

Asignatura: Optimización

Profesor: D. Sc. Gerardo García Gil
2023-A

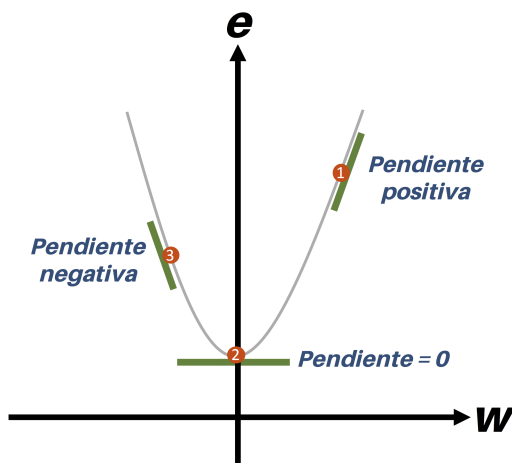
Ingeniería en Desarrollo de Software
Centro de Enseñanza Técnica Industrial (CETI)

Investigación.

El algoritmo gradient descent es el método de aprendizaje más utilizado para entrenar redes neuronales artificiales debido a su sencillez y facilidad de implantación.

Este algoritmo de optimización permite encontrar de forma automática el mínimo de una función. Para ello hace uso del gradiente (o derivada) de dicha función, que permite “guiar” al algoritmo para de manera progresiva acercarse al mínimo ideal de dicha función.

El gradiente es sinónimo de pendiente o inclinación, y matemáticamente recibe el nombre de derivada.



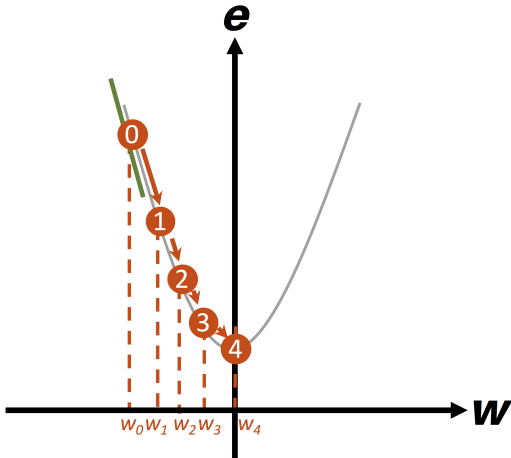
El algoritmo es iterativo y requiere que el usuario introduzca dos parámetros:

- El número de iteraciones, es decir cuántas veces se repetirá el procedimiento
- La tasa de aprendizaje (o learning rate), conocido como el parámetro α . Este definirá la cantidad de iteraciones requerida para que el algoritmo encuentre el mínimo de la función.

Una vez definidos estos dos parámetros, el algoritmo procede de la siguiente manera:

- Escoge de forma aleatoria el valor inicial de la variable
- En cada iteración actualiza el valor de w usando la siguiente ecuación: $w \leftarrow w - \alpha \cdot \text{gradiente}$

- Se repite el paso (2) hasta completar el número de iteraciones definido por el usuario.



Código.

```
clear all

funstr='10-(exp(-1*(x^2+3*y^2)))';
f=vectorize(inline(funstr));
range=[-1 1 -1 1];
% Draw the function
Ndiv=50;
dx=(range(2)-range(1))/Ndiv; dy=(range(4)-range(3))/Ndiv;
[x,y]=meshgrid(range(1):dx:range(2),range(3):dy:range(4));
z=(f(x,y));
figure(1); surf(x,y,z);
% Define the number of iterations
k=0;
niter=200;
% Gradient step size h definition
hstep = 0.001;
% Step size of the Gradient descent method
alfa=0.05;
```

```

%Initial point selection
xrange=range(2)-range(1);
yrange=range(4)-range(3);
x1=rand*xrange+range(1);
x2=rand*yrange+range(3);

% Optimization process
while (k<niter)

% Function evaluation
zn=f(x1,x2);

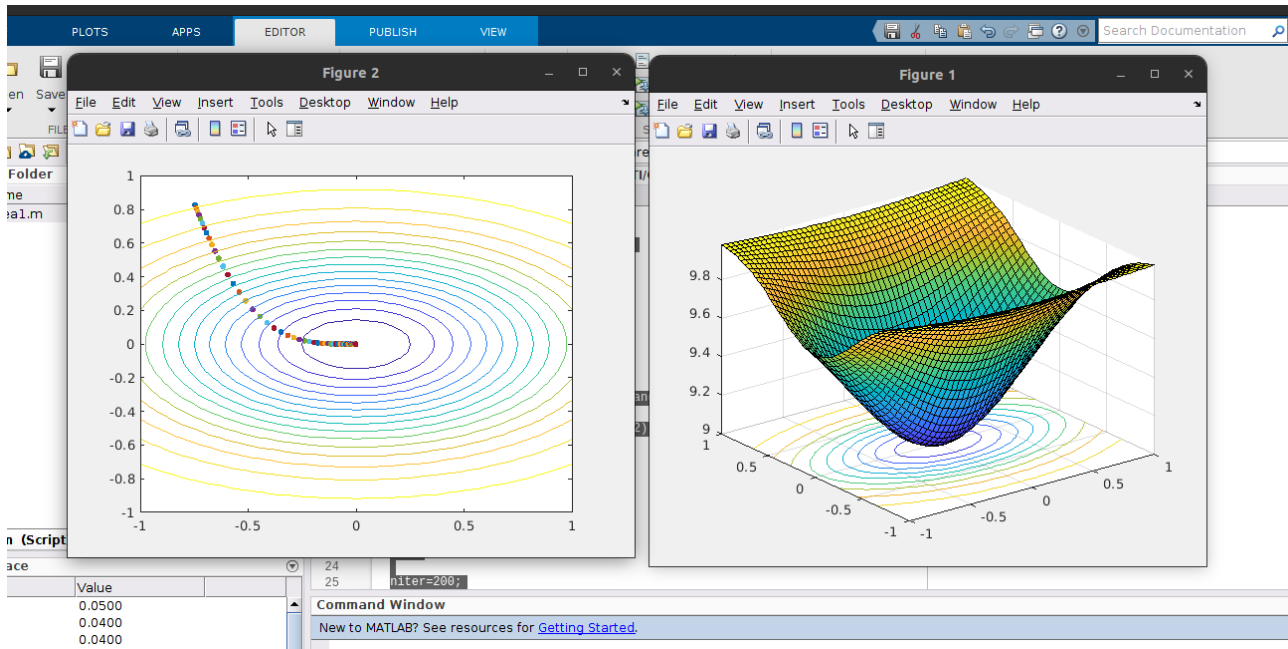
% Computation of gradients gx1 y gx2
vx1=x1+hstep;
vx2=x2+hstep;
gx1=(f(vx1,x2)-zn)/hstep;
gx2=(f(x1,vx2)-zn)/hstep;

% Draw the current position
figure(2)
contour(x,y,z,15); hold on;
plot(x1,x2,'.','markersize',10,'markerfacecolor','g');
hold on;

% Computation of the new solution
x1=x1-alfa*gx1;
x2=x2-alfa*gx2;
k=k+1;
end

```

Visualización de datos.



Conclusiones.

Este es uno de los algoritmos más básicos para la optimización, es fácil de entender aunque requiere conocimiento básico de cálculo integral el cual debemos repasar, matlab facilita mucho el manejo de las funciones y derivadas así como la visualización de los datos en tiempo real.

Referencias.

1. ¿Qué es el Gradiente Descendente? (s. f.). Codificando Bits. <https://www.codificandobits.com/blog/el-gradiente-descendente/>
2. Optimización por Descenso Gradiente. (2019, 17 junio). Optimización por Descenso Gradiente - File Exchange - MATLAB Central. <https://la.mathworks.com/matlabcentral/fileexchange/71867-optimizacion-por-descenso-gradiente>