SEMINARIO AADECA-UBA

“Lógica difusa, redes neuronales y control predictivo. Técnicas modernas de control”

Profesora: Dra. Doris Sáez

Apuntes II: Control basado en Redes Neuronales

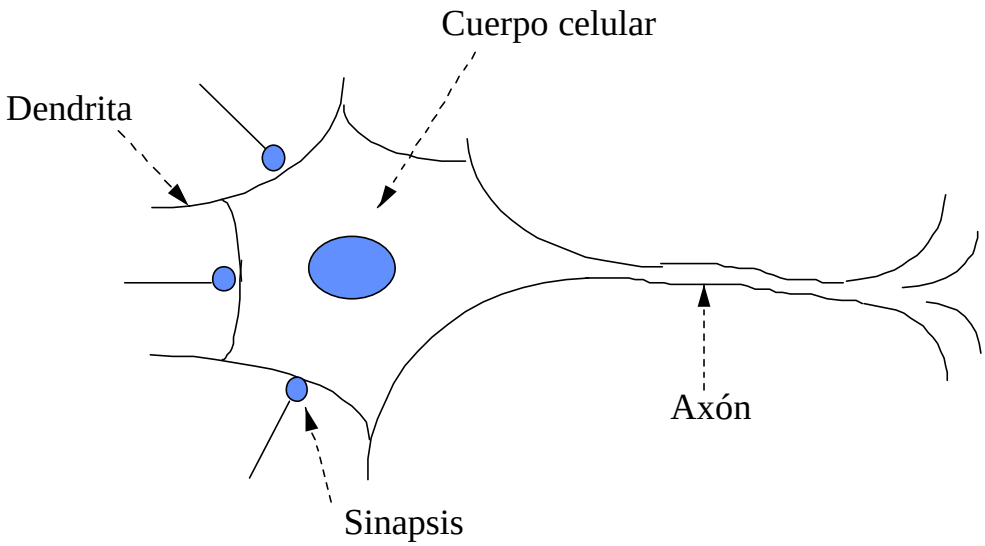
Doris Sáez (Marzo, 2002). Apuntes II. Control basado en Redes Neuronales. Seminario AADECA-UBA, Buenos Aires.

REDES NEURONALES

Las redes neuronales constituyen una poderosa herramienta para modelar sistemas, especialmente no lineales, sean dinámicos o estáticos.

El cerebro humano es una sistema muy complejo formado por muchas células llamadas neuronas; se estima que existen entre 10^{10} y 10^{11} de células en el cerebro. Las redes neuronales artificiales emulan la arquitectura y capacidades de sistemas neuronales biológicos.

Una esquema simplificado de una neurona se muestra en la siguiente figura.



Doris Sáez (Marzo, 2002). Apuntes II. Control basado en Redes Neuronales. Seminario AADECA-UBA, Buenos Aires.

En el cuerpo celular se realizan la mayoría de las funciones lógicas de la neurona. El axón es el canal de salida final de la neurona. Las dentritas reciben las señales de entrada de los axones de otras neuronas y se conectan al cuerpo celular por medio de las sinapsis.

NEURONA BIOLOGICA

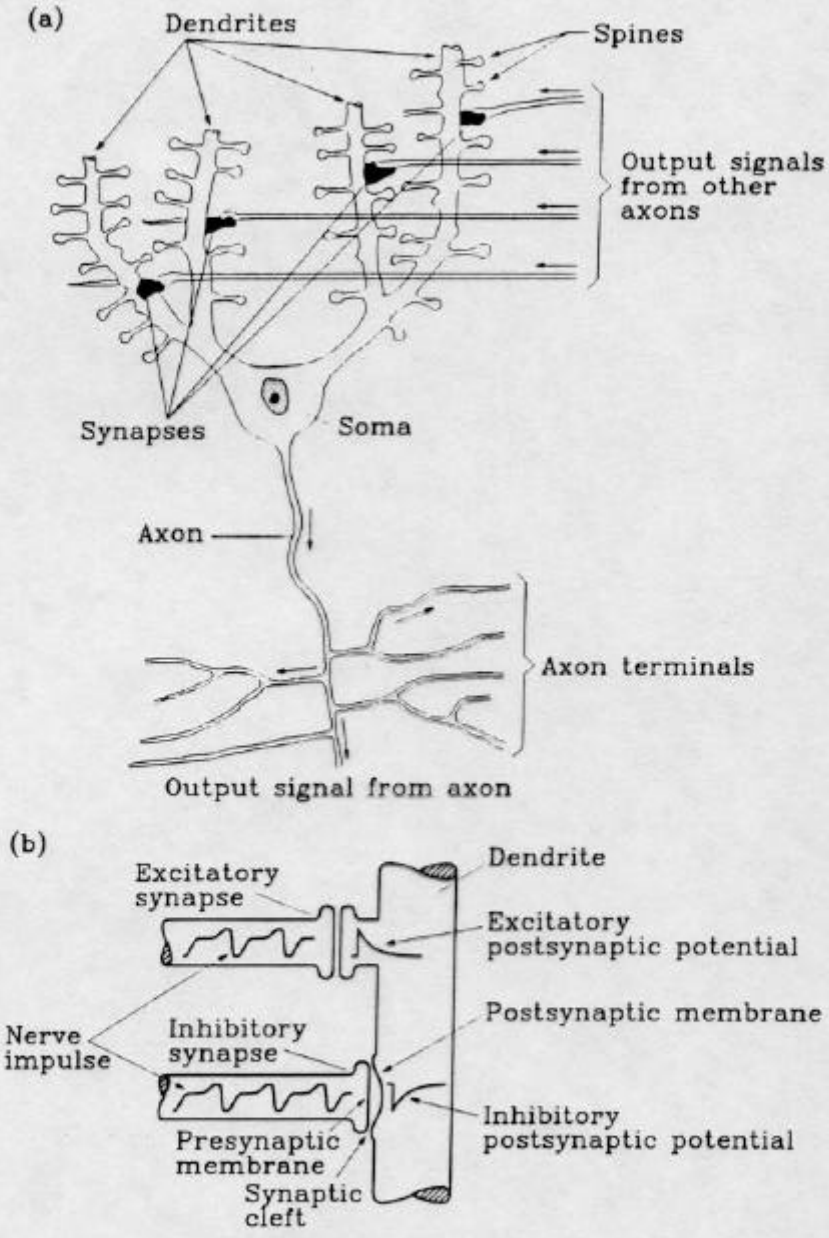
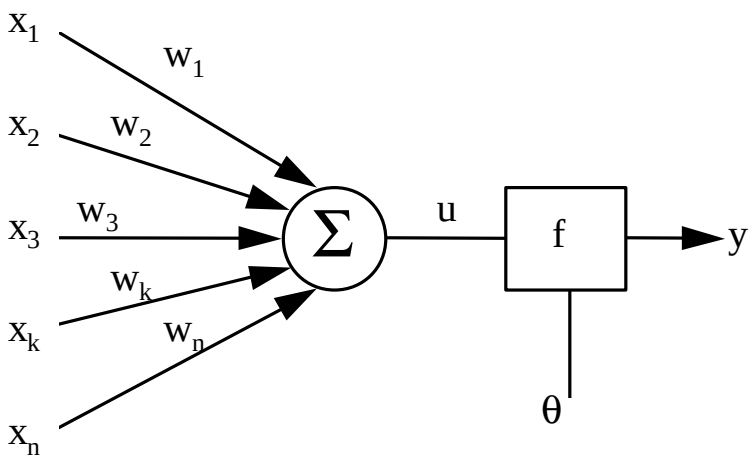


Fig. 2.1: (a) Schematic structure of a biological neuron
(b) Simplified sketch of synapses

Doris Sáe
UBA, Buenos Aires.

REPRESENTACION MATEMATICA DE UNA NEURONA

En la siguiente figura se observa la estructura de una neurona artificial con múltiples entradas.



En esta estructura, se tiene

$$u = \sum_{i=1}^{n} w_i x_i$$

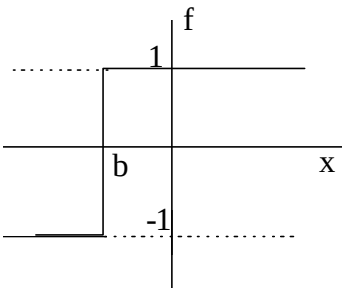
donde w_i son los pesos de la neurona (sinápsis)
x_i son las entradas a la neurona
n es el número de entradas a la neurona

Doris Sáez (Marzo, 2002). Apuntes II. Control basado en Redes Neuronales. Seminario AADECA-UBA, Buenos Aires.

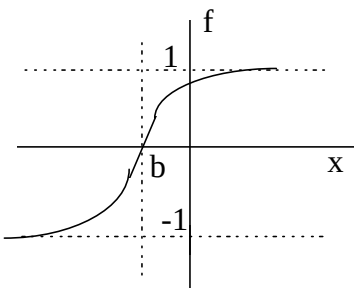
$$y = f(u) = f\left(\sum_{i=1}^{n} w_i x_i - \theta \right)$$

donde y es la salida de la neurona (axón)
f es la función de activación, correspondiente, en general, a una función no lineal (cuerpo celular)
θ es el sesgo

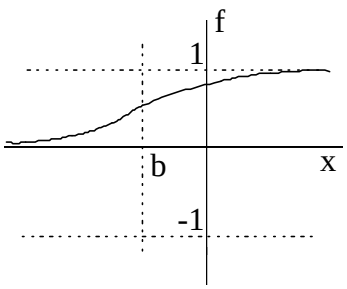
En general, se utilizan las siguientes funciones de activación:



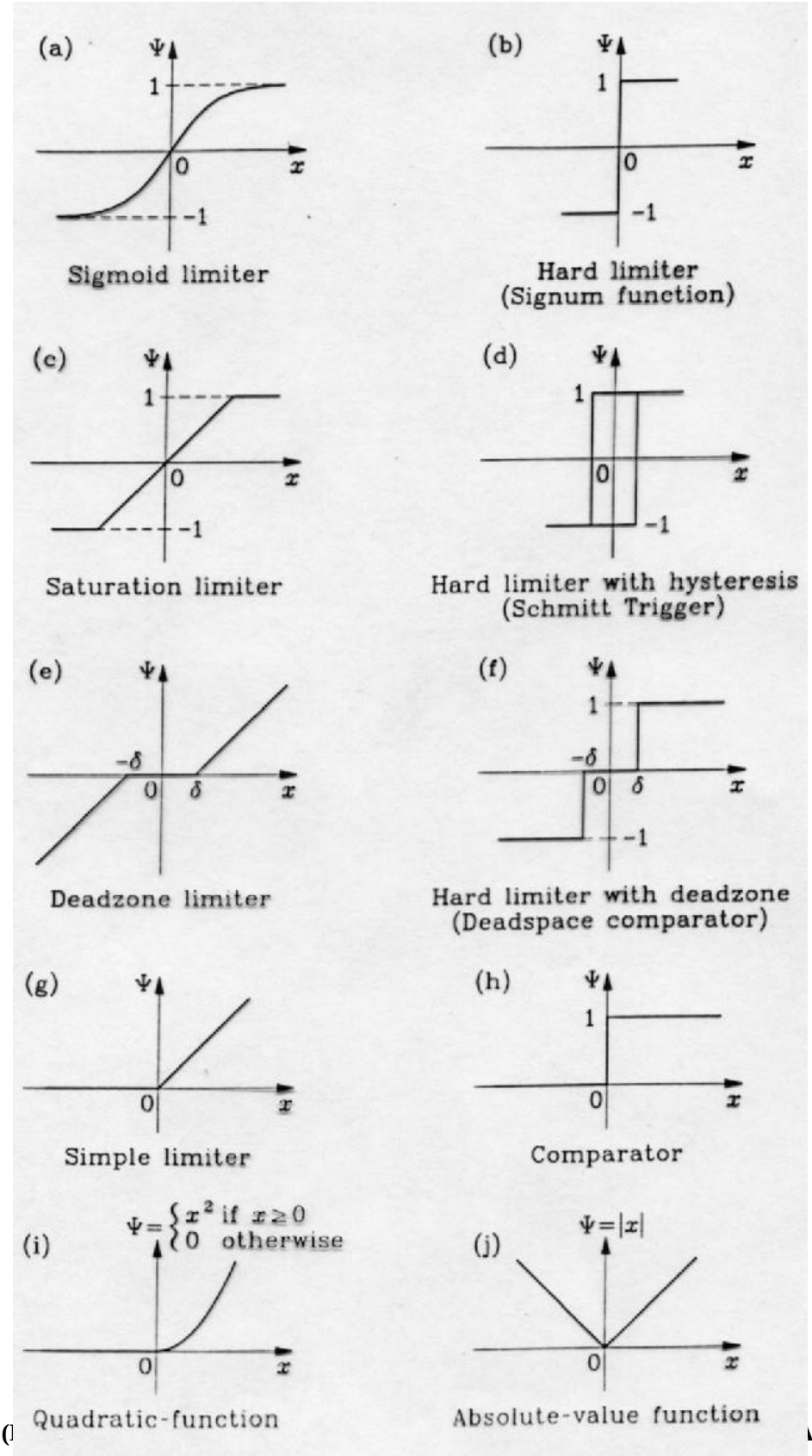
Limitador duro



Hiperbólica



Sigmoidal



Doris Sáez (I ADECA-UBA, Buenos Aires.

Las redes neuronales son estructuras de procesamiento formadas por una gran cantidad de neuronas, que operan en paralelo.

Además, los distintos tipos de redes neuronales se generan a partir de la interconexión de neuronas.

Las principales redes neuronales que se utilizan para modelación no lineal son:

- Redes perceptrón multicapa
- Redes recurrentes
- Redes de funciones de base radiales (RBFN)

Doris Sáez (Marzo, 2002). Apuntes II. Control basado en Redes Neuronales. Seminario AADECA-UBA, Buenos Aires.

VENTAJAS DE LAS REDES NEURONALES

Las redes neuronales deben su capacidad de procesamiento de información a su estructura distribuida y paralela, a su capacidad de apredizaje y por tanto de generalización.

Tareas

- Reconocimiento de patrones
- Memorias asociativas
- Aproximación funcional
- Etc.

Propiedades

- No linealidad. Las neuronas son elementos de proceso generalmente no lineales. La interconexión de estos elementos genera estructuras dde transformación de datas donde este carácter no lineal queda distribuido a lo largo y ancho de la red.

- Modelado de relaciones de entrada/salida.

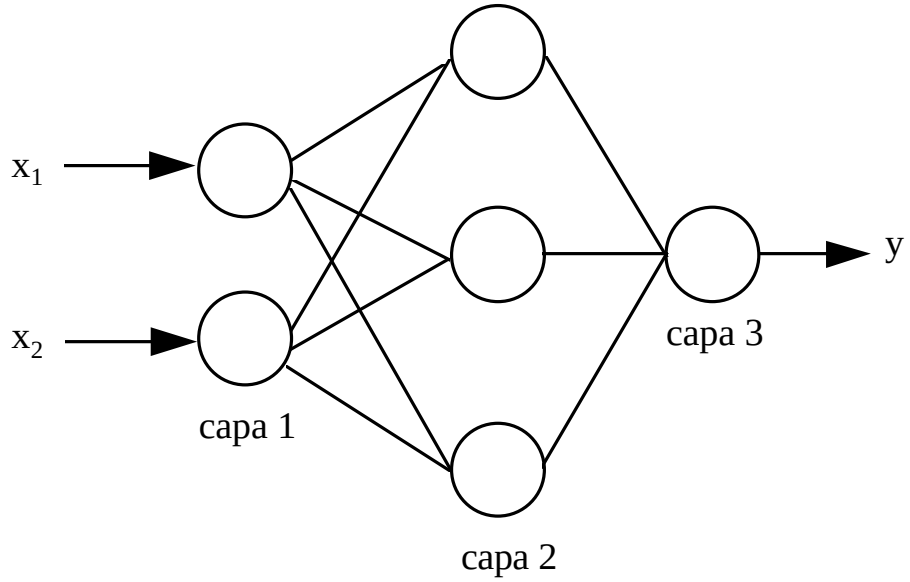
- Adaptibilidad. Las redes neuronales son por definición estructuras adaptivas capaces de ajustar sus pesos, y por

tanto su función de transferencia, a cambios en su entorno.

- Tolerancia ante fallos. Una red neuronal tiene la capacidad de seguir respondiendo de forma no catastrófica cuando parte de su estructura no está dañada. Esto es debido al tratamiento distribuido de la información y a la redundancia implícita en su estructura.

PERCEPTRÓN MULTICAPA

El perceptrón multicapa es una estructura jerárquica que consiste en varias capas de neuronas totalmente interconectadas, que admiten como entradas las salidas de los elementos de proceso (neuronas) de la capa anterior.



En las redes perceptrón multicapa se distinguen tres tipos de capas:

- Capa de entrada. Esta formada por n unidades (siendo n el número de entradas externas) que se limitan a distribuir las señales de entrada a la capa siguiente.

Doris Sáez (Marzo, 2002). Apuntes II. Control basado en Redes Neuronales. Seminario AADECA-UBA, Buenos Aires.

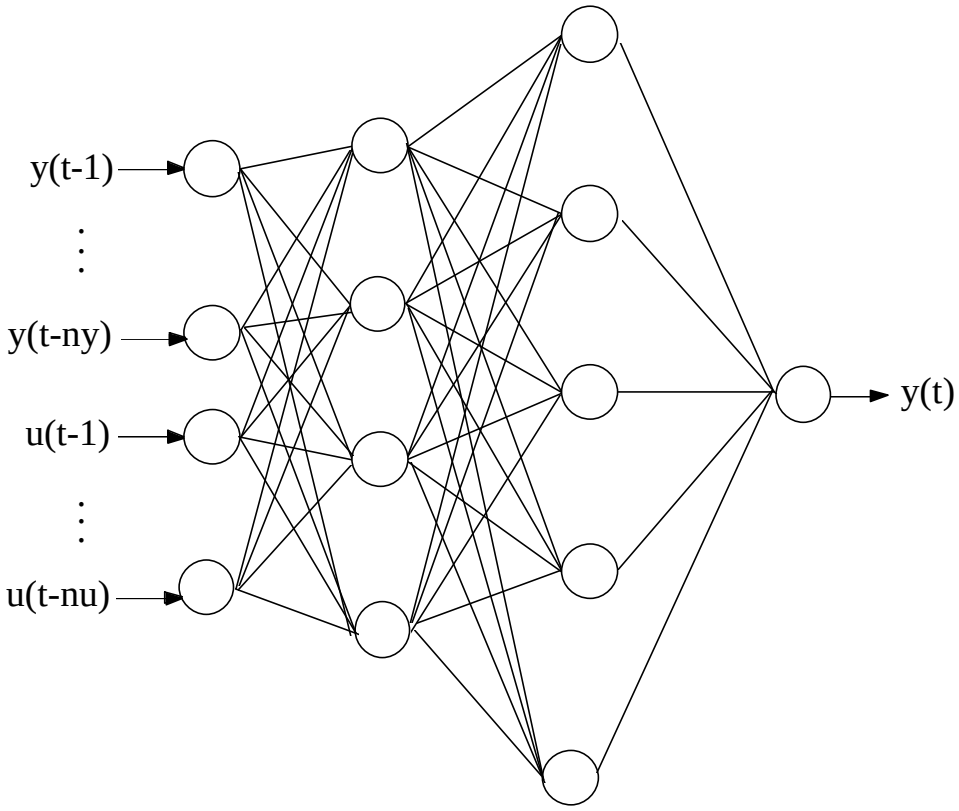
- Capas ocultas. Están formadas por neuronas que no tienen contacto físico con el exterior. El número de capas ocultas es variable, pudiendo incluso ser nulo.

- Capa de salida. Está formado por m neuronas (siendo m el número de salidas externas) cuyas salidas constituyen el vector de salidas externas del perceptrón multicapa.

Los modelos dinámicos neuronales están dados por:

$$y(t) = N(y(t-1),\ldots,y(t-ny),u(t-1),\ldots,u(t-nu))$$

donde N es la red neuronal que puede ser un perceptrón multicapa, como se muestra en la siguiente figura.



Doris Sáez (Marzo, 2002). Apuntes II. Control basado en Redes Neuronales. Seminario AADECA-UBA, Buenos Aires.

Aplicaciones

- Aproximación funcional
- Reconocimiento de patrones
- Filtrado de señales
- Eliminación de ruido
- Segmentación de imágenes y señales
- Control adaptivo
- Compresión de datos
- Etc.

Ventajas

- Capacidad de representación funcional universal. Gran rapidez de procesamiento. Genera buenas representaciones internas de las características de los datos de entrada. Ampliamente estudiada. Es la red neuronal más aplicada en la práctica

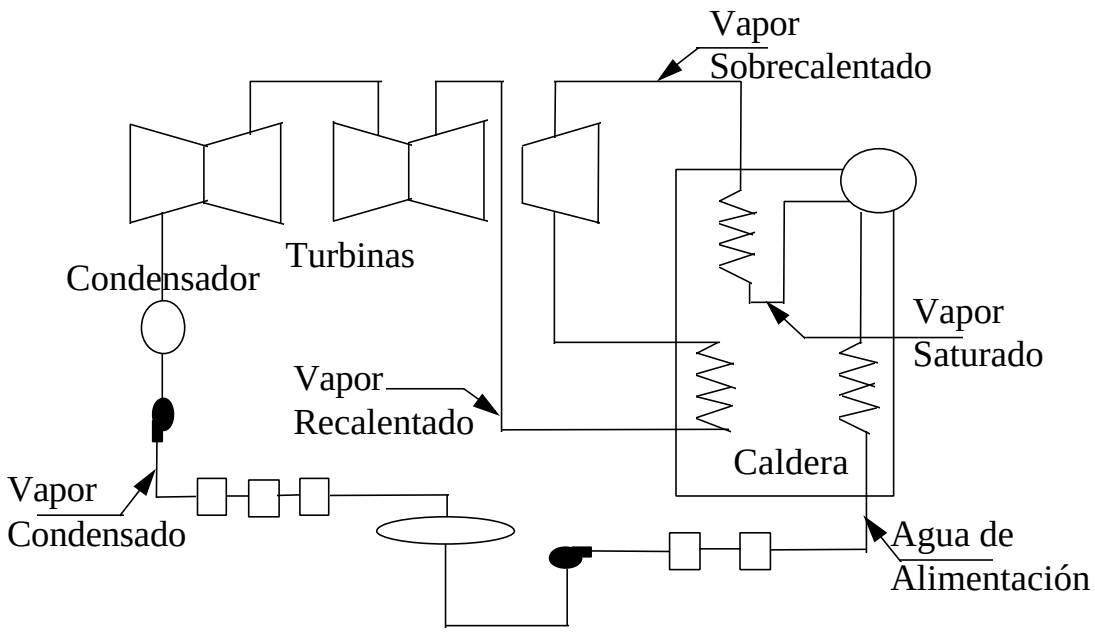
Desventajas

- Tiempo de aprendizaje elevado para estructuras complejas

Doris Sáez (Marzo, 2002). Apuntes II. Control basado en Redes Neuronales. Seminario AADECA-UBA, Buenos Aires.

Ejemplo Modelación de la química del agua de una central térmica utilizando redes neuronales.

Se considera la central térmica a carbón Anllares (350 MW), propiedad de la empresa Unión Eléctrica Fenosa (UEFSA), España. Esta central tiene en operación un sistema experto denominado SEQA que permite adquirir variables relacionadas con las propiedades químicas de los siguientes flujos del ciclo agua-vapor: vapor condensado, agua de alimentación, vapor saturado, vapor sobrecalentado y vapor recalentado.



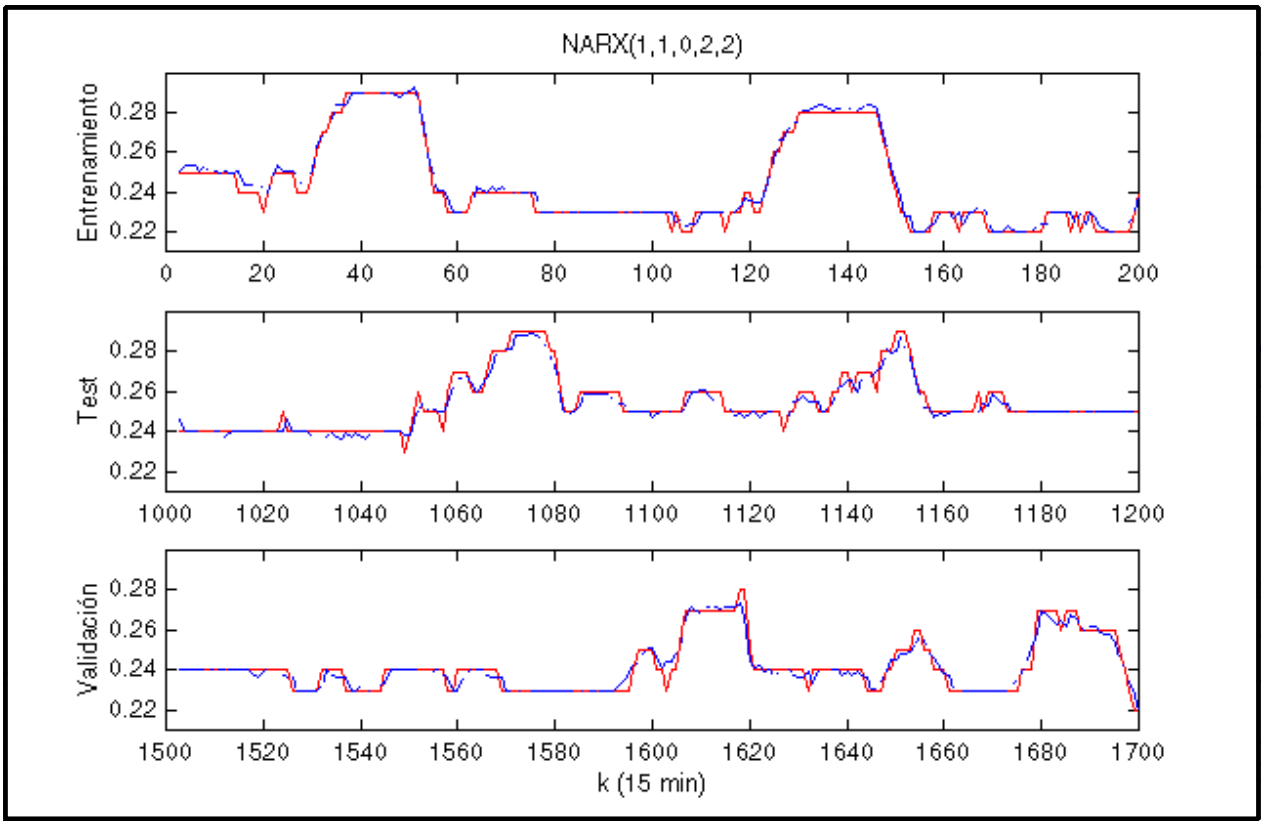
Las propiedades químicas analizadas para los flujos considerados son: la conductividad catiónica, la

conductividad específica, el pH y el porcentaje de O_2. La utilización de modelos predictivos para estas propiedades químicas, en el sistema experto SEQA, permite controlar los problemas de corrosión de componentes presentes en la producción de energía eléctrica. Especialmente, es importante la modelación de la conductividad catiónica del ciclo agua-vapor, debido a que esta propiedad es muy representativa de las impurezas del agua.
Como ejemplo de la modelación neuronal de las propiedades químicas del agua, se presentan los resultados obtenidos para la modelación de la conductividad catiónica del agua de alimentación (CC_{aa}). Las variables de entrada al modelo son: la potencia generada de la central (P) y la conductividad catiónica del condensado (CC_{cond}, flujo precedente). Los datos son adquiridos con un período de muestreo de 15 minutos.

El modelo neuronal para la conductividad catiónica del agua de alimentación está dada por:

$$CC_{aa}(k) = N(CC_{aa}(k-1), P(k-1), P(k-2), CC_{cond}(k), CC_{cond}(k-1))$$

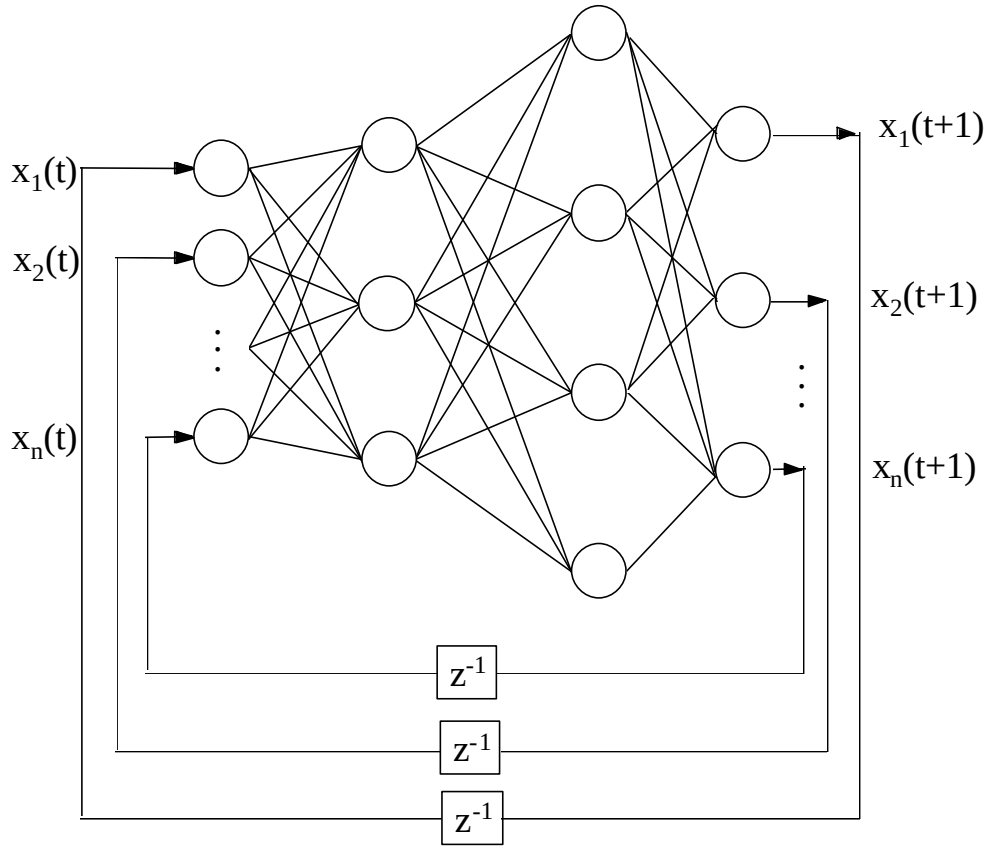
donde N es un perceptrón multicapa con una capa oculta de neuronas de funciones de activación tangente hiperbólica y una capa de salida lineal.



Doris Sáez (Marzo, 2002). Apuntes II. Control basado en Redes Neuronales. Seminario AADECA-UBA, Buenos Aires.

REDES RECURRENTES

Estos modelos son capaces de representar sistemas realimentados dinámicos no lineales (Narendra, 1990).



Además, se debe mencionar que existen diversos modelos neuronales que son combinaciones de las redes perceptrón multicapa y redes recurrentes.

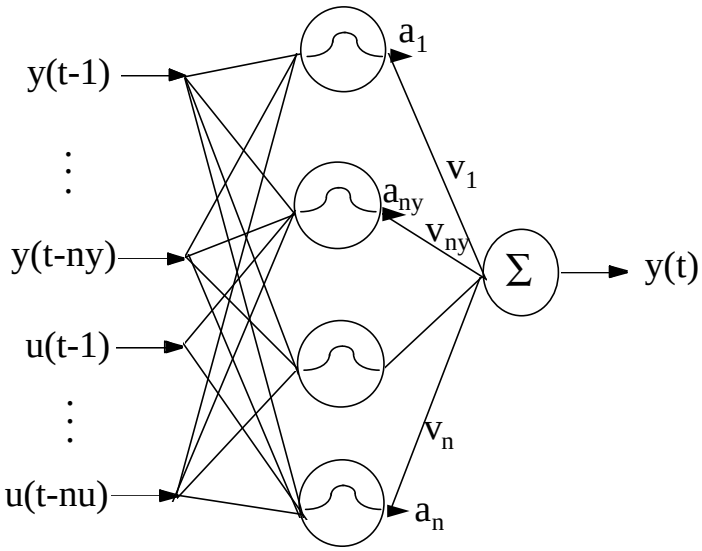
Doris Sáez (Marzo, 2002). Apuntes II. Control basado en Redes Neuronales. Seminario AADECA-UBA, Buenos Aires.

REDES DE FUNCIONES DE BASE RADIALES (RBFN)

Las redes de funciones de base radiales (RBFN "Radial Basis Function Networks") consisten en dos capas (Jang, 1993). Los modelos dinámicos basados en las redes RBFN están dados por:

$$y(t) = N(y(t-1), \ldots, y(t-ny), u(t-1), \ldots, u(t-nu))$$

donde N es una red neuronal como se muestra en la siguiente figura con n = ny + nu.



Doris Sáez (Marzo, 2002). Apuntes II. Control basado en Redes Neuronales. Seminario AADECA-UBA, Buenos Aires.

La capa oculta esta compuesta por n unidades radiales totalmente conectadas al vector de entrada. Las funciones de transferencia de la capa oculta son similares a una función de densidad gaussiana, es decir:

$$a_i = \exp\left(-\frac{\left\| x - r_i \right\|^2}{\sigma_i^2} \right)$$

donde $x = \left[y(t-1), \ldots, y(t-ny), u(t-1), \ldots u(t-nu) \right]$ es el vector de entradas de la red, r_i son los centros de las unidades radiales, σ_i representan los anchos.

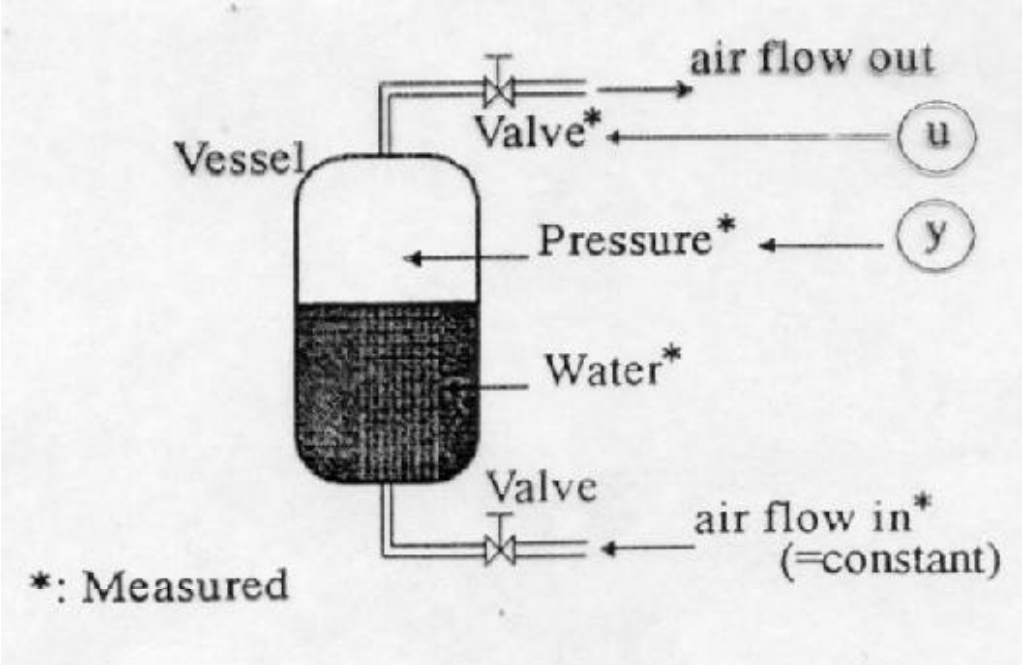
La salida de la red está dada por:

$$y(t) = \sum_{i=1}^{n} v_i a_i$$

donde v_i son los pesos de las unidades radiales.

Ejemplo Modelación neuronal basada en RBFN para un fermentador batch de alimentación.

La presión en el estanque de fermentación puede ser controlada a través del cambio de flujo de aire de salida manteniendo constante el flujo de aire de entrada.



Doris Sáez (Marzo, 2002). Apuntes II. Control basado en Redes Neuronales. Seminario AADECA-UBA, Buenos Aires.

El modelo de la red está dado por:

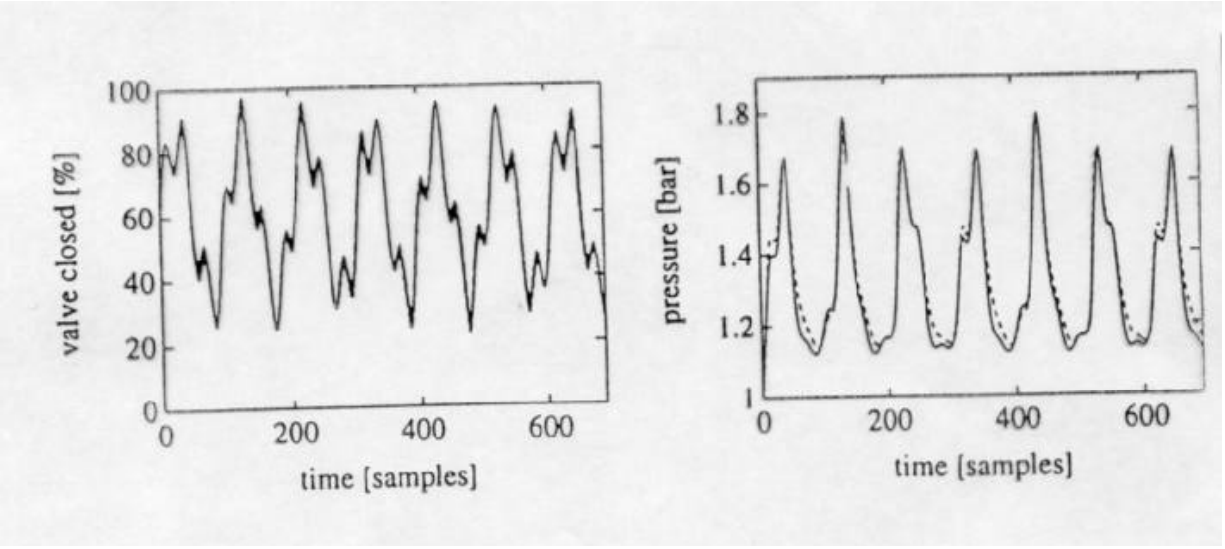
$$\hat{y}(k+1) = N\big(y(k), u(k)\big)$$

donde y(k) es la presión en el estanque y u(k) es el flujo de salida. Además, N es una red neuronal lineal/RBF dada por las siguientes ecuaciones:

$$y(k+1) = w_0 + \sum_{i=1}^{n} w_{1i}\phi_i r_i(k) + w_2^T x(k)$$

$$r_i(k) = \|x(k) - c_i\|$$

$$x(k) = \big[y(k),\ldots,y(k-n_y),u(k),\ldots,u(k-n_u)\big]^T$$



Aplicaciones

- Aproximación funcional
- Reconocimiento de patrones

Ventajas

Capacidad de representación funcional universal. La estructura de esta red tiene interpretación directa, lo que permite realizar una buena inicialización de los pesos de la red, y extraer conocimiento de las estructuras ajustadas. La buena inicialización de los pesos acelera el proceso de aprendizaje.

Desventajas

El procesamiento realizado es algo más complejo que en el caso del perceptrón multicapa.

OTROS TIPOS DE REDES

Adaline. Estas neuronas tienen capacidad de aprendizaje debido a que sus pesos son cambiados adaptivamente de acuerdo a un algoritmo adaptivo. Sus aplicaciones principales son: filtrado adaptivo de señales, reconocimiento de patrones. Son fácilmente implementables en hardware debido a su sencillez y homogeneidad, sin embargo sólo son capaces de resolver problemas de clasificación linealmente separables y llevar a cabo transformaciones lineales.

Mapas autoorganizativos de Kohonen. En este caso, las neuronas están ordenadas topológicamente. Frente a la presentación de un patrón n-dimensional de entrada, compiten lateralmente hasta que sólo una de ellas queda activa. El objetivo es que patrones de entrada con características parecidas queden asociados a neuronas topológicamente cercanas. Sus principales aplicaciones son: agrupación y representación de datos, compresión de datos y optimización.

ENTRENAMIENTO DE REDES NEURONALES

Se entiende por entrenamiento el cálculo de pesos y sesgos de manera que la red se comporte de una manera deseada. De acuerdo al tipo de entrenamiento, las redes se pueden subdividir en dos grandes grupos:

- Redes con entrenamiento supervisado. Estas redes se entrenan presentando, para cada combinación de entradas, las salidas que se espera ellas produzcan. Los algoritmos de entrenamiento calculan pesos y sesgos nuevos de manera de minimizar el error entre la salida deseada y la obtenida realmente.

- Redes sin supervisión. Los algoritmos de entrenamiento calculan nuevos pesos libremente. Estas redes se utilizan como clasificadores, pues se caracterizan por asociar una combinación de entradas especifica con una sola salida.

ALGORITMO DE ENTRENAMIENTO BACKPROPAGATION

El algoritmo de entrenamiento backpropagation se utiliza para ajustar los pesos y sesgos de un red, con el fin de minimizar la suma del cuadrado de los errores de la red.

El algoritmo backpropagation es un método iterativo de optimización de descenso según el gradiente, cuyos detalles se presentan a continuación.

Para una neurona j en una capa oculta o en la salida, la señal de salida es

$$o_j = f\left(\sum_{i=1}^{n} w_{ij} o_i - b_j \right)$$

donde f es la función de activación de la neurona

w_{ij} son los pesos de las conexiones entre la neurona considerada, j, y la neurona i, perteneciente a la capa precedente.

o_i es la salida de la neurona i de la capa precedente

b_j es el sesgo de la neurona j

En este caso, se considera funciones de activación sigmoide logarítmicas.

Doris Sáez (Marzo, 2002). Apuntes II. Control basado en Redes Neuronales. Seminario AADECA-UBA, Buenos Aires.

Además, se define

$$\text{net}_j = \sum_{i=1}^{n} w_{ij} o_i - b_j$$

La salida de la neurona j, entonces, está dada por

$$o_j = f\left(\text{net}_j\right) = \frac{1}{1+e^{-\text{net}_j}}$$

Para el entrenamiento, el valor $-b_j$ se considera como un peso correspondiente a la conexión de la neurona j con una supuesta neurona de la capa precedente cuya salida es constante e igual a uno.

El algoritmo de backpropagation permite ajustar los pesos de la red neuronal con el fin de minimizar el error cuadrático sobre un conjunto de entradas y salidas asociadas (patrones) que la red debe ser capaz de aprender para luego realizar generalizaciones a partir de ellas.

Además, se define como superficie de error a la función multivariable generada por la expresión del error de ajuste en términos de los pesos y sesgos de las neuronas de la red.

El algoritmo backpropagation permite determinar los valores de los pesos para los cuales la función de error es mínima. Esto no siempre se logra, convergiendo muchas veces el algoritmo a mínimos locales, no al mínimo global buscado, o simplemente no convergiendo.

Se considera una red con M neuronas en la capa de salida y suponiendo que se dispone de un conjunto de entrenamiento con P patrones, uno de los cuales, denominado p, tiene salidas dadas por

$$t_p = \left[t_{p1}, t_{p2}, \ldots, t_{pM} \right]$$

el error cuadrático tiene, para ese patrón, la siguiente expresión

$$E_p = \frac{1}{2} \sum_{i=1}^{M} \left(t_{pi} - o_{pi} \right)^2$$

que corresponde al error tomado para derivar la regla de optimización.

Los valores t_{pi} representan las salidas deseadas ante las entradas correspondientes al patrón p. Cuando dicho patrón es presentado a la red, los pesos se modifican según una

regla iterativa derivada del método de optimización según el gradiente, con lo cual el peso w_{ij} según la ecuación

$$w_{ij}(h) = w_{ij}(h-1) + \Delta w_{ij}(h)$$

donde h corresponde al contador dentro de una iteración. En este caso, una iteración se define como la presentación (una vez) de todos los patrones entrada/salida de los cuales se dispone para el entrenamiento.

El valor de $\Delta w_{ij}(h)$ se calcula como

$$\Delta w_{ij}(h) = \eta\left(-\frac{\partial E_p}{\partial w_{ij}}\right) = \eta\left(-\frac{\partial E_p}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}}\right) \qquad (*)$$

donde η es la tasa de aprendizaje (constante de proporcionalidad) $(0 < \eta < 1)$

En general, los pesos se inicializan entre cero y uno aleatoriamente.

Se define el parámetro δ_j como

$$\delta_j = -\frac{\partial E_p}{\partial net_j} = -\frac{\partial E_p}{\partial o_j}\frac{\partial o_j}{\partial net_j}$$

En las expresión siguientes, el subíndice p se ha omitido por simplicidad.

Para calcular las derivadas es necesario tener en cuenta que la función de activación escogida es una sigmoide logarítmica, cuya derivada es

$$\frac{df(x)}{dx} = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right) = \frac{1}{1+e^{-x}}\left(1-\frac{1}{1+e^{-x}}\right) = f(x)\left(1-f(x)\right)$$

Para una neurona j en la capa de salida se tiene, entonces,

$$\delta_j = \left(t_j - o_j\right)o_j\left(1-o_j\right)$$

Para una neurona en la capa oculta o en la capa de entrada, se tiene

$$\delta_j = o_j\left(1-o_j\right)\sum_k\left(\delta_k w_{jk}\right)$$

donde el contador k cubre las neuronas de la capa posterior a la j.

Entonces, la corrección de los pesos se comienza por la capa de salida y se propaga hacia atrás hasta llegar a la capa de entrada.

Con esto, el término (*) se puede expresar como

$$\Delta w_{ij} = \eta \delta_j o_i$$

Ahora bien, normalmente no se emplea sólo esta expresión sino que se agrega un término denominado momentum, que corresponde al cambio anterior en el peso ponderado por el coeficiente de momentum. Entonces, se tiene

$$\Delta w_{ij} = \eta \delta_j o_i + \alpha \Delta w_{ij}(h-1)$$

donde α es el coeficiente de momento. Este término permite suavizar la convergencia del método y ayuda a que la convergencia de los pesos no se vea demasiado afectada por irregularidades en la superficie de error.

Considerando los P patrones de que se dispone y con los cuales se realizará el entrenamiento, la expresión para el error total, o error de ajuste, es la siguiente

$$E = \sum_{p=1}^{P} E_p = \sum_{p=1}^{P} \left(\frac{1}{2} \sum_{i=1}^{M} \left(t_{pi} - o_{pi} \right)^2 \right)$$

En general, el entrenamiento se considera acabado cuando el valor de E es menor o igual que un límite preestablecido.