



Universidad
Carlos III de Madrid

Programación Automática

MÁSTER EN CIENCIA Y TECNOLOGÍA INFORMÁTICA

Ricardo Aler Mur, Fernando Fernández Rebollo,
Raquel Fuentetaja Pizan, David Quintana Montero





Programación Automática (PA)

Ricardo Aler Mur
Fernando Fernández Rebollo
Raquel Fuentetaja Pizán
David Quintana Montero

Universidad Carlos III de Madrid



¿Qué es la PA?

- Que un ordenador sea capaz de generar otros programas de ordenador
- Que un programa sea capaz de generar otros programas de ordenador
- El campo de aplicación es vasto.
- Se puede enfocar desde varios puntos de vista:
 - Ingeniería del Software:
 - Optimización de código (compiladores), generación automática de código a partir de UML, ...
 - Inteligencia Artificial:
 - Aprendizaje automático avanzado



Motivación de la asignatura Programación Automática

- Aprendizaje automático avanzado
- Estudiar diversas maneras en que un ordenador puede aprender estructuras equivalentes a programas, en dos sentidos:
 - Que incluyan el tiempo o varios pasos (bucles, secuencias, ...)
 - Representaciones potentes (lógica, lenguajes Turing-completos, políticas, automatas finitos, ...)
- Un programa es lo más **GENERAL** que el aprendizaje automático puede producir (mas que los árboles de decisión, redes de neuronas, ...)



UN EJEMPLO DE PROBLEMA DE CLASIFICACIÓN:

- Concesión de créditos bancarios
 - Un banco por Internet desea obtener reglas para predecir qué personas de las que solicitan un crédito no van a devolverlo.
 - La entidad bancaria cuenta con una gran base de datos correspondientes a los créditos concedidos (o no) a otros clientes con anterioridad.
 - Instancias (de la base de datos del banco):
 - Atributos de entrada: años del crédito, cuantía del crédito, tiene cuentas morosas, tiene casa propia
 - Clase: si/no
 - Modelo que se podría aprender:
 - SI** (cuentas-morosas > 0) **ENTONCES** Devuelve-crédito = no
 - SI** (cuentas-morosas = 0) **Y** ((salario > 2500) **O** (años > 10)) **ENTONCES** devuelve-crédito = si

UN EJEMPLO DE PROBLEMA DE CLASIFICACIÓN CON REGLAS:

Años	Euros	Salario	Casa propia	Cuentas morosas	Crédito
10	50000	3000	Si	0	??

Años	Euros	Salario	Casa propia	Cuentas morosas	Crédito
15	60000	2200	Si	2	No
2	30000	3500	Si	0	Si
9	9000	1700	Si	1	No
15	18000	1900	No	0	Si
10	24000	2100	No	0	No
...

Algoritmo

IF CM > 0 THEN NO

**IF CM = 0 Y
S > 2500 THEN SI**

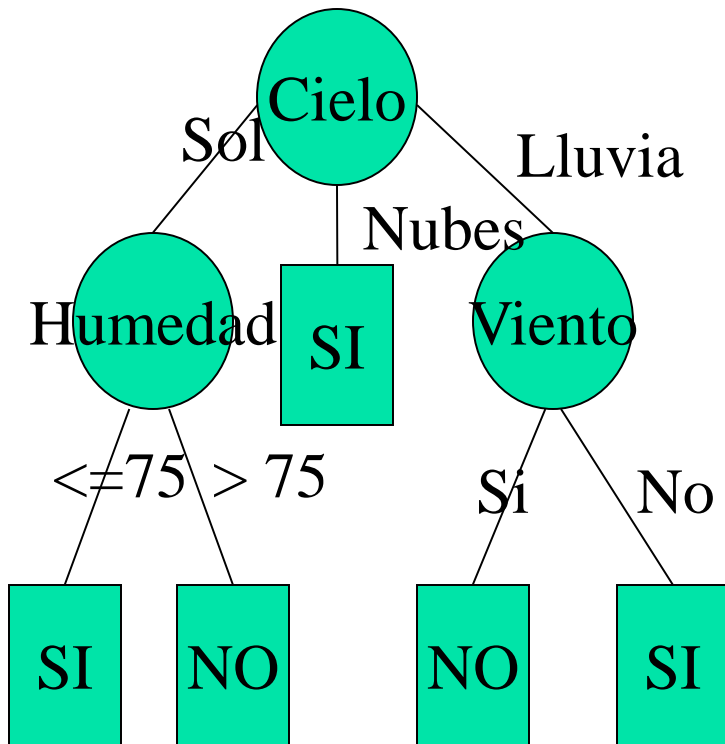
Crédito = Si

x (o atributos de entrada)

y (clase, o atributo de salida)

Aprendizaje Automático

Proposicional



IF Cielo = Sol

 Humedad \leq 75 **THEN** Tenis = Si

ELSE IF Cielo = Sol

 Humedad $>$ 75 **THEN** Tenis = No

ELSE IF Cielo = Nubes **THEN** Tenis = Si

ELSE IF Cielo = Lluvia

 Viento = Si **THEN** Tenis = Si

ELSE Tenis = No



Características Aprendizaje Automático proposicional

- Las estructuras que se aprenden actúan en un solo paso:
 - Instancia a la entrada
 - Clase a la salida (moroso / no moroso)
- El lenguaje de las estructuras que se aprenden es sencillo (proposicional)
 - Ej: no aparecen bucles



Ejemplo programación automática

- **Entrada:**

- `([2,1], [1,2]); ([2,3,1], [1,2,3]);`
- `([3,5,4], [3,4,5]); ([],[]); ...`

- **Instrucciones primitivas:**

- `(dobl start end work) (wismaller x y)`
- `(swap x y) (wibigger x y)`
- `(e1+ x) (e- x y) (e1- x)`



Programación Genética

```
(dobl (wismaller (wismaller (el- *len*) *len*)
                 index)
      (dobl (wismaller index
                (wismaller
                 (el- index)
                 (el+ (el- index))))
            (el- *len*)
            (swap (swap (el- *len*) index)
                  index))
      (dobl (swap (wibigger index (e- index *len*))
                (e- index *len*))
            (el- *len*)
            (swap (wismaller (el+ index) index)
                  index))))
```



Equivalente a Bubble-sort

```
(dobl 0
  (el- *len*)
  (dobl 0
    (el- *len*)
    (swap (wismaller (el+ index) index)
           index)))
```

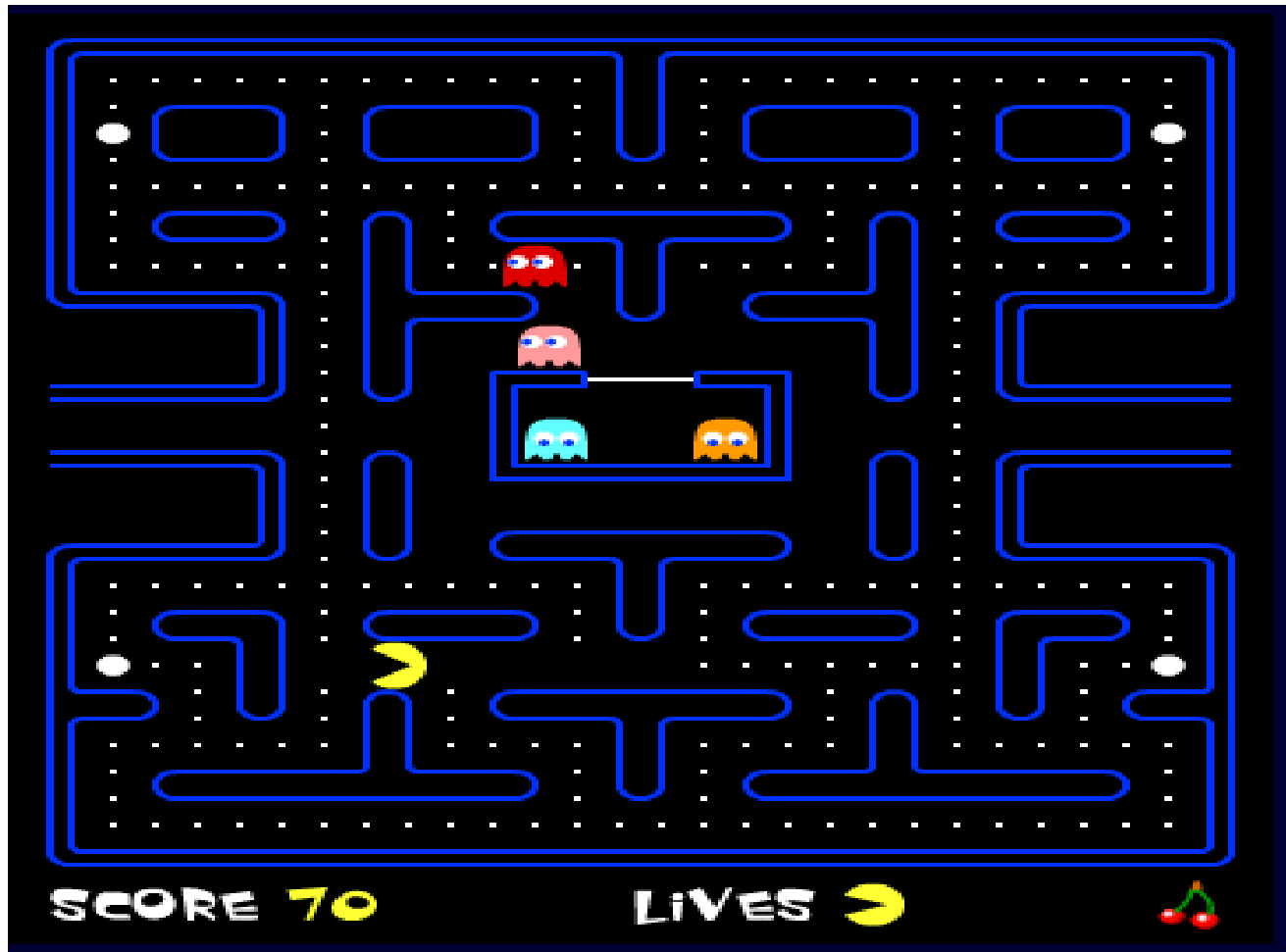


Programación lógica inductiva

```
; logical program for quicksort
qsort([X|Xs], Ys) :-
    partition(Xs, X, S1, S2),
    qsort(S1, S1s),
    qsort(S2, S2s),
    append(S1s, [X|S2s], Ys).
qsort([], []).
```



Ejemplo 2: Pac-Man





Ejemplo: Aprender Estrategia para Pacman

- Primitivas:
 - si-obstaculo, si-punto, si-punto-gordo, si-fantasma, (son del tipo if-then-else(A1,A2))
 - avanzar, girar-izquierda, girar-derecha
- **QUÉ:** comer todos los puntos del tablero



Ejemplo de programa en el Pacman con Programación Genética

(si-fantasma

(secuencia3 (girar-izquierda)

(girar-izquierda)

(avanzar))

(si-punto-gordo

(avanzar)

(girar-derecha)))



Ejemplo de programa en el Pacman con aprendizaje por refuerzo

- Aprender función estrategia:

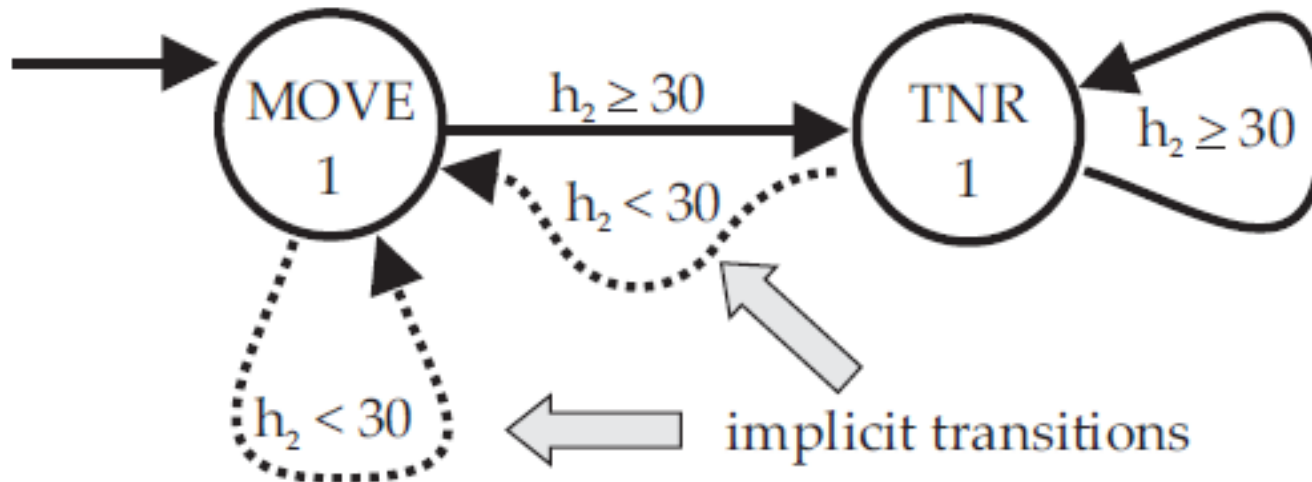
- $\Pi(\text{estado}) \rightarrow \text{accion}$

$\pi(\text{obstaculo, punto, puntogordo, fantasma, memoria-estados-anteriores})$

->

{avanzar, girar-izquierda, girar-derecha}

Ejemplo 3. Automata de estado finito (FSA)

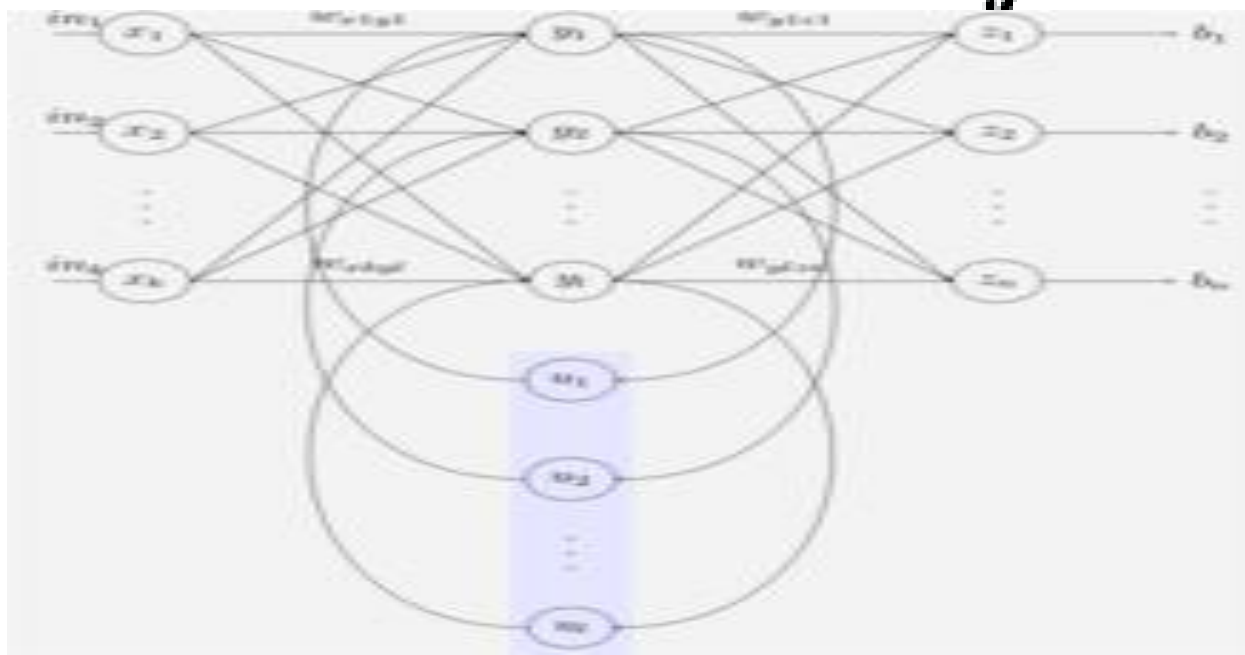


Se mueve hacia adelante mientras no haya un obstáculo delante ($h_2 \geq 30$) en cuyo caso, gira hacia la izquierda.

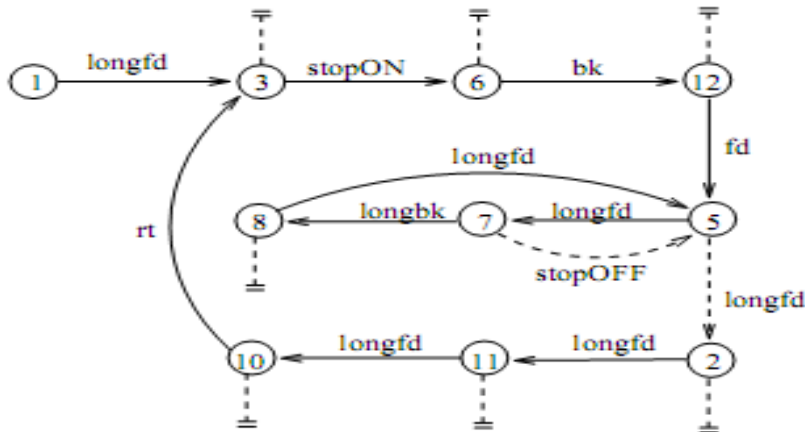
Ejemplo 4 Reconocimiento de escritura

never die. You with they did.

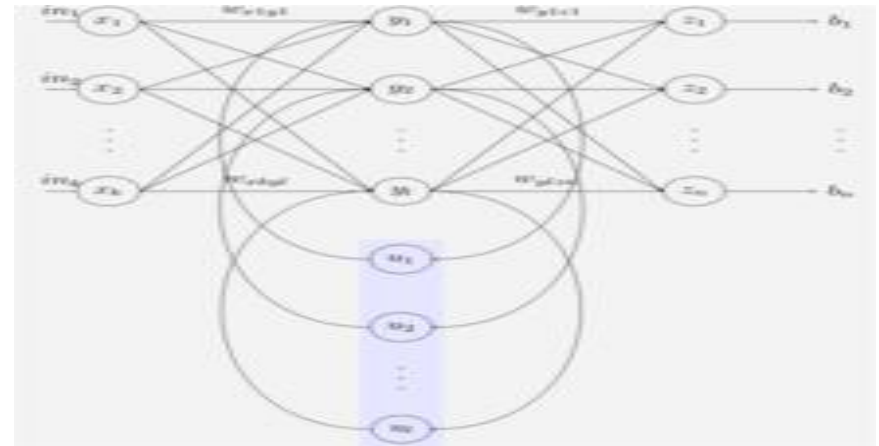
never die. You with they did.



Sistemas de varios pasos (incluyen el tiempo)



Finite State Automata



Recurrent Neural Networks

Reinforcement Learning: $\Pi(\text{estado}) \rightarrow \text{accion}$



Objetivos de la asignatura

- Ir más allá del aprendizaje automático proposicional (árboles de decisión, redes de neuronas, máquinas de vectores de soporte) o que incluyan la idea de tiempo / varios pasos, mediante:
 - Políticas / Aprendizaje por refuerzo
 - Representaciones relacionales / Programación Lógica Inductiva (ILP)
 - Aprendizaje de programas con lenguajes Turing-completos (bucles, ...)