5. Transformers

Computational Music Creativity

Universitat Pompeu Fabra Barcelona

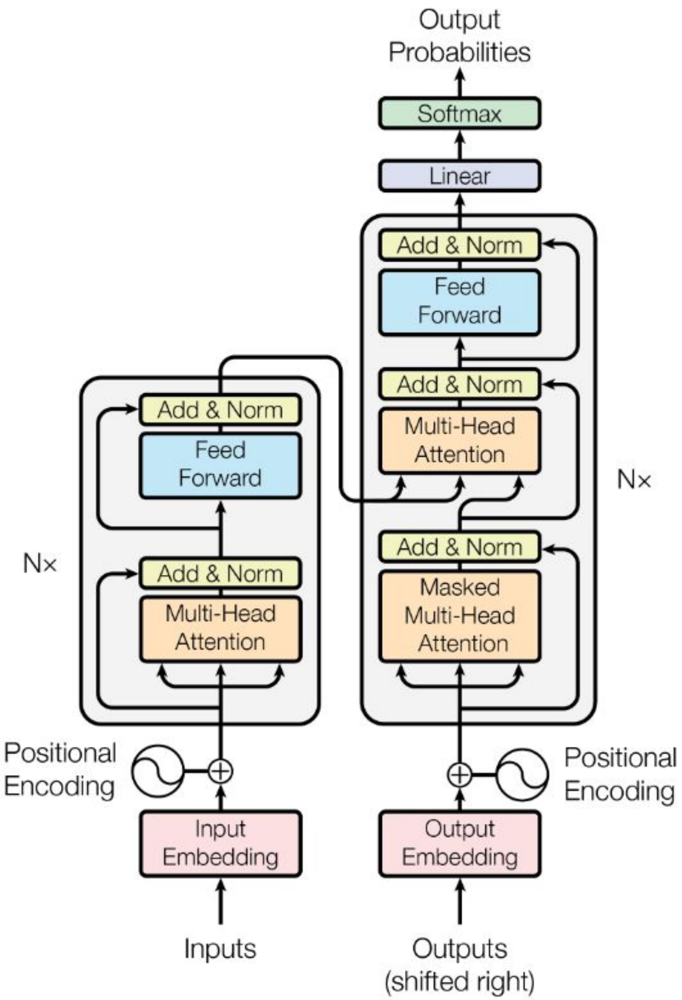
MTG Music Technology Group


ANY AI
MUSIC ENGINEER
60+ YEARS
OF SOLID
TECHNIQUES
NEW,
SHINY AI MODEL
imgflip.com


THEY'RE EATING THE CATS!
LIVE
NEWS
2024 PRESIDENTIAL DEBATE


Pet Savers
Pearlyarts

Real-time scores


Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)

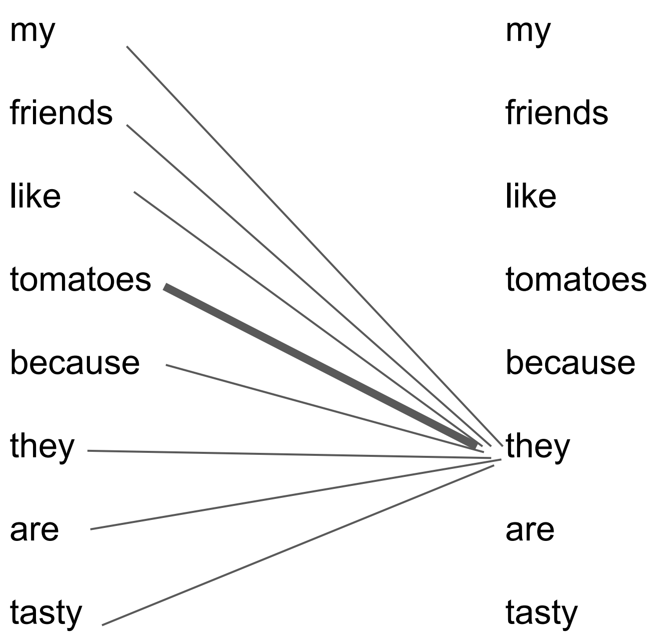
A reference problem

My friends like tomatoes because they are tasty

A reference problem

My friends like tomatoes because they are tasty

Self-attention: Intuition



What matrices do we have in self-attention?

Query, key, value matrices

Query (Q)

$$\begin{array}{c} \text{I} \\ \text{like} \\ \text{cats} \end{array} \begin{bmatrix} 1.3 & 0.8 \\ 0.7 & 3.5 \\ 1.9 & 0.1 \end{bmatrix}$$

Key (K)

$$\begin{array}{c} \text{I} \\ \text{like} \\ \text{cats} \end{array} \begin{bmatrix} 0.6 & 2.4 \\ 0.8 & 1.7 \\ 2.5 & 0.3 \end{bmatrix}$$

Value (V)

$$\begin{array}{c} \text{I} \\ \text{like} \\ \text{cats} \end{array} \begin{bmatrix} 0.4 & 1.0 \\ 1.2 & 2.8 \\ 1.7 & 0.2 \end{bmatrix}$$

How do we derive Q, K, V?

How do we derive Q, K, V?

- Multiply input matrix by 3 weight matrices
- Learn weights during training

$$IW_Q = Q$$
$$IW_K = K$$
$$IW_V = V$$

Self-attention: Formalisation

$$Z(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

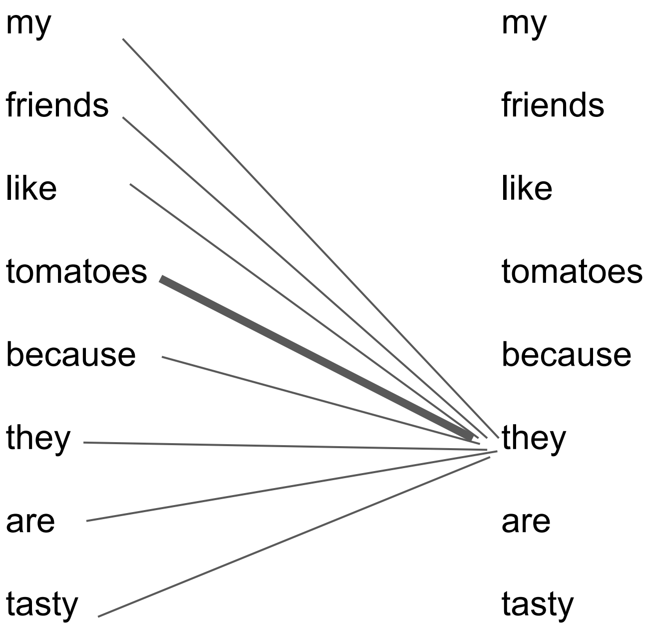
Self-attention: Step 1

$$Z(Q, K, V) = \mathrm{softmax}\left(\frac{\boxed{QK^T}}{\sqrt{d_k}}\right) V$$

Self-attention: Step 1

$$QK^T = \begin{array}{c c c} & & \\ \begin{array}{c} \text{I} \\ \text{like} \\ \text{cats} \end{array} & \underbrace{\begin{bmatrix} 1.3 & 0.8 \\ 0.7 & 3.5 \\ 1.9 & 0.1 \end{bmatrix}}_{Q} & \begin{array}{c} q_1 \\ q_2 \\ q_3 \end{array} \end{array} \begin{array}{ccc} \text{I} & \text{like} & \text{cats} \\ \multicolumn{3}{c}{\underbrace{\begin{bmatrix} 0.6 & 0.8 & 2.5 \\ 2.4 & 1.7 & 0.3 \end{bmatrix}}_{K^T}} \\ k_1 & k_2 & k_3 \end{array} = \begin{bmatrix} q_1k_1 & q_1k_2 & q_1k_3 \\ q_2k_1 & q_2k_2 & q_2k_3 \\ q_3k_1 & q_3k_2 & q_3k_3 \end{bmatrix} = \begin{array}{cc} & \begin{array}{ccc} \text{I} & \text{like} & \text{cats} \end{array} \\ \begin{array}{c} \text{I} \\ \text{like} \\ \text{cats} \end{array} & \begin{bmatrix} 2.7 & 2.4 & 3.49 \\ 8.82 & 6.51 & 2.8 \\ 1.38 & 1.69 & 4.78 \end{bmatrix} \end{array}$$

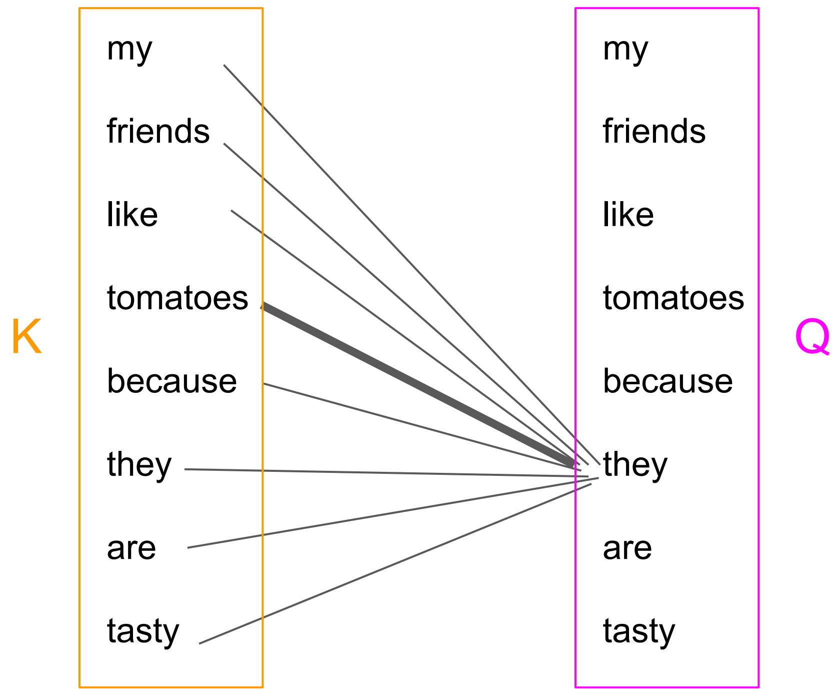
What are Q and K really?



Self-attention: Step 2

$$Z(Q, K, V) = \mathrm{softmax}\left(\boxed{\frac{QK^T}{\sqrt{d_k}}}\right)V$$

What are Q and K really?



Self-attention: Step 3

$$Z(Q, K, V) = \boxed{\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)} V$$

Self-attention: Step 3

- Normalize similarity scores
- Apply *softmax*
- Each word vector (row) adds up to 1 (probability)

$$\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) = \begin{array}{c c} & \begin{array}{ccc} \text{I} & \text{like} & \text{cats} \end{array} \\ \begin{array}{c} \text{I} \\ \text{like} \\ \text{cats} \end{array} & \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix} \end{array}$$

*values in the matrix completely made up

Self-attention: Step 3

$$\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

Self-attention: Step 3

$$\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

Attention score

Relevance of different parts of the sequence to each other

Self-attention: Step 4

$$Z(Q, K, V) = \boxed{\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V}$$

Self-attention: Step 4

$$Z = \begin{array}{c} \\ \text{I} \\ \text{like} \\ \text{cats} \end{array} \begin{array}{c} \begin{array}{ccc} \text{I} & \text{like} & \text{cats} \end{array} \\ \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix} \end{array} \begin{array}{c} \text{I} \\ \text{like} \\ \text{cats} \end{array} \begin{bmatrix} 0.4 & 1.0 \\ 1.2 & 2.8 \\ 1.7 & 0.2 \end{bmatrix} \begin{array}{c} v_1 \\ v_2 \\ v_3 \end{array} = \begin{array}{c} \text{I} \\ \text{like} \\ \text{cats} \end{array} \begin{bmatrix} 0.69 & 1.28 \\ 1.14 & 1.92 \\ 1.13 & 0.78 \end{bmatrix} = \begin{bmatrix} \vec{z_1} \\ \vec{z_2} \\ \vec{z_3} \end{bmatrix}$$

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \qquad V$$

Self-attention for word “I”

$$Z = \begin{array}{cc} & \begin{array}{ccc}\text{I} & \text{like} & \text{cats}\end{array} \\ \begin{array}{c}\text{I}\\ \text{like}\\ \text{cats}\end{array} & \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix} \end{array} \begin{array}{c}\text{I}\\ \text{like}\\ \text{cats}\end{array} \begin{bmatrix} 0.4 & 1.0 \\ 1.2 & 2.8 \\ 1.7 & 0.2 \end{bmatrix} \begin{array}{c}v_1\\ v_2\\ v_3\end{array} = \begin{array}{c}\text{I}\\ \text{like}\\ \text{cats}\end{array} \begin{bmatrix} 0.69 & 1.28 \\ 1.14 & 1.92 \\ 1.13 & 0.78 \end{bmatrix} = \begin{bmatrix} \vec{z}_1 \\ \vec{z}_2 \\ \vec{z}_3 \end{bmatrix}$$

$$\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \qquad V$$

$$\vec{z}_1 = \underset{\text{I}}{0.7\vec{v}_1} + \underset{\text{like}}{0.2\vec{v}_2} + \underset{\text{cats}}{0.1\vec{v}_3} = 0.7\underset{\text{I}}{\begin{bmatrix}0.4 & 1.0\end{bmatrix}} + 0.2\underset{\text{like}}{\begin{bmatrix}1.2 & 2.8\end{bmatrix}} + 0.1\underset{\text{cats}}{\begin{bmatrix}1.7 & 0.2\end{bmatrix}}$$

Sum of the value vectors weighted by the scores

A reference problem: Solved

My friends like tomatoes because they are tasty

$$\vec{z}_{they} = \underset{\text{my}}{0.0\vec{v}_1} + \underset{\text{friends}}{0.0\vec{v}_2} + \underset{\text{like}}{0.0\vec{v}_3} + \underset{\text{tomatoes}}{0.9\vec{v}_4} + \underset{\text{because}}{0.0\vec{v}_5} + \underset{\text{they}}{0.1\vec{v}_6} + \underset{\text{are}}{0.0\vec{v}_7} + \underset{\text{tasty}}{0.0\vec{v}_8}$$

What's multi-head attention?

What's multi-head attention?

- Run multiple instances of the self-attention mechanism in parallel
- Compute as many Q, K, V, Z matrices as the number of heads

$$Z = concatenate(Z_1, Z_2, Z_3, ..., Z_n)W_0$$


WHY MULTIPLE HEADS?
imgflip.com

Why positional encoding?

Positional encoding: Strategy

$$I' = \underbrace{\begin{bmatrix} 0.2 & 1.2 \\ 0.5 & 4.1 \\ 2.1 & 0.4 \end{bmatrix}}_{I} + \underbrace{\begin{bmatrix} 0.5 & 1.0 \\ 2.5 & 1.3 \\ 1.1 & 0.3 \end{bmatrix}}_{P} = \begin{bmatrix} 0.7 & 2.2 \\ 3.0 & 5.4 \\ 3.2 & 0.7 \end{bmatrix}$$

How do we compute P?

How do we compute P?

$$\mathrm{P}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/dimension_{model}}}\right)$$

$$\mathrm{P}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/dimension_{model}}}\right)$$

How do we compute P?

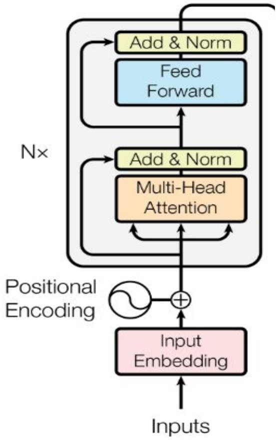
$$\mathrm{P}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/dimension_{model}}}\right)$$

$$\mathrm{P}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/dimension_{model}}}\right)$$

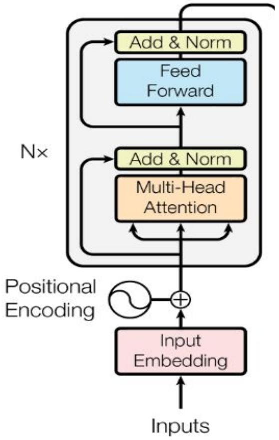
$$P = \begin{array}{c} \text{Spaghetti} \\ \text{monster} \\ \text{is} \\ \text{great} \end{array} \begin{bmatrix} \sin\left(\frac{0}{10000^{2\cdot 0/3}}\right) & \cos\left(\frac{0}{10000^{2\cdot 1/2}}\right) & \sin\left(\frac{0}{10000^{2\cdot 2/3}}\right) \\ \sin\left(\frac{1}{10000^{2\cdot 0/3}}\right) & \cos\left(\frac{1}{10000^{2\cdot 1/2}}\right) & \sin\left(\frac{1}{10000^{2\cdot 2/3}}\right) \\ \sin\left(\frac{2}{10000^{2\cdot 0/3}}\right) & \cos\left(\frac{2}{10000^{2\cdot 1/2}}\right) & \sin\left(\frac{2}{10000^{2\cdot 2/3}}\right) \\ \sin\left(\frac{3}{10000^{2\cdot 0/3}}\right) & \cos\left(\frac{3}{10000^{2\cdot 1/2}}\right) & \sin\left(\frac{3}{10000^{2\cdot 2/3}}\right) \end{bmatrix}$$

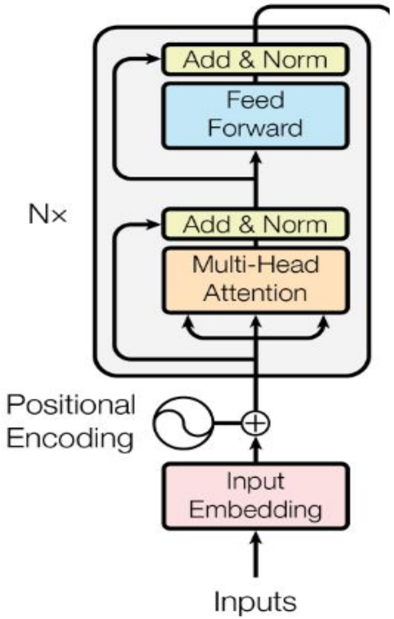
Other components missing from encoder?

- Feed-forward
- Add & Norm

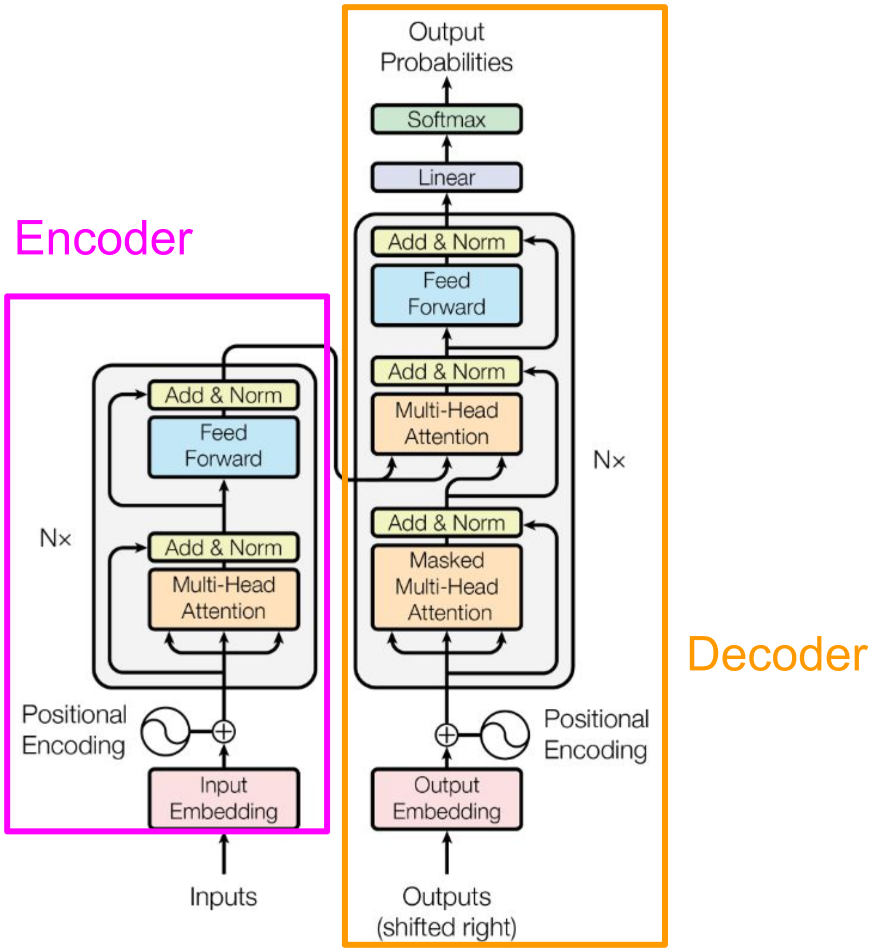


Other components missing from encoder?



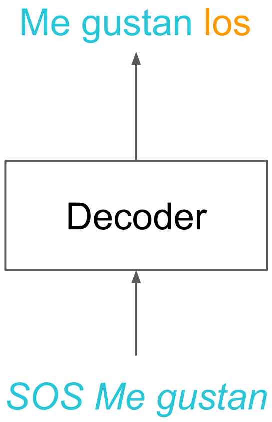


Add & Norm
Feed Forward
Nx
Add & Norm
Multi-Head Attention
Positional Encoding
Input Embedding
Inputs

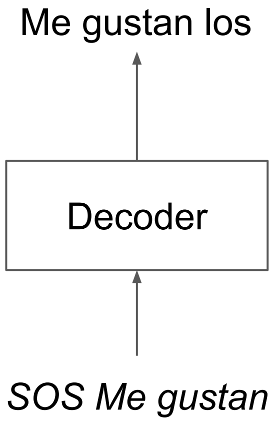


Encoder
Decoder
Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Input
Embedding
Inputs
Positional
Encoding
Output
Embedding
Outputs
(shifted right)

Training / inference discrepancy



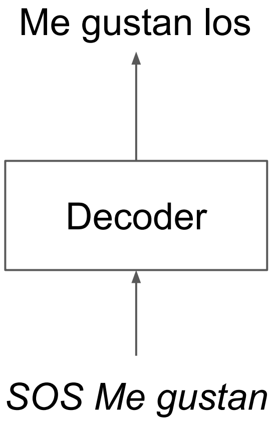
Training / inference discrepancy



What decoder knows during inference

SOS me gustan

Training / inference discrepancy



What decoder knows during inference

SOS me gustan

What decoder knows during training

SOS me gustan los gatos

Masked multi-head attention

$$Z_i(Q_i, K_i, V_i) = \mathrm{softmax}\left(\boxed{\frac{Q_i K_i^T}{\sqrt{d_k}}}\right) V_i$$

Masked multi-head attention

$$\frac{Q_i K_i^T}{\sqrt{d_k}} = \begin{array}{c c} & \begin{array}{ccccc} \text{SOS} & \text{me} & \text{gustan} & \text{los} & \text{gatos} \end{array} \\ \begin{array}{c} \text{SOS} \\ \text{me} \\ \text{gustan} \\ \text{los} \\ \text{gatos} \end{array} & \begin{bmatrix} 1.3 & 0.8 & 1.3 & 2.8 & 2.3 \\ 2.4 & 2.8 & 2.3 & 6.8 & 1.9 \\ 1.6 & 7.4 & 1.6 & 0.3 & 0.5 \\ 2.1 & 1.2 & 9.3 & 5.2 & 0.2 \\ 4.3 & 3.8 & 6.3 & 1.8 & 2.3 \end{bmatrix} \end{array}$$

Masked multi-head attention

$$\frac{Q_i K_i^T}{\sqrt{d_k}} = \begin{array}{c c} & \begin{array}{ccccc} \text{SOS} & \text{me} & \text{gustan} & \text{los} & \text{gatos} \end{array} \\ \begin{array}{c} \text{SOS} \\ \text{me} \\ \text{gustan} \\ \text{los} \\ \text{gatos} \end{array} & \begin{bmatrix} 1.3 & 0.8 & 1.3 & 2.8 & 2.3 \\ 2.4 & 2.8 & 2.3 & 6.8 & 1.9 \\ 1.6 & 7.4 & 1.6 & 0.3 & 0.5 \\ 2.1 & 1.2 & 9.3 & 5.2 & 0.2 \\ 4.3 & 3.8 & 6.3 & 1.8 & 2.3 \end{bmatrix} \end{array}$$

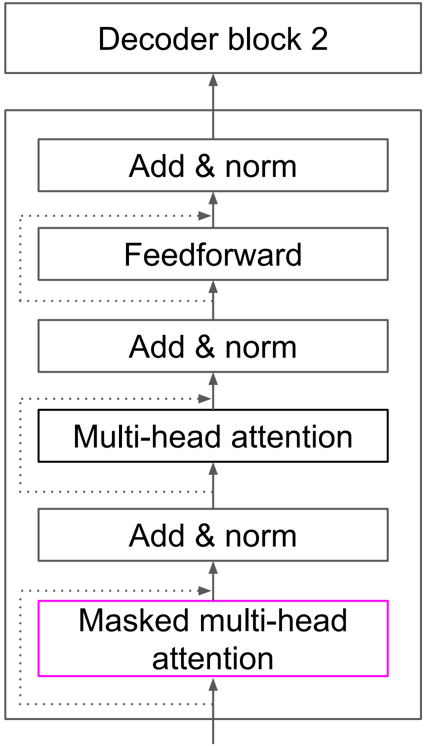
Masked multi-head attention

$$\frac{Q_i K_i^T}{\sqrt{d_k}} = \begin{array}{c c} & \begin{array}{ccccc} \text{SOS} & \text{me} & \text{gustan} & \text{los} & \text{gatos} \end{array} \\ \begin{array}{c} \text{SOS} \\ \text{me} \\ \text{gustan} \\ \text{los} \\ \text{gatos} \end{array} & \begin{bmatrix} 1.3 & \cancel{0.8} & \cancel{1.3} & \cancel{2.8} & \cancel{2.3} \\ 2.4 & 2.8 & 2.3 & 6.8 & 1.9 \\ 1.6 & 7.4 & 1.6 & 0.3 & 0.5 \\ 2.1 & 1.2 & 9.3 & 5.2 & 0.2 \\ 4.3 & 3.8 & 6.3 & 1.8 & 2.3 \end{bmatrix} \end{array}$$

Masked multi-head attention

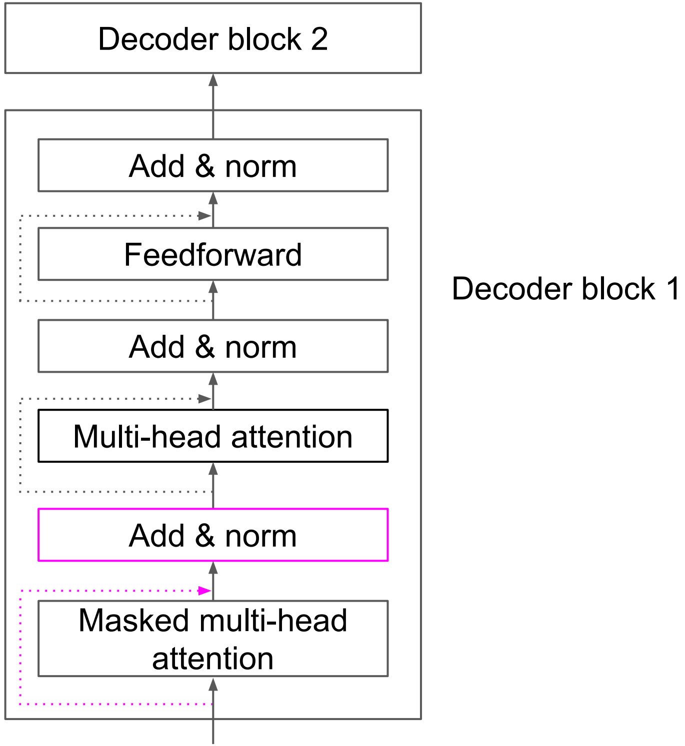
$$
\frac{Q_i K_i^T}{\sqrt{d_k}} =
\begin{array}{c c}
 & \begin{array}{ccccc} \text{SOS} & \text{me} & \text{gustan} & \text{los} & \text{gatos} \end{array} \\
\begin{array}{c} \text{SOS} \\ \text{me} \\ \text{gustan} \\ \text{los} \\ \text{gatos} \end{array} &
\begin{bmatrix}
1.3 & -\infty & -\infty & -\infty & -\infty \\
2.4 & 2.8 & -\infty & -\infty & -\infty \\
1.6 & 7.4 & 1.6 & -\infty & -\infty \\
2.1 & 1.2 & 9.3 & 5.2 & -\infty \\
4.3 & 3.8 & 6.3 & 1.8 & 2.3
\end{bmatrix}
\end{array}
$$

Decoder block

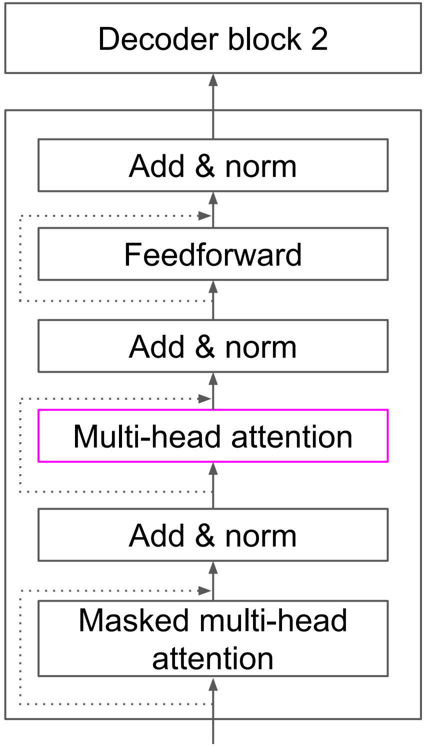


Decoder block 1

Decoder block

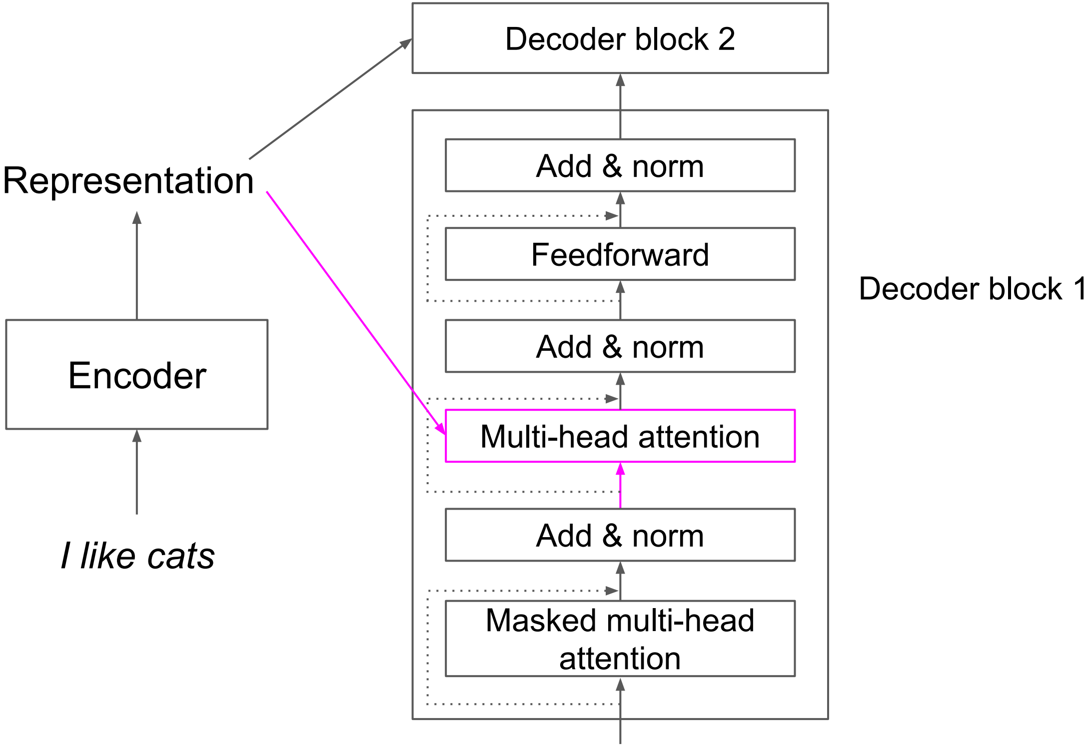


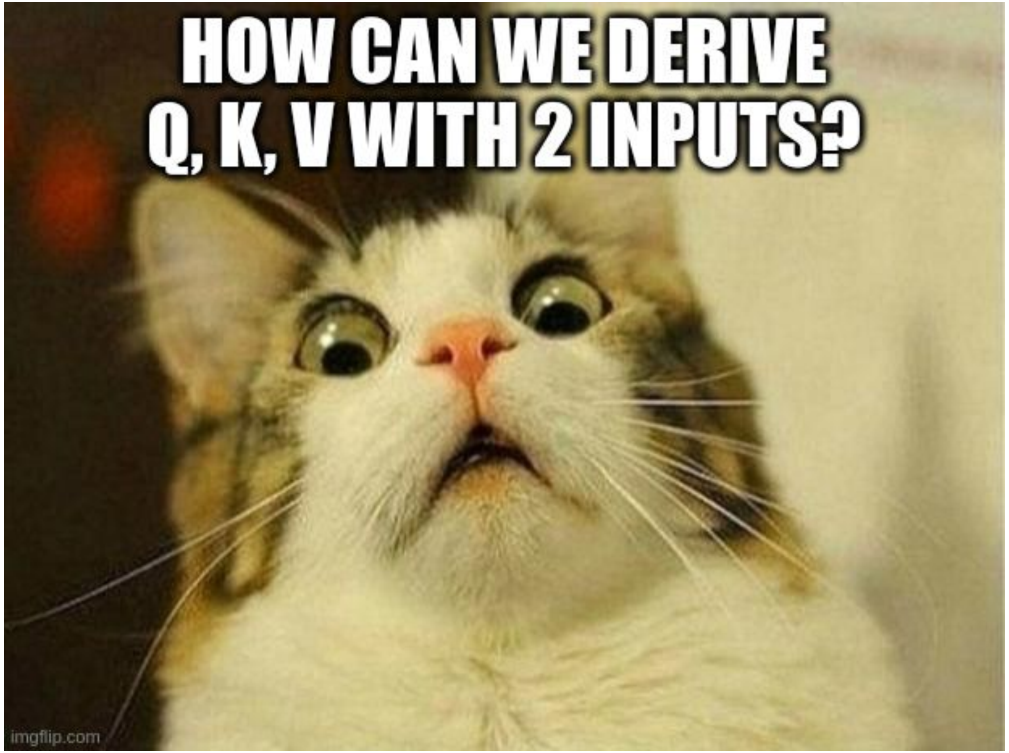
Decoder block



Decoder block 1

Decoder block




HOW CAN WE DERIVE
Q, K, V WITH 2 INPUTS?
imgflip.com

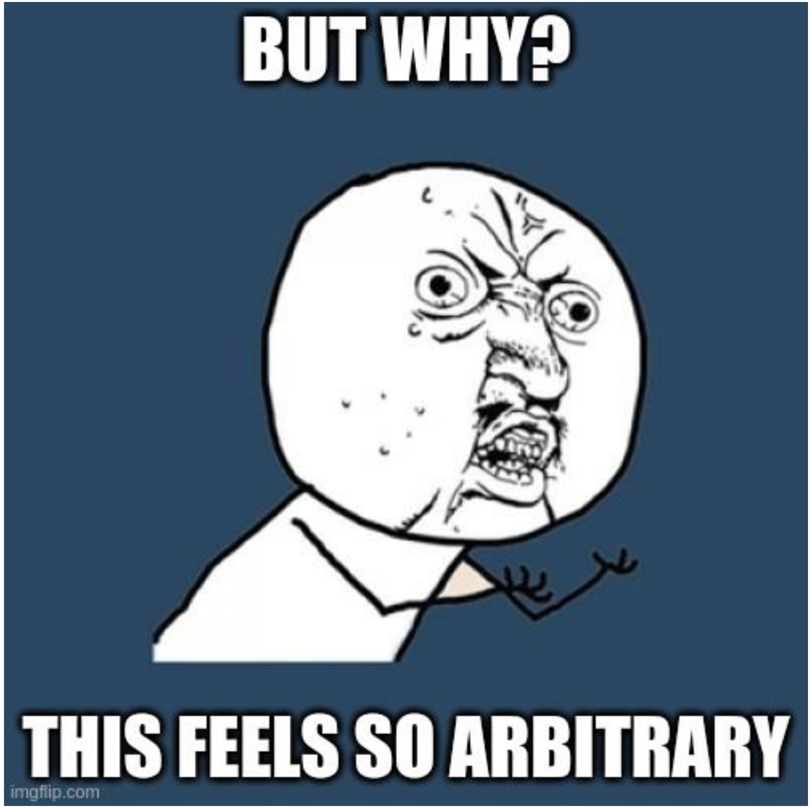
Deriving Q, K, V

- Query matrix (Q) from masked attention input
- Key (K) and value (V) matrices from encoder representation

$$MW_Q = Q$$
$$RW_K = K$$
$$RW_V = V$$

Deriving Q, K, V

- Q holds representation of target sentence
- K, V hold representation of source sentence

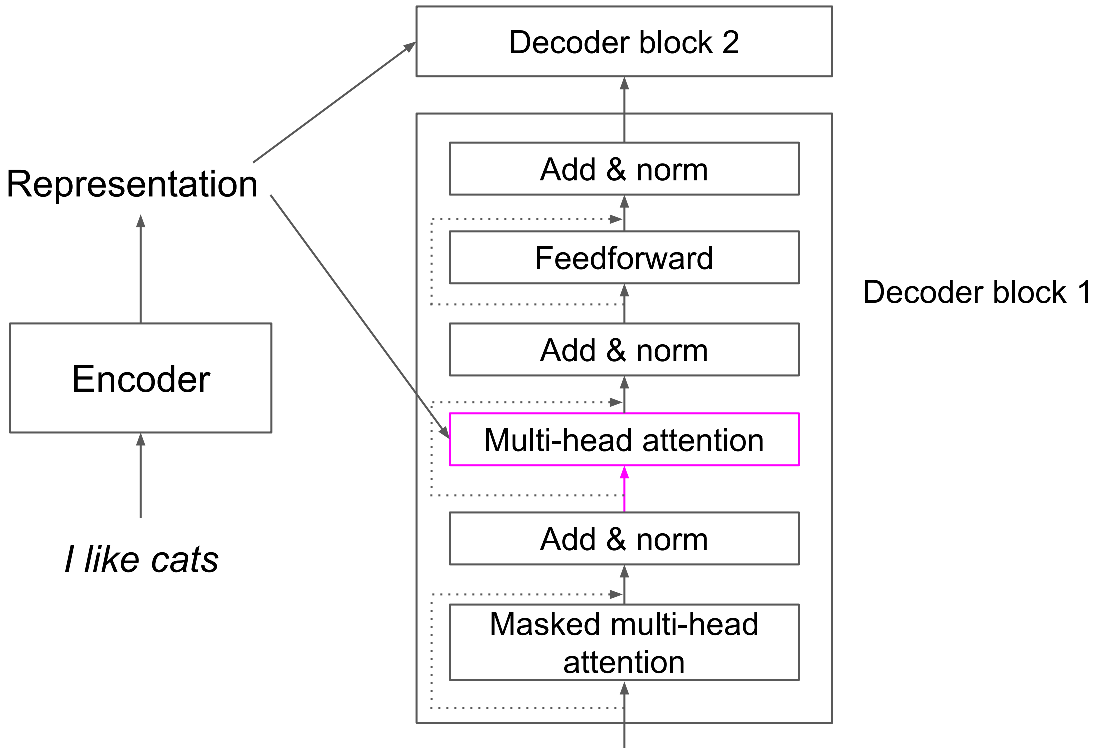

BUT WHY?
THIS FEELS SO ARBITRARY
imgflip.com

Deriving attention matrix

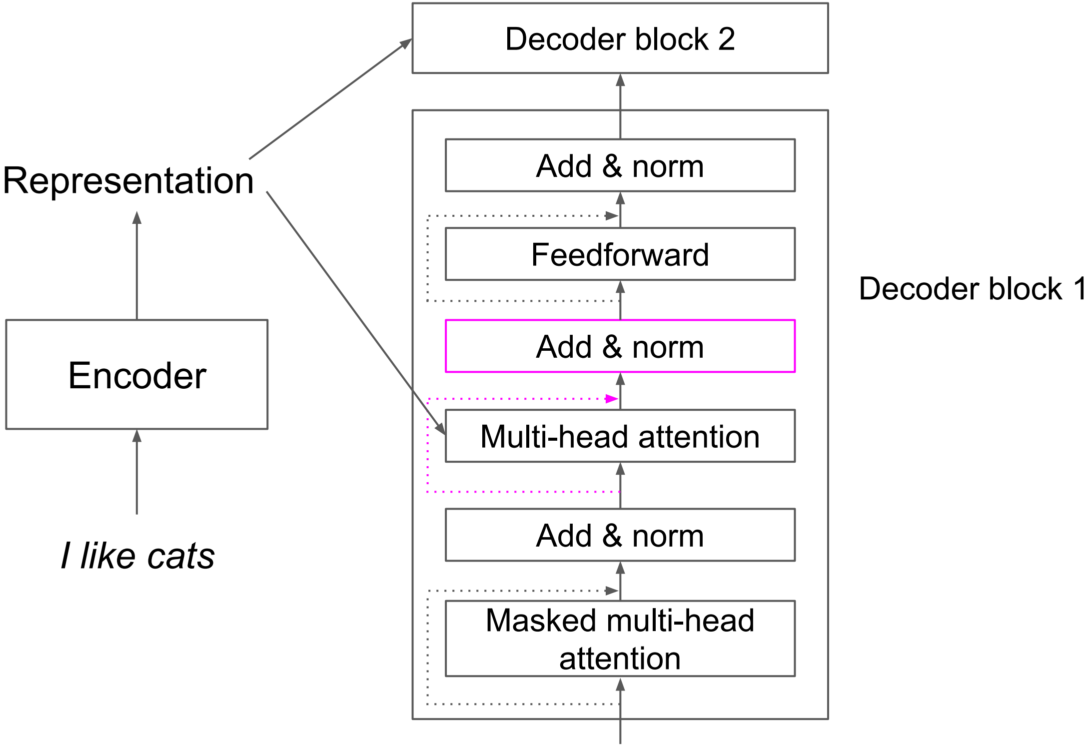
$$Z = \begin{array}{r} \\ \text{SOS} \\ \text{me} \\ \text{gustan} \\ \text{los} \\ \text{gatos} \end{array} \begin{array}{c} \begin{array}{ccc} \text{I} & \text{like} & \text{cats} \end{array} \\ \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.6 & 0.3 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.3 & 0.6 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \end{array} \quad \begin{array}{r} \text{I} \\ \text{like} \\ \text{cats} \end{array} \begin{bmatrix} 0.4 & 1.0 \\ 1.2 & 2.8 \\ 1.7 & 0.2 \end{bmatrix} \begin{array}{l} v_1 \\ v_2 \\ v_3 \end{array} = \begin{array}{r} \text{SOS} \\ \text{me} \\ \text{gustan} \\ \text{los} \\ \text{gatos} \end{array} \begin{bmatrix} \vec{z}_1 \\ \vec{z}_2 \\ \vec{z}_3 \\ \vec{z}_4 \\ \vec{z}_5 \end{bmatrix}$$

$$\underbrace{\vec{z}_3}_{\text{gustan}} = 0.1\underbrace{\vec{v}_1}_{\text{I}} + 0.8\underbrace{\vec{v}_2}_{\text{like}} + 0.1\underbrace{\vec{v}_3}_{\text{cats}}$$

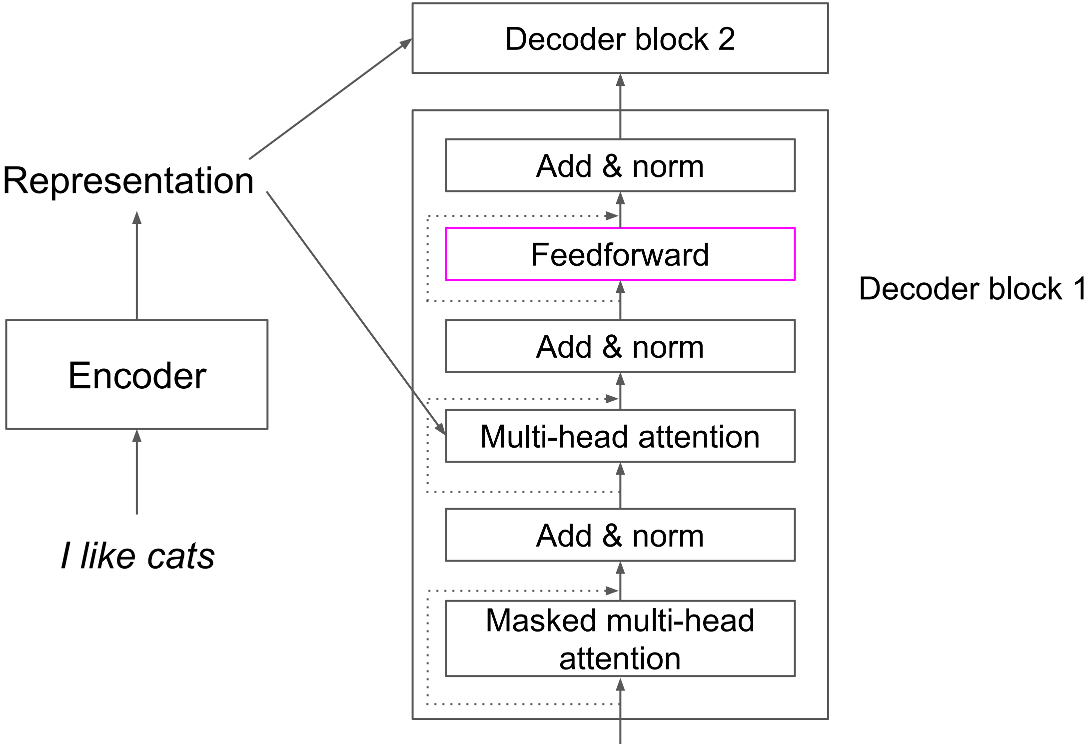
Decoder block



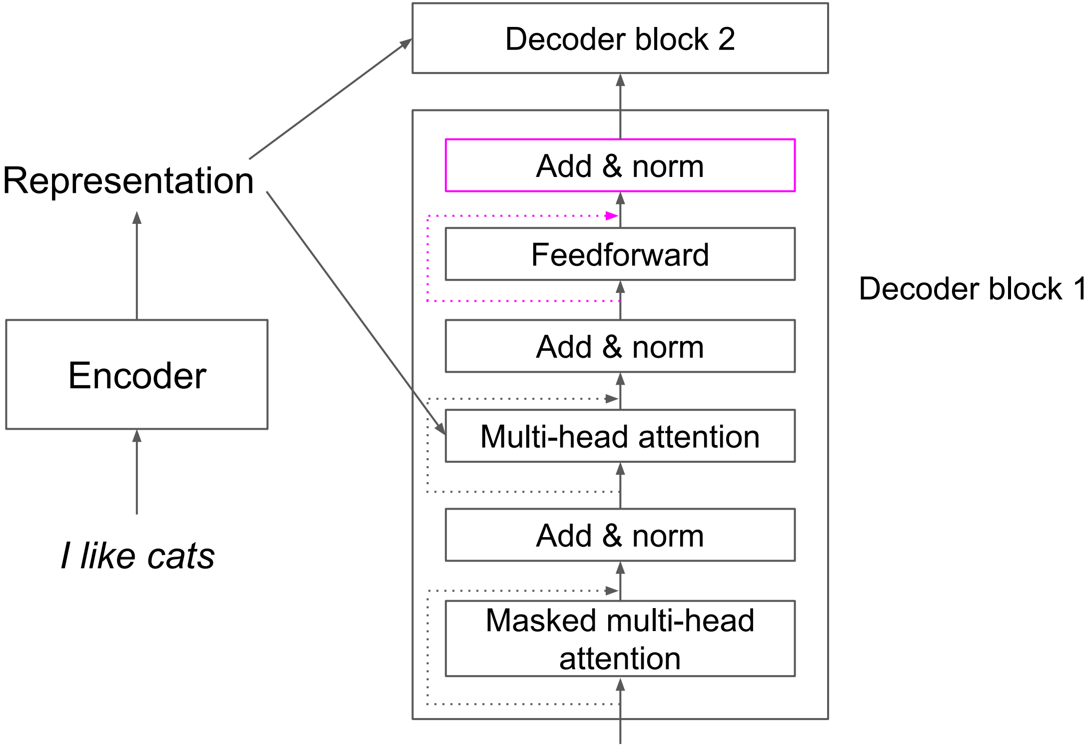
Decoder block



Decoder block

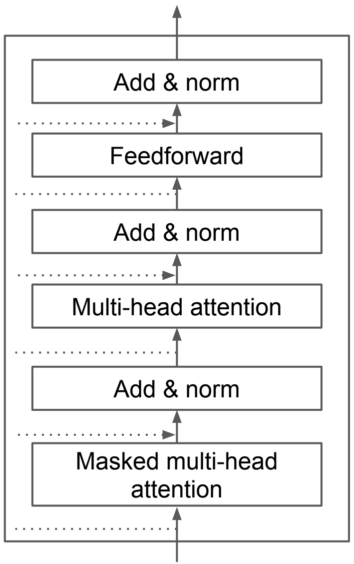


Decoder block

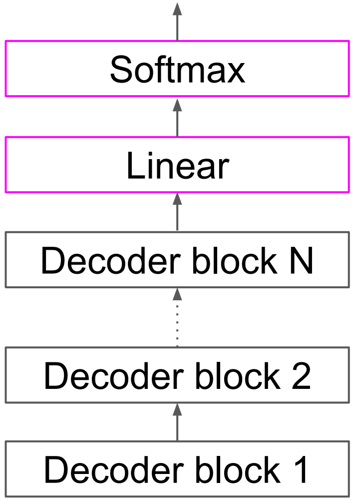


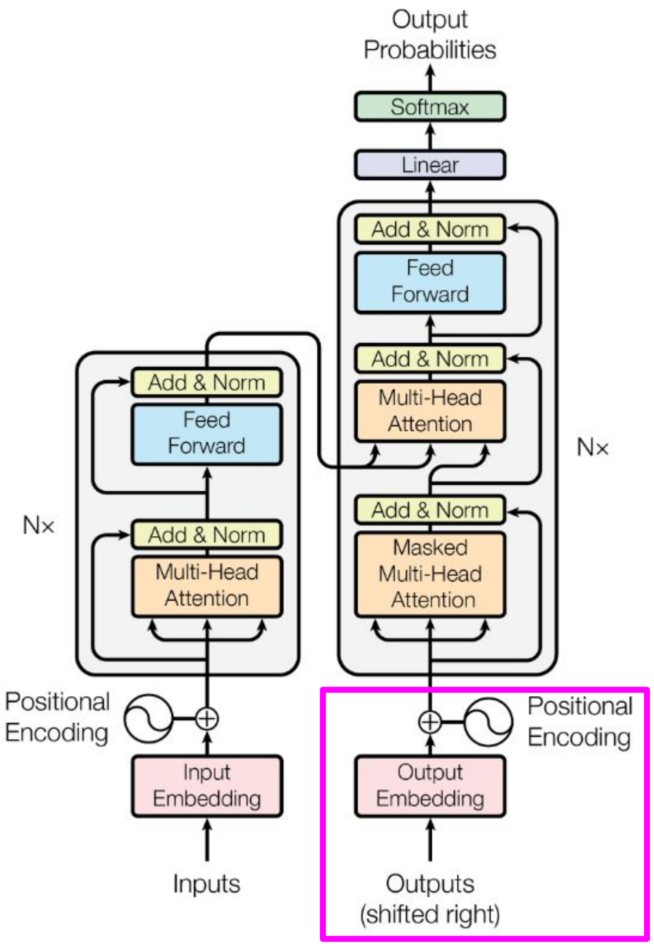
What’s the deeper meaning?

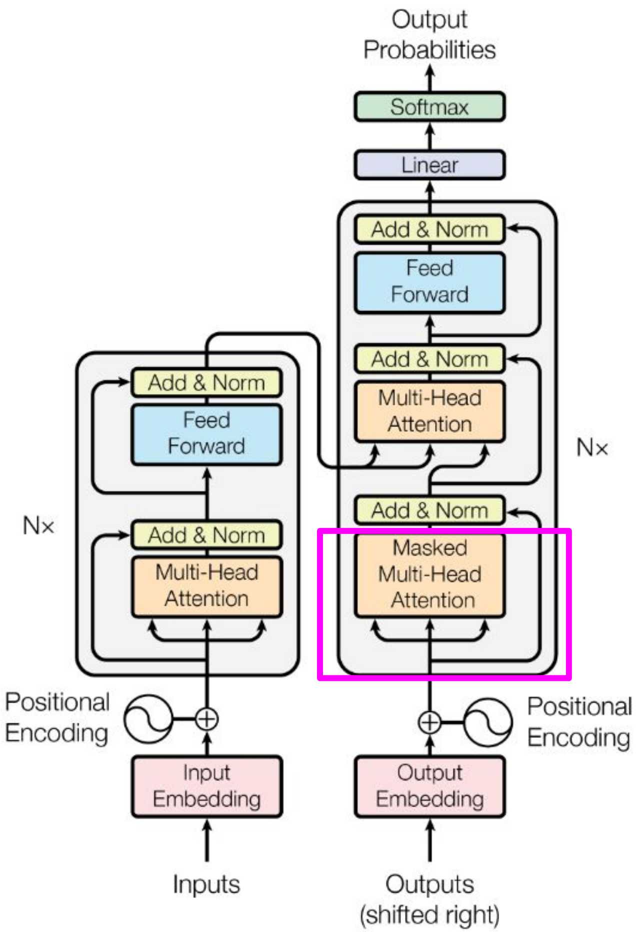
- Masked multi-head attention
- Multi-head attention
- Feedforward
- Add & Norm

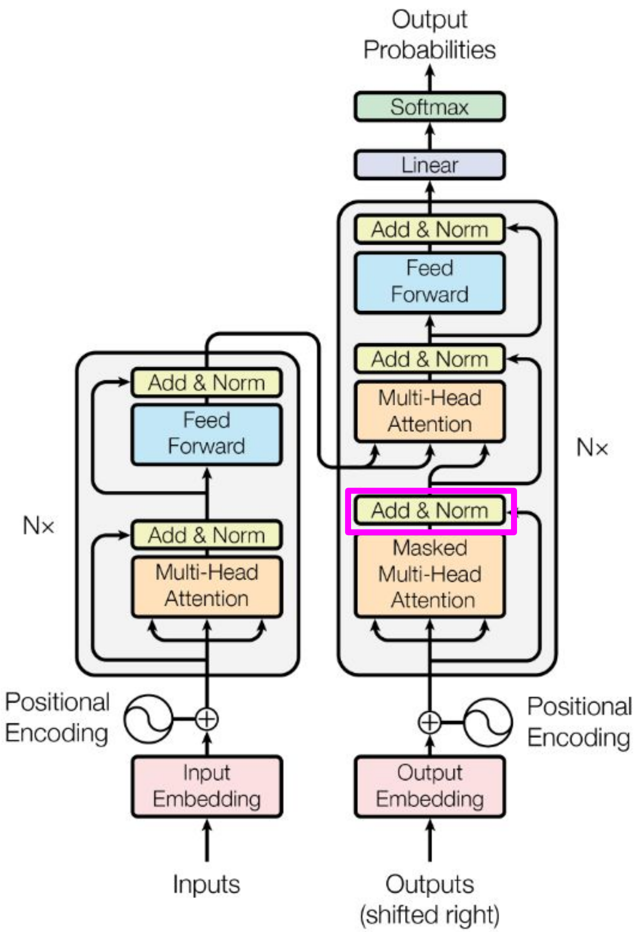


Linear & softmax layers

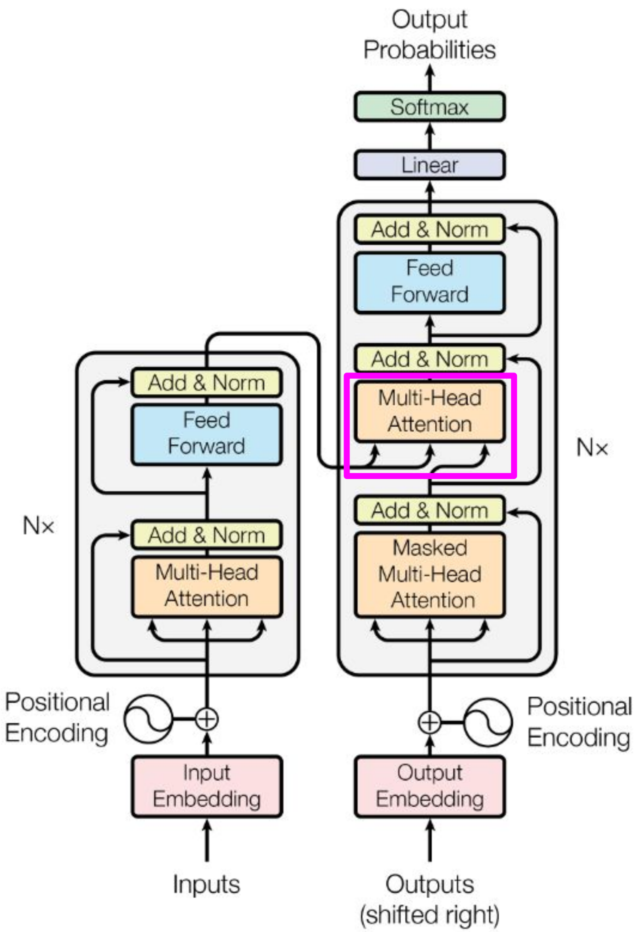


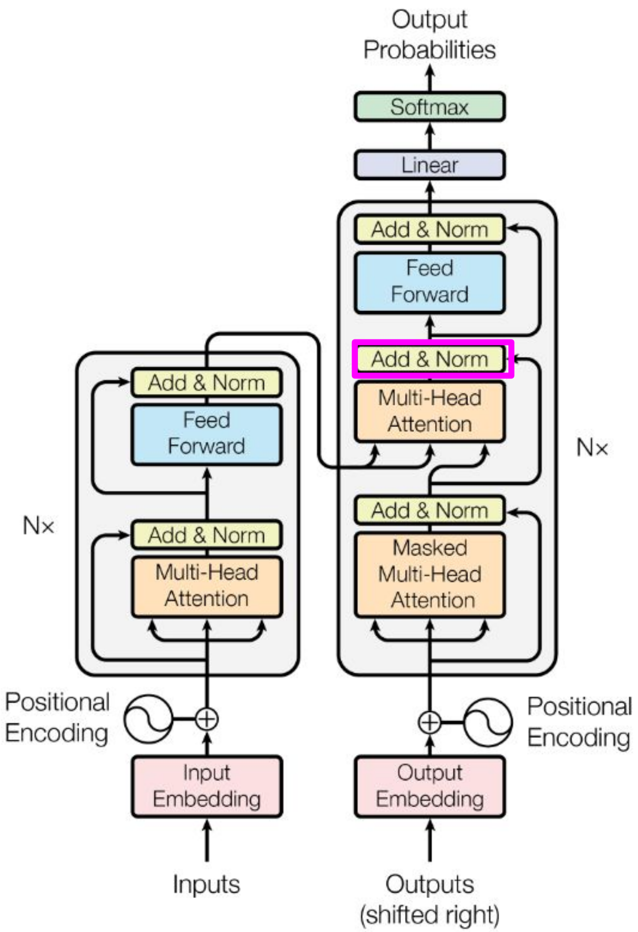

Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Input Embedding
Inputs
Positional Encoding
Output Embedding
Outputs (shifted right)

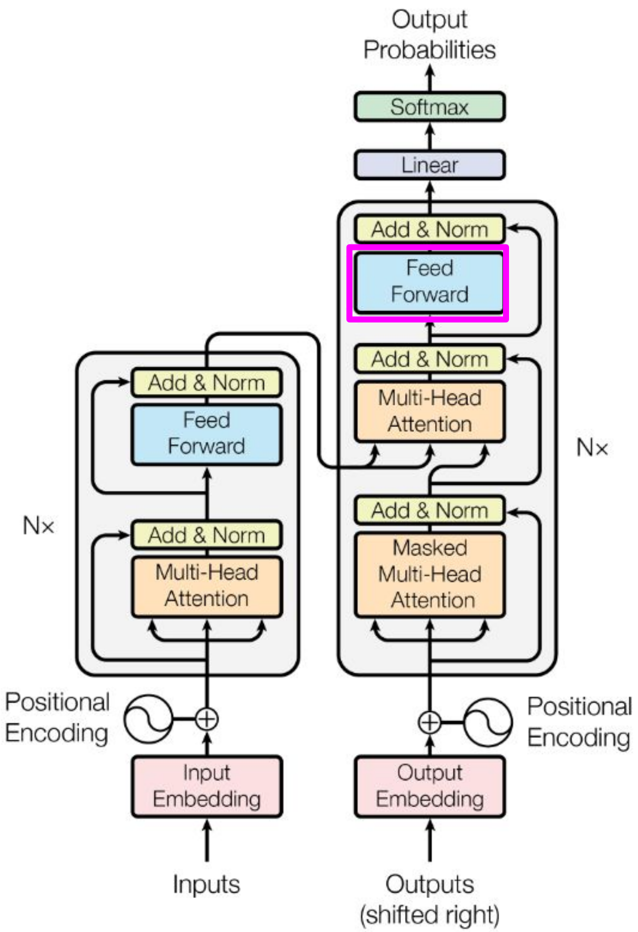

Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)



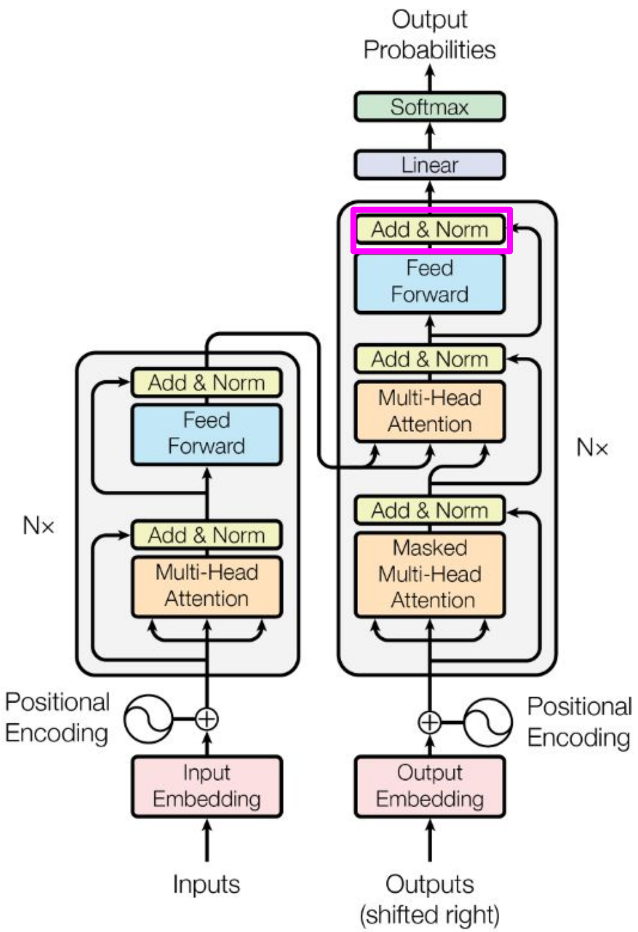
Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Add & Norm
Masked
Multi-Head
Attention
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)

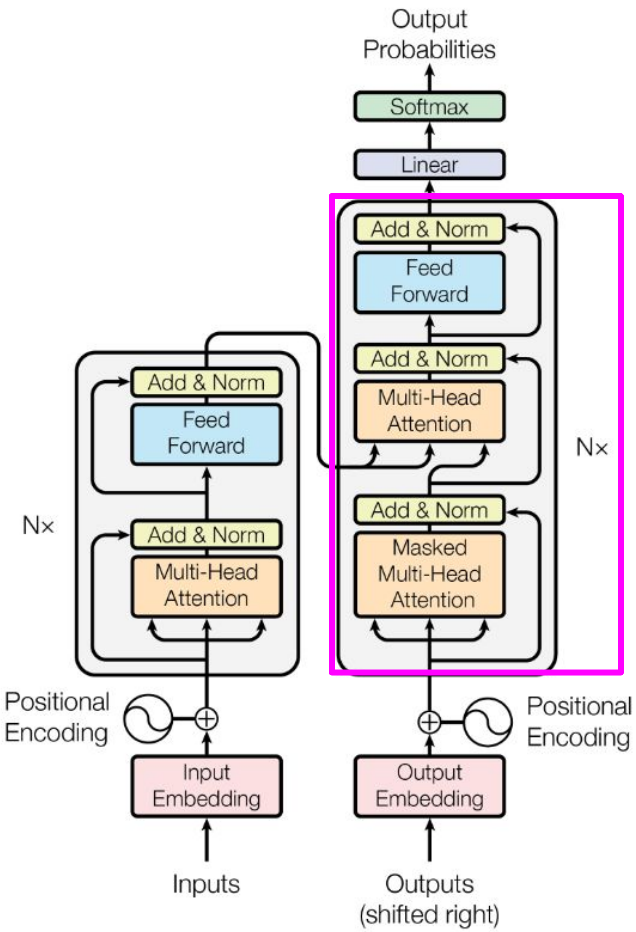

Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
N×
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
N×
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)


Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)

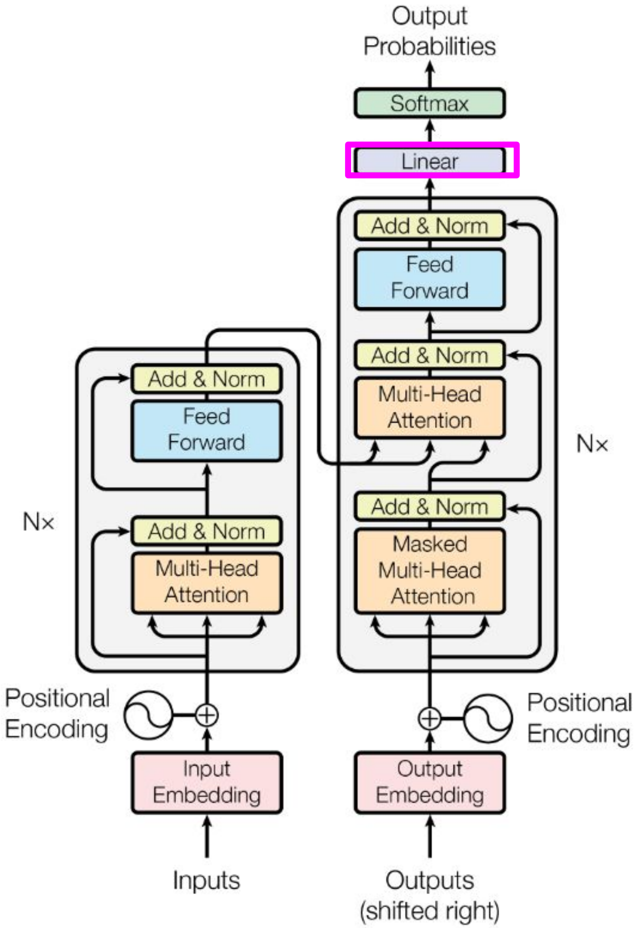


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

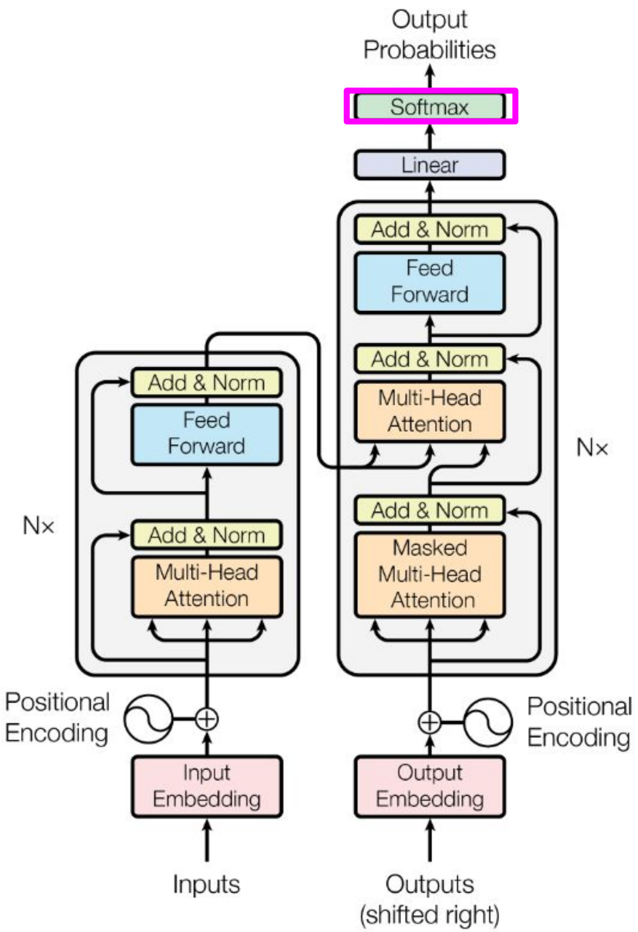

Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)

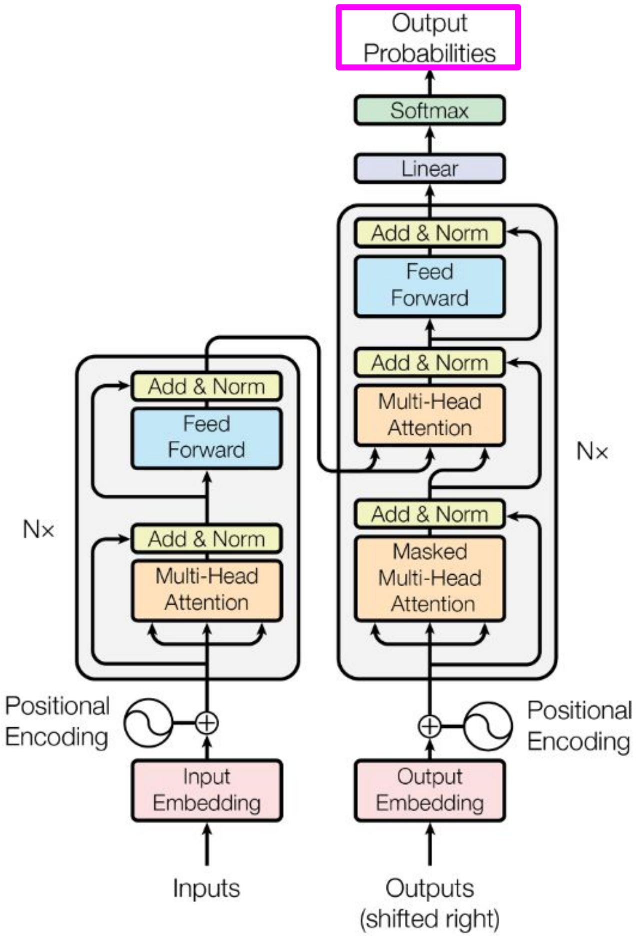


Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)



Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)


Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)


Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
N×
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
N×
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)

How can you use a transformer to generate chords?

How can you condition generation on emotion?

How would you evaluate the output of a transformer that generates melodies?

Do we care about overfitting? Do we care about perfect prediction?

Can transformers create truly original music? Can they get us to transformational creativity?


YOU'VE MADE IT
CONGRATS!
imgflip.com

My experience with Transformers

- Most capable model

My experience with Transformers

- Most capable model
- Massive amount of music data needed

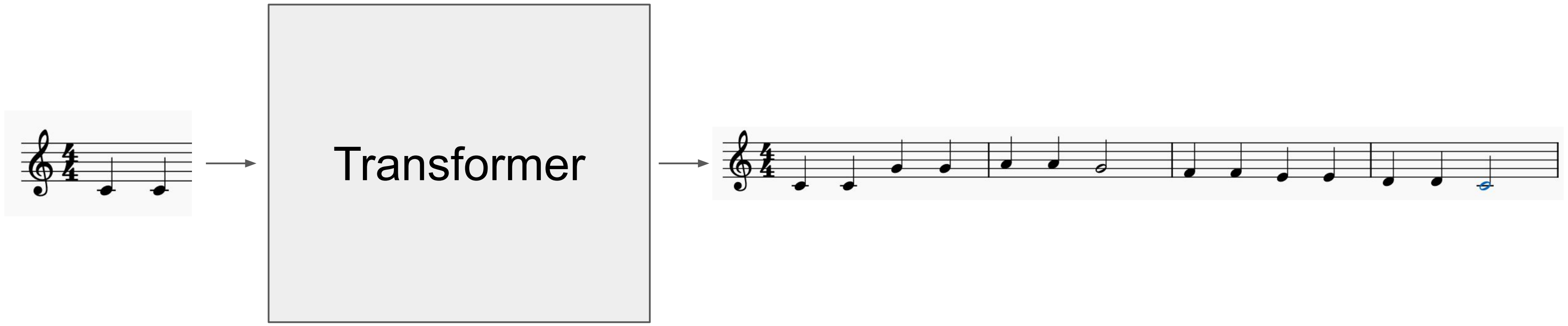
My experience with Transformers

- Most capable model
- Massive amount of music data needed
- Music theory bozo

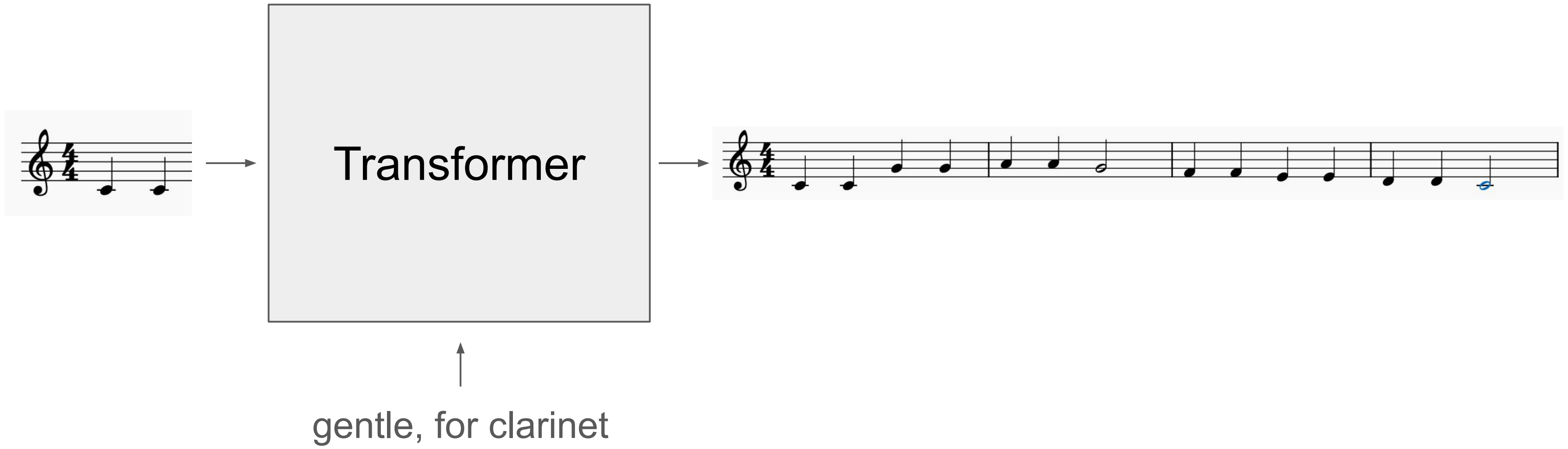
My experience with Transformers

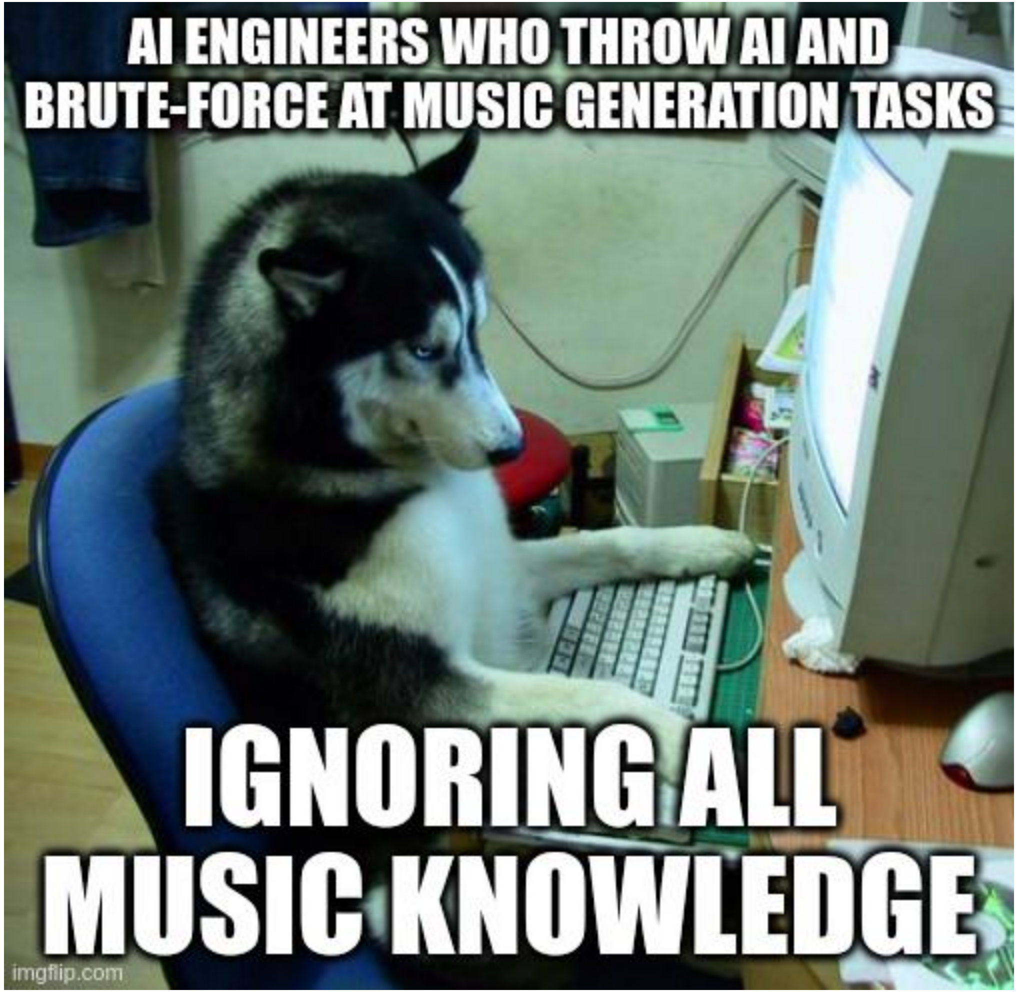
- Most capable model
- Massive amount of music data needed
- Music theory bozo
- Music representation is everything

Lack of creative control



Lack of creative control




AI ENGINEERS WHO THROW AI AND BRUTE-FORCE AT MUSIC GENERATION TASKS
IGNORING ALL MUSIC KNOWLEDGE
imgflip.com

My idea

- 2-level transformer
- Level 1: Generate high-level music representation
- Level 2: Fill the notes for level 1

My idea: Music representation

My idea: Level 1 music representation

- High-level description of symbolic music

My idea: Level 1 music representation

- High-level description of symbolic music
- Easier to learn
- More coherent generation
- Controllable

Tips for using Transformers

- Get as much (consistent) data as possible

Tips for using Transformers

- Get as much (consistent) data as possible
- Music-informed tokenization

Tips for using Transformers

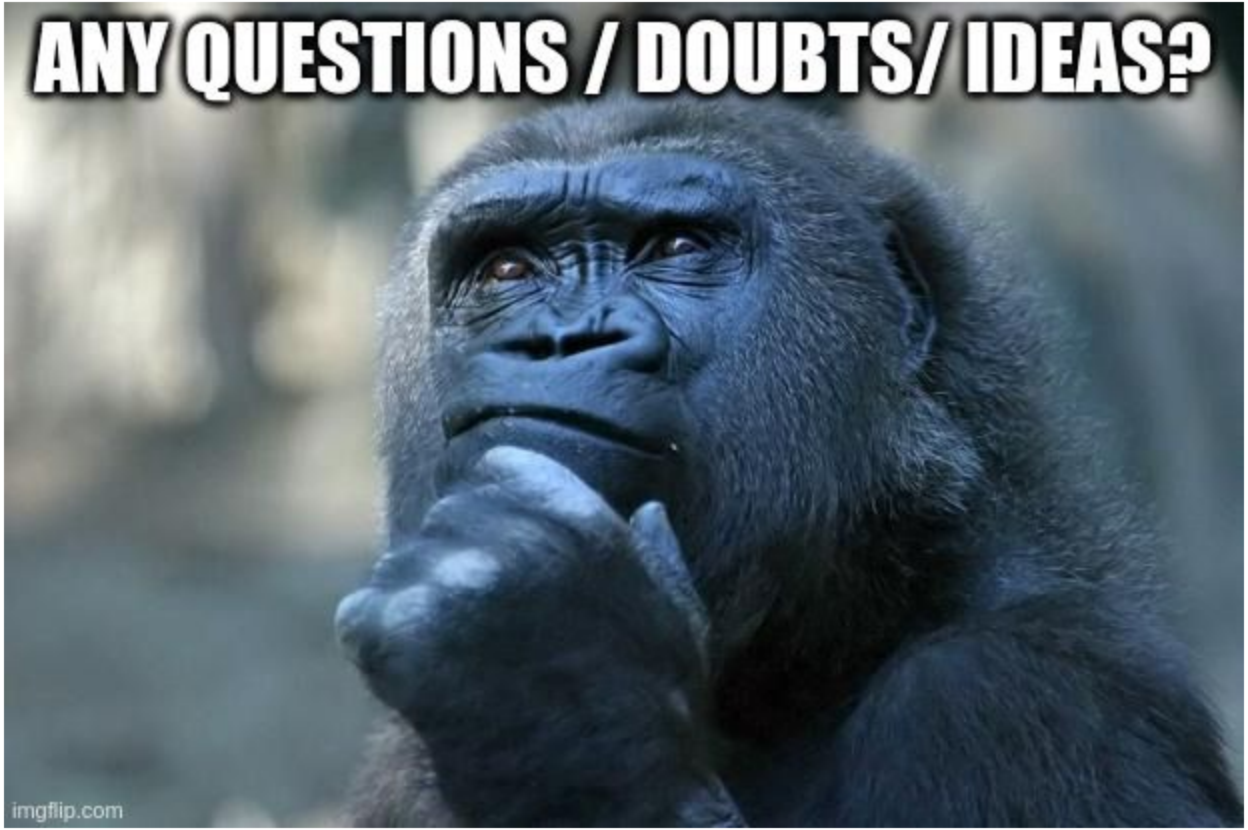
- Get as much (consistent) data as possible
- Music-informed tokenization
- Transpose data to C / Amin

Tips for using Transformers

- Get as much (consistent) data as possible
- Music-informed tokenization
- Transpose data to C / Amin
- Augment music data

Tips for using Transformers

- Get as much (consistent) data as possible
- Music-informed tokenization
- Transpose data to C / Amin
- Augment music data
- When using pre-trained models:
    - Fine-tune
    - Distillation


ANY QUESTIONS / DOUBTS/ IDEAS?
imgflip.com

Activity 1: Bridging symbolic and audio

Come up with a strategy / new architecture to repurpose the Transformer architecture for audio-based music gen. What challenges would you face?

Instructions:

- Work in groups (5 people)
- 7’ to come up with a solution
- 5’ to discuss together

[Museformer](https://ai-muzic.github.io/museformer/)

https://ai-muzic.github.io/museformer/

Activity 2: Transformers go long

Long-term coherence has long been a problem in gen mus. Skim through the *Museformer* paper and answer the questions:

- What's the model about?
- How does it try to address long-term coherence?

Instructions:

- Work in pairs
- 10' to read / study
- 5' to discuss together

Activity 3: LLMs generate music

Use ChatGPT (or any other LLM) to generate music. Try to sonify it.

- Is it any good? What surprised you (good / bad)?
- What are its shortcomings?
- How could you improve the generation?

Instructions:

- Work in groups (5 people)
- 10’ to play around
- 5’ to discuss together

Assignment 4: Transformer training

Train Transformer on Irish folk tunes dataset.

Deadline: 25 January at midnight