

Roteiro Aula Prática



**FUNDAMENTOS DA INTELIGÊNCIA
ARTIFICIAL**

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: FUNDAMENTOS DA INTELIGÊNCIA ARTIFICIAL

Unidade: U3 _ LÓGICA NEBULOSA

Aula: A4_ LÓGICA NEBULOSA EM SISTEMAS ESPECIALISTA

Tempo previsto de execução de aula prática: 3h (CAMPO OBRIGATÓRIO – NÃO APARECER EM NENHUM RAP)

OBJETIVOS (campo obrigatório – exibição para todos)

Definição dos objetivos da aula prática:

- Introduzir o uso do Octave na resolução de problemas práticos de Lógica Nebulosa.
- Comparar um programa não baseado em Lógica Nebulosa com um programa baseado nessa técnica, apontando as vantagens do segundo tipo em uma classe de problemas propícia para seu uso.
- Resolver um problema simples usando Lógica Nebulosa

INFRAESTRUTURA (OBRIGATÓRIO SE HOVER – EXIBIÇÃO DOCENTE/TUTOR)

Instalações – Materiais de consumo – Equipamentos:

NOME DO LABORATÓRIO

- Laboratório de Informática

Equipamentos:

- Computador.
- ~ 1 unid. 1 aluno

SOLUÇÃO DIGITAL (OBRIGATÓRIO SE HOVER - APARECER PARA TODOS)

- Octave Simulador:

GNU Octave é uma linguagem computacional, desenvolvida para computação matemática. Possui uma interface em linha de comando para a solução de problemas numéricos, lineares e não-lineares, também é usada em experimentos numéricos. Faz parte do projeto GNU, é um software livre sob os termos da licença GPL.

<https://octave.org>

EQUIPAMENTO DE PROTEÇÃO INDIVIDUAL (EPI) (CAMPO OBRIGATÓRIO – APARECER PARA TODOS)

Não se aplica

PROCEDIMENTOS PRÁTICOS (OBRIGATÓRIO – TODOS)

Procedimento/Atividade nº 1

Atividade proposta:

Implantar a solução para um problema simples de cálculo de gorjeta com base em dois parâmetros SEM USAR A LÓGICA NEBULOSA PARA FINS DE COMPARAÇÃO:

- A qualidade da comida servida
- A Qualidade do serviço provido pelo restaurante

Como o julgamento de “bom” e “ruim” é difuso, a Lógica Nebulosa é uma ferramenta adequada para atacar o problema do cálculo, assumindo que podemos atribuir menos de 10% quando estamos insatisfeitos e mais de 10% quando formos positivamente surpreendidos pela comida e/ou pelo serviço.

Em, seguida, o mesmo problema (com os mesmos parâmetros), deverá ser resolvido usando a Lógica Nebulosa.

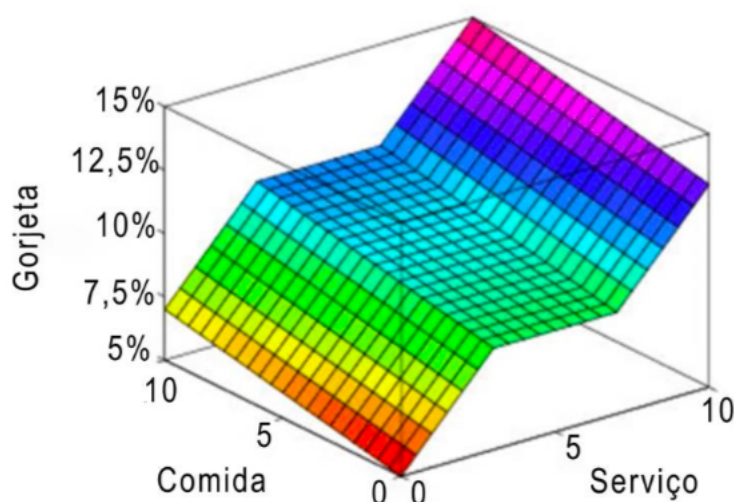
Imagine-se em um restaurante onde a gorjeta deve ser especificada pelo cliente, que inclusive discrimina o valor a ser cobrado no cartão (como está cada vez mais comum em restaurantes nos dias de hoje, nas capitais e nas grandes cidades). Para gerarmos o valor da gorjeta, devemos fazer um pequeno cálculo mental em função da qualidade da comida e também do serviço prestado pelo estabelecimento. Imaginemos que:

- Se a comida foi de qualidade ruim ou fomos mal atendidos, vamos deixar uma gorjeta pequena, mais próxima de 5%;
- Se o serviço for bom, vamos deixar uma gorjeta mediana, mais próxima de 10%;
- Se a comida ou o serviço nos surpreenderam positivamente, vamos deixar uma gorjeta generosa, mais próxima de 15%;

Vamos, ainda, graduar a comida e o serviço em uma escala de 0 a 10.

Com esses parâmetros em mente, poderíamos criar um gráfico com a gorjeta a ser deixada, conforme a Figura 1, a seguir:

Figura 1 – Gráfico de Definição da Gorjeta com base na qualidade da comida e do serviço



Fonte: autor

Nesta primeira fase do exercício, vamos implementar a solução do problema proposto usando os comandos procedurais do Octave, sem utilizar as ferramentas de Lógica Nebulosa, para fins de comparação.

Agora, usando a mesma definição de grupos utilizada nos passos anteriores, você deverá criar o mesmo resultado, por meio da Lógica Nebulosa.

Alguns passos serão seguidos para que esse resultado seja obtido, a saber:

1. Definir as entradas e saídas do procedimento
2. Criar Funções que determinam o quanto um elemento pertence a um conjunto
3. Criar regras para combinar os dados
4. Simular o sistema de Lógica Nebulosa resultante.

Checklist:

- Octave está funcionando corretamente.
- Criar as variáveis necessárias.
- Inserir as regras corretamente.
- Gerar valores para a gorjeta com base em diferentes qualidades de serviço e comida.
- Registrar seus resultados em um relatório.

RESULTADOS (obrigatório – aparecer para todos)

Resultados de Aprendizagem:

Resolução de um problema usando Lógica Nebulosa

ESTUDANTE, VOCÊ DEVERÁ ENTREGAR (não obrigatório – aparecer para todos)

Descrição orientativa sobre a entregada da comprovação da aula prática:

Entrega do relatório com os resultados obtidos e do código desenvolvido.

REFERÊNCIAS BIBLIOGRÁFICAS (não obrigatório – aparecer para todos)

Descrição (em abnt) das referências utilizadas

Roteiro Aula Prática



FUNDAMENTOS DA INTELIGÊNCIA ARTIFICIAL

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: FUNDAMENTOS DA INTELIGÊNCIA ARTIFICIAL

Unidade: U4 _ REDES NEURAIAS ARTIFICIAIS

Aula: A4_ ALGORITMOS DE REDES NEURAIAS

Tempo previsto de execução de aula prática: 3h (CAMPO OBRIGATÓRIO – NÃO APARECER EM NENHUM RAP)

OBJETIVOS (campo obrigatório – exibição para todos)

Definição dos objetivos da aula prática:

Compreensão e desenvolvimento de um rede neural utilizando a linguagem Python.

INFRAESTRUTURA (OBRIGATÓRIO SE HOVER – EXIBIÇÃO DOCENTE/TUTOR)

Instalações – Materiais de consumo – Equipamentos:

NOME DO LABORATÓRIO

- Laboratório de Informática

Equipamentos:

- Computador.
- ~ 1 unid. 1 aluno

SOLUÇÃO DIGITAL (OBRIGATÓRIO SE HOVER - APARECER PARA TODOS)

Infraestrutura mínima necessária para execução.

- Jdoodle

O JDoodle Python é uma plataforma online que oferece um ambiente de desenvolvimento integrado (IDE) para programação em Python diretamente no navegador. Ideal para iniciantes e programadores experientes, o JDoodle Python permite escrever, executar e compartilhar código Python de maneira simples e eficiente. Com recursos como realce de sintaxe, sugestões de código e feedback imediato, os usuários podem testar e depurar seus programas com facilidade. Além disso, o JDoodle Python suporta uma variedade de bibliotecas e módulos Python, facilitando o desenvolvimento de projetos complexos e a exploração de novas funcionalidades da linguagem.

<https://www.jdoodle.com/python3-programming-online>

EQUIPAMENTO DE PROTEÇÃO INDIVIDUAL (EPI) (CAMPO OBRIGATÓRIO – APARECER PARA TODOS)

Não se Aplica

PROCEDIMENTOS PRÁTICOS (OBRIGATÓRIO – TODOS)

Procedimento/Atividade nº 1

Atividade proposta:

Implemente uma rede neural de uma camada utilizando a função de ativação sigmoide. O objetivo é treinar a rede para realizar a classificação binária. Você receberá uma matriz de entrada e uma matriz de saída desejada e deverá ajustar os pesos da rede (sinapse) através do processo de aprendizado supervisionado.

Passos do Exercício

- **Importação das Bibliotecas Necessárias:**
Utilize a biblioteca NumPy para realizar operações matemáticas e manipulações de arrays.
- **Definição da Função Sigmoide:**
Implemente a função de ativação sigmoide que será utilizada tanto para a ativação quanto para a derivada no cálculo do gradiente.
- **Definição das Entradas e Saídas:**
Crie a matriz de entrada X e a matriz de saída y .
- **Inicialização dos Pesos:**
Inicialize os pesos da sinapse com valores aleatórios.
- **Treinamento da Rede Neural:**
Implemente o loop de treinamento que ajustará os pesos da sinapse. O treinamento deve ser executado por 10.000 iterações.
- **No loop de treinamento, execute os seguintes passos:**
 - Realize a propagação para frente calculando as saídas.
 - Calcule o erro.
 - Calcule o delta (ajuste dos pesos) utilizando a derivada da função sigmoide.
 - Atualize os pesos da sinapse.
- **Exibição dos Resultados:**
Após o treinamento, imprima a saída da rede.

Checklist:

- Entender o conceito de redes neurais artificiais;
- Implementar através do Python uma RNA simples.
- Realizar e entender os resultados da quantidade de treinamentos.

RESULTADOS (obrigatório – aparecer para todos)

Resultados de Aprendizagem:

Compreenda a implementação de uma RNA e o conceito de treinamento.

ESTUDANTE, VOCÊ DEVERÁ ENTREGAR (não obrigatório – aparecer para todos)

Descrição orientativa sobre a entrega da comprovação da aula prática:

Código desenvolvido.

REFERÊNCIAS BIBLIOGRÁFICAS (não obrigatório – aparecer para todos)

Descrição (em abnt) das referências utilizadas