

**IA353 – Redes Neurais (1s2024)**  
**Projeto Computacional – Parte 2 – PCs #06 a #12**  
**Atividade em Dupla – Peso 7**  
**Data de entrega dos resultados solicitados: 11/06/2024**

**1 Projeto Computacional 06: Resolvendo o problema do caixeiro viajante com uma rede neural de Hopfield**

Mesmo cientes de que existem, neste momento de evolução tecnológica, estratégias mais competentes para resolver o problema do caixeiro viajante, como o Concorde TSP Solver [ <https://www.math.uwaterloo.ca/tsp/concorde.html> ], execute o notebook [PC06\_HN\_TSP.ipynb] para obter soluções de caixeiro viajante para ao menos duas instâncias dentre as disponíveis, apresentando o melhor resultado em cada caso.

- (a) O desempenho da técnica é sensível à escolha dos hiperparâmetros A, B, C e D, sendo que não foi feito um esforço tão grande em sua sintonia, razão pela qual não estamos considerando, nesta atividade específica, casos de estudo com mais de 10 cidades. Qual é o papel desempenhado por esses hiperparâmetros?
- (b) Por que boa parte das soluções não atende as condições de contorno do problema, como passar por todas as cidades uma única vez?
- (c) Por que, mesmo com o atendimento de todas as condições de contorno em algumas soluções candidatas, produzindo, assim, uma solução válida para o problema, nem sempre o percurso é minimizado?

**2 Projeto Computacional 07: Resolvendo o problema do caixeiro viajante com um mapa de Kohonen**

Com a mesma ressalva feita no enunciado do PC06, execute o notebook [PC07\_SOM\_TSP.ipynb] para obter soluções de caixeiro viajante empregando um mapa de Kohonen. Para o primeiro caso de estudo, é necessário carregar o arquivo [qa194.tsp] fornecido pelo professor. Você é livre para escolher o segundo caso de estudo, dentre as opções em:

<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/>

Nesta atividade, as instâncias podem ser bem maiores que aquelas do PC06.

- (a) A solução adota um mapa auto-organizável unidimensional. Repare que a proposta difere daquela apresentada em aula. Na proposta apresentada em aula, há inserção e remoção de neurônios ao longo do processo de auto-organização, o que não ocorre aqui. Descreva o princípio de operação da técnica, que opera como “um anel elástico que vai se expandindo e aderindo às coordenadas das cidades”. Para tanto, consulte o material em: [ <https://diego.codes/post/som-tsp/> ].
- (b) Sabemos, então, que existem pelo menos três hardwares diferentes para resolver o problema do caixeiro viajante: (1) O computador digital padrão von Neumann; (2) Uma rede neural de Hopfield; e (3) Um mapa auto-organizável unidimensional em anel. Como a solução para o TSP é concebida em cada um desses hardwares?
- (c) Sabendo que a dificuldade para se resolver um problema depende da forma como o problema é formulado e do hardware a ser empregado na busca de sua solução, aponte as possíveis consequências práticas de se dispor de hardwares alternativos para resolver o TSP.

### 3 Projeto Computacional 08: Comparação entre modelos de predição de séries temporais

A predição de séries temporais pode ser implementada a partir de um grande número de técnicas, com tratamentos específicos para a série temporal e com objetivos variados. Nesta atividade, vamos empregar redes neurais recorrentes (SimpleRNN e LSTM), não-recorrentes (MLP) e um modelo de regressão linear na predição de duas séries temporais: Sunspot (média mensal de 1749 até 1983) e Unemployment rate (Fonte: U.S. Bureau of Labor Statistics) (de 01/1948 até 03/2024). As predições podem ser de um passo à frente ou de múltiplos passos à frente. Para efeito de comparação, entre as abordagens recorrentes e não-recorrentes, o número de desdobramentos no tempo e o número de atrasos (*tapped delay line*) serão os mesmos. As séries temporais sofrem apenas um escalamento, embora outros tratamentos, como eliminação de tendências, pudessem ser considerados. Não cabe estender os resultados das análises comparativas a outras séries temporais, pois cada série tem suas particularidades e vai desafiar os modelos de aprendizado de formas distintas. Por exemplo, a série Unemployment rate tem um evento anômalo nos dados de teste, pois esses incluem o efeito do lockdown da pandemia causada pelo coronavírus SARS-CoV-2. Já a série Sunspot é bem-comportada, permitindo um bom desempenho de modelos de predição lineares. Execute o notebook [PC08\_RNNs\_MLP\_LP\_TS.ipynb] para os dois casos de estudo: Sunspot e Unemployment rate. Apresente, assim, dois notebooks em seu relatório. Responda as 5 questões das células finais em apenas um dos dois notebooks, mesmo que envolva resultados dos dois casos de estudo.

### 4 Projeto Computacional 09: Captura de manifolds 2D da base MNIST empregando autoencoders e uma técnica de desemaranhamento

Tomando a base de dados MNIST, treinar um *autoencoder* com camadas densas e constatar o sucesso da tarefa de codificação e de decodificação (mesmo que ainda com algumas imperfeições), indicando que duas variáveis latentes podem ser suficientes para codificar a essência de toda a variedade exibida pelas versões manuscritas de 10 dígitos. Por fim, explorar o *manifold* gerado no gargalo do *autoencoder*. Para tanto, execute o notebook [PC09\_P1\_AE\_MNIST.ipynb] e procure compreender o que está sendo feito em cada trecho de código (não é preciso reportar esta atividade). Para as células [Moving along a circle], [Moving along an ellipse] e [Moving along a grid], procure ajustar o intervalo de excursão para melhor explorar a “nuvem de pontos” 2D. Em seguida, execute o notebook [PC09\_P2\_AE\_MNIST.ipynb], que implementa o *autoencoder* utilizando camadas convolucionais, com *downsampling* no codificador e *upsampling* no decodificador, realizando as 4 atividades requeridas.

### 5 Projeto Computacional 10: Generative Adversarial Networks (GANs)

O treinamento de redes neurais adversárias foi proposto em 2014:

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, **Generative adversarial nets**, Advances in Neural Information Processing Systems, pp. 2672-2680, 2014.

no contexto de síntese de máquinas generativas. As GANs conseguem conduzir o processo de treinamento a desempenhos nunca antes vislumbrados em aprendizado de máquina. Evidentemente, a exploração de todo este potencial requer muita memória e muito processamento, algo que não será exigido neste curso. Busca-se a execução completa do treinamento de uma GAN para reproduzir dígitos manuscritos, a partir da base MNIST. Para isso, vamos utilizar a biblioteca Keras-GAN disponível em:

<https://github.com/eriklindernoren/Keras-GAN>

O objetivo é conseguir executar o exemplo disponível em:

<https://github.com/eriklindernoren/Keras-GAN/blob/master/gan/gan.py>

que já foi organizado no notebook [PC10\_P1\_GAN\_MNIST.ipynb]. O código executa o treinamento por 30.000 épocas e salva exemplos de imagens a cada 1000 épocas. Execute o código e exiba: a primeira imagem obtida, e imagens obtidas após 1000, 10000, 20000 e 30000 épocas. Por fim, adapte o código ou use algum outro código capaz de treinar uma máquina generativa para as imagens de **uma das classes** da base CIFAR-10, apresentando os resultados de forma similar ao caso da base MNIST. Como material de apoio, considere:

<https://machinelearningmastery.com/how-to-develop-a-generative-adversarial-network-for-a-cifar-10-small-object-photographs-from-scratch/>

Repare que o desempenho do discriminador cai e estabiliza em um valor baixo ao longo das épocas de treinamento, conforme esperado.

## 6 Projeto Computacional 11: Processamento de Linguagem Natural

Nesta atividade, vamos implementar um Encoder-Decoder LSTM para executar uma tarefa análoga à tradução de frases realizada por vários aplicativos da internet, mas envolvendo bem menos “palavras” das duas linguagens. Trata-se de um preditor de uma sequência de saída a partir de uma sequência de entrada (mapeamento seq2seq). Execute o notebook [PC11\_PLN\_LSTM.ipynb] e explique como são gerados os dados de treinamento. Em seguida, explique como uma calculadora simples pode operar baseada no conceito de tradução de frases, ou seja, sem realizar operações algébricas. Este notebook está baseado no material presente em [Brownlee, J. “Long Short-Term Memory Networks With Python – Develop Sequence Prediction Models With Deep Learning”, Machine Learning Mastery, 2017], mais especificamente no [Chapter 9].

## 7 Projeto Computacional 12: Aprendizado por reforço (Deep Q-Learning) para percursos em labirintos

Estude o notebook [PC12\_RL\_Maze.ipynb], procurando compreender o que está sendo feito em cada trecho de código (não é preciso reportar esta atividade). Para tanto, é esperado que o (a) aluno(a) acompanhe atentamente as explicações fornecidas em <https://www.samyzaf.com/ML/rl/qmaze.html>. Em seguida, execute o notebook para o labirinto  $5 \times 5$  apresentado (labirintos maiores tornam o aprendizado mais demorado, mas poderiam ser considerados), apresentando os resultados do treinamento até a convergência. A estimativa de tempo de execução está em torno de 1h15. Por fim, responda o que se pede nas 3 atividades das células finais do notebook.