



ESCOLA REGIONAL DE INFORMÁTICA DO ESPÍRITO SANTO

WORKSHOP: INTRODUÇÃO À ARQUITETURA
TRANSFORMER E VISION TRANSFORMER PARA
APLICAÇÕES EM SAÚDE

Realização:



GOVERNO DO ESTADO
DO ESPÍRITO SANTO
Secretaria de Ciência, Tecnologia,
Inovação e Educação Profissional

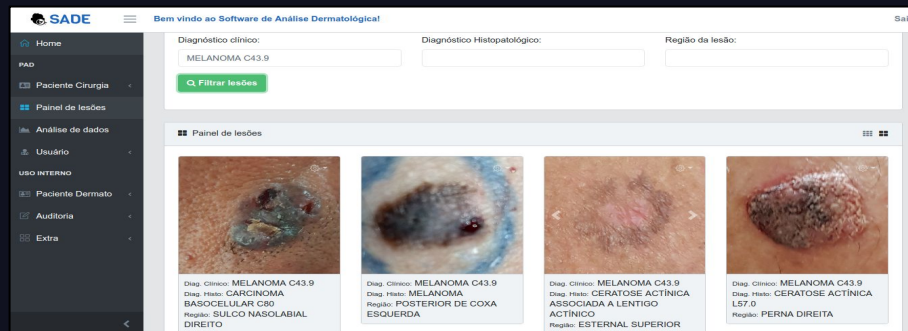
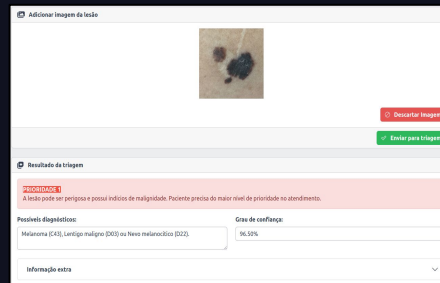


Apresentação

- Mestrando em Informática (2024-Atual) - UFES
- Membro do Programa de Assistência Dermatológica da UFES
- Graduado em Engenharia de Controle e Automação (2014-2019) - UFLA
- Contato:
 - pedro.bouzon@edu.ufes.br
 - www.linkedin.com/in/pedro-bouzon/

Áreas de atuação

- Machine learning
- Deep Learning
- Medical imaging
- Medical datasets



Informações sobre o Workshop

Informações sobre o Workshop

- **Tópico:** Introdução à arquitetura Transformers e Visual Transformers para aplicações em saúde
- **Objetivos:**
 - Apresentar a arquitetura do Transformer e implementar cada uma de suas partes.
 - Apresentar o Bidirectional Encoder Representations from Transformers (BERT).
 - **Aplicação 1:** classificação de especialidade médica a partir de transcrições médicas.
 - Introduzir a arquitetura do Vision Transformer.
 - **Aplicação 2:** classificação de câncer de pele.
 - Propor um desafio.

Informações sobre o Workshop

- **Dinâmica:**
 - Slides vão introduzir tópicos/temas
 - Implementações serão apresentadas em Jupyter notebooks.
 - Código disponível em: [drive](#).

Apresentações

Apresentações

- **Apresentação individual dos participantes**
 - Seu nome
 - Formação (concluída ou em andamento)
 - O que te trouxe aqui? Trabalha na área? Quer começar? Entusiasta? etc

Introdução à arquitetura Transformers

Transformers

- **A arquitetura Transformer foi proposta por Vaswani et al. (2017) no paper “Attention is all you need”**
 - Alternativa às Redes Neurais Recorrentes (RNN) para neural machine translation.
 - Mudança de paradigma.

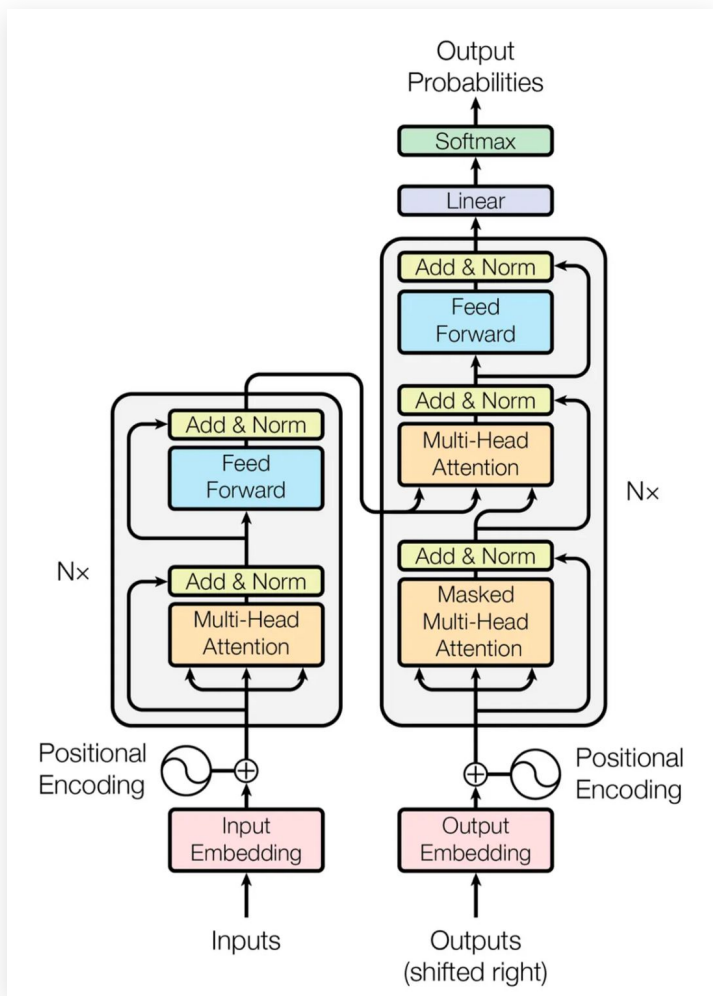
Transformers

- **Arquitetura Encoder-Decoder**



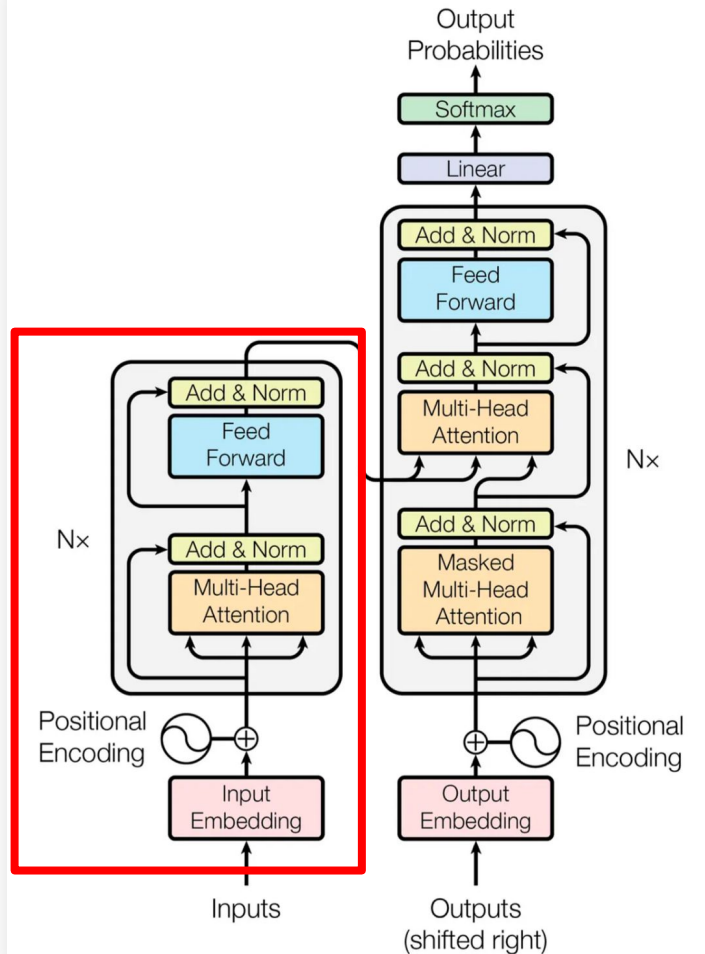
Transformers

- **Arquitetura Encoder-Decoder**



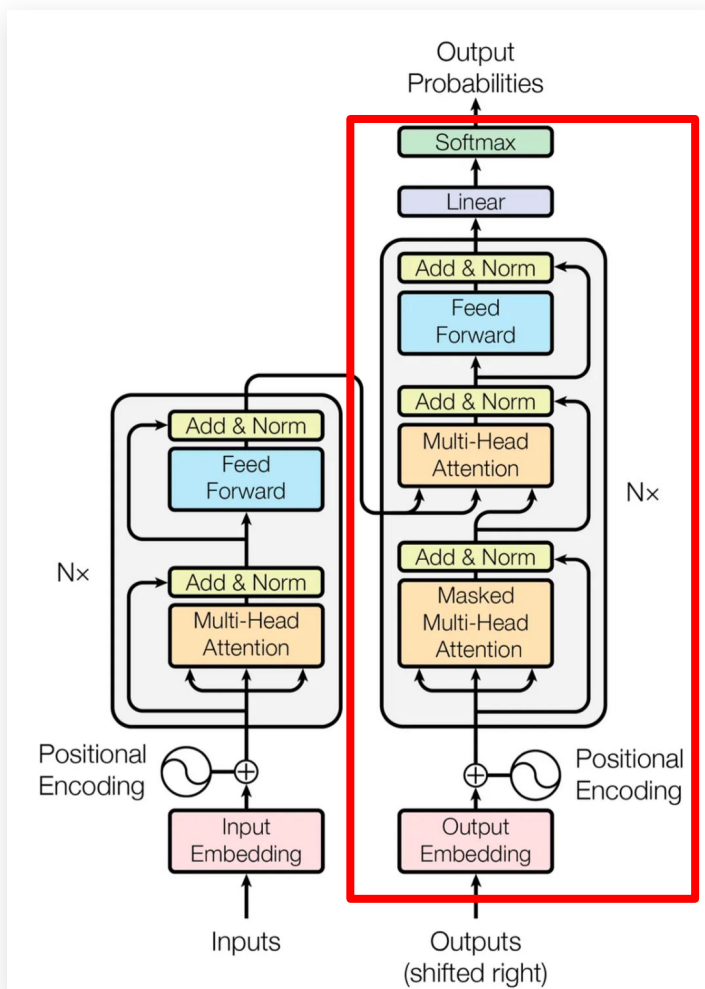
Transformers

- **Encoder:**
 - Cria uma representação rica e contextualizada da frase na língua original



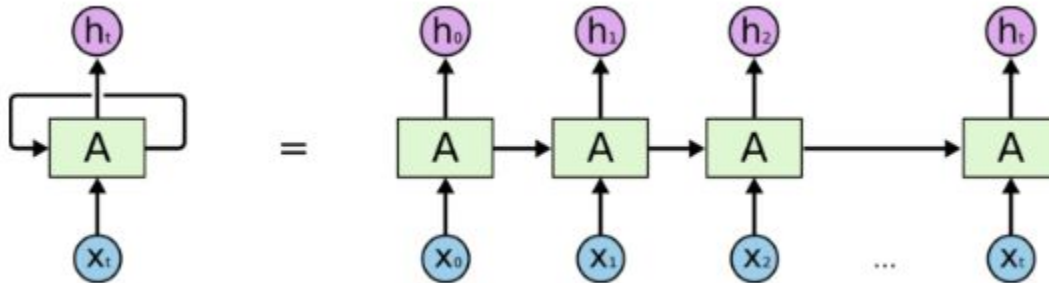
Transformers

- **Decoder:**
 - Gera o texto na língua desejada levando em conta o contexto fornecido pelo **Encoder**.



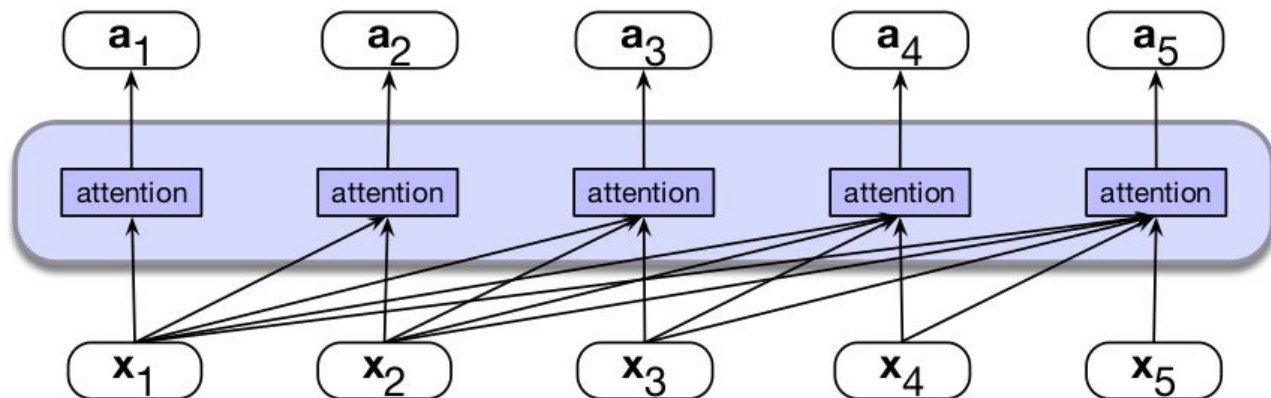
Transformers

- Principais contribuições
 - Diminuiu o problema do “Vanishing gradient”.



Transformers

- Principais contribuições
 - Aumentou a eficiência do treinamento (mecanismo de atenção)

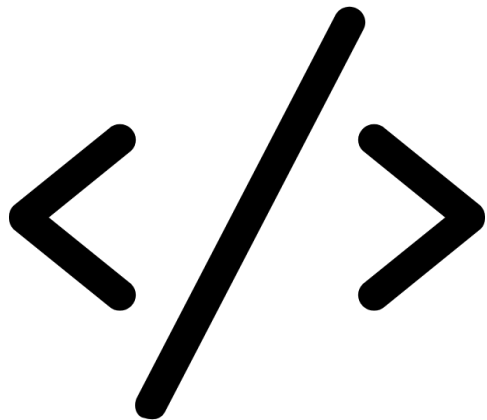


Transformers

- **Principais contribuições**

- Definiu um novo estado da arte para neural machine translation (WMT14):
 - De 40.4 BLEU para 41.8 BLEU (En-Fr).
 - De 26.03 BLEU para 28.4 BLEU (En-De)

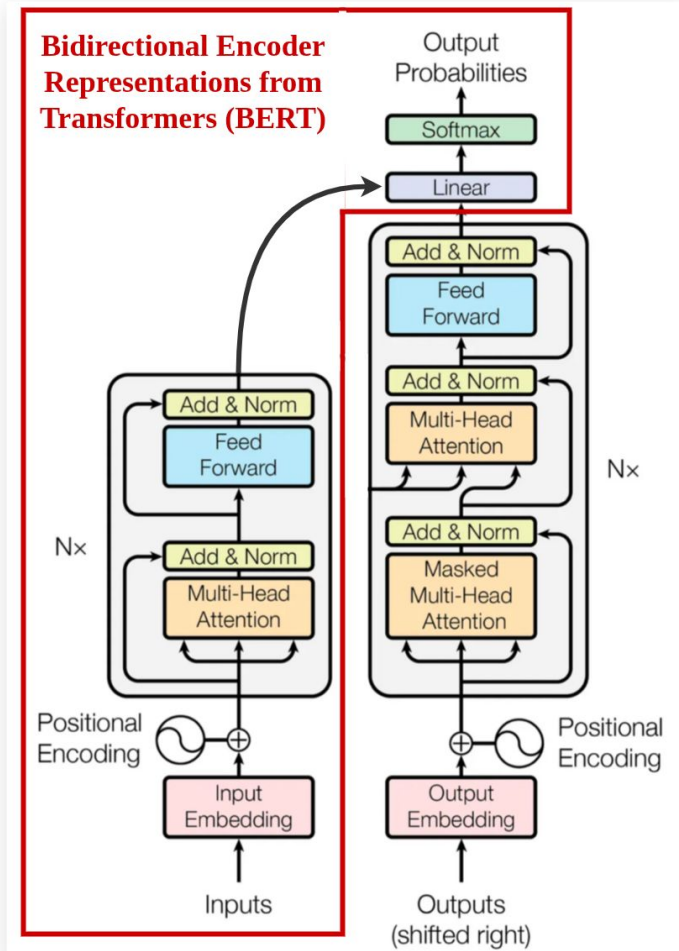
Implementação dos blocos do Transformer



Let's code!

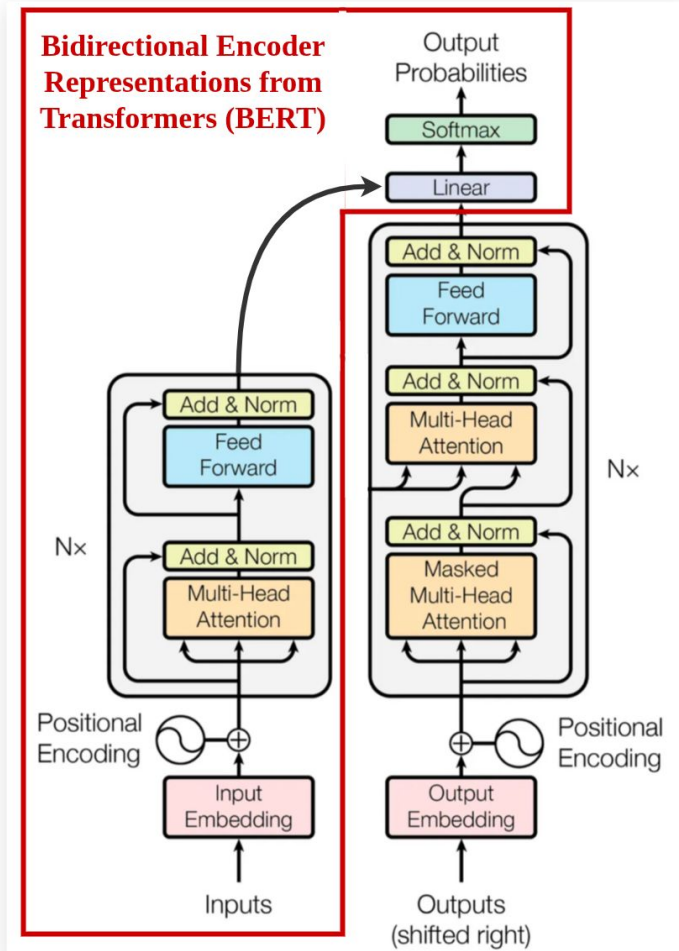
Bidirectional Encoder Representations from Transformers (BERT)

- Proposto por Devlin et al. 2018 (Google)
 - Baseado no Encoder do Transformer



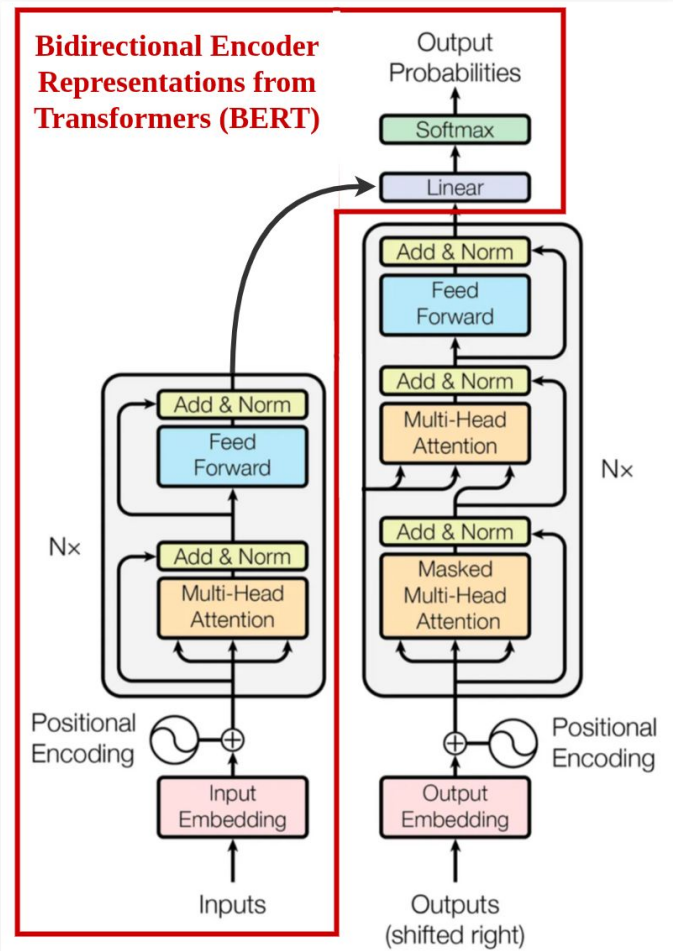
Bidirectional Encoder Representations from Transformers (BERT)

- Proposto por Devlin et al. 2018 (Google)
 - Baseado no Encoder do Transformer
 - [CLS] Token



Bidirectional Encoder Representations from Transformers (BERT)

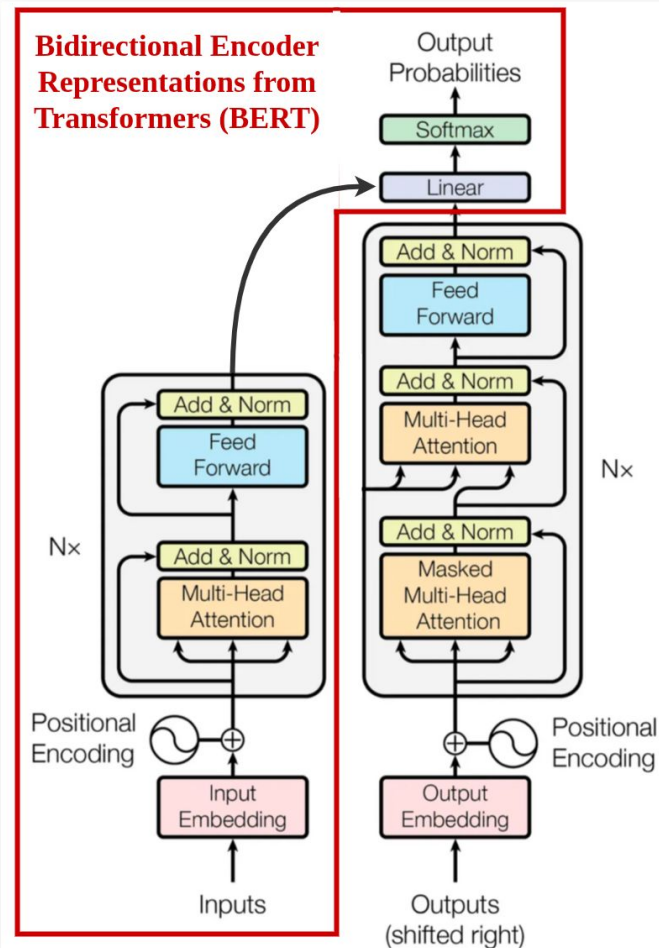
- Proposto por Devlin et al. 2018 (Google)
 - Baseado no Encoder do Transformer
 - [CLS] Token
 - Treinamento não supervisionado (Wikipedia e BookCorpus)
 - Masked Language Modeling
 - Next Sentence Prediction



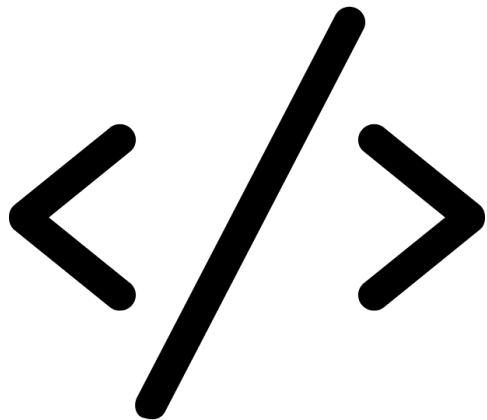
Bidirectional Encoder Representations from Transformers (BERT)

- Principais contribuições

- Avançou o estado da arte em 3 tarefas e 8 benchmarks diferentes:
 - Inferência de linguagem natural (RTE, QNLI, MNLI)
 - Similaridade de sentenças (STS-B, MRPC, QQP)
 - Classificação de sentenças (SST-2, CoLA)



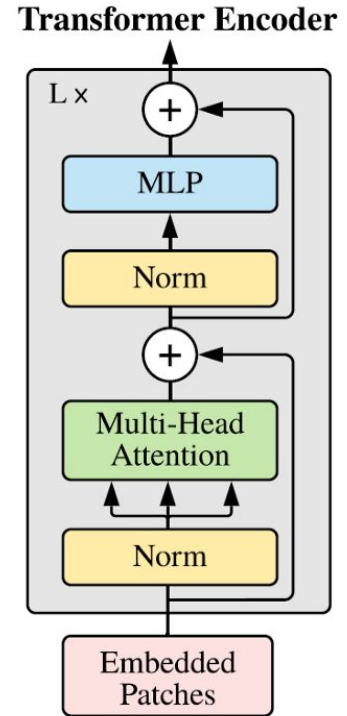
Aplicação 1: classificação de especialidade médica a partir de transcrições médicas



Let's code!

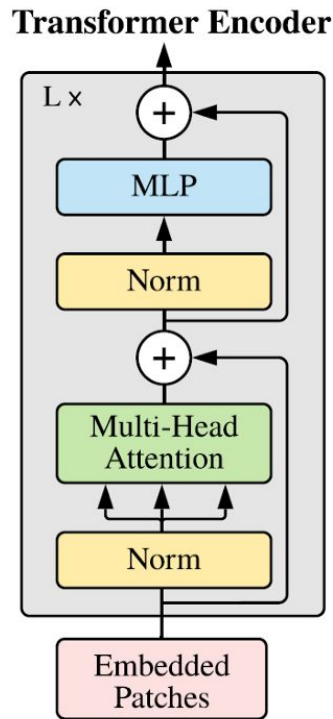
Vision Transformer

- **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (Dosovitskiy et al. 2020)**
 - Desafiou a soberania das Redes Neurais Convolucionais (CNN).



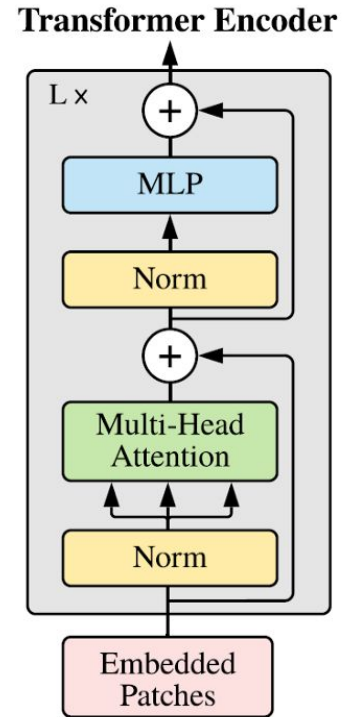
Vision Transformer

- **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (Dosovitskiy et al. 2021)**
 - Desafiou a soberania das Redes Neurais Convolucionais (CNN).
 - Atingiu resultados competitivos na ImageNet1K.
 - Acurácia: 88.55 vs 88.5 (EfficientNet-L2)



Vision Transformer

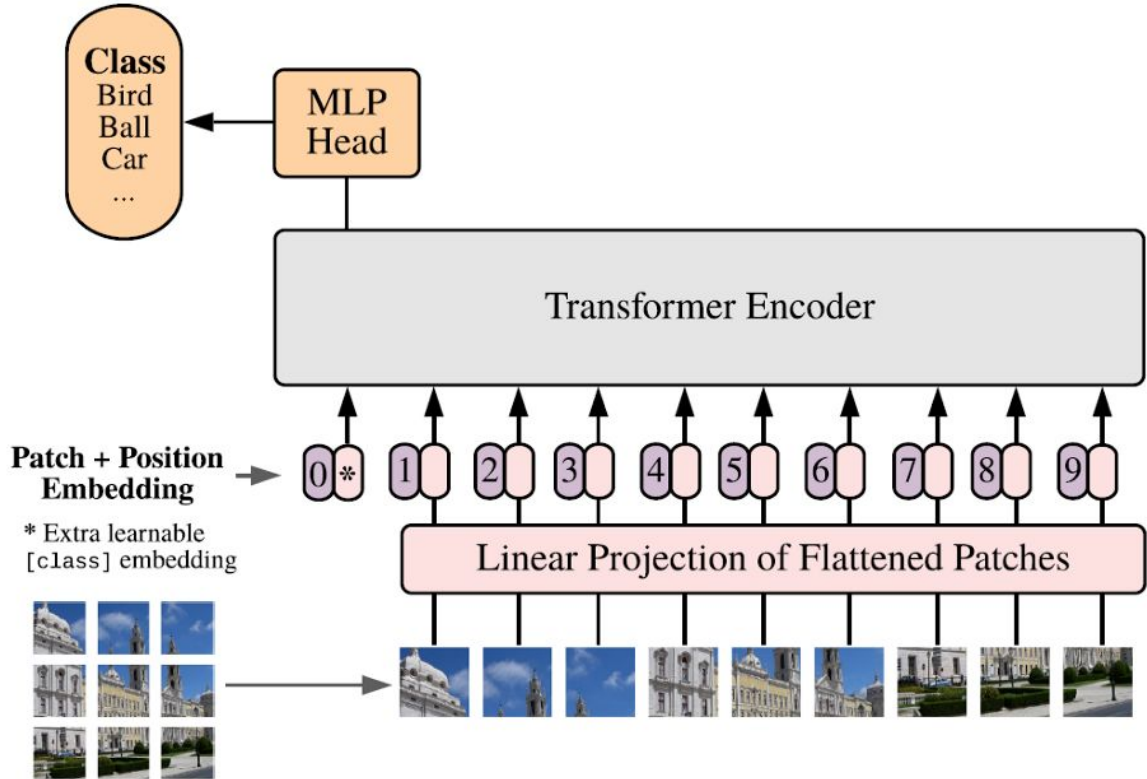
- **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (Dosovitskiy et al. 2020)**
 - Desafiou a soberania das Redes Neurais Convolucionais (CNN).
 - Atingiu resultados competitivos na ImageNet1K.
 - Acurácia: 88.55 vs 88.5 (EfficientNet-L2)
 - Mais eficiente que CNNs: 2.5k vs 12.3k TPUv3-core-days (JFT-300M).



Vision Transformer

- **Patch Embeddings**

- 16x16 pixels
- $(224/16) \times (224/16) = 196$ patches.

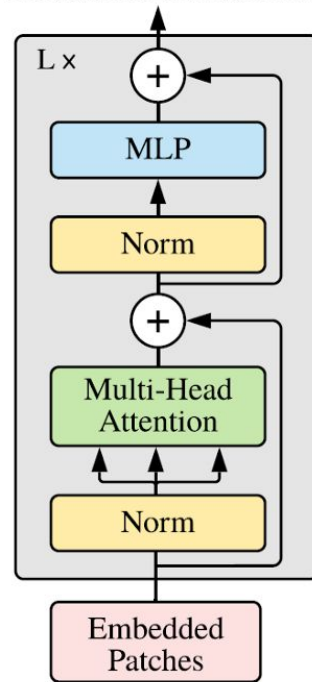


Vision Transformer

- **Pontos negativos:**

- Não possui o viés indutivo das CNNs
 - Necessita de grandes volumes de dados ou formas alternativas de treinamento.

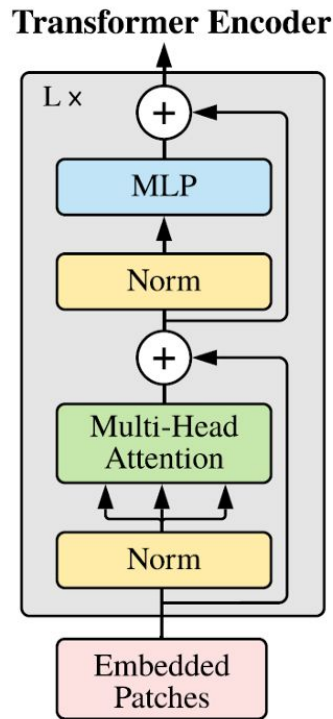
Transformer Encoder



Vision Transformer

- **Pontos negativos:**

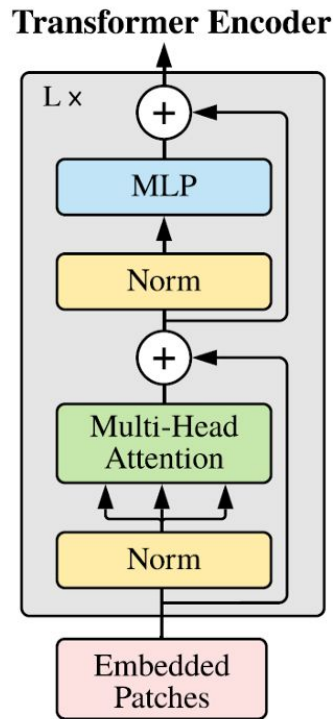
- Não possui o viés indutivo das CNNs
 - Necessita de grandes volumes de dados ou formas alternativas de treinamento.
 - Data-efficient Image Transformers (Touvron et al. 2020)



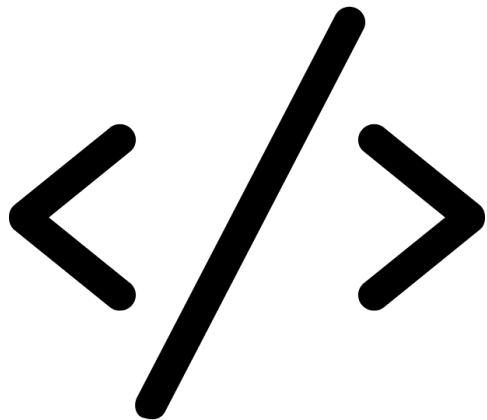
Vision Transformer

- **Pontos negativos:**

- Não possui o viés indutivo das CNNs
 - Necessita de grandes volumes de dados ou formas alternativas de treinamento.
 - Data-efficient Image Transformers (Touvron et al. 2020)
- Ineficiente com imagens de alta resolução
 - Problema atacado pelo Swin Transformer (Liu et al. 2021)



Aplicação 2: classificação de câncer de pele



Let's code!

Desafio

- **Parte 1: Transferência de aprendizado:**

- Altere o exemplo do VIT para utilizar um transformer pré-treinado em outro conjunto de dados.
- Você pode obter este modelo em [HuggingFace](#).
- Como o resultado se compara com o modelo treinado do zero?

- **Parte 2: Mão na massa**

- Altere o exemplo do BERT ou VIT para utilizar dados relacionados a sua área de pesquisa ou atuação.

Dúvidas?



Obrigado!

