

**Instituto Politécnico Nacional**

**Escuela Superior de Cómputo**

*Asignatura*:

Neural Networks

*Grupo*: 3CM2

Práctica .4 **Red ADALINE**

*Alumno*:

Garcia Garcia Rafael

*Profesor*: Moreno Armendariz Marco Antonio

**Introducción:**

En la práctica pasada se explicó del nacimiento, funcionamiento y comportamiento de la red perceptrón y como puede ser aplicada a problemas sencillos. Una red muy similar es la Adaptative Linear Network o mejor conocida como red ADALINE. A finales de los años 50 Bernard Widrow empezaba a trabajar en redes neuronales, en el tiempo en el que Frank Rosenblatt desarrolló la regla de aprendizaje del perceptrón, años después, en 1960 se presenta la red ADALINE creada por Widrow y su asesorado, asimismo presentaron una regla de aprendizaje la cual denominaron algoritmo LMS (Least Mean Square).

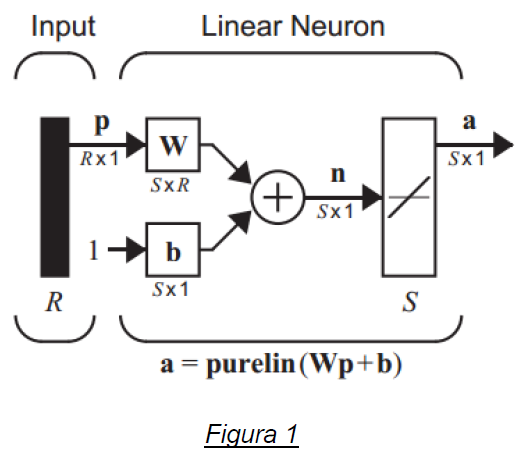
La red ADALINE al tener similitudes al perceptrón y al ser su única diferencia que usa una función de transferencia linear en vez de hard limit, presenta las mismas

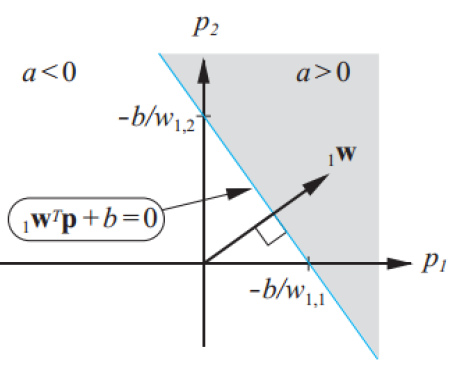
problemáticas: Solo pueden resolver problemas linealmente separables.

**Marco teórico.**

***Explicación de la arquitectura,***

ADALINE fue desarrollada por el profesor Bernard Widrow y su alumno Ted Hoff en la Universidad de Stanford en 1960. Como se observa en la figura 1 (Hagan), tiene la misma arquitectura que el perceptrón a diferencia de la función de transferencia purelin().

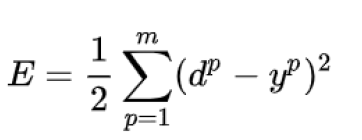




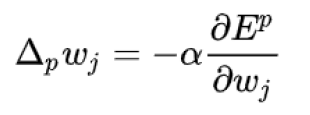
La salida de la neurona es mayor que 0 en el área gris. En el área blanca la salida es menor que cero. Ahora, ¿qué implica esto sobre el ADALINE?

Dice que ADALINE se puede usar para clasificar objetos en dos categorías. Sin embargo, solo puede hacerlo si los objetos son linealmente separables. Por lo tanto, a este respecto, el ADALINE tiene la misma limitación que el perceptrón.

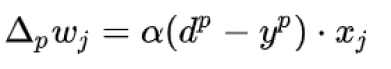
Esto nos lleva a poder hablar más de la regla de aprendizaje de la red ADALINE: El algoritmo LMS, Dado que el objetivo del ADALINE es poder estimar de la manera más exacta la salida, se busca minimizar la desviación de la red para todos los patrones de entrada, eligiendo una medida del error global. Normalmente se utiliza el error cuadrático medio.



La manera de reducir este error global es ir modificando los valores de los pesos al procesar cada entrada, de forma iterativa, mediante la regla del descenso del gradiente. Suponiendo que tenemos una constante de aprendizaje alfa.



Si operamos con la derivada, queda:



Que será la expresión que utilizaremos por cada entrada para modificar los pesos. Con respecto al perceptrón, la red ADALINE posee la ventaja de que su gráfica de error es un hiperparaboloide que posee o bien un único mínimo global, o bien una recta de infinitos mínimos, todos ellos globales. Esto evita la gran cantidad de problemas que da el perceptrón a la hora del entrenamiento debido a que su función de error (también llamada de coste) posee numerosos mínimos locales.

**Diagrama de Flujo:**

**Experimentos.**

**Modo regresor:**

Experimento 1:

Para este experimento se tuvo que desarrollar una función el cuál pudiera crear un codificador de 8 bits, este codificador se mete como entrada para el entrenamiento de nuestra red neuronal ADALINE. Para este ejemplo decidimos poner un valor de EEPOCH de 0.001, un número de EpochMax = 10 y Alpha de 0.04 para ver si nuestra red converge.

Se realiza el experimento y como se observa en las gráficas la red converge de manera muy rápida, además de que el valor de Eepoch desciende de manera significativa

Experimento 2:

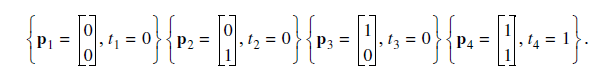
Ahora se empleó un codificador de 12 bits, este codificador se mete como entrada para el entrenamiento de nuestra red neuronal ADALINE. Para este ejemplo decidimos poner un valor de EEPOCH de 0.001, un número de EpochMax = 10 y Alpha de 0.04 para ver si nuestra red converge.

De igual manera como en el ejercicio anterior convergió muy rápido, además de que los resultados a la hora de abrir la salida es bastante parecido a el dataset con el que se entrenó las

**Modo clasificador**:

Experimento 1:

Este primer experimento se realizó con el Dataset de la compuerta AND por lo cuál el data set de entrada es el de la figura 2 (Hagan) donde sustituiremos los 0’s por -1 ya que por la función de transferencia de Adaline tenemos valores de -1 a 1:

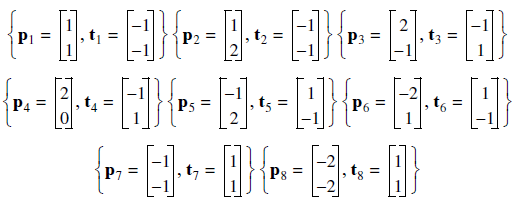


*Figura 2*

La salida de la red es la siguiente:

Experimento 2:

Para el segundo ejemplo se usa un data set compuesto de 4 clases como el que se muestra en la figura 3 (Hagan).



*Figura 3*

Para este ejercicio se requirieron 2 neuronas debido al número de clases, estas neuronas tienen un vector de pesos los cuales definirán la frontera de decisión de nuestro ejercicio las cuales estarán ilustradas en las siguientes gráficas junto a los valores de EEPOCH y de pesos y bias.

Experimento 3:

Este es el último experimento de nuestra donde tenemos un dataset que no es linealmente separable, se puede obviar que nuestra res ADALINE no va a converger correctamente debido a esta limitante por lo cuál se verá reflejado en nuestro entrenamiento.

Si observamos los valores de pesos y bias en este ejercicio observaremos que los valores oscilan los unos a los otros lo cual como se vio en clase es señal de que el problema no es linealmente separable y requerimos muchas más capas para darle solución.

**Discusión:**

Como pudimos observar en nuestros experimentos la parte teórica vista en nuestras clases fue de gran utilidad para poderlas realizar además de que tuvimos una manera más visual de observar como una red como ADALINE que es para problemas lineales no puede ayudarnos mucho en problemas no lineales ya que los valores de pesos y bias oscilan. Esto para otro tipo de problemas nos será útil porque podremos identificar cuando nuestra red neuronal tiene un problema de bias(diseño) o varianza.

Ahora si retomamos lo visto la red Adaline comparada con el perceptrón tiene muchas más implementaciones y su regla de aprendizaje nos dio mejores resultados que el perceptrón de una sola capa, esto es importante porque también sabremos cuando usar una ante la otra.

**Conclusiones:**

Es notoria la diferencia con la red perceptrón a pesar de tener un gran parecido lo que nos lleva al pensamiento del peso de las funciones de transición ya que es lo único que cambian, sin embargo, eso hace que cambie totalmente la manera de resolver los problemas y la forma de programar los mismos. Es necesario entender el primero para tener una mayor facilidad de programar una red ADALINE ya que sirven como un gran complemento y de igual forma, muchas funciones que ya se habían utilizado en la red perceptrón son útiles también en esta práctica.

# Bibliografía

Hagan, M. T. (s.f.). Neural Network Design. En M. T. Hagan, *Neural Network Design* (pág. 1010).

Pereira, U. T. (2000). *http://www.medicinaycomplejidad.org/pdf/redes/Competitivas.pdf*. Obtenido de medicinaycomplejidad.