

EXERCICIOS LPO

Vamos traduzir as sentenças para a Lógica de Primeira Ordem (LPO):

"Marcelo gosta de todos os carros da Ford"

Para todos x, se x é um carro da Ford, então Marcelo gosta de x.

Em LPO: $\forall x (\text{CarroDaFord}(x) \rightarrow \text{Gosta}(\text{Marcelo}, x))$

"Maçãs e Frango são comida."

Maçã é comida e Frango é comida.

Em LPO: $\text{Comida}(\text{Maçã}) \wedge \text{Comida}(\text{Frango})$

"Qualquer coisa que alguém coma e que não cause sua morte é comida."

Para todos x, para todo y, se y come x e x não causa a morte de y, então x é comida.

Em LPO: $\forall x \forall y (\text{Come}(y, x) \wedge \neg \text{CausaMorte}(x, y) \rightarrow \text{Comida}(x))$

"Paulo comprou um carro novo e passou a ficar mais feliz."

Paulo comprou um carro novo e Paulo está mais feliz.

Em LPO: $\text{ComprouCarroNovo}(\text{Paulo}) \wedge \text{MaisFeliz}(\text{Paulo})$

"Susana namora Paulo porque ele comprou um carro."

Susana namora Paulo e Paulo comprou um carro.

Em LPO: $\text{Namora}(\text{Susana}, \text{Paulo}) \wedge \text{ComprouCarro}(\text{Paulo})$

Por favor, note que a LPO é uma forma de representar o conhecimento que é mais formal e precisa do que a linguagem natural, e pode ser útil para a resolução de problemas em inteligência artificial.

Modus Ponens

Modus Ponens é uma regra de inferência válida e uma das mais comuns na lógica. A forma geral do Modus Ponens é a seguinte:

Se P, então Q.

P é verdadeiro.

Portanto, Q é verdadeiro.

Em outras palavras, se temos uma afirmação condicional (Se P, então Q) e sabemos que a primeira parte dessa afirmação (P) é verdadeira, então podemos concluir que a segunda parte (Q) também é verdadeira.

Por exemplo, considere as seguintes afirmações:

Se está chovendo, então a rua está molhada.

Está chovendo.

Usando Modus Ponens, podemos concluir que:

Portanto, a rua está molhada.

Modus Ponens é uma ferramenta importante na lógica e é frequentemente usada em provas matemáticas, programação e inteligência artificial.

Exercício de Resolução

3 - Exercício 7.9 (Livro). Resolva a questão abaixo usando Resolução. As afirmações são:

Se um unicórnio é mítico, então é imortal.

Porém se ele não é mítico, então é um mamífero mortal.

Se o unicórnio é imortal ou mamífero, então ele tem chifre.

O unicórnio é mágico se tem chifre. Pergunta-se: O unicórnio é mágico ?

Para resolver essa questão usando resolução, primeiro precisamos converter as afirmações para a forma normal conjuntiva (FNC). As afirmações são:

Se um unicórnio é mítico, então é imortal. ($M \rightarrow I$)

Se ele não é mítico, então é um mamífero mortal. ($\neg M \rightarrow (Mam \wedge \neg I)$)

Se o unicórnio é imortal ou mamífero, então ele tem chifre. ($(I \vee Mam) \rightarrow C$)

O unicórnio é mágico se tem chifre. ($C \rightarrow Mag$)

Convertendo para a FNC, temos:

$\neg M \vee I$

$M \vee (Mam \wedge \neg I)$

$\neg I \vee \neg Mam \vee C$

$\neg C \vee Mag$

Agora, queremos saber se o unicórnio é mágico, então queremos provar Mag . Para fazer isso, negamos Mag e adicionamos à nossa lista:

$\neg Mag$

Agora, usamos a resolução para tentar obter uma contradição. Podemos combinar as cláusulas 4 e 5 para obter $\neg C$. Então, podemos combinar $\neg C$ com a cláusula 3 para obter $\neg I \vee \neg Mam$. No entanto, isso não nos leva a uma contradição, o que significa que não podemos provar que o unicórnio é mágico a partir das afirmações fornecidas. Portanto, a resposta para a pergunta "O unicórnio é mágico?" é que não podemos determinar com base nas informações fornecidas.

5 – (Visão Computacional) Qual a diferença entre os filtros de fechamento e de abertura ?

Os filtros de abertura e fechamento são técnicas utilizadas no processamento de imagens, especialmente na morfologia matemática, para remover ruídos e melhorar a qualidade da imagem.

Filtro de Abertura: Este filtro é uma operação de erosão seguida por uma operação de dilatação. A erosão é um processo que "afina" ou "erode" os objetos na imagem, enquanto a dilatação é um processo que "engrossa" ou "expande" os objetos. O filtro de abertura é útil para remover ruídos pequenos e separar objetos que estão conectados. Ele preserva a forma dos objetos maiores enquanto elimina os menores.

Filtro de Fechamento: Este filtro é uma operação de dilatação seguida por uma operação de erosão. O filtro de fechamento é útil para preencher pequenos buracos e conectar objetos próximos. Ele tende a preservar a forma dos objetos maiores enquanto preenche lacunas nos menores.

Em resumo, a principal diferença entre os dois é a ordem das operações. A abertura é útil para remover ruídos e separar objetos, enquanto o fechamento é útil para conectar objetos e preencher lacunas.

6 – (Visão Computacional) Cite pelo menos 4 métodos de visão computacional que você aplicaria para extrair o círculo, corretamente de imagem, que permitisse gerar uma imagem B a partir de uma imagem A ?

Transformada de Hough para círculos: A Transformada de Hough é uma técnica que pode ser usada para detectar formas geométricas simples, como linhas e círculos. No caso de círculos, a Transformada de Hough para círculos seria uma escolha adequada.

Segmentação de imagem: A segmentação de imagem é o processo de dividir uma imagem em várias partes, que são geralmente regiões de interesse. Métodos de segmentação como o crescimento de regiões ou a segmentação baseada em watershed podem ser usados para isolar o círculo.

Detecção de bordas: A detecção de bordas é uma técnica que identifica os pontos em uma imagem onde o brilho muda bruscamente. Métodos de detecção de bordas, como o operador de Sobel ou o detector de bordas de Canny, podem ser usados para identificar o contorno do círculo.

Filtragem morfológica: A filtragem morfológica é uma técnica que pode ser usada para remover ruído ou outros objetos pequenos da imagem que não são de interesse. Operações morfológicas, como erosão e dilatação, podem ser usadas para limpar a imagem antes da extração do círculo.

Depois de extrair o círculo usando um ou mais desses métodos, você pode criar a imagem B a partir da imagem A, colocando o círculo extraído em uma nova imagem ou alterando seu contexto na imagem original.

8 – (CSP) Exercício 5.6 (Livro). Resolva o problema criptoaritmético (modelado em aula e abaixo), na mão, usando busca em retrocesso, verificação prévia e a heurística VRM.

T W O
+ T W O

F O U R

O problema criptoaritmético é um tipo de problema de satisfação de restrições (CSP) onde dígitos são atribuídos a letras de tal forma que a equação seja satisfeita. A equação dada é:

markdown

Copy code

```
T W O
+ T W O
-----
F O U R
```

Aqui estão as restrições que temos:

1. Cada letra representa um dígito distinto.
2. A letra F deve ser 1 porque é o único número que pode ser carregado quando dois números de um dígito são somados.
3. O valor de O deve ser par, pois a soma de dois números iguais sempre resulta em um número par.
4. $T + T$ deve ser igual a $10 + O$, pois a soma de dois números iguais é sempre um número par e menor que 20.

Usando a heurística de Valor Mais Restritivo (VRM), começamos atribuindo valores às variáveis com o maior número de restrições. Nesse caso, começamos com F e O.

Atribuímos $F = 1$, pois é a única opção disponível para F. Agora, precisamos escolher um número par para O. Vamos escolher $O = 0$, pois é o número par disponível mais restritivo.

Agora, $T + T = 10 + O$, então $T = 5$.

Finalmente, $W + W + 1$ (carregado de $O + O$) deve ser igual a $10 + U$, então $W = 2$ e $U = 4$.

Portanto, a solução para o problema criptoaritmético é $T = 5, W = 2, O = 0, F = 1, U = 4, R = 0$.

9 - O que é Overfitting ? Como evitá-lo ?

Overfitting é um conceito em aprendizado de máquina que ocorre quando um modelo é treinado de tal forma que se ajusta muito bem aos dados de treinamento, mas não generaliza bem para dados novos ou não vistos. Em outras palavras, o modelo aprende o "ruído" nos dados de treinamento, resultando em um desempenho pobre quando aplicado a dados de teste ou a novos dados.

Overfitting pode ser causado por várias razões, incluindo a complexidade excessiva do modelo (por exemplo, muitos parâmetros em relação ao número de observações), treinamento por tempo demais, ou falta de dados de treinamento suficientes.

Existem várias estratégias para evitar o overfitting:

Validação Cruzada: Esta é uma técnica onde os dados de treinamento são divididos em vários subconjuntos. O modelo é então treinado em alguns desses subconjuntos e validado nos restantes. Isso ajuda a garantir que o modelo generalize bem para dados não vistos.

Regularização: A regularização adiciona uma penalidade à função de perda do modelo para evitar que os coeficientes se tornem muito grandes. Isso ajuda a prevenir o overfitting, reduzindo a complexidade do modelo.

Coleta de mais dados: Se possível, coletar mais dados de treinamento pode ajudar a melhorar a capacidade do modelo de generalizar.

Simplificar o modelo: Se o modelo for muito complexo (por exemplo, muitas camadas em uma rede neural), pode ser útil simplificar o modelo.

Early stopping: Durante o treinamento, monitorar o desempenho do modelo nos dados de validação. Se o desempenho começar a piorar, isso pode ser um sinal de overfitting e o treinamento pode ser interrompido.

Dropout: Em redes neurais, o dropout é uma técnica onde certos neurônios são "desligados" durante o treinamento. Isso ajuda a prevenir o overfitting, forçando a rede a aprender representações mais robustas.

Lembrando que a escolha da estratégia depende do tipo de modelo e do conjunto de dados específico com o qual você está trabalhando.

11 - (Aprendizado Indutivo) - Calcule o ganho de cada atributo sobre os 23 exemplos de treinamento

Atributo Chego_atrasado = {Muito, AsVezes, Nunca} Atributo Presto_atencao = {Muito, Pouco}

Atributo Saio C/ Duvida = {Sempre, Asvezes, Nunca} Atributo Conteudo = {Muito, Pouco}

O ganho de informação é uma métrica usada em aprendizado de máquina para selecionar os atributos que são mais úteis para a previsão. É baseado no conceito de entropia da teoria da informação.

Para calcular o ganho de informação para cada atributo, precisamos seguir os seguintes passos:

Calcule a entropia do conjunto de dados antes da divisão (Entropia(S)).

Para cada atributo:

Divida o conjunto de dados com base nos valores do atributo.

Calcule a entropia do conjunto de dados após a divisão (Entropia(S_v)) para cada valor v do atributo.

Calcule a entropia média ponderada após a divisão para o atributo.

Calcule o ganho de informação para o atributo como a diferença entre a entropia antes da divisão e a entropia média ponderada após a divisão.

No entanto, para calcular o ganho de informação, precisamos dos dados de treinamento, que não foram fornecidos na pergunta. Se você fornecer os dados de treinamento, posso ajudá-lo a calcular o ganho de informação para cada atributo.

Além disso, é importante notar que o ganho de informação favorece os atributos com muitos valores. Portanto, em alguns casos, pode ser mais apropriado usar uma métrica diferente, como o coeficiente de ganho, que normaliza o ganho de informação pelo número de valores do atributo.

12 – (Algoritmo Genético) – Explique como funciona o método da Roleta para seleção de soluções que irão fazer a reprodução em um Algoritmo Genético e responda, com base na tabela abaixo, que possui as soluções e as respectivas aptidões, qual seria a solução escolhida pelo método da roleta se o número aleatório escolhido for 51.

Solução

1

2

3

4

5

6

7

8

9

10

Aptidão

8

2

17

7

2

12

11

7

3

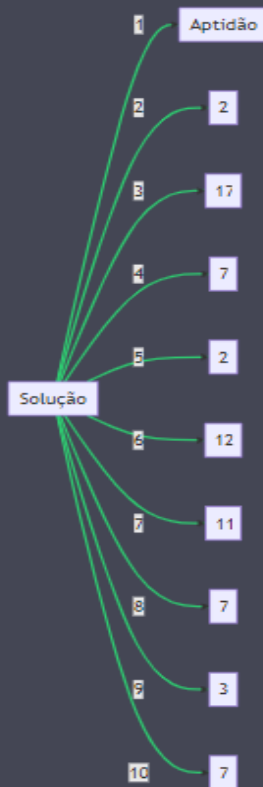
7

Me retorne uma tabela para melhorar a visualização da solução

O método da roleta é uma técnica de seleção usada em algoritmos genéticos. A ideia é dar a cada solução uma chance de ser selecionada que é proporcional à sua aptidão. Aqui está como funciona:

1. Calcule a aptidão total somando as aptidões de todas as soluções.
2. Gere um número aleatório entre 0 e a aptidão total.
3. Percorra as soluções, acumulando a aptidão de cada uma. Quando a soma acumulada for maior ou igual ao número aleatório, selecione essa solução.

Aqui está a tabela de soluções e aptidões que você forneceu:



A aptidão total é 76. Se o número aleatório escolhido for 51, começamos a acumular as aptidões das soluções. Quando chegamos à solução 6, a soma acumulada é 46. Quando adicionamos a aptidão da solução 7 (11), a soma acumulada é 57, que é maior que 51. Portanto, a solução 7 seria a escolhida pelo método da roleta.

13 – (Algoritmo Genético) – Explique o que é elitismo e o que é mutação.

Elitismo e mutação são dois conceitos importantes em algoritmos genéticos.

Elitismo: O elitismo é uma técnica usada em algoritmos genéticos para garantir que as melhores soluções (ou indivíduos) de uma geração sejam preservadas para a próxima geração. Isso é feito selecionando um pequeno número dos melhores indivíduos e os passando diretamente para a próxima geração, sem alterá-los. O elitismo ajuda a garantir que a qualidade da população não diminua com o tempo e que a melhor solução encontrada até agora não seja perdida.

Mutação: A mutação é um operador genético usado para manter e introduzir diversidade na população genética. Durante a mutação, os genes individuais de um cromossomo são alterados. Por exemplo, em uma representação binária, um bit pode ser invertido de 0 para 1 ou de 1 para 0. A mutação é importante porque ajuda a explorar novas áreas do espaço de soluções e evita que o algoritmo fique preso em mínimos locais. No entanto, a taxa de mutação geralmente é mantida baixa para evitar a introdução de demasiada aleatoriedade no processo de busca.

14 – (sistema fuzzy) Aplicando um sistema fuzzy com as características altura, habilidade e probabilidade de se tornar jogador, abaixo, pergunta-se: Uma pessoa Baixa e habilidosa poderia ser um jogador de futebol ? Considere as funções de pertinências a seguir:

Altura Habilidade Probabilidade de ser jogador

Baixo Médio Alto

Sofrível Regular Habilidoso

baixa média alta

1,4 1,6 1,8 2,0 (metros)

Sabe-se que as seguintes regras são válidas:

2 4 6 8 10 (Nota)

2 4 6 8 10 (x10%)

- Se pessoa for baixa E for Habilidoso, então tem alta probabilidade de ser jogador

- Se pessoa for alta E for Regular, então tem média probabilidade de ser jogador

- Se pessoa for baixa OU for Sofrível, então tem baixa probabilidade de ser jogador

Considerando as três regras acima, além das funções de pertinência dadas, determine qual deve ser a PROBABILIDADE DE UMA PESSOA, sabendo que ela tem a altura de 1,50 m e ganhou nota 3 de habilidade quando jogou um rachaço no campo da FEI, em ser um jogador de futebol.

Um sistema fuzzy é uma forma de lógica que lida com raciocínio aproximado, em vez de precisamente fixado. É uma extensão da lógica booleana que lida com conceitos que podem ser em algum grau verdadeiros ou falsos, em vez de completamente verdadeiros ou completamente falsos.

No contexto deste problema, podemos usar a lógica fuzzy para determinar a probabilidade de uma pessoa se tornar um jogador de futebol com base em sua altura e habilidade.

Dadas as regras e funções de pertinência, podemos proceder da seguinte maneira:

Avalie a pertinência da altura e habilidade da pessoa nas funções de pertinência dadas. Por exemplo, uma pessoa com 1,50 m de altura seria considerada "baixa" e uma pessoa com nota de habilidade 3 seria considerada "sofrível".

Aplicar as regras fuzzy. No caso da primeira regra, a pessoa é baixa e habilidosa, então ela tem alta probabilidade de ser um jogador. No entanto, a pessoa em questão não é habilidosa, então essa regra não se aplica.

A terceira regra se aplica porque a pessoa é baixa ou sofrível (neste caso, ambas). Portanto, a pessoa tem baixa probabilidade de ser um jogador de futebol.

Portanto, com base nas regras e funções de pertinência dadas, a probabilidade de uma pessoa com 1,50 m de altura e nota de habilidade 3 se tornar um jogador de futebol é baixa. No entanto, isso é apenas uma avaliação baseada em um modelo fuzzy simplificado e pode não refletir a realidade complexa de se tornar um jogador de futebol profissional.

17 – (Colônia de Formigas) - Quando a formiguinha, em um algoritmo de colônia de formigas, chega em um ponto onde ela deverá escolher um caminho a seguir, como ela irá decidir qual caminho seguir , se ela encontra 4 possibilidades sendo a primeira com 80 de feromônio, a segunda com 34, a terceira com 15 e a quarta com 20 ?

Em um algoritmo de colônia de formigas, a decisão de uma formiga sobre qual caminho seguir é geralmente baseada na quantidade de feromônio presente em cada caminho. O feromônio é uma substância química depositada por formigas anteriores que seguiram aquele caminho. Quanto mais feromônio em um caminho, mais atraente ele é para as formigas subsequentes.

No entanto, a decisão não é determinística, mas probabilística. Isso significa que a formiga não escolhe necessariamente o caminho com a maior quantidade de feromônio, mas a probabilidade de escolher um caminho é proporcional à quantidade de feromônio nele.

A probabilidade P_{ij} de uma formiga escolher o caminho j a partir do nó i é geralmente calculada usando a seguinte fórmula:

$$P_{ij} = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{k \in N_i} (\tau_{ik})^\alpha (\eta_{ik})^\beta}$$

onde:

- τ_{ij} é a quantidade de feromônio no caminho de i para j ,
- η_{ij} é a visibilidade (ou atratividade) do caminho de i para j ,
- N_i é o conjunto de nós que podem ser visitados a partir do nó i ,
- α e β são parâmetros que controlam a importância relativa do feromônio e da visibilidade.

No seu caso, se considerarmos que a visibilidade é a mesma para todos os caminhos, a formiga escolherá o primeiro caminho com a maior probabilidade, pois é o caminho com a maior quantidade de feromônio (80). No entanto, ainda há uma chance de que a formiga escolha um dos outros caminhos.

18 – (CBR/RBC) Quais são os passos de um Sistema de Raciocínio Baseado em Casos ? Explique cada um.

O Raciocínio Baseado em Casos (RBC), ou Case-Based Reasoning (CBR) em inglês, é uma abordagem de inteligência artificial que envolve a solução de novos problemas com base na solução de problemas anteriores semelhantes. O RBC passa por quatro etapas principais, conhecidas como os 4Rs do RBC:

Recuperação: Nesta etapa, o sistema procura em sua base de dados de casos anteriores para encontrar os casos que são mais semelhantes ao novo problema. A semelhança é geralmente determinada usando uma medida de distância, como a distância euclidiana ou a distância de Manhattan.

Reutilização: Uma vez que um ou mais casos semelhantes são recuperados, o sistema tenta adaptar a solução desses casos para resolver o novo problema. Isso pode envolver a modificação da solução do caso antigo para se adequar ao novo problema.

Revisão: A solução proposta é então testada para verificar se é válida. Isso pode envolver a execução da solução em um ambiente de teste ou a verificação da solução com um especialista humano.

Retenção: Se a solução for bem-sucedida, o novo problema e sua solução são armazenados na base de dados de casos para uso futuro. Isso permite que o sistema aprenda com a experiência e melhore suas soluções ao longo do tempo.

19 – (Aprendizado por reforço) Qual a importância do método ϵ -Greedy para o sucesso de um sistema de Aprendizado por Reforço ?

O método ϵ -Greedy é uma estratégia de equilíbrio entre exploração e exploração usada em muitos algoritmos de aprendizado por reforço. A importância deste método reside na sua simplicidade e eficácia em lidar com o dilema exploração-exploração.

No contexto do aprendizado por reforço, "exploração" refere-se à ação de tentar ações diferentes para descobrir quais delas podem levar a melhores recompensas no futuro. Por outro lado, "exploração" refere-se à ação de escolher a ação que se acredita ser a melhor com base na informação disponível.

O método ϵ -Greedy aborda este dilema da seguinte maneira: a maior parte do tempo ($1 - \epsilon$), o agente escolhe a ação que ele acredita ter a maior recompensa esperada (exploração). No entanto, uma pequena fração do tempo (ϵ), o agente escolhe uma ação aleatória (exploração).

A importância do método ϵ -Greedy é que ele permite que o agente continue aprendendo sobre o ambiente, tentando novas ações e descobrindo possivelmente melhores estratégias. Ao mesmo tempo, o agente também aproveita o conhecimento que já adquiriu para obter recompensas. A escolha do valor de ϵ pode ter um grande impacto no desempenho do agente e geralmente requer algum ajuste.

20 – (Aprendizado por reforço) Explique, pela formula abaixo, como o valor de Q é atualizado:

$$Q(st,a) = r(st,a) + \gamma \max_{a'} [Q(st+1,a')]$$

Please write in Portuguese language.

A fórmula que você forneceu é uma equação fundamental em muitos algoritmos de aprendizado por reforço, conhecida como equação de atualização Q-Learning.

Vamos quebrar cada componente da equação:

- $Q(st, a)$: Esta é a função de valor de ação, que estima o valor esperado de tomar uma ação a no estado st .
- $r(st, a)$: Esta é a recompensa imediata recebida após tomar a ação a no estado st .
- γ : Este é o fator de desconto, que determina a importância das recompensas futuras em comparação com as recompensas imediatas. Um valor de γ próximo a 0 fará com que o agente se concentre principalmente nas recompensas imediatas, enquanto um valor de γ próximo a 1 fará com que o agente leve em conta as recompensas futuras.
- $\max_{a'} [Q(st + 1, a')]$: Este é o valor máximo esperado de todas as possíveis ações a' no próximo estado $st + 1$.

A equação de atualização Q-Learning é usada para atualizar a estimativa do agente do valor de tomar a ação a no estado st . A nova estimativa é uma combinação da recompensa imediata $r(st, a)$ e do valor descontado da melhor ação possível no próximo estado $\gamma \max_{a'} [Q(st + 1, a')]$.

Em outras palavras, o agente está tentando aprender a ação que maximiza a soma das recompensas futuras, descontadas de volta ao presente.