5. Transformers

Generative Algorithms for Sound and Music

Universitat Pompeu Fabra Barcelona

MTG Music Technology Group


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

Transformer Battle

Organize 4 teams

- Guinea pigs
- Dogs
- Cats
- Hamsters

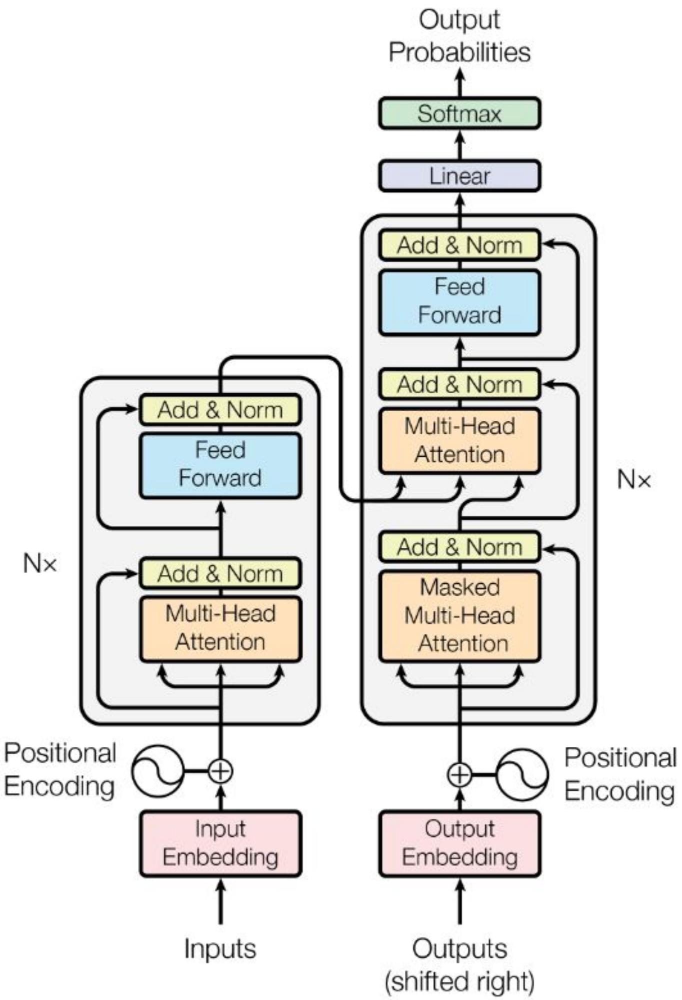
Rules of engagement

- Good explanation = +2
- Tough question = +4
- Wrong answer / explanation = -1
- If you can’t answer, another group will jump in

Tough question


RUTHLESS WINS THE DAY
imgflip.com

Real-time scores


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

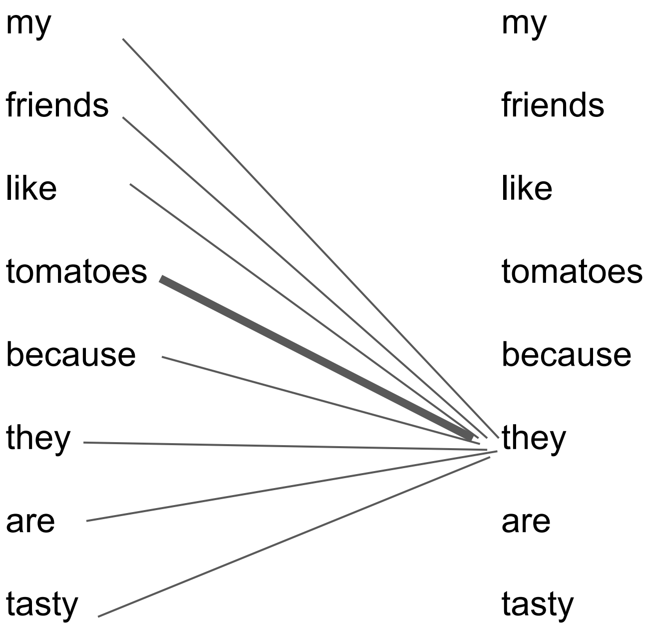
A reference problem

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

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

Why not use fixed K instead of learnable per head?

Self-attention: Formalisation

Self-attention: Formalisation

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

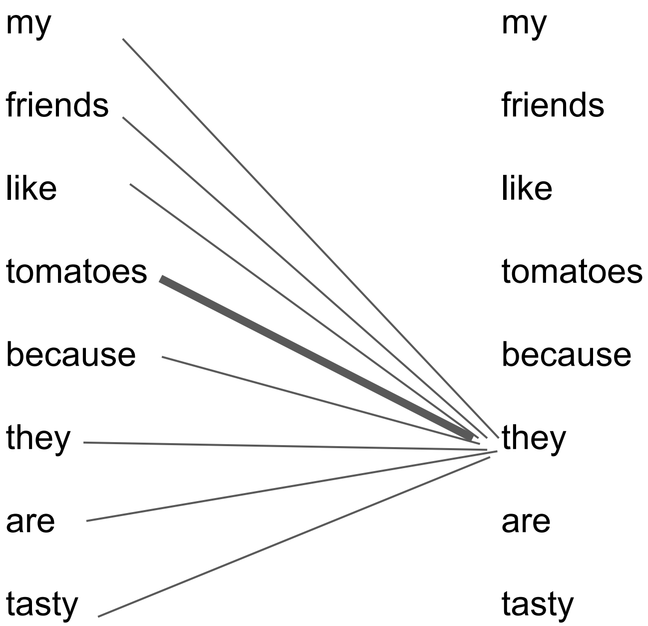
Self-attention: Step 1

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

Self-attention: Step 1

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

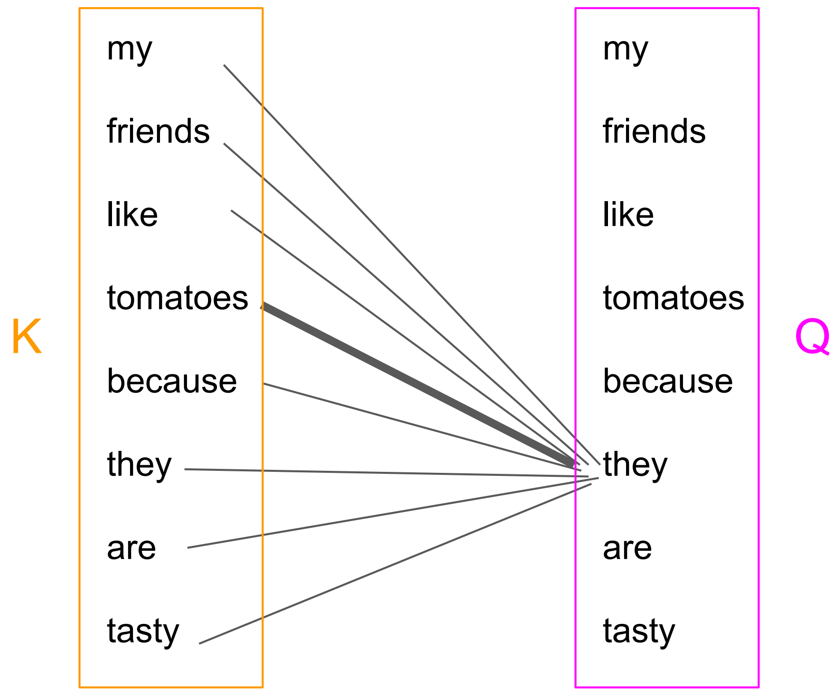
Which ones are Q and K really?



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?



Why Q and K are separate matrices?

What happens if Q = K?

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)

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

*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

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

Self-attention: Step 4

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

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

Self-attention for word “I”

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

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

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

Sum of the value vectors weighted by the scores

A reference problem: Solved

My friends like tomatoes because they are tasty

$$\vec{z}_{they} = 0.0\vec{v}_1 + 0.0\vec{v}_2 + 0.0\vec{v}_3 + 0.9\vec{v}_4 + 0.0\vec{v}_5 + 0.1\vec{v}_6 + 0.0\vec{v}_7 + 0.0\vec{v}_8$$

my friends like tomatoes because they are tasty

Why not use K for V?

Analogy: Search engine

Q = ?

K = ?

V = ?

Attention scores = ?

Result = ? x ?

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

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

What does concatenation **really** mean mathematically?

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


WHY MULTIPLE HEADS?
imgflip.com

Why do we multiply concatenated Z by Wo?

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

Positional encoding: Strategy

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

$$\qquad\qquad I \qquad\qquad\qquad P$$

Why transformer doesn't know about order?

How do we compute P?

How do we compute P?

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

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

Why sinusoidal?

How’s the position encoded, for real?

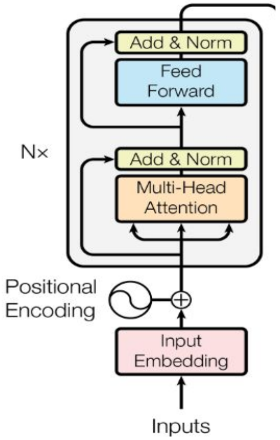
How do we compute P?

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

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

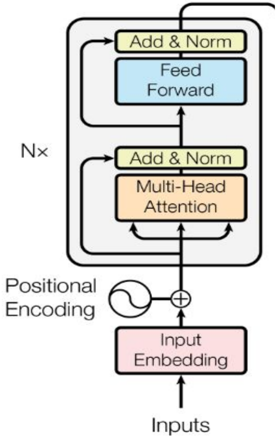
$$P = \begin{array}{r} \text{Spaghetti} \\ \text{monster} \\ \text{is} \\ \text{great} \end{array} \begin{bmatrix} \sin\left(\frac{0}{10000^{2\cdot 0/3}}\right) & \cos\left(\frac{0}{10000^{2\cdot 1/2}}\right) & \sin\left(\frac{0}{10000^{2\cdot 2/3}}\right) \\ \sin\left(\frac{1}{10000^{2\cdot 0/3}}\right)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

Other components missing from encoder?



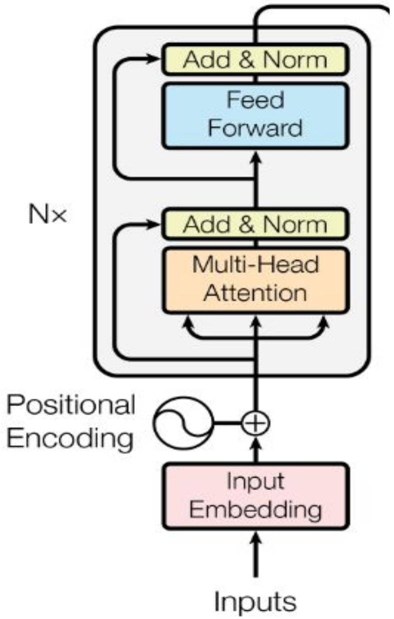
Other components missing from encoder?

- Feed-forward
- Add & Norm

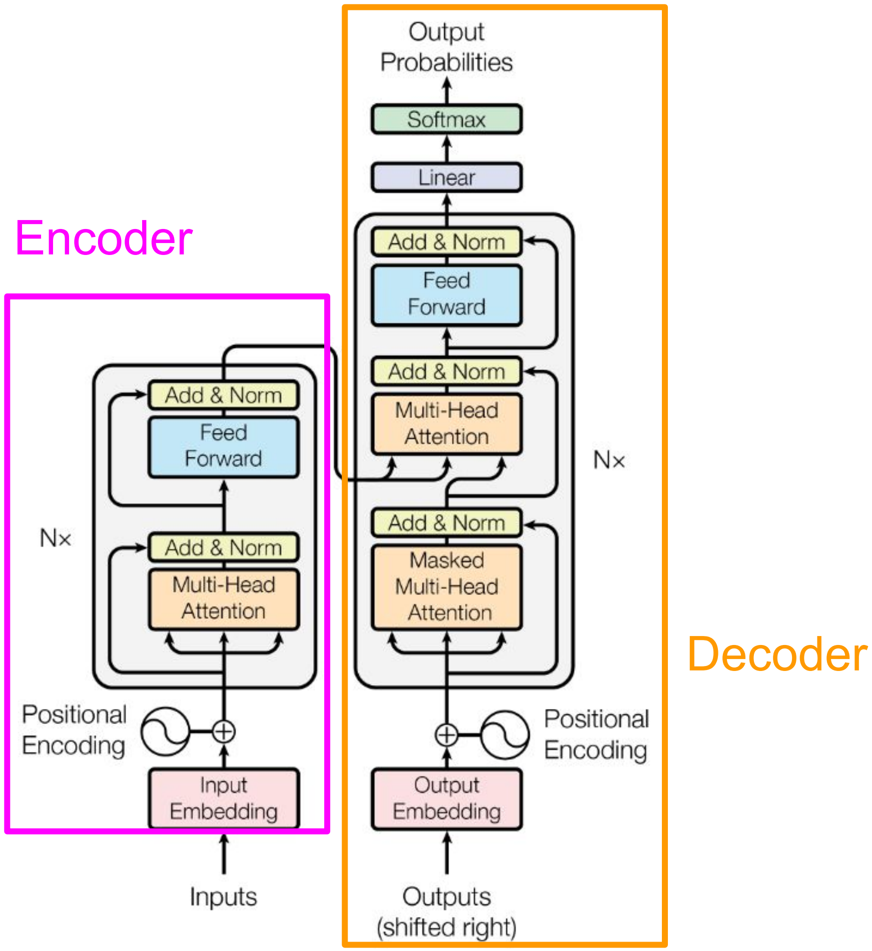


Why feedforward?

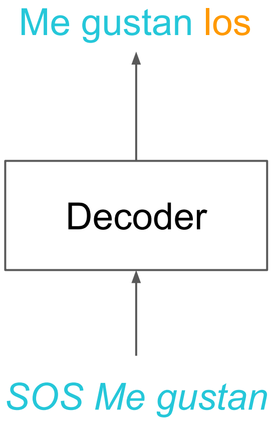
How do Add and Norm work for real?



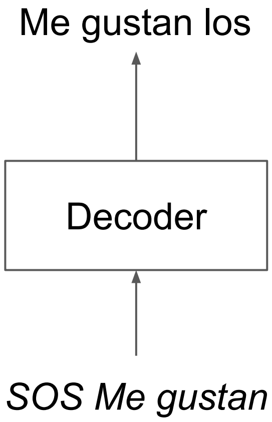
Add & Norm
Feed Forward
Nx
Add & Norm
Multi-Head Attention
Positional Encoding
Input Embedding
Inputs


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

Training / inference discrepancy



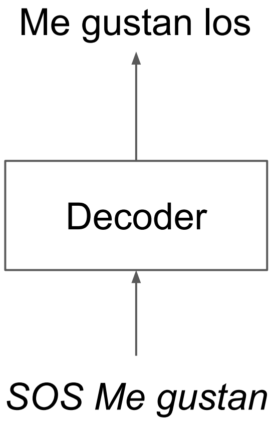
Training / inference discrepancy



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

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

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

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

Masked multi-head attention

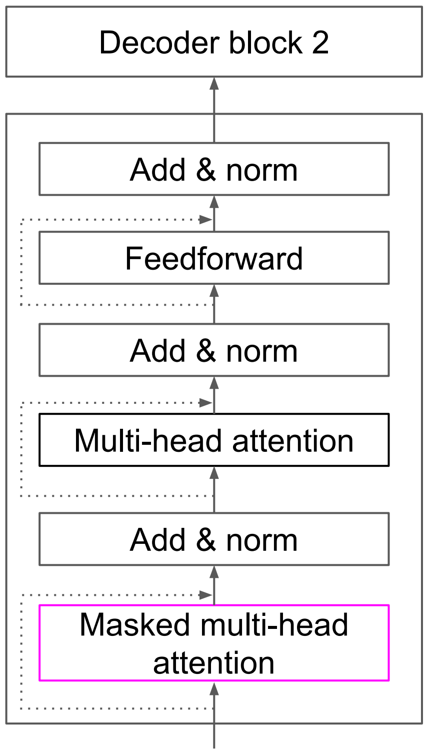
$$\frac{Q_i K_i^T}{\sqrt{d_k}} = \begin{array}{c c} & \begin{array}{ccccc} \text{SOS} & \text{me} & \text{gustan} & \text{los} & \text{gatos} \end{array} \\ \begin{array}{c} \text{SOS} \\ \text{me} \\ \text{gustan} \\ \text{los} \\ \text{gatos} \end{array} & \begin{bmatrix} 1.3 & \cancel{0.8} & \cancel{1.3}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

Masked multi-head attention

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

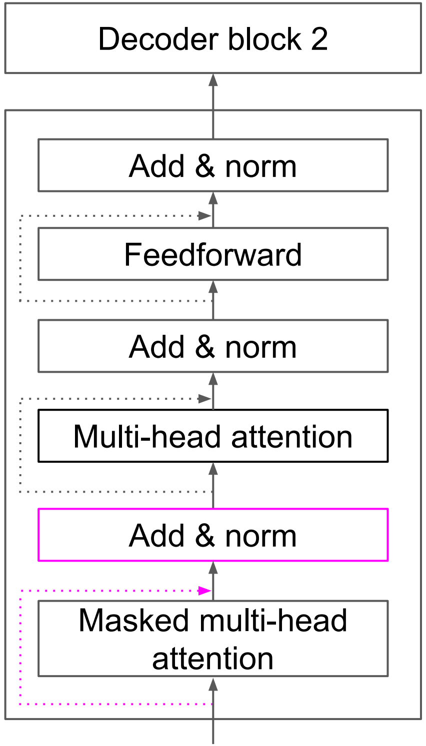
What does *softmax* spit when it encounters the -infinity mask?

Decoder block



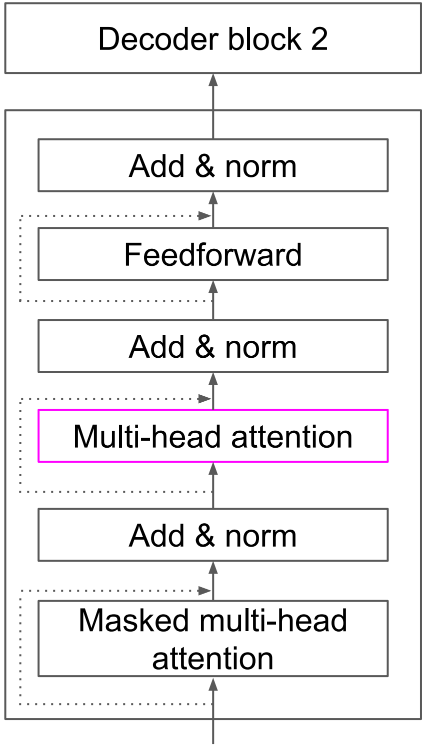
Decoder block 1

Decoder block



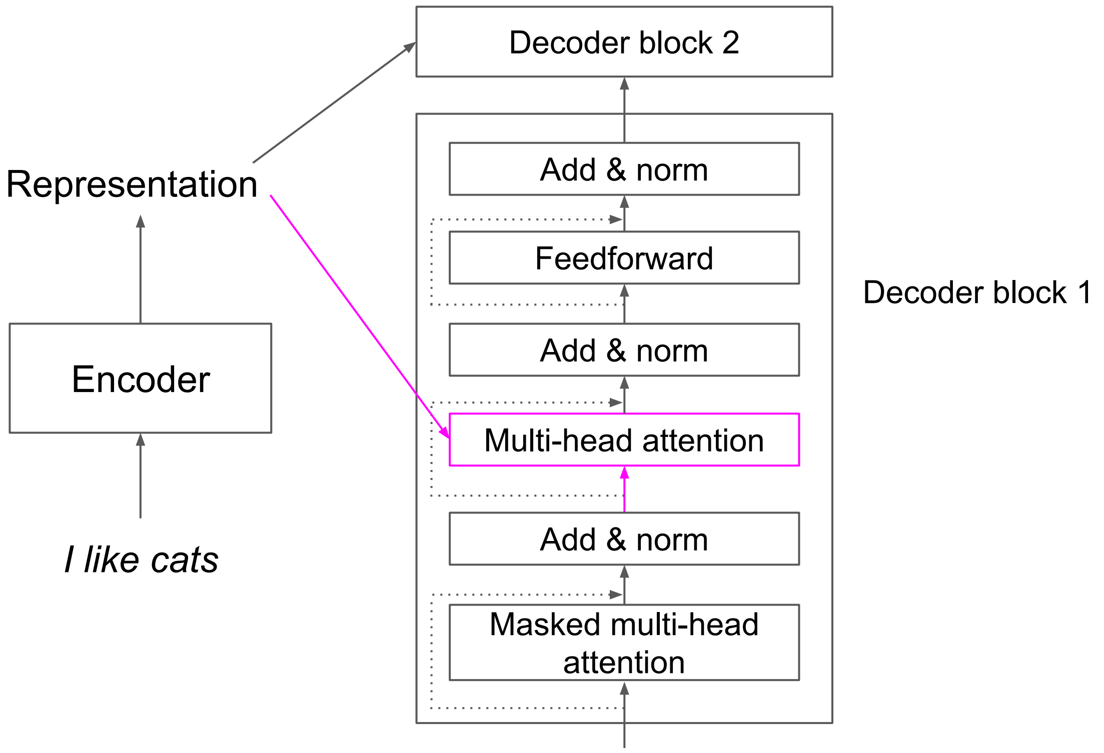
Decoder block 1

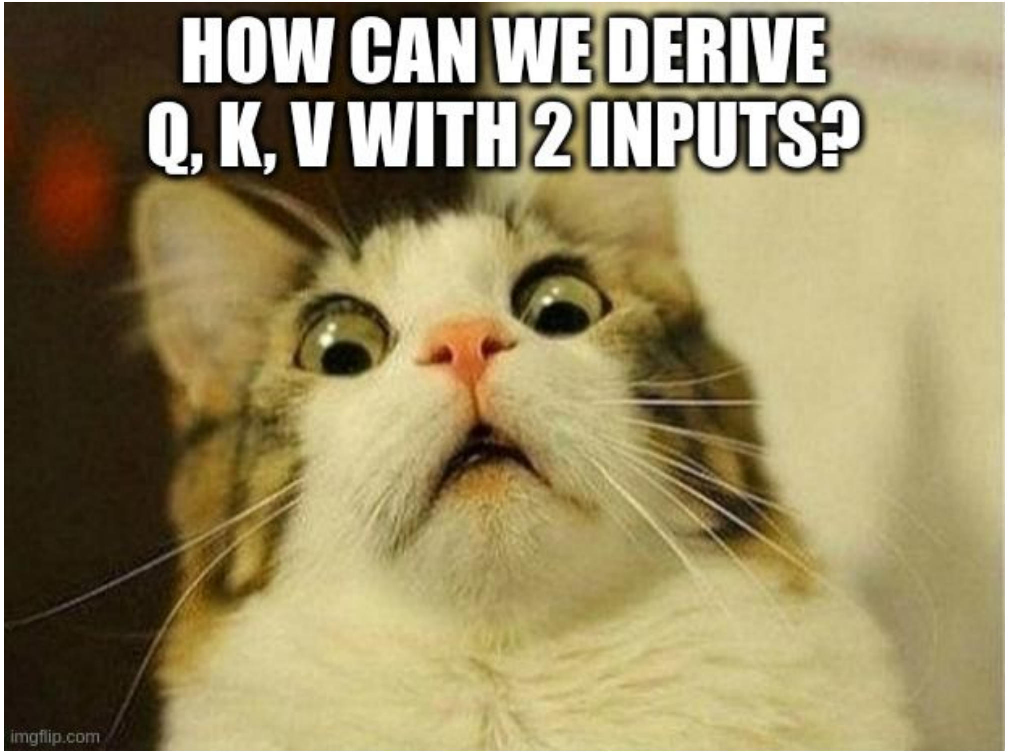
Decoder block



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

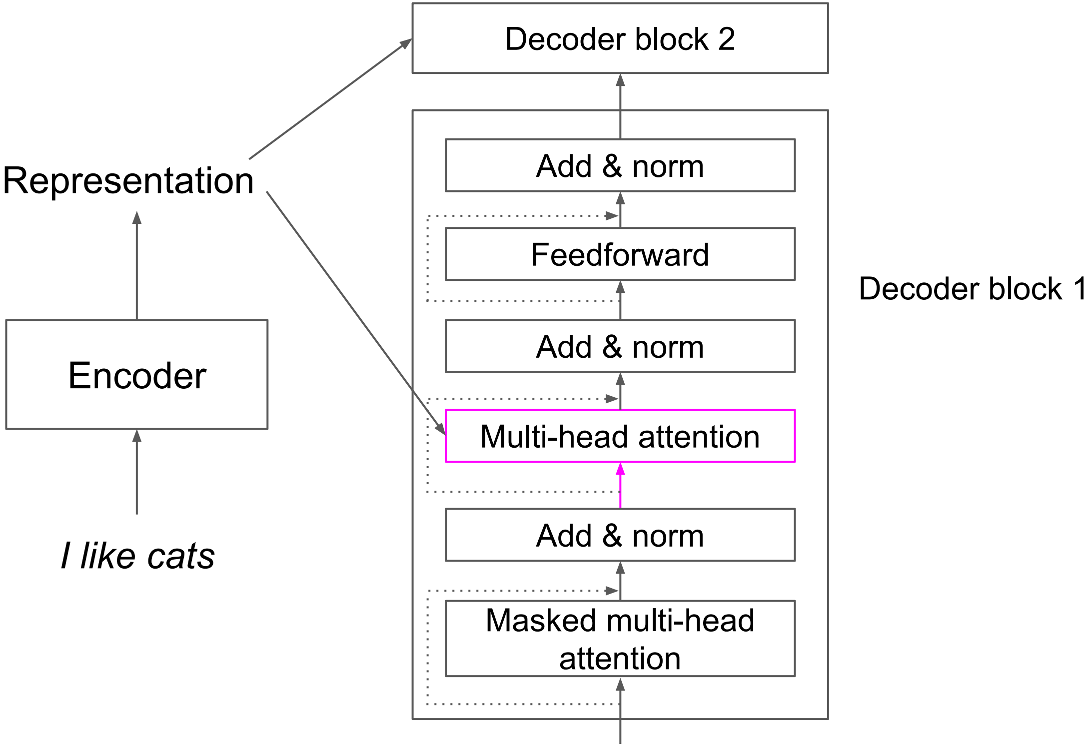
Why do we derive Q from target, and K, V from source?

Deriving attention matrix

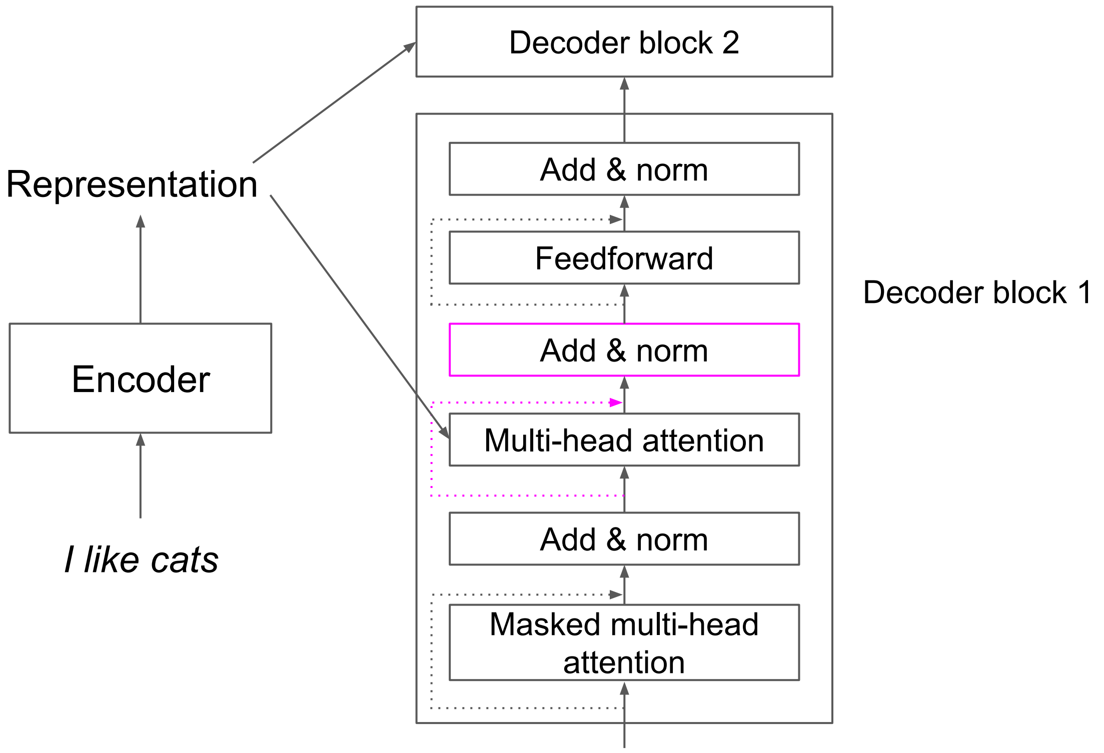
$$Z = \begin{array}{c} \\ \text{SOS} \\ \text{me} \\ \text{gustan} \\ \text{los} \\ \text{gatos} \end{array} \begin{array}{c} \begin{array}{ccc} \text{I} & \text{like} & \text{cats} \end{array} \\ \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.6 & 0.3 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.3 & 0.6 \\ 0.1 & 0.1 &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

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

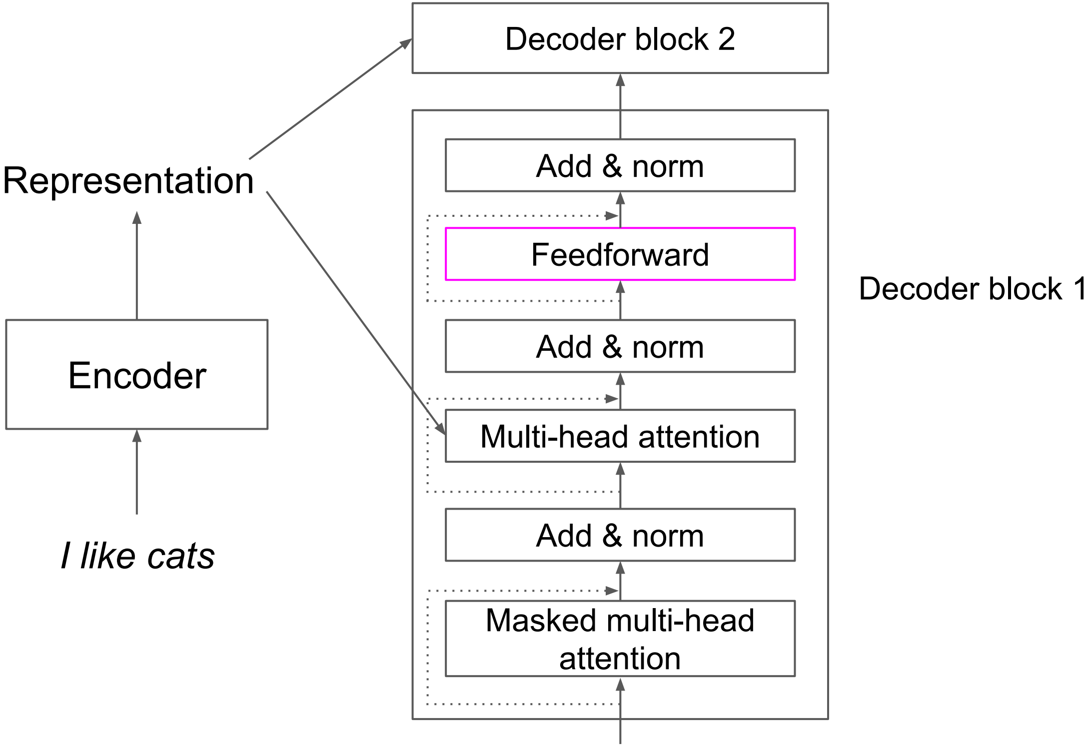
Decoder block



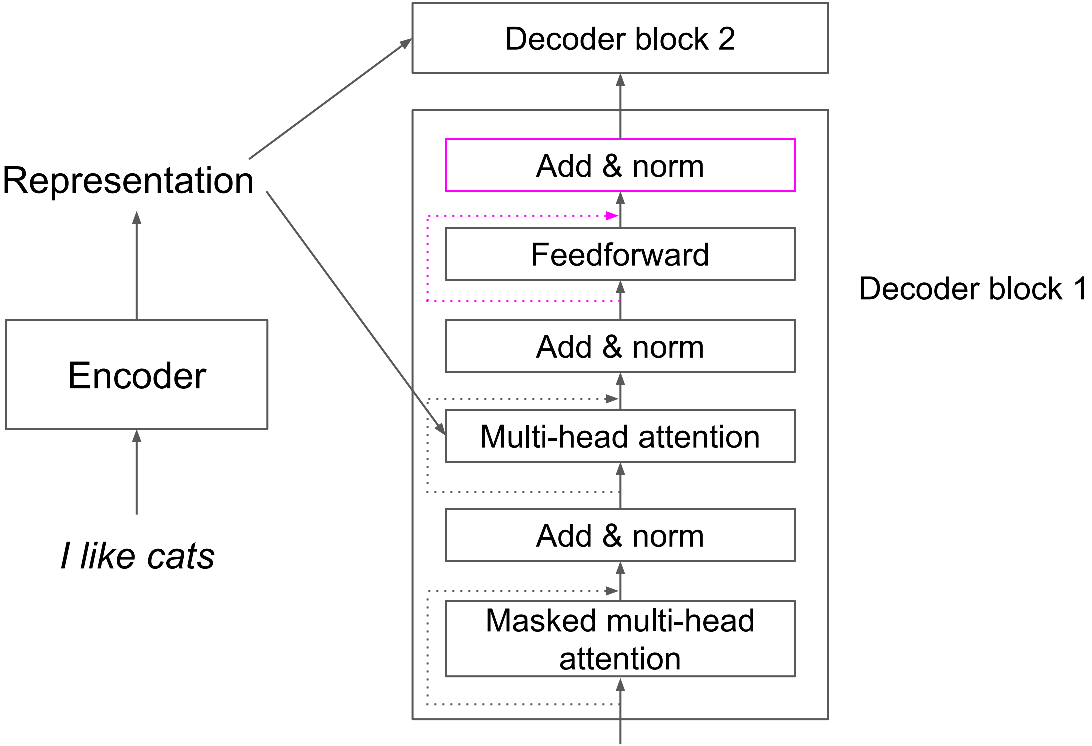
Decoder block



Decoder block

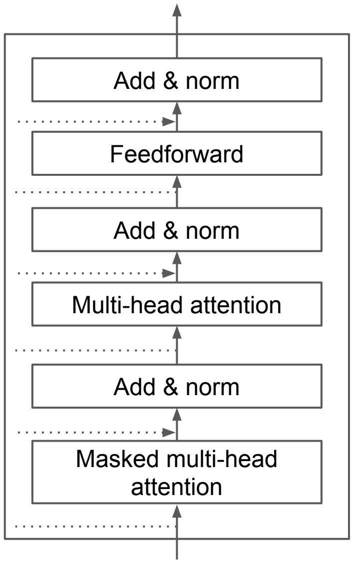


Decoder block

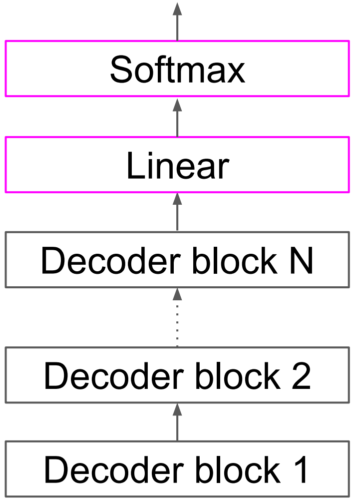


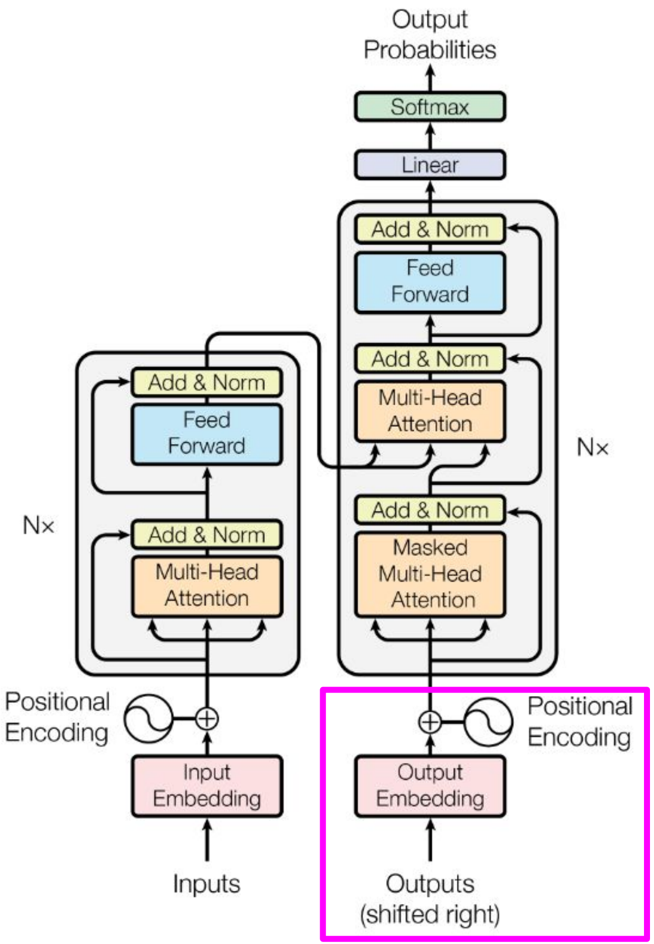
What’s the deeper meaning?

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

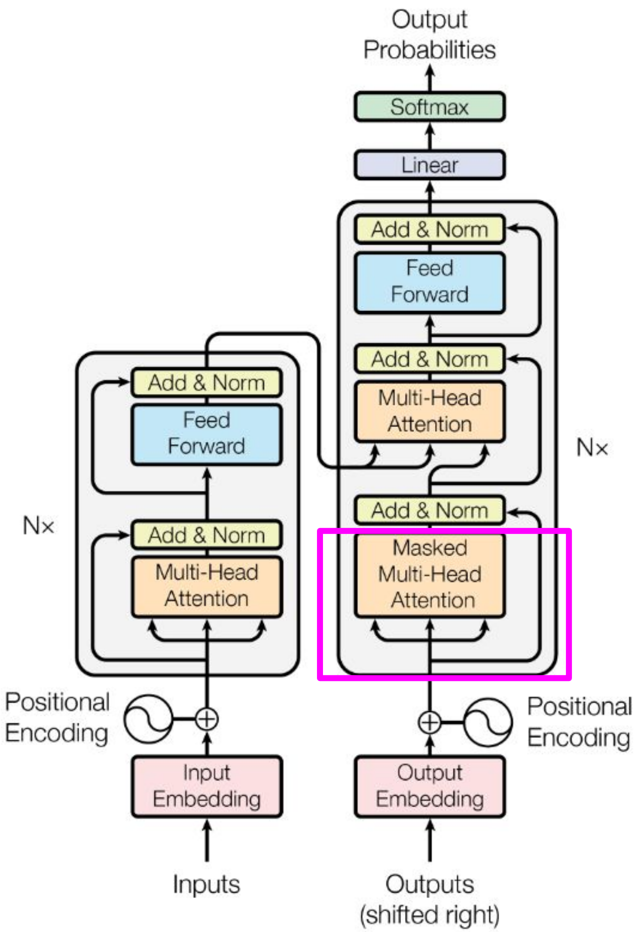


Linear & softmax layers

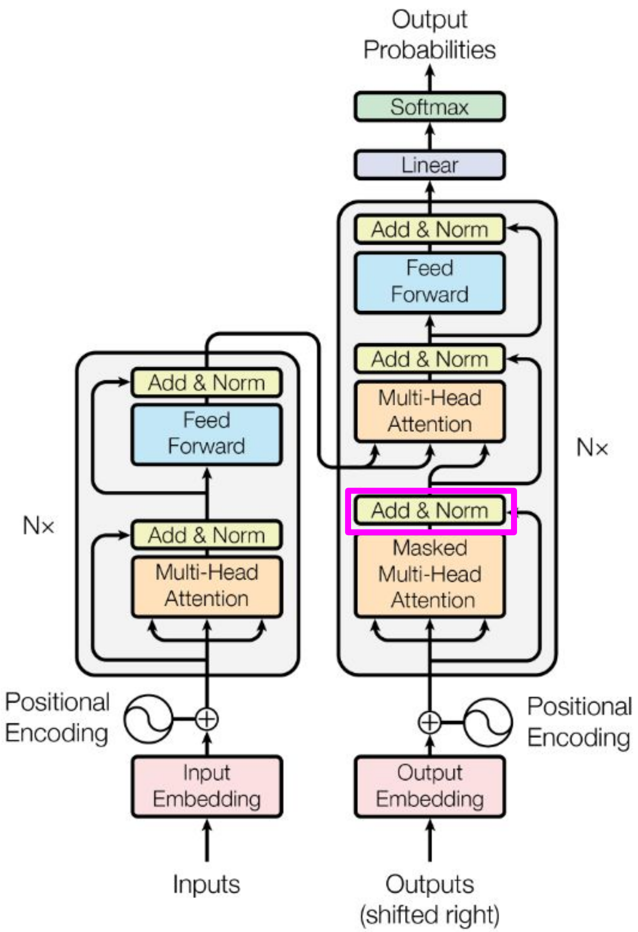


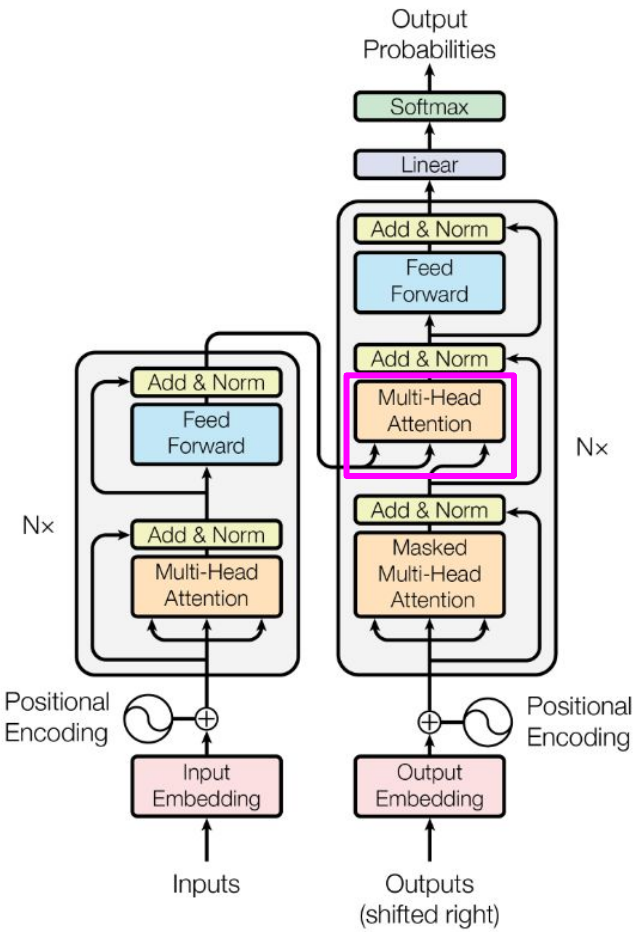


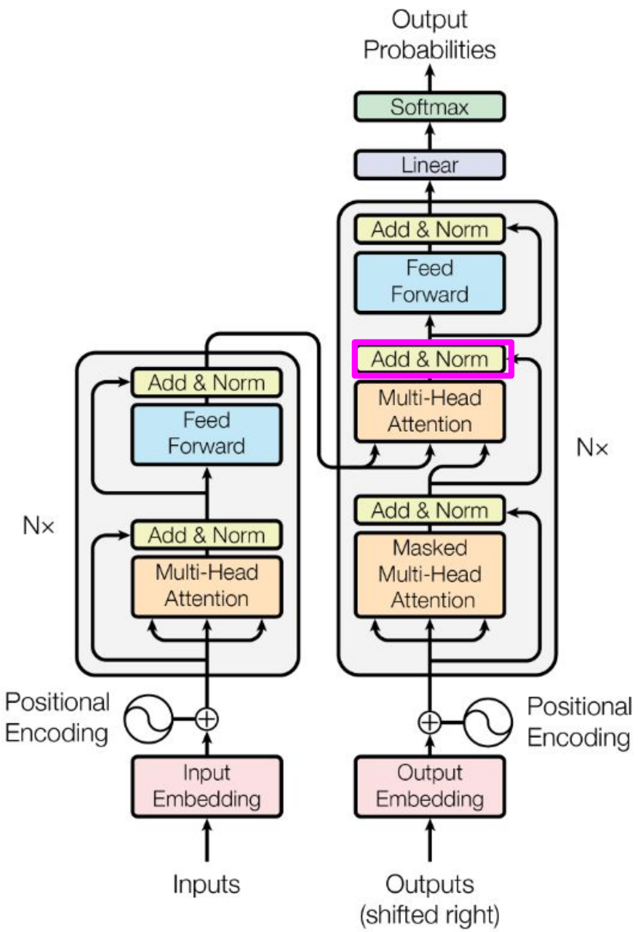
Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
N×
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
N×
Positional
Encoding
Input
Embedding
Inputs
Positional
Encoding
Output
Embedding
Outputs
(shifted right)

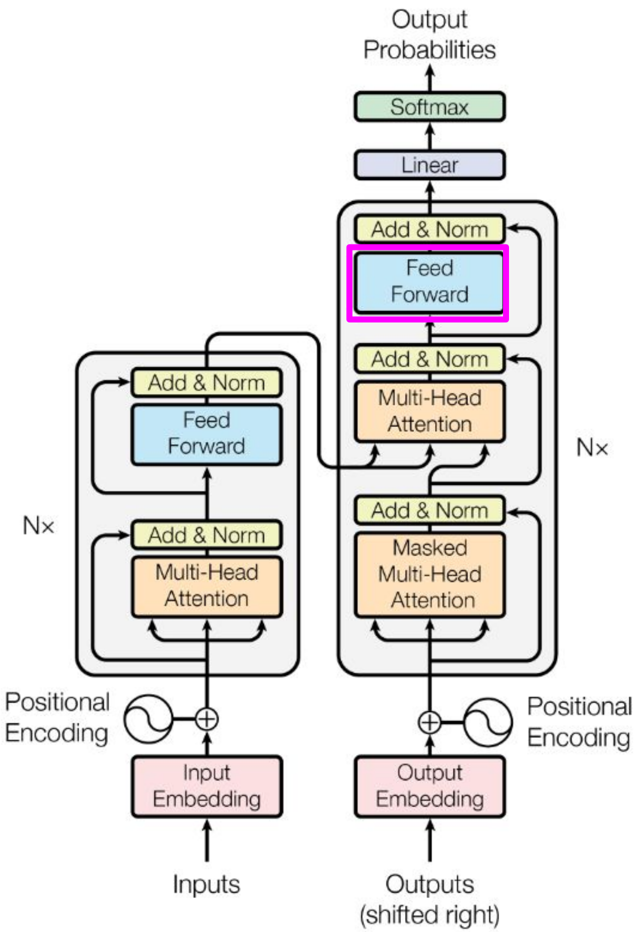


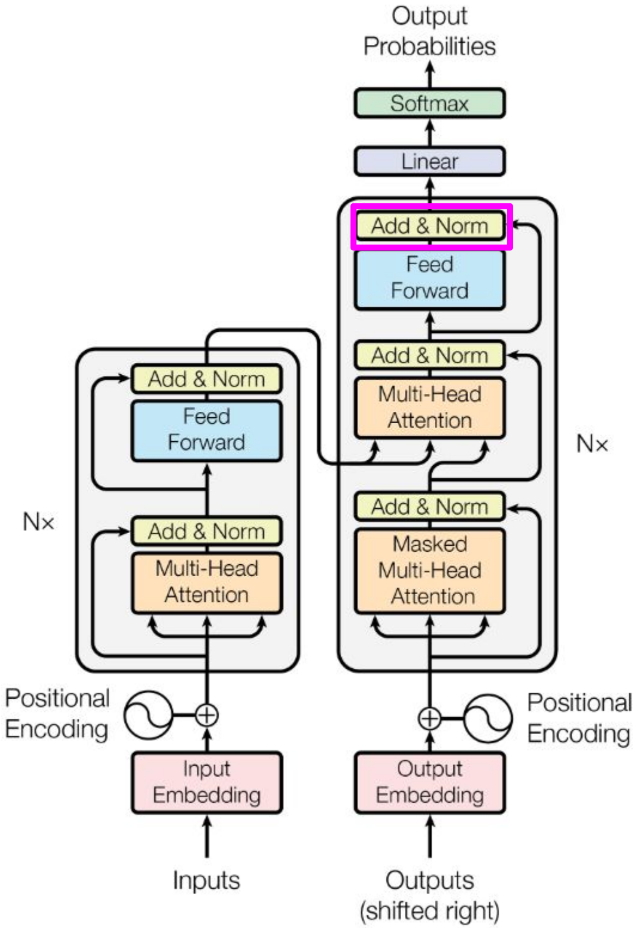
Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Add & Norm
Masked
Multi-Head
Attention
Add & Norm
Feed
Forward
Nx
Add & Norm
Multi-Head
Attention
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)


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)

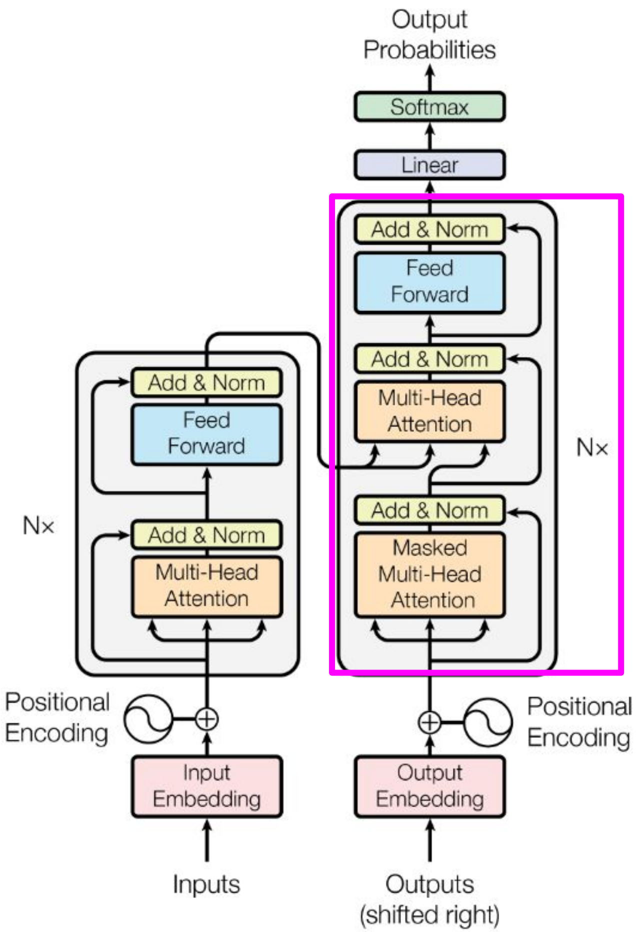

Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)


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

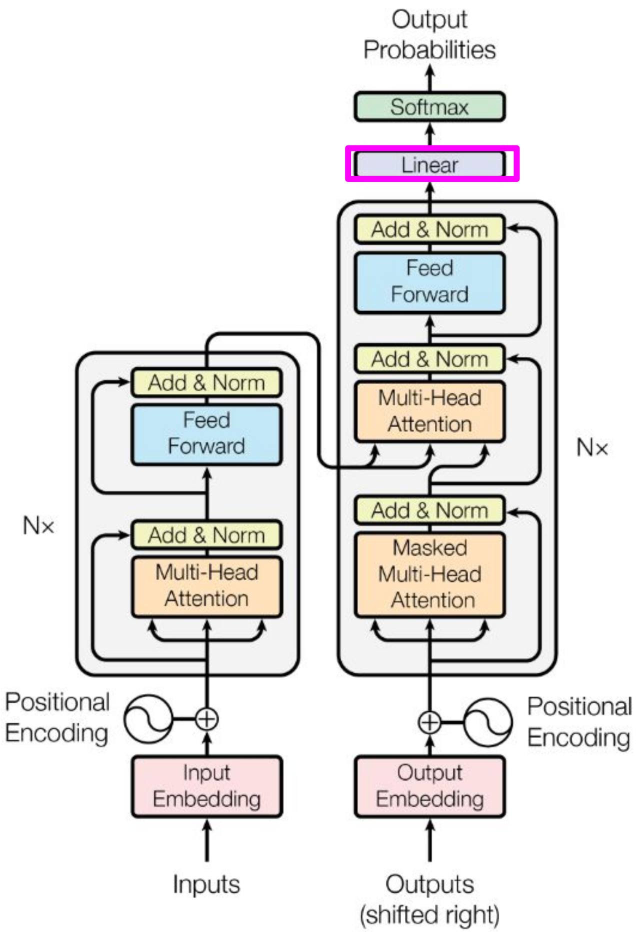

Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Add & Norm
Masked
Multi-Head
Attention
Add & Norm
Feed
Forward
Nx
Add & Norm
Multi-Head
Attention
Positional
Encoding
Input
Embedding
Inputs
Output
Embedding
Positional
Encoding
Outputs
(shifted right)

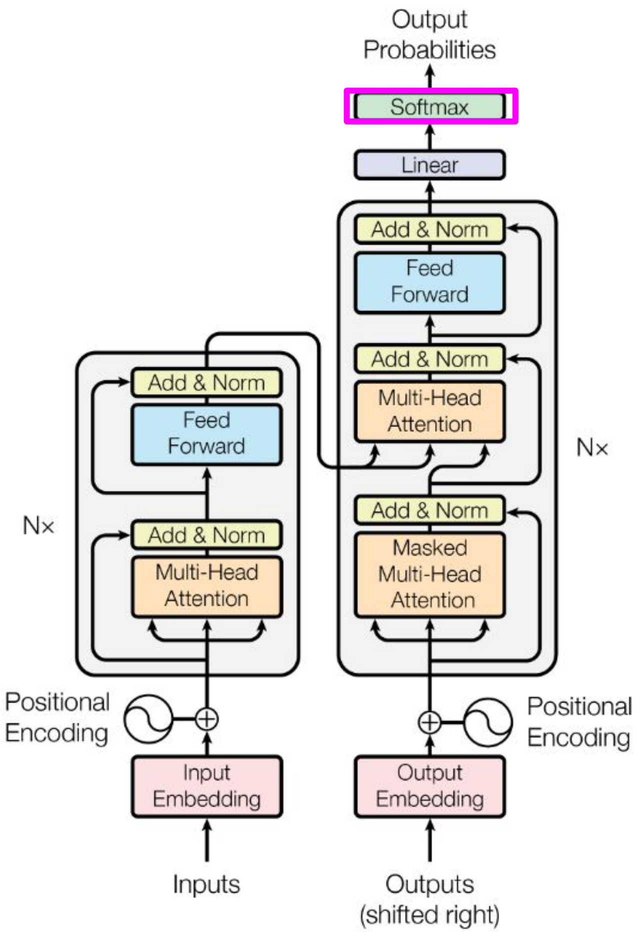


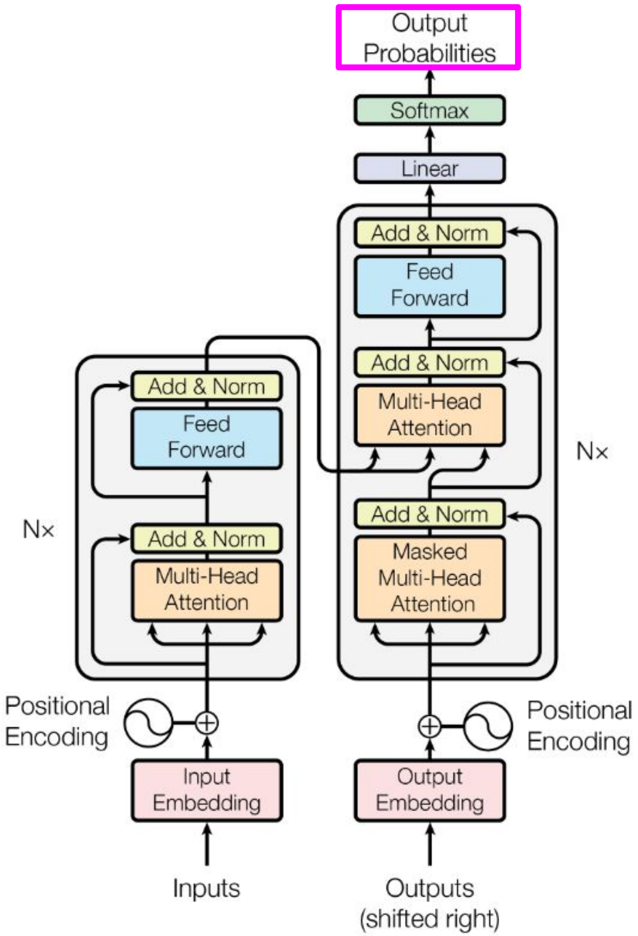
Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)



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)


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)

After all of this, why encoder + decoder?


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

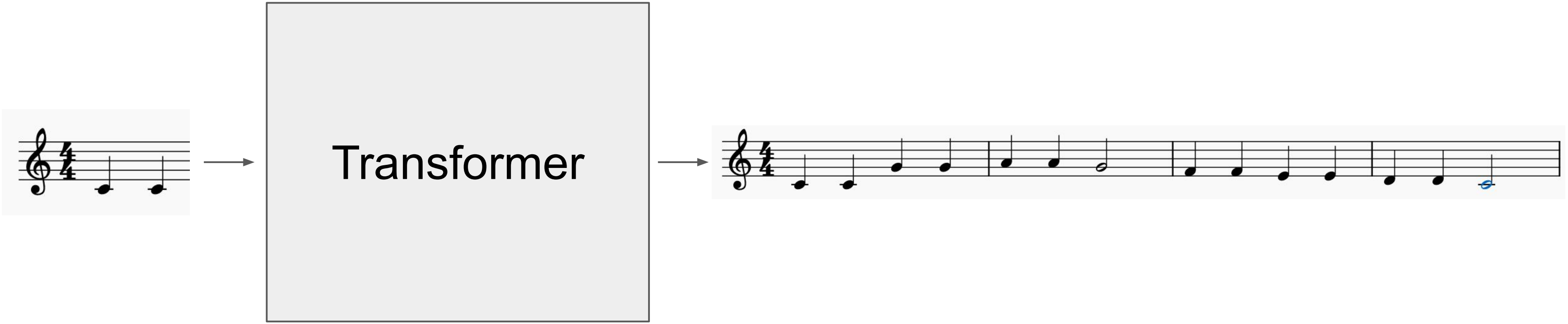
My experience with Transformers

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

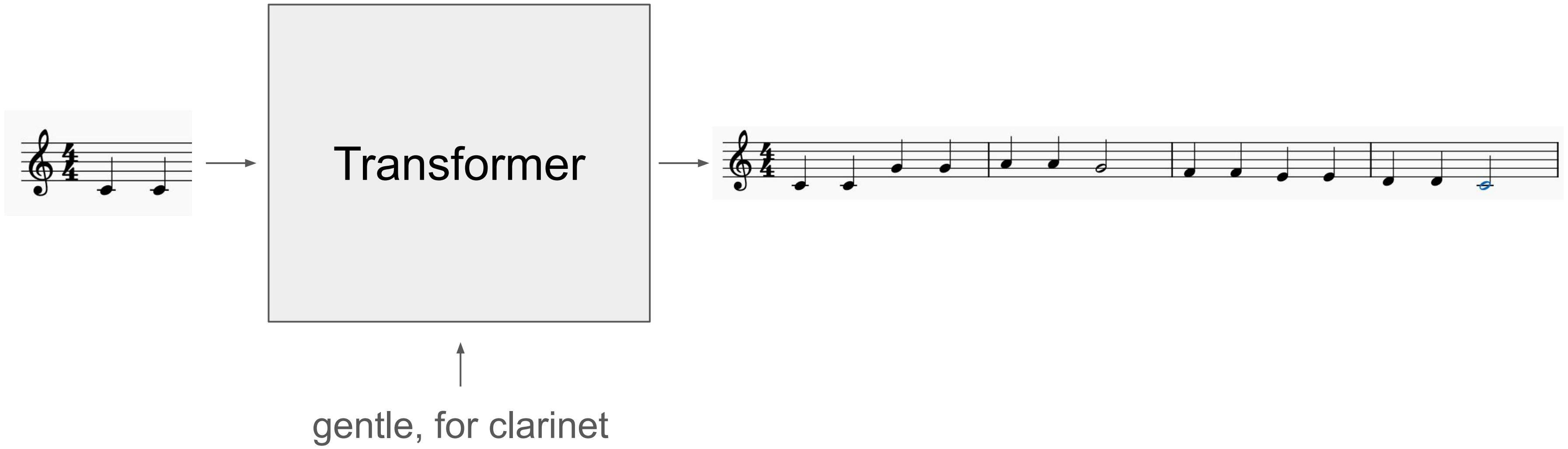
My experience with Transformers

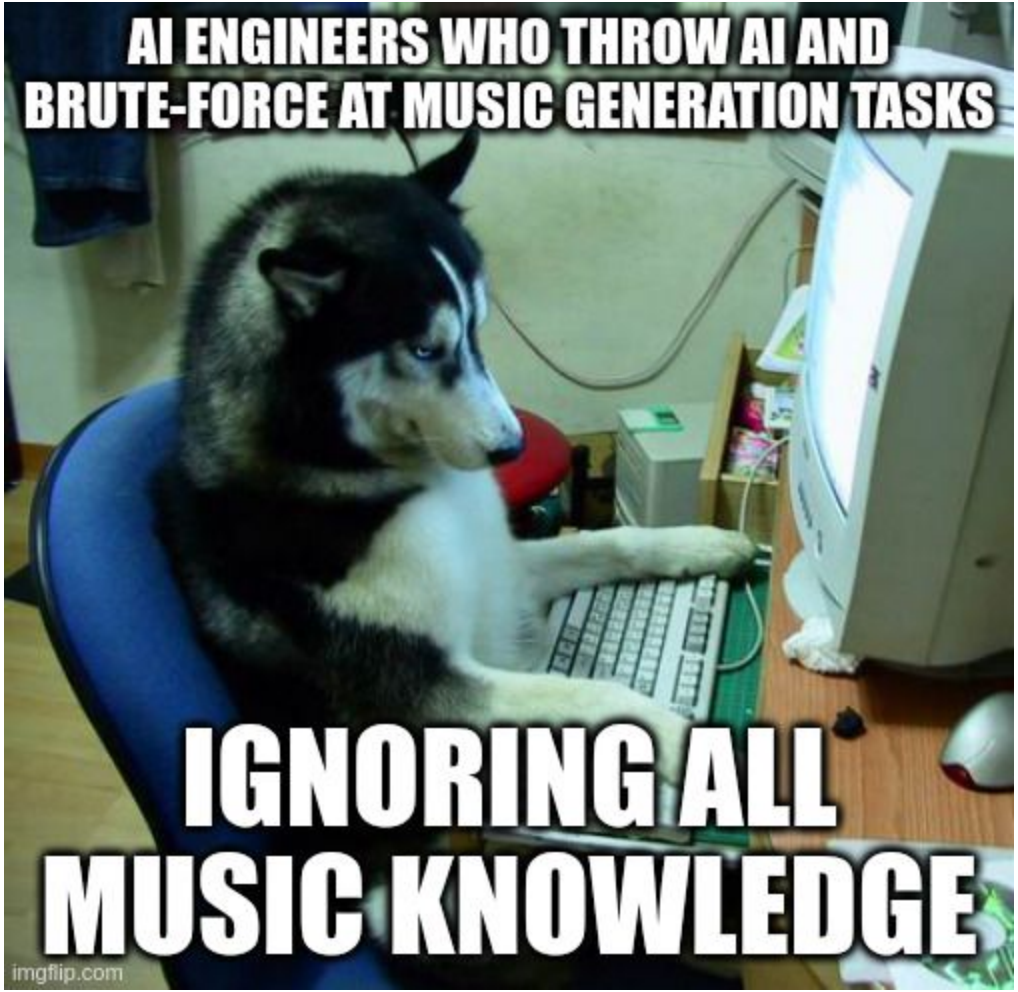
- Most capable model
- Massive amount of music data needed
- Music theory bozo
- Music representation is everything

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

Music gestures: Examples

- Melodic
    - Running scale up
    - Running scale down
- Harmonic
    - Perfect cadence
- Structure
    - Repetition
    - Variation

BEING

AN ESSAY

ON

VARIOUS

SCHEMATA

CHARACTERISTIC OF

EIGHTEENTH-CENTURY

MUSIC

FOR COURTLY CHAMBERS,

CHAPELS, AND THEATERS,

INCLUDING TASTEFUL

PASSAGES OF MUSIC

DRAWN FROM

MOST EXCELLENT

CHAPEL MASTERS

IN THE EMPLOY OF

NOBLE AND NOTEWORTHY

PERSONAGES,

Said Music

All Collected for the

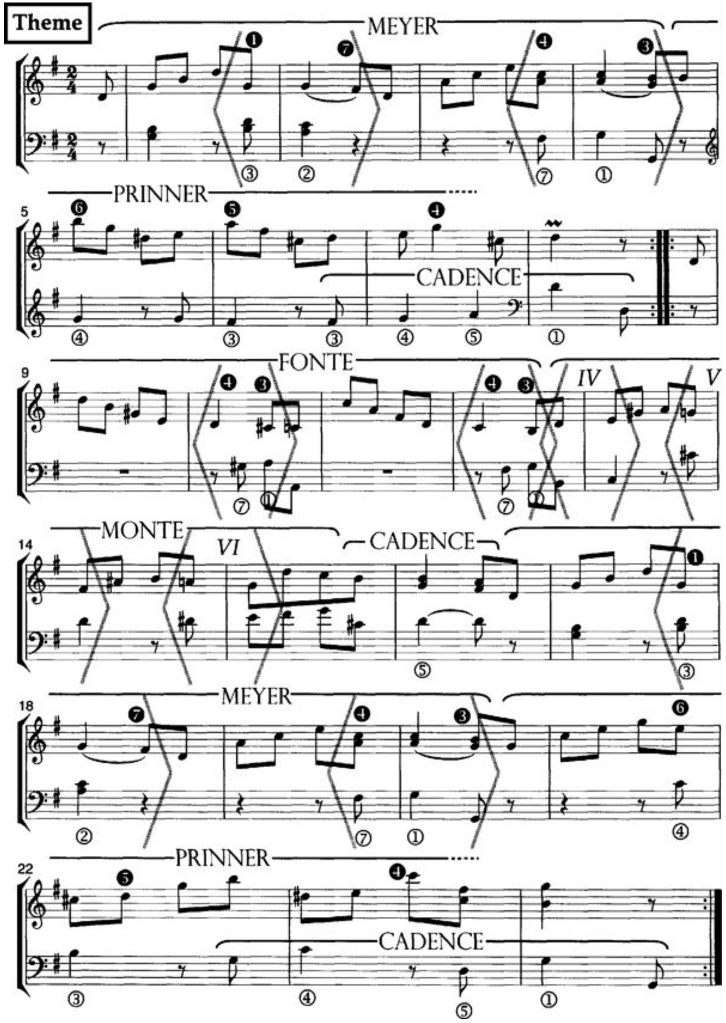
Reader's Delectation

on the

World Wide Web

MUSIC *in the* GALANT STYLE

ROBERT O. GJERDINGEN


Theme
MEYER
PRINNER
CADENCE
FONTE
IV
V
MONTE
VI
CADENCE
MEYER
PRINNER
CADENCE

My idea: Level 1 music representation

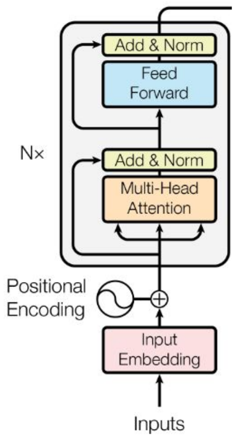
- High-level description of symbolic music

My idea: Level 1 music representation

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

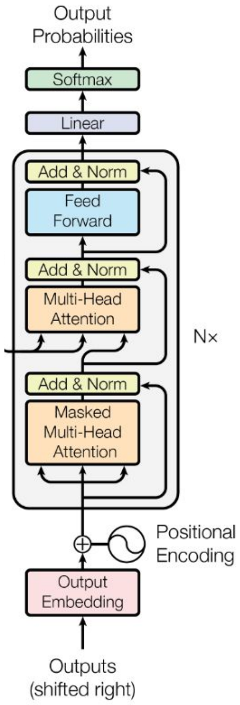
3 types of Transformer architectures

- Encoder only (BERT, MERT)



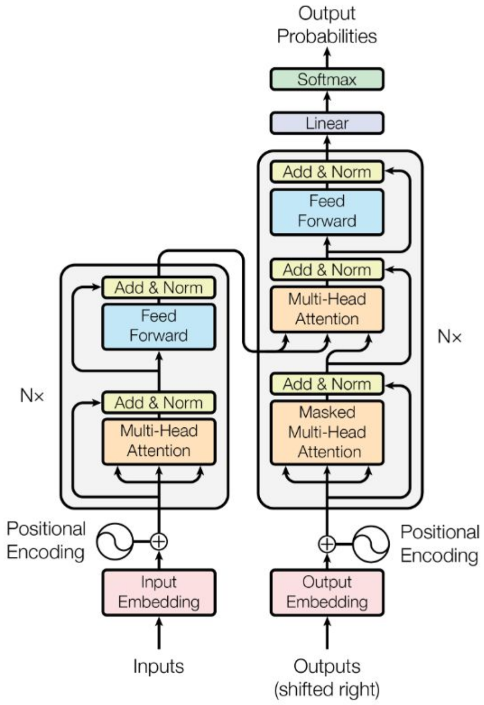
3 types of Transformer architectures

- Encoder only (BERT, MERT)
- Decoder only (GPT, MuPT)



3 types of Transformer architectures

- Encoder only (BERT, MERT)
- Decoder only (GPT, MuPT)
- Encoder - decoder



How can you condition music generation on text (*a la* Suno)?

How do we condition generation?

1. Single prompt (decoder)

[text tokens] <SEP> [music tokens]

How do we condition generation?

1. Single prompt (decoder)
2. Special tokens + segment embeddings (decoder)

<TEXT> blah blah </TEXT>
<MUSIC> … </MUSIC>

How do we condition generation?

1. Single prompt (decoder)
2. Special tokens + segment embeddings (decoder)
3. Prefix conditioning with learned embeddings (decoder)

[prefix embeddings] + [music tokens]

How do we condition generation?

1. Single prompt (decoder)
2. Special tokens + segment embeddings (decoder)
3. Prefix conditioning with learned embeddings (decoder)
4. Encoder - decoder conditioning

How do we condition generation?

1. Single prompt (decoder)
2. Special tokens + segment embeddings (decoder)
3. Prefix conditioning with learned embeddings (decoder)
4. Encoder - decoder conditioning
5. Multi-stream encoder - decoder conditioning

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

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

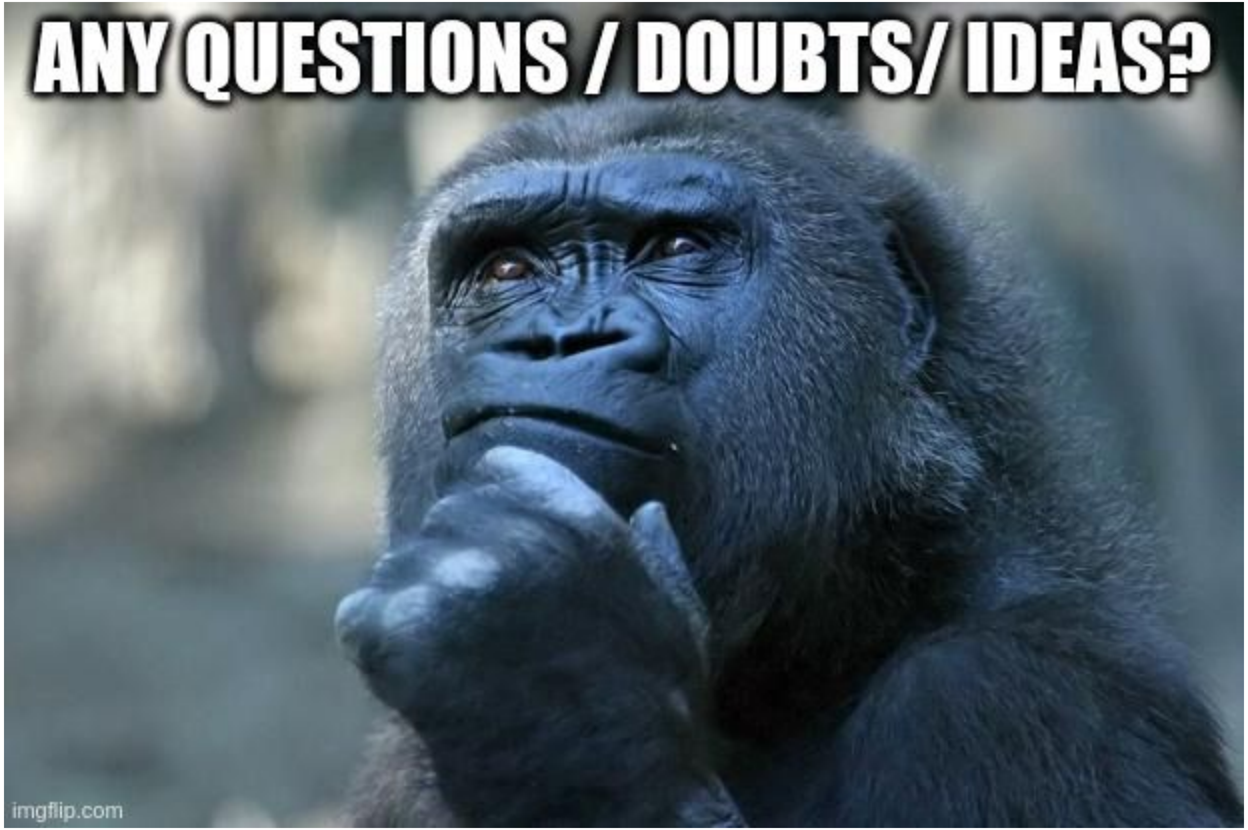
Tips for using Transformers

- 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

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?


ANY QUESTIONS / DOUBTS/ IDEAS?
imgflip.com