

# Optimización - Tarea 5

Erick Salvador Alvarez Valencia, *CIMAT A.C.*

## I. INTRODUCCIÓN

En el presente reporte se analizarán los resultados obtenidos al aplicar los algoritmos de: *Backtracking*, *Interpolación cuadrática* e *Interpolación cúbica* para obtener el tamaño de paso  $\alpha$  en el método de descenso por gradiente. Los cuales fueron aplicados a las funciones de *Rosenbrock* con  $n = 2$ ,  $n = 100$  y *Wood*. Se mostrarán los resultados finales como: número de iteraciones, vector  $x_k$ , entre otras.

De la misma forma se presentará el resultado de aplicar el método de descenso por gradiente a un método de regresión logística para realizar la clasificación de una serie de imágenes en sus correspondientes dígitos. Para dicho algoritmo se utilizó el método de backtracking para calcular el tamaño de paso. Al final se mostrarán los resultados de la clasificación con el modelo entrenado.

mds

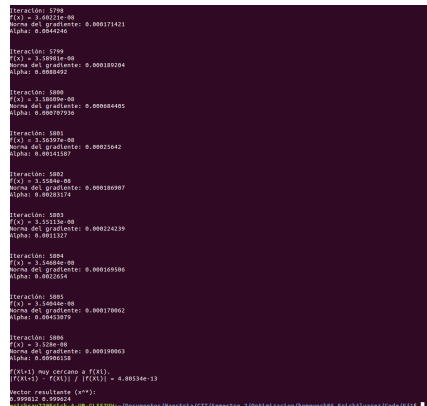
Marzo, 2018

## II. ALGORITMOS DE CÁLCULO DE TAMAÑO DE PASO

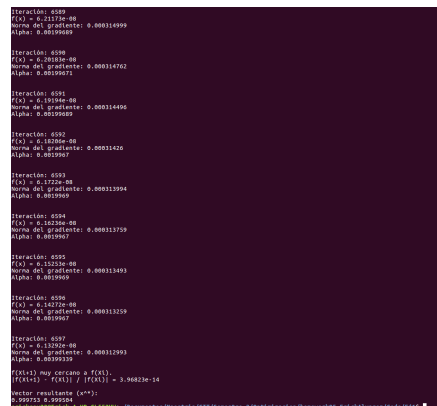
Para esta parte se hizo la implementación de los algoritmos de búsqueda de tamaño de paso: Backtracking, interpolación cuadrática e interpolación cúbica, y para la realización de pruebas se utilizaron las funciones de Rosenbrock para  $n = 2$ ,  $n = 100$  y Wood.

Los métodos anteriormente mencionados se caracterizan por el hecho de que no necesitan la matriz Hessiana para poder ejecutarse. Otra cosa importante es que estos métodos nacen por el hecho de que existe ya un tamaño de paso propuesto pero el mismo no cumple las condiciones de Armijo y por lo tanto tratan de remediar este hecho ya que aplican ciertas correcciones para encontrar un  $\alpha$  que si cumpla esto anterior.

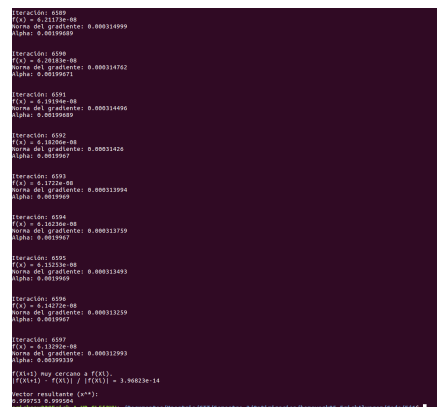
La forma en que se implementaron estos métodos sigue completamente esta filosofía, ya que se genera un  $\alpha$  inicial  $\alpha = 1$  y dependiendo el método que se va a aplicar se comprueba si este  $\alpha$  cumple o no las condiciones de Armijo, en caso de que no las cumpla se aplica el método y se genera un  $\alpha$  que si las cumpla, finalmente ese valor es el que se usará como tamaño de paso. Esta comprobación siempre se hace dentro del método el cual calcula el  $\alpha$ . Otra cosa a tener en cuenta es que tras cada nueva iteración del método de descenso por gradiente el  $\alpha$  inicial que se pasa al método es igual al  $\alpha$  anterior escalado por algún factor, esto porque puede que el  $\alpha$  anterior si cumpla las condiciones de Armijo pero sea demasiado pequeño y el avance hacia el óptimo local sea lento, de esta forma se le da un poco de ventaja a encontrar un tamaño de paso más grande. A continuación se mostrarán los resultados obtenidos de aplicar estos métodos a las funciones antes descritas.



(a) Figura 1. Descenso por gradiente usando backtracking para la función de Rosembrock con  $n = 2$ .



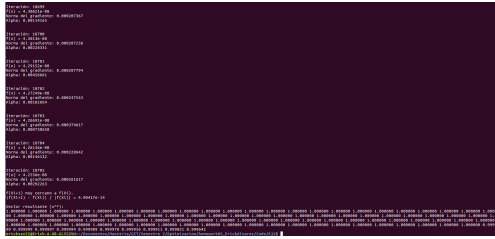
(b) Figura 2. Descenso por gradiente usando i. cuadrática para la función de Rosembrock con  $n = 2$ .



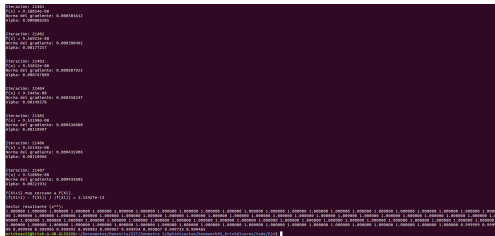
(c) Figura 3. Descenso por gradiente usando i. cúbica para la función de Rosembrock con  $n = 2$ .

Podemos observar en las Figuras 1, 2 y 3 el resultado de los tres algoritmos aplicados a la función de Rosembrock para

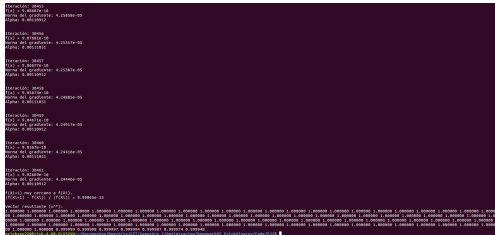
$n = 2$ . En las cuales se nota el hecho de que el algoritmo de Backtracking tuvo convergencia más rápida que los de interpolación.



(d) Figura 4. Descenso por gradiente usando backtracking para la función de Rosembrock con  $n = 100$ .

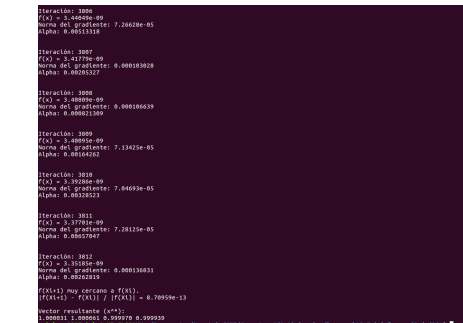


(e) Figura 5. Descenso por gradiente usando i. cuadrática para la función de Rosembrock con  $n = 100$ .

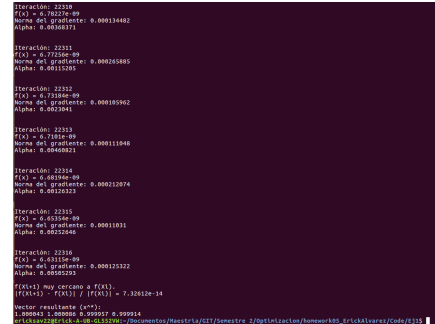


(f) Figura 6. Descenso por gradiente usando i. cúbica para la función de Rosembrock con  $n = 100$ .

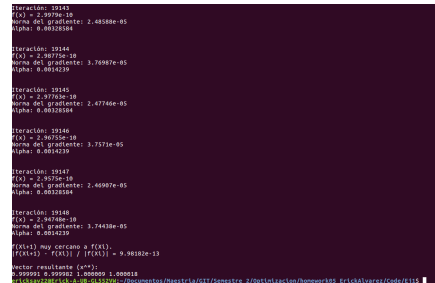
En las tres Figuras anteriores podemos ver que aplicando los métodos en la función de Rosembrock con  $n = 100$  se logra convergencia. Además se aprecia que el método más rápido es el de Backtracking, posteriormente le sigue la interpolación Cuadrática.



(g) Figura 7. Descenso por gradiente usando backtracking para la función de Wood.



(h) Figura 8. Descenso por gradiente usando  $i$ . cuadrática para la función de Wood.



(i) Figura 9. Descenso por gradiente usando i. cúbica para la función de Wood.

Finalmente tenemos la función de Wood, podemos notar que de igual manera el método converge, y también se aprecia que el método de Backtracking es el que produce la convergencia más rápida.

### III. MODELO DE REGRESIÓN LOGÍSTICA

En esta parte se uso el método de descenso por gradiente con el algoritmo de backtracking para optimizar un modelo de clasificación binaria el cual sirvió para clasificar imágenes de dígitos escritos a mano con sus valores correspondientes. El modelo usado es el siguiente:

$$\begin{aligned} h(\beta, \beta_0) &= \sum_{i=1}^n y_i \log \pi_i + (1 - y_i) \log(1 - \pi_i) \\ \pi_i &:= \pi_i(\beta, \beta_0) = \frac{1}{1 + \exp(-x_i^T \beta - \beta_0)} \end{aligned} \quad (1)$$

El gradiente de la función con respecto a  $\beta$ ,  $\beta_0$  es:

$$\nabla h = \sum_{i=1}^n (\pi_i \exp(x_i^T \beta - \beta_0 + y_i - 1)) x_i \quad (2)$$

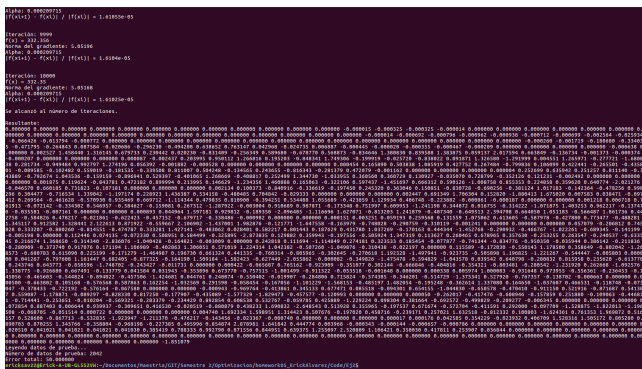
Nota: Para el caso de  $\frac{\partial h}{\partial \beta_0}$  la derivada es la descrita en (2) simplemente no se multiplica por  $x_i$ .

Para lo anterior podemos ver que la función  $\pi_i$  corresponde a la sigmoide la cual es muy usada en procesos de aprendizaje y clasificación. También hay que destacar que  $x_i \in R^{784}$  es el vector que corresponde a la imagen linealizada, además para cada  $x_i$  hay asociado un  $y_i$  que corresponde al valor de la imagen.

Para hacer la optimización primero se hizo lectura de los datos y para ello se eligieron dos números para los cuales se iba a realizar la clasificación. Posteriormente se segmentaron los datos asociados a esos dos números, y como el número de datos resultante era alto (10000 aprox.) se tomó un subconjunto de los mismos elegido arbitrariamente.

Ahora, como la función descrita en (1) representa una verosimilitud, se sabe que hay que maximizarla para que tenga buena clasificación. Y para esto se optó el convertirla en un problema de minimización invirtiendo esta función, osea  $\hat{h}(\beta, \beta_0) = -h(\beta, \beta_0)$ . Y por lo tanto el gradiente cambió tal que  $\nabla \hat{h} = -\nabla h$ , todo lo demás siguió igual.

Se hicieron varias pruebas con el modelo y en efecto, mientras más datos se añadían a la optimización mejores resultados brindaba. A continuación se mostrarán tres ejemplos realizados con pares de números distintos para la clasificación.



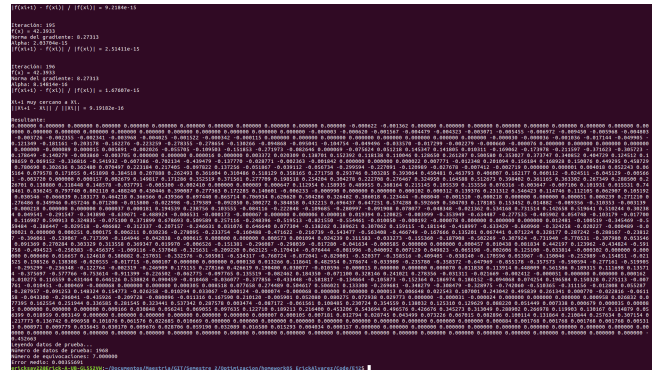
(j) Figura 10. Resultados de la clasificación para  $n_1 = 2$  y  $n_2 = 3$  y 5000 datos de entrenamiento.

Para el caso que describe la Figura 10 notamos que se realizaron 10mil iteraciones y se eligieron los números 2 y 3, obteniéndose 50 equivocaciones a la hora de clasificar 2042 datos.



(k) Figura 11. Resultados de la clasificación para  $n_1 = 1$  y  $n_2 = 8$  y 10000 datos de entrenamiento.

En la Figura 11. se muestran los resultados obtenidos para clasificar los números 1 y 8, en esta ocasión se necesitaron 244 iteraciones para optimizar el modelo y a la hora de clasificar el modelo se equivocó 19 veces de 2109 obteniendo un error medio de 0.009.



(l) Figura 12. Resultados de la clasificación para  $n_1 = 6$  y  $n_2 = 3$  y 7000 datos de entrenamiento.

En la Figura 12. se muestran los resultados obtenidos para clasificar los números 6 y 3, en esta ocasión se necesitaron 196 iteraciones para optimizar el modelo y a la hora de clasificar el modelo se equivocó 7 veces de 1968 obteniendo un error medio de 0.003.

#### IV. CONCLUSIÓN

En este reporte se mostraron los resultados de usar los métodos de Backtracking e interpolaciones cuadrática y cubica para estimar el tamaño de paso  $\alpha$  en el algoritmo de descenso por gradiente para encontrar óptimos locales de una función. De la misma forma se optimizó un modelo de regresión logística para clasificación binaria de imágenes de números escritos a mano. Para concluir se denota que los algoritmos trabajaron de una manera correcta, así como se logró una buena clasificación con el modelo anteriormente descrito usando el método de backtracking para la elección del  $\alpha$ .

Cabe destacar que se tuvieron que hacer varias pruebas con los métodos para tener buenos resultados.