Aprendizaje por Refuerzo y Redes neuronales

Instituto de Computación
Facultad de Ingeniería
Universidad de la República

mina
network management | artificial intelligence

Contenido

- Aprendizaje por refuerzo.
- Redes neuronales y aprendizaje por refuerzo.

Aprendizaje por Refuerzo

- Introducción
- La tarea a aprender
- Q Learning
- Ejemplo ilustrativo
- Convergencia
- Estratergias de exploración
- Acciones y Recompensas NO Deterministas
- Generalización

Aprendizaje por Refuerzo (1/2)

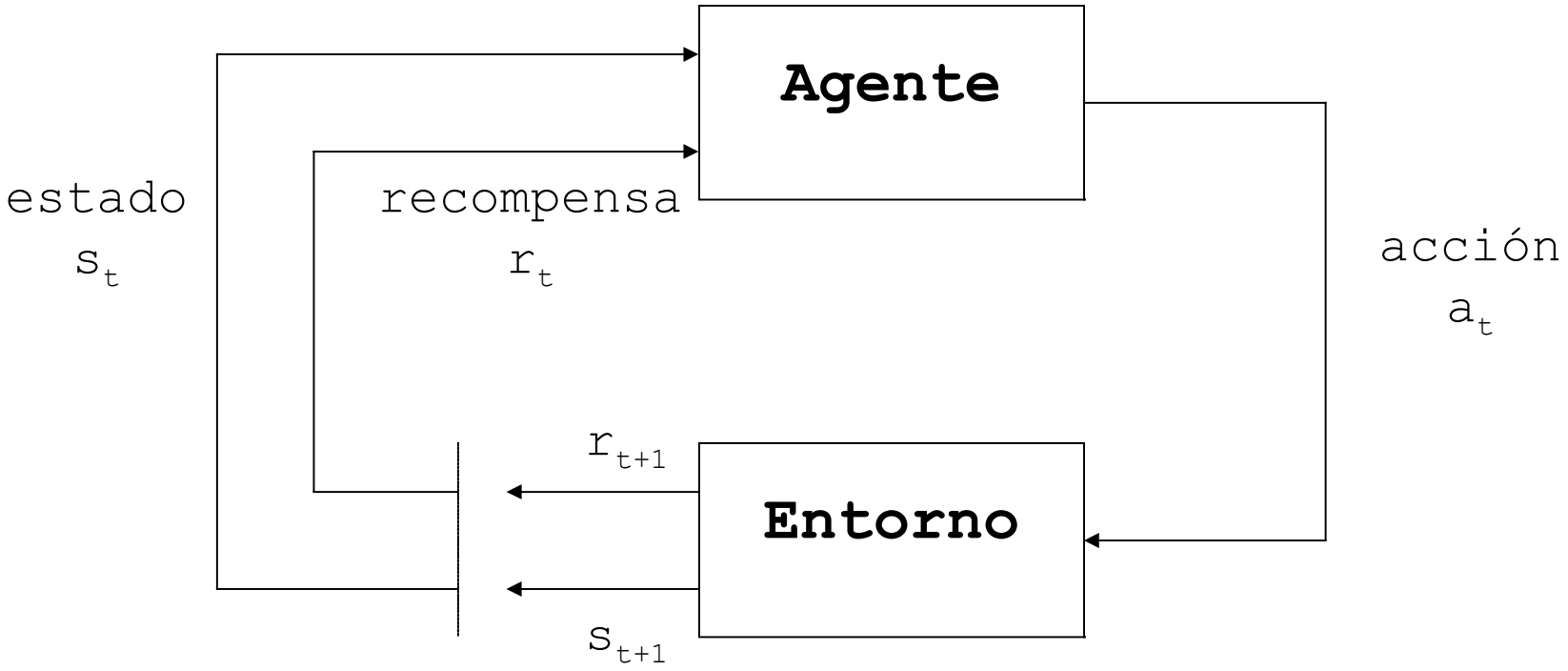
- ¿Cómo un agente autónomo que percibe y actúa en un entorno dinámico puede aprender a elegir acciones óptimas de modo de alcanzar sus objetivos?
- La idea detrás del aprendizaje por refuerzo es tan simple como atractiva: aprender del ensayo y el error mediante la interacción con el entorno.

Aprendizaje por Refuerzo (2/2)

- Cada vez que el agente realiza una acción en el entorno, o el entrenador, proporciona una recompensa o una penalización.
- La tarea del agente es aprender indirectamente, a partir de los refuerzos retardados, a elegir la secuencia de acciones que producen mayor acumulación de refuerzo.

Introducción

- Queremos que un agente aprenda a comportarse.



Principales características

- Ensayo y error.
- Recompensas y penalizaciones.
- Dilema exploración – explotación.
- Qué y no cómo.

La tarea a aprender (1/2)

- Proceso de Decisión Markoviano (MDP)
  - S: conjunto de estados.
  - A: conjunto de acciones.
  - r: función de refuerzo.
  - δ: función de transición. El siguiente estado depende del estado actual y de la acción elegida

La tarea a aprender (2/2)

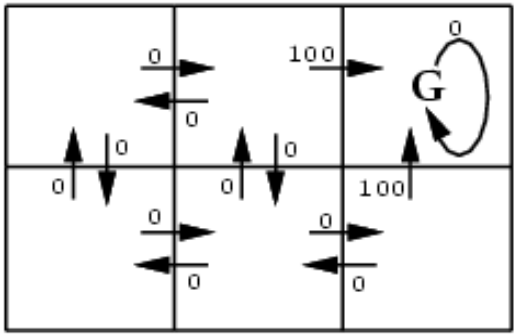
- La tarea
  - Política π: S->A
  - Valor acumulado

$$V^{\pi}(s_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$$

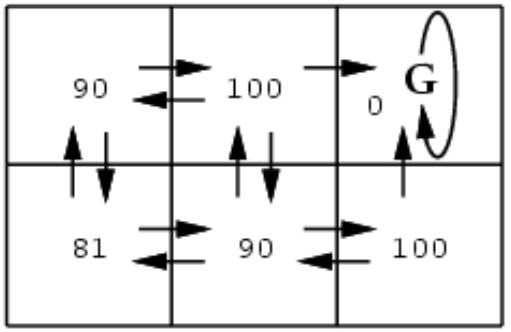
  - Política óptima

$$\pi^* = \arg\max_{\pi} V^{\pi}(s), \forall s$$

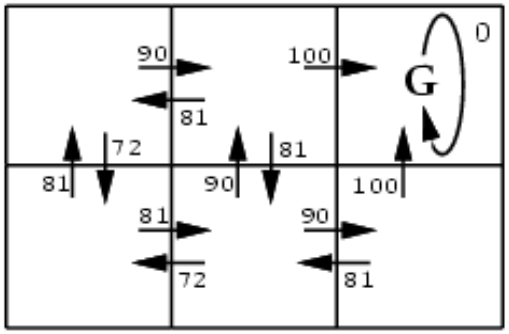
Ejemplo – Mundo grilla



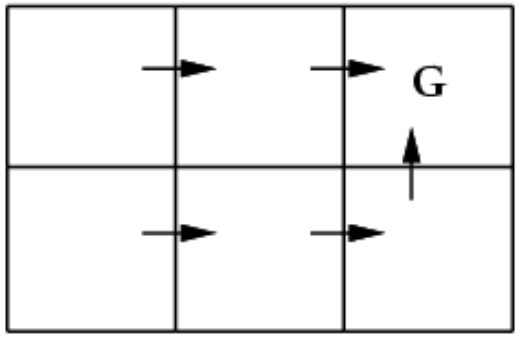
$r(s,a)$ (immediate reward) values



$V^*(s)$ values



$Q(s,a)$ values



One optimal policy

mina
network management | artificial intelligence

Q Learning

- ¿Cómo puede aprender el agente la política óptima en un entorno arbitrario?

$$\pi^*(s) = \arg\max_a \left[r(s,a) + \gamma V^*(\delta(s,a)) \right]$$

- Por lo cual el agente obtiene la política óptima aprendiendo V*, proporcionando conocimiento perfecto de refuerzos inmediatos y la función de transición de estados.

La función Q

- Definamos la función evaluación Q(s,a)

$$Q(s,a) = r(s,a) + \gamma V^*(\delta(s,a))$$

- Observar que Q es el valor que se desea maximizar en la ecuación de la política óptima

$$\pi^*(s) = \arg\max_a [Q(s,a)]$$

Un algoritmo para Q (1/2)

- Notemos la estrecha relación entre Q y V

$$V^*(s) = \max_{a'} Q(s, a')$$

- Permite rescribir la ecuación de la siguiente manera

$$Q(s, a) = r(s, a) + \gamma \max_{a'} Q(\delta(s, a), a')$$

- Regla de actualización del estimador para Q

$$\hat{Q}(s, a) = r + \gamma \max_{a'} \hat{Q}(s', a')$$

Un algoritmo para Q (2/2)

For each s, a initialize table entry $\hat{Q}(s, a) \leftarrow 0$

Observe current state s

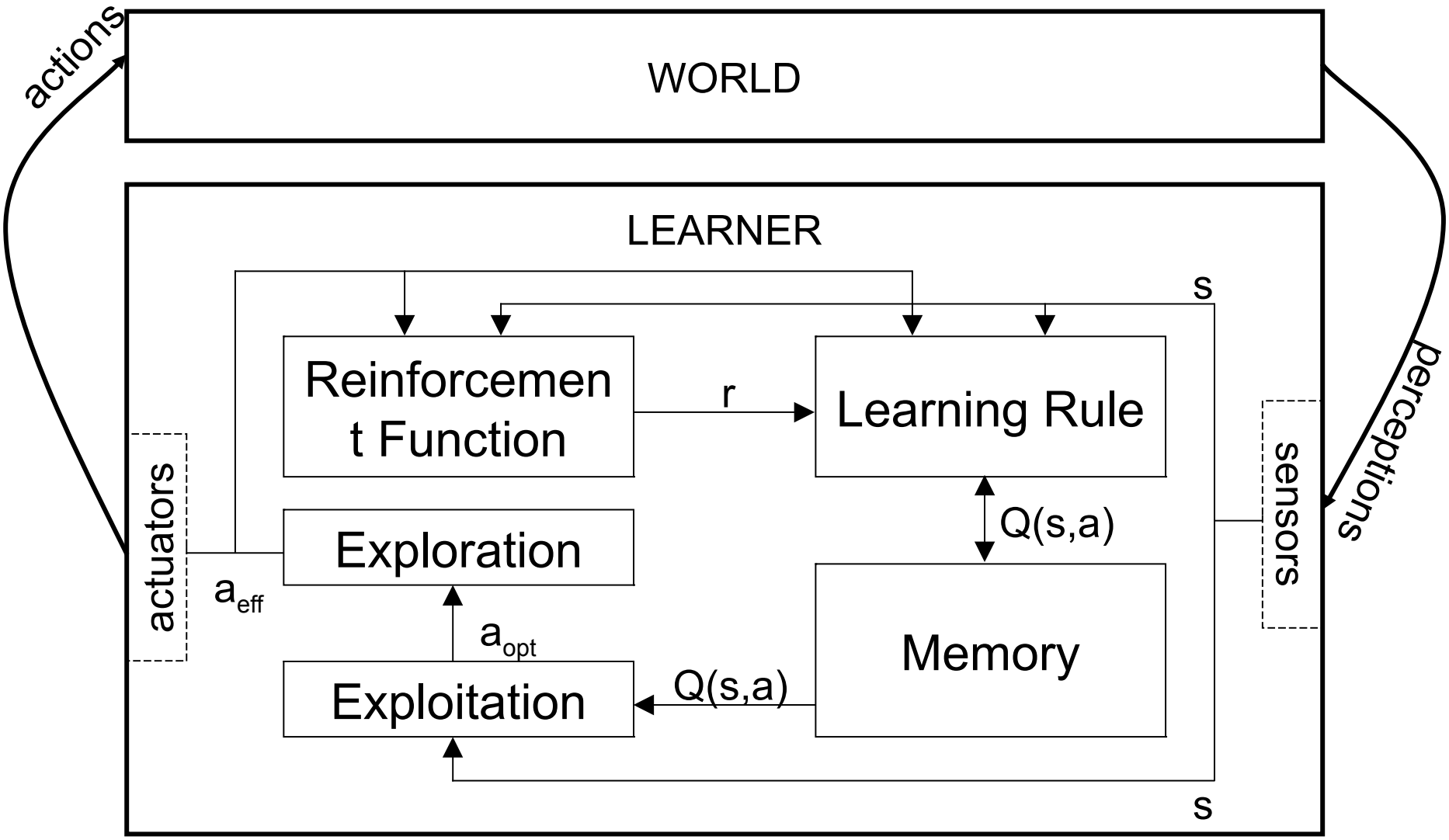
Do forever:

- Select an action a and execute it
- Receive immediate reward r
- Observe the new state s'
- Update the table entry for $\hat{Q}(s, a)$ as follows:

$$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$

- $s \leftarrow s'$

Agente RL

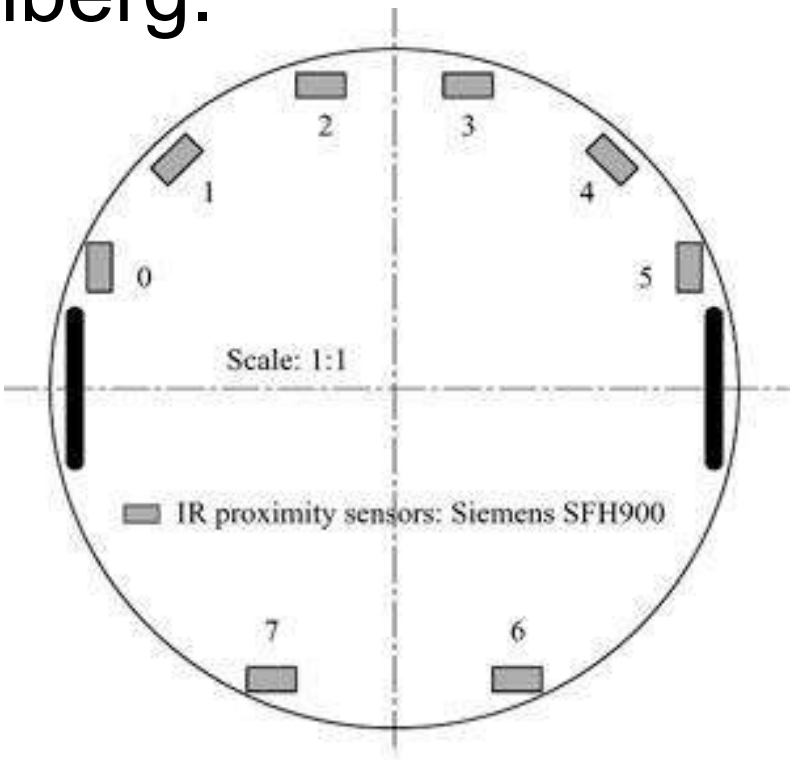


mina
network management | artificial intelligence

Un ejemplo ilustrativo
Evitar Obstáculos

- Método exhaustivo.
- Algoritmo de Braitenberg.
- Q-Learning

$$r(s^t, s^{t-1}) = \begin{cases} +1 \text{ si está evitando} \\ -1 \text{ si ocurrió colisión} \\ 0 \text{ en otro caso} \end{cases}$$



Convergencia

- Se ha demostrado la convergencia del estimador de Q bajo ciertas circunstancias al utilizar el algoritmo desarrollado por Watkins en 1989.
- Condiciones
  1. Es sistema se puede modelar como un MDP determinista.
  2. Los refuerzos inmediatos están acotados.
  3. El agente selecciona acciones de modo de visitar infinitamente a menudo los pares estado-acción. Esta es una restricción muy fuerte para dominios grandes (o infinitos).

Estrategias de exploración

- El algoritmo propuesto no especifica de que forma se eligen las acciones.
- Ejemplos
  - Codicioso y Codicioso-ε
  - Otra opción que asigna a toda acción una probabilidad de ser elegida

$$P(a_i \mid s) = \frac{k^{Q(s,a_i)}}{\sum_j k^{Q(s,a_j)}}$$

Acciones y Recompensas NO Deterministas (1/2)

- En este caso las funciones $\delta(s,a)$ y $r(s,a)$ pueden verse como si primero producen una distribución de probabilidad sobre las salidas.
- El entorno debe ser estacionario!
- La generalización es redefinir el valor de la política p como

$$V^{\pi}(s_t) = E\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i}\right]$$

Acciones y Recompensas NO Deterministas (2/2)

- Generalizamos la definición de Q

$$\begin{aligned} Q(s,a) &= E\left[r(s,a) + \gamma V^{\pi}(\delta(s,a))\right] \\ &= E\left[r(s,a)\right] + E\left[\gamma V^{\pi}(\delta(s,a))\right] \\ &= E\left[r(s,a)\right] + \gamma \sum_{s'} P(s'|s,a) V^{\pi}(s') \end{aligned}$$

- Al igual que antes

$$Q(s,a) = E\left[r(s,a)\right] + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a')$$

$$\hat{Q}_n(s,a) = (1 - \alpha_n)\hat{Q}_n(s,a) + \alpha_n(r + \gamma \max_{a'} \hat{Q}(s',a'))$$

Generalización

- Una de las restricciones más fuertes en procedimiento de Q Learning es que la función objetivo es representada en una tabla.
- Las hipótesis de convergencia requieren que todos los pares sean visitados infinitamente a menudo.
- Podemos tratar de estimar el valor Q de pares estado-acción no visitados generalizando a partir de los pares visitados.
- Clusterización.
- Redes neuronales.

Aprendizaje por Refuerzo y Redes Neuronales

- Motivación
- Uso de RN en RL
- Ejemplos

Motivación

- El objetivo del aprendizaje por refuerzo en robótica es sintetizar comportamientos que maximicen las recompensa en el tiempo.
- Problemas
  - Memoria (sensores y acciones continuas)
  - Generalización
  - Exploración

Uso de RN en RL (1/2)

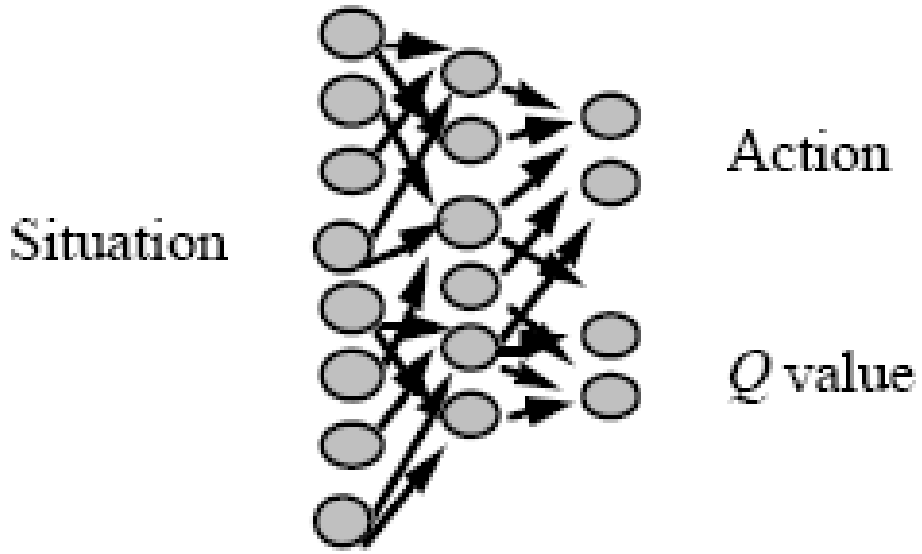
- Las redes neuronales se utilizan como funciones.
- Principalmente para aproximar Q.
- También se utilizan para atacar el inmenso espacio de búsqueda situación-acción.

Uso de RN en RL (2/2)

- Memoria para Q
  - Consultas realizadas sobre Q
    - valor Q(s,a)
    - arg $\max_a$ Q(s,a)
  - Actualización de Q (regla de aprendizaje)

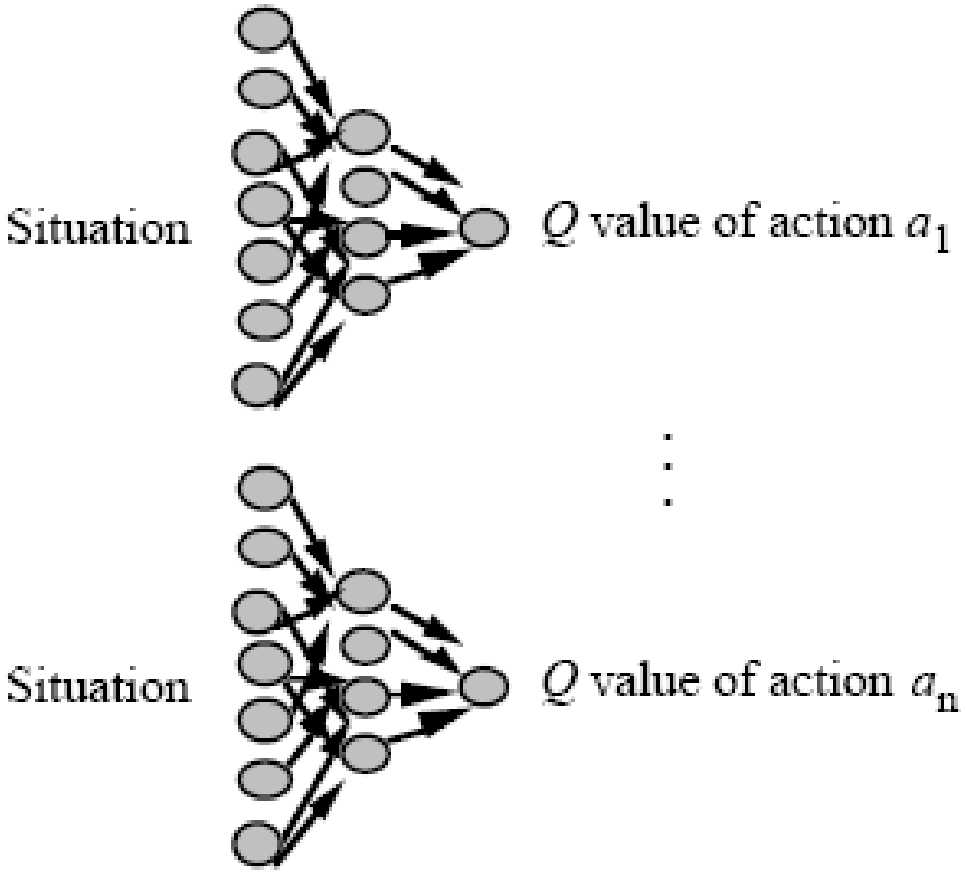
Red Ideal

- Ventaja
  - Dada las situación responde rápidamente con la acción a realizar.
  - Memoria
- Desventaja
  - Para una determinada situación existen varias acciones a tomar y sus respectivos valores Q.

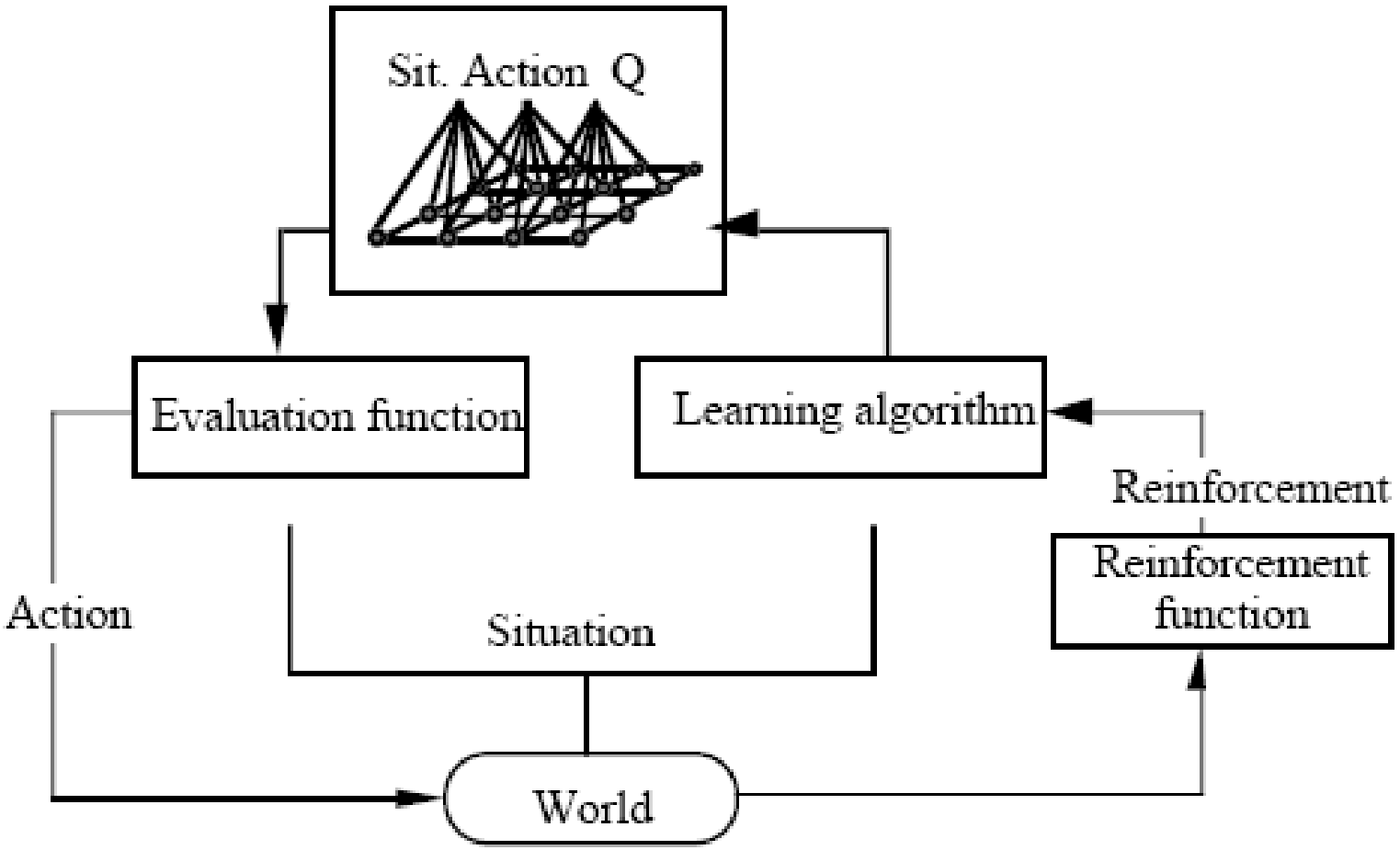


QCON

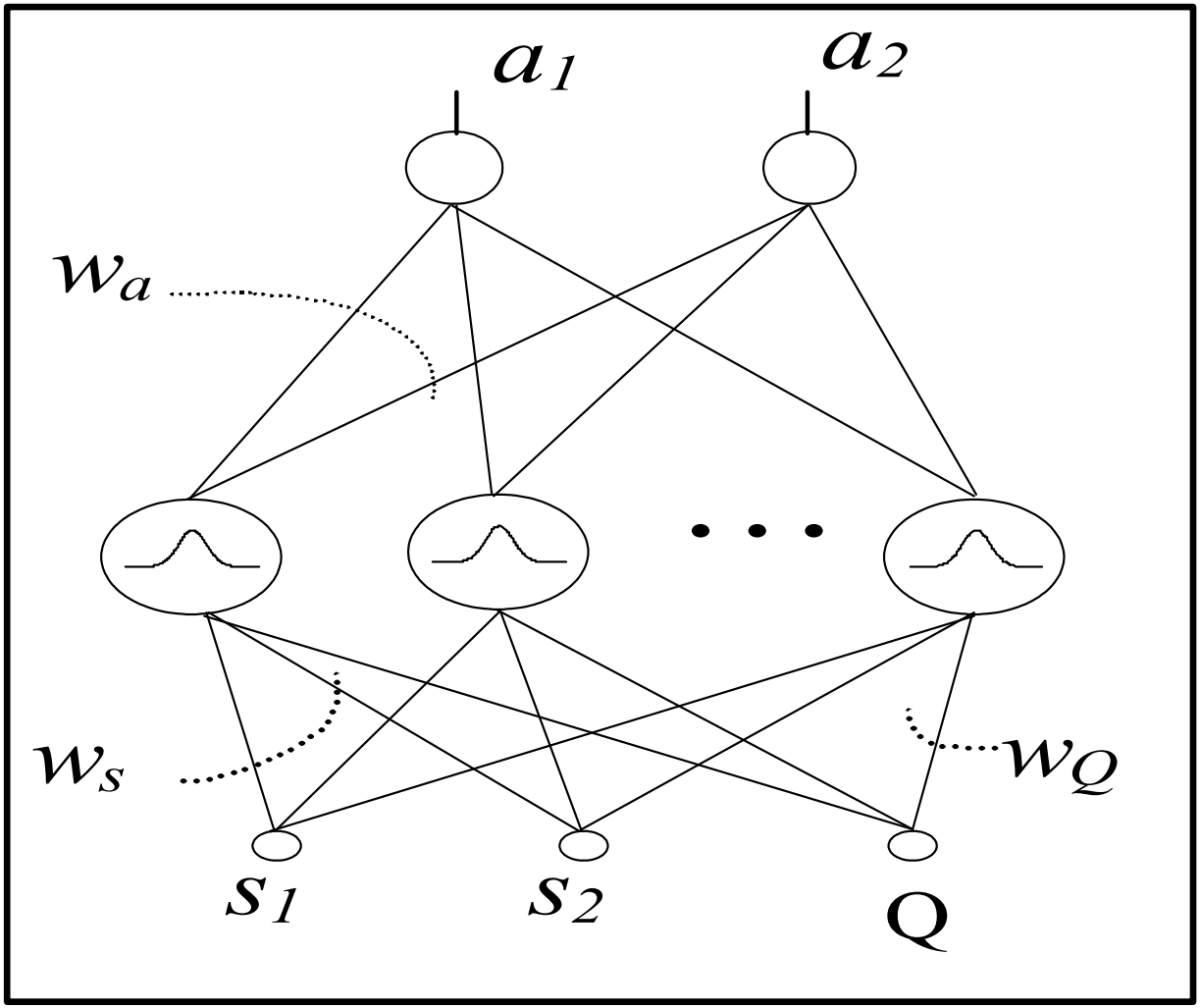
- Ventaja
  - Permite manejar varias acciones y valores Q para una misma situación
- Desventaja
  - No generaliza entre acciones.
  - Acciones fijas
    - Cantidad
    - Sentido



Q-KOHON


Sit. Action Q
Evaluation function
Learning algorithm
Reinforcement
Reinforcement function
Action
Situation
World

Redes RBF



Algoritmo Q

For each s, a initialize table entry $\hat{Q}(s, a) \leftarrow 0$

Observe current state s

Do forever:

- Select an action a and execute it
- Receive immediate reward r
- Observe the new state s'
- Update the table entry for $\hat{Q}(s, a)$ as follows:

$$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$

- $s \leftarrow s'$

Usando la red como memoria asociativa

- *Entrada: situación (s) y Q valor (q)*

- Paso 1. elegir el valor *i** según:

$$i^* = \arg\max_i \left[y_i = e^{-\left[(s - w_s)^t (s - w_s) + \left| (q - w_Q)/2 \right|^2 \right] / \sigma^2} \right]$$

- Paso 2.

si $y_i^* > acceptance\_threshold$

$$action = w_a(i^*)$$

else

agregar una neurona oculta (s,0,random)

mina
network management | artificial intelligence

Algoritmo Q

For each s, a initialize table entry $\hat{Q}(s,a) \leftarrow 0$

Observe current state s

Do forever:

- Select an action a and execute it
- Receive immediate reward r
- Observe the new state s'
- Update the table entry for $\hat{Q}(s,a)$ as follows:

$$\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a')$$

- $s \leftarrow s'$

mina
network management | artificial intelligence

Actualización de los pesos

- Entrada: situación (s), acción (a), máximo Q para la nueva situación (q) y el refuerzo (*r)*
- Paso 1. elegir el valor *i** según:

$$i^* = \arg\max_i \left[e^{-\left((s-w_s)^t(s-w_s)+|a-w_a|^2\right)/\sigma^2} \right]$$

- Paso 2.

$$if\ e^{-\left((s-w_s)^t(s-w_s)+|a-w_a|^2\right)/\sigma^2} > \text{acceptance\_threshold}$$

$$w_Q(i^*) = w_Q(i^*) + \eta_Q\left(r + \gamma q - w_Q(i^*)\right)$$

$$w_s(i^*) = w_s(i^*) + \eta_s\left(s - w_s(i^*)\right)$$

if r > 0

$$w_a(i^*) = w_a(i^*) + \eta_a\left(a - w_a(i^*)\right)$$

if r < 0

$$w_a(i^*) = w_a(i^*) + \eta_a\left(1 - a - w_a(i^*)\right)$$

else

agregar neuron oculta **(s,r,a)**

Algoritmo Q

For each s, a initialize table entry $\hat{Q}(s, a) \leftarrow 0$

Observe current state s

Do forever:

- Select an action a and execute it
- Receive immediate reward r
- Observe the new state s'
- Update the table entry for $\hat{Q}(s, a)$ as follows:

$$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$

- $s \leftarrow s'$

Obtener Q-max de la red

- *Entrada: nueva situación (s)*

- Paso 1. elegir el valor *i** según:

$$i^* = \arg\max_i \left[y_i = e^{-\left[(s - w_s)^t (s - w_s) + \left| (1 - w_Q)/2 \right|^2 \right] / \sigma^2} \right]$$

- Paso 2. retornar como Q-max el valor de

$$w_Q(i^*)$$

Algunas recetas

- Valores usuales
  - η_a=0.2
  - η_s=0.01
  - η_q=0.5
  - σ=0.4
  - *acceptance_thereshold*=0.1
  - Neuronas ocultas ~150

Referencias

- Redes neuronales
  - Dayhoff J., Neural Network Architectures: An Introduction, Van Nostrand Reinhold, 1990.
  - Hertz J.A., A. Krogh y R.G. Palmer, Introduction to the Theory of Neural Computation, Addison Wesley,1991.
  - Haykin S., Neural Network: A Comprehensive Foundation, Macmillan, 1994.
- Aprendizaje por refuerzo
  - Russell S. y P. Norvig, Inteligencia Artificial: un enfoque moderno, Prentice Hallm, Second Edition, 2002.
  - Mitchell T., Machine Learning, McGraw Hill, 1997.
- Artículos
  - Santos J. M. y C. Touzet , "Dynamic Update of the Reinforcement Function during Learning" , Connection Science Journal. Special Issue on Adaptive Robotics, Volume 11, Number 3-4, page 267--290 – 1999.
  - Kaelbling L. P., M. L. Littman y A. W. Moore. Reinforcement learning: A survey. Journal of Arti cial Intelligence Research, 4:237-285, 1996.

Preguntas