Redes Neurais e Aprendizagem em Profundidade

Departamento de Computação - Universidade Federal de Ouro Preto - MG
Professores: **Eduardo Luz** e **Vander Freitas**

Trabalho Prático I (TP I) - 10 pontos, peso 10% da nota.

- Data de entrega: 08/10/2025 até 23:59. O que vale é o horário do *Google Drive*, e não do *seu*, ou do *meu* relógio!!!

- Clareza, identação e comentários no código também vão valer pontos. Por isso, escolha cuidadosamente o nome das variáveis e torne o código o mais legível possível.

- Entrevistas podem ser realizadas para complementar a avaliação.

- Códigos cuja autoria não seja do aluno, com alto nível de similaridade em relação a outros trabalhos, ou que não puder ser explicado, acarretará na perda da nota.

- Bom trabalho!

O Desafio da PyCab: Prevendo o Tempo no Trânsito de Nova York

Você é o(a) cientista de dados líder de uma startup inovadora, a *PyCab*. A missão da *PyCab* é revolucionar o transporte urbano na cidade de Nova York, oferecendo um serviço mais eficiente e transparente que os concorrentes. Para isso, é crucial fornecer aos usuários estimativas de tempo de viagem extremamente precisas. Previsões incorretas podem levar à frustração do cliente e à perda de confiança na plataforma.

Sua tarefa é construir um modelo de regressão capaz de prever a duração de uma corrida de táxi em Nova York, em segundos. Para isso, você utilizará o famoso dataset *NYC Taxi Trip Duration*[1], disponível no Kaggle, que contém informações detalhadas sobre milhões de viagens, incluindo horários, coordenadas de partida e chegada, e outras características. O sucesso do seu modelo impactará diretamente a viabilidade e o sucesso da *PyCab* no mercado competitivo.

Imposições e Requisitos Técnicos

O objetivo principal deste trabalho é introduzir os conceitos de tensores e computação em GPU, que são a base para o Deep Learning. Para isso, existem requisitos técnicos específicos:

- **Implementação com PyTorch:** A principal restrição deste trabalho é que o modelo de Machine Learning (Regressão Linear) **deve ser implementado do zero utilizando PyTorch**.
- É preciso deixar explícito e comentado, com as respectivas equações, quando possível, onde está sendo realizado o treinamento, qual a função de perda (por exemplo MSE), como exatamente é feita a atualização dos pesos (gradiente descendente), etc.
- **Uso Obrigatório de GPU:** O treinamento deve ser acelerado por GPU. Parte da avaliação será a comparação do tempo de treinamento entre CPU e GPU, demonstrando o ganho de performance. Você deve usar o ambiente **Google Colab**, que já fornece acesso a GPUs gratuitas.
- **Bibliotecas Auxiliares:** Bibliotecas como `pandas`, `numpy`, e `scikit-learn` são permitidas e incentivadas para as etapas de **pré-processamento**, **análise exploratória de dados (EDA)** e **avaliação de métricas** (e.g., `sklearn.metrics.mean_squared_error`). Implementação e treinamento do modelo devem ser realizados usando **torch**.

[1] https://www.kaggle.com/competitions/nyc-taxi-trip-duration/data

Estrutura do Trabalho e Critérios de Avaliação

O seu notebook `.ipynb` deve ser auto-contido e bem organizado, funcionando como um relatório técnico do seu projeto. Ele deve conter as seguintes seções, com explicações claras para cada passo:

1. **Identificação:** Nome completo do aluno e o resultado final da métrica principal (RMSE) no conjunto de teste.
2. **Análise Exploratória e Pré-processamento:** Carregamento dos dados, visualizações, tratamento de outliers e valores faltantes, e justificativa das suas decisões.
3. **Feature Engineering:** Criação de novas features a partir das existentes (e.g., cálculo de distância, extração de informações de data/hora) para melhorar o desempenho do modelo.
4. **Implementação do Modelo com PyTorch:** Definição do modelo de regressão linear, da função de perda (MSE) e da lógica do otimizador (gradiente descendente), tudo usando operações de tensores.
5. **Treinamento e Avaliação:**
   - Código do loop de treinamento que itera sobre os dados em mini-batches.
   - Medição e comparação do tempo de treinamento em CPU vs. GPU.
   - Gráfico da evolução da perda (loss) ao longo das épocas.
   - Cálculo da métrica final no conjunto de teste. A métrica de avaliação principal será a **Raiz do Erro Quadrático Médio (Root Mean Squared Error - RMSE)**.
6. **Conclusão:** Uma breve análise dos resultados, discutindo o desempenho do modelo, o impacto do *feature engineering* e a importância da aceleração por GPU.

O desempenho do seu modelo (RMSE) será um componente importante da nota. Alunos que demonstrarem um pré-processamento e feature engineering mais sofisticados, resultando em um RMSE menor, serão valorizados.

Como deve ser feita a entrega

Verifique se seu programa executa corretamente do início ao fim (deixe salvas as saídas de cada célula também) antes de efetuar a entrega. Quando estiver pronto, entregue via FORM (https://forms.gle/LdhELrn9iLuD4Dbr7), até 08/10/2025 até 23:59, um único arquivo **.ipynb** com o nome no formato `nome-sobrenome.ipynb`.

Esse arquivo deve conter: (i) seu nome e o RMSE obtido no conjunto de teste, (ii) todo o código utilizado, com as saídas visíveis, e (iii) o relatório integrado nas células de texto (Markdown), explicando a lógica por trás do seu código e das suas decisões.