

Trabalho de redes neurais com uso de interfaces gráficas

Prof. José Alfredo F. Costa

Julho de 2024

Título do trabalho: "Desenvolvimento de um Sistema Interativo de Classificação com Redes Neurais: Uma Abordagem Prática com Interface Gráfica"

Resumo: Este projeto visa o desenvolvimento de uma aplicação de software com interface gráfica para treinamento, avaliação e análise de redes neurais em tarefas de classificação. Os alunos, trabalhando em duplas, irão implementar um sistema que permite a configuração flexível de redes neurais, visualização em tempo real do processo de treinamento, e análise comparativa de diferentes arquiteturas e hiperparâmetros. Utilizando bases de dados do UCI Machine Learning Repository e diversas bibliotecas de interface gráfica, o projeto integra conceitos de aprendizado de máquina, desenvolvimento de software e interação homem-computador. O objetivo é proporcionar uma experiência prática abrangente no design e implementação de sistemas de inteligência artificial interativos.

Palavras-chave: Redes Neurais, Classificação, Interface Gráfica, Python, Aprendizado de Máquina, Visualização de Dados, Interação Homem-Computador

Objetivos de Aprendizagem:

1. Aplicar conhecimentos teóricos de redes neurais em um contexto prático
2. Desenvolver habilidades em programação Python e criação de interfaces gráficas
3. Aprimorar competências em análise e visualização de dados
4. Fomentar o pensamento crítico na avaliação de modelos de aprendizado de máquina
5. Desenvolver habilidades de trabalho em equipe e comunicação técnica

Estrutura do Projeto:

1. Planejamento e Design
2. Implementação do Backend (Rede Neural)
3. Desenvolvimento da Interface Gráfica
4. Integração e Testes
5. Análise de Resultados
6. Documentação e Apresentação

Critérios de Avaliação:

1. Funcionalidade e robustez da aplicação (30%)
2. Qualidade e usabilidade da interface gráfica (20%)
3. Análise e interpretação dos resultados (20%)

4. Código-fonte e documentação (15%)
5. Apresentação oral e relatório final (15%)

Cronograma:

- Semana 1: Planejamento e design inicial (início em 3/7/24)
- Semana 2: Implementação do backend e interface gráfica
- Semana 3: Integração, testes e refinamento
- Semana 4: Análise de resultados e preparação da documentação
- Semana 5: Apresentações finais (4 duplas apresentarão já em 31/7/24)

Recursos Necessários:

- Computadores com Python instalado
- Bibliotecas: TensorFlow/Keras, Scikit-learn, Matplotlib, Pandas
- Bibliotecas de interface gráfica (específicas para cada dupla)
- Acesso às bases de dados do UCI Machine Learning Repository
- Ambiente de desenvolvimento integrado (IDE) de preferência
- Conta no GitHub para hospedagem do código

Impacto Esperado: Este projeto proporcionará aos alunos uma experiência prática valiosa no desenvolvimento de sistemas de IA interativos, preparando-os para desafios reais na indústria e pesquisa. Além disso, fomentará habilidades essenciais como pensamento crítico, resolução de problemas e comunicação técnica.

Painel Geral para Todos os Trabalhos:

1. Seleção da base de dados (dropdown ou lista)
2. Configuração da rede neural:
 - Número de camadas ocultas (slider ou input numérico)
 - Neurônios por camada (input numérico para cada camada)
 - Função de ativação (dropdown: 'relu', 'tanh', 'sigmoid')
 - Algoritmo de otimização (dropdown: 'adam', 'sgd', 'rmsprop')
3. Parâmetros de treinamento:
 - Taxa de aprendizagem (input numérico)
 - Número de épocas (input numérico)
 - Tamanho do batch (input numérico)
4. Divisão dos dados:
 - Proporção treino/teste (slider: 50/50, 60/40, 70/30, 80/20)
5. Botões:
 - "Carregar Dados"
 - "Treinar"
 - "Testar"
 - "Gerar Relatório"
6. Áreas de saída:
 - Progresso do treinamento (barra de progresso + texto)
 - Métricas de desempenho (acurácia, precisão, recall, F1-score)
 - Gráfico de evolução do treinamento (perda e acurácia por época)

- Matriz de confusão (após teste)
- Área para comparação de experimentos

Enunciados específicos para cada dupla:

1. Dupla 1: Base de dados: Wine Dataset Biblioteca: Tkinter
 - Enunciado: "Implemente o sistema de classificação para o Wine Dataset usando Tkinter como biblioteca de interface gráfica. O Wine Dataset contém 178 amostras de vinhos com 13 atributos químicos, classificados em 3 tipos.
 - Foque em criar uma interface intuitiva para manipulação dos parâmetros da rede neural e visualização dos resultados."
2. Dupla 2: Base de dados: Iris Dataset Biblioteca: PyQt5
 - Enunciado: "Desenvolva o sistema de classificação para o Iris Dataset utilizando PyQt5 para a interface gráfica. O Iris Dataset inclui 150 amostras de 3 espécies de íris, cada uma com 4 características.
 - Explore os recursos avançados do PyQt5 para criar gráficos interativos dos resultados."
3. Dupla 3: Base de dados: Breast Cancer Wisconsin Dataset Biblioteca: MESOP ou wxPython
 - Enunciado: "Crie o sistema de classificação para o Breast Cancer Wisconsin Dataset com wxPython como biblioteca de interface. Este dataset contém 569 amostras com 30 características, classificadas como benignas ou malignas.
 - Foque em criar uma interface responsiva e amigável para análise dos resultados."
4. Dupla 4: Base de dados: Digits Dataset Biblioteca: Kivy
 - Enunciado: "Implemente o sistema de classificação para o Digits Dataset usando Kivy para a interface gráfica. O Digits Dataset contém 1797 imagens de dígitos manuscritos (0-9).
 - Aproveite as capacidades de touch do Kivy para criar uma interface interativa de visualização dos dígitos e resultados."
5. Dupla 5: Base de dados: Heart Disease Dataset Biblioteca: MESOP ou PyGObject (GTK)
 - Enunciado: "Desenvolva o sistema de classificação para o Heart Disease Dataset utilizando PyGObject (GTK) para a interface. Este dataset inclui 303 amostras com 13 atributos, classificando a presença ou ausência de doença cardíaca.
 - Crie uma interface robusta e eficiente para análise detalhada dos resultados."
6. Dupla 6: Base de dados: Bank Marketing Dataset Biblioteca: PySide2
 - Enunciado: "Crie o sistema de classificação para o Bank Marketing Dataset com PySide2 como biblioteca de interface. O dataset contém 45211 amostras com 17 características, prevendo se um cliente fará um depósito a prazo.
 - Foque em criar visualizações detalhadas para análise de features importantes."
7. Dupla 7: Base de dados: Car Evaluation Dataset Biblioteca: PySimpleGUI

- Enunciado: "Implemente o sistema de classificação para o Car Evaluation Dataset usando PySimpleGUI para a interface. Este dataset possui 1728 amostras com 6 atributos, classificando carros em 4 classes.
 - Aproveite a simplicidade do PySimpleGUI para criar uma interface clara e funcional."
8. Dupla 8: Base de dados: Adult Dataset Biblioteca: Pygame
- Enunciado: "Desenvolva o sistema de classificação para o Adult Dataset utilizando Pygame para a interface gráfica. O dataset contém 48842 amostras com 14 atributos, prevendo se o rendimento excede \$50K/ano.
 - Use os recursos do Pygame para criar visualizações animadas dos resultados e do processo de treinamento."
9. Dupla 9: Base de dados: Mushroom Dataset Biblioteca: DearPyGui
- Enunciado: "Crie o sistema de classificação para o Mushroom Dataset com DearPyGui como biblioteca de interface. Este dataset inclui 8124 amostras de cogumelos com 22 atributos, classificados como comestíveis ou venenosos.
 - Explore os recursos modernos do DearPyGui para criar uma interface elegante e responsiva."
10. Dupla 10: Base de dados: Glass Identification Dataset Biblioteca: MESOP ou CustomTkinter
- Enunciado: "Implemente o sistema de classificação para o Glass Identification Dataset usando CustomTkinter para a interface gráfica. O dataset contém 214 amostras com 9 atributos, classificando tipos de vidro.
 - Aproveite os elementos personalizáveis do CustomTkinter para criar uma interface moderna e atraente."
11. Dupla 11: Base de dados: Abalone Dataset Biblioteca: MESOP
(<https://google.github.io/mesop/>)
- Enunciado: "Desenvolva o sistema de classificação para o Abalone Dataset utilizando MESOP para a interface gráfica. O Abalone Dataset contém 4177 amostras com 8 atributos, visando prever a idade de abalone.
 - Aproveite as capacidades do MESOP para criar uma aplicação web interativa e responsiva, explorando sua facilidade de uso para desenvolvimento rápido de aplicações internas."
12. Dupla 12: Base de dados: Poker Hand Dataset Biblioteca: Dash
- Enunciado: "Implemente o sistema de classificação para o Poker Hand Dataset usando Dash para a interface gráfica. Este dataset contém 1025010 amostras, cada uma sendo uma mão de poker com 5 cartas descritas usando 10 atributos categóricos.
 - Utilize os componentes interativos do Dash para criar visualizações dinâmicas e um painel de controle intuitivo para análise dos resultados."

Obs.: Cada dupla deve implementar o painel geral descrito no início, adaptando-o à sua biblioteca específica e às particularidades de sua base de dados. Encoraje os alunos a explorarem recursos únicos de suas bibliotecas escolhidas para enriquecer a experiência do usuário e a apresentação dos resultados.

Passo a Passo para Execução do Projeto:

1. Configuração do Ambiente
 - Quais bibliotecas específicas você precisa instalar para seu projeto?
 - Como você verificará se todas as dependências estão corretamente instaladas e funcionando?
2. Carregamento e Pré-processamento de Dados
 - Quais etapas de pré-processamento são necessárias para sua base de dados específica?
 - Como você lidará com valores ausentes ou outliers, se existirem?
 - Qual método você usará para normalizar ou padronizar os dados?
3. Implementação da Rede Neural
 - Como você estruturará seu código para permitir a fácil modificação dos parâmetros da rede?
 - Quais funções de ativação você implementará e por quê?
 - Como você implementará diferentes algoritmos de otimização?
4. Desenvolvimento da Interface Gráfica
 - Quais widgets da sua biblioteca de interface gráfica são mais adequados para cada elemento do layout proposto?
 - Como você organizará os elementos na interface para garantir uma boa experiência do usuário?
 - Que estratégias você usará para tornar a interface responsiva e intuitiva?
5. Integração da Rede Neural com a Interface
 - Como você conectará as funções da rede neural aos elementos da interface?
 - Qual método você usará para atualizar o progresso do treinamento em tempo real?
 - Como você tratará erros potenciais durante a interação usuário-sistema?
6. Implementação das Funcionalidades de Treinamento e Teste
 - Como você estruturará as funções de treinamento e teste para serem chamadas pela interface?
 - Que informações você fornecerá ao usuário durante o processo de treinamento?
 - Como você implementará a divisão dos dados em conjuntos de treino e teste?
7. Visualização de Resultados
 - Quais tipos de gráficos você usará para representar a evolução do treinamento?
 - Como você implementará a matriz de confusão de forma visualmente eficaz?
 - Que outras visualizações você pode adicionar para enriquecer a análise dos resultados?
8. Implementação da Comparação de Experimentos
 - Que estrutura de dados você usará para armazenar os resultados de diferentes experimentos?
 - Como você apresentará a comparação entre diferentes configurações de rede?

- Que funcionalidades adicionais você pode incluir para facilitar a análise comparativa?
- 9. Geração de Relatórios
 - Quais informações você incluirá no relatório gerado?
 - Como você estruturará o relatório para ser claro e informativo?
 - Que formato você usará para o relatório (PDF, HTML, texto simples)?
- 10. Testes e Otimização
 - Quais cenários de teste você implementará para garantir a robustez do sistema?
 - Como você avaliará e otimizará o desempenho da aplicação?
 - Que estratégias você usará para testar diferentes configurações de rede eficientemente?
- 11. Documentação
 - Como você estruturará o manual do usuário para ser claro e abrangente?
 - Que informações técnicas você incluirá na documentação do código?
 - Como você documentará as decisões de design e implementação tomadas durante o projeto?
- 12. Reflexão e Análise Final
 - Quais foram os principais desafios encontrados durante o desenvolvimento e como você os superou?
 - Que insights você obteve sobre o desempenho de diferentes configurações de rede neural em sua base de dados?
 - Como você avalia a eficácia da interface gráfica em facilitar a experimentação com redes neurais?

Explicação do Trabalho e sua Importância:

Este projeto integra conceitos fundamentais de aprendizado de máquina, desenvolvimento de software e interface gráfica, proporcionando uma experiência prática e abrangente no campo da inteligência artificial. Ao desenvolver um sistema de classificação com redes neurais e uma interface gráfica interativa, os alunos irão:

1. Aprofundar o entendimento prático de redes neurais e seus parâmetros.
2. Ganhar experiência em desenvolvimento de software com Python.
3. Aprender a criar interfaces gráficas intuitivas para sistemas complexos.
4. Desenvolver habilidades em análise e visualização de dados.
5. Praticar o trabalho em equipe e a comunicação técnica.

Resultados Esperados:

1. Código-fonte: Os alunos devem gerar um código bem estruturado e comentado, que deve ser hospedado em um repositório público no GitHub. Isso não apenas facilitará a avaliação, mas também servirá como um portfólio para futuros empregos ou projetos.
2. Relatório Técnico: Um documento detalhado descrevendo a implementação, desafios encontrados, decisões de design, análise dos resultados e conclusões. O

relatório deve seguir um formato acadêmico, incluindo introdução, metodologia, resultados, discussão e conclusão.

3. Manual do Usuário: Um guia claro e conciso sobre como utilizar a aplicação desenvolvida.
4. Apresentação de Seminário: Cada dupla fará uma apresentação de 15 minutos, seguida de 5 minutos para perguntas e discussão. A apresentação deve cobrir os principais aspectos do projeto, demonstrar o funcionamento da aplicação e discutir os resultados obtidos.

Metodologia:

Os alunos devem seguir uma metodologia de desenvolvimento iterativo:

1. Planejamento inicial e divisão de tarefas.
2. Implementação incremental das funcionalidades.
3. Testes regulares e refinamento do código.
4. Documentação contínua do processo e decisões tomadas.
5. Análise final dos resultados e preparação da apresentação.

Resultados Esperados:

1. Uma aplicação funcional que permita a configuração, treinamento e avaliação de redes neurais através de uma interface gráfica.
2. Análises comparativas do desempenho de diferentes configurações de rede neural.
3. Insights sobre a eficácia de vários parâmetros de rede para o problema de classificação específico.

Conclusões Esperadas:

Espera-se que os alunos concluam sobre:

1. A eficácia de diferentes arquiteturas de rede neural para seu problema específico.
2. O impacto de vários hiperparâmetros no desempenho do modelo.
3. A importância da interface do usuário no processo de experimentação com ML.
4. Desafios e soluções no desenvolvimento de aplicações de ML com interface gráfica.

Bibliografia Esperada (pode acrescentar, na realidade, deve):

1. Documentação oficial de TensorFlow/Keras e Scikit-learn.
2. Documentação da biblioteca de interface gráfica específica.
3. Artigos científicos relacionados à base de dados utilizada.
4. Livros-texto sobre redes neurais e aprendizado de máquina.
5. Recursos online sobre desenvolvimento de software e boas práticas de codificação.

Competências Necessárias (Antes do Trabalho):

1. Conhecimento básico de Python.
2. Compreensão dos fundamentos de redes neurais e aprendizado de máquina.
3. Familiaridade com conceitos de programação orientada a objetos.
4. Noções básicas de interfaces gráficas de usuário.
5. Habilidades básicas de análise de dados.

Competências e Ganhos (Após o Trabalho):

1. Proficiência em desenvolvimento de aplicações de ML com Python.
2. Habilidade em criar interfaces gráficas interativas para sistemas complexos.
3. Experiência prática em ajuste e otimização de redes neurais.
4. Capacidade de analisar e visualizar resultados de experimentos de ML.
5. Competência em documentação técnica e apresentação de projetos de software.
6. Experiência em trabalho colaborativo e controle de versão com Git.
7. Compreensão aprofundada da relação entre parâmetros de rede neural e desempenho do modelo.
8. Habilidade em projetar experimentos para comparar diferentes configurações de ML.
9. Capacidade de comunicar resultados técnicos para audiências diversas.

Finalizando: Este projeto desafiador e abrangente proporcionará aos alunos de ELE 606 (2024.1) uma **experiência que combina teoria e prática**, preparando-os para futuras carreiras em inteligência artificial, desenvolvimento de software e análise de dados.