2.6 REDES RECURRENTES

En el contexto de las redes recurrentes existen redes dinámicas por naturaleza como lo son la red de Hopfield, la red de Jordan y la red de Elman y redes dinámicas que siendo de naturaleza estática como lo son las redes multicapa logran el comportamiento dinámico realimentando sus entradas con muestras anteriores de las salidas, el comportamiento dinámico de las redes recurrentes hace que sean una poderosa herramienta para simular e identificar sistemas dinámicos no lineales.

2.6.1 Red de Hopfield.

2.6.1.1 Antecedentes. En la década de los 80´s con el fin de estudiar procesos que involucran sistemas gobernados por ecuaciones diferenciales no lineales surge la teoría clásica de control geométrico basada en la geometría diferencial; simultáneamente renace el estudio de las Redes Neuronales debido al redescubrimiento del algoritmo Backpropagation, este hecho sumado al fracaso de las metodologías tradicionales aplicadas a la inteligencia artificial y a la disponibilidad de herramientas computacionales de bajo costo, permitieron el desarrollo las redes neuronales recurrentes cuya principal aplicación es el control e identificación de sistemas no lineales. Este desarrollo es posible debido a que las propiedades matemáticas de las redes recurrentes están enmarcadas en las mismas propiedades que fundamentan el control geométrico, la primera red

neuronal recurrente de naturaleza dinámica fue propuesta por Hopfield en 1984 bajo el contexto de las memorias asociativas.

2.6.1.2 Estructura de la red. En búsqueda de una implementación práctica, Hopfield presentó su modelo básico como un circuito eléctrico, el cual se muestra en la figura 2.6.1, donde cada neurona se representa por un amplificador operacional y una red asociada formada por una capacitancia y una resistencia, la entrada a cada amplificador es la suma de las corrientes I_i más las realimentaciones provenientes de otros amplificadores, por ejemplo el segundo amplificador realimenta al amplificador S a través de la resistencia R_{S2}, en caso de necesitarse realimentaciones con signo negativo, éstas se hacen por medio de la salida inversora de cada amplificador; la ecuación para el modelo de Hopfield basado en las leyes de Kirchhoff se muestra en la (2.6.1).

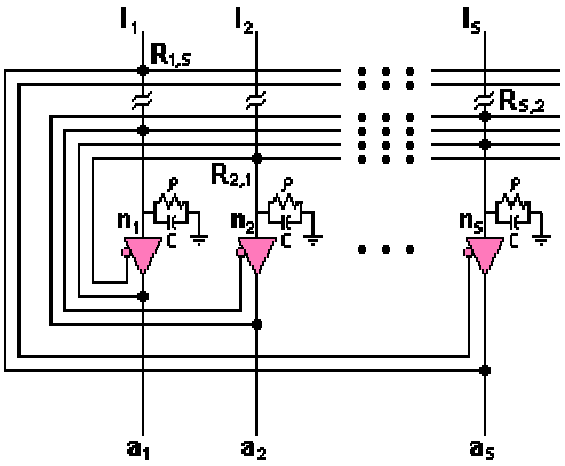

Figura 2.6.1 Circuito Eléctrico red Hopfield

$$C\frac{dn_i(t)}{dt}=\sum_{j=1}^{S}T_{ij}a_j(t)-\frac{n_i(t)}{R_i}+I_i \qquad (2.6.1)$$

Donde n_i es el voltaje de entrada a cada amplificador y $a_i = f(n_i)$ su salida, con característica de amplificación f la cual es generalmente de tipo sigmoidal, $\left|T_{ij}\right|=\frac{1}{R_{i,j}}$ y $\frac{1}{R_i}=\frac{1}{\rho}+\sum_{j=1}^{S}\frac{1}{R_{ij}}$.

Multiplicando a ambos lados de la ecuación (2.6.1) por R_i y definiendo $\in = R_i C$, $\omega_{ij}=R_i T_{ij}$ y $b_i=R_i I_i$, ésta puede reescribirse en la ecuación (2.6.2) la cual describe el comportamiento de cada una de las neuronas dinámicas que componen el circuito eléctrico de la red de Hopfield.

$$\in \frac{dn_i(t)}{dt}=-n_i(t)+\sum_{j=1}^{S}\omega_{ij}a_j+b_i \qquad (2.6.2)$$

Utilizando la ecuación (2.6.2) y escribiéndola en su forma matricial con $\boldsymbol{a}(t)=\boldsymbol{f}(\boldsymbol{n}(t))$, se obtiene (2.6.3), en esta ecuación se describe el comportamiento de la red de Hopfield

$$\in \frac{d\mathbf{n}(t)}{dt}=-\mathbf{n}(t)+W\mathbf{a}(t)+\mathbf{b} \qquad (2.6.3)$$

La red de Hopfield en notación compacta se muestra en la figura 2.6.2, en donde el vector de $\boldsymbol{p}$ no se considera como la entrada a la red sino como la condición inicial de la red

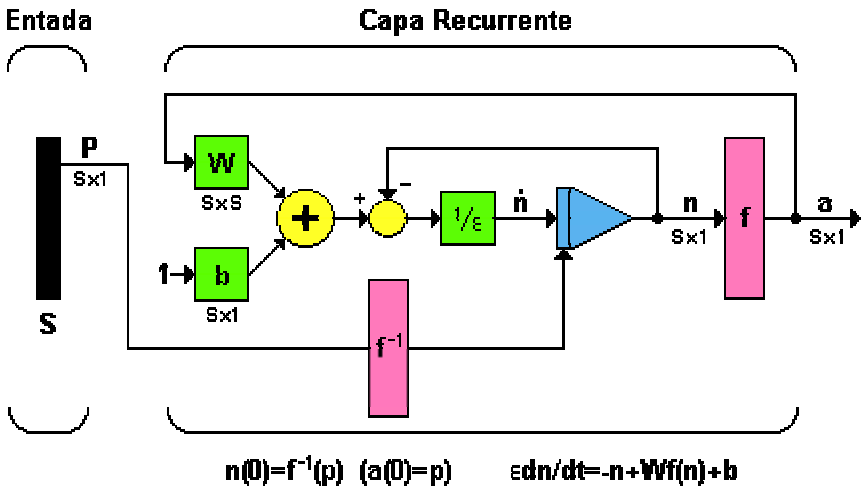

Figura 2.6.2 Notación compacta red de Hopfield

Como se observa, la red de Hopfield esta compuesta de neuronas dinámicas altamente interconectadas gobernadas por ecuaciones diferenciales no lineales, esta red funciona como una memoria asociativa no lineal que puede procesar patrones presentados de forma incompleta o con ruido, siendo útil como una poderosa herramienta de optimización

En el libro “Neural Network Design” [23], se muestra que una de las principales contribuciones de Hopfield fue la aplicación de la teoría de estabilidad de Lyapunov al análisis de las redes recurrentes, la teoría de estabilidad de Lyapunov

se aplica a través del teorema de LaSalle y para su utilización el primer paso es escoger una función de Lyapunov, para lo cual Hopfield sugirió la siguiente función:

$$V(\boldsymbol{a}) = -\frac{1}{2}\boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{a} + \sum_{i=1}^{S}\left\{\int_0^{a_i} f^{-1}(u)du\right\} - \boldsymbol{b}^{\boldsymbol{T}}\boldsymbol{a} \qquad (2.6.4)$$

Donde $\boldsymbol{a}$ es la salida de la red, $\boldsymbol{W}$ es la matriz de pesos y $\boldsymbol{b}$ es el vector de ganancias.

La escogencia de esta particular función, fue clave en el desarrollo de Hopfield, pues el primer y el tercer término de esta ecuación conforman una función cuadrática, las cuales pueden aproximar gran cantidad de funciones en un pequeño intervalo, especialmente cerca de puntos donde se encuentre un mínimo local.

Para usar el teorema de LaSalle se necesita evaluar la derivada de la ecuación 2.6.4, por claridad se evaluará cada uno de los tres términos de forma independiente, tomando la derivada del primer término de la ecuación 2.6.4 se obtiene:

$$\frac{d}{dt}\left(-\frac{1}{2}\boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{a}\right) = -\frac{1}{2}\nabla\left[\boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{a}\right]^{T}\frac{d\boldsymbol{a}}{dt} = -\left[\boldsymbol{W}\boldsymbol{a}\right]^{T}\frac{d\boldsymbol{a}}{dt} = -\boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{W}\frac{d\boldsymbol{a}}{dt} \qquad (2.6.5)$$

Derivando el segundo término de la ecuación 2.6.4, el cual consiste de una sumatoria de integrales y considerando una de estas integrales se obtiene:

$$\frac{d}{dt}\left\{\int_0^{a_i} f^{-1}(u)du\right\}=\frac{d}{da_i}\left\{\int_0^{a_i} f^{-1}(u)du\right\}\frac{da_i}{dt}=f^{-1}(a_i)\frac{da_i}{dt}=n_i\frac{da_i}{dt} \quad (2.6.6)$$

Tomando en consideración todas las integrales, en forma matricial la derivada del segundo término es:

$$\frac{d}{dt}\left[\sum_{i=1}^{S}\left\{\int_0^{a_i} f^{-1}(u)du\right\}\right]=\boldsymbol{n}^{\boldsymbol{T}}\frac{d\boldsymbol{a}}{dt} \quad (2.6.7)$$

Derivando el tercer término de la ecuación 2.6.4 y apoyándose en las propiedades de las funciones cuadráticas se obtiene la ecuación 2.6.8

$$\frac{d}{dt}\left\{-\boldsymbol{b}^{\boldsymbol{T}}\boldsymbol{a}\right\}=-\nabla\left[\boldsymbol{b}^{\boldsymbol{T}}\boldsymbol{a}\right]^{T}\frac{d\boldsymbol{a}}{dt}=-\boldsymbol{b}^{\boldsymbol{T}}\frac{d\boldsymbol{a}}{dt} \quad (2.6.8)$$

La derivada total de la ecuación 2.6.8 se obtiene al unir los resultados de las ecuaciones 2.6.5, 2.6.7 y 2.6.8

$$\frac{d}{dt}V(\boldsymbol{a})=-\boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{W}\frac{d\boldsymbol{a}}{dt}+\boldsymbol{n}^{\boldsymbol{T}}\frac{d\boldsymbol{a}}{dt}-\boldsymbol{b}^{\boldsymbol{T}}\frac{d\boldsymbol{a}}{dt}=\left[-\boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{W}+\boldsymbol{n}^{\boldsymbol{T}}-\boldsymbol{b}^{\boldsymbol{T}}\right]\frac{d\boldsymbol{a}}{dt} \quad (2.6.9)$$

comparando con la ecuación (2.6.3) del modelo eléctrico de Hopfield, se tiene que:

$$\left[-\boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{W}+\boldsymbol{n}^{\boldsymbol{T}}-\boldsymbol{b}^{\boldsymbol{T}}\right]\frac{d\boldsymbol{a}}{dt}=-\varepsilon\left[\frac{d\boldsymbol{n}(t)}{dt}\right]^{T} \qquad (2.6.10)$$

Esto permite reescribir la ecuación 2.6.9 así como sigue:

$$\frac{d}{dt}V(\boldsymbol{a})=-\varepsilon\left[\frac{d\boldsymbol{n}(t)}{dt}\right]^{T}\frac{d\boldsymbol{a}}{dt}=-\varepsilon\sum_{i=1}^{S}\left(\frac{dn_i}{dt}\right)\left(\frac{da_i}{dt}\right) \qquad (2.6.11)$$

ya que $n_i = f^{-1}(a_i)$, es posible expandir la derivada de n_i de la siguiente forma:

$$\frac{dn_i}{dt}=\frac{d}{dt}\left[f^{-1}(a_i)\right]=\frac{d}{da_i}\left[f^{-1}(a_i)\right]\frac{da_i}{dt} \qquad (2.6.12)$$

Con esto la ecuación (2.6.11) puede ser reescrita como:

$$\frac{d}{dt}V(\boldsymbol{a})=-\varepsilon\sum_{i=1}^{S}\left(\frac{dn_i}{dt}\right)\left(\frac{da_i}{dt}\right)=-\varepsilon\sum_{i=1}^{S}\left(\frac{d}{da_i}\left[f^{-1}(a_i)\right]\right)\left(\frac{da_i}{dt}\right)^{2} \qquad (2.6.13)$$

si se asume que $f^{-1}(a_i)$ es una función incremental, como sucede en los amplificadores operacionales, entonces:

$$\frac{d}{da_i}\left[f^{-1}(a_i)\right] > 0 \qquad (2.6.14)$$

Este resultado implica en la ecuación 2.6.12 que:

$$\frac{d}{dt}V(\boldsymbol{a}) \leq 0 \qquad (2.6.15)$$

De esta manera, si $f^{-1}(a_i)$ es una función incremental, todos los valores propios de la función $dV(\boldsymbol{a})/dt$ son no positivos lo cual implica que la red sea estable, entonces $V(\boldsymbol{a})$ es una función de Lyapunov válida

Los atractores de Hopfield son puntos estacionarios de la función de Lyapunov que satisfacen la ecuación (2.6.16)

$$\frac{d\boldsymbol{a}}{dt} = 0 \qquad (2.6.16)$$

Estos puntos estacionarios son puntos donde se encuentra un mínimo de la función $V(\boldsymbol{a})$ descrita en la ecuación (2.6.4), en estos puntos el gradiente de la función $V(\boldsymbol{a})$ igual a cero [21].

$$\nabla V(\boldsymbol{a}) = \left[\frac{\partial V}{\partial a_1} \quad \frac{\partial V}{\partial a_2} \quad \cdots \quad \frac{\partial V}{\partial a_S} \right]^T = 0 \qquad (2.6.17)$$

La función de Lyapunov descrita por la ecuación (2.6.4) puede simplificarse si se considera que la ganancia γ es grande, como sucede en los amplificadores con los que se implementa la red, una función de transferencia típica para estos amplificadores no lineales se muestra a continuación:

$$a = f(n) = \frac{2}{\pi} tan^{-1}\left(\frac{\gamma \pi n}{2} \right) \qquad (2.6.18)$$

Para evaluar el segundo término de la función de Lyapunov se requiere el cálculo de $f^{-1}(u)$.

$$f^{-1}(u) = \frac{2}{\gamma \pi} tan\left(\frac{\pi u}{2} \right) \qquad (2.6.19)$$

Si la ganancia γ es muy grande y la salida de la red se mantiene en el rango $1 > a > -1$, el segundo término de la función de Lyapunov tiende a cero y puede definirse la función de alta ganancia de Lyapunov como:

$$V(\boldsymbol{a}) = -\frac{1}{2} \boldsymbol{a}^T \boldsymbol{W} \boldsymbol{a} - \boldsymbol{b}^T \boldsymbol{a} \qquad (2.6.20)$$

2.6.1.3 Regla de Aprendizaje. La red de Hopfield no tiene una ley de aprendizaje asociada, esto significa que la red no es entrenada ni realiza un proceso de aprendizaje, sin embargo, es posible determinar la matriz de pesos por medio de un procedimiento basado en la función de alta ganancia de Lyapunov descrita por la ecuación 2.6.20.

$$V(\boldsymbol{a}) = -\frac{1}{2}\boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{a} - \boldsymbol{b}^{\boldsymbol{T}}\boldsymbol{a} \qquad (2.6.21)$$

El procedimiento consiste en escoger la matriz de pesos $\boldsymbol{W}$ y el vector de ganancias $\boldsymbol{b}$ tal que V toma la forma de la función que se quiere minimizar, convirtiendo el problema que se quiere resolver, en un problema de minimización cuadrática, puesto que la red de Hopfield minimizará a V

Una red de Hopfield puede diseñarse como una memoria asociativa, en este caso es llamada memoria de contenido direccionable, porque la memoria recupera la información almacenada con base en parte de su contenido, en contraste con las memorias estándar de computo, donde la información se recupera con base en sus direcciones, por ejemplo si se tiene una base de datos de contenido direccionable que contiene nombres y direcciones de los empleados de una empresa, la información completa se recupera por ejemplo suministrando el nombre (o parte de él), este tipo de memoria es la misma memoria autoasociativa

excepto que en este caso se utilizará la red recurrente de Hopfield en vez del asociador lineal estudiado en la sección 2.4.

Cuando se le presenta un patrón de entrada a la red de Hopfield, el estado inicial de la salida será el mismo patrón de entrada y luego la red convergerá al patrón prototipo almacenado que se encuentre más cercano (o que más se parezca) al patrón de entrada, para que la red memorice un patrón prototipo, este debe ser un mínimo de la función de Lyapunov

Asumiremos que los patrones prototipo son $\{p_1, p_2, \ldots, p_Q\}$ y que cada uno de estos vectores se compone de S elementos, al asumir que $Q<<S$, el espacio de estado es amplio y los patrones prototipo se encuentran bien distribuidos y por lo tanto no están cercanos uno de otro.

Para garantizar que los patrones prototipo a almacenar son mínimos de la función de Lyapunov, se propone la siguiente función para evaluar el error en la aproximación.

$$J(\boldsymbol{a}) = -\frac{1}{2}\sum_{\boldsymbol{q=1}}^{\boldsymbol{Q}} \left(\left[\boldsymbol{p_Q} \right]^T \boldsymbol{a} \right)^2 \qquad (2.6.22)$$

Si los elementos de $\boldsymbol{a}$ son restringidos a valores de ±1, la función es minimizada en los patrones prototipo como se mostrara a continuación:

Asumiendo que los patrones prototipo son ortogonales, y evaluando el error en uno de ellos, se tendrá que.

$$J(\boldsymbol{a}) = -\frac{1}{2}\sum_{\boldsymbol{q=1}}^{\boldsymbol{Q}}\left(\left[\boldsymbol{p_Q}\right]^T \boldsymbol{p_j}\right)^2 = -\frac{1}{2}\left(\left[\boldsymbol{p_Q}\right]^T \boldsymbol{p_j}\right)^2 = -\frac{S}{2} \qquad (2.6.23)$$

La segunda igualdad de la ecuación 2.6.23 se debe a la ortogonalidad de los patrones prototipo y la ultima igualdad a que todos los elementos de $\boldsymbol{p}_j$ son ±1, evaluando el error del patrón aleatorio de entrada, el cual presumiblemente no esta cercano a ningún patrón prototipo, cada elemento de la sumatoria en la ecuación (2.6.22) es el producto punto entre un patrón prototipo y la entrada, el producto punto se incrementará cuando la entrada se mueva cerca del patrón prototipo, sin embargo, si la entrada no se encuentra cerca de algún patrón prototipo, todos los términos de la sumatoria serán pequeños y por lo tanto $J(\boldsymbol{a})$ será la mayor (menos negativa) y cuando $\boldsymbol{a}$ sea igual a alguno de los patrones prototipo $J(\boldsymbol{a})$ será mas pequeña (mas negativa).

La ecuación (2.6.22) es una función cuadrática que indica con precisión el desempeño del contenido de la memoria direccionable, el próximo paso es escoger la matriz de pesos $\boldsymbol{W}$ y ganancias $\boldsymbol{b}$, tal que la función de Lyapunov de Hopfield V sea equivalente al desempeño de la función cuadrática J.

Si se utiliza la regla de aprendizaje supervisado de Hebb para calcular la matriz de pesos (con patrones objetivo iguales a los patrones de entrada)

$$\mathbf{W} = \sum_{q=1}^{Q} p_q \left(p_q\right)^T \text{ y } \boldsymbol{b}=0 \qquad (2.6.24)$$

entonces la función de Lyapunov será:

$$V(\boldsymbol{a}) = -\frac{1}{2}\boldsymbol{a}^{\boldsymbol{T}}\left[\sum_{q=1}^{Q} p_q \left(p_q\right)^T\right]\boldsymbol{a} = -\frac{1}{2}\sum_{q=1}^{Q}\boldsymbol{a}^{\boldsymbol{T}}\, p_q \left(p_q\right)^T \boldsymbol{a} \qquad (2.6.25)$$

y puede ser reescrita como:

$$V(\boldsymbol{a}) = -\frac{1}{2}\sum_{q=1}^{Q}\left[\left(p_q\right)^T \boldsymbol{a}\right]^2 = J(\boldsymbol{a}) \qquad (2.6.26)$$

Podemos observar que la función de Lyapunov es igual al desempeño del error del contenido de la memoria direccionable, la salida de la red de Hopfield tenderá a converger a los patrones prototipo almacenados, en el caso que todos los patrones prototipo sean ortogonales, cada uno será un punto de equilibrio de la red; la red puede tener muchos otros puntos de equilibrio indeseables, una regla práctica para evitarlos consiste en que cuando se utilice la regla de Hebb, el

número de patrones almacenados no debe superar el 15% del número de neuronas de la red.

2.6.1.4 Identificación de Sistemas No Lineales: El comportamiento dinámico de las redes recurrentes hace que sean una poderosa herramienta en la identificación de sistemas dinámicos no lineales.

En la forma estándar una neurona dinámica esta regida por la siguiente ecuación y se muestra en la figura 2.6.3

$$\dot{\chi}_i = -\chi_i + \sum_{j=1}^{N} \omega_{ij}\sigma(\chi_j) + \gamma_i u \qquad i = 1,\ldots,N \tag{2.6.27}$$

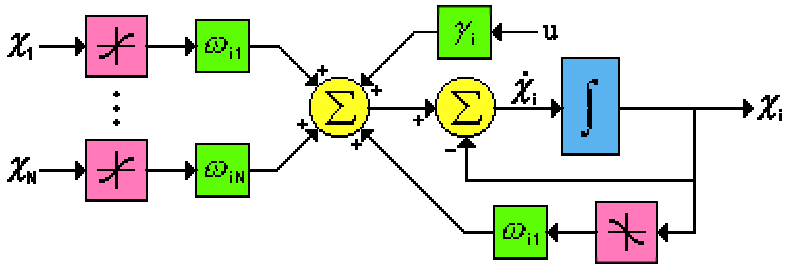

Figura 2.6.3 Neurona dinámica

o en forma matricial:

$$\dot{\chi} = A\chi + W\sigma(\chi) + \tilde{g}u \tag{2.6.28}$$

donde $A = -I$, $W = [\omega_{ij}]$, $\sigma(\chi) = [\sigma(\chi_1) \quad \cdots \quad \sigma(\chi_N)]^T$ y $\tilde{g} = [\gamma_i]$

En la figura 2.6.4 se observa una red neuronal dinámica recurrente, donde cada unidad de procesamiento es una neurona dinámica y cada punto es un peso.

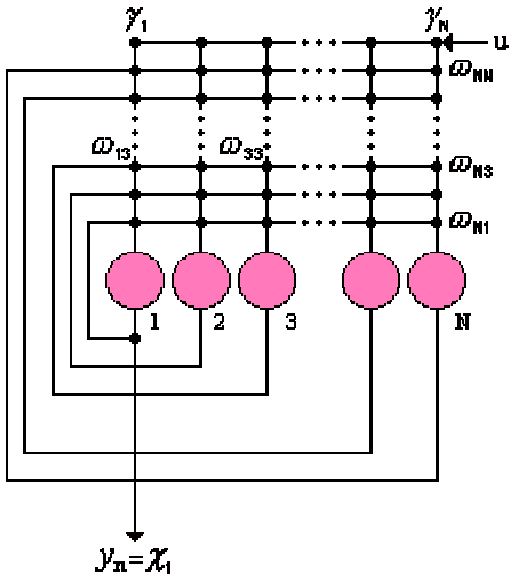

Figura 2.6.4 Red neuronal dinámica recurrente

Para garantizar la estabilidad de las redes dinámicas recurrentes en el proceso de identificación de sistemas no lineales, Delgado[9] formuló condiciones estrictas para los pesos la red y su desarrollo se basa en la función de Lyapunov.

Para el entrenamiento de la red de Hopfield en identificación de sistemas, se utiliza el algoritmo de Chemotaxis, el cual permite entrenar redes neuronales de cualquier tipo sin calcular el gradiente del error, este algoritmo fue formulado considerando el movimiento de una bacteria en un medio donde hay un gradiente

de solución alimenticia; la bacteria se mueve inicialmente al azar hasta detectar un aumento en la concentración de la solución y luego continúa en esa dirección.

El algoritmo de Chemotaxis toma los pesos iniciales al azar con distribución Gaussinana, cuando una iteración es exitosa (disminuye el valor de la función de error) el algoritmo continúa en esta dirección hasta que la función de error J no muestra cambios

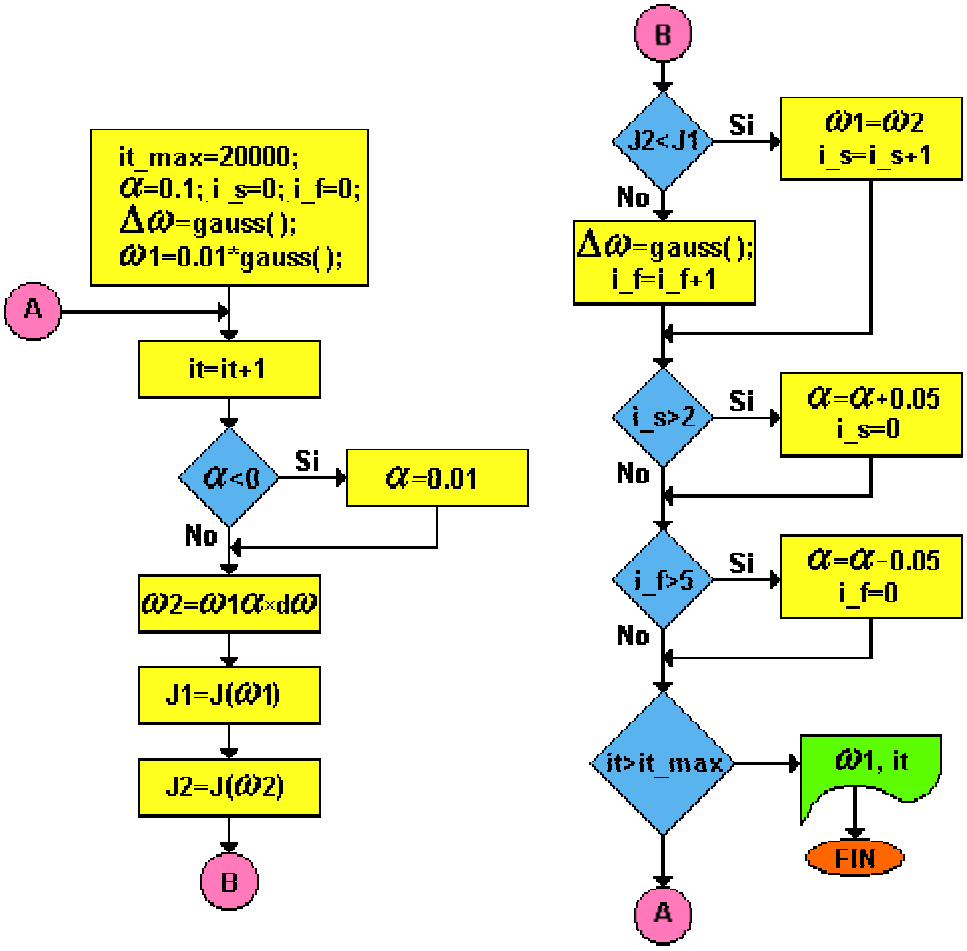

Figura 2.6.5 Algoritmo de Chemostaxis

it_max: Número máximo de iteraciones

it: Contador de iteraciones

i_s: Contador de iteraciones exitosas

α: Rata de aprendizaje

i_f: Contador de iteraciones no exitosas

$\omega 1$: Antigua matriz de pesos

$\Delta\omega$: Perturbación en la matriz de pesos

$\omega 2$: Antigua matriz de pesos

gauss( ): Generador de números aleatorios con distribución Gaussiana

Ji: Indice de la función de error correspondiente a la matriz de pesos ω_i.

La función de error Ji relaciona la salida del sistema a aproximar con la salida de la red dinámica entrenada con NP patrones de entrenamiento.

$$J = \sum_{k=1}^{NP} (d_k - y_k)^2 \qquad (2.6.29)$$

d_k: Salida deseada para el patrón de entrenamiento *k.*

y_k: Salida actual de la red ante el patrón de entrenamiento *k.*

2.6.2 Redes Multicapa

2.6.2.1 Estructura de la red. Las redes multicapa son de naturaleza estática, o sea su salida no evoluciona con el tiempo (para un patrón de entrada existe una salida asociada), pero pueden adquirir un comportamiento dinámico (para un patrón entrada la salida posee un estado transitorio y converge a un valor en el

estado estacionario) realimentando sus entradas con estados anteriores de sus salidas.

La red esta compuesta de una capa estática la cual generalmente posee un número de neuronas superior al número de variables de estado del sistema a identificar, la salida de la capa estática va a un sumador donde se le resta el valor anterior de la variable de estado Z_i identificada por el sistema, de esta operación se genera la derivada de cada una de las i variables de estado identificadas por el sistema.

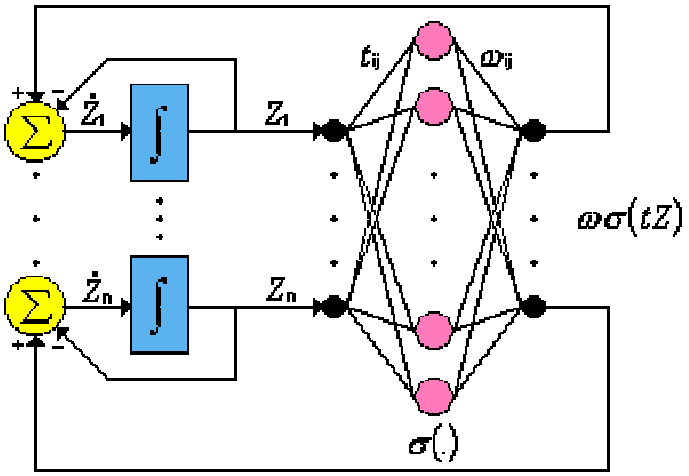

Figura 2.6.6 Red Dinámica Multicapa

La red recurrente dinámica multicapa cuyo comportamiento lo describe la ecuación (2.6.30) puede identificar el comportamiento de un sistema autónomo (u=0) descrito por la ecuación (2.6.31)

$$\frac{d}{dt} z = \overline{f}(z) = Ax + \omega\sigma(Tz) \qquad (2.6.30)$$

$$\frac{d}{dt} x = f(x) = Ax + f_o(x) \qquad (2.6.31)$$

donde $x, z \in \Re^n$, $A \in \Re^{nxn}$, $f(x): \Re^n \to \Re^n$, $\overline{f}(z): \Re^n \to \Re^n$, $W \in \Re^{nxN}$, $T \in \Re^{nxN}$, $\sigma(z) = [\sigma(z_1), \sigma(z_2), \ldots, \sigma(z_n)]$ y función de transferencia $\sigma(\theta)=tansig(\theta)$, n es el número de variables de estado del sistema y N el número de neuronas en la capa oculta.

Según Delgado[9], sin pérdida de generalidad, si el origen se asume como punto de equilibrio, el sistema (2.6.31) será identificado con la red (2.6.30) alrededor de su región de atracción y se garantiza que el error en la aproximación $e(t)$ es limitado.

2.6.2.2 Regla de Aprendizaje. La etapa estática que hace parte de la red multicapa dinámica recurrente generalmente es entrenada con el algoritmo de Chemotaxis o cualquier algoritmo de propagación inversa (Backpropagation), estos algoritmos fueron descritos en la sección 2.3, el algoritmo de Chemotaxis fue explicado en el contexto de la identificación de sistemas dinámicos por medio de la red de Hopfield donde es realmente indispensable, pues para redes dinámicas multicapa los algoritmos de propagación inversa son más eficientes y rápidos.

Los patrones de entrenamiento de la capa estática de la figura (2.6.6) son diferentes combinaciones de valores de las variables de estado y los patrones objetivo están dados por la suma de cada variable de estado con su correspondiente derivada como se muestra en la figura 2.6.7

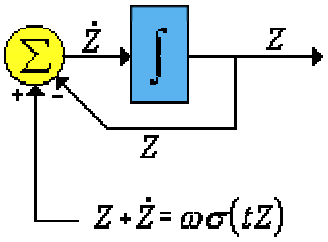

Figura 2.6.7 Patrones de entrenamiento de la red multicapa

La red después de entrenada tiene la estructura de la ecuación (2.6.32)

$$\frac{d}{dt}\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} -z_1 \\ -z_2 \\ \vdots \\ -z_n \end{bmatrix} + \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1n} \\ W_{21} & W_{22} & \cdots & W_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ W_{n1} & W_{n2} & \cdots & W_{nn} \end{bmatrix} \begin{bmatrix} \sigma(t_{11}z_1 + t_{12}z_2 + \ldots + t_{1n}z_n) \\ \sigma(t_{21}z_2 + t_{22}z_2 + \ldots + t_{2n}z_n) \\ \vdots \\ \sigma(t_{n1}z_1 + t_{n2}z_2 + \ldots + t_{nn}z_n) \end{bmatrix} \quad (2.6.32)$$

Para garantizar que la red ha identificado la dinámica del sistema, el Jacobiano de la red en el origen (2.6.33) debe tener valores propios muy cercanos a los del sistema que ha sido aproximado.

$$J_M = -I_n + WT \quad (2.6.33)$$

- Transformado una red dinámica multicapa en una red dinámica recurrente tipo Hopfield

La red dinámica multicapa de la figura (2.6.6), puede transformarse en una red dinámica tipo Hopfield por medio de la siguiente transformación lineal descrita en la ecuación (2.6.34)

$$\chi = Tz \qquad \frac{d\chi}{dt} = T\frac{dz}{dt} \tag{2.6.34}$$

Generalmente la matriz T es cuadrada, pero en caso no ser cuadrada la transformación se realiza por medio de la inversa generalizada; la red transformada tendrá la estructura (2.6.35)

$$\frac{d}{dt}\chi = -I_N\chi + TW\sigma(\chi) \tag{2.6.35}$$

donde el nuevo vector de estado $\chi \in \Re^N$, $TW \in \Re^{NxN}$, I_N es la matriz identidad de dimensión N, la transformación (2.6.34) extiende la red dinámica multicapa (2.6.32) en la red dinámica recurrente de Hopfield (2.6.35), aunque en la red de Hopfield no existen neuronas en la capa oculta el número de estados es mayor o igual al número de estados de la red multicapa N≥n

Después de realizar la transformación, la red tiene la estructura (2.6.36)

$$\frac{d}{dt}\begin{bmatrix}\chi_1\\ \chi_2\\ \vdots\\ \chi_N\end{bmatrix}=\begin{bmatrix}-\chi_1\\ -\chi_2\\ \vdots\\ -\chi_N\end{bmatrix}+[TW]\begin{bmatrix}\sigma(\chi_1)\\ \sigma(\chi_2)\\ \vdots\\ \sigma(\chi_N)\end{bmatrix} \qquad (2.6.36)$$

El Jacobiano de la red descrito en la ecuación 2.6.37 debe tener valores propios muy cercanos a los del sistema que ha sido aproximado e iguales a los de la red multicapa.

$$J_H = -I_N + TW \qquad (2.6.37)$$

2.6.3 Red de Elman

2.6.3.1 Estructura de la Red. La red de Elman típicamente posee dos capas, cada una compuesta de una red tipo Backpropagation, con la adición de una conexión de realimentación desde la salida de la capa oculta hacia la entrada de la misma capa oculta, esta realimentación permite a la red de Elman aprender a reconocer y generar patrones temporales o variantes con el tiempo.

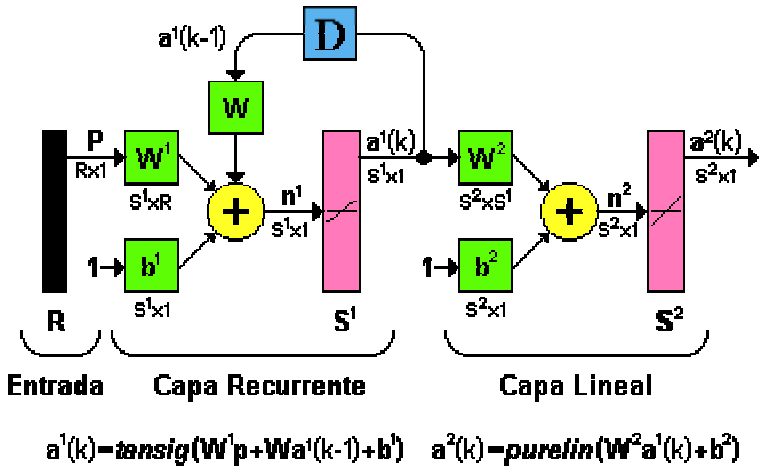

Figura 2.6.8 Red de Elman

La red de Elman generalmente posee neuronas con función transferencia sigmoidal en su capa oculta, en este caso *tansig* y neuronas con función de transferencia tipo lineal en la capa de salida, en esta caso *purelin,* la ventaja de la configuración de esta red de dos capas con este tipo de funciones de transferencia, es que según lo demostrado por Funahashi [16], puede aproximar cualquier función con la precisión deseada mientras que ésta posea un número finito de discontinuidades, para lo cual la precisión de la aproximación depende de la selección del número adecuado de neuronas en la capa oculta.

Para la red de Elman la capa oculta es la capa recurrente y el retardo en la conexión de realimentación almacena los valores de la iteración previa, los cuales serán usados en la siguiente iteración; dos redes de Elman con los mismos parámetros y entradas idénticas en las mismas iteraciones podrían producir salidas diferentes debido a que pueden presentar diferentes estados de realimentación.

2.6.3.2 Entrenamiento de la red. Debido a la estructura similar de la red de Elman con una red tipo Backpropagation, esta red puede entrenarse con cualquier algoritmo de propagación inversa como los explicados en la sección 2.3 de este capítulo, entre los cuales se destacan los algoritmos basados en técnicas de optimización como el del gradiente conjugado o el algoritmo de Levemberg Marquard.

El entrenamiento de la red puede resumirse en los siguientes pasos:

- Presentar a la red, los patrones de entrenamiento y calcular la salida de la red con los pesos iniciales, comparar la salida de la red con los patrones objetivo y generar la secuencia de error.

- Propagar inversamente el error para encontrar el gradiente del error para cada conjunto de pesos y ganancias,

- Actualizar todos los pesos y ganancias con el gradiente encontrado con base en el algoritmo de propagación inversa.

La red de Elman no es tan confiable como otros tipos de redes porque el gradiente se calcula con base en una aproximación del error, para solucionar un problema con este tipo de red se necesitan más neuronas en la capa oculta que si se solucionara el mismo problema con otro tipo de red.