

# Red Neuronal

August 7, 2022

La forma de la red neuronal hacia delante es la siguiente. Sea  $x$  una posición  $(X, Y)$  de la punta del robot y sea  $y$  un vector en  $R^8$  con la posición de los motores impares. Lo primero que necesitas implementar es el rescalamiento de  $x$  a  $x_t$ , que es una transformación lineal para que la nueva  $x_t \in [-1, 1]$ :

$$x_t = 2(x - x_{min}) / (x_{max} - x_{min}) - 1 \quad (1)$$

donde  $x_{min}$  y  $x_{max}$  te los doy yo ya en un txt y  $.$  es la división elemento a elemento.

Ahora viene la red neuronal. Es una red neuronal de tipo feed-forward con una capa escondida para un problema de regresión, así que su fórmula es esta:

$$h = g(W_1 x_t + b_1) \quad (2)$$

$$y_t = W_2 h + b_2 \quad (3)$$

dónde  $h$  es un vector que representa el estado de la capa escondida ( $h$  de hidden en inglés),  $W_1$  y  $W_2$  son matrices que te doy yo en un archivo (son las matrices de pesos, se llaman  $W$  por weight en inglés),  $b_1$  y  $b_2$  son vectores que yo te doy (son los sesgos o las ordenadas en el origen de las regresiones lineales, digamos, se llaman  $b$  por bias en inglés) y  $g$  es la función de activación. He probado con varias y la que mejor funciona (no es nada sorprendente) es la función ReLU (Rectified linear unit):

$$g(t) = \max(t, 0), \quad (4)$$

es decir, es la función  $y = x$  cuando en el eje positivo y es la función  $y = 0$  en el eje negativo. Ten en cuenta que es una función elemento a elemento sobre un vector, porque  $W_1 x_t + b_1$  es un vector. Matlab tiene ReLUs en algún sitio, pero si no funciona, para implementártela tú quizá lo más fácil (y eficiente, podrías hacer un loop, pero sería más lento) sea hacer concat del vector  $t$  con un vector de ceros para obtener una matriz y luego hacer max en el eje de las columnas.

Vale, la red neuronal te da  $y_t$ , que son las posiciones de los motores impares reescaladas a  $[-1, 1]$ . Lo primero es reescalarlas de vuelta a la realidad. Eso lo puedes hacer usando:

$$y = \frac{y_t + 1}{2} * (y_{max} - y_{min}) + y_{min} \quad (5)$$

si no me he equivocado al invertir la fórmula que hemos usado para las  $x$  (que es la misma pero al revés). Las  $y_{min}$  y las  $y_{max}$  te las doy yo también en un archivo.

Y ya lo único que te queda es obtener los motores pares a partir de los impares.