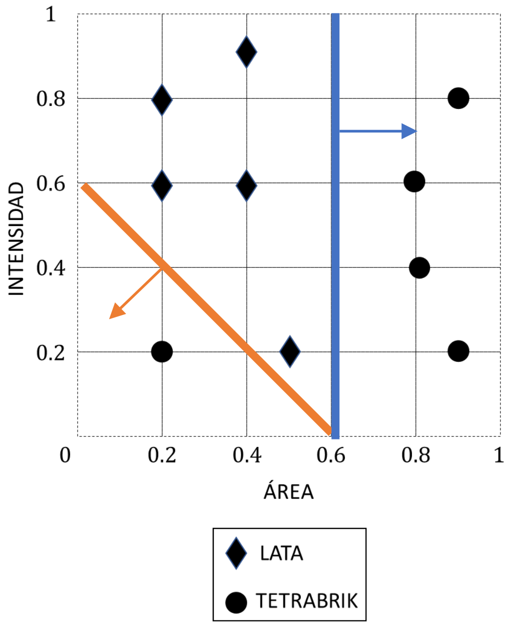
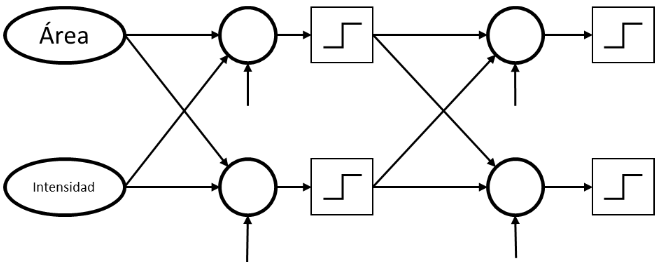
4 de enero de 2021 – Redes Neuronales





No existe una solución única de diseño. Aquí se propone una solución.

La primera neurona, dibujada de azul, tiene parámetros

$$w_1 = (1, 0), \qquad b_1 = -0.6$$

La segunda neurona, dibujada de granate, tiene parámetros

$$w_2 = (-1, -1), \qquad b_2 = 0.6$$

Ambas neuronas con función de activación escalón. La salida de la primera capa para las latas será siempre el vector (0,0), mientras que la salida de la primera capa para los tetrabriks será (1,0) o bien (0, 1).

Nos piden dos neuronas en la capa de salida, una para distinguir cada clase, ambas con función de activación escalón. La neurona asociada a la clase lata,

$$w_3 = (-1, -1), \qquad b_3 = 0.5$$

La neurona asociada a la clase tetrabrik se puede coger negando la neurona de la clase lata

$$w_4 = (1,1), \qquad b_4 = -0.5$$

b) La salida de la primera capa para el dato de entrada (0.6, 0.4) es

$$n = (0.6 - 0.6, -0.6 - 0.4 + 0.6) \Rightarrow y = (1,0)$$

La salida para la segunda capa es

$$n = (-1 + 0.5, 1 - 0.5) \Rightarrow y = (0,1)$$

Por tanto el dato será clasificado como tetrabrik.

Clase	Y_lata	Y_brik	Y_otro
Lata	0.8	0.4	0.7
Lata	0.7	0.5	0.5
Lata	0.75	0.3	0.7
Lata	0.65	0.4	0.75
Brik	0.4	0.75	0.7
Brik	0.4	0.9	0.5
Brik	0.75	0.6	0.7
Brik	0.2	0.7	0.8
Otro	0.8	0.3	0.7
Otro	0.3	0.5	0.6

Matriz de confusión

LATA	3	0	1
BRIK	1	2	1
OTRO	1	0	1
Clase real Clase predicha	LATA	BRIK	OTRO

Matriz de confusión para clase LATA

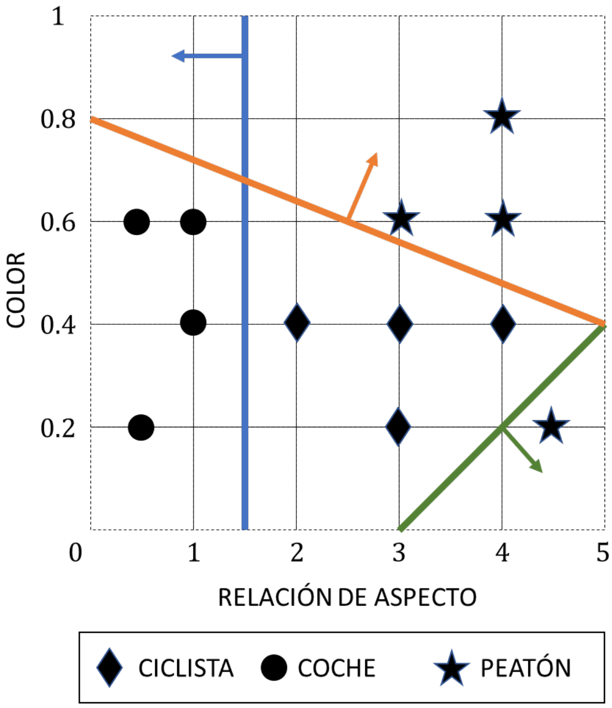
LATA	3	1
NO LATA	2	4
Clase real Clase predicha	LATA	NO LATA

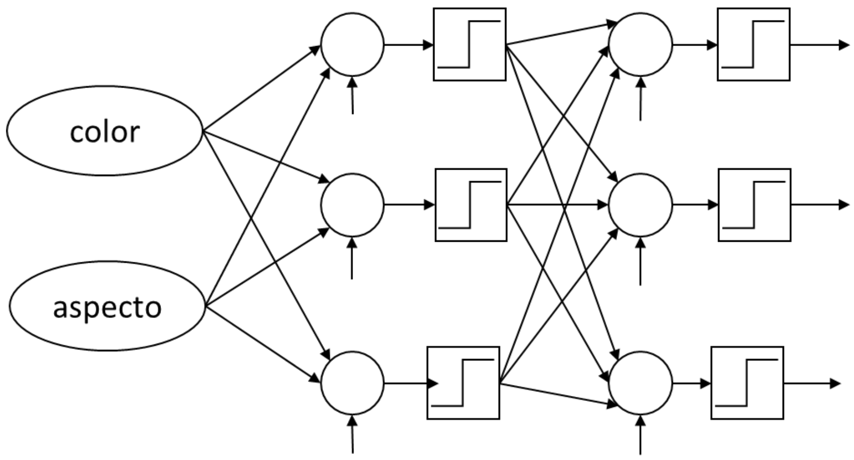
Precisión $= \frac{3}{5} = 0.6$

Recall: $\frac{3}{4} = 0.75$

F1-Score: $\frac{(2 \cdot 0.6 \cdot 0.75)}{0.6+0.75} = 0.66$

9 de septiembre de 2021 – Redes Neuronales
No existe una solución única de diseño. Aquí se propone una solución.





La primera neurona, dibujada de azul, tiene parámetros

$$w_1 = (-1, 0), \qquad b_1 = 1.5$$

La segunda neurona, dibujada de naranja, tiene parámetros

$$w_2 = (0.4, 5), \qquad b_2 = -4$$

La segunda neurona, dibujada de verde, tiene parámetros

$$w_3 = (0.2, -1), \qquad b_3 = -0.6$$

Todas con función de activación escalón. La salida de la primera capa para la clase ciclista será siempre el vector (0,0,0). Para la clase coche la salida será del tipo (1, 0, 0). Para la clase peatón la salida será (0, X, 1-X).

Nos piden tres neuronas en la capa de salida, una para distinguir cada clase. La clase coche se distingue utilizando como entrada la primera neurona

$$w_4 = (1, 0, 0), \qquad b_4 = -0.5$$

La clase ciclista se puede distinguir chequeando que las tres neuronas de la primera capa son 0. Con pesos negativos y un pequeño offset positivo la salida para las otras dos clases será negativa

$$w_5 = (-1, -1, -1), \qquad b_5 = 0.5$$

Por último, la clase peatón se distingue haciendo un OR de la segunda y tercera neurona de la primera capa

$$w_6 = (0, 1, 1), \qquad b_6 = -0.5$$

APARTADO B

Para el umbral $\tau = 0.9$ el sistema considera que ninguna muestra se corresponde con un ciclista. Los círculos azules en la tabla muestran que datos son clasificados como ciclista para el umbral $\tau = 0.3$ mientras que los círculos naranjas muestran los datos clasificados como ciclista para el umbral $\tau = 0.4$

Clase	Y_coche	Y_ciclista	Y_peatón
Coche	0.85	0.1	0.05
Coche	0.75	0.2	0.05
Coche	0.8	0.1	0.1
Ciclista	0.2	● 0.35	0.45
Ciclista	0.1	● 0.5 ●	0.4
Ciclista	0.15	● 0.35	0.5
Ciclista	0.2	● 0.45 ●	0.35
Peatón	0.05	● 0.35	0.6
Peatón	0.15	● 0.3	0.55
Peatón	0.05	● 0.5 ●	0.45

Ahora es necesario calcular la precisión y el recall en cada uno de los tres casos. Para ello construimos primero las matrices de confusión (para la clase ciclista). De izquierda a derecha umbrales 0.9, 0.4 y 0.3.

En la primera tabla (0.9) todas las muestras son clasificadas como no ciclista, tanto los 4 ciclistas como los 3 peatones y los 3 coches. En la segunda tabla

real / predicción	Ciclista	No Ciclista
Ciclista	0	0
No ciclista	4	6

real / predicción	Ciclista	No Ciclista
Ciclista	2	1
No ciclista	2	5

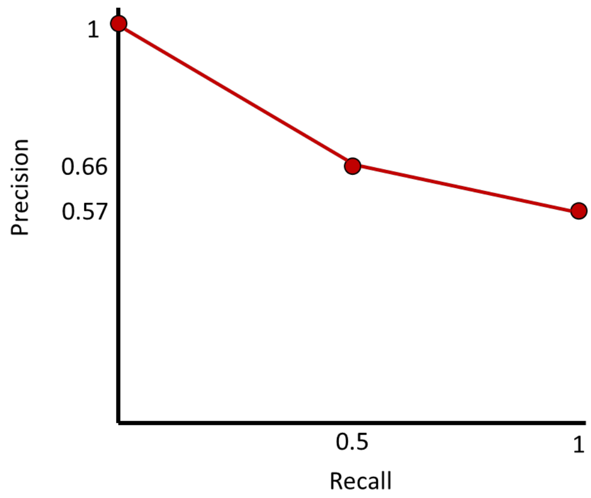
real / predicción	Ciclista	No Ciclista
Ciclista	4	3
No ciclista	0	3

Valores de precisión y recall en cada tabla

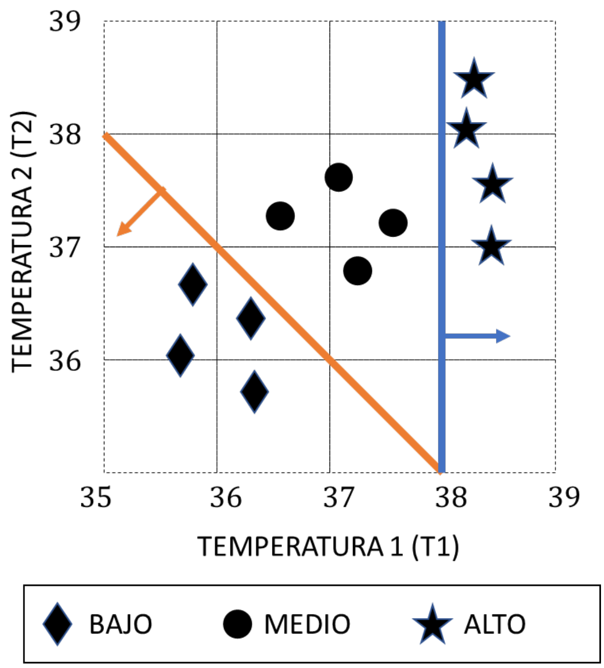
Precisión = $\frac{0}{0+0} = 1$ Recall: $\frac{0}{4} = 0$

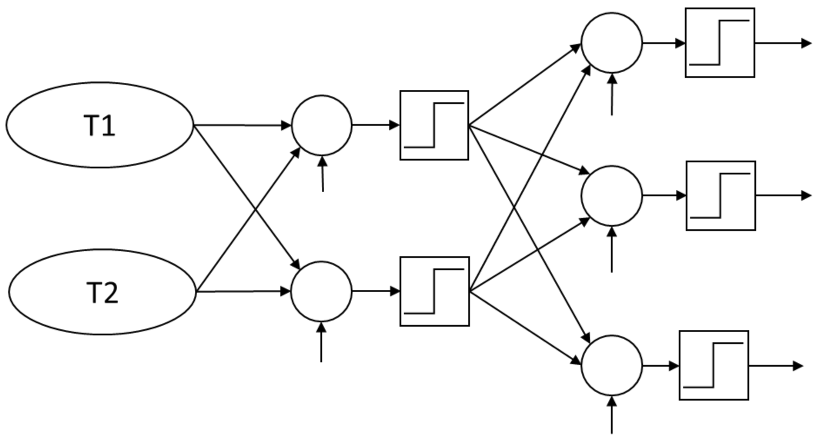
Precisión = $\frac{2}{2+1} = 0.66$ Recall: $\frac{2}{4} = 0.5$

Precisión = $\frac{4}{4+3} = 0.57$ Recall: $\frac{4}{4} = 1$



20 de enero de 2022 – Redes Neuronales
No existe una solución única de diseño. Aquí se propone una solución.





La primera neurona, dibujada de azul, tiene parámetros

$$w_1 = (1, 0), \qquad b_1 = -38$$

La segunda neurona, dibujada de naranja, tiene parámetros

$$w_2 = (-1, -1), \qquad b_2 = 73$$

Ambas con función de activación escalón. La salida de la primera capa para la clase BAJO será siempre el vector (0,1). Para la clase MEDIO la salida será del tipo (0, 0). Para la clase ALTO la salida será (1,0).

Nos piden tres neuronas en la capa de salida, una para distinguir cada clase. La clase ALTO se distingue utilizando como entrada la primera neurona

$$w_3 = (1, 0), \qquad b_3 = -0.5$$

La clase BAJO se puede distinguir utilizando el valor de la segunda neurona. Con pesos negativos y un pequeño offset positivo la salida para las otras dos clases será negativa

$$w_4 = (0, 1), \qquad b_4 = -0.5$$

Por último, la clase MEDIO se distingue verificando que ninguna de las dos entradas de la capa es positiva

$$w_5 = (-1, -1), \qquad b_5 = 0.5$$

APARTADO B

En primer lugar, calculamos el resultado de la capa convolucional. Se aplica el kernel centrado en cada uno de los 4 valores de la matriz x, considerando como valor 0 las multiplicaciones de valores del kernel que no están en la matriz. El resultado es otra matriz de las mismas dimensiones que x

$y_{11} = 2 \cdot 3 + 0{,}8 \cdot 5 + 0{,}5 \cdot 1 + 0 \cdot 4 = 10{,}5$ (kernel centrado en x_{11})
$y_{12} = -1 \cdot 3 + 2 \cdot 5 - 0{,}2 \cdot 1 + 0{,}5 \cdot 4 = 8{,}8$ (kernel centrado en x_{12})
$y_{21} = -1 \cdot 3 - 0{,}3 \cdot 5 + 2 \cdot 1 + 0{,}8 \cdot 4 = 0{,}7$ (kernel centrado en x_{21})
$y_{22} = 0 \cdot 3 - 1 \cdot 5 - 1 \cdot 1 + 2 \cdot 4 = 2$ (kernel centrado en x_{22})

$$y_1 = Conv(x) = \begin{pmatrix} 10{,}5 & 8{,}8 \\ 0{,}7 & 2 \end{pmatrix}$$

La segunda capa hace una selección del máximo en grupos de 2x2, devolviendo en este caso un único valor (el máximo de la matriz)

$$y_2 = MaxPool(y_1) = 10{,}5$$

La tercera capa es un perceptrón con una sola neurona de peso w=1 y offset b=-0,5, con función de activación escalón,

$$y_3 = \mathrm{Fesc}(w_1 y_2 + b_1) = \mathrm{Fesc}(0{,}5 \cdot 10{,}5 - 5) = 1$$

APARTADO C

Se muestran con círculo azul los datos clasificados como ALTO

Clase	Y_ALTO	Y_BAJO
ALTO	0.65	0.35
ALTO	0.75	0.25
ALTO	0.80	0.20
ALTO	0.45	0.55
ALTO	0.65	0.35
BAJO	0.55	0.45
BAJO	0.65	0.35
BAJO	0.45	0.55

La matriz de confusión resultante es

real / predicción	ALTO	NO ALTO
ALTO	4	2
NO ALTO	1	1

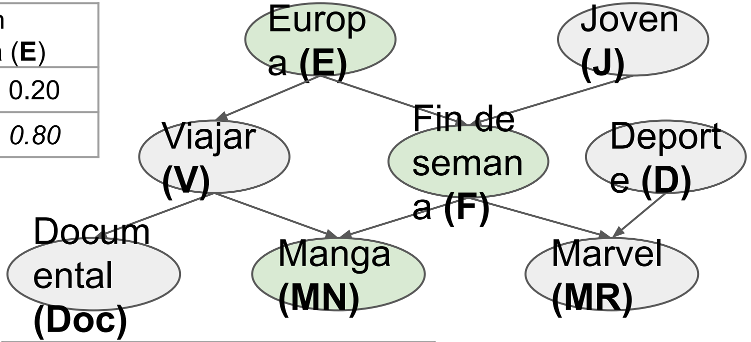
La precisión y el recall

$$P = \frac{4}{4+2} = 0{,}66$$

$$R = \frac{4}{4+1} = 0{,}8$$

Viajar es uno de sus *hobbies* principales	
P(v | +e)	**0.7**
P(v | -e)	0.4
P(-v | +e)	***0.3***
P(-v | -e)	*0.6*

Vive en Europa (**E**)	
P(e)	0.20
P(-e)	*0.80*



Solicita película en fin de semana (**F**)	
P(f | +e, +j)	**0.6**
P(f | +e, -j)	**0.3**
P(f | -e, +j)	0.7
P(f | -e, -j)	0.6
P(-f | +e, +j)	**0.4**
P(-f | +e, -j)	**0.7**
P(-f | -e, +j)	0.3
P(-f | -e, -j)	0.4

Tiene menos de 25 años (**J**)	
P (j)	0.4
P (-j)	0.6

El **Deporte** es uno de sus *hobbies* principales	
P (d)	0.6
P (-d)	0.4

Le gusta el **Documental** propuesto	
P(Doc | +v)	0.6
P(Doc | -v)	0.3
P(-Doc | v)	*0.4*
P(-Doc | -v)	*0.7*

Le gusta la serie **Manga** propuesta	
P(mn | v, f)	0.6
P(mn | v, -f)	0.4
P(mn | -v, f)	0.2
P(mn |- v, - f)	0.1
P(-mn | v, f)	***0.4***
P(-mn | v, -f)	*0.6*
P(-mn | -v, f)	***0.8***
P(-mn |- v, - f)	*0.9*

Le gusta la película de **Marvel** propuesta	
P(mr | d, f)	0.7
P(mr | d, -f)	0.5
P(mr | -d, f)	0.4
P(mr | -d, -f)	0.3
P(-mr | d, f)	
P(-mr | d, -f)	
P(-mr | -d, f)	
P(-mr | -d, -f)	

P(DOC | +e, +f, -mn) → Ignorar D, MR (no ancestro), J (tripleta J-F-MN y VEF INACT lo que hacen los dos caminos inactivos).

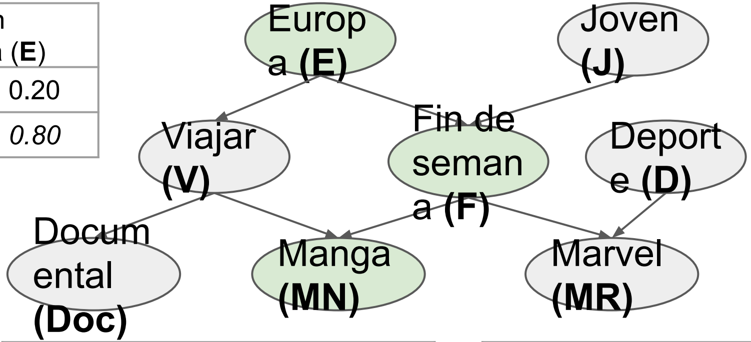
ELIMINAR SOLO V → p(+doc | +f, +e -mn) = **46%**

Versiones mas largas (eliminar V y J por ejemplo) lleva a lo mismo y tambien se considera correcto

20 de enero de 2022

Viajar es uno de sus *hobbies* principales	
P(v | +e)	**0.7**
P(v | -e)	0.4
P(-v | +e)	***0.3***
P(-v | -e)	*0.6*

Vive en Europa (**E**)	
P(e)	0.20
P(-e)	*0.80*



Solicita película en fin de semana (**F**)	
P(f | +e, +j)	0.6
P(f | +e, -j)	0.3
P(f | -e, +j)	0.7
P(f | -e, -j)	0.6
P(-f | …)	

Tiene menos de 25 años (**J**)	
P (j)	0.4
P (-j)	*0.6*

El **Deporte** es uno de sus *hobbies* principales	
P (d)	0.6
P (-d)	*0.4*

Le gusta el **Documental** propuesto	
P(Doc | +v)	0.6
P(Doc | -v)	0.3
P(-Doc | v)	*0.4*
P(-Doc | -v)	*0.7*

Le gusta la serie **Manga** propuesta	
P(mn | v, f)	**0.6**
P(mn | v, -f)	0.4
P(mn | -v, f)	**0.2**
P(mn |- v, - f)	0.1
P(-mn | v, f)	*0.4*
P(-mn | v, -f)	*0.6*
P(-mn | -v, f)	*0.8*
P(-mn |- v, - f)	*0.9*

Le gusta la película de **Marvel** propuesta	
P(mr | d, f)	**0.7**
P(mr | d, -f)	0.5
P(mr | -d, f)	**0.4**
P(mr | -d, -f)	0.3
P(-mr | d, f)	***0.3***
P(-mr | d, -f)	*0.5*
P(-mr | -d, f)	***0.6***
P(-mr | -d, -f)	*0.7*

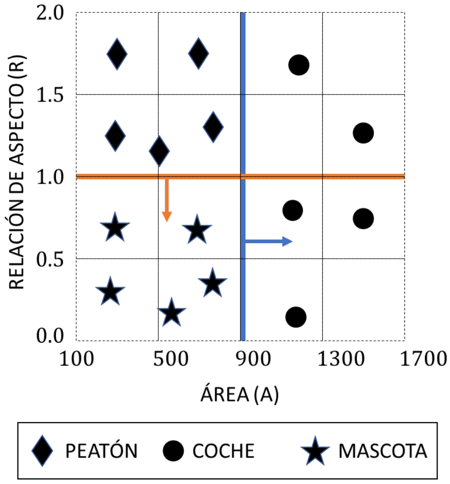
P(MR | +e, +f, -mn) → Ignorar Doc (no ancestro), Ignorar J (tripletas J-F-MR y VEF inactivos) y E (tripletas E-F-MR, y MN-F-MR inactivas), y MN y V (caminos cortados arriba y abajo, inactivos MN-F-MR y V-E-F),

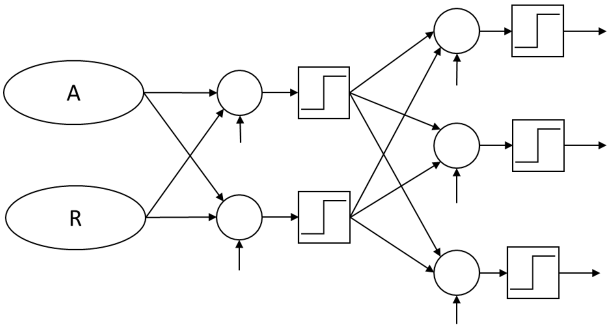
Eliminar SOLO D→ P(MR | +e, +f, -mn) = **P(MR | +f)** → **58%**

Aunque no se ignoren todas las que se puede, se puede hacer un proceso mas largo que lleve al mismo punto

9 de septiembre de 2022 – Redes Neuronales

No existe una solución única de diseño. Aquí se propone una solución.





La primera neurona, dibujada de azul, tiene parámetros

$$w_1 = (1, 0), \qquad b_1 = -900$$

La segunda neurona, dibujada de naranja, tiene parámetros

$$w_2 = (0, - \quad 1), b_2 = 1$$

Ambas con función de activación escalón. La salida de la primera capa para la clase PEATÓN será siempre el vector (0,0). Para la clase COCHE la salida será del tipo (1, X). Para la clase MASCOTA la salida será (0,1).

Nos piden tres neuronas en la capa de salida, una para distinguir cada clase. La clase COCHE se distingue utilizando como entrada la primera neurona

$$w_3 = (1, 0 \quad), b_3 = -0.5$$

La clase MASCOTA se puede distinguir utilizando el valor de ambas neuronas.

$$w_4 = (-1, 1 \quad), b_4 = -0.5$$

Por último, la clase PEATÓN se distingue verificando que ninguna de las dos entradas de la capa es positiva

$$w_5 = (-1, -1), \qquad b_5 = 0.5$$

APARTADO B

La entrada es $x = (600{,}0.9)$

La salida de la primera neurona es $n_1 = (1{,}0)(600{,}0.9)^T - 900 = -300 < 0 \Rightarrow y_1 = 0$

La salida de la segunda neurona es $n_2 = (0, -1)(600{,}0.9)^T + 1 = 0.1 > 0 \Rightarrow y_2 = 1$

La salida de la tercera neurona es $n_3 = (1{,}0)(y_1, y_2)^T - 0.5 = -0.5 < 0 \Rightarrow y_3 = 0$
La salida de la cuarta neurona es $n_4 = (0{,}1)(y_1, y_2)^T - 0.5 = 0.5 > 0 \Rightarrow y_4 = 1$
La salida de la quinta neurona es $n_5 = (-1, -1)(y_1, y_2)^T + 0.5 = -0.5 < 0 \Rightarrow y_5 = 0$

La red neuronal dirá que la muestra es una MASCOTA

APARTADO C

Se muestran con círculo azul la clase ganadora para cada muestra

Clase	Y_coche	Y_mascota	Y_peatón
Coche	0. 3	0.5 ●	0.2
Coche	0.7 ●	0.2	0.1
Coche	0.8 ●	0.1	0.1
Mascota	0.2	0.3	0.5 ●
Mascota	0.1	0.6 ●	0.3
Mascota	0.1	0.8 ●	0.1
Mascota	0.5 ●	0.2	0.3
Peatón	0.1	0.1	0.8 ●
Peatón	0.2	0.3	0.5 ●
Peatón	0.1	0.5 ●	0.4

Se muestra la matriz de confusión para las 3 clases, la matriz de confusión para la clase mascota y las ecuaciones de la precisión, el recall y el F1.

	C	M	P
C	2	1	0
M	1	2	1
P	0	1	2

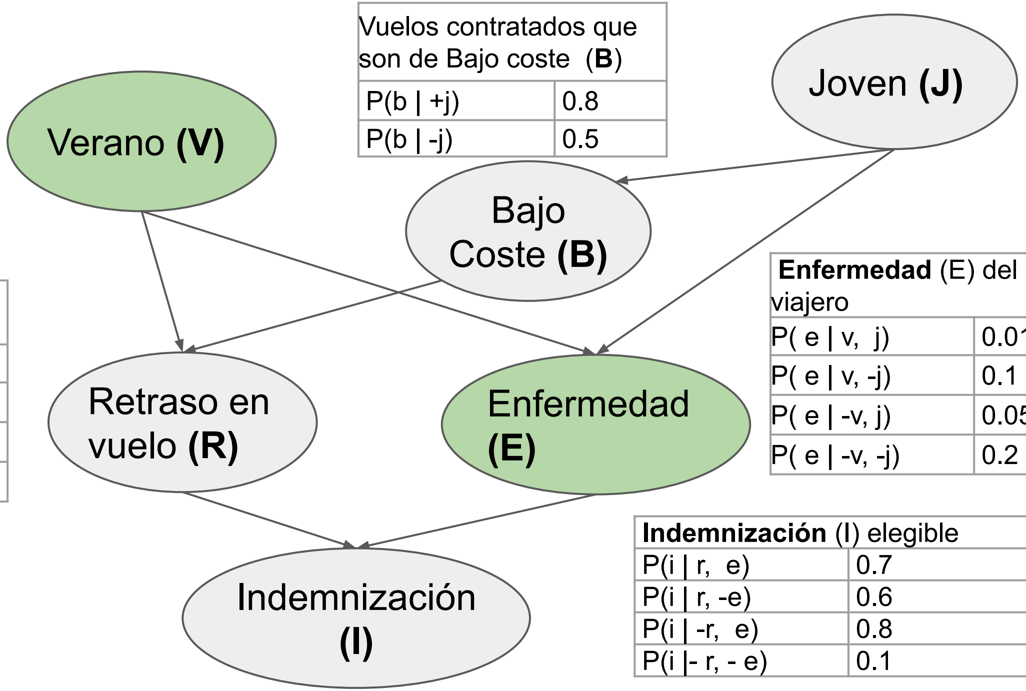
	M	-M
M	2	2
-M	2	4

$$P = \frac{2}{2+2} = 0{,}5$$

$$R = \frac{2}{4} = 0{,}5$$

$$F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = 0{,}5$$

9 de septiembre de 2022



Vuelos que se realizan en **verano** (V)	
P(v)	0.70

Vuelos contratados que son de Bajo coste (**B**)	
P(b | +j)	0.8
P(b | -j)	0.5

Viajero tiene menos de 25 años (**J**)	
P (j)	0.3

Retraso (R) del vuelo	
P(r | v, b)	0.7
P(r | v, -b)	0.4
P(r | -v, b)	0.35
P(r | -v, -b)	0.2

Enfermedad (E) del viajero	
P(e | v, j)	0.01
P(e | v, -j)	0.1
P(e | -v, j)	0.05
P(e | -v, -j)	0.2

Indemnización (I) elegible	
P(i | r, e)	0.7
P(i | r, -e)	0.6
P(i | -r, e)	0.8
P(i |- r, - e)	0.1

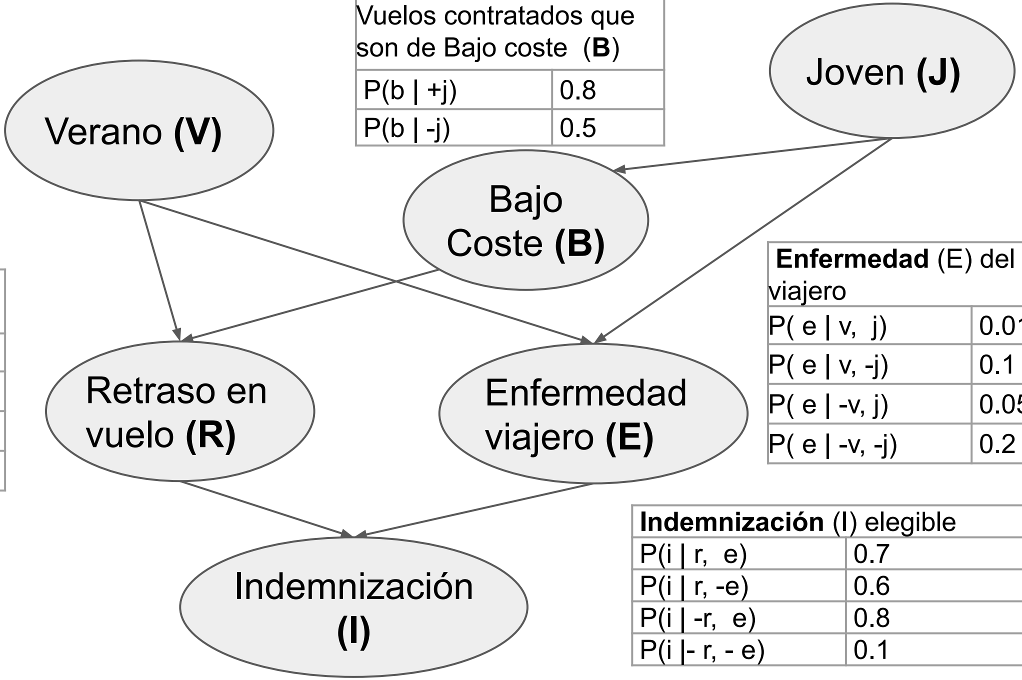
Una agencia de viajes quiere estudiar si debería cobrar el seguro de viaje más alto según la franja de edad de los viajeros (<=25 años, o no). Para ello estudia las probabilidades de tener que pagar indemnización a los viajeros en distintas circunstancias. Responde a estas dos preguntas:

- ¿Este verano, sería lógico cobrar más a los viajeros menores de 25 años o no? (es decir, estima la probabilidad de indemnización en verano, es mayor dado que el pasajero es menor de 25 o si no lo es?). Estima estas probabilidades aplicando el algoritmo de eliminación de variables. Detalla claramente el proceso seguido y los motivos claros si decides ignorar alguna variable.
- Si sabemos que el viajero/a no está enfermo, ¿sigue influyendo la edad en la probabilidad de indemnización? (es decir, ¿son I y J dependientes en este caso?) → **El camino IEJ se vuelve inactivo si marcamos E, pero el IRBJ sigue activo asi que al haber camino, hay dependencia. Sería necesario que no haya ningun camino activo para que sean independientes.**

Vuelos que se realizan en **verano** (**V**)	
P(v)	0.70

Vuelos contratados que son de Bajo coste (**B**)	
P(b | +j)	0.8
P(b | -j)	0.5

Viajero tiene menos de 25 años (**J**)	
P (j)	0.3



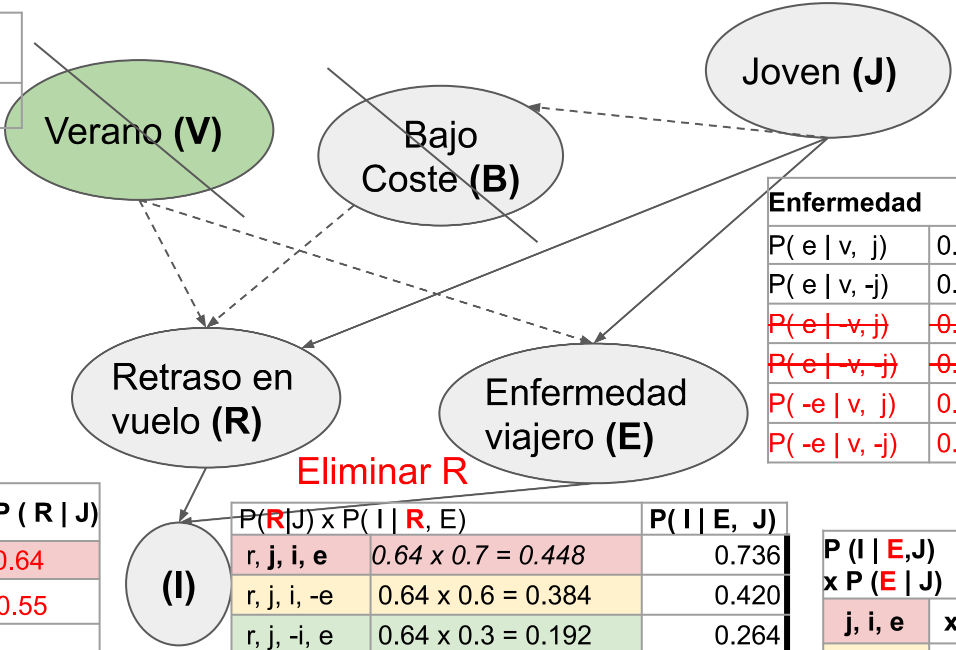
Retraso (R) del vuelo	
P(r | v, b)	0.7
P(r | v, -b)	0.4
P(r | -v, b)	0.35
P(r | -v, -b)	0.2

Enfermedad (E) del viajero	
P(e | v, j)	0.01
P(e | v, -j)	0.1
P(e | -v, j)	0.05
P(e | -v, -j)	0.2

Indemnización (I) elegible	
P(i | r, e)	0.7
P(i | r, -e)	0.6
P(i | -r, e)	0.8
P(i |- r, - e)	0.1

Retraso	
P(r | v, b)	0.7
P(r | v, -b)	0.4
~~P(r | -v, b)~~	~~0.35~~
~~P(r | -v, -b)~~	~~0.2~~
P(-r | v, b)	0.3
P(-r | v, -b)	0.6

Vuelos en verano (**V**)	
P(v)	0.70



Tiene menos de 25 años (**J**)	
P (j)	0.3
P(-j)	0.7

Bajo coste (**B**)	
P(b | +j)	0.8
P(b | -j)	0.5
P(-b | +j)	0.2
P(-b | -j)	0.5

Enfermedad	
P(e | v, j)	0.01
P(e | v, -j)	0.1
~~P(e | -v, j)~~	~~0.05~~
~~P(e | -v, -j)~~	~~0.2~~
P(-e | v, j)	0.99
P(-e | v, -j)	0.9

Indemnización (I) elegible	
P(i | r, e)	0.7
P(i | r, -e)	0.6
P(i | -r, e)	0.8
P(i |- r, - e)	0.1
P(-i | r, e)	0.3
P(-i | r, -e)	0.4
P(-i | -r, e)	0.2
P(-i |- r, - e)	0.9

Query: I; Evidencia: +V;

P(I | J) → Eliminar B, R y E

Eliminar B

P(R | ~~V~~, **B**) x P (**B** | J) → P (R, B | J)		**P (R | J)**
r, v, b, **j**	0.7 x 0.8 = 0.56	0.64
r, v, b, -j	0.7 x 0.5 = 0.35	0.55
r, v, -b, **j**	0.08	
r, v, -b, **-j**	0.20	
- r, v, b, **j**	0.24	0.36
- r, v, b, **-j**	0.15	0.45
- r, v, -b, **j**	0.12	
- r, v, -b, **-j**	0.30	

Eliminar R

P(**R**|J) x P(I | **R**, E)		**P(I | E, J)**
r, **j, i, e**	*0.64 x 0.7 = 0.448*	0.736
r, j, i, -e	0.64 x 0.6 = 0.384	0.420
r, j, -i, e	0.64 x 0.3 = 0.192	0.264
r, j, -i, -e	0.64 x 0.4 = 0.256	0.580
r, -j, i, e	0.55 x 0.7 = 0.385	0.745
r, -j, i, -e	0.55 x 0.6 = 0.330	0.375
r, -j, -i, e	0.55 x 0.3 = 0.165	0.255
r, -j, -i, -e	0.55 x 0.4 = 0.220	0.625
-r, **j, i, e**	0.36 x 0.8 = 0.288	
-r, j, i, -e	0.36 x 0.1 = 0.036	
-r, j, -i, e	0.36 x 0.2 = 0.072	
-r, j, -i, -e	0.36 x 0.9 = 0.324	
-r, -j, i, e	0.45 x 0.8 = 0.360	
-r, -j, i, -e	0.45 x 0.1 = 0.045	

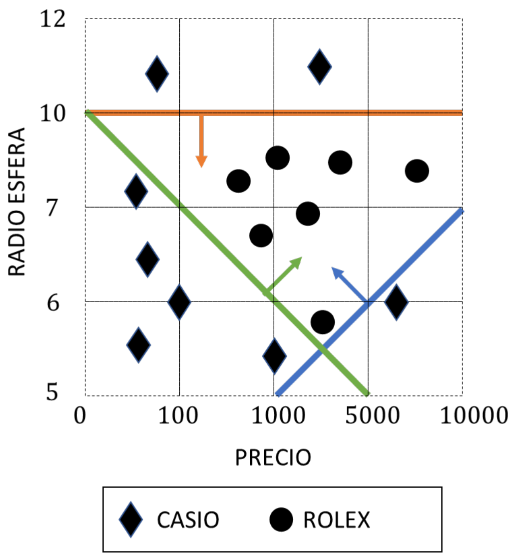
P (I | E,J) x P (E | J)			**P(I | J)**	**Ya esta NORMAL.**
j, i, e	**x 0.01**	**0.0074**	**0.423**	**+j**
j, i, -e	x 0.99	0.4158		
j, -i, e	**x 0.01**	**0.0026**	0.577	
j, -i, -e	x 0.99	0.5742		
-j, i, e	x 0.1	**0.0745**	**0.412**	**-j**
-j, i, -e	x 0.9	0.3375		
-j, -i, e	x 0.1	**0.0255**	0.588	
-j, -i, -e	x 0.9	0.5625		

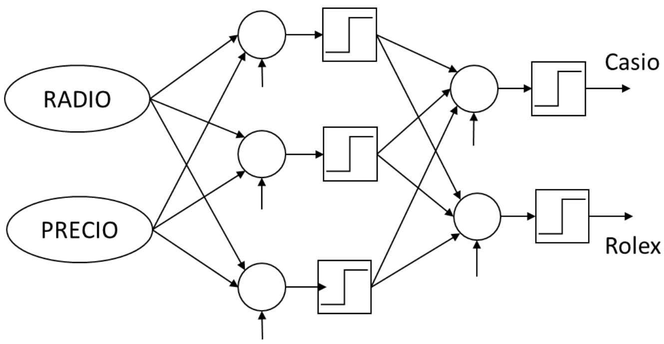
Eliminar E y normalizar

0.412 < 0.423

Duración total del examen: 3 horas

SOLUCIÓN EJ 5





Neurona 1 (al no haber puesto una escala lineal, depende de qué puntos se elijan en el dibujo salen diferentes soluciones que pueden estar bien)

$$d1 = (10000{,}7) - (1000{,}5) = (9000{,}2) \Rightarrow w_1 = (-2{,}9000)$$
$$b_1 = -w_1^T x_1 = 2 \cdot 1000 - 9000 \cdot 5 = -43000$$

Neurona 2

$$d2 = (0{,}10) - (5000{,}5) = (-5000{,}5) \Rightarrow w_2 = (5, 5000)$$
$$b_2 = -w_2^T x_2 = -5 \cdot 0 - 5000 \cdot 10 = -50000$$

Neurona 3

$$w_3 = (0, -1)$$
$$b_3 = -w_3^T x_3 = 0 \cdot 0 + 1 \cdot 10 = 10$$

Salida de la primera capa

Casio = (1,1,0), (1,0,1), (0,1,1)

Rolex = (1,1,1)

Segunda capa

Neurona Rolex:

$$w_4 = (1{,}1{,}1),\ b_4 = -2.5$$

Neurona Casio:

$$w_4 = (-1, -1, -1),\ b_4 = 2.5$$

Apartado b

Marcamos con círculos las muestras clasificadas como Rolex para cada umbral (ver colores)

CURVA PR

$$\tau = (0, 0.2, 0.6, 0.8, 1)$$

Clase	Y_rolex	Clase	Y_rolex
Rolex	0.85 ● ● ● ●	Casio	0.75 ● ● ●
Rolex	0.75 ● ● ●	Casio	0.05 ●
Rolex	0.95 ● ● ● ●	Casio	0.55 ● ●
Rolex	0.4 ● ●	Casio	0.65 ● ● ●
Rolex	0.4 ● ●	Casio	0.1 ●
Rolex	0.9 ● ● ● ●	Casio	0.25 ● ●

Construimos las matrices de confusión para cada umbral (ver colores)

real / pred	R	-R
R	6	6
-R	0	0

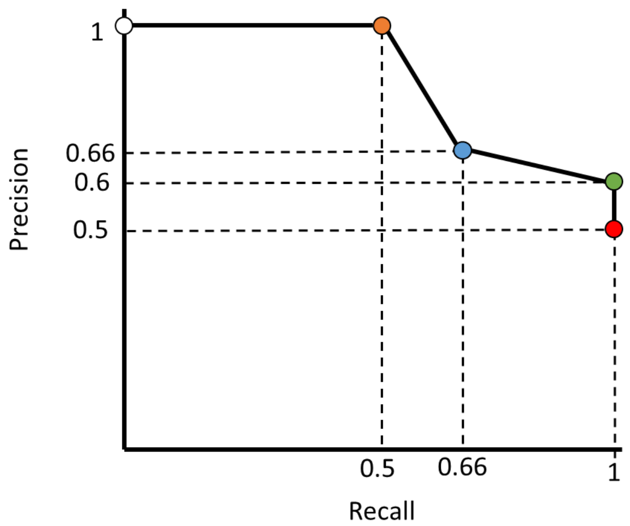
real / pred	R	-R
R	6	4
-R	0	2

real / pred	R	-R
R	4	2
-R	2	4

real / pred	R	-R
R	3	0
-R	3	6

real / pred	R	-R
R	0	0
-R	6	6

Para cada tabla calculamos la precisión (TP/TP+FP) y el recall (TP/TP+FN) y generamos un punto de la gráfica (ver colores). Cuando en la precisión aparece un 0/0 se considera una precisión igual a 1



Duración total del examen: 3 horas

Apartado c

Capa convolucional

$$x = \begin{pmatrix} 10 & 20 \\ 5 & 2 \end{pmatrix} \qquad \text{Kernel} = \begin{pmatrix} -0.1 & 0.1 & 0 \\ 0.1 & 1 & -0.2 \\ -3 & 0.1 & 0 \end{pmatrix}$$

$$y_{11} = 10 \cdot 1 + 20 \cdot -0.2 + 5 \cdot 0.1 + 2 \cdot 0 = 6.5$$
$$y_{12} = 10 \cdot 0.1 + 20 \cdot 1 + 5 \cdot -3 + 2 \cdot 0.1 = 6.2$$
$$y_{21} = 10 \cdot 0.1 + 20 \cdot 0 + 5 \cdot 1 + 2 \cdot -0.2 = 5.6$$
$$y_{22} = 10 \cdot -0.1 + 20 \cdot 0.1 + 5 \cdot 0.1 + 2 \cdot 1 = 3.5$$

Capa flatten

$$y^2 = (6.5,\ 6.2,\ 5.6,\ 3.5)$$

Capa densa (MLP) con $w = (1,\ -1, -1,\ 1)$ y $b = 2$

$$w^T y^2 + b = 6.5 - 6.2 - 5.6 + 3.5 + 2 = 0.2 > 0$$

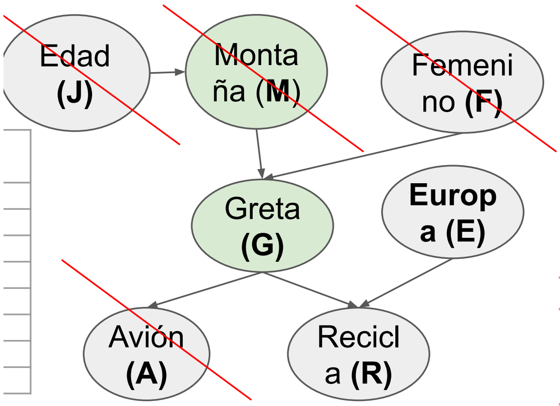
Por tanto la salida es reloj ROLEX

SOLUCIÓN EJ 6

a. A influye en R → tripleta activa

b. P(R | +m, +g)?

- Ignoramos A. no es ancestro de query o evidencia
- Ignoramos J y M (tripleta MGR inactiva, hace que el único camino hasta R desde ellas sea inactivo). Tambien ignoramos F (FGR tripleta inactiva). **P (R | +m, +g) = P(R| +g)**
- Solo hay que eliminar E:



Viaja frecuentemente en **Avión (A)**	
P(a | +g)	0.3
P(a | -g)	0.7
P(-a | g)	*0.7*
P(-a | -g)	*0.3*

Sexo Femenino (F)	
P(f)	0.45
P(-f)	0.55

Vive en **Europa (E)**	
P (e)	0.1
P (-e)	0.9

Tiene menos de 20 años (**J**)	
P (j)	0.33
P (-j)	0.77

Es seguidor/a de **Greta (G)**	
P(g | m, f)	0.9
P(g | m, -f)	0.8
P(g | -m, f)	0.4
P(g |- m, - f)	0.3
P(-g | m, f)	***0.1***
P(-g | m, -f)	*0.2*
P(-g | -m, f)	***0.6***
P(-g |- m, - f)	*0.7*

Recicla de manera habitual **(R)**	
P(r | g, e)	0.95
P(r | g, -e)	0.7
P(r | -g, e)	0.4
P(r | -g, -e)	0.2
P(-r | g, e)	0.05
P(-r | g, -e)	0.3
P(-r | -g, e)	0.6
P(-r | -g, -e)	0.8

Es seguidor/a de un club de **montaña (M)**	
P(m | +j)	0.4
P(m | -j)	0.6
P(-m | +j)	*0.6*
P(-m | - j	*0.4*

Eliminar E											
1) multiplicar									2) marginalizar E		
P(E)		**x**	**P(R|G,E)**			**R, G, E**		**P(R,E | +g)**			**P(R | +g)**
P(e)	**0.1**		**P(r | g, e)**	**0.95**		*r,g,e*	0.1 x 0.95	**0.095**			
P(-e)	**0.9**		**P(r | g, -e)**	**0.7**		*r,g,-e*	0.9 x 0.7	**0.63**	0.095+0.63	**r,g**	**0.725**
			P(r | -g, e)	0.4		*-r,g,e*	0.1 x 0.05	**0.005**			
			P(r | -g, -e)	0.2		*-r,g,-e*	0.9 x 0.3	**0.27**	0.005+0.27	-r,g	0.275
			P(-r | g, e)	**0.05**							
			P(-r | g, -e)	**0.3**							
			P(-r | -g, e)	0.6							
			P(-r | -g, -e)	0.8							

Ya queda normalizada la tabla P(R | +g)

Y la respuesta es P(+r | +g) = 0.725

23 de junio de 2023

Cálculos de la primera capa

$$\mathbf{n}_1 = \mathbf{W}_1\mathbf{x} + \mathbf{b}_1 = \begin{bmatrix} -0.3 \\ 1.2 \end{bmatrix}, \qquad \mathbf{y}_1 = ReLU(\mathbf{n}_1) = \begin{bmatrix} 0 \\ 1.2 \end{bmatrix}$$

Cálculos de la segunda capa

$$\mathbf{n}_2 = \mathbf{W}_2\mathbf{y}_1 + \mathbf{b}_2 = \begin{bmatrix} -0.18 \\ 0.76 \\ -0.52 \end{bmatrix}, \qquad \mathbf{y}_2 = Softmax(\mathbf{n}_2) = \begin{bmatrix} 0.23 \\ 0.60 \\ 0.17 \end{bmatrix}$$

Matriz de confusión

Clase predicha

Clase real		y1	y2	y3
	Y1	4	0	1
	Y2	0	2	3
	y3	0	0	5

Accuracy clase 1

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} = \frac{4 + 10}{15} = 0.93$$

Curva precision – recall

CURVA PR $\quad \tau = (0, 0.2, 0.5, 0.8, 1)$

Clase	y_1	y_2	y_3
1	0.6	0.1	0.3
1	0.8	0.1	0.1
1	0.4	0.1	0.5
1	0.9	0.0	0.1
1	0.5	0.1	0.4
2	0.3	0.3	0.4
2	0.1	0.4	0.5
2	0.0	0.9	0.1
2	0.2	0.6	0.2
2	0.1	0.2	0.7
3	0.0	0.2	0.8
3	0.1	0.2	0.7
3	0.1	0.1	0.8
3	0.0	0.1	0.9
3	0.3	0.1	0.6

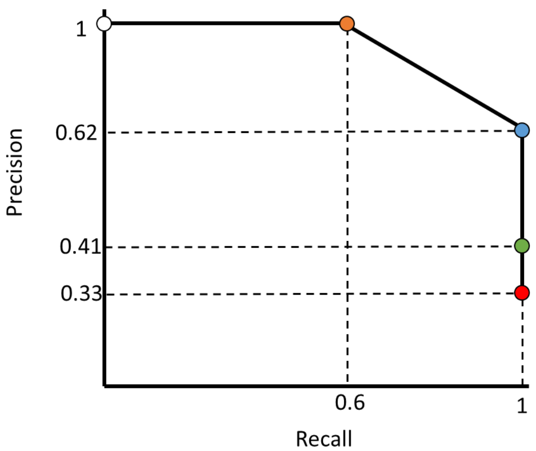
real / pred	Y3	-Y3
Y3	5	10
-Y3	0	0

real / pred	Y3	-Y3
Y3	5	7
-Y3	0	3

real / pred	Y3	-Y3
Y3	5	3
-Y3	0	7

real / pred	Y3	-R
Y3	3	0
-R	2	10

real / pred	Y3	-Y3
Y3	0	0
-Y3	5	10



						P(V) x P(D|v) x P(C|v)		**Elimino V**	**P(D,C)**	x
P (v)	0.25	P(d | +v)	0.4	P(c | +v)	0.6	*v, d, c*	0.06	***d,c***	**0.09**	
P(-v)	0.75	P(d | -v)	0.2	P(c | -v)	0.2	*-v, d, c*	0.03			
		P(-d | +v)	0.6	~~P(-c | +v)~~	~~0.4~~	*v, -d, c*	0.09	***-d,c***	**0.21**	
		P(-d | -v)	0.8	~~P(-c | -v)~~	~~0.8~~	*-v, -d, c*	0.12			

	P(L)		**P(Inu | L, +t)**	**ELIMINO L**			**P(Inu | +t)**
P(L=M)	0.2	**P(Inu | t, m)**	**0.001**		t, L=m	0.0002	**0.0627**
P(L=C)	0.45	**P(Inu | t, c)**	**0.1**		t, L=c	0.045	
P(L=I)	0.35	**P(Inu | t, i)**	**0.05**		t, L=i	0.0175	
		~~P(Inu | -t, m)~~	~~0.0001~~				
		~~P(Inu | -t, c)~~	~~0.001~~				
		~~P(Inu | -t, i)~~	~~0.001~~				**P(- Inu | +t)**
		P(-Inu | t, m)	**0.999**			0.1998	**0.9373**
		P(-Inu | t, c)	**0.9**			0.405	
		P(-Inu | t, i)	**0.95**			0.3325	
		~~P(-Inu | -t, m)~~	~~0.9999~~				
		~~P(-Inu | -t, c)~~	~~0.999~~				
		~~P(-Inu | -t, i)~~	~~0.999~~				

P(Inc|D, t)			P(D,C) x P(Inc|D, t)	**Elimino D**	**P(+c, +Inc | T)**	x	**P(T | c)**		P (+c, T, +inc)		**P(T| +c, +Inc)**
P(inc | +d +t)	0.4	*d, c, t, Inc*	0.036		**0.0885**		P(t | c)	0.05	0.004425	t, c, inc	**0.067**
P(inc | +d -t)	0.25	*-d, c, t, Inc*	0.0525				~~P(t | -c)~~	~~0.005~~			
P(inc | -d +t)	0.25	*d, c, -t, Inc*	0.0225		**0.0645**		P(-t | c)	0.95	0.061275	-t, c, inc	**0.933**
P(inc | -d -t)	0.2	*-d, c, -t, Inc*	0.042				~~P(-t | -c)~~	~~0.995~~			
~~P(-inc | +d +t)~~	~~0.6~~										
~~P(-inc | +d -t)~~	~~0.75~~										
~~P(-inc | -d +t)~~	~~0.75~~										
~~P(-inc | -d -t)~~	~~0.8~~										

16 ENERO 2024

APARTADO A) **P(G1 | L1, +c, -o1)**

Se pueden ignorar G0 y O0, solo queda como variable Hidden la variable COV

1) ELIMINAR COV

P(O1 | G1, C, COV)		x	P(COV | L1)		"="	*JOIN* P(-o1,COV|G1,+c, L1)	
~~o1, +g1, +c, +COV~~	~~0.85~~		**cov, L1**	**0.2**		~~0.17~~	~~o1, +g1, +c, +COV, L1~~
~~o1, -g1, +c, +COV~~	~~0.05~~		~~cov, L2~~	~~0.4~~		~~0.01~~	~~o1, -g1, +c, +COV, L1~~
~~o1, +g1, -c, +COV~~	~~0.8~~		~~cov, L3~~	~~0.6~~			**-o1, +g1, +c, +COV, L1**
~~o1, -g1, -c, +COV~~	~~0.1~~		**-cov, L1**	**0.8**			**-o1, -g1, +c, +COV, L1**
~~o1, +g1, +c, -COV~~	~~0.95~~		~~-cov, L2~~	~~0.6~~		~~0.76~~	
~~o1, -g1, +c, -COV~~	~~0.01~~		~~-cov, L3~~	~~0.4~~		~~0.008~~	
~~o1, +g1, -c, -COV~~	~~0.9~~						
~~o1, -g1, -c, -COV~~	~~0.05~~						
-o1, +g1, +c, +COV	0.15					0.03	
-o1, -g1, +c, +COV	0.95					0.19	
~~-o1, +g1, -c, +COV~~	~~0.2~~						
~~-o1, -g1, -c, +COV~~	~~0.9~~						
-o1, +g1, +c, -COV	0.05					0.04	
-o1, -g1, +c, -COV	0.99					0.792	
~~-o1, +g1, -c, -COV~~	~~0.1~~						
~~-o1, -g1, -c, -COV~~	~~0.95~~						

MARGINALIZE — **2) JUNTAR TODO y NORMALIZAR**

P(-o1 | G1, +c, L1)	P(L1)		P(+c)		P(G1 | L1)		P(-o1, G1 | L1, +c)	P(G1 | -01, L1, +c)	
~~0.93~~	**L1**	**0.5**	**c**	**0.6**	g1, L1	0.5	0.0105	**0.067**	**"+g1"**
~~0.018~~	~~L2~~	~~0.3~~	~~-c~~	~~0.4~~	~~g1, L2~~	~~0.6~~	0.1473	0.933	"-g1"
0.07	~~L3~~	~~0.2~~			~~g1, L3~~	~~0.7~~		1.000	
0.982					-g1, L1	0.5			
					~~-g1, L2~~	~~0.4~~			
					~~-g1, L3~~	~~0.3~~			

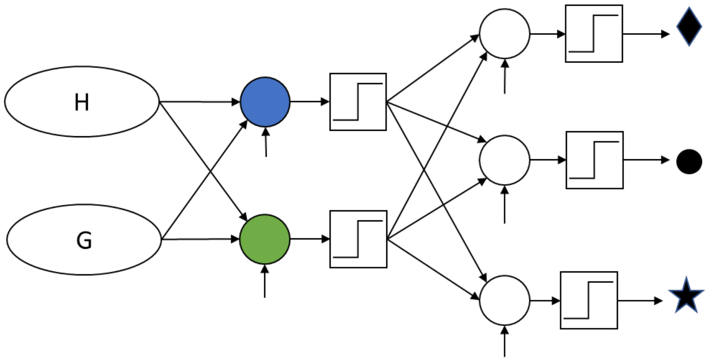
APARTADO B) **P(G0 | L1, +c, -o0, COV)** **NADA QUE ELIMINAR. DIRECTAMENTE, JUNTAR TODO y NORMALIZAR**

Se pueden ignorar G1 y O1, todo lo demas son query o evidencias, asi que no hay que eliminar nada

P(G0 | L1)		X	P(O0 | G0, +c, +COV)		X	P(+COV | L1)		X	P(L1)		X	P(+c)		"="	P(G0, -o0, +COV, L1, +c)		P(G0 |-o0, +COV, L1, +c)	
g0, L1	**0.7**		~~o0, +g0, +c, +COV~~	0.85		**cov, L1**	**0.2**		**L1**	**0.5**		**c**	**0.6**		0.0063	*g0*	**0.27**	*g0*
~~g0, L2~~	~~0.8~~		~~o0, -g0, +c, +COV~~	0.05		~~cov, L2~~	~~0.4~~		~~L2~~	~~0.3~~		~~-c~~	~~0.4~~		0.0171	*-g0*	**0.73**	*-g0*
~~g0, L3~~	~~0.9~~		~~o0, +g0, -c, +COV~~	0.8		~~cov, L3~~	~~0.6~~		~~L3~~	~~0.2~~								
-g0, L1	**0.3**		~~o0, -g0, -c, +COV~~	0.1		~~-cov, L1~~	~~0.8~~											
~~-g0, L2~~	~~0.2~~		~~o0, +g0, +c, -COV~~	0.95		~~-cov, L2~~	~~0.6~~											
~~-g0, L3~~	~~0.1~~		~~o0, -g0, +c, -COV~~	0.01		~~-cov, L3~~	~~0.4~~											
			~~o0, +g0, -c, -COV~~	0.9														
			~~o0, -g0, -c, -COV~~	0.05														
			-o0, +g0, +c, +COV	**0.15**														
			-o0, -g0, +c, +COV	**0.95**														
			~~-o0, +g0, -c, +COV~~	~~0.2~~														
			~~-o0, -g0, -c, +COV~~	~~0.9~~														
			~~-o0, +g0, +c, -COV~~	~~0.05~~														
			~~-o0, -g0, +c, -COV~~	~~0.99~~														
			~~-o0, +g0, -c, -COV~~	~~0.1~~														
			~~-o0, -g0, -c, -COV~~	~~0.95~~														

16 ENERO 2024

A) Como los datos no son linealmente separables, necesitamos una arquitectura de dos capas. La primera capa se puede diseñar con dos neuronas, mientras que la segunda capa necesita tres neuronas, una para cada clase:



Diseño de la primera capa

Neurona azul

$$w_1 = (-1{,}0),\ b_1 = 0.35$$

Neurona verde

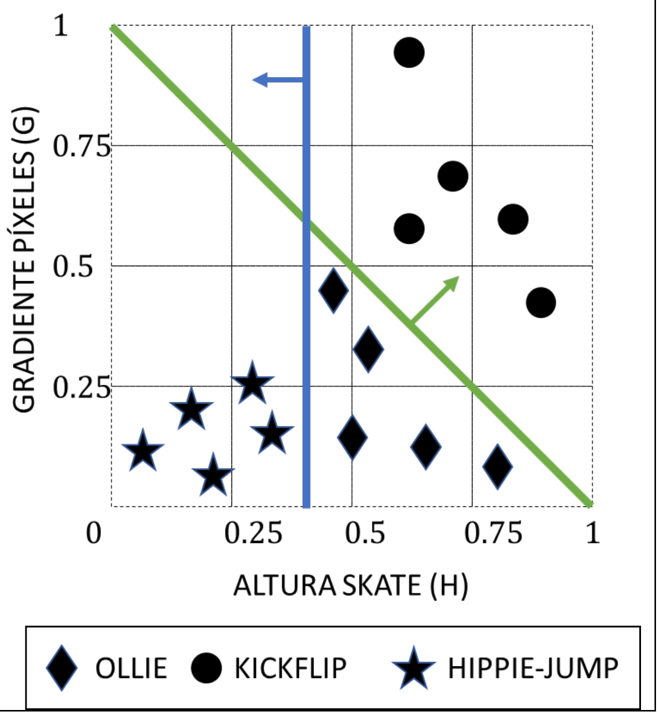
$$w_2 = (1{,}1),\ b_2 = -1$$

Salida de la primera capa para cada clase:

Ollie: $y_1 = (0{,}0)$

Kickflip: $y_2 = (0{,}1)$

Hippie: $y_3 = (1{,}0)$



Diseño de la segunda capa.

Neurona para reconocer Ollie: $w_3 = (-1,-1),\ b_3 = 0{,}5$

Neurona para reconocer Kickflip: $w_4 = (0{,}1),\ b_4 = -0{,}5$

Neurona para reconocer Hippie-Jump: $w_5 = (1{,}0),\ b_5 = -0{,}5$

B) Para cada uno de los 20 datos marcamos la clase predicha por la red neuronal (en verde):

Truco	y_{ol}	y_k	y_h	y_{ot}	Truco	y_{ol}	y_k	y_h	y_{ot}
Ollie	0,83	0,85	0,27	0,37	Hippie	0,78	0,61	0,92	0,76
Ollie	0,93	0,83	0,33	0,66	Hippie	0,67	0,39	0,52	0,68
Ollie	0,88	0,68	0,25	0,78	Hippie	0,67	0,42	0,83	0,73
Ollie	0,85	0,76	0,15	0,58	Hippie	0,47	0,55	0,87	0,76
Ollie	0,70	0,74	0,28	0,07	Hippie	0,58	0,5	0,69	0,79
Kickflip	0,64	0,79	0,11	0,88	Other	0,79	0,73	0,82	0,83
Kickflip	0,76	0,86	0,32	0,98	Other	0,71	0,83	0,61	0,9
Kickflip	0,79	0,70	0,19	0,47	Other	0,66	0,83	0,62	0,87
Kickflip	0,52	0,72	0,17	0,53	Other	0,69	0,67	0,9	0,67
Kickflip	0,63	0,96	0,26	0,44	Other	0,83	0,69	0,69	0,61

A partir de estos datos construimos la matriz de confusión:

True\pred	Ollie	Kickflip	Hippie	Other
Ollie	3	2	0	0
Kickflip	1	2	0	2
Hippie	0	0	3	2
Other	1	0	1	3

C) Se calcula la precisión individual de cada clase:

P_ollie $= \frac{3}{5} = 0.6$

P_kickflip $= \frac{2}{4} = 0.5$

P_hippie $= \frac{3}{4} = 0.75$

P_other $= \frac{3}{7} = 0.43$

P_media $= 0.57$

20 JUNIO 2024

a) CALCULAR P (G0 | -o0, D) y P (G0 | -o0, N)

P(T)	
Day	0.7
Night-N	0.3

P(I | T)	
"+i, D	0.6
"+i, N	0.1
"-i, D	**0.4**
"-i, N	**0.9**

P(G0)		P(G1)	
"+g0"	0.7	"+g1"	0.6
"-g0"	**0.3**		**0.4**

1) ELIMINAR I

2) JUNTAR TODO Y NORMALIZAR (EN EL CASO DE T = DAY

2) JUNTAR TODO Y NORMALIZAR (EN EL CASO DE T = Night

P(O0 | G0, T, I)		"x"	**P(I | T)**		"="	P(O0, I | G0, T)	SUM	**P(O0 = -o0 | G0, T)**	**" x P(T = D) x P(G0)**	"="	**P(-o0, D, G0)**	**P(G0 | -o0, D)**		**" x P(T = N) x P(G0)**	**P(-o0, N, G0)**	**P(G0 | -o0, N)**		
~~**"+o0, +g0, D, +i"**~~	~~0.5~~		"+i, D	0.6		~~**0.3**~~		~~0.68~~										
~~**"+o0, -g0, D, +i"**~~	~~0.05~~		"+i, N	0.1		~~0.03~~		~~0.034~~										
~~"+o0, +g0, N, +i"~~	~~0.3~~		**"-i, D**	**0.4**		~~0.03~~		~~0.66~~										
~~"+o0, -g0, N, +i"~~	~~0.15~~		**"-i, N**	**0.9**		~~0.015~~		~~0.105~~										
~~"+o0, +g0, D, -i~~	~~0.95~~					~~0.38~~												
~~"+o0, -g0, D, -i~~	~~0.01~~					~~0.004~~												
~~"+o0, +g0, N, -i~~	~~0.7~~					~~0.63~~												
~~"+o0, -g0, N, -i~~	~~0.1~~					~~0.09~~						~~1.00~~						
"-o0, +g0, D, +i"	**0.5**					0.3		**0.32**	x 0.7 x 0.7		0.16	**0.44**	**+g0**					
"-o0, -g0, D, +i"	0.95					0.57		0.966	x 0.7 x 0.3		0.20	**0.56**	-g0			~~1.00~~		
"-o0, +g0, N, +i"	0.7					0.07		0.34						x 0.3 x 0.7	0.07	**0.47**	**+g0**	**POR LA NOCHE ES MAS PROBABLE QUE HAYA FANTASMA 0!**
"-o0, -g0, N, +i"	0.85					0.085		**0.895**						x 0.3 x 0.3	0.08	0.53	-g0	
"-o0, +g0, D, -i	**0.05**					0.02												
"-o0, -g0, D, -i	0.99					0.396												
"-o0, +g0, N, -i	0.3					0.27												
"-o0, -g0, N, -i	0.9					0.81												

b) CALCULAR P (I | +g1, -o1, D)

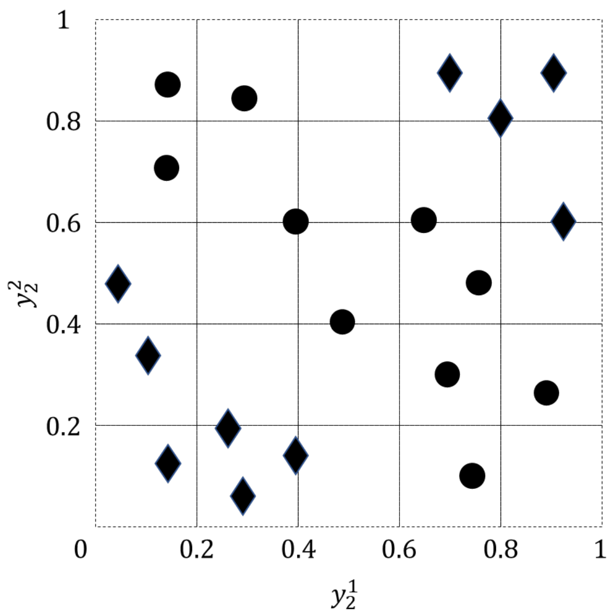
NADA QUE ELIMINAR, JUNTAR Y NORMALIZAR

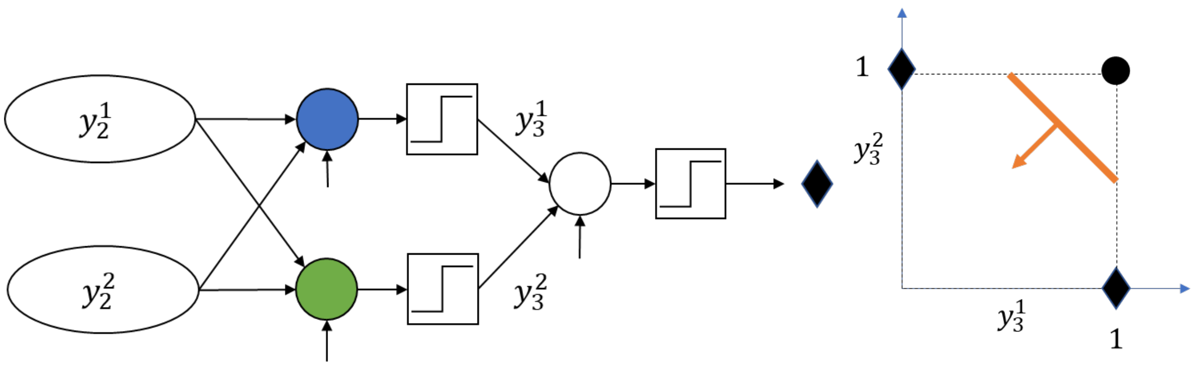
P(-o1 | +g1, D, I)		**x**	**P(+g1)**	**x**	**P(I | T=D)**		**P(D)**	P(I, D, +g1, -o1)	NORM	**P(I | D, +g1, -o1)**	
"-o1, +g1, D, +i	0.5		0.6		"+i, D	0.6	0.7	0.126		**0.94**	"+i"
"-o1, +g1, D, -i	0.05		0.6		"-i, D	0.4	0.7	0.0084		**0.06**	"-i"

20 JUNIO 2024

$$y_1^1 = RELU\begin{pmatrix} 1-4+2 & 4-1+3 \\ 2-1-3 & 3-2-4 \end{pmatrix} = RELU\begin{pmatrix} -1 & 6 \\ -2 & -3 \end{pmatrix} = \begin{pmatrix} 0 & 6 \\ 0 & 0 \end{pmatrix} \Rightarrow y_2^1 = 6$$

$$y_1^2 = RELU\begin{pmatrix} 1-3 & 4+2 \\ 2-4 & 3+1 \end{pmatrix} = RELU\begin{pmatrix} -2 & 6 \\ -2 & 4 \end{pmatrix} = \begin{pmatrix} 0 & 6 \\ 0 & 4 \end{pmatrix} \Rightarrow y_2^2 = 6$$





$$\boldsymbol{w_1 = (-1,-1),\ b_1 = 1.4}$$
$$\boldsymbol{w_2 = (1,1),\ b_2 = -0.6}$$
$$\boldsymbol{w_3 = (-1,-1),\ b_3 = 1.5}$$