

XVI

APLICACIONES DE LAS REDES NEURONALES EN ROBÓTICA

Carme Torras

CSIC-Universidad Politécnica de Cataluña

1. INTRODUCCIÓN

La programación de robots suele realizarse en términos de coordenadas cartesianas del espacio de trabajo para comodidad del usuario, recayendo en el controlador la tarea de traducir dicha especificación a variables articulares o motoras, que son las que gobiernan los movimientos del robot. Algunos lenguajes de programación avanzados contemplan incluso la posibilidad de especificar ciertos movimientos del robot en términos de variables sensoriales (por ejemplo, moverse en una determinada dirección hasta que la lectura del sensor de fuerza supere un determinado valor). Por consiguiente, el control de robots depende críticamente de la disponibilidad de funciones que permitan pasar del espacio físico (o del espacio sensorial) al espacio de variables articulares o motoras.

Una función básica de este tipo es pues la que relaciona las coordenadas cartesianas del espacio de trabajo con las coordenadas articulares de un robot manipulador. Esta función se denomina *cinemática inversa*, porque la función natural (directa) es la que a cada vector de valores de las variables articulares le hace corresponder la posición y orientación del elemento terminal (garra, herramienta) del robot en la configuración definida por dicho vector. Otra función relaciona la trayectoria del elemento terminal con las fuerzas y pares ejercidos en las distintas articulaciones. Esta función se denomina *dinámica inversa* por la misma razón anterior. El denominado *mapa sensoriomotor*, que relaciona patrones sensoriales con las pertinentes órdenes motoras, es una función genérica de este tipo que engloba varios casos particulares, como los requeridos en el servocontrol visual,

las operaciones de inserción y la evitación de obstáculos por parte de robots móviles.

El problema es que estas funciones suelen ser altamente no lineales y resulta difícil, cuando no imposible, derivarlas analíticamente. Además, debido a cambios en el entorno y al desgaste del propio robot, estas funciones pueden ser variantes en el tiempo y entonces es deseable que la estructura de control se adapte a las variaciones.

No hace falta decir que muy pocas de estas funciones quedan caracterizadas por su relación de entrada-salida, dependiendo la mayoría de variables de estado (o de algunos valores previos de las entradas) como sucede habitualmente en los sistemas dinámicos.

Desde un punto de vista computacional, las redes neuronales no son más que procedimientos generales para el aprendizaje de funciones no lineales. En este sentido, ofrecen una buena respuesta a la necesidad expresada anteriormente de esquemas adaptativos para codificar este tipo de funciones. Además, la dependencia de las variables de estado puede representarse de forma natural utilizando bien redes recurrentes, bien redes con conectividad hacia adelante provistas de retardos en las entradas.

Este capítulo está estructurado del siguiente modo. En el apartado 2 se mencionan las reglas de aprendizaje neuronal que han sido utilizadas en el ámbito de la Robótica, clasificándolas según el nivel de supervisión que requieren durante el entrenamiento. El apartado 3 está dedicado a la exposición de los esquemas que se han planteado para el control neuronal. Los tres apartados siguientes describen modelos de redes neuronales utilizados para el aprendizaje de las tres funciones básicas para el control de robots descritas más arriba, esto es, la cinemática inversa, la dinámica inversa y los mapas sensoriomotores. Finalmente, en el apartado 7, se presentan las principales conclusiones derivadas del estudio y desarrollo de modelos neuronales para su aplicación en Robótica.

2. REGLAS DE APRENDIZAJE NEURONAL

Como ya se ha descrito en capítulos previos, el aprendizaje en una red neuronal consiste en la aplicación de una regla de modificación de pesos en paralelo a todas las conexiones de la red.

En este apartado se esboza una clasificación de las reglas de aprendizaje neuronal utilizadas en Robótica, sin entrar en una descripción pormenorizada de las mismas, que ya ha sido realizada en el capítulo 6. El criterio utilizado para la clasificación es el tipo de realimentación requerido durante el entrenamiento.

En un extremo del espectro se sitúan las *reglas correlacionales*, que no utilizan realimentación alguna. Estas reglas ajustan el peso de una conexión en función de la correlación entre los niveles de actividad de las neuronas que dicha conexión interconecta. Derivan todas ellas de la clásica regla hebbiana [Hebb, 1949].

Reglas de este tipo han sido incorporadas a modelos de aprendizaje competitivo [Rumelhart y Zipser, 1985], modelos de resonancia adaptativa [Grossberg, 1987] y mapas auto-organizativos de Kohonen [1988].

El objetivo de estos modelos es la agrupación de las muestras de entrada en función de la similitud de sus características. En el contexto del control de robots, dichos modelos suelen utilizarse para representar un espacio de estados de forma compacta y preservando la topografía. Esta última característica hace referencia al hecho de que neuronas próximas en la red aprendan a ser activadas por entradas similares según una determinada métrica, por ejemplo, la euclídea en los mapas auto-organizativos de Kohonen.

En los próximos apartados se describen dos aplicaciones basadas en representaciones de este tipo, utilizadas para caracterizar la topografía del espacio de trabajo del robot (apartado 4.2) y para expresar de forma compacta el espacio de las posiciones, velocidades y aceleraciones de las articulaciones de un robot (apartado 5.1).

Las *reglas de minimización del error* se sitúan en el polo opuesto del espectro, puesto que requieren una señal de error exacta que se utiliza como realimentación. Ello equivale a decir que precisan conocer las salidas que corresponden a las entradas o secuencias de entradas que se presentarán durante el entrenamiento, ya que sólo a partir de dicha información es posible generar las necesarias señales de error. El objetivo de estas reglas es sintetizar, a partir de los ejemplos de entrenamiento, una función que generalice adecuadamente. Son las reglas más utilizadas, tanto en solitario como en combinación con reglas correlacionales.

Las dos reglas de este tipo utilizadas en el dominio del control de robots son la regla LMS [Widrow y Hoff, 1960] y la regla de retropropagación del error [LeCun, 1985; Rumelhart y col., 1986], que, de hecho, es una extensión de la anterior para su aplicación a redes multicapa.

Esta última regla es la más conocida y ampliamente utilizada aunque, como técnica de minimización basada en el gradiente que es, suele caer en mínimos locales y su velocidad de convergencia es baja. Por ello, se han planteado numerosos procedimientos de aceleración para ser utilizados conjuntamente con esta regla. Otro problema asociado a la retropropagación de estrictamente el error cuadrático medio es que no permite el aprendizaje incremental, esto es, al aprender un nuevo par de

entrada-salida, necesariamente se produce un cierto olvido de los pares previamente aprendidos. Ello ha llevado al planteamiento de extensiones que explícitamente minimizan dicho olvido [Ruiz y Torras, 1994a], así como a la introducción de ruido tanto en las entradas de la red como en los pesos [Ruiz y Torras, 1994b].

Las aplicaciones descritas en los apartados 4.1 y 6.1 utilizan redes basadas en la regla de retropropagación del error, mientras que las descritas en los apartados 4.2 y 5.1 combinan reglas correlacionales con la regla LMS.

Entre ambos extremos del espectro se sitúan las *reglas basadas en una señal de refuerzo*. Esta señal puede entenderse como una realimentación cualitativa, ya que sólo proporciona una indicación de cuán buena o mala es la respuesta a un determinado estímulo. Normalmente, la señal de refuerzo adopta valores continuos entre un máximo (premio) y un mínimo (castigo). Se persigue maximizar el refuerzo total obtenido durante la realización de una determinada tarea.

La regla de este tipo más ampliamente utilizada es la de aprendizaje asociativo [Barto y col., 1983]. Los modelos neuronales que incorporan este tipo de regla se denominan redes de búsqueda asociativa [Barto y col., 1981] y han sido frecuentemente utilizados junto con métodos de diferencias temporales [Sutton, 1988]. Estos métodos, procedentes del ámbito de la programación dinámica, permiten la integración de la señal de refuerzo a lo largo del tiempo (o a lo largo de una trayectoria que lleve un robot móvil hasta la meta, como en el caso que nos ocupará en el apartado 6.3). También la aplicación descrita en el apartado 6.2 utiliza una regla de este tipo.

En [Torras, 1985, 1989] se presenta una más detallada exposición de la clasificación aquí esbozada, mientras que en [Barto, 1985; Fogelman, 1987; Hecht-Nielsen, 1990; Hinton, 1989; Kröse y van der Smagt, 1993] pueden hallarse sistematizaciones del mismo dominio llevadas a cabo por distintos autores que ofrecen puntos de vista alternativos.

3. UTILIZACIÓN DE LAS REDES NEURONALES EN CONTROL

Generalmente, un sistema a controlar puede caracterizarse mediante una función de transición f y una función de salida g . La señal de control $\mathbf{u}(t)$, conjuntamente con el estado actual del sistema $\mathbf{x}(t)$, determina el estado siguiente $\mathbf{x}(t+1)$:

$$\mathbf{x}(t+1) = f(\mathbf{u}(t), \mathbf{x}(t)) \quad (1)$$

En cada estado $\mathbf{x}(t)$, el sistema produce una salida $\mathbf{y}(t)$:

$$\mathbf{y}(t) = g(\mathbf{x}(t)) \quad (2)$$

Un controlador puede ser pensado como un modelo inverso del sistema puesto que, dados una salida deseada $y^*(t)$ y el estado actual, el controlador debe generar la señal de control que producirá dicha salida.

El esquema más simple de control neuronal, denominado *modelización inversa directa* (en inglés, *direct inverse modelling*), utiliza el propio sistema para generar pares de entrada-salida y entrena el modelo inverso directamente mediante el intercambio de entradas y salidas (figura 1). Este esquema es únicamente aplicable a sistemas caracterizados por funciones inyectivas (puesto que, en caso contrario, la inversa no es una función) y su éxito depende de la calidad del muestreo (las entradas deben ser seleccionadas de modo que las salidas generadas cubran adecuadamente el espacio de salidas). Jordan y Rumelhart [1992] ofrecen un tratamiento detallado de estas y otras cuestiones.

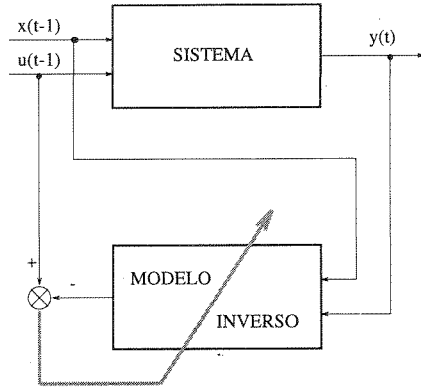


Figura 1. Esquema de modelización inversa directa.

El esquema de *modelización hacia adelante* (en inglés, *forward modelling*) procede en dos etapas. La primera consiste en hacer que una red neuronal aprenda el modelo del sistema a partir de pares de entrada-salida. En la segunda etapa, se antepone a dicho modelo otra red neuronal y se entrena el sistema compuesto por ambas para que aproxime la función identidad (figura 2). Los pesos del modelo se mantienen fijos en esta segunda etapa, en tanto que los pesos del controlador son los que se ajustan mediante aprendizaje. En el caso de que la función subyacente al sistema no sea inyectiva, puede introducirse un cierto sesgo en este esquema para que encuentre una pseudo-inversa con unas determinadas propiedades.

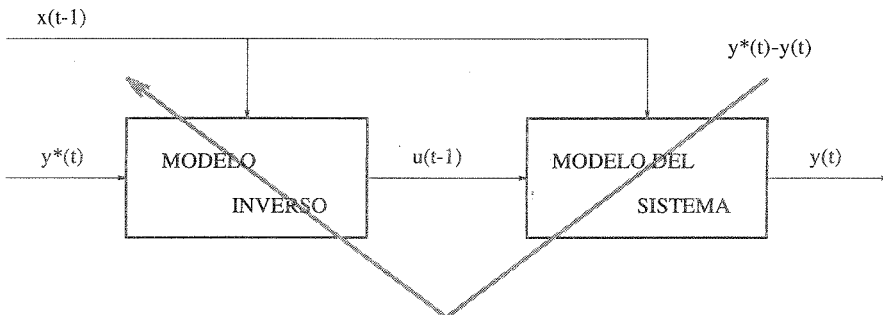


Figura 2. Esquema de modelización hacia adelante.

La primera etapa puede obviarse si se conoce el jacobiano del sistema. El jacobiano es la versión multidimensional de la derivada, es decir, es la matriz de derivadas parciales. Una derivación didáctica del jacobiano de un robot manipulador, ilustrada con numerosos ejemplos, es la proporcionada por Paul [1981]. En este caso particular, el jacobiano permite derivar las velocidades lineal y angular del elemento terminal del robot a partir de las velocidades de las articulaciones. El jacobiano puede aproximarse modificando ligeramente la entrada al sistema y midiendo los cambios que ello ocasiona en la salida. Por aplicación directa de la regla de la cadena, el jacobiano puede ser usado para derivar los errores en la entrada como función de los errores de salida [Kröse y van der Smagt, 1993]. Al ser precisamente ésta la función que cumple el modelo neuronal del sistema en el esquema de modelización hacia adelante, la primera etapa puede eliminarse.

Destaquemos que este esquema también puede ser aplicado aun cuando, en lugar de las salidas deseadas, sólo se disponga de una señal de refuerzo (es decir, una medida de cuán adecuadamente se comporta el controlador neuronal). En este caso, el modelo del sistema engloba tanto al sistema como a la función de refuerzo, y es utilizado igualmente en la segunda etapa para derivar las modificaciones a llevar a cabo en los pesos del controlador neuronal. Alternativamente, puede obviarse la primera etapa de modelización y utilizarse directamente la señal de refuerzo para determinar las modificaciones de los pesos.

Por último, el enfoque del *aprendizaje basado en la señal de error realimentada* (en inglés, *feedback error learning*) requiere disponer de un controlador convencional conectado al sistema, siendo la función del controlador neuronal hacer que la señal de error (la realimentación del controlador convencional) tienda a cero (figura 3). Una característica interesante de este enfoque es que no precisa una fase de aprendizaje previa, sino que el controlador aprende mientras está en operación.

Jordan [1993] ofrece un tratamiento detallado de los conceptos expuestos en esta sección. En el capítulo 3 de [Miller y col., 1990b], Kawato lleva a cabo una comparación de los tres esquemas descritos. La presentación de aplicaciones ofrecida en los próximos apartados también está estructurada de acuerdo a los mencionados esquemas.

Para una exposición más detallada de la utilización de las redes neuronales en control, el lector puede consultar [Cembrano y Wells, 1992; Miller y col., 1990b]. En cuanto al tema más específico del las aplicaciones al control de robots, existen ya algunas recopilaciones disponibles [Bekey y Goldberg, 1993; Torras, 1993, 1994].

A continuación se describen distintos modelos utilizados para el aprendizaje de las tres funciones básicas subyacentes al control de robots, esto es, la cinemática inversa, la dinámica inversa y los mapas sensoriomotores. Para cada uno de dichos

modelos, se indica la regla (o reglas) de aprendizaje en que se basa, de acuerdo a la taxonomía expuesta en el apartado 2, así como el esquema de control utilizado según la nomenclatura introducida en el presente apartado.

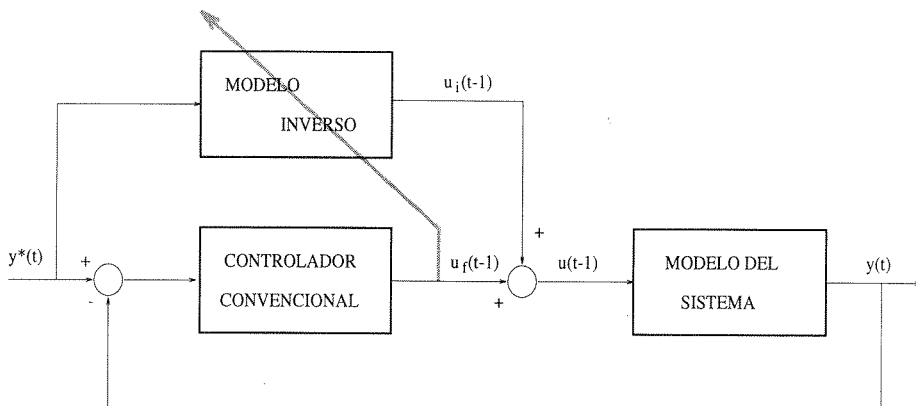


Figura 3. Esquema de aprendizaje basado en el error realimentado.

4. CINEMÁTICA INVERSA DE UN ROBOT MANIPULADOR

Los robots suelen programarse en términos de las coordenadas cartesianas de su entorno, mientras que su control se realiza mediante variables articulares. De aquí la importancia que, para el control de robots, tiene el disponer de funciones precisas que hagan corresponder a una posición y orientación cualesquiera en el espacio, los valores de las variables articulares que hacen que el elemento terminal del robot alcance la posición y orientación mencionadas.

La utilización de redes neuronales para aprender la cinemática inversa de robots manipuladores presenta especial interés cuando no se dispone de un modelo preciso de algunas de las articulaciones (éste es el caso, por ejemplo, de los robots flexibles) o cuando, debido a las condiciones de operación del robot (en el espacio, bajo el mar, etc.), no es posible recalibrarlo.

4.1. Redes multicapa con retropropagación del error

La manera más sencilla de abordar el aprendizaje de la cinemática inversa es aplicar el esquema de *modelización inversa directa* a una red multicapa que incorpore la *regla de retropropagación del error*. Muchos investigadores han seguido este

enfoque y lo han aplicado a una gran variedad de modelos de robot [Kröse y van der Smagt, 1993; Torras, 1993]. La manera de asegurar que la función sea inyectiva, requisito indispensable para poder aplicar este esquema tal como se ha indicado en el apartado precedente, es elegir para el conjunto de aprendizaje sólo configuraciones del robot en que las articulaciones permanezcan siempre en los mismos semiespacios.

Jordan y Rumelhart [1992] han aplicado el esquema de *modelización hacia adelante* al aprendizaje de la cinemática inversa de un robot manipulador con 3 articulaciones que se mueve en un plano. Aunque en este caso la cinemática inversa no es una función, sino una correspondencia de 2 a 3 variables, el controlador neuronal consigue converger a una función cinemática inversa de 2 a 2 variables. Los autores mencionan que es fácil incorporar al procedimiento de aprendizaje la restricción de minimización de la norma o restricciones de suavidad temporal para guiar la selección de la función inversa obtenida.

La conclusión alcanzada después de una extensa experimentación con redes multicapa siguiendo ambos enfoques es que una aproximación burda de la cinemática inversa puede obtenerse rápidamente, pero que una representación precisa de la verdadera función es extremadamente difícil e incluso a veces imposible. La razón de esta dificultad parece ser el carácter global de la aproximación obtenida con este tipo de redes, ya que ésta se ve afectada por todos y cada uno de los pesos [Kröse y van der Smagt, 1993].

Una manera obvia de evitar este efecto global es por supuesto la utilización de representaciones locales, de modo que cada parte de la red sea responsable de sólo una pequeña porción del espacio de entradas. Esta es la idea subyacente al trabajo descrito en el siguiente subapartado.

4.2. Mapas topológicos auto-organizativos

Ritter y col. [1992] han combinado un mapa auto-organizativo de Kohonen, basado en una *regla correlacional*, con la *regla de minimización del error LMS* para el aprendizaje de la cinemática inversa de un robot manipulador con 3 grados de libertad. El esquema de control seguido es el de *modelización inversa directa* con un modo de funcionamiento enteramente no-supervisado. La posición deseada para el elemento terminal se define mediante un punto visualizado por dos cámaras fijas situadas en el entorno del robot.

Las neuronas están organizadas en una malla tridimensional que se pretende que, mediante el aprendizaje, evolucione hacia una representación discretizada del espacio de trabajo del robot. Las entradas a cada neurona son las coordenadas $\mathbf{u}(t)$, en las dos imágenes captadas, del punto que se pretende alcanzar con el elemento

terminal. Las salidas, una vez entrenada la red, son las variables articulares θ_i y los jacobianos A_i correspondientes a dicho punto del espacio. Dada una entrada \mathbf{u} , la red de Kohonen determina la neurona k que mejor correlaciona con dicha entrada, según la inequación:

$$\sum_i w_{ik} u_i \geq \sum_i w_{ij} u_i, \quad \forall j \quad (3)$$

La salida se calcula entonces mediante la expresión:

$$\theta(x) = \theta_k + A_k (\mathbf{u} - \mathbf{w}_k) \quad (4)$$

Un ciclo de aprendizaje comprende los siguientes cuatro pasos:

1. En primer lugar, se aplica la regla de aprendizaje de Kohonen a los pesos de la red:

$$w_{ij}(t+1) = w_{ij}(t) + c h_k(j)(u_j(t) - w_{ik}(t)) \quad (5)$$

donde $h_k(\cdot)$ es una función gaussiana centrada en k , que se usa para modular la adaptación de los pesos de entrada a una neurona en función de su distancia a la neurona activa.

2. Moviendo el robot a las coordenadas articulares $\theta(\mathbf{u})$ obtenidas mediante la ecuación (4), el elemento terminal se sitúa en la posición \mathbf{u}' en coordenadas de las imágenes. La diferencia entre la posición deseada \mathbf{u} y la realmente alcanzada \mathbf{u}' constituye una señal de error, que permite aplicar una regla de minimización del mismo, en este caso la regla LMS:

$$\theta^* = \theta_k + \Delta\theta = \theta_k + A_k (\mathbf{u} - \mathbf{u}') \quad (6)$$

3. Aplicando el incremento de corrección $A_k(\mathbf{u} - \mathbf{u}')$ a las articulaciones del robot, se ajusta la posición del elemento terminal obteniéndose las nuevas coordenadas \mathbf{u}'' . A continuación se aplica la regla LMS al jacobiano, utilizando $\Delta\mathbf{u} = (\mathbf{u} - \mathbf{u}'')$ como señal de error:

$$A^* = A_k + (\Delta\theta - A_k \Delta\mathbf{u}) \frac{\Delta\mathbf{u}^T}{\|\Delta\mathbf{u}\|^2} \quad (7)$$

4. Finalmente, se utiliza la regla de Kohonen para actualizar los valores articulares:

$$\theta_i^{\text{new}} = \theta_i^{\text{old}} + c' h_k(i)(\theta_i^* - \theta_k(t)) \quad (8)$$

y la matriz jacobiana:

$$A_i^{\text{new}} = A_i^{\text{old}} + c' h'_k(i)(A_i^* - A_k(t)) \quad (9)$$

donde c' es una constante que determina la velocidad de aprendizaje, y $h'_k(\cdot)$ es también aquí una función gaussiana centrada en k , utilizada para modular los pasos de aprendizaje en función de la distancia a la neurona activa k .

La amplia experimentación llevada a cabo por los autores mencionados muestra que la red se auto-organiza, constituyendo una representación adecuada del espacio de trabajo, en aproximadamente 30.000 iteraciones del ciclo de entrenamiento. Este resultado muestra la gran capacidad de aprendizaje de este esquema, puesto que las condiciones de operación a las que se ha sometido la red son las peores imaginables: completo desconocimiento del modelo del robot, inicialización aleatoria de los parámetros y muestreo aleatorio del espacio de trabajo durante el entrenamiento.

En la práctica, parece más razonable utilizar este esquema para adaptar una cinemática inversa nominal a la cinemática inversa real de un robot en operación, sometido a un cierto desgaste y descalibración. Además, tanto la inicialización de parámetros como el muestreo del espacio de entrada pueden realizarse de forma más eficiente que la aleatoria. Esta es la línea de trabajo que se está siguiendo en el Instituto de Cibernética en colaboración con la empresa espacial alemana MBB-Erno, como parte del proyecto CONNY.

5. DINÁMICA INVERSA

Por ésta se entiende la función que, dada una trayectoria deseada del elemento terminal del robot, proporciona las fuerzas y pares que deben ser aplicados a las distintas articulaciones para que efectivamente se siga dicha trayectoria. Sólo muy raramente puede calcularse esta función de forma analítica y, en este caso, la función obtenida es adecuada para un único robot sometido a unas determinadas condiciones de carga. Esta es la razón por la cual se suelen aplicar técnicas de control adaptativo, que no requieren conocimiento a priori de esta función, al control dinámico de robots. Sin embargo, el coste computacional de estas técnicas aumenta muy rápidamente con el número de variables de estado, haciendo prohibitiva en muchos casos su aplicación en tiempo real.

Recientemente, se han utilizado modelos de redes neuronales para el aprendizaje de la dinámica inversa bien directamente o a través del error proporcionado por un controlador convencional de ganancia fija, como se describirá en los dos

subapartados que siguen. Conviene destacar que el caso dinámico difiere del cinemático en que, para generar los pares de entrada-salida, se precisa algún tipo de controlador que guíe al robot desde el principio. A medida que progresa el aprendizaje de la función dinámica inversa, la salida de la red neuronal deviene más precisa y el efecto del controlador tiende progresivamente a cero.

5.1. Red CMAC

Miller y col. [1990a] han adaptado el modelo CMAC (Cerebellar Model Articulation Controller), desarrollado por Albus [1975], al control dinámico de un robot industrial con 5 grados de libertad. La adaptación consiste en combinar la rapidez de acceso a tablas proporcionada por la red CMAC con un procedimiento de corrección de errores similar a la *regla LMS*.

La idea subyacente a esta combinación es similar, aunque no exactamente igual, a la de extender los mapas auto-organizativos de Kohonen con la regla LMS para cinemática inversa, como se ha descrito en el apartado 4.2. Aquí el CMAC se utiliza para representar el espacio de estados de forma compacta y localizada (tal como los mapas auto-organizativos de Kohonen se utilizaban para modelar el espacio de trabajo del robot) y, mediante el procedimiento de corrección de errores, los pesos sinápticos se modifican proporcionalmente a la diferencia entre la salida deseada y la obtenida mediante la red (mientras que la regla LMS se utilizaba para estimar la salida deseada en el caso cinemático).

La tarea consiste en enseñar a un robot a seguir una determinada trayectoria. Para ello, se muestran puntos sucesivos de la trayectoria tanto a la red neuronal como al controlador de ganancia fija y se usa la suma de sus respuestas como consigna para guiar el robot. Después de cada ciclo, la consigna junto con el estado del robot se utilizan como par de entrada-salida para entrenar la red neuronal, siguiendo el esquema de *modelización inversa directa*.

A medida que progresa el aprendizaje, la red CMAC aproxima la función dinámica inversa, de modo que la diferencia entre el estado actual y el deseado tiende a cero, y, por consiguiente, la red neuronal toma el control que antes tenía el controlador de ganancia fija.

Miller y col. [1990a] presentan los resultados de varios experimentos usando dos tipos de trayectorias, tres valores de ganancia de velocidad y cuatro redes con distinto número de pesos. La conclusión es que la red converge a un error pequeño (entre 1 y 2 unidades de los codificadores de posición) en 10 intentos, suponiendo que se utilicen un número suficiente de pesos. Puesto que el modelo CMAC se basa en el acceso a tablas, los resultados muestran una gran dependencia respecto del número de pesos utilizados.

5.2. Aprendizaje a partir del error realimentado

La misma tarea de aprendizaje de una trayectoria que acabamos de describir ha sido abordada por Kawato y col. [1987] bajo el esquema del *aprendizaje basado en la señal de error realimentada*. El robot utilizado tiene 3 grados de libertad y la red neuronal está formada sólo por 3 neuronas (una por articulación) cuyas entradas son 13 funciones no-lineales de las velocidades y aceleraciones de las articulaciones. Así pues, 39 pesos son repetidamente modificados utilizando una regla de corrección de errores similar a la *regla LMS*.

El proceso de entrenamiento es el mismo descrito en el subapartado precedente, esto es, se proporciona la trayectoria deseada tanto a la red neuronal como al controlador de ganancia fija y la suma de sus respuestas se usa como consigna para gobernar el robot. La diferencia entre ambos enfoques radica en la señal de error utilizada para modificar los pesos. En el enfoque de Miller y col., se generan pares de entrada-salida correctos y se usa como señal de error la diferencia entre la salida deseada y la proporcionada por la red frente a una determinada entrada, siguiendo pues estrictamente la regla LMS.

Por el contrario, Kawato y col. no generan pares de entrada-salida, sino que usan la salida del controlador de ganancia fija (que, de algún modo mide la desviación del estado actual respecto del deseado en términos de la señal de control necesaria para que el primero se aproxime al segundo) directamente como señal de error. Obsérvese que esta medida del error es menos precisa que la anterior, pero tiene la ventaja de ser directamente accesible en el bucle de control, obviándose así la necesidad de determinar el estado actual del robot. Si se considera este error como una medida cualitativa en lugar de cuantitativa, entonces se convierte en una especie de refuerzo y este esquema puede pensarse como aprendizaje basado en una señal de refuerzo.

Los autores presentan resultados mostrando que, después de entrenar 300 veces al robot a seguir una trayectoria de 6 segundos de duración, el par medio realimentado decrece dos órdenes de magnitud, demostrando que la red neuronal ha tomado el control en detrimento del controlador de ganancia fija. Además, el error cuadrático medio en los ángulos de las articulaciones decrece un orden de magnitud y medio.

6. MAPAS SENSORIOMOTORES

Funciones que relacionen patrones sensoriales con consignas motoras son necesarias para llevar a cabo operaciones tales como el posicionamiento visual de robots manipuladores, la manipulación fina, y la navegación en el caso de robots

móviles. A continuación se describen los modelos neuronales propuestos para abordar estas tres operaciones.

6.1. Posicionamiento visual de robots

El paso clave hacia el control basado en visión es el de mover una cámara de modo que la imagen captada se corresponda con una imagen de referencia dada. El objetivo a alcanzar no es pues ya una posición en el espacio, sino una sensación óptica. Existen numerosas aplicaciones que se beneficiarían de un enfoque como éste, por ejemplo la inspección o el agarre de objetos que no puedan ser posicionados con precisión, o el acople entre dos elementos móviles (piénsese en aplicaciones submarinas o espaciales).

Este posicionamiento por visión se aborda habitualmente prefijando un conjunto de características en la imagen y calculando a continuación la matriz de interacción que relaciona desplazamientos 2D de dichas características en la imagen con movimientos 3D de la cámara [Samson y col., 1990]. Las potenciales ventajas derivadas de la aplicación de redes neuronales a esta tarea son: (1) la automatización de la programación del sistema, puesto que la matriz de interacción se obtiene por aprendizaje; y (2) el ahorro de tiempo de ejecución, al obviar la necesidad de calcular la correspondencia entre las características de la imagen captada y la de referencia.

La primera ventaja resulta ya patente en el sistema desarrollado por Hashimoto y col. [1992], mientras que un primer paso hacia la consecución de la segunda se ha dado en el marco del proyecto CONNY, en una colaboración del Instituto de Cibernética con la empresa francesa THOMSON, con la definición de parámetros globales de la imagen que preserven información de posición y orientación [Venaille y col., 1994]. Obsérvese que, en cierto sentido, se persigue el objetivo opuesto al del reconocimiento de objetos, donde se requiere que las características utilizadas para la discriminación sean invariantes a la posición y orientación de los objetos.

El sistema de Hashimoto y col. [1992] funciona sobre la base de que es posible detectar cuatro puntos característicos en la imagen. Se utilizan dos redes neuronales para mover la cámara de modo que se capten esos cuatro puntos en una configuración predeterminada. Las dos redes funcionan esencialmente de la misma manera, pero a diferentes resoluciones: una controla los movimientos de largo alcance, mientras que la otra guía los movimientos finos. La única diferencia es que la primera red utiliza la posición de la cámara como entrada, mientras que la segunda funciona independientemente de la posición absoluta.

El entrenamiento de la red se consigue moviendo la cámara desde la posición de referencia a posiciones arbitrarias, y aplicando entonces el algoritmo de

retropropagación del error para aprender la asociación entre los desplazamientos de los cuatro puntos en la imagen y el movimiento realizado. Obsérvese pues que, en términos de la clasificación descrita en el apartado anterior, el aprendizaje se basa aquí exclusivamente en una regla de minimización del error. Una vez entrenada la red, a partir de la imagen captada, se calculan los desplazamientos de los cuatro puntos respecto a sus posiciones en la imagen de referencia y se entran dichos desplazamientos a la red. La salida proporcionada por la red será el movimiento inverso al que debe aplicarse a la cámara para que la imagen captada sea semejante a la de referencia.

Los resultados obtenidos mediante simulación muestran que, después de 30.000 iteraciones del algoritmo de retropropagación del error, la posición de los cuatro puntos después de resituarse la cámara difiere de la de referencia en, como máximo, un píxel. Desde el punto de vista de la precisión, los resultados son pues satisfactorios. La principal limitación radica en la identificación de los cuatro puntos y el establecimiento de la correspondencia entre dichos puntos en la imagen captada y la de referencia.

Con objeto de superar dicha limitación, Venaille y col. [1994] han definido unos coeficientes de Fourier normalizados, que permiten la determinación de la posición y orientación de un objeto a partir de su contorno. Se han usado 32 de dichos coeficientes como entrada a una red neuronal que, también mediante retropropagación, aprende a asociar diferencias entre coeficientes a movimientos de la cámara. La red ha sido probada, no ya en simulación, sino en una célula robotizada en la que se ha incorporado una cámara al elemento terminal de un robot manipulador.

El enfoque descrito es sólo aplicable al caso en que el contorno de un objeto esté enteramente contenido en la imagen, lo cual supone una restricción considerable. Los trabajos que se están llevando a cabo en la actualidad están encaminados a la definición de parámetros globales de una mayor generalidad.

6.2. Manipulación fina

La inserción de componentes con tolerancias muy pequeñas es un ejemplo de operación para la cual el diseño de una estrategia detallada de control de fuerza, que funcione correctamente en todas las situaciones posibles y sujeta a las condiciones de incertidumbre y ruido propias de un entorno real, es extremadamente difícil. Además, la estrategia debe de ser particularizada para cada operación de inserción específica. En consecuencia, la posibilidad de usar una red neuronal para aprender la acción que debe aplicarse frente a cada patrón de fuerza (es decir, el apropiado mapa sensoriomotor) resulta muy atractiva.

Gullapalli y col. [1992] han utilizado un sistema de *aprendizaje asociativo basado en una señal de refuerzo* para aprender la estrategia de control por acomodación activa necesaria para insertar un vástago en un agujero en un entorno 2D. Su sistema toma la posición de vástago junto con los valores de fuerza y par como entrada, y produce una consigna de velocidad como salida. La señal de refuerzo depende de la discrepancia entre la posición deseada y la posición real del vástago, con un término de penalización que se activa cada vez que las fuerzas medidas sobre el vástago superan un valor máximo predeterminado.

El sistema está formado por una red multicapa con 3 neuronas estocásticas en la capa de salida y neuronas sigmoideas deterministas en las restantes capas. Todas las neuronas toman valores reales. La estocasticidad en la salida es un ingrediente esencial del aprendizaje por refuerzo (véase el apartado 2), ya que permite realizar una búsqueda de la consigna más apropiada a aplicar en respuesta a un patrón de fuerza dado.

Durante el entrenamiento, cada intento se inicia con el vástago situado en una posición y una orientación aleatorias respecto al agujero y finaliza con el vástago satisfactoriamente insertado o bien después de transcurridas 100 unidades de tiempo. Los experimentos realizados con el robot real muestran que, después de 150 intentos, el robot siempre es capaz de completar la inserción. Además, la duración de la inserción decrece continuamente desde 100 a 20 unidades de tiempo a lo largo de 500 intentos. Hay que resaltar la importancia de estos resultados, al haber sido obtenidos con un robot equipado con un sensor de fuerza, en lugar de con un simulador.

6.3. Generación de trayectorias para robots móviles

Los robots móviles realizan a menudo tareas en entornos no previamente conocidos. Por ello, es de gran importancia que puedan llegar a un determinado objetivo sin colisionar con los obstáculos que encuentren en su camino. Igualmente es de desear que vayan aprendiendo de la experiencia de modo que, después de un cierto tiempo de permanecer en el mismo entorno, sigan trayectorias más eficientes que las generadas inicialmente.

Millán y Torras [1992] han desarrollado un controlador, basado en *aprendizaje por refuerzo*, que permite generar trayectorias progresivamente más cortas y seguras. Un supuesto básico de dicho controlador es que el entorno en que se desenvuelve el robot no es laberíntico, es decir, para generar trayectorias en dicho entorno no se requiere una sofisticada capacidad de planificación, sino sobretodo habilidad para sortear obstáculos.

La entrada al sistema está formada por una fuerza de atracción ejercida por la meta y cuatro fuerzas de repulsión ejercidas por los obstáculos. La salida del sistema codifica una acción del robot en términos de una orientación y una longitud de paso. Una señal de refuerzo valora la adecuación del paso dado por el robot a la situación estimular presente, en función del criterio de optimización perseguido, que supone un compromiso entre minimizar la longitud de la trayectoria y maximizar la distancia a los obstáculos. Todas las señales involucradas están codificadas como números reales, lo cual supone la extensión de los algoritmos de aprendizaje por refuerzo (clásicamente propuestos para señales binarias) a dicho caso.

El sistema se compone de dos redes: un generador de pasos y un módulo crítico. Este último es una red de tres capas que funciona mediante un algoritmo de retropropagación, en tanto que el generador de pasos se construye incrementalmente a partir de una red de dos capas provista de dos neuronas de salida estocásticas. Una vez esta red mínima inicial ha aprendido una somera asociación entre entradas y salidas, los pesos se dejan fijos y se añade a la red una capa intermedia. La adaptación de los nuevos pesos mediante aprendizaje por refuerzo permite ajustar la asociación.

Después de 75.000 ciclos de aprendizaje, el generador de pasos ha aprendido un mapa sensoriomotor suficientemente preciso como para generar trayectorias satisfactorias desde prácticamente cualquier posición inicial del robot en el entorno. Obsérvese que, como la realimentación es mucho menos informada en el caso del aprendizaje por refuerzo que en el de minimización del error, el primero requiere habitualmente un período de entrenamiento considerablemente más largo.

Hay que destacar que el mapa sensoriomotor obtenido es altamente insensible al ruido en las entradas (la generación de trayectorias sigue siendo satisfactoria aún cuando se añada un 20% de ruido blanco), muestra una buena capacidad de generalización frente a cambios en la situación de la meta o en el número y distribución de los obstáculos, e incluso funciona adecuadamente cuando la meta y los obstáculos son móviles. La principal limitación es el largo tiempo de entrenamiento que, sin embargo, se reduce considerablemente si se utiliza una red modular y se inicializan adecuadamente los parámetros mediante un procedimiento de búsqueda local [Millán, 1992].

En el marco del proyecto B-Learn II, y en colaboración con el Centro de Investigación que la Comunidad Europea tiene en Ispra (Italia), este sistema de generación de trayectorias, considerablemente mejorado y adaptado a las nuevas entradas y salidas, ha sido incorporado a un robot móvil de la familia NOMAD 200 [Millán y Torras, 1994]. La adaptación afecta especialmente a la codificación de las entradas, puesto que el robot móvil está equipado con dos anillos de sensores, uno de infrarrojos y otro de ultrasonidos, así como con sensores de contacto utilizados para evitar que el robot colisione durante la fase de entrenamiento. Otra modificación

importante es la incorporación de una serie de «reflejos» al robot, que facilitan -y consecuentemente, aceleran- notablemente el aprendizaje.

7. CONCLUSIONES

Los controladores de robots basados en redes neuronales son en la actualidad todavía prototipos, que han mostrado interesantes aptitudes en la resolución de una gran variedad de tareas de control simplificadas. Entre las simplificaciones habituales se encuentran las de trabajar en simulación, considerar un número reducido de grados de libertad, asumir la capacidad de medir con exactitud la posición del robot en su espacio de trabajo, considerar una forma de robot idealizada, y rebajar la precisión requerida por la aplicación.

El siguiente estadio de desarrollo no sólo deberá eliminar dichas simplificaciones, sino también atacar cuestiones como la robustez frente a perturbaciones en las entradas, velocidad de aprendizaje, y el desarrollo de un marco conceptual para guiar la asignación de parámetros. Este último punto hace referencia al problema encontrado a menudo por los usuarios de redes neuronales de tener que probar numerosos valores para los parámetros antes de conseguir adaptar un determinado modelo neuronal a una aplicación particular.

Para alcanzar el último estadio de ser ampliamente utilizados en la industria y los servicios, los neurocontroladores deberán poder integrarse fácilmente con sistemas geométricos y simbólicos, que puedan explotar conocimiento a priori sobre el dominio para ganar eficacia. Por ejemplo, el controlador para robots móviles descrito en el apartado precedente puede llegar a evitar obstáculos de forma eficiente, pero necesita ser acoplado a un sistema de planificación para poder generar trayectorias adecuadas en entornos laberínticos.

AGRADECIMIENTOS

La investigación descrita en los apartados 4.2, 6.1 y 6.3 ha sido parcialmente financiada por el Programa ESPRIT III de la Comunidad Europea, a través de los contratos No. 6715 (proyecto «CONNY: Robot Control based on Neural Network Systems») y No. 7274 (proyecto «B-Learn II: Behavioural Learning: Combining Sensing and Action»). La autora agradece a G. Cembrano, G. Wells, C. Doherty, E. Celaya y J. del R. Millán su valiosa contribución a los proyectos mencionados.

REFERENCIAS

- Albus, J.S., «A new approach to manipulator control: The cerebellar model articulation controller (CMAC)», *Trans. of the ASME, Journal of Dynamic Systems, Measurement and Control*, 97, (1975), 220-227.
- Barto, A.G., «Learning by statistical cooperation of self-interested neuron-like computing elements», *Human Neurobiology*, 4, (1985), 229-256.
- Barto, A.G., Sutton, R.S. y Anderson, C.W., «Neuron-like adaptive elements that can solve difficult learning control problems», *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 13, N. 5, (1983), 834-846.
- Barto, A.G., Sutton, R.S. y Brouwer, P.S., «Associative Search Network: A reinforcement learning associative memory», *Biological Cybernetics*, 40, (1981), 201-211.
- Bekey, G.A. y Goldberg, K.Y., *Neural Networks in Robotics*, Kluwer Academic Publishers, 1993.
- Cembrano, G. y Wells, G., «Neural networks for control», *Artificial Intelligence in Process Control*, Pergamon Press, (1992).
- Fogelman-Soulié, F., «Le connexionnisme», *Support de cours MARI 87 - COGNITIVA 87*, Paris, mayo, (1987).
- Grossberg, S., «Competitive learning: from interactive activation to adaptive resonance», *Cognitive Science*, 11, (1987), 23-63.
- Gullapalli, V., Gruben, R. y Barto, A., «Learning reactive admittance control», *IEEE Conf. on Robotics and Automation*, 2, (1992), 1475-1480.
- Hashimoto, H., Takashi, K., Kudou, M. y Harashima, F., «Self-organizing visual servo system based on neural networks», *IEEE Control Systems*, abril, (1992), 31-36.
- Hebb, D.O., *The Organization of Behavior*, Wiley, New York, 1949.
- Hecht-Nielsen, R., *Neurocomputing*, Addison-Wesley, 1990.
- Hinton, G.E., «Connectionist learning procedures», *Artificial Intelligence*, 40, (1989), 185-234.
- Jordan, M.I., «Computational aspects of motor control and motor learning», en Heuer, H. y Keele, S. (Eds.), *Handbook of Perception and Action: Motor Skills*, Academic Press, New York, 1993.
- Jordan, M.I. y Rumelhart, D.E., «Forward models: Supervised learning with a distal teacher», *Cognitive Science*, 16, (1992), 307-354.

- Kawato, M., Uno, Y., Isobe, M. y Suzuki, R., «A hierarchical model of voluntary movement and its application to robotics», *Proc. IEEE 1st Intl. Conf. on Neural Networks*, San Diego, Vol. IV, IEEE Press, Piscataway, New Jersey, (1987), 573-582.
- Kohonen, T., *Self-Organization and Associative Memory*, Springer-Verlag, Berlin Heidelberg New-York Tokyo, 1988².
- Kröse, B.J.A. y van der Smagt, P.P., *An Introduction to Neural Networks*, University of Amsterdam, 1993⁵.
- LeCun, Y., «Une procédure d'apprentissage pour réseau au seuil assymétrique», *Proceedings of COGNITIVA*, (1985), 599-604.
- Millán, J. del R., «Building reactive path-finders through reinforcement connectionist learning: Three issues and an architecture», *Proc. 10th European Conference on Artificial Intelligence (ECAI'92)*, (1992), 661-665.
- Millán, J. del R. y Torras, C., «A reinforcement connectionist approach to robot path finding in non-maze-like environments», *Machine Learning*, Vol. 8, N. 3/4, (1992), 363-395.
- Millán, J. del R. y Torras, C., «Efficient reinforcement learning of navigation strategies in an autonomous robot», *Intl. Conf. on Intelligent Robotics Systems (IROS'94)*, septiembre, (1994).
- Miller, W.T., Hewes, R.P., Glanz, H.H. y Kraft, L.G., «Real-time dynamic control of an industrial manipulator using a neural-network-based learning controller», *IEEE Trans. on Robotics and Automation*, Vol. 6, N. 1, (1990a), 1-9.
- Miller, W.T., Sutton, R.S. y Werbos, P.J., *Neural Networks for Control*, The MIT Press, Cambridge, 1990b.
- Paul, R.P., *Robot Manipulators: Mathematics, Programming and Control*, The MIT Press, Cambridge, 1981.
- Ritter, H., Martinetz, T. y Schulten, K., *Neural Computation and Self-Organizing Maps*, Addison Wesley, New York, 1992.
- Ruiz de Angulo, V. y Torras, C., «Random weights and regularization», *Proc. Intl. Conf. on Artificial Neural Networks (ICANN'94)*, Sorrento, mayo, (1994).
- Ruiz de Angulo, V. y Torras, C., «On-line learning with minimal degradation in feedforward networks», *IEEE Trans. on Neural Networks*, pendiente de publicación.

- Rumelhart, D.E. y Zipser, D., «Feature discovery by competitive learning», *Cognitive Science*, 9, (1985), 75-112.
- Rumelhart, D.E., Hinton, G.E. y Williams, R.J., «Learning representations by back-propagating errors», *Letters to Nature*, 323, (1986), 533-535.
- Samson, C., LeBorgne, M. y Espiau, B., *Robot Control: The Task Function Approach*, Oxford Engineering Science Series 22, Oxford Science Publications, Oxford, 1990.
- Sutton, R.S., «Learning to predict by the methods of temporal differences», *Machine Learning*, 3, (1988), 9-44.
- Torras, C., *Temporal-Pattern Learning in Neural Models*, Lecture Notes in Biomathematics N. 63, Springer-Verlag, 1985.
- Torras, C., «Relaxation and neural learning: points of convergence and divergence», *Journal of Parallel and Distributed Computing*, 6, (1989), 217-244.
- Torras, C., «From geometric motion planning to neural motor control in robotics», *AI Communications*, Vol. 6, N. 1, (1993), 3-17.
- Torras, C., «Neural learning for robot control», en Cohn, A. (Ed.), *Proc. 11th European Conf. on Artificial Intelligence (ECAI'94)*, Amsterdam, agosto, 1994, 814-819.
- Venaille, C., Wells, G. y Torras, C., «A neural network approach to image-based robot positioning», *Proc. of SICICA*, Budapest, junio, (1994).
- Widrow, B. y Hoff, M.E., «Adaptative switching capatibility and its relation to the mechanisms of association», *Kybernetik*, 12, (1960), 204-215.