Alexa, faça de conta que você não está escutando.

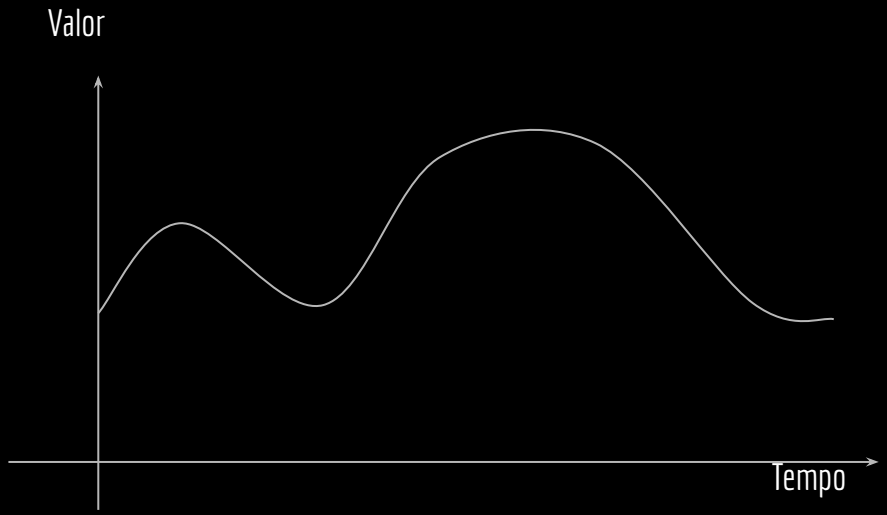
Transformers Multimodais

Paulo Ricardo Lisboa de Almeida

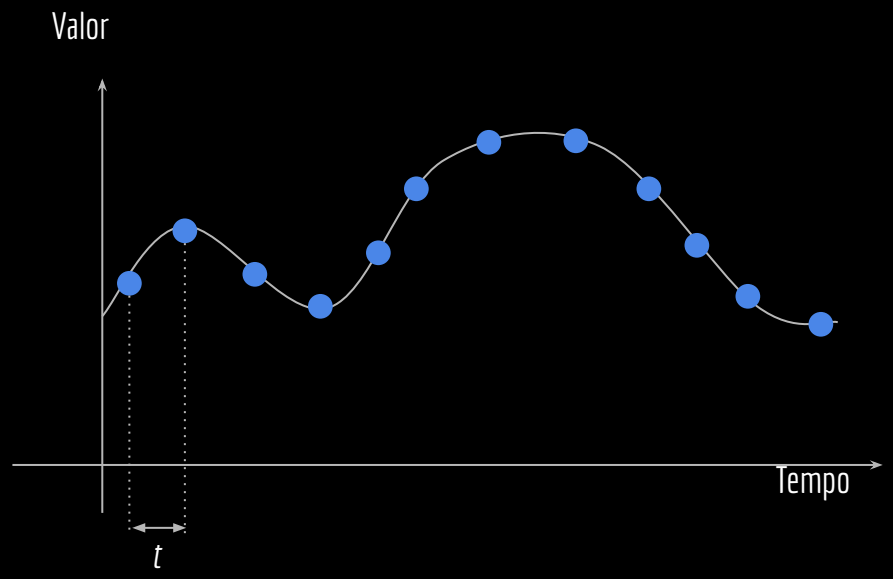
DSBD

UFPR

Mundo analógico e discreto



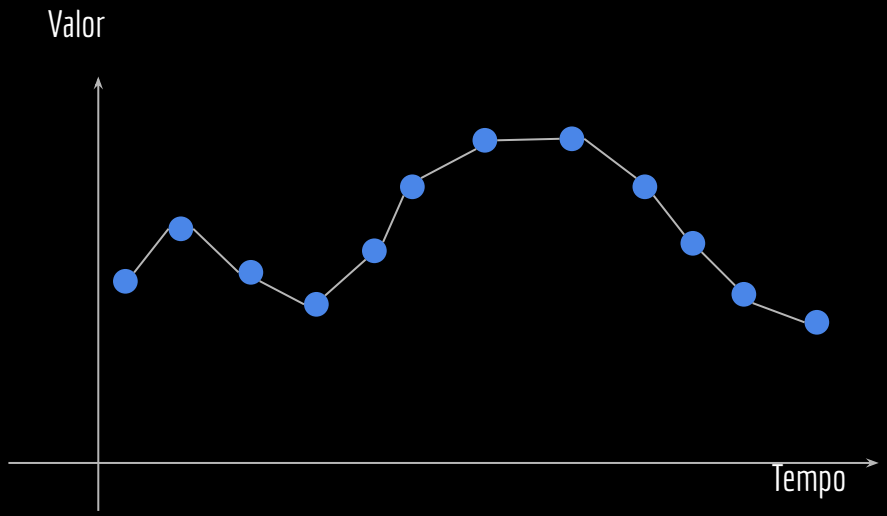
Mundo analógico e discreto



Amostragem.

Podemos discretizar um sinal analógico, verificando o seu valor a cada *t* instantes de tempo (*t* é o período).

Mundo analógico e discreto



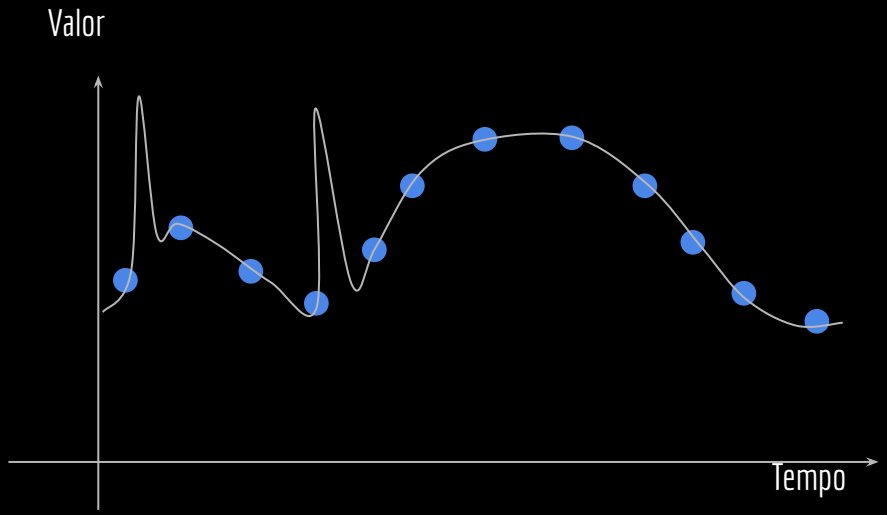
Amostragem.

Podemos discretizar um sinal analógico, verificando o seu valor a cada *t* instantes de tempo.

Quanto menor for *t*, mais fiel é o sinal discreto quando comparado ao analógico.

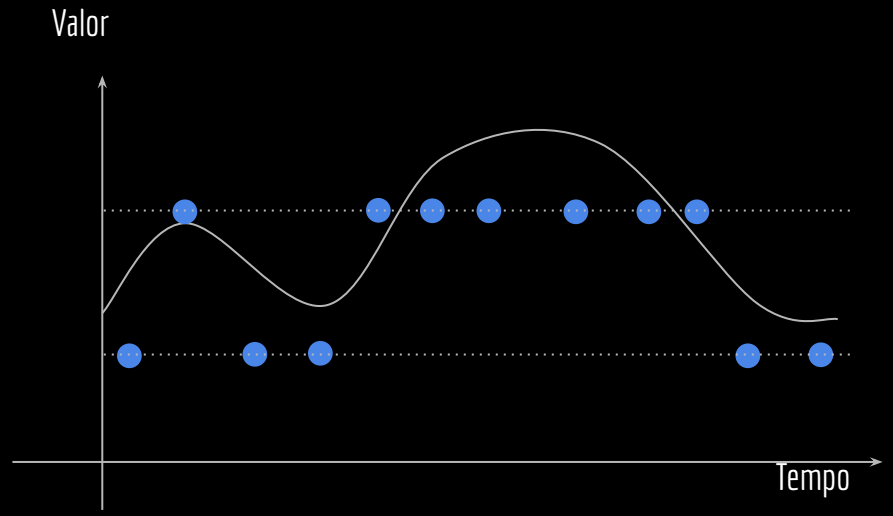
Podemos interpolar o sinal discreto para reconstruir (algo próximo) do sinal original.

Mundo analógico e discreto



Durante a discretização, algumas informações podem ser perdidas.

Mundo analógico e discreto

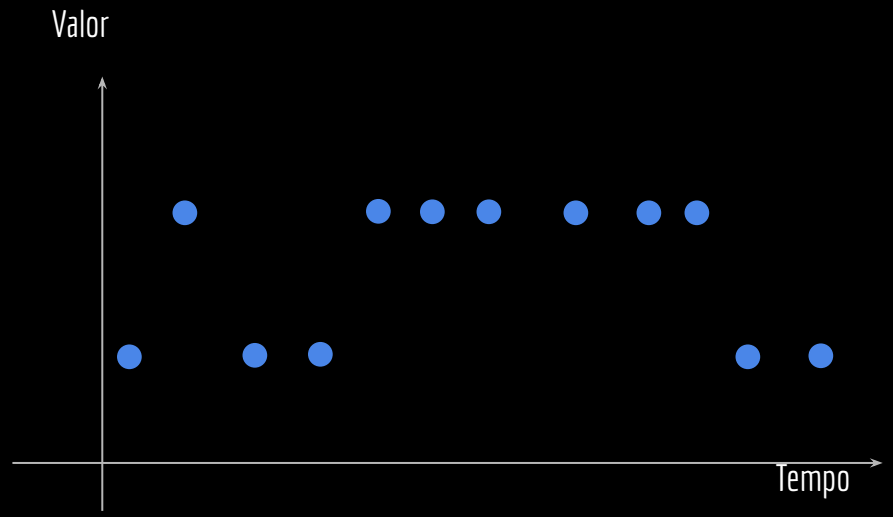


Com 2 valores

Quantização.

No eixo y, quantos valores podemos representar? Quanto mais valores valores, maior a fidelidade no eixo y com o sinal original, mas o hardware se torna mais complicado.

Mundo analógico e discreto

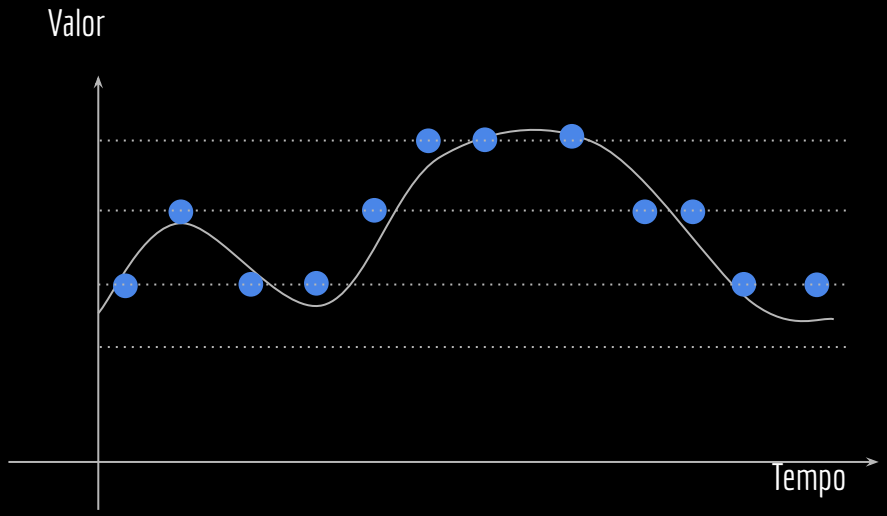


Com 2 valores

Quantização.

No eixo y, quantos valores podemos representar? Quanto mais valores valores, maior a fidelidade no eixo y com o sinal original, mas o hardware se torna mais complicado.

Mundo analógico e discreto

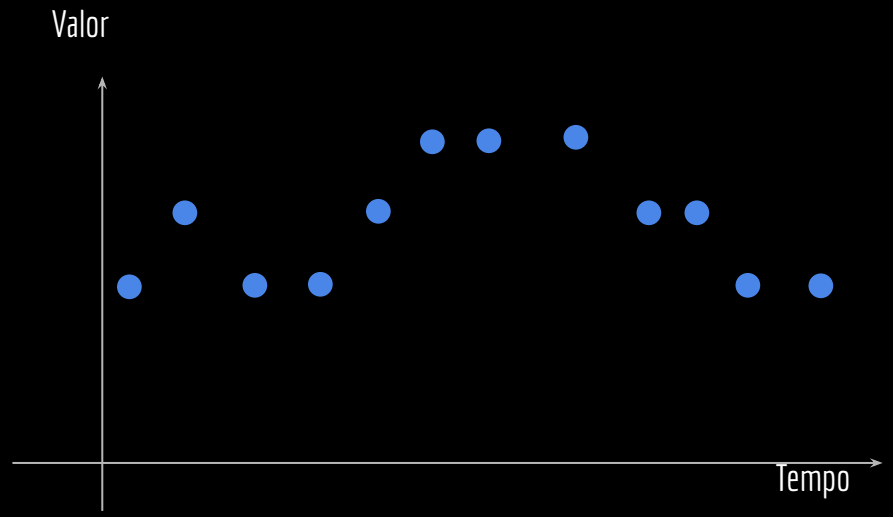


Com 4 valores

Quantização.

No eixo y, quantos valores podemos representar? Quanto mais valores valores, maior a fidelidade no eixo y com o sinal original, mas o hardware se torna mais complicado.

Mundo analógico e discreto



Com 4 valores

Quantização.

No eixo y, quantos valores podemos representar? Quanto mais valores valores, maior a fidelidade no eixo y com o sinal original, mas o hardware se torna mais complicado.

Exemplo

Geralmente as músicas que ouvimos possuem uma amostragem de 44.100Hz.

Isso significa que a cada $t = 1/441000 \approx 0{,}000023$ segundos um sinal é amostrado.

Exemplo

Geralmente as músicas que ouvimos possuem uma amostragem de 44.100Hz.

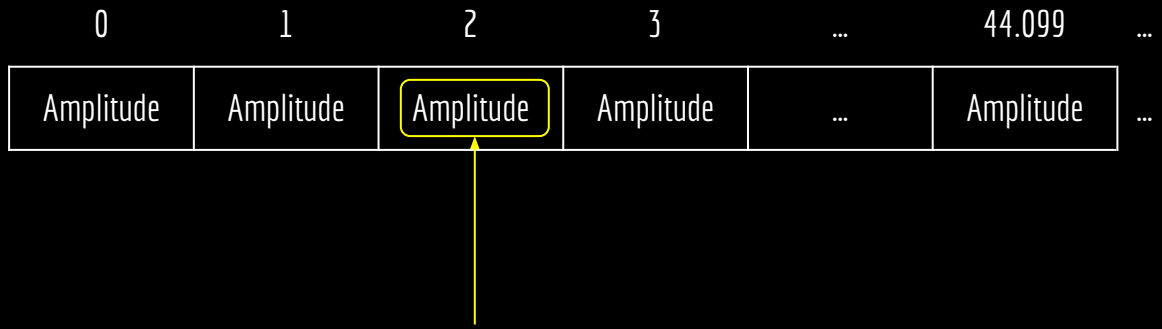
Isso significa que a cada t = 1/441000 ≈ 0,000023 segundos um sinal é amostrado.

f = 1/t, onde f é a frequência, e t o período.

Exemplo - CD

Técnica chamada de Pulse-Code Modulation.

Exemplo - CD



Cada amostra de amplitude é armazenada como um valor de 16 bits.

Técnica chamada de Pulse-Code Modulation.

Exemplo - CD

Podemos adicionar mais fluxos para ter múltiplos canais de áudio. Por exemplo, em um CD com dois canais (stereo).

Canal 1.

Canal 2.

Técnica chamada de Pulse-Code Modulation.

Convoluções

Podemos aplicar uma CNN.

Convoluções de *1 x N x C*, onde *C* é o número de canais, e *N* é o tamanho do Kernel.

Transformer

Ou podemos aplicar um transformer, que vai ser capaz de capturar o contexto global.
Mas possui um custo quadrático de acordo com o tamanho da entrada.

Ideia

A mesma ideia aplicada para modelos de linguagem pode ser aplicada.

Dado um conjunto de tokens (de áudio) prever, por exemplo, qual o token (de texto) pertence a esse trecho.

Nesse exemplo, o modelo se torna um *speech to text*.

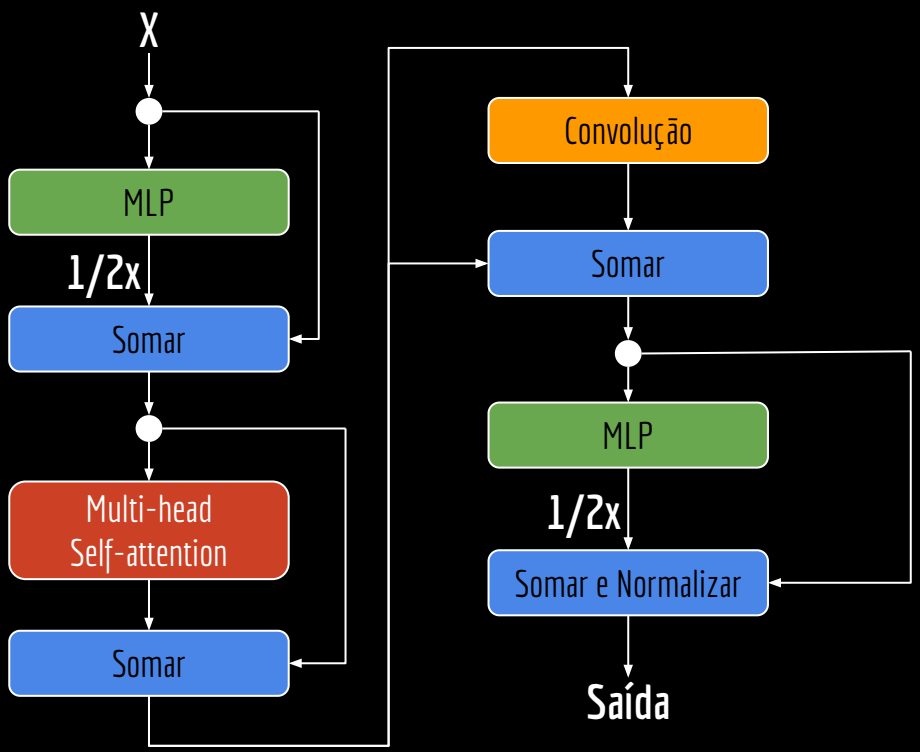
Estado da arte

Técnicas do estado da arte combinam convoluções com transformers.

Convoluções para capturar as correlações locais e reduzir a dimensionalidade dos dados.

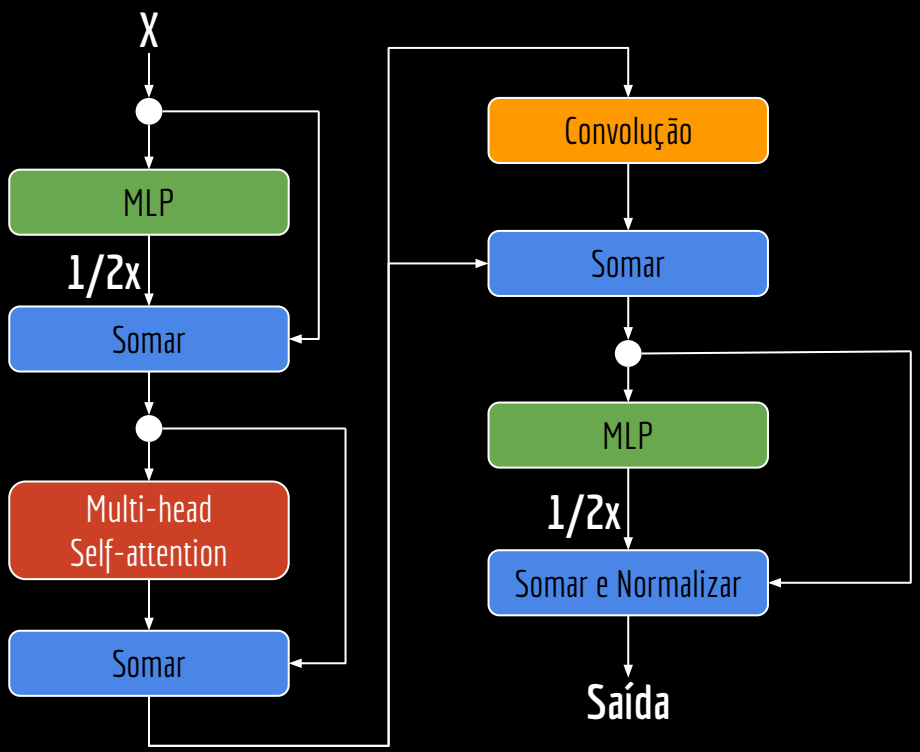
Transformers para capturar as correlações globais.

Conformer



GULATI, Anmol et al. Conformer: Convolution-augmented transformer for speech recognition. 2020.

Conformer



GULATI, Anmol et al. Conformer: Convolution-augmented transformer for speech recognition. 2020.

$$\tilde{\mathbf{x}} = \mathbf{x} + \frac{1}{2} MLP(\mathbf{x})$$

$$\mathbf{x}' = \tilde{\mathbf{x}} + MHSA(\tilde{\mathbf{x}})$$

$$\mathbf{x}'' = \mathbf{x}' + Conv(\mathbf{x}')$$

$$\mathbf{y} = normalizar(\mathbf{x}'' + \frac{1}{2} MLP(\mathbf{x}''))$$

Conformer

GULATI, Anmol et al. Conformer: Convolution-augmented transformer for speech recognition. 2020.


Conformer: Convolution-augmented Transformer for Speech Recognition

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, Ruoming Pang

Google Inc.

{anmolgulati, jamesqin, chungchengc, nikip, ngyuzh, jiahuiyu, weihan, shibow, zhangzd, yonghui, rpang}@google.com

Abstract

Cabeças de atenção ficam em um “sanduíche” de MLPs.

O primeiro MLP projeta os dados para uma dimensão 4x menor. O segundo desfaz essa operação.

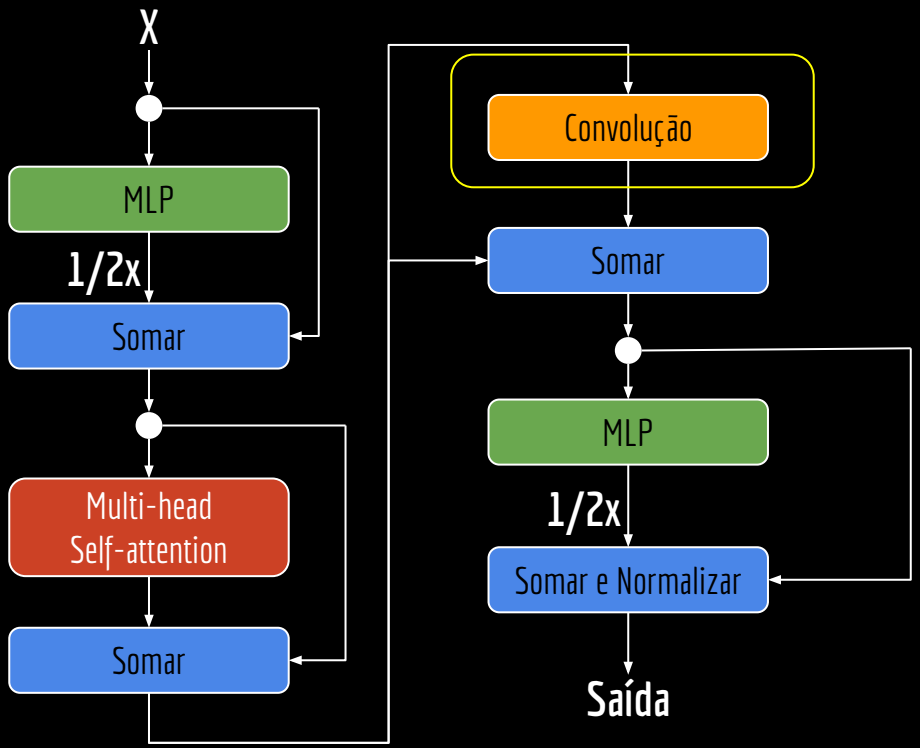
$$\tilde{\mathbf{x}} = \mathbf{x} + \frac{1}{2} MLP(\mathbf{x})$$

$$\mathbf{x}' = \tilde{\mathbf{x}} + MHSA(\tilde{\mathbf{x}})$$

$$\mathbf{x}'' = \mathbf{x}' + Conv(\mathbf{x}')$$

$$\mathbf{y} = normalizar(\mathbf{x}'' + \frac{1}{2} MLP(\mathbf{x}''))$$

Conformer



GULATI, Anmol et al. Conformer: Convolution-augmented transformer for speech recognition. 2020.


Conformer: Convolution-augmented Transformer for Speech Recognition

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, Ruoming Pang

Google Inc.

Abstract

1. Introduction

Figure 1: Conformer encoder model architecture. Conformer comprises of two macaron-like feed-forward layers with half-step residual connections sandwiching the multi-headed self-attention and convolution modules. This is followed by a post-layernorm.

Camada convolucional aplica convolução de tamanho 1 (pointwise) para combinar os canais, e depois aplica uma convolução de tamanho 31 para capturar as correlações locais.

Fast Conformer

Em 2023, os Fast Conformers foram propostos.

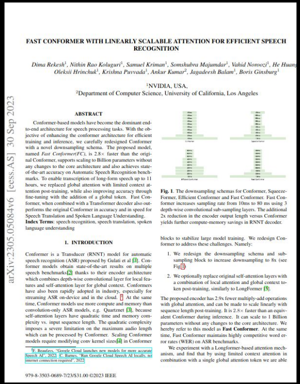
Mesma ideia básica do Conformer.

Troca do tokenizador.

    Possibilitou fazer o donwsampling de 4x para 8x.

Kernels de tamanho 9.

REKESH, Dima et al. Fast conformer with linearly scalable attention for efficient speech recognition. IEEE Automatic Speech Recognition and Understanding Workshop. 2023.



FAST CONFORMER WITH LINEARLY SCALABLE ATTENTION FOR EFFICIENT SPEECH RECOGNITION

Dima Rekesh[1], Nithin Rao Koluguri[1], Samuel Kriman[1], Somshubra Majumdar[1], Vahid Noroozi[1], He Huang
Oleksii Hrinchuk[1], Krishna Puvvada[1], Ankur Kumar[2], Jagadeesh Balam[1], Boris Ginsburg[1]

[1]NVIDIA, USA,
[2]Department of Computer Science, University of California, Los Angeles

ABSTRACT

Conformer-based models have become the dominant end-to-end architecture for speech processing tasks. With the objective of enhancing the conformer architecture for efficient training and inference, we carefully redesigned Conformer with a novel downsampling schema. The proposed model, named *Fast Conformer(FC)*, is 2.8× faster than the original Conformer, supports scaling to Billion parameters without any changes to the core architecture and also achieves state-of-the-art accuracy on Automatic Speech Recognition benchmarks. To enable transcription of long-form speech up to 11 hours, we replaced global attention with limited context attention post-training, while also improving accuracy through fine-tuning with the addition of a global token. Fast Conformer, when combined with a Transformer decoder also outperforms the original Conformer in accuracy and in speed for Speech Translation and Spoken Language Understanding.
Index Terms: speech recognition, speech translation, spoken language understanding

1. INTRODUCTION

Conformer is a Transducer (RNNT) model for automatic speech recognition (ASR) proposed by Gulati et al [1]. Conformer models obtain state-of-the-art results on multiple speech benchmarks[2] thanks to their encoder architecture which combines depth-wise convolutional layer for local features and self-attention layer for global context. Conformers have also been rapidly adopted in industry, especially for streaming ASR on-device and in the cloud. [1] At the same time, Conformer models use more compute and memory than convolution-only ASR models, e.g. Quartznet [3], because self-attention layers have quadratic time and memory complexity vs. input sequence length. The quadratic complexity imposes a severe limitation on the maximum audio length which can be processed by Conformer. Scaling Conformer models require modifying conv kernel sizes[4] in Conformer blocks to stabilize large model training. We redesign Conformer to address these challenges. Namely:

1. We redesign the downsampling schema and sub-sampling block to increase downsampling to 8x (see Fig[1])
2. We optionally replace original self-attention layers with a combination of local attention and global context token post-training, similarly to LongFormer [5].

The proposed encoder has 2.9x fewer multiply-add operations with global attention, and can be made to scale linearly with sequence length post-training. It is 2.8× faster than an equivalent Conformer during inference. It can scale to 1 Billion parameters without any changes to the core architecture. We hereby refer to this model as **Fast Conformer**. At the same time, Fast Conformer maintains highly competitive word error rates (WER) on ASR benchmarks.

We experiment with a Longformer-based attention mechanism, and find that by using limited context attention in combination with a single global attention token we are able

Fig. 1. The downsampling schemas for Conformer, Squeeze-Former, Efficient Conformer and Fast Conformer. Fast Conformer increases sampling rate from 10ms to 80 ms using 3 depth-wise convolutional sub-sampling layers. The additional 2x reduction in the encoder output length versus Conformer yields further compute-memory savings in RNNT decoder.

[1]F. Beaufays, "Google Cloud launches new models for more accurate Speech AI", 2022. C. Barnes, "Run Google Cloud Speech AI locally, no internet connection required", 2022

Zipformer

Proposta de:

- Novas funções de ativação (*Swoosh*).
- Nova normalização (*Biasnorm*).

Rede profunda onde blocos Multi-head Self-attention enxergam diferentes sub amostragens (*downsamplings*) do áudio.

YAO, Zengwei et al. Zipformer: A faster and better encoder for automatic speech recognition. In: International Conference on Learning Representations. 2024.


Published as a conference paper at ICLR 2024

ZIPFORMER: A FASTER AND BETTER ENCODER FOR AUTOMATIC SPEECH RECOGNITION

Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, Daniel Povey
Xiaomi Corp., Beijing, China
dpovey@xiaomi.com

ABSTRACT

The Conformer has become the most popular encoder model for automatic speech recognition (ASR). It adds convolution modules to a Transformer to learn both local and global dependencies. In this work we describe a faster, more memory-efficient, and better-performing Transformer, called Zipformer. Modeling changes include: 1) a U-Net-like encoder structure where middle stacks operate at lower frame rates; 2) reorganized block structure with more modules, within which we re-use attention weights for efficiency; 3) a modified form of LayerNorm called BiasNorm allows us to retain some length information; 4) new activation functions SwooshR and SwooshL work better than Swish. We also propose a new optimizer, called ScaledAdam, which scales the update by each tensor's current scale to keep the relative change about the same, and also explictly learns the parameter scale. It achieves faster convergence and better performance than Adam. Extensive experiments on LibriSpeech, Aishell-1, and WenetSpeech datasets demonstrate the effectiveness of our proposed Zipformer over other state-of-the-art ASR models. Our code is publicly available at https://github.com/k2-fsa/icefall.

1 INTRODUCTION

End-to-end models have achieved remarkable success in automatic speech recognition (ASR). An effective encoder architecture that performs temporal modeling on the speech sequence plays a vital role in end-to-end ASR models. A most prominent example is Conformer (Gulati et al., 2020), which combines the advantages of the convolutional neural network (CNN) models (Zhang et al., 2017; Li et al., 2019; Kriman et al., 2020) and Transformer models (Dong et al., 2018; Karita et al., 2019; Zhang et al., 2020b). By integrating CNN into Transformer (Vaswani et al., 2017), Conformer is able to extract both local and global dependencies on speech sequences, and achieves state-of-the-art performance in ASR.

In this work, we propose a faster, more memory-efficient, and better-performing Transformer as ASR encoder, called *Zipformer*. First, unlike Conformer that operates on the sequence at a constant frame rate, *Zipformer* adopts a U-Net-like (Ronneberger et al., 2015) structure, which consists of multiple stacks downsamping the sequence to various lower frame rates. Second, we re-design the block structure, which is equipped with more modules like two Conformer blocks, and reuses the attention weights for efficiency. We propose *BiasNorm* as a simpler replacement of LayerNorm, which allows for retaining length information in normalization. We also replace Swish with our new activation functions *SwooshR* and *SwooshL* to achieve better results. In addition, we devise a parameter-scale-invariant version of Adam, called *ScaledAdam*, which scales the update by the current parameter scale and also explicitly learns the parameter scale. Compared to Adam, *ScaledAdam* enables faster convergence and better performance.

Extensive experiments are conducted on LibriSpeech, Aishell-1, and WenetSpeech datasets, and results demonstrate the effectiveness of the proposed modeling and optimization-related innovations. *Zipformer* achieves state-of-the-art results on all three datasets. It is worth mentioning that *Zipformer* is the first model ever to achieve results comparable to those reported in the Conformer paper on the LibriSpeech dataset (these results have proved difficult for others to reproduce). In terms of efficiency, *Zipformer* converges faster during training and speeds up the inference by more than

1

Projeto

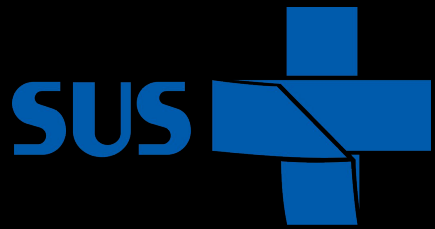
O uso de transformers está sendo estudado no Projeto “Comunicação direta entre usuários e a Atenção Primária à Saúde (APS)”.

Pesquisa atual: como converter as respostas de usuários dadas via ligação de voz para dados textuais.

Algo parecido com um “Quiz” que a pessoa pode responder utilizando a própria voz, sem formulários.

Os dados podem ser processados para identificar possíveis melhorias em determinadas regiões, hospitais, postos de saúde, ...

Obs.: os dados são processados apenas com o consentimento dos usuários.



Outras modalidades

Dados que podemos projetar em um espaço (*embedding*) de tokens, e onde há uma dependência sequencial entre os dados, podem ser processados através de transformers.

O processamento de áudio foi um exemplo.

Outras modalidades

Dados que podemos projetar em um espaço (*embedding*) de tokens, e onde há uma dependência sequencial entre os dados, podem ser processados através de transformers.

O processamento de áudio foi um exemplo.

Outro exemplo: processamento de imagens.

Transformers para visão

Em visão, podemos utilizar cada pixel individual como um *token*.

Vai ser computacionalmente custoso, mas funciona.

Transformers para visão

Em visão, podemos utilizar cada pixel individual como um *token*.

Vai ser computacionalmente custoso, mas funciona.

Para reduzir o custo podemos, por exemplo:

- Utilizar patches de *N x N* das imagens, e projetar cada patch como um token, ou
- Utilizar filtros de convolução para reduzir a dimensionalidade das imagens originais.

Ideia do ViT: *Vision Transformer*

Transformers para visão

O embedding final gerado pelos transformers pode ser usado para, por exemplo, passar por um MLP para classificar a imagem.

Ou podemos montar um modelo baseado em auto regressão:

Dado o início de uma imagem, prever os próximos pixels dela (terminar de desenhar).

Multimodalidade

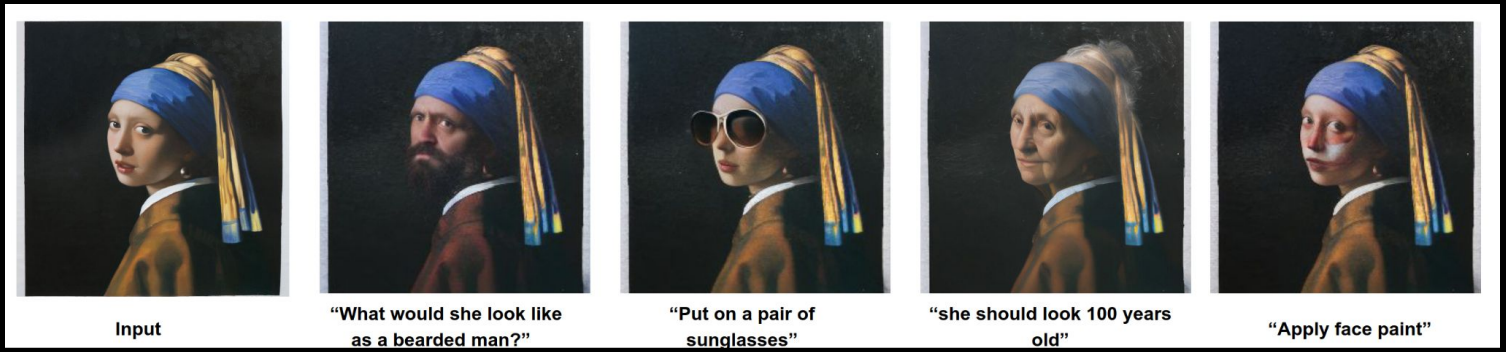
Ao transformar as entradas e saídas em tokens, é relativamente trivial unir múltiplos modais de informação em uma única rede.

Exemplos:

- Uma rede que recebe ao um prompt de texto, e uma imagem de base, e a modifica de acordo com o prompt.
- Uma rede que recebe um prompt, e gera uma imagem de saída (na forma de auto regressão, na próxima iteração, recebe o prompt, e o primeiro pixel já gerado, ...).

Exemplo - CM3Leon

YU, Lili et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. arXiv preprint arXiv:2309.02591. 2023.



Timeline


1943 - Warren McCulloch e Walter Pitts. Neurônio humano modelado como um perceptron eletrônico.

Timeline

1943

Neurônio Eletrônico

1957 - Frank Rosenblatt.
Implementação de um perceptron em um computador IBM 704.

Timeline

1943

Neurônio Eletrônico

1957

Perceptron de Rosenblatt

1970/1971 - Seppo Linnainmaa e Paul Werbos. Desenvolvimento da retropropagação.

Timeline

1969 - Kunihiko Fukushima propõe a ativação ReLU.

1943

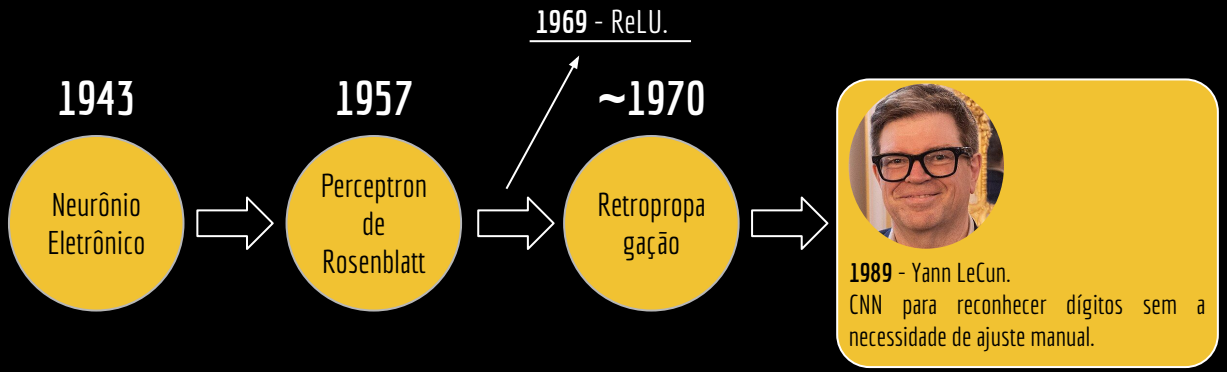
Neurônio Eletrônico

1957

Perceptron de Rosenblatt

1970/1971 - Seppo Linnainmaa e Paul Werbos. Desenvolvimento da retropropagação.

Timeline


1969 - ReLU.
1943
1957
~1970
Neurônio Eletrônico
Perceptron de Rosenblatt
Retropropagação
1989 - Yann LeCun.
CNN para reconhecer dígitos sem a necessidade de ajuste manual.

Timeline

1980 - Kunihiko Fukushima propõe a neocognitron. Primeira CNN, com camadas de convolução e pooling.

1969 - ReLU.

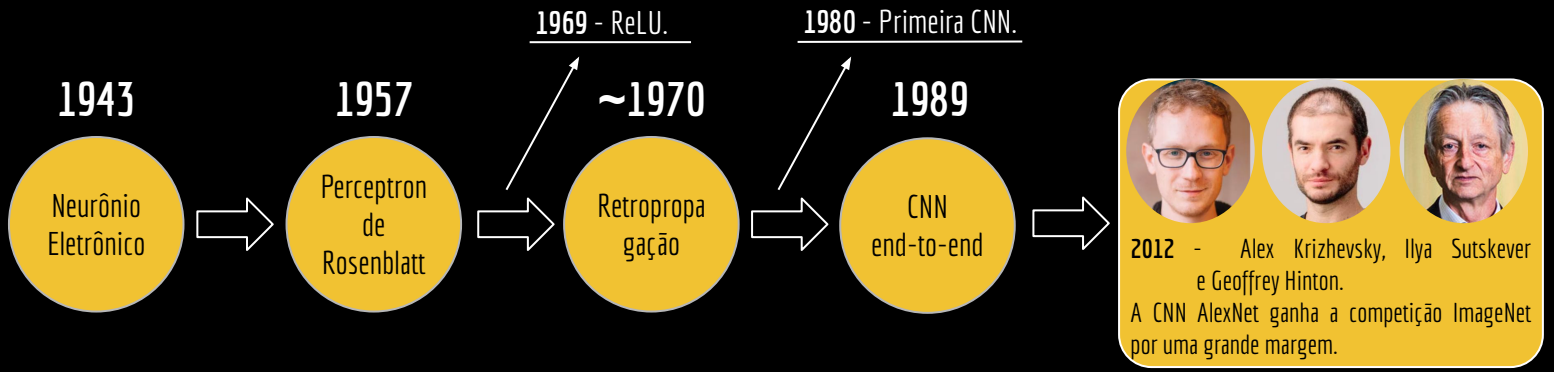
1943 Neurônio Eletrônico

1957 Perceptron de Rosenblatt

~1970 Retropropagação

1989 - Yann LeCun.
CNN para reconhecer dígitos sem a necessidade de ajuste manual.

Timeline


1943
Neurônio Eletrônico
1957
Perceptron de Rosenblatt
1969 - ReLU.
~1970
Retropropagação
1980 - Primeira CNN.
1989
CNN end-to-end
2012 - Alex Krizhevsky, Ilya Sutskever e Geoffrey Hinton.
A CNN AlexNet ganha a competição ImageNet por uma grande margem.

Timeline

1943 Neurônio Eletrônico

1957 Perceptron de Rosenblatt

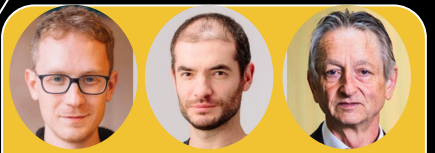
~1970 Retropropagação

1969 - ReLU.

1980 - Primeira CNN.

1989 CNN end-to-end

2009 - Rajat Raina, Anand Madhavan e Andrew Y. Ng usam 30 GPUs GTX 280 para treinar um modelo de 100M de parâmetros.



2012 - Alex Krizhevsky, Ilya Sutskever e Geoffrey Hinton.
A CNN AlexNet ganha a competição ImageNet por uma grande margem.

Timeline

1943 Neurônio Eletrônico

1957 Perceptron de Rosenblatt

~1970 Retropropagação

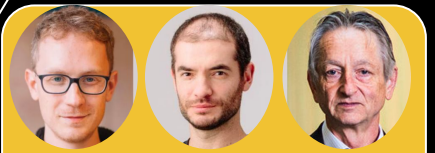
1969 - ReLU.

1980 - Primeira CNN.

1989 CNN end-to-end

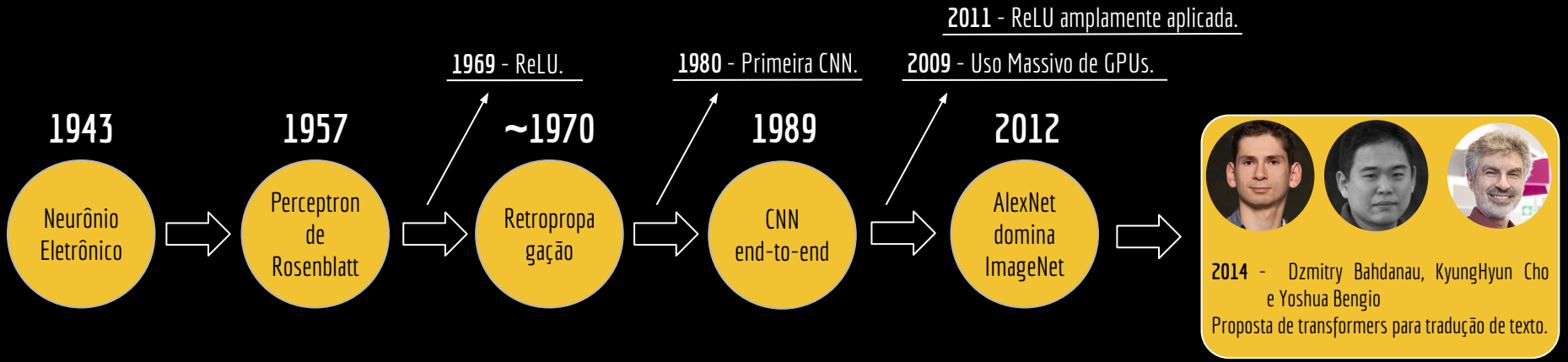
2009 - Rajat Raina, Anand Madhavan e Andrew Y. Ng usam 30 GPUs GTX 280 para treinar um modelo de 100M de parâmetros.

2011 - Xavier Glorot, Antoine Bordes e Yoshua Bengio descobrem que a ReLU funciona bem em redes profundas.



2012 - Alex Krizhevsky, Ilya Sutskever e Geoffrey Hinton.
A CNN AlexNet ganha a competição ImageNet por uma grande margem.

Timeline


1943
Neurônio Eletrônico
1957
Perceptron de Rosenblatt
1969 - ReLU.
~1970
Retropropagação
1980 - Primeira CNN.
1989
CNN end-to-end
2009 - Uso Massivo de GPUs.
2011 - ReLU amplamente aplicada.
2012
AlexNet domina ImageNet
2014 - Dzmitry Bahdanau, KyungHyun Cho e Yoshua Bengio
Proposta de transformers para tradução de texto.

Timeline

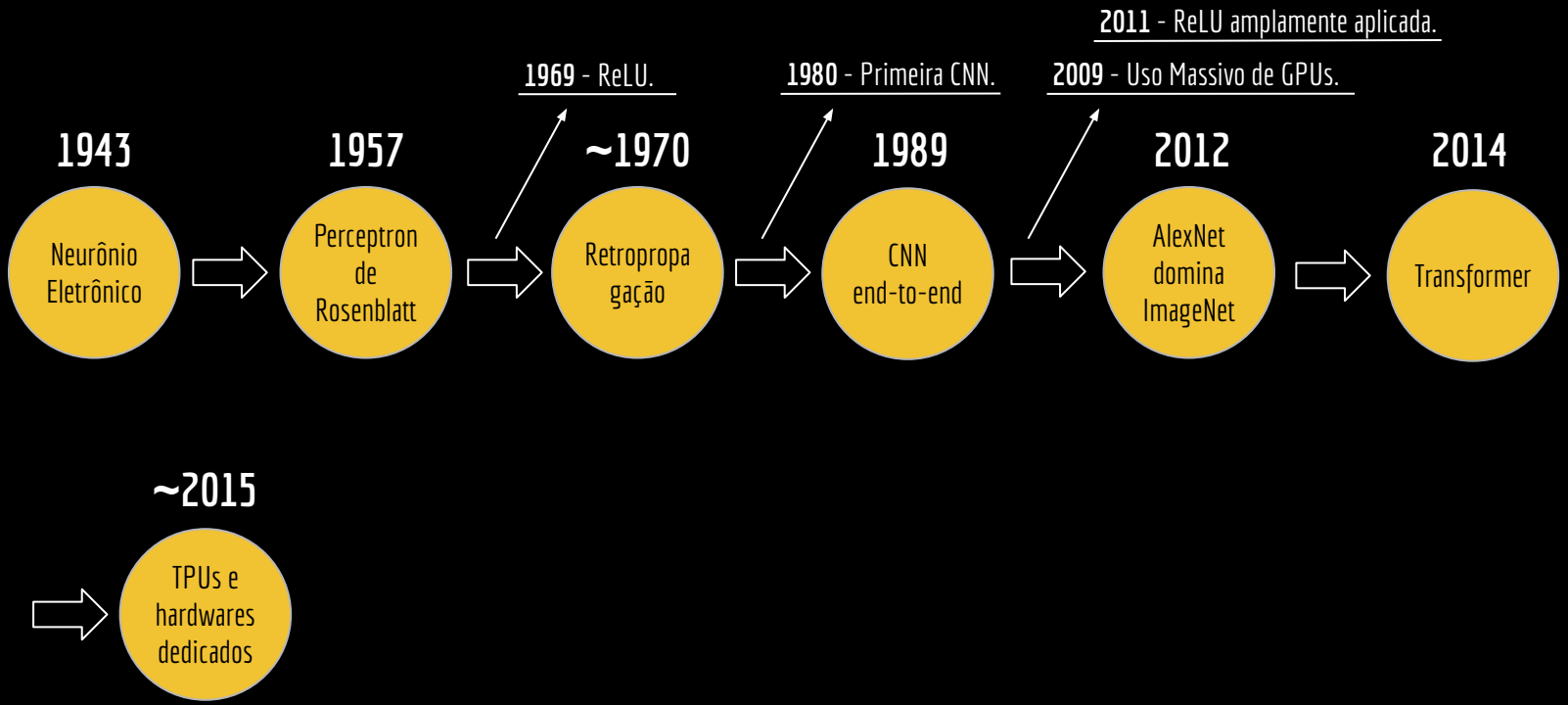
2011 - ReLU amplamente aplicada.

1969 - ReLU. **1980** - Primeira CNN. **2009** - Uso Massivo de GPUs.

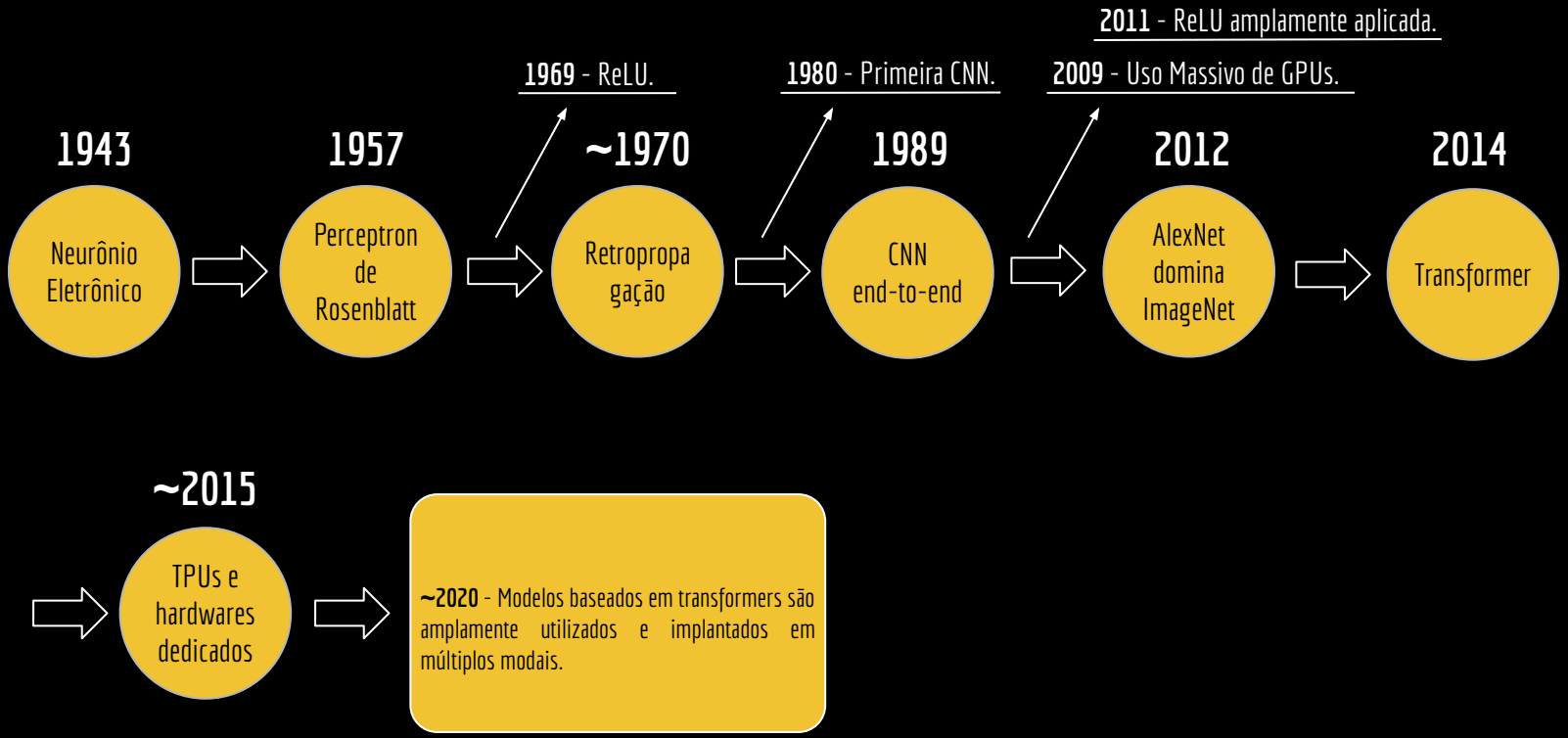
1943 Neurônio Eletrônico → **1957** Perceptron de Rosenblatt → **~1970** Retropropagação → **1989** CNN end-to-end → **2012** AlexNet domina ImageNet → **2014** Transformer

2015 - Patterson et. al.
Desenvolvimento de hardwares massivamente paralelos para ML, como TPUs.

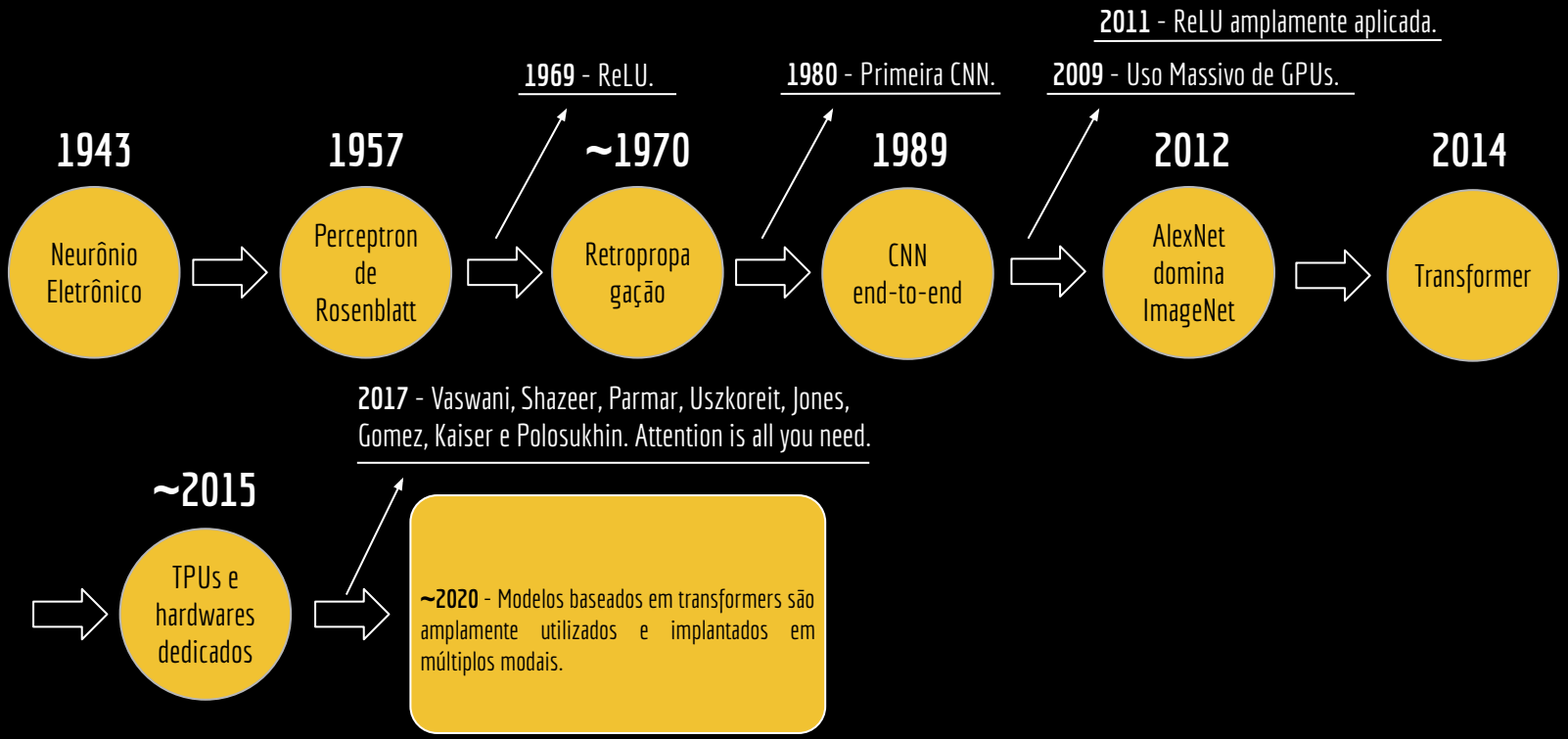
Timeline


2011 - ReLU amplamente aplicada.
1969 - ReLU.
1980 - Primeira CNN.
2009 - Uso Massivo de GPUs.
1943
Neurônio Eletrônico
1957
Perceptron de Rosenblatt
~1970
Retropropagação
1989
CNN end-to-end
2012
AlexNet domina ImageNet
2014
Transformer
~2015
TPUs e hardwares dedicados

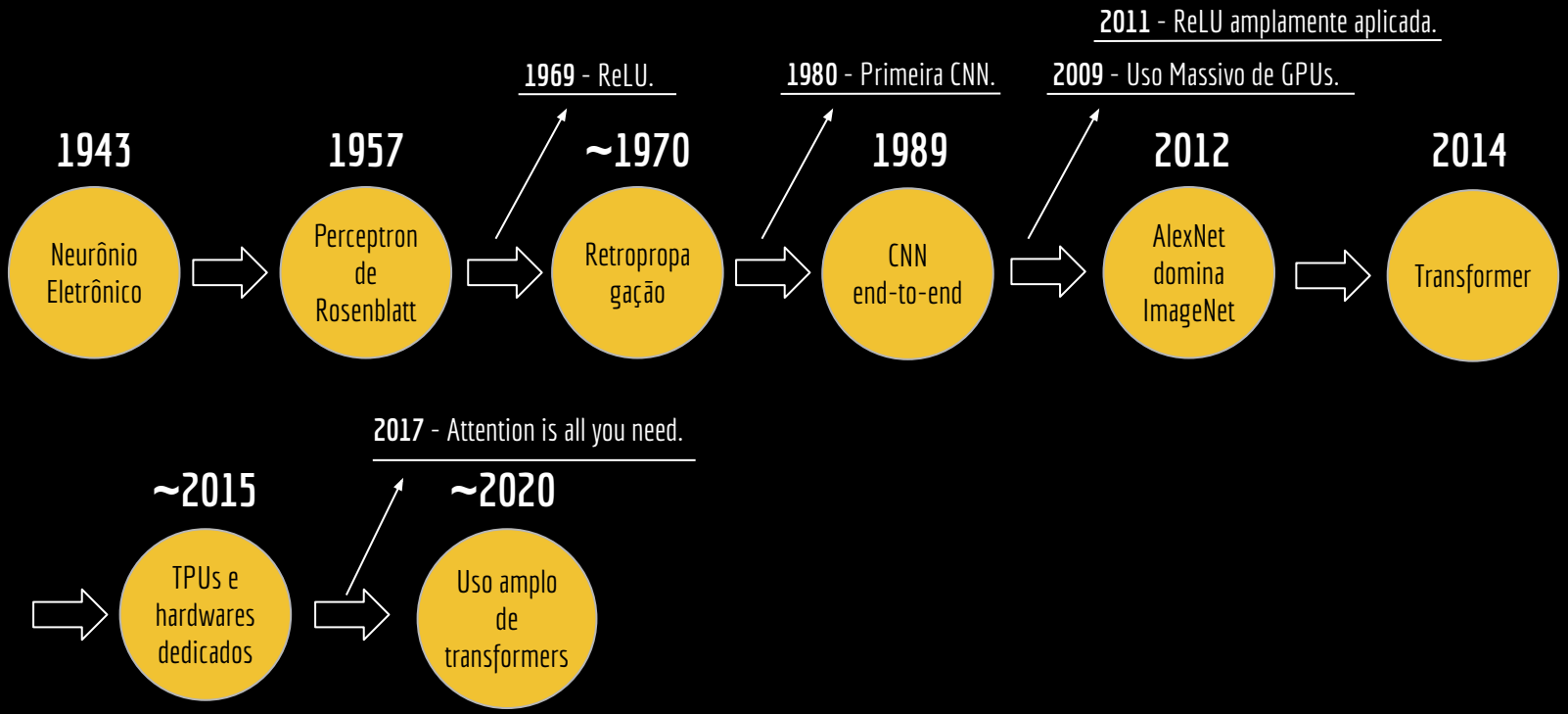
Timeline


2011 - ReLU amplamente aplicada.
1969 - ReLU.
1980 - Primeira CNN.
2009 - Uso Massivo de GPUs.
1943
Neurônio Eletrônico
1957
Perceptron de Rosenblatt
~1970
Retropropagação
1989
CNN end-to-end
2012
AlexNet domina ImageNet
2014
Transformer
~2015
TPUs e hardwares dedicados
~2020 - Modelos baseados em transformers são amplamente utilizados e implantados em múltiplos modais.

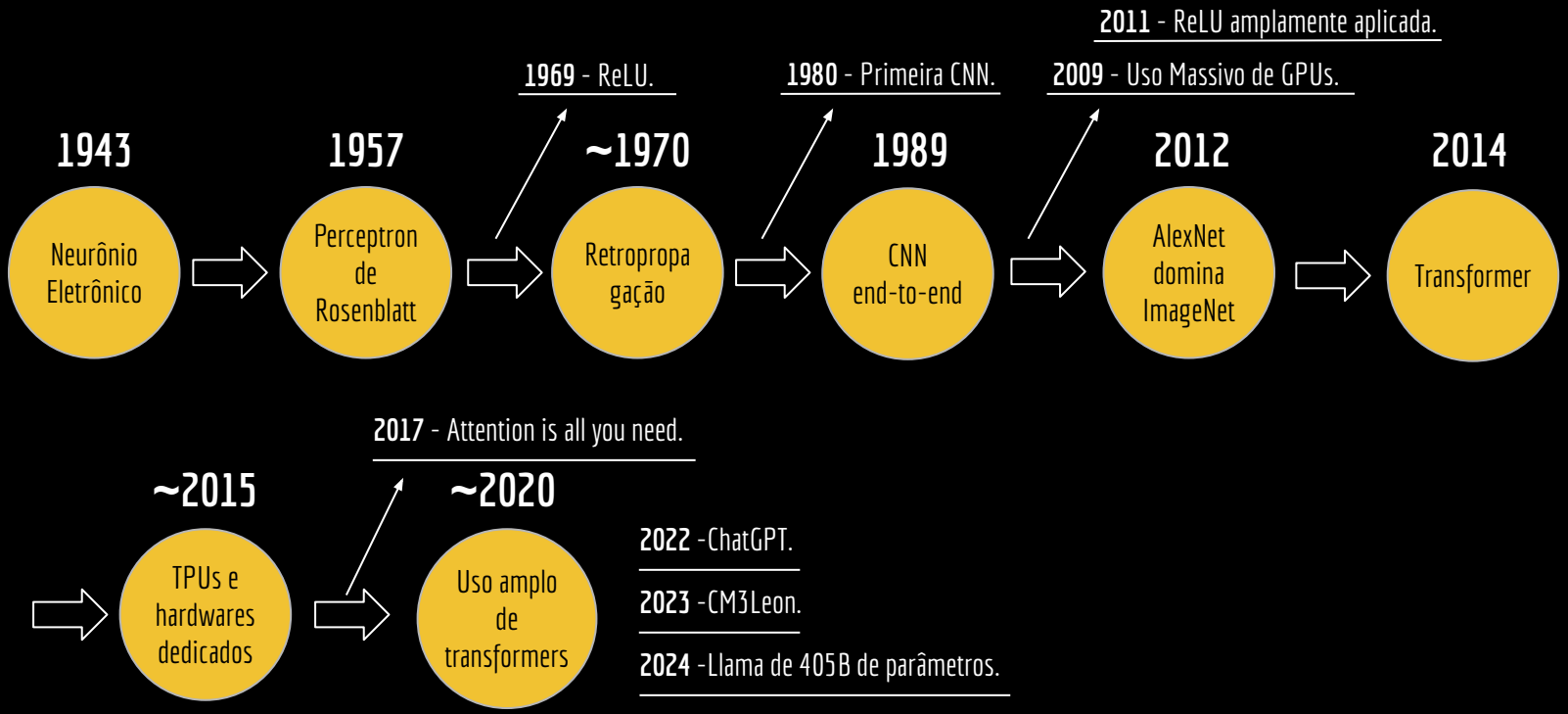
Timeline


1943
Neurônio Eletrônico
1957
Perceptron de Rosenblatt
1969 - ReLU.
~1970
Retropropagação
1980 - Primeira CNN.
1989
CNN end-to-end
2009 - Uso Massivo de GPUs.
2011 - ReLU amplamente aplicada.
2012
AlexNet domina ImageNet
2014
Transformer
2017 - Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser e Polosukhin. Attention is all you need.
~2015
TPUs e hardwares dedicados
~2020 - Modelos baseados em transformers são amplamente utilizados e implantados em múltiplos modais.

Timeline


1943
Neurônio Eletrônico
1957
Perceptron de Rosenblatt
1969 - ReLU.
~1970
Retropropagação
1980 - Primeira CNN.
1989
CNN end-to-end
2009 - Uso Massivo de GPUs.
2011 - ReLU amplamente aplicada.
2012
AlexNet domina ImageNet
2014
Transformer
~2015
TPUs e hardwares dedicados
2017 - Attention is all you need.
~2020
Uso amplo de transformers

Timeline


1943
Neurônio Eletrônico
1957
Perceptron de Rosenblatt
1969 - ReLU.
~1970
Retropropagação
1980 - Primeira CNN.
1989
CNN end-to-end
2009 - Uso Massivo de GPUs.
2011 - ReLU amplamente aplicada.
2012
AlexNet domina ImageNet
2014
Transformer
~2015
TPUs e hardwares dedicados
2017 - Attention is all you need.
~2020
Uso amplo de transformers
2022 - ChatGPT.
2023 - CM3Leon.
2024 - Llama de 405B de parâmetros.

2011 - ReLU amplamente aplicada.

1969 - ReLU. **1980** - Primeira CNN. **2009** - Uso Massivo de GPUs.

1943 Neurônio Eletrônico → **1957** Perceptron de Rosenblatt → **~1970** Retropropagação → **1989** CNN end-to-end → **2012** AlexNet domina ImageNet → **2014** Transformer

2017 - Attention is all you need.

~2015 TPUs e hardwares dedicados → **~2020** Uso amplo de transformers

2022 - ChatGPT.

2023 - CM3Leon.

2024 - Llama de 405B de parâmetros.

Algumas perguntas em aberto para › 2024:

Como reduzir o custo computacional?

Como aprender continuamente?

Como inserir raciocínio nos modelos?

Como modelar causalidade?

...

Referências

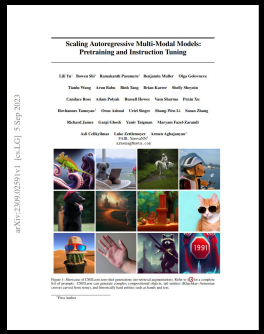
GULATI, Anmol et al. Conformer: Convolution-augmented transformer for speech recognition.2020.

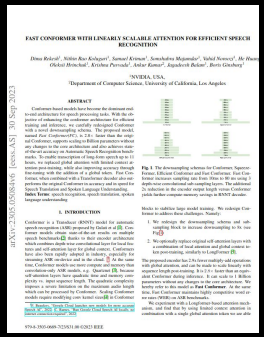
YAO, Zengwei et al. Zipformer: A faster and better encoder for automatic speech recognition. In: International Conference on Learning Representations. 2024.

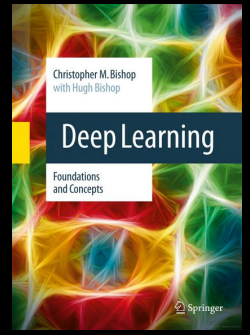
YU, Lili et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. arXiv preprint arXiv:2309.02591. 2023.

REKESH, Dima et al. Fast conformer with linearly scalable attention for efficient speech recognition. IEEE Automatic Speech Recognition and Understanding Workshop. 2023.

Bishop, C. M., Bishop, H. Deep Learning: Foundations and Concepts. 2023.







Licença