
UFC – UNIVERSIDADE FEDERAL DE CEARÁ
CAMPUS DE SOBRAL
CURSO: ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA: INTELIGÊNCIA COMPUTACIONAL
PROF. JARBAS JOACI DE MESQUITA SÁ JUNIOR

CONTEÚDO

SUMÁRIO DO CONTEÚDO

Aula 01 - 00/00/2025.....	2
Aula 0 - 00/00/2025.....	2
Aula 0 - 00/00/2025.....	3
Aula 0 - 00/00/2025.....	4
Aula 0 - 00/00/2025.....	5
Aula 0 - 00/00/2025.....	6
Aula 0 - 00/00/2025.....	7
Aula 0 - 00/00/2025.....	8
Aula 0 - 00/00/2025.....	9
Aula 0 - 00/00/2025.....	10
Aula 0 - 00/00/2025.....	12
Aula 0 - 00/00/2025.....	12
Aula 0 - 00/00/2025.....	12

Slide 01 - Introdução e Breve Histórico

As visões comumente aceitas são:

1) **Agir como humanos: (Teste de Turing)**

Para passar no teste de Turing o computador precisa ter as seguintes habilidades:

- Processamento de linguagem natural.
- Representação de conhecimento
- Raciocínio automatizado.
- Aprendizado de máquina.

2) **Pensar como humanos:** Abordagem baseada na modelagem cognitiva

3) **Pensar racionalmente:** Aristóteles se preocupou com as leis do pensamento que governam a operação da mente;

4) **Agir racionalmente:** Comportamento racional: fazendo a “coisa certa”.

Os Fundamentos de IA...

Slide 02 - Busca Sem Informação

Resolução de Problemas Via Busca: Estrutura da Apresentação

- Tipos de Problemas: Estados únicos (totalmente **observável**) e os ambientes devem ser **discretos** e **determinísticos**.
- Formulação do Problema: Problemas exemplo.
- Algoritmos de Busca Básicos: Não informado.

Exemplos:

→ Mapa da Romênia

→ Gráfico do Espaço de Estados do Aspirador de Pó.

→ 8-puzzle.

→ Problema das 8 rainhas.

Soluções de Problemas por Busca: Quatro passos gerais

1. **Formulação da Meta:** Qual ou quais estados correspondem à solução do problema?
2. **Formulação do problema:** Quais ações e estados considerar dada a meta?
3. **Busca pela solução:** Encontre uma sequência (ou a melhor das sequências) de ações que leve à meta.
4. **Execução:** Implemente as ações.

Algoritmos de Busca Básicos

Em geral, a busca se dá, na verdade, sobre um **grafo** (mesmo estado alcançado por múltiplos caminhos).

Implementação: nodos vs. estados

- Um estado é uma representação de uma configuração física.

- Um nodo é uma estrutura de dados constituindo parte da árvore de busca e que inclui estado, nodo pai, ação, custo de caminho $g(x)$ e profundidade.
- Vários nós diferentes podem representar o mesmo estado.

Estratégias de Buscas

É basicamente uma ordem de expansão dos nodos.

Medidas de desempenho para diferentes estratégias:

- Completude: Sempre encontra uma solução (se existir)?
- Otimalidade: Sempre encontra a solução de custo mais baixo?
- Complexidade (tempo): Número de nodos explorados?
- Complexidade (espaço): Número de nodos armazenados?

Complexidade usualmente medida em função da dificuldade do problema:

- b : máximo fator de ramificação da árvore de busca.
- d : profundidade da solução de menor custo.
- m : máxima profundidade do espaço de estados (pode ser ∞).

Estratégias Não-Informadas

Também denominadas de busca cega: Usam estritamente a informação disponível na formulação do problema. Quando é possível utilizar **informação adicional** para determinar se um nodo não-meta é mais promissor do que outro temos uma **busca informada**.

Diferenciam-se pela abordagem de expansão: | *Conteúdo de prova* |

- Busca em largura (Breadth-first search).
- Busca uniforme (Uniform-cost search).
- Busca em profundidade (Depth-first search).
- Busca em profundidade limitada (Depth-limited search).
- Busca em profundidade iterativa (Iterative deepening search).
- Busca Bidirecional.

Evitar Geração de Estados Repetidos

1. Não retornar ao estado “pai”.
2. Não criar caminhos com ciclos.
3. Não gerar qualquer estado que já tenha sido criado antes (em qualquer ramo).

Tensão (tradeoff) Básica.

Aula 0 - 00/00/2025

Slide 03 - Busca Com Informação

Busca Best-First | *Conteúdo de prova* |

- Nodo é selecionado para expansão baseado em uma função de avaliação $f(n)$.
- Ideia: Estimar a distância até a meta. Escolhe o nodo que aparenta ser o melhor - segundo $f(n)$

Casos especiais: **Busca Gulosa** e **Busca A***.

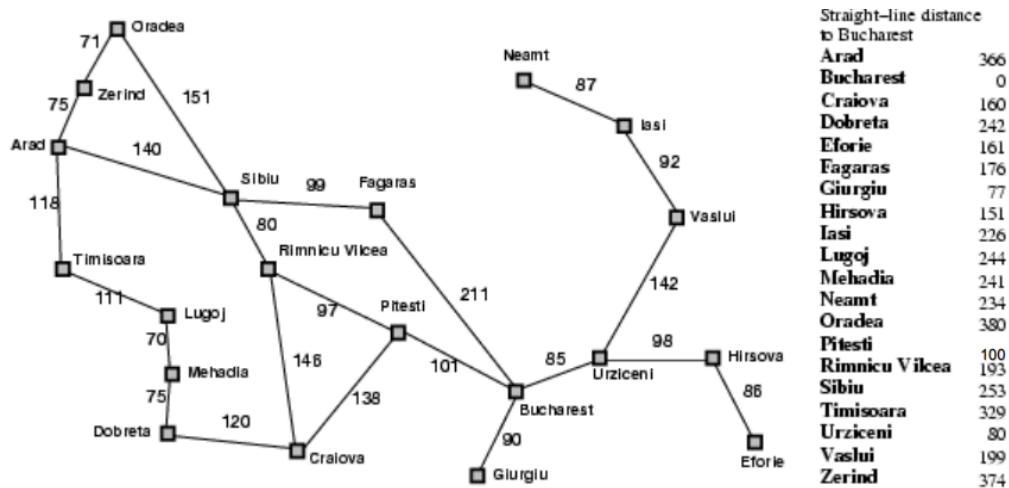
Função Heurística | *Conteúdo de prova* |

- $b(n)$ = custo estimado do caminho mais curto do nodo n até o nodo meta.

- Condição Básica: Se n é a meta então $b(n) = 0$.

Exemplo:

→ Mapa da Romênia



→ Busca Gulosa.

→ Busca A*.

Heurísticas Admissíveis | *Conteúdo de prova* |

Exemplo:

→ 8-puzzle

Dominância...

Problema relaxado | *Conteúdo de prova* |

Aula 0 - 00/00/2025

Slide 04 - Busca Agentes Lógicos

Agentes Baseados em Conhecimento

Abordagem declarativa para construir um agente:

- **Tell:** Informar o que o agente precisa saber.
- **Ask:** Para o agente obter respostas, devem seguir o que foi informado à Base de conhecimento (BC) = *conjunto de sentenças numa linguagem formal*.

Exemplo:

→ **Mundo de Wumpus:** É um ambiente clássico da inteligência artificial usado para demonstrar conceitos de lógica e inferência. Ele é um jogo de tabuleiro onde um agente (um jogador) deve explorar um ambiente desconhecido para encontrar ouro sem cair em armadilhas (poços) ou ser morto pelo Wumpus, um monstro. O agente recebe pistas sobre o ambiente, como:

- Brisa → indica um poço próximo.
- Fedor → indica que o Wumpus está perto.
- Brilho → indica que o ouro está na sala.

O agente usa a lógica para inferir informações sobre o ambiente e tomar decisões baseadas nas percepções recebidas.

- **Noções sobre dedução.**
- **Noções sobre Modelos.**
- **Noções sobre inferência.**

Sistemas de Lógica: Geralmente consiste de:

- *Sintaxe:* Símbolos e combinações de símbolos que constituem a linguagem lógica.
- *Semântica:* Significado para as partes da linguagem, permitindo sua interpretação.
- *Teoria da Prova:* Conjunto de regras de inferência.

Tabela verdade para os conectivos lógicos						
P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	false	true	false	false	true	true
false	true	true	false	true	true	false
true	false	false	false	true	false	false
true	true	false	true	true	true	true

Teoria da Prova (regras de inferência):

1. Modus Ponens.
2. Eliminação do E.
3. Introdução do E.
4. Introdução do OU.
5. Eliminação da Dupla Negação.
6. Silogismo disjuntivo.
7. Resolução.
8. Silogismo hipotético.

Regras de Inferência e Equivalência.

Aula 0 - 00/00/2025

Slide 05 - Lógica Fuzzy I

É uma extensão da lógica clássica que lida com conceitos vagos ou imprecisos, em vez de apenas valores verdadeiros binários (verdadeiro ou falso).

Grau de Pertinência

- Na lógica clássica, um elemento pertence ou não a um conjunto (valores 0 ou 1).
- Na Lógica Fuzzy, um elemento pode pertencer a um conjunto com um grau de pertinência entre 0 e 1. Isso permite representar conceitos como "parcialmente verdadeiro" ou "parcialmente falso".

Exemplo:

Considere o conjunto "pessoas altas". Na lógica clássica, alguém com 1,80m pode ser considerado "alto" (1) e alguém com 1,70m "não alto" (0).

Na Lógica Fuzzy, uma pessoa com 1,75m pode ter um grau de pertinência de 0,7 ao conjunto "pessoas altas", indicando que ela é "mais ou menos alta".

Funções de Pertinência para Conjuntos Fuzzy: É definido por uma função de pertinência que mapeia elementos de um universo de discurso para valores no intervalo [0, 1].

Operadores de Conjunto: As operações básicas da lógica clássica (E, OU, NÃO) são estendidas para conjuntos fuzzy.

- Como resultado, os operadores de conjuntos são comutativos, associativos e distributivos e obedecem às leis de De Morgan, e o complemento de um conjunto.
- Temos o operador de inclusão: $A \supset B$. | conjunto A contém um conjunto B $\Rightarrow A \cup B = A \Rightarrow A \cap B = B$ |

Hedge (modificador): É um qualificador de conjunto fuzzy, como "muito", "extremamente", "um pouco". Quando um desses qualificadores é aplicado ao conjunto fuzzy, tal como "pessoas altas", nós produzimos um novo conjunto.

Lógica Fuzzy

- É não-monotônica, no sentido de que se um novo fato fuzzy for adicionado à base de dados, esse fato pode contradizer conclusões que foram previamente obtidas da base de dados.
- Da mesma forma que MIN e MAX foram usadas para a intersecção e união de conjuntos fuzzy, elas também podem ser usadas para calcular a conjunção e disjunção de variáveis fuzzy.
- Possui o operador \rightarrow , de acordo com a lógica clássica, a seguinte igualdade é verdadeira $A \rightarrow B = \neg A \vee B$.
- **implicação de Gödel**, pode ser definida como: $A \rightarrow B = (A \leq B) \vee B$.

Aula 0 - 00/00/2025

Slide 06 - Lógica Fuzzy II

Inferência Nebulosa

- Uma alternativa à implicação de Gödel chamada implicação de Mamdani (ou inferência de Mamdani) é geralmente usada em sistemas nebulosos.
- É um processo de raciocínio utilizado em sistemas baseados em Lógica Fuzzy para derivar conclusões a partir de regras fuzzy e entradas imprecisas.
- Permite combinar regras fuzzy (baseadas em conjuntos fuzzy) com entradas fuzzy para produzir uma saída fuzzy.
- Diferente da lógica clássica, que opera com valores verdadeiros binários (verdadeiro ou falso), a inferência nebulosa lida com graus de pertinência, permitindo conclusões mais flexíveis e próximas do raciocínio humano.

Componentes da Inferência Nebulosa

- **Conjuntos Fuzzy:** Representam conceitos vagos (como "quente", "frio", "alto", "baixo") por meio de funções de pertinência.
- **Regras Fuzzy:** São expressas na forma "Se... Então..." (ex.: "Se a temperatura está quente, então aumente a potência do ar-condicionado").

- **Entrada Fuzzy:** Dados de entrada são "fuzzificados", ou seja, convertidos em graus de pertinência aos conjuntos fuzzy.
 - **Saída Fuzzy:** O resultado da inferência é um conjunto fuzzy, que pode ser **desnebulização** (*defuzzification*) para obter um valor numérico.
-

Aula 0 - 00/00/2025

Slide 07 - Reconhecimento de Padrões - RP

É um ramo do aprendizado de máquina e consiste em atribuir um rótulo (ou classe) para uma certa amostra ou valor de entrada. Em geral, visa fornecer uma resposta razoável para todas as entradas possíveis e realizar a correspondência "mais provável" das entradas, tendo em conta a sua variação estatística.

Para definir um problema de RP precisamos

- Um número finito de k classes: C_1, C_2, \dots, C_k .
- Um número finito de N_i objetos por classe C_i .
- Um número finito de p atributos (features) para representar numericamente cada objeto físico.
- Um mecanismo de memória e/ou aprendizado.
- Uma regra de decisão para classificar novos objetos.
- Um critério de avaliação do classificador.

Definições Básicas - Atributo

Tipos de atributos possíveis:

- Nominal: Cor, identificação, profissão, ...
- Ordinal: Gosto (ruim, médio, bom), dias da semana, ...
- Intervalar: Temperatura em Celsius, ...
- Racional: Peso, tamanho, idade ...
- **Atributos Discretos:** Assumem um número contável (enumerável) de valores.
- **Atributos Contínuos:** Assumem uma quantidade incontável de valores (valores que são números reais).
- **Atributos Inadequados:** São atributos que não contém nenhuma informação relevante para a separação das classes, não importa o classificador usado. *Solução: definir novos atributos.*
- **Atributos Correlacionados:** Pode acontecer de 2 atributos distintos serem influenciados por um mecanismo comum de modo que variem juntos. Isso pode degradar o desempenho do classificador.
 - Ex.: raio e área do círculo.

Tipos de Aprendizado

Principais Paradigmas de Treinamento:

- **Supervisionado:** Consiste em apresentar um padrão a ser reconhecido juntamente com a resposta que o sistema deve fornecer ao deparar-se novamente com o esse padrão. Guiado por um "professor" externo que possui conhecimento sobre o ambiente.
- **Semi-supervisionado:** Combina uma pequena quantidade de amostras classificadas com um grande número de amostras não classificadas para produzir melhores classificadores. Tem um "professor" externo apenas para parte dos exemplos de treinamento.

- **Não supervisionado:** Não tem “crítico” ou “professor” externo, apenas os dados de entrada e busca extrair as propriedades estatisticamente relevantes.
- **Reforço:** Tipo de treinamento intermediário entre o supervisionado e o não supervisionado. Guiado por um “crítico” externo.
 - Se ação tomada por sistema é seguida por estado satisfatório, sistema é fortalecido, caso contrário, sistema é enfraquecido (lei de Thorndike).
 - Tipos de reforço: Positivo (recompensa), Negativo (punição) e Nulo.

Função Alvo: Trata-se da função objetivo. Ela estabelece qual conhecimento será aprendido. Permite também verificar quão bem o conhecimento foi aprendido.

- Função discriminante entre classes.
 - Função de similaridade intra grupos.
 - Etc.
-

Aula 0 - 00/00/2025

Slide 08 - Regressão Simples

A análise de regressão abrange uma série de técnicas voltadas para a modelagem e a investigação de relações entre duas ou mais variáveis aleatórias.

Definição do Problema...

Conceito Básico

- **Variável Dependente (y):** A variável que queremos prever ou explicar.
- **Variável Independente (x):** A variável usada para explicar ou prever y .
- **Relação:** A regressão simples assume que a relação entre x e y pode ser aproximada por uma função linear:

$$y = \beta_0 + \beta_1 x + \epsilon.$$

- β_0 : Intercepto (valor de y quando $x = 0$).
- β_1 : Coeficiente angular (mudança em y para cada unidade de x).
- ϵ : Erro aleatório (diferença entre o valor observado e o valor previsto).

Métodos Mínimos Quadrados (MQ)

- Minimiza a soma dos quadrados dos resíduos (diferenças entre os valores observados e previstos).
- As fórmulas para calcular β_0 e β_1 são:

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

$$\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}.$$

- \bar{x} : Média dos valores de x .
- \bar{y} : Média dos valores de y .

Entendendo o problema: Minimizar a função-custo equivale a fazer com que a soma dos quadrados dos desvios entre os valores medidos (observações) e a reta de regressão seja mínima.

Análise de Resíduos: Ela consiste em examinar os resíduos, que são as diferenças entre os valores observados e os valores previstos pelo modelo. A análise de resíduos ajuda a verificar se o modelo está bem ajustado aos dados e se as suposições da regressão são válidas.

Exercício Proposto

Determinar a reta de regressão associada a cada uma das regiões **R1**, **R2** e **R3**. Ou seja, determinar

- **R1:** $\hat{y} = \hat{\beta}_0^{(1)} + \hat{\beta}_1^{(1)} x$.
- **R2:** $\hat{y} = \hat{\beta}_0^{(2)} + \hat{\beta}_1^{(2)} x$.
- **R3:** $\hat{y} = \hat{\beta}_0^{(3)} + \hat{\beta}_1^{(3)} x$.

Em que $\hat{\beta}_0^{(i)}$ e $\hat{\beta}_1^{(i)}$ definem o intercepto e a inclinação da i -ésima da reta de regressão, $i = 1, 2$ e 3 .

Aula 0 - 00/00/2025

Slide 09 - Regressão Múltipla

É uma extensão da regressão linear simples que permite modelar a relação entre uma variável dependente y e **duas ou mais variáveis independentes** (x_1, x_2, \dots, x_n) . Ela é amplamente utilizada para prever resultados, entender relações entre variáveis e testar hipóteses.

Em geral, a variável de saída ou resposta, y , pode ser relacionada a k variáveis de entrada.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_k x_k + \varepsilon.$$

Regularização de Tikhonov: Também conhecida como **Regularização Ridge**, é uma técnica usada para resolver problemas de **regressão linear** (simples ou múltipla) quando há **multicolinearidade** (alta correlação entre variáveis independentes) ou quando o número de variáveis é maior que o número de observações. Ela adiciona uma penalização aos coeficientes do modelo para evitar sobreajuste (overfitting) e melhorar a generalização.

1. **Problema da Regressão Linear sem Regularização.**
2. **Regularização de Tikhonov (Ridge).**
3. **Efeito da Regularização**

Coeficiente de Determinação Ajustado: (R^2 ajustado) é uma versão modificada do R^2 tradicional que leva em conta o número de variáveis independentes no modelo de regressão. Ele é usado para avaliar a qualidade do ajuste de um modelo, penalizando a adição de variáveis que não contribuem significativamente para a explicação da variável dependente.

Regressão Polinomial

- É uma extensão da regressão linear que permite modelar relações não lineares entre a variável dependente (y) e uma variável independente (x) por meio de **termos polinomiais** (como x^2 , x^3 , etc.). Ela é útil quando a relação entre as variáveis não pode ser adequadamente descrita por uma linha reta.
- São amplamente usados nos casos em que a relação entre a variável de saída e de entrada é curvilínea (i.e. não-linear).

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_k x^k + \varepsilon.$$

Aula 0 - 00/00/2025

Slide 10 - Redes Neurais

1. Introdução e Ementa

- O material aborda os fundamentos das redes neurais artificiais (RNAs), desde conceitos básicos até aplicações práticas.
- Tópicos principais incluem: neurônio biológico, neurônio artificial de McCulloch-Pitts (MP), Perceptron Simples e Multicamadas (MLP), algoritmos de aprendizagem (como backpropagation), e aplicações em problemas reais.

2. Neurônio Biológico

- Descrição das partes do neurônio biológico: dendritos (entrada), sinapses (transmissão), corpo celular (processamento), e axônio (saída).
- Funcionamento: potencial de ação, limiar de disparo (-55 mV), e codificação da informação na frequência dos pulsos.
- Curiosidades: número de neurônios no cérebro (100 bilhões), sinapses por neurônio (10.000), e consumo energético (20% do corpo em repouso).

3. Neurônio Artificial de McCulloch-Pitts (MP)

- Modelo matemático simplificado do neurônio biológico, proposto em 1943.
- Componentes:
 - Entradas (x_1, x_2, \dots, x_p).
 - Pesos sinápticos (w_1, w_2, \dots, w_p).
 - Limiar de ativação (θ).
 - Função de ativação degrau: *saída binária* (+ 1 se $u > 0$, - 1 caso contrário).
- Aplicação em portas lógicas (AND, OR, NOT) e análise geométrica (separação linear de classes).

4. Perceptron Simples (PS)

- Combina o neurônio MP com uma regra de aprendizagem para ajuste automático dos pesos.
- Regra de aprendizagem:
$$w(t + 1) = w(t) + ne(t)x(t),$$
onde n é o passo de aprendizagem e $e(t)$ é o erro.
- Exemplo passo a passo: Implementação da porta OR.
- Limitação: só resolve problemas linearmente separáveis.

5. Problemas Não-Linearmente Separáveis (Porta XOR)

- A porta XOR não pode ser implementada com um único neurônio MP devido à não linearidade.
- Solução: uso de redes multicamadas com pelo menos três neurônios (duas camadas ocultas).

6. Perceptron Multicamadas (MLP)

- Estrutura:
 - Camada de entrada, uma ou mais camadas ocultas, e camada de saída.
 - Neurônios com funções de ativação sigmoidais (logística ou tangente hiperbólica).
- Algoritmo de treinamento: Backpropagation, que ajusta os pesos retropropagando o erro da saída para as camadas ocultas.
- Função-custo: Erro Quadrático Médio (EQM).
- Universalidade: MLP pode aproximar qualquer função contínua com precisão arbitrária, dado número suficiente de neurônios ocultos.

7. Dicas e Heurísticas

- Normalização dos dados de entrada para melhor desempenho.
- Evitar paralisia da rede (saturação dos neurônios) com inicialização adequada de pesos e adição de pequenos valores à derivada da função de ativação.
- Determinação do número de neurônios ocultos: regras heurísticas como valor médio, raiz quadrada, ou Kolmogorov.
- Balanceamento entre underfitting (poucos neurônios) e overfitting (muitos neurônios).

8. Aplicações Práticas

- Exemplos incluem diagnóstico médico (dermatologia), identificação de padrões em aerogeradores, e análise de crédito bancário.
- Demonstração da versatilidade das redes neurais em problemas complexos e não lineares.

9. Conclusão

- O material fornece uma base sólida para entender e implementar redes neurais artificiais, desde modelos simples até arquiteturas avançadas como MLP.
- Ênfase na importância do aprendizado supervisionado (backpropagation) e na escolha adequada de parâmetros para otimizar o desempenho da rede.

Aula 0 - 00/00/2025

Slide 11 - Rede Neural MLP

Aula 0 - 00/00/2025

Slide 12 - Rede Neural RBF

Aula 0 - 00/00/2025

Slide - Rede Neural ELM