

	Universidade Federal de Goiás – Instituto de Informática Disciplina: INF 0092 - INTELIGÊNCIA COMPUTACIONAL Docente: Prof. Me. Bruno Brandão	
	Trabalho (Redes Neurais Artificiais)	13/05/2025

Orientações Gerais:

1. O trabalho deve ser entregue na data combinada em sala. Atrasos serão aceitos mediante apresentação de justificativa plausível. Ademais, atrasos terão dedução de nota.
2. Não é permitido plagiar código ou relatório de outros grupos. Quaisquer suspeitas de plágio serão confrontadas com os grupos podendo chegar ao conhecimento da coordenação.
3. A entrega deverá ser feita pelo SIGAA em um único arquivo comprimido por APENAS UM integrante do grupo.

1. Objetivos e Organização

1. O trabalho tem como objetivo colocar em prática os conhecimentos adquiridos em sala de aula sobre Redes Neurais Artificiais.
2. O trabalho poderá ser feito em grupo ou individualmente, com um máximo de 3 integrantes por grupo.
3. O trabalho terá dois entregáveis, o código fonte em python e o relatório técnico detalhado de experimentos.
4. Os grupos **devem** utilizar RNAs para resolver o problema, não podendo utilizar de outros métodos de aproximação.
 - a. Outros métodos de Aprendizado de Máquina podem ser utilizados como auxiliares, por exemplo, para extração de características.

2. Problema

1. O problema a ser resolvido é o desafio “House Prices - Advanced Regression Techniques” disponibilizado na plataforma Kaggle através do link:
<https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/overview>
Ele consiste na predição de preços de casas a partir de diversos fatores.
2. Na aba “Data” é possível encontrar os links para download dos dados de treino e teste bem como uma breve descrição de cada coluna.
3. A coluna alvo é o preço da casa, denominada SalePrice.
 - a. O alvo de predição não está presente no conjunto de teste, isso porque as predições de um modelo podem ser submetidas na plataforma e avaliadas em segredo para evitar treino no conjunto de avaliação. Sendo assim, é recomendado que o conjunto de treino seja dividido em treino e teste para experimentos locais.

3. Sobre o Código

1. O código poderá ser entregue em formato de arquivos .py com README.md explicativo ou Jupyter Notebook.
2. Tanto o README quanto o Notebook devem conter instruções de execução bem como uma lista de bibliotecas usadas.
3. A única biblioteca de autograd permitida é o Pytorch.
4. Deixe o código modularizado contendo de forma clara e explícita:
 - a. O carregamento dos dados, quaisquer trabalhos de normalização, one hot encoding, etc.

- b. O loop de treino, testes, e cálculo da loss.
 - c. A forma como o dataset é carregado em batches.
5. Deixe a escolha de hiperparâmetros fácil e destacada no código, por exemplo, argumentos de execução dos scripts, usando um arquivo .yaml de configuração, ou uma célula do notebook separada apenas para a definição dos mesmos.
 6. A seed de números aleatórios deve ser explicitamente configurada para todas as bibliotecas no início do código para garantir reprodutibilidade.
 7. É permitido entregar um código que execute mais de um experimento por vez, contudo, os hiperparâmetros do melhor experimento ainda devem estar descritos no relatório para reprodução.
 8. O treinamento do melhor modelo não deve durar mais que 1h (com uma margem de erro aceitável de 10 minutos)

4. Sobre o Relatório

1. O relatório deverá ser entregue em um único documento pdf.
2. O relatório deverá conter as seguintes seções:
 - a. **Dados**
 - i. Descrever quais colunas foram usadas ou não, e porque. Não é necessário discorrer sobre todas as colunas usadas, afinal mais dados significam mais informação. Porém, nem todas são úteis. Ao rejeitar algumas colunas, discorre sobre qual foi o critério utilizado para sua remoção.
 - ii. Descrever como foram definidos os outliers e sua remoção, aqui é possível agrupar colunas por métodos de normalização, distribuição de dados, escala etc. Porém, é preciso demonstrar sua justificativa, por exemplo, histogramas, máximo e mínimo de um conjunto, etc. O leitor não deve simplesmente acreditar no que é dito sobre os dados.
 - iii. Descrever como foi feita a normalização dos dados, se possível, demonstrar a distribuição antes e depois da normalização. Outra possível sugestão aqui é agrupar vários gráficos, mesmo que pequenos, para diversas colunas.
 - iv. Descrever também como foram tratadas colunas categóricas.
 - b. **Modelo**
 - i. Descrever como foi a definição do modelo, tamanho da rede, número de camadas, função de ativação (com justificativa), inicialização de pesos (com justificativa), quantidade de parâmetros (pesos).
 - c. **Experimentos**
 - i. Os experimentos devem demonstrar sua trajetória científica sobre o problema. Dificilmente você terá testado uma única configuração, portanto, demonstre aqui curvas de loss e erros que justifiquem mudar algum hiperparâmetro do modelo e registre os resultados comparativos que comprovem se a mudança foi positiva ou negativa.
 - ii. Experimentos mínimos: no mínimo o relatório deve ter experimentos com dois otimizadores, onde cada otimizador possui uma curva com média e desvio padrão de diferentes taxas de aprendizado. Avaliações de outros aspectos após a escolha do otimizador (como tamanho de rede, normalização, colunas do dataset, função de ativação, etc.) e outros gráficos com suas devidas interpretações são muito bem vindos e aumentarão a nota.
 - iii. Lembre-se que testes comparativos fixam todos os hiperparâmetros e alteram apenas um. Assim, é possível ver o seu efeito no aprendizado.
 - d. **Conclusões**

- i. Discorra sobre os resultados dos experimentos relacionados ao hiperparâmetros ou aos dados. Algumas sugestões de perguntas a se responder são:
 1. Quais as colunas mais importantes para a decisão do modelo?
 2. Os exemplos em que ele erra mais têm algo em comum?
 3. Qual o efeito de um hiperparâmetro específico no aprendizado?
 4. Alguma normalização é mais, ou menos, eficiente?

5. Competição (Bônus)

1. Uma vez que o Kaggle permite a submissão de predições e sua avaliação de forma secreta, os times poderão submeter suas predições do conjunto de testes na plataforma e observar suas posições no ranking.
2. Para que um time esteja competindo oficialmente, envie o nome de usuário (apenas um por time) que realizará submissões ao professor. Estes serão públicos para a turma com o ranking atualizado periodicamente.
3. Ao final da entrega do trabalho, os times que ficarem em primeiro e segundo lugar dentre os grupos da disciplina poderão escolher um prêmio dentre os seguintes sem repetição, com time em primeiro escolhendo primeiro:
 - a. 1,0 pontos extra na avaliação de RNAs (aplicável a cada integrante, a nota não passará de 10,0)
 - b. Escolher o tema de uma aula (data a ser agendada no calendário).
 - c. Escolher o problema do próximo trabalho (haverá uma pré-seleção para garantir alinhamento com o tema e possibilidade de solução).
 - d. Possibilidade de realização da última prova em duplas (alunos podem escolher fazer sozinhos).
4. Em caso de empate, o vencedor será sorteado entre os grupos com mesmo desempenho.