

Implementação Redes Neurais Artificiais

1. Definição do Trabalho

Este trabalho deverá ser realizado em equipes de 2 (dois) a 4 (quatro) alunos e valerá 100 pontos para a 2ª avaliação da disciplina e consiste na modelagem e implementação de um sistema utilizando Redes Neurais Artificiais para a resolução de um problema à escolha da equipe.

Abaixo, uma lista de modelos de RNAs disponíveis:

- MLP (*Multilayer Perceptron*)
- Redes Recorrentes (p. ex. Rede de Hopfield ou Máquinas de Boltzmann)
- BAM (*Bi-directional Associative Memory*)
- Adaline/Madaline
- Redes auto-organizáveis (p. ex. Mapas de Kohonen)
- Redes Convolucionais
- *Long Short-Term Memory* (LSTM)
- *Deep Auto-Encoders*
- *Generative Adversarial Networks* (GANs)
- Outros modelos...

Cada equipe deverá escolher um domínio de conhecimento diferente e um modelo de RNA. Caso duas ou mais equipes escolham o mesmo domínio, este ficará com a primeira equipe a defini-lo e as demais deverão escolher outro domínio.

Neste trabalho, as equipes podem escolher o mesmo modelo de RNA. Caso a equipe queira implementar outro modelo que não esteja na lista, deve procurar a professora antes da escolha.

Sugestões de implementação:

- Páginas com bases de dados que podem ser resolvidas com RNAs:
 - <http://archive.ics.uci.edu/ml/index.php>
 - <https://www.kaggle.com/datasets>
- Outras bases de dados públicas, que possuam dados para serem usados com RNAs.

2. Datas de Entrega

Dias 09 e 16/09/2024, no horário da aula: discussão com as equipes sobre os problemas pretendidos.

Até dia 20/09/2024, 17hs: Constituição da equipe e escolha do modelo e do problema a ser resolvido. Essa escolha deverá ser feita no documento `RNAs.xlsx`, disponível na pasta Arquivos, na plataforma Teams. Não preencher a coluna “Data de Apresentação”. Equipes que não respeitarem esse prazo terão suas notas descontadas.

Até 27/10/2024, 23hs: Entrega, via plataforma Teams, do trabalho. Não serão aceitos trabalhos entregues em outros meios, nem trabalhos atrasados. A implementação (parte do treinamento e a aplicação, conforme itens 3.1 e 3.2 abaixo), a base de dados, o manual do usuário e o artigo devem ser compactados em um arquivo .zip, nomeado com o nome dos integrantes da equipe. Basta que apenas um dos integrantes submeta o trabalho.

Dias 28/10 e 01/11/2024, no horário da aula: Apresentação, pela equipe, do sistema desenvolvido, durante o horário da aula. A avaliação da apresentação fará parte da nota e será individual. Os alunos que não participarem da apresentação ou não demonstrarem conhecimento no assunto, receberão nota 0 (zero) no item “apresentação”. Caso alguma equipe ou aluno não apresente o trabalho, no dia definido, a equipe/aluno irá receber nota 0 (zero) no trabalho.

3. O que deve ser entregue

3.1. Implementação – Treinamento e ajustes

O trabalho pode ser implementado em C/C++, Java ou Python. Todos os códigos devem estar bem comentados e com o nome dos integrantes da equipe. Devem ser entregues todos os códigos-fontes da implementação e todos os arquivos necessários à compilação, para a parte do treinamento da rede.

A implementação deve executar nas máquinas disponíveis nos laboratórios da instituição. Será separada uma máquina, no Laboratório de Robótica Inteligente (LRI – Sala 12, bloco F), onde todas as implementações devem ser instaladas para poderem ser corrigidas. É uma máquina (máquina 4, ao entrar no laboratório mesa à esquerda) com Windows e Linux, com poderes de administrador, para que vocês possam instalar o que for necessário. O usuário e senha de cada SO estão descritos no próprio gabinete. Cada equipe deve criar uma pasta com o os arquivos do trabalho e instalar os softwares. Pegar a chave com a professora ou com o laboratorista.

É possível que a equipe utilize o Google Colab (<https://colab.research.google.com>). Neste caso, a equipe precisa enviar o notebook da implementação e demais arquivos utilizados, e a correção acontecerá diretamente no Colab.

A base de dados usada deve ser entregue completa. No código entregue deve estar disponível a separação da base em treino/teste/validação. Incluir um arquivo (pode ser .txt) com as instruções para a realização do treinamento.

Como trata-se da implementação de uma RNA, vários hiperparâmetros influenciam no desempenho da rede. Esses hiperparâmetros incluem, mas não são os únicos:

- Número de camadas ocultas;
- Número de neurônios em cada camada oculta;
- Funções de ativação;
- Algoritmos de treinamento;
- Otimizadores;
- Entre outros.

Para esse trabalho, as equipes devem realizar uma busca (não precisa ser exaustiva, mas deve ser feita) desses e outros hiperparâmetros (a escolha de qual hiperparâmetro deve ser utilizado vai depender da RNA escolhida). O código para essa escolha também deve ser entregue (para quem for utilizar as ferramentas do Python, há uma função específica para isso, a GridSearchCV¹, que pode ser usada para facilitar essa busca. Para outras linguagens, verificar a existência de ferramentas próprias, caso contrário, implementar um código de busca).

¹ https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

O modelo treinado e otimizado deve ser salvo para ser usado na próxima etapa: aplicação prática.

3.2. Aplicação prática

Após o treinamento, a equipe deve salvar o modelo para ser usado nesta etapa, que envolve a criação de uma aplicação prática que utilize o modelo de RNA treinado. Esta aplicação pode ser desenvolvida em ambiente *Web* ou *mobile* e deve ser voltada para um usuário final, ou seja, alguém sem conhecimentos técnicos em RNAs. O objetivo é mostrar que a equipe tem capacidade não apenas de treinar e otimizar um modelo de RNA, mas também de aplicá-lo em um contexto real, criando uma solução prática e acessível para o usuário final.

A aplicação deve integrar o modelo treinado de forma intuitiva, permitindo que o usuário final possa interagir com ela e utilizar os resultados do modelo de maneira simples e eficaz. Por exemplo, a aplicação poderia permitir que o usuário insira dados e receba previsões ou classificações geradas pelo modelo, tudo isso com uma interface amigável e de fácil uso.

Além disso, espera-se que a aplicação seja bem documentada, com instruções claras sobre seu funcionamento, e que seja funcional, sem erros que comprometam a experiência do usuário. A interface deve ser clara e objetiva, com uma navegação intuitiva que não exija conhecimentos técnicos avançados.

A aplicação também deve executar na máquina do laboratório, se a aplicação for *Web*. Caso seja uma aplicação *mobile*, a equipe deverá agendar com a professora a apresentação do aplicativo.

3.3. Manual do usuário

Como parte do projeto, a equipe deverá desenvolver um manual do usuário que acompanhe a aplicação final. Este manual tem como objetivo orientar o usuário final sobre o uso da aplicação, fornecendo instruções claras e acessíveis para que ele possa tirar o máximo proveito das funcionalidades oferecidas. O manual do usuário deve incluir as seguintes seções:

1. **Introdução:** uma breve descrição da aplicação, seu propósito, e como ela pode beneficiar o usuário final.
2. **Instalação e configuração:** instruções passo a passo sobre como instalar e configurar a aplicação, tanto em ambiente *Web* quanto *mobile*. Deve-se incluir informações sobre requisitos do sistema, procedimentos de instalação, e configurações iniciais necessárias.
3. **Navegação pela interface:** explicação das principais áreas da interface da aplicação, com descrições de cada menu, botão e funcionalidade disponível. Capturas de tela podem ser incluídas para facilitar a compreensão.
4. **Uso das funcionalidades:** detalhes sobre como o usuário pode utilizar as funcionalidades principais da aplicação, como inserir dados, realizar consultas, interpretar os resultados gerados pelo modelo de RNA, e outras ações importantes.

O manual deve ser escrito de forma clara e concisa, evitando jargões técnicos e utilizando uma linguagem acessível ao público geral. O objetivo é garantir que qualquer pessoa, independentemente de seu nível de conhecimento técnico, possa utilizar a aplicação com facilidade e confiança.

O manual deve ser incluído no arquivo `.zip` que a equipe irá submeter ao Teams, e deverá estar em formato `.pdf`.

3.4. Artigo Científico

A equipe deverá desenvolver um artigo científico, em LaTeX, disponível no Overleaf, no endereço: <https://www.overleaf.com/latex/templates/sbc-conferences-template/blbxwjwzdng>, que deverá descrever, pelo menos:

- Problema resolvido, incluindo se é classificação, regressão ou outro, e quais os objetivos da aplicação implementada.
- Arquitetura utilizada (deve contemplar, no mínimo, sua estrutura e seu modo de funcionamento)
- Algoritmo de treinamento utilizado
- Parâmetros utilizados na solução desenvolvida, entre eles:
 - Variáveis de entrada e de saída
 - Quantas camadas ocultas
 - Quantos neurônios em cada camada
 - Codificação dos dados de entrada e das saídas da rede
 - Função (ou funções) de ativação utilizada
 - Taxa de aprendizado
 - Otimizadores (se utilizado)
 - Demais parâmetros de configuração dependentes da arquitetura escolhida
- Resultados e conclusões: deve conter os resultados que a rede alcançou na sua tarefa de classificação/regressão, qual o percentual de acertos e de erros, e quais as tentativas que foram realizadas para acertar a estrutura e calibrar os parâmetros para que a rede tivesse o melhor desempenho. Caso seja um problema de classificação, deve apresentar também a matriz de confusão da rede implementada.
- Referências bibliográficas, conforme o modelo disponibilizado.

O artigo deve ser incluído no arquivo .zip que a equipe irá submeter ao Teams, e deverá estar em formato .pdf.

4. Sobre a apresentação

Nos dias 28/10 e 01/11/2024, no horário da aula, cada equipe terá até 10 minutos para apresentar o problema, a arquitetura escolhida, hiperparâmetros utilizados, o processo de treinamento e ajustes e a aplicação final, além das análises do processo. Não é necessário descrever a teoria das RNAs. A equipe deve focar no problema, objetivos e solução desenvolvida.