Capítulo 7

Otimização Contínua

André Rodrigues Coimbra, Rayane Araújo Lima e Renan R. de Oliveira

Abstract

This report presents a theoretical summary of Chapter 7 of the book "Mathematics for Machine Learning" [Deisenroth et al. 2020] which deals with the topic of "Continuous Optimization", focusing on the essential mathematical foundations for Data Science. The chapter covers two main branches of continuous optimization: unconstrained optimization and constrained optimization. We will assume in this chapter that our objective function is differentiable. In this case, we have access to the gradient value to help us find the local or global optimal value in the case of unconstrained optimization. For the case of constrained optimization, other concepts will be introduced to manage constraints. This chapter also covers a special class of problems (convex optimization problems) where we can make statements about how to achieve the global optimum. This report is part of the work of the discipline "Special Topics in Computing Fundamentals - Mathematics and Statistics for Data Sciences" of the Postgraduate Program at the Federal University of Goiás (UFG), in Computer Science, of the Institute of Informatics , in the 2024/1 academic semester, aiming to equip students with mathematical skills for the field of Data Science.

Resumo

Este relatório apresenta um resumo teórico do Capítulo 7 do livro "Mathematics for Machine Learning" [Deisenroth et al. 2020] que trata do tópico sobre "Otimização Contínua", focando nas bases matemáticas essenciais para a Ciência de Dados. O capítulo aborda dois ramos principais da otimização contínua: otimização irrestrita e otimização restrita. Assumiremos neste capítulo que a nossa função objetivo é diferenciável. Neste caso, temos acesso ao valor do gradiente para nos ajudar a encontrar o valor ótimo local ou global no caso da otimização irrestrita. Para o caso da otimização restrita, serão introduzidos outros conceitos para gerenciar as restrições. Este capítulo também aborda uma classe especial de problemas (problemas de otimização convexa) onde podemos fazer afirmações sobre como atingir o ótimo global. Este relatório é parte do trabalho da

disciplina "Tópicos Especiais em Fundamentos de Computação - Matemática e Estatística para Ciências de Dados" do programa de Pós-Graduação da Universidade Federal de Goiás (UFG), em Ciência da Computação, do Instituto de Informática, no semestre letivo 2024/1, visando equipar os estudantes com habilidades matemáticas para o campo da Ciência de Dados.

7.1. Introdução

Os algoritmos de aprendizado de máquina implementados em computadores transformam-se em métodos de otimização numérica, conforme as teorias matemáticas são aplicadas. Este capítulo aborda os métodos numéricos fundamentais utilizados para o treinamento de modelos de aprendizado de máquina. O processo de treinamento de um modelo de aprendizado de máquina frequentemente envolve a identificação de um conjunto ótimo de parâmetros. Para isso, é necessário otimizar uma função objetivo específica utilizando algoritmos de otimização, que buscam o valor mais adequado para alcançar os melhores resultados.

A Figura 7.1 apresenta o mapa mental dos conceitos de otimização explorados neste capítulo. Neste caso, serão abordados os métodos de otimização irrestrita e otimização restrita, que são os dois principais segmentos da otimização contínua. Na Figura 7.1, destaca-se duas ideias principais: gradiente descendente e otimização convexa.

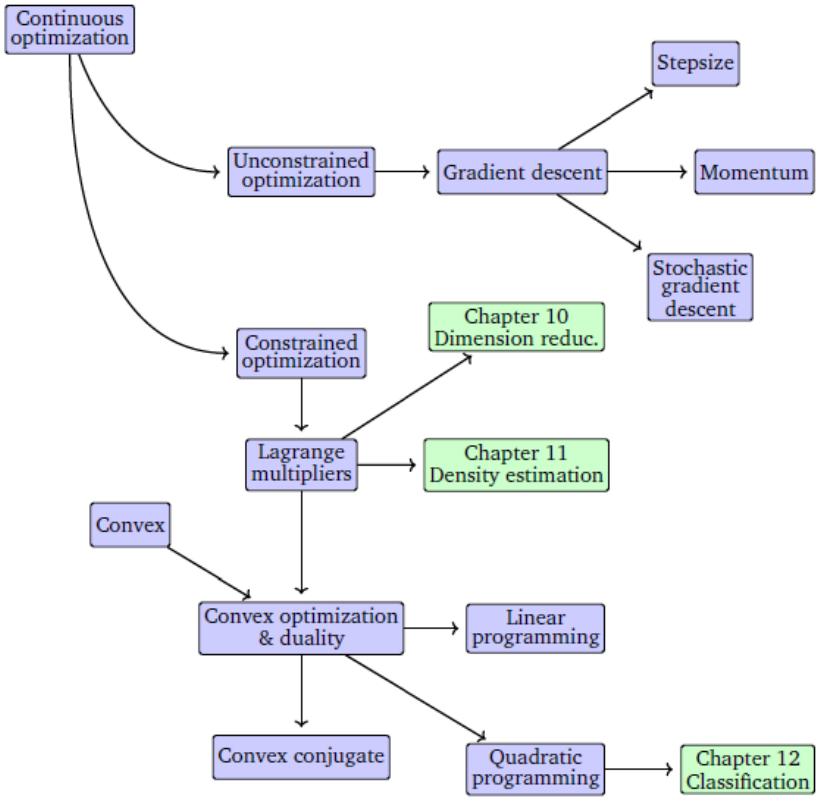


Figura 7.1: Mapa mental dos conceitos relacionados à otimização explorados neste capítulo. Existem duas ideias principais: gradiente descendente e otimização convexa.

Ao longo deste capítulo, partiremos do pressuposto de que a função objetivo é di-

ferenciável, o que nos permite acessar o gradiente em cada ponto do espaço para auxiliar na busca pelo valor ótimo. Na maioria dos casos em aprendizado de máquina, as funções objetivo são configuradas para serem minimizadas, isto é, busca-se alcançar o valor mínimo como o melhor resultado da função de treinamento.

Intuitivamente, encontrar esse melhor valor pode ser comparado à busca pelos vales em um terreno montanhoso, onde os gradientes nos indicam a direção ascendente. A estratégia consiste em "descer a colina" (mover-se na direção oposta ao gradiente) na esperança de encontrar o ponto mais baixo. No contexto da otimização irrestrita, essa é a principal abordagem requerida. Já para a otimização restrita, é essencial incorporar conceitos adicionais para lidar com as restrições. Adicionalmente, introduziremos uma categoria especial de problemas, denominados problemas de otimização convexa, onde é possível fazer afirmações definitivas sobre como alcançar o ótimo global.

7.2. Otimização usando Gradiente Descendente

Considere o problema de encontrar o valor que minimiza uma função

$$\min_x f(x), \tag{1}$$

onde $f : \mathbb{R}^d \to \mathbb{R}$ é uma função objetivo que reflete um problema de aprendizado de máquina. Assumimos que a função f é diferenciável e não é possível encontrar analiticamente uma solução direta utilizando operações elementares.

O gradiente descendente é um algoritmo de otimização amplamente utilizado, classificado como de primeira ordem devido ao seu uso das primeiras derivadas para encontrar mínimos locais de funções. Este método é essencial em muitos campos, incluindo aprendizado de máquina, onde é utilizado para minimizar funções de custo ou perda associadas a modelos. O processo fundamental do gradiente descendente envolve a atualização iterativa dos parâmetros da função em direção ao mínimo local.

Matematicamente, o gradiente de uma função multivariável é um vetor que contém todas as derivadas parciais da função. Ele aponta na direção da maior taxa de aumento da função e tem magnitude que indica quão rápido a função aumenta nessa direção. No contexto de otimização, estamos interessados em encontrar onde a função atinge seu valor mínimo. O gradiente descendente faz isso ao tomar passos na direção oposta ao gradiente da função no ponto atual. Isto é baseado na lógica de que se o gradiente aponta para a direção de maior aumento, o valor do gradiente negativo aponta para a direção de maior decréscimo. Portanto, ao seguir na direção do negativo do gradiente, espera-se alcançar o ponto de mínimo local.

Considere a superfície descrita pela função f e uma bola posicionada em um ponto inicial x_0. Se a bola for solta, ela naturalmente rolará para baixo, seguindo a trajetória da descida mais íngreme. O método do gradiente descendente se baseia na observação de que o valor de $f(x_0)$ diminui mais rapidamente se movermos o ponto x_0 na direção oposta ao gradiente $-(\nabla f(x_0))^\top$ da função f nesse ponto. Portanto, se

$$x_1 = x_0 - \gamma((\nabla f)(x_0))^\top \tag{2}$$

para um tamanho de passo pequeno $\gamma \geq 0$, então $f(x_1) \leq f(x_0)$. Considerando que f pode ser uma função de várias variáveis, aplicamos a transposição ao gradiente para assegurar a consistência dimensional.

Dessa forma, podemos definir um algoritmo simples de gradiente descendente para encontrar um ótimo local $f(x_*)$ de uma função começando com uma estimativa inicial x_0 como o valor do parâmetro que desejamos otimizar e então iterar de acordo com

$$x_{i+1} = x_i - \gamma((\nabla f(x_i))^\top. \tag{3}$$

Para um tamanho de passo adequado γ_i, a sequência $f(x_0) \geq f(x_1) \geq ... \geq f(x_n)$ converge para um mínimo local. A descida do gradiente pode ser relativamente lenta perto do mínimo. Usando a analogia da bola rolando colina abaixo, a descida é rápida quando a superfície é íngreme. Porém, quando a superfície é pouco inclinada, a bola desce lentamente e pode fazer movimentos de zigue-zague devido à orientação quase lateral dos gradientes.

7.2.1. Gradiente Descendente com Função Univariada

Considere a seguinte função polinomial de uma única variável que mapeia valores de x para $f(x)$ no conjunto dos números reais.

$$f(x) = x^4 + 7x^3 + 5x^2 - 17x + 3 \tag{4}$$

De acordo com a Figura 7.2, a função possui um mínimo global próximo de $x = -4.5$ com um valor aproximado de $f(x) = -47$. Devido a diferenciabilidade contínua da função, o cálculo dos gradientes são úteis para orientar a busca pelo valor mínimo, indicando se devemos avançar para a direita ou para a esquerda. Na Figura 7.2, o sentido oposto dos gradientes são indicados por setas e o mínimo global é indicado pela linha azul tracejada. No caso da busca pelo valor mínimo da função, deve-se pressupor que estamos na "bacia" de atração correta, visto que há também um mínimo local próximo de $x = 0.7$.

Para funções de uma única variável, o gradiente da função é simplesmente a derivada da função com relação à sua única variável. O gradiente é um conceito que se generaliza para funções de múltiplas variáveis, onde é representado por um vetor de derivadas parciais. A derivada de uma função polinomial pode ser encontrada aplicando a regra da potência a cada termo da função. Neste caso, a derivada $f'(x)$ da função $f(x)$ é

$$f'(x) = 4x^3 + 21x^2 + 10x - 17. \tag{5}$$

A derivada $f'(x)$ pode ser utilizada para analisar o comportamento da função e para aplicar métodos de otimização. O valor de $f'(x)$ indica a direção de maior crescimento da função. Em funções de uma única variável, $f'(x) > 0$ indica que a função está aumentando à medida que x aumenta. Por outro lado, se $f'(x) < 0$, a função está diminuindo. Os valores de $f'(x) = 0$ indicam os pontos críticos da função, que podem ser mínimos, máximos ou pontos de inflexão. Neste caso, em busca de um ponto onde a

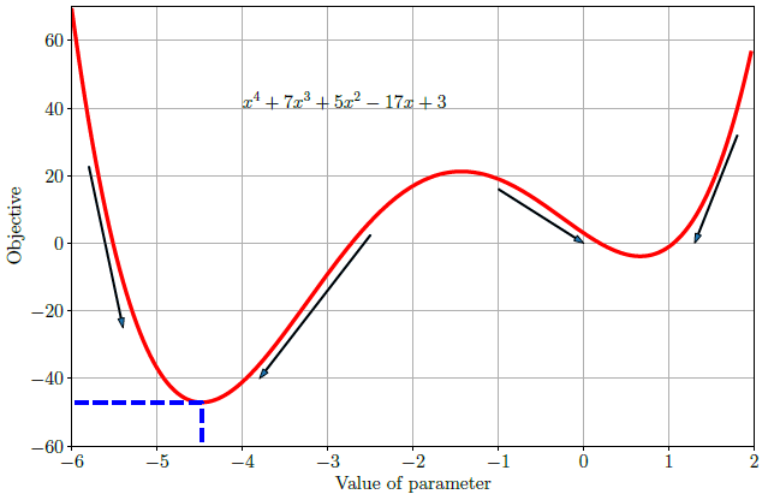


Figura 7.2: Exemplo de função objetivo. Os gradientes negativos são indicados por setas e o mínimo global é indicado pela linha azul tracejada.

função potencialmente atinge um valor mínimo, o método do gradiente descendente deve mover-se na direção de $-f'(x)$.

Os pontos estacionários são as raízes reais da derivada, ou seja, pontos que possuem gradiente zero. Podemos resolver todos os pontos estacionários de uma função calculando a sua derivada e definindo-a como zero. Como $f'(x)$ é uma equação cúbica, em geral esta função possui três soluções quando definida como zero. Na Figura 7.2, existem dois pontos são mínimos e um é máximo (em torno de $x = -1.4$).

Para verificar se um ponto estacionário é mínimo ou máximo, precisamos calcular a derivada uma segunda vez e verificar se a segunda derivada é positiva ou negativa no ponto estacionário $x = c$. No exemplo da Figura 7.2, a segunda derivada de $f'(x)$ é

$$f''(x) = 12x^2 + 42x + 10. \tag{6}$$

Neste caso, $f''(c) > 0$ indica que a função tem concavidade voltada para cima no ponto c, indicando que c é um ponto de mínimo local da função f. Por outro lado, $f''(c) < 0$ indica que a função tem uma concavidade voltada para baixo, indicando que c é um ponto de máximo local da função f. Ao substituir os valores de c em $f''(c)$ estimados visualmente como $c = -4.5, -1.4$ e 0.7, observamos que, como esperado, o ponto médio é um máximo, pois $f''(-1.4) < 0$, e os outros dois pontos estacionários são mínimos.

Para polinômios de alta ordem, torna-se impraticável analisar o comportamento local da função de maneira analítica. Em tais situações, é necessário iniciar com um valor inicial, por exemplo, $x_0 = -6$, e seguir a direção do gradiente. A direção do gradiente orienta para onde devemos nos mover na busca pelo ponto mínimo ou máximo, mas não especifica a magnitude do deslocamento, conhecida como "tamanho do passo".

Como exemplo, apresenta-se a seguir a execução de três passos do método do gradiente descendente para a função da Equação 4.

Configuração Inicial:

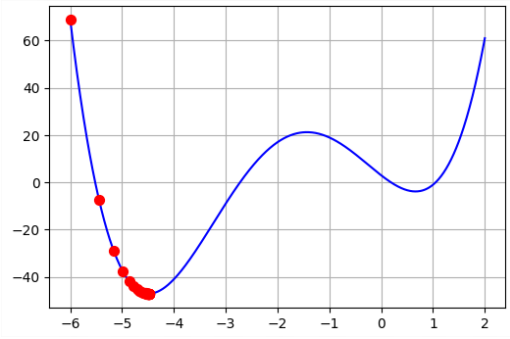
- Função:
  - $f(x) = x^4 + 7x^3 + 5x^2 - 17x + 3$
- Gradiente da função:
  - $\nabla f(x) = 4x^3 + 21x^2 + 10x - 17$
- Ponto Inicial:
  - $x_0 = -6$
- Taxa de Aprendizagem:
  - $\gamma = 0.001$

Método do Gradiente Descendente:

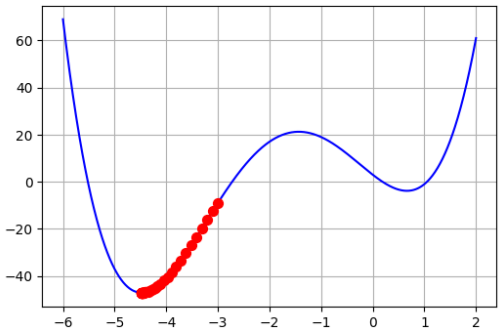
- Passo 1:
  - Calcular o gradiente em $x_0 = -6$:
    $\nabla f(-6) = 4(-6)^3 + 21(-6)^2 + 10(-6) - 17$
    $= 4(-216) + 21(36) - 60 - 17$
    $= -864 + 756 - 60 - 17$
    $= -185$
  - Atualizar x:
    $x^{Novo} = -6 - 0.001 \times (-185)$
    $\approx -6 + 0.185$
    ≈ -5.815
- Passo 2:
  - Calcular o gradiente em $x_1 = -5.815$:
    $\nabla f(-5.815) = 4(-5.815)^3 + 21(-5.815)^2 + 10(-5.815) - 17$
    $\approx 4(-197.753) + 21(33.834) - 58.15 - 17$
    $\approx -791.012 + 710.514 - 58.15 - 17$
    ≈ -155.648
  - Atualizar x:
    $x^{Novo} = -5.815 - 0.001 \times (-155.648)$
    $\approx -5.815 + 0.155648$
    ≈ -5.659
- Passo 3:
  - Calcular o gradiente em $x_2 = -5.659$:
    $\nabla f(-5.659) = 4(-5.659)^3 + 21(-5.659)^2 + 10(-5.659) - 17$
    $\approx 4(-181.225) + 21(32.024) - 56.59 - 17$
    $\approx -724.9 + 672.504 - 56.59 - 17$
    ≈ -125.986
  - Atualizar x:
    $x^{Novo} = -5.659 - 0.001 \times (-125.986)$
    $\approx -5.659 + 0.125986$
    ≈ -5.533

- Passo $4, 5, ..., n$:
  - ◇ Repetindo o mesmo procedimento. O método deve continuar até que uma condição de parada seja atingida.

A Figura 7.3 apresenta o gráfico com diversos passos do método do gradiente descendente para a função $f(x) = x^4 + 7x^3 + 5x^2 - 17x + 3$ implementado em Python. O algoritmo segue a direção do gradiente negativo na busca pelo ponto mínimo, com a condição de parada quando $|\nabla f(x)| < \varepsilon$, onde $\varepsilon = 0.1$ e $\gamma = 0.003$.



(a) O valor inicial foi definido como $x_0 = -6$, com a aproximação do valor do mínimo global após 30 iterações.



(b) O valor inicial foi definido como $x_0 = -3$, com a aproximação do valor do mínimo global após 40 iterações.

Figura 7.3: Exemplo do método do gradiente descendente para a função $f(x) = x^4 + 7x^3 + 5x^2 - 17x + 3$. O algoritmo segue a direção do gradiente negativo na busca pelo ponto mínimo, com a condição de parada quando $|\nabla f(x)| < \varepsilon$, onde $\varepsilon = 0.1$ e $\gamma = 0.003$.

Neste caso, considerando a inicialização dos valores de x_0 na "bacia" de atração correta, o algoritmo converge para uma aproximação do valor que minimiza a função objetivo nas duas situações. No entanto, caso x_0 fosse definido à direita (por exemplo, $x_0 = 0$), o gradiente negativo orientaria em direção ao mínimo local mais próximo, localizado entre $x = 0$ e $x = 1$, que não é o mínimo global.

7.2.2. Gradiente Descendente com Função Multivariada

Considere a seguinte função quadrática de duas variáveis, x_1 e x_2, que é definida em um espaço bidimensional e expressa a relação entre estas duas variáveis através de termos quadráticos e lineares.

$$f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 2 & 1 \\ 1 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 5 \\ 3 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{7}$$

A função da Equação 7 em sua forma matricial pode ser representada da forma geral como

$$f(x) = \frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x, \tag{8}$$

onde $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $A = \begin{bmatrix} 2 & 1 \\ 1 & 20 \end{bmatrix}$ e $b = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$.

Para transformar a função f da forma matricial em uma expressão algébrica, vamos aplicar algumas operações e simplificação nos termos envolvidos.

- Cálculo de Ax:
  - $Ax = \begin{bmatrix} 2 & 1 \\ 1 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \times x_1 + 1 \times x_2 \\ 1 \times x_1 + 20 \times x_2 \end{bmatrix} = \begin{bmatrix} 2x_1 + x_2 \\ x_1 + 20x_2 \end{bmatrix}$
- Cálculo de $x^{\mathrm{T}}Ax$:
  - $x^{\mathrm{T}}Ax = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2x_1 + x_2 \\ x_1 + 20x_2 \end{bmatrix} = x_1(2x_1 + x_2) + x_2(x_1 + 20x_2)$
    $= 2x_1^2 + x_1x_2 + x_1x_2 + 20x_2^2 = 2x_1^2 + 2x_1x_2 + 20x_2^2$
- Cálculo de $\frac{1}{2}x^{\mathrm{T}}Ax$:
  - $\frac{1}{2}x^{\mathrm{T}}Ax = \frac{1}{2}(2x_1^2 + 2x_1x_2 + 20x_2^2) = x_1^2 + x_1x_2 + 10x_2^2$
- Cálculo de $b^{\mathrm{T}}x$:
  - $b^{\mathrm{T}}x = \begin{bmatrix} 5 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 5x_1 + 3x_2$
- Cálculo de $\frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x$:
  - $\frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x = (x_1^2 + x_1x_2 + 10x_2^2) - (5x_1 + 3x_2)$
    $= x_1^2 + x_1x_2 + 10x_2^2 - 5x_1 - 3x_2$

Dessa forma, temos que $f(x)$ é algebricamente equivalente a

$$f(x_1, x_2) = x_1^2 + x_1x_2 + 10x_2^2 - 5x_1 - 3x_2 \tag{9}$$

Para funções de várias variáveis, o gradiente é o vetor composto pelas derivadas parciais da função em relação a cada uma das variáveis. De forma geral, temos que

$$\nabla f(x) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \quad \frac{\partial f(\mathbf{x})}{\partial x_2}, \quad \ldots \quad \frac{\partial f(\mathbf{x})}{\partial x_n}, \right]^{\mathrm{T}} \tag{10}$$

Além de indicar a direção de maior crescimento da função, o vetor de gradientes também representa a taxa de variação da função em cada dimensão do espaço. A seguir, vamos calcular o gradiente da função da Equação 9, que é um vetor das derivadas parciais em relação a x_1 e x_2.

- Cálculo da derivada parcial em relação a x_1:
  - $\frac{\partial}{\partial x_1}(x_1^2 + x_1x_2 + 10x_2^2 - 5x_1 - 3x_2) = 2x_1 + x_2 - 5$
- Cálculo da derivada parcial em relação a x_2:
  - $\frac{\partial}{\partial x_2}(x_1^2 + x_1x_2 + 10x_2^2 - 5x_1 - 3x_2) = x_1 + 20x_2 - 3$

Portanto, temos que o gradiente de $f(x_1, x_2)$ é

$$\nabla f(x_1, x_2) = \begin{bmatrix} 2x_1 + x_2 - 5 \\ x_1 + 20x_2 - 3 \end{bmatrix} \tag{11}$$

A Figura 7.4 apresenta o comportamento do algoritmo gradiente descendente na função da Equação 9 em uma superfície quadrática bidimensional. O valor inicial é definido como $x_0 = [-3, -1]$ e a linha representa a sequência de estimativas que convergem para o valor mínimo.

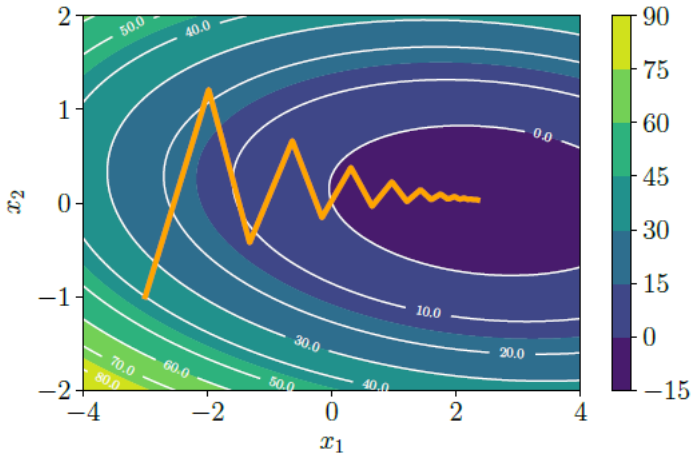


Figura 7.4: Exemplo do método do gradiente descendente para a função $f(x_1, x_2) = x_1^2 + x_1x_2 + 10x_2^2 - 5x_1 - 3x_2$ em uma superfície quadrática bidimensional com $\gamma = 0.085$.

Como exemplo, apresenta-se a seguir a execução de três passos do método do gradiente descendente para a função da Equação 9.

Configuração Inicial:

- Função:
  - $f(x_1, x_2) = x_1^2 + x_1x_2 + 10x_2^2 - 5x_1 - 3x_2$
- Gradiente da função:
  - $\nabla f(x_1, x_2) = \begin{bmatrix} 2x_1 + x_2 - 5 \\ x_1 + 20x_2 - 3 \end{bmatrix}$
- Ponto Inicial:
  - $x_0 = [-3, -1]$
- Taxa de Aprendizagem:
  - $\gamma = 0.085$

Método do Gradiente Descendente:

- Passo 1:

- ◇ Calcular o gradiente em $x_0 = [-3, -1]$:
  $\frac{\partial}{\partial x_1} = 2(-3) - 1 - 5 = -12$
  $\frac{\partial}{\partial x_2} = -3 + 20(-1) - 3 = -26$
- ◇ Atualizar x:
  $x_1^{Novo} = -3 - 0.085 \times (-12) = -3 + 1.02 \approx -1.98$
  $x_2^{Novo} = -1 - 0.085 \times (-26) = -1 + 2.21 \approx 1.21$

- Passo 2:
  - ◇ Calcular o gradiente em $x_1 = [-1.98, 1.21]$:
    $\frac{\partial}{\partial x_1} = 2(-1.98) + 1.21 - 5 = -8.75$
    $\frac{\partial}{\partial x_2} = -1.98 + 20(1.21) - 3 = 19.82$
  - ◇ Atualizar x:
    $x_1^{Novo} = -1.98 - 0.085 \times (-8.75) = -1.98 + 0.74375 \approx -1.23625$
    $x_2^{Novo} = 1.21 - 0.085 \times (19.82) = 1.21 - 1.6847 \approx -0.4747$
- Passo 3:
  - ◇ Calcular o gradiente em $x_2 = [-1.23625, -0.4747]$:
    $\frac{\partial}{\partial x_1} = 2(-1.2362) - 0.4747 - 5 = -7.9472$
    $\frac{\partial}{\partial x_2} = -1.23625 + 20(-0.4747) - 3 = -11.73$
  - ◇ Atualizar x:
    $x_1^{Novo} = -1.23625 - 0.085 \times (-7.9472) = -1.236 + 0.675 \approx -0.561$
    $x_2^{Novo} = -0.4747 - 0.085 \times (-11.73) = 0.4747 + 0.997 \approx 1.4717$
- Passo $4, 5, ..., n$:
  - ◇ Repetindo o mesmo procedimento. O método deve continuar até que uma condição de parada seja atingida.

A Figura 7.5 apresenta o gráfico com diversos passos do método do gradiente descendente para a função $f(x_1, x_2) = x_1^2 + x_1x_2 + 10x_2^2 - 5x_1 - 3x_2$ implementado em Python. O valor inicial foi definido como $x_0 = [-6, 8]$, seguindo a direção do gradiente negativo na busca pelo ponto mínimo, com a condição de parada quando $|\nabla f(x_1, x_2)| < \varepsilon$, onde $\varepsilon = 0.1$ e $\gamma = 0.01$. Neste caso, considerando que a função $f(x_1, x_2)$ é estritamente convexa (implicando que a função tem um único mínimo global), o algoritmo convergiu para uma aproximação do valor que minimiza a função objetivo depois de 263 iterações.

7.2.3. Tamanho do Passo

No contexto de algoritmos de otimização, como o gradiente descendente, o termo "tamanho do passo" é frequentemente usado como sinônimo para "taxa de aprendizagem". Esta taxa de aprendizagem, denotada muitas vezes por γ, determina o tamanho do passo tomado em direção ao mínimo de uma função a cada iteração do algoritmo.

A Figura 7.6 apresenta algumas intuições sobre a escolha adequada do tamanho do passo para o gradiente descendente implementado em Python. Esta escolha pode ser baseada em experiência, experimentação ou métodos mais sofisticados como o ajuste automático da taxa de aprendizagem durante o treinamento. Na Figura 7.6a, observa-se que quando o tamanho do passo é muito pequeno, a descida do gradiente pode ser

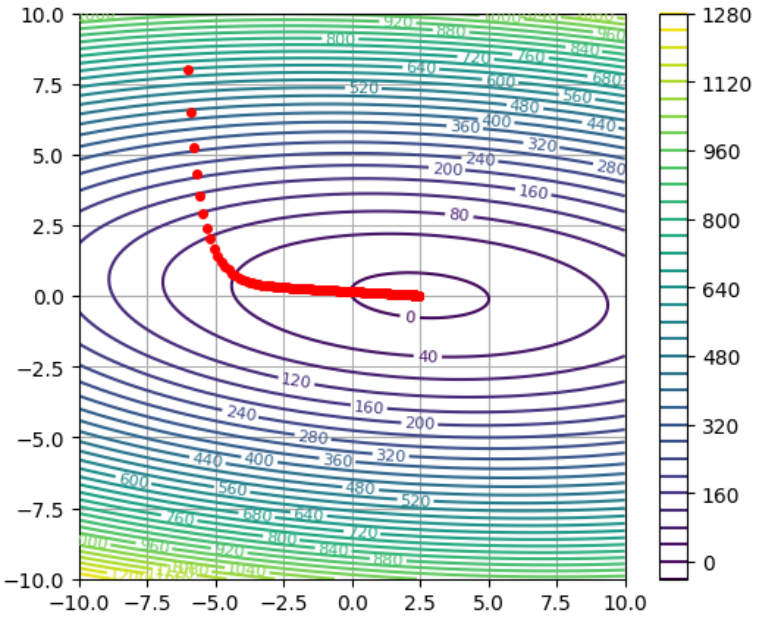
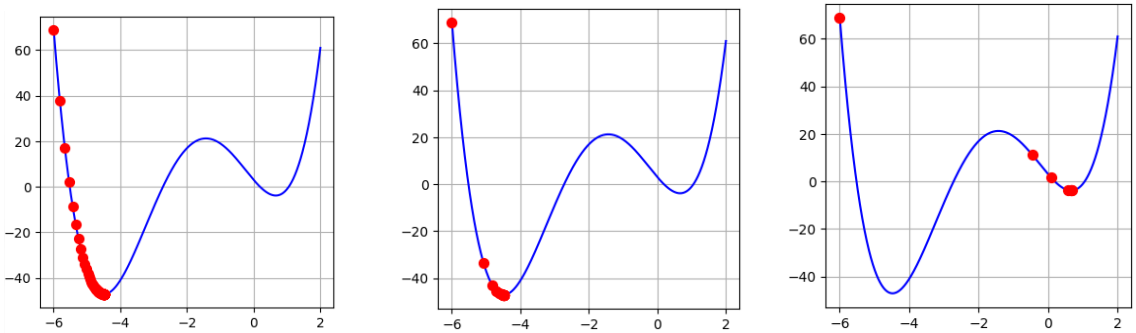


Figura 7.5: Exemplo do método do gradiente descendente para a função $f(x_1,x_2) = x_1^2 + x_1x_2 + 10x_2^2 - 5x_1 - 3x_2$. O valor inicial foi definido como $x_0 = [-7,7.5]$, seguindo a direção do gradiente negativo na busca pelo ponto mínimo, com a condição de parada quando $|\nabla f(x_1,x_2)| < \varepsilon$, onde $\varepsilon = 0.1$ e $\gamma = 0.01$.

lenta. Além do mais, na Figura 7.6b, observa-se que se o tamanho do passo é um pouco maior, a descida do gradiente pode ser mais rápida. No entanto, na Figura 7.6c, mesmo considerando a inicialização de x_0 na "bacia" de atração correta, quando o tamanho do passo é muito grande, a descida do gradiente pode ultrapassar o valor do mínimo global, e portanto, pode não convergir para o valor que minimiza a função objetivo.



(a) Aproximação do valor do mínimo global após 96 iterações, com $\gamma = 0.001$

(b) Aproximação do valor do mínimo global após 16 iterações, com $\gamma = 0.005$

(c) Divergência do valor do mínimo global após 6 iterações, com $\gamma = 0.03$.

Figura 7.6: Exemplo do método do gradiente descendente para a função $f(x) = x^4 + 7x^3 + 5x^2 - 17x + 3$. O valor inicial foi definido como $x_0 = -6$, com a condição de parada quando $|\nabla f(x)| < \varepsilon$, onde $\varepsilon = 0.1$ com variações do valor de γ.

Os métodos de gradiente adaptativo redimensionam o tamanho do passo a cada

iteração, dependendo das propriedades locais da função. De forma geral, existem duas heurísticas simples [Toussain 2012]:

- Redução do tamanho do passo quando o valor da função aumenta.

  Esta heurística baseia-se na observação de que se o valor da função objetivo aumenta após um passo de gradiente, é provável que o passo tenha sido excessivamente grande. Em outras palavras, o movimento foi tão agressivo que ultrapassou o mínimo local, levando a um aumento no valor da função em vez de uma diminuição. Este é um indicativo claro de que o algoritmo está potencialmente se afastando do mínimo desejado.

  Procedimento:

  - ◇ Após uma iteração, deve-se verificar se o valor da função *f(x)* aumentou.
  - ◇ Neste caso, cancela-se a última atualização realizada (desfazendo o último passo) para diminuir o tamanho do passo. Isso é feito multiplicando a taxa de aprendizado atual por um fator $\delta < 1$.

- Aumento do tamanho do passo quando o valor da função diminui.

  A segunda heurística parte do princípio de que se o valor da função diminui após um passo de gradiente, o passo atual poderia ser potencialmente maior para acelerar a convergência. Isso sugere que o passo tomado foi conservador e que um passo maior poderia alcançar uma redução ainda mais significativa no valor da função, otimizando assim o processo de otimização.

  Procedimento:

  - ◇ Após uma iteração, deve-se verificar se o valor da função *f(x)* diminuiu.
  - ◇ Neste caso, tenta-se aumentar o tamanho do passo. Isso pode ser feito multiplicando a taxa de aprendizado atual por um fator $\delta' > 1$.

Essas heurísticas são simples, porém eficazes, e são frequentemente incorporadas em métodos mais sofisticados de gradiente adaptativo, como Adam, RMSprop, entre outros, que ajustam automaticamente o tamanho do passo com base em estimativas de momentos de primeira e segunda ordem dos gradientes. A escolha de implementar tais heurísticas depende das características específicas do problema e do comportamento desejado para a convergência do algoritmo de otimização.

7.2.4. Gradiente Descendente com *Momentum*

O gradiente descendente com *momentum* [Rumelhart et al. 1986] é uma extensão da técnica tradicional de descida do gradiente que adiciona um componente de "momento" que torna o algoritmo mais eficiente para lidar com superfícies de erro que possuem curvaturas irregulares ou mínimos locais. O componente de momento é um termo extra que captura a essência do que ocorreu na iteração anterior.

O conceito fundamental do *momentum* é simular o comportamento de uma partícula se movendo em uma superfície de erro com alguma inércia. Em termos físicos, isso

significa que a partícula (ou os parâmetros do modelo) não somente reage ao gradiente (força) local mas também mantém uma direção e velocidade consistentes baseadas em seu movimento anterior.

A implementação clássica do *momentum* incorpora uma fração do gradiente anterior. A atualização do vetor de *momentum* é caracterizado por

$$m = \beta m - \gamma \nabla f(x_i), \tag{12}$$

onde m é o vetor de *momentum* e β é um fator de decaimento que determina o quanto do movimento anterior será retido em comparação ao novo gradiente. Um valor alto significa que mais do *momentum* anterior é preservado, o que pode ajudar a suavizar a atualização dos parâmetros e permitir que o otimizador escape de mínimos locais ou avance mais rapidamente em superfícies planas.

A atualização do ponto x_{i+1} é caracterizado pelo valor de x_i adicionado ao vetor de *momentum*, que é definido por

$$x_{i+1} = x_i + m, \tag{13}$$

onde o vetor de *momentum* contém uma combinação do gradiente atual (direção imediata de descida mais íngreme) e uma fração do vetor de *momentum* anterior, permitindo que o algoritmo mantenha a direção geral de movimento anterior enquanto ainda responde ao gradiente local.

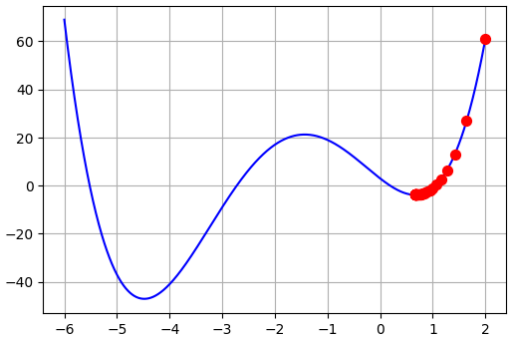
A Figura 7.7 apresenta o gráfico com diversos passos do método do gradiente descendente com *momentum* implementado em Python, destacando que o passo dado em cada iteração é influenciado tanto pelo gradiente atual quanto pelo impulso adquirido pela iteração anterior. O valor inicial como o valor de $x_0 = 2$. A condição de parada é definida quando $|\nabla f(x)| < \varepsilon$, onde $\varepsilon = 0.1$ e $\gamma = 0.005$ e $\beta = 0.9$. Observa-se que mesmo definindo o valor de x_0 do lado da "bacia" de atração contendo um vale com um mínimo local, o algoritmo gradiente descendente com *momentum* utilizou o impulso adquirido pela iteração anterior para escalar para o vale vizinho e convergir para o mínimo global.

Exitem outras variantes de *momentum* que incluem não apenas o gradiente mas também uma diferença dos parâmetros (Δx) entre iterações. Isso ajuda a ajustar a magnitude da atualização não só com base na inclinação (gradiente) mas também considerando a variação no espaço de parâmetros.

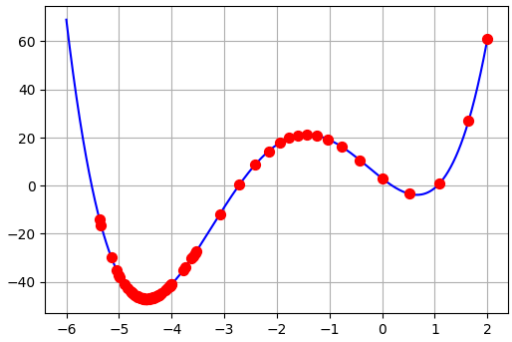
7.2.5. Descida do Gradiente Estocástica

Calcular o gradiente exato de uma função pode ser um processo oneroso em termos de tempo e recursos computacionais, especialmente em contextos onde as funções envolvidas são complexas ou os conjuntos de dados são grandes. Essa aproximação ainda é válida contanto que indique uma direção próxima à do gradiente real. No entanto, frequentemente é viável obter uma aproximação econômica do gradiente. Essa aproximação ainda é válida contanto que indique uma direção próxima à do gradiente real.

Uma das técnicas que utilizam essa ideia é a Descida de Gradiente Estocástica



(a) Implementação sem *momentum*, onde o algoritmo atinge um mínimo local após 41 iterações.



(b) Implementação com *momentum*, onde o algoritmo atinge uma aproximação do valor do mínimo global após 83 iterações.

Figura 7.7: Exemplo do método do gradiente descendente para a função $f(x) = x^4 + 7x^3 + 5x^2 - 17x + 3$. O valor inicial foi definido como $x_0 = 2$ com a condição de parada quando $|\nabla f(x)| < \varepsilon$, onde $\varepsilon = 0.1$, $\gamma = 0.003$ e $\beta = 0.92$.

(SGD), uma adaptação estocástica do método tradicional de descida de gradiente. A descida de gradiente clássica minimiza uma função objetivo atualizando os parâmetros na direção oposta ao gradiente dessa função. Quando a função objetivo é a soma de várias funções diferenciáveis, como é comum em problemas de aprendizado de máquina, o SGD oferece uma alternativa eficiente ao calcular o gradiente de um subconjunto aleatoriamente dos pontos de dados a cada iteração.

O termo "estocástico" neste contexto significa que não temos conhecimento exato do gradiente, mas sim de uma estimativa aproximada e ruidosa dele. Ao controlar a distribuição de probabilidade desses gradientes aproximados, é possível assegurar teoricamente que o SGD convergirá. No aprendizado de máquina, dado $n = 1, ..., N$ pontos de dados, frequentemente consideramos funções objetivo caracterizadas pela soma das perdas L_n incorridas por cada exemplo n. Em notação matemática, temos a forma

$$L(\theta) = \sum_{n=1}^{N} L_n(\theta), \tag{14}$$

onde θ é o vetor de parâmetros, ou seja, queremos encontrar θ que minimize L.

A descida gradiente padrão, conforme introduzido anteriormente, é um método de otimização em "lote", ou seja, a otimização é realizada usando o conjunto de treinamento completo, atualizando o vetor de parâmetros de acordo com

$$\theta_{i+1} = \theta_i - \gamma_i \left(\nabla L(\theta_i)\right)^{\mathrm{T}} = \theta_i - \gamma_i \sum_{n=1}^{N} \left(\nabla L_n(\theta_i)\right)^{\mathrm{T}}. \tag{15}$$

A avaliação do gradiente da soma das funções pode envolver cálculos dispendiosos dos gradientes para cada valor de L_n. Especialmente quando o conjunto de dados de treinamento é extenso e/ou as fórmulas para os gradientes não são diretas, o custo

associado à avaliação dessas somas de gradientes pode ser proibitivo.

Neste caso, podemos reduzir a quantidade de cálculo calculando a soma da Equação 15 sobre um conjunto menor de L_n. Em contraste com a descida gradiente em lote, que usa todo L_n para $n = 1, ..., N$, opcionalmente podemos escolher aleatoriamente um subconjunto de L_n para descida gradiente de minilote. No caso extremo, podemos selecionar aleatoriamente apenas um único L_n para estimar o gradiente.

Um dos principais motivos para considerar o uso de gradientes aproximados são as restrições práticas de implementação, como o tamanho da memória da unidade de processamento central (CPU) e/ou unidade de processamento gráfico (GPU), bem como, os limites de tempo computacional. Tamanhos grandes de minilotes fornecerão estimativas precisas do gradiente, reduzindo a variação na atualização dos parâmetros. Além disso, grandes minilotes aproveitam operações matriciais altamente otimizadas em implementações vetorizadas de custo e gradiente. Em contraste, pequenos minilotes são rápidos de estimar. Se mantivermos o tamanho do minilote pequeno, o ruído em nossa estimativa de gradiente nos permitirá sair de alguns ótimos locais ruins, nos quais poderíamos ficar presos.

No aprendizado de máquina, o objetivo geral é melhorar o desempenho da generalização. Neste caso, como o objetivo no aprendizado de máquina não precisa necessariamente de uma estimativa precisa do mínimo da função objetivo, gradientes aproximados usando abordagens de minilote têm sido amplamente utilizados, sendo eficazes em problemas de aprendizado de máquina em grande escala.

7.3. Otimização Restrita e Multiplicadores de Lagrange

Nesta seção, continuaremos a analisar o problema de encontrar o mínimo de uma função $f : \mathbb{R}^D \to \mathbb{R}$, contudo considerando a imposição de *restrições*. Desta forma, para funções com valores reais $g_i : \mathbb{R}^D \to \mathbb{R}$ para $i = 1, \ldots, m$, consideramos o problema de otimização restrita

$$\begin{aligned} \min_x \quad & f(x) \\ \text{sujeito a} \quad & g_i(x) \leq 0 \quad \text{para todo} \quad i = 1, \ldots, m. \end{aligned} \tag{16}$$

A Figura 7.8 apresenta uma ilustração de um problema de minimização restrito. Na oportunidade, vale ressaltar que as funções f e g_i podem ser não convexas, uma vez que consideraremos o caso convexo na próxima seção. Além disso, neste momento, destaca-se que as restrições g_i estão expressas como inequações maiores ou iguais a 0 (zero).

Uma maneira óbvia, mas não muito prática, de converter o problema restrito (20) em um problema irrestrito é usar uma função indicadora

$$J(x) = f(x) + \sum_{i=1}^{m} 1(g_i(x)), \tag{17}$$

onde $1(z)$ é uma função degrau infinita

$$1(z) = \begin{cases} 0 & \text{se } z \leq 0 \\ \infty & \text{caso contrário} \end{cases}. \tag{18}$$

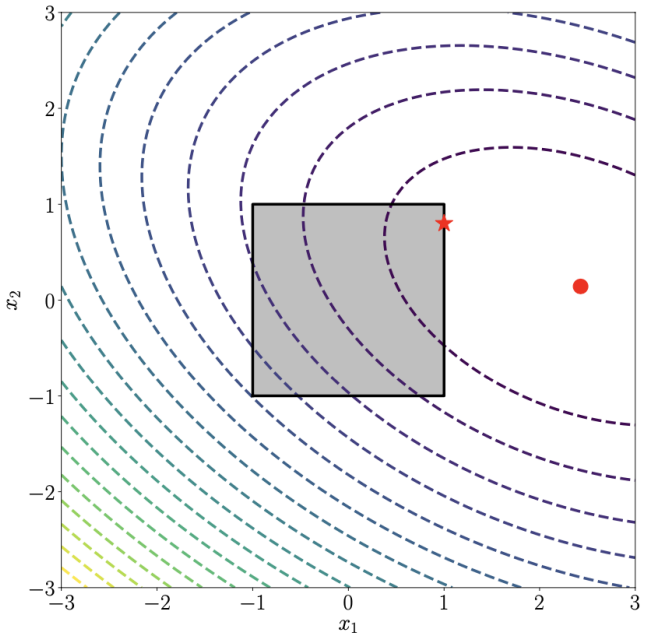


Figura 7.8: Ilustração de otimização restrita. O problema irrestrito (indicado pelas curvas de nível) tem um mínimo no lado direito (indicado pelo círculo). As restrições da caixa $-1 \leq x_1 \leq 1$ e $-1 \leq x_2 \leq 1$ exigem que a solução ótima esteja dentro da caixa, resultando em um valor ótimo indicado pela estrela.

O que pode ser compreendido como a aplicação de uma penalidade infinita se a restrição não for satisfeita e, portanto, forneceria a mesma solução. No entanto, esta função de passo infinito é igualmente difícil de otimizar.

Para superar esta dificuldade, utilizamos os multiplicadores de Lagrange, com os quais visa-se substituir a função degrau por uma função linear. Desta forma, associamos ao problema (20) o Lagrangiano introduzindo os multiplicadores de Lagrange $\lambda_i \leq 0$ correspondentes a cada restrição de desigualdade respectivamente de modo que

$$\begin{aligned} \mathfrak{L}(x, \lambda) &= f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) \\ &= f(x) + \lambda^\top g(x), \end{aligned} \tag{19}$$

sendo que na última linha concatenamos todos os multiplicadores de Lagrange em um vetor $\lambda \in \mathbb{R}^m$, e todas as restrições $g_i(x)$ em um vetor $g(x)$.

Quando trabalhamos com Multiplicadores de Lagrange, nos deparamos com o *Problema Dual de Lagrange*. Desta forma, é importante lembrar que Dualidade neste contexto é a ideia de converter um problema de otimização em um conjunto de variáveis x (chamadas de variáveis primárias), em outro problema de otimização em um conjunto diferente de variáveis λ (chamadas de variáveis duais).

Assim, o problema em (20)

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{sujeito a} \quad & g_i(x) \leq 0 \quad \text{para todo} \quad i = 1, \ldots, m. \end{aligned} \tag{20}$$

é considerado o Problema Primordial (Primal) em relação às variáveis primárias x. Logo, o Problema Dual de Lagrange correspondente pode ser expresso como

$$\begin{aligned} \max_{\lambda \in \mathbb{R}^m} \quad & \mathfrak{D}(\lambda) \\ \text{sujeito a} \quad & \lambda \geq 0 \end{aligned} \tag{21}$$

sendo λ as variáveis duais e $\mathfrak{D}(\lambda) = \min_{x \in \mathbb{R}^d} \mathfrak{L}(x, \lambda)$.

Inicialmente, pode não ficar tão evidente porque o problema dual passa a ser um problema de maximização, mas para entender esse aspecto é importante lembrar que as restrições são expressas como $g_i(x) \leq 0$. Desta maneira, o resultado dessas funções deverá ser sempre negativo ou no máximo igual a zero. E como $\lambda \geq 0$, o resultado da multiplicação de λ por $g_i(x)$ por também sempre negativo ou no máximo zero.

Assim, quando as restrições são atendidas temos que o resultado da função de Lagrange será o resultado da função $f(x)$ menos um valor ou no máximo mais 0. Desta forma, $\mathfrak{L}(x, \lambda) \leq f(x)$, logo o objetivo agora é maximizar essa função em relação à λ, para encontrarmos um resultado o mais próximo possível de $f(x)$.

Para ficar ainda mais claro, vamos considerar uma solução viável $\tilde{x}$ para o Problema Primordial (PP),

$$\mathfrak{D}(\lambda) = \min_{x \in \mathbb{R}^d} \mathfrak{L}(\tilde{x}, \lambda)$$

$$\mathfrak{L}(\tilde{x}, \lambda) = f(\tilde{x}) + \sum_{i=1}^{m} \lambda_i g_i(\tilde{x}) \quad \leq \quad f(\tilde{x})$$

Consequentemente, se considerarmos uma solução ótima p^* para o Problema Primordial, temos que

$$\mathfrak{D}(\lambda) \leq p^*$$

Desta forma, temos que a função Dual de Lagrange funciona como um limite inferior para o valor ótimo do Problema Primordial. Portanto, nosso objetivo será encontrar o melhor limite inferior, ou seja, o expresso no problema (21).

Quando trabalhamos com o problema Dual de Lagrange, é importante conhecermos o conceito da *desigualdade minimax*, que diz que para qualquer função com dois argumentos $\varphi(x, y)$, o *maximin* é menor que o *minimax*, ou seja,

$$\max_y \min_x \varphi(x, y) \leq \min_x \max_y \varphi(x, y).$$

Em outras palavras, caso busquemos o mínimo de uma função em relação a x e depois busquemos o máximo em relação a y, o resultado obtido será sempre menor ou igual ao resultado caso eu busquemos primeiro o máximo em relação a y e depois o mínimo em relação a x.

Isso pode parecer confuso, mas na verdade é simples de provar considerando uma desigualdade mais básica expressa a seguir

$$\textit{Para todo } x,y \quad \min \varphi(x,y) \leq \max \varphi(x,y).$$

Assim, é óbvio que caso eu busquemos o mínimo de uma a função em relação a x o resultado será sempre menor ou igual ao resultado caso eu busquemos o máximo em relação a y. Para ficar mais claro, vamos considerar dois pontos x_0 e y_0 quaisquer, nós temos que

$$\min_x \varphi(x,y) \leq \varphi(x_0,y_0) \leq \max_y \varphi(x,y)$$

$$\min_x \varphi(x,y) \leq \max_y \varphi(x,y)$$

$$\max_y \min_x \varphi(x,y) \leq \min_x \max_y \varphi(x,y).$$

Portanto, percebemos que ao suprimir $\varphi(x_0,y_0)$, chegamos na desigualdade mais básica e em seguida chegamos na desigualdade *minimax*. É importante observar que a todo tempo estamos utilizando uma desigualdade não-estrita, ou seja, o símbolo de menor ou igual ($\leq$), pois os valores encontrados de fato podem ser iguais.

Além do conceito de desigualdade *minimax*, quando trabalhamos com o Problema Dual de Lagrange é importante conhecer o conceito de *dualidade fraca*, para mostrar que os valores primordiais são sempre maiores ou iguais aos valores duais.

Assim, considerando o problema inicial em relação à função indicadora, nós temos que $J(x)$ poderia ser expresso como

$$J(x) = \max_{\lambda \geq 0} \mathfrak{L}(x,\lambda) \tag{22}$$

Uma vez que que o Lagrangiano $\mathfrak{L}(x,\lambda)$ pode ser visto como um limite inferior da função indicadora. Consequentemente, o problema poderia ser reescrito da seguinte forma em relação a $J(x)$

$$\min_{x \in \mathbb{R}^d} \max_{\lambda \geq 0} \mathfrak{L}(x,\lambda) \tag{23}$$

Portanto, considerando a *desigualdade minimax*, ou seja, ao inverter a ordem do mínimo e do máximo sabemos que encontraremos um valor ainda menor,

$$\min_{x \in \mathbb{R}d} \max_{\lambda \geq 0} \mathfrak{L}(x,\lambda) \geq \max_{\lambda \geq 0} \min_{x \in \mathbb{R}^d} \mathfrak{L}(x,\lambda) \tag{24}$$

Assim, observe que o lado direito da inequação acima é exatamente a definição do problema dual de Lagrange $\mathfrak{D}(\lambda)$, sobre a qual podemos afirmar o seguinte:

- Ao contrário do problema de otimização original, que possuía restrições, $\min_{x \in \mathbb{R}^d} \mathfrak{L}(x,\lambda)$ é um problema de otimização irrestrito para um determinado valor de λ;

- Se resolver $\min_{x\in\mathbb{R}^d} \mathfrak{L}(x,\lambda)$ for fácil, então o problema geral será fácil de se resolver;
- Além disso, sabendo que $\mathfrak{L}(x,\lambda)$ é afim em relação a λ, podemos afirmar que $\min_{x\in\mathbb{R}^d} \mathfrak{L}(x,\lambda)$ é um mínimo pontual de funções afins de λ;
- $\mathfrak{D}(\lambda)$ é côncavo mesmo que $f(\cdot)$ e $g_i(\cdot)$ possam ser não-convexos;
- Logo, o problema externo, maximização sobre λ, é o máximo de uma função côncava e pode ser calculado com eficiência.
- Assumindo que $f(\cdot)$ e $g_i(\cdot)$ são diferenciáveis, encontramos o problema dual de Lagrange diferenciando o Lagrangiano em relação a x, definindo o diferencial como zero e resolvendo o valor ideal.

É importante ressaltar que até o momento trabalhamos apenas com restrições expressas como inequações. Contudo, caso tivéssemos restrições adicionais de igualdade, conforme descrito abaixo

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{sujeito a} \quad & g_i(x) \leq 0 \quad \text{para todo} \quad i = 1,\ldots,m \\ \text{sujeito a} \quad & h_j(x) = 0 \quad \text{para todo} \quad j = 1,\ldots,n. \end{aligned} \tag{25}$$

7.3.1. Exemplo de Aplicação dos Multiplicadores de Lagrange

Considere o seguinte problema de minimização com apenas uma restrição:

$$\begin{aligned} \min \quad & f(x,y) = x^2+y^2 \\ \text{sujeito a} \quad & g(x,y) = x+y = 2 \end{aligned} \tag{26}$$

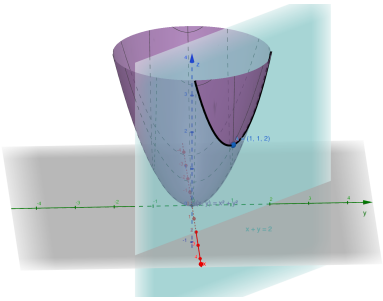
Consideremos inicialmente a análise geométrica para este problema apresentado na Figura 7.9, em que podemos ver que a intersecção entre a forma gerada pela função objetivo e a restrição resulta em uma parábola. É possível observar que o mínimo global da função irrestrita seria no ponto (0,0), contudo considerando a restrição a qual o problema está sujeito o mínimo global passa a ser no ponto (1,1).

É importante notar que no ponto ótimo encontrado, os vetores gradientes da função objetivo e da função de restrição são paralelos, ou seja, linearmente dependentes, portanto podem ser escritos da seguinte forma:

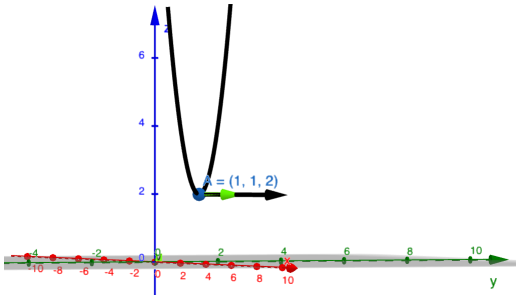
$$\nabla f(x,y) = \lambda \nabla g(x,y)$$

sendo λ o Multiplicador de Lagrange. Portanto, para encontrar a solução para esse problema a partir de uma análise algébrica poderíamos iniciar estruturando as funções conforme a equação anterior, portanto teríamos que:

$$(2x,2y) = \lambda \cdot (1,1)$$



(a) Função objetivo mostrada pela forma em roxo e função de restrição representada pelo plano em azul.



(b) Parábola da intersecção entre a forma e o plano e vetores gradientes de ambas as funções no ponto (1,1).

Figura 7.9: Problema de Otimização Restrito (a) Função objetivo e função de restrição do problema 26 no plano tridimensional; (b) Parábola dos pontos que atendem à restrição, com ponto mínimo global encontrado e vetores gradientes da funções.

Consequentemente, obteríamos o seguinte sistema de equações:

$$\begin{cases} 2x = \lambda \cdot 1 \\ 2y = \lambda \cdot 1 \\ x + y = 2 \end{cases}$$

Resolvendo o sistema de equações acima, teríamos que:

$$\begin{cases} x = y = 1 \\ \lambda = 2 \end{cases}$$

Portanto, o ponto mínimo do nosso problema seria o (1,1), e substituindo na função objetivo teríamos que o valor mínimo sería 2, o que condiz com o ponto encontrado a partir da análise geométrica. Portanto, como este é um problema simples, nem analisamos a segunda derivada para verificar se o ponto crítico encontrado é o mínimo, máximo ou ponto de sela, pois obviamente ele é o mínimo.

Na Figura 7.10 vemos de forma clara que a linha tracejada da restrição corta a linhas de contorno da função objetivo até encontrar um ponto em os vetores gradientes de ambas as funções são paralelos. Além disso, a partir desse ponto, verificamos que a linha de restrição volta a cortar linhas de contorno que já haviam sido cortadas anteriormente em outro ponto.

7.4. Otimização Convexa

Um conjunto C é convexo se para qualquer $x, y \in C$ e algum θ com $0 \leq \theta \leq 1$ nós temos

$$\theta x + (1 - \theta) y \in C \tag{27}$$

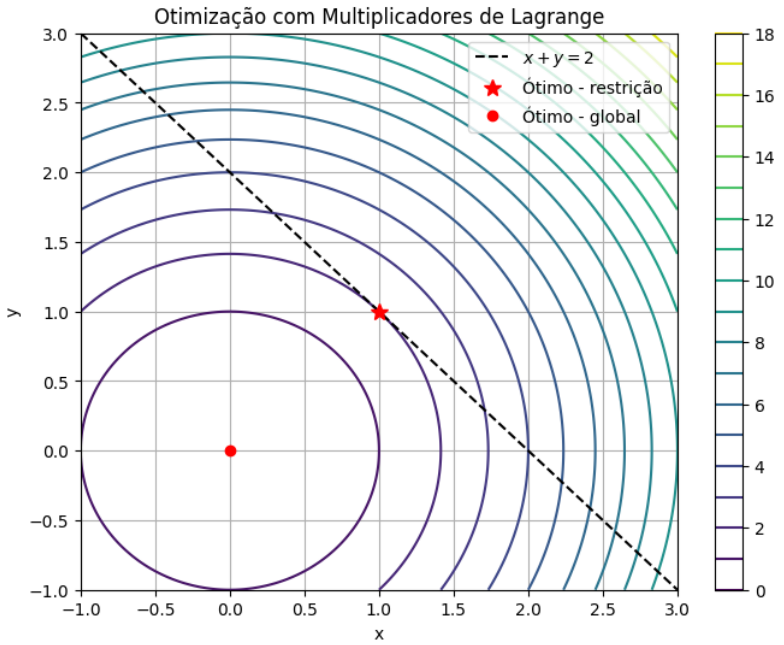


Figura 7.10: Gráfico do problema de otimização restrita 7.9 que apresenta a função objetivo através da linhas de contorno. O ponto ótimo para o problema irrestrito é indicado pelo círculo, enquanto para o problema com restrição é indicado pela estrela.

Conjuntos convexos são aqueles conjuntos que ao conectar uma linha entre dois elementos do conjuntos fica dentro do conjunto. A função $f : R^n \longrightarrow R$ é convexa se o **dom** f é um conjunto convexo e para todo $x, y \in$ **dom** f e $0 \leq \theta \leq 1$, nós temos

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y) \tag{28}$$

Observação: uma função côncava é uma função convexa negativa.

Em resumo, um problema de otimização restrita é chamado de problema de otimização convexa se

$$\begin{aligned} \text{minimizar} \quad & f_0(x) \\ \text{sujeito a} \quad & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{aligned} \tag{29}$$

onde $f_0(x)$ e $f_i(x)$ são funções convexas e $h_i(x) = 0$ são conjuntos convexos e $x \in R^n$ são as variáveis de otimização e f_0 é a função objetivo. Descreveremos duas classes de problemas de otimização convexa que são amplamente utilizados.

7.4.1. Programação Linear

Considere o caso especial quando todas as funções anteriores são lineares, tal que

$$\begin{aligned} \min_{x\in\mathbb{R}} \quad & c^T x \\ \text{sujeito a} \quad & Ax \leq b \end{aligned} \tag{30}$$

- Problema convexo com funções objetivos e restrições afins;

Onde $A \in \mathbb{R}^{m\times d}$ e $b \in \mathbb{R}^m$. Conhecido como uma programação linear. Possui d variáveis e m restrições lineares. O Lagrangiano é dado por

$$\mathfrak{L}(x,\lambda) = c^T x + \lambda^T (Ax - b) \tag{31}$$

Onde $\lambda \in \mathbb{R}^m$ é o vetor de um multiplicador de Lagrange não negativo. Reorganizando os termos correspondentes a x produz

$$\mathfrak{L}(x,\lambda) = (c + A^T\lambda)^T x - \lambda^T b \tag{32}$$

Derivando em relação a x e defini-lo como zero

$$c + A^T\lambda = 0 \tag{33}$$

Entretanto o Lagrangiano dual é $\mathfrak{D}(\lambda) = -\lambda^T b$. O objetivo é minimizar $\mathfrak{D}(\lambda)$. Além da restrição devido à derivada de $\mathfrak{L}(x,\lambda)$ ser zero, também temos o fato de que $\lambda \geq 0$, resultando no seguinte problema de otimização dual

$$\begin{aligned} \max_{\lambda\in\mathbb{R}^m} \quad & -b^T\lambda \\ \text{sujeito a} \quad & c + A^T\lambda = 0 \\ & \lambda \geq 0. \end{aligned} \tag{34}$$

Também é um programa linear, mas com m variáveis

Exemplo de Programação Linear

Considere o seguinte problema de minimização:

$$\min_{x\in\mathbb{R}^2} \quad c^T x = 3x_1 + 2x_2 \tag{35}$$

$$\text{sujeito a} \quad \begin{cases} x_1 + x_2 \geq 4 \\ x_1 - x_2 \geq 1 \\ x_1, x_2 \geq 0 \end{cases} \tag{36}$$

Aqui, $c = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$.

Formulando o Lagrangiano

Reescrevendo as restrições na forma $Ax \leq b$:

$$\begin{cases} -x_1 - x_2 \leq -4 \\ -x_1 + x_2 \leq -1 \\ x_1, x_2 \geq 0 \end{cases}$$

Então, $A = \begin{pmatrix} -1 & -1 \\ -1 & 1 \end{pmatrix}$ e $b = \begin{pmatrix} -4 \\ -1 \end{pmatrix}$.

O Lagrangiano é dado por:

$$\mathfrak{L}(x, \lambda) = 3x_1 + 2x_2 + \lambda_1(-x_1 - x_2 + 4) + \lambda_2(-x_1 + x_2 + 1)$$

Simplificando o Lagrangiano

Expandindo e agrupando termos:

$$\mathfrak{L}(x, \lambda) = (3 - \lambda_1 - \lambda_2)x_1 + (2 - \lambda_1 + \lambda_2)x_2 + 4\lambda_1 + \lambda_2$$

Condições de Otimalidade

Derivando em relação a x_1 e x_2 e igualando a zero:

$$\frac{\partial \mathfrak{L}}{\partial x_1} = 3 - \lambda_1 - \lambda_2 = 0 \implies \lambda_1 + \lambda_2 = 3$$

$$\frac{\partial \mathfrak{L}}{\partial x_2} = 2 - \lambda_1 + \lambda_2 = 0 \implies \lambda_1 - \lambda_2 = 2$$

Resolvendo este sistema de equações:

$$\begin{cases} \lambda_1 + \lambda_2 = 3 \\ \lambda_1 - \lambda_2 = 2 \end{cases}$$

Somando as equações:

$$2\lambda_1 = 5 \implies \lambda_1 = \frac{5}{2}$$

Substituindo λ_1 na primeira equação:

$$\frac{5}{2} + \lambda_2 = 3 \implies \lambda_2 = 3 - \frac{5}{2} = \frac{1}{2}$$

Problema Dual

A função dual é:

$$\mathfrak{D}(\lambda) = -\lambda_1 b_1 - \lambda_2 b_2 = -\lambda_1(-4) - \lambda_2(-1) = 4\lambda_1 + \lambda_2$$

Substituindo λ_1 e λ_2:

$$\mathfrak{D}\left(\frac{5}{2}, \frac{1}{2}\right) = 4 \cdot \frac{5}{2} + \frac{1}{2} = 10 + \frac{1}{2} = 10.5$$

O problema dual se torna:

$$\max_{\lambda_1, \lambda_2 \geq 0} \quad 4\lambda_1 + \lambda_2 \tag{37}$$

$$\text{sujeito a} \quad \begin{cases} \lambda_1 + \lambda_2 = 3 \\ \lambda_1 - \lambda_2 = 2 \end{cases} \tag{38}$$

7.4.2. Programação Quadrática

Considere o caso de uma função objetivo quadrática convexa, onde as restrições são afins

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & \frac{1}{2} x^T Q x + c^T x \\ \text{sujeito a} \quad & Ax \leq b \end{aligned} \tag{39}$$

onde $A \in \mathbb{R}^{m \times d}$ e $b \in \mathbb{R}^m$. Este é conhecido como programação linear com d variáveis e m restrições lineares. O Lagrangiano é dado por

$$\mathfrak{L}(x, \lambda) = \frac{1}{2} x^T Q x + c^T x + \lambda^T (Ax - b) \tag{40}$$

$$= \frac{1}{2} x^T Q x + (c + A^T \lambda)^T x - \lambda^T b \tag{41}$$

Rearranjando os termos, pegando a derivada $\mathfrak{L}(x, \lambda)$ rem relação a x e definindo como zero, temos

$$Qx + (c + A^T \lambda) = 0 \tag{42}$$

Como Q é positivo definido e isolando o x, temos

$$x = -Q^{-1} + (c + A^T \lambda) \tag{43}$$

Substituindo a equação anterior no Lagrangiano primal $\mathfrak{L}(x, \lambda)$, obtemos o Lagrangiano dual

$$\mathfrak{D}(\lambda) = -\frac{1}{2}(c + A^T\lambda)^T Q^{-1}(c + A^T\lambda)^T - \lambda^T b \tag{44}$$

Portanto o problema de otimização dual é dado por

$$\begin{aligned} \max_{x\in\mathbb{R}^m} \quad & -\frac{1}{2}(c + A^T\lambda)^T Q^{-1}(c + A^T\lambda)^T - \lambda^T b \\ \text{sujeito a} \quad & \lambda \geq 0 \end{aligned} \tag{45}$$

Exemplo Programação Quadrática

Considere o seguinte exemplo:

$$\begin{aligned} \min_{x\in\mathbb{R}^2} \quad & \frac{1}{2}x^T \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} x + \begin{pmatrix} -2 \\ -5 \end{pmatrix}^T x \\ \text{sujeito a} \quad & \begin{pmatrix} 1 & 2 \\ -1 & 2 \end{pmatrix} x \leq \begin{pmatrix} 2 \\ 2 \end{pmatrix} \end{aligned} \tag{46}$$

O Lagrangiano para este problema é:

$$\mathfrak{L}(x,\lambda) = \frac{1}{2}x^T \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} x + \begin{pmatrix} -2 \\ -5 \end{pmatrix}^T x + \lambda^T \left(\begin{pmatrix} 1 & 2 \\ -1 & 2 \end{pmatrix} x - \begin{pmatrix} 2 \\ 2 \end{pmatrix} \right) \tag{47}$$

Rearranjando os termos, temos:

$$\mathfrak{L}(x,\lambda) = \frac{1}{2}(2x_1^2 + 2x_2^2) + (-2 + \lambda_1 - \lambda_2)x_1 + (-5 + 2\lambda_1 + 2\lambda_2)x_2 - 2\lambda_1 - 2\lambda_2 \tag{48}$$

Derivando em relação a x e igualando a zero:

$$\frac{\partial \mathfrak{L}}{\partial x_1} = 2x_1 + (-2 + \lambda_1 - \lambda_2) = 0 \tag{49}$$

$$\frac{\partial \mathfrak{L}}{\partial x_2} = 2x_2 + (-5 + 2\lambda_1 + 2\lambda_2) = 0 \tag{50}$$

Isolando x:

$$x_1 = 1 - \frac{1}{2}\lambda_1 + \frac{1}{2}\lambda_2 \tag{51}$$

$$x_2 = \frac{5}{2} - \lambda_1 - \lambda_2 \tag{52}$$

Substituindo no Lagrangiano primal, obtemos o dual:

$$\mathfrak{D}(\lambda) = -\frac{1}{2}(c + A^T\lambda)^T Q^{-1}(c + A^T\lambda) - \lambda^T b \tag{53}$$

Portanto, o problema dual é:

$$\begin{aligned} \max_{\lambda \in \mathbb{R}^m} \quad & -\frac{1}{2}(c + A^T\lambda)^T Q^{-1}(c + A^T\lambda) - \lambda^T b \\ \text{sujeito a} \quad & \lambda \geq 0 \end{aligned} \tag{54}$$

7.5. Conclusão

Este capítulo apresenta importantes conceitos de *Otimização Contínua*, uma área de pesquisa ainda ativa e com aplicação direta em problemas de aprendizado de máquina.

Inicialmente, foi discutido em detalhes o Gradiente Descendente, um algoritmo amplamente utilizado no contexto otimização, bem como algumas de suas limitações como a sensibilidade à escolha da taxa de aprendizado, a convergência a mínimos locais, a necessidade de gradientes calculáveis, a sensibilidade à inicialização, entre outras. Entretanto, foram discutidas estratégias para lidar com alguns desses desafios como o Gradiente Descente com *Momentum* e a Descida do Gradiente Estocástica.

Em seguida, analisou-se problemas de otimização sujeitos a restrições e como resolver essa classe de problemas aplicando os Multiplicadores de Lagrange para tornar um problema restrito em um problema irrestrito. Ademais, notou-se como os conhecimentos de Álgebra Linear e Cálculo Multivariável são fundamentais na compreensão deste método.

Por conseguinte, trabalhou-se o conceito de dualidade no contexto de problemas de otimização, mais especificamente com o Problema Dual de Lagrange. Desta forma, verificou-se a possibilidade de estabelecer limites inferiores ou superiores para o valor ótimo do problema primal, por meio da solução do problema dual. Em outras palavras, constatou-se que a função dual, devido à dualidade fraca, pode ser vista como um limite inferior ou superior à função primal. Além disso, observou-se que, em alguns casos, resolver o problema dual pode ser mais fácil do que resolver o problema primal diretamente.

Posteriormente, apresentou-se o conceito de funções e conjuntos côncavos e convexos, bem como suas propriedades quando aplicados a problemas de otimização. Destaca-se a questão do ótimo global, uma vez que em um problema de otimização convexo, qualquer mínimo local é também um mínimo global, o que simplifica significativamente a busca pela solução ótima. Soma-se, ainda, o aspecto da eficiência computacional, pois algoritmos de programação convexa costumam convergir a uma solução ótima de maneira eficiente.

7.5.1. Material Disponibilizado

Por fim, informa-se que este grupo organizou o material do trabalho em um repositório no *GitHub*, que pode ser acessado pelo endereço `https://github.com/cmprenan/`

`TFC-UFG-Cap-7` e contém o seguintes documentos:

- **Relatório:** o presente relatório do trabalho em formato PDF.
- **Videoaula:** apresentação dos principais conceitos de *Otimização Contínua* em uma videoaula organizada em capítulos e disponível no *YouTube* pelo endereço `https://youtu.be/8r9uPzhuwIs`.
- **Slides:** arquivo de apresentação utilizado durante a videoaula em formato PDF.
- **Códigos:** os códigos apresentados ao longo da videoaula que foram desenvolvidos em *Python* utilizando *Notebooks* no *Google Colab* e encontram-se disponíveis para análise, teste e modificação.
- **Livro-texto:** hiperlink para acessar a última edição do livro *"Mathematics for Machine Learning"* de [Deisenroth et al. 2020], disponível de forma gratuita no endereço: `https://mml-book.github.io/book/mml-book.pdf`. O livro em questão é o livro-texto da disciplina de "Tópicos Especiais em Fundamentos de Computação – Matemática e Estatística para Ciência de Dados" ministrada pelo Prof. Dr. Rommel Melgaço Barbosa, no Programa de Pós-Graduação em Ciência da Computação, do Instituto de Informática, da Universidade Federal de Goiás, no semestre letivo 2024/1 e foi a base para o desenvolvimento deste trabalho.

Referências

[Deisenroth et al. 2020] Deisenroth, M. P., Faisal, A. A., and Ong, C. S. (2020). *Mathematics for Machine Learning*. Cambridge University Press.

[Rumelhart et al. 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). *Learning Representations by Back-Propagating Errors*. Nature.

[Toussain 2012] Toussain, M. (2012). Some Notes on Gradient Descent. `https://www.user.tu-berlin.de/mtoussai/notes/gradientDescent.pdf`.