

# Redes neuronales dinámicas

Diego Milone  
Inteligencia Computacional  
Departamento de Informática  
FICH-UNL

# Organización

- Introducción: 3 dinámicas simples
- Clasificación
- Redes de Hopfield
  - Arquitectura
  - Entrenamiento
  - Prueba
- Redes de Boltzman
  - Arquitectura
  - Templado simulado
- Retropropagación a través del tiempo
- Redes parcialmente recurrentes



# Introducción

¿Por qué dinámicas?



# Introducción

¿Por qué dinámicas?

Aproximación 1: entradas desplazadas



# Introducción

¿Por qué dinámicas?

Aproximación 1: entradas desplazadas

Aproximación 2: realimentación de las salidas



# Introducción

¿Por qué dinámicas?

Aproximación 1: entradas desplazadas

Aproximación 2: realimentación de las salidas

Aproximación 3: realimentación de estados internos

# Introducción

¿Por qué dinámicas?

Aproximación 1: entradas desplazadas

Aproximación 2: realimentación de las salidas

Aproximación 3: realimentación de estados internos

Caso general:

$$y(n) = f(\mathbf{x}(n), \mathbf{z}'(n), \mathbf{y}'(n))$$



# Clasificación

## Redes neuronales dinámicas (DNN)

- Redes con retardos en el tiempo (TDNN)

# Clasificación

## Redes neuronales dinámicas (DNN)

- Redes con retardos en el tiempo (TDNN)
- Redes recurrentes (RNN)
  - Redes totalmente recurrentes
  
- Redes parcialmente recurrentes (PRNN)



# Clasificación

## Redes neuronales dinámicas (DNN)

- Redes con retardos en el tiempo (TDNN)
- Redes recurrentes (RNN)
  - Redes totalmente recurrentes
    - Redes de Hopfield (memorias asociativas)
    - Redes de Boltzman (supervisadas)
    - Teoría de la resonancia adaptativa (ART)
  - Redes parcialmente recurrentes (PRNN)

# Clasificación

## Redes neuronales dinámicas (DNN)

- Redes con retardos en el tiempo (TDNN)
- Redes recurrentes (RNN)
  - Redes totalmente recurrentes
    - Redes de Hopfield (memorias asociativas)
    - Redes de Boltzman (supervisadas)
    - Teoría de la resonancia adaptativa (ART)
  - Redes parcialmente recurrentes (PRNN)
    - Retropropagación a través del tiempo (BPTT)
    - Redes de Elman
    - Redes de Jordan



# Redes de Hopfield

## Arquitectura

# Redes de Hopfield

## Arquitectura

Modelo matemático:

$$y_j = \operatorname{sgn} \left( \sum_{i=1}^N w_{ji} y_i - \theta_j \right) \dots \begin{cases} +1 & x > 0 \\ y_j(n-1) & x = 0 \\ -1 & x < 0 \end{cases}$$

$$w_{ji} = w_{ij} \quad \forall i \neq j$$

$$w_{ii} = 0 \quad \forall i$$

# Redes de Hopfield: generalidades

- Cada neurona tiene un disparo probabilístico
- Conexiones simétricas

# Redes de Hopfield: generalidades

- Cada neurona tiene un disparo probabilístico
- Conexiones simétricas
- El entrenamiento es no-supervisado

## Redes de Hopfield: generalidades

- Cada neurona tiene un disparo probabilístico
- Conexiones simétricas
- El entrenamiento es no-supervisado
- Puede utilizarse como memoria asociativa

## Redes de Hopfield: generalidades

- Cada neurona tiene un disparo probabilístico
- Conexiones simétricas
- El entrenamiento es no-supervisado
- Puede utilizarse como memoria asociativa
- Cada patrón es un “valle” de energía
- Se busca el valle iterativamente a partir de ciertas condiciones iniciales (patrones sucios o incompletos)

# Entrenamiento (almacenamiento)

Dado un conjunto de patrones (memorias fundamentales o datos limpios):

$$X^* = \{\mathbf{x}_k^* \in \mathbb{R}^N\}$$

# Entrenamiento (almacenamiento)

Dado un conjunto de patrones (memorias fundamentales o datos limpios):

$$X^* = \{\mathbf{x}_k^* \in \mathbb{R}^N\}$$

Aprendizaje Hebbiano:

$$w_{ji} = \frac{1}{N} \sum_{k=1}^P x_{kj}^* x_{ki}^*$$

# Entrenamiento: observaciones

- El proceso de entrenamiento NO es iterativo

## Entrenamiento: observaciones

- El proceso de entrenamiento NO es iterativo
- $w_{ji}$  es mayor cuando las neuronas  $i$  y  $j$  se tienen que activar juntas (regla de Hebb)

## Entrenamiento: observaciones

- El proceso de entrenamiento NO es iterativo
- $w_{ji}$  es mayor cuando las neuronas  $i$  y  $j$  se tienen que activar juntas (regla de Hebb)
- La capacidad de almacenamiento está limitada a:

$$P_{max} = \frac{N}{2 \ln(N)}$$

con un 1 % de error.

# Prueba (recuperación)

Dado un patrón  $\mathbf{x}$  (incompleto, ruidoso...) se fuerza:

$$\mathbf{y}(0) = \mathbf{x}$$

# Prueba (recuperación)

Dado un patrón  $\mathbf{x}$  (incompleto, ruidoso...) se fuerza:

$$\mathbf{y}(0) = \mathbf{x}$$

Iteración:

1.  $j^* = \text{rnd}(N)$
2.  $y_{j^*}(n+1) = \text{sgn}\left(\sum_{i=1}^N w_{ji}y_i(n)\right)$
3. volver a 1 hasta no observar cambios en las  $y_j$

# Prueba: observaciones

- El proceso de recuperación ES iterativo (dinámico)



## Prueba: observaciones

- El proceso de recuperación ES iterativo (dinámico)
- En general no se utilizan los  $\theta_j$

## Prueba: observaciones

- El proceso de recuperación ES iterativo (dinámico)
- En general no se utilizan los  $\theta_j$
- La salida final es  $\mathbf{y}(M)$  cuando no hay cambios al “recorrer” todas las salidas

## Prueba: observaciones

- El proceso de recuperación ES iterativo (dinámico)
- En general no se utilizan los  $\theta_j$
- La salida final es  $\mathbf{y}(M)$  cuando no hay cambios al “recorrer” todas las salidas
- Se pueden obtener estados espúreos y oscilaciones...

# Campos energéticos de Hopfield

- Almacenamiento
- Recuperación

# Redes de Boltzman: generalidades

- Conexiones simétricas
- Entrenamiento supervisado
- Posee capa oculta
- Templado simulado (annealing)



# Redes de Boltzman

## Arquitectura

# Redes de Boltzman: entrenamiento

- ¿Cuál es el principal problema de los métodos de gradiente?

## Redes de Boltzman: entrenamiento

- ¿Cuál es el principal problema de los métodos de gradiente?
- Principio del templado simulado: *ir gradiente abajo la mayoría de las veces pero NO TODAS.*

## Redes de Boltzman: entrenamiento

- ¿Cuál es el principal problema de los métodos de gradiente?
- Principio del templado simulado: *ir gradiente abajo la mayoría de las veces pero NO TODAS.*
- Ejemplo: templado de acero



# Retropropagación a través del tiempo (BPTT)

- Arquitectura con recurrencia total



# Retropropagación a través del tiempo (BPTT)

- Arquitectura con recurrencia total
- Expansión en red de propagación hacia adelante pura

# Retropropagación a través del tiempo (BPTT)

- Arquitectura con recurrencia total
- Expansión en red de propagación hacia adelante pura
- Método de truncado (recurrencia parcial)



# Otras redes que modelan dinámicas temporales

- Redes neuronales con retardos en el tiempo (TDNN)  
Arquitectura y entrenamiento

# Otras redes que modelan dinámicas temporales

- Redes neuronales con retardos en el tiempo (TDNN)  
Arquitectura y entrenamiento  
Ejemplo: Waibel89, clasificación espacio-temporal,  
reconocimiento automático del habla



## Otras redes que modelan dinámicas temporales

- Redes neuronales con retardos en el tiempo (TDNN)  
Arquitectura y entrenamiento  
Ejemplo: Waibel89, clasificación espacio-temporal,  
reconocimiento automático del habla
- Arquitectura neuronal de Elman



## Otras redes que modelan dinámicas temporales

- Redes neuronales con retardos en el tiempo (TDNN)  
Arquitectura y entrenamiento  
Ejemplo: Waibel89, clasificación espacio-temporal,  
reconocimiento automático del habla
- Arquitectura neuronal de Elman
- Arquitectura neuronal de Jordan