

APROXIMACION DE LA DINAMICA INVERSA DE UN MANIPULADOR ROBOTICO MEDIANTE UN ALGORITMO ESTOCASTICO DE APRENDIZAJE EN REDES NEURONALES

Enrique Carlos Segura
Departamento de Computación, Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
esegura@dc.uba.ar

RESUMEN

Se utiliza el algoritmo SAGA [Segura, 1996] para aproximar la dinámica inversa de un manipulador robótico de dos juntas rotacionales. SAGA (Simulated Annealing + Gradiente + Adaptacion) es una estrategia estocástica para construcción aditiva de una red neuronal artificial del tipo perceptrón bicapa basada en tres elementos esenciales: a) modificación de los pesos de la red mediante información del gradiente de la función de costo; b) aceptación o rechazo del cambio propuesto según una técnica de simulated annealing clásico; y c) crecimiento progresivo de la red neuronal, conforme su estructura resulta insuficiente, según una estrategia conservadora de adición de unidades a la capa oculta.

Se realizan experimentos y se analiza la eficiencia en términos de la relación entre error relativo promedio -en el conjunto de entrenamiento y en el de testeo-, tamaño de la red y tiempo de cálculo.

Se destaca la aptitud de la técnica propuesta para realizar buenas aproximaciones minimizando la complejidad de la arquitectura de la red y, por ende, la memoria computacional requerida. Asimismo se discute la evolución del proceso de minimización conforme la superficie de costo va siendo modificada a medida que la red se agranda.

I INTRODUCCION

Una de las propiedades más popularizadas del Simulated Annealing (SA) clásico y, a la vez, más frecuentadas en la bibliografía [2][6][10], es su capacidad para obtener en los problemas de optimización complejos (i.e. dominios de muchas variables y función de costo con muchos mínimos locales y de gradiente muy alto) buenas aproximaciones "gruesas", es decir, acercamiento a un óptimo o subóptimo en tiempos razonables. Como contraparte, el SA suele fallar o ser muy lento en el "ajuste fino", es decir la localización precisa del óptimo buscado. Por otro lado, los algoritmos de minimización (optimización) basados en el gradiente (y en el caso particular de las redes neuronales, el Backpropagation [11]) resultan casi siempre incapaces de escapar de una zona de mínimo local pero muy adecuados para, ya dentro de la región deseada, precisar la ubicación del mínimo. Así surge, naturalmente, la idea de un algoritmo híbrido que cooperativamente las buenas propiedades de ambos enfoques.

El otro ingrediente que se trata de incluir apunta a la búsqueda de la optimalidad. Son bien conocidas [3][4] las condiciones mínimas que deben imponerse sobre la estructura de una red neuronal de tipo feedforward para garantizar su universalidad en tanto aproximadora de funciones continuas. Atento a ellas, se buscó incorporar a la red propuesta o, mejor dicho, a su estrategia de aprendizaje, un mecanismo de adición de unidades de procesamiento que hiciera al sistema capaz de obtener estructuras de complejidad mínima o cuasi-mínima para aproximar con errores aceptables la función objetivo.

II HACIA UNA ESTRUCTURA SIMPLE

Muchos investigadores han buscado reducir clases amplias de funciones a combinaciones simples de funciones simples. Uno de los primeros antecedentes es el de Kolmogorov [7] quien, muy alejado del discurso conexionista, estableció la representabilidad (exacta) de cualquier función continua en muchas variables por medio de sumas y composiciones de funciones de una variable. Este resultado, así como sus sucedáneos de unos pocos años después [1][13], puede traducirse, en términos de redes neuronales, como la demostración de que cualquier función continua puede ser representada como una red neuronal de tres

capas (si bien lo que Kolmogorov se proponía era, en realidad, responder negativamente al tridécimo problema de Hilbert). Pero estos resultados son presentados de manera por completo no constructiva.

A partir de entonces, distintos trabajos jalonan este camino. En 1989, dos matemáticos presentan, casi simultáneamente, demostraciones de que los perceptrones de tres capas (una de entrada, una oculta, una de salida) con funciones sigmoideas de activación en sus unidades intermedias pueden aproximar con un error arbitrariamente pequeño, en cualquier métrica L^p , una función continua arbitraria [3][4]. Enunciamos aquí el resultado en la versión de [4].

TEOREMA (Funahashi)

Sea ϕ función monótona creciente y acotada; $C \in R^N$ compacto, $f: C \rightarrow R^L$ continua.

Dado $\varepsilon > 0$ arbitrario, existe $K \in N$ y constantes reales W_{ij}, u_j ($j=1, \dots, K, i=1, \dots, L$) y

w_{jk} ($j=1, \dots, K; k=1, \dots, N$) tales que si $\tilde{f}(\xi_1, \dots, \xi_N) = \sum_{j=1}^K W_{ij} \phi(\sum_{k=1}^N w_{jk} \xi_k - u_j)$

entonces $\|f - \tilde{f}\|_{L^\infty(C)} = \max_{\xi \in C} |f(\xi_1, \dots, \xi_N) - \tilde{f}(\xi_1, \dots, \xi_N)| < \varepsilon$

Este resultado vale, además, para cualquier métrica L^p , en particular para la cuadrática.

Resultados más recientes [5][8] han precisado como condición necesaria y suficiente sobre ϕ que no sea un polinomio.

La red neuronal resultante tendrá la forma que exhibe la figura 1.

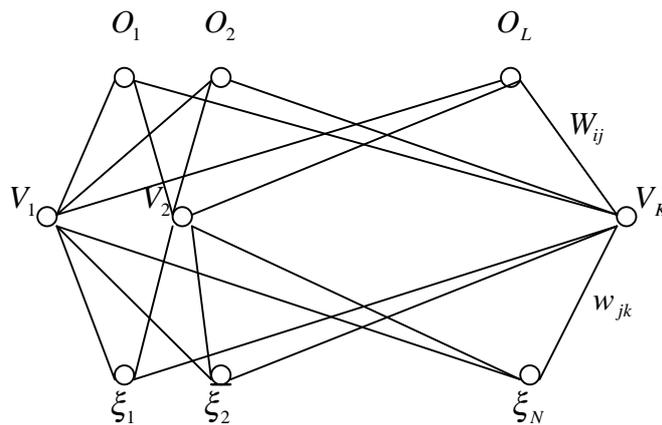


Fig. 1: la estructura básica de un perceptrón de tres capas (una oculta). Las unidades intermedias tienen activación sigmoidea; las de entrada y las de salida, lineal.

Nos restringiremos, con evidente conservación de toda generalidad, al caso de una sola variable de salida (L perceptrones en lugar de uno). Luego, el conjunto $\{W_{ij}\}$ de parámetros será simplemente $\{c_i\}$

En esto se inspira la idea de obtener, con la estructura más económica posible, aproximaciones de cierta utilidad. El algoritmo que propondremos realiza, específicamente, una estrategia simple -¿la más simple?- de agregado de unidades ocultas.

III EL ALGORITMO

Se presenta a continuación un esquema del algoritmo SAGA (Simulated Annealing + Gradiente + Adaptación). Dicha técnica ya ha sido aplicada, con ligeras diferencias, a problemas de calibración de cámaras de video para robots móviles [9][12].

```

- Obtener un conjunto de entrenamiento (TRS) y otro de testeo (TES) a partir de f (de una sola variable de salida).
- Inicializar la red al azar /*  $c_i(0), w_{ij}(0), u_i(0) \approx U(0,1) \quad i = 1, \dots, K; j = 1, \dots, N$  */
  con una adecuada cantidad inicial
  K de unidades intermedias
- Mientras  $t < t_{\text{máx}}$  /* o bien  $E_{TRS} > \varepsilon$  o bien  $E_{TES} > \varepsilon$  */
  {
  T  $\leftarrow$  T(t); /* típicamente  $T(t) = A/\log(1+t)$  */
  proponer_cambio(W, W*);
  generar  $p \approx U(0,1)$ 
  Si  $\exp\{-\Delta E/T\} > p$ ; /*  $\Delta E = E_x(W^*) - E_x(W)$  */
    W  $\leftarrow$  W*;
  Si no
    rechazos++;
  Si rechazos > tol y  $K < K_{\text{máx}}$ 
    agregar(W, K);
  t++;
  }

proponer_cambio(W, W*)
{
  Elegir x perteneciente a TRS al azar;
  Elegir  $r \approx U(0,R)$ ; /* R paso máximo fijado heurísticamente */
   $W^* \leftarrow W - r \frac{\nabla E_x(W)}{|\nabla E_x(W)|}$ ; /*  $E_x(W)$  término de la función de error debido a x */
}

agregar(W, K)
{
  K++; /* crea una nueva unidad oculta */
  Elegir m entre 1 y K-1; /* típicamente m tal que  $|C_m|$  es máximo */
   $u_K \leftarrow u_m$ ;
   $w_{Kj} \leftarrow w_{mj}$  para  $j = 1, \dots, N$ ;
   $c_K = c_m \leftarrow c_m/2$ ;
  rechazos  $\leftarrow$  0;
}

```

Si bien el algoritmo parece altamente parametrizado, debe destacarse que de las experiencias realizadas surge que los parámetros relevantes son sólo tres: uno para la curva de enfriamiento del SA (variable según la elección de dicha curva), el paso máximo R y el valor de tolerancia (tol) que, al ser superado por el número de rechazos consecutivos, produce la invocación a la rutina de ampliación de la capa oculta de la red, la que, al tiempo que incrementa en N+2 la dimensión del espacio de los pesos, mantiene el valor de la función de costo obtenida hasta ese momento.

IV UNA APLICACIÓN

SAGA fue aplicado en el presente trabajo a la aproximación de la dinámica inversa de un manipulador robótico de dos grados de libertad. Es decir, dado un esquema de control como el de la figura 2, donde $y(t)$ es la salida en tiempo t, $y^*(t)$ la salida deseada y $u(t)$ la señal de control, se trata de modelar la aplicación inversa: qué señal de control habría correspondido aplicar para obtener la salida deseada. En el caso del

manipulador, el modelo inverso relaciona aceleraciones $\ddot{\theta}_1$ y $\ddot{\theta}_2$ en las juntas con torques τ_1 y τ_2 a aplicar (i.e. voltajes en los motores), dadas posiciones angulares θ_1 y θ_2 y velocidades $\dot{\theta}_1$ y $\dot{\theta}_2$.

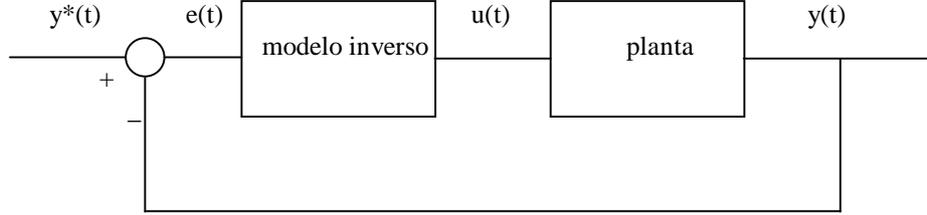


Fig. 2: Esquema de control con modelo inverso

Si bien se conoce una expresión analítica para dicha inversa, dada por las ecuaciones IV.1 y IV.2, estas ecuaciones no representan modelos exactos de la dinámica inversa de ningún manipulador real, la cual debería ser modelada a partir de mediciones experimentales sobre la planta. De todas formas, a los efectos de evaluar la eficiencia del método propuesto, se realizaron simulaciones computacionales utilizando las aproximaciones dadas por dichas ecuaciones.

$$\tau_1 = m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 \cos(\theta_2) (2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 \sin(\theta_2) \dot{\theta}_2^2 - 2m_2 l_1 l_2 \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g \cos(\theta_1 + \theta_2) + (m_1 + m_2) l_1 g \cos(\theta_1) \quad \text{IV.1}$$

$$\tau_2 = m_2 l_1 l_2 \cos(\theta_2) \ddot{\theta}_1 + m_2 l_1 l_2 \sin(\theta_2) \dot{\theta}_1^2 + m_2 l_2 g \cos(\theta_1 + \theta_2) + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \quad \text{IV.2}$$

siendo l_i y m_i la longitud y la masa, respectivamente, del link i .

V EXPERIMENTOS

Se aplicó SAGA a la función $\tau = (\tau_1, \tau_2)$ en el dominio de R^6 definido por $\theta_1, \theta_2 \in [0, \pi]$, $\dot{\theta}_1, \dot{\theta}_2 \in [-0.5, 0.5]$ y $\ddot{\theta}_1, \ddot{\theta}_2 \in [-0.1, 0.1]$ con $l_1 = 0.4$ m., $l_2 = 0.2$ m., $m_1 = 1$ kg. y $m_2 = 0.5$ kg.

La función aproximante, es decir la red neuronal, fue de la forma

$$\tilde{\tau}_i(\vec{x}) = \sum_{j=1}^K c_j \phi \left(\sum_{k=1}^6 w_{jk} x_k - u_j \right)$$

para $y = 1, 2$, siendo $\vec{x} = (\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2, \ddot{\theta}_1, \ddot{\theta}_2)$ y $\phi(h) = \tanh(\beta h) = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$

Exhibimos aquí algunos de los resultados obtenidos para el caso de τ_1 . La figura 3 muestra la evolución del error relativo promedio conjuntos de entrenamiento independientes. Se observan buenos resultados para 300 ejemplos (6.6 %) y para 400 ejemplos (7.4 %). La figura 2 muestra, para los mismos conjuntos, la evolución del error de entrenamiento a medida que se fueron agregando unidades a la capa oculta. Fijando un máximo de 30 unidades se llega a las aproximaciones precisadas. Debe destacarse, sin embargo, que no siempre una mayor cantidad de ejemplos produce mayor error en la aproximación. Por ejemplo, si hubiesemos decidido que el proceso terminara a las 20000 iteraciones (fig. 1) o en un máximo de 12-14 unidades ocultas (fig. 2), claramente habría resultado más exitoso para el conjunto de 400 ejemplos. Se destaca, de cualquier forma, la gran economía y simplicidad de la estructura de la red

neuronal resultante. La figura 5 presenta, para el conjunto anterior de 400 ejemplos, y otro de testeo de 1500, la evolución temporal de los errores (siempre relativos y promedio). Aquí se ilustra con claridad el fenómeno del sobreentrenamiento: a partir de las 50000 iteraciones el error de testeo crece, lo cual sugiere que la fase útil del proceso se ha cumplido.

CONCLUSIONES

Se presentó una técnica estocástica para la construcción de RNAs de tipo perceptrón bicapa que permite obtener buenas aproximaciones de funciones continuas cualesquiera mediante estructuras muy simples y número mínimo o cuasi-mínimo de unidades de procesamiento.

La propuesta fue ensayada en el caso de la inversa dinámica de un manipulador robótico de dos grados de libertad (juntas rotacionales), lo que implicó modelar en forma aproximada una función de seis variables de entrada y dos de salida. Si bien los resultados obtenidos pueden considerarse muy buenos, sería recomendable ampliar las experiencias a más grados de libertad, así como a conjuntos (de testeo y de entrenamiento) de mayor cardinalidad. En cuanto al dominio sobre el que se trabajó, puede considerarse bastante cercano a un caso de la realidad.

RECONOCIMIENTO: el presente trabajo forma parte del proyecto EX157 de la Secretaría de Ciencia y Técnica de la Universidad de Buenos Aires, que es sostenido con un subsidio de esa institución.

REFERENCIAS

- [1] V. I. Arnold: "On Functions of three Variables", Dokl. Akad. Nauk SSSR, 114, 679-681 (1957).
- [2] V. Cerny; Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm"; J. Opt. Th. and Appl., vol. 45, no. 1, enero 1985, pp. 41-51.
- [3] G. Cybenko; "Approximation by superpositions of a sigmoidal function"; Math. Control Signal System, 2, 303-314, 1989.
- [4] K. Funahashi; "On the approximate realization of continuous mappings by neural networks", Neural Networks, 2, 183-92, 1989.
- [5] Y. Ito; Extension of Approximation Capability of Three Layered Neural Networks to Derivatives; Proc. IEEE Int. Conf. Neural Networks, San Francisco, 377-381, 1993.
- [6] S. Kirkpatrick, C. D. Gelatt y M. P. Vecchi; "Optimization by Simulated Annealing"; Science, vol. 220, pp. 671-680, 1983.
- [7] A. N. Kolmogorov: "On the Representation of Functions of Many Variables by Superposition of Continuous Functions of one Variable and Addition" (1957), Am. Math. Soc. Tr., 28, 55-59 (1963).
- [8] M. Leshno, V. Y. Lin, A. Pinkus y S. Schocken; Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function; Neural Networks, 6, 861-867, 1993.
- [9] A. B. Martínez, R. M. Planas y E. C. Segura; "Disposición anular de cámaras sobre un robot móvil"; Actas XVII Jornadas de Automática Santander 96, Santander, septiembre de 1996
- [10] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. H. Teller y E. Teller: "Equation of State Calculations by Fast Computing Machines", J. Chem. Phys., 21,6,1087-91 (1953).
- [11] D. E. Rumelhart, G. E. Hinton y R. J. Williams; "Learning representations by back-propagating errors"; Nature 323, 533-536 (1986).
- [12] E. C. Segura; "A non parametric method for video camera calibration using a neural network"; Proc. Int. Symp. Multi-Technology Information Processing, Hsinchu, Taiwan, diciembre de 1996.
- [13] D. A. Sprecher: "On the Structure of Continuous Functions of Several Variables", Tr. Am. Math. Soc., 115, 340-355 (1963).

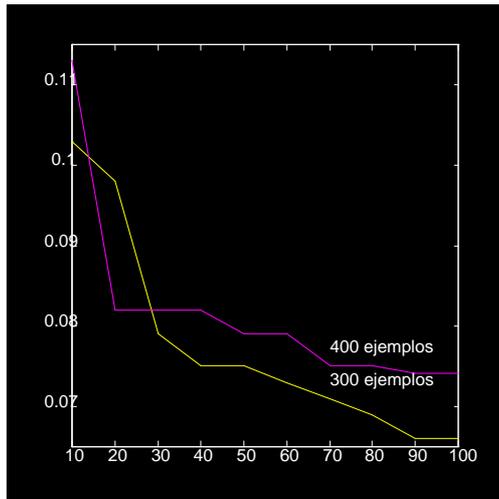


Fig. 3: Error de entrenamiento (relativo promedio) en función del tiempo (miles de iteraciones).

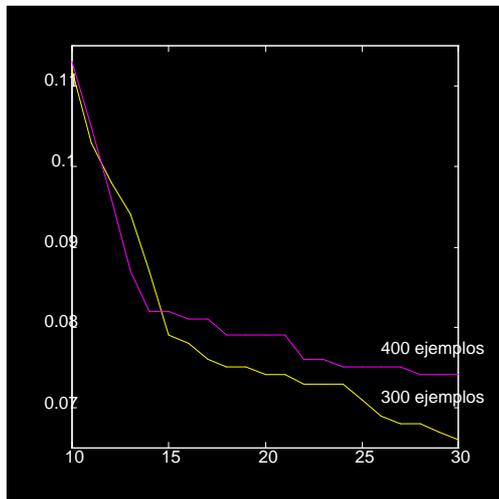


Fig. 4: error de entrenamiento (relativo promedio) en función del número de unidades ocultas.

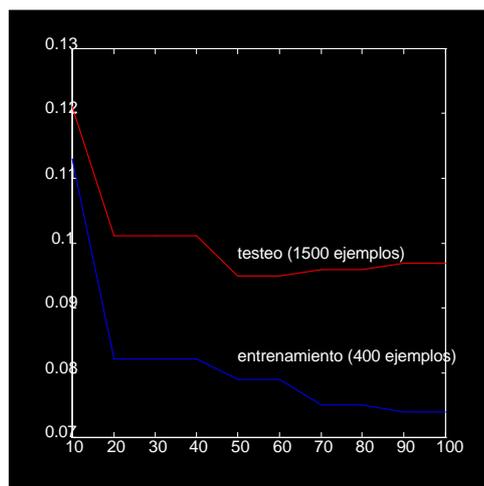


Fig. 5: errores en el entrenamiento y en el testeo en función del tiempo (miles de iteraciones).