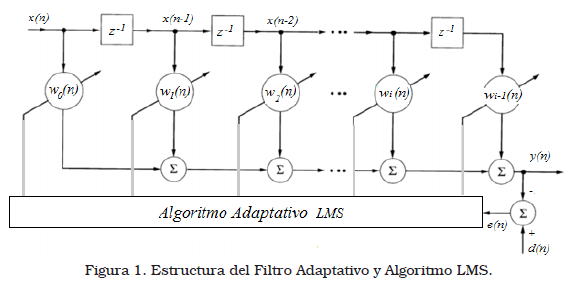
**Informe del LMS**

El **algoritmo LMS** (del inglés, *Least-Mean-Square algorithm*) se usa en filtros adaptativos para encontrar los coeficientes del filtro que permiten obtener el valor esperado mínimo del cuadrado de la señal de error, definida como la diferencia entre la señal deseada y la señal producida a la salida del filtro.

Pertenece a la familia de los algoritmos de gradiente estocástico, es decir, el filtro se adapta en base al error en el instante actual únicamente. Fue inventado en 1960 por el profesor de la Universidad de Stanford Bernard Widrow y su primer estudiante de doctorado, Ted Hoff. Este algoritmo consiste de dos procesos básicos:

* Un *proceso de filtrado*, que involucra:
  + el cómputo de la salida de un filtro lineal en respuesta a una señal de entrada, y
  + la generación de una estimación del error mediante la comparación de esta salida con la señal deseada.
* Un *proceso adaptativo*, que involucra el ajuste automático de los parámetros del filtro de acuerdo al error estimado.

Cuando se habla de filtros adaptativos, está implícito que los parámetros que caracterizan al filtro, tales como el ancho de banda y frecuencias de los ceros, entre otros, cambian con el tiempo, esto es, los coeficientes de los filtros adaptativos cambian con el tiempo, en contraposición a los coeficientes de los filtros fijos que son, teóricamente, invariantes con el tiempo.



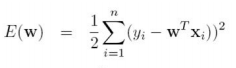
Este tipo de algoritmo no requiere de una función de activación, las funciones de activación son normalmente usadas para los sistemas de clasificación no para los sistemas de predicción.

Posteriormente, implementamos el funcionamiento del algoritmo LMS, eligiendo el algoritmo aproximado no el matemáticamente riguroso, debido a que encontrar matrices inversas no siempre es posible o analíticamente tener que y (salida) sea una combinación lineal de las características o vector de entradas X, es un poco incómodo, entonces nuestro algoritmo consiste en:

1. Es un proceso iterativo donde los pesos se inicializan en un valor aleatorio, recordemos que hay n+1 elementos de pesos siendo n la cantidad de valores o características de entrada.
2. Buscamos el gradiente “Ir colina hacia abajo”, si bien, el gradiente en un punto de nuestra función estará en la dirección para la que hay un mayor grado de inclinación. La magnitud del gradiente nos mostrará cuán empinada se encuentra la pendiente, es decir, encontrar el gradiente, estamos analizando hacia donde tiende nuestro próximo dato según el valor de la función en cierto punto, entonces los pesos se van actualizando de esta manera:



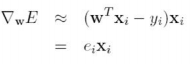
Pero no se calcula exactamente, debido a que nuestra función tal vez tenga cambios bruscos o analíticamente no sea rigurosa entonces procedemos a analizar que, siendo el error E:



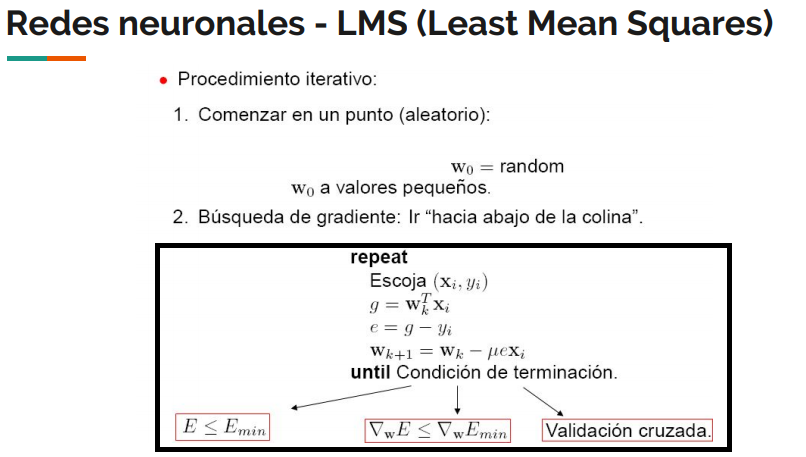
Entonces la derivada del error con respecto al vector de pesos, vendría siendo:



Tal que para un solo dato entonces se tiene que:



Entonces procedemos a implementar el siguiente código:



En especial el cuadro seleccionado, este se encuentra en un círculo while en nuestro código, después del subtítulo: “Implementando LMS (Least Mean Squares - Mínimos Cuadrados Medios) y Función de Validación”.

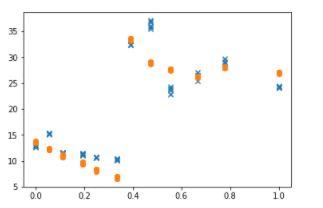
De hecho, nuestro código consta de los siguientes pasos:

1. Obteniendo y normalizando dato.
2. Dividir la matriz de datos de entrada para entrenamiento.
3. Implementando LMS (Least Mean Squares - Mínimos Cuadrados Medios) y Función de Validación.
4. Obteniendo un Alpha Óptimo.
5. Analizando el Alpha Óptimo.
6. Visualizar rendimiento final.

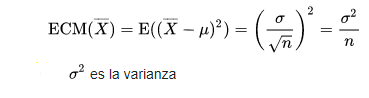
En la forma en la cual culminamos nuestro procedimiento, puede ser entonces por cantidad de iteraciones o por superar un porcentaje de validación, realizado por el error (ECM) que después de analizar entonces obtuvimos un error mínimo:

Los puntos 4 y 5, es porque en la expresión: 

La forma en cómo se varían los pesos y se obtienen unos pesos óptimos dependen de miu que en mi caso lo trate como alpha (Siguen siendo solo una variable de representación) su valor se encuentra entre 0 y 1, después de analizar, se culmino viendo que el valor de error fluctúa menos y es mejor si se encuentra entre: 10\*\*-2 y 10\*\*-1, nuestro alpha adecuado fue de 0.064



La anterior gráfica se encuentran los valores predichos y los esperados y al ver que un error medio cuadrático es bajo para los valores finales y teniendo en cuenta su relación con la varianza, se puede concluir que para este ejercicio la predicción se encuentra estimada adecuadamente entre un intervalo.



Así que el LMS es un tipo de algoritmo deseado para funciones sobre las cuales sus datos y sus tendencias están muy marcadas.

La predicción fue adecuada.