

ISBN: 970-27-0770-6

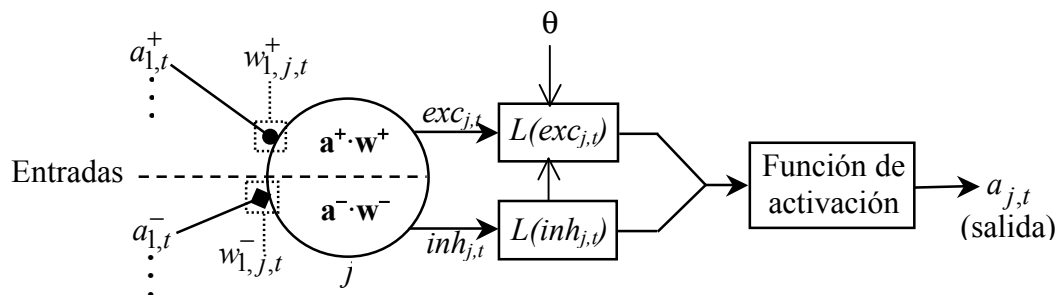
BLOQUEO EN REDES NEURALES ARTIFICIALES

Jorge Grajeda Velázquez*
 Centro Universitario de la Costa
jorgegrajeda@hotmail.com

José Enrique Burgos Triano
 Centro de Estudios e Investigaciones en Comportamiento

El bloqueo es un fenómeno de condicionamiento pavloviano que fue reportado por primera vez por Kamin (1968). El diseño básico consiste de dos grupos. El grupo experimental (*Exp*) es primero expuesto a un procedimiento pavloviano estándar anterógrado demorado, en el cual un estímulo condicionado EC1 es apareado con un estímulo incondicionado (EI). En una segunda fase, *Exp* es sometido a un procedimiento en el cual EC1 se presenta en compuesto con un nuevo estímulo condicionado EC2 y el compuesto es apareado con el EI. El grupo control (*Con*) recibe sólo el entrenamiento con el compuesto. El hallazgo principal es que la respuesta condicionada a EC2 es significativamente menor en *Exp* que en *Con*. Se dice entonces que en *Exp* la presencia de EC1 “bloquea” la adquisición de la respuesta condicionada a EC2. Este fenómeno ha sido explicado de varias formas. El objetivo de este trabajo es demostrar que puede ser simulado (y por tanto explicado) mediante el modelo neurocomputacional propuesto inicialmente por Donahoe, Burgos y Palmer (1993).

El modelo describe el elemento procesador neural (EPN), una especie de neurona abstracta que funge de unidad estructural y funcional básica de una red neural artificial. La siguiente figura muestra un EPN genérico. El EPN incluye sensores que pueden detectar señales ($a_{i,t}$) del medio ambiente local del EPN (sea externo o interno éste último constituido por señales de otros EPNs) en un momento t . Cada sensor está conectado a la unidad j con una cierta fuerza representada numéricamente por un peso ($w_{ij,t}$). Un EPN puede recibir hasta dos



tipos diferentes de sensores: excitatorios e inhibitorios. Para todo momento t , la unidad j calcula el producto interno entre el vector de señales de entrada y el vector de pesos separadamente para cada tipo de sensor. Estos productos son entonces pasados por funciones logísticas (L), sobre las cuales se calcula la activación del elemento según la siguiente función:

$$a_{j,t} = \begin{cases} L(exc_{j,t}) + \tau_j L(exc_{j,t-1})[1 - L(exc_{j,t})] - L(inh_{j,t}) & \text{si } L(exc_{j,t}) > L(inh_{j,t}) \text{ and } L(exc_{j,t}) \geq \theta_j \\ a_{j,t-1} - \kappa a_{j,t-1}(1 - a_{j,t-1}) - L(inh_{j,t}) & \text{si } L(exc_{j,t}) > L(inh_{j,t}) \text{ and } L(exc_{j,t}) < \theta_j \\ 0 & \text{si } L(exc_{j,t}) \leq L(inh_{j,t}) \end{cases}$$

donde

$$L(x) = \frac{1}{1 + e^{-\frac{x-\mu}{\sigma}}}; \quad x = \sum_{i=1}^s a_{i,t} w_{i,j,t}$$

θ es un número aleatorio según una función gaussiana con media de 0.2 y desviación de 0.15, s es el número total de entradas de cierto tipo (excitatorias o inhibitorias) conectadas a j , τ es un parámetro de sumación temporal (el cual se fijó en 0.1), κ es un parámetro de decaimiento temporal (el cual se fijó en 0.2), μ y σ son respectivamente la media y desviación de la distribución logística (las cuales se fijaron en 0.5 y 0.1, respectivamente). Toda activación oscila entre 0.0 y 1.0.

En este tipo de modelo, el aprendizaje se define como cambio en uno o más pesos. En el presente modelo, dicho cambio viene dado por la siguiente función:

$$\Delta w_{i,j,t} = \begin{cases} \alpha_j a_{j,t} d_t p_{i,t} r_{j,t} & \text{si } d_t \geq 0.001 \\ -\beta_j w_{i,j,t-1} a_{i,t} a_{j,t} & \text{si } d_t < 0.001 \end{cases}$$

donde $d_t = d_{s,t} = \phi_t + v_t(1 - v_t)$, si j es un EPN *sa* o *cal* (ver modelo de red), o

$d_t = d_{m,t} = v_t$, si j es un EPN *ma*, salida, o *vta*

$\phi_t = |\text{Media}(\mathbf{h}_t - \mathbf{h}_{t-1})|$; $v_t = \text{Media}(\mathbf{v}_t - \mathbf{v}_{t-1})$

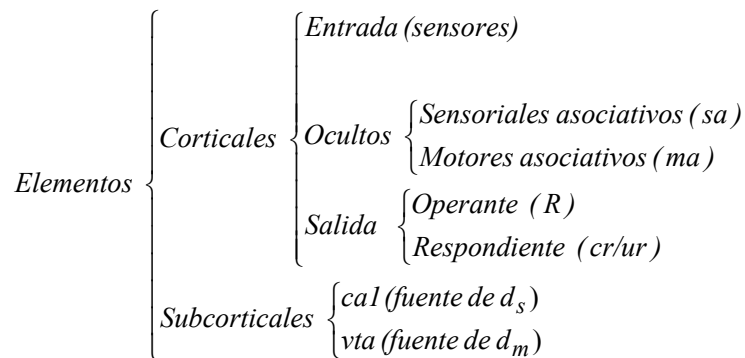
$$p_{i,t} = \frac{a_{i,t} w_{i,j,t-1}}{N}, \quad r_{j,t} = 1 - \sum_{i=1}^s w_{i,j,t}$$

$N = exc_{j,t}$ o $N = inh_{j,t}$

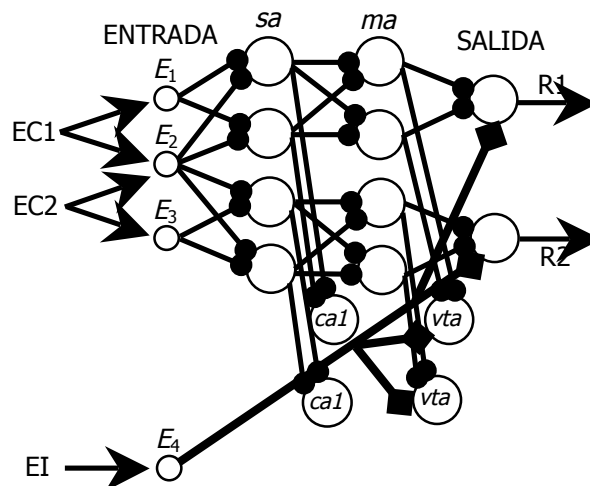
$$\alpha = 0.5\beta = 0.05$$

Todo peso oscila entre 0.0 y 1.0. Los factores p y r muestran que esta función es competitiva. Ello significa que hay una cantidad limitada de peso disponible (1.0) que pueden ganar las conexiones que coinciden en un mismo EPN. Esta cantidad se reduce a medida que las conexiones van ganando peso.

Una red neural es un sistema de EPNs conectados de cierta manera. La siguiente tabla muestra los tipos de elementos que pueden constituir una red neural en el presente modelo. Estrictamente, los elementos de entrada no son EPNs, ya que sus activaciones no son calculadas según la regla de activación. Tales activaciones son asignadas manualmente y representan la ocurrencia de estímulos en el medio ambiente de la red. Las activaciones de algunos elementos de entrada representan la ocurrencia de estímulos sensoriales como luces o tonos, comúnmente utilizados como estímulos condicionados (ECs) en investigación en condicionamiento pavloviano. La ocurrencia de estímulos como comida, agua, choques eléctricos, comúnmente utilizados como estímulos incondicionados (EIs), se representa por la activación de un elemento especialmente destinado para tal función.



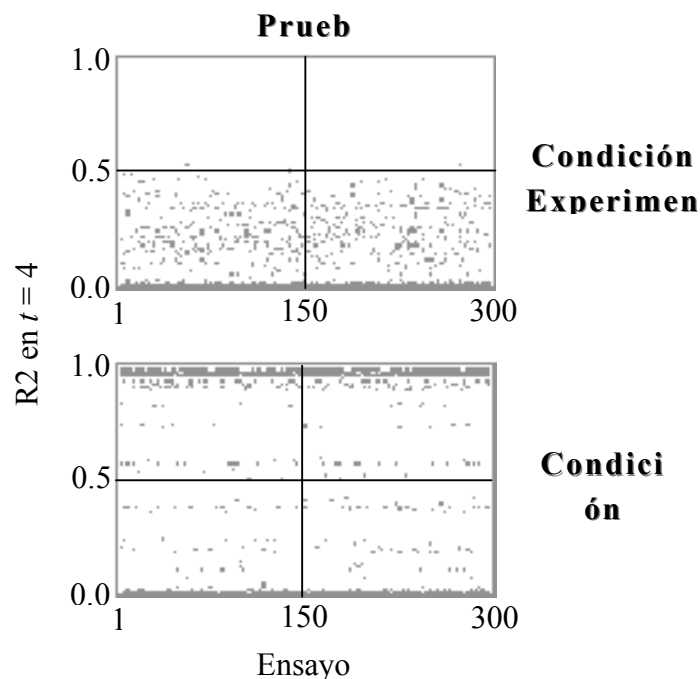
En la simulación, una red experimentalmente ingenua (i.e., con pesos iniciales de 0.01) con la siguiente arquitectura:



fue expuesta al siguiente procedimiento. En la Fase 1 se presentaron 300 ensayos de EC1 apareado con el EI (EC1+). EC1 fue definido como $a(E_1) = 1.0$ y $a(E_2) = 0.25$ por cinco momentos temporales. EI fue definido como $a(E_4) = 1.0$ en el quinto momento de EC1. En

la Fase 2, se presentaron 300 ensayos de EC2 en compuesto con EC1, donde EC2 se definió como $a(E_2) = 0.25$ y $a(E_3) = 1.0$ en los últimos cuatro momentos temporales de EC1. EC1 y EC2, entonces, fueron patrones no ortogonales de entrada, lo cual simula una cierta semejanza entre ellos. Esta semejanza representa un contexto tónico que es común a dos estímulos fásicos. Tal contexto se simuló en forma de un intervalo entre ensayos de 46 momentos temporales entre el final de un ensayo y el inicio de otro, en los cuales la presencia del contexto sólo se definió como $a(E_2) = 0.25$. Otra red con la misma arquitectura fue sometida sólo al apareamiento del EI con el compuesto EC1|EC2. Por último, ambas redes recibieron 300 ensayos de prueba de EC2 sólo en los la función de aprendizaje fue deshabilitada.

La siguiente figura muestra los resultados tal y como son presentados por el simulador.



La activación de R2 fue sustancialmente menor en la condición experimental que en la condición control, lo cual simula el efecto básico de bloqueo. El modelo explica este efecto de la siguiente manera. El apareamiento previo de EC1 con EI en la Fase 1 de la condición experimental causó un incremento significativo en los pesos de las conexiones E_1 - sa (mucho menos significativa en las conexiones E_2 - sa , debido a la baja activación de E_2). Este incremento permitió a EC1 activar de manera sustancial el EPN $vta1$, la fuente de la señal de discrepancia motora, moduladora de los incrementos de pesos. Cuando en la Fase 2 de la condición experimental se presentó EC1 en conjunción con EC2, dicha discrepancia era mucho menor en el momento de ocurrencia de EI, debido a que EC1 ya causaba una alta activación de $vta1$ en el momento inmediatamente anterior a EI. Ello resultó en un incremento poco significativo de los pesos en esta fase y, por consiguiente, de la activación de R2. En la condición control, por el contrario, las activaciones de $vta1$ en el momento inmediatamente anterior a EI fueron considerablemente más bajas, ya que el EC1 no había

sido preentrenado. Ello resultó en una discrepancia más sustancial en el momento de EI y, por tanto, un incremento más significativo en los pesos y en la activación de R2.

Referencias

- Donahoe, J. W., Burgos, J. E., & Palmer, D. C. (1993). A selectionist approach to reinforcement. *Journal of Experimental Analysis of Behavior*, 60, 17-40.
- Kamin, L. J. (1968): 'Attention-like' processes in classical conditioning. En M. R. Jones (Ed.), *Miami Symposium on the Prediction of Behavior: Aversive stimulation* (pp. 9-33). Miami: University of Miami Press.