Estudio de las redes neuronales de Perceptrón, Adaline y Madaline

Juan Pablo Requez Vivas
1398086
juanrequez@gmail.com

Resumen: Para comprobar las habilidades de clasificación sobre datos separables de manera lineal o no en un espacio bidimensional, se utilizan redes neuronales de Perceptrón, Adaline y Madaline. El uso de estas redes muestra que las formas más simples, el Perceptrón y Adaline, son incapaces de clasificar conjuntos que no son separables linealmente mientras que la red Madaline puede resolver este problema. Un estudio acerca del tiempo requerido para el entrenamiento de las redes neuronales y el número de presentaciones que debe hacerse de los datos para lograrlo es presentado, mostrando que el perceptrón requiere, para los datos elegidos, menor cantidad de tiempo y épocas de entrenamiento que la red Adaline con factor de aprendizaje de η=0.1 en el plano y la incapacidad de ambas redes de adaptarse a la solución de un problema de separación no lineal sencillo, descrito por la función lógica XOR. Además, se muestra la relación del número de épocas y tiempo requerido para la utilización de redes Madaline con distintos tamaños en una sola capa, mostrando que la relación entre el número de neuronas y la velocidad del aprendizaje no es lineal o fácilmente predecible.

I. INTRODUCCIÓN

Las redes neuronales más simples, el Perceptrón y Adaline, permiten hacer clasificación de conjuntos con ciertas características particulares. Estas redes describen la región a través de una superficie de clasificación dada por un hiperplano, que divide a R^n en dos regiones separadas; si un conjunto de datos produce una salida positiva de la neurona, su salida es clasificada como pertenenciente a un tipo, por el contrario, si la salida es negativa, es clasificada como el otro. Esta habilidad es una simplificación del funcionamiento de la neurona biológica y para cada una se ha establecido una regla de entrenamiento que, intentando simplificar el aprendizaje de seres vivos, utiliza una cuantificación del error establecido por la red neuronal artificial para hacer correcciones sobre su funcionamiento hasta alcanzar el deseado.

Una de las primeras observaciones que puede hacerse de la descripción indicada es que un hiperplano clasifica elementos en dos categorías que están separadas linealmente, es decir, cuya curva de clasificación es lineal. Esto presenta una desventaja del uso de estas estructuras, el perceptrón y Adaline, para clasificación. Una generalización de la red Adaline, que requiere múltiples neuronas de este tipo, fue introducida para superar esta limitación, llamada Madaline. La red Madaline puede hacer clasificación usando múltiples superficies lineales en el universo de clasificación, lo que permite resolver problemas más complejos.

En este documento se describe el funcionamiento de estas tres redes, el perceptrón, Adaline y Madaline (aunque debe aclararse que, técnicamente, el Perceptrón y Adaline no son redes pues ambas están compuestas por una neurona artificial), y son puestos a prueba para distintos tipos de conjuntos separables linealmente o no, en especial sobre los conjuntos clásicos correspondientes a la función lógica OR, AND y XOR, y posteriormente sobre una región uniforme. El objetivo es observar la dificultad del entrenamiento, comparar los distintos tiempos de cálculo requeridos para realizar tal tarea y entender, de manera práctica, la capacidad de clasificación de estas tres redes, sobre datos descritos en un espacio bidimensional.

Este documento está estructurado de la siguiente manera: en la sección II se presenta una descripción de las redes en cuestión; en la sección III se describe la metodología y el conjunto de datos elegidos para las pruebas; en la sección IV se muestran los resultados de este análisis para luego, en la sección V concluir al respecto; finalizando con las referencias utilizadas en la realización de este trabajo en la sección VI.

II. GENERALIDADES DE LAS REDES NEURONALES PERCEPTRÓN, ADALINE Y MADALINE

Las redes neuronales artificiales computacionales han sido usadas desde la década de 1950, aproximadamente. Los perceptrones, uno de los primeros tipos de redes neuronales, fueron inventadas por Frank Rosenblatt en 1957, en un esfuerzo por entender la memoria humana, así como el aprendizaje y demás procesos cognitivos. [1]

Una neurona Perceptrón está constituida de una salida que está conectada con un factor de peso con todas las entradas [2]. Cuando se considera la neurona más simple, solo tiene dos entradas y un bias como se muestra en la Figura 1.

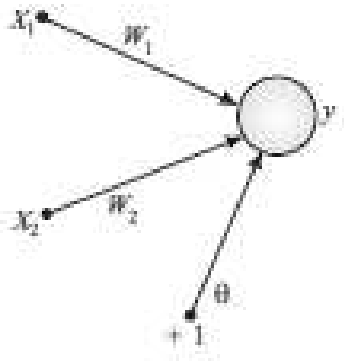


Figura 1. Neurona simple de dos entradas. [2]

Para un perceptrón, la salida y se calcula como:

$$y = w_1 x_1 + w_2 x_w + \theta \tag{1}$$

La función de activación F toma la salida *y* y ejecuta una operación sobre ella. Para el perceptrón, la función de activación es la función de umbral definida como [2]:

$$F(y) = \begin{cases} 1 \; si \; y > 0 \\ 0 \; si \; y \leq 0 \end{cases} \tag{2}$$

Esta red Perceptrón puede ser usada como clasificador de entradas. Si el patrón correspondiente a la entrada es positivo, el patrón es asignado a la clase “1” y si es negativo o cero, a la clase “0”. La separación entre las clases está dada por un hiperplano, y en el caso de esta neurona de dos entradas, por una recta dada por la ecuación:

$$w_1 x_1 + w_2 x_2 + \theta = 0 \tag{3}$$

Esta recta está representada en la Figura 2

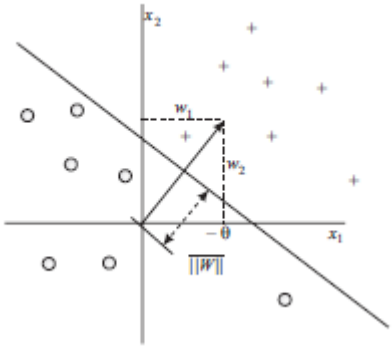


Figura 2. Recta de separación de las clases [2]

El perceptrón posee un algoritmo de entrenamiento asociado, conocido como Regla de Entrenamiento de Perceptrón, en este procedimiento los pesos se ajustan según :

$$w(t+1) = w(t) + (y-d)x \qquad (4)$$
$$\theta(t+1) = \theta(t) + (y-d)$$

Si la neurona responde correctamente a lo que se espera para un dato de entrada, la neurona no es actualizada.

La convergencia del entrenamiento está garantizada. En [2] se indica que "si existe un conjunto de pesos que son capaces de realizar la clasificación en cuestión, la regla de entrenamiento de Perceptrón convergerá a una de ellas en un número finito de pasos para una elección inicial de pesos". Una demostración detallada de este teorema está en [2] [3].

Una generalización del perceptrón es la neurona Adaline. En la década de 1960, el Elemento Lineal Adaptativo (ADALINE) fue propuesto por Widrow y Hoff, en donde se utilizaba un sistema adaptativo que fue usado eventualmente para eliminar los ecos en los sistemas telefónicos. [4]

En realidad, la arquitectura de la neurona es escencialmente la misma del perceptrón mostrado en la Figura 1, la principal diferencia estriba en el procedimiento con el que la neurona es entrenada. Inicialmente, sus creadores, Widrow y Hoff, implementaron esta red como un circuito [2], como se muestra en la Figura 3.

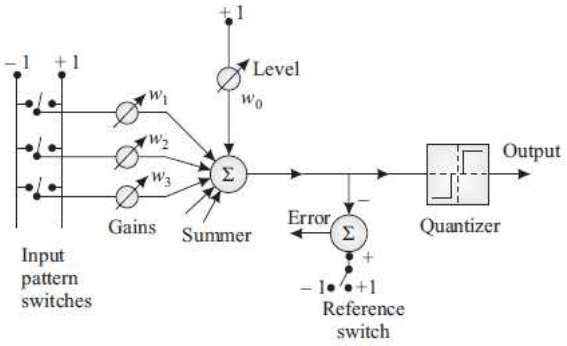


Figura 3. Circuito de una neurona Adaline. [5]

Para una neurona Adaline, la función de umbral es usada como bipolar, es decir, con ambas salidas unitarias de diferente signo como se muestra

$$umbral(y) = \begin{cases} 1 \; si \; y > 0 \\ -1 \; si \; y \leq 0 \end{cases} \qquad (5)$$

La diferencia primordial entre la neurona Perceptrón y la Adaline subyace en el método de entrenamiento. Para una neurona Adaline, el entrenamiento utiliza la suma ponderada de las entradas como parte del algoritmo en vez de la salida de la neurona [3]. En la regla de entrenamiento de perceptrón, se usa la salida del umbral para el entrenamiento mientras que Adaline usa la salida ponderada sin la función de umbral. Para un entrenamiento de una neurona Adaline se utiliza la regla delta, dada por

$$w(t+1) = w(t) + \eta(d - w(t)x)x \qquad (6)$$

Aunque el Perceptrón y Adaline pueden ser usados para la clasificación, solo pueden aplicarse en problemas de separación lineal [6] [2] [3]. Esto fue demostrado por Minsky y Papert, indicando que una red perceptrón no puede usarse para la clasificación de datos correspondientes a una función lógica de O-Excluyente (XOR). En [2] se presenta una demostración para esto.

Una extensión de la neurona Adaline a una red es conocida como Madaline (Multiple Adaline). Su estructura más simple se muestra en la Figura 4.

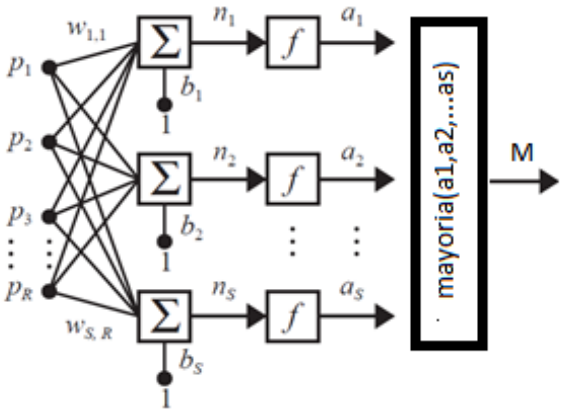


Figura 4. Red Madaline. [5]

El entrenamiento de una red Madaline difiere considerablemente del de la red Adaline en que no se conoce cuanto es la salida esperada de cada neurona que la constituye. Un algoritmo de entrenamiento se presenta en la siguiente sección.

La red Madaline es intuitiva, pero ineficiente [3]. Muchos pasos de entrenamiento son necesarios, y puede incurrir en miles de pasos para un problema sencillo, y lamentablemente es muy sensible al ruido.

III. METODOLOGÍA

A. Estructura de las redes neuronales

Para cada conjunto de datos de prueba, se entrenará una neurona Adaline, una neurona Perceptrón y una red Madaline. La neurona Adaline y el perceptrón tienen dos pesos y un bias para el conjunto de datos de análisis, en una estructura como la que se muestra en la Figura 5.

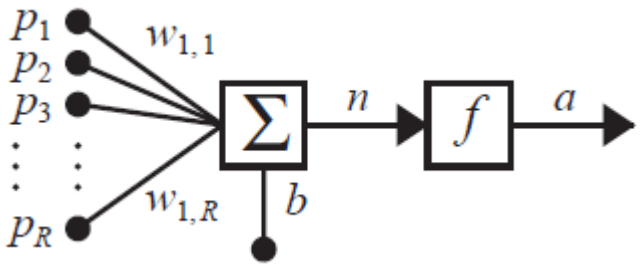


Figura 5. Neurona de múltiples entradas. [5]

La red Madaline, en cambio, está compuesta de múltiples neuronas en número impar, y sobre las salidas se ejecuta una función de mayoría como la indicada en el algoritmo de entrenamiento. Un diagrama esquemático de la red se puede ver en la Figura 4.

B. Conjunto de prueba para clasificación

Para el estudio comparativo de las redes neuronales de tipo Perceptrón, Adaline y Madaline, se utiliza un conjunto de datos generados para tal sentido. Estos datos son en 2 dimensiones y corresponden a una interpretación de la utilización de una función lógica OR, una AND, una XOR y posteriormente, una región en el plano separada linealmente.

La función lógica OR está implementada a través de 400 datos, distribuidos en cuatro regiones del plano, cada una es una rejilla que contiene 10x10 elementos. El significado de los valores como una función lógica fue representada a través del razonamiento para los signos de las variables de entrada, X1 y X2, presentado en la Tabla 1. Una representación gráfica de las regiones en el plano cartesiano de los valores utilizados se presenta en la Figura 6, donde se muestra en (a) las regiones definidas por la función OR, en (b) las regiones definidas por la función AND y en (c) las regiones para la funciónXOR.

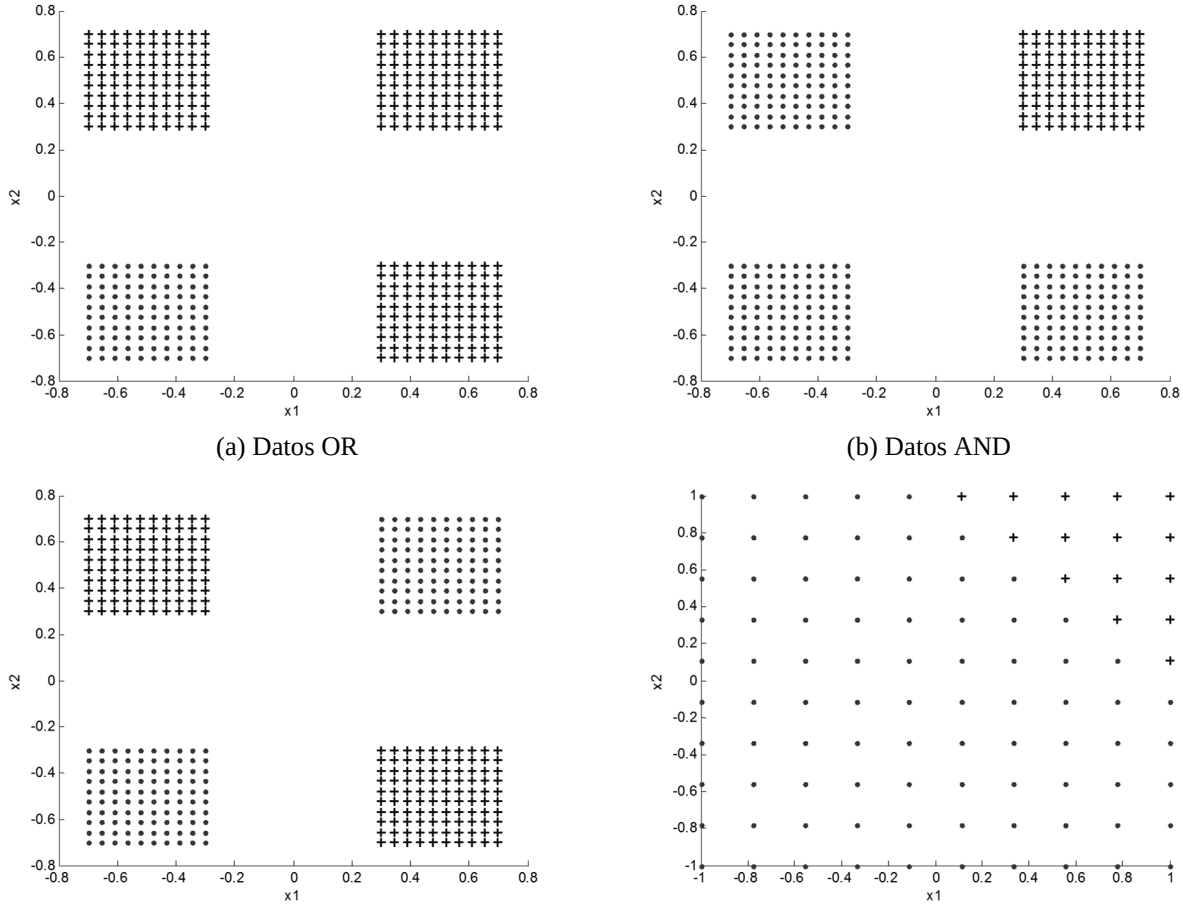
Tabla 1: regla para la construcción de los datos de prueba de las redes neuronales

Función lógica	X1	X2	Resultado
OR	Negativo	Negativo	0 (ó -1)
	Negativo	Positivo	1
	Positivo	Negativo	1
	Positivo	Positivo	1
AND	Negativo	Negativo	0 (ó -1)
	Negativo	Positivo	0 (ó -1)
	Positivo	Negativo	0 (ó -1)
	Positivo	Positivo	1
XOR	Negativo	Negativo	0 (ó -1)
	Negativo	Positivo	1
	Positivo	Negativo	1
	Positivo	Positivo	-1

Estas regiones implementadas en la Figura 6a, Figura 6b y Figura 6c están conformadas por cuatro sectores, con $0.3 \leq |x_i| \leq 0.7, i = 1,2$. Cada sector posee 100 datos, espaciados uniformemente y con suficiente espaciamiento entre las regiones para que las redes neuronales de prueba puedan conseguir superficies que puedan separarlas. En total, son cuatrocientos puntos.

También se utiliza una región en el plano separada linealmente, como se muestra en la Figura 6d, con datos en forma de una rejilla cuadrada en el plano cartesiano separadas en dos regiones, indicadas por

puntos y cruces. Estos datos están conformados por cien puntos distribuidos uniformemente,con $0 \leq |x_i| \leq 1, i = 1,2$.



(a) Datos OR (b) Datos AND

(c) Datos XOR (d) Datos "Región"

Figura 6: Representación gráfica de los datos de prueba para las redes neuronales

Las pruebas son etiquetadas con los nombres "OR", "AND", "XOR" y "Región" para hacer referencia al razonamiento utilizado para su selección. Aunque otras regiones pueden ser utilizadas (en un espacio n-dimensional), estas fueron elegidas de manera que puedan representarse gráficamente y que sus resultados puedan servir para aclarar el desempeño de las redes de clasificación en estudio.

C. Algoritmo de entrenamiento del preceptrón

Para el entrenamiento del perceptrón, se utilizó el algoritmo indicado a continuación [7]:

Entrada: se tienen n datos de entrada $x \in R^m$ *y salida* $d \in R$
Inicio: Los pesos de la neurona, $w \in R^{n+1}$ *son generados al azar*
Paso 1: para algún $i \in \{1,2, \dots n\}$ *calcular la salida de la neurona como* $y(i) = umbral(w \cdot [1\ x(i)])$ *donde* $umbral(p) = \begin{cases} 1\ si\ p > 0 \\ 0\ si\ p \leq 0 \end{cases}$
Paso 2: si $d(i) - y(i) \neq 0$ *entonces* $w = w + \big(d(i) - y(i)\big)[1\ x(i)]$
Paso 3: Volver a paso 1.

Este algoritmo debe ser programado con un criterio de parada o un conteo de repeticiones. Este algoritmo converge siempre que los datos sean linealmente separables, por lo tanto habrá un número finito de pasos a ejecutar en él.

D. Algoritmo de entrenamiento de Adaline

Para el entrenamiento de una red Adaline se utiliza el procedimiento descrito en [8] :

Entrada: se tienen n datos de entrada $x \in R^m$ *y salida* $d \in R$ *y se elige* $\eta \in (0,1)$
Inicio: Los pesos de la neurona, $w \in R^{n+1}$ *son generados al azar*
Paso 1: para algún $i \in \{1,2,\dots n\}$ *calcular la salida de la neurona como* $y(i) = umbral(w \cdot [1\ x(i)])$ *, donde* $umbral(p) = \begin{cases} 1\ si\ p > 0 \\ -1\ si\ p \leq 0 \end{cases}$
Paso 2: si $y(i) - d(i) \neq 0$ *entonces* $w = w + \eta(d(i) - w \cdot [1\ x(i)])[1\ x(i)])$
Paso 3: Volver a paso 1.

Como en el caso del perceptrón, Este algoritmo debe ser programado con un criterio de parada o un conteo de repeticiones. Este algoritmo converge siempre que los datos sean linealmente separables, por lo tanto habrá un número finito de pasos a ejecutar en él.

E. Algoritmo de entrenamiento de Madaline

Las redes Madaline tienen un algoritmo de entrenamiento diferente al de Adaline debido a que no se conoce la salida deseada de cada neurona. Este algoritmo está basado en el principio de mínima perturbación. [3]

Entrada: se tienen n datos de entrada $x \in R^m$ *y salida* $d \in R$ *y se elige* $\eta \in (0,1)$
Inicio: Los pesos de las k neuronas de la red neuronal, $(w_1, w_2, \dots w_k) \in R^{n+1}$ *son generados al azar*
Paso 1: para algún $i \in \{1,2,\dots n\}$ *calcular la salida de cada neurona k como* $y_k(i) = umbral(w \cdot [1\ x(i)])$ *, donde* $umbral(p) = \begin{cases} 1\ si\ p > 0 \\ -1\ si\ p \leq 0 \end{cases}$
Paso 2: calcular la salida de la red Madaline como M(t)=mayoría(y_k*)*
Paso 3: si si $M(i) - d(i) \neq 0$ *entonces determinar la neurona ganadora, que es esa que tiene la salida más cercana a cero pero con el signo incorrecto. Luego actualizar los pesos de esta neurona g según* $w_g = w_g + \eta\big(d(i) - w_g \cdot [1\ x(i)]\big)[1\ x(i)])$
Paso 4: volver al paso 1

Este algoritmo debe ser programado con un criterio de parada sobre el error de clasificación.

F. Estructura de los experimentos

Los datos de prueba descritos se suministran a las tres redes neuronales suministradas. Para la red Adaline, se presentan los datos en cien entrenamientos y se calcula el número de épocas promedio en que la red alcanza el objetivo de entrenamiento y se determina el tiempo promedio, en segundos, necesario para esto. De igual manera se procede usando el algoritmo de entrenamiento del perceptrón. Las desviaciones estándar del número de épocas de entrenamiento y del tiempo utilizado también son calculadas.

Para la red Madaline, se presentan los datos para diferentes arquitecturas, usando tres, cinco, siete y nueve neuronas en la red de una sola capa, para todos los tipos de datos en el plano descritos. De igual manera, se calcula el número promedio de épocas necesarias y el tiempo promedio para alcanzar el entrenamiento.

Todos los cálculos se realizan en una computadora portátil con procesador Intel i5, 4GB de memoria RAM, con sistema operativo Windows 10 y sobre la versión de MATLAB 2014a, trabajando fuera de linea y sin otros procesos, más que los mínimos requeridos para el funcionamiento del sistema operativo, funcionando.

IV. RESULTADOS

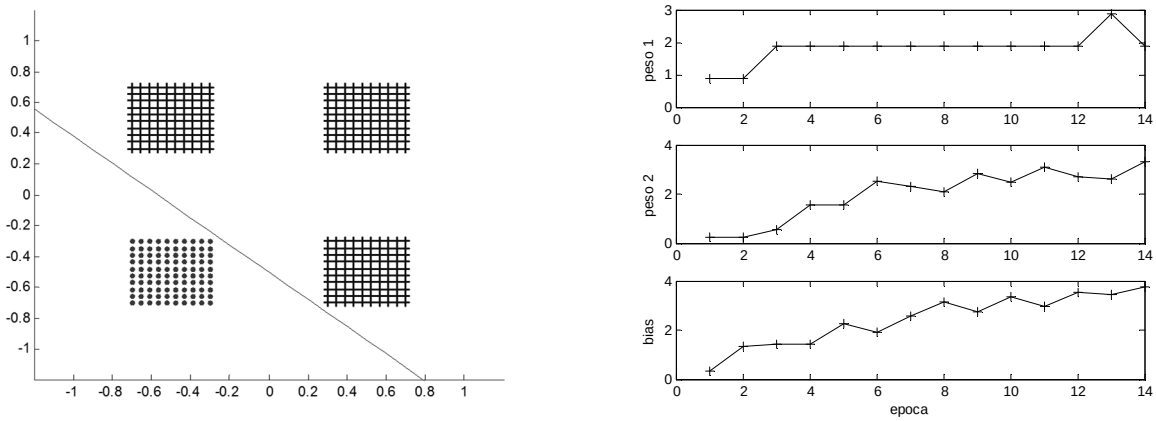
A. Entrenamiento de neurona Adaline y Perceptrón

Los grupos de datos comentados en la Figura 6 fueron suministrados a las redes neuronales. Debido a que una red Adaline y un Perceptrón solo pueden usarse para la clasificación de conjuntos linealmente separables [5] [7] [3], no se presentan los resultados para los datos etiquetados como XOR, ya que todos requieren el número máximo de épocas de entrenamiento dispuestas como criterio de parada, que fue de cien épocas. En la Tabla 2 se presentan los resultados obtenidos, con las desviaciones estándar de cada valor presentado entre paréntesis a su lado. Puede observarse que la red Adaline entrenada con factor η de 0.1 requirió, en todos los casos, más épocas de entrenamiento que el Perceptrón para todos los casos indicados, requiriendo mayor tiempo de entrenamiento en general, aunque ligeramente mayor. Los datos "Región" presentan un tiempo de cálculo inferior, debido a que la solución del entrenamiento no es abierta, es decir, que no tiene la libertad de selección de una superficie de clasificación tan amplia como los datos "AND" y "OR".

Tabla 2: Entrenamiento de Adaline y Perceptrón

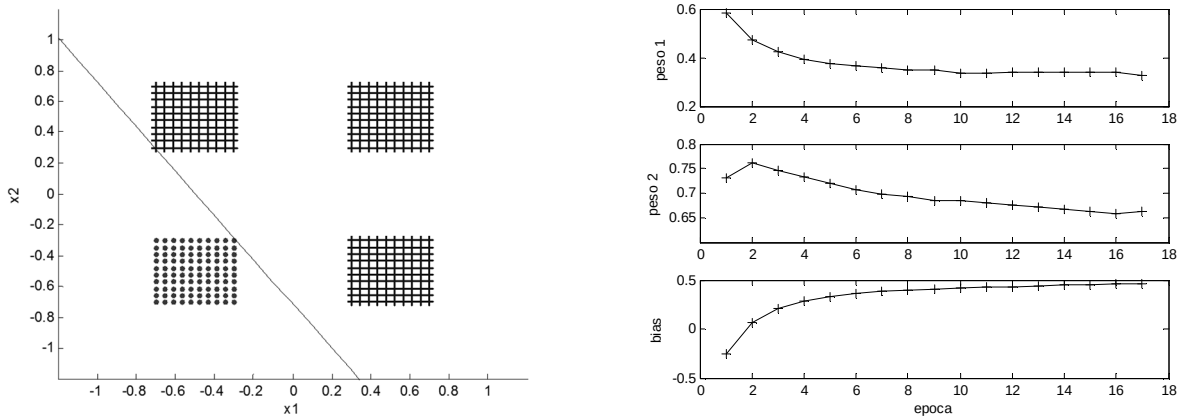
Datos	Tipo de red	Número de épocas promedio	Tiempo de cálculo promedio en segundos
OR	Adaline	9.45(3.84)	0.270 (0.104)
	Perceptrón	8.1(3.51)	0.261(0.114)
AND	Adaline	10.83(4.99)	0.304 (0.133)
	Perceptrón	7.39(3.83)	0.237 (0.123)
Región	Adaline	8.93(4.05)	0.068 (0.028)
	Perceptrón	4.58(2.25)	0.038 (0.018)

Para el entrenamiento usando los datos OR, una respuesta típica de la curva de clasificación obtenida y la evolución por época de los pesos de las neuronas es presentado en la Figura 7 para Perceptrón y en la Figura 8 para Adaline. Obsérvese que en ambos casos, la superficie de respuesta resultante logra clasificar los datos en dos grupos, a ambos lados de la recta que representa el frente de decisión de la neurona. En el caso del entrenamiento con la red Perceptrón, los pesos tienen una mayor variabilidad alrededor del valor final resultante, mientras que en la red Adaline, estos tienen un comportamiento más suave hacia el valor final. Esto es debido a la selección del factor de entrenamiento η y a que el entrenamiento Adaline tiene mejoras en el algoritmo respecto a Perceptrón [7] [4], entre las que se incluye la utilización del resultado de estimación lineal en la red Adaline que es mejor que la utilización directa de su salida implementado en perceptrón.



(a) superficie resultante de clasificación (b) avance de los pesos en un entrenamiento

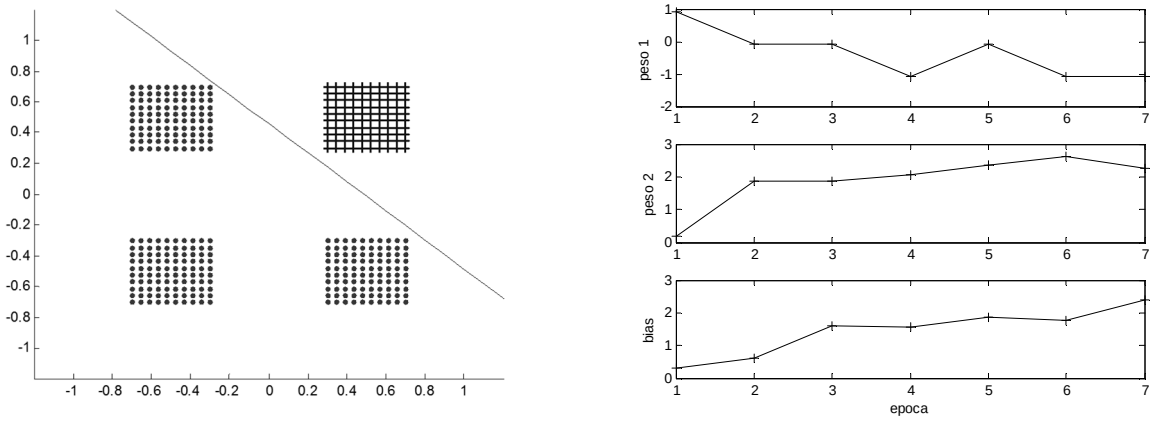
Figura 7: Resultados de un entrenamiento típico para un Perceptrón con los datos OR



(a) superficie resultante de clasificación (b) avance de los pesos en un entrenamiento

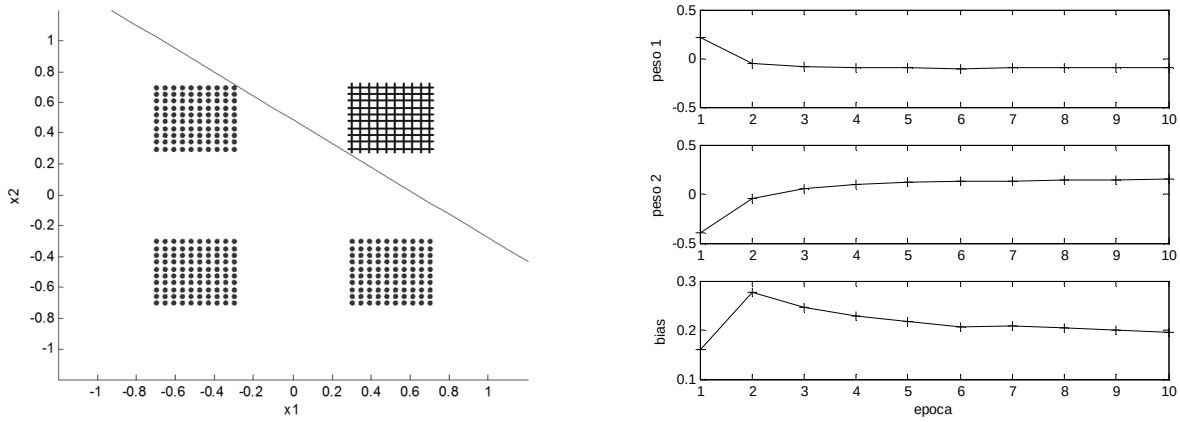
Figura 8: Resultados de un entrenamiento típico para Adaline con los datos OR

Para el entrenamiento con los datos AND, los resultados de un entrenamiento típico son similares a los mostrados para el caso OR, lo que puede verse en la Figura 9 para un perceptrón y en la Figura 10 para Adaline. Nuevamente, los pesos de la red evolucionan más suavemente para el entrenamiento Adaline, aunque los resultados definitivos muestran que ambas configuraciones obtienen una superficie de respuesta adecuada para el problema de clasificación.



(a) superficie resultante de clasificación (b) avance de los pesos en un entrenamiento

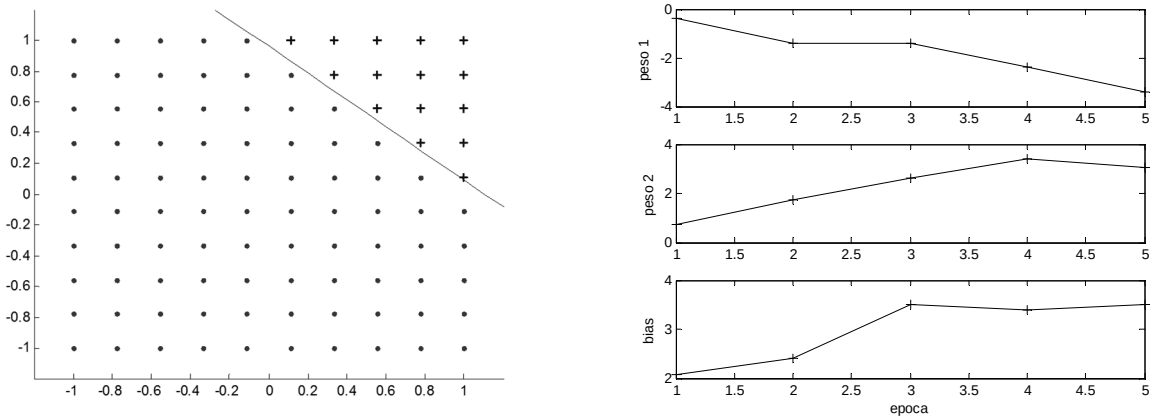
Figura 9: Resultados de un entrenamiento típico para un Perceptrón con los datos AND



(a) superficie resultante de clasificación (b) avance de los pesos en un entrenamiento

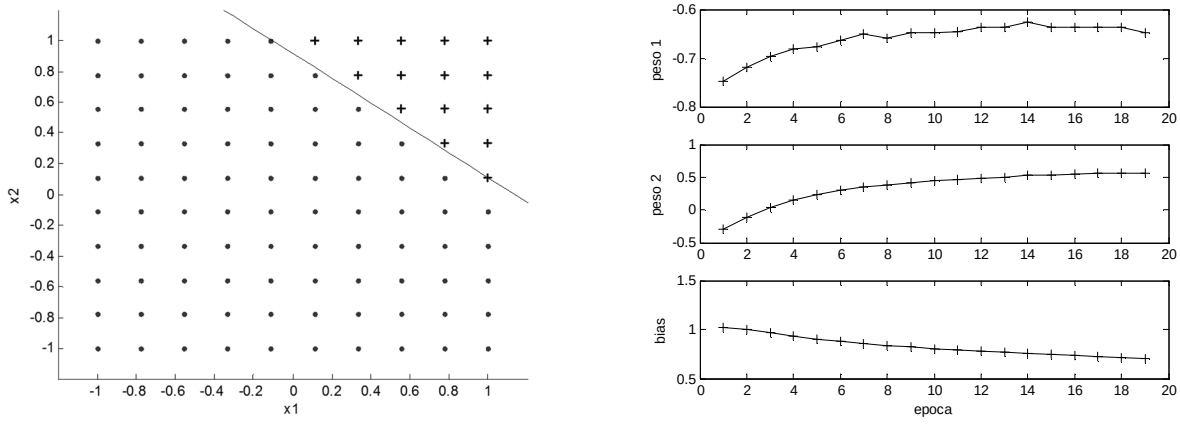
Figura 10: Resultados de un entrenamiento típico para Adaline con los datos AND

Respecto a la utilización de los datos “Región”, puede observarse que la curva de clasificación separa a las regiones correctamente para la red de Perceptrón en la Figura 11 y Adaline en Figura 12. Tambien puede observarse el mismo comportamiento discutido con el avance de los pesos en cada entrenamiento.



(a) superficie resultante de clasificación (b) avance de los pesos en un entrenamiento

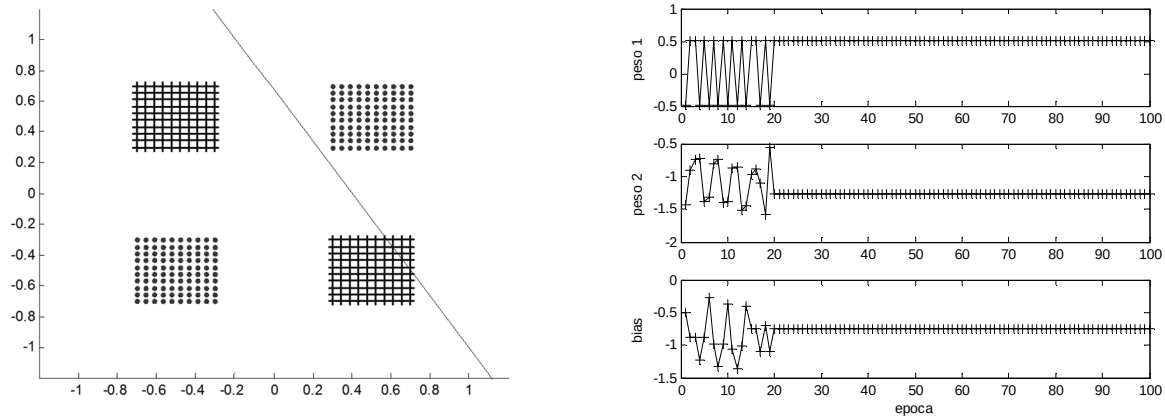
Figura 11: Resultados de un entrenamiento típico para un Perceptrón con los datos Región



(a) superficie resultante de clasificación (b) avance de los pesos en un entrenamiento

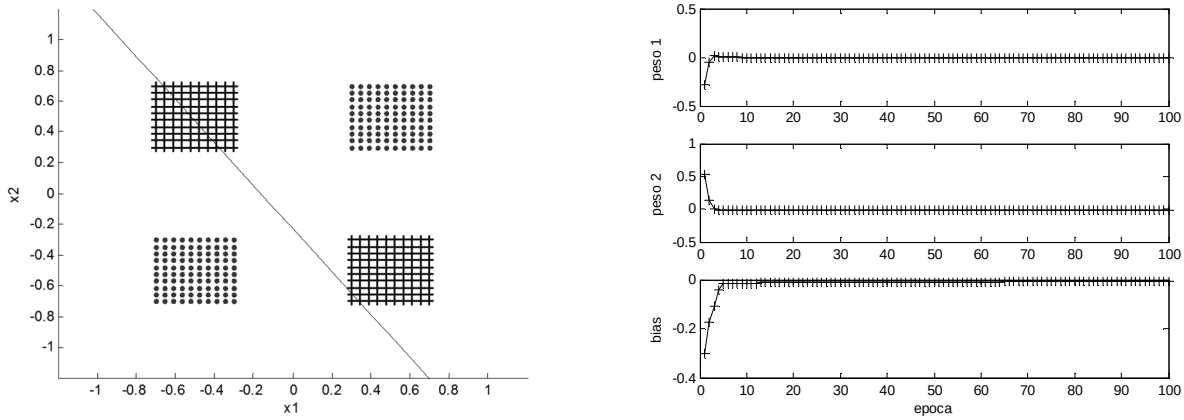
Figura 12: Resultados de un entrenamiento típico para Adaline con los datos Región

Cuando a las redes se le suministró el conjunto de datos correspondiente a “XOR”, tanto la red Perceptrón (Figura 13) y la red Adaline (Figura 14), se puede comprobar que ambas fallan en la tarea de separar los grupos en cuestión. Esto era de esperarse como ya se comentó previamente, sin embargo el avance de los pesos se observa que converge a un valor constante para no modificarse más luego de una búsqueda. La red Adaline converge todos sus pesos a cero mientras que la red Perceptrón converge a diferentes valores.



(a) superficie resultante de clasificación (b) avance de los pesos en un entrenamiento

Figura 13: Resultados de un entrenamiento típico para el Percetrón con los datos XOR



(a) superficie resultante de clasificación (b) avance de los pesos en un entrenamiento

Figura 14: Resultados de un entrenamiento típico para Adaline con los datos XOR

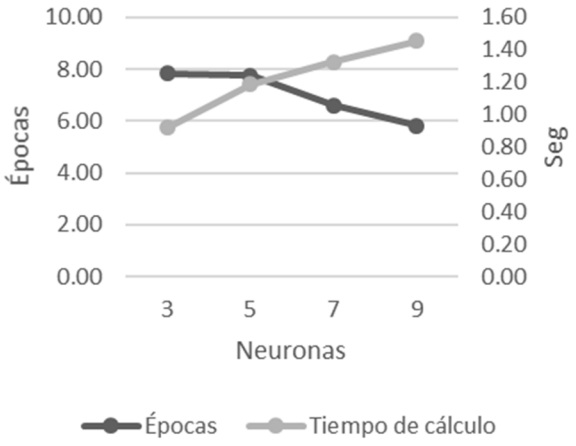
B. Entrenamiento de red Madaline

La red Madaline es una mejora al funcionamiento de la red Adaline y permite la clasificación no lineal de grupos a través de la clasificación en subgrupos. [3]. Para esta red, se hizo un estudio del entrenamiento para distintos tamaños impares de la capa (la capa debe tener elementos impares para que la función *mayoría* pueda tener una salida en cada utilización). El entrenamiento de esta red fue realizado cien veces en cada grupo y tamaño, y se presenta el promedio de épocas y tiempo requerido para el entrenamiento así como la desviación estándar de estos en la Tabla 3. Véase que a medida que se incrementa el tamaño de la red neuronal para los datos OR y AND suministrados, el número de épocas de entrenamiento disminuye, sin embargo, el tiempo de entrenamiento requerido aumenta. Esta tendencia puede observarse en Figura 15a y Figura 15b.

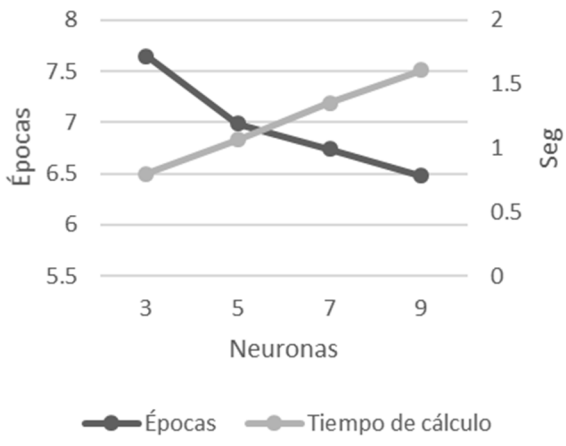
Tabla 3: Entrenamiento de Madaline

Datos	Número de neuronas (red Madaline)	Número de épocas promedio	Tiempo de cálculo promedio en segundos
OR	3	7.84(4.92)	0.920 (0.590)
	5	7.77(4.88)	1.187(0.739)
	7	6.62(4.30)	1.326 (0.849)
	9	5.82(3.99)	1.454(0.984)
AND	3	7.65(4.85)	0.799 (0.498)
	5	6.99(4.75)	1.065 (0.710)
	7	6.74(4.39)	1.353 (0.878)
	9	6.48(4.32)	1.610(1.056)
XOR	3	28.93(35.32)	2.956 (3.584)
	5	17.16(25.29)	2.596 (3.786)
	7	12.84(17.41)	2.553(3.415)
	9	10.11(10.26)	2.513(2.509)
Región	3	22.27(24.80)	0.597 (0.652)
	5	30.29(29.16)	1.177 (1.113)
	7	31.88(31.12)	1.643(1.571)
	9	32.15(28.52)	2.054(1.790)

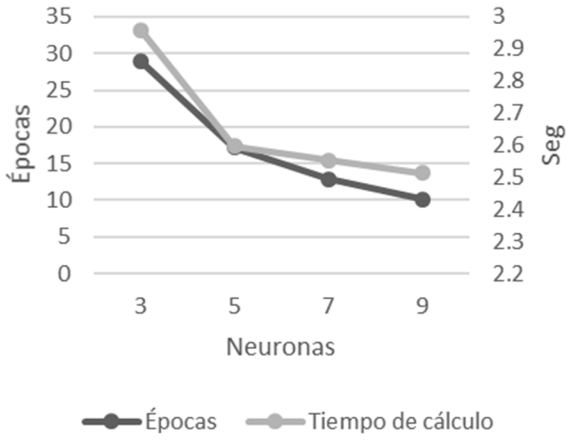
Respecto a los datos XOR, la disminución del número de épocas estuvo relacionada con la disminución del tiempo de ejecución. A medida que se incrementa el número de neuronas de la red, más fácil es para la red hacer la clasificación y menos tiempo es requerido, como se observa en la Figura 15c. Esto puede deberse a que al haber más grados de libertad en el entrenamiento, mayor cantidad de superficies de clasificación pueden establecerse en el plano que permitan una correcta clasificación por grupos.



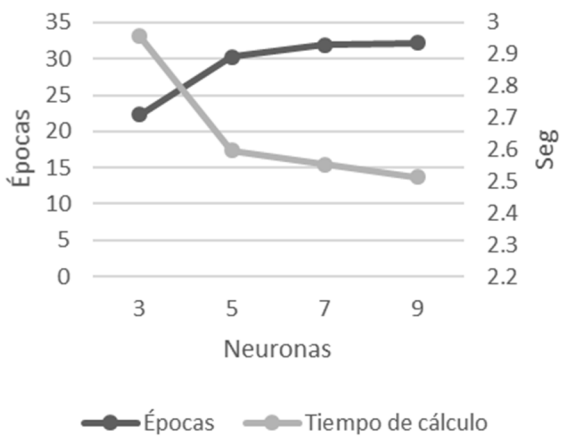
(a) Datos OR



(b) Datos AND



(a) Datos XOR

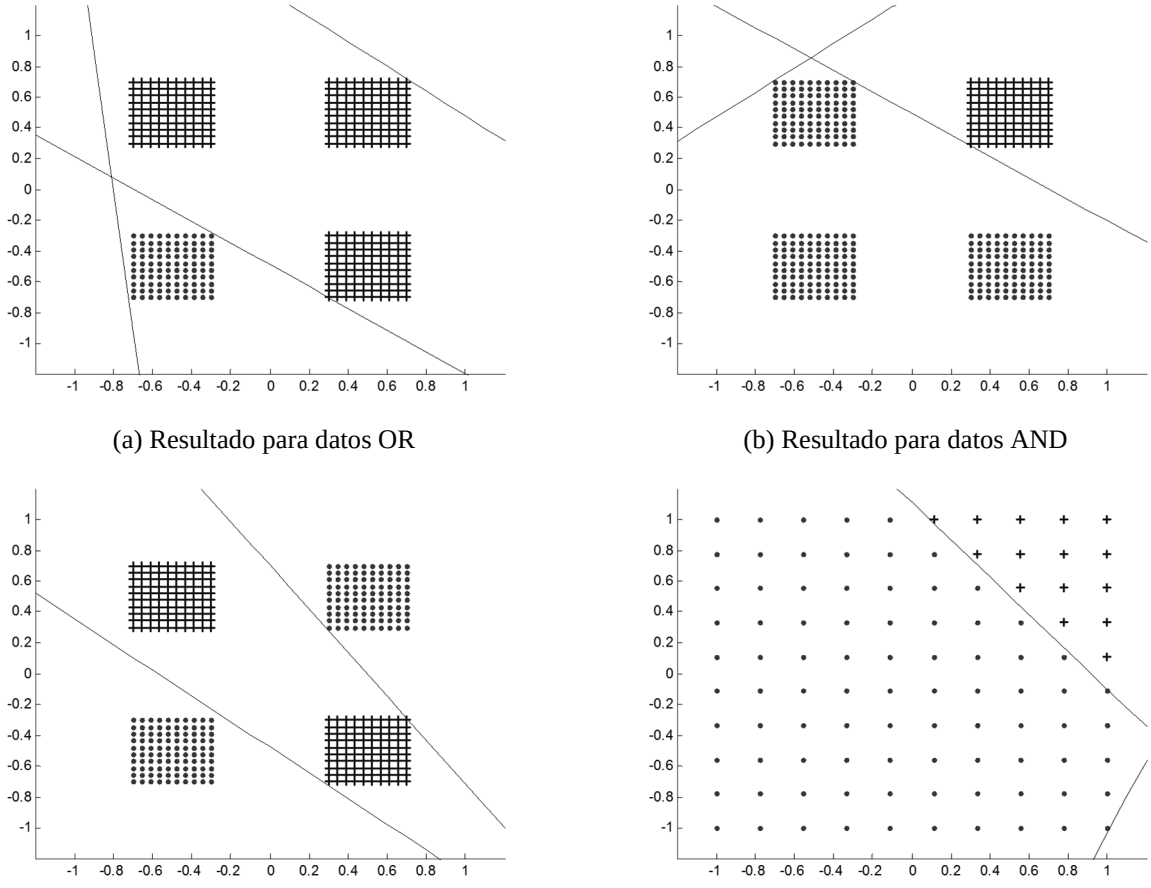


(b) Datos “Región”

Figura 15: Épocas y tiempo requerido para entrenamiento con red Madaline

El caso de la clasificación de los datos “Región” tiene un comportamiento diferente. El número de épocas promedio fue mayor a medida que se incrementó el tamaño de la red, también asociado con un incremento del tiempo de cálculo utilizado para ello. Es evidente, del manejo de mayor número de datos relacionado con más elementos en una capa, que el incremento de neuronas en ella conlleva a un mayor tiempo de cómputo, pero la ubicación de la superficie de decisión requiere ubicarla en trayectorias que no dividan a la región. Esto parece haber influido en el número de etapas requeridas al incrementar el número de neuronas porque el problema consiste en introducir más superficies en el plano que separen a la información suministrada en grupos.

En la Figura 16 se ejemplifican los resultados obtenidos con una selección de resultados para un entrenamiento usando la red Madaline de 3 neuronas. Por cada neurona adicional, una nueva recta aparece en el plano. La Figura 16(b) y (c) muestran solo dos líneas porque dos de ellas están superpuestas. Aunque la ubicación de las tres rectas no es la misma en cada entrenamiento, en cada caso se logró separar los datos en grupos. No se incluyen resultados en forma gráfica para otras arquitecturas con el fin de la brevedad, pero pueden determinarse usando los pesos resultantes de la red.



(a) Resultado para datos OR

(b) Resultado para datos AND

(c) Resultado para datos XOR

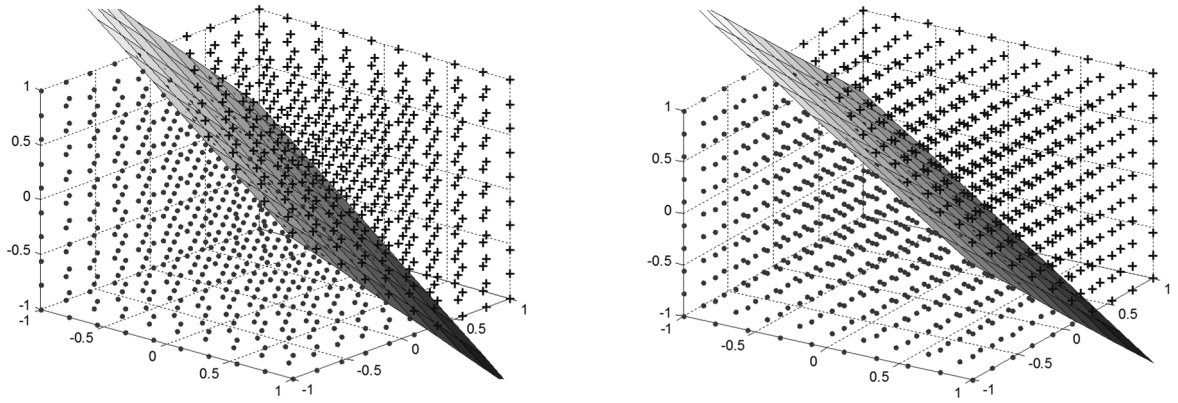
(d) Resultado para datos “Región”

Figura 16: Resultados de un entrenamiento típico para red Madaline con tres neuronas.

C. Otros experimentos

Las redes neuronales Perceptrón, Adaline y Madaline fueron usadas para la clasificación de grupos de datos de mayor tamaño o de mayor dimensión, sin embargo la representación gráfica para la

ejemplificación no produce resultados agradables visualmente. A pesar de esto, se eligió una región del espacio como generalización de los datos “Región” a tres dimensiones, y se presentan en la Figura 17 para una neurona Adaline y una neurona Perceptrón, prácticamente indistinguibles en las gráficas. En ambos casos, la figura muestra un plano que separa a los datos en dos grupos. La representación espacial de los resultados de una red Madaline son difíciles de apreciar porque involucran múltiples planos y por eso no se incluyen en este documento.



(a) Perceptrón (b) Adaline

Figura 17: Resultados de un entrenamiento para datos en tres dimensiones.

V. CONCLUSIONES

Los experimentos utilizados permitieron comprobar que las redes Perceptrón y Adaline no pueden hacer clasificación de conjuntos que no son separables linealmente, tal como es indicado en [2] [1] [8] [9]. Sin embargo, para conjuntos separables linealmente, ambas hacen una clasificación de los datos a través de la creación de una superficie de decisión, que está representada matemáticamente por una recta en el plano o un plano en el espacio. Esto se extiende al espacio multidimensional, requiriendo para la separación la determinación de un hiperplano.

Los experimentos muestran que, con factor de aprendizaje $\eta = 0.1$ para la red Adaline, el entrenamiento del Perceptrón requiere un número inferior de épocas y tiempo que Adaline, sin embargo, el proceso de búsqueda de Adaline es más suave que el de Perceptrón, lo que conduce a menor variabilidad de los pesos durante el procedimiento respecto a la época anterior.

Las redes Madaline pueden hacer clasificación de conjuntos que no son linealmente separables a través de la división del espacio en múltiples regiones representadas por múltiples rectas en el plano bidimensional. Esta clasificación es, por supuesto, una ventaja respecto a las redes de Perceptrón y Adaline pero requieren mayor cantidad de épocas y de tiempo para lograr realizar la tarea.

La relación entre el número de neuronas de una red Madaline y la velocidad de aprendizaje o el tiempo requerido para ello no es una relación lineal, y depende, como muestran los experimentos, del tipo de datos y del número de neuronas utilizadas.

VI. REFERENCIAS

[1] J. Heaton, Introduction to Neural Networks for C#, St. Louis: Heaton Research, Inc., 2008.

[2] C. Alavala, Fuzzy Logic and Neural Networks: Basic Concepts and applications, New Delhi: New Age International Publishers.

[3] D. Graupe, Principles of Artificial Neural Networks, Singapore: World Scientific Publishing, 2007.

[4] K. L. Priddy y P. E. Keller, Artificial Neural Networks - An introduction, Bellingham: SPIE Press, 2005.

[5] M. Hagan, H. Demuth y O. De Jesus, «An Introduction to the Use of Neural Networks in Control Systems,» *International Journal of Robust and Nonlinear Control,* vol. 12, nº 11, pp. 959-985, 2002.

[6] J. Freeman y D. Skapura, Neural Networks. Algorithms, Applications and Programming Techniques, Addison-Wesley Publishing Company, Inc., 1991.

[7] R. Rojas, Neural Networks. A Systematic Introduction, Berlin: Springer, 1996.

[8] B. Yegnanarayana, Artificial Neural Networks, New Delhi: Prentice-Hall, 2005.

[9] A. Poznyak, E. Sanchez y W. Yu, Differential Neural Networks for Robust Nonlinear Control, Singapore: World Scientific Publising, 2001.

[10] B. Müller y J. Reinhardt, Neural Network - An introduction, Frankfurt: Springer-Verlag, 1990.