Redes neuronales dinámicas

Diego Milone

Inteligencia Computacional
Departamento de Informática

FICH-UNL

Introducción: ¿por qué dinámicas?

Introducción: ¿por qué dinámicas?

Aproximación 1: entradas desplazadas

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

Introducción: ¿por qué dinámicas?

Aproximación 1: entradas desplazadas

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

Aproximación 2: realimentación de las salidas

$y(n) = f(\mathbf{x}(n), \mathbf{y}_1(n))$

Introducción: ¿por qué dinámicas?

Aproximación 1: entradas desplazadas

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

Aproximación 2: realimentación de las salidas

$y(n) = f(\mathbf{x}(n), \mathbf{y}_1(n))$

Aproximación 3: realimentación de estados internos

$y(n) = f(\mathbf{x}(n), \mathbf{z}_1(n))$

Introducción: ¿por qué dinámicas?

Aproximación 1: entradas desplazadas
$y(n) = f(\mathbf{x}(n))$

Aproximación 2: realimentación de las salidas
$y(n) = f(\mathbf{x}(n), \mathbf{y}_1(n))$

Aproximación 3: realimentación de estados internos
$y(n) = f(\mathbf{x}(n), \mathbf{z}_1(n))$

Caso general:

$$y(n) = f(\mathbf{x}(n), \mathbf{z}_1(n), \mathbf{y}_1(n))$$

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

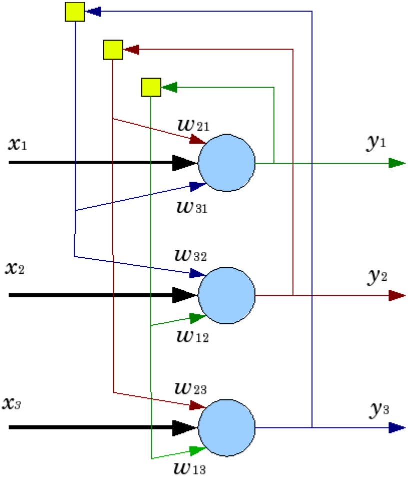
Diego Milone

Inteligencia Computacional
Departamento de Informática

FICH-UNL

Redes de Hopfield

Arquitectura



Redes de Hopfield

Modelo matemático

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

$$w_{ji} = w_{ij} \quad \forall\, i \neq j$$
$$w_{ii} = 0 \quad \forall i$$

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
Almacenamiento y recuperación

Diego Milone

Inteligencia Computacional
Departamento de Informática

FICH-UNL

Entrenamiento (almacenamiento)

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

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

Entrenamiento (almacenamiento)

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

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

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. $j^* = rnd(N)$
2. $y_{j^*}(n) = sgn\left(\sum_{i=1}^{N} w_{ji} y_i(n-1)\right)$
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 θ_j

Prueba: observaciones

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

Retropropagación a través del tiempo

Diego Milone

Inteligencia Computacional
Departamento de Informática

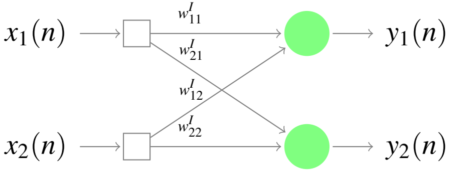
FICH-UNL

Retropropagación a través del tiempo (BPTT)

Arquitectura con recurrencia total

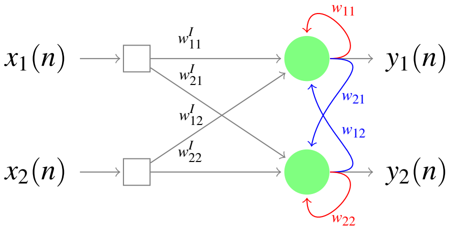
Retropropagación a través del tiempo (BPTT)

Arquitectura con recurrencia total

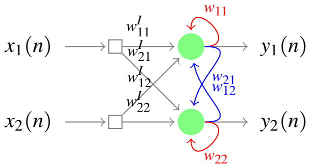


Retropropagación a través del tiempo (BPTT)

Arquitectura con recurrencia total



Expansión: propagación hacia adelante pura



Redes neuronales con retardos en el tiempo

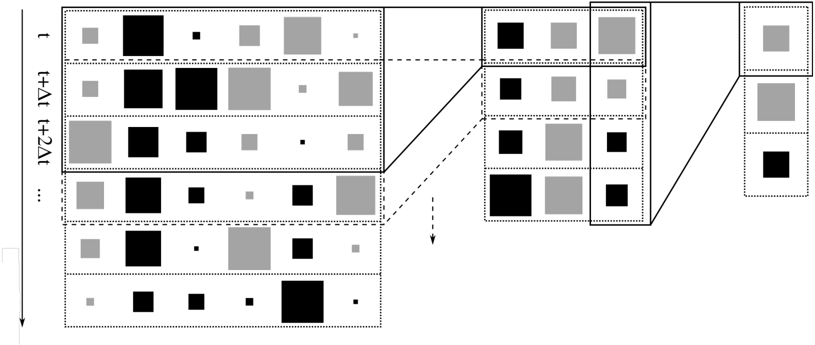
Diego Milone

Inteligencia Computacional
Departamento de Informática

FICH-UNL

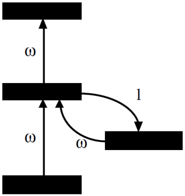
Arquitectura de una TDNN

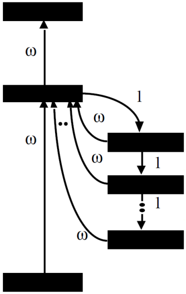
TDNN: clasificación espacio-temporal



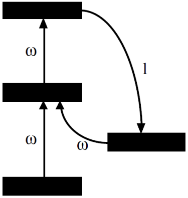
Arquitecturas neuronales de Elman y Jordan

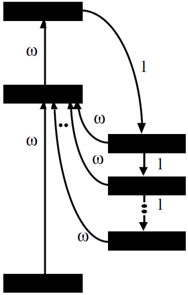
Elman





Arquitecturas neuronales de Elman y Jordan





Jordan