



PONTIFICIA  
**UNIVERSIDAD**  
**CATÓLICA**  
DEL PERÚ

# *INTELIGENCIA ARTIFICIAL*

## *INTRODUCCION A REDES NEURONALES ARTIFICIALES*

Dr. Edwin Rafael Villanueva Talavera

[ervillanueva@pucp.edu.pe](mailto:ervillanueva@pucp.edu.pe)

- Motivacion
- Perceptron
- Redes Perceptron Multicapa
- Entrenamiento
- Aplicacion

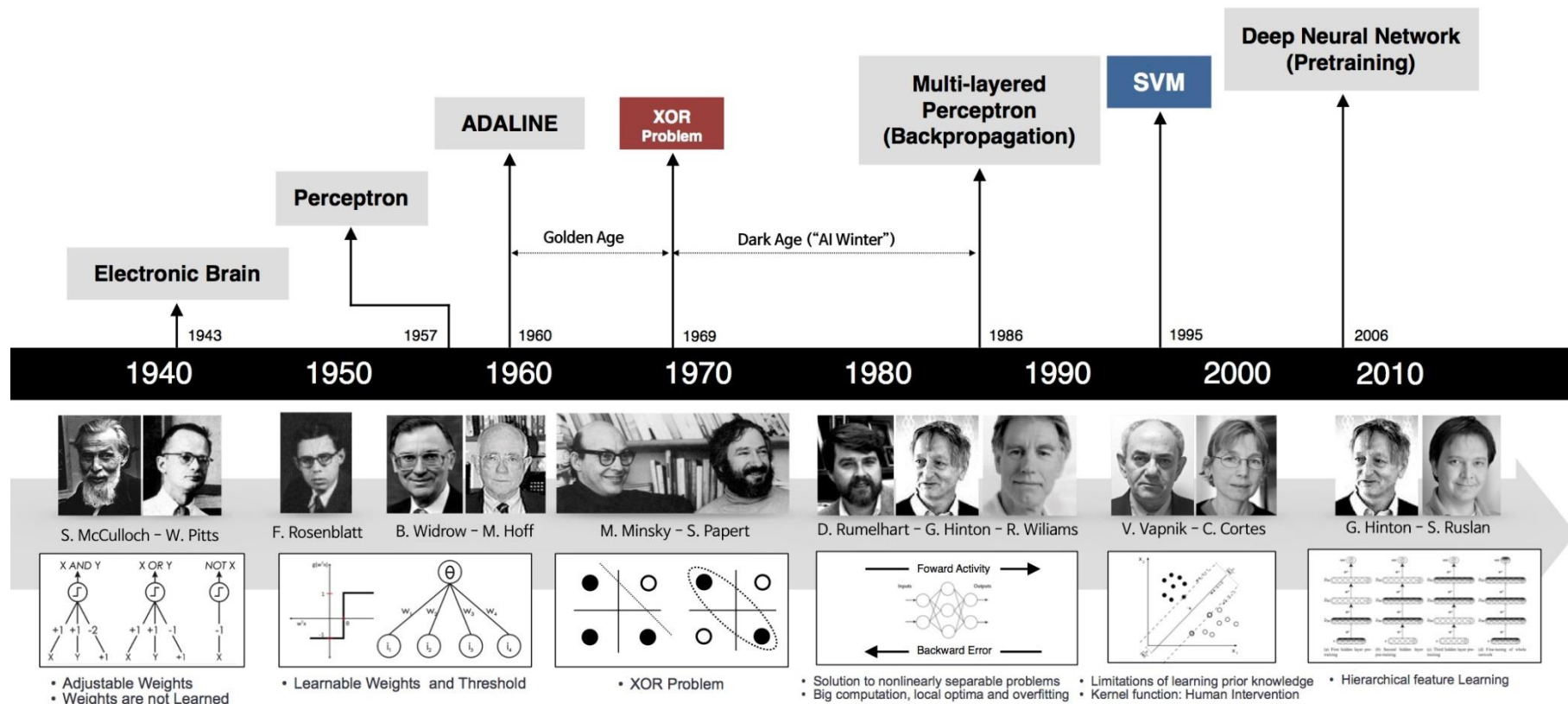
**RNA son resultado del deseo de construir artefactos capaces de exhibir comportamiento inteligente**



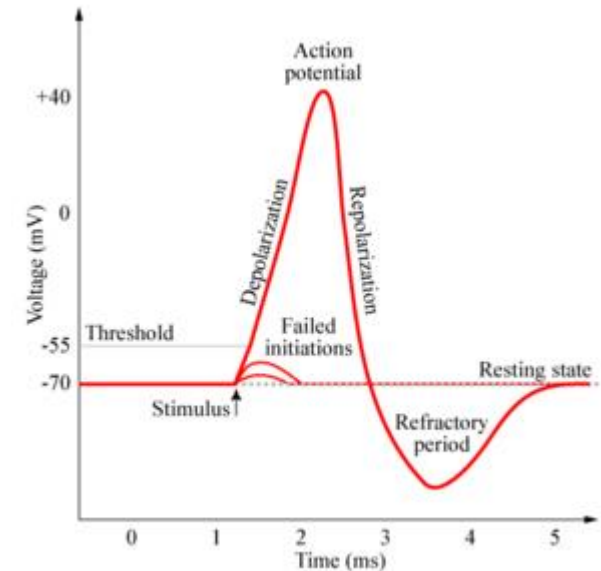
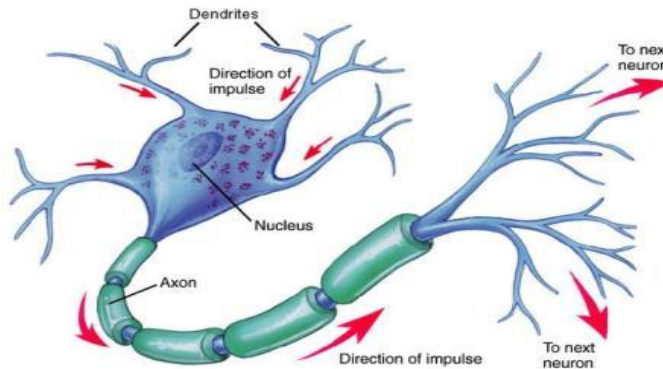
**RNA son inspiradas en el sistema nervioso central de animales:**

- El cerebro resuelve eficientemente problemas de procesamiento de imágenes, reconocimiento de habla, recuperación de información, aprendizaje basado en ejemplos, etc .

# Hitos Históricos



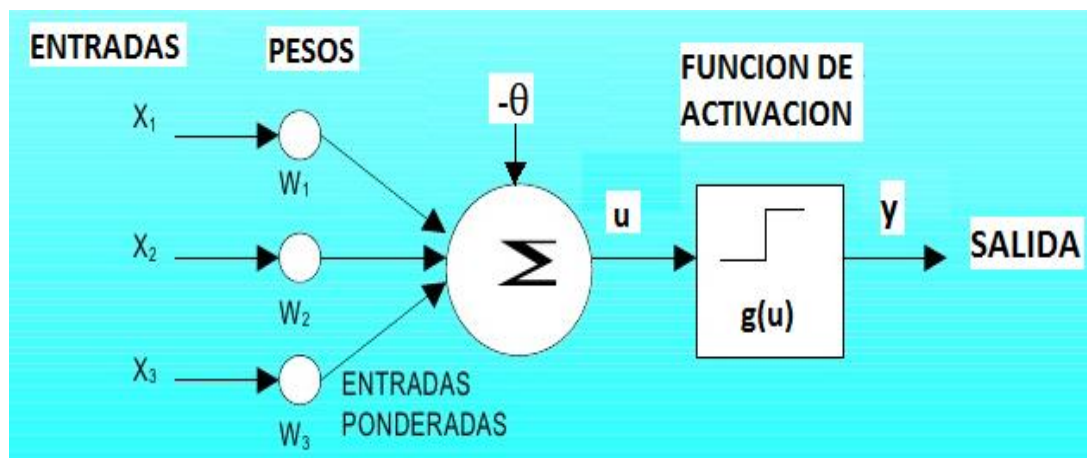
# Neurona Biológica



- Potencial eléctrico a través de la membrana de la célula exhibe picos.
- Pico se origina en el cuerpo celular, pasa por el axón, y hace que las terminaciones sinápticas generen neurotransmisores.
- Neurotransmisores pasan a través de las sinapsis hacia las dendritas de otras neuronas.
- Si la entrada total de neurotransmisores hacia una neurona ultrapasa un cierto limite, la neurona se dispara (genera un pico).

- Las neuronas biológicas se “activan” y “desactivan” en algunos milisegundos, mientras que el hardware actual hace lo mismo en apenas nanosegundos.
- Sistemas neuronales biológicos realizan tareas cognitivas complejas (visión, reconocimiento de voz) en décimas de segundo.
- Un sistema neuronal usa “paralelismo masivo”.
- El cerebro humano tiene  $10^{11}$  neuronas con una media de  $10^4$  conexiones por cada neurona.

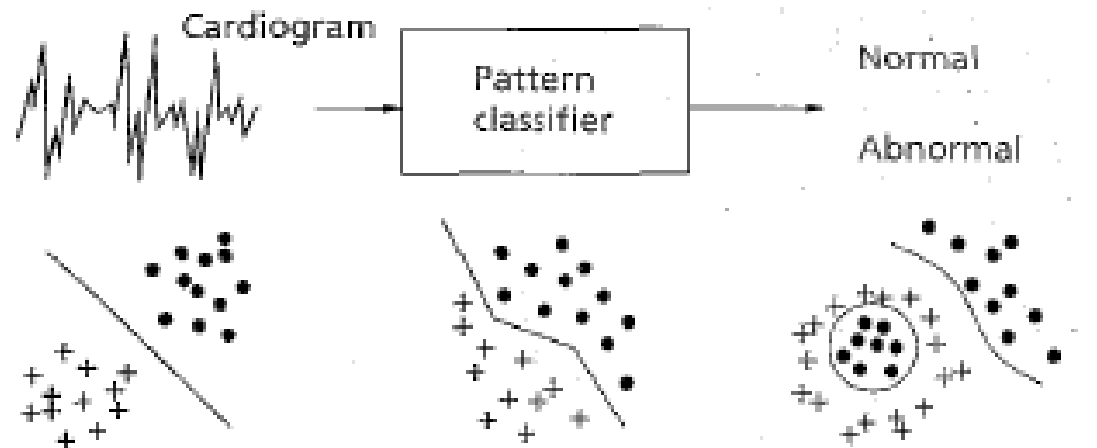
# Neurona Artificial



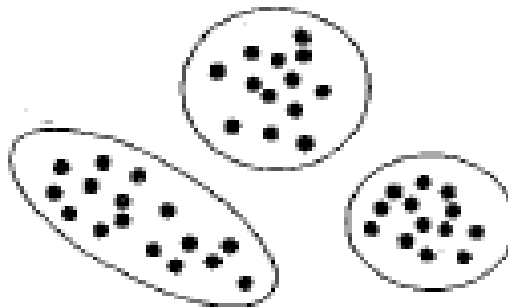
$$u = \left( \sum_{i=1}^N x_i \cdot w_i \right) - \theta$$

$$y = g(u)$$

- **Clasificación de patrones:** reconocimiento de imágenes, voz, retina, texto escrito, etc.

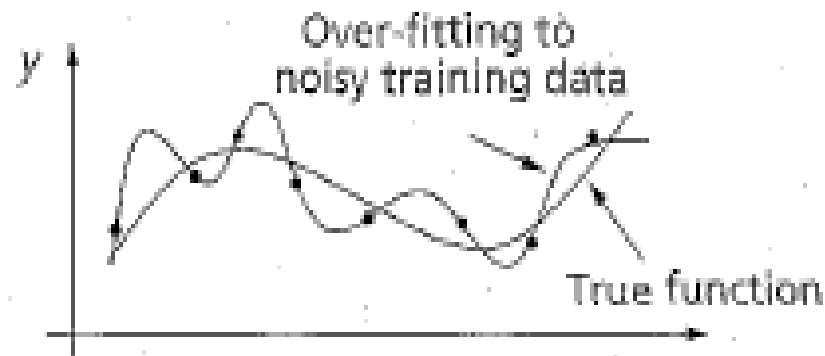


- **Clustering:** explorar similitudes y grupos, compresión de datos, etc.

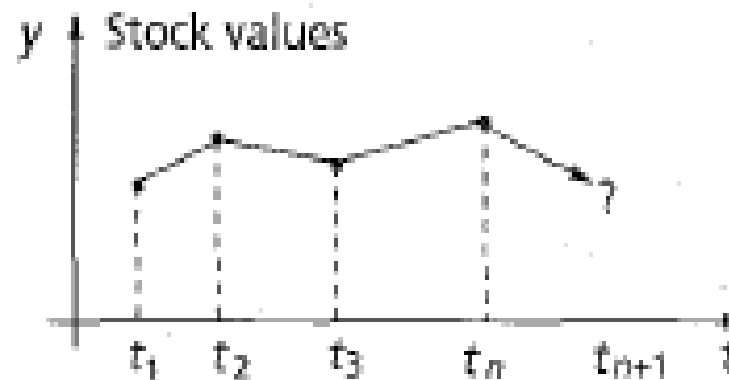




- **Aproximación de funciones:** modelamiento científico y ingeniería



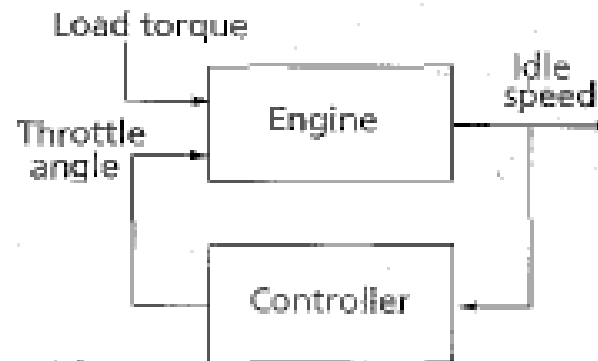
- **Previsión/Estimación:** Mercado financiero, previsión de clima, etc.



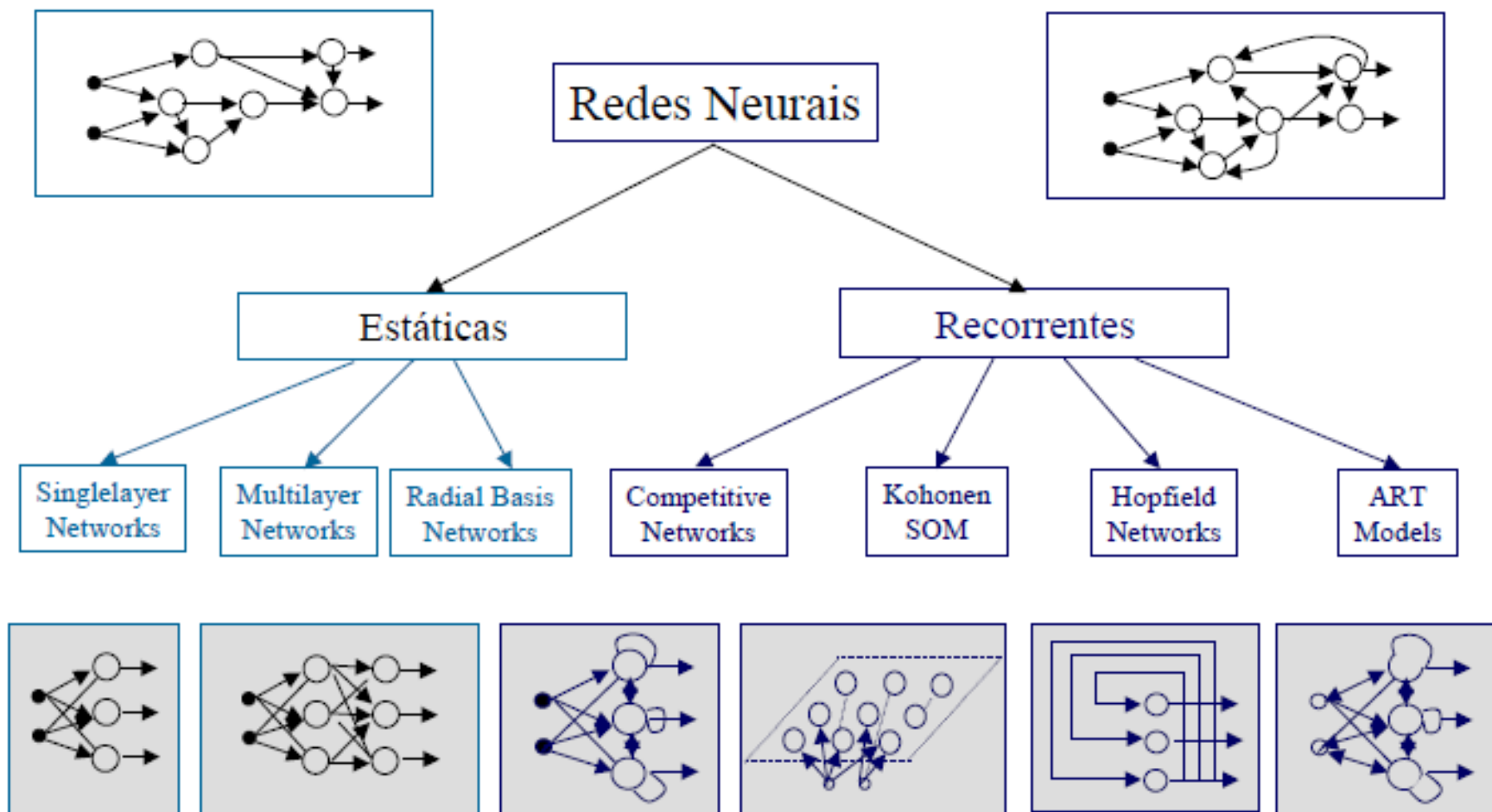
- **Memorias asociativas:** recuperar ítems por contenido, aún cuando la entrada sea distorsionada. (recuperación de imágenes, bases de datos, etc.)



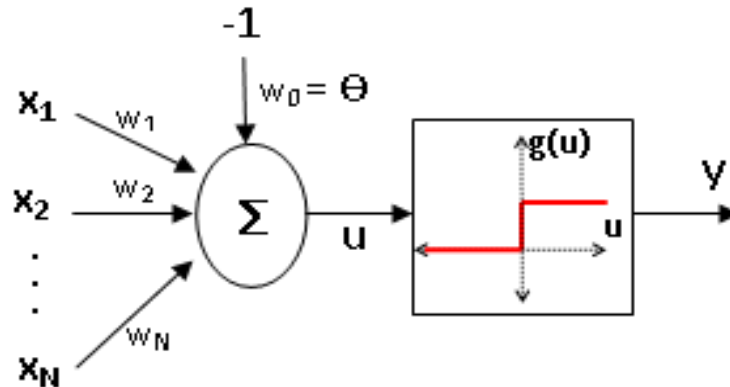
- **Control:** controlar sistemas de ingeniería (control de procesos, robótica, etc)



# Tipos de RNA



# Perceptron

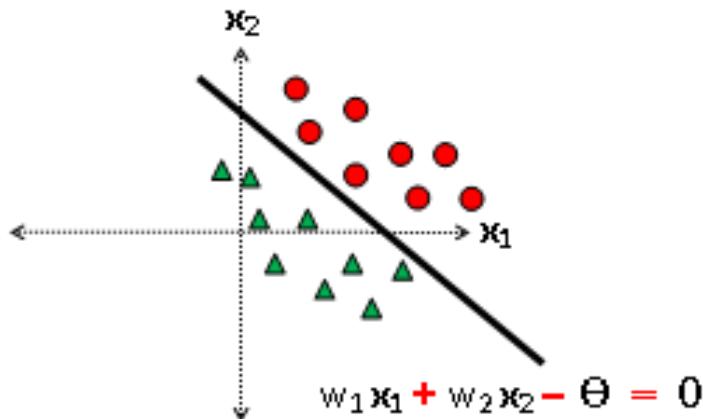


$$u = \left( \sum_{i=1}^N x_i \cdot w_i \right) - \theta$$

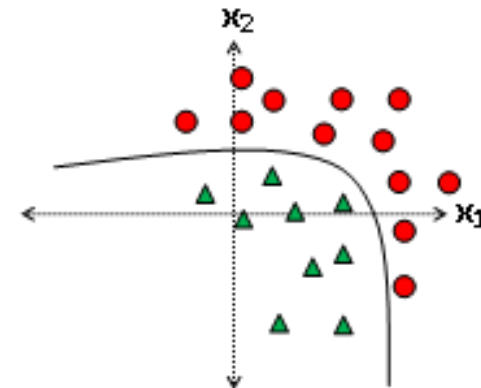
$$y = g(u)$$

## □ Clasificación con Perceptrons

Separa clases  
linealmente separables

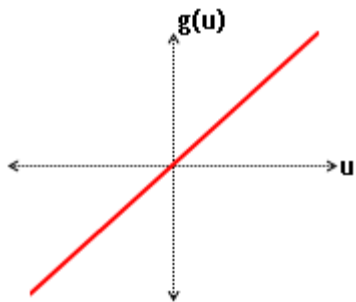


Es posible separar clases  
linealmente **No** separables

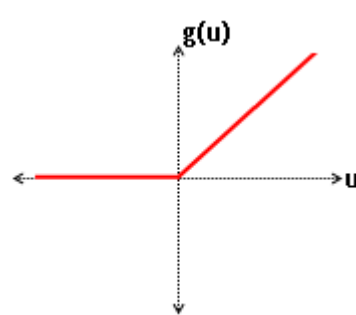


# Funciones de activación populares

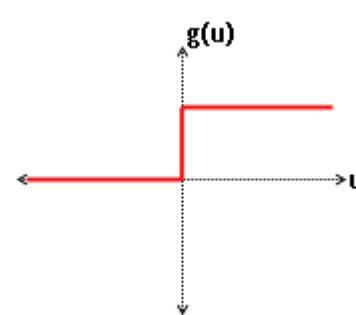
Función lineal



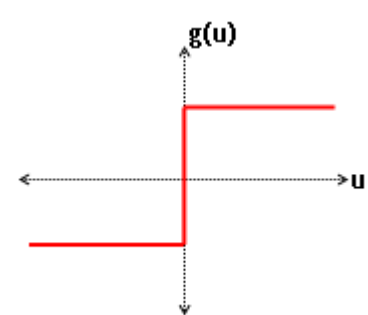
Función RELU



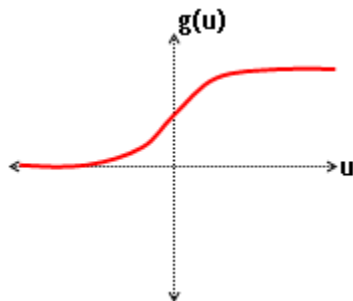
Función escalón



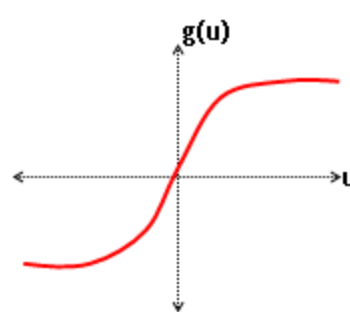
Función signo



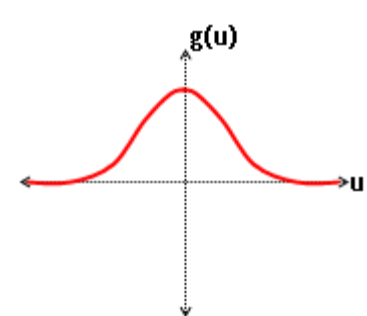
Función sigmoidea



Función tangente hiperbólica

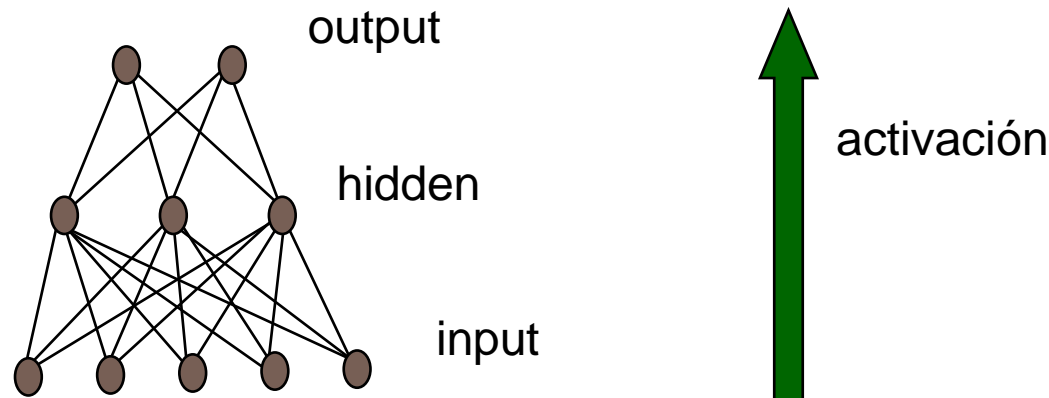


Función gaussiana



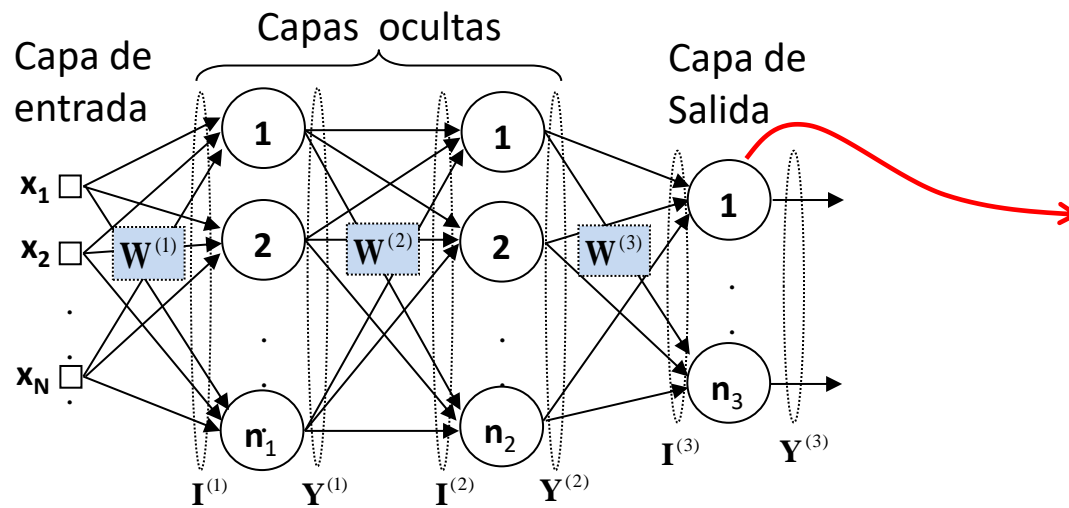
# Redes Perceptron Multicapa

- Las redes multicapa pueden representar funciones arbitrarias, sin embargo aprender esas redes era considerado un problema de difícil solución antes de los 90s.
- Una red multicapa **feed-forward** típica consta de capas de entrada, interna y salida, cada una totalmente conectada a la siguiente, con la activación yendo para adelante.

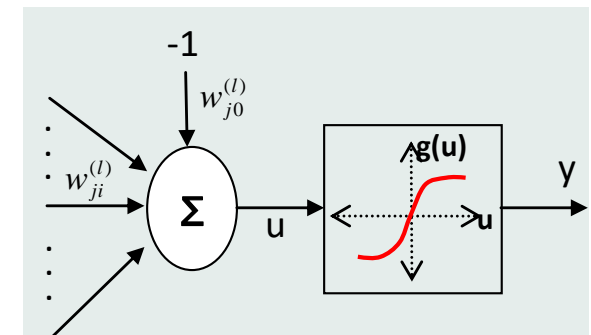


- Los pesos determinan la función calculada.

# Redes Perceptron Multicapa



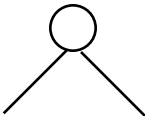
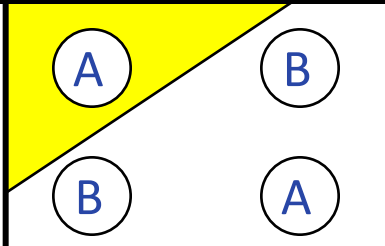
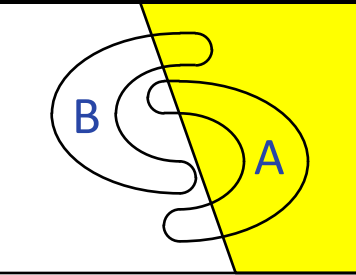
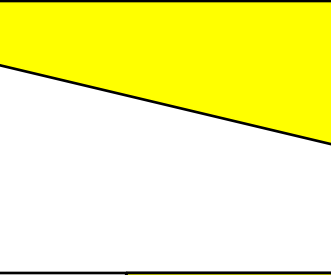
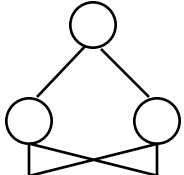
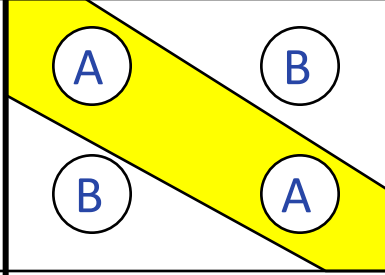
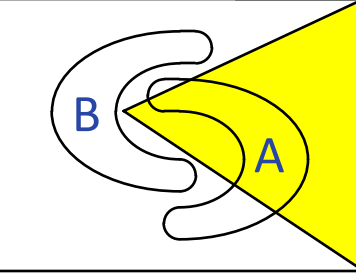
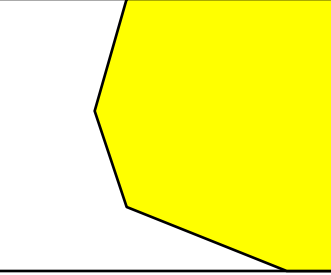
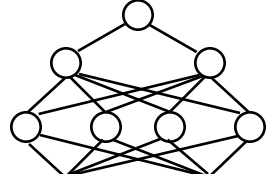
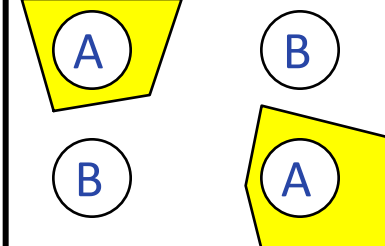
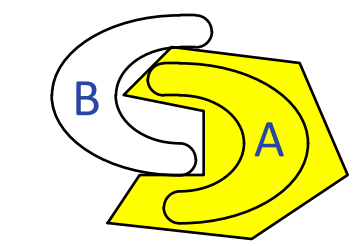
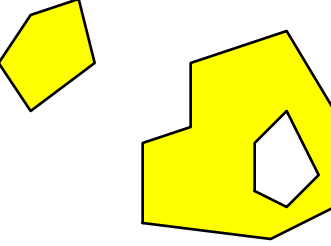
Cada neurona:



$$I_j^{(1)} = \sum_{i=1}^N w_{ji}^{(1)} \cdot x_i$$

$$Y_j^{(1)} = g(I_j^{(1)})$$

## □ Estructuras de redes Perceptron para clasificación

Estructura	XOR	Clases No-Convexas	Configuraciones Posibles
1 capa 			
2 capas 			
3 capas 			



- Utiliza un conjunto de ejemplos de entrenamiento con tuplas  $\langle \text{entrada}, \text{salida\_deseada} \rangle$
- El objetivo es **ajustar los pesos sinápticos** de tal forma que la red neuronal aproxime la salida deseada para cada ejemplo.
- Un algoritmo mas popular para el ajuste es el **backpropagation**. Este realiza actualizaciones iterativamente hasta conseguir resultados adecuados.

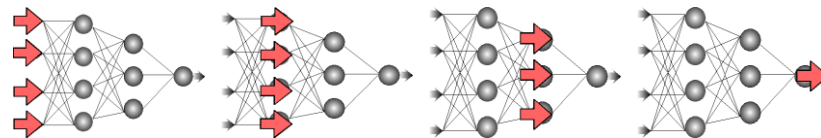
## Algoritmo backpropagation

1. Inicializar los pesos de la red (por ejemplo, aleatoriamente)
2. While (not\_critério\_parada)

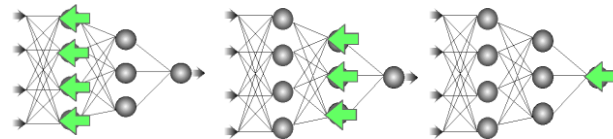
Para cada par entrada-salida  $\{\mathbf{x}^{(k)}, \mathbf{d}^{(k)}\}$

Forward pass: Calcular salida  $Y_k$  para entrada  $X^{(k)}$

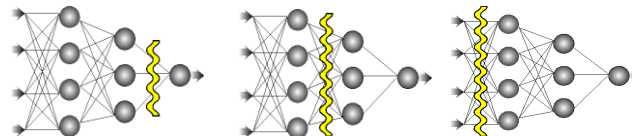
Calcular  $e_k = (Y_k - d^{(k)})$ , donde  $d^{(k)}$  es el target



Backward pass: Calcular  $\Delta w_{j,i}$  para cada capa  $j$  usando  
gradientes del error de cada neurona

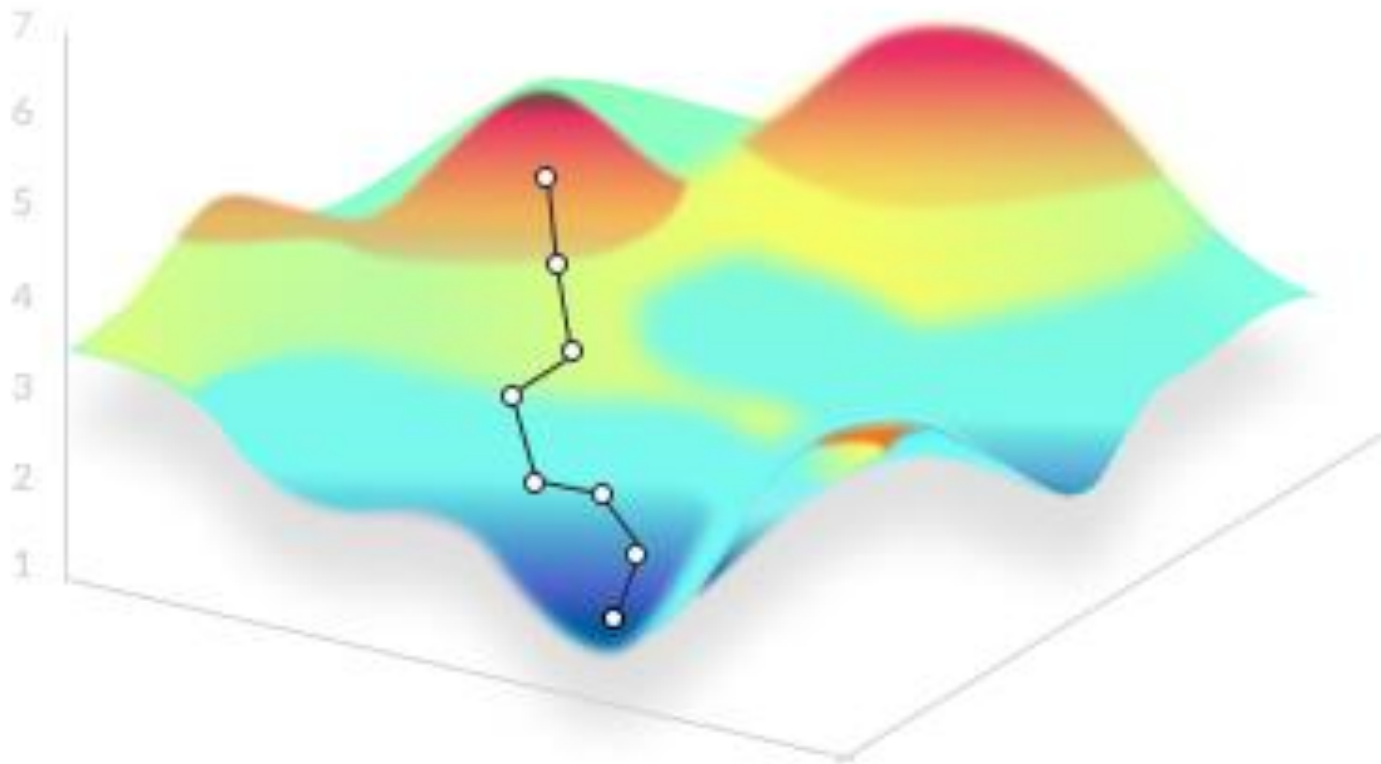


Actualizar pesos



end

## Superficie de optimización

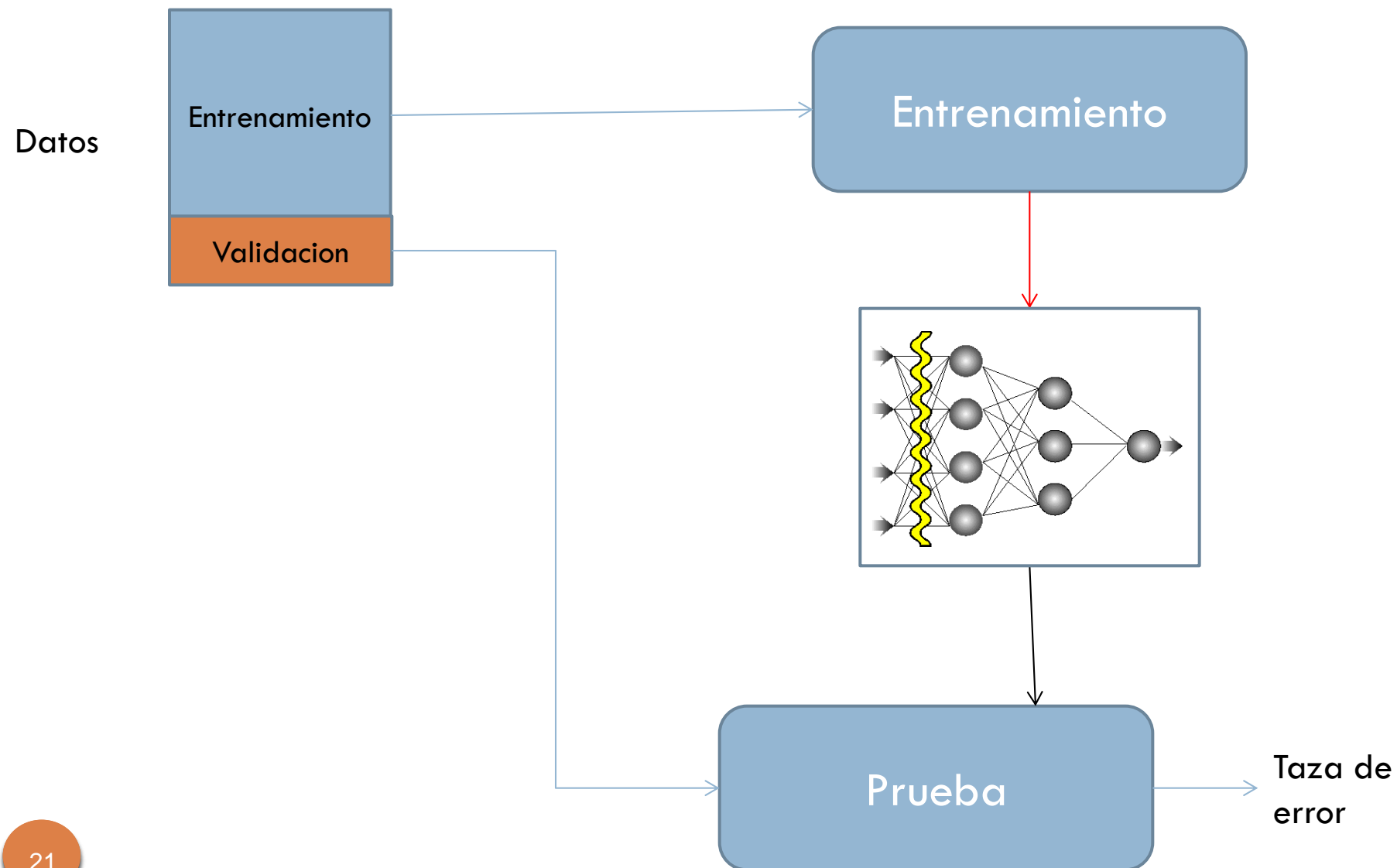


# Comentarios sobre backpropagation



- No garantiza optimalidad – puede convergir para óptimos locales u oscilar indefinidamente.
- Puede ser necesario elevada cantidad de épocas, lo que significa horas de entrenamiento para grandes redes.

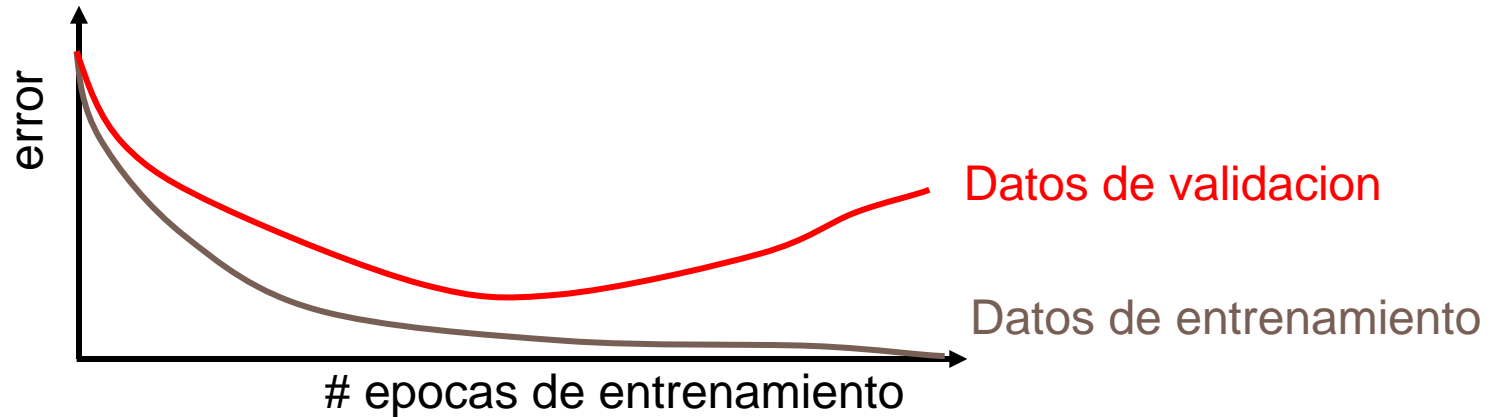
# Validación de Redes Neuronales



# Prevención de Sobreajuste (*Overfitting*)



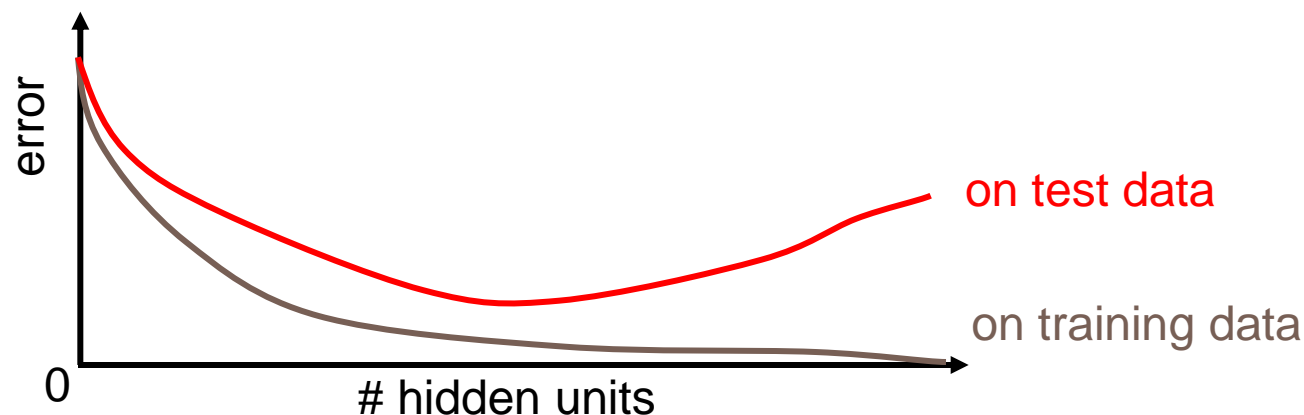
- Entrenar con muchas épocas puede llevar a sobreajuste.



- Una estrategia es usar un conjunto de validación y parar cuando el error comience a aumentar.

# Determinando el mejor número de neuronas

- Pocas neuronas pueden impedir que la red se adecue totalmente a los datos.
- Muchas neuronas pueden generar un sobreajuste.



- Usar validación cruzada interna para determinar empíricamente el mejor número de neuronas internas.

# Material Complementar

## □ Tutorial:

<https://missinglink.ai/guides/neural-network-concepts/backpropagation-neural-networks-process-examples-code-minus-math/>

## □ Demos para jugar con redes neuronales

- Redes Neuronales en tu browser

(<http://playground.tensorflow.org>)

- **ConvNetJS**, Convolutional Neural Network demo

(<https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html>)