Matilde Santos Peñas
Facultad de Informática
Universidad Complutense de Madrid

Asignatura: Inteligencia Artificial aplicada al Control

Prácticas del Tema 5:
NEURO CONTROL

PRÁCTICA 1:
a) Diseñar una red neuronal mediante la herramienta *nntool* de Matlab/Simulink que realice la función XOR. Observar la influencia de los parámetros de la red sobre la convergencia del error. Probar diversas estrategias de aprendizaje.
Nota: esta práctica no se evalúa, pero es necesaria para aprender a manejar la herramienta, sobre todo aprender a introducir los datos y algunos conceptos fundamentales.

PRÁCTICA 2:
En esta práctica vamos a hacer uso de un conjunto de redes neuronales para mover un brazo robot. Emplearemos el modelo de brazo que se muestra en la figura 1 como plataforma para obtener datos del movimiento del brazo, y entrenar y validar las redes construidas. Dicho brazo posee cinco articulaciones:

1. Base: El brazo puede rotar en torno a su base.
2. Hombro: permite rotar verticalmente el brazo entero.
3. Codo: permite rotar verticalmente todas las piezas del brazo desde dicha articulación en adelante.
4. Muñeca, articulación vertical: permite rotar la pinza verticalmente.
5. Muñeca articulación horizontal: permite rotar la pinza a derecha e izquierda.

Todas las articulaciones se han simulado sin topes, lo que quiere decir que pueden girar libremente 360º. Además el modelo no tiene en cuenta las colisiones de sus partes móviles entre sí. Por tanto, el modelo puede realizar movimientos que serían imposibles para un brazo real.

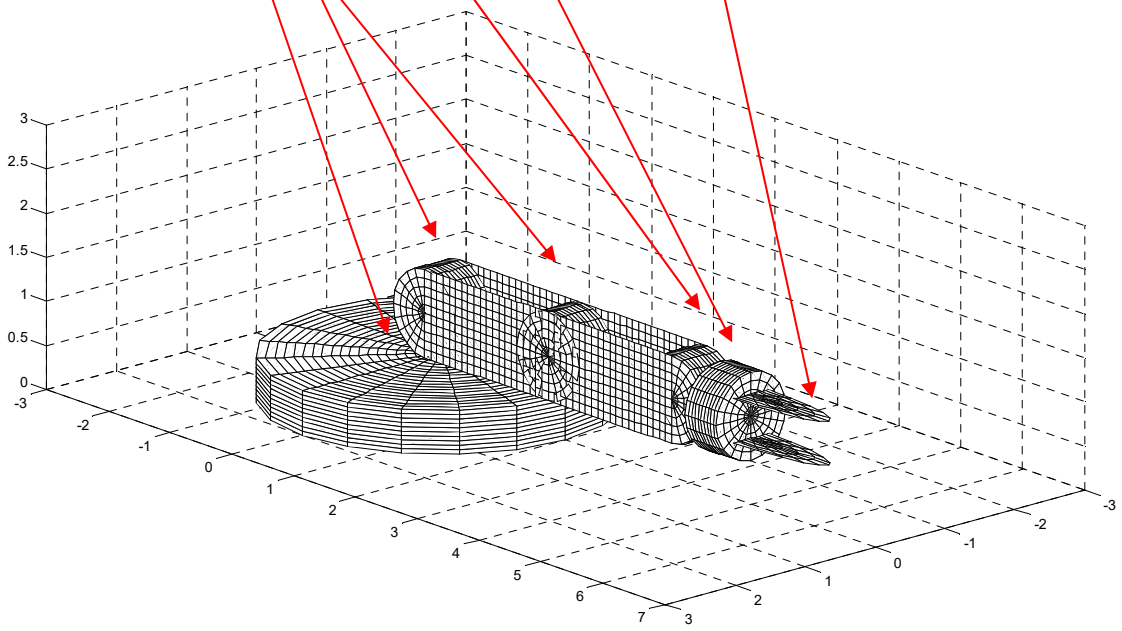
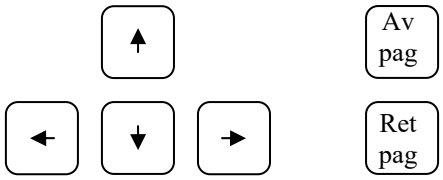


Figura 1. Esquema del brazo robot.

El brazo esta simulado en Matlab. Para utilizar el modelo basta escribir en la línea de comandos de Matlab: *brazo*, y se abrirá una ventana como la de la figura 2. A esta ventana se le han añadido algunos menús desplegables que permiten trabajar más fácilmente con el robot. El uso de dichos menús, así como la forma de mover el robot, se describe a continuación:

1. Manipulación manual del brazo. El brazo puede manejarse empleado el teclado del ordenador; para ello hace falta que la ventana del modelo sea la ventana activa. Las teclas que permiten manipular el brazo son:



   - Las dos flechas horizontales permiten girar el brazo respecto a la base.
   - Las flechas verticales mueven la articulación (hombro, codo, muñeca vertical, muñeca horizontal) que indique la leyenda situada encima de la imagen del brazo.
   - Las teclas de avance y retroceso permiten cambiar la articulación activa; activan un menú circular en el que se va pasando sucesivamente de una articulación a la siguiente, y cuando se han recorrido las cincos posibilidades, vuelve a empezar. Avance página recorre el menú en un sentido y retroceso página lo recorre en el sentido contrario.

   Cada vez que se pulsa una de las flechas, el brazo gira un ángulo equivalente a un grado, en la dirección correspondiente.

2. Uso de la barra de herramientas. A la barra de herramientas (Figura 2, menú barra superior) se le han añadido tres entradas nuevas: 'datos', 'red' y 'reset datos'. Cada vez que se pulsa una de las flechas, el modelo guarda en una variable interna, denominada datos, el valor de la articulación rotada, así como la nueva posición del extremo del efector (pinza del robot). La variable datos es una estructura con dos matrices de datos:

   a. Datos.efector: contiene la posición del efector (x,y,z), y tantas filas como movimientos sucesivos se hayan realizado.

   b. Datos.angulo. contiene una matriz de cuatro columnas. (b,h,c,mv). El primer elemento, b, contiene el ángulo 1,0 ó -1, que se ha rotado la base desde su posición anterior. El segundo, h, el ángulo en que se ha rotado el hombro; el tercer elemento, c, el ángulo en que se ha rotado el codo, y el cuarto, mv, el ángulo en que se ha rotado la muñeca en la dirección vertical. La rotación horizontal de la muñeca no se guarda, porque no afecta a la posición en el espacio de efector. De nuevo esta matriz guarda todos los movimientos realizados.

   Mediante la entrada 'datos' de la barra de herramientas se pueden guardar los datos capturados después de mover el brazo. También se pueden exportar al workspace y resetear los valores guardados para capturar una nueva secuencia de movimientos.
   La entrada 'red', permite importar redes neuronales, diseñadas y entrenadas con la herramienta 'nntool' de Matlab, para mover automáticamente el brazo desde una posición inicial a una posición final, siguiendo la trayectoria 'aprendida' por las redes. El modelo del brazo robot trabaja con cuatro redes neuronales: una para mover la base,

otra para el hombro, una tercera para el codo, y una cuarta para el movimiento vertical de la muñeca.

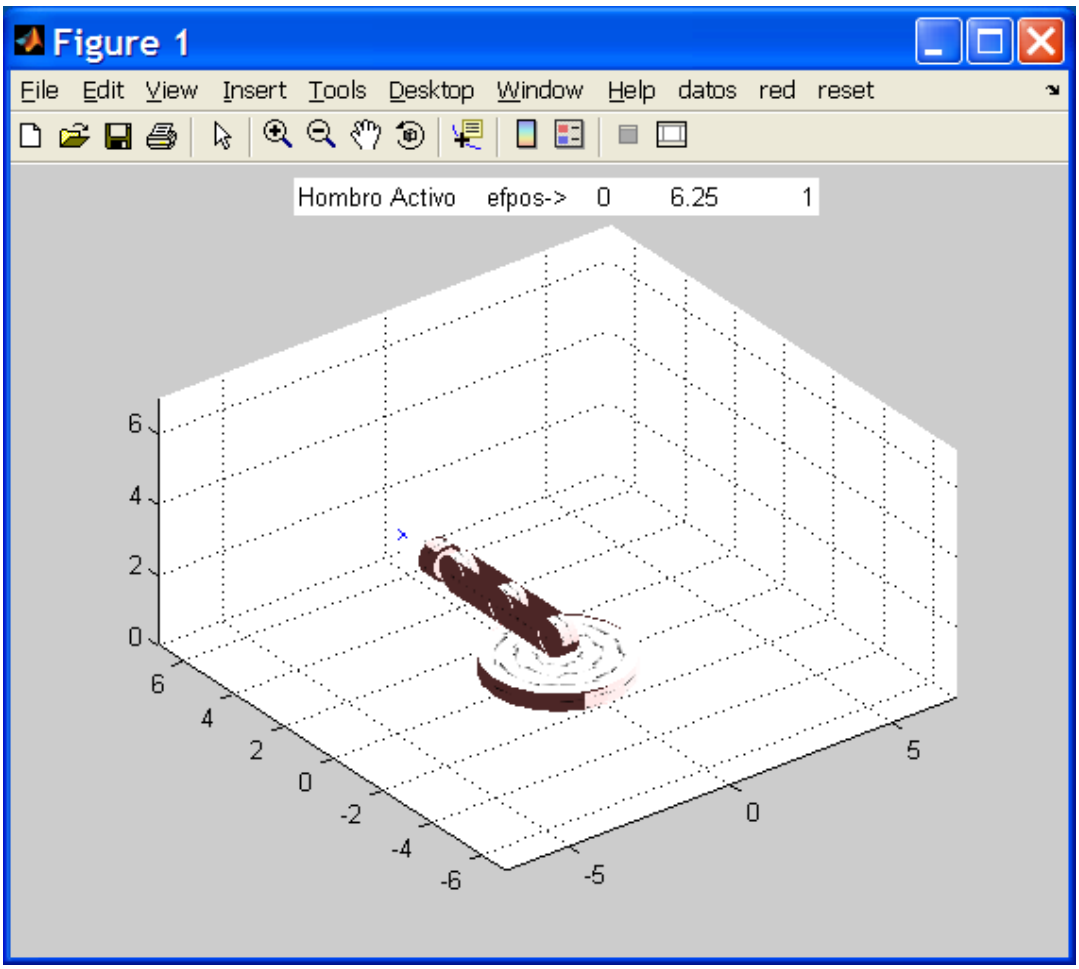


Figura 2. Vista de la ventana del modelo.

La entrada 'red' de la barra de herramientas también permite fijar las posiciones iniciales y finales de la pinza, para su movimiento automático. En el menú de red se puede establecer la 'tolerancia' con la que las redes considerarán que la pinza del brazo robot ha alcanzado su posición final. Por último 'red' permite arrancar el movimiento automático de la red.
La entrada 'reset' lleva el robot a su posición de reposo.

Encima del gráfico del robot, en la figura 2, puede verse una leyenda en la que se indica la articulación actualmente activa del brazo y la posición actual de la pinza (x,y,z).

Práctica.

1. Mover manualmente el robot desde la posición de reposo (la posición en la que aparece por defecto) hasta otra posición final. Al hacerlo, procurar emplear movimientos simples y continuos, por ejemplo, levantar el hombro unos 45º, bajar el codo hasta volver a poner la pinza horizontal, rotar la muñeca hacia abajo otros 45º, girar sobre la base 90º, deshacer los primeros los tres primeros movimientos, hasta volver a poner el brazo horizontal estirado, etc. Evitar mezclar movimientos de las articulaciones o hacerlas avanzar un paso hacia adelante y e inmediatamente otro hacia atrás (esto puede complicar mucho el aprendizaje de la red).

Nota: en la captura de los datos no conviene mantener pulsadas las teclas de forma continua, sino hacerlo movimiento a movimiento, una a una, para evitar posibles errores con los datos de entrenamiento.

2. Guardar los datos capturados en un archivo y exportarlos al workspace.

3. Emplear un conjunto de cuatro redes neuronales para controlar el movimiento del brazo. El esquema del sistema, con la redes como controlador, tendría la estructura mostrada en la figura 3.

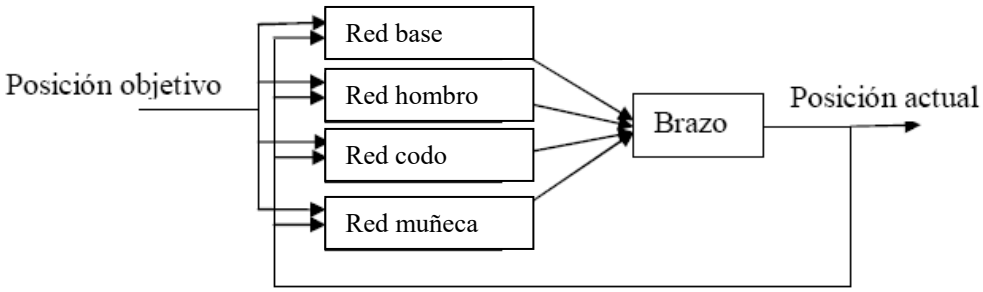


Figura 3. Esquema del sistema de control.

Para diseñar las redes empleamos la herramienta de Matlab *nntool*. Creamos cuatro redes distintas del tipo perceptrones multicapa con dos capas ocultas y funciones de transferencia purelin para la última capa, y las funciones de activación que se quiera para las demás capas. Se usa como conjunto de entrenamiento los datos capturados en los apartados 1 y 2.

A cada una de las redes se le tiene que proporcionar, como datos de entrada para el entrenamiento, las séxtuplas (X,Y,Z,u,v,w), donde (X,Y,Z) son las posiciones de los movimientos efectuados previamente con el brazo robot que se guardan en datos.efector, y (u,v,w) es la posición del último movimiento efectuado. Para ello, **se debe eliminar dicha posición de la tabla datos.efector**.

Como datos de salida hay que proporcionar a cada red neuronal los ángulos de base, hombro, codo y muñeca vertical, respectivamente; es decir, el valor del ángulo obtenido para cada articulación en cada movimiento. Así para entrenar la red que moverá el robot en torno a la base usaremos como salida los datos de la primera columna de datos.angulo; para el hombro, los de las segunda, etc.

Es conveniente usar la **función de transposición (operador ')** para convertir vectores columna en vectores fila, tanto para la entrada como para cada una de las salidas, a la hora de introducirlas en las redes neuronales.

Es posible conseguir buenos resultados empleando redes que tengan entre 10 y 25 neuronas en las capas ocultas. La base suele necesitar menos neuronas y la muñeca más (deducir cuál es la razón de que ocurra esto).

Como el movimiento del brazo supone que se mueva cada vez una articulación un grado (1 ó -1), la red estará ajustada cuando el error en el conjunto de entrenamiento sea del orden de 10^{-3} o menor, en valor absoluto, para cada uno de las salidas del conjunto de entrenamiento. Esto se puede comprobar sin más que exportar los errores de entrenamiento desde la herramienta nntool al workspace, y chequear su valor (se puede hacer una gráfica para mostrar la evolución del

error). Si las redes están bien entrenadas deberían ser capaces de llevar el brazo desde cualquiera de la posiciones de la trayectoria aprendida hasta la posición final; comprobarlo.

Para aplicar las redes entrenadas al robot basta exportarlas desde nntool al workspace, e importarlas al brazo robot.

4. Una vez entrenadas las redes, probar a partir de una posición inicial distinta que forme parte del recorrido. Probar también con una posición final distinta. ¿Qué ocurre? Explicar por qué.