Urdániz, Estefanía; Ruiz Quinteros, Elena

Cuando nos ponemos a pensar en la manera en la cual funciona el cerebro, su mecanismo, las correlaciones entre las neuronas, entramos en una zona de nebulosa de la cual aun no sabemos con certeza cómo opera tan rápido y con tanta precisión.

Nos propusimos desarrollar un modelo de red neuronal donde nos concentramos especialmente en la actividad neuronal (*neuronal firing*) y en la plasticidad neuronal (el cambio en la fuerza de la sinapsis), y tratamos de comprobar que el mejor mecanismo para el aprendizaje a corto tiempo se basa en la corrección de las conexiones neuronales indeseadas.

Introducción

Una red neuronal (RN) consiste en la simulación de las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos recreados mediante mecanismos artificiales. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. Cada neurona recibe una serie de entradas a través de las interconexiones y emite una salida. El objetivo es conseguir que las máquinas den respuestas similares a las que es capaz de dar el cerebro, caracterizadas por su generalización y su robustez.

Biológicamente, un cerebro aprende mediante la reorganización de las conexiones sinápticas entre las neuronas que lo componen. En la simulación de las redes, la reorganización de las conexiones sinápticas biológicas se modela mediante un mecanismo de *pesos*, que son ajustados durante la fase de aprendizaje. En una red entrenada, el conjunto de los pesos determina el *conocimiento* de la red y tiene la propiedad de resolver el problema para el cual ha sido entrenada.

Por otra parte, en la red, además de los pesos y las conexiones, cada neurona tiene asociada una función matemática denominada función de transferencia. Dicha función genera la señal de salida de la neurona a partir de las señales de entrada.

Es creído y aceptado que el aprendizaje en el cerebro reside en la plasticidad asociada con alteraciones en la eficacia sináptica. Este modelo de aprendizaje sigue las ideas de Hebb en el cual el refuerzo de las conexiones sinápticas (pesos) se dan para neuronas excitadas que han conectado los inputs y outputs de manera correcta para un dado patrón. Este proceso se conoce como *"Long term potentiation*" (LTP) y es el utilizado comúnmente para entrenar redes neuronales con ciertos patrones a través de algoritmos de "back propagation".-

En las RN que utilizan estos algoritmos, el ambiente provee una señal de error usada para corregir las conexiones sinápticas. El modelo funciona de la siguiente manera:

1. Se presenta un estímulo simple a la RN.
2. Se comparan los output de la red a los del estímulo deseado y se calcula el error para cada neurona de salida.
3. Para cada neurona, se calcula qué output sería el correcto y un factor escalar, o sea cuanto más o menos debería haber sido el output para ajustarse al deseado. Éste es el error local.
4. Se ajustan los pesos para cada neurona para bajar el error local.
5. Se asigna la "culpa" para el error local de las neuronas en el nivel anterior, dando mayor responsabilidad a las neuronas conectadas por pesos más fuertes (o sea premiar a las neuronas de la conexión correcta)

El modelo descripto arriba enfatiza el rol de potenciación de las conexiones sinápticas más fuertes como el mecanismo crucial del aprendizaje. Sin embargo, existen modelos de redes neuronales llamadas "*Long-term depression***"** (LTD), que funcionan de la manera opuesta a un LTP.

Si uno cambia las cosas de arriba para abajo y vive por un rato en el reino del revés de M. E. Walsh, se puede explicar cómo la depresión de la eficacia sináptica es el mecanismo fundamental de la mecánica del aprendizaje y adaptación.

Las diferencias fundamentales con el enfoque citado más arriba radican en que:

- Aprender de manera que sólo se refuercen las respuestas correctas es un proceso que por definición nunca se detiene; no hay una regla para la cual una vez alcanzado el objetivo no se deje de reforzar las conexiones "correctas"; el proceso, de esta manera, se vuelve un tanto "adictivo". Sin embargo, aprender corrigiendo los errores implica un proceso que se detiene cuando la meta ha sido alcanzada.
- Si un sistema adaptativo es posicionado en un nuevo ambiente o sometido a un nuevo aprendizaje, la posibilidad de que aprenda sin errores es muy baja. Es mucho más probable que se equivoque a medida que va encontrando "la" conexión correcta. De esto último se extrae que la oportunidad de moldear la sinapsis es más grande para un mecanismo adaptativo, que sólo confíe en los errores, llevando a una convergencia más rápida.

En el modelo planteado a continuación, asumimos específicamente que el área modelada en la RN recibe una señal global que solamente dice si el resultado de la acción causada por el output ha llegado a un resultado favorable o no. Esta señal es todo sobre lo que podemos actuar, el "ambiente" no nos dice qué neuronas específicas deberían haber sido disparadas o que músculos deberían haber sido activados. El ambiente actúa como un crítico y no como un maestro.

Por otro lado, asumimos que inicialmente todos los inputs del ambiente están alimentados al azar y las conexiones entre neuronas son azarosas, de igual manera que los outputs a los músculos. Por lo tanto, el sistema actúa por su cuenta, asumiendo que todas las señales de feedback para la plasticidad, están distribuidas a todas las neuronas modificándose luego usando la información que está disponible localmente.

El modelo – "Turning things upside down"

Consideramos una red neuronal de dos capas, donde *s* representan los outputs, *i* los inputs y *m* la capa media de neuronas (Figura 1). Las señales de entrada son muy simples, consisten en una sola neurona de entrada disparando, lo cual podría representar un estímulo visual mientras que las neuronas de salida podrían representar una respuesta motora.

.

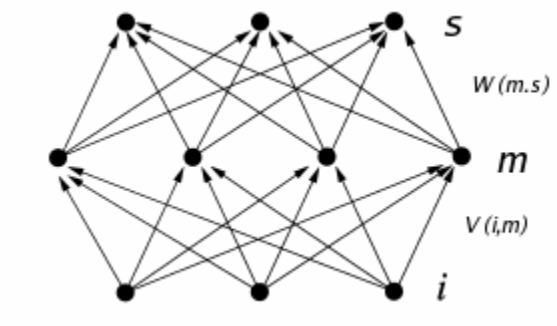

Figura 1 – Modelo de red neuronal propuesto. *i* representan los inputs, *m* la capa neuronal media y *s* los outputs. Los pesos de las interacciones entre neuronas están representadas por las matrices V y W.

Cada neurona de entrada está conectada con otra de la capa media mediante un interacción de peso V(i,m) , que representa la fuerza sináptica. Asimismo, las neuronas de la capa media se conectan con las de salida mediante interacciones de peso W(m,s). La red debe aprender a conectar cada input con un output apropiado para una situación arbitrariamente establecida.

Mediante este mecanismo, invocamos la dinámica de "winner-take-all", donde solamente una neurona *s* con el W (m,s) más fuerte responde en cada paso.

La dinámica del proceso completo es el siguiente: una neurona *i* es elegida para ser activada. La neurona *m* de la capa media con el V (i,m) más fuerte de interacción, dispara. Luego, una neurona de salida con el W(m,s) de interacción más grande recibe este estímulo y dispara. Si la neurona de salida *s* es la correcta para un dado patrón, nada pasa. De lo contrario, las dos sinapsis involucradas en esta "mala decisión", V(i,m) y W(m,s), son deprimidas en un determinado valor delta (δ). Éste es el único parámetro del modelo; sin embargo, este valor no es importante, ya que para prevenir que la fuerza sináptica decaiga por siempre, es redistribuido entre las otras sinapsis. Esta redistribución, no puede relacionarse a algún evento en particular pero puede proceder como un índice compensatorio uniforme.

La dinámica del proceso que utilizamos asegura que sólo pocas sinapsis están activas en el proceso y por lo tanto, sean más aptas a ser "castigadas" posteriormente. Esto permite que los patrones asociados con otros inputs permanezcan intactos.

Existe un proceso de selección seguido de un proceso de posible depresión lo que asegura que la actividad se propague a través de la conexión más fuerte. En el caso de resultados indeseados, las neuronas que no intervinieron en la conexión podrían llegar a ofrecer una mejor solución en el próximo estímulo. Esto es exactamente la esencia de la selección natural en el contexto de la evolución. El organismo más adaptado lo es, no porque haya proliferado intensificando intrínsicamente sus habilidades, sino simplemente porque los demás no fueron lo suficientemente fuertes.

Resultados al modelo

A partir de nuestro modelo se programó en la interfaz Matlab de acuerdo a lo establecido en la sección anterior (ver códigos en Apéndice). Se estudió en primera instancia el error en el aprendizaje en función del tiempo para una red neuronal en la cual se estimulan las neuronas N veces, siendo Ni=Ns= 2, Nm=3 . Se considera que cuando el error es cero entonces, la red "aprendió" el patrón dado o "camino correcto" según lo asignado en los códigos. En la figura 2 puede observarse que a medida que aumenta la cantidad de estímulos o señales de input, el tiempo de error disminuye.

Por otro lado, la característica fundamental de este modelo es su capacidad para recordar viejos patrones rápidamente que han sido corregidos anteriormente. Esto se ilustra en la figura 3, en la cual vemos cómo disminuye el error en el aprendizaje y los tiempos de aprendizaje una vez q se expone a la red varias veces ante el mismo estímulo. El tiempo de aprendizaje disminuye más todavía si se expone la red a un número mayor de ciclos de aprendizaje (= experimentos).

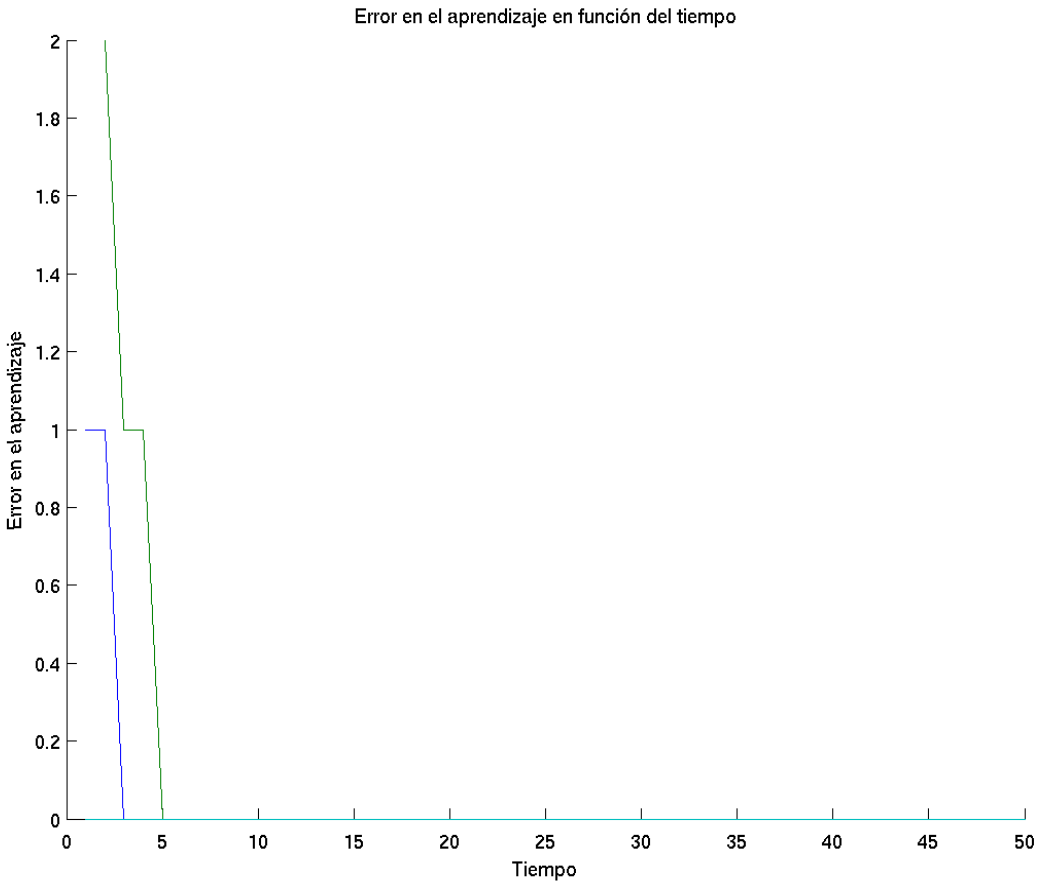

Figura 2 – Error en el aprendizaje en función del tiempo para N=1000 (azul), N=800 (verde), N=500 (celeste).

Por otro lado, nos preguntamos cómo cambia el funcionamiento de la red a medida que se varía la cantidad de neuronas de la misma: Cerebros más grandes, más inteligentes? Con este fin, variamos el número de neuronas de las tres capas y vemos que existe una marcada diferencia en los tiempos de aprendizaje cuando aumentamos la cantidad en la capa media. Esto se diferencia de otros modelos en los que a medida que la red se hace más grande, funciona peor y más despacio. Una red con una capa neuronal media extensa, ofrece más opciones al sistema para seleccionar cuando un patrón incorrecto es suprimido. Esto acelera el aprendizaje de asociaciones correctas.

Para un sistema con número constante de nueronas input y output y un número creciente en la capa media, se plotea el tiempo de aprendizaje. Puede verse en la figura 4 que el funcionamiento de la red mejora al aumentar el número de neuronas intermedias (Nm). Así, vemos que el aprendizaje es más rápido cuando el cerebro es más grande, siempre hasta cierto punto, ya que la función obtenida tiende a hacerse asintótica. Para nuestro sistema, cantidad es calidad.

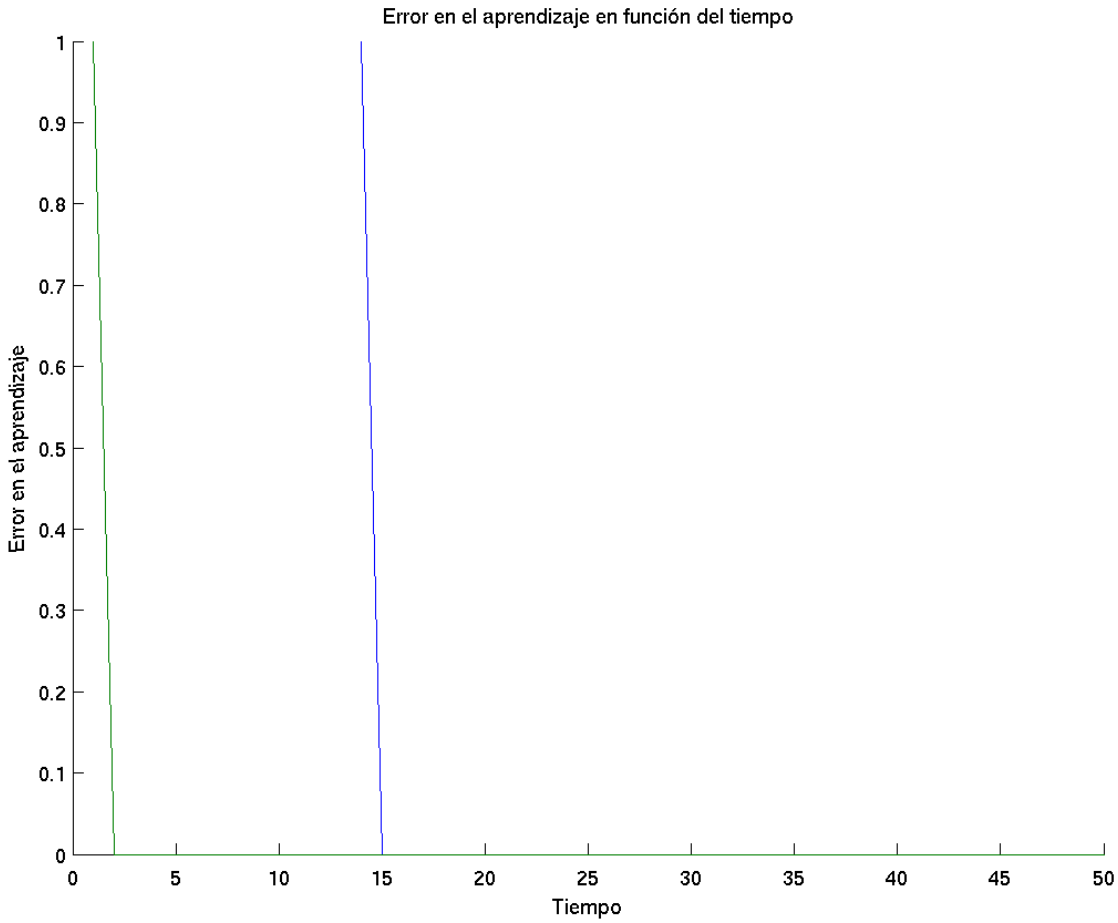

Figura 3 – Si se expone la red al mismo estímulo, el tiempo de aprendizaje la segunda vez es menor (línea verde).

Por último, probamos la gran plasticidad que brinda el sistema. Ante la presentación de distintos estímulos, la red responde más rápidamente la segunda vez que se exponen los mismos. La recuperación rápida de los patrones correctos y pasados, se intensifica más si las sinapsis que nunca han estado involucradas en outputs correctos son castigadas más severamente que aquellas que dispararon al menos una vez produciendo buenos resultados. Esta es la mayor diferencia con el enfoque hebbiano, existe una mayor capacidad de adaptación ante nuevos estímulos y la red puede "recordar" estímulos ya aprendidos. En las redes hebbianas traicionales las diferencias entre "buenas" y "malas" conexiones son siempre significativas, generándose mínimos de gran estabilidad, mientras que en nuestro modelo no. Esta característica es lo que hace a nuestro diseño una base propicia para el estudio de la memoria a corto plazo.

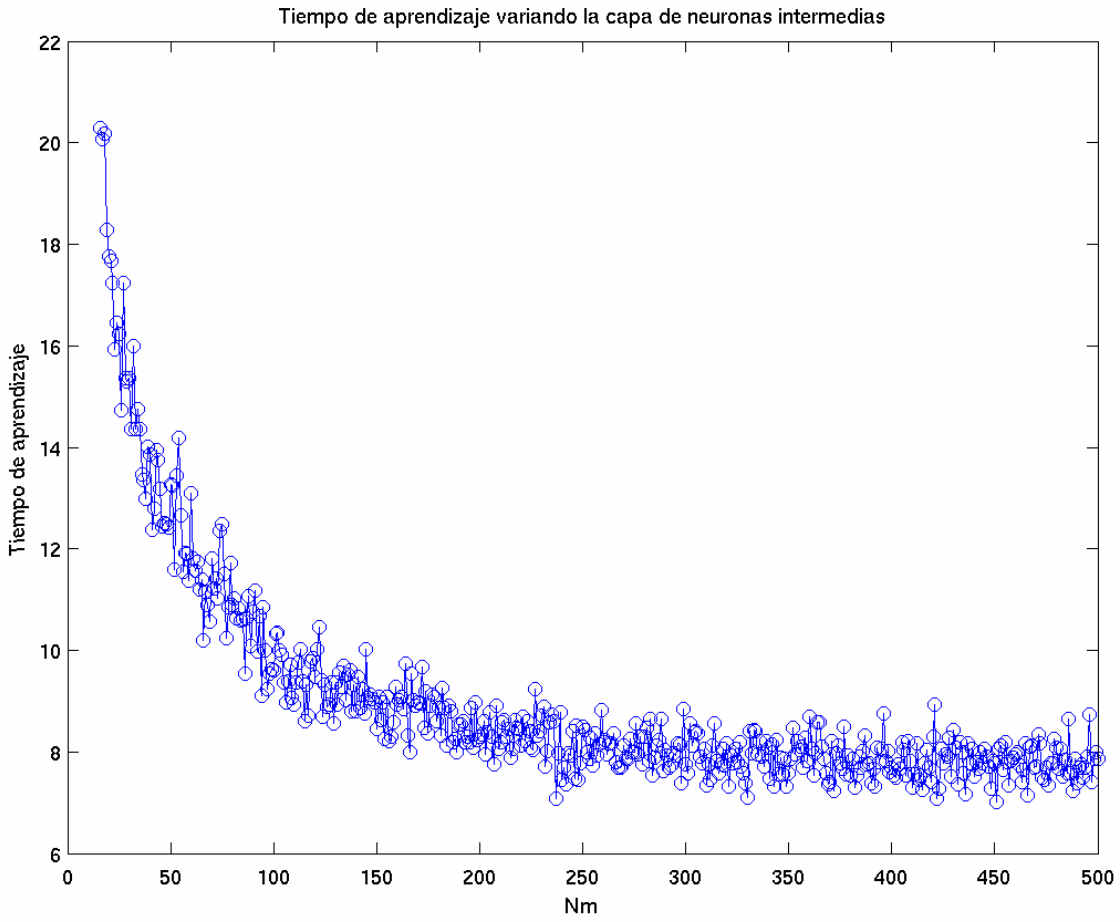

Figura 4 – El tiempo de aprendizaje disminuye a medida que aumenta el número de neuronas en la capa intermedia.

Solo una pequeña perturbación en la fuerza de la interacción es necesaria para suprimir la conexión activa ("the winner"), y las sinapsis previamente activas pueden ser activadas nuevamente de forma sencilla. Esto hace contraste con el escenario de refuerzo clásico donde, al final de un periodo preestablecido de aprendizaje, las sinapsis correctas dominan sobre las incorrectas. En LTP, la adaptación es lenta ya que nuevos aprendizajes deberán empezar desde cero y no hay memoria de los viejos patrones. De esta manera, volvemos a afirmar que el mecanismo de LTD es útil para patrones de aprendizaje no-permanente.

Conclusión

Aunque en nuestro modelo no consideramos factores fisiológicos o químicos, nos concentramos en un modelo simplista donde el foco estuvo puesto en el disparo neuronal y en la plasticidad de la red, pudimos comprobar algunos puntos:

- La red se basa en LTD. Este tipo de proceso permitió a nuestra red ganar una cierta plasticidad ante la presentación de nuevos estímulos en dos casos diferentes:

- Cuando se le presentó a la red un estímulo determinado tardaba una cierta cantidad de pasos el tener error cero, o sea aprender. Ante la presentación consecutiva del mismo estímulo los pasos que tardaba en "aprender" disminuían de manera considerable
- De la misma manera si a la red se le presentaba estímulos diferentes A,B,C, etc y luego se le volvía a presentar cualquiera de los estímulos "aprendidos", el tiempo de aprendizaje también disminuía.

Esto comprueba la plasticidad que tiene la red para aprender nuevos patrones a corto plazo. En el modelo Hebbiano (donde se utiliza LTP) que se cuestiona aquí, al premiar siempre al ganador, el proceso no termina nunca una vez alcanzada la meta. Siempre se reforzaría la conexión exitosa de manera que ésta quedaría en un pozo de estabilidad muy grande. En ese caso, la plasticidad de la red ante nuevos estímulos no permitiría a la misma conocer nuevos patrones ya que se volvería lento, requiriendo tal vez una cifra significativa de ruido en la red. Es por eso que se cree que la red mediante el proceso de LTP, reforzando conexiones, ayudaría en la memoria a largo plazo de la misma manera que el LTD lo haría a corto plazo.

- En nuestro modelo de la escala del tiempo de aprendizaje a medida que aumentábamos en la red la capa intermedia, aceleraba el tiempo de aprendizaje de las conexiones correctas (*Figura 4*). Eso se debe a que la red al tener más neuronas de interconexión entre las de entrada y salida, ofrece más opciones al sistema si la elección anterior hecha en el mismo fue la incorrecta.- En nuestro pequeño modelo cantidad es calidad (hasta cierto punto).

Bibliografía específica

- Chialvo, D.R.; Bak, P.;"Learning from mistakes"; Division of Neural Systems, Memory and Aging; University of Arizona; USA.

Anexos

Códigos de Matlab uitlizados

Básico

```
Ni = 2;
Nm = 3;
Ns = 2;

%Definimos vectores de caminos de interacción correctos entre neuronas de entrada y
%salida. Este vector tiene que tener tamaño Ni

camino_correcto = [1  2];

%Definimos matrices V y W que contendrán los pesos de las interacciones
%entre las neuronas de entrada y salida con las de la capa intermedia.

V = rand(Ni,Nm);
W = rand(Nm,Ns);
```

```
N=1000;           % N veces se estimularán las neuronas
delta=0.05;

for   n = 1:N;
      e(n) = 0;    % Determinamos que el error en el paso final es cero => Neurona "aprendió"

      for i=1:Ni;

          m = find ( V(i,:)== max( V(i,:) ) );
          s = find (W(m,:) == max ( W(m,:) ));

         if  ( camino_correcto(i) ~= s )

             V(i,m)=V(i,m)-delta;
             W(m,s) = W(m,s)-delta;

             e(n)=e(n)+1;
      end

   end

end

plot(e)
xlabel('Tiempo')
ylabel('Error de aprendizaje')
title ('Error en el aprendizaje de un patrón determinado')
```

Variando Nm

```
Ni = 4;
Ns = 4;

ind = 1;
for Nm=16:100

   Nm

 %Definimos vectores de caminos de interacción correctos entre neuronas de entrada y
   %salida. Este vector tiene que tener tamaño Ni

 camino_correcto=[1 2 3 4];

   for experimento=1:200

      %Definimos matrices V y W que contendrán los pesos de las interacciones
      %entre las neuronas de entrada y salida con las de la capa intermedia.
```

```
        V = rand(Ni,Nm);
        W = rand(Nm,Ns);

        N = 4000;      % N veces se estimularán las neuronas
        delta = 0.05;

        for n = 1:N;
           e (n) = 0;     % Determinamos que el error en el paso final es cero => Neurona "aprendió"

           for i=1:Ni;

              m = find ( V(i,:) = = max( V(i,:) ) );
              s = find (W(m,:) = = max ( W(m,:) ));

              if  ( camino_correcto(i) ~= s )
                 V(i,m) = V(i,m)-delta;
                 W(m,s) =  W(m,s)-delta;
                 e (n) = e(n)+1;
              end

           end

              if (e(n)==0)
              tiempo_aprendizaje (experimento) = n;
              break
            end
        end

        if (e(n)>0)
           tiempo_aprendizaje (experimento) = N;
        end

     end

     tiempo_medio(ind) = mean(tiempo_aprendizaje);
     tiempo_std(ind) = std(tiempo_aprendizaje);
     ind=ind+1;

  end

  plot(16:100, tiempo_medio,'o-')
  %errorbar (1:50,tiempo_medio,tiempo_std)
  xlabel ('Nm')
  ylabel ('Tiempo de aprendizaje')
  title ('Tiempo de aprendizaje variando la capa de neuronas intermedias')
```

Se fueron modificando las variables en cada código.