

Técnicas de Inteligencia Artificial Examen evaluación continua - 16 de enero de 2023

1.- Aprendizaje Supervisado (1.5 puntos)

1.1- Responde a las siguientes preguntas (0.4 puntos):

Nota Importante: Razona tu respuesta, si el razonamiento es correcto se contabilizará el valor de la pregunta pero si la respondes sin razonamiento o con razonamiento incorrecto se te descontará la puntuación que esta pregunta valga.

1.- Teniendo un perceptrón simple, en cada actualización de los pesos, ¿la instancia que ha condicionado esa actualización sería bien etiquetada trás la actualización que acabamos de realizar?.

- Siempre Algunas Vezes con perceptrón y siempre con MIRA Nunca, solo sucederá si empleamos MIRA con el perceptrón no

2.- Dadas la red-1 y la red-2:

¿Hay alguna que represente un aprendizaje multiclas?.

- No, ninguna Una Ambas

3.- Dadas la red-1 y la red-2:

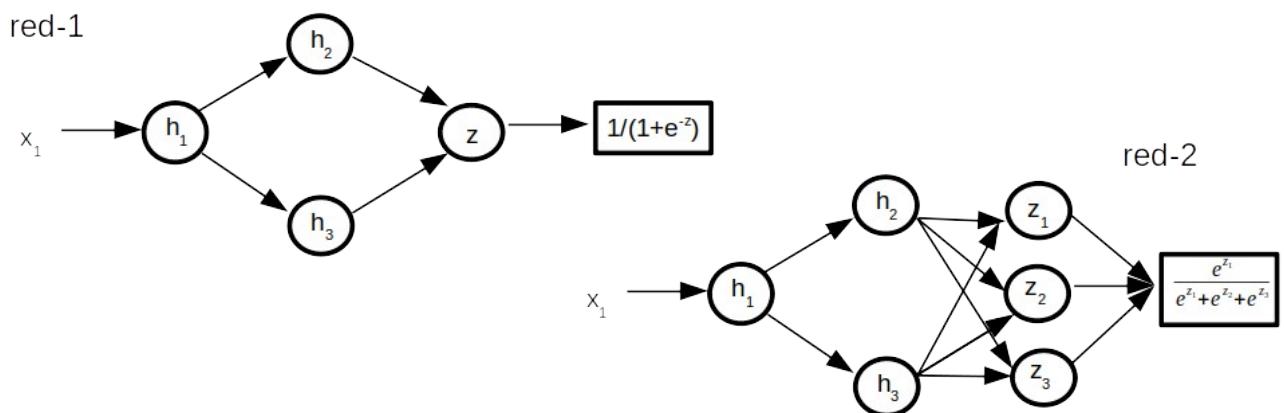
¿Para qué se emplean las ReLUs?.

- Para introducir una capa más y mejorar la predicción Para introducir no linearidad
 Para convertir una regresión en clasificación

4.- Dadas la red-1 y la red-2:

En la red-1 tras el sigmoide, ¿cuál será el punto de corte para determinar si la instancia pertenece o no a cada clase?. Quizás te sirva para razonar recordar que en el perceptrón el valor de corte es 0, intenta hacer un paralelismo.

- 1 0.5 0



1.2- Soluciona el siguiente problema (1.1 puntos)

Dada la siguiente red, sabiendo que los valores son los siguientes:

$$x_1 = 5 \rightarrow [0.1] \begin{pmatrix} 0.01 & 0.05 \end{pmatrix} \begin{pmatrix} 0.02 \\ 0.03 \end{pmatrix} \rightarrow 1/(1+e^{-z}) \rightarrow \text{MSE}$$

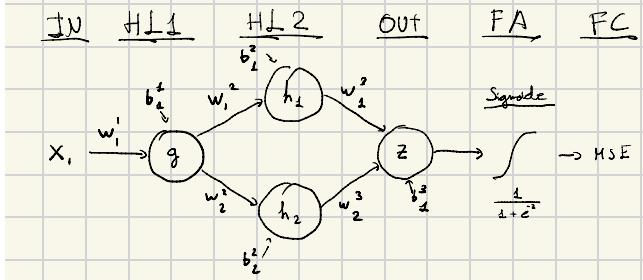
$$y_{real} = 1$$

La función de activación a es en este caso es el sigmoide, es decir, $\text{sigmoid}(z) = \frac{1}{1+e^{-z}}$ (sabemos que $\frac{\partial \text{sigmoid}(z)}{\partial z} = \text{sigmoid}(z) * (1 - \text{sigmoid}(z))$)

La función de error es el Mean Square Error $MSE: (y_{real} - y_{predicha})^2$ y su derivada

$$\frac{\partial MSE}{\partial a} = 2 * (y_{real} - y_{predicha})$$

Calcula la actualización que se producirá con el descenso del gradiente para el peso w_1^1 (el peso 0.1 en la representación matricial). Para ello deberás calcular el error (0.3 puntos), calcular el gradiente asociado a w_1^1 (0.7 puntos) y realizar su actualización con lr=0.01 (0.1 puntos)



$$g = X_1 \cdot w_1^1 + b_1^1$$

$$h_1 = g \cdot w_2^1 + b_2^1$$

$$h_2 = g \cdot w_2^2 + b_2^2$$

$$z = h_1 w_3^1 + h_2 w_3^2 + b_3^1$$

$$z = 0,00085$$

$$\text{sigmoide}(z) = \frac{1}{1+e^{-0,00085}} = 0,5$$

$$MSE = (1 - 0,5)^2 = 0,25$$

$$w_1^1 = w_1^1 - 0,01 \frac{\frac{\partial FC}{\partial w_1^1}}{\frac{\partial FC}{\partial w_1^1}} = 0,1 - 0,01 \cdot 0,002125 = 0,09998$$

$$\frac{\partial FC}{\partial w_1^1} = \frac{\partial FC}{\partial FA} \cdot \frac{\partial FA}{\partial z} \cdot \left(\frac{\partial z}{\partial h_1} \frac{\partial h_1}{\partial g} \frac{\partial g}{\partial w_1^1} + \frac{\partial z}{\partial h_2} \frac{\partial h_2}{\partial g} \frac{\partial g}{\partial w_1^1} \right)$$

$$\frac{\partial FA}{\partial z} = 2(\hat{y} - y) = 2(1 - 0,5) = 1$$

$$\frac{\partial FA}{\partial z} = \text{sigmoide}(z) * (1 - \text{sigmoide}(z)) = 0,5 * (1 - 0,5) = 0,25$$

$$\frac{\partial z}{\partial h_1} \frac{\partial h_1}{\partial g} \frac{\partial g}{\partial w_1^1} = w_1^3 \cdot w_1^2 \cdot x_1 = 0,02 \cdot 0,01 \cdot 5 = 0,001$$

$$\frac{\partial z}{\partial h_2} \frac{\partial h_2}{\partial g} \frac{\partial g}{\partial w_1^1} = w_2^3 \cdot w_2^2 \cdot x_1 = 0,03 \cdot 0,05 \cdot 5 = 0,0075$$

$$\frac{\partial FC}{\partial w_1^1} = 1 \cdot 0,25 \cdot (0,001 + 0,0075) = 0,002125$$

2.1-Iteración del Valor (0.75 puntos) 16,075 por esto !

Suponiendo que disponemos de la siguiente cuadrícula (Grid):



Encontrar **un solo diamante** supone una recompensa de +0.06, encontrar **varios diamantes** de +0.2, y los **lingotes** de +0.5, encontrar **el tesoro** supone una recompensa de +10 y la muerte -10 con la consecuente finalización del juego.

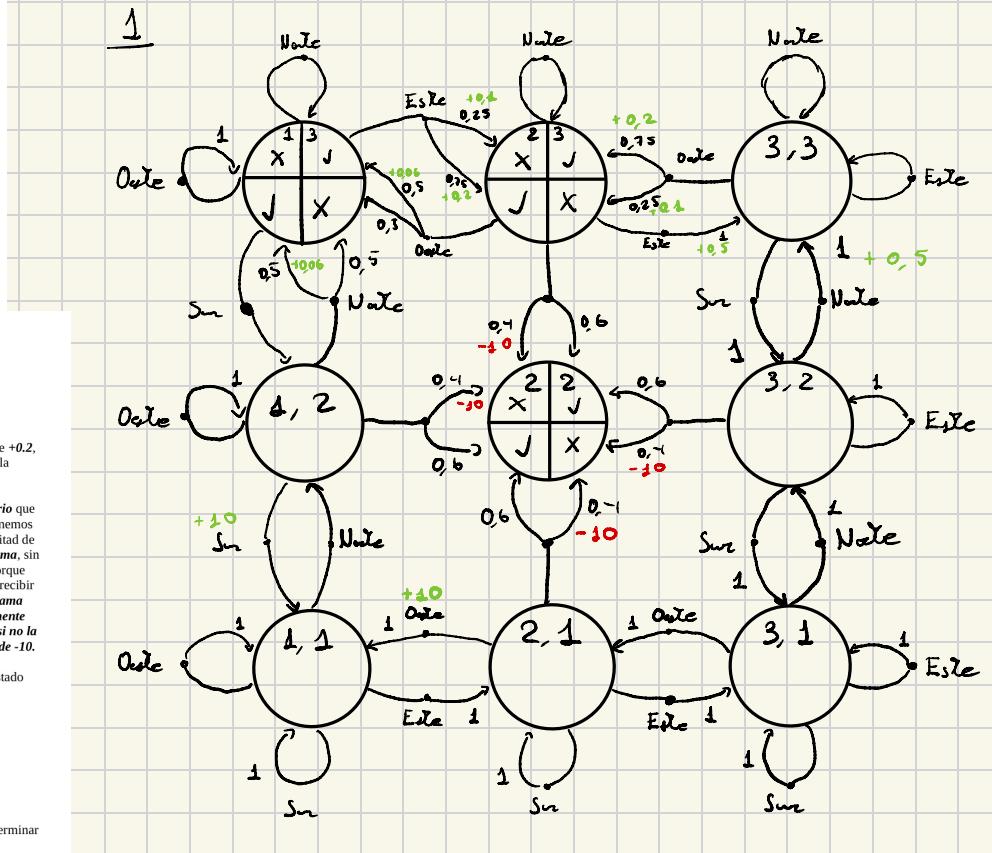
Sabemos que los diamantes que están solos están protegidos por el **espíritu de pirata solitario** que aparece con una probabilidad de 0.5 y cuando aparece esconde el diamante, así que no obtenemos ninguna recompensa. En los lugares con **muchos diamantes** hay un dragón protegiendo la mitad de ellos. Este suele estar dormido el 0.75 de las veces, y si el caso te deja cogerlos sin problema, sin embargo si **está despierto** debes contar que tu **recompensa no será de 0.2** sino de la mitad porque no podrás hacerlo con todos los diamantes. El **tesoro** no está protegido y por lo tanto podrás recibir la recompensa asociada a él en cuanto caigas en el lugar donde se encuentra. Por último, la **dama de la muerte** si caes en su casilla, te hará una pregunta que normalmente responde correctamente el 0.6 de la gente. Si la aciertas el juego acaba, pero no te penalizan con -10, sin embargo, si no la aciertas, no solo finalizará el juego, sino que morirás y recibiras la penalización esperada de -10.

Suponiendo una $\gamma = 0.1$ y sabiendo que la fórmula de Bellman para calcular el valor de un estado es:

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

Calcular el MDP, es decir:

- 1.- las probabilidades de transición y de premio
- 2.- una vez que las hemos identificado y disponemos de ellas, realizar 1 iteración (la 0) y determinar el valor de los estados tras esa iteración. Aplicaremos la versión sincrona.



No pienso escribir esto en notación $T(s, a, s')$...

2)

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

V_0 0 0 0 0 0 0 0 0 0 0

V_1 0 10 0 10 0 0,5 0,275 0,5 0,275

$$V(2,1) = 1 (10 + 0,1 \cdot 0) = 10$$

$$V(1,2) = 1 (10 + 0,1 \cdot 0); 0,5 \cdot 0,06 = \max \left(\underline{10}, 0,03 \right)$$

$$V(1,3) = 0,75 (0,2) + 0,25 (0,1) = 0,175$$

$$V(2,3) = 0,5 (0,06) + 0,5 \cdot 0, 1 \cdot 0,5 = \max (0,03; \underline{0,5})$$

$$V(3,3) = 0,75 (0,2) + 0,25 (0,1) = 0,175$$

$$V(3,2) = 1 \cdot 0,5 = 0,5$$

2.2- Q-learning

En este caso supongamos que en el mismo escenario no disponemos de información sobre el entorno, es decir no conocemos ni la función de transición, ni la de recompensas, y por lo tanto no nos queda más remedio que aplicar aprendizaje-Q (Q-learning), exponiendo al agente al entorno para que éste a través de la experimentación determine cuál es la política que habrá de seguir.

Sabiendo que la fórmula de Bellman para calcular los valores Q es la siguiente y sabiendo que $\alpha=0.1$ y una $\gamma=0.5$.

$$NewQ(s, a) = Q(s, a) + \alpha[R(s, a) + \gamma \max_a Q'(s', a') - Q(s, a)]$$

Diagrama de la fórmula:

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

Se dispone de los siguientes valores-Q (Q-values) tras X episodios,

			0.00	0.00	0.00
			1, 3	2, 3	3, 3
			0.05	0.25	0.62
0.00	0.01	0.00	0.00	0.00	0.00
0.00	1, 2	0.00	0.00	2, 2	0.00
0.94	0.00	0.00	-0.9	3, 2	0.00
0.01	0.00	-0.9	0.00	0.15	0.00
0.00	4, 1	0.00	0.01	2, 1	0.00
0.00	0.00	0.00	0.01	3, 1	0.00

Q Valores actuales

2.2.1 (0.1 puntos) Generar la Tabla-Q (Q-table) asociada a estos valores transformando el dibujo Q Valores actuales en la Tabla-Q.

2.2.2 (0.4 puntos) Suponiendo que el siguiente episodio sea partiendo del {1,2}: ({Arriba, Derecha, Abajo}) actualizar los valores-Q (Q-values) asociados empleando la tabla-Q (Q-table) y escribir las operaciones realizadas a tal efecto.

2.2.3 (0.1 punto) Explica cuando se emplea ϵ -greedy porque se suelen modificar a la vez la α y ϵ .

2.2.4 (0.15 punto) Escribe el algoritmo Q-learning empleando ϵ -greedy e identifica en qué paso del algoritmo se emplea el valor ϵ y que supondría tener un ϵ de 0.8.

2.2.3 ϵ : Es la probabilidad de explorar respecto a explotar

α : Es el learning rate o lo dice si Q cambia rápidamente.

Vamos bajando estos valores porque al principio que no teníamos conocimientos del mundo nos interesa explorar y aprender de él pero a medida que tenemos más datos nos merece más tomar acciones en base a esos datos, lo que se llama explotar.

2.2.4 Si hardcodamos ϵ a 0,8 el program tomará el 80% de las acciones de forma aleatoria.

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$

Initialize $Q(s, a)$, for all $s \in S^+, a \in A(s)$, arbitrarily except that $Q(\text{terminal}, \cdot) = 0$

Loop for each episode:

 Initialize S

 Loop for each step of episode:

 Choose A from S using policy derived from Q (e.g., ϵ -greedy)

 Take action A , observe R, S'

$Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma \max_a Q(S', a) - Q(S, A)]$

$S \leftarrow S'$

 until S is terminal

Arriba	Abajo	Derecha	Izquierda
↑	↓	→	←
2.2.1			
N	S	E	O
1, 1	0,01	0	0
2, 1	-0,9	0	0
3, 1	0,15	0	0
1, 2	0,61	0,94	0
2, 2	0	0	0
3, 2	0,25	0	-0,9
1, 3	0	0	0,05
2, 3	0	0	0,25
3, 3	0	0	0,62

2.2.2

$Q(1, 1, N)$

$$-0,9 + 0,1 [0 + 0,5 \cdot 0 + 0,9] = -0,99$$

$Q(2, 2, E)$

$$0 + 0,1 [0 + 0,5 \cdot 0,25 + 0] = 0,0325$$

$Q(3, 2, S)$

$$0 + 0,1 [0 + 0,5 \cdot 0,15 + 0] = 0,0075$$