M2.2.2 Modelos Supervisados y No Supervisados

Programa Big Data y Business Intelligence

Enrique Onieva
enrique.onieva@deusto.es
https://twitter.com/EnriqueOnieva
https://www.linkedin.com/in/enriqueonieva/

Deusto
Facultad de Ingeniería
Ingeniaritza Fakultatea

Redes Neuronales

- Motivación, paralelismo con el cerebro biológico
- Funcionamiento de la Neurona Artificial
- Aprendizaje de una neurona
- Modelos y aprendizaje de estructuras de multiples capas

Motivación

- Durante varias décadas los científicos han perseguido la construcción de algoritmos capaces de procesar información al igual que el cerebro
- En la actualidad existen muchas estructuras de redes de neuronas artificiales
- Se caracterizan porque poseen propiedades como la capacidad de aprendizaje a partir de ejemplos

Motivación

- 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

Motivación

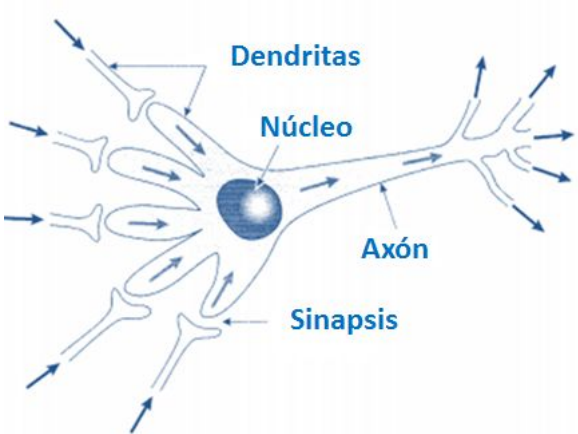
- Objetivo: Usar los principios de organización del cerebro para construir sistemas inteligentes.
  - RNA → Emulación (modelo matemático) del funcionamiento del cerebro a bajo nivel.
- Cerebro/RNAs
  - Sistemas masivamente paralelos formados por un gran número de elementos simples (neuronas) interconectados

Aplicaciones

- En dominios difíciles (necesidad aprendizaje)
  - Entradas/salidas muchas dimensiones o con ruido.
  - Tareas clasificación/reconocimiento de patrones.
  - Comprensión por humanos poco importante.
- Aplicaciones:
  - Reconocimiento/generación de voz
  - Reconocimiento de formas (OCR, ...)
  - Identificación personas (voz, huellas, iris,...)
  - Predicción, series temporales...
  - Clasificación, aproximación funciones, filtrado de señales...

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 y 200.000 millones
  - Cada una con hasta 30.000 conexiones con otras
- Es la base del procesamiento del cerebro humano



Neurona

- Funcionamiento
  - Neurona (soma) ”acumula” todos los potenciales 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
  - Las conexiones (sinapsis) pueden modificarse.
  - Conexiones más o menos fuertes.
  - Permite modificar comportamiento de la neurona para adaptarse a nuevas situaciones (aprendizaje)

Neurona

- Se nace con alguna estructura neuronal
  - Se crean nuevas conexiones y otras se gastan.
  - Se desarrollan por el aprendizaje de la etapa de crecimiento
  - La estructura neuronal cambia durante la vida.
  - Estos cambios consisten en reforzamiento o debilitamiento de conexiones
- El hecho de ser capaz de memorizar la cara de una persona que nos presentan, consiste en alterar “varias” sinapsis de nuestra red neuronal

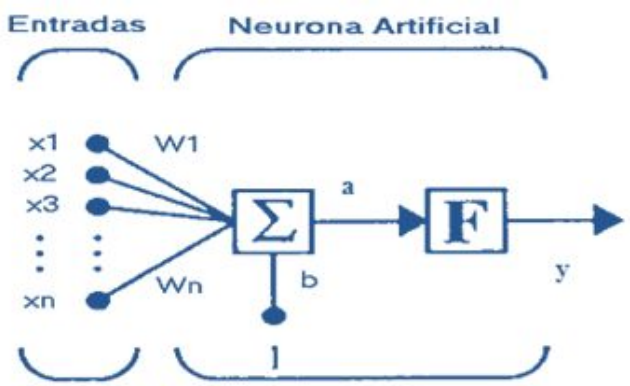
Neurona Artificial

- El cerebro humano constituye una computadora muy notable, es capaz de interpretar información imprecisa suministrada por los sentidos a un ritmo increíblemente veloz.
  - Características del cerebro deseables para un sistema
    - Es robusto y tolerante a fallos: Diariamente mueren neuronas sin afectar su desempeño.
    - Es flexible:Se ajusta a nuevos ambientes por aprendizaje, no hay que programarlo.
    - Puede manejar información difusa, con ruido o inconsistente.
    - Es altamente paralelo
    - Es pequeño, compacto y consume poca energía

Neurona Artificial

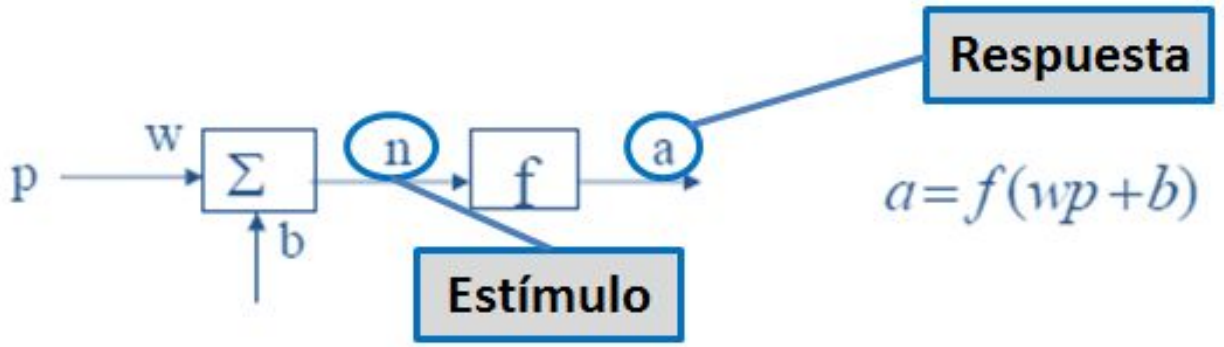
- Funcionamiento
  - Cada neurona recibe diferentes entradas (x_1,x_2, x_3,…)
  - Cada señal se multiplica por un valor, denominado peso, que da una idea de la fuerza de esa conexión (w_1,w_2, w_3,…)
  - Se calcula una salida, por una función de Transferencia (F)
    - Función de activación
      - $a=x_1 \cdot w_1+\ldots+x_n \cdot w_n+b$
    - Función de transferencia
      - $y=F(x_1 \cdot w_1+\ldots+x_n \cdot w_n+b)$
    - Bias o polarización: entrada constante de magnitud 1, y peso b que se introduce en el sumador

¿Os recuerda a algo?



Neurona Artificial

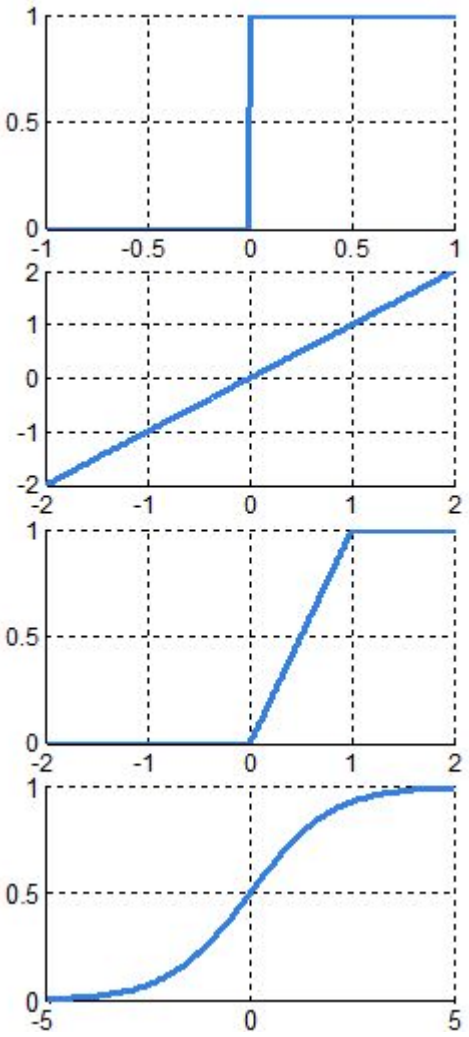
- La entrada (p) se multiplica por el peso (w).
- Otra entrada (1), se multiplica por un sesgo (b).
- La salida del sumador (n) se conoce como estímulo
- Va a la función de transferencia (f), la cual produce la salida (a) de la neurona.



Funciones de Transferencia

- 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: $a = \frac{1}{1+e^{-n}}$



Funciones de Transferencia

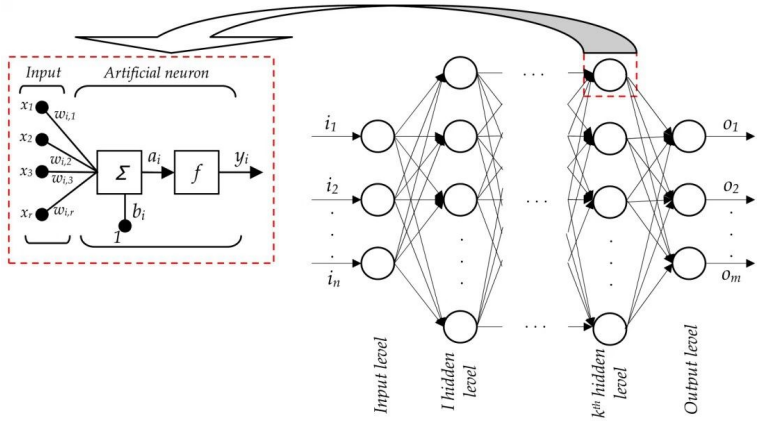
- Más funciones

¿qué las diferencia?

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	$(-1, 1)$	$\frac{2}{\pi}\arctan(g)$

Redes Neuronales

- 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

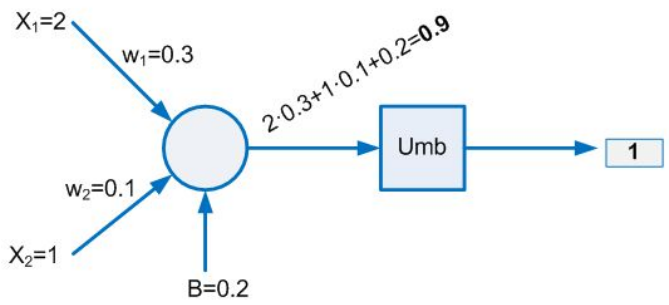


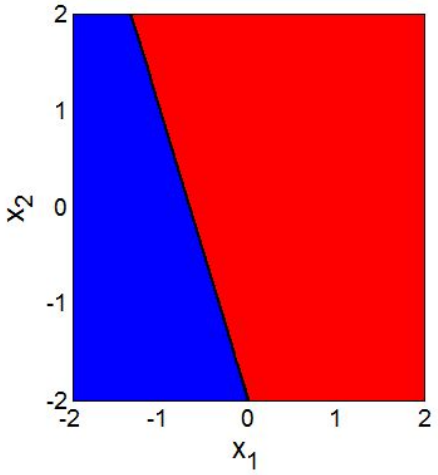
Redes Neuronales

- Capa de salida: es la capa final, que genera la respuesta a partir de la información recibida
- Para problemas de Regresión
  - Una neurona con función lineal o sigmoidal
- Para problemas de Clasificación
  - Binarios: con una neurona con función de activación umbral es suficiente
    - Puede usarse una función contínua (“probabilidad”)
  - Multi-clase: tantas neuronas como clases

Funcionamiento de la Neurona

- El modelo más sencillo es el perceptrón
  - Una neurona con función de activación umbral por cada salida necesaria recibe las entradas, las multiplica por los pesos y genera un resultado



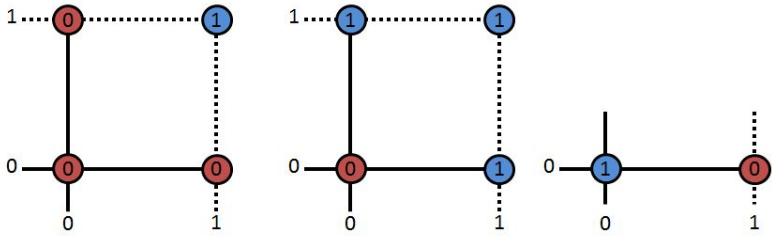


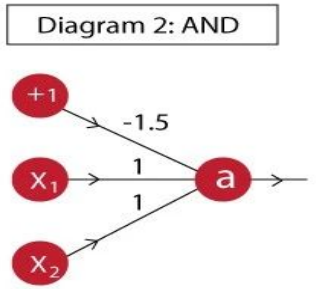
- Evaluación

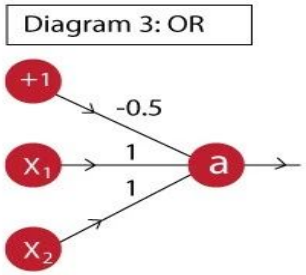
$$- \ a = Umb(b + \sum w \cdot x) = \begin{cases} 1, si\ (0{,}2 + 0{,}3 \cdot x_1 + 0{,}1 \cdot x_2) \geq 0 \\ -1, si\ (0{,}2 + 0{,}3 \cdot x_1 + 0{,}1 \cdot x_2) < 0 \end{cases}$$

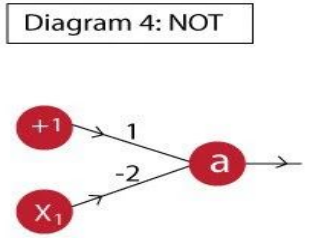
Funcionamiento de la Neurona

- Con un único perceptrón, se pueden implementar clasificadores lineales básicos, para datos tipo AND, OR y NOT
  - AND → a = f(-1.5 + x1 + x2)
  - OR → a = f(-0.5 + x1 + x2)
  - NOT → a = f(1 – 2*x1)









Deep neural network: Redes neuronales con múltiples capas ocultas

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

Mecanismo de aprendizaje

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

Mecanismo de aprendizaje

- 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.

Aprendizaje Supervisado

- Se provee a la red con un conjunto de datos de entrada y la “salida”
- Comparamos la salida deseada con la obtenida
  - Si coinciden No hay cambios
  - Si no coinciden
  - Ajustamos los pesos para que pueda tener una respuesta mejor en el futuro ante esa entrada
  - Las RNA, propagan esa diferencia hacia las capas para reajustar todos los pesos

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

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

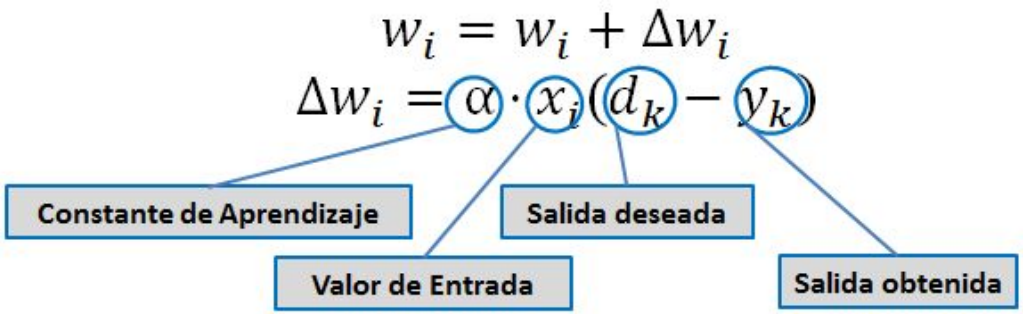
$$w_i = w_i + \Delta w_i$$

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

- α: Constante de Aprendizaje
- x_i: Valor de Entrada
- d_k: Salida deseada
- y_k: Salida obtenida

Aprendizaje del Perceptrón

- 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 xi negativos). Análogamente ocurre si es y=1 e d = 0
  - Cuando y=d, los w_i no se modifican

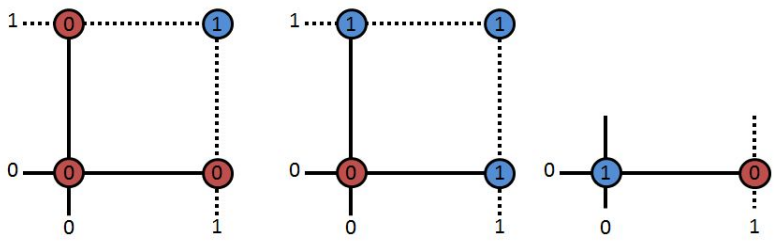


Aprendizaje del Perceptrón

- 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 α 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

Modelos de varias capas

- ¿Por qué más capas?
  - Una única neurona implementa una función lineal sobre los datos de entrada
    - (Corta los datos con una línea recta)
    - Para datos más complejos, se necesitarán más capas
  - Esa función será perfecta si los datos son linealmente separables
    - Eso no va a ocurrir en casos reales

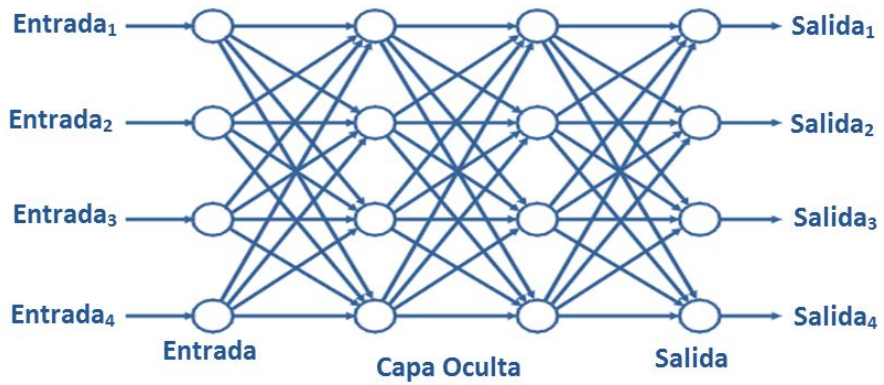


Modelos de varias capas

- 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

Modelos de varias capas

- 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
    - Funciones de transferencia sigmoideas en capas ocultas, normalmente



Aprendizaje en varias capas

- Backpropagation:
  - La red empieza fijando pesos de forma aleatoria
  - Proceso iterativo hasta alcanzar un criterio de parada
- Ciclos (épocas) 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

Número de capas ocultas

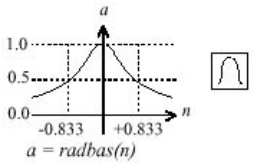
- Con muchos conjuntos de datos, una sóla capa puede ser suficiente
  - No hay teoría sobre cuántas capas ocultas son necesarias
  - Deep neural network: Redes con varias capas ocultas
- Teorema
  - *"It has been proven that a neural network with at least one hidden layer of sufficient neurons is a universal function approximator. This means that neural networks can be used to approximate any continuous function to an arbitrary precision over a finite interval"*

Número de Neuronas por Capa

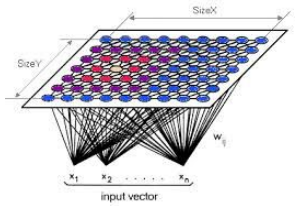
- Entrenar varias redes con distinto número de neuronas ocultas y estimar el error generalizado de cada red
  - Procedimiento simple: empezar sin neuronas ocultas y añadir una neurona cada vez.
  - Calcular el error de generalización de cada red
  - Dejar de añadir neuronas si aumenta la generalización del error (overfitting)
- Para una red con 2 capas ocultas
  - ¿Mejor con 1 neurona en la 1ª y 1000 en la 2ª?
  - ¿Mejor con 1000 neurona en la 1ª y 1 en la 2ª?

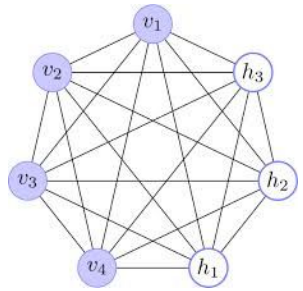
Otros modelos de redes

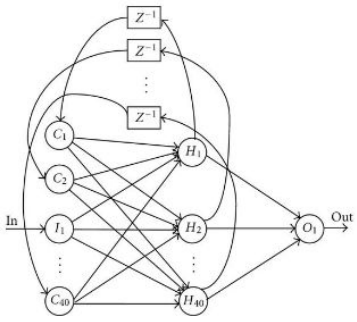
- Radial Basis Neural Network
  - Similar, neuronas con función de base radial
- Mapas auto-organizados
  - Redes Kohonen y Bolzman
    - Las neuronas se conectan en red
- Redes Neuronales Recurrentes
  - Redes Elman, Hopfield y Jordan
    - Las neuronas se conectan con neuronas de capas anteriores



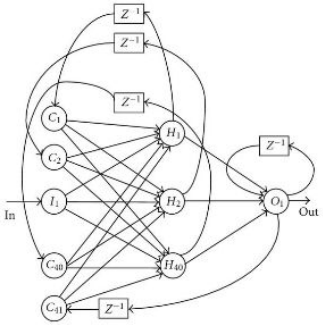
Radial Basis Function



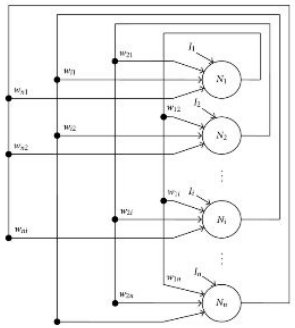




Elman Neural Network

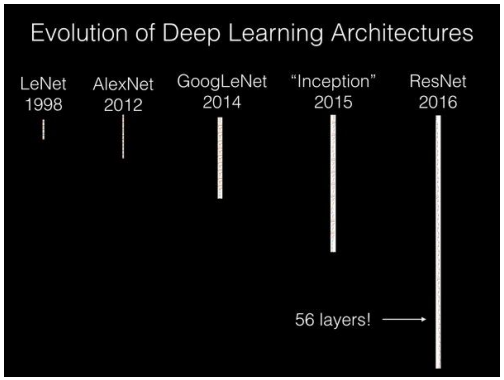


Jordan Neural Network


Hopfield Neural Network

Deep Learning (En 3 frases)

- Se basa en redes neuronales “clásicas”, pero...
  - Muchas (pero muchas) entradas
    - No le paso características de una imagen, le paso todos los valores de todos y cada uno de los píxeles
    - No le paso frecuencias de palabras, le paso todo el documento
  - Muchas capas
    - Del orden de decenas: podéis probar el tiempo que se tarda en entrenar una red conforme el número de capas (y neuronas) aumenta
  - Coste computacional muy (pero muy) grande
    - Mi ordenador echa humo
    - Trabajo en plataformas “Big”



Resúmen

- Un consejo: Al igual que con otros modelos, estandarizar/normalizar antes las entradas
- Ventajas
  - Pocas presunciones sobre relaciones entre las variables
  - Valen para clasificación y regresión
  - Modelizan cualquier problema complejo
  - Gran potencia (precisión)
- Desventajas
  - Caja negra, difícil interpretación
  - Fácil overfitting con estructuras neuronales complejas
  - Computacionalmente intensivas

Enrique Onieva
enrique.onieva@deusto.es
https://twitter.com/EnriqueOnieva
https://www.linkedin.com/in/enriqueonieva/