CENTRO UNIVERSITÁRIO FEI

VICTOR BIAZON
RA: 119.115-4

RELATÓRIO IV – PROGRAMAÇÃO CIENTIFICA
DESCIDA DE GRADIENTE

SÃO BERNARDO DO CAMPO
2019

VICTOR BIAZON

RA: 119.115-4

RELATÓRIO IV – PROGRAMAÇÃO CIENTIFICA

DESCIDA DE GRADIENTE

Relatório de desenvolvimento do algoritmos descida de gradiente, desenvolvido pelo aluno Victor Biazon, RA 119.115-4, para disciplina PEL216 – Programação Cientifica, ministrada pelo professor Reinaldo Bianchi.

São Bernardo do Campo

2019

Sumário:

Motivação

Implementação de algoritmo para encontrar o mínimo local de funções matemáticas.

Objetivo:

Desenvolver o algoritmo que através de iterações realiza o cálculo do ponto mínimo local de uma função matemática através do uso da descida de gradiente baseada no uso de derivadas da função para encontrar tal objetivo.

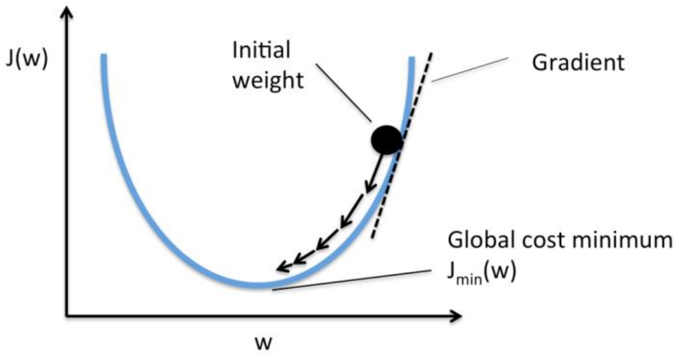
Teoria

Para se encontrar o mínimo de uma função utilizando a descida de gradiente é necessário calcular a derivada de tal função para a condição inicial e ajusta seu valor iterativamente até que se encontre o ponto onde a derivada é igual a zero (ou muito próximo disso, já que pelas iterações geralmente atingir o valor zero é impossível).

No entanto é extremamente importante observar qual será a condição inicial utilizada já que para alguns casos a escolha errônea de tal condição pode levar a uma iteração não convergente, ou seja, que estoura para o infinito negativo.

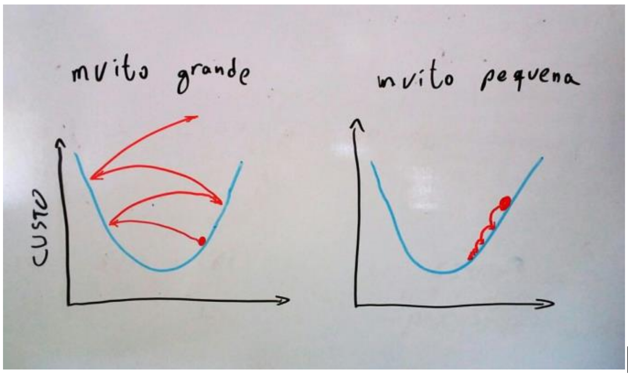
Também é importante observar a taxa de aprendizado, que deve ser adequada ao problema. Se por exemplo se utilizar uma taxa de aprendizado muito pequena, o problema levará muito tempo para ser solucionado. Por outro lado com uma taxa grande demais o problema pode não chegar a convergir, pois passa de um lado para o outro do gradiente e não consegue atingir o ponto de mínimo, ou no pior caso, se distancia cada vez mais do objetivo.

Abaixo está ilustrado como as iterações chegam ao ponto de mínimo utilizando a descida de gradiente, sendo a reta tangente a função a representação gráfica da derivada da função naquele ponto.



[1]

Abaixo uma ilustração das taxas de aprendizado:



[2]

Implementação

Pseudo-código – Descida de Gradiente

```
DescidaGradiente()
{
    x = x0; //x recebe o valor inicial x0
    f = Funcao(x); //calculo de f0
    f_linha = Derivada(x); //derivada da função em X

    while (abs(f_linha) > 0.0000001) { //avalia enquanto o modulo da derivada de f for maior que um número muito próximo de 0, pois não é possível chegar ao zero.
        s = Derivada(x); //calcula derivada da função em x
        x = x - taxa_ap * s; //realiza o desconto da derivada multiplicada pela taxa de variação
        f_linha = Derivada(x); //calcula a derivada novamente para ser avaliada pelo while
    }
    Imprime (x, f) // mostra o valor de x, e da funcao f, onde o mínimo local ocorre.
```

O pseudocódigo acima é uma generalização, onde as funções “Funcao” e “Derivada” são cálculos dependentes da função a ser avaliada.

Experimentos e resultados

Para testar o algoritmo foram propostas duas funções e duas taxas de aprendizado. Sendo elas:

a) x^2 com condição inicial x0 = 2;

b) $x^3 - 2x^2 + 2$ com condição inicial x0 = 2;

E taxas de aprendizado 0,1 e 1.

Com isso tivemos os seguintes resultados:

com condição inicial x0 = 2 e taxa de aprendizado 0.1.

Iniciando as iterações...

```
X: 1.6 s: 4
X: 1.28 s: 3.2
X: 1.024 s: 2.56
X: 0.8192 s: 2.048
X: 0.65536 s: 1.6384
X: 0.524288 s: 1.31072
X: 0.41943 s: 1.04858
X: 0.335544 s: 0.838861
X: 0.268435 s: 0.671089
X: 0.214748 s: 0.536871
X: 0.171799 s: 0.429497
X: 0.137439 s: 0.343597
X: 0.109951 s: 0.274878
X: 0.0879609 s: 0.219902
X: 0.0703687 s: 0.175922
X: 0.056295 s: 0.140737
X: 0.045036 s: 0.11259
```

Vindo a convergir em:

```
X: 6.76921e-14 s: 1.6923e-13
X: 5.41537e-14 s: 1.35384e-13
X: 4.3323e-14 s: 1.08307e-13
X: 3.46584e-14 s: 8.66459e-14
X: 2.77267e-14 s: 6.93167e-14
X: 2.21814e-14 s: 5.54534e-14
X: 1.77451e-14 s: 4.43627e-14
X: 1.41961e-14 s: 3.54902e-14
X: 1.13569e-14 s: 2.83921e-14
X: 9.08548e-15 s: 2.27137e-14
X: 7.26839e-15 s: 1.8171e-14
X: 5.81471e-15 s: 1.45368e-14
X: 4.65177e-15 s: 1.16294e-14
O minimo local da funcao x^2 se encontra em x: 4.65177e-15 e neste ponto a funcao vale: 2.16389e-29
Arredondando x: 0 e o valor da funcao: 0
```

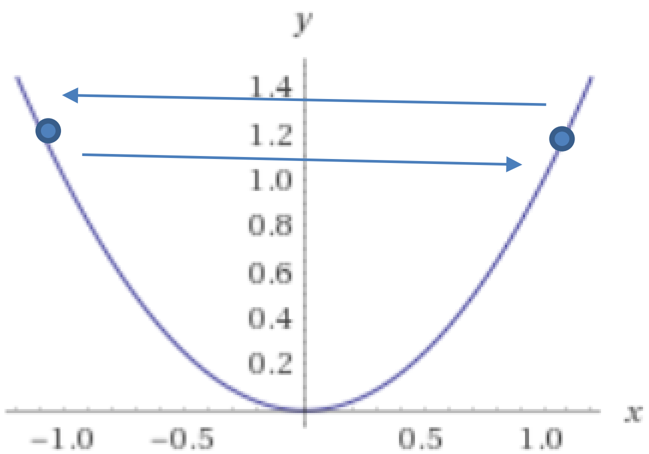
Desta forma pudemos extrair da execução que o ponto mínimo ocorre na função x^2 quando: x = 0, resultando em y = 0.

Já para a mesma função mas com taxa de aprendizado = 1:

As iterações retornam:

```
X: 2 s: -4
X: -2 s: 4
X: 2 s: -4
X: -2 s: 4
X: 2 s: -4
X: -2 s: 4
X: 2 s: -4
X: -2 s: 4
X: 2 s: -4
X: -2 s: 4
X: 2 s: -4
X: -2 s: 4
X: 2 s: -4
X: -2 s: 4
```

Nesta pode-se notar que o x varia entre 2 e -2 apenas, sendo assim o valor da taxa de aprendizado causou uma impossibilidade de convergência pois o valor de x nunca deixará de mudar de 2 para -2 e vice-versa. Com isto a taxa de aprendizado de 1, impossibilita a solução. Tal variação de posições é ilustrada na imagem abaixo.



(x from -1.2 to 1.2)

[3]

Como podemos ver a acima a imagem confirma o ponto de mínimo da função ocorrendo no ponto x = 0.

Avaliando agora a função $x^3 - 2x^2 + 2$ com condição inicial x0 = 2 e taxa de aprendizado 0.1.

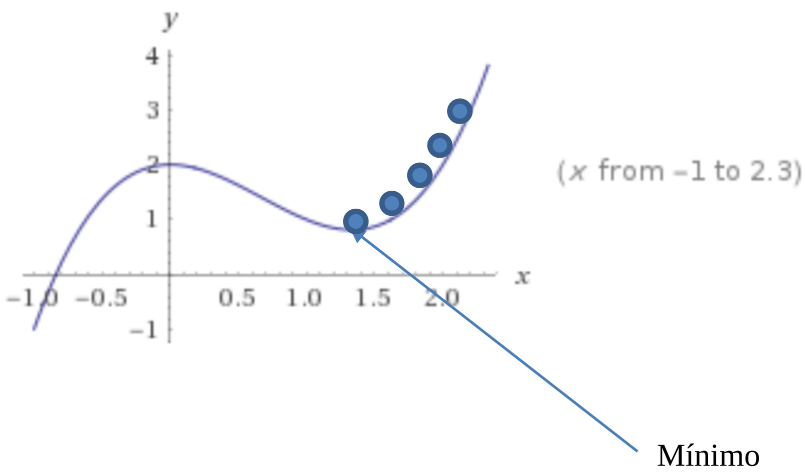
As iterações rapidamente convergem para:

```
X: 1.6 s: 4
X: 1.472 s: 1.28
X: 1.41076 s: 0.612352
X: 1.37799 s: 0.327713
X: 1.35953 s: 0.184624
X: 1.34885 s: 0.10685
X: 1.34257 s: 0.062773
X: 1.33885 s: 0.0371977
X: 1.33663 s: 0.022154
X: 1.33531 s: 0.0132338
X: 1.33452 s: 0.00791934
X: 1.33404 s: 0.00474409
X: 1.33376 s: 0.00284376
X: 1.33359 s: 0.00170528
X: 1.33349 s: 0.00102282
X: 1.33343 s: 0.000613568
X: 1.33339 s: 0.000368095
X: 1.33337 s: 0.000220841
X: 1.33335 s: 0.000132499
X: 1.33335 s: 7.94971e-05
X: 1.33334 s: 4.76975e-05
X: 1.33334 s: 2.86182e-05
X: 1.33334 s: 1.71708e-05
X: 1.33333 s: 1.03025e-05
X: 1.33333 s: 6.18147e-06
X: 1.33333 s: 3.70888e-06
X: 1.33333 s: 2.22533e-06
X: 1.33333 s: 1.33519e-06
X: 1.33333 s: 8.01117e-07
X: 1.33333 s: 4.8067e-07
X: 1.33333 s: 2.88402e-07
X: 1.33333 s: 1.73041e-07
X: 1.33333 s: 1.03825e-07
O minimo local da funcao x^3-2*x^2+2 se encontra em x: 1.33333 e neste ponto a funcao vale: 0.814815
```

Desta forma pudemos extrair da execução que o ponto mínimo local ocorre na função $x^3 - 2x^2 + 2$ quando:

x = 1.333, resultando em y = 0,8148.

Como podemos ver abaixo a função plotada confirma o ponto de mínimo local.



[4]

Agora analisando a mesma função com taxa de aprendizado = 1:

As iterações resultam em:

```
X: -2 s: 4
X: -22 s: 20
X: -1562 s: 1540
X: -7.32734e+06 s: 7.32578e+06
X: -1.6107e+14 s: 1.6107e+14
X: -7.78305e+28 s: 7.78305e+28
X: -1.81728e+58 s: 1.81728e+58
X: -9.90748e+116 s: 9.90748e+116
X: -2.94474e+234 s: 2.94474e+234
X: -inf s: inf
X: -inf s: inf
X: -inf s: inf
X: -inf s: inf
X: -inf s: inf
X: -inf s: inf
X: -inf s: inf
X: -inf s: inf
X: -inf s: inf
X: -inf s: inf
X: -inf s: inf
X: -inf s: inf
```

Como pode-se notar rapidamente o x tende a infinito negativo com inclinação infinita. Isto se deve a esta função ter como característica tender ao infinito negativo para um limite: $\lim_{n \to -\infty} x^3 - 2x^2 + 2$. Por isso ao utilizar esta taxa de aprendizado grande demais o algoritmo "pulou" para outra descida da função que no caso não encontra ponto onde a derivada é zero pois seu valor diminui cada vez mais.

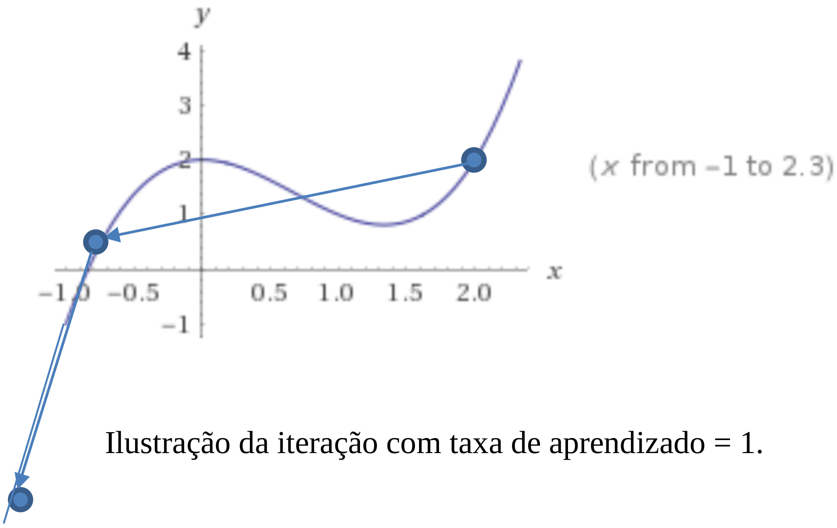


Ilustração da iteração com taxa de aprendizado = 1.

Trabalhos correlatos

Stochastic Gradient Descent as Approximate Bayesian Inference

MANDT, Stephan; HOFFMAN, Matthew D.; BLEI, David M.; 2017

http://www.jmlr.org/papers/volume18/17-214/17-214.pdf

Globally Optimal Gradient Descent for a ConvNet with Gaussian Inputs

BRUTZKUS, Alon; GLOBERSON, Amir; 2017

https://pdfs.semanticscholar.org/fc75/6b45678ef7ffc1a796de62365013011b659e.pdf

Conclusão

Com sete experimento pudemos concluir que a descida de gradiente, para encontrar numericamente o ponto mínimo de uma função é extremamente útil quando não se é possível determinar formalmente, ou analiticamente, a função matemática em questão. Desta forma é possível soluções empíricas não necessitando modelar matematicamente o problema.
Para as duas funções propostas foi evidenciado a grande dependência para o sucesso do êxito do algoritmo a determinação do ponto de início das iterações, ou seja as condições iniciais. E também é importantíssimo ressaltar a relevância de se determinar uma taxa de aprendizado coerente para que o algoritmo chegue à solução rapidamente e possa convergir em um resultado. Sendo que para taxas muito baixas se leva muito tempo, e taxas muito grandes o algoritmo não converge ou se perde.

Referências bibliográficas

[1] http://deeplearningbook.com.br/aprendizado-com-a-descida-do-gradiente/

[2] https://matheusfacure.github.io/2017/02/20/MQO-Gradiente-Descendente/

[3] https://www.wolframalpha.com/input/?i=x%5E2

[4] https://www.wolframalpha.com/input/?i=x%5E3-2x%5E2%2B2