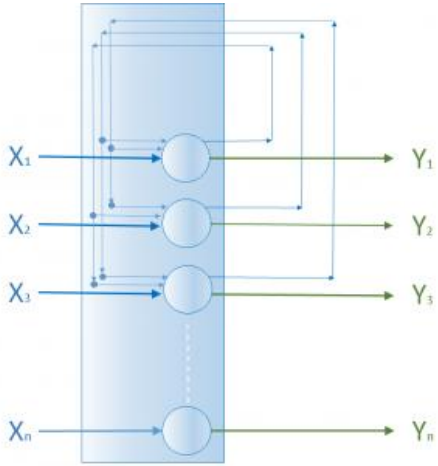
- Mapas auto-organizados (Redes Kohonen)
  - Se basan en ciertas evidencias descubiertas a nivel cerebral y usan un aprendizaje no supervisado competitivo.
  - Se orienta a la clusterización o clasificación de los datos de entrada (categorización).
  - La red auto-organizada debe descubrir rasgos comunes, regularidades, correlaciones o categorías en los datos de entrada, e incorporarlos a su estructura interna de conexiones.
  - Se dice, por tanto, que las neuronas deben auto-organizarse en función de los estímulos (datos) procedentes del exterior.



16. Otros modelos de redes neuronales

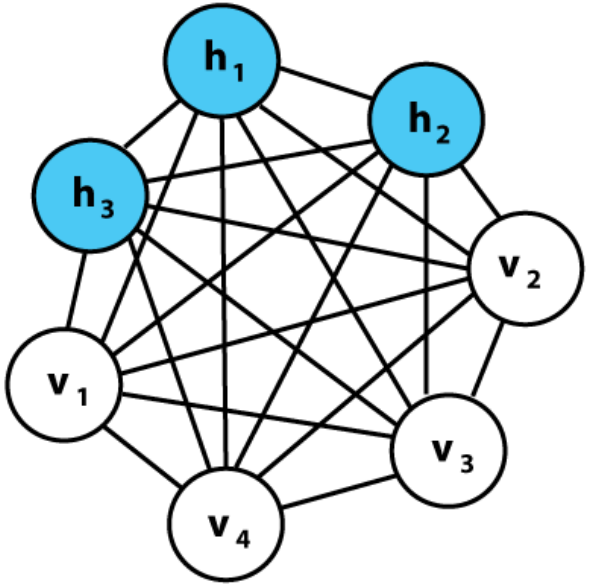
- Red de Hopfield
  - Es probabilístico
  - Al ser un modelo recurrente todas las neuronas están conectadas entre sí, cada neurona recibe señal de todas las demás y emite hacia todas ellas, pero no a sí.
  - Consta de una matriz de pesos fija (W), simétrica y de diagonal nula (0).
  - Cada neurona puede tomar dos valores distintos (binarios-bipolares).
  - Al introducir un patrón de entrada, la información se propaga hacia adelante y hacia atrás. La convergencia a uno de los patrones no está; garantizada, esta situación puede provocar que no se alcance la estabilidad del sistema y que la red nunca se detenga.





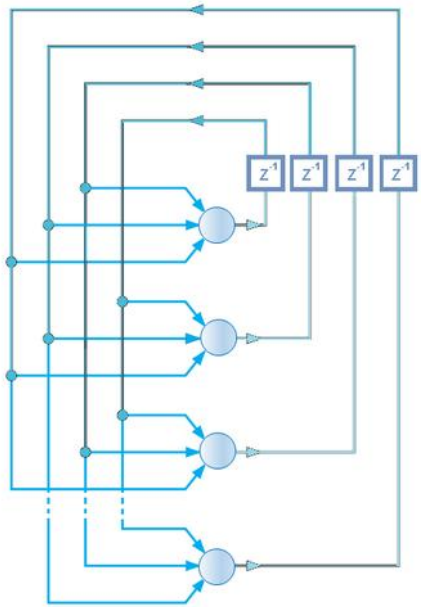
16. Otros modelos de redes neuronales

- Máquinas de Boltzmann
  - Son redes estocásticas de Hopfield con unidades ocultas y recurrentes que representan la información a partir de una distribución de probabilidad.
  - Los pesos se inician aleatoriamente y la red aprende por backpropagation.
  - Se usa para resolver 2 problemas diferentes:
    1. Búsquedas. Los pesos en las conexiones son fijos y se usan para representar una función de coste.
    2. Aprendizaje. Hacen muchas actualizaciones pequeñas en los pesos y cada actualización requieren que resuelvan problemas de búsqueda diferentes.



16. Otros modelos de redes neuronales

- Redes neuronales recurrentes
  - Son un grupo de redes neuronales especializadas en procesar datos secuenciales
  - Deben tener al menos un circuito de retroalimentación
  - 2 tipologías: de una capa y capa oculta
  - Su principal ventaja estriba en la posibilidad de almacenar (memorizar) una representación de la historia reciente de la secuencia. Son más potentes que las redes neuronales feed-forward, debido a su memoria.
  - Sin embargo son más difíciles de entrenar



17. Play with it

- Prueba a jugar con una RNA aquí:
  - https://playground.tensorflow.org/