

# Resumen Inteligencia Artificial 2

Por: Pablo Federico Martín Luna

## Tema 1: Introducción al Procesamiento Subsimbólico

### Inteligencia Artificial Subsimbólico

Técnicas de IA que simulan los elementos de más bajo nivel que interviene en los procesos inteligentes para que de su combinación emerja de forma espontánea el comportamiento inteligente.

### Soft Computing

Rama de la IA que se dedica al desarrollo de algoritmos y técnicas que pueden resolver problemas que son típicamente difíciles o no factibles de resolver utilizando los métodos tradicionales. (No tiene por qué ser la mejor solución)

### Técnicas subsimbólicas basadas en sistemas biológicos

- Estructura del cerebro: Redes Neuronales
- Evolución biológica: Computación evolutiva

### Áreas

- Búsqueda
- Aprendizaje automático (Machine Learning/Data Mining)
- Simulación sensorial

### Machine Learning (Aprendizaje Automático)

Rama de la IA que desarrolla algoritmos de análisis de datos que permitan al sistema aprender y tomar decisiones autónomas en base a la predicción de las situaciones que se puedan producir.

### Aprender

Generalizar comportamientos mediante detección de patrones en la información suministrada en forma de ejemplos y experiencia.

### Optimizar

Calcular el máximo y mínimo de una función.

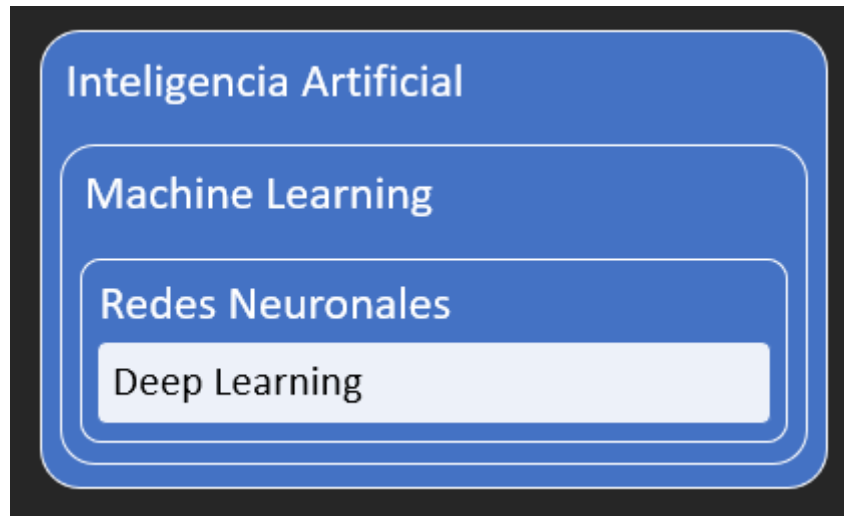
Función de error, Loss function, Función objetivo.

En machine Learning ha aprendido porque nuestra función de error es un mínimo.

### Factores que influyen en el aprendizaje

- Qué **conocimiento previo** tiene el sistema. (un buen set de datos)
- Qué **representación** se usa para los datos que usan el sistema.
- Qué **feedback** está disponible para que el sistema aprenda.
  - Aprendizaje supervisado:
    - Cualitativa – clasificación.
    - Cuantitativa – regresión.
  - Aprendizaje no supervisado:
    - Clustering
  - Aprendizaje por refuerzo.

- Aprendizaje semi-supervisado



### ¿Qué son las Redes Neuronales?

Sistema compuesto por un conjunto de elementos de procesamiento simples (neuronas), conectadas entre sí (red), cuyo comportamiento está determinado por la topología y los pesos de sus conexiones.

### Características

#### Propiedades emergentes

El comportamiento global de la red es distinto del comportamiento individual de las neuronas que la componen. (aprender de la experiencia, autoorganizarse, generalizar y abstraer)

#### Aprendizaje adaptativo

El sistema modifica su comportamiento para ajustarlo a los datos que se le proporcionan.

#### Autoorganización

Emplean su capacidad de aprendizaje adaptativo para autoorganizar la información que recibe durante el aprendizaje.

#### Generalización

Facultad de responder apropiadamente cuando se les presentan datos que no habían sido presentados anteriormente.

#### Procesamiento distribuido

El conocimiento no está en ninguna neurona concreta.

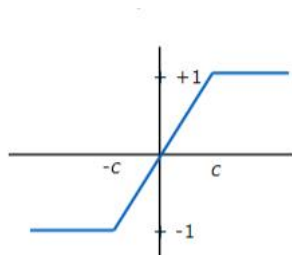
## Conceptos básicos

### Neurona

Unidad de procesamiento elemental que genera una salida como resultado de la integración de un conjunto de entradas.

Formada por: entradas, salidas, procesamiento y pesos.

- Procesamiento: operaciones elementales, suma ponderada, funciones de activación.
  - Funciones de activación:
    - No lineal:

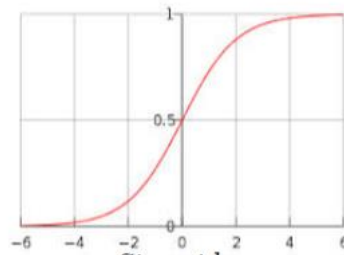


*Lineal*

$$Y = +1 \text{ si } S > c$$

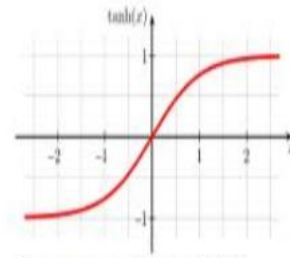
$$Y = -1 \text{ si } S < -c$$

$$Y = aS \text{ en otro caso}$$



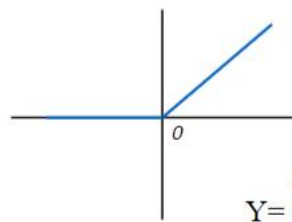
*Sigmoidea*

$$Y = \frac{1}{1 + e^{-S}}$$



*Tangente hiperbólica*

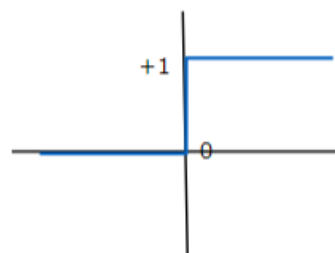
$$Y = \frac{e^S - e^{-S}}{e^S + e^{-S}}$$



*ReLU*

$$Y = \max(0, S)$$

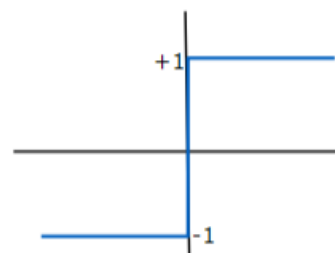
- Lineal:



*Escalón*

$$Y = 1 \text{ si } S \geq 0$$

$$Y = 0 \text{ si } S < 0$$



*Signo*

$$Y = +1 \text{ si } S \geq 0$$

$$Y = -1 \text{ si } S < 0$$

### Capa

Elemento estructural que agrupa neuronas con comportamiento similar.

## Red

Conjunto de reglas que define la forma en que se conectan las capas y/o PE.

Multicapa: Capa de Entrada (IL), Capa/s intermedia/s (HL) y Capa de Salida (OL).

## Procesamiento

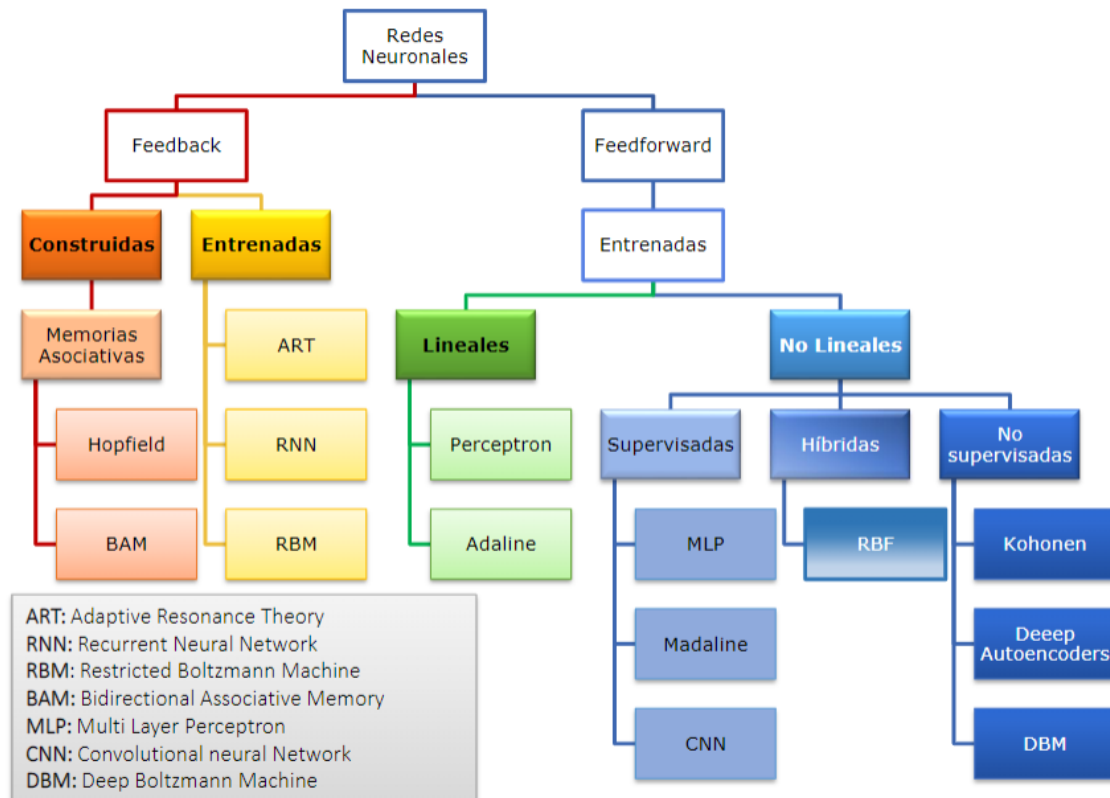
A partir de la capa de entrada cada neurona

1. Procesa la información de sus conexiones de entrada.
2. Evalúa su función de activación.
3. Transmite su salida  $Y_j$  a las neuronas con las que está conectada.

## Aprendizaje

Proceso (algoritmo) por el que la red modifica sus pesos en función de las salidas presentadas asociando patrones de entrada a patrones de salida.

## Tipos de arquitecturas



## Tema 2: Aprendizaje NO Supervisado: Aprendizaje Competitivo

Algoritmos de aprendizaje no supervisado están generalmente relacionados con la detección de características: Clustering y Clasificar.

### Clustering

Dividir el espacio N-dimensional de características en K regiones excluyentes.

Las características N-dimensionales deben ser multinormales con igual covarianza.

### Clasificar

Asignar una de las K regiones excluyentes obtenidas en el Clustering a los patrones.

Asignar cada patrón a uno de los grupos/clases definidos Previamente.

### Tipos de clasificadores

#### Discriminación

- Discriminantes lineales o no lineales.
  - Clasificación supervisada.
  - Parten de los valores de una serie limitada de parámetros.
  - Comparación de características del individuo y de los grupos.
- Regresión logística.

#### Clustering

- Métodos de Clustering o estimación de densidades.
  - Clasificación no supervisada.
  - No se dispone de una muestra previamente clasificada. A priori no se conocen los grupos y lo que precisamente se desea es establecerlos a partir de los datos que poseemos.
  - La agrupación de individuos se realiza minimizando ciertas funciones de distancia mediante técnicas estadísticas:
    - Distancia Euclídea:  $\sqrt{|x_1 - x_2|^2 + |y_1 - y_2|^2}$
    - Distancia de Manhattan:  $|p_1 - q_1| + |p_2 - q_2|$
    - Distancia de Mahalanobis.

### Autoorganización

Proceso en el cual, por medio de interacciones locales, se obtiene un ordenamiento global mediante un proceso de aprendizaje no supervisado.

### Tipos de aprendizaje no supervisado

#### Aprendizaje Hebbiano o Asociativo

1. Si dos neuronas en cada lado de una conexión se activan sincrónicamente, entonces el peso de esa conexión se incrementa.
2. Si dos neuronas en cada lado de una conexión se activan asincrónicamente, entonces el peso de esa conexión se decrementa.

$$\vec{W}^{nuevo} = \vec{W}^{viejo} + \vec{y}_i \vec{x}_i$$

#### Aprendizaje por Principal Component Analysis (PCA)

(Esto lo estudiamos del Resumen Data Mining)

## Aprendizaje Competitivo (SOM y ART)

El aprendizaje consiste en reforzar las conexiones de la unidad ganadora y debilitar las otras.

### Mapas Autoorganizativos

#### Arquitectura

- Capas: Entrada + Kohonen.

$$\text{cantidadVectoresPesos} = \text{numNeuronasKohonen}$$

$$\text{cantidadPesos} = \text{numEntradas} * \text{ladoMapa} * \text{ladoMapa}$$

- Elementos de un SOM:
  - Matriz multidimensional.
  - Función discriminante.
  - Mecanismo de competición.
  - Mecanismo de cooperación.
  - Mecanismo de adaptación.

#### Procesamiento

1. Propagación de la entrada.
2. Cálculo de la función discriminante.

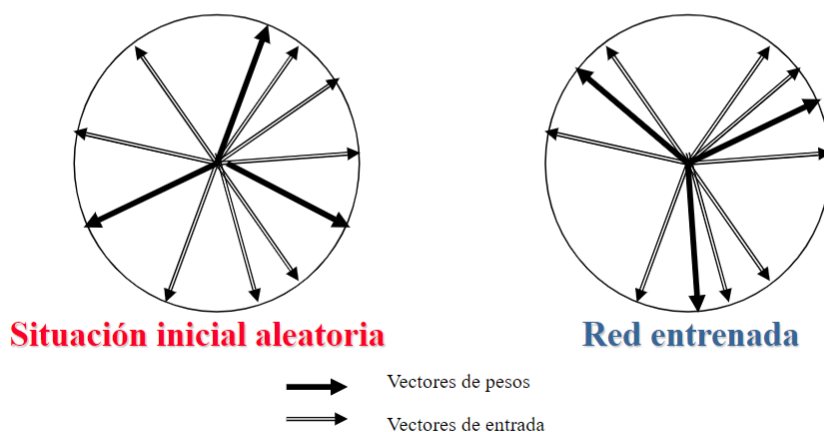
$$D_j = \|X - W_j\| = \sqrt{\sum_{i=1}^N (x_i - w_{ij})^2} \quad E = XW_j = \sum_{i=1}^N x_i w_{ij}$$

3. Competición.

$$O_j = \begin{cases} 1 & E_j = \max_{i=1 \dots k^2} E_i \\ 0 & E_j \neq \max_{i=1 \dots k^2} E_i \end{cases}$$

#### Aprendizaje

Ajuste de la matriz de pesos de las neuronas de la capa de Kohonen a medida que se presentan patrones (vectores) de entrada en un proceso iterativo (entrenamiento).



Proceso:

1. Se inicializa la matriz de pesos con valores aleatorios.
2. Se presentan uno a uno los patrones del conjunto de entrenamiento, también normalizados.
3. Se determina la neurona ganadora (BMU).
4. Se actualizan los pesos de la BMU.

Vecindario

El aprendizaje no se limita a la BMU, sino que se extiende a las neuronas adyacentes.

Las funciones de aprendizaje y vecindario están ligadas.

Mapas topológicos

La capa de Kohonen preserva el orden y compacta la representación de una nube de datos en un espacio N-dimensional proyectándola sobre un mapa bidimensional.

Parámetros

- Periodo. (p)
- Coeficiente de aprendizaje. ( $\eta_0$ )
- Vecindario. ( $v_0$ )
- Amortiguación. ( $A_i$ )
- Número de neuronas del lado de la capa de Kohonen. (k)

Iteración: Activación de una neurona + Aprendizaje.

Número de presentaciones. (t)

Variación del coeficiente de aprendizaje.  $\eta_{(t)} = \eta_0(1 - \frac{t}{p})$

Variación del vecindario.  $v_{(t)} = 1 + v_0(1 - \frac{t}{p})$

Amortiguación.  $A_i = e^{-\frac{d_i^2}{2v_t^2}}$

Ecuación de aprendizaje.  $W_{i,j}^{t+1} = W_{i,j}^t + \eta_{(t)} A_j (X_i - W_{i,j}^t)$

Modificaciones al proceso básico

- Refuerzo
  - Después del primer periodo de aprendizaje, se realizan sucesivos períodos de refuerzo.
  - En cada periodo de refuerzo se cumple que:

$$p_r = p_{factor}^r \quad \eta_{0,r} = \eta_0 \eta_{shrink}^r$$

- Mecanismo de consciencia
  - Función de penalización respecto a la frecuencia de activación.
  - Esto se realiza modificando la distancia neurona – patrón a partir de la frecuencia de activación de cada neurona.

$$D'_j = D_j - B_j$$

## Métricas para medir la adecuación de cada mapa

- Error de Cuantificación.

Distancia promedio entre cada vector de datos y su BMU.

- Distancia media de cada clase.
- Distancia media de la red.

- Error Topológico.

Proporción de todos los vectores de datos para los que la primera y la segunda BMU no son unidades adyacentes.

- Primera y segunda BMU adyacentes – 0.
- Primera y segunda BMU no adyacentes – 1.



## Tema 3: Aprendizaje Supervisado: Modelos Lineales

Algoritmos de aprendizaje supervisado aprenden a reproducir comportamientos y razonamientos mediante estímulos.

### Predicción

Proporcionar valores futuros de una variable con una cierta antelación (horizonte), en función de los valores históricos (serie temporal) que ha tenido dicha variable en el pasado.

### Serie Temporal

Sucesión de medidas realizadas a intervalos regulares de tiempo resultado de un proceso estocástico.

### Definiciones

- Proceso estocástico: conjunto de variables aleatorias  $X_t$  cuya distribución varía de acuerdo al tiempo  $t$ .
- Proceso de Márkov: el valor de la serie a tiempo  $t + 1$  sólo depende del valor a tiempo  $t$  y no de los anteriores.
- Proceso estacionario: un proceso es estacionario en sentido amplio o débilmente estacionario cuando la media, varianza y covarianza existen y son estables e independientes del tiempo.

### Elementos de una Serie Temporal

- Tendencia: movimiento que se mantiene durante el periodo de observación.
- Variación estacional: oscilaciones periódicas que dependen del período mínimo al cabo del cual aparecen patrones repetitivos en la serie temporal.
- Movimientos cíclicos: oscilaciones sobre la tendencia basadas en algún tipo de funciones trigonométricas.
- Proceso aleatorio: oscilaciones aleatorias que se superponen a los demás componentes resultado de las fluctuaciones estocásticas del sistema real.

### Técnicas de predicción

- Cuantitativas: estimaciones numéricas a partir de valores o propiedades que se conocen de la variable a predecir:
  - Deterministas.
  - Estadísticas/Estocásticas.
  - Neuronales.
- Cualitativas: estimación subjetiva a partir de opiniones de expertos. Carecen de bases teóricas. Sirven para aglutinar opiniones.

### Técnicas estadísticas

- Métodos tradicionales. Sencillez de cálculo, bajo coste, utilidad práctica. Casos particulares de modelos más complejos.
- Métodos de Descomposición. Identifican las componentes deterministas de la serie (tendencia, estacionalidad, periodicidad) descartando la componente aleatoria que se engloba en un término de error
- Modelos Box-Jenkins. Modelos: AR, MA, ARMA, ARIMA y SARIMA.

Redes a las que hay que enseñar

Entrenamiento

Proceso iterativo que busca minimizar el error en el conjunto de entrenamiento.

Ley de aprendizaje

Formulación matemática que ajuste la matriz de pesos en función de los pares entrada/salida con que se entrena la red.

Aprendizaje = Entrenamiento + Ley de aprendizaje

Perceptrón

- Cálculo de la entrada a la neurona (**S**)

$$S = w_1x_1 + w_2x_2$$

- Cálculo de la salida de la neurona (**Y** real)

$$Y = \begin{cases} +1 & \text{si } S \geq \theta \\ 0 & \text{si } S < \theta \end{cases}$$

- Error

$$e = d_t - y_t$$

- Actualización de los pesos (Ley de Hebb)

$$\vec{W}_{t+1} = \vec{W}_t + \eta e \vec{X}_t$$

## Tema 4: Aprendizaje Supervisado: Modelos No Lineales

### Historia

El MLP es una variación del modelo Adaline de Widrow con aprendizaje por minimización del MSE mediante Regla Delta.

Aplicaciones de un perceptrón:

- Clasificación.
- Reconocimiento de señales.
- Monitorización en tiempo real.
- Control adaptativo sobre múltiples variables en tiempo real.
- Predicción de series temporales.

Ventajas:

- Capacidad de generalización.
- Soluciona la mayoría de los problemas a los que se aplica.
- Una vez entrenada, produce en ejecución resultados muy rápidos.

### Arquitectura

#### ¿Cuántas capas ocultas?

En una capa oculta no hay forma de saber cuál debería ser la salida adecuada de las neuronas de esta capa.

#### ¿Cuántas neuronas en cada capa?

- Balance entre capacidad y velocidad de aprendizaje.
- Menor que el número de patrones de entrenamiento. Lo contrario hace que cada neurona de la capa intermedia memorice un patrón de entrenamiento en lugar de generalizar a partir de casos individuales.
- Pocas neuronas -> underfitting.
- Demasiadas neuronas -> overfitting.
- Empíricamente se ha comprobado que el número de elementos en las capas ocultas debe de ser:

$$\frac{M}{2N} < n < \frac{2M}{N}$$

#### ¿Podemos resolver cualquier problema?

##### Teorema de Aproximación Universal

Cualquier función continua  $R^N \xrightarrow{f} R^M$  definida sobre un conjunto compacto de datos puede aproximarse con precisión arbitraria mediante una red neuronal de una sola capa oculta.

## Regla Delta Generalizada o Backpropagation

Método de obtención del vector gradiente en redes neuronales de múltiples capas y funciones de transferencia no lineal y diferenciales para poder aplicar a cada capa el descenso de gradiente.

Oculto  $\rightarrow$  Salida

$$\vec{w}_{no}^{t+1} = \vec{w}_{no}^t + \Delta \vec{w}_{no}^t$$
$$\Delta \vec{w}_{no}^t = -\eta \nabla E(\vec{w}_{no}^t)$$
$$\nabla E(\vec{w}_{no}^t) = \frac{\partial E^p}{\partial \vec{w}_{no}^t} = \frac{\partial E^p}{\partial s_o^p} \frac{\partial s_o^p}{\partial \vec{w}_{no}^t}$$
$$\frac{\partial E^p}{\partial s_o^p} = -\delta_o^p = (d_o^p - o_o^p) g'(s_o^p) = \delta_o^p$$
$$\frac{\partial s_o^p}{\partial \vec{w}_{no}^t} = \frac{\partial}{\partial \vec{w}_{no}^t} \left( \sum_{j=1}^Q w_{jo}^p h_j^p \right) = f_o'(s_o^p) = h_n^p$$
$$\nabla E(\vec{w}_{no}^t) = -\delta_o^p h_n^p$$
$$\Delta \vec{w}_{no}^t = \eta \delta_o^p h_n^p$$
$$\vec{w}_{no}^{t+1} = \vec{w}_{no}^t + \eta \delta_o^p h_n^p$$

Entrada  $\rightarrow$  Oculta

$$\vec{w}_{in}^{t+1} = \vec{w}_{in}^t + \Delta \vec{w}_{in}^t$$

$$\Delta \vec{w}_{in}^t = -\eta \nabla E(\vec{w}_{in}^t)$$

$$\nabla E(\vec{w}_{in}^t) = \frac{\partial E}{\partial \vec{w}_{in}^t}$$

$$\frac{\partial E}{\partial \vec{w}_{in}^t} = \frac{\partial E}{\partial s_n^p} \frac{\partial s_n^p}{\partial \vec{w}_{in}^t}$$

$$\frac{\partial E}{\partial s_n^p} = -\delta_n^p = \frac{\partial}{\partial s_n^p} \frac{1}{2} (d_n^p - y_n^p)^2$$

Como não conhecemos  $d_n^p$  nos queda:

$$-\delta_n^p = g_h'(s_n^p) \sum_{k=1}^m w_{nk} \delta_k^p$$

$$\frac{\partial s_n^p}{\partial \vec{w}_{in}^t} = \frac{\partial}{\partial \vec{w}_{in}^t} \left( \sum_{k=1}^p w_{ik} x_i^p \right) = x_i^p$$

$$\frac{\partial E}{\partial \vec{w}_{in}^t} = -\delta_n^p x_i^p$$

$$\nabla E(\vec{w}_{in}^t) = -\delta_n^p x_i^p$$

$$\Delta \vec{w}_{in}^t = \eta \delta_n^p x_i^p$$

$$\vec{w}_{in}^{t+1} = \vec{w}_{in}^t + \eta \delta_n^p x_i^p$$

### Posibles situaciones la concluir el entrenamiento

- Entrenamiento correcto: generaliza.
- Subentrenamiento: entrenamiento insuficiente.
- Sobre-entrenamiento: exceso de entrenamiento → memoriza.

### Validación cruzada simple

Método estadístico de estimación del error de generalización basado en usar un conjunto de datos de validación elegido aleatoriamente del dataset que no se usarán en el entrenamiento.

### Validación cruzada de k iteraciones

Los datos de muestra se dividen en k subconjuntos. Uno de los subconjuntos se utiliza como datos de validación y el resto (k-1) como datos de entrenamiento.

### Problemas con el algoritmo

- Tiempo de computación necesario para el entrenamiento.
- Existencia de mínimos locales en la función.
- Conjunto de entrenamiento.
- Tamaño de la red.
- Sobre-entrenamiento.
- Parálisis de la red.

Posibles soluciones:

- Variaciones al modelo básico.
- Técnicas optimizadoras del entrenamiento.
  - Extracción de características.
  - Entrenamiento Adaptativo.
  - Simulated Annealing.
- Otras técnicas de entrenamiento.

### Redes de Función de Base Radial (RBF)

**Mirar directamente de las diapositivas.**

## Tema 5: Deep Learning

### Problema de los algoritmos de retropropagación

El error se va diluyendo de forma exponencial a medida que atraviesa capas hacia la capa de entrada.

### Aprendizaje Profundo o Deep Learning

Área del Machine Learning que utiliza diferentes algoritmos de aprendizaje automático para modelar abstracciones de datos de alto nivel usando arquitecturas jerárquicas llamadas redes neuronales profundas (DNN).

### Teorema de Aproximación Universal VS. Deep Learning

1. Las redes generalizan mejor cuanto mayor es la profundidad. Una red neuronal con una sola capa oculta no es capaz de aprender representaciones jerárquicas.
2. Las redes superficiales necesitan más anchura para alcanzar la misma presión.
3. Las redes superficiales se sobreentrenan más fácilmente.

### Problemas de las DNN

1. La no disponibilidad de grandes cantidades de datos para entrenar a los modelos.
2. No tener algoritmos de entrenamiento que permitan entrenar las capas más profundas.
3. No tener hardware en los ordenadores lo suficientemente potente como para la ejecución de los modelos.

### Solución de Hinton

#### Uso de RBM

1. Cada capa de la red es entrenada utilizando un algoritmo de preentrenamiento no supervisado comenzando con la capa de entrada a la red.
2. Después de preentrenar todas las capas, la red puede ser ajustada utilizando un entrenamiento supervisado: descenso estocástico de gradiente (SGD) + Backpropagation.

#### Entrenamiento de Autoencoders

1. Entrenar cada capa encoder de forma no supervisada, especialmente las primeras.
2. A continuación, se usa esa capa oculta como capa de entrada y se repite el proceso.
3. Por último, se combinan y se entrena toda la red de forma supervisada para hacer únicamente un ajuste fino de los pesos.

**El resto del tema se puede ver directamente de las diapositivas.**

## Tema 6: Redes convolucionales y Autoencoders

### Convolución

Operador matemático que transforma dos funciones  $f$  y  $g$  de una sola variable en una nueva función  $f * g$  que representa la magnitud en la que se superponen  $f$  y una versión trasladada e invertida de  $g$ .

### Transformación global VS. Transformación local

- Global. Cada píxel de salida depende sólo de un píxel de entrada y no se tiene en cuenta la relación de vecindad entre píxeles.
- Local. El valor de un píxel depende de la vecindad local de ese píxel.

### Redes convolucionales

Redes que pueden aprender relaciones entrada-salida basadas en operaciones de convolución, donde la entrada es una imagen.

### Arquitectura

- Feedforward: variación del MLP.
- Locally connected: Existen capas, pero se restringen las conexiones entre capas.
- Capas:
  - Input.
  - Capa(s) de convolución: capa compuesta por neuronas convolucionales que detectan patrones aplicando varios filtros de 1 o más dimensiones a los datos que le llegan.
    - Hiperparámetros: número de filtros, dimensión de cada filtro, tamaño del stride y tamaño del padding.
  - Capa(s) de reducción (pooling): capa que agrupa los valores de varias celdas contiguas del mapa de características de un filtro usando alguna función de agrupación (subsampling).
    - Hiperparámetros: tipo de pooling, tamaño del pooling y tamaño del stride.
  - Capa(s) clasificadora (fully connected): capa compuesta por un MLP con:
    - Capa(s) oculta(s) no lineal(es): función de activación relu.
    - Capa de salida (que también es la capa OUTPUT de la CNN).
    - Hiperparámetros: número de capas y número de neuronas.

### Aplicaciones de las CNN

- 1D CNN: texto, señales secuenciales
- 2D/3D CNN: imágenes escala de grises/color, señales en frecuencias.
- 3D CNN: vídeo, imágenes volumétricas/tomográficas.

### Deep Autoencoders

Modelo neuronal Feedforward no supervisado de compresión de datos en el que las funciones de compresión y descompresión son llevadas a cabo por distintas capas que son:

- Específicas de los datos: solo pueden comprimir datos similares a los usados para entrenamiento.
- Con pérdida: las salidas descomprimidas se desgradarán en comparación con las entradas originales.
- Aprenden automáticamente (auto supervisado): entrada y salida iguales.



El algoritmo necesita:

- Una función codificadora: codifica un input  $\vec{x}$  mediante una función  $h = f(\vec{x})$ .
- Una función decodificadora: decodificar los valores de  $f(\vec{x})$  utilizando una función  $\hat{x} = g(f(\vec{x}))$ .
- Una función de distancia (loss function) que mide la cantidad de información perdida entre la representación comprimida y la representación descomprimida. El objetivo es minimizar el error de reconstrucción entre el input  $\vec{x}$  y el output  $\hat{x}$ .

#### Undercomplete Autoencoders

Sirven para identificar y capturar las características más importantes en los datos usando una capa oculta menor que la capa de entrada.

#### Sparse Autoencoders

La capa oculta es mayor que la entrada. Se obtienen las características más importantes, pero evita que la salida de la red copie la entrada recibida.

#### Denoising Autoencoders

Añaden ruido de forma intencionada a los datos antes de ser introducidos en la red. La red aprende las características más importantes de los datos en lugar de copiarlos en su salida.

#### Contractive Autoencoders

Tienen como objetivo obtener una representación robusta del aprendizaje para que sea menos sensible a las pequeñas variaciones en los datos.

#### Stacked Autoencoders

Modelo Deep consistente en una red neuronal multicapa compuesta por múltiples Autoencoders. Equivalente a Deep Belief Network.

#### Deep Autoencoders

Dos redes neuronales simétricas y conectadas, donde una se encarga de la codificación y la otra de la decodificación.

#### Aplicaciones

- Reducción de la dimensionalidad.
- Motores de recomendación.
- Extracción de características.
- Reconocimiento de imágenes.

## Tema 8: Algoritmos Genéticos

### Historia

#### Mirar de las diapositivas

### ¿Qué es un AG?

El proceso de optimización de un problema mediante un AG consiste en:

- Identificar de qué parámetros depende la solución al problema.
- Generar aleatoriamente un conjunto de soluciones potenciales a ese problema usando esos parámetros y codificadas de alguna manera.
- Definir una métrica que permita evaluar cuantitativamente a cada solución candidata.
- Ejercer una presión selectiva sobre cada solución de forma que se produzca una selección.
- Crear variaciones de las soluciones usando los principios de la genética molecular.

### Definición

Algoritmo matemático sistemático de resolución/optimización de problemas mediante búsqueda.

Altamente paralelo.

Donde un conjunto (población) de soluciones (individuos) a un problema, codificadas en forma de secuencia (cromosoma), asociadas con una cierta función de coste (fitness) y que pueden ir cambiando mediante operadores (evolución), compiten para ver cuál satisface mejor el problema.

Aunque no necesariamente se obtiene la mejor de todas las soluciones posibles.

### Pseudo Código

BEGIN

1. Generar una población inicial.
2. Calcular la función de evaluación de cada individuo (fitness).
3. WHILE NOT Terminado DO

    FOR Tamaño Población DO

1. Seleccionar progenitores
2. Cruzar
3. Mutar
4. Calcular nuevo fitness
5. Insertar

    END

IF la población ha convergido THEN

    Terminado = TRUE

END

END

## Fundamentos biológicos

- Meiosis – Recombinación
- Mitosis – Mutación

## Elementos de un AG

- Codificación. La solución al problema tiene que representarse (codificarse) en un cromosoma.
  - Decodificación – obtención del fenotipo
  - Tipos:
    - Codificación binaria (alfabeto cardinalidad = 2)
    - Codificación numérica (alfabetos cardinalidad  $\neq$  2)
    - Codificación por valor directo (alfabetos cardinalidad  $\neq$  2)
  - El teorema de los esquemas afirma que la codificación sobre la que los algoritmos genéticos funcionan mejor es con un alfabeto de cardinalidad 2.
- Individuo. Cromosoma (secuencia de caracteres/dígitos) que constituye el código de información sobre el cual opera el AG.
- Población. Un conjunto de individuos. (Cromosomas)
- Función Fitness/Objetivo/Coste: Función que informa de cuan bueno es un individuo dado en la solución de un problema. (evalúa el fenotipo)
- Parámetros:
  - Tamaño de la población. Número de cromosomas en nuestra población para una generación determinada.
  - Condición de terminación. Convergencia del algoritmo genético. (fitness no varía después de varias generaciones)
  - Probabilidad de Selección. Indica la probabilidad de elegir un cromosoma para ser replicado.
  - Probabilidad de Entrecruzamiento. Indica la frecuencia con la que se producen cruces entre los cromosomas padre.
  - Probabilidad de Mutación. Indica la frecuencia con la que cada gen de un cromosoma es mutado. Se aplica gen a gen.

## Operadores

### Operador Selección

Mecanismo que favorece la replicación de los individuos con mejor valoración, pero que no impide la diversidad.

Se compone de dos partes:

- Determinación de que individuos de la población serán utilizados para la reproducción.
- Determinación de los individuos de la población que serán sustituidos por los recién generados.

### SELECCIÓN DE LOS PADRES

- Selección por Ruleta. (Probabilística)
- Selección por Rango.

## SELECCIÓN DE INDIVIDUOS ELIMINADOS

- Selección elitista.
- Selección pura.
- Selección por Estado Estacionario.

### Operador Entrecruzamiento

Consiste en unir en alguna forma los cromosomas de los padres que han sido previamente seleccionados de la generación anterior para formar dos descendientes.

### Operador Mutación

Consiste en la alteración aleatoria de cada uno de los genes del individuo con una probabilidad de mutación  $P_M$ .

### Aplicaciones

- Optimización.
- Aprendizaje de máquina.
- Bases de datos.
- Reconocimiento de patrones.
- Generación de gramáticas.
- Planeación de movimientos de robots.
- Predicción.