

Os *Transformers* (especificamente a autoatenção) impulsionaram um progresso recente significativo em *PNL*. Eles permitiram que modelos como *BERT*, *GPT-3* formassem modelos de linguagem poderosos que podem ser usados para gerar texto, traduzir texto, responder perguntas, classificar documentos, resumir texto e afim.

Os transformers mostraram grande capacidade de modelagem para dependências e interações de longo alcance em dados sequenciais e, portanto, estão aparecendo na modelagem de séries temporais. Muitas variantes do Transformer foram propostas para abordar desafios especiais na modelagem de séries temporais e foram aplicadas com sucesso a várias tarefas de séries temporais, como previsão [Liet al., 2019; Zhou et al., 2021; Zhou et al., 2022], detecção de anomalias [Xu et al., 2022; Tuli et al., 2022], classificação [Zerveaset al., 2021; Yanget al., 2021], e assim por diante [artigo, p.1].

Apesar do sucesso do *deep learning* em visão computacional, muitos modelos de séries temporais ainda são superficiais. Particularmente, na indústria, muitos cientistas de dados ainda utilizam modelos autoregressivos simples em vez de *deep learning*.

1.1 TRABALHOS

A rede neural *LSTM* é um dos primeiros [trabalhos](#) que propõe o uso de um mecanismo *LSTM* + *attention* para séries temporais de previsão multivariada. A autoatenção revela alguns problemas em relação a séries temporais, um deles é que a chave/valor geralmente podem vir de fontes diferentes e variam de acordo com a camada do codificador e decodificador, outro é que a autoatenção geralmente requer codificações posicionais, pois não tem conhecimento da ordem da sequência. Além disso, esse [artigo](#) demonstra que, o mecanismo de *attention* se especializa em atender o token anterior e seguinte, enquanto o token atual tem pouca atenção, o que é problemático para séries temporais, em certas sequências de séries temporais, a causalidade pode vir de etapas muito mais antigas.

1 RESPOSTA

1.2 VANILLA TRANSFORMER

Ao contrário do LSTM ou RNN, o Transformer não tem recorrência e nem convolução. Em vez disso, ele utiliza a codificação posicional adicionada nos embeddings de entrada para modelar as informações de sequência. Um Transformer vanilla [Vaswani et al., 2017] segue os modelos de sequência neural mais competitivos com uma estrutura de codificador-decodificador. Tanto o codificador quanto o decodificador são compostos de vários blocos idênticos. Cada bloco do codificador consiste em um módulo de *multi-head self-attention* e uma *feed-forward network* (FFN) enquanto cada bloco de decodificador insere modelos de *cross-attention* entre as várias *multi-head self-attention* e a *feed-forward network* (FFN).

1.3 INPUT ENCODING AND POSITIONAL ENCODING

Alguns trabalhos [Liet et al., 2019] simplesmente introduzem a codificação posicional vanilla (Seção 2.2) usada em [Vaswani et al., 2017], que é então adicionada às incorporações de séries temporais de entrada e alimentadas no Transformer. Embora esse aplicativo simples possa extrair algumas informações posicionais de séries temporais, eles não conseguiram explorar totalmente os recursos importantes dos dados de séries temporais.

1.4 LEARNABLE POSITIONAL ENCODING

Como a codificação posicional vanilla é artesanal e menos expressiva e adaptativa, vários estudos descobriram que aprender os encaixes de posição apropriados a partir de dados de séries temporais pode ser muito mais eficaz. Em comparação com a fixed vanilla positional encoding, as incorporações aprendidas são mais

flexíveis e podem se adaptar a tarefas específicas. [Zerveaset al., 2021] introduz uma camada de incorporação no Transformer que aprende vetores de incorporação para cada índice de posição em conjunto com outros parâmetros do modelo.[Limetal., 2019] usaram uma rede LSTM para codificar embeddings posicionais, com o objetivo de melhor explorar informações de ordenação sequencial em séries temporais.

1.5 TIMESTAMP ENCODING

Ao modelar séries temporais em cenários do mundo real, as informações de *timestamps* de data/hora são comumente acessíveis, incluindo *timestamps* de data/hora do calendário (por exemplo, segundo, minuto, hora, semana, mês e ano) e carimbos de data/hora especiais (por exemplo, feriados e eventos) . Esses timestamps são bastante informativos em aplicações reais, mas dificilmente são aproveitados em vanilla Transformers. Para mitigar o problema, [Zhouet al.,2021] propôs codificar carimbos de data/hora como codificação de posição adicional usando camadas de incorporação apreensíveis. Um esquema de codificação de carimbo de data e hora semelhante foi usado no Autoformer [Wuetal., 2021] e FEDformer [Zhouet al., 2022].

2 MÓDULO DE AUTOATENÇÃO

A central de um Transformer é o módulo de auto-atenção. Ele pode ser visto como uma camada totalmente conectada com os pesos que são gerados dinamicamente com base na semelhança de pares de padrões de entrada. Como resultado, ele compartilha o mesmo comprimento de caminho máximo que camadas totalmente conectadas, mas com um número muito menor de parâmetros, tornando-o adequado para modelar dependências de longo prazo. O transformador tem uma complexidade de tempo e memória de $O(L^2)$ — onde o L é o comprimento da série

temporal de entrada — que se torna o gargalo computacional ao lidar com sequências longas.

3 APLICAÇÕES DO TRANSFORMADOR DE SÉRIE TEMPORAL

- PREVISÃO ESPACIAL TEMPORAL
 - Necessário reduzir a complexidade computacional;
 - Introduzir sparse bias;
 - Modificar o módulo de autoatenção;
- PREVISÃO DE SÉRIES TEMPORAIS
 - Dependência temporal e espaço-temporal;
 - Necessário que o encoder-decoder tenha um módulo self-attention para capturar dependências temporais em um módulo Graph Neural Network para capturar dependências espaciais;
 - Rede de convolução de grafos;
 - Uma arquitetura de decomposição de tendência de média móvel proposta em [Wu et al., 2021] para testar vários módulos de atenção, pode aumentar significativamente o desempenho do modelo em 50% a 80%.
- PREVISÃO DE EVENTOS
 - A previsão ou predição de eventos visa prever os tempos e marcas de eventos futuros dada a história de eventos passados, e muitas vezes é modelada por temporal point processes (TPP);
 - Arquitetura: alteração na codificação posicional traduzindo intervalos de tempo em função senoidal de modo que os intervalos entre eventos possam ser utilizados;
- TRANSFORMADORES NA DETECÇÃO DE ANOMALIAS

- [Menget al., 2019] revelou a vantagem de usar o Transformer para detecção de anomalias sobre outros modelos tradicionais de dependência temporal (por exemplo, LSTM). Além de uma maior qualidade de detecção, a detecção de anomalia baseada em transformador é significativamente mais eficiente do que os métodos baseados em LSTM, principalmente devido à computação paralela na arquitetura Transformer.

4 ANÁLISE

Conforme [WEN al., 2022] quando comparamos os resultados da previsão com o prolongamento do comprimento de entrada, vários modelos baseados em *Transformer* se deterioram rapidamente. Esse fenômeno torna muitos *transformers* cuidadosamente projetados impraticáveis em tarefas de previsão de longo prazo, uma vez que eles não podem utilizar efetivamente informações de entrada longas.

4 CONCLUSÃO

Alguns desses problemas se devem ao fato de o *Bert* ter sido treinado numa base linguística e outros devido a sua arquitetura. Nesse outro [artigo](#), os autores sugerem um *framework* pré-treinado utilizando como base, dados de séries temporais. Esse outro [artigo](#), propõe uma abordagem de *transformer* customizável com algumas modificações na sequência e ordem da autoatenção. Apesar de vários estudos, ainda é complicado aplicar séries temporais ao mecanismo de autoatenção. A Google tem [desenvolvido](#) um novo transformer, o [Reformer](#), que serve inclusive para séries temporais, entretanto, não há muitos estudos em relação a ele. Há outros *frameworks* como o [Flow-Forecast](#), mas não encontrei artigos de sua aplicação.

[Wen al, 2022] *Transformers* pré-treinados para séries temporais Modelos de transformadores pré-treinados em grande escala aumentaram significativamente o desempenho para várias tarefas em NLP [Kenton e Toutanova, 2019; Brown et al., 2020] e CV [Chen et al., 2021a]. No entanto, existem trabalhos limitados sobre *Transformers* pré-treinados para séries temporais, e os estudos existentes focam principalmente na classificação de séries temporais [Zerveaset al., 2021; Yanget al., 2021]. Portanto, como desenvolver modelos de Transformadores pré-treinados apropriados para diferentes tarefas em séries temporais ainda precisa ser examinado no futuro.

Portanto, observa-se que os modelos de *Transformers* tem melhorado o desempenho das previsões de séries temporais e séries espaciais. No entanto, identifica-se que muitos problemas ainda são enfrentados. Alguns dos trabalhos apresentados tiveram trabalhos exitosos para análise de séries temporais utilizando, na sua maioria, vanilla transformers modificados para ter melhor desempenho com sua predição de séries temporais. Logo, é uma linha de estudos promissora, no entanto, ainda não há modelos pré-treinados estáveis para séries temporais que resolvem muitos problemas, tendo, dessa maneira, que trabalhar a arquitetura da rede neural para ajustar especificamente para séries temporais.

Transformers in Time Series: A Survey <<https://arxiv.org/pdf/2202.07125.pdf>>

<https://ai.googleblog.com/2020/01/reformer-efficient-transformer.html>

<https://openreview.net/pdf?id=rkgNKkHtvB>

<https://github.com/AIStream-Peelout/flow-forecast>

<https://arxiv.org/pdf/2001.08317.pdf>

<https://www.iccs-meeting.org/archive/iccs2021/papers/127430203.pdf>

<https://arxiv.org/pdf/1906.04341.pdf>

<https://arxiv.org/pdf/1809.04206.pdf>

<https://towardsdatascience.com/attention-for-time-series-classification-and-forecasting-261723e0006d>