

Optimización de Motores de Inferencia Conexionistas Mediante Algoritmos de Poda y Lógica Difusa

José Talavera-Herrera, Omar Vivas-Arapa
Escuela Profesional de Ingeniería de Sistemas
Universidad Nacional de San Agustín
Arequipa, Perú
{jtalaverahe,ovivas}@unsa.edu.pe

and

Ernesto Cuadros-Vargas
Instituto de Ciências Matemática e de Computação
Universidade São Paulo-Sao Carlos-SP, Brasil
ecuadros@icmc.usp.br

Abstract

In the design of connectionist inference motors based on Artificial Neural Networks Back-Propagation, to determine the number of nodes of the hidden layer is a problem of great importance, due to their serious influence in the learning process. Therefore it should be had a optimal structure neural that generates appropriate and satisfactory results for the user. To get it, the to experiment with several sizes of nets and to choose the best of them, it can consume too much time and effort. This work presents the automation of this process, using the foundations of Algorithms of Pruning; based on the Singular Value Decomposition, that allows to calculate the utility of the neurons to the end of the learning process, and the Fuzzy Logic applied to the control of the elimination of the nodes in function of the calculated utility, being able to obtain compressed structures of optimal neural generalization.

Keywords: Decomposition of Values Singular, Algorithm of Pruning, Fuzzy Logic, Neural Networks.

Resumen

En el diseño de motores de inferencia conexionistas basados en Redes Neuronales Artificiales Back-Propagation, determinar el número de nodos de la capa oculta es un problema de gran importancia, debido a su influencia seria en el proceso de aprendizaje. Por lo tanto se debe contar con una estructura neuronal óptima que genere resultados adecuados y satisfactorios para el usuario. Para conseguirlo, el experimentar con varios tamaños de redes y escoger la mejor de ellas, puede consumir demasiado tiempo y esfuerzo. Este trabajo presenta la automatización de este proceso, utilizando los fundamentos de Algoritmos de Poda; basados en la Descomposición de Valores Singulares, que permite calcular la utilidad de los nodos al final del proceso de aprendizaje, y la Lógica Difusa aplicada al control de la eliminación de los nodos en función de la utilidad calculada, logrando obtener estructuras compactadas de optima generalización neuronal.

Palabras claves: Descomposición de Valores Singulares, Algoritmo de Poda, Lógica Difusa, Red Neuronal.

1 Introducción

La Red Neuronal Artificial (RNA), también llamada procesador paralelo distribuido o arquitectura conexionista, es una herramienta de procesamiento de información inspirada en la estructura paralela del cerebro. Las RNAs en general son colecciones de modelos matemáticos que emulan algunas de las propiedades observadas en los sistemas nerviosos biológicos. Una típica RNA, está compuesta por un gran número de elementos de procesamiento, similar a la cantidad de neuronas en el cerebro, que se unen a través de conexiones (pesos) que hacen referencia a la sinapsis.

A pesar de muchos adelantos, aún se debe de contestar varias preguntas antes de poder generar RNAs aplicadas a los problemas del mundo real. Responder a la pregunta *¿Cuántos nodos tendrá la capa oculta?*, es el paso más importante para determinar una RNA satisfactoria, pero también es uno de los problemas que no se ha resuelto. Esta cantidad mencionada anteriormente, es relacionada generalmente con el *tamaño de la red neuronal*.

Las redes grandes y pequeñas presentan ventajas. Cuando una red tiene demasiados parámetros (es decir pesos y /o elementos) no sólo aprende rápidamente, también evitan los mínimos locales de manera más fácil [2]. Sin embargo, tal red neuronal podría dar resultados inexactos cuando se introduce patrones fuera del conjunto de datos. En el caso donde haya sólo algunos grados de variación, se espera que la red se adapte a las regularidades de los datos, ignorando los más pequeños o insignificante. Esta *capacidad de generalización* es sumamente importante para problemas con dominios grandes, continuos o discretos, donde es imposible generar una muestra satisfactoria para cada patrón de entrada [13]. También en el caso de la *limitación de los recursos computacionales*, pequeñas RNAs serian menos costosas en su construcción. Debido a este dilema, conocer el tamaño óptimo de la RNA, se vuelve un problema importante. Probar varios tamaños de la red podría ser una solución, pero este proceso a menudo requiere mucho tiempo.

El trabajo realizados por Psychogios y Ungar [10], propone un algoritmo que permite obtener el número óptimo de nodos ocultos, utiliza la Descomposición de Valores Singulares (DVS) para calcular los pesos de la capa oculta a la capa de salida de la RNA y a la vez calcular los nodos redundantes, obteniendo óptimos resultados. Y el trabajo realizado por [13], basado en el trabajo de Psychogios y Ungar, proponen usar los valores de salida de los nodos de la capa oculta como parámetros de medición de utilidad en el aprendizaje, y poder eliminar los nodos más redundantes como resultado de aplicar la DVS. El presente trabajo pretende optimizar el algoritmo propuesto por [13], en el proceso de eliminación de nodos en la capa oculta, utilizando lógica difusa para dar una mayor flexibilidad al proceso y poder identificar el tamaño óptimo de la RNA, algoritmo que lo hemos titulado Poda Difusa.

Este trabajo está estructurado de la siguiente manera: en la sección 2 se describe brevemente una visión general de los algoritmos de poda. En la sección 3 se expone el criterio de optimización de RNAs utilizado en el trabajo de [13], la sección 4 se explica los fundamentos de DVS, en la sección 5 se describe el algoritmo de poda difusa propuesto para la optimización de RNAs, en la sección 6 se explica el papel de la lógica difusa en el algoritmo, en la sección 7 se define el caso de estudio y se realizan las pruebas pertinentes, y finalmente en la sección 8 se muestran las conclusiones.

2 Algoritmos de Poda

Aunque la intuición parece muy certera para probar un número mínimo de nodos ocultos como primer paso, y luego proceder con otro más grande. Es un hecho que las redes pequeñas y factibles sean sensibles a condiciones iniciales y a los parámetros de aprendizaje, y más probables de ser capturadas en mínimos locales [12]. Por lo tanto es conveniente empezar con una red grande y podarla paulatinamente, es decir ir suprimiendo un número apropiado de nodos ocultos, hasta lograr un sistema más estable con una mejor generalización.

Muchos de los algoritmos de poda se pueden clasificar en dos grupos. Un grupo de estimación de la sensibilidad, donde los elementos con menor efecto son removidos. Usualmente se asume que cuando se lleva a cabo este método la sensibilidad de los pesos no cambia cuando un elemento es removido; note que esto no puede ser válido, considerando la naturaleza no lineal de las RNAs [13].

Un buen acercamiento puede calcular la sensibilidad de los pesos y aplicar iterativamente la poda, para permitir que la red reaccione dinámicamente a los cambios [1].

El otro grupo agrega parámetros adicionales a la función objetivo que premia a la red para ser escogida como una solución eficaz; por ejemplo, una condición proporcional al número de nodos ocultos favorece soluciones con redes relativamente pequeñas. Una revisión extensa de los métodos que entran en ambas categorías puede encontrarse en el estudio realizado por Reed [12].

Para la mayoría de los algoritmos de poda propuestos, la optimización que lleva la supresión de las unidades ocultas es a menudo más efectiva que la introducción de parámetros adicionales, parámetros que son normalmente específicos para un cierto conjunto de datos. El algoritmo utilizado tomara esta idea como fundamento de eliminación.

3 Criterio de Optimización de una RNA Back-Propagation

La red neuronal back-propagation, conocida también como perceptrón multicapa por su *alimentación hacia adelante*, es una clase de red neuronal importante por su simple topología y su poderosa capacidad de generalización.

Las fases de aprendizaje incluyen una técnica de búsqueda de gradiente para minimizar la función de costo, que es igual a la diferencia de cuadrados entre la salida deseada y los valores reales obtenidos por la red. La red es entrenada, primero inicializando sus pesos con valores pequeños, seleccionados en forma aleatoria, y luego presentando repetidamente todos los datos de entrenamiento. Los pesos son ajustados “en línea”, es decir, después de cada presentación de los datos de entrenamiento, hasta que los pesos converjan y la función de costo sea reducida a un valor aceptable. El aprendizaje back-propagation puede ser realizado con sólo una capa sin perder generalidad [4], ya que redes con más de una capa oculta generan regiones de decisión arbitrariamente complejas [7].

El problema principal al diseñar una RNA es determinar cuantos nodos ocultos son necesarios. Un criterio de optimización es buscar aquellos valores de salida insignificantes de los nodos ocultos al finalizar el proceso de aprendizaje, innecesarios en este proceso [13].

Podemos organizar la información de los valores de salidas de los nodos ocultos en una matriz de N columnas (#nodos ocultos) y M filas (#patrones a aprender), de esta manera podemos analizar los valores de salida asociados a cada neurona y en conjunto decidir si esa neurona es influyente en el aprendizaje. Este proceso se observa en la figura 1.

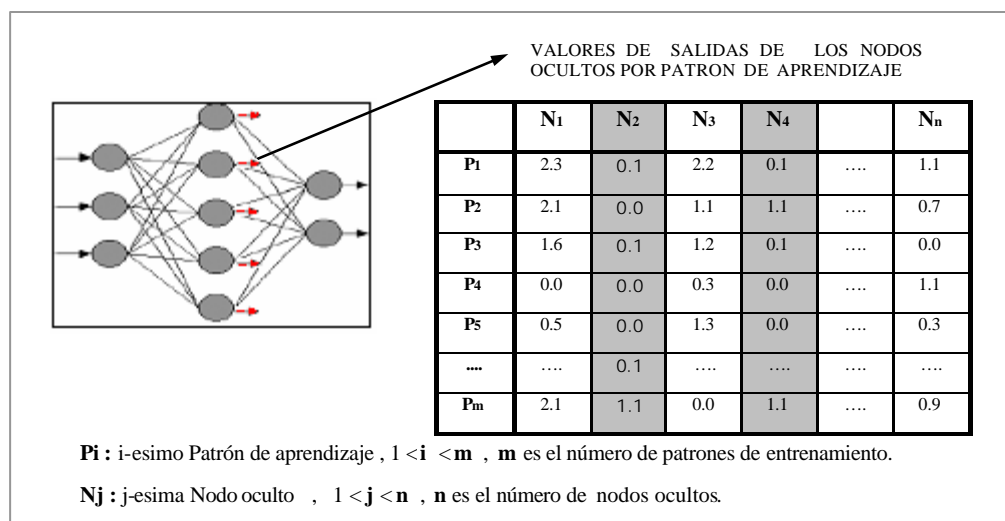


Fig.1. Red Neuronal Back-Propagation con 3 capas

La matriz describe la participación de cada nodo oculto por cada patrón de aprendizaje, de lo cual podemos distinguir a simple vista aquellos que son útiles y aquellos que no lo son. Esta opción puede ser óptima para redes

pequeñas, pero resulta ser muy complicada de utilizar cuando tratamos con redes medianas y peor aun con redes de mayor tamaño, para lo cual necesitamos una representación más explícita de la información la de utilidad de las salidas de los nodos ocultos representados por las columnas de la matriz.

Es muy importante examinar la matriz ya que la posible degeneración de sus columnas implica que las salidas de los nodos ocultos están correlacionadas; es decir los datos disponibles no ayudan a establecer una diferencia entre los mismos. En condiciones prácticas esto hace pensar en la presencia de nodos ocultos redundantes [10]. Una solución a este problema es utilizar la DVS, que nos permitirá determinar una aproximación de la utilidad de cada nodo oculto en el aprendizaje.

4 Descomposición de Valores Singulares

La DVS de una matriz $A_{M \times N}$ es definida como cualquier factorización de la forma [6]:

$$A = U \Sigma V^T \tag{1}$$

Donde U es una matriz $M * M$ ortogonal, V es una matriz $N * N$ ortogonal, y Σ es la matriz diagonal es $M * N$ con elementos $s_{ij} = 0$ si $i \neq j$ y $s_{ii} = s_i$ además $s_1 \geq s_2 \geq \dots \geq s_N \geq 0$ [11].

Las cantidades s_i son llamados los *valores singulares* de A , y las columnas de U y V son llamados *vectores singulares izquierdos y derechos respectivamente*.

La DVS nos proporciona un método sencillo para determinar el rango de la matriz. El rango se define por la cantidad de valores singulares diferentes de cero que representan el número de columnas linealmente independientes. Los valores singulares cercanos a cero representarían la cantidad de columnas linealmente dependientes.

Retomando el problema planteado, las columnas degeneradas, que representan la utilidad de las neuronas, pueden ser identificadas como vectores linealmente dependientes debido a su norma cercana a cero. Por consiguiente estas columnas no son importantes en la estructura de esta matriz. Este efecto se muestra en la figura 2.

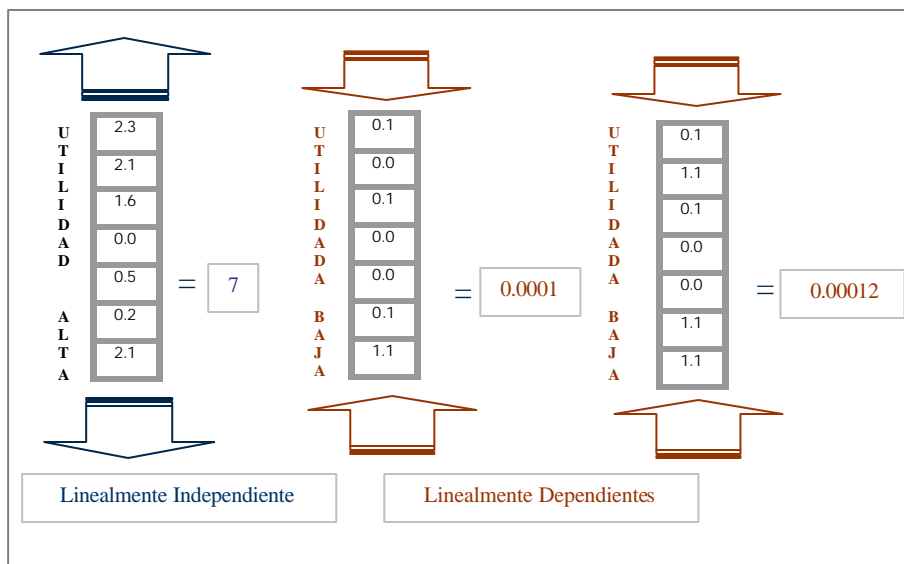


Fig. 2 Efecto SVD

5 Algoritmo de Poda Difusa

La Estrategia o algoritmo que se plantea como poda difusa contiene las estrategias mencionadas anteriormente, la creación de la matriz con los valores de salidas de los nodos ocultos y la DVS de dicha matriz, los cuales son complementados con la lógica difusa para poder hacer más efectivo el proceso de podamiento de la estructura neuronal actual. Este algoritmo se describe en la figura 3.

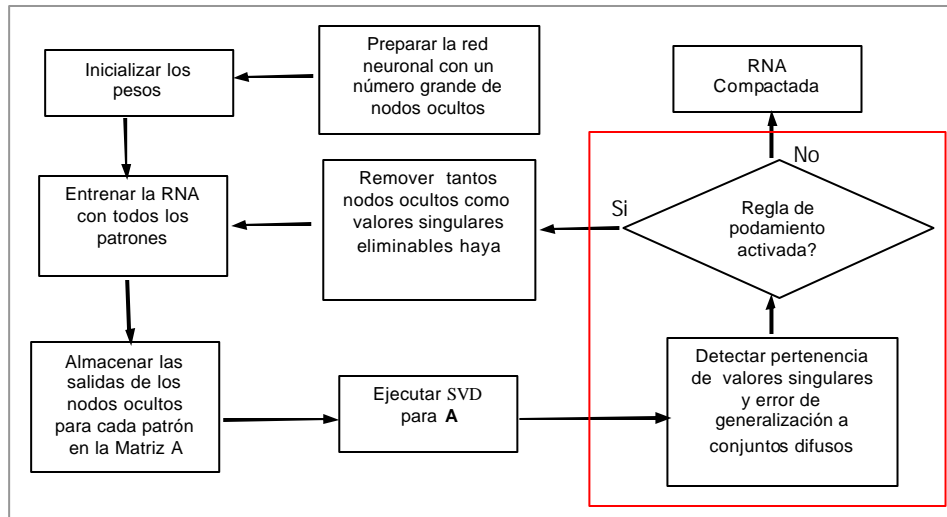


Fig. 3. Algoritmo de Poda Difusa de RNAs.

El modulo difuso procesa los valores singulares que representan la utilidad de los nodos ocultos. Los resultados parciales del tratamiento difuso de los valores singulares de la figura 4, necesitan ser complementados por el error de generalización promedio de la RNA, que fortalecerá la inferencia difusa de eliminación. En la siguiente sección se detallarán los conjuntos difusos y reglas de inferencia necesarias para asociar los dos conceptos mencionados anteriormente.

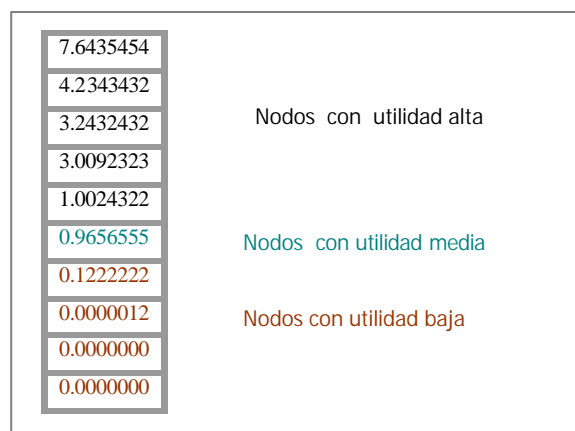


Fig.4 Utilidad de los Nodos Ocultos

6 Utilizando Lógica difusa

La lógica difusa es un sistema que provee un fundamento matemático para capturar relaciones inciertas asociadas con sistemas cognoscitivos humanos tales como el pensamiento y el razonamiento. Simplemente, simula el

pensamiento humano que opera más sobre símbolos que con valores exactos. De hecho, la comunicación y nuestros pensamientos se basan totalmente en estos símbolos o expresiones difusas [9] [14].

En este trabajo se estudiaron y se establecieron conjuntos difusos en base a pruebas relacionadas con los valores singulares y el error de generalización de la RNA.

Es importante considerar el error de generalización de la RNA, pues está relacionado con el tamaño de la RNA, por ejemplo, una RNA de tamaño considerable podría tener un poder de generalización bajo provocando la eliminación de sus nodos ocultos, por otro lado, una RNA pequeña podría tener un poder de generalización alto, lo que determinaría la conservación de sus nodos.

Cada variable difusa contiene a su vez variables lingüísticas representadas por funciones de membresía o pertenencia [3, 8], que puede ser de naturaleza lineal o del orden que requiera la aplicación. En la figura 5 se muestran los conjuntos difusos que serán necesarios para el criterio de eliminación de los valores singulares.

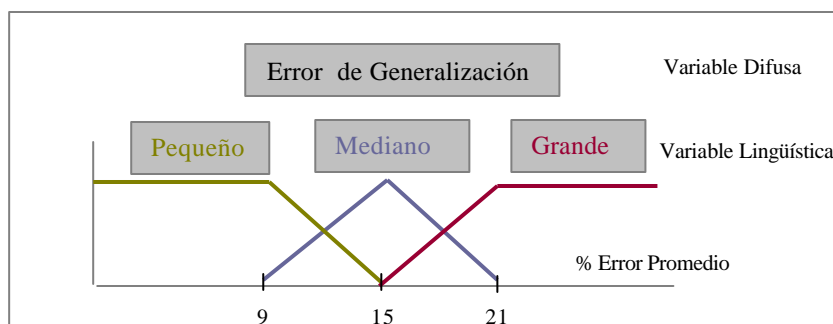


Fig. 5. Conjuntos difusos de Errores de Generalización

Planteando un acercamiento a la utilidad de las neuronas se realizaron pruebas y se obtuvieron los siguientes conjuntos difusos en la figura 6.

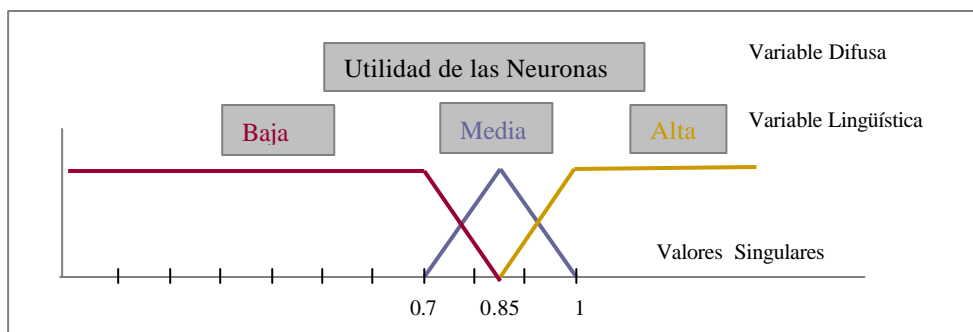


Fig. 6. Conjuntos difusos DVS

Cabe señalar que no siempre será necesaria la eliminación de neuronas después del algoritmo de poda, pues se puede alcanzar una RNA óptima antes de tiempo.

Las reglas difusas necesarias para la inferencia se establecieron en la tabla 1:

EG: Error de Generalización; **VS:** Valor Singular **EVS:** Eliminación del Valor Singular

Puede ser una RNA pequeña que pueda generalizar mejor si (Mpequeño(EG) y Mbaja(VS)) entonces (EVS es Alta)	(1)
Puede ser una RNA pequeña que pueda generalizar mejor o peor, hasta 0.9 se podría eliminar si (Mpequeño(EG) y Mmedia(VS)) entonces (EVS es Media)	(2)
<i>Regla Fija de no Eliminación</i> si (Mpequeño(EG))y Malta(VS) entonces (EVS es Baja)	(3)
<i>Puede ser una RNA mediana que puede generalizar mejor</i> si (Mmediano (EG) y Mbaja(VS)) entonces (EVS es Alta)	(4)
Puede ser una RNA mediana que pueda generalizar mejor o peor, hasta 0.9 se podría eliminar si (Mmediano(EG) y Mmedia(VS)) entonces (EVS es Media)	(5)
<i>Regla Fija de no Eliminación</i> si (Mmediano (EG) y Malta(VS) entonces (EVS es Baja)	(6)
Posiblemente es una RNA Grande que no generaliza bien si (Mgrande(EG) y Mbaja(VS)) entonces (EVS es Alta)	(7)
Posiblemente la RNA pequeña que no tenga mayor poder de aprendizaje si (Mgrande(EG) y Mmedia(VS)) entonces (EVS es Media)	(8)
<i>Regla Fija de no Eliminación</i> si (Mgrande(EG))y Malta(VS) entonces (EVS es Baja)	(9)

Tabla 1. Reglas Difusas

7 Caso de Estudio: Diagnóstico de Enfermedades

El uso de RNA como motores de inferencia de sistemas expertos en el diagnóstico de enfermedades está siendo muy utilizado como ayuda al área de la medicina.

Aquí se propone una RNA con conexiones hacia adelante, que reconoce síntomas que corresponden a un tipo de enfermedad. Los patrones de entrenamiento son síntomas definidos en un vector de longitud 77. Por lo tanto, la RNA contará con 77 neuronas de entrada y 8 neuronas de salida, que son asignados a cada una de las enfermedades a diagnosticar. El número de nodos ocultos en la red neuronal es definido inicialmente por la intuición del diseñador. La RNA iniciará su entrenamiento con un gran número de nodos ocultos y después de haber finalizado el proceso de entrenamiento back-propagation, la aplicación del algoritmo de poda difusa eliminará los nodos ocultos con grado de utilidad bajo.

La tasa de aprendizaje definido como 0.4, determinará la velocidad de convergencia. Se dio un valor de 0.1 al umbral de error, al cual la estructura neuronal tendrá que converger después de 3000 iteraciones, caso contrario saldrá del bucle back-propagation. Debido a la existencia de mínimos locales se agrega el termino momento con un

valor de 0.7

10 casos fueron escogidos según sea la relación por cada enfermedad. En el modo de entrenamiento, estos 80 patrones generados se presentan a la red neuronal en una secuencia arbitraria. Cuando el proceso de entrenamiento culmina la red neuronal estará lista para trabajar. En esta fase, los patrones de prueba son introducidos en las neuronas de entrada y el algoritmo de propagación aplica los pesos óptimos calculados anteriormente. Entonces los valores de las neuronas de salida proporcionaran un grado de aproximación de los datos de prueba a una de las 8 enfermedades aprendidas por la red neuronal.

Un grupo de patrones (síntomas) son seleccionados para cada enfermedad, mostrados en la tabla 2. La proximidad de las salidas de la red a los valores que se muestran a continuación mostrará el éxito de la implementación de la red:

- S1: 1 0 0 0 0 0 0 0 -> Anorexia
- S2: 0 1 0 0 0 0 0 0 -> Cancer de Tiroides
- S3: 0 0 1 0 0 0 0 0 -> Diabetes Mellitus
- S4: 0 0 0 1 0 0 0 0 -> Hipertiroidismo
- S5: 0 0 0 0 1 0 0 0 -> Hipotiroidismo
- S6: 0 0 0 0 0 1 0 0 -> Hirsutismo
- S7: 0 0 0 0 0 0 1 0 -> Obesidad
- S8: 0 0 0 0 0 0 0 1 -> Vaginitis

	ANOREXIA	CÁNCER DE TIROIDES	DIABETES MELLITUS	HIPERTIROIDISMO	HIPOTIROIDISMO	HIRSUTISMO	OBESIDAD	VAGINITIS
Alteración visual	0	0	1	0	0	0	0	0
Alteraciones de la memoria	1	0	1	1	1	0	0	0
Ardor vaginal	0	0	0	0	0	0	0	1
Aumento de apetito	0	0	1	1	0	0	1	0
Aumento de la frecuencia cardiaca	0	0	0	1	0	0	1	0
Aumento de la frecuencia urinaria y la cantidad	0	0	1	1	0	0	0	0
Aumento de la sudoración	0	0	0	1	0	0	0	0
Aumento de peso	0	0	1	0	1	0	1	0
Ausencia de sueño	0	0	0	1	0	0	0	0
Confusión	0	0	1	1	1	0	0	0
Debilidad muscular	1	0	1	1	0	0	0	0
Diabetes	0	0	0	0	0	0	1	0
Dificultad para concentrarse	1	0	1	1	0	0	1	0
Dificultad respiratoria	0	0	0	1	0	0	0	0
Dificultar al tragar	0	1	0	0	0	0	0	0
Disminución de la libido	0	0	0	0	1	0	0	0
Distorsión de la apreciación del peso	1	0	0	0	0	0	0	0
Dolor abdominal	1	0	0	1	1	0	1	0
Dolor al orinar	0	0	0	0	0	0	0	1
Dolor durante el coito	0	0	0	0	0	0	0	1
Dolor en el cuello	0	1	0	0	0	0	0	0
Embarazo	0	0	0	0	0	0	1	0
Estreñimiento	1	0	0	0	1	0	0	0
Fatiga	0	1	1	1	1	0	1	0
Frecuencia cardiaca disminuida	0	0	0	0	1	0	0	0
Ganglios	0	1	0	0	0	0	0	0
Ginecomastia	0	0	0	1	0	0	0	0
Habla lenta	0	0	0	0	1	0	0	0
Hiperactividad	1	0	0	1	0	0	0	0

Hiperdefecación	0	0	0	1	0	0	0	0
Hiperlipemia	0	0	0	0	0	0	1	0
Hipertensión arterial	0	0	0	1	0	0	1	0
Hiperuricemia	0	0	0	0	0	0	1	0
Impotencia	0	0	0	0	1	0	0	0
Infecciones recurrentes	0	0	1	0	0	0	0	0
Intolerancia a la luz	0	0	0	1	0	0	0	0
Intolerancia al calor	0	0	0	1	0	0	0	0
Intolerancia al frío	0	0	0	1	1	0	0	0
Irritación e hinchazón en vulva y vagina.	0	0	0	0	0	0	0	1
Latidos percivibles del corazón	0	0	0	1	0	0	0	0
Letargo	0	0	0	0	1	0	1	1
Malestar	1	1	1	1	1	0	1	1
Menstruación irregular	1	0	0	1	1	0	0	1
Mentiras	1	0	0	0	0	0	0	0
Miedo intenso al aumento de peso	1	0	0	0	0	0	0	0
Movimientos lentos	0	0	0	0	1	0	1	0
Nerviosismo	0	0	1	1	1	0	0	0
Nódulo en el cuello	0	1	0	0	0	0	0	0
Obsesión por la báscula	1	0	0	0	0	0	0	0
Pelo en el tórax	0	0	0	0	0	1	0	0
Pelo en la cara	0	0	0	0	0	1	0	0
Pelo en las orejas	0	0	0	0	0	1	0	0
Pelo en zonas donde solo lo tienen los hombres	0	0	0	0	0	1	0	0
Pérdida de cabello	0	0	0	1	1	0	0	0
Pérdida de peso	0	1	1	1	0	0	0	0
Picor	0	0	1	0	0	0	0	1
Piel seca y áspera	0	0	0	0	1	0	0	0
Preocupación obsesiva por los estudios	1	0	0	0	0	0	0	0
Preocupación por el frío	1	0	0	0	0	0	0	0
Preocupación por la imagen	1	0	0	0	0	0	0	0
Preocupación por la preparación de las comidas	1	0	0	0	0	0	0	0
Preocupación por las calorías de los alimentos	1	0	0	0	0	0	0	0
Problemas de fertilidad	0	0	0	0	1	0	0	0
Problemas psicológicos	1	0	0	0	0	0	1	1
Protusión del globo ocular	0	0	0	1	0	0	0	0
Rechazo a mantener el peso corporal	1	0	0	0	0	0	0	0
Reflejos disminuidos	0	0	0	1	1	0	1	0
Restricción progresiva de alimentos	1	0	0	0	0	0	0	0
Secreción anormalmente excesiva	0	0	0	0	0	0	0	1
Sed exagerada	1	0	1	0	0	0	0	0
Separación de las uñas del lecho ungueal	0	0	0	1	0	0	0	0
Sequedad de la boca	0	0	1	0	0	0	0	0
Somnolencia	0	0	0	0	1	0	0	0
Temblor fino de las manos	0	0	0	1	0	0	0	0
Tumefacción facial	0	0	0	0	1	0	0	0
Vómitos	1	0	1	0	0	0	0	0
Voz ronca	0	0	0	0	1	0	0	0

Tabla 2. Síntomas para cada una de las enfermedades propuestas.

La prueba muestra que el algoritmo de poda DVS propone 6 nodos ocultos para obtener resultados óptimos. Para verificar la validez de esta proposición, un proceso iterativo es aplicado, la RNA se construye y se prueba de nuevo, con un número de nodos ocultos que varían entre 6 y 40¹.

En la primera iteración del algoritmo ejecutado en un simulador con 40 nodos iniciales indicó un error promedio de generalización de 9.947862%, indicando 32 valores singulares eliminables [0, 0.1], en este caso la inferencia difusa aprobará su eliminación inmediata.

La segunda iteración pero con 8 nodos ocultos, indica 1 valor singular a eliminar, con un error de generalización promedio de 12.957577%.

La tercera iteración con 7 nodos ocultos indica 1 valor singular eliminable con un error de generalización promedio de 14.041162%.

¹ El número de nodos inicial en el proceso es definido como la media del número de nodos de la capa de entrada.

La Cuarta iteración con 6 nodos ocultos indica 0 valores singulares eliminables, con un error de generalización de 10.120831% mejorando significativamente, con respecto a las anteriores configuraciones de RNAs.

En esta simulación inicial con 40 nodos ocultos se pudo concluir que en 4 iteraciones el algoritmo de poda difusa obtuvo una RNA óptima. Sin embargo este proceso, con la misma cantidad de nodos ocultos, probado en una nueva sesión concluyó en solo 3 iteraciones y con un error promedio mínimo de 6.555290%, resultado que supera la simulación anterior como observamos en la figura 7. Este efecto se debe a que el mínimo local alcanzado en cada entrenamiento será el primero que encuentre, valor diferente en cada iteración, por lo que la inferencia asume una RNA óptima. Estos resultados se observan en la figura 8.

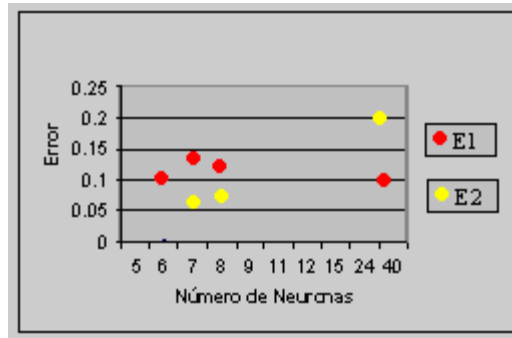


Fig. 7. Error de Generalización de la RNA

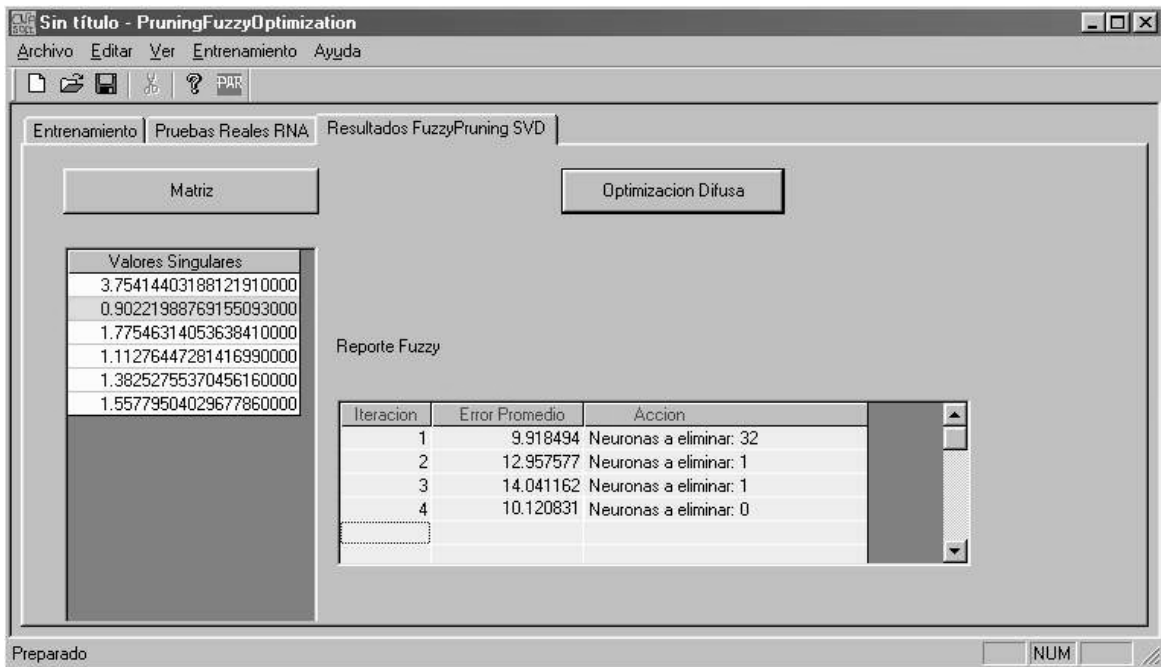


Fig. 8. Simulador de Optimización

8 Conclusiones

La aplicación de lógica difusa en el algoritmo propuesto, proporciona un mayor grado de automatización y flexibilidad en el proceso de eliminación de nodos ocultos en comparación con procesos iterativos. Con lo anterior se puede concluir que la determinación de la función de membresía es muy importante para lograr un buen acercamiento a la eliminación de los valores singulares sin perder poder de generalización en la RNA

Usar la DVS de una matriz en el cálculo, es mucho mejor que la matriz original, ya que esta expresa mejor la estructura geométrica de la matriz además de ser más robusta al error numérico. La DVS presta una definición práctica de rango, así como cuantificar el significado de la noción de aproximación de rango deficiente. Aplicar todas estas definiciones permite construir una forma de podamiento más potente y eficaz de optimización de RNAs y mayor aun si se combina con las relaciones inciertas de lógica difusa.

Los algoritmos y técnicas propuestas han demostrado ser eficientes buscando una topología adecuada, no sólo refiriéndose a encontrar la mínima topología sino también a encontrar aquella que pueda generalizar eficientemente ante cualquier entrada.

La eliminación de nodos ocultos está relacionado directamente con las columnas linealmente dependientes de norma próxima a cero, existiendo la posibilidad de identificar columnas linealmente dependientes pero con una norma mucho mayor a cero, eliminando columnas que representan neuronas verdaderamente útiles. Sería bueno controlar este efecto mediante el cálculo de la norma de dichos vectores, evitando de esta manera su eliminación.

9 Bibliografía y Referencias

- [1] Babri, H.A., Kot, A.C., Tan, N.T. and Tang, J.G., 1997, "Dynamic Pruning Algorithms for Improving Generalization of Neural Networks", Proceedings of the International Conference on Information, Communications and Signal Processing, ICICS'97, Singapore.
- [2] Castellano, G., Fanelli, A.M. and Pelillo, M., 1997, "An Iterative Pruning Algorithm for Feedforward Neural Networks", IEEE Transactions on Neural Networks, Vol 8, pp. 519-531.
- [3] Corzo Y, "La Lógica Difusa", Porlamar, Venezuela Departamento de Matemática, Física, Historia. <http://www.arrakis.es/~casanchi/difusa01.htm#01>
- [4] Gao, Xiao-Zhi, 1999, "Soft Computing Methods for Control and Instrumentation", Thesis for the Degree of Doctor of Science in Technology, Department of Electrical and Communications Engineering, Helsinki University of Technology.
- [5] Hilera G. J.R, "Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones". Addison Weley. Iberoamericana. RA-MA, 1995.
- [6] Leach, S., 1999, "Singular Value Decomposition - A Primer", Technical Report, Department of Computer Science, Brown University, <http://www.cs.brown.edu/research/ai/dynamics/tutorial/Documents/SingularValueDecomposition.html>
- [7] Lippman, R.P., 1987, "An Introduction To Computing With Neural Nets", IEEE ASSP Magazine, April, pp. 4-22.
- [8] Nauck, D. and F. Klawonn, "Neuro-Fuzzy Classification Initialized by Fuzzy Clustering", Proc. 4th European Congress on Intelligent Techniques and Soft Computing (EUFIT96), Aachen 1996.
- [9] Ojala T. (1994) Neuro-Fuzzy Systems in Control. Department of Electrical Engineering. Tampere University of Technology. Tampere, Finland.
- [10] Psychogios, D.C. and Ungar, L.H., 1994, "SVD-NET: An Algorithm that Automatically Selects Network Structure", IEEE Transactions on Neural Networks, Vol 5, pp. 513-515.

- [11] P. Dewilde and Ed. F. Deprettere. Singular value decomposition: An introduction. In Ed. F. Deprettere, editor, SVD and Signal Processing: Algorithms, Applications, and Architectures, pages 341. Elsevier Science Publishers, North Holland, 1988.
- [12] Reed, R., 1993, "Pruning Algorithms – A Survey", IEEE Transactions on Neural Networks, Vol 4, pp. 740-747.
- [13] Serkan Güroglu, Onur L. Çetin, and Abdülkadir Erden, "Application of a Pruning Algorithm to Automate the Design of Artificial Neural Networks". 5th Int. Conf. Mechatronic Design and Modeling, Konya, 12-14 Sept. 2001.
- [14] Tanaka, K. (1962). An Introduction to fuzzy logic for practical applications. Rassel, Inc.
- [15] Thimm, G. and Fiesler, E., 1996, "Neural Network Pruning and Pruning Parameters", 1st Online Workshop on Soft Computing, Nagoya, Japan, <http://www.bioele.nuee.nagoya-u.ac.jp/wsc1>