APROXIMACIONES ENTRE LAS REDES NEURONALES Y LOS OPERADORES BORROSOS

Pedro Tejedor, Miguel Ángel Sanz
Universidad Pontificia Comillas
Escuela Técnica Superior de Ingeniería ICAI
Instituto de Investigación Tecnológica
Alberto Aguilera, 23
28015 Madrid
T^{o}: 915422800. Fax: 915423176
<tejedor,masanz>@iit.upco.es

Resumen

This paper describes the main results concerning a comparative study between simple architectures of trained neural networks using the logistic function as activation function, and the performance of the classic Łukasiewicz operators. The paper shows how those operators can explain trained neural networks. Other fuzzy operators are tested and the results explained. In all cases the results obtained promise an advance in the explanation of trained neural networks using fuzzy logic.
Palabras clave: Rule extraction, neural networks, fuzzy logic.

1 INTRODUCCIÓN

El uso de redes neuronales constituye una práctica muy extendida para modelar relaciones complejas de variables características de determinados procesos. Esta técnica de Inteligencia Artificial permite abordar problemas tanto de clasificación como de regresión. Incluso lo primero puede ser visto como un caso particular de lo segundo[3]. Sin embargo, las ventajas que poseen las redes neuronales tienen un coste: las redes crean modelos del tipo "caja negra", esto es, que no dan información sobre la razón de su estructura interna. El conocer la razón de las relaciones que define una red neuronal es especialmente deseable en entornos que modelan relaciones críticas complejas, o en aquellos en los que sea deseable conocer el razonamiento interno para estudiar las posibles incoherencias o su comportamiento ante los casos poco frecuentes. Saber qué significado tiene el modelo que describe una red neuronal es muy importante. Este significado se ha intentado obtener por varios métodos, todos ellos desarrollados alrededor de la extracción de reglas.

Este artículo pretende ser una contribución al conocimiento de un modelo definido por una red neuronal, entrenada previamente, a través de reglas de naturaleza borrosa. Por lo tanto este estudio habría que encuadrarlo entre las denominadas técnicas "neuro-fuzzy" ([8]), aunque también estaría próximo a las técnicas de minería de datos.

2 TÉCNICAS DE EXTRACCIÓN DE REGLAS DESDE REDES NEURONALES

La extensión de este artículo no permite hacer una revisión pormenorizada de todas las técnicas desarrolladas en el campo de la extracción de reglas con técnicas basadas en lógica borrosa, sin embargo se pueden enunciar sus principales características.

En este sentido son muchos los trabajos que se encuentran en la literatura, testigos de la fecunda interacción entre las redes neuronales y los sistemas borrosos [4]. Muchos se orientan a borrosificar en alguna medida las redes, bien sea porque trabajan con entradas y salidas reales, pero con pesos borrosos como las *neo-fuzzy* de [12], bien con entradas y salidas borrosas, pero con pesos reales como en [9], o bien con entradas, salidas y pesos borrosos como en [10] o [11].

La gran mayoría de los trabajos en el campo borroso se orientan hacia el estudio del aprendizaje en redes modificadas en una u otra forma. Esto les aleja del propósito de nuestro trabajo, de explanación de redes clásicas, según estas resultan de un entrenamiento. Una muy notable excepción la constituyen los trabajos de [5] y el excelente trabajo de [1], [2]. Mediante el uso de un nuevo operador borroso propuesto por los autores, el *i-or*. Éste es capaz de desacoplar perceptrones, hallar un equivalente borroso exacto de un perceptrón multicapa entrenado y darle una interpretación.

Nuestro trabajo constituye también un paso en el intento de explicación de los perceptrones multicapa

utilizados para regresión. En concreto aquellos que utilizan la función logística, definida como $\Psi(x) = 1/(1 + \exp(-\omega \cdot x - b))$, como función de activación. Buscaremos ante todo la interpretabilidad, aun a costa de la precisión de nuestros modelos. Para ello utilizaremos un camino en cierta forma inverso: basados en la aproximación propuesta por [7], vamos a entrenar perceptrones multicapa con dos entradas y una sola capa oculta para que emulen la superficie generada por distintos operadores borrosos. De esta forma podremos tener una primera aproximación del comportamiento lógico de los perceptrones con sus distintas combinaciones de pesos.

El resto de este trabajo está organizado como sigue. En la sección 3 comprobaremos el comportamiento de redes sencillas en términos de los operadores lógicos propuestos por Łukasiewicz. Esto nos permitirá tener una herramienta para examinar otras analogías más complejas, como las que se establecen con las t-norma y t-conorma clásicas, lo que se expondrá en la sección 4. Finalizaremos con las conclusiones y propuestas de futuros desarrollos.

3 EXPLICACIÓN DE LOS PARÁMETROS DE UNA RED NEURONAL MEDIANTE OPERADORES BORROSOS DE ŁUKASIEWICZ

Nuestro primer paso va a consistir en entrenar un par de redes neuronales para modelar la t-norma y la t-conorma de Łukasiewicz. Las formas de ambas superficies son las mostradas en las figuras 1, donde se han marcado las líneas de contorno en el plano horizontal. La red que usaremos tendrá dos perceptrones con función de activación sigmoidal. La capa de salida será proporcional. Su estructura se representa esquemáticamente en la figura 2, en la que se han rotulado los pesos de la primera capa como "W", los de la segunda como "V" y los sesgos como "b". Exponemos los resultados de este entrenamiento en la tabla 1.

Tabla 1: Resultados del entrenamiento con una t-norma de Łukasiewicz

	N1	N2
W1	4.91	-15.34
W2	4.89	-55.88
b	-7.03	28.68

	Sal.
VS1	0.96
VS2	0.04
b	-0.05

Un examen detallado de los valores de la tabla 1 nos indica que la neurona N2 prácticamente está desactivada, dado que su peso en la salida es de sólo 0.04,

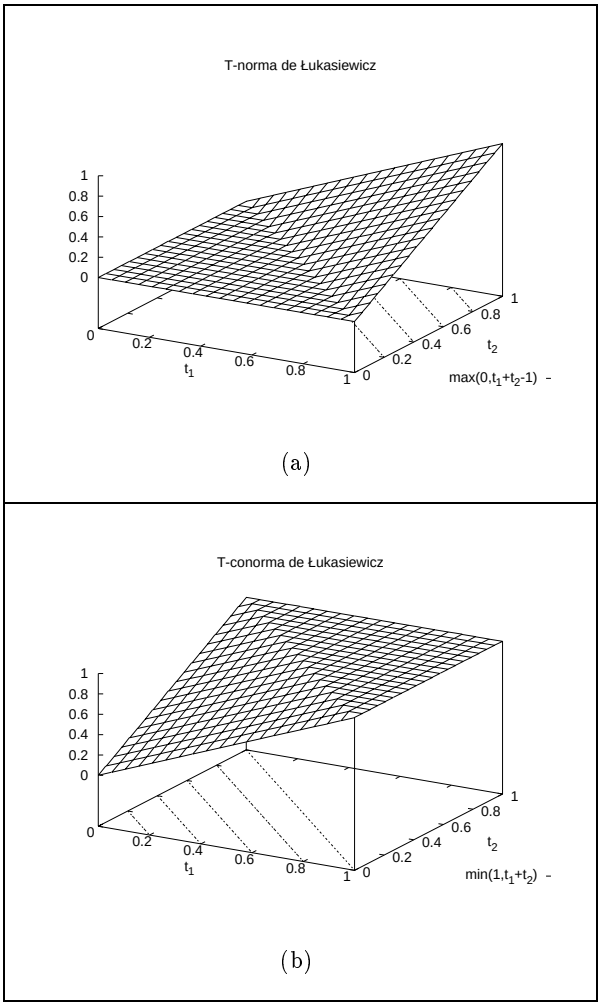


Figura 1: Operadores de Łukasiewicz

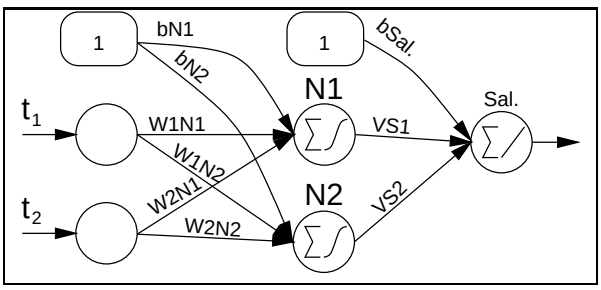


Figura 2: Estructura de la red usada en este trabajo

frente al peso de la neurona N1 de valor 0.96. Los pesos que llegan a N1 tienen valores cercanos a los 5.02 previstos (cf. [7]). Podemos apreciar en la figura 3

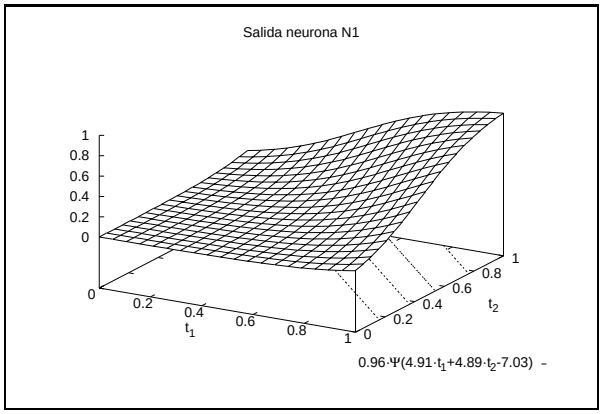


Figura 3: Superficie generada por la neurona N1, corregido por su peso

cómo la información de la t-norma de la figura 1 es aproximada por la superficie generada por la neurona N1. Merece la pena destacar el valor del sesgo en esta neurona, en la línea propuesta por [6]. Como es sabido, el sesgo consigue un desplazamiento de toda la sigmoidal en un valor ponderado por el valor de los pesos. En este caso su valor consigue que la neurona sólo alcance su nivel de saturación cuando tanto t_1 como t_2 están cerca del máximo valor de su rango, como cabría esperar de una función que efectúa una t-norma.

Veamos a continuación el ejemplo equivalente cuando la función a entrenar es una t-conorma de Łukasiewicz. Los resultados los expondremos de forma análoga al caso anterior. La tabla 2 muestra los resultados de un

Tabla 2: Resultados del entrenamiento con una t-conorma de Łukasiewicz

	N1	N2
W1	5.77	22.56
W2	5.72	-24.83
b	-3.29	11.71

	Sal.
VS1	0.89
VS2	-0.005
b	0.13

entrenamiento tipo. En él de nuevo observamos que la neurona N2 no presenta actividad significativa a la salida. El valor de su peso es de -0.005. La neurona N1 es la que ha aproximado la función borrosa. Su semejanza con la misma se puede apreciar comparando la figura 1(b) con la figura 4.

De nuevo un examen atento de los pesos y su relación con el sesgo nos indica que la función alcanzará la saturación con que cualquiera de las variables de entrada alcance un valor elevado, sólo haciéndose cero en el caso de que ambas entradas estén muy próximas a ese

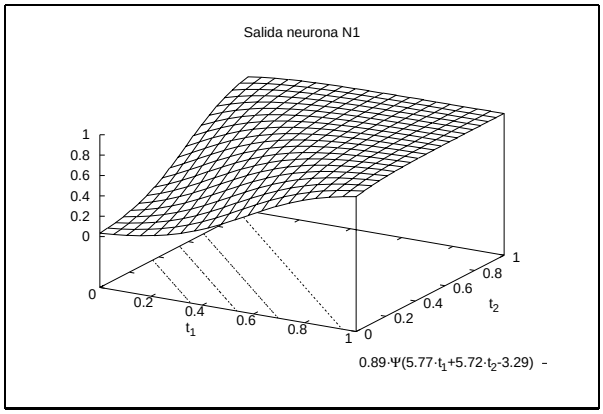


Figura 4: Superficie generada por la neurona N1 por su peso

valor. Este es el comportamiento que cabría esperar de una t-conorma.

Recapitulando lo visto hasta ahora, las funciones de Łukasiewicz presentan una superficies sencillas y fáciles de entrenar con una sola neurona. Los resultados concuerdan en gran medida con los propuestos en [7], excepto en que ese trabajo se planteó con perceptrones aislados. Aquí podremos afirmar que la diferencia entre una neurona que está efectuando una función que emula una t-norma y una que lo haga con una t-conorma estará sobre todo en el valor del sesgo. También veremos que la interrelación entre los valores de pesos y sesgo es de una importancia capital a la hora de definir cuál es la función borrosa que puede estar detrás de un determinado comportamiento en las neuronas.

Hasta ahora las superficies que hemos entrenado han sido extremadamente sencillas. Intentemos ahora una función no tan sencilla, pero que mantenga su interpretabilidad. Entrenemos la salida generada por XOR. Esta función se puede caracterizar como $\mathrm{XOR}(t_1, t_2) = (\neg t_1 \wedge t_2) \vee (t_1 \wedge \neg t_2)$, y para generarla utilizaremos las t-norma y t-conorma de Łukasiewicz.

La superficie generada por esta función la podemos apreciar en la figura 5. Los parámetros resultado del entrenamiento los detallamos en la tabla 3. Se pue-

Tabla 3: Resultados del entrenamiento con XOR

	N1	N2
W1	3.5	-3.42
W2	-3.49	3.41
b	-0.33	-0.21

	Sal.
VS1	6.81
VS2	6.86
b	-5.84

de apreciar la alta simetría de los parámetros. Como ilustración pondremos la neurona N2 (figura 6) que de acuerdo con la primera parte de esta sección está

efectuando la función $\neg t_1 \wedge t_2$. La neurona N1 será la simétrica, y su salida es análoga a $t_1 \wedge \neg t_2$. La neurona de salida se asimila a la función OR. Por tanto, el resultado concuerda con lo que cabría esperar de las analogías expuestas.

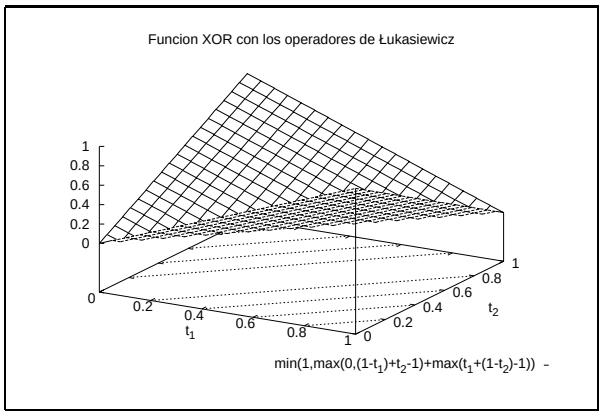


Figura 5: Superficie de la función XOR

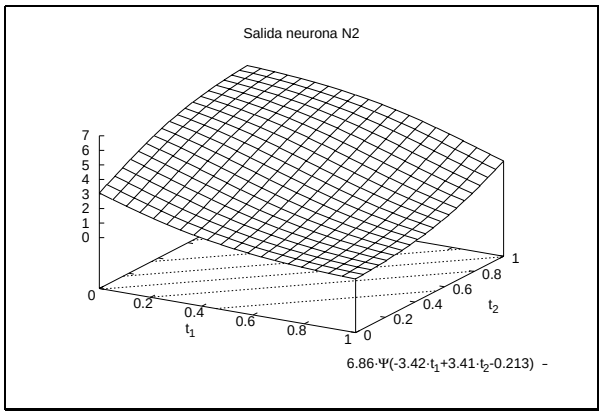
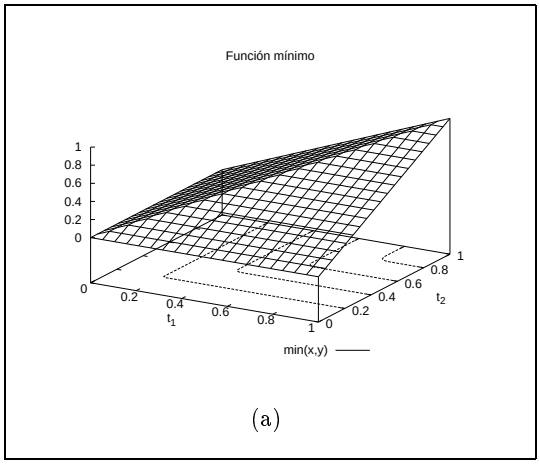


Figura 6: Salida de la neurona N2 en el entrenamiento de XOR

4 OTROS OPERADORES BORROSOS

Se realizará un estudio similar usando ahora la t-norma y la t-conorma clásicas (ver figuras 7 y 9). En este apartado entrenaremos una red sucesivamente con cada uno de los conjuntos de entrenamientos mostrados en la figura 7. La estructura de la red será la misma que en el apartado anterior (ver figura 2). Trataremos de asociar las superficies generadas por las funciones logísticas después del entrenamiento con los operadores de Łukasiewicz, tal como hemos visto en la sección 3.

Después de entrenada la red de la figura 2 con la superficie $\min(x, y)$ (figura 7), los resultados son los que se indican en la tabla 4. Las superficies generadas



(a)

Figura 7: t-norma clásica

Tabla 4: Resultados del entrenamiento con $\min(t_1, t_2)$

	N1	N2
W1	-4.67	2.37
W2	4.64	-0.32
b	1.99	-1.07

	Sal.
VS1	1.17
VS2	1.99
b	-1.55

por las distintas neuronas se muestran en las figuras 8. En ellas se puede apreciar cómo (a) la neurona N1 adopta una forma análoga a la función lógica $\neg t_1 \vee t_2$ utilizando los operadores de Łukasiewicz . La neurona N2 produce una extensión cilíndrica de t_1 (b), y la neurona de salida tiene un lugar geométrico (c) muy similar a la t-norma de Łukasiewicz. Todo parece apuntar a que la función lógica efectuada por la red es: $AND_a(OR_a(\neg t_1, t_2), t_1)$. Es sencillo demostrar que es idéntica a $\min(t_1, t_2)$, utilizando los operadores de Łukasiewicz.

Un resultado análogo obtenemos del entrenamiento con la superficie $\max(x, y)$ (figura 9). Los resultados se muestran en la tabla 5. De nuevo podemos apreciar

Tabla 5: Resultados del entrenamiento con $\max(t_1, t_2)$

	N1	N2
W1	-6.93	2.87
W2	7.94	-0.07
b	-3.24	-1.73

	Sal.
VS1	0.737
VS2	1.44
b	0.112

las semejanzas con operadores lógicos de Łukasiewicz. La superficie generada por N1 (a) es similar a $\neg t_1 \wedge t_2$. La generada por N2 es análoga a la extensión cilíndrica de t_2. El lugar geométrico de la superficie de la neurona de salida apunta al mismo que $t_1 \vee t_2$. En conjunto, la red parece estar efectuando la operación lógica $OR_a(t_1, AND_a(\neg t_1, t_2))$, que es sencillo mostrar

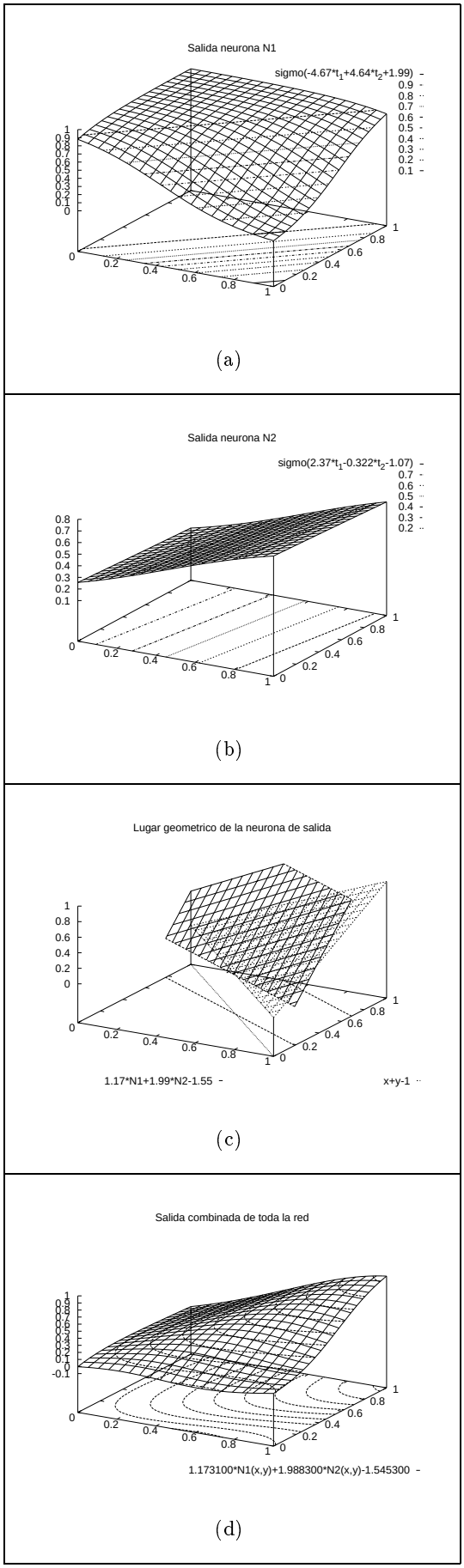


Figura 8: Salidas de las neuronas de la red

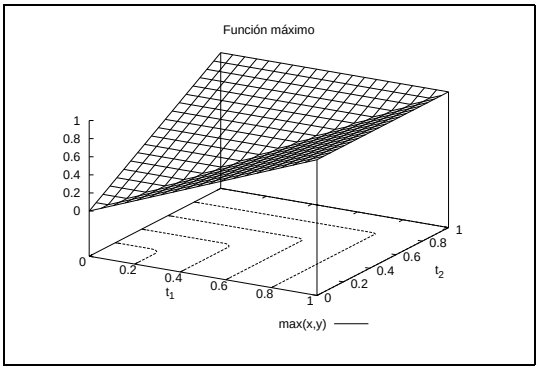


Figura 9: t-conorma clásica

como idéntica a $\max(t_1, t_2)$.

5 CONCLUSIONES

En este artículo se han explorado las analogías entre una red neuronal sencilla que utiliza la función logística como función de activación de un perceptrón multicapa y los operadores lógicos de Łukasiewicz. Éstos se han aplicado a las redes entrenadas para modelar la t-norma y t-conorma clásicas, encontrando que las redes se ajustan para conseguir una expresión equivalente a los operadores clásicos utilizando operadores de Łukasiewicz. En los casos estudiados de redes entrenadas ha sido posible reconstruir la función entrenada con operadores lógicos borrosos. Estos resultados son muy prometedores, y abren el camino para una comprensión más profunda de los mecanismos de las redes, utilizando las técnicas de lógica borrosa.

Referencias

[1] J. M. Benítez, J. L. Castro, and I. Requena. Are artificial neural networks black boxes? *IEEE Transactions on neural networks*, 8(5):1156–1164, sep 1997.

[2] José Manuel Benítez. *Extracción de reglas difusas con y de redes neuronales artificiales*. PhD thesis, E.T.S. de Ingeniería Informática. Departamento de Ciencias de la Computación e Inteligencia Artificial, feb 1998.

[3] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, 1995.

[4] James J. Buckley and Yoichi Hayashi. Fuzzy neural networks: A survey. *Fuzzy Sets and Systems*, 66:1–13, 1994.

[5] J.L. Castro and E. Trillas. The logic of neural networks. *Mathware and Soft Computing*, 5:23–37, 1998.

[6] LiMin Fu. Rule generation from neural networks. *IEEE Transactions on Systems, Man and Cybernetics*, 24(8):1114–1124, aug 1994.

[7] Pierre-Yves Glorennec. Neuro-fuzzy logic. In *5th IEEE International Conference on Fuzzy Systems*, volume 2, pages 899–904. IEEE Neural Networks Council, sep 1996.

[8] Madan M. Gupta. Fuzzy neural networks: theory and applications. In *Proceedings of Spie - The international society for optical engineering*, volume 2353, pages 303–325, 1994.

[9] Hisao Ishibuchi, Ryosuke Fujioka, and Hideo Tanaka. An architecture of neural networks for input vectors of fuzzy numbers. *IEEE*, pages 1293–1300, 1992.

[10] Hisao Ishibuchi, Hideo Tanaka, and Hideiko Okada. Fuzzy neural networks with fuzzy weights and fuzzy biases. *IEEE*, pages 1650–1655, 1993.

[11] W. Pedrycz. Fuzzy neural networks with reference neurons as pattern classifiers. *IEEE Transactions on neural networks*, 3(5):770–775, Septiembre 1992.

[12] Takeshi Yamakawa, Eiji Uchino, Tsutomu Miki, and Hiroaki Kusanagi. A neo fuzzy neuron and its applications to system identification and prediction of the system behavior. In *Proceedings of the 2nd International Conference on Fuzzy Logic, Iizuka, Japan*, pages 477–483, jul 1992.