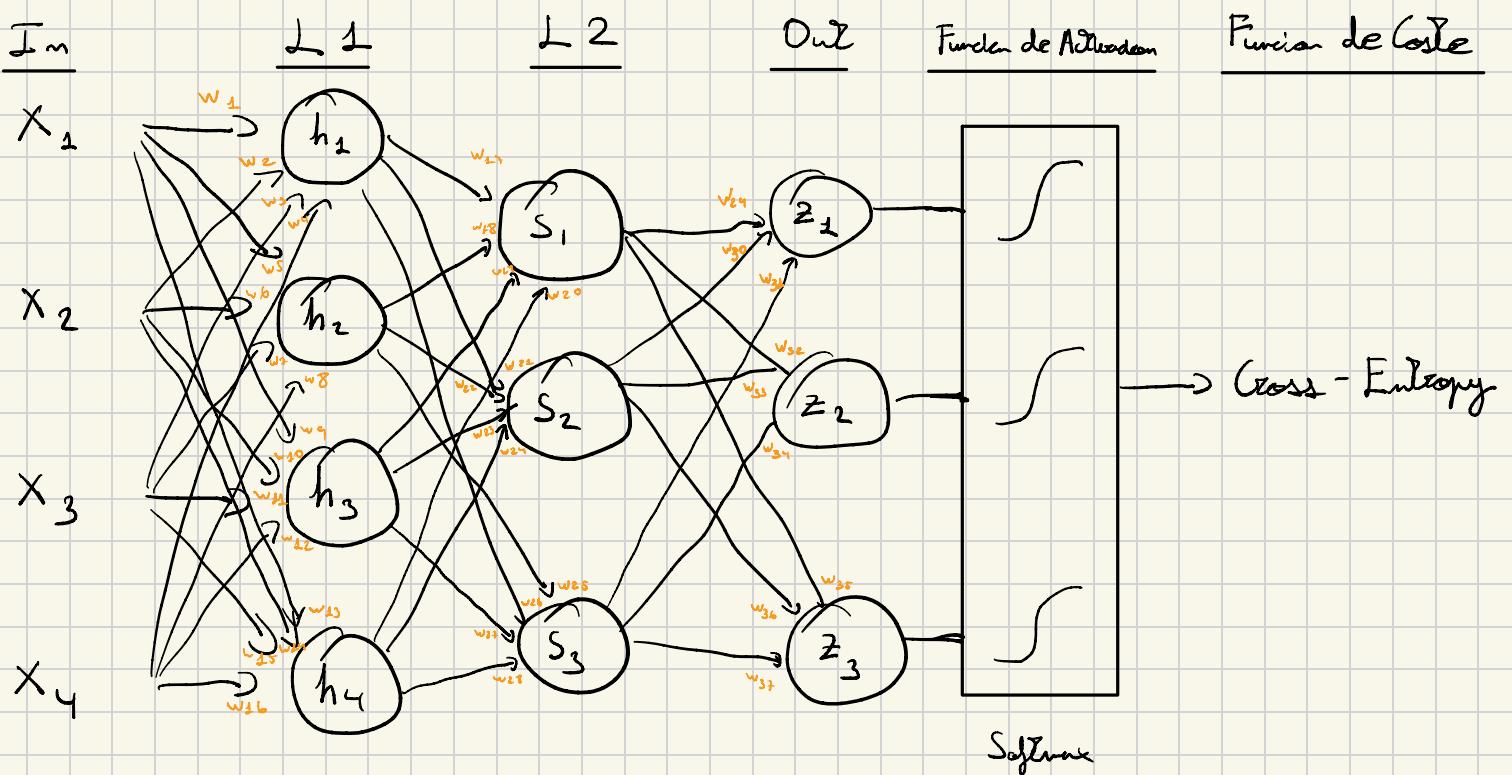


### 1.1 Dibujar la red (1 punto)

Dados items de entrada que vienen representados por 4 rasgos (features) es decir, que cada ítem del entrenamiento viene representado por un vector de dimensión [1x4] dibuja una red neuronal de 2 capas intermedias, donde la primera capa consta de 4 neuronas y la segunda de 3, sabiendo que esta red se empleará para realizar clasificación multiclase consistente en 3 clases. Decide tu la dimensión de la última capa. Inicializa los pesos con valores al azar, pero dibuja dichos valores también en el dibujo, por ejemplo  $w_1=0.2$ ,  $w_2=0.15$ , etc.



$$\begin{aligned} w_1 &= 0,2 \\ w_2 &= 0,15 \\ w_3 &= 0,3 \\ w_4 &= 0,4 \\ w_5 &= 0,22 \\ w_6 &= 0,24 \\ w_7 &= 0,3 \\ w_8 &= 0,1 \\ w_9 &= 0,4 \\ w_{10} &= 0,6 \end{aligned}$$

$$\begin{aligned} w_{11} &= 0,33 \\ w_{12} &= 0,12 \\ w_{13} &= 0,16 \\ w_{14} &= 0,24 \\ w_{15} &= 0,31 \\ w_{16} &= 0,28 \\ w_{17} &= 0,64 \\ w_{18} &= 0,6 \\ w_{19} &= 0,71 \\ w_{20} &= 0,33 \end{aligned}$$

$$\begin{aligned} w_{21} &= 0,8 \\ w_{22} &= 0,1 \\ w_{23} &= 0,33 \\ w_{24} &= 0,15 \\ w_{25} &= 0,25 \\ w_{26} &= 0,33 \\ w_{27} &= 0,45 \\ w_{28} &= 0,11 \\ w_{29} &= 0,07 \\ w_{30} &= 0,15 \end{aligned}$$

$$\begin{aligned} w_{31} &= 0,21 \\ w_{32} &= 0,36 \\ w_{33} &= 0,44 \\ w_{34} &= 0,22 \\ w_{35} &= 0,11 \\ w_{36} &= 0,23 \\ w_{37} &= 0,17 \end{aligned}$$

Softmax

## 1.2 Calcular el error (3 puntos)

Una vez hecho el dibujo, calcula las salida y el error empleando cross-entropy sabiendo que la clase real es la 2, es decir el vector  $y_{\text{real}}$  será 0,1,0. Presenta las operaciones (no hace falta que las resuelvas) para actualizar un peso de la segunda capa. Consulta la hoja de fórmulas.

$$\begin{aligned} w_1 &= 0,2 \\ w_2 &= 0,15 \\ w_3 &= 0,3 \\ w_4 &= 0,4 \\ w_5 &= 0,22 \\ w_6 &= 0,24 \\ w_7 &= 0,3 \\ w_8 &= 0,2 \\ w_9 &= 0,4 \\ w_{10} &= 0,6 \end{aligned}$$

$$\begin{aligned} w_{11} &= 0,33 \\ w_{12} &= 0,12 \\ w_{13} &= 0,16 \\ w_{14} &= 0,24 \\ w_{15} &= 0,31 \\ w_{16} &= 0,28 \\ w_{17} &= 0,64 \\ w_{18} &= 0,6 \\ w_{19} &= 0,71 \\ w_{20} &= 0,33 \end{aligned}$$

$$\begin{aligned} w_{21} &= 0,8 \\ w_{22} &= 0,1 \\ w_{23} &= 0,33 \\ w_{24} &= 0,15 \\ w_{25} &= 0,25 \\ w_{26} &= 0,33 \\ w_{27} &= 0,15 \\ w_{28} &= 0,11 \\ w_{29} &= 0,07 \\ w_{30} &= 0,15 \end{aligned}$$

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 2 \\ x_3 &= 3 \\ x_4 &= 4 \end{aligned}$$

$$b_x = b_x \cdot w_{bx} = 1$$

$$\begin{aligned} L_1: h_1 &= x_1 w_1 + x_2 w_2 + x_3 w_3 + x_4 w_4 + b_1 \rightarrow h_1 = 4 \\ h_2 &= x_1 w_5 + x_2 w_6 + x_3 w_7 + x_4 w_8 + b_2 \rightarrow h_2 = 3 \\ h_3 &= x_1 w_9 + x_2 w_{10} + x_3 w_{11} + x_4 w_{12} + b_3 \rightarrow h_3 = 4,07 \\ h_4 &= x_1 w_{13} + x_2 w_{14} + x_3 w_{15} + x_4 w_{16} + b_4 \rightarrow h_4 = 3,69 \end{aligned}$$

$$\begin{aligned} L_2: S_1 &= h_1 w_{17} + h_2 w_{18} + h_3 w_{19} + h_4 w_{20} + b_5 \rightarrow S_1 = 8,95 \\ S_2 &= h_1 w_{21} + h_2 w_{22} + h_3 w_{23} + h_4 w_{24} + b_6 \rightarrow S_2 = 6,4 \\ S_3 &= h_1 w_{25} + h_2 w_{26} + h_3 w_{27} + h_4 w_{28} + b_7 \rightarrow S_3 = 5,29 \end{aligned}$$

$$\begin{aligned} \text{Out: } Z_1 &= S_1 w_{29} + S_2 w_{30} + S_3 w_{31} + b_8 \rightarrow Z_1 = 3,86 \\ Z_2 &= S_1 w_{32} + S_2 w_{33} + S_3 w_{34} + b_9 \rightarrow Z_2 = 8,2 \\ Z_3 &= S_1 w_{35} + S_2 w_{36} + S_3 w_{37} + b_{10} \rightarrow Z_3 = 3,72 \end{aligned}$$

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^m e^{z_j}}$$

$$\left. \begin{array}{l} \text{softmax}(Z_1) = 0,013 \\ \text{softmax}(Z_2) = 0,976 \\ \text{softmax}(Z_3) = 0,011 \end{array} \right\}$$

Increíble que me haya dado el resultado correcto

$$\begin{aligned} \text{Cross-Entropy} &= -\frac{1}{n} \sum_{i=1}^n y_i \ln(y_i) \rightarrow -\frac{1}{3} (0 \cdot \ln(0,013) + 1 \cdot \ln(0,976) + 0 \cdot \ln(0,011)) \\ &\quad -\frac{1}{3} \ln(0,976) = \underline{0,008} \end{aligned}$$

Actualizar  $w_{17}$

$$w_{17} = w_{17} - \alpha \frac{dFC}{dw_{17}}$$

$$\frac{dFC}{dw_{29}} = \frac{dFC}{dFA} \cdot \left[ \left( \frac{dFA}{dz_1} \cdot \frac{dz_1}{ds_1} \cdot \frac{ds_1}{dw_{17}} \right) + \left( \frac{dFA}{dz_2} \cdot \frac{dz_2}{ds_1} \cdot \frac{ds_1}{dw_{17}} \right) + \left( \frac{dFA}{dz_3} \cdot \frac{dz_3}{ds_1} \cdot \frac{ds_1}{dw_{17}} \right) \right]$$

## 2. Value Iteration (3 puntos)

Dada la fórmula de Bellman adaptada al Value Iteration:

$$V_{k+1} \leftarrow \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V_k(s')]$$

- Conjunto de estados:  $S = \{S_0, S_1, S_2\}$
- Conjunto de acciones:  $A = \{A_0, A_1\}$
- Función de transición de estados:  $T: S \times A \times S \rightarrow P(S)$  donde

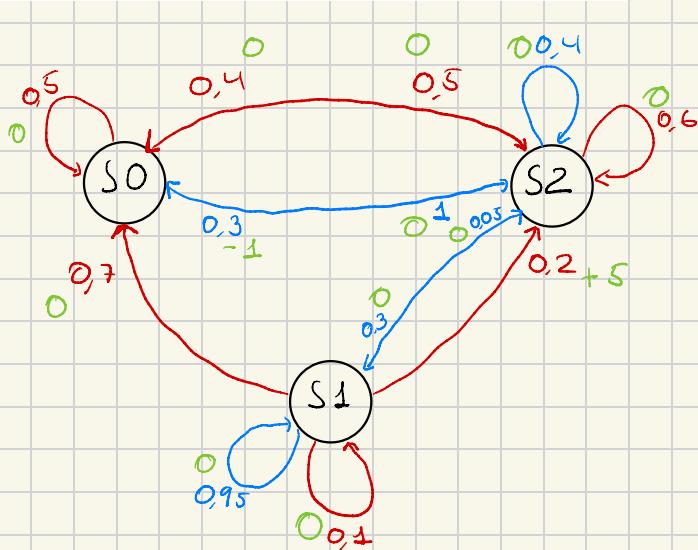
- $T(S_0, A_0, S_0) = 0,5$
- $T(S_0, A_0, S_1) = 0$
- $T(S_0, A_0, S_2) = 0,5$
- $T(S_0, A_1, S_0) = 0$
- $T(S_0, A_1, S_1) = 0$
- $T(S_0, A_1, S_2) = 1$
- $T(S_1, A_0, S_0) = 0,7$
- $T(S_1, A_0, S_1) = 0,1$
- $T(S_1, A_0, S_2) = 0,2$
- $T(S_1, A_1, S_0) = 0$
- $T(S_1, A_1, S_1) = 0,95$
- $T(S_1, A_1, S_2) = 0,05$

- $T(S_2, A_0, S_0) = 0,4$
- $T(S_2, A_0, S_1) = 0$
- $T(S_2, A_0, S_2) = 0,6$
- $T(S_2, A_1, S_0) = 0,3$
- $T(S_2, A_1, S_1) = 0,3$
- $T(S_2, A_1, S_2) = 0,4$

$R: S \times A \times S \rightarrow R$  donde  
 +5 si  $(s, a, s') = (S_1, A_0, S_2)$   
 -1 si  $(s, a, s') = (S_2, A_1, S_0)$   
 0 en otro caso

Dibuja gráficamente el MDP correspondiente (1 punto).

Calcula el valor del estado  $S_1$  empleando el algoritmo de Value Iteration o Iteración por valor, en su versión síncrona, suponiendo que todos los estados se inicialicen a valor 0.0 (2 puntos)



$V_0$	$S_0$	$S_1$	$S_2$
0	0	0	0

$$V_1(S_1) = \max \{ 0,7[0 + \gamma \cdot 0], 0,1[0 + \gamma \cdot 0], 0,2[5 + \gamma \cdot 0], 0,095[0 + \gamma \cdot 0], 0,05[0 + \gamma \cdot 0] \}$$

$$V_1(S_1) = \max \{ 0, 0, 1, 0, 0, 0 \}$$

$$V_1(S_1) = 1$$

## 2. Value Iteration (2 puntos)

Dada la fórmula de Bellman adaptada al Value Iteration que encontrarás en la hoja de fórmulas, considera el siguiente MDP. Los estados son cuadrados de cuadrícula, identificados por su número de fila y columna (primera fila). El agente siempre comienza en el estado (1,1), marcado con la letra S. Hay dos estados objetivo terminales, (2,3) con recompensa +5 y (1,3) con recompensa -5. Las recompensas son 0 en estados no terminales. La función de transición para las acciones Norte, Sur, Oeste o Este aparece representada en el dibujo, pero si ocurre una colisión con una de las paredes que delimitan las cuadrículas, el agente permanece en el mismo estado.

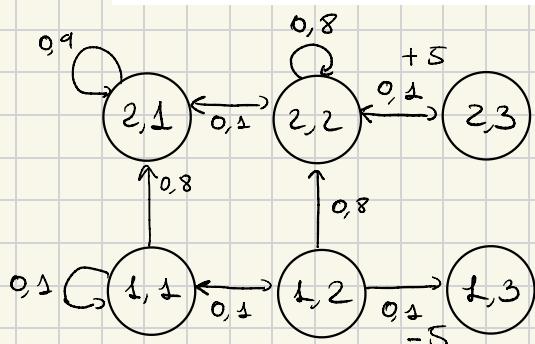


(a)

(b)

Dibuja gráficamente el MDP correspondiente (1 punto).

Calcula el valor de los estados dadas dos iteraciones empleando el algoritmo de Value Iteration o Iteración por valor, en su versión síncrona, suponiendo que todos los estados se inicialicen a valor 0.0 y que el valor de descuento gamma es de 0.9 (1 punto)



(No tengo ya duda si el  
MDP está bien)

(1,1) (1,2) (1,3) (2,1) (2,2) (2,3)

$V_0$  0 0 0 0 0 0

$V_1$  0 0 0 0 0.5 0

$V_2$  0 0.36 0 0.045 0.5 0.045

$$V_1(2,2) = \max \left[ 0.1(0 + 0.9 \cdot 0), 0.8(0 + 0.9 \cdot 0), 0.1(5 + 0.9 \cdot 0) \right]$$

$$V_2(2,2) = 0.5$$

$$V_2(2,3) = 0.1(0 + 0.9 \cdot 0.5) = 0.045$$

$$V_2(1,2) = 0.8(0 + 0.9 \cdot 0.5) = 0.36$$

$$V_2(1,1) = 0.045$$

### 3. Q-learning (3 puntos)

Dada la cuadrícula que se muestra a continuación y un agente que está tratando de aprender la política óptima, sabemos que las recompensas se consiguen solamente por realizar la acción **salir** desde uno de los estados coloreados. Al realizar esta acción, el agente pasa al estado **Final (D)** que aunque no aparece en la cuadrícula es un estado al que se puede transitar desde cualquiera de los estados coloreados aunque con distinto premio, al salir desde los estados verdes se obtiene un premio de +30 y desde los estados rojos un premio de -100. En cualquier caso, al salir el episodio terminaría. Supongamos  $\gamma = 1$  y  $\alpha = 0.5$  para todos los cálculos.



Sabiendo que la fórmula de Bellman adaptada para Q-learning es la siguiente:

$$\text{New } Q(s, a) = Q(s, a) + \alpha [R(s, a) + \gamma \max_{a'} Q'(s', a') - Q(s, a)]$$

New value for that state and action  
 Current Q value  
 Reward for taking that action at that state  
 Learning Rate  
 Discount rate  
 Maximum expected future reward given the new state and all possible actions at that new state

Dados los episodios que aparecen a continuación, completa el momento en el que los siguientes valores de Q se vuelven distintos de cero por primera vez. Tu respuesta debe tener el formato (# de episodio, # de iter) donde # es la iteración de actualización de Q-learning en ese episodio. Si el valor Q especificado nunca pasa a ser distinto de cero, escribe nunca.

5 de Episodio

6 de Episodio

$Q((1,2), E) = \underline{\text{Nunca}}$

$Q((2,2), E) = \underline{3 \text{ de Iteración}}$

$Q((3,2), S) = \underline{4 \text{ de Iteración}}$

#### Episodio 1      Episodio 2      Episodio 3      Episodio 4      Episodio 5

Episodio 1	Episodio 2	Episodio 3	Episodio 4	Episodio 5
(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0
(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0
(2,2), E, (3,2), 0	(2,2), S, (2,1), 0	(2,2), E, (3,2), 0	(2,2), E, (3,2), 0	(2,2), E, (3,2), 0
(3,2), N, (3,3), 0	(2,1), Exit, D, -100	(3,2), S, (3,1), 0	(3,2), N, (3,3), 0	(3,2), S, (3,1), 0
3,3), Exit, D, +50	(3,1), Exit, D, +30	(3,3), Exit, D, +50	(3,1), Exit, D, +30	(3,1), Exit, D, +30

#### Episodio 1

$$\text{New } Q((1,3), S) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((1,2), E) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((2,2), E) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((3,2), N) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((3,3), \underline{\text{Exit}}) = 0 + 0,5[50 + 1 \cdot 0 - 0] = 25$$

#### Episodio 2

$$\text{New } Q((1,3), S) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((1,2), E) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((2,2), S) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((2,1), \underline{\text{Exit}}) = 0 + 0,5[-100 + 1 \cdot 0 - 0] = -50$$

#### Episodio 3

$$\text{New } Q((1,3), S) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((1,2), E) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((2,2), E) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((3,2), S) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((3,1), \underline{\text{Exit}}) = 0 + 0,5[30 + 1 \cdot 0 - 0] = 15$$

#### Episodio 4

$$\text{New } Q((1,3), S) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((1,2), E) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((2,2), E) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((3,2), N) = 0 + 0,5[0 + 1 \cdot 25 - 0] = 12,5$$

$$\text{New } Q((3,1), \underline{\text{Exit}}) = 25 + 0,5[50 + 1 \cdot 0 - 25] = 37,5$$

#### Episodio 5

$$\text{New } Q((1,3), S) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((1,2), E) = 0 + 0,5[0 + 1 \cdot 0 - 0] = 0$$

$$\text{New } Q((2,2), E) = 0 + 0,5[0 + 1 \cdot 12,5 - 0] = 6,25$$

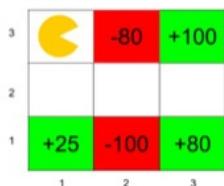
$$\text{New } Q((3,2), S) = 0 + 0,5[0 + 1 \cdot 15 - 0] = 7,5$$

$$\text{New } Q((3,1), \underline{\text{Exit}}) = 15 + 0,5[30 + 1 \cdot 0 - 15] = 22,5$$

No hace falta realmente los cálculos. Los hago por una suerte de masoquismo.

### 3. Q-learning (2,5 puntos)

Bajo el mundo de cuadrícula que se muestra a continuación, Pacman, está tratando de aprender la política óptima. Si una acción resulta en aterrizar en uno de los estados sombreados, se otorga la recompensa correspondiente durante esa transición. Todos los estados sombreados son estados terminales, es decir, el episodio termina una vez que llega a un estado sombreado. Los otros estados tienen como acciones posibles Norte, Este, Sur, y Oeste, que mueven de forma determinista a Pacman al estado vecino correspondiente (o mantienen al Pacman en su lugar si la acción que se intenta le hace salir de la cuadrícula). Suponiendo que el factor de descuento  $\gamma = 0.5$  y learning rate  $\alpha = 0.5$  para todos los cálculos y sabiendo que el Pacman comienza en el estado (1, 3).



Teniendo la fórmula de Bellman adaptada para Q-learning:

The diagram illustrates the components of the Q-learning update rule:

- New Q value for that state and that action**: Represented by a downward-pointing bracket under the term  $Q(s, a) + \alpha[R(s, a) + \gamma \max Q'(s', a') - Q(s, a)]$ .
- Current Q value**: Represented by a downward-pointing bracket under the term  $Q(s, a)$ .
- Reward for taking that action at that state**: Represented by a downward-pointing bracket under the term  $R(s, a)$ .
- Learning Rate**: A vertical line separating the new Q value from the current Q value.
- Discount Loss**: A vertical line separating the reward from the maximum expected future reward.
- Maximum expected future reward given the new s' and all possible actions at that new state**: Represented by a downward-pointing bracket under the term  $\gamma \max Q'(s', a')$ .

Dados los episodios que aparecen a continuación, asigna uno de los valores de la lista (0, 12.5, 50) a los valores Q solicitados. No será suficiente con que asigne el resultado, tendrás que razonarlo. Una forma puede ser calcular todos los valores Q tras los 3 episodios, esto debería de ser rápido porque la mayoría de ellos serán 0.

<b>Episodio 1</b>	<b>Episodio 2</b>	<b>Episodio 3</b>
(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0
(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0
(2,2), S, (2,1), -100	(2,2), E, (3,2), 0	(2,2), E, (3,2), 0
	(3,2), N, (3,3), +100	(3,2), S, (3,1), +80

$$Q((3,2), N) = \underline{\text{50}} \quad Q((1,2), S) = \underline{\text{0}} \quad Q((2,2), E) = \underline{\text{12, 6}}$$