Reconocimiento de patrones

Categorías de reconocimiento:

1) Reconocimiento de objetos concretos

   Es el reconocimiento **perceptual**
   - forma (patrón espacial)
   - secuencia (patrón temporal)
2) Reconocimiento de objetos abstractos

   Es el reconocimiento **conceptual**

Vamos a trabajar con el reconocimiento **perceptual**

El reconocimiento de patrones pasa por tres etapas:

1) Sensado:

   Permite representar la información obtenida a través de un dispositivo sobre algún objeto a ser reconocido.
   Tomamos las medidas que van a formar los descriptores de un objeto.
2) Extracción de características:

   Es tratar de reducir las características para hacer menos complejo el cálculo de la clasificación sin perder información importante y representativa de estos patrones.
3) Clasificación:

   Son las técnicas de reconocimiento de patrones

Nosotros vamos a trabajar sobre la etapa de **clasificación**, con el cálculo de la **función de decisión lineal** que traza una recta y clasifica a las clases
Cuando trabajamos con patrones vamos a buscar que un patrón pertenezca a una cierta clase
La agrupación de objetos de una misma clase se llama clúster

La función de decisión lineal está formada por los (W_i) que más adelante van a ser los pesos en lo que serían las redes neurales y ahora son los coeficientes de la matriz A.

Existen dos tipos de descripciones de patrones:

1) Cuantitativa a través de vectores -> los que vamos a usar
2) Estructural a través de cadenas y árboles

$$d(x) = \sum_{i=0}^{n} w_i * x_i = 0$$

Los vectores X_i son las variables de entrada
Los vectores W_i son el vector de parámetros

X_i representa el i-ésimo descriptor siendo n el número de descriptores
Los patrones vectoriales se representan como columnas (matrices n x 1):

$$\begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \quad \text{ó} \quad x = (x_1, x_2, \dots, x_n)^T$$

Ejemplo sencillo: https://youtu.be/YlULgHj1mv8?list=PLsI9mUiT17yyWuesfvYmHRtKyacQIsWeP&t=509

Ejemplo paso a paso: https://youtu.be/YlULgHj1mv8?list=PLsI9mUiT17yyWuesfvYmHRtKyacQIsWeP&t=753

Cálculo de los coeficientes de la matriz A que es el vector de los pesos

$$A = (X^T * X)^{-1} * X^T * Y$$

Y es el vector de salidas

La inversa $(X^T * X)^{-1}$ se da como dato porque es engorroso así que tengo que multiplicar ese dato por $X^T * Y$

1) Primero dejamos expresada X (entradas) como vector aumentado:
Por ejemplo para las entradas: (0 0), (0 1), (1 0), (1 1)

$$X = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Se agrega esa columna de unos rojos para facilitar los cálculos matriciales y se aproxima a lo que en redes neurales se llama umbral.

2) Se calcula la transpuesta X^T
Cada columna pasa a ser fila
¿Cómo? Columna 1 = Fila 1, … Columna n = Fila n
Girando los valores en sentido antihorario

$$X^T = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

3) Producto de $X^T * X$ (viene como dato la inversa así que no haría falta este paso)
Recordatorio de como multiplicar:
Para calcular el elemento C_{ij} siendo i la fila y j la columna multiplico cada elemento de la fila i con su correspondiente elemento en la columna j. Para el primer elemento sería:
0*0 + 0*0 + 1*1 + 1*1 = 1+1 = 2

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 2 \\ 2 & 2 & 4 \end{bmatrix}$$

4) Calcular la inversa (viene como dato)

$$(X^T . X)^{-1} = \begin{bmatrix} 1 & 0 & -0{,}5 \\ 0 & 1 & -0{,}5 \\ -0{,}5 & -0{,}5 & 0{,}75 \end{bmatrix}$$

5) Calcular el producto de la inversa por X^T

$$(X^T . X)^{-1} = \begin{bmatrix} 1 & 0 & -0{,}5 \\ 0 & 1 & -0{,}5 \\ -0{,}5 & -0{,}5 & 0{,}75 \end{bmatrix} \blacksquare \; X^T = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -0{,}5 & -0{,}5 & 0{,}5 & 0{,}5 \\ -0{,}5 & 0{,}5 & -0{,}5 & 0{,}5 \\ 0{,}75 & 0{,}25 & 0{,}25 & -0{,}25 \end{bmatrix}$$

6) Multiplicar por el vector de salidas Y obteniendo la matriz A

$$(X^T . X)^{-1} . X^T = \begin{bmatrix} -0{,}5 & -0{,}5 & 0{,}5 & 0{,}5 \\ -0{,}5 & 0{,}5 & -0{,}5 & 0{,}5 \\ 0{,}75 & 0{,}25 & 0{,}25 & -0{,}25 \end{bmatrix} \blacksquare \; Y = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 1 \end{bmatrix}$$

7) Comprobación -> $X * A = Y$

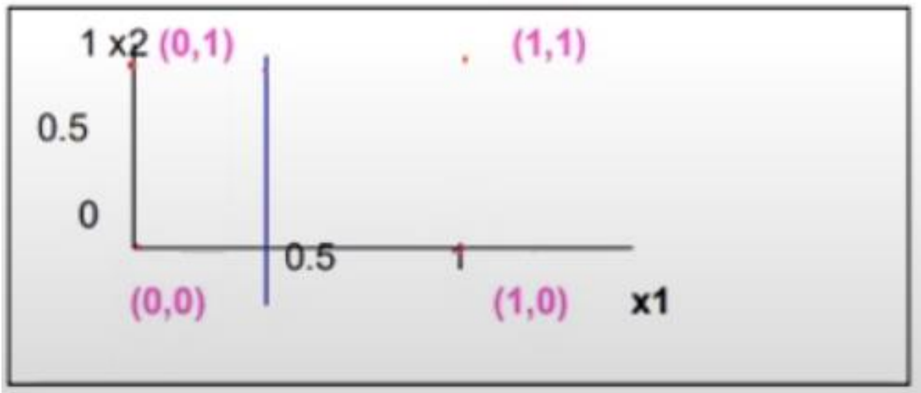
$$X = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \blacksquare \; A = \begin{bmatrix} -2 \\ 0 \\ 1 \end{bmatrix} \begin{matrix} (a1) \\ (a2) \\ (a3) \end{matrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

8) Planteo de la función de decisión lineal

$$a_n = w_n\ (pesos) = elementos\ del\ vector\ A$$
$$d(x) = a_1 * x_1 + a_2 * x_2 + a_3$$
$$= -2 * x_1 + 0 * x_2 + 1 = 0$$

9) Despejo las variables para graficar

$$-2 * x_1 + 0 * x_2 + 1 = 0$$
$$-2 * x_1 = -1$$
$$x_1 = 0{,}5$$



Así nos queda distinguidas ambas clases

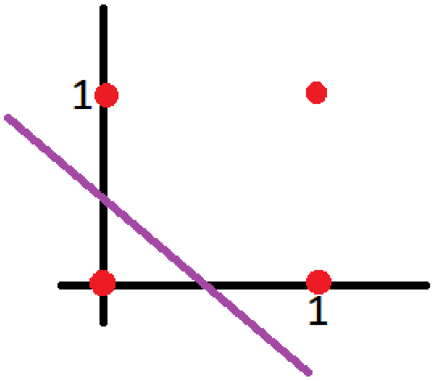
Redes neuronales

Perceptron

Explicación ejercicio 1

Dada una red perceptrón con comportamiento de una compuerta lógica OR se pide:

1) Sabiendo que las salidas son 0 y 1, graficar la solución (sin hacer cálculos)
   Se que las entradas posibles son 0 y 1
   Entonces teniendo las entradas (0 0) (0 1) (1 0) (1 1) se que la primera va a resultar 0 y el resto 1 así que puedo aproximar una partición de clases



2) Dado el siguiente cuadro de iteración de la red responder
   Esta es una iteración completa que es pasar cada entrada una vez
   El primer uno de cada entrada es lo que en reconocimiento de patrones era la columna del vector aumentado, en este tema toma el nombre de umbral, siempre es 1 y lo tomamos como una entrada más (va a tener la posición X_0)
   Cada entrada X_n va a tener un peso asociado W_n

(x,s)		Pesos	Salida	Ajuste de pesos	Error
Entrada	Salida				
1 0 0	0	-2,5 0,5 1,5	f (-2,5) = 0	$\begin{pmatrix} -2,5 \\ 0,5 \\ 1,5 \end{pmatrix} + (0) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -2.5 \\ 0,5 \\ 1,5 \end{pmatrix}$	0
1 0 1	1	-2,5 0,5 1,5	f (-1,0) = 0	$\begin{pmatrix} -2,5 \\ 0,5 \\ 1,5 \end{pmatrix} + (1) \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -1,5 \\ 0,5 \\ 2,5 \end{pmatrix}$	1
1 1 0	1	-1,5 0,5 2,5	f (-1,5) = 0	$\begin{pmatrix} -1,5 \\ 0,5 \\ 2,5 \end{pmatrix} + (1) \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -0,5 \\ 1,5 \\ 2,5 \end{pmatrix}$	1
1 1 1	1	-0,5 1,5 2,5	f (3,5) = 1	$\begin{pmatrix} -0,5 \\ 1,5 \\ 2,5 \end{pmatrix} + (0) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -0,5 \\ 1,5 \\ 2,5 \end{pmatrix}$	0

   a. Que representan los 3 elementos que se dan como entrada?
      El umbral y las otras son las entradas
   b. Porque se modifican los pesos en algunos casos?
      Porque hay error
   c. Como se obtiene la columna de la "Salida calculada"?
      Se aplica una función, no sé si siempre es $\sum X_n * W_n$ o puede variar

   Procedimiento de la iteración:
   Partimos de los pesos, hacemos la salida y luego calculamos un error
   La salida se calcula haciendo la sumatoria de cada peso por su entrada -> $\sum X_n * W_n$

Esa salida se evalúa a través de una función de trasferencia o activación que tiene forma de escalón y puede tomar los valores de 0 y 1 ó 1 y -1 dependiendo de cómo la fijemos.
Esa función escalón permite que la salida tenga dos valores extremos
Esa salida calculada (Columna 4) se compara con la salida deseada (Columna 2)
Con eso se calcula el **error -> salida deseada - salida calculada**
Si el error es cero, los pesos no se van a modificar y con esos pesos voy a pasar la otra entrada (Por esto las dos primeras filas de pesos son iguales y después cambian).
Importante: **En el ajuste de pesos** también se incluye el alfa (α) que viene como dato que es el factor de ganancia. Entonces quedaría:

$$\{\boldsymbol{Pesos\ actuales}\} + \boldsymbol{\alpha} * \boldsymbol{Error} * \{\boldsymbol{Entrada}\} = \{\boldsymbol{Pesos\ ajustados}\}$$

Si hubo, aunque sea un error, esa red no está estable, no aprendió, es necesario seguir iterando.
Se vuelve a calcular la salida y en base a esa salida el error.
Si hay error hay que ajustar los pesos, a los pesos actuales se le suma el producto del error por la entrada, esto se hace con todas las entradas solo que cuando no hubo error se está multiplicando por cero

Adaline

Diferencia entre perceptrón y adaline: La diferencia fundamental radica en la función de transferencia o de activación, en perceptrón tiene una forma de escalón mientras que en adaline toma la forma de una rampa porque permite que haya un margen de error, este es el error cuadrático aceptable

Ejercicio adaline: https://youtu.be/vUYazAzwOWM?t=2159

Pasos:

1) Fijar el error medio cuadrático aceptable
2) Fijar valor de alfa (α)
3) Asignar valores aleatorios a los pesos
4) Presentar vector de entrada
5) Obtener la salida lineal (función rampa) de la red:
$$\sum X_n * W_n$$
6) Calcular el error -> Salida esperada – Salida calculada
7) Actualizar pesos
8) Con los nuevos pesos probar la siguiente entrada

Es igual a perceptrón, lo único a tener en cuenta es que hay que guardar los errores para compararlos con el error medio cuadrático para definir si la red es estable o no. Esto se hace una vez que se terminaron las entradas, o sea luego de la iteración.

$$< \varepsilon^2 >= \frac{1}{2L} * \sum \varepsilon^2$$

Donde:

- L es la cantidad de entradas que tuvo la red
- ε son los errores de cada entrada

Modelos Bayesianos

Probabilidad condicional o a posteriori: Es la probabilidad de que ocurra un evento ***A*** dado que ocurre otro evento ***B*** (evidencia).

$$P(a|b) = \frac{P(a \wedge b)}{P(b)}$$

$$P(a \wedge b) = P(a|b) * P(b)$$

Regla de Bayes: Me permite calcular una probabilidad condicional cuando tenemos la probabilidad condicional en sentido contrario.

$$P(b|a) = \frac{P(a|b) * P(b)}{P(a)}$$

Distribución conjunta completa

	Evento 2			
Evento 1		**Positivo**	**Negativo**	**P(Evento 2)**
	Positivo	$\boldsymbol{E_1 \wedge E_2}$	$\boldsymbol{E_1 \wedge \overline{E_2}}$	$\boldsymbol{P(E_2)}$
	Negativo	$\boldsymbol{\overline{E_1} \wedge E_2}$	$\boldsymbol{\overline{E_1} \wedge \overline{E_2}}$	$\boldsymbol{\overline{P(E_2)}}$
	P(Evento 1)	$\boldsymbol{P(E_1)}$	$\boldsymbol{\overline{P(E_1)}}$	**1**

Independencia

Condición: $P(a|b) = P(a)$

Si a y b son independientes, $P(a \wedge b) = P(a) * P(b)$

La dependencia no implica causalidad, o sea que un evento sea la causa de otro

Independencia condicional

Condición: $P(a|b \wedge c) = P(a|c)$

Redes bayesianas
Estructuras de datos que representan las dependencias entre variables
Muestran una descripción compacta de cualquier distribución de probabilidad conjunta completa
Son grafos dirigidos y cada nodo contiene información probabilística
Especificación completa:

- Un conjunto de variables aleatorias forman los nodos
- Un conjunto de arcos dirigidos conectan pares de nodos
- Si hay un arco de X a Y, es padre de Y
- Cada nodo X_i tiene una distribución de probabilidad condicionada $P\left(X_i \middle| Padres(X_i)\right)$
- El grafo no tiene ciclos dirigidos, es un grafo acíclico dirigido (GAD)

Inferencia automática

- Exacta
  - Por enumeración

Por enumeración

- Una probabilidad conjunta se puede calcular como un producto de probabilidades condicionales

$$\mathbf{P}(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i|padres(X_i))$$

- Sumando las probabilidades conjuntas de los casos donde se cumplen las proposiciones se puede calcular la probabilidad buscada.

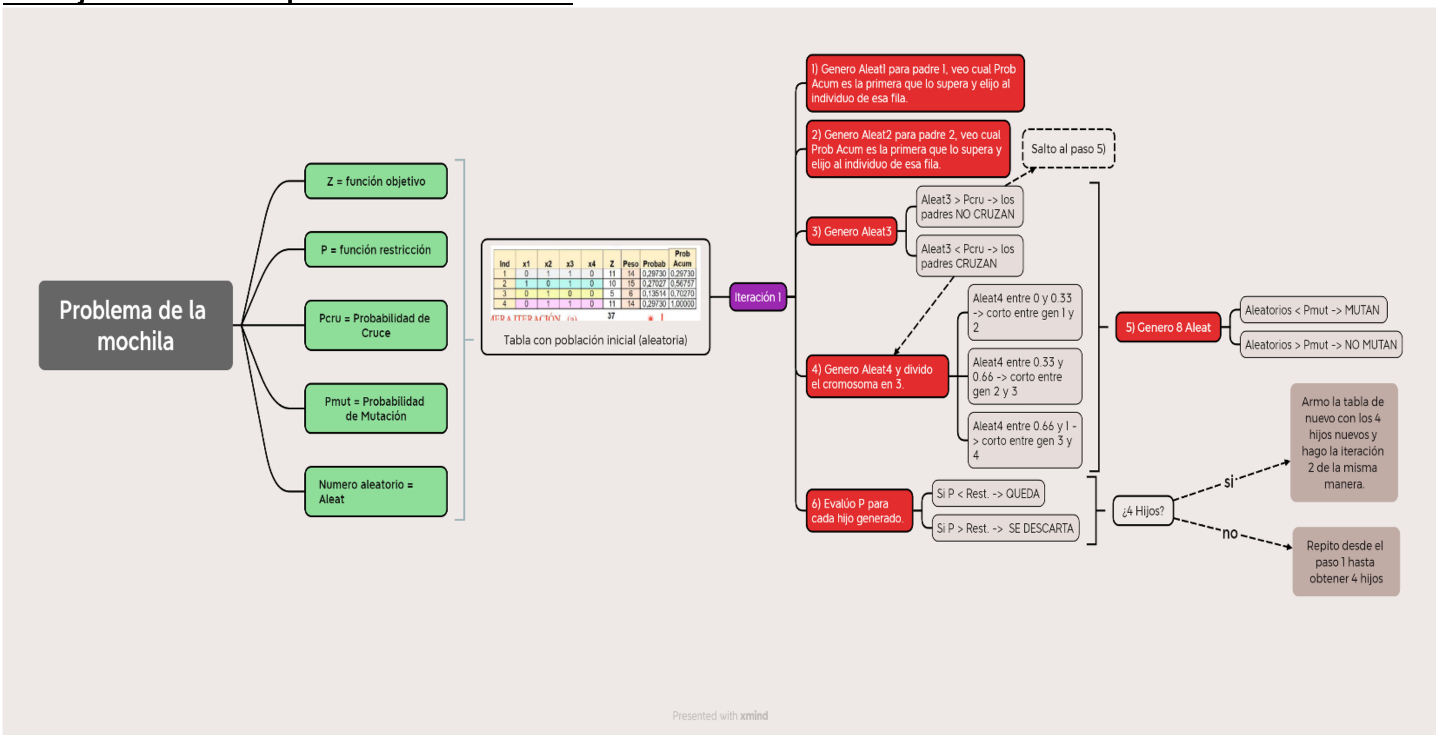
$$\mathbf{P}(X|\mathbf{e}) = \alpha\,\mathbf{P}(X,\mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X,\mathbf{e},\mathbf{y})$$

- Para el caso de nuestra pregunta del robo:

$$P(r|j,m) = \alpha \sum_{t^*\in\langle t,\neg t\rangle} \sum_{a^*\in\langle a,\neg a\rangle} P(r)P(t^*)P(a^*|r,t^*)P(j|a^*)P(m|a^*)$$

Metaheurísticas

Para ejercicios como el problema de la mochila



Si no nos dan las probabilidades de cruce y mutación -> AVERIGUAR

Para ejercicios donde nos dan una función f(x) el procedimiento sería igual que el anterior, pero f(x) reemplazaría a la función Z y no habría función P (no hay restricción). SEGÚN ENTIENDO VIENDO LA RESOLUCION DEL EJERCICIO 2