Transformers 2

Frederick Vandermoeten

Overview

- Multi-Head Attention
- Complete Transformer Model
- Cross-Attention
- Training
- Transformer Variants
- Optimizations

Multi-Head Attention

- many Attention Heads in parallel create one Layer

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O$$

- each head uses a linear transformation on Q, K and V

$$\mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Ouput Matrix Splitting - Example GPT3

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O$$

$$\mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$d_{Embedding} = d_E = 12{,}288; d_k = 128$

W_i^Q and $W_i^K \in \mathbb{R}^{d_E \times d_k}$

$d_e * d_k = 12{,}288 * 128 = 1{,}572{,}864$

$W_{i,pre}^V \in \mathbb{R}^{d_E \times d_E}$

$d_e * d_e = 12{,}288 * 12{,}288 = 150{,}994{,}944$

Ouput Matrix Splitting - Example GPT3

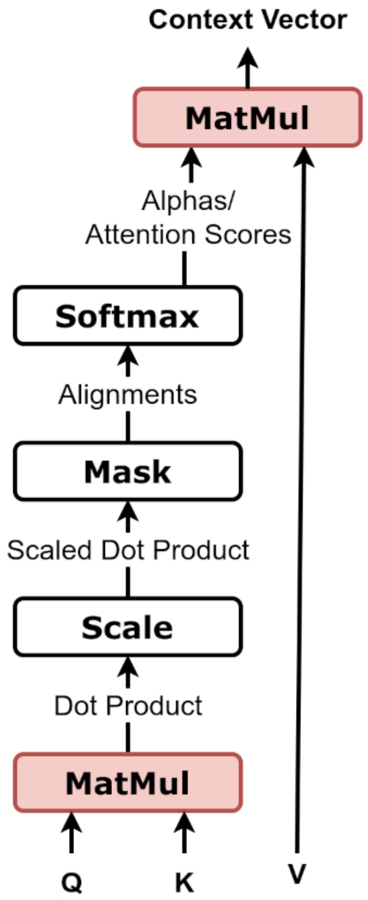
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O$$

$$\mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

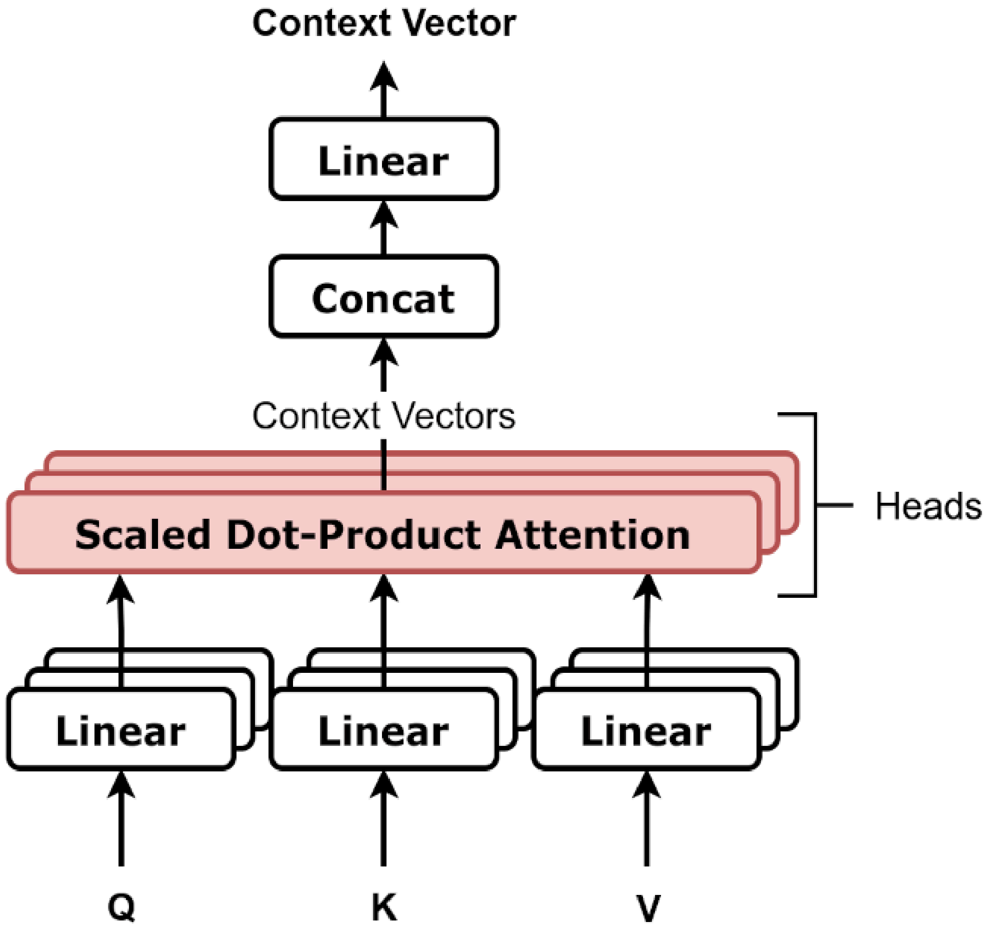
W_i^Q and $W_i^K \in \mathbb{R}^{d_E \times d_k}$
$d_e * d_k = 12{,}288 * 128 = 1{,}572{,}864$
$W_{i,pre}^V \in \mathbb{R}^{d_E \times d_E}$
$d_e * d_e = 12{,}288 * 12{,}288 = 150{,}994{,}944$
We split $W_{i,pre}^V$ into $W_i^V \in \mathbb{R}^{d_E \times d_v}$ and $W_i^O \in \mathbb{R}^{d_v \times d_E}$ with $W_{i,pre}^V = W_i^V W_i^O$
If we use $d_v = d_k$, we get the same amount of parameters for W_i^V and W_i^O as for W_i^Q and W_i^K

$$W^O = \mathrm{Vertcat}(W_1^O, \ldots, W_h^O)$$

Multi-Head Attention 2

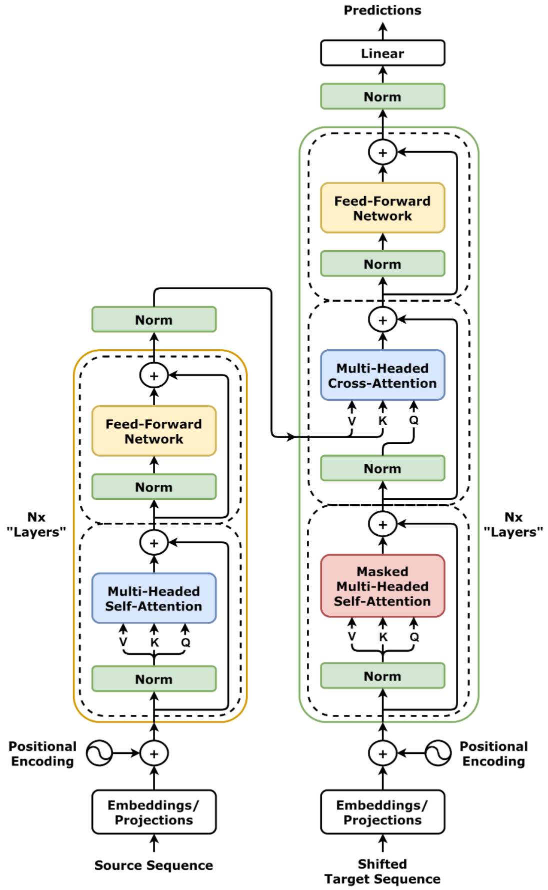


Scaled Dot-Product
Attention



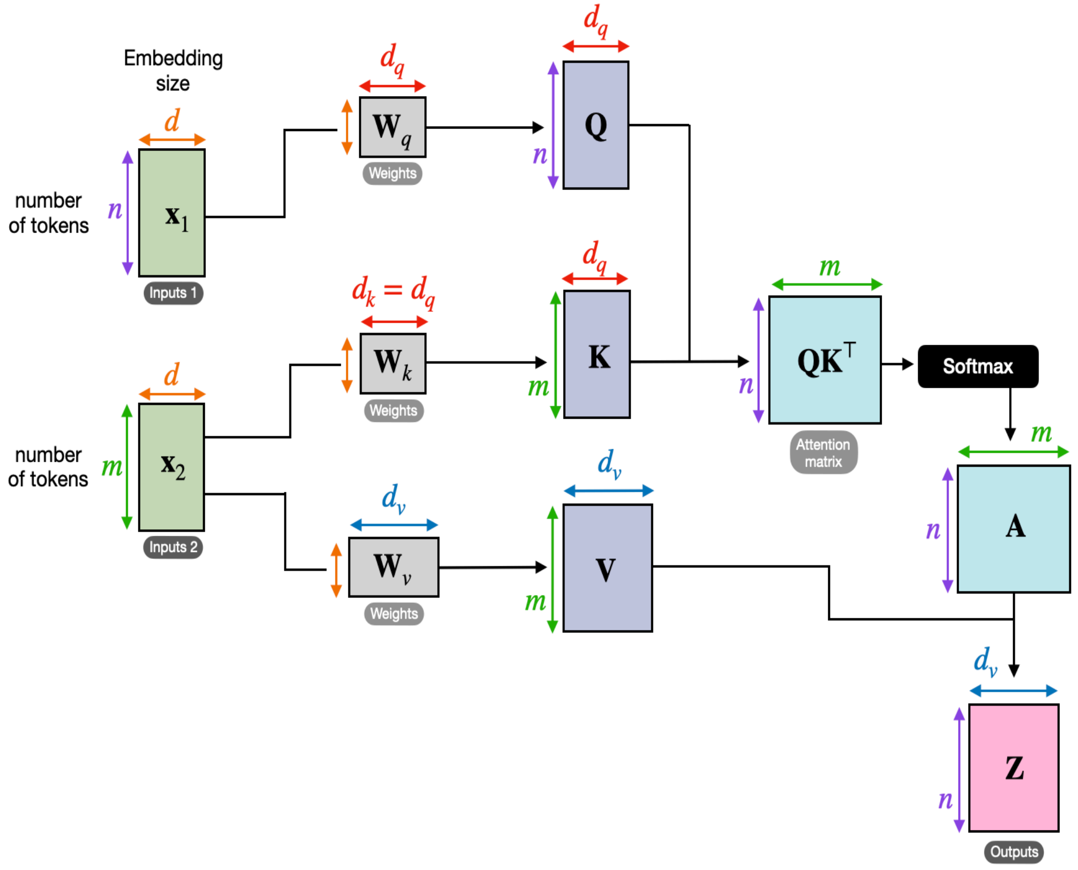
Complete Transformer Model

- Encoder-Decoder architecture
- Embedding
- Positional Encoding
- Multi-Head Attention
- Feed Forward Network
- Skip Connections
- Layer Normalization
- Cross-Attention



Cross-Attention

- Enables interaction between encoder and decoder
- Decoder uses encoder output as input
- Query comes from the decoder, key and value from the encoder
- Encoder and decoder could be from different modals (text, image, video)

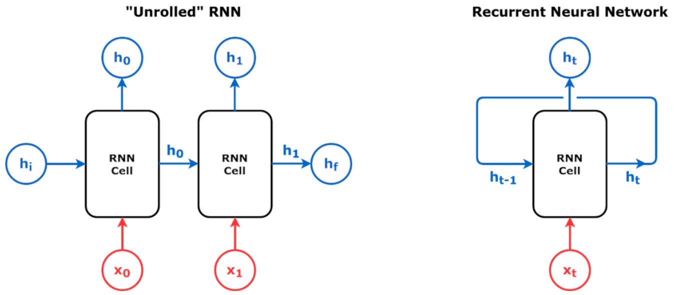


RAS2024 (modified)

Training

Parallelization

- Unlike RNNs, no sequential dependencies between inputs
- Just a bunch of matrix multiplications that can be parallelized very well with modern hardware (GPUs, TPUs)
- We still get order information due to the positional encoding
- Efficient use of training data



Training: Example GPT-2 vs GPT-3

GPT-2

- 1.5B parameters
- 40 Gb of data

GPT-3

- 176B parameters
- 570 Gb of data

Transformer Variants

Transformers were initially designed for NLP, but have since been applied to other tasks as well.

Text Transformers

GPT - Generative Pre-trained Transformer

- Decoder-only Transformer
- Used for text generation, summarization, question answering, etc.

BERT - Bidirectional Encoder Representations from Transformers

- Encoder-only Transformer
- Used for text classification, sentiment analysis, etc.

T5 - Text-to-Text Transfer Transformer

- Encoder-Decoder Transformer
- Less specialized so it needs fine-tuning, useful for e.g. translation

Vision Transformers

ViT - Vision Transformer

- Images are encoded in patches, then the patches are fed into the Transformer
- Self-Attention is used to learn the global relationships
- Used for image classification, object detection, etc.

Stable Diffusion

- Diffusion models are encoder-decoder models that use noise for the hidden state
- Cross-Attention is used to integrate text prompts
- Used for image generation, image inpainting, etc.

Audio Transformers

Wav2Vec

- Operates directly on waveforms
- Speech recognition, especially for low-resource languages

Audio Spectrogram Transformer

- Uses ViT on spectrograms
- Audio classification (genre, instrument, etc.), speech emotion recognition, etc.

Optimizations

Sparse Transformers

- Attention Layers grow $O(n^2)$ for the context size n
- most results of $\text{softmax}(\frac{Q^T K}{\sqrt{d_k}})$ are close to zero
- different ways to reduce the scaling by only considering close values or random feature attention

Model Distillation

- Train a smaller, faster model with data generated by a larger, slower model
- Teacher/ Student Model

Conclusion

- Transformers revolutionized NLP
- Attention is what drives their success
- They are versatile and can be used for many different tasks
- They scale well to large datasets
- Work in progress

Image Sources

All not locally credited images are from dvgodoy: Deep Learning Visuals CC BY

Main Conceptual Sources

- Vaswani, A. "Attention is all you need." Advances in Neural Information Processing Systems (2017).
- Raschka: Understanding and coding self-attention
- 3Blue1Brown: Neural Networks Playlist
- SBert Wiki