

|   |  |
|---|--|
| Universidade Federal da Paraíba<br>Centro de Informática<br>Departamento de Informática | Aprendizado Profundo<br>Período 2025.1<br>Professor: Tiago Maritan |
|---|--|

**LISTA DE EXERCÍCIOS**  
**Data de Entrega: 02/09/2025**

**ORIENTAÇÕES:**

- A lista pode ser resolvida em grupo de até 3 integrantes.
- No dia da entrega da lista de exercícios, o(s) grupo(s) deverão(m) fazer uma apresentação para a turma de cerca de 15 a 20 min, e enviar um link com a sua resolução da lista (contendo códigos-fontes, resultados, etc) no formulário de envio de atividades próprio disponibilizado pelo professor no SIGAA.

1) Implemente uma rede perceptron de múltiplas camadas e utilize-a para aproximar as funções abaixo.

a)  $f(x) = \sin(2x) + \cos(3x)$ , onde  $0 \leq x \leq 5$

b)  $f(x) = 10x^5 + 5x^4 + 2x^3 - 0.5x^2 + 3x + 2$ , onde  $0 \leq x \leq 5$

Para cada função a ser aproximada, gere um conjunto de treinamento e um conjunto de testes. Nesses conjuntos, cada amostra deve ser representada da seguinte forma:  $x$  é a entrada e  $f(x)$  é a saída desejada - rótulo. Treine um perceptron de múltiplas camadas para que ele aprenda a aproximar a função a partir do conjunto de treinamento, e vá testando com o conjunto de validação.

Apresente os gráficos das funções reais vs. funções aproximadas e as curvas de erro de treinamento e validação.

2) O naufrágio do Titanic é um dos eventos mais emblemáticos do século XX. Embora a sobrevivência de um passageiro tivesse um elemento de acaso, estudos mostram que fatores como sexo, idade e classe socioeconômica influenciaram significativamente as chances de escapar. Implemente uma Rede Perceptron de Múltiplas Camadas que faça a predição se um passageiro sobreviveu ou não ao acidente do Titanic. Neste exercício, você deve tentar criar um modelo preditivo, usando os dados de passageiros (ou seja, nome, idade, sexo, classe socioeconômica, etc.), que procure responder à pergunta: "que tipo de pessoa tem mais probabilidade de sobreviver?" A base de dados do problema pode ser acessada através do seguinte link: <https://www.kaggle.com/c/titanic/data>.

Apresente também a curva do erro médio de treinamento com relação ao número de épocas e a curva do erro médio com o conjunto de validação ou teste. Calcule e apresente também a matriz de confusão do modelo.

Algumas recomendações:

- Explore a base de dados, identificando valores ausentes, inconsistências e outliers.
- Realize o preenchimento de dados faltantes usando técnicas adequadas  
Remova ou transforme atributos irrelevantes ou redundantes;
- Avalie e, se necessário, aplique normalização ou padronização das variáveis numéricas;
- Analise a correlação entre atributos e o rótulo, justificando suas escolhas.  
Caso exista desbalanceamento entre as classes, aplique técnicas de balanceamento.

3) Implemente e treine uma Rede Neural Convolutacional (CNN) para resolver o problema de classificação de objetos em imagens, utilizando a base de dados CIFAR-10, disponível em: <https://www.cs.toronto.edu/~kriz/cifar.html>.

Apresente a curva do erro médio ao longo do treinamento, bem como a matriz de confusão do modelo avaliado sobre o conjunto de testes.

4) (a) Implemente e treine um autoencoder utilizando o dataset Fashion-MNIST, disponível em: <https://www.kaggle.com/datasets/zalando-research/fashionmnist>. O objetivo deste exercício é compreender como esse tipo de rede neural é capaz de aprender representações comprimidas de dados e realizar reconstruções a partir dessas representações. O autoencoder pode ser construído utilizando apenas camadas densas (fully connected) ou, alternativamente, camadas convolucionais e deconvolucionais.

Após o treinamento, selecione 10 imagens do conjunto de teste e utilize o modelo treinado para gerar as reconstruções correspondentes. Apresente os resultados visualmente, exibindo lado a lado as imagens originais e suas respectivas reconstruções, de modo que seja possível avaliar qualitativamente o desempenho do autoencoder. Apresente também a curva do erro médio ao longo do treinamento.

(b) Complete o exercício da questão 3(a) adicionando ruído aleatório às imagens de entrada, com o objetivo de treinar um *denoising autoencoder*. Após o treinamento, utilize as 10 imagens selecionadas na questão anterior (com o ruído aplicado) e apresente os resultados visualmente, exibindo lado a lado as imagens originais e suas respectivas reconstruções. Avalie qualitativamente o desempenho do modelo em remover o ruído e preservar as características das imagens originais.

5) Implemente uma Rede Neural Recorrente (RNN), como por exemplo uma LSTM ou GRU, para o problema de Análise de Sentimentos em postagens do Twitter. O conjunto de dados **Twitter Sentiment Analysis Dataset** contém frases curtas extraídas do Twitter, rotuladas como positivas ou negativas. Essa base de dados está disponível em:

<https://www.kaggle.com/datasets/kazanova/sentiment140>

Descreva o processo de pré-processamento dos dados, a arquitetura da RNN implementada, os parâmetros utilizados no treinamento e os principais resultados obtidos como acurácia e curva de erro. Apresente também 5 (cinco) exemplos de postagens do conjunto de teste, acompanhados da predição do modelo e do rótulo verdadeiro.

Algumas recomendações:

- **Pré-processamento dos dados**

- A base original contém **1,6 milhão de tweets**. Para este exercício, use apenas uma **amostra menor** (ex.: 50 mil registros) para reduzir o tempo de treinamento.
- Remova ou substitua URLs, menções (**@usuario**), hashtags e caracteres especiais.
- Converta todo o texto para minúsculas.
- Aplique *padding* para que todas as sequências tenham o mesmo tamanho.
- Opcional: aplique *stemming* ou *lemmatization* para reduzir variações de palavras.

- **Implementação da Rede**

- Implemente uma RNN usando **LSTM** ou **GRU**.
- Defina o tamanho do vocabulário, a dimensão dos embeddings;
- Opcional: experimente **embeddings pré-treinados** (ex.: GloVe) para comparar resultados.

Bom trabalho!!!!