Attention is All you Need

JUHYEONG LEE, KNUAIR

Contents

Vaswani, Ashish, et al. "Attention is all you need. " Advances in neural information processing systems 30 (2017). Citations : 31,758 (2022. 5)

Jurafsky, Dan, et al. "Speech and Language Processing", 3rd, (2021)

1. Introduction
2. Self-Attention Networks: Transformers
3. Transformer Blocks
4. Multi-Head Attention
5. Positional Encoding

Introduction

Timeline

LSTM
(1997)

Seq-to-Seq
(2014)

Attention
Mechanism
(2015)

Transformer
(2017)

BERT
(2019)

Contribution

- Previous works : RNN or CNN encoder-decoder + attention mechanism
- Proposal : Based *solely on* ***attention mechanism***

Recurrent Models

- Sequential nature precludes parrellelization

- Becomes critical at longer sequence lengths,
  as memory constraints limit batching across examples

Attention Mechanisms

- Become an integral part of sequence modeling
- Allow modeling of dependencies **without regard to** their distance in the input or output sequences
- Most of attention models : Based on RNN

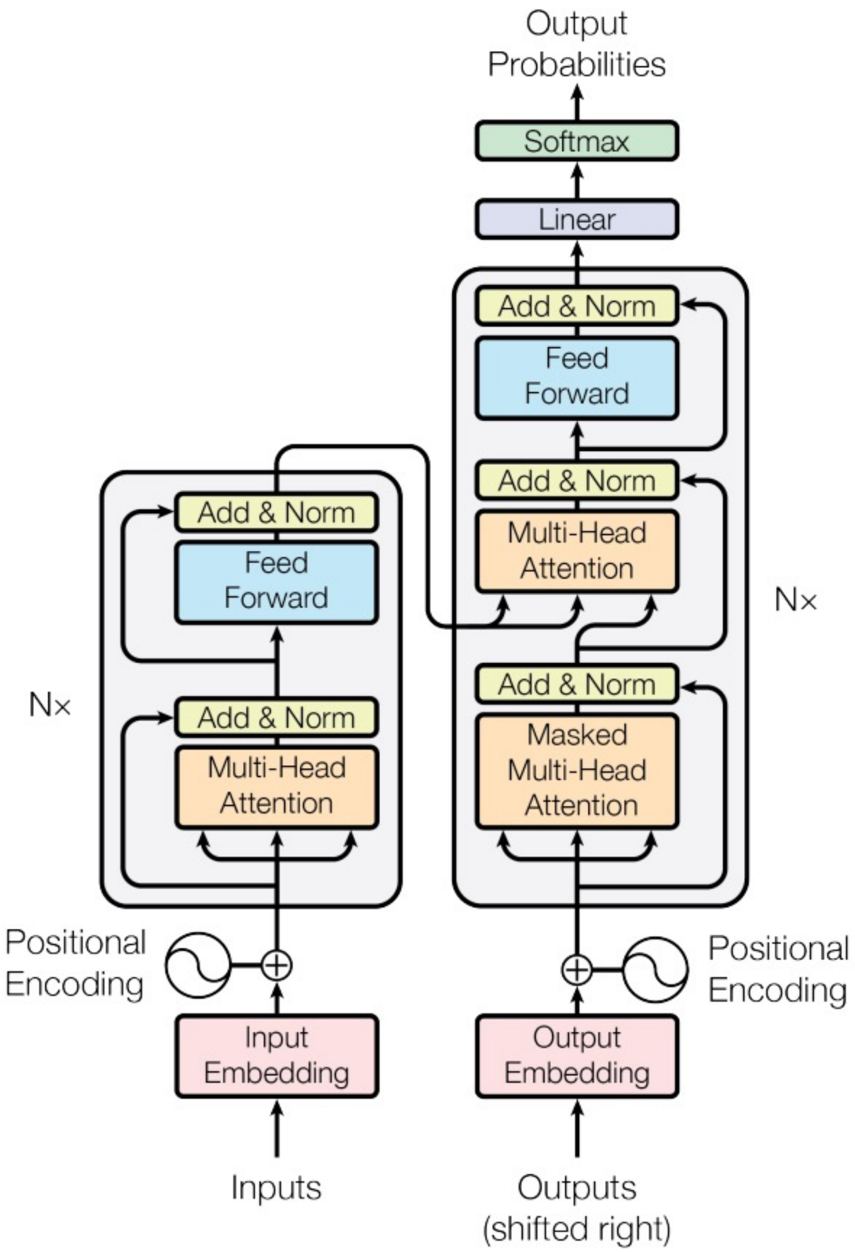
Transformer

- A model architecture *relying entirely on attention mechanism*
- Allows parallelization
- Faster training time

Self-Attention Networks: Transformers

Transformers

- Model Architecture :
  N transformer blocks
  - Simple Linear Layers
  - Self-Attention Layers
  - (Position-Wise) Feed Forward Networks
- Input : $(x_1, \cdots, x_n)$
- Output : $(y_1, \cdots, y_n)$

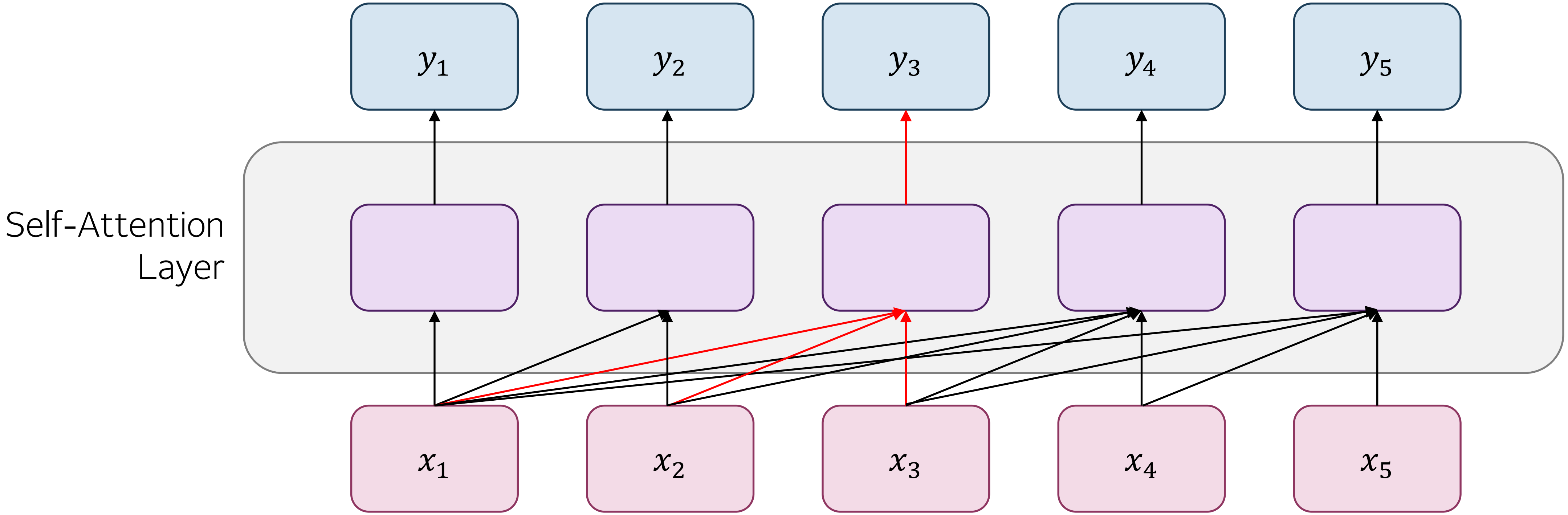


Self-Attention

- Allows a network to directly extract and use information from arbitrarily large contexts
- Dose not need to pass the information through recurrent networks

Self-Attention

- When processing each item in the input, the model has access to preceding inputs and itself (no access beyond current one).



Attention Revisited

- Attention-based approach
  - Compare an item of interest to a collection of other items
  - Reveals their relevance in the current context
- Self-Attention
  - the set of comparisons are to other elements within a given sequence

Attention Revisited

- Attention Score
  - $\text{score}(x_i, x_j) = x_i \cdot x_j \rightarrow$ Dot Product
  - With normalize, $\alpha_{ij} = \text{softmax}(\text{score}(x_i, x_j)), \forall j \leq i$
- Output y_i
  - $y_i = \sum_{j \leq i} \alpha_{ij} x_j$

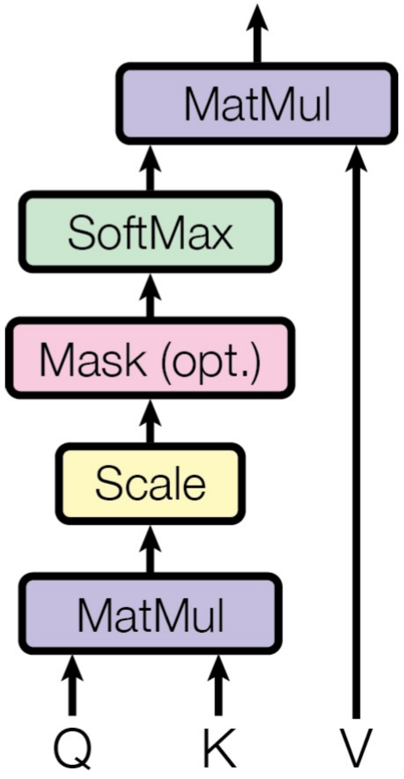
Query, Key, Value

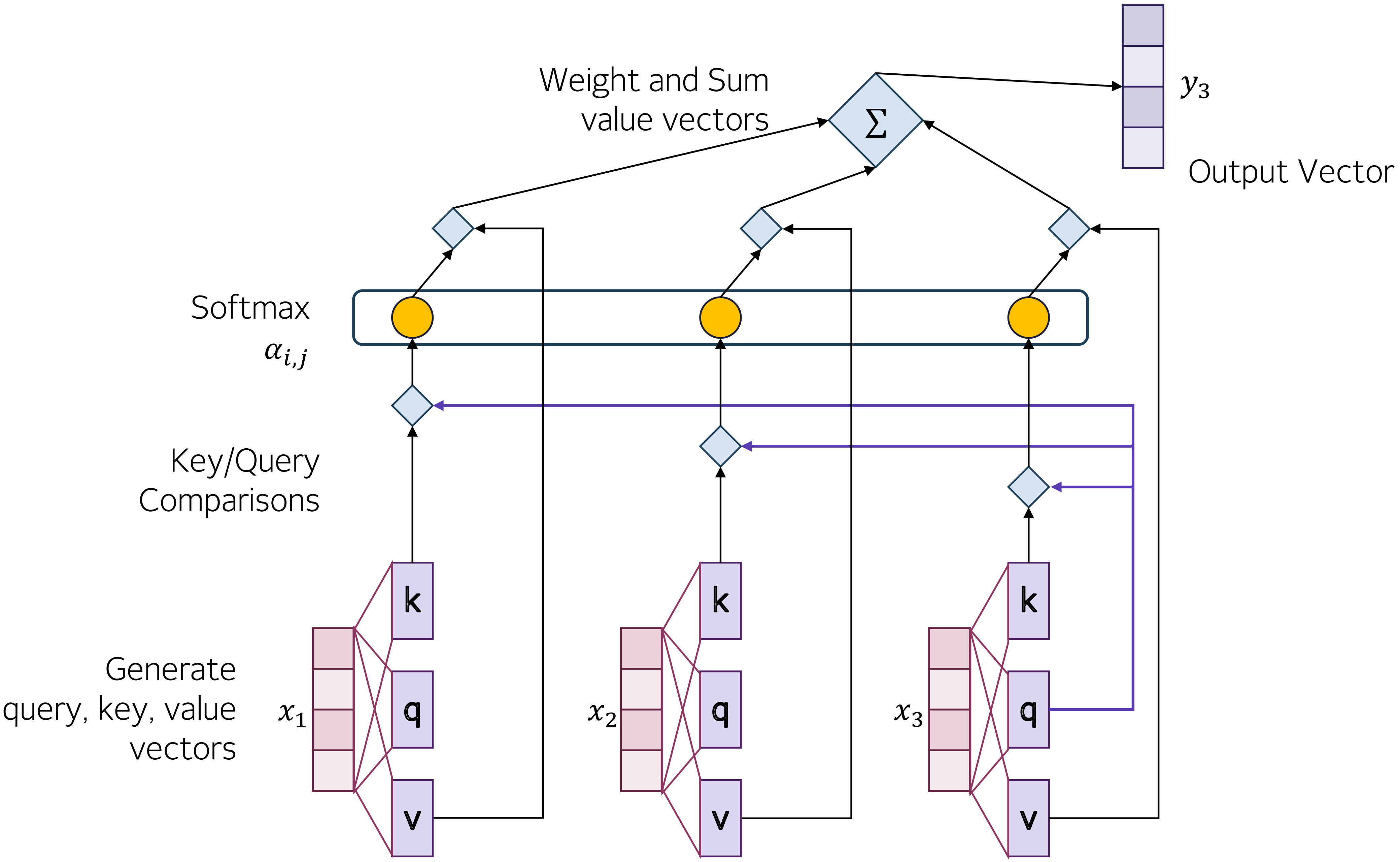
- Three different input embeddings
- Query $q_i = W^Q x_i$
    - Current focus of attention
- Key $k_i = W^K x_i$
    - Preceding input
- Value $v_i = W^V x_i$
    - Used to compute the output for the query q

Query, Key, Value

- Attention Score →
  Dot Product (Self) Attention
  - $\text{score}(x_i, x_j) = q_i \cdot k_j$
- Modified Attention Score →
  Scaled Dot Product (Self) Attention
  - $\text{score}(x_i, x_j) = \frac{q_i \cdot k_j}{\sqrt{d_k}}$
- Output y_i
  - $y_i = \sum_{j \leq i} \alpha_{ij} v_j$

Scaled Dot-Product Attention



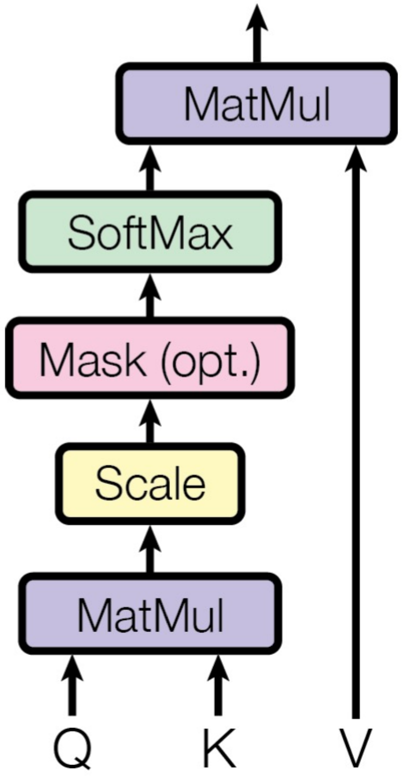


Weight and Sum
value vectors
Σ
y_3
Output Vector
Softmax
$\alpha_{i,j}$
Key/Query
Comparisons
Generate
query, key, value
vectors
x_1
x_2
x_3
k
q
v
k
q
v
k
q
v

From Vectors To Matrices

- Each processes are independent; can be parallelized
- $Q = XW^Q; K = XW^K; V = XW^V$
- $\mathrm{SelfAttention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$

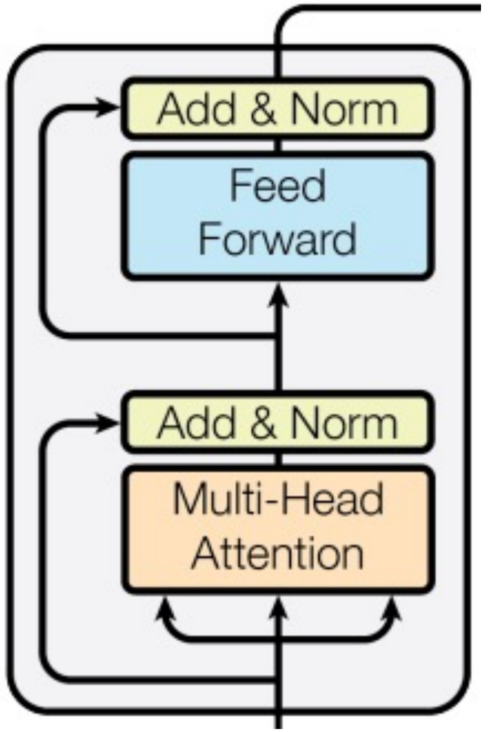
Scaled Dot-Product Attention



Transformer Blocks

Residual Connections

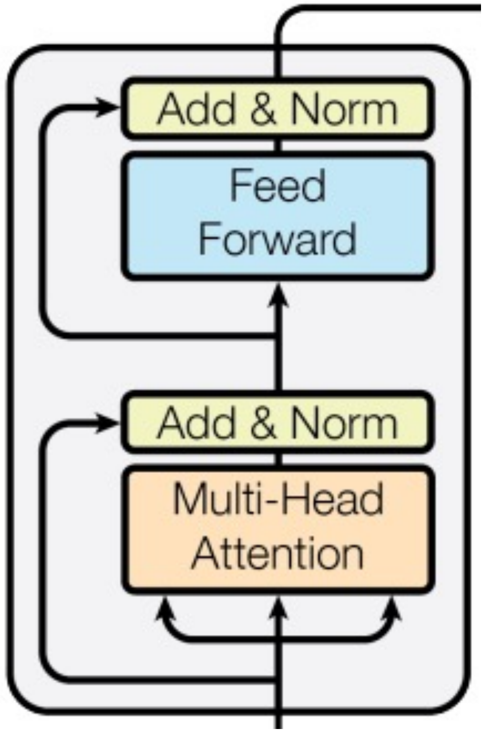
- *Connections that pass information from a lower to a higher layer*
- Allows information to skip a layer improves learning
- Gives higher level layers direct access to information from lower layers



He, K., X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. CVPR

Layer Normalization

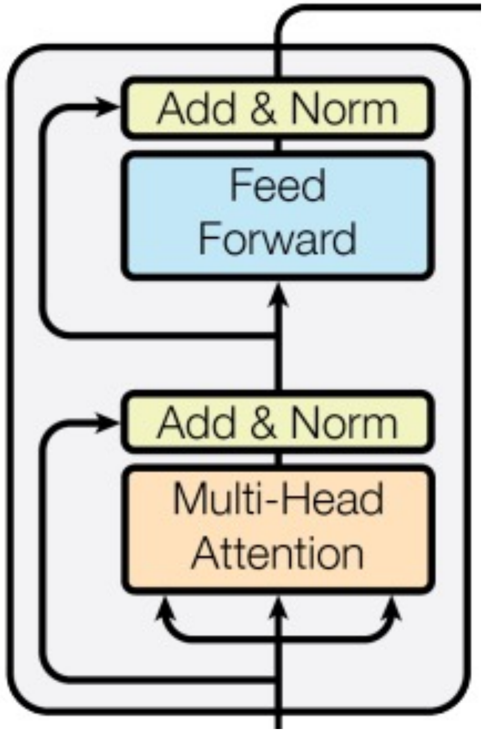
- Facilitates gradient-based training by keeping the values of hidden layer in a range
- $\hat{x} = \frac{(x-\mu)}{\sigma}$
- two learnable parameters : γ, β
- $\mathrm{LayerNorm} = \gamma\hat{x} + \beta$



Ba, J. L., J. R. Kiros, and G. E. Hinton. 2016. Layer normalization. NeurIPS workshop.

Transformer Block

- $z = \mathrm{LayerNorm}\big(x + \mathrm{SelfAttn}(x)\big)$
- $y = \mathrm{LayerNorm}(z + \mathrm{FFNN}(z))$



Ba, J. L., J. R. Kiros, and G. E. Hinton. 2016. Layer normalization. NeurIPS workshop.

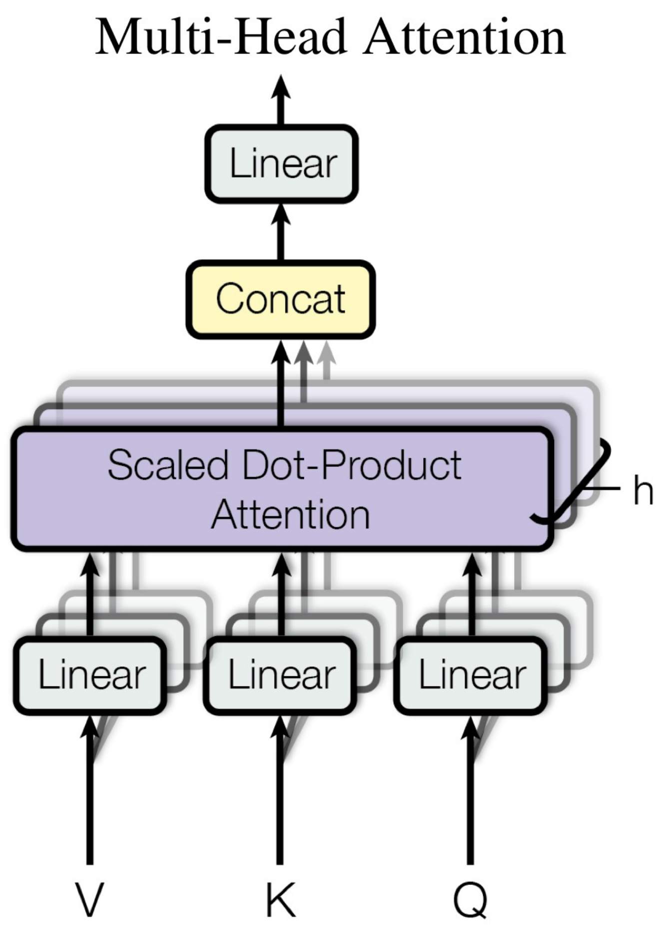
Multi-Head Attention

Multi-Head Attention

- Difficult for a single transformer block *to learn to capture all the different kinds of parallel relations among its inputs*
- Now we have multi head(self-attention) layers
- Each head can learn *different aspects of the relationships*

Multi-Head Attention

- Own set of Q, K, V matrices:
  $W_i^Q,\ W_i^K,\ W_i^V$
- $\mathrm{MultiHeadAttn}(X) =$
  $(\mathrm{head}_1 \oplus \mathrm{head}_2 \cdots \oplus \mathrm{head}_h)W^O$
- $Q = XW_i^Q; K = XW_i^K; V = XW_i^V$
- $\mathrm{head}_i = \mathrm{SelfAttn}(Q, K, V)$



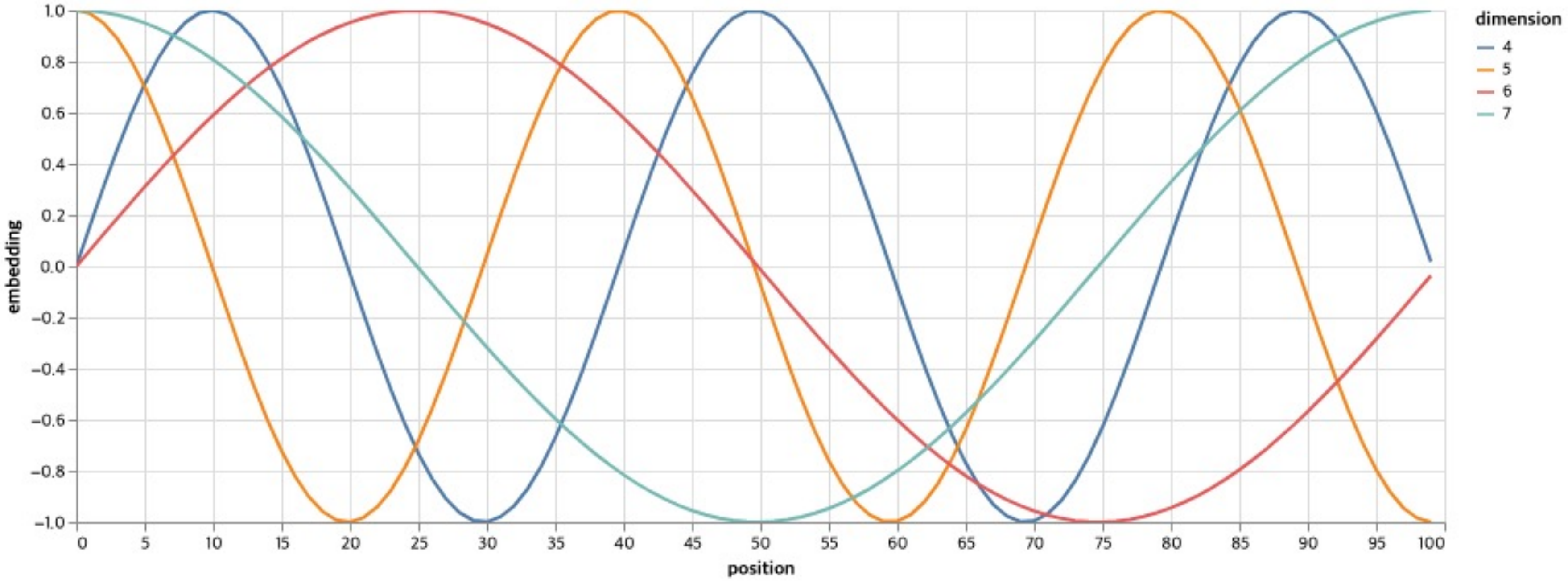
Positional Encoding

Positional Encoding

- Motivation : Transformer model doesn't have any notion of *relative, or absolute, positions of the input*
- Solution : modify the input embeddings by *combining them with positional embeddings*

Positional Encoding

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



http://nlp.seas.harvard.edu/annotated-transformer/#position-wise-feed-forward-networks

Q&A