



Universidad Simón Bolívar  
Departamento de Computación y Tecnología de la Información  
CO-6612, Introducción a las redes neuronales  
Trimestre Enero-Marzo 2023

Eros Cedeño 16-10216

## Tarea 3: Adaline LMS

El perceptrón es una excelente herramienta para clasificar datos linealmente separables, sin embargo este busca ajustar los pesos sinápticos tal que las respuestas deseadas tengan valores enteros, sin embargo el poder de perceptrón se puede extender a valores continuos como respuestas deseadas. Esto trae consigo ciertas connotaciones, entre ellas por ejemplo que la función de activación pasa de ser la función signo a convertirse en la función lineal o función identidad. Además, el adaline permite extender los usos de la red neuronal, permitiéndole ser usada para realizar regresiones lineales o representación de modelos autorregresivos. Para esta tarea se realizó la implementación del algoritmo del LMS para construir un dispositivo Adaline (ADAPtative LINear Element) en el lenguaje de programación Python 3. Se creó además un programa cliente que entrena dos neuronas para clasificar entre dos categorías de datos cada una. La primera de ellas clasifica entre datos de Ciencias de la Tierra y el espacio VS Ciencias Médicas, y la segunda clasifica entre Ciencias de la vida VS Agricultura.

Se creó adicionalmente un dispositivo Adaline para interpolar el conjunto de datos del archivo datosT3 - datosT3.csv.

A continuación se muestra la salida estándar al ejecutar el programa cliente del adaline con el comando

```
$ python3 adaline/lms.py
```

```

(env) eycer@debian:~/Desktop/EneMar2023/Intro Redes Neuronales/Codigo/tarea2$ python3 lms.py
### Ciencias de la tierra y el espacio vs Ciencias medicas ###
Error cuadratico medio con tasa: 0.001
0.3146452703330074
Error cuadratico medio con tasa: 0.01
0.17952724346396437
Error cuadratico medio con tasa: 0.1
0.024896577087465253
Error cuadratico medio con tasa: 0.25
0.012040327358686733

### Ciencias de la vida vs Agricultura ###
Error cuadratico medio con tasa: 0.001
0.32381983427206823
Error cuadratico medio con tasa: 0.01
0.04555939138108818
Error cuadratico medio con tasa: 0.1
0.005375444600030511
Error cuadratico medio con tasa: 0.25
0.0026964852582943397

### Interpolador ###
Error cuadratico medio con tasa: 0.001
0.18201106922532875
Error cuadratico medio con tasa: 0.01
0.17873690796387767
Error cuadratico medio con tasa: 0.1
0.0795942391185194
Error cuadratico medio con tasa: 0.25
0.07555763664494011

```

Cada dispositivo se entrenó durante 100 épocas máximas y se varió la tasa de aprendizaje entre los valores 0.001, 0.01, 0.1 y 0.25.

En comparación con la tarea anterior, el adaline no busca clasificar correctamente todos los datos recibidos, sino que busca conseguir un hiperplano tal que minimice lo mas posible el error de la clasificación. Con el perceptrón y los mismos estímulos se obtuvieron resultados muy positivos de hasta la clasificación correcta del 98% de los datos, sin embargo si no fuera por el máximo de épocas impuesto no tendríamos cómo asegurar que el algoritmo convergerá, esto porque no tenemos certeza si los datos a clasificar son linealmente separables o no. Por otra parte los datos con errores de clasificación en el perceptrón no tienen ni siquiera aproximaciones al valor esperado pero en el caso del adaline con el LMS se puede esperar una aproximación aunque no sea muy precisa.

Es importante señalar que aunque en el perceptrón la convergencia estaba garantizada para cualquier tasa de aprendizaje  $\eta$  por grande o pequeña que sea siempre y cuando los datos sean linealmente separables, en el caso del Adaline no es así. La escogencia de la tasa de aprendizaje es decisiva para lograr minimizar el error cuadrático medio que es justamente el objetivo del LMS. En los casos estudiados se observa como algunas tasas son mejores para minimizar el error cuadrático. Pero como contraste, se

ejecutó el mismo algoritmo sobre los mismos datos pero fijando la tasa de aprendizaje en 1.2.

```
(env) eycer@debian:~/Desktop/EneMar2023/Intro Redes Neuronales/Codigo/tarea2$ python3 lms.py
### Ciencias de la tierra y el espacio vs Ciencias medicas ###
Error cuadratico medio con tasa: 1.01
328773.87518159713

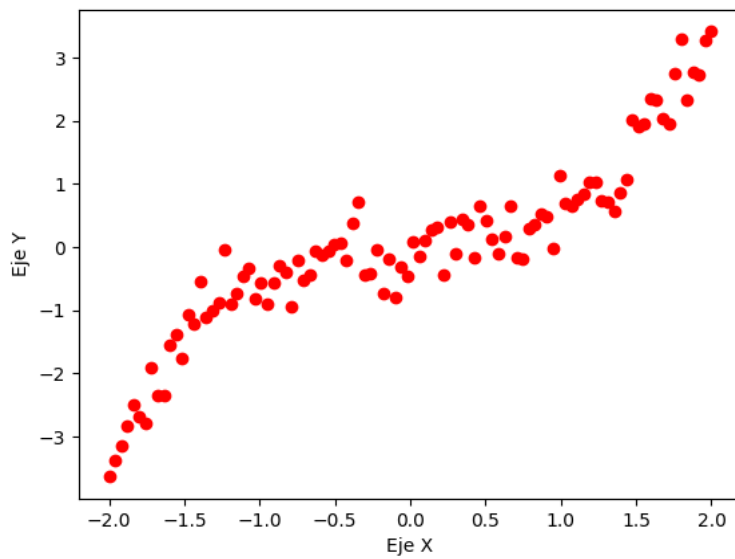
### Ciencias de la vida vs Agricultura ###
Error cuadratico medio con tasa: 1.01
4.066141940853883e+22

### Interpolador ###
/home/eycer/Documents/USB/EneMar2023/Intro Redes Neuronales/Codigo/tarea2/LMSStochastic.py:37: Runtime
Warning: overflow encountered in double_scalars
    costo = 0.5 * error**2
/home/eycer/Documents/USB/EneMar2023/Intro Redes Neuronales/Codigo/tarea2/LMSStochastic.py:36: Runtime
Warning: invalid value encountered in add
    self.pesos += self.eta * xi.dot(error)
Error cuadratico medio con tasa: 1.01
nan
```

Acá se observa que para clasificar entre Ciencias de la tierra y el espacio VS Ciencias médicas el error cuadrático medio se mantiene en unos escandalosos  $3,28 \times 10^5$  unidades. Al igual Ciencias de la vida VS Agricultura con un error de  $4.06 \times 10^{22}$ . Y en el caso del interpolador superamos la capacidad de representación máxima de los flotantes y ocurrió overflow de los datos.

Con esto se muestra que la escogencia del  $\eta$  no debe ser muy grande porque de otro modo el error aumenta en vez de disminuir, pero tampoco debe ser muy pequeño o si no la tendencia de disminución del error sería muy lenta. Para estos entrenamientos se realizaron pruebas con diferentes tasas  $\eta$  y se vio que en las dos primeras clasificaciones la tasa de 0.1 logra disminuir a un mínimo muy cercano al cero. Y en el caso de la interpolación se logra un error cuadrático medio muy cercano a cero de 0.075 con la tasa de  $\eta = 0.25$ .

Se implementaron los LMS con actualizaciones por lotes y también se implementó con actualizaciones en línea o estocástica. Y se adjunta con este informe el código de ambas implementaciones. Se observó adicionalmente que el método estocástico aunque teóricamente es menos preciso, operacionalmente mostró una convergencia más rápida

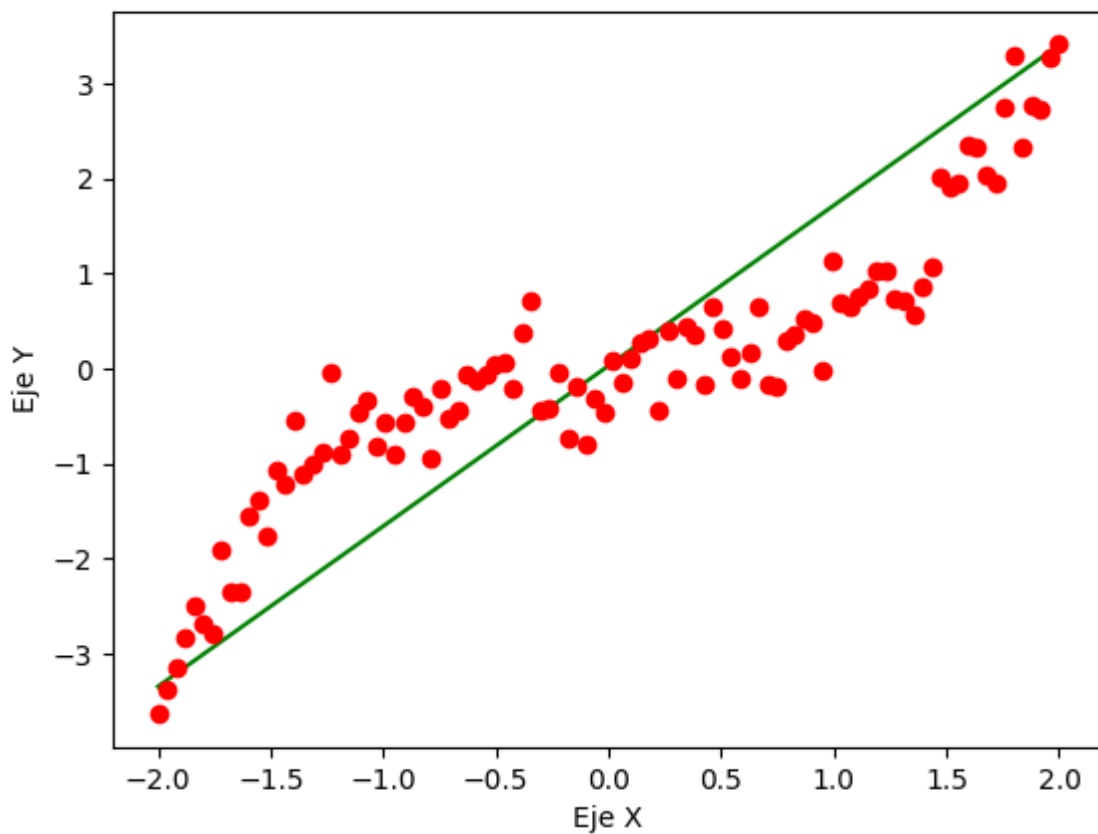


que al actualizar por lotes dado que se actualizan los pesos cada vez que se evalúa un estímulo.

Luego se procedió a interpolar los datos recibidos para tal fin. Para conocer la naturaleza de estos datos se graficaron los puntos recibidos en el archivo datosT3 - datosT3.csv, esta gráfica se muestra a la izquierda.

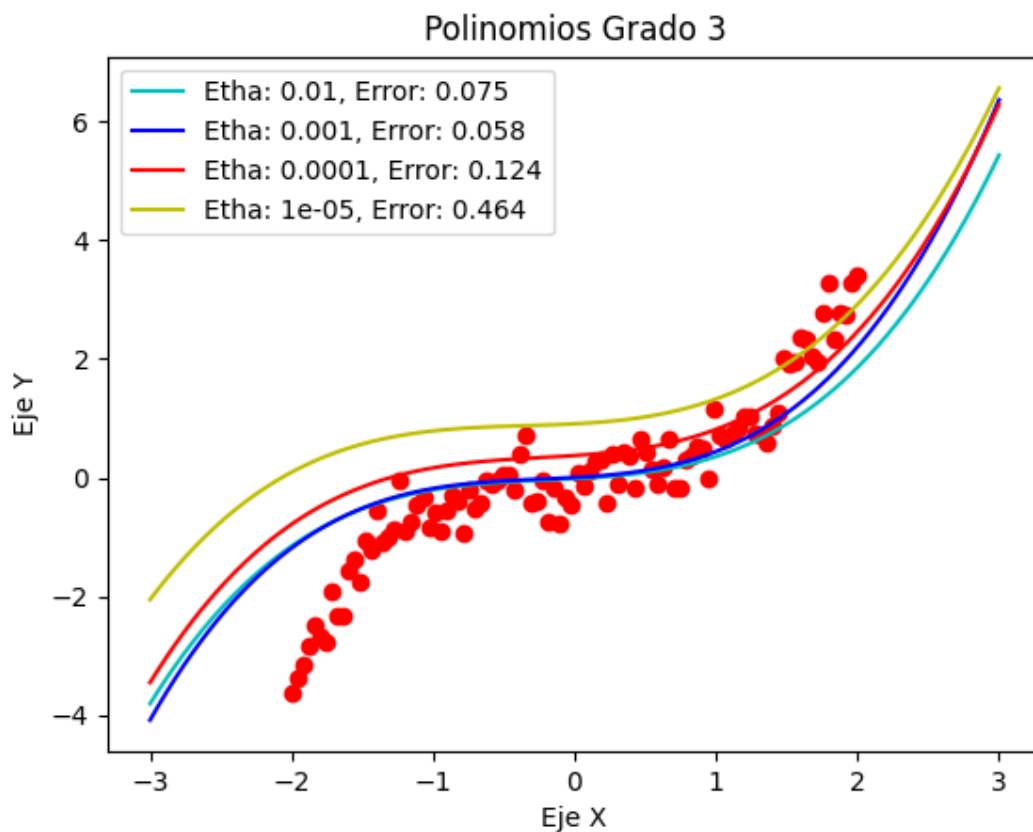
Se logró obtener la recta interpolante de los puntos. Tanto la forma analítica como la grafica de dicha recta se muestran a continuación:

$$y = (1.688)x + 0.0312$$



Dado que los puntos claramente corresponden a una función no lineal, la recta obtenida no es completamente interpolante a los puntos (no pasa por todos los puntos), sin embargo puede caracterizar los datos con un error cuadrático medio muy cerca de cero, de 0.075.

Por la forma de los datos se observa que es posible que una función polinómica de grado 3 se ajuste mejor a los datos. Por lo tanto se repitió el experimento pero buscando ajustar los coeficientes de un polinomio de tercer grado con tasas de aprendizaje 0.01, 0.001, 0.0001 y  $1 \times 10^{-5}$ .



Este resultado nos da curvas más ajustadas a los datos recibidos. Y se considera como mejor aproximación aquella con tasa de aprendizaje de 0.01 dado que el error cuadrático medio es tan solo de 0.058 unidades. Con ecuación del polinomio:

$$p(x) = (0.1787)x^3 + (0.1265)x^2 + (0.1323)x + (0.0016)$$