

# Redes Neuronales & Java

M. D. López De Latorre

**Próximias charlas**

IEEE ARGENTINA UP Universidad de Palermo

Java y el XML	(D. López De Latorre)	27/10/05 (19:00hs)
La IA en Java	(D. López De Latorre)	17/11/05 (19:00hs)
Cognitive Memory	(S. Fichtelma)	21/11/05 (19:00hs)
Webbrowsing con Java	(D. López De Latorre)	24/11/05 (19:00hs)

informes:  
upgrade@palermo.edu sec.argentina@ieee.org

M. D. López De Latorre

## Objetivo

Presentar los fundamentos de las Redes Neuronales y su aplicación desde la perspectiva del lenguaje Java.

## Temario

- Terminología de la neurona
- El modelo neuronal
- Modelos y codificación de información
- Algunas redes famosas
- Librerías open source

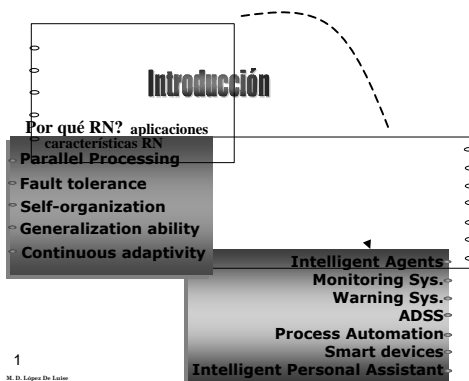
0  
M. D. López De Latorre

## Introducción

### Por qué RN?

- 1958 Surge con F. Rosenblat: Perceptrón
- 1969 M. Minsky y S. Papert: limitaciones del Perceptrón
- 1982 J. Hopfield: descripción matemática del modelo
- 1985 American Institute of Physics: annual meeting of NN
- 1987 IEEE: First International Conference on NN
- 1990 US Dep. of Defense: Smart Business Innovation Research Pgm
- 2001 IEEE: Neural Networks Society
- 2003 IEEE: Neural Networks local Chapter
- 2005 Soft(computing (CIS))
  - ▶ Genetic algorithms
  - ▶ Heuristics algorithms
  - ▶ Neural Networks aplicaciones
  - ▶ Fuzzy Logic
  - ▶ Evolutionary algorithms

1  
M. D. López De Latorre

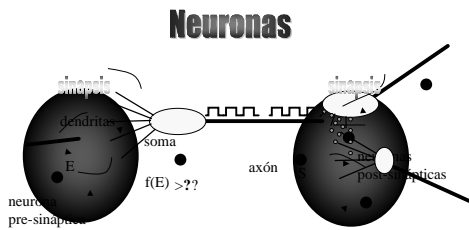


1  
M. D. López De Latorre

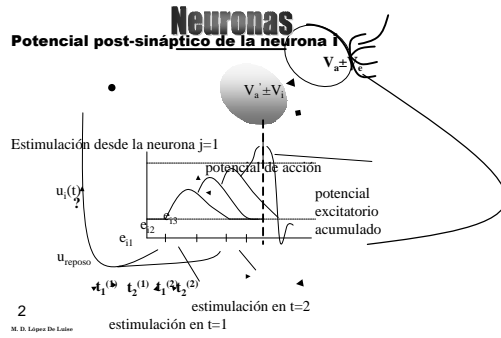
### Por qué no RN? algunos prejuicios

- Comportamiento interno complicado
- Precisión menor al 100%
- Cantidad de datos para entrenar
- Es un arte?

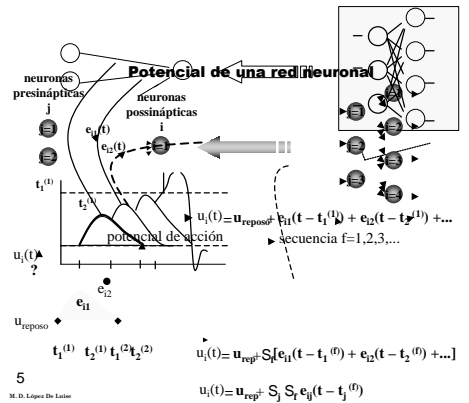
0  
M. D. López De Latorre



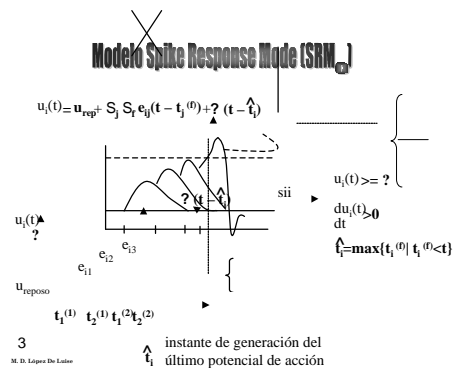
1  
M. D. López De Latorre



2  
M. D. López De Latorre



5  
M. D. López De Latorre



3  
M. D. López De Latorre

### Modelo SRM<sub>0</sub> : limitaciones del modelo

**Simplificaciones realizadas con este modelo**

- A** No todos los potenciales postsinápticos tienen igual forma.
- B** No es verdad que la dinámica de una neurona *i* depende principalmente del momento del último disparo.

**Limitaciones provocadas**

- A** No modeliza la adaptación celular hasta el estado de régimen
- B** No modeliza la dependencia del potencial con el estado pre-sináptico
- C** No modeliza la dependencia del estado interno de la neurona
- D** No modeliza la dependencia de la ubicación

1  
M. D. López De Latorre

### La codificación

**Avg. rate**  $\rightarrow ? = \frac{n_{sp}(T)}{T}$  Hz

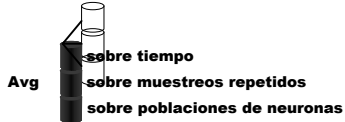
**Vs**

**Spikes (pulsos)**  $\rightarrow$

0  
M. D. López De Latorre

### Codificación de rates

#### Modelos de rate



0  
M. D. López De Latorre

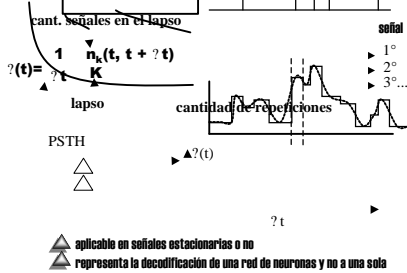
#### Modelos de rate: promedio temporal

El más común  $n_{sp}(T)$  Hz

- o adecuado p/ señales estacionarias o de lento cambio
    - sistema sensorial: mayor presión sobre la piel  $\Rightarrow$  más pulsos
    - sistema motor: más pulsos  $\Rightarrow$  mayor fuerza del músculo
  - o problemas:
    - hay actividades que rondan los 14 ms y no son bien modeladas así
    - esta no es bueno para señales no estacionarias (mayoría en el mundo real)
    - una célula neuronal no depende de una simple entrada
    - no representa las saturaciones (límites físicos en la reacción) (hay una curva de ganancia o función de transferencia)
- $V = g(I/I_0)$

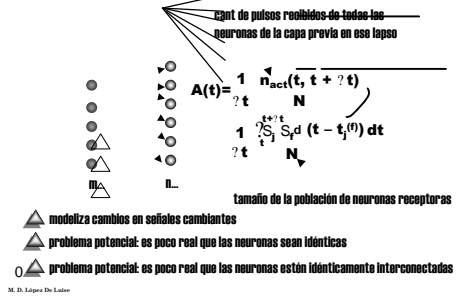
2  
M. D. López De Latorre

#### Modelos de rate: promedio sobre repeticiones



2  
M. D. López De Latorre

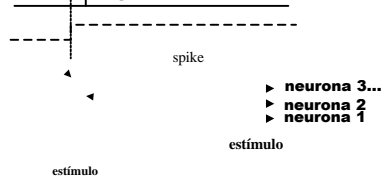
#### Modelos de rate: promedio sobre varias neuronas



M. D. López De Latorre

### Codificación de spikes

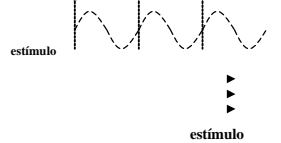
#### Modelos de spike: momento del primer pulso



ej. retina

0  
M. D. López De Latorre

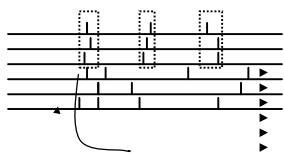
#### Modelos de spike: fase



ej. hipocampo

0  
M. D. López De Latorre

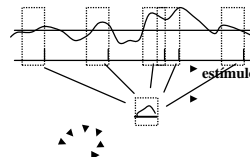
**Modelos de spike: correlaciones y sincronización**



simultaneidad de los spikes

0  
M. D. López De Latorre

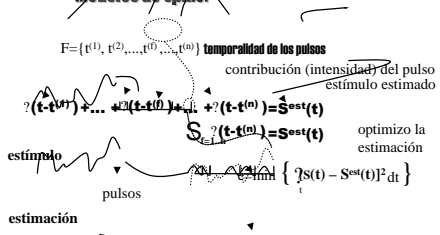
**Modelos de spike: correlación invertida**



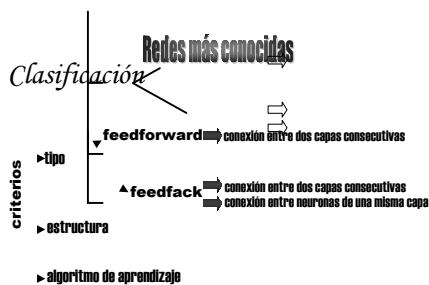
ej. córtex visual

0  
M. D. López De Latorre

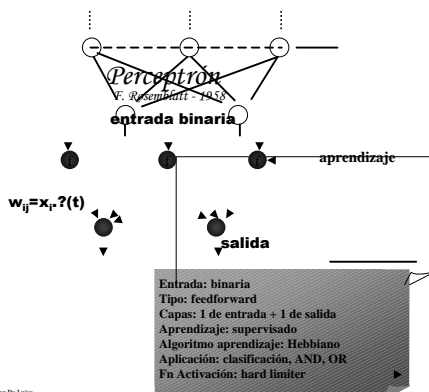
**Modelos de spike: reconstrucción de estímulo s(t)**



4  
M. D. López De Latorre

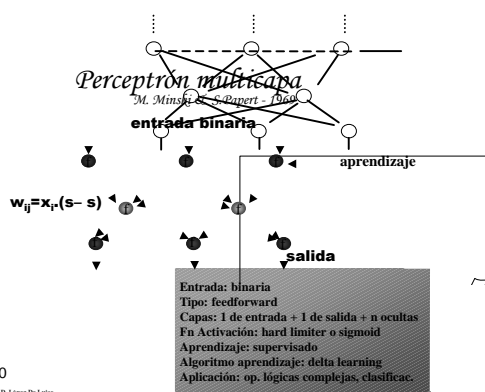


0  
M. D. López De Latorre



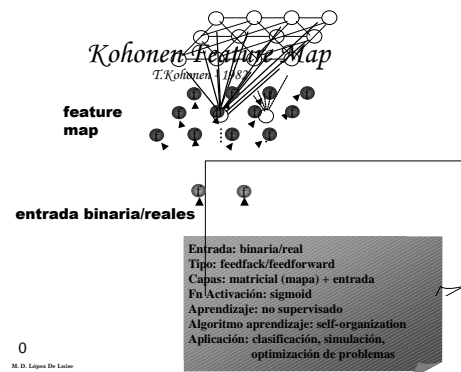
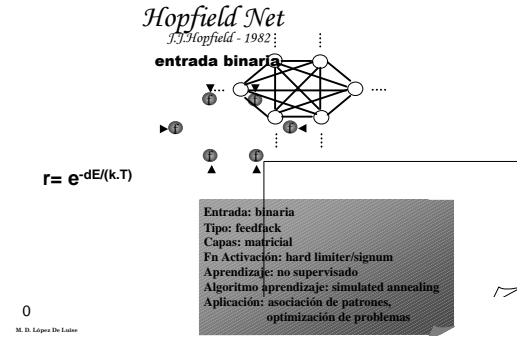
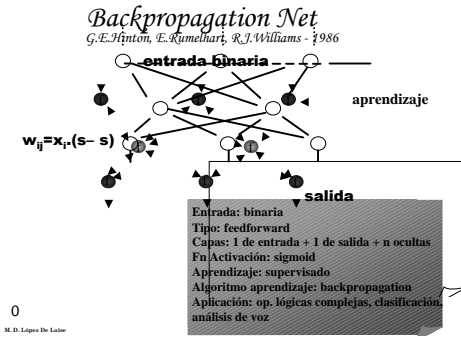
1  
M. D. López De Latorre

Entrada: binaria  
Tipo: feedforward  
Capas: 1 de entrada + 1 de salida  
Aprendizaje: supervisado  
Algoritmo aprendizaje: Hebbiano  
Aplicación: clasificación, AND, OR  
Fn Activación: hard limiter



0  
M. D. López De Latorre

Entrada: binaria  
Tipo: feedforward  
Capas: 1 de entrada + 1 de salida + n ocultas  
Fn Activación: hard limiter o sigmoid  
Aprendizaje: supervisado  
Algoritmo aprendizaje: delta learning  
Aplicación: op. lógicas complejas, clasificac.



**GNU GPL 2.0**  
 jahuwaldt.tools.NeuralNets

- **NeuronFactory**: generador de Neurons
- **BasicNeuron**: implementa una neurona
- **BasicNeuronFactory**: generador de BasicNeurons
- **HyperbolicNeuron**: neurona con fn activación = tg hiperbólica
- **HyperbolicNeuronFactory**: generador de BasicNeurons
- **InputNeuron**: implementa una input neuron
- **FeedForwardNet**: implementa una red feedforward
- **FeedForwardNetBP**: implementa una red con backpropagation
- **FeedForwardNetSCG**: implementa una red con *Scaled Conjugated Gradient*

**NeuronObservable** implementa una interface observable

**TrInstGenerator** implementa una sesión de training

0  
M. D. Lopez De Latorre



**F. Regensburg**  
 Dep. Computer Science

- Consiste en...
- Tiene un ejemplo completo de BackPN
- Tiene Aplica de demostración
- Recetas...

1. object declaration (ej. BackpropagationNet bpn;)
2. constructor call (ej. bpn = new BackpropagationNet();)
3. read conversion file (ej. bpn.readConversionFile("fileName");)
4. create input layer (ej. bpn.addNeuronLayer(i);)
5. create hidden layer(s) (ej. bpn.addNeuronLayer(h);)
6. create output layer (ej. bpn.addNeuronLayer(o);)
7. connect all layers (ej. bpn.connectLayers(i);)
8. read pattern file (ej. bpn.readPatternFile("fileName");)
9. perform a learning cycle (ej. bpn.learn();)

0  
M. D. Lopez De Latorre



M. D. Lopez De La Torre

## NNDef Toolkit 1.0.1

Usa un conjunto de archivos NNDEF.XML  
 (Soporta solo algunas redes simples (ej Perceptron y Perceptron(Multilayer))  
 Se pueden generar nuevos XML pero

usando NNDEF Generator  
 conforme a NNDEF DTD (document Type Def)  
 debe instalar NNDEFRun Library

Version command line (ej. java -jar NNDefRun.jar)  
 Herramientas de debugging & testing (java -jar NNDefRun.jar -log )  
 Puede usarse en multithreading

M. D. Lopez De La Torre



M. D. Lopez De La Torre

## wiki JOONE 1.2.0

Librerías de  
 aprendizaje supervisado  
 aprendizaje no supervisado  
 redes feedforward  
 redes feedback  
 redes temporales  
 preprocesamiento de datos  
 posibilidades de uso con JavaScript

M. D. Lopez De La Torre



M. D. Lopez De La Torre



M. D. Lopez De La Torre