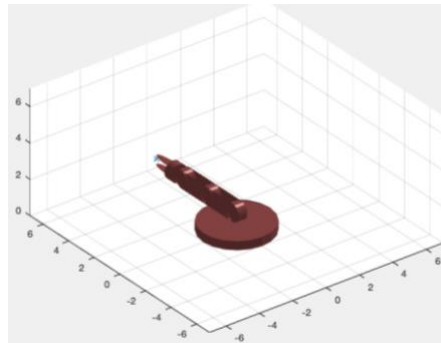




Universidad Complutense de Madrid.
Facultad de Ingeniería Informática.
Inteligencia Artificial Aplicada al Control.



Práctica 5

Redes Neuronales

Alumno:

Borges Noronha, Frederick Ernesto

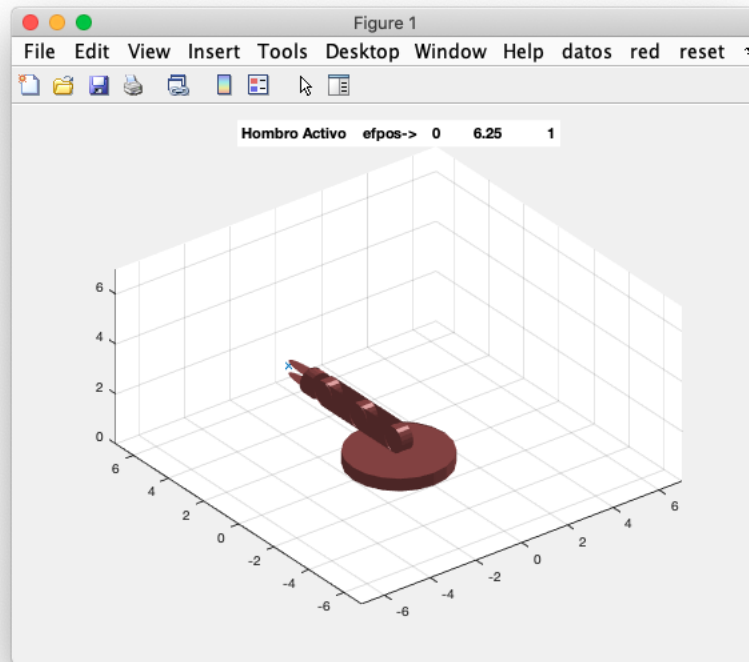
Tabla de contenido

<i>Pasos para reproducir la práctica</i>	<i>3</i>
Generar los datos.....	3
Crear la red.....	4
Configuración de las redes.....	9
Red Neuronal para el base.	9
Red Neuronal para el hombro.	10
Red Neuronal para el codo.	10
Red Neuronal para la muñeca vertical.....	11
Importar la red en el brazo robot.....	11
Probar el modelo	12
<i>Análisis de resultados</i>	<i>12</i>

Pasos para reproducir la práctica

Generar los datos.

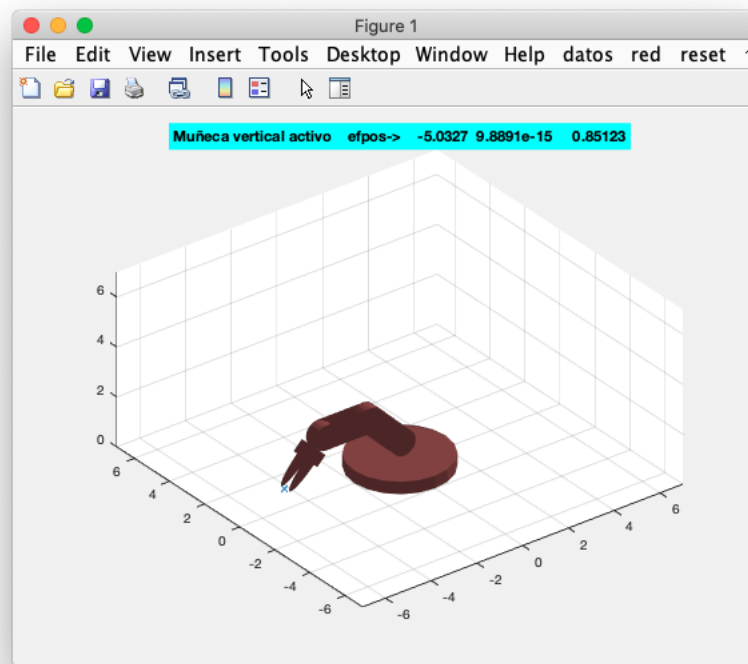
En la línea de comandos del MATLAB ejecutar el comando **brazo**, esto abrirá la siguiente ventana, donde podremos generar los datos para entrenar la red:



Ahora debemos llevar el brazo robot a la posición indicada en la práctica para lo cual presionaremos:

1. **Levantar el hombro 45°:** Presionar 45 veces la tecla `UP`
2. **Cambiar el selector para mover el codo:** Presionar la tecla `RE PAG`
3. **Bajar el codo hasta que este en posición horizontal:** Presionar 45 veces la tecla `DOWN`
4. **Cambiar el selector para mover la muñeca:** Presionar la tecla `RE PAG`
5. **Rotar la muñeca vertical 45° hacia abajo:** Presionar 45 veces la tecla `DOWN`
6. **Girar sobre la base 90°:** Presionar 90 veces la tecla `LEFT`

Luego de seguir estos pasos el brazo debería estar como en la siguiente figura:



Luego deshacemos algunos movimientos:

1. **Rotar la muñeca vertical 45° hacia arriba:** Presionar 45 veces la tecla `UP`
2. **Cambiar el selector para mover el codo:** Presionar la tecla `AV PAG`
3. **Subir el codo 45°:** Presionar 45 veces la tecla `UP`
4. **Cambiar el selector para mover el hombro:** Presionar la tecla `AV PAG`
5. **Bajar el hombro 45°:** Presionar 45 veces la tecla `DOWN`

Luego procedemos a guardar los datos haciendo clic en `datos` -> guardar datos` y luego los exportamos al workspace haciendo clic en `datos` -> exportar datos al workspace`.

Crear la red.

Para crear la red mostrada anteriormente hay que seguir los siguientes pasos:

1. En los datos exportados se tiene una tupla (X,Y,Z) en `datos.efector`. Una vez dicho esto ejecutamos el siguiente código para obtener los datos de entrada:

```
1. input = datos.efector
2. % Tomamos los valores finales
3. u = input(361,1)
4. v = input(361,2)
5. w = input(361,3)
6. % Creamos 3 columnas con los valores anteriores
7. u_column = ones(360,1) * u
8. v_column = ones(360,1) * v
9. w_column = ones(360,1) * w
10. input(361,:) = [] % Eliminamos la última fila
11. input = [input u_column v_column w_column]
```

```
12. input = input.' % Transponemos la matriz
13.
```

2. En **datos.angulo** se tienen las salidas obtenidas por cada uno de los movimientos según la siguiente tabla:

Base	Hombro	Codo	Muñeca Vertical
0	0	0	0

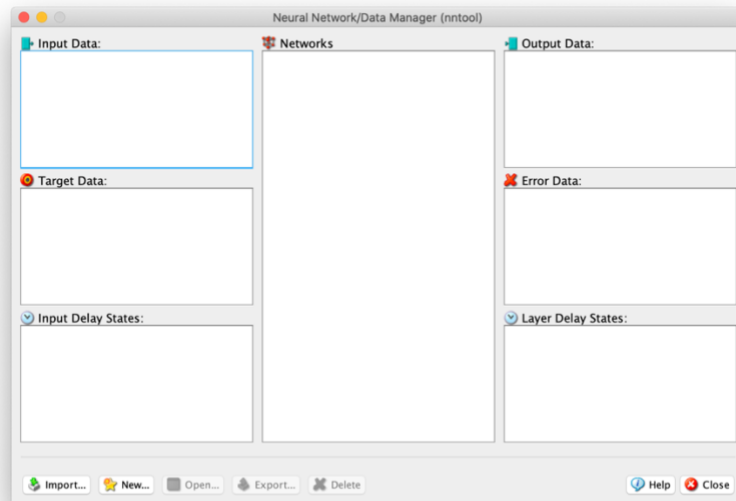
Sabiendo esto, procedemos a crear las variables de salida de la siguiente manera:

```
1. output_base = datos.angulo(:,1,:).'
2. output_hombro = datos.angulo(:,2,:).'
3. output_codo = datos.angulo(:,3,:).'
4. output_muneca = datos.angulo(:,4,:).'
```

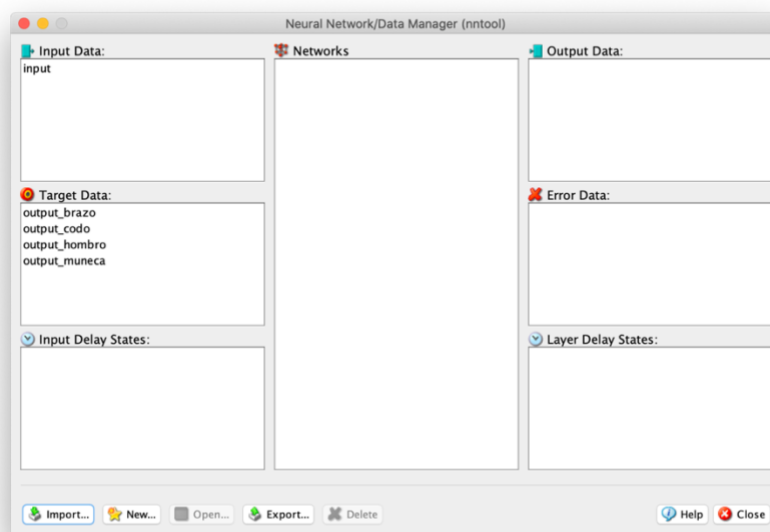
3. Antes de ir a la herramienta `nntool` vamos a expandir nuestros vectores de entrada y salida para obtener mejores resultados y para eso utilizamos el siguiente código:

```
1. input = [input input input input]
2. output_base = [output_base output_base output_base output_base]
3. output_hombro = [output_hombro output_hombro output_hombro
output_hombro]
4. output_codo = [output_codo output_codo output_codo output_codo]
5. output_muneca = [output_muneca output_muneca output_muneca
output_muneca]
```

4. Abrir la herramienta `nntool` ejecutando en la línea de comandos el siguiente comando **nntool**. Ahora deberíamos tener abierta una ventana como esta:



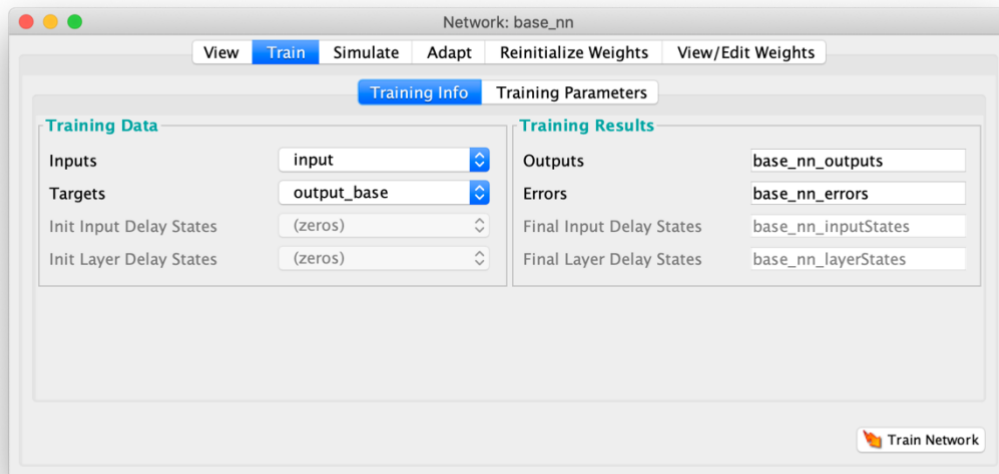
5. Importamos los datos haciendo clic en `Import` y procedemos a importar la variable `input` como `Input Data`, las variables de output como `Target Data`, luego hacemos clic en `Close` y deberíamos ver ahora una pantalla como esta:



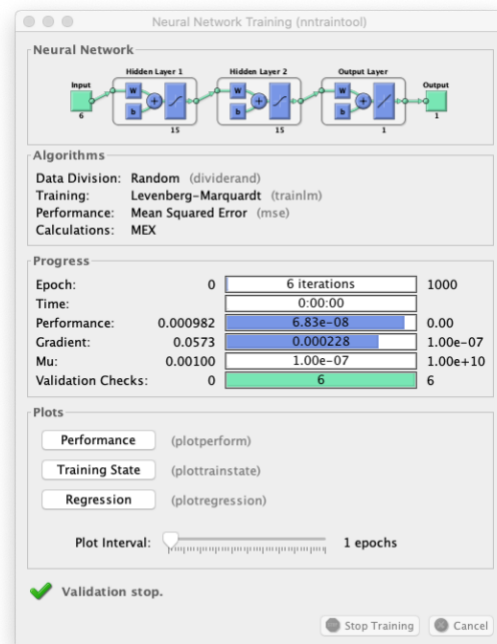
6. Hacemos clic en `New...` y procedemos a crear las redes **nn_base**, **nn_codo**, **nn_hombro**, **nn_muneca**.

Nota: se pueden poner entre 10 y 25 Neuronas. La base suele necesitar menos neuronas y la muñeca más.

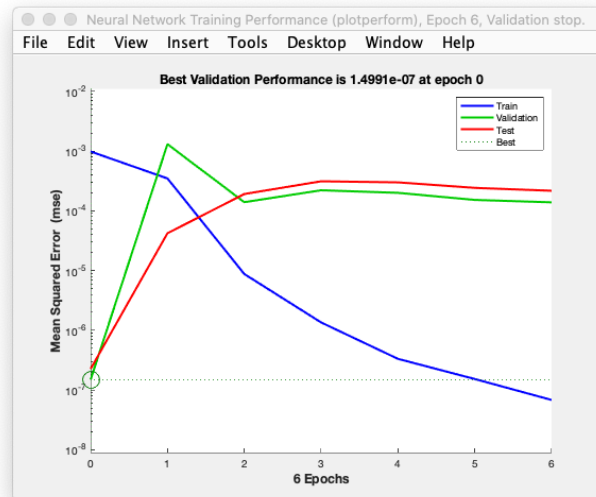
7. Ahora vamos a entrenar nuestras redes, hacemos doble clic en una red y hacemos clic en la pestaña de `train` y ponemos el `Inputs` y el `Targets` según la red que estemos entrenando:



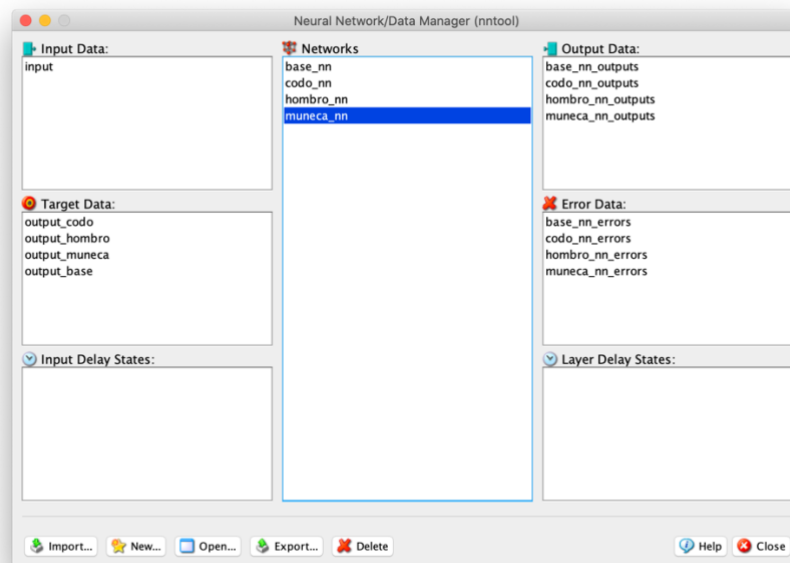
Luego hacemos clic en `Train Network` y se nos abrirá una ventana con los resultados del entrenamiento:



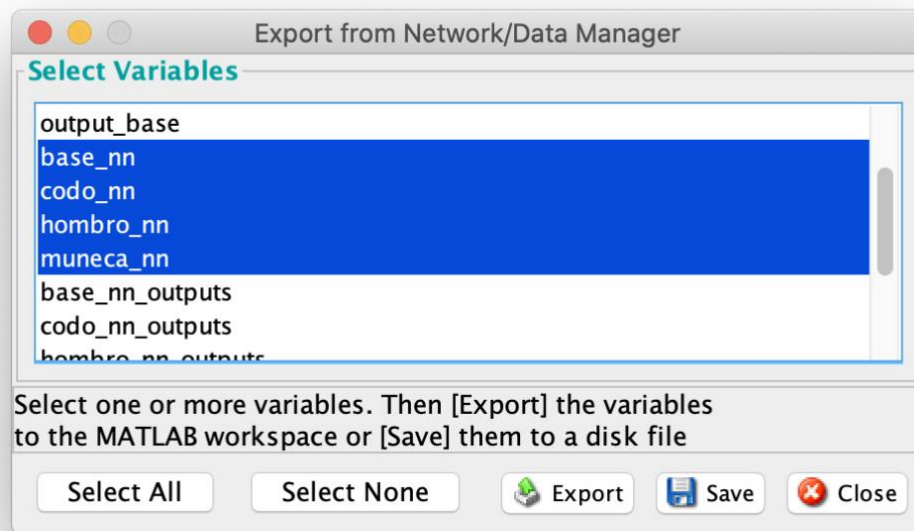
Para este entrenamiento a mi me ha resultado en el siguiente `Performance`:



8. Ahora vamos a la pestaña de `View/Edit Weights` y guardamos los pesos haciendo clic en `Set Weights`.
9. Una vez hecho el entrenamiento y guardados los pesos para todas las redes procedemos a exportar dichas redes al workspace, deberíamos tener el siguiente resultado:

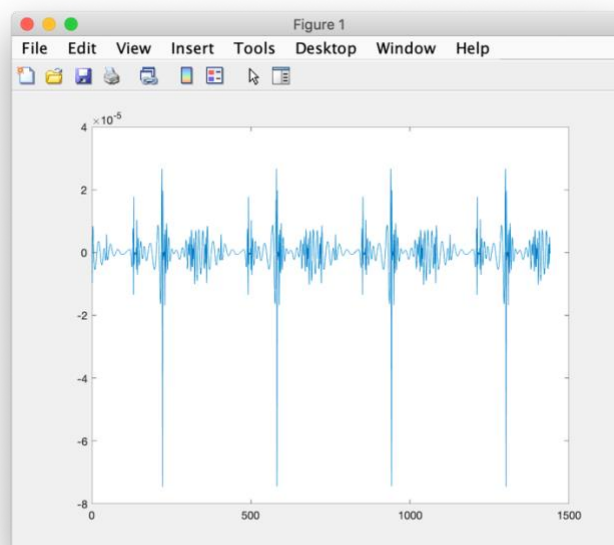
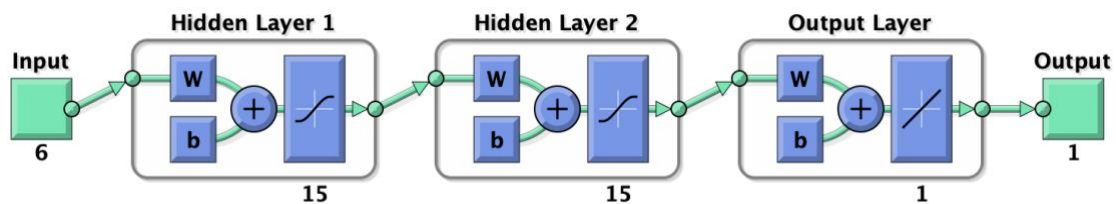


Hacemos clic en `Export...` y seleccionamos las redes y luego hacemos clic en `Export`.

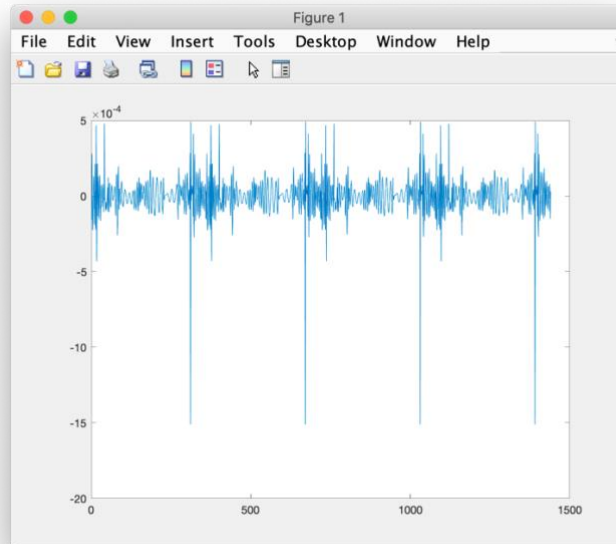
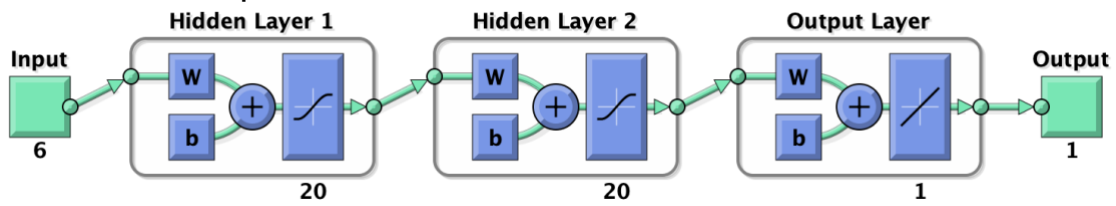


Configuración de las redes

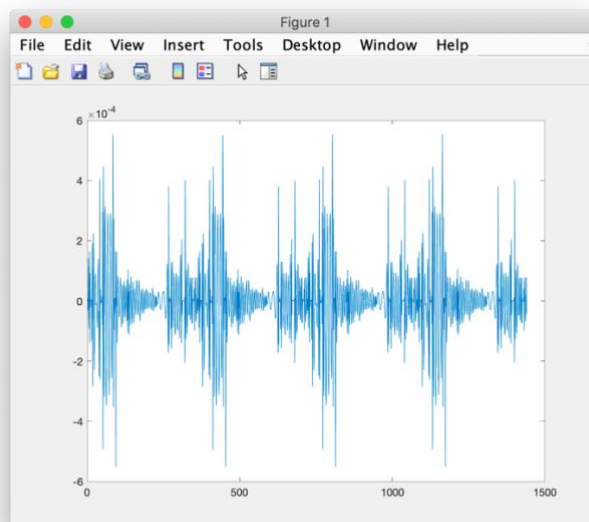
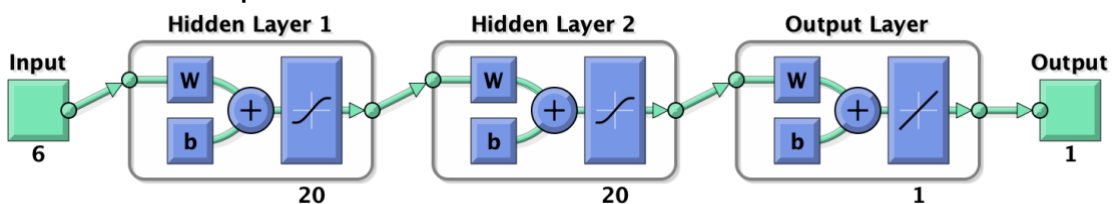
Red Neuronal para el base.



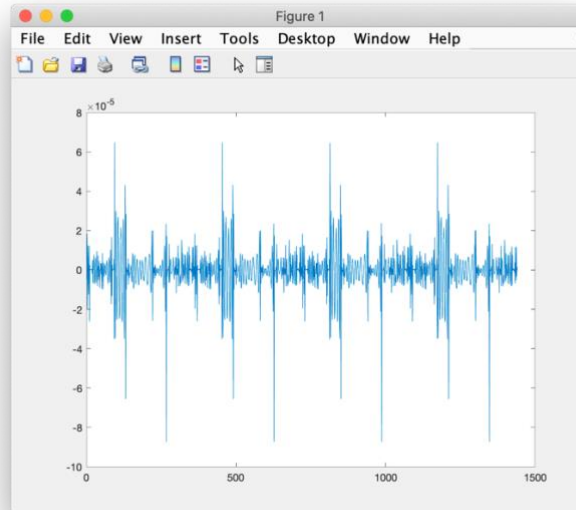
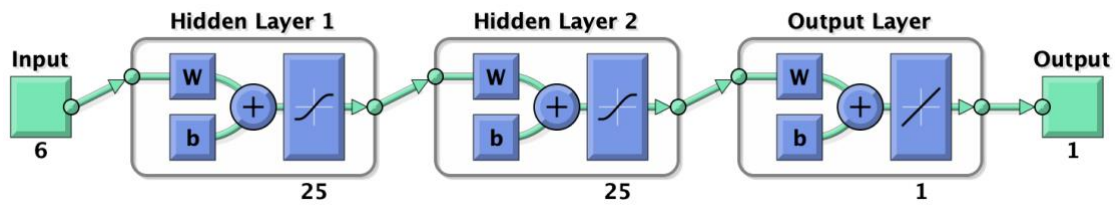
Red Neuronal para el hombro.



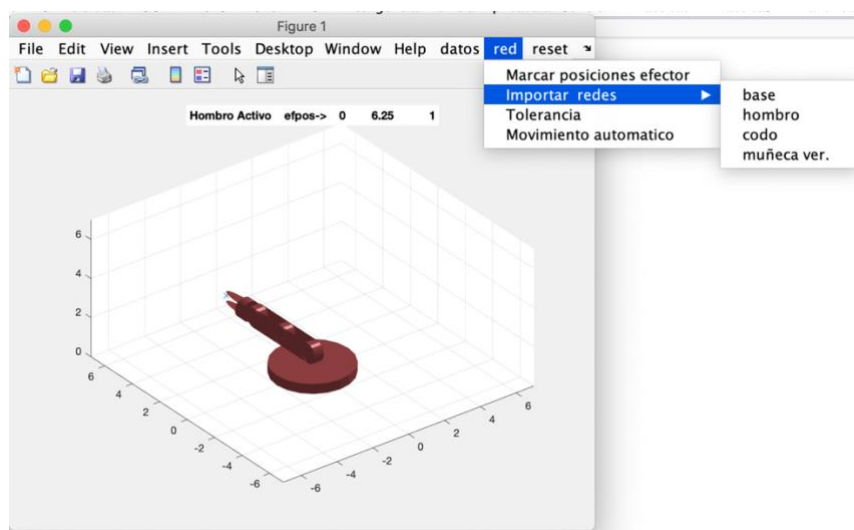
Red Neuronal para el codo.



Red Neuronal para la muñeca vertical.

*Importar la red en el brazo robot*

Para llevar a cabo este paso solo hace falta ejecutar el comando `brazo` y luego en la pestaña de `red` y luego `Importar redes` y allí importamos una a una cada una de las redes creadas.



Probar el modelo

En la misma ventana del brazo robot luego de importar las redes hacemos clic en `Movimiento Automático` y nos pedirá la posición inicial y final. Al hacer clic en ok deberíamos ver como el brazo robot se mueve por su cuenta hasta la posición final.

Análisis de resultados

Con lo que se ha hecho en esta practica podemos llegar a una conclusión importante, la cual es que esta red no ha aprendido, ha memorizado, ya que si se intenta ir a una posición que no estuviese en los movimientos de entrenamiento entonces el brazo robot no sabrá que hacer. Para solucionar este problema se deben realizar dos acciones, la primera es tener un conjunto de datos mas grande para hacer el entrenamiento y la segunda es dividir los datos en conjuntos de entrenamiento y pruebas.