



Seminario de Solución de Problemas de Inteligencia Artificial

Practica 1

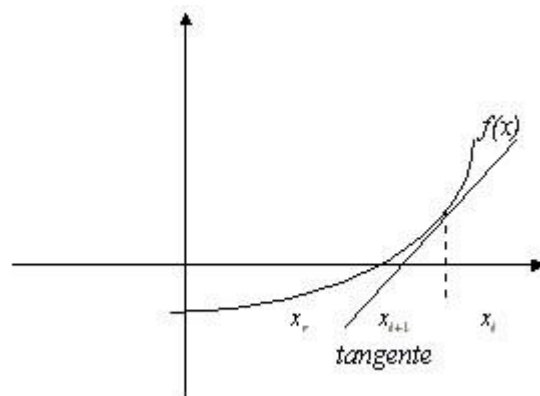
Introducción

Este método es uno de los mas utilizados para localizar raíces ya que en general es muy eficiente y siempre converge para una función polinomial.

Se requiere que las funciones sean diferenciables, y por tanto, continuas, para poder aplicar este método.

Se debe partir de un valor inicial para la raíz: x^i , este puede ser cualquier valor, el método convergirá a la raíz mas cercana.

Si se extiende una tangente desde el punto $(x_i, f(x_i))$, el punto donde esta tangente cruza al eje x representa una aproximación mejorada de la raíz.



La fórmula de Newton-Raphson se deduce a partir de la fórmula de la pendiente de una recta.

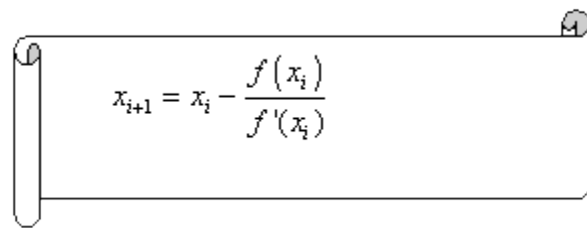
Pendiente de una recta:

$$m = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} = \frac{0 - f(x_i)}{x_{i+1} - x_i}$$

$$m(x_{i+1} - x_i) = -f(x_i)$$

$$x_{i+1} - x_i = \frac{f(x_i)}{m}$$

$$x_{i+1} = x_i - \frac{f(x_i)}{m}$$


$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Se define la derivada de una función en un punto dado como la pendiente a la recta tangente de dicho punto, por lo tanto:
 $m = f'(x)$

Desarrollo

Se busco aplicar la formula de newthon raphson en ruby de la siguiente manera

```
def newthonRapson(exercise, slimit_iterations = 50)
  (0..limit_iterations).map do |i|
    xi = xi - (fi(xi, exercise) / fii(xi, exercise))
  end.last
end
exclude(arr)
end
```

De esta manera se podria calcular una de las raices, pero como esto es poco practico puesto que se tendria que realizar a prueba y error para encontrar las demas, se opto por hacer una leve variación

```
def newthonRapson(excercise, start, ended, steps = 1, limit_iterations = 50)
  arr = (start..ended).step(steps).to_a().map do |xi|
    (0..limit_iterations).map do |i|
      xi = xi - (fi(xi, excercise) / fii(xi, excercise))
    end.last
  end
  exclude(arr)
end
```

De esta manera se tomaban todos puntos dentro de un intervalo posible de encontrar las raices, sin embargo esto creo un problema, al momento de evaluar los ceros saldrian n cantidad de ceros tales como el rango que se especifico, para solucionar esto se creo una funcion de filtrado para solo obtener las raices necesarias y no tener elementos duplicados

```
def exclude(arr)
  sorted = arr.reject{|element| element.nan?}.to_a.sort
  filtered = Array.new()
  to_filter = Array.new()
  for i in 0..sorted.count-2
```

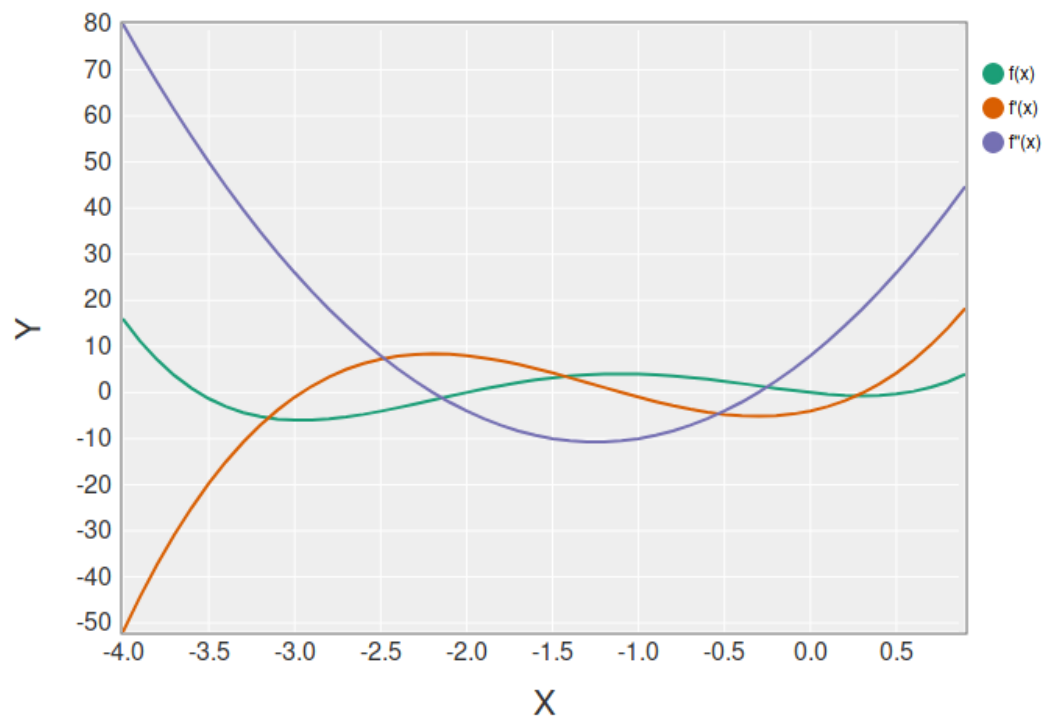
```
if !to_filter.include?(sorted[i].round)
  filtered << sorted[i]
  to_filter << sorted[i].round
end
end
return filtered
end
```

Este metodo fue el mas dificil ya que habia ocaciones donde se tenian que filtrar elementos del tipo NaN propio de ruby y luego revisar si ya existia o no dentro de los ya seleccionados, para esto se probaron varios metodos, pero el mas eficaz y versatil para ruby resulto ser el que se indico.

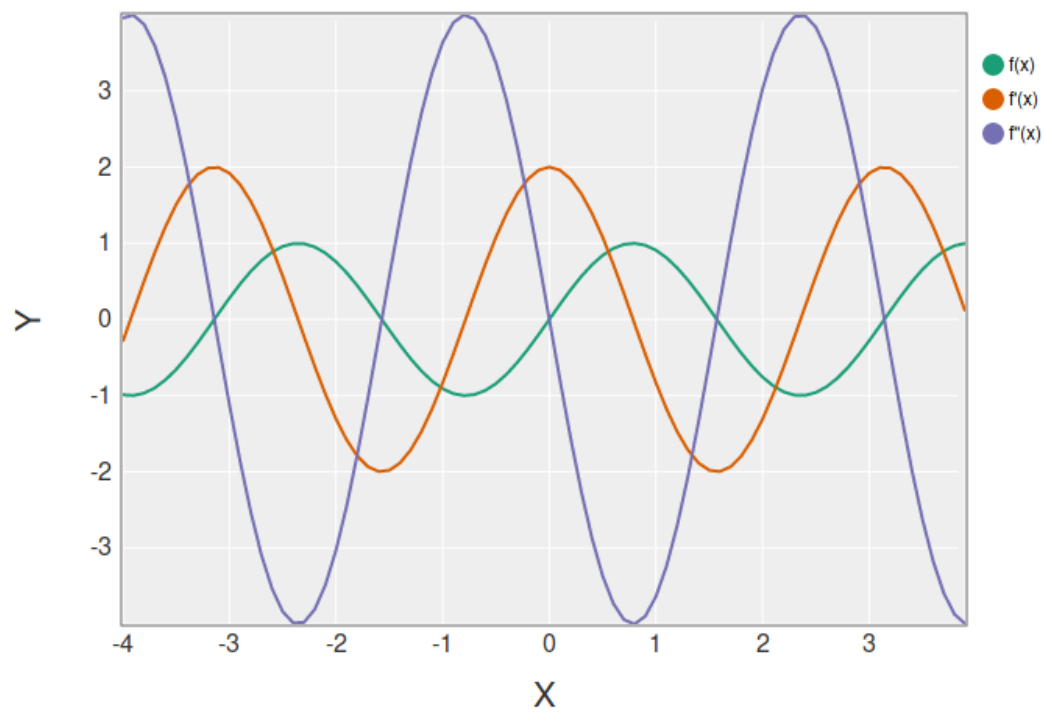
Para crear las graficas, se utilizo una gema (librerias de ruby) llamada nyaplot, para esta tiene el grave problema que su metodo show, el cual debe de mostrar la grafica, esta corrupto y retorna un error al stacktrace, para evitar esto, se opto por usar otra de las funciones disponibles, la cual era export_html, de esta manera se comento, al momento de la entrega para no generar archivos innecesarios dentro de la carpeta del evaluador, de igual manera solo basta con descomentar la linea de codigo 115 o donde se encuentra la funcion export_html, y para avanzar entre funcion y funcion, se es necesario dar enter ya que espera el bufer de entrada del teclado

Graficas generadas

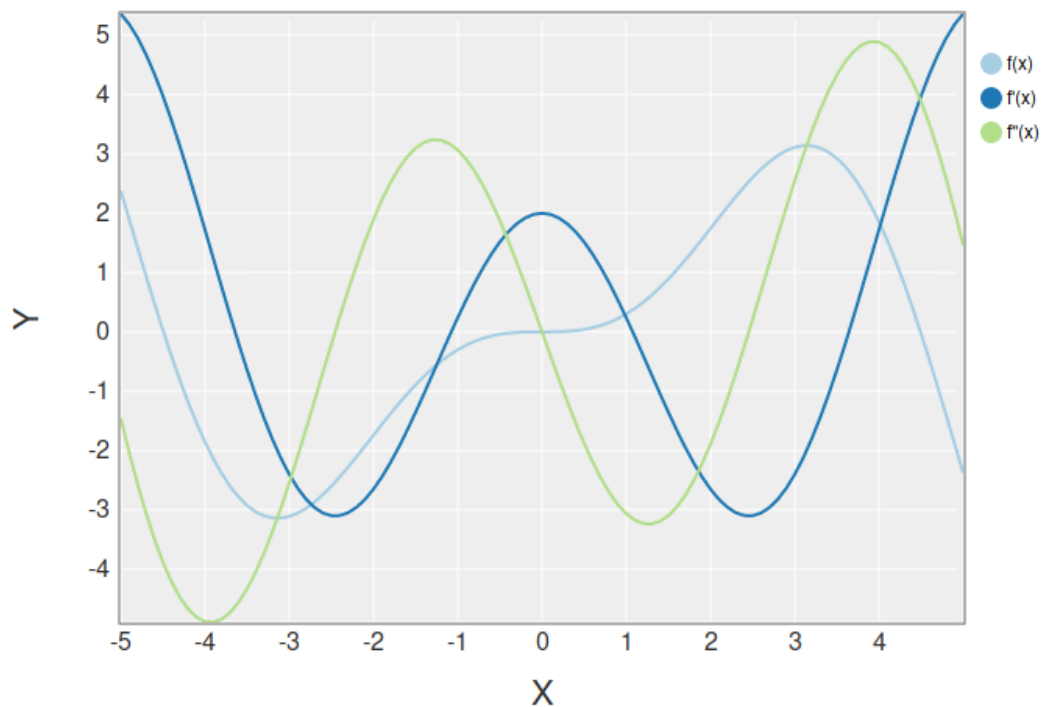
F1 otorgada



F2 otorgada



F3 otorgada



Conclusion

El metodo newthon raphson es realmente sencillo de programar, y su velocidad de ejecucion ademas de su adaptivilidad en cualquier funcion lo hacen realmente eficas, quitando el hecho de que se debe de tener conocimiento de derivaciones, y que se deben de ingresar estas por cada funcion diferente.

Extras

Se es necesario para ejecutar el programa instalar ruby y uno de sus administradores de gemas, en mi caso se utiliza rbenv y con el se itiliza el comando `gem install nyaplot`