



Descrição

Esta tarefa explora conceitos de cálculo vetorial, otimização e métodos de gradiente descendente no contexto de aprendizado de máquina, com foco em classificação multinomial.

1. Funções Convexas e Otimização

- Dado uma função $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$ faça três versões mudando a matriz \mathbf{A} para que a função seja:
 - Fortemente Convexa.
 - Convexa.
 - Não Convexa.

Evidencia cada uma das propriedades.

- Compare a progressão da otimização:
 - Implemente o método de gradiente descendente para cada função.
 - Visualize a progressão da otimização em cada caso, plotando o valor da função em relação ao número de iterações.
 - Opcional: Calcule a expressão do gradiente para cada função e analise como ele influencia a direção e a velocidade de convergência do algoritmo.

2. Classificação Multinomial e Gradiente Descendente

- Classificação Multinomial:
 - Utilize a base de dados **MNIST** (dígitos manuscritos) para classificação multiclasse.
 - Implemente um classificador multinomial utilizando **regressão logística multinomial** (também conhecida como *softmax regression*).
- Gradiente Descendente:
 - Derive o gradiente da função de custo da regressão logística multinomial.

- Implemente os seguintes métodos de otimização:
 - Gradiente Descendente (GD)
 - Gradiente Descendente Estocástico (SGD)
- Comparação e Análise:
 - Compare a progressão da otimização para GD e SGD, utilizando a função de custo exata e o gradiente automático (utilizando bibliotecas como TensorFlow ou PyTorch).
 - Avalie o desempenho dos métodos em termos de acurácia na classificação dos dígitos do MNIST.
 - Analise o impacto do tamanho do passo (*learning rate*) na convergência dos algoritmos.
 - ****Opcional:**** Explore outros métodos de otimização, como *mini-batch gradient descent* e algoritmos de otimização com momentum (ex: Adam).

3. Entregáveis

- Um notebook Jupyter (ou similar) contendo:
 - Implementações dos métodos de otimização e do classificador multinomial.
 - Visualizações da progressão da otimização para as três funções e para os métodos GD e SGD.
 - Resultados da classificação no MNIST, incluindo métricas de desempenho (acurácia).
 - Análise e discussão dos resultados, incluindo o impacto do tamanho do passo e a comparação entre os métodos.

4. Critérios de Avaliação

- Clareza do código e organização do notebook.
- Correção e eficiência das implementações.
- Qualidade da análise e discussão dos resultados, incluindo a interpretação das visualizações e a comparação entre os métodos.
- Profundidade da compreensão dos conceitos e técnicas exploradas (cálculo vetorial, otimização, gradiente descendente, classificação multinomial).

5. Observações

- Utilize bibliotecas como NumPy, Pandas e Matplotlib para manipulação de dados, cálculos numéricos e visualizações.
- Para a implementação do gradiente automático, utilize bibliotecas como TensorFlow ou PyTorch.
- Explore a documentação e tutoriais online para aprofundar seus conhecimentos sobre os métodos e as bibliotecas utilizadas.

Essa tarefa oferece uma oportunidade de aprofundar seus conhecimentos em conceitos fundamentais de aprendizado de máquina, combinando teoria e prática. Ao implementar os métodos de otimização e o classificador multinomial, você poderá visualizar a progressão da otimização, analisar o impacto de diferentes parâmetros e comparar o desempenho dos algoritmos.