Trabajo práctico – Inteligencia Artificial (MCT5738)

Instrucciones de entrega

Entrega:
Vía correo electrónico.

Formato de entrega:
Enviar los archivos **.m** correspondientes a cada función creada en un archivo comprimido **cedula_nombre_apellido.zip** según corresponda.

Dirección de correo para entrega:
guidovalenzano@ieee.org

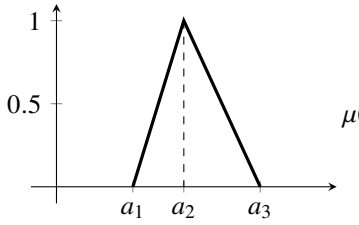
Asunto del correo electrónico:
«Trabajo Práctico - Inteligencia Artificial - Nombre Apellido», **sin** comillas.

Fecha de entrega:
Jueves 6 de abril de 2017, hasta las 11:59 horas.

Tema 1 (10 %)

Escribir una función `fuzzyTri` en lenguaje M que permita determinar el grado de pertenencia $\mu(x)$ de una variable x, suponiendo que la función de pertenencia es triangular con parámetros a_1, a_2, y a_3. Estos parámetros son pasados a la función en forma de **vector fila**.

```
p = [a1, a2, a3];
function mu = fuzzyTri(x, p)
        ⋮
        ⋮
end
```



$$\mu(x) = \begin{cases} 0 & x < a_1 \\ (x - a_1)/(a_2 - a_1) & a_1 \leq x \leq a_2 \\ (a_3 - x)/(a_3 - a_2) & a_2 \leq x \leq a_3 \\ 0 & x > a_3 \end{cases}$$

Tema 2 (10 %)

Escribir una función `fuzzyInv` en lenguaje M que calcule la inversa del tema anterior. Es decir, dado un grado de pertenencia $\mu(x) > 0$ y un **vector fila** $p = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}$, devuelva un conjunto de valores u, v tales que $\mu(u) = \mu(v) = \mu(x)$.

```
function [u, v] = fuzzyInv(mu, p)
        ⋮
end
```

Tema 3 (10 %)

Escribir una función `fuzzyInput` en lenguaje M que permita fuzzificar la variable de entrada x y devuelva un **vector fila** $\tilde{x}_f$ con los grados de pertenencia correspondiente a cada valor difuso $A, B, \ldots, N$ en s. En la matriz s, cada **fila** corresponde a los elementos a_1, a_2 y a_3 de una función triangular que caracteriza al valor difuso. **Observación:** utilizar la función del Tema 1.

```
function xf = fuzzyInput(x, s)
        ⋮
        ⋮
end
```

$$s = \begin{bmatrix} a_1^A & a_2^A & a_3^A \\ a_1^B & a_2^B & a_3^B \\ \vdots & \vdots & \vdots \\ a_1^N & a_2^N & a_3^N \end{bmatrix} \qquad \tilde{x}_f = \begin{bmatrix} \mu_A(x) & \mu_B(x) & \ldots & \mu_N(x) \end{bmatrix}$$

Tema 4 (10 %)

Escribir una función `defuzzy` en lenguaje M que permita defuzzificar una salida difusa $\tilde{z}_f$ utilizando el método de la altura (también llamado promedio ponderado). La función recibe un **vector fila** $\tilde{z}_f$ donde cada elemento representa el grado de pertenencia correspondiente a cada valor difuso $A, B, \ldots, N$ en s. En la matriz s, cada **fila** corresponde a los elementos a_1, a_2 y a_3 de una función triangular que caracteriza al valor difuso. **Observación:** en el método de la altura es posible que el denominador se haga cero. Utilizar una sentencia **if-else** para que $\tilde{z}_f = 0$ en dicho caso.

```
function z = fuzzyInput(zf, s)
        ⋮
        ⋮
end
```

$$\tilde{z}_f = \begin{bmatrix} \mu_A(z) & \mu_B(z) & \ldots & \mu_N(z) \end{bmatrix} \qquad s = \begin{bmatrix} a_1^A & a_2^A & a_3^A \\ a_1^B & a_2^B & a_3^B \\ \vdots & \vdots & \vdots \\ a_1^N & a_2^N & a_3^N \end{bmatrix}$$

Tema 5 (60 %)

Dados dos **vectores fila** $\tilde{x}_f$ y $\tilde{y}_f$ correspondientes a entradas difusas x e y. Asuma que las mismas cuentan con el **mismo número** de valores difusos, es decir:

$$\begin{aligned} \tilde{x}_f &= \begin{bmatrix} \mu_A(x) & \mu_B(x) & \mu_C(x) & \dots & \mu_N(x) \end{bmatrix} \\ \tilde{y}_f &= \begin{bmatrix} \mu_{\mathcal{A}}(x) & \mu_{\mathcal{B}}(x) & \mu_C(x) & \dots & \mu_N(x) \end{bmatrix} \end{aligned} \qquad \text{length}(\tilde{x}_f) = \text{length}(\tilde{y}_f)$$

Dada además una matriz r correspondiente a las reglas, de forma que $\tilde{x}$ varía en las columnas e $\tilde{y}$ varía en las filas, y donde cada valor en $r(i, j) = r(\tilde{y}_f(i), \tilde{x}_f(j))$ está vinculado a un valor difuso correspondiente a la salida difusa $\tilde{z}_f = \begin{bmatrix} \mu_1(z) & \mu_2(z) & \mu_3(z) & \dots & \mu_n(z) \end{bmatrix}$. Como por ejemplo:

$$r = \begin{bmatrix} 1 & 1 & 2 & 3 & 4 \\ 1 & 1 & 3 & 4 & 4 \\ 1 & 2 & 3 & 4 & 5 \\ 2 & 2 & 3 & 5 & 5 \\ 2 & 3 & 4 & 5 & 5 \end{bmatrix}$$

Escribir una función `fuzzyInference` que, utilizando el método de inferencia Mamdani y los operadores **max** y **min**, determine las reglas activadas y devuelva un **vector fila** correspondiente a los grados de pertenencia en $\tilde{z}_f$.

Observación: las funciones **max**, **min**, **unique**, **find** y **meshgrid** pueden llegar a ser útiles.

```
function zf = fuzzyInference(xf, yf, r)
        ⋮
        ⋮
end
```