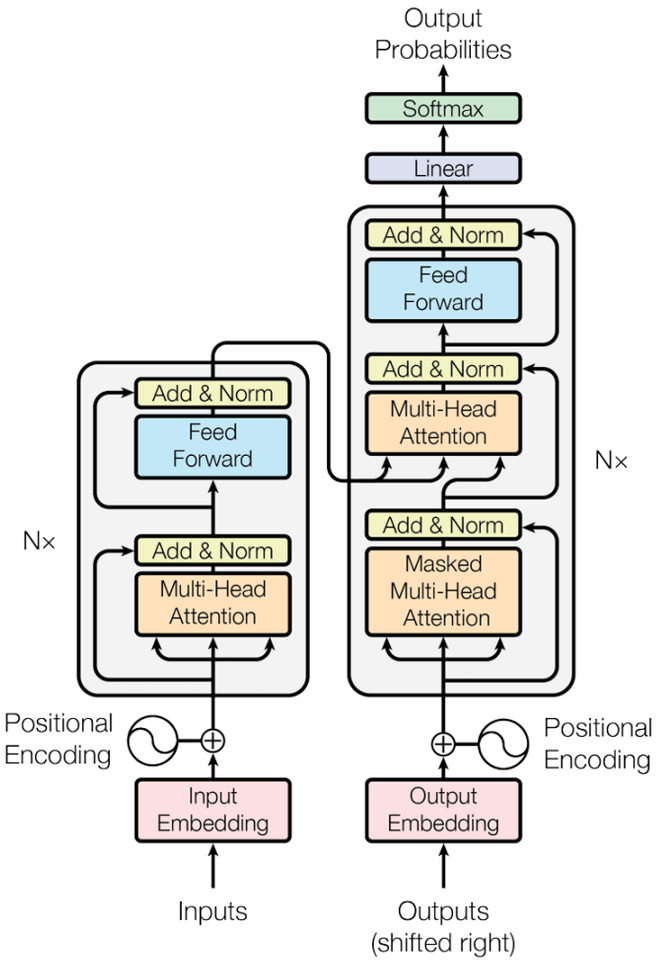
Microsoft Azure

<|start|>user<|message|> Como modelos linguísticos funcionam?

George Luiz Bittencourt



Sobre mim

SCAN ME

George Luiz Bittencourt

Arquiteto de soluções cloud com mais de 20 anos de experiência em infraestrutura e desenvolvimento de software.

/glzbcrt

/glzbcrt

george.bittencourt@microsoft.com

Quais os desafios da linguagem natural?

- **Ambiguidade**: Muitas palavras e frases têm múltiplos significados dependendo do contexto. Por exemplo, “banco” pode significar uma instituição financeira ou um assento.
- **Contexto e inferência**: Humanos usam conhecimento prévio e contexto para entender nuances, algo que é desafiador para máquinas.
- **Variação linguística**: Existem diferentes dialetos, gírias, erros de digitação e formas de expressão que tornam a linguagem imprevisível.
- **Implicitude**: Muitas vezes, informações são deixadas subentendidas em uma conversa, exigindo interpretação além do que está explicitamente dito.
- **Estrutura complexa**: A gramática e a sintaxe das línguas naturais são altamente complexas e cheias de exceções.

O PROBLEMA EM SER PROGRAMADOR

MINHA MULHER DISSE:
– AMOR, VÁ ATÉ O MERCADO E COMPRE 1 GARRAFA DE LEITE. SE ELES TIVEREM OVOS, TRAGA 6

~ EU VOLTEI PARA CASA COM 6 GARRAFAS DE LEITE ~

ELA DISSE:
– PORQUE DIABOS VOCÊ COMPROU 6 GARRAFAS DE LEITE?

EU RESPONDI:
– PORQUE ELES TINHAM OVOS.

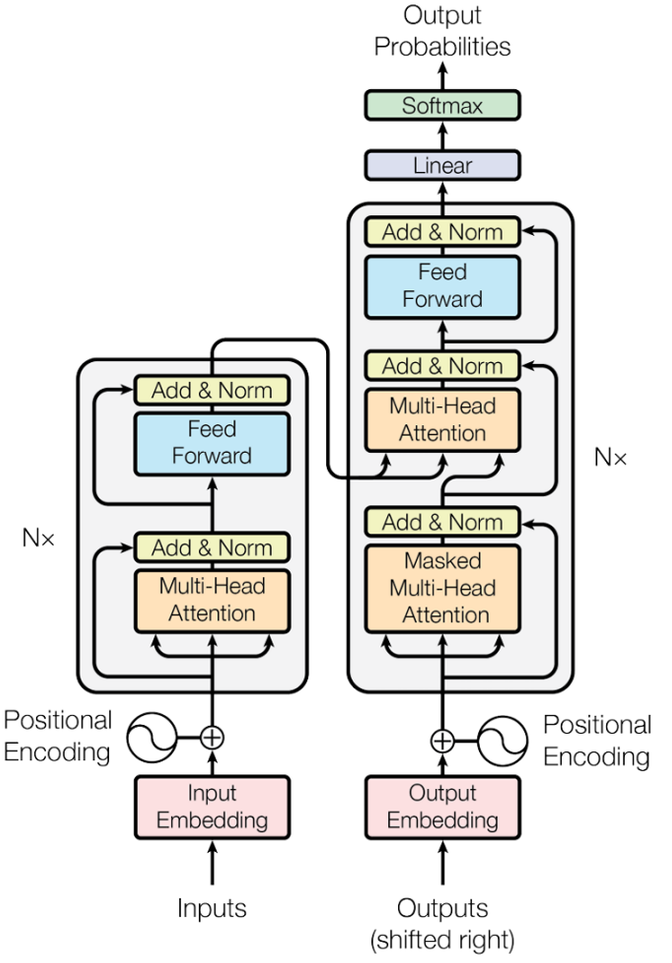
Attention Is All You Need

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after



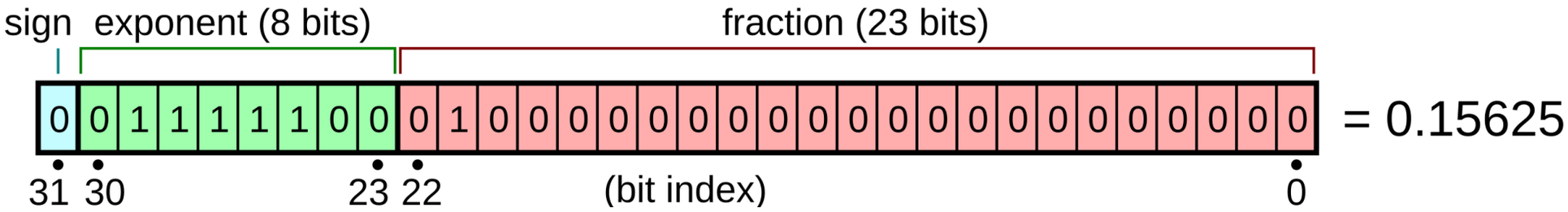
O que é um modelo?

- É uma **fórmula matemática** com **múltiplas variáveis** de entrada e saída **otimizada** a partir de **exemplos anotados**.

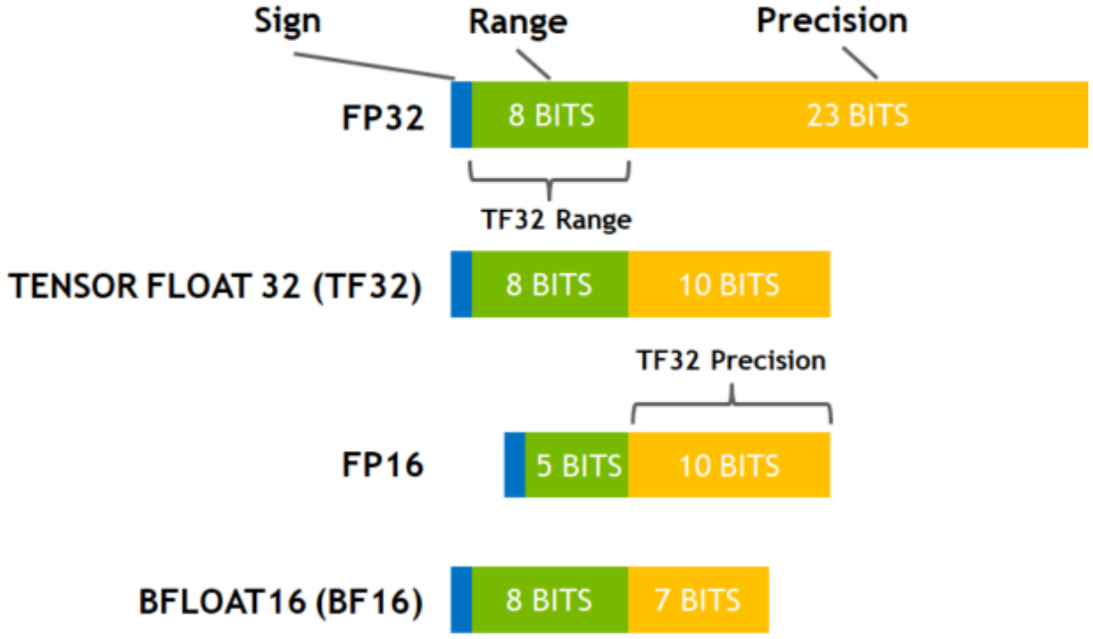
$$Y = in^0 w^0 + in^1 w^1 + in^2 w^2 + in^n w^n + \beta$$

- Um modelo é composto de **parâmetros**, que são **números** encontrados durante o processo de **treinamento** que tem por objetivo **reduzir o erro** entre o **valor calculado** e o **valor anotado**.
- **Os valores de entrada e saída precisam ser numéricos**, já que são utilizados em milhares de cálculos matemáticos. Etapas de pré-processamento e pós-processamento fazem as conversões necessárias.
- Durante a execução milhões de cálculos são executados. Esses cálculos podem ser executados tanto na CPU quanto utilizando aceleradores como as GPUs da NVIDIA ou ainda TPUs. A unidade FLOPS é muito importante, pois mede quantos cálculos é possível fazer por segundo. Quanto maior, menor será a latência da inferência.

Números



Fonte: Single-precision floating-point format – Wikipedia

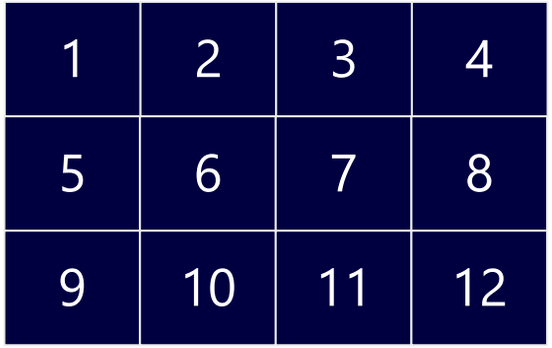


Fonte: Accelerating AI Training with NVIDIA TF32 Tensor Cores | NVIDIA Technical Blog

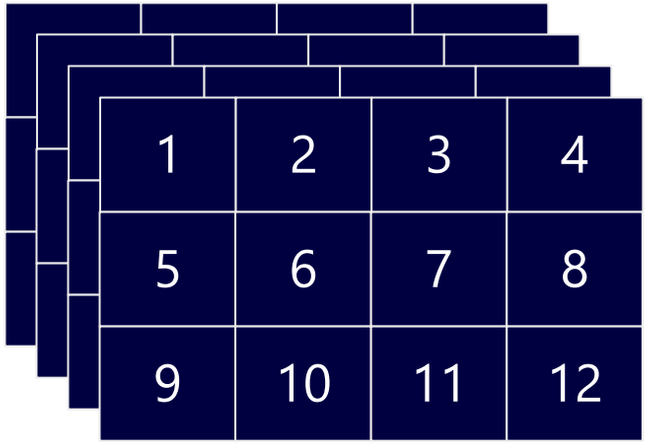
Como o dado é representado?

escalar

vetor



matriz



cubo

TENSOR

- Não existe limite na quantidade de dimensões, porém o cérebro humano lida com até 3 dimensões facilmente.
- Um tensor tem um formato, que é a quantidade de dimensões e elementos por dimensão.
- Um local. Ele pode estar armazenado na memória RAM ou ainda na GPU/TPU.
- + um tipo de dado que define a faixa de valores e quantidade de memória necessária. Os mais comuns:
  - FP32
  - FP16
  - BF16
  - INT8

Do que é feito um modelo?

Camada de entrada

```
return tf.keras.Sequential([
    tf.keras.layers.Rescaling(
        1./255, input_shape=(img_height, img_width, img_channels)),
    tf.keras.layers.Conv2D(16, 3, padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(len(output_classes), name="output")
])
```

Camada de saída

Layers

- *Cada layer parâmetros, aprendidos treinamento.*
- *Algumas layers não possuem parâmetros, com a Dropout e Flatten.*

$$Y = in^0w^0 + in^1w^1 + in^2w^2 + in^nw^n + \beta$$

Outra forma de escrever o modelo:

```
output = Dense(Dense(Dropout(Flatten(MaxPooling2D(Conv2D(MaxPooling2D(Conv2D(MaxPooling2D(Conv2D(Rescaling(input)))))))))))
```

Como o modelo aprende?

- Da mesma maneira que nós! Ao sermos expostos a vários exemplos corretos e incorretos conseguimos criar regras para entender.
- No caso do computador ele calcula para cada exemplo o quão errado ele estava e ajusta seus parâmetros para na próxima iteração reduzir o erro.
- Existem várias formas de calcular o erro, sendo o MSE uma das mais comuns.

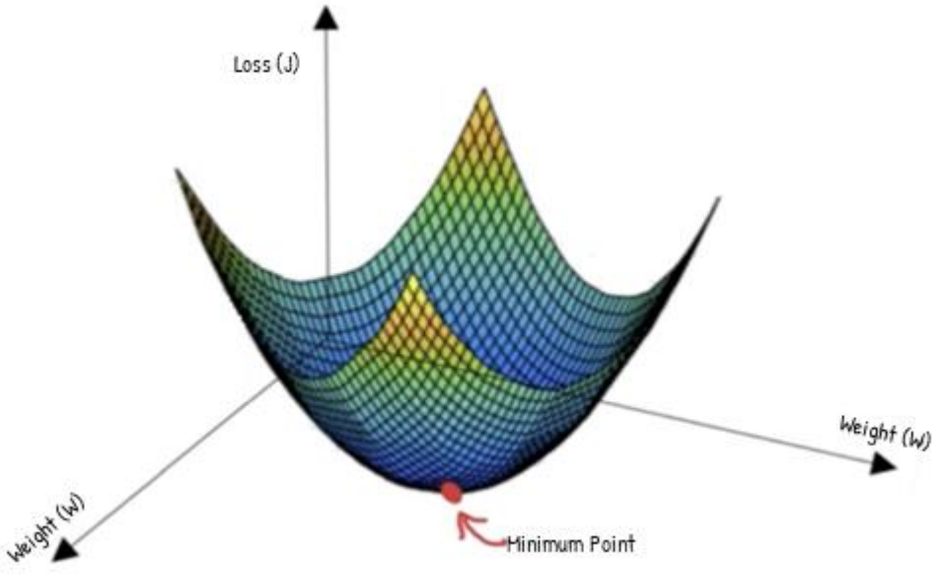
$$Y = in^0w^0 + in^1w^1 + in^2w^2 + in^nw^n + \beta$$

Exemplo	Valor Esperado	Valor Calculado #1	Valor Calculado #2	Valor Calculado #N
1	5	2	4	5.3
2	6	15	8	6.7
3	2	2.3	2.7	2.1
4	1	7	3	0.6
5	7	2	9	6.6

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Como o modelo aprende?

- Durante o treinamento do modelo o **otimizador** em conjunto com o **algoritmo de erro** busca os **melhores valores**. Somente os valores iniciais são aleatórios.
- Depois da primeira iteração utilizando matemática diferencial e calculando os gradientes eles são aplicados aos parâmetros gerando novos valores e tudo recomeça. Esse passo é conhecido como *backpropagation* e é um dos princípios mais importantes em redes neurais.
- O objetivo é buscar o conjunto de parâmetros que reduz ao menor valor possível o erro **dentro do tempo alocado** para o treinamento.
- Existem **várias** respostas para o mesmo *dataset*.



Fonte: Gradient Descent vs. Backpropagation: What's the Difference? (analyticsvidhya.com)

Acesso em: 11/03/2024

Alguns datasets utilizados

Nome	Tamanho Aproximado
Common Crawl	60 TB
The Pile	825 GB
C4 (Colossal Clean Crawled)	750 GB
Wikipedia	20 GB
BookCorpus	6 GB
Red Pajama	1,2 TB
RefinedWeb	600 GB
OpenWebText	40 GB

Life of a Prompt

How a prompt gets processed

API REQUEST

```
{
    max_tokens = 5,
    who founded microsoft?
}
```

TOKENIZER

Key Metrics

Latency → TTFT + (TPOT X # of generated tokens)
Throughput → # of tokens generated by unit of time

Context, Prompt

14965	18538	80996	30	30742	35634	323	68111	51141

TTFT = Time to First Token

TPOT = Time per Output Token

TOKENIZER

bill gates and paul allen

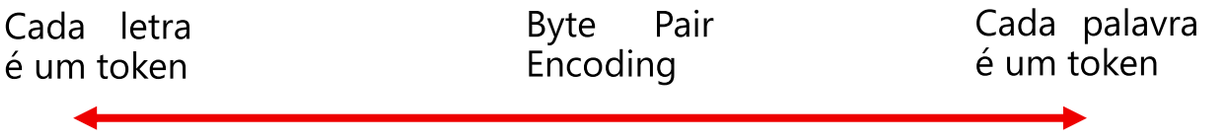
API RESPONSE

😉

You can see it in action in the Code Llma open-source model here.

Tokenizer

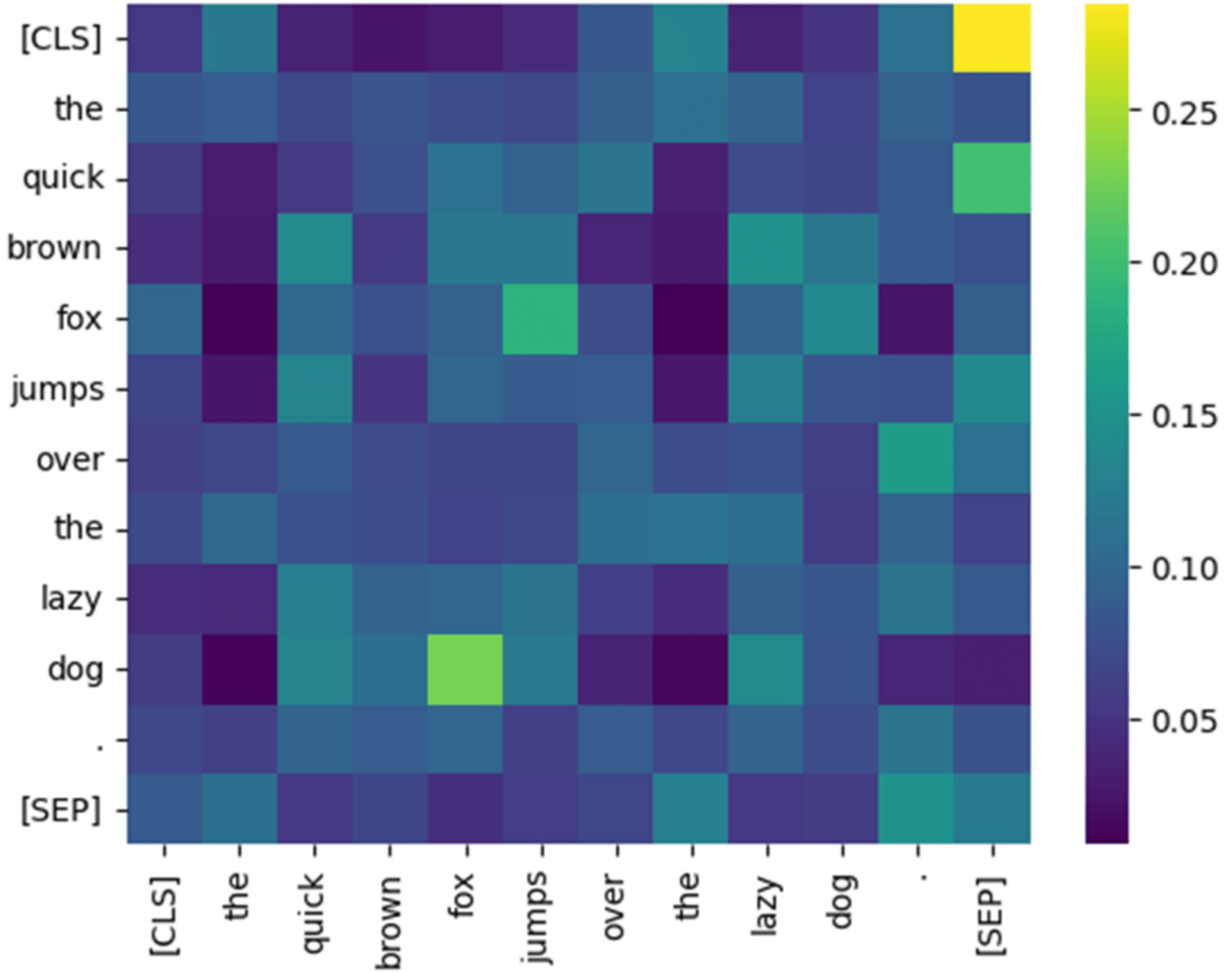
- É usado no início e no fim da requisição ao modelo, pois converte um texto em um vetor de números e o contrário, um vetor de números em um texto.
- Todo tokenizer tem um **vocabulário**, ou seja, um conjunto de tokens que o tokenizer suporta. No caso do gpt-oss o vocabulário tem **201.088 tokens**.




```
GitHub Copilot adapts to your unique needs allowing you to select the best model
 for your project, customize chat responses with custom instructions, and utilize
 agent mode for AI-powered, seamlessly integrated peer programming sessions.
```

```
[56279, 27052, 28412, 56775, 38082, 1561, 316, 634, 5746, 4414, 16246, 481, 316,
4736, 290, 1636, 2359, 395, 634, 2993, 11, 38440, 7999, 22488, 483, 2602, 15543,
11, 326, 24570, 11793, 6766, 395, 20837, 69943, 11, 77640, 21781, 24770, 23238,
19791, 13]
```

Matriz de Atenção



Fonte: How to Visualize Model Internals and Attention in Hugging Face Transformers – Kdnuggets

Logits e Sampling

Parabéns para você →

Vocabulário

Token	Probabilidade
nessa	85%
nesta	6%
por	4%
joinville	3%
...	

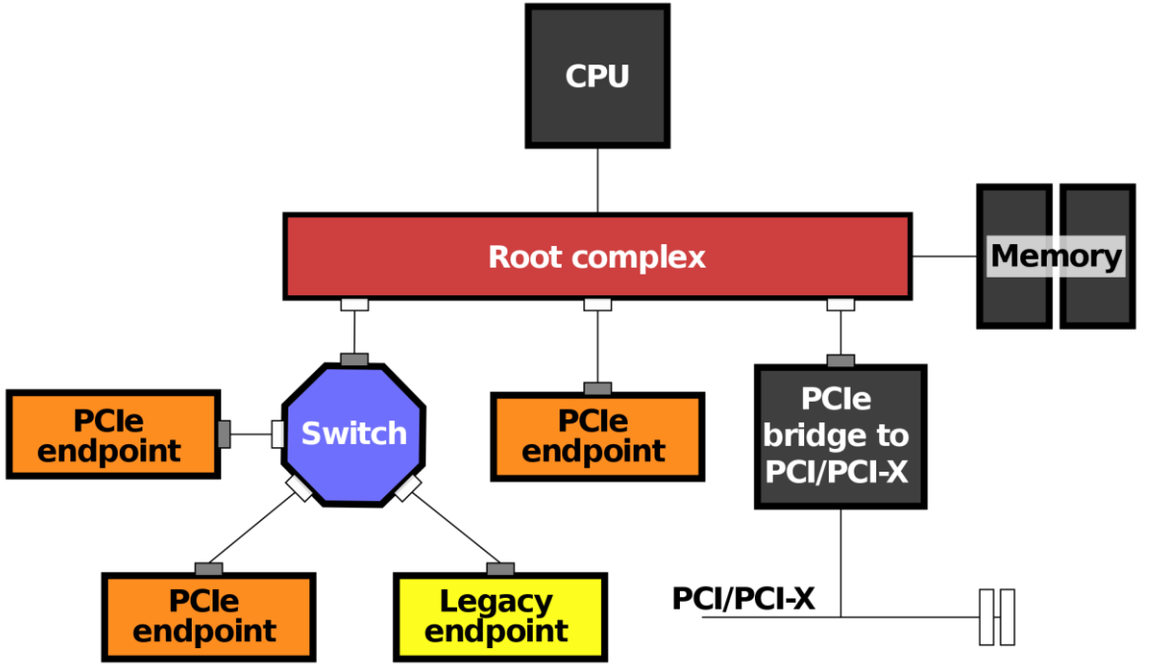
Parabéns para você nessa →

Vocabulário

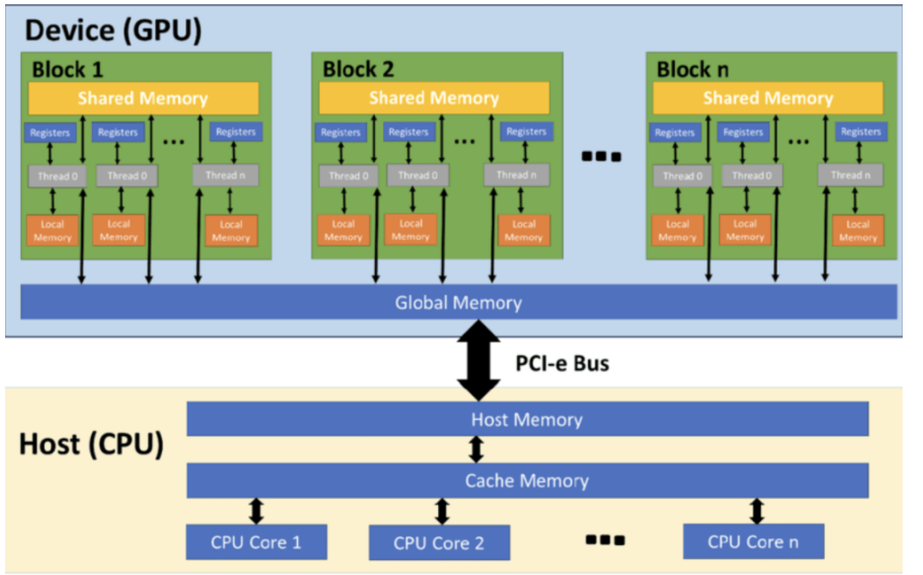
Token	Probabilidade
data	75%
escolha	15%
nova	10%
morango	0%
...	

- **Temperatura**: altera os logits e com isso tokens com baixa probabilidade ganham um amento. Quanto maior, mais "criativo" o processo se torna, pois tokens com baixa probabilidade podem ser selecionados.
- **Top P**: seleciona os tokens onde o acumulado da probabilidade é maior que o valor parametrizado.
- **Top K**: seleciona os # primeiros tokens com maior probabilidade.

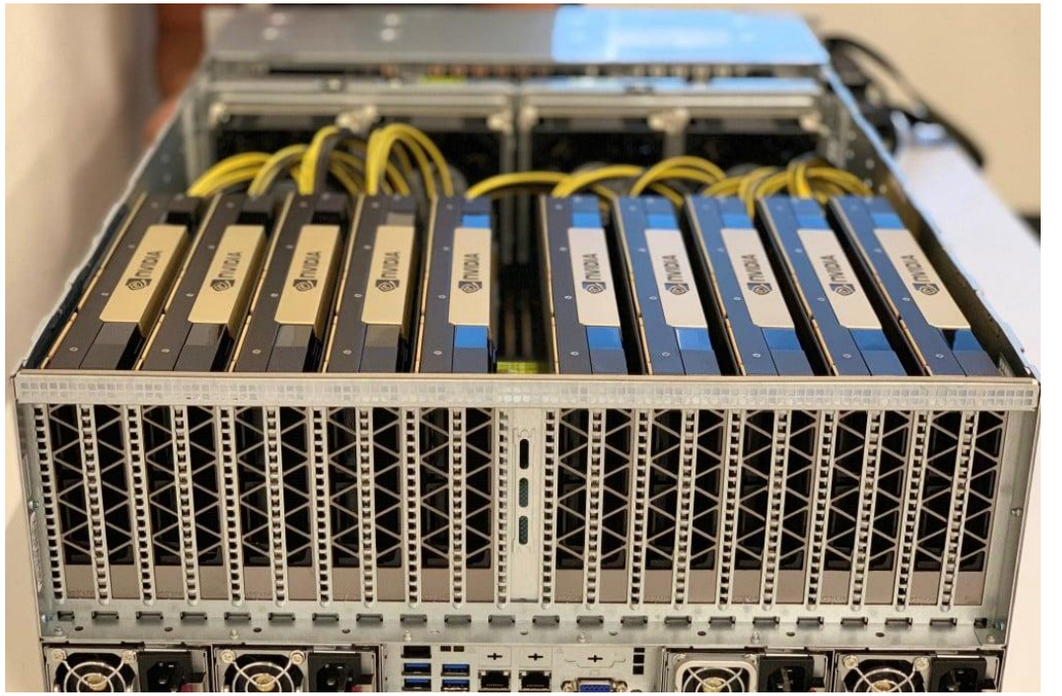
GPU Is All You Need!



Fonte: [PCIe / PCI Express: what it is and terminology « Adafruit Industries – Makers, hackers, artists, designers and engineers!]



Fonte: [A Technical Deep Dive Into CPU & GPU Internals – Tamal Dutta Chowdhury]



Microsoft Azure

EOP

/glzbcrt

/glzbcrt

george.bittencourt@microsoft.com

SCAN ME