

Relatório CARD 26 - Prática: Modelos Generativos (III)

Willian Augusto Soder de Souza

O objetivo deste relatório é explicar os principais conhecimentos adquiridos ao assistir aos dois cursos indicados, que abordam modelos generativos. O primeiro curso, 'Machine Learning, Data Science and Generative AI with Python', disponível na Udemy, apresenta, na seção sobre modelos generativos, uma explicação equilibrada entre teoria e prática, destacando os conceitos mais importantes desses modelos. Já o segundo curso, 'Let's build GPT: from scratch, in code, spelled out', disponível no YouTube, tem um enfoque mais prático, mas também aborda o contexto teórico. A seguir, apresento um resumo dos principais conceitos discutidos em ambos os cursos.

- **Modelos Generativos:** são algoritmos de aprendizado de máquina que visam gerar novos dados semelhantes aos dados nos quais foram treinados. Eles aprendem a capturar a distribuição dos dados de entrada e, a partir disso, conseguem criar exemplos novos que seguem as mesmas características. Esses modelos são amplamente utilizados em tarefas como a geração de imagens, texto e áudio. Funcionam, de maneira geral, aprendendo as relações e padrões nos dados por meio de técnicas como redes generativas adversariais ou modelos baseados em probabilidade.
- **GPT-2/GPT-3:** são modelos de linguagem desenvolvidos pela OpenAI, baseados na arquitetura de Transformer. Esses modelos são treinados com grandes volumes de texto para prever a próxima palavra em uma sequência, o que lhes permite gerar textos coerentes e contextualmente relevantes. O GPT-3 possui aproximadamente 100 vezes mais parâmetros do que o GPT-2. Isso faz com que o GPT-3 tenha uma capacidade muito mais avançada de compreensão e geração de texto, permitindo-lhe realizar tarefas mais complexas.
- **Transformers:** são uma arquitetura de deep learning que utilizam o 'mecanismo de atenção' para processar sequências de dados de forma mais eficiente. Em modelos generativos, como GPT, eles permitem prever a próxima palavra em uma sequência, gerando textos realistas e coerentes.
- **Self Attention:** é um mecanismo utilizado em arquiteturas de Transformer que permite que cada palavra em uma sequência avalie e ajuste sua relevância em relação a todas as outras palavras da mesma sequência. Ele usa vetores de queries, keys e values para calcular a atenção e criar representações ajustadas, melhorando a compreensão do contexto e as relações entre palavras.
- **Masked Self Attention:** é uma variante do mecanismo de Self-Attention usada principalmente em modelos de linguagem. Em vez de permitir que cada palavra olhe para todas as palavras na sequência, a Masked Self-Attention restringe a atenção de cada palavra para apenas as palavras anteriores na sequência.
- **Multi-Headed Self Attention:** é uma técnica usada em Transformers onde o mecanismo de Self-Attention é dividido em múltiplas "cabeças" ou subespaços. Cada cabeça realiza a atenção de forma independente, permitindo ao modelo capturar diferentes aspectos e relações contextuais da sequência. Os resultados dessas cabeças são então combinados, proporcionando uma representação mais rica e abrangente da sequência.
- **Encoders/Decoders:** são partes essenciais de modelos de Transformer. O Encoder recebe e processa a entrada, transformando-a em uma representação numérica que captura suas

características e contexto. Já o Decoder utiliza essa representação para gerar a saída, convertendo-a em uma sequência final, como texto traduzido ou gerado em formato de string.

- **Embedding:** é uma técnica que converte dados categóricos, como palavras, em vetores numéricos densos. Esses vetores capturam semântica e relações entre os dados, facilitando o processamento e a análise em modelos de aprendizado de máquina.
- **Reinforcement Learning:** é uma área do aprendizado de máquina onde um agente aprende a tomar decisões através de interações com um ambiente. O agente realiza ações e recebe recompensas ou penalidades com base nas suas escolhas, ajustando seu comportamento para maximizar a recompensa acumulada ao longo do tempo. O processo envolve a exploração de diferentes ações para descobrir quais trazem os melhores resultados e a exploração de ações conhecidas para melhorar a performance.
- **Proximal policy gradients:** são algoritmos de aprendizado por reforço que otimizam políticas de decisão de forma estável e eficiente, garantindo que as atualizações na política não sejam muito grandes. Isso é feito limitando o tamanho das mudanças para evitar alterações drásticas que possam prejudicar o desempenho do agente. Um exemplo de algoritmo baseado em PPG é o Proximal Policy Optimization (PPO).
- **Bigram Model:** é um modelo de linguagem que prevê a probabilidade de uma palavra com base na palavra imediatamente anterior a ela. Ele utiliza pares de palavras (bigrams) para estimar a probabilidade de uma sequência de palavras, simplificando a previsão de palavras subsequentes ao considerar apenas o contexto da palavra anterior. Isso é útil para tarefas como geração de texto e correção automática.
- **Pre-Training:** é o processo inicial de treinamento de um modelo em um grande conjunto de dados para aprender padrões e representações gerais. Ele fornece ao modelo um conhecimento básico e abrangente antes de ser ajustado para tarefas específicas.
- **Fine-Tuning:** é o ajuste final de um modelo pré-treinado em um conjunto de dados específico para melhorar seu desempenho em uma tarefa particular.

CONCLUSÃO:

Compreender modelos generativos e seus conceitos fundamentais é crucial para avançar no campo da inteligência artificial e aprendizado de máquina. Estes modelos, que incluem técnicas como Self-Attention, Transformers e embeddings, oferecem ferramentas poderosas para gerar e manipular dados de forma inovadora. O conhecimento profundo sobre esses conceitos permite não apenas o desenvolvimento de aplicações avançadas, como geração de texto e tradução automática, mas também proporciona uma base sólida para a resolução de problemas complexos e a criação de soluções personalizadas. A habilidade de aplicar e adaptar modelos generativos, desde o pre-training até o fine-tuning, é essencial para otimizar o desempenho em tarefas específicas e explorar todo o potencial dessas tecnologias emergentes, que têm se destacado nos últimos anos, como, por exemplo, o Chat GPT da OpenAI.