GCT634/AI613: Musical Applications of Machine Learning

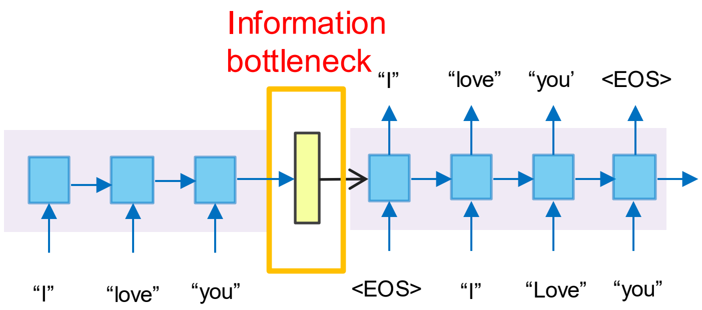
Symbolic Music Generation: Advanced Models

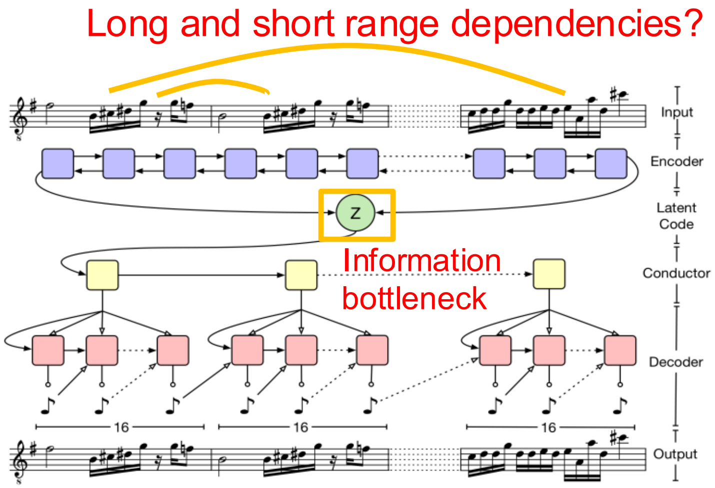
KAIST Graduate School of Culture Technology

Juhan Nam

Issues with RNN

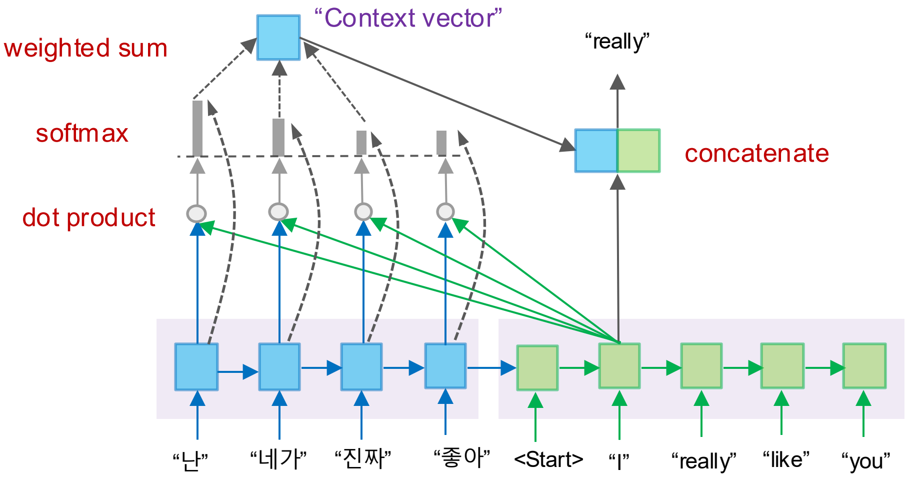
- Sequential computation inhibits parallelization (not like CNN)
- No explicit modeling of long and short range dependencies
- Information bottleneck in the encoder

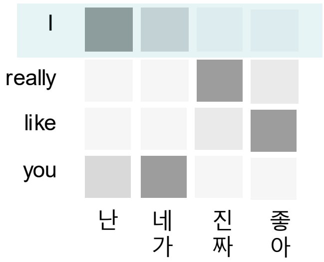




Attention Mechanism

- Direct connections between words in the encoder and decoder
  - A weighted sum of the input (“context vector”) is concatenated to each of the output words
  - The weights are computed from the one-to-one correspondence between words in the encoder and decoder





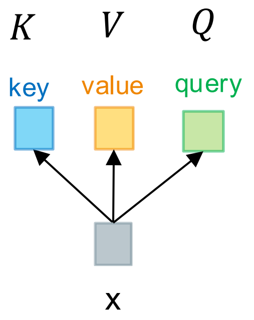
Neural Machine Translation by Jointly Learning to Align and Translate, Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, ICLR, 2015

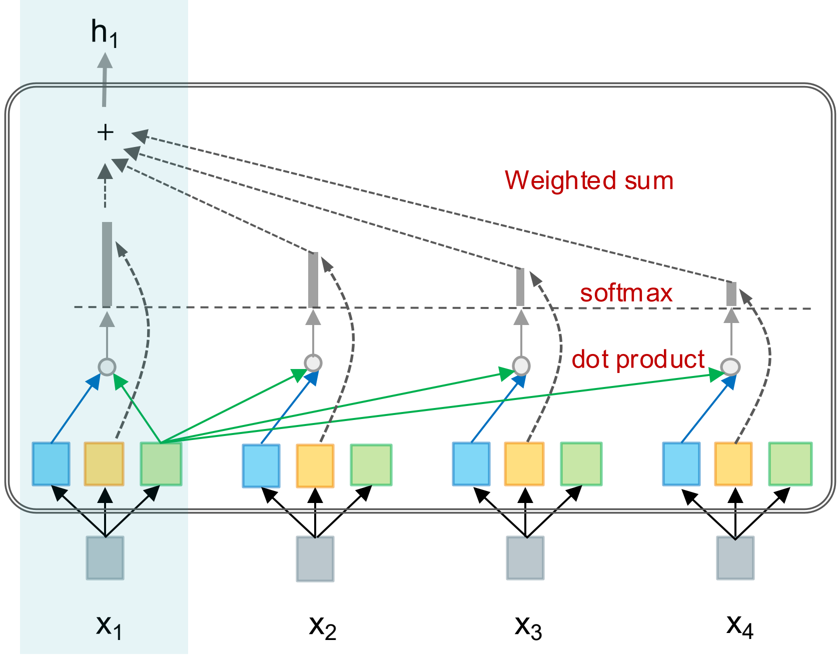
Self-Attention

- Direct connections can be made between elements within a sequence
  - Each input element is transformed into **key**, **query** and **value** via linear transforms

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

Scaling factor





“Self-attention layer”

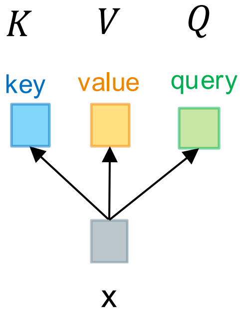
Attention Is All You Need, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, 2017

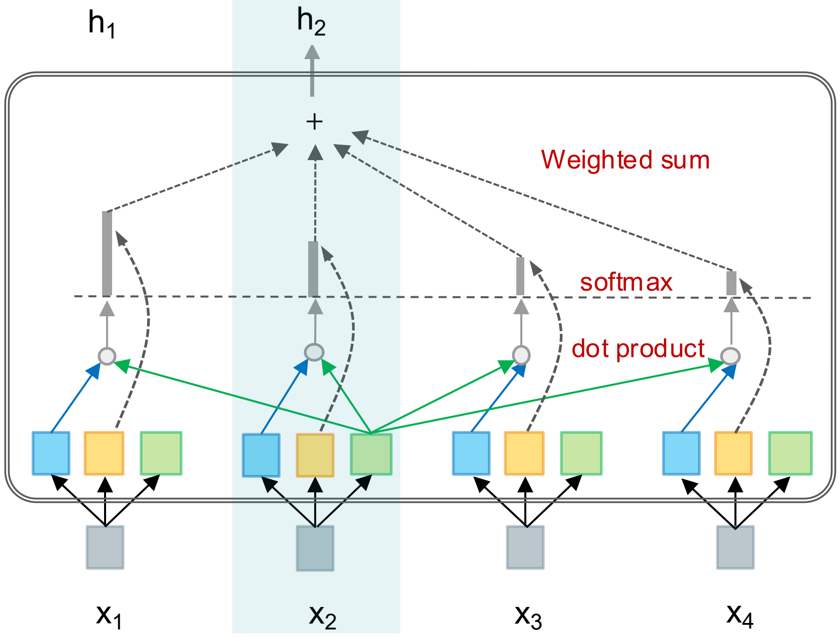
Self-Attention

- Direct connections can be made between elements within a sequence
  - Each input element is transformed into **key**, **query** and **value** via linear transforms

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

Scaling factor





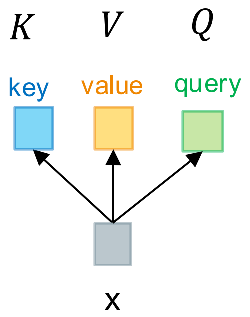
“Self-attention layer”

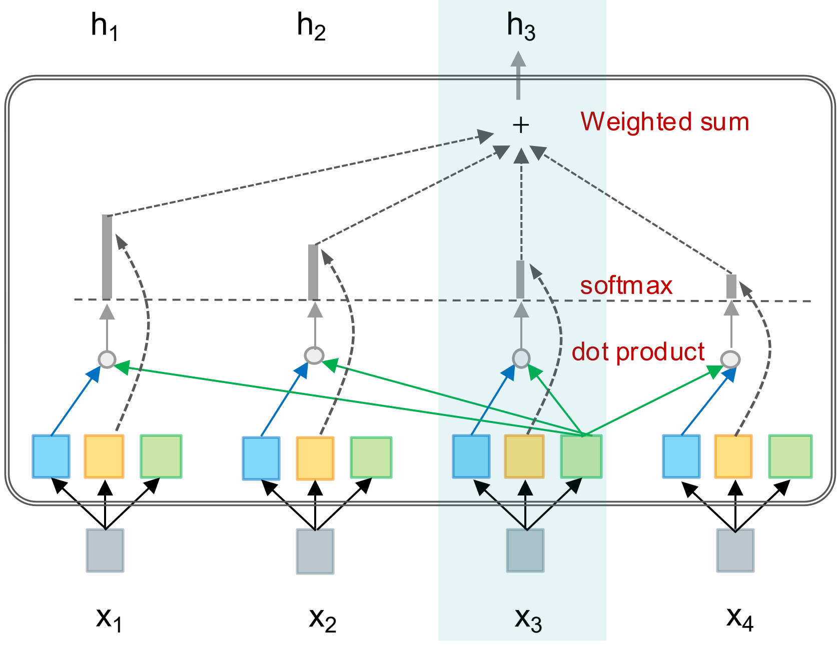
Self-Attention

- Direct connections can be made between elements within a sequence
  - Each input element is transformed into **key**, **query** and **value** via linear transforms

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

Scaling factor





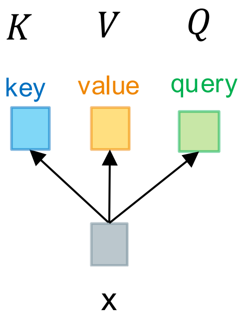
“Self-attention layer”

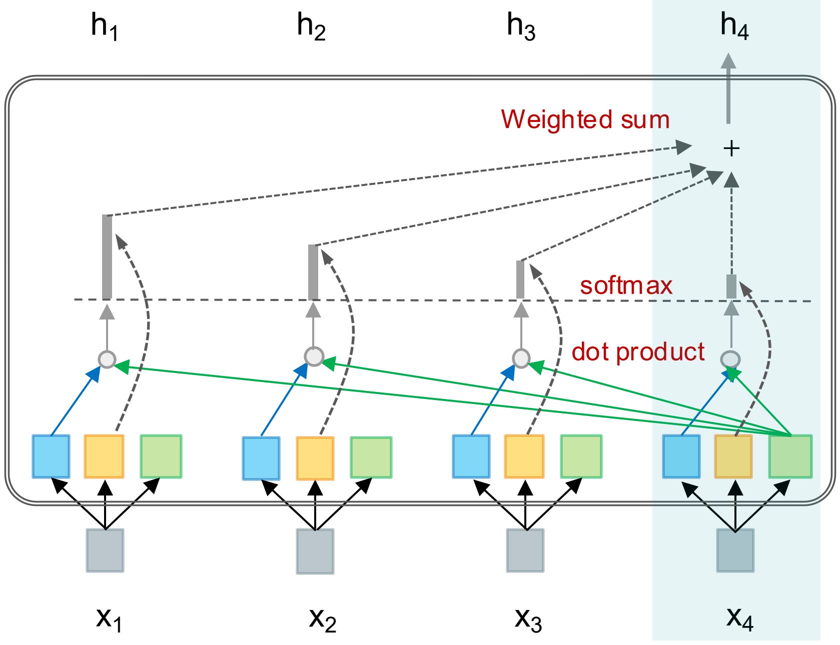
Self-Attention

- Direct connections can be made between elements within a sequence
    - Each input element is transformed into **key**, **query** and **value** via linear transforms

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

Scaling factor

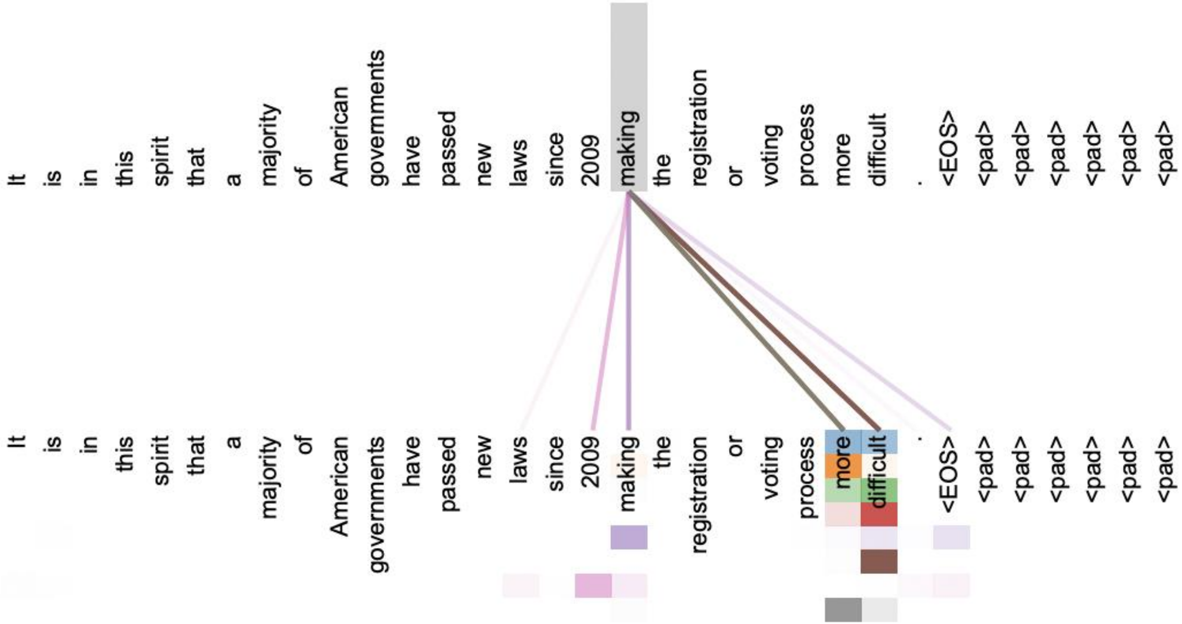




"Self-attention layer"

Self-Attention

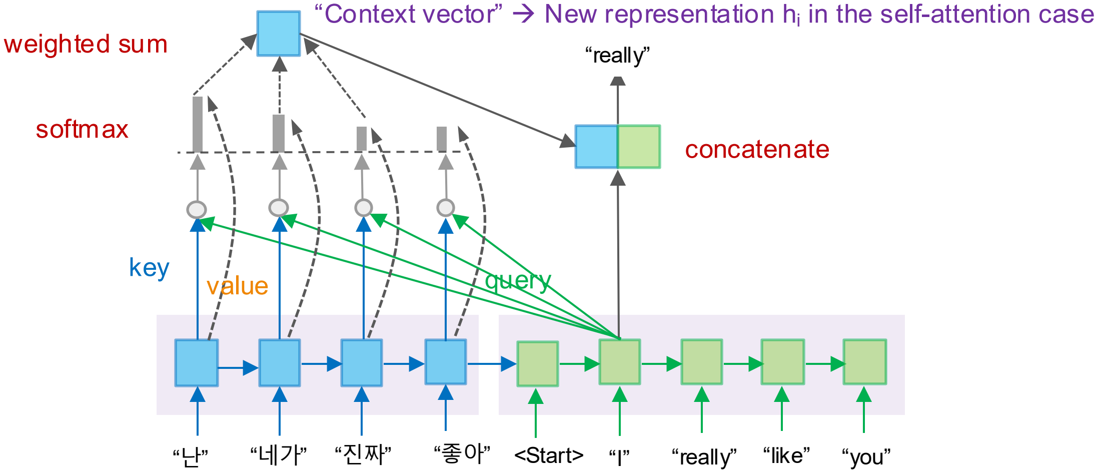
- We can visualize the interaction of input elements from the self-attention

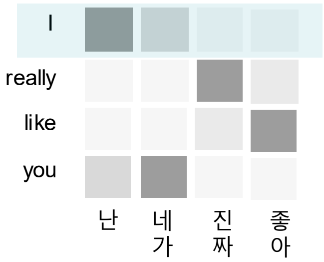


Attention Is All You Need, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, 2017

Attention Mechanism

- Direct connections between words in the encoder and decoder
  - A weighted sum of the input (“context vector”) is concatenated to each of the output words
  - The weights are computed from the one-to-one correspondence between words in the encoder and decoder

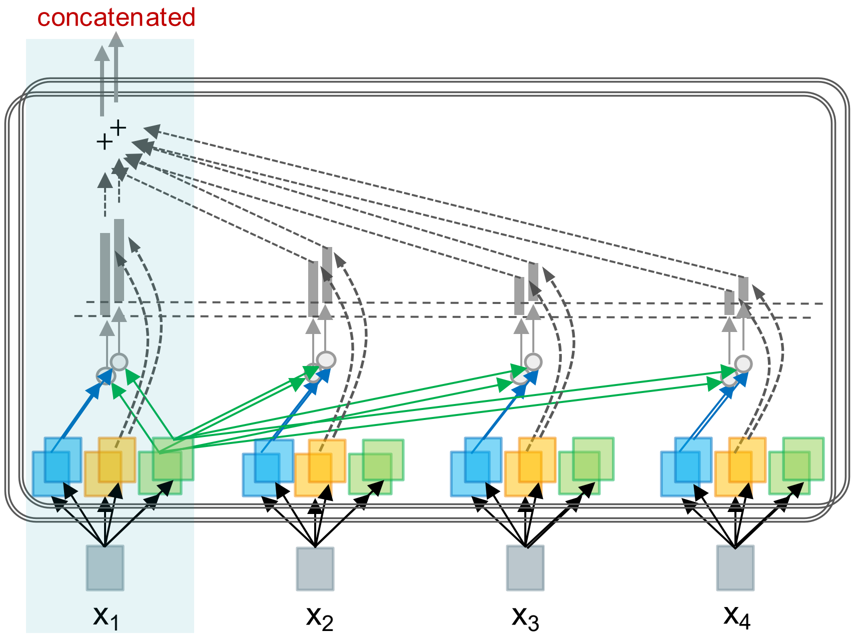




Neural Machine Translation by Jointly Learning to Align and Translate, Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, ICLR, 2015

Self-Attention

- Multi-head attention
  - Multiple independent **key**, **query** and **value** that capture different types of dependency in the sequence



Self-Attention

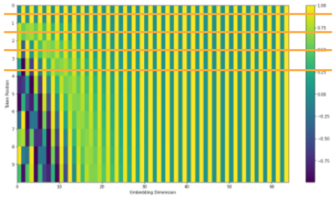
- "Re-representation" of input
    - Based on interactions between input elements
- Constant "path length" between two arritrary positions (unlike RNN)
    - Permutation-invariant
    - Need to add positional information for sequence modeling
- Trivial to parallelize
    - Effective use of GPU

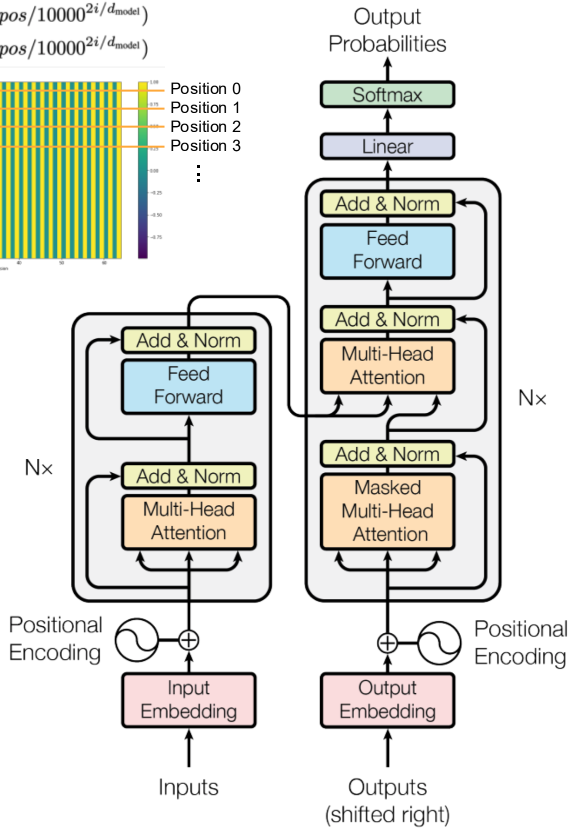
Attention Is All You Need, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, 2017

Transformer

- **Position encoding** is added to the input
  - Self-attention is permutation-invariant
- A single module is composed of
  - Multi-head attention layer
  - Position-wise feed forward layer
  - Add skip connections and normalization layers
    - The skip connection carries the position information
- Masking is added to the attention in the decoder
  - For causal self-attention

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$



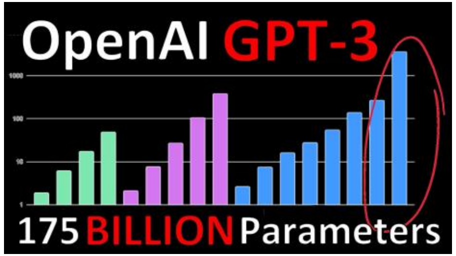


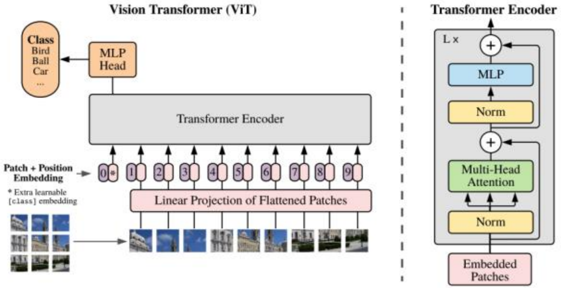
Attention Is All You Need, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, 2017

Transformer

- Used in the state-of-the-arts models in natural language processing
  - Machine Translation
  - Language modeling
- Used in computed vision as well
  - Image classification
  - Image generation
- Recently used in MIR
  - Music auto-tagging

Bert





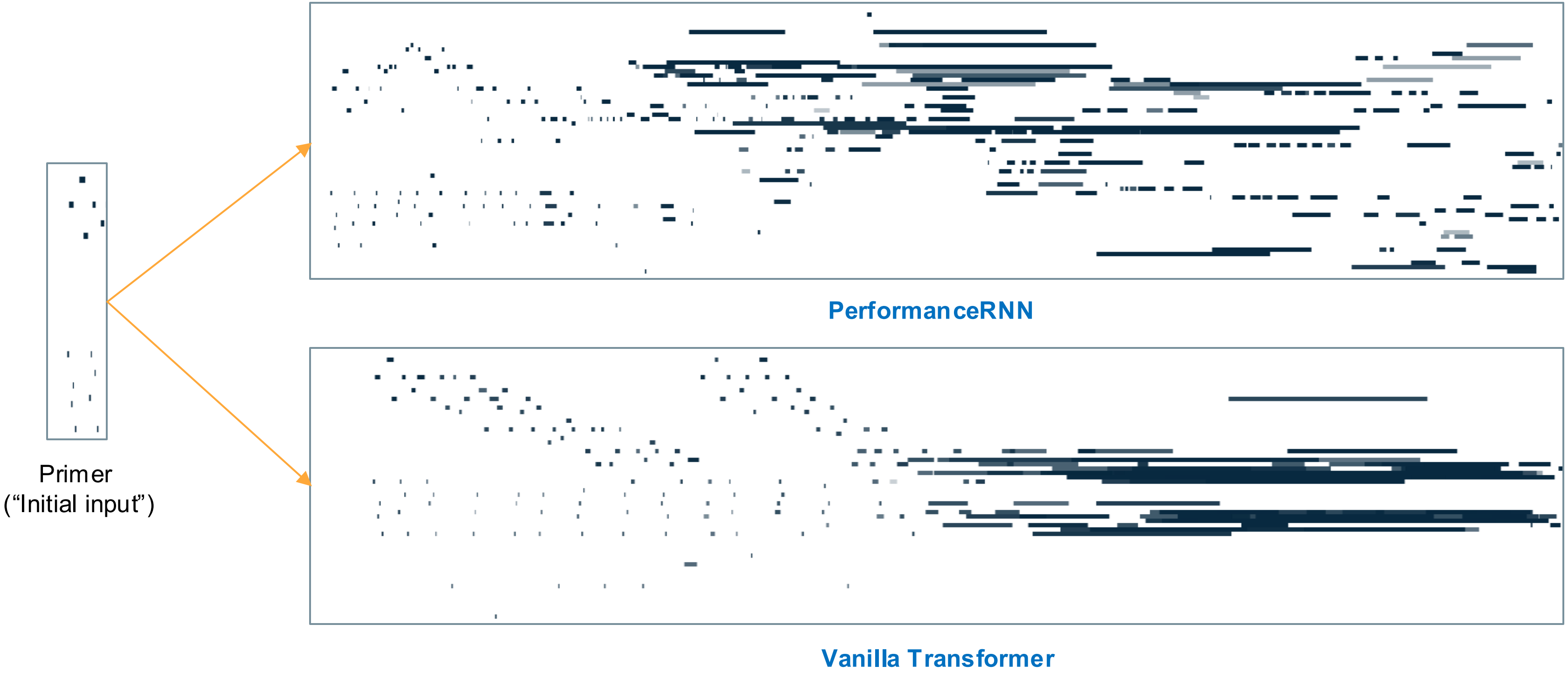
Vision transformer

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, Alexey Dosovitskiy et al., ICLR, 2020

Semi-Supervised Music Tagging Transformer, Minz Won, Keunwoo Choi, Xavier Serra, ISMIR, 2021

Music Transformer

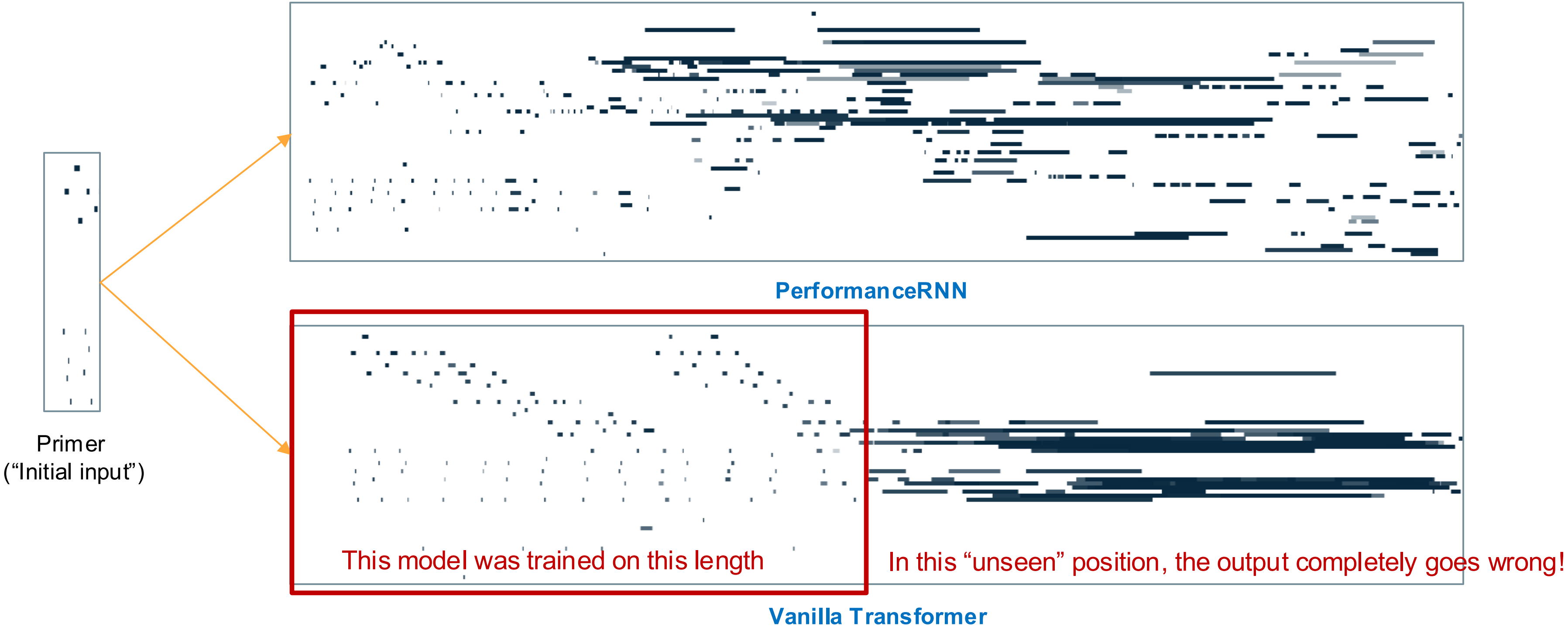
- How about applying transformer to music generation?



Source: https://magenta.tensorflow.org/music-transformer

Transformer for Music Generation

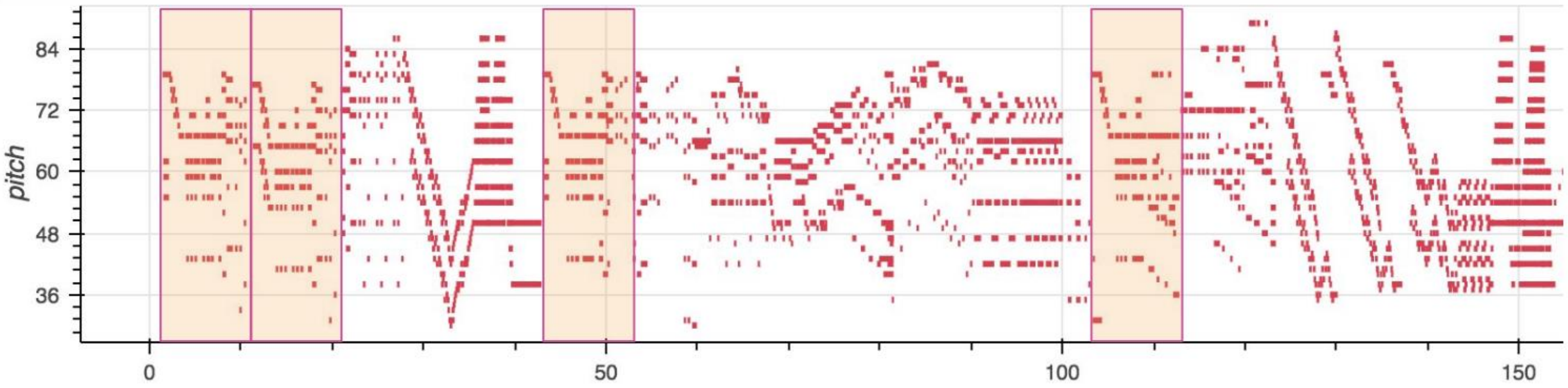
- What's wrong?



Source: https://magenta.tensorflow.org/music-transformer

Self-Similarity in Music

- Music has a motif and it is repeated immediately and also at a distance

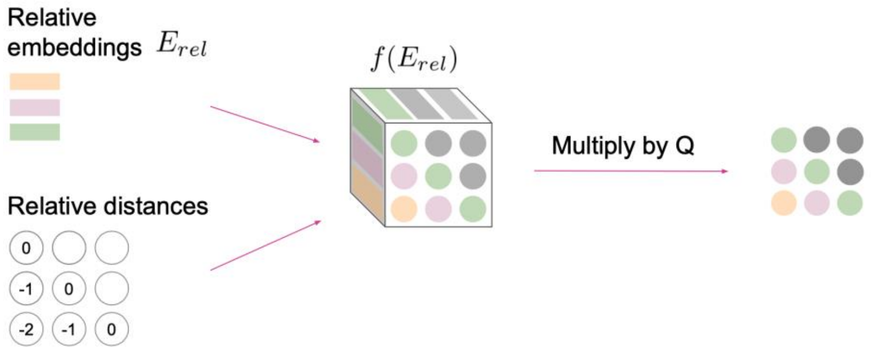


Source: Cheng-Zhi Anna Huang

Music Transformer

- Take the advantage of self-similarity in music by using the **relative position** instead of the absolute position
  - A straightforward way is using a pair-wise distance between two points: the relative distance becomes 2D
  - A 3D tensor is necessary for positional encoding: too much memory !

$$softmax(QK^\top + \underbrace{Qf(E_{rel})}_{\text{Relative position encoding}})$$



Music Transformer, Cheng-Zhi Anna Huang, et al, ICLR, 2019

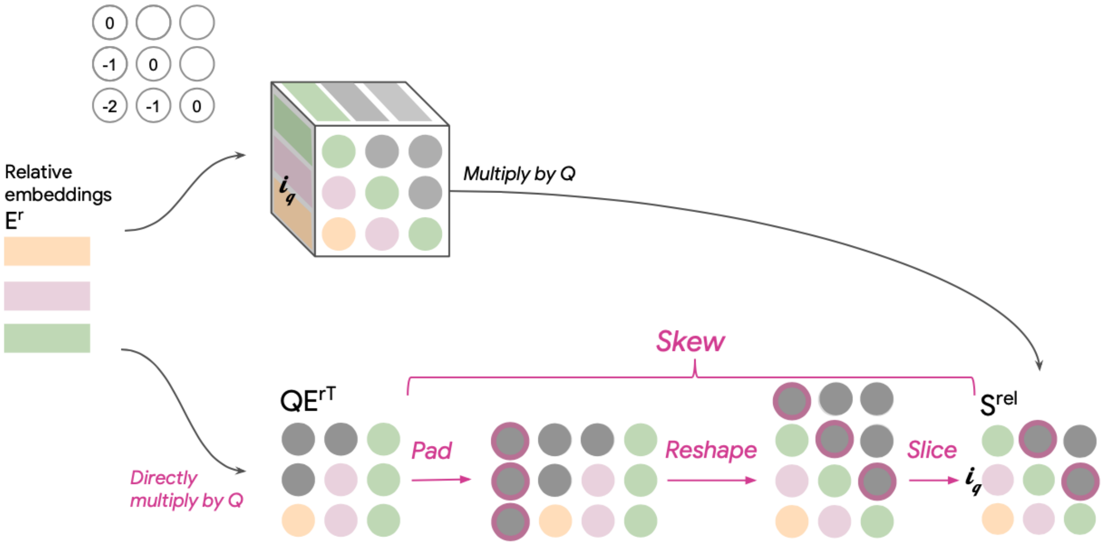
Music Transformer

- Skewing to reduce relative memory

Per layer, L=2048, D=512

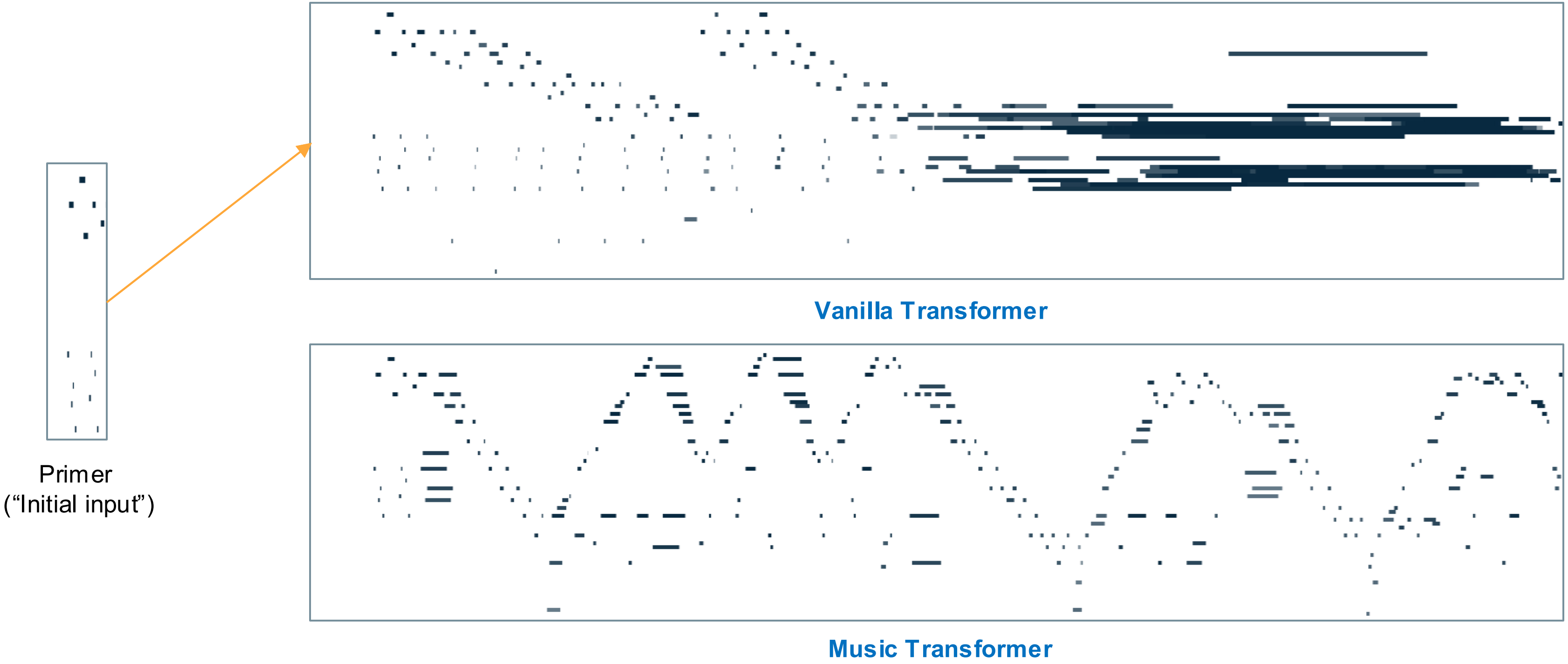
Previous work
$O(L^2D)$: **8.5 GB**

Our work
$O(LD)$: **4.2 MB**



Music Transformer

- Consistent generation!



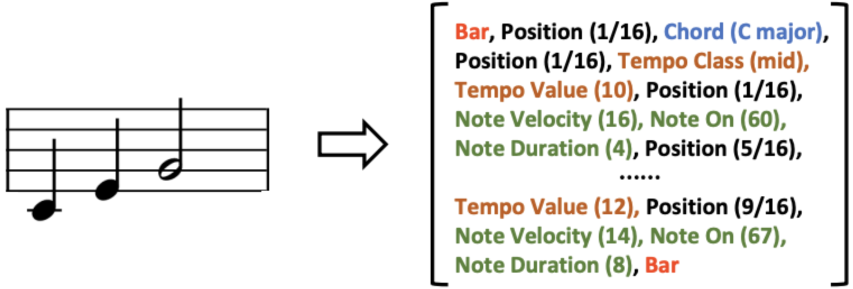
Source: https://magenta.tensorflow.org/music-transformer

Issues in Music Transformer

- Music transformer does not have a built-in notion of beats/downbeats
- If a **time-shift event** is generated at a wrong position, the entire notes in the following time steps are affected and the rhythm becomes unstable
  - This is a serious problem in Pop music

Sheet music as multi-element events

- Parse the music notation and tokenize them
    - A structured text sequence
        - Bar, chord, tempo, note, and so on
    - This rich information can be useful in generating more musical output
        - But, it may need manual annotations
    - No standard method



REvamped MIDI-derived events (REMI)

Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions, Yu-Siang Huang, Yi-Hsuan Yang, 2020

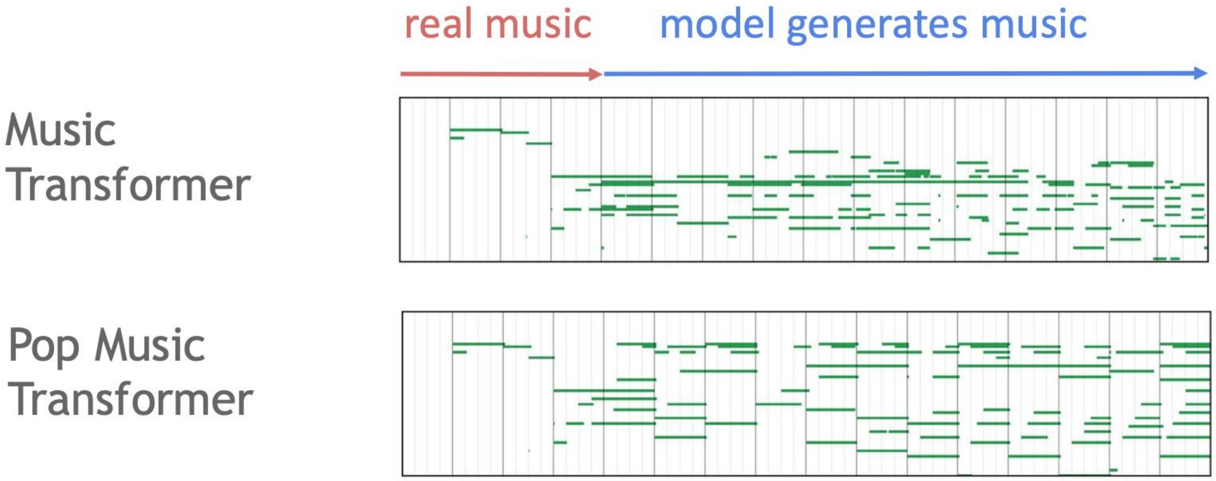
Pop Music Transformer

- Revamped MIDI-derived events (REMI): a token representation for pop music
  - Use **position & bar (beat & downbeat)** instead of **time-shift**
  - Use **quantized note-duration** instead of **note-off**
  - Add chord & tempo related tokens

	MIDI-like	REMI
Note onset	NOTE-ON (0-127)	NOTE-ON (0-127)
Note offset	NOTE-OFF (0-127)	NOTE DURATION (32th note multiples)
Note velocity	NOTE VELOCITY (32 bins)	NOTE VELOCITY (32 bins)
Time grid	TIME-SHIFT (10-1000ms)	POSITION (16 bins) & BAR (1)
Tempo changes	✗	TEMPO (30-209 BPM)
Chord	✗	CHORD (60 types)

Pop Music Transformer

- The input symbolic representation is designed for **pop music** which features **steady beats**
  - Use Transformer-XL instead of the vanilla Transformer



Demo: https://soundcloud.com/yating_ai/sets/ai-piano-generation-demo-202004

Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions, Yu-Siang Huang, Yi-Hsuan Yang, et al, 2019