

# Computação Natural

## Inteligência Coletiva

1. O algoritmo de colônia de formigas realiza uma busca local no espaço de soluções através do depósito de feromônio.

A Busca realizada é global. O feromônio depositado nos caminhos sub-ótimos tende a desaparecer em detrimento do melhor caminho com as iterações.

2. No algoritmo de colônias de formigas, as soluções são construídas probabilisticamente considerando a quantidade de feromônio associada a cada solução candidata.

Cada formiga decide o próximo vértice visitado de forma probabilística, considerando a quantidade de feromônio nas arestas possíveis de serem percorridas.

A 2 falta a desejabilidade, a escolha é tomada com desej. + feromônio, não só o feromônio. (Verdade mas incompleta)

3. O feromônio associado a cada aresta é atualizado logo após a finalização da construção da solução, de acordo com a qualidade da solução final.

Após a construção de um caminho, a regra delta de atualização leva em consideração a desejabilidade do caminho. Quanto melhor, mais feromônio depositado.

4. Os algoritmos de enxames de partículas se baseiam na ideia de avaliar o comportamento dos vizinhos e imitá-los.

Essa imitação é traduzida matematicamente no termo global da atualização da velocidade.

5. Cada partícula no PSO representa uma solução para um problema, associada a uma posição no espaço e uma velocidade.

A busca realizada pelo PSO é justamente atualizando a velocidade e a posição, a cada iteração, em busca de uma melhora.

6. A partícula no algoritmo de PSO aprende atualizando sua posição no espaço de busca.

Vale ressaltar que além da própria posição, a posição dos vizinhos também é considerada. (V mas incompleta)

7. A velocidade de uma partícula é atualizada de acordo com sua melhor posição até o momento e a posição das partículas da vizinhança.

Estes são os fatores cognitivos e social, respectivamente

8. A vizinhança mais comum no PSO é a geográfica, que pode, por exemplo, ser calculada através da distância Euclidiana.

Utiliza-se mais comumente as vizinhanças do tipo estrela, anel e focal

9. A atualização da velocidade depende tanto da memória da melhor posição do indivíduo quanto do melhor vizinho.

Estes são os fatores cognitivos e social, respectivamente

10. O fator de inércia no PSO serve para convergência da partícula.

A inércia é responsável por manter o movimento das partículas.

11. O exploration e exploitation no PSO são controlados pela velocidade.

Exploration e exploitation são controlados pelas constantes  $C1$  e  $C2$  e, até certo ponto, pelas taxas de aprendizado.

## Redes Neurais

12. O perceptron foi a primeira rede a conseguir resolver o problema do XOR, seguido das RBF.

Não faz separações de forma a resolver o XOR.

O perceptron não tem capacidade de realizar separações de forma a resolver o problema XOR.

13. O treinamento de uma rede perceptron de múltiplas camadas pode ser visto como um problema de encontrar os vetores de peso da rede que levem ao menor erro nos exemplos de treinamento.

Essa busca é exatamente o treinamento (backpropagation, por exemplo), que atualiza os pesos buscando minimizar a função de perda / erro.

14. Pode-se dizer que uma rede sofre de overfitting quando o erro do treinamento é alto e o erro do teste é baixo.

O erro de treino é baixo.

O overfitting pode ser considerado como uma "especialização" do algoritmo nos dados de treino, resultando numa acurácia altíssima nestes, em detrimento da capacidade de generalização, o que se manifesta como uma baixa acurácia nos testes.

15. O treinamento da rede em batch só atualiza os pesos da rede depois de passar por todos os exemplos. Nele, a ordem dos exemplos pode afetar o aprendizado.

Como não há atualização entre os exemplos, a ordem dos dados dentro de um mesmo batch (ou mini-batch) não importa.

16. Redes RBF são redes supervisionadas que aprendem mais rápido que as MLP.

Por possuírem estrutura fixa e utilizarem kernels (funções) ao invés de pesos, as redes RBF precisam fazer menos atualizações a cada iteração, tornando-se mais rápidas.

17. A arquitetura da rede RBF é composta por 3 camadas, sendo uma linear seguida de uma não linear. Os pesos da primeira camada são ajustados com um algoritmo do tipo K-means.

É o contrário, uma não linear seguida de uma linear.

A camada de entrada não possui pesos, somente utilizam kernels para medir a distância do ponto para o centro do kernel do neurônio.

18. Nas redes que utilizavam funções do tipo RBF, quanto mais próximo do centro da função, maior a influência de um exemplo no aprendizado.

19. Não é possível treinar uma rede RBF apenas com a descida do gradiente.

É possível e é feito em apenas uma fase

A descida do gradiente é um dos algoritmos possíveis para treinamento de uma rede BRF, funciona em uma fase única.

A descida de gradiente é uma das três formas mais comuns de treinamento dos RBFs, juntamente com K-means e centros aleatórios.

20. Verdadeiro. A função já se explica.

20. Redes SOM são redes não supervisionadas cujo aprendizado ocorre em 3 fases: competição, cooperação e adaptação.

21. Redes RBF criam um mapeamento do espaço original para um grid 2D, que é uma aproximação do espaço original.

Quem faz isso é o SOM

Esta é a descrição da rede SOM e não de um RBF.

22. Uma das maiores aplicações de redes SOM é em tarefas de visualização de dados de alta dimensão.

Por mapear dados de alta dimensionalidade para espaços 1D e 2D, as redes SOM facilitam a visualização, e por isso sua popularidade.

23. Uma rede neural com uma camada e um número suficientemente grande de neurônios na camada escondida consegue representar qualquer função com um grau de acurácia arbitrário.

A utilização de camadas extras ou mais neurônios numa camada oculta (ou seja, o aumento do número total de neurônios) torna possível a aproximação de funções mais complexas, incluindo as não lineares.

24. O uso da função de ativação ReLU em redes profundas torna o processo de aprendizado mais rápido.

A função ReLU é muito mais fácil de calcular e derivar que as funções sigmóide e tangente hiperbólica.