Lista Teórica 3 - Introdução a Aprendizagem Profunda

March 10, 2025

Questão 1

Descreva o conceito de vanishing gradient problem nas redes neurais recorrentes (RNNs). Como esse problema impacta o treinamento dessas redes? Quais são as soluções propostas para mitigar esse efeito? Explique e compare técnicas como Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU).

Questão 2

As redes neurais recorrentes são amplamente utilizadas para modelar sequências temporais. Compare a arquitetura de uma RNN tradicional, uma LSTM e uma GRU, destacando suas diferenças estruturais e vantagens/desvantagens. Ilustre sua resposta com diagramas dessas redes.

Questão 3

Você está desenvolvendo um modelo de previsão de séries temporais para prever a demanda de energia elétrica de uma cidade.

1. Qual arquitetura de RNN você utilizaria? Justifique sua escolha.
2. Como organizaria os dados de entrada e saída para treinar esse modelo?
3. Como validaria a qualidade das previsões feitas pelo modelo?

Esboce um fluxo de dados para representar o processo de treinamento e inferência desse sistema.

Questão 4

Explique a diferença entre Backpropagation Through Time (BPTT) e Truncated Backpropagation Through Time (TBPTT). Em quais situações cada uma dessas técnicas é mais adequada? Qual a importância cada uma? Apresente exemplos práticos de sua aplicação.

Questão 5

Qual o funcionamento de uma Rede Neural Recorrente Bidirecional (Bi-RNN) e como ela difere de uma RNN unidirecional? Em quais situações o uso de uma Bi-RNN é mais vantajoso? Dê um exemplo de aplicação prática.

Questão 6

Em uma **Rede Neural Recorrente (RNN) simples**, a saída h_t em um determinado instante de tempo t é calculada pela seguinte equação:

$$h_t = \tanh(W_h x_t + U_h h_{t-1} + b_h)$$

onde:

- x_t é a entrada no tempo t,
- h_{t-1} é o estado oculto do tempo anterior,
- W_h e U_h são os pesos da rede,
- b_h é o viés (bias),
- $\tanh(x)$ é a função de ativação, dada por:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Agora, suponha os seguintes valores:

- **Entrada**: $x_t = 0.8$
- **Estado oculto anterior**: $h_{t-1} = 0.4$
- **Pesos e bias**:
  - $W_h = 0.6$
  - $U_h = 0.5$
  - $b_h = 0.2$

Pergunta: Calcule o valor da **saída h_t da RNN** no instante t, mostrando o passo a passo do cálculo.

Questão 7

O conceito de Attention Mechanism tem sido amplamente utilizado em redes neurais modernas, especialmente em arquiteturas como Transformers. Explique o funcionamento desse mecanismo, sua importância no processamento de sequências e como ele difere das redes neurais recorrentes tradicionais. Apresente um exemplo prático de aplicação.

Bônus

Questão 8

O mecanismo de **Self-Attention** utilizado em modelos como Transformers para atribuir pesos diferentes às palavras de uma sequência com base na importância relativa entre elas. Uma das formas de medir essa importância é através da **similaridade do cosseno**, que quantifica a relação entre dois vetores.

A **similaridade do cosseno** entre dois vetores A e B é dada pela equação:

$$\cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

onde:

- $A \cdot B$ é o produto escalar entre os vetores A e B,
- $\|A\|$ e $\|B\|$ são as normas dos vetores A e B, calculadas como:

$$\|A\| = \sqrt{A_1^2 + A_2^2 + \cdots + A_n^2}, \quad \|B\| = \sqrt{B_1^2 + B_2^2 + \cdots + B_n^2}$$

Agora, considere os seguintes vetores representando palavras em um espaço vetorial:

$$A = (2, 3, 4), \quad B = (1, 0, 5)$$

Perguntas:

1. Calcule a **similaridade do cosseno** entre os vetores A e B.
2. Com base no valor encontrado, os vetores A e B estão mais próximos ou distantes no espaço vetorial? Explique o que isso significa no contexto de Self-Attention.

Mostre todos os cálculos passo a passo.

Questão 9

Os modelos de **Self-Attention**, como o utilizado no Transformer, funcionam atribuindo pesos diferentes às partes da entrada. Para isso, cada entrada é transformada em três matrizes principais: **Query (Q)**, **Key (K)** e **Value (V)**.

- **Query (Q)**: Representa a consulta sobre qual informação deve ser extraída.
- **Key (K)**: Representa as referências que serão comparadas com a Query para calcular a atenção.
- **Value (V)**: Contém as informações reais que serão combinadas com os pesos de atenção.

O processo de atenção é calculado pela seguinte equação:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

onde d_k é a dimensão dos vetores Key.

Perguntas:

1. Explique, em suas próprias palavras, o papel de **Query (Q)**, **Key (K)** e **Value (V)** no mecanismo de Self-Attention.
2. Por que a matriz **Key (K)** é multiplicada pela **Query (Q)** antes da aplicação da Softmax?
3. Qual o impacto de um valor alto ou baixo na matriz de atenção gerada?