**Adaline Neuronal Network**

**Pablo Saavedra López**

Implementación de un programa que simule una estructura de capa de Adalines, aplicando en su entrenamiento la regla LMS.

Asociación de Patrones

Caso 1:

Se presentan los patrones de entrada (3,1) y (1,4) con salidas deseadas respectivas (4,5). Se utilizaran las velocidades de aprendizaje 20 y 0.025 .Los PE's no tienen tendencia y el error máximo permitido es 0.0001

Con velocidad de aprendizaje igual a 20, el EMC tiende a crecer cada vez más. La estructura no es capaz de encontrar un mínimo en la superficie de error, ya que la modificación de los pesos es muy grande de una iteración a otra.

Con velocidad de aprendizaje igual a 0.025 la estructura entrena perfectamente en aproximadamente 10 iteraciones. Los pesos de las entradas tienden a 1, porque, si la función de salida es

Neta = ((X1\*W1) + (X2\*W2))

y los pesos son 1, entonces la salida deseada es la suma de las entradas.

Al probar la neurona se aprecia que las salidas se aproximan mucho al resultado correcto.

Caso 2:

En este caso se realiza lo mismo que en el anterior, pero la neurona tiene tendencia. En este caso la función de salida es

Neta = ((X1\*W1) + (X2\*W2) + W0)

donde W0 es el peso de la tendencia.

Cuando se realiza el entrenamiento con velocidad de aprendizaje 20 pasa lo mismo que en el caso anterior. Pero al entrenar con velocidad de aprendizaje 0.025 la neurona entrena en aproximadamente 10 iteraciones.

En este caso, al probar la neurona, el error de la salida es más grande cuanto más diferentes son las entradas de los patrones de entrenamiento.

Caso 3:

En este caso se utilizan 10 patrones de entrenamiento:

(5,3) -> 8 (2,4) -> 6 (3,2) -> 5 (1,6) -> 7 (7,2) -> 9

(1,5) -> 6 (1,2) -> 3 (4,2) -> 6 (3,3) -> 6 (2,6) -> 8

Cuando se realiza el entrenamiento con velocidad de aprendizaje 20 pasa lo mismo que en los casos anteriores.

Pero si utilizamos 0.025 la estructura entrena más rápidamente (aprox. 4 iteraciones) porque la muestra de patrones es mayor.

Caso 4:

Realizamos lo mismo que en el caso 1, pero cambiamos los patrones de entrenamiento por los patrones (5,3) y (2,4) con salidas deseadas respectivas (8,6). Cuando se realiza el entrenamiento con velocidad de aprendizaje 20 pasa lo mismo que en los casos anteriores. Con velocidad de aprendizaje 0.025 la estructura entrena.

Si se entrena la neurona sin tendencia, ocurre lo mismo que en el caso 1.

He aquí varios ejemplos de este caso:

Peso Entrada 1 Peso Entrada 2

Entrenamiento 1 => 1.0009761479E+00 9.9951192606E-01

Entrenamiento 2 => 1.0008571055E+00 9.9957144727E-01

Entrenamiento 3 => 9.9914289455E-01 1.0004285527E+00

Si se entrena con tendencia ocurre lo mismo que en el caso 2. Veamos unos cuantos ejemplos de de este caso:

Peso Entrada 1 Peso Entrada 2 Peso Tendencia

Entrenamiento 1 => 9.3637284040E-01 8.1087939638E-01 8.8350078782E-01

Entrenamiento 2 => 9.2344816902E-01 7.6656103023E-01 1.0873665186E+00

Entrenamiento 3 => 8.6842064025E-01 6.0699245778E-01 1.8349570010E+00

Tratamiento de Señales

Caso1:

Este caso trata de asociar una señal de entrada a una de salida. La señal de salida elegida ha sido y=4x. Para la estructura se utiliza una sola entrada en la neurona y no se utiliza tendencia.

Los patrones de entrenamiento son 10 y son los siguientes:

Patrones de entrada (1),(2),(3),(4),(5),(6),(7),(8),(9),(10) y

Salidas deseadas (4 8, 12, 16, 20, 24, 28, 32, 36, 40).

La velocidad de aprendizaje utilizada es 0.025 y el error m ximo permitido es 0.0001. Tras 16 iteraciones aproximadamente la estructura entrena, y el peso de la única entrada es muy cercano a 4. (P.ej. W1 = 3.9958671597E+00)

Ya que la función de salida es Neta = (X1\*W1) al ser W1=4 entonces

Salida = 4\*Entrada

En la gráfica se superponen la función de salida deseada y obtenida, debido al la buena aproximación de la neurona.

Caso 2:

La señal a generar será y = cos(x)\*(1/x). Para generarla se utilizará una estructura de Adaline con 5 entradas sin tendencia.

El conjunto de patrones para el entrenamiento de la neurona son los valores del muestreo de la señal en el intervalo (1,25). En este intervalo la señal es continua y variante a lo largo del tiempo.

Para entrenar la neurona se le aplicaran cada vez cinco valores consecutivos de la señal muestreada y como salida deseada se tendrá el siguiente valor a esos cinco. En este caso se realizan 100 muestreos de la señal para el entrenamiento. El error máximo permitido es de 0.0001 y la velocidad de aprendizaje es de 0.1