TRANSFORMER

中央研究院 古倫維

關於我

- ✓中央研究院資訊科學所-研究員 since 2012
- ✓國立陽明交通大學合聘教授
- ✓ACL 2024 Program Chair 技術主席
- ✓自然語言處理研究 since 1998

@lunweiku 或搜尋 古倫維

古老師研究室

@LUNWEIKU

OUTLINE

MAY. 10: WORD EMBEDDINGS, FROM CNN TO TRANSFORMER, TRANSFORMER FRAMEWORK
MAY. 17: TRANSFORMER ENCODER MODELS, LAB 1
MAY. 24: TRANSFORMER DECODER MODELS, LAB 2

LAB TA: YILI AND LUANA

Word Features

Why word embeddings?

BOW – Bag of words

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

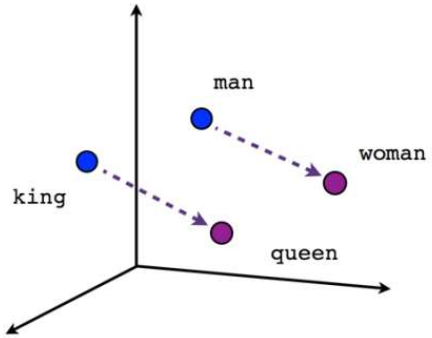
Issues

- ✖ Memory Explosion
  - ✖ The vocabulary of a speaker: 20,000 words
  - ✖ Old bag of words -- Unigram: 20,000, Bigram (first-order): 400 million, Trigram (second-order): 8 trillion
- ✖ Semantic Computation

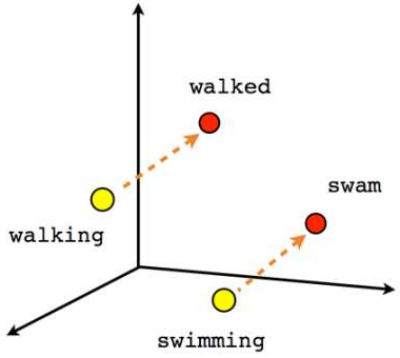
Word Embeddings (1)

Word embedding與過去使用的詞向量差異點：

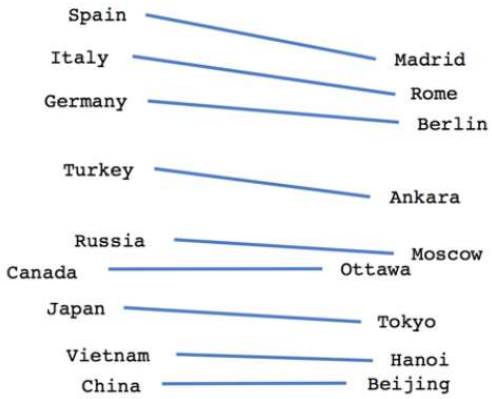
可以做語意運算: king + woman – man = queen



Male-Female



Verb tense



Country-Capital

Word embeddings for texts

- Word: king

- Sentence: I am a king of this small world.

- Document:

Biden Warns Putin of Economic Consequences if Aggression Continues

In a tense, two-hour virtual summit, President Biden also warned that an invasion of Ukraine would result in a repositioning of NATO troops in Europe.

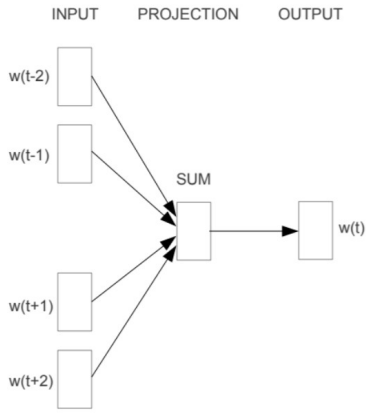
But it is too early to tell if the call will ease tensions. President Vladimir Putin gave no indication of his ultimate intent, officials said.

Word Embeddings (2)

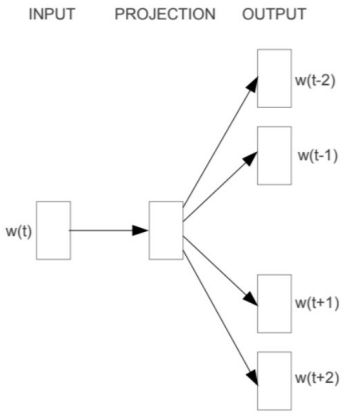
Pre-trained or train by yourself!

One-hot vector

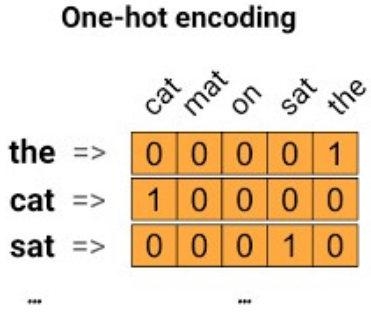
- w2v (Tomas Mikolov, 2013)
- Glove
- Fasttext (char level w2v)
- Bert Pretrained



CBOW



Skip-gram



You can find various of embeddings on the Web and then use ML models.

WORD EMBEDDINGS(3)

- Counting-Based: Glove

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

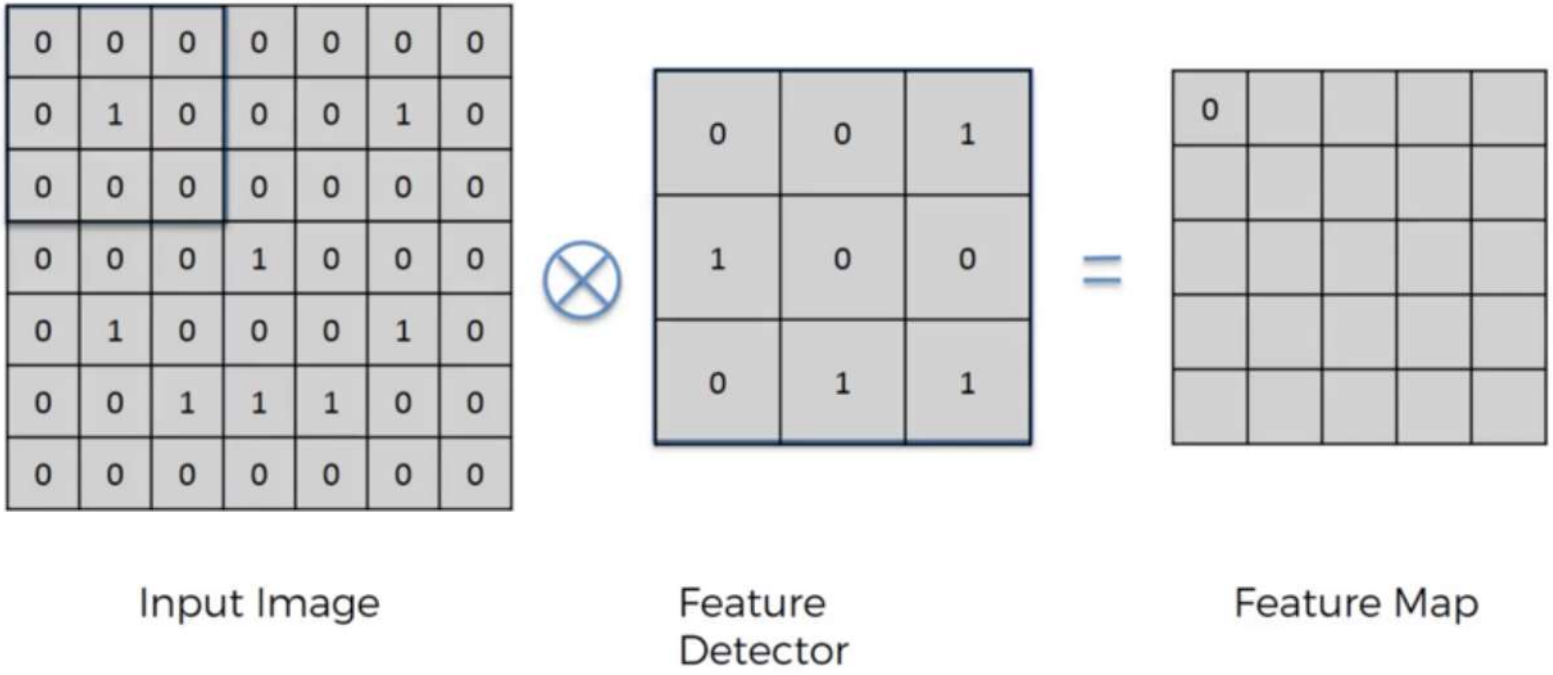
- Prediction-Based: CBOW, Skip-gram, Bert
- Word analogy task: A is to A’ as B is to B’ (spearman’s rank correlation coefficient) is usually used to evaluate word embeddings

CBOW		
Sample	Input	Output
1	[love]	I
2	[I, natural]	love
3	[love, language]	natural
4	[natural, processing]	language
5	[language]	processing

Skip-gram		
Sample	Input	Output
1	I	[love]
2	love	[I, natural]
3	natural	[love, language]
4	language	[natural, processing]
5	processing	[language]

From CNN to Transformer

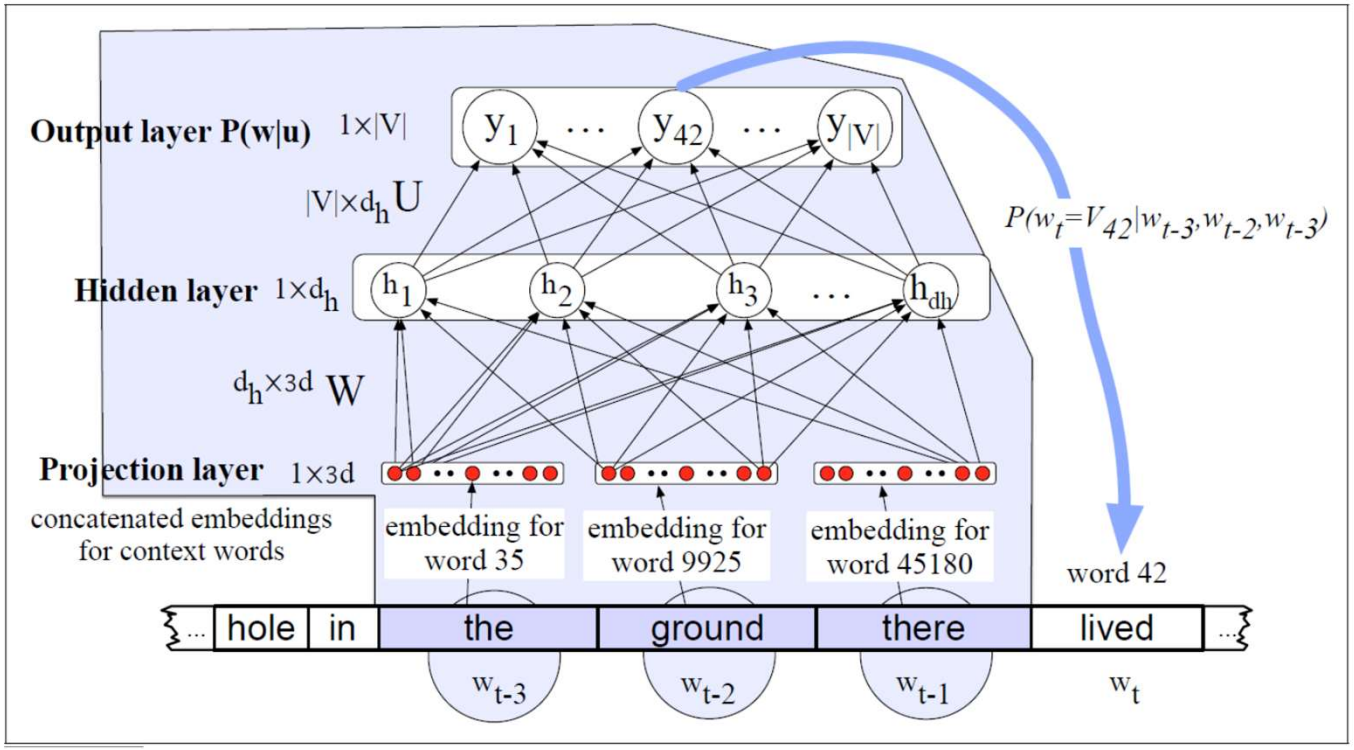
CNN Filter as n-gram



Basic Models

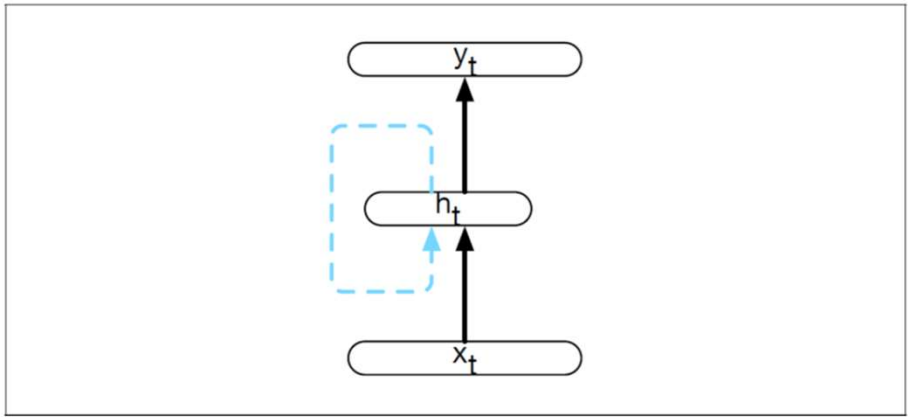
RNN / (Bi)LSTM & NLP: Language is a temporal phenomenon

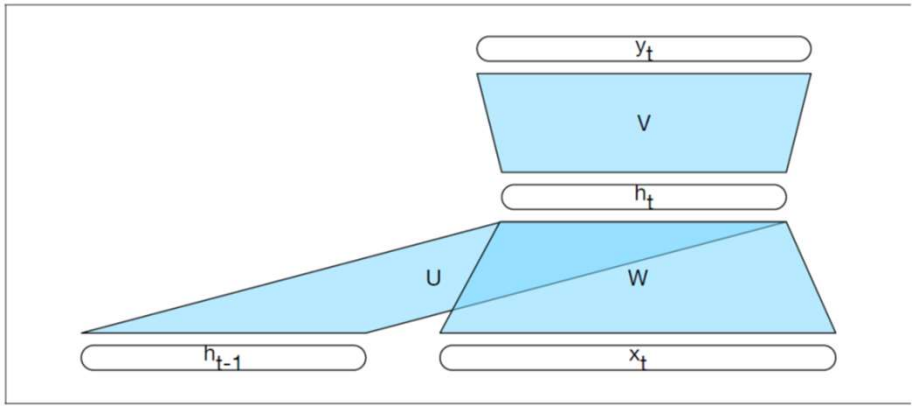
A simple feedforward NN: ctx len=3



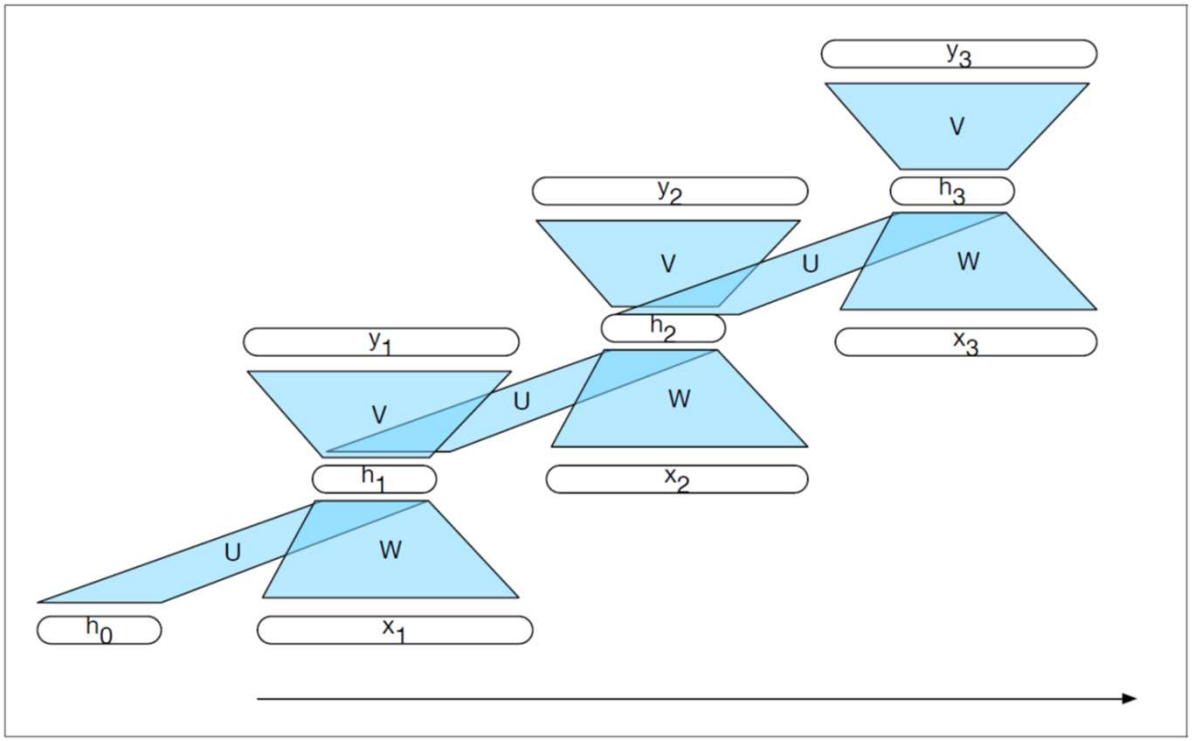
RNN

Recurrent connection in hidden layer: the activation value of the hidden layer depends on current input and the activation value of the hidden layer from the previous time step.





RNN Unrolled



RNN LM

Teacher forcing – Input ground truth at time *t-1.* (<-> autoregressive)

Teacher forcing is important in transformer.

Exposure bias.

Scheduled sampling to mitigate exposure bias.

Use cross entropy as loss function: measure the difference between predicted probability distribution and the correct distribution

$$L_{CE} = -\sum_{w \in V} y_w^t \log \hat{y}_w^t$$

$$L_{CE}(\hat{y}^t, y^t) = -\log \hat{y}_{w_{t+1}}^t$$

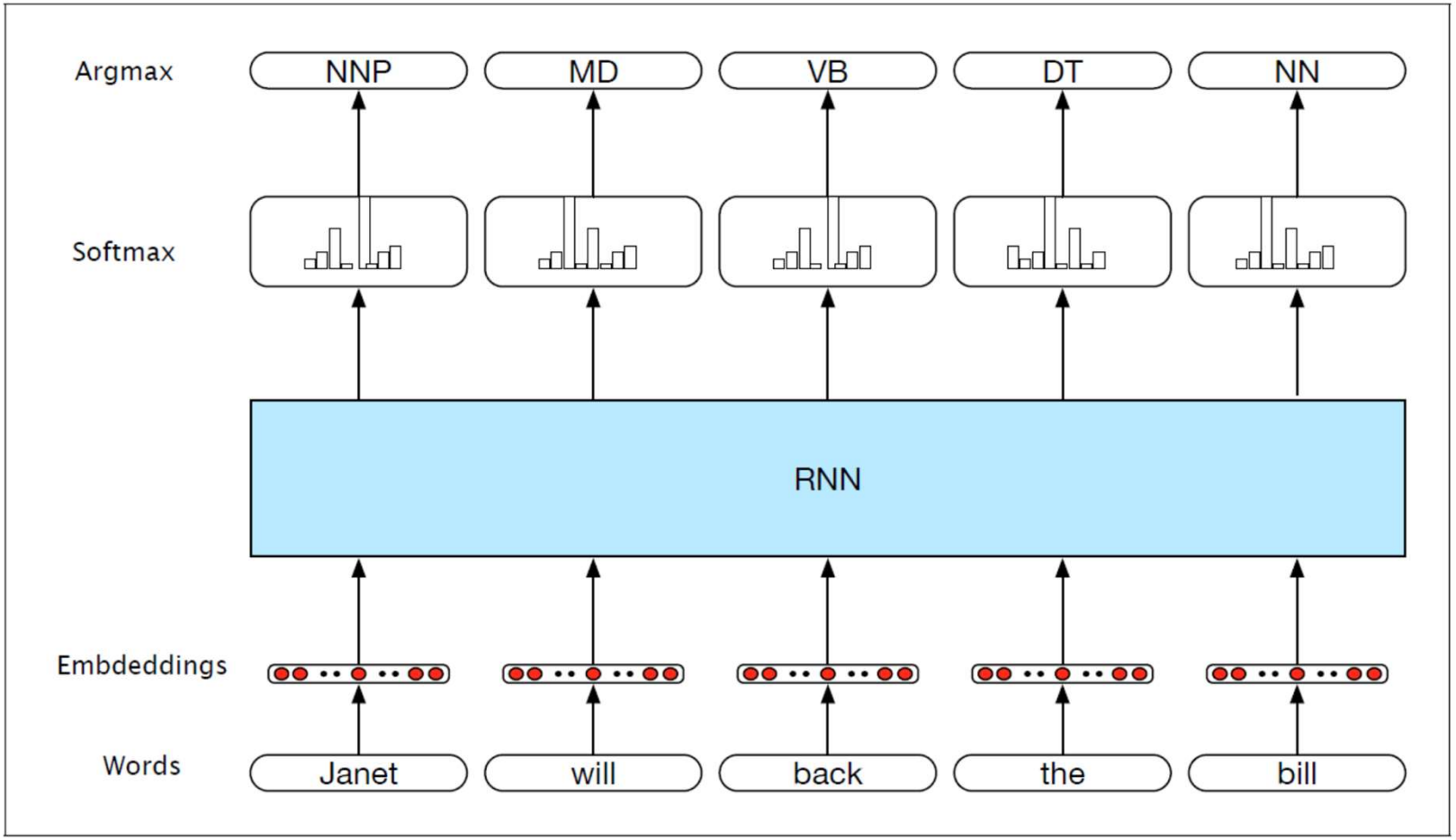
NN Models

- Sequence generation
- Sequence labeling
- Sequence classification

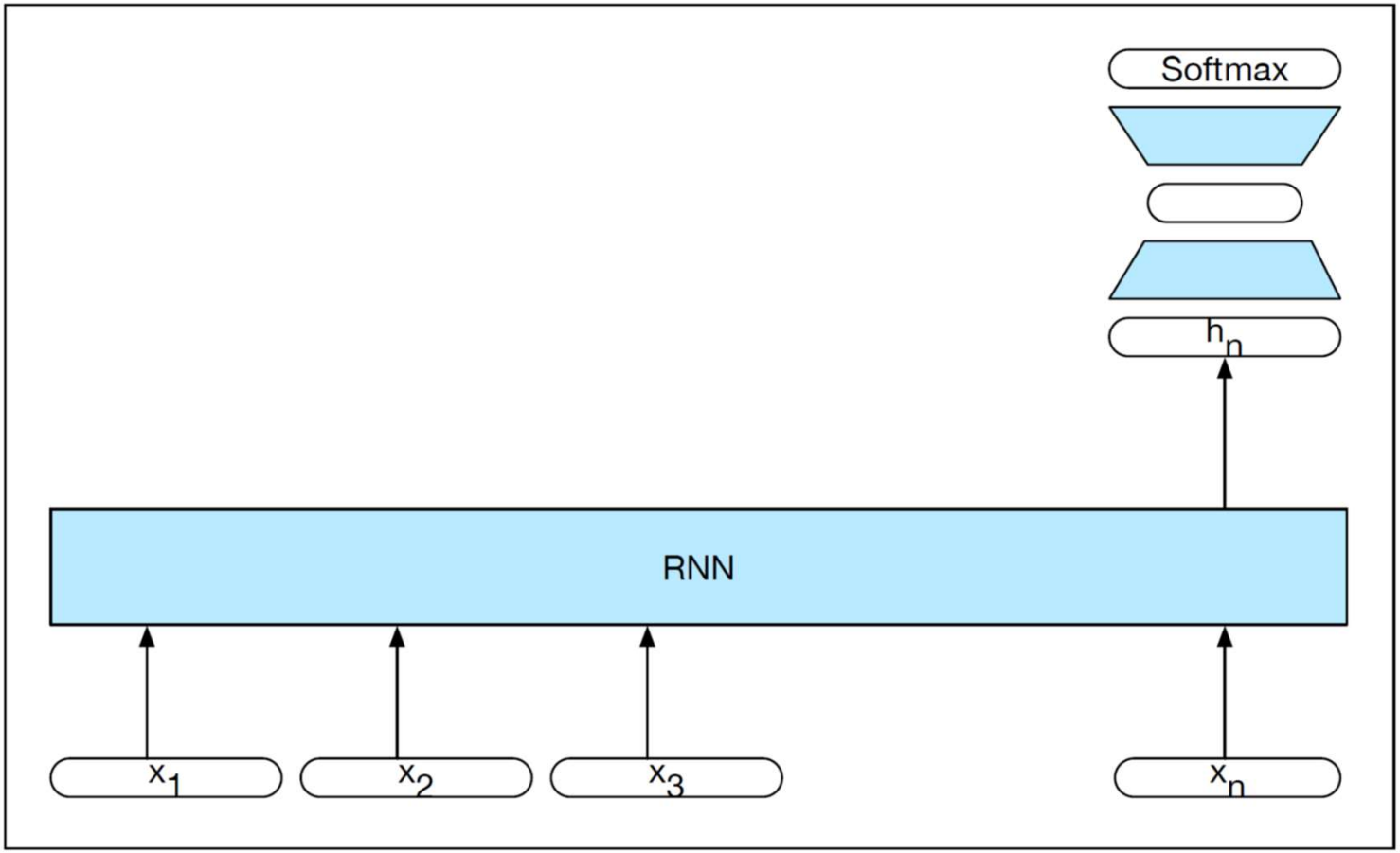
RNN for sequence generation

- Starting from <s> as input
- Sample the next word
- Stop until </s> or the limited length is reached

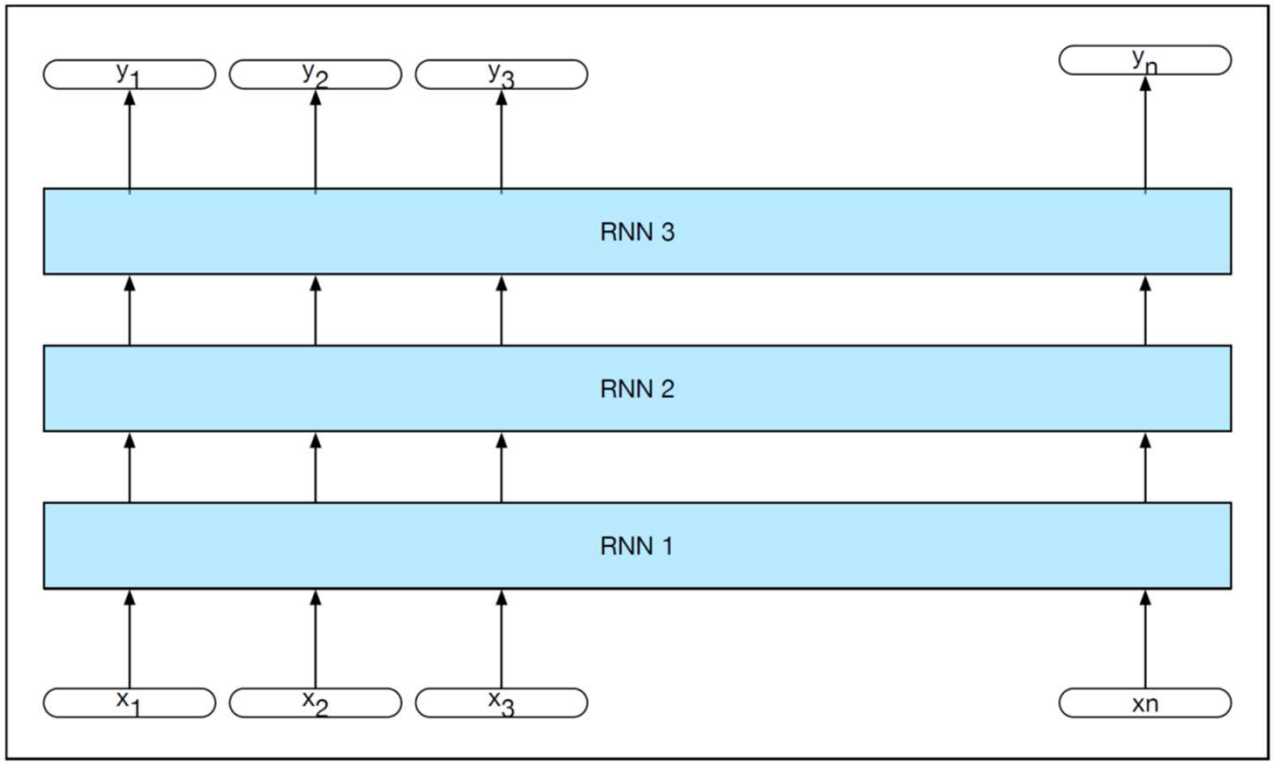
RNN for sequence labeling



RNN for sequence classification



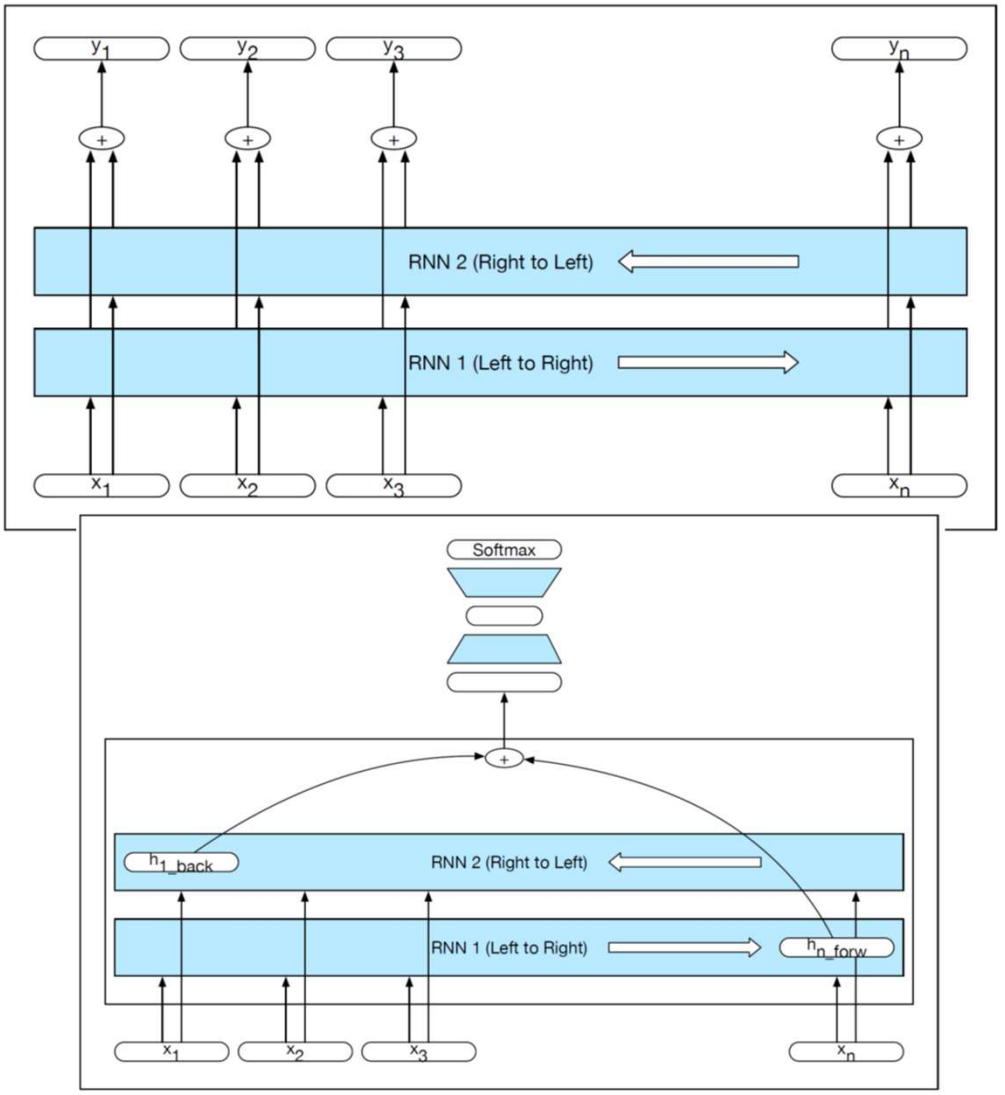
Stacked RNN



Bi-directional RNN

Hidden layer rather local

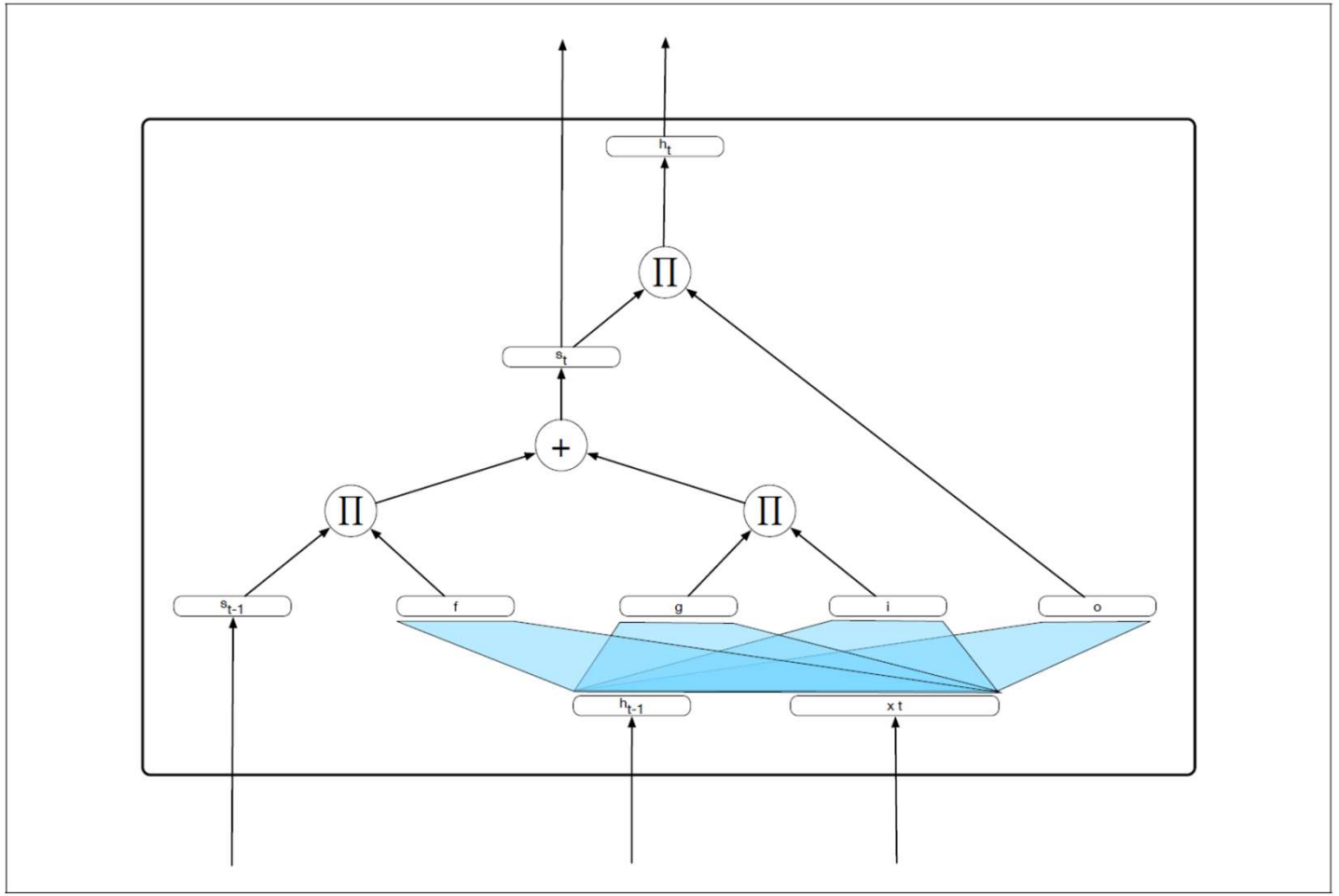
Gradient vanishment



Long short-term memory (LSTM)

Removing information no longer needed & adding information likely to be needed for later decision making

Forget gate, add get, output gate



Gated recurrent units

- Stop using a separate context vector
- Reset gate r, update gate z
- Reduce 8 U/W parameters (for 4 gates) to 4 (for 2 gates)

$$\tilde{h}_t = tanh(U(r_t \odot h_{t-1}) + Wx_t)$$

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t$$

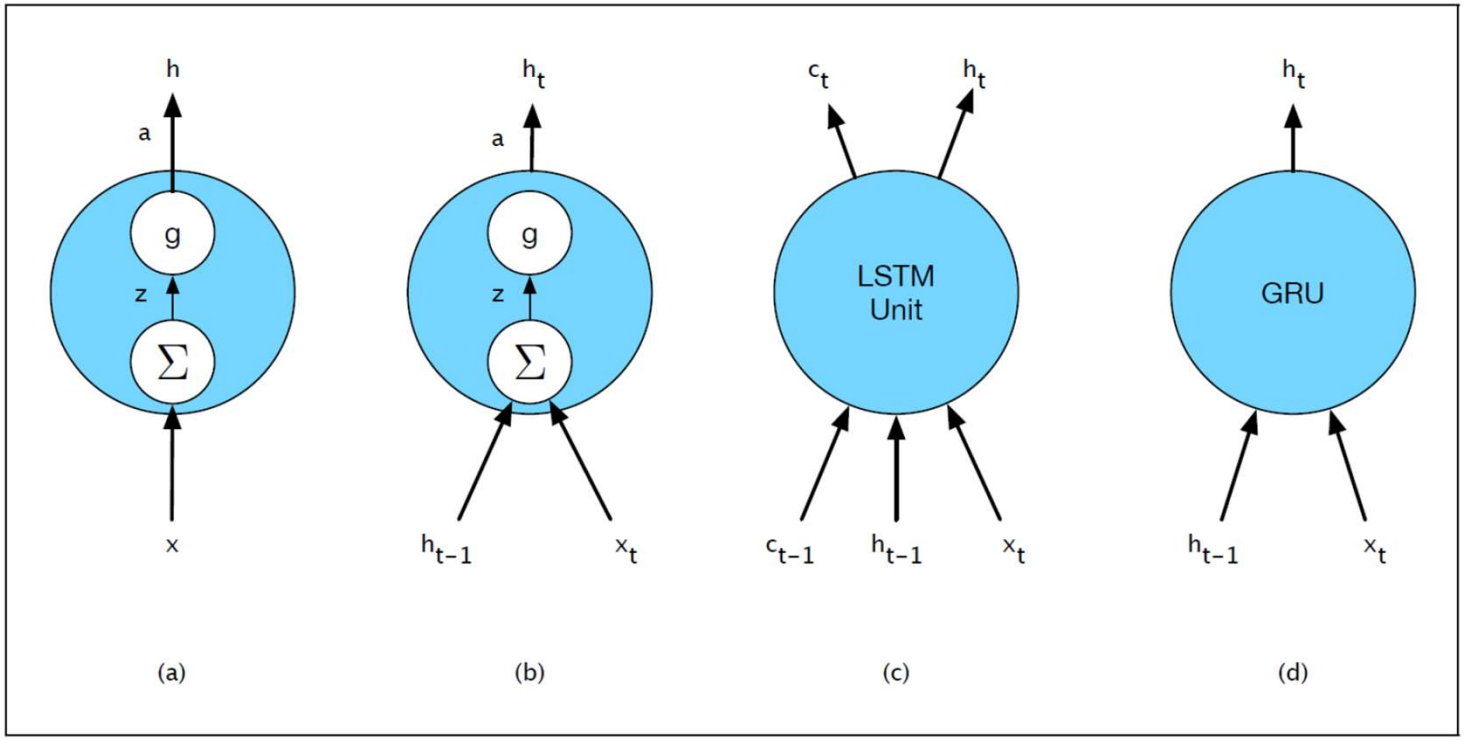
Basic Neural Units

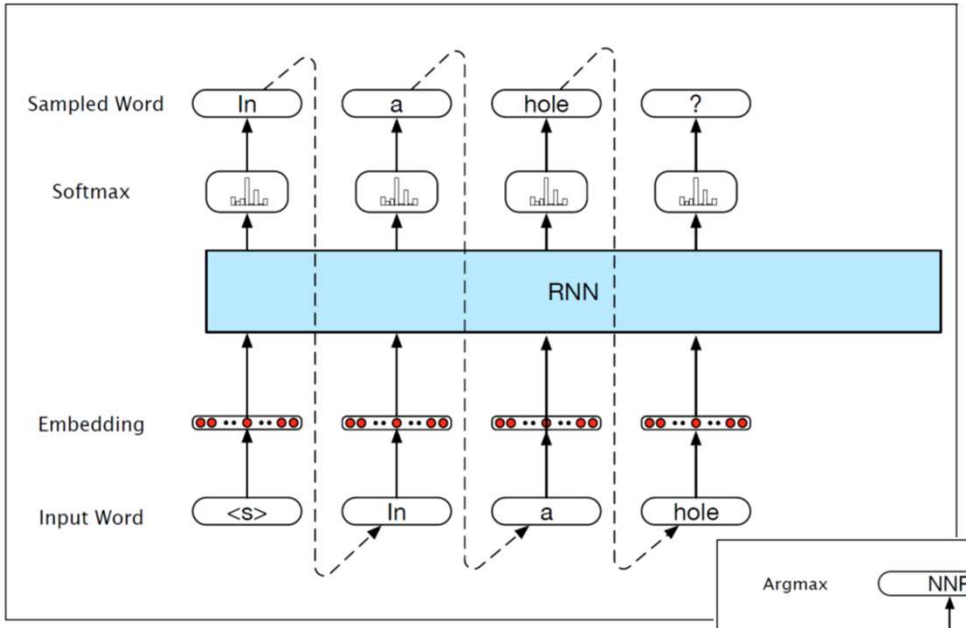
(a)Feedfoward

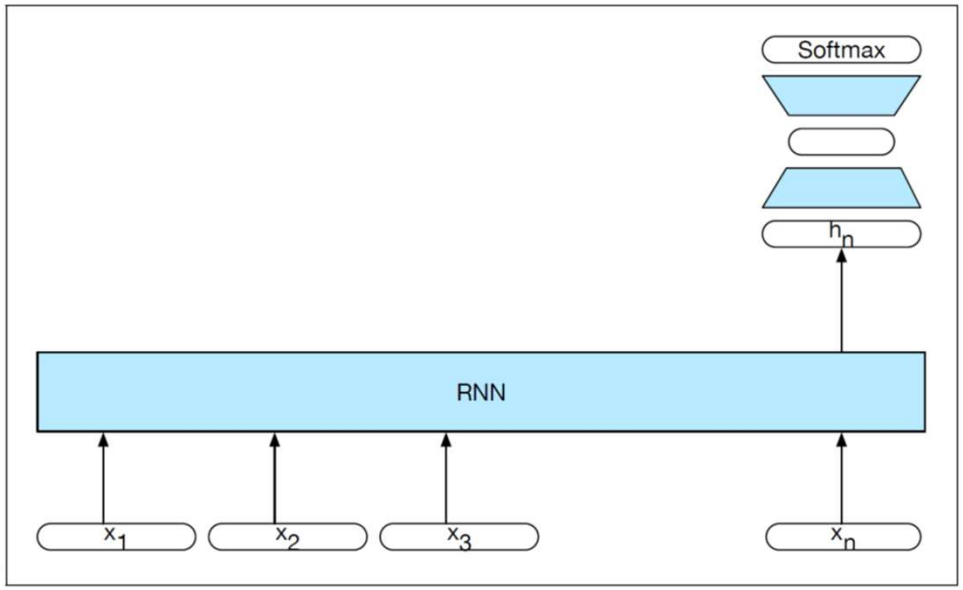
(b)Simple recurrent network

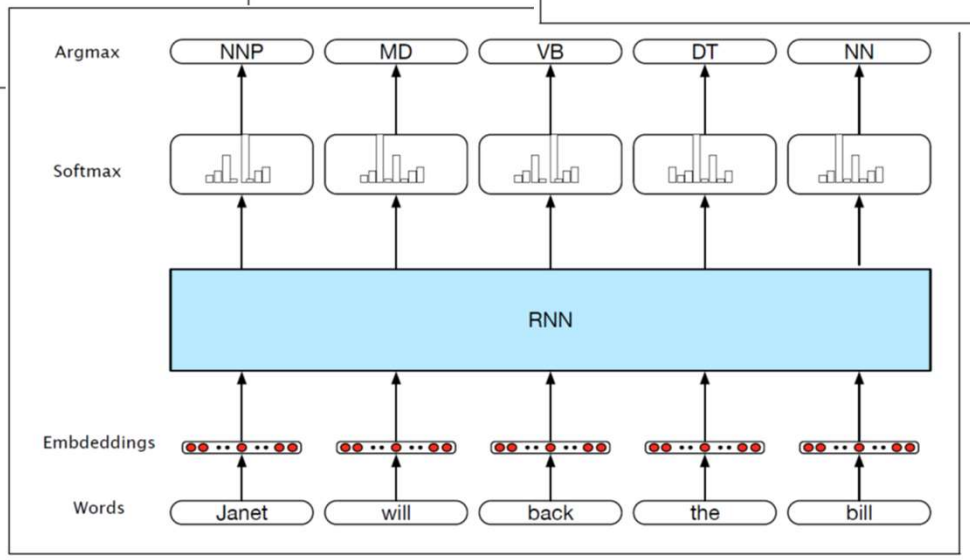
(c)LSTM

(d)GRU









↑Autoregression

→ Sequential Labeling

↗ Sequence Classification

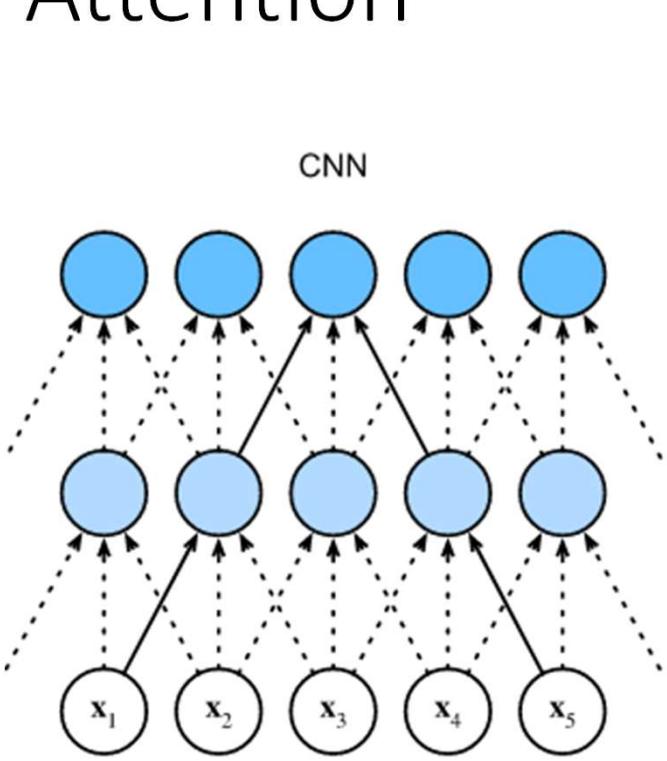
Transformer:
Attention is All You Need

Self-Attention, Multi-head attention, Positional encoding

Self-attention network: Transformer

- LSTM: mitigate the loss of distant information due to the recurrence in RNNs
- Issue: loss of relevant information and difficulties in training
- Transformers: from sequential network back to **fully connected network**
- Innovation: the use of self-attention layers

Self-Attention



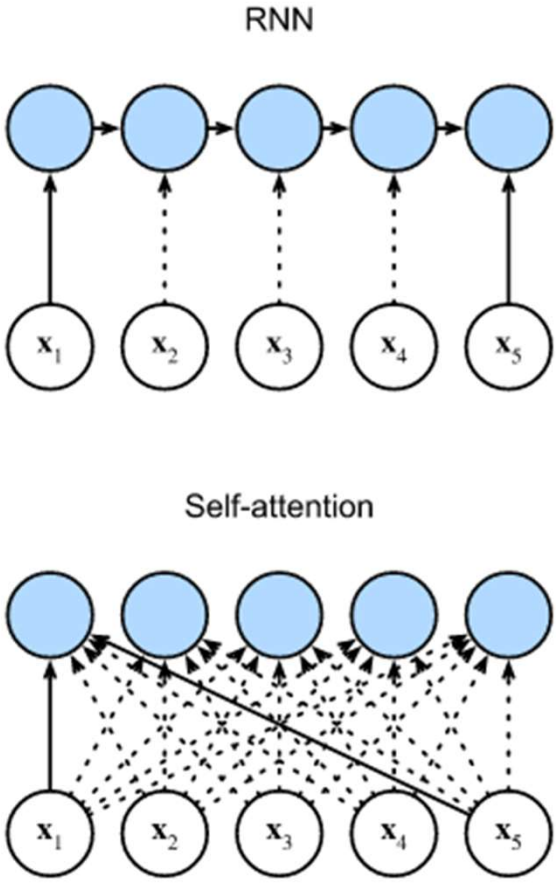
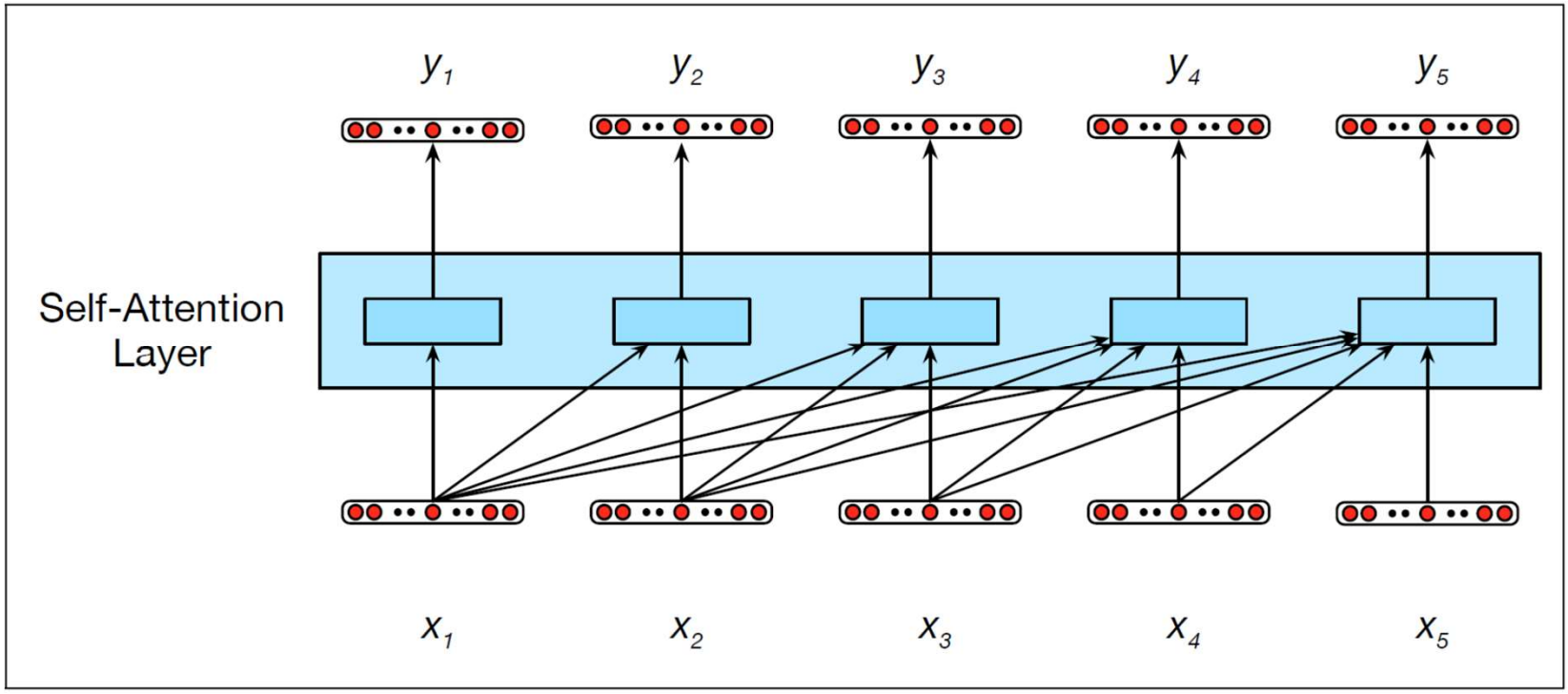


Figure from gluon.ai

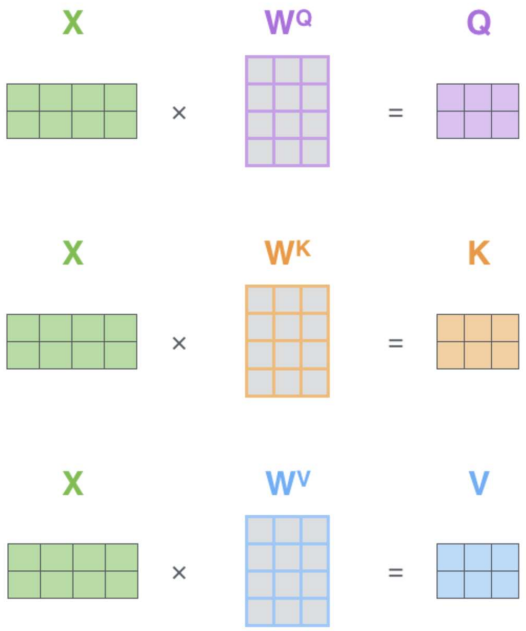
transformer

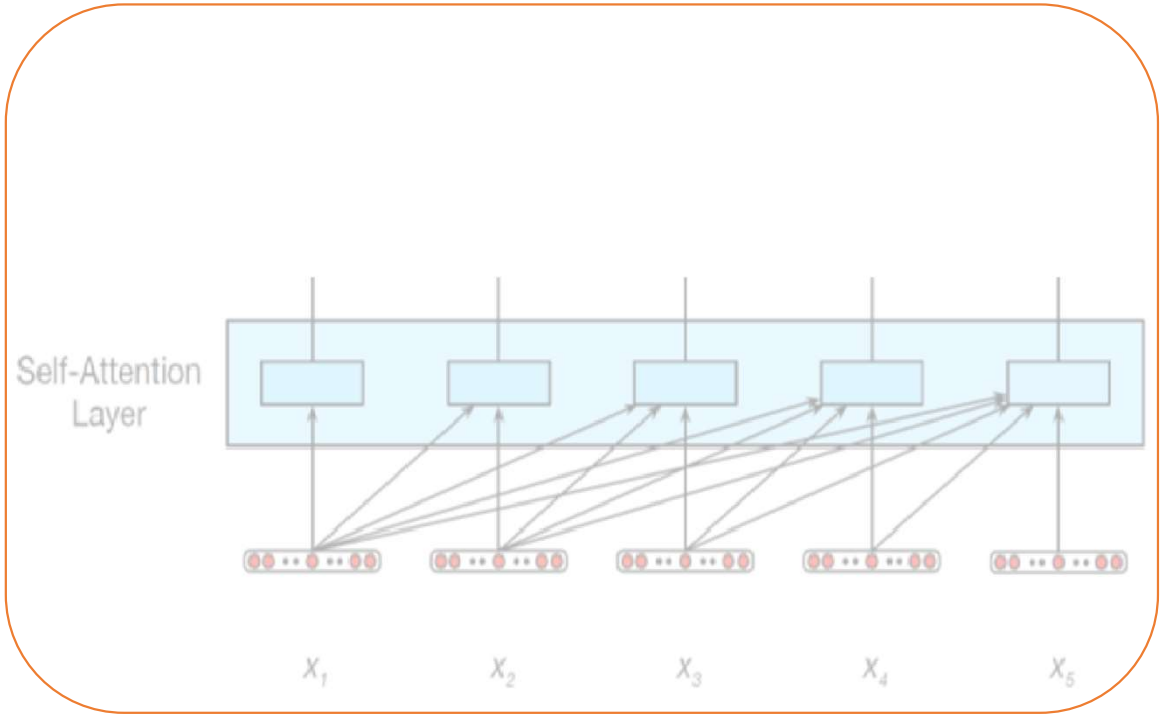
Information flow

Map x to the same length y

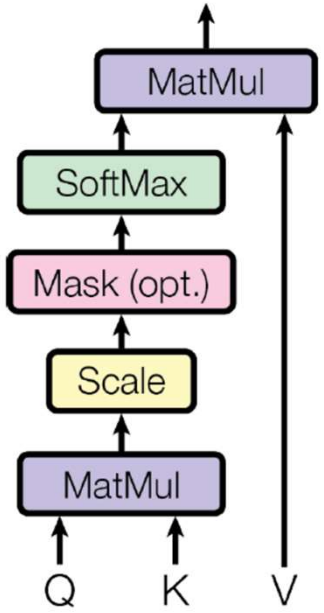


Self attention layer
Query, Key, Value in self attention

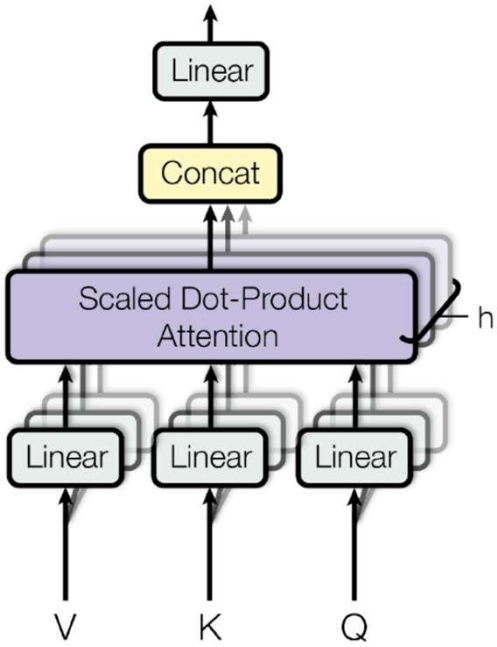




Scaled Dot-Product Attention



Multi-Head Attention



Transformer

Query, Key, Value (different role of embeddings)

Rewrite formula

$$q_i = W^Q x_i; \;\; k_i = W^K x_i; \;\; v_i = W^V x_i$$

$$score(x_i, x_j) \;=\; q_i \cdot k_j$$

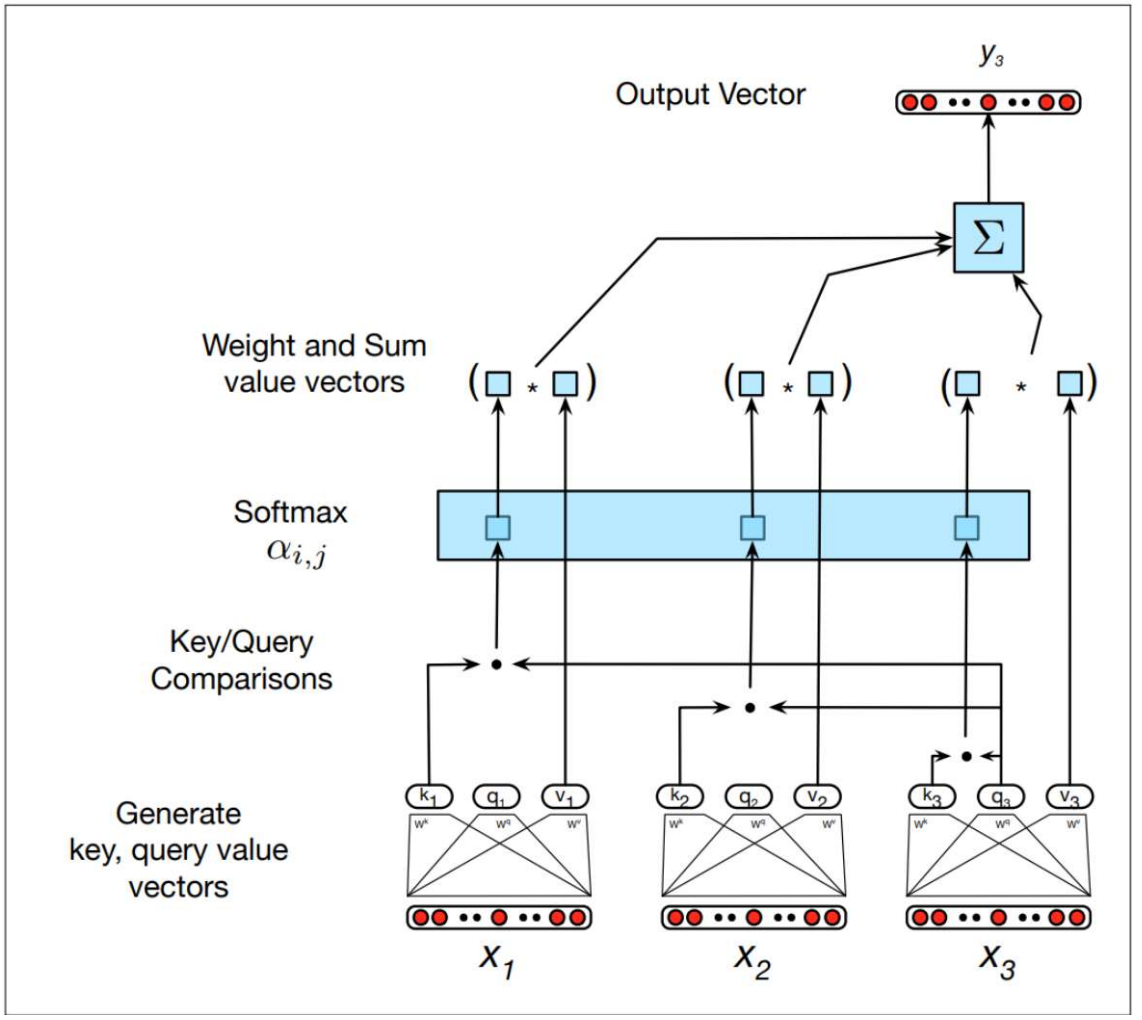
$$y_i \;=\; \sum_{j \leq i} \alpha_{ij} v_j$$

Transformer

Calculate Key, Query, Value

Dot product generates large value for softmax exponential operation ->

$$score(x_i, x_j) = \frac{q_i \cdot k_j}{\sqrt{d_k}}$$



Self-attention

Comparisons

Current input to itself and the preceding contexts

$$score(x_i, x_j) = x_i \cdot x_j$$

$$\begin{aligned} \alpha_{ij} &= \text{softmax}(score(x_i, x_j)) \ \forall j \leq i \\ &= \frac{\exp(score(x_i, x_j))}{\sum_{k=1}^{i} \exp(score(x_i, x_k))} \ \forall j \leq i \end{aligned}$$

$$y_i = \sum_{j \leq i} \alpha_{ij} x_j$$

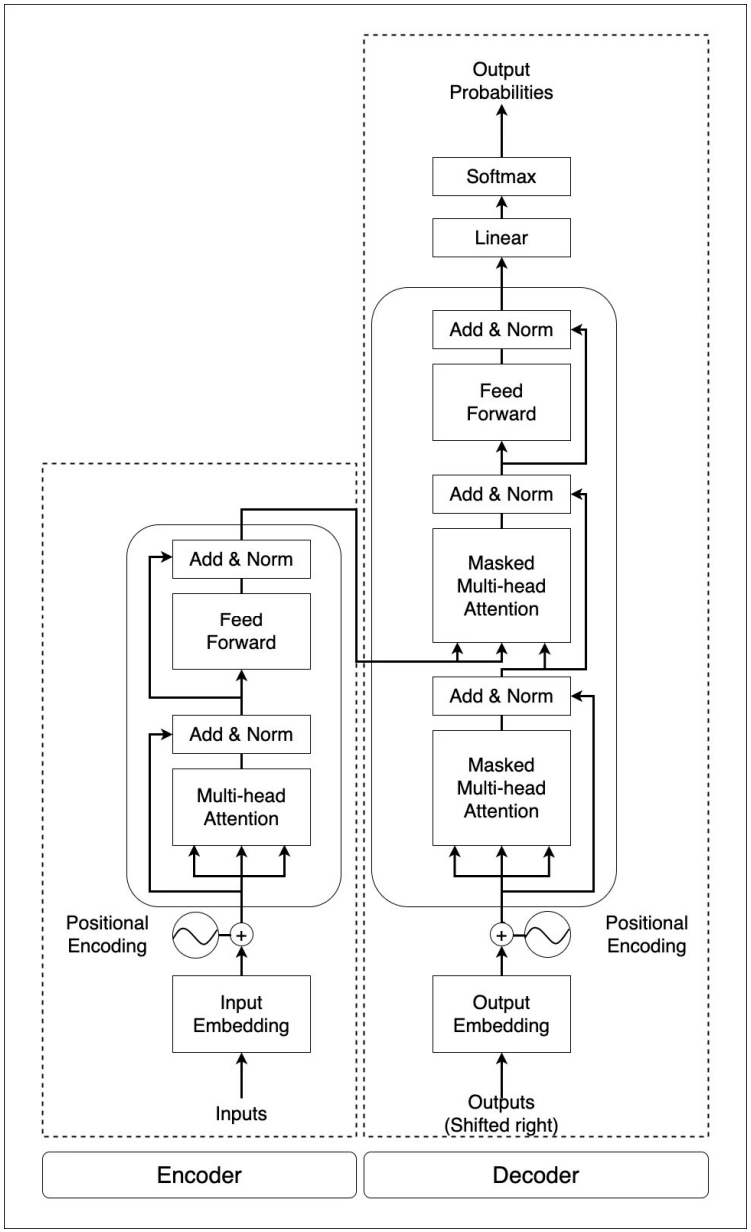
Transformer

$$Q = W^Q X;\ \ K = W^K X;\ \ V = W^V X$$

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

Transformer

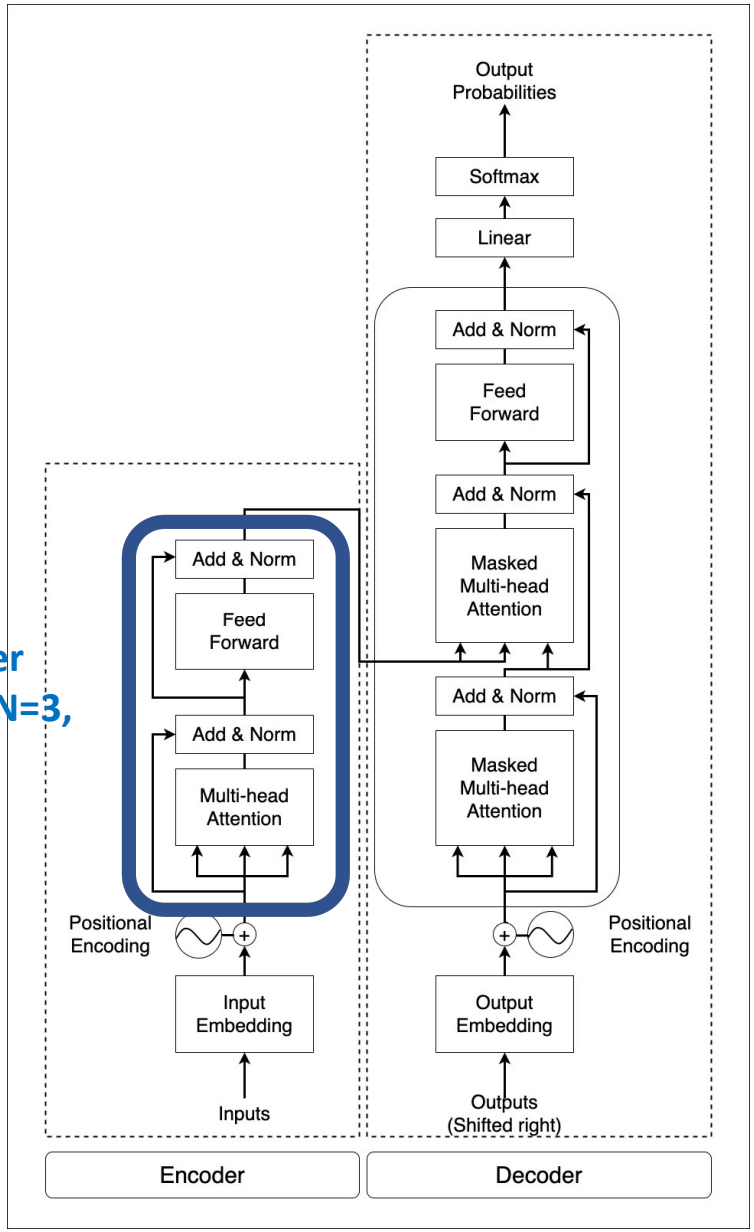
Framework



Transformer

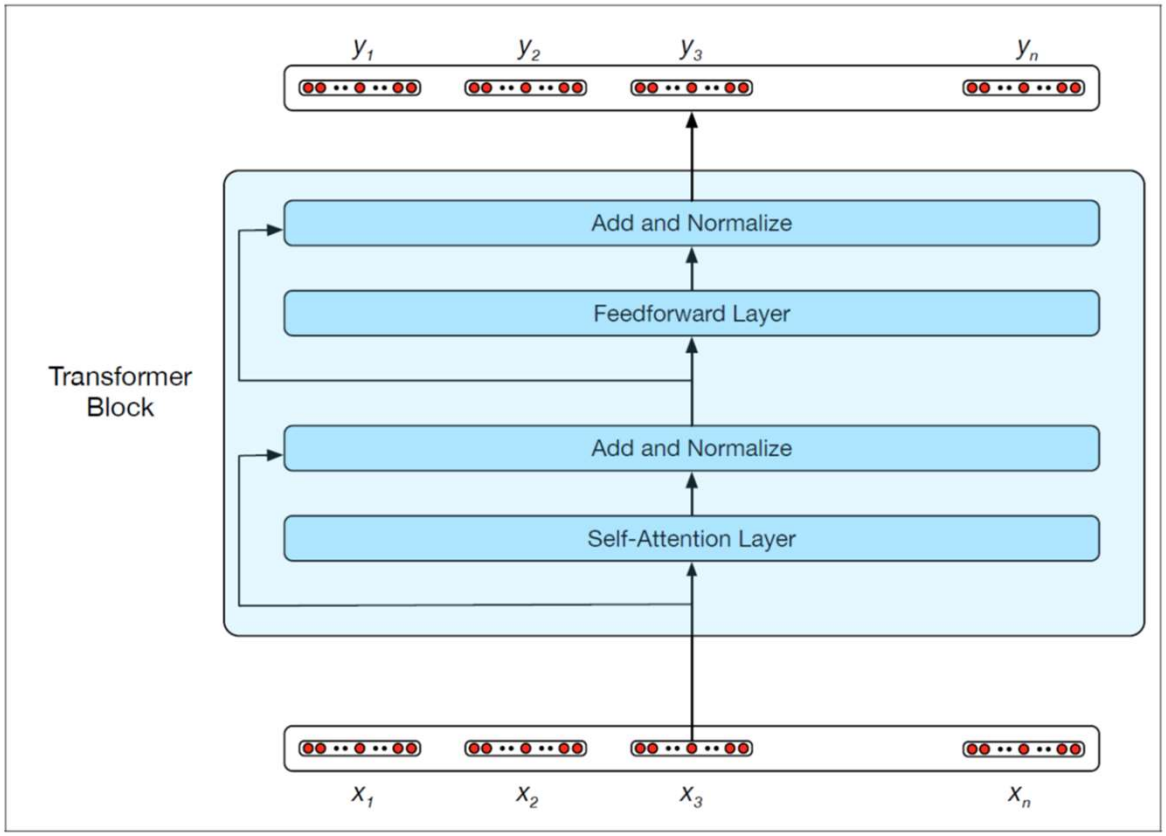
Framework

Transformer Block x N, N=3, 6, or more



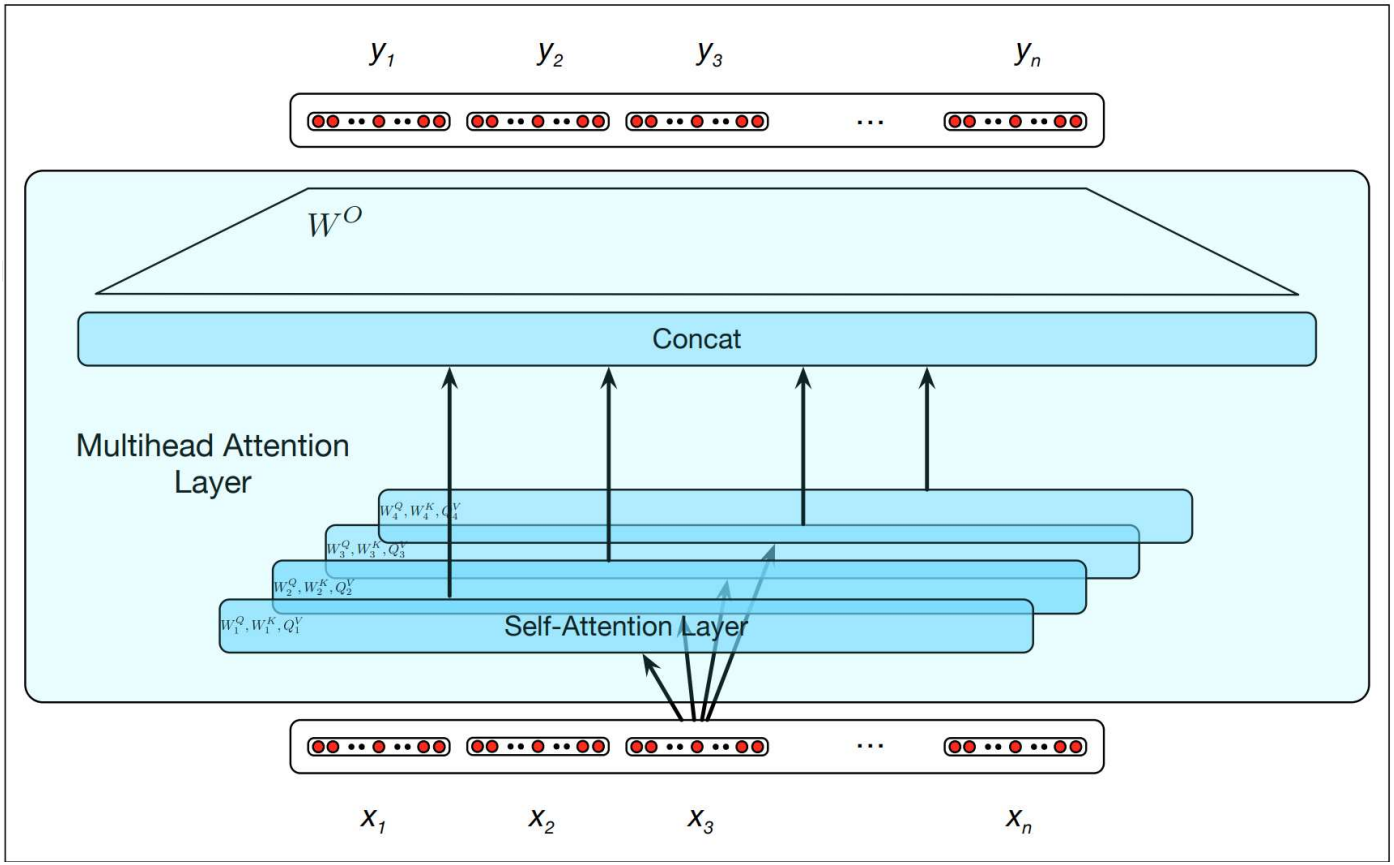
Transformer block

Elements in Upper-triangular portion of the comparison matrix are zeroed out (set to -∞)



Multi-head self-attention

Different words relate to each othe in many different ways, e.g., syntactic, semantic, discourse relations…

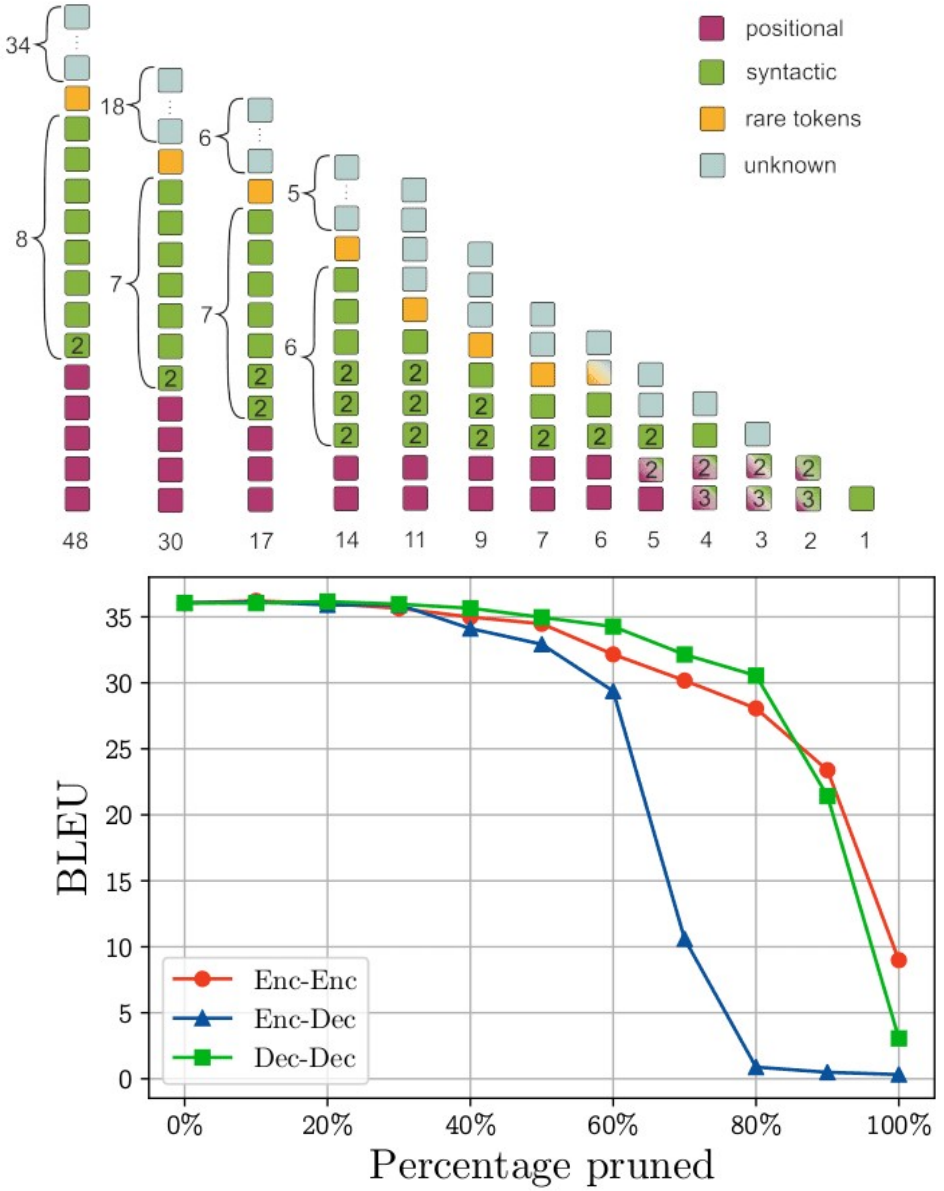


$$MultiHeadAttn(Q,K,V) = W^O(head_1 \oplus head_2 ... \oplus head_h)$$
$$head_i = SelfAttention(W_i^Q X, W_i^K X, W_i^V X)$$

Multi-head? How many?

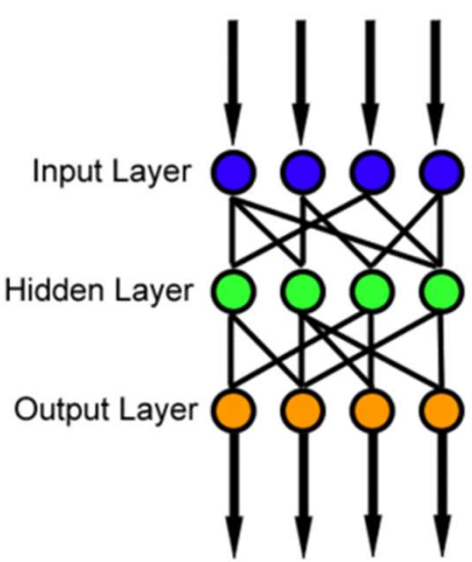
Voita et al., “Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned”.

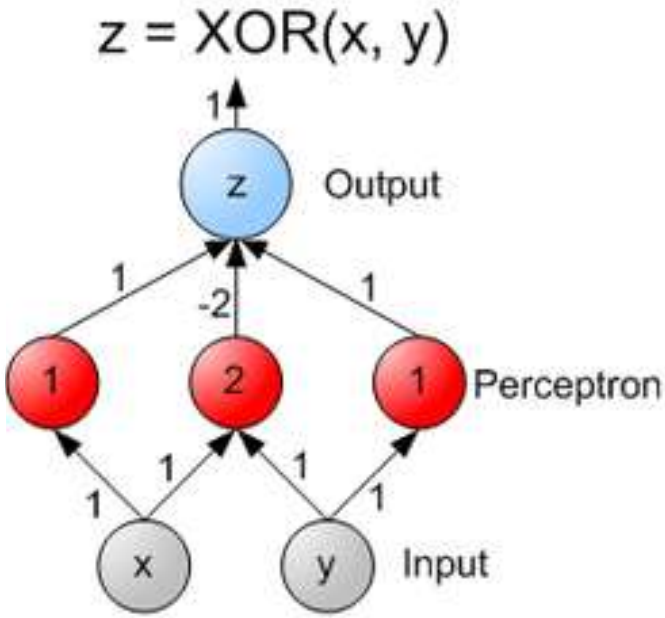
1. Positional heads
2. Syntactic heads
3. Heads pointing to rare words



Feed Forward Network

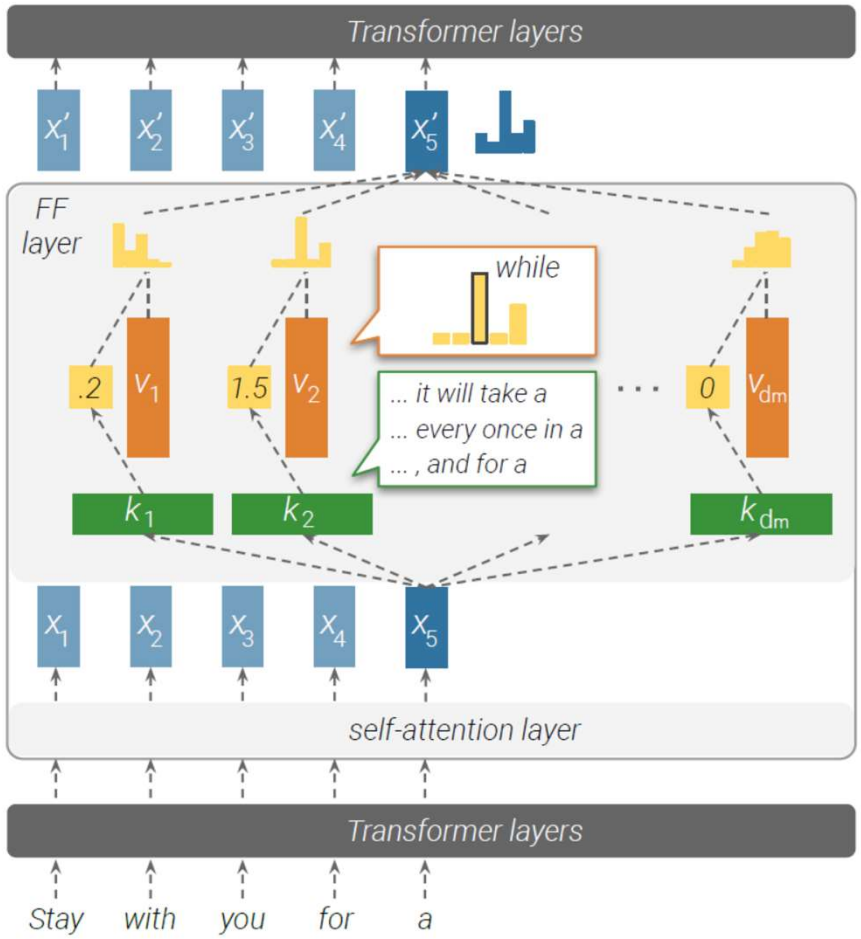
- From dimension *d* to 4*d* and back to *d* by ReLu
- Project to higher dimension and back to the original dimension
- Better distill important information





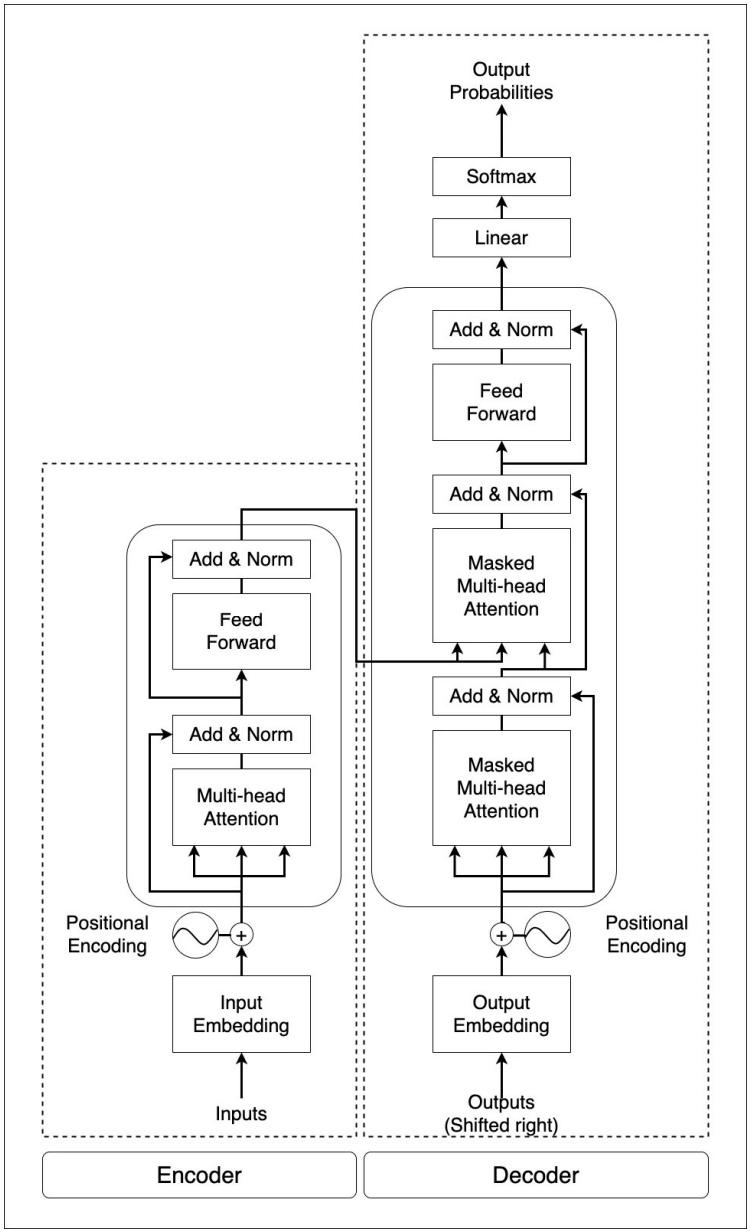
Feed Forward Network

- Transformer Feed-Forward Layers Are Key-Value Memories (Mor et al., 2021)



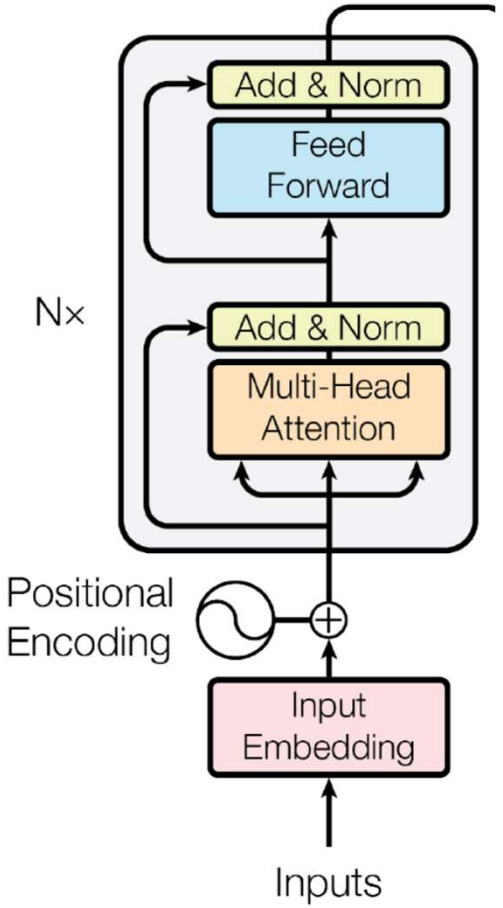
Transformer

Framework



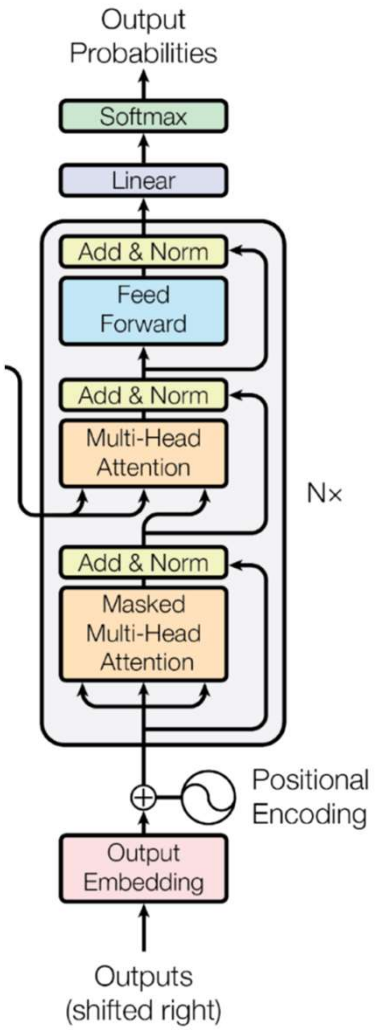
Transformer

Encoder

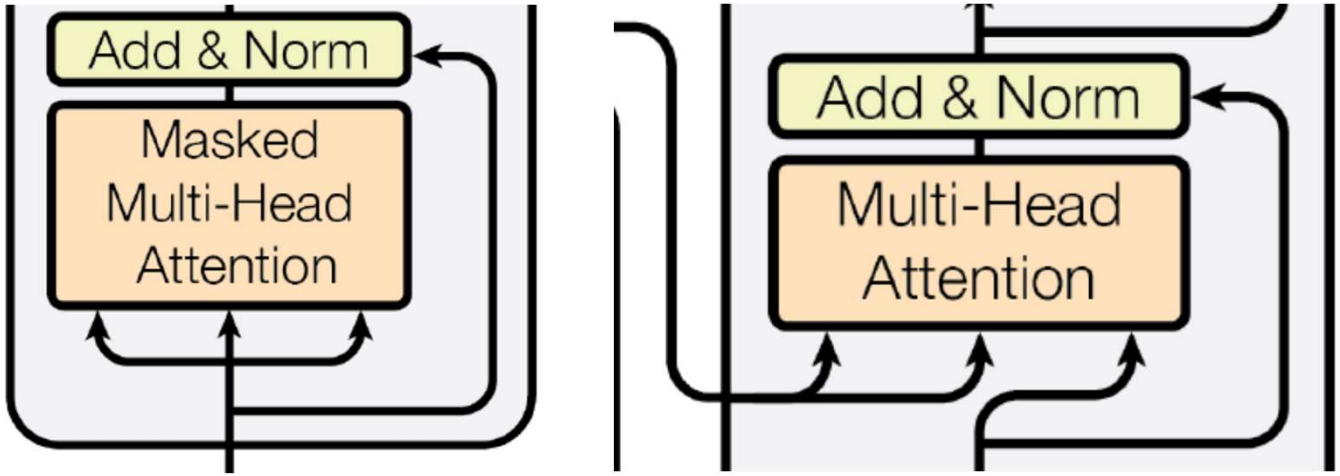


Transformer

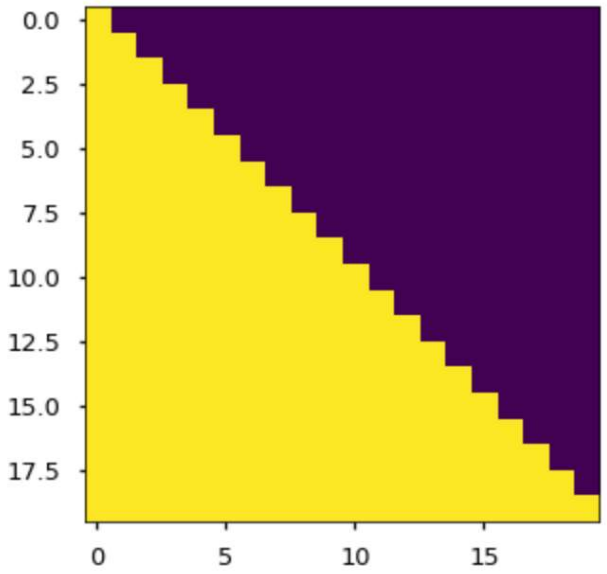
Decoder

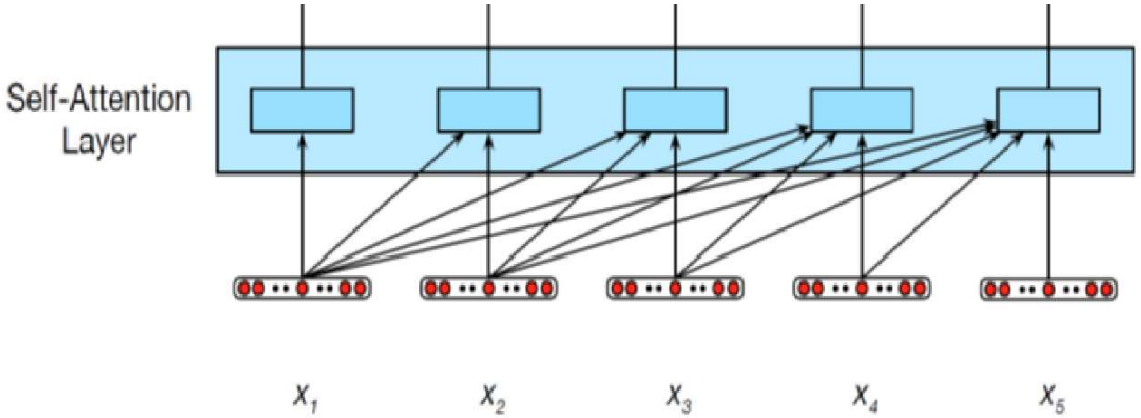


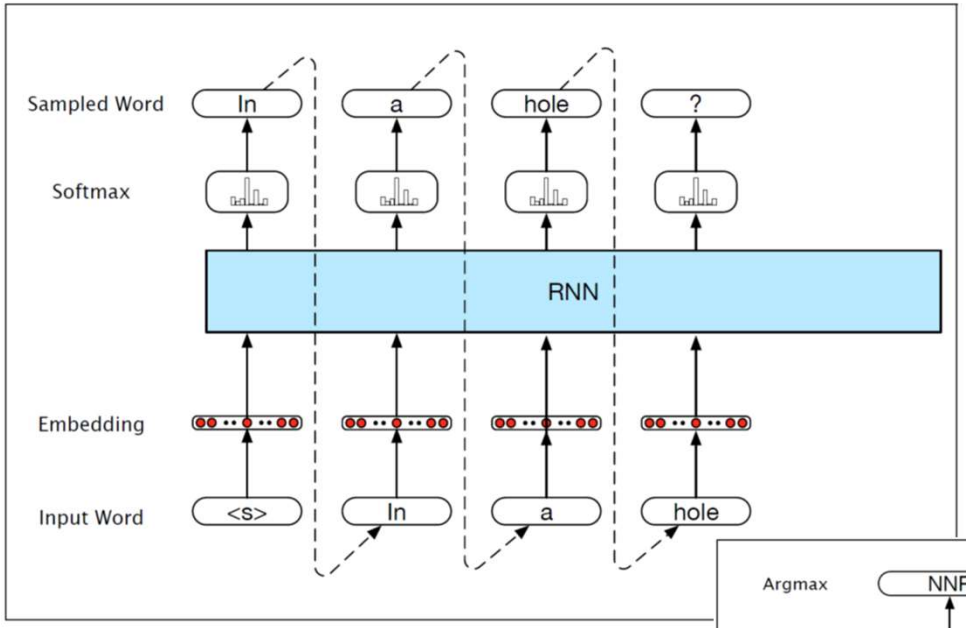
Transformer

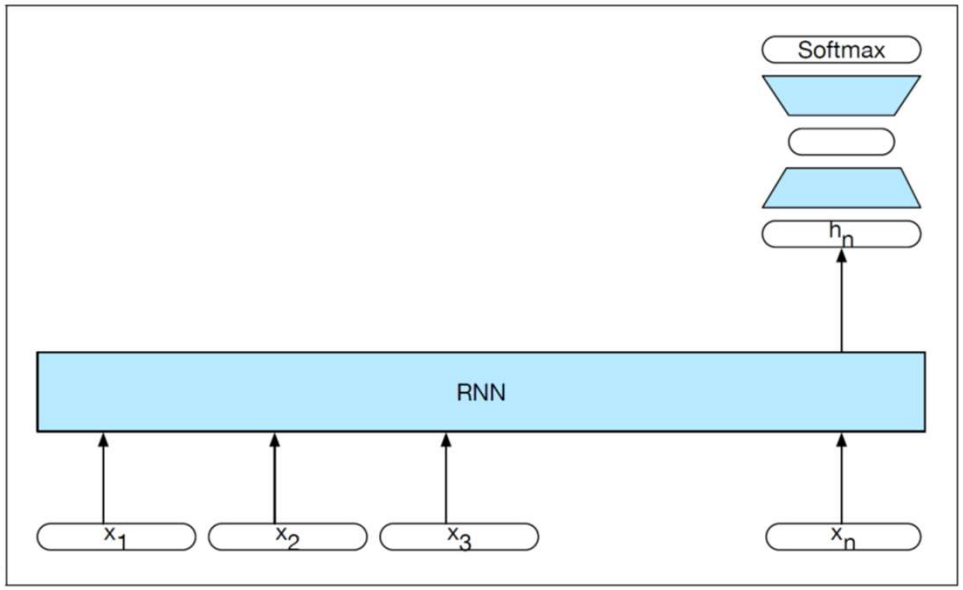


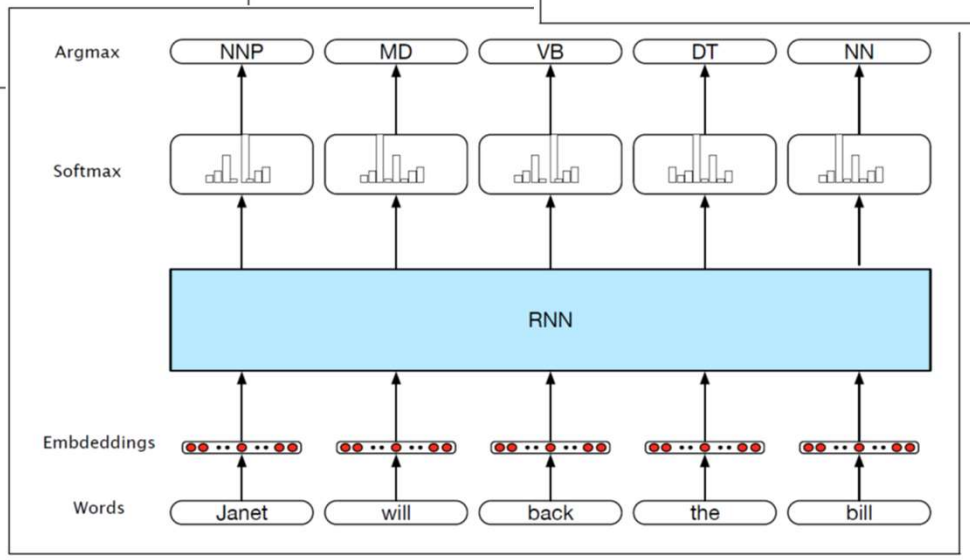
Masked


0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0
5
10
15


Self-Attention
Layer
x_1
x_2
x_3
x_4
x_5







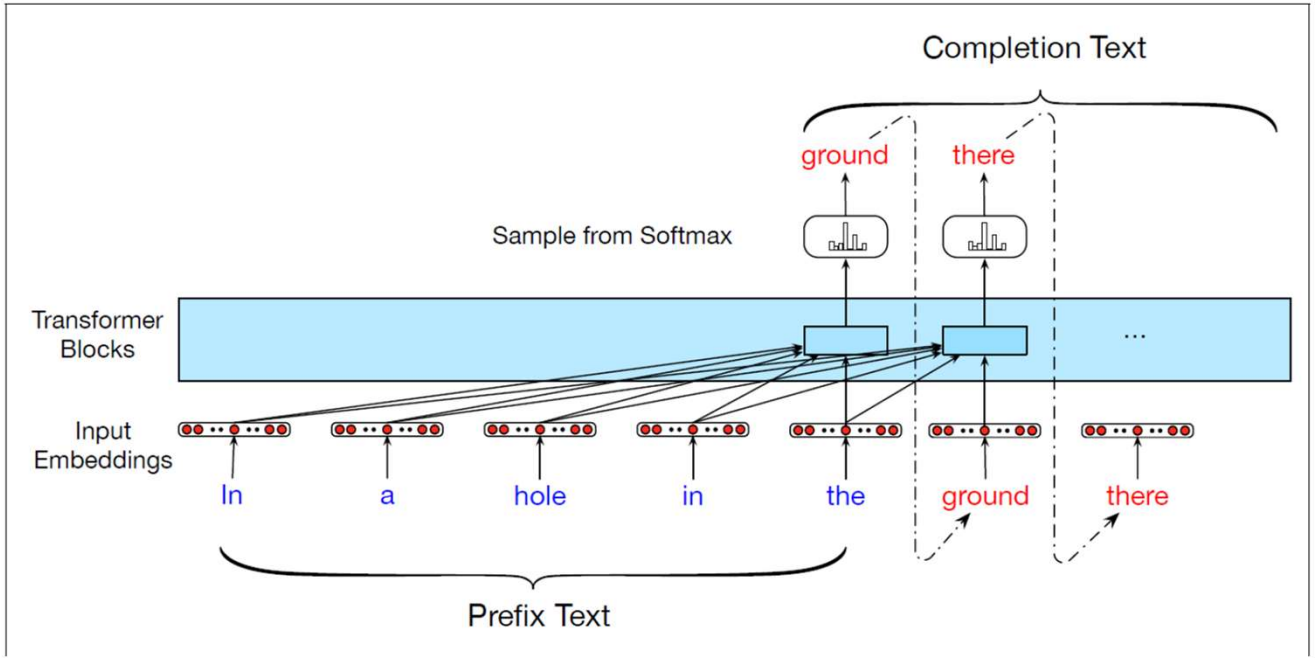
↑Autoregression

→ Sequential Labeling

↗ Sequence Classification

transformer

Autoregressive text completion



Positional Encoding & Positional Embedding

Positional Encoding: Keep Position Information

- Use 1, 2, …, N?

Three requirements (for generalizability):

- Different position has different encoding.
- Intervals should be consistent
- Can be applied to sentences of any length

Positional encoding

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

- Ex. Position *pos* = 1 (second token), *i* = start from 0 to dim/2

$$PE(1) = \left[\sin\left(1/10000^{0/512}\right), \cos\left(1/10000^{0/512}\right), \sin\left(1/10000^{2/512}\right), \cos\left(1/10000^{2/512}\right), \ldots\right]$$

- f = 1 to 1/10000, λ = 2π to $10000 \cdot 2\pi$

Positional encoding

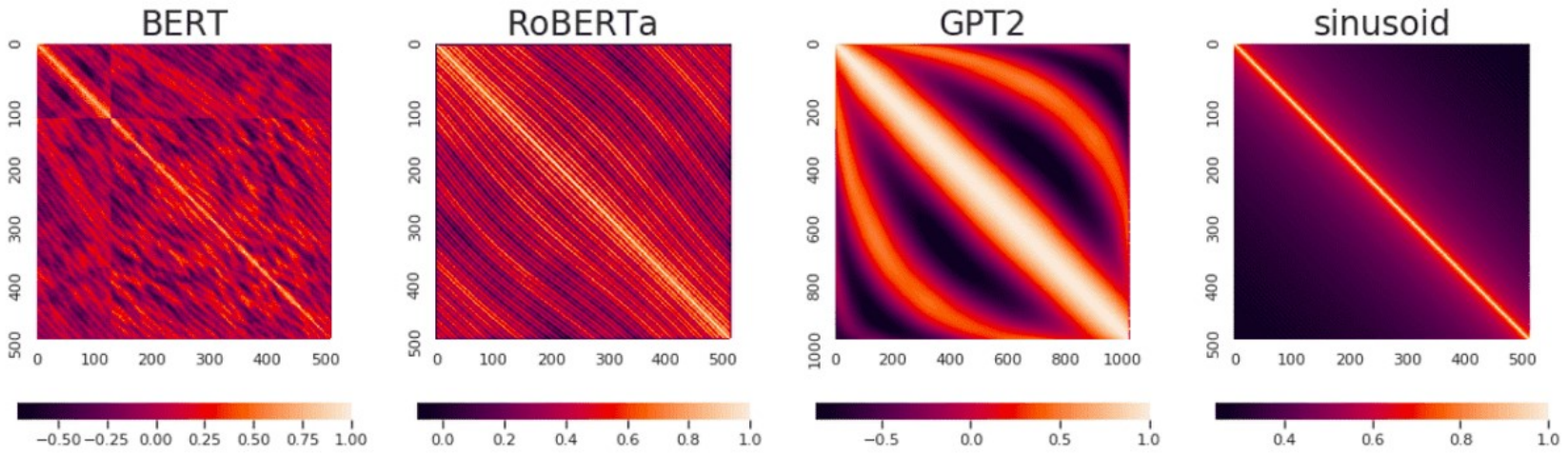
$$\overrightarrow{p_t}^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k . t), & \text{if } i = 2k \\ \cos(\omega_k . t), & \text{if } i = 2k+1 \end{cases}$$

$$\omega_k = \frac{1}{10000^{2k/d}}$$

$$\overrightarrow{p_t} = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d \times 1}$$

Positional Embeddings

- Different from positional encoding
- Each position is mapped to a trainable vector of size dim
- Positional embeddings are trainable, positional encodings are fixed.
- Wang and Chen (2020) What Do Position Embeddings Learn? An Empirical Study of Pre-Trained Language Model Positional Encoding.



Positional Embedding

- Absolute position

- Relative position

Transformer Training/Testing Phase

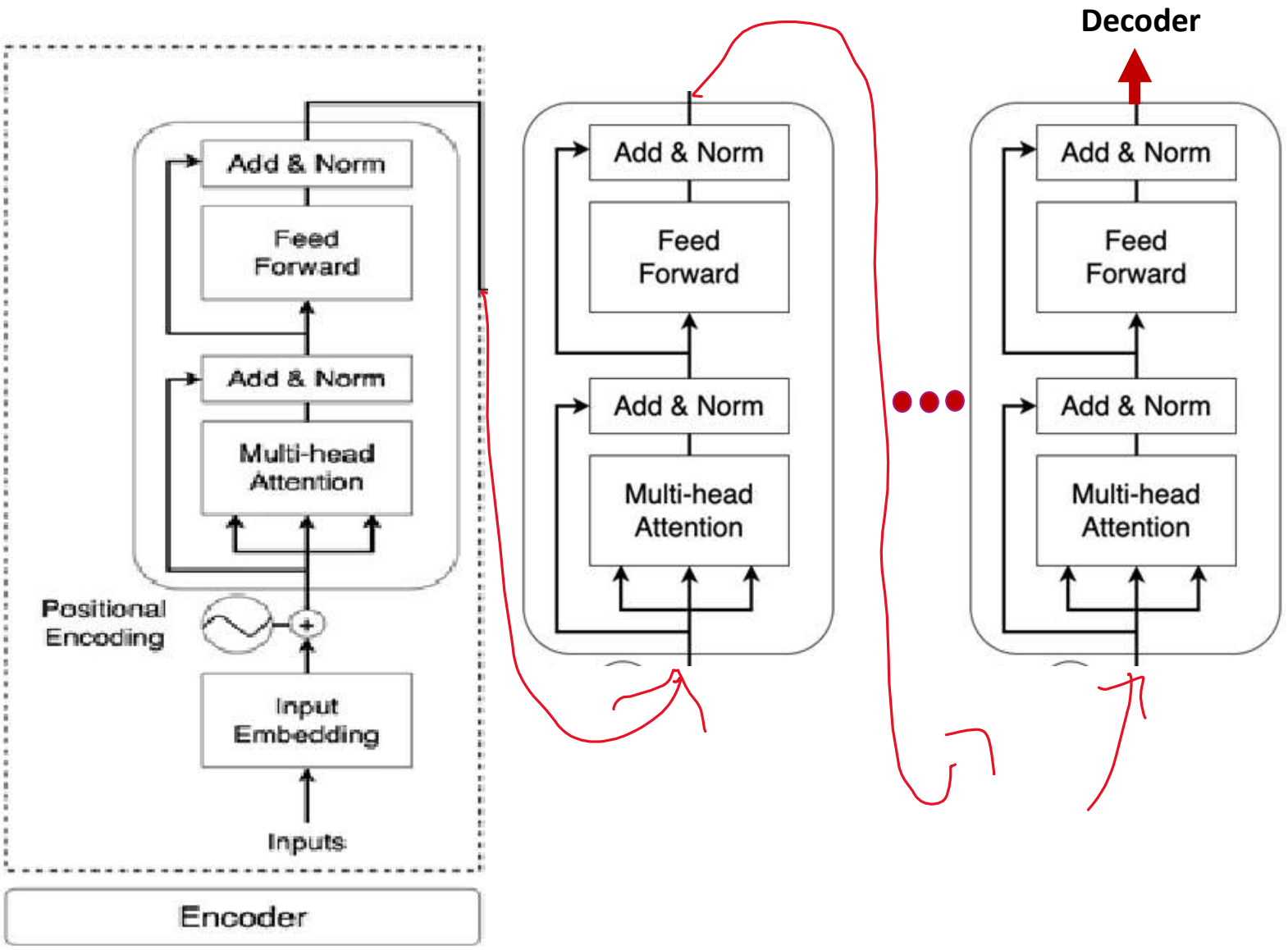
Transformer Training/Inference Input/Output

- Transformer training phase: parallel
- Use the technique of **teacher forcing**

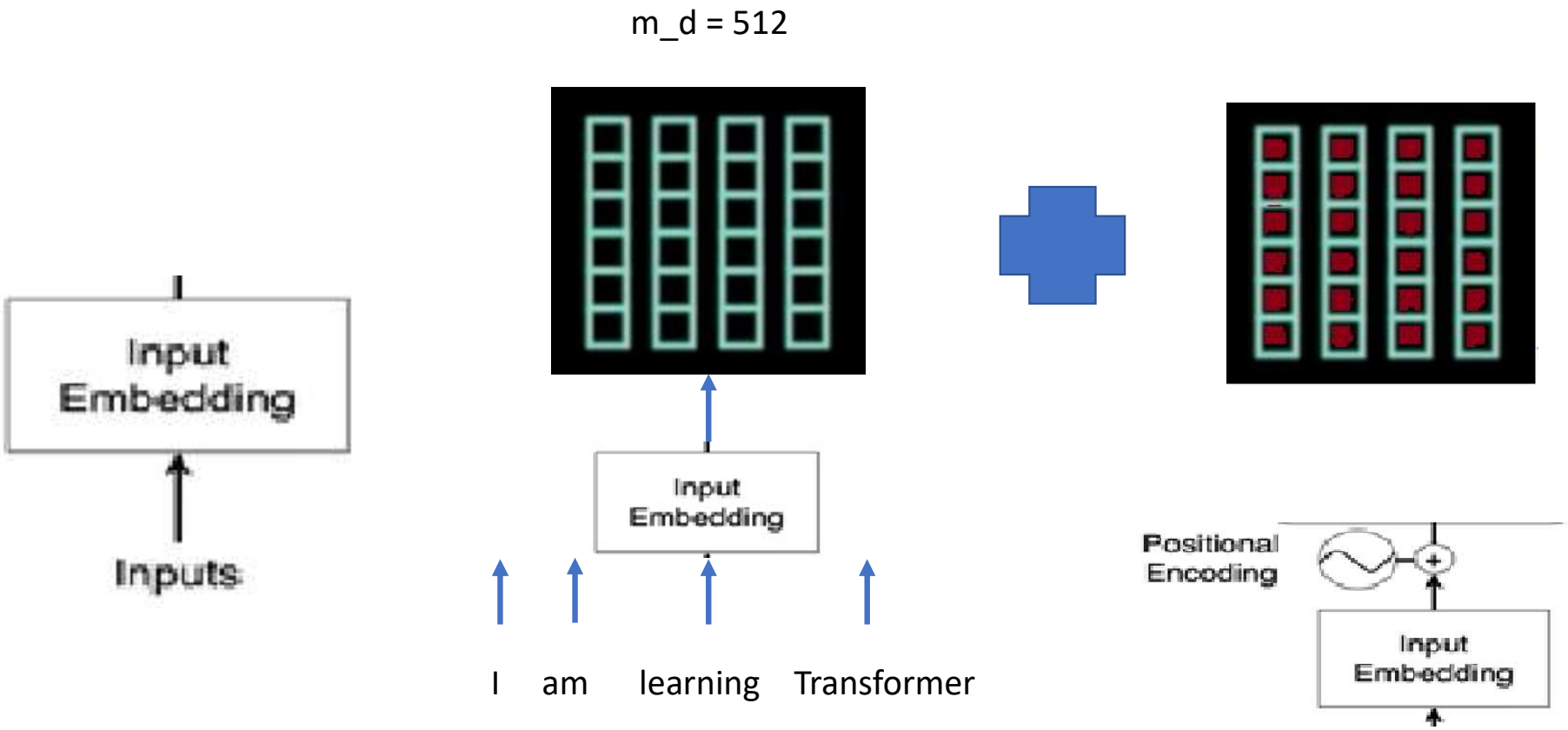
- Transformer testing phase: **decoder not parallel**
- Generate output by looking at the input from the previous timestamp

Encoder

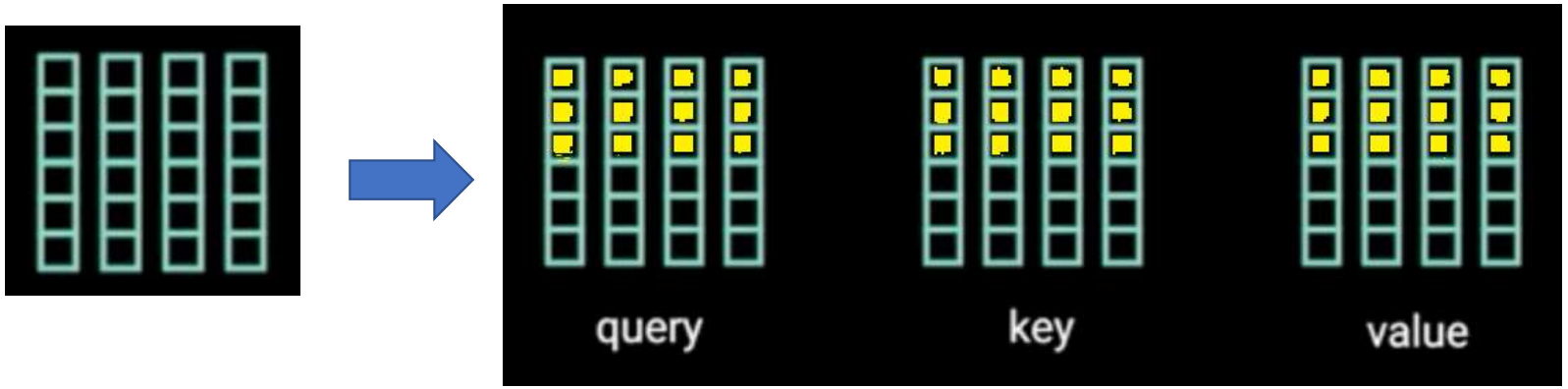
- Stack
  N = 6 (layers)



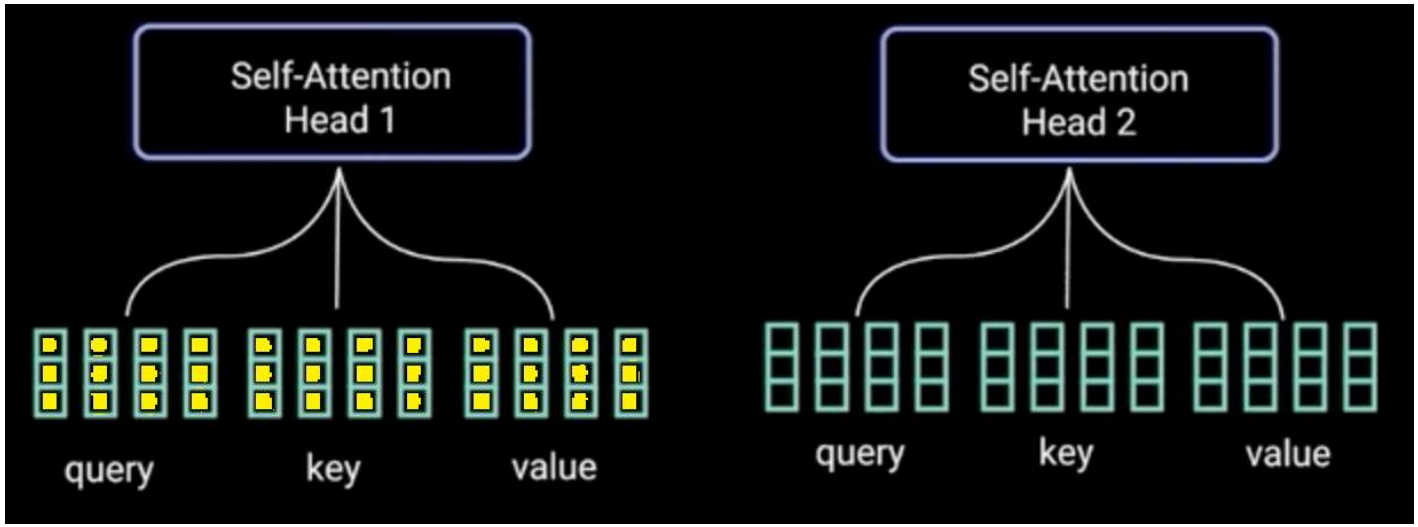
Input

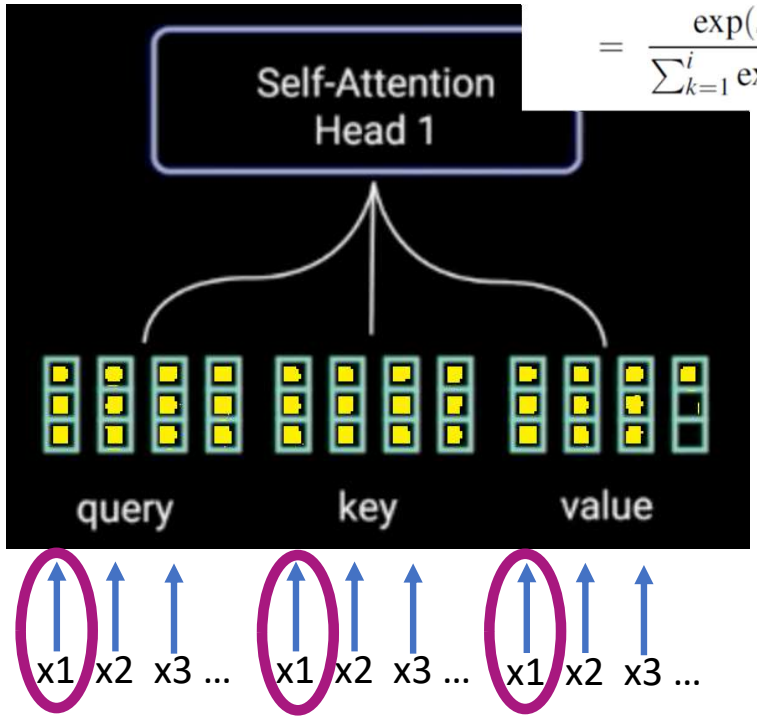

m_d = 512
Input Embedding
Inputs
Input Embedding
I
am
learning
Transformer
Positional Encoding
Input Embedding

Multi-head h=2 as an example (h=8, h_d=512/8=64)

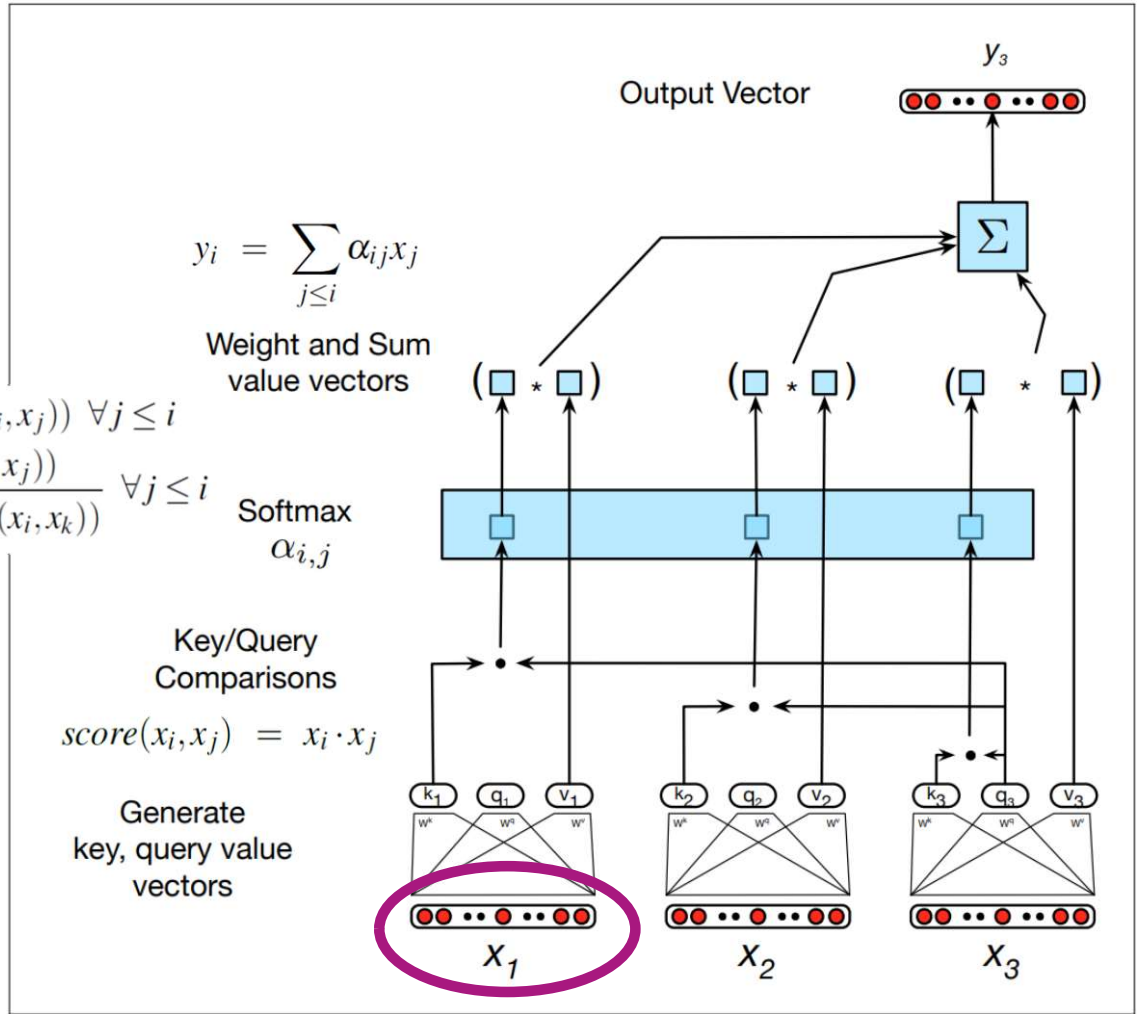


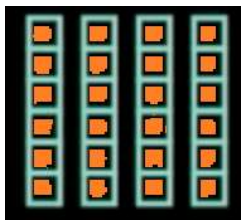
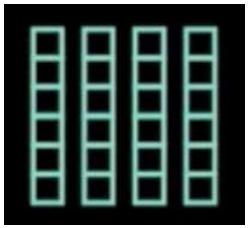
$$Q = W^Q X;\ \ K = W^K X;\ \ V = W^V X$$





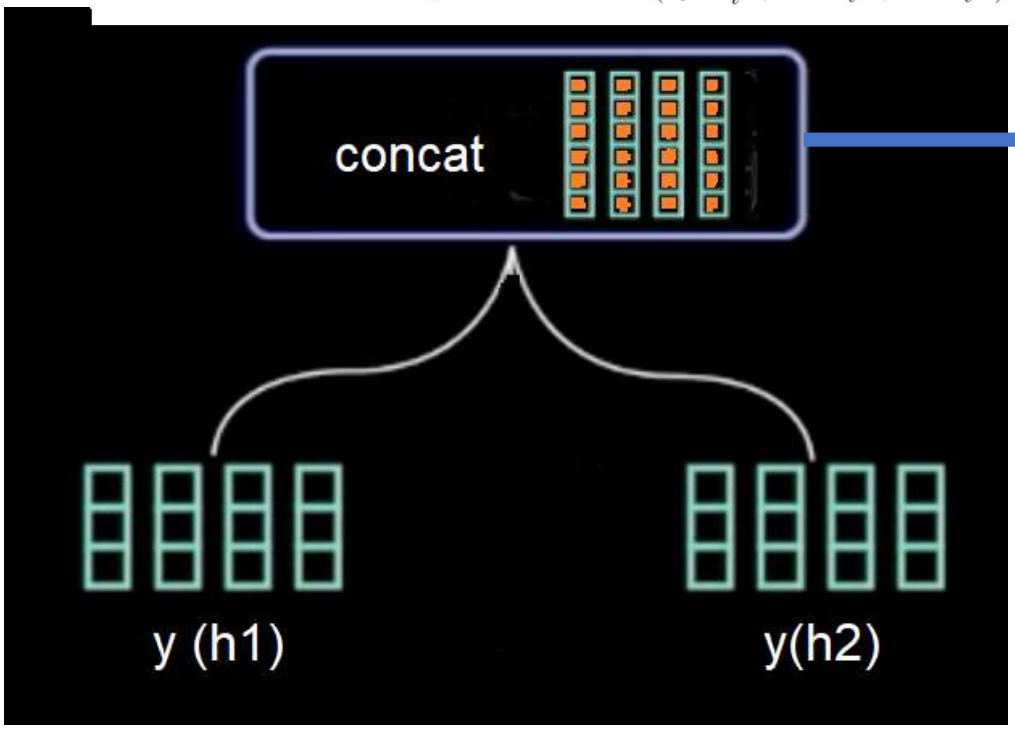
$$\alpha_{ij} = \text{softmax}(score(x_i, x_j)) \;\; \forall j \le i$$
$$= \frac{\exp(score(x_i, x_j))}{\sum_{k=1}^{i} \exp(score(x_i, x_k))} \;\; \forall j \le i$$

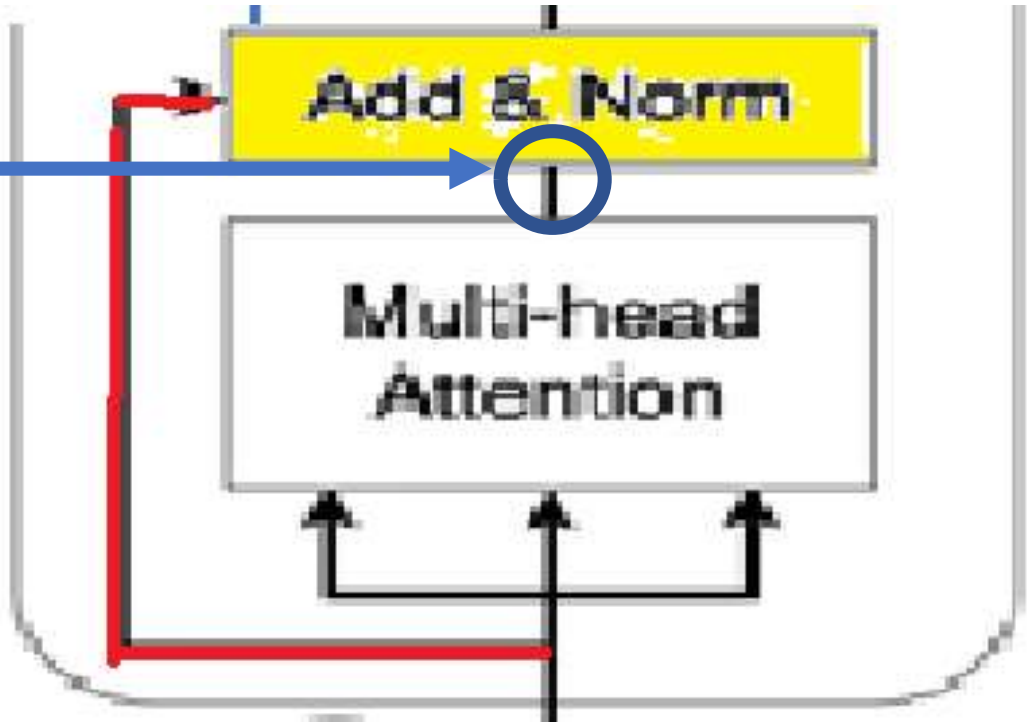


Residual Connection, LayerNorm( + )

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$

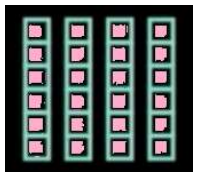
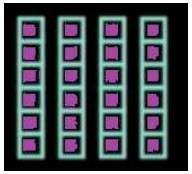
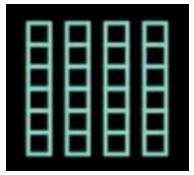
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



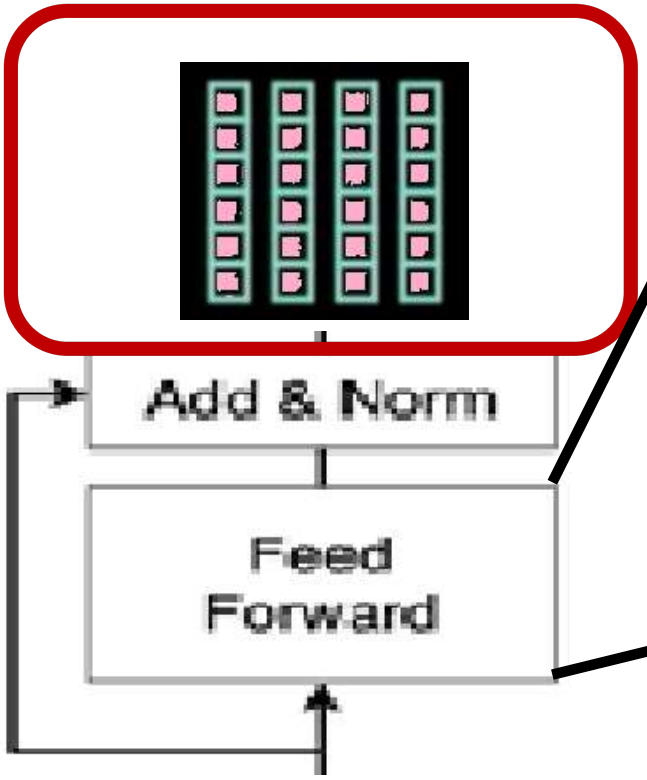


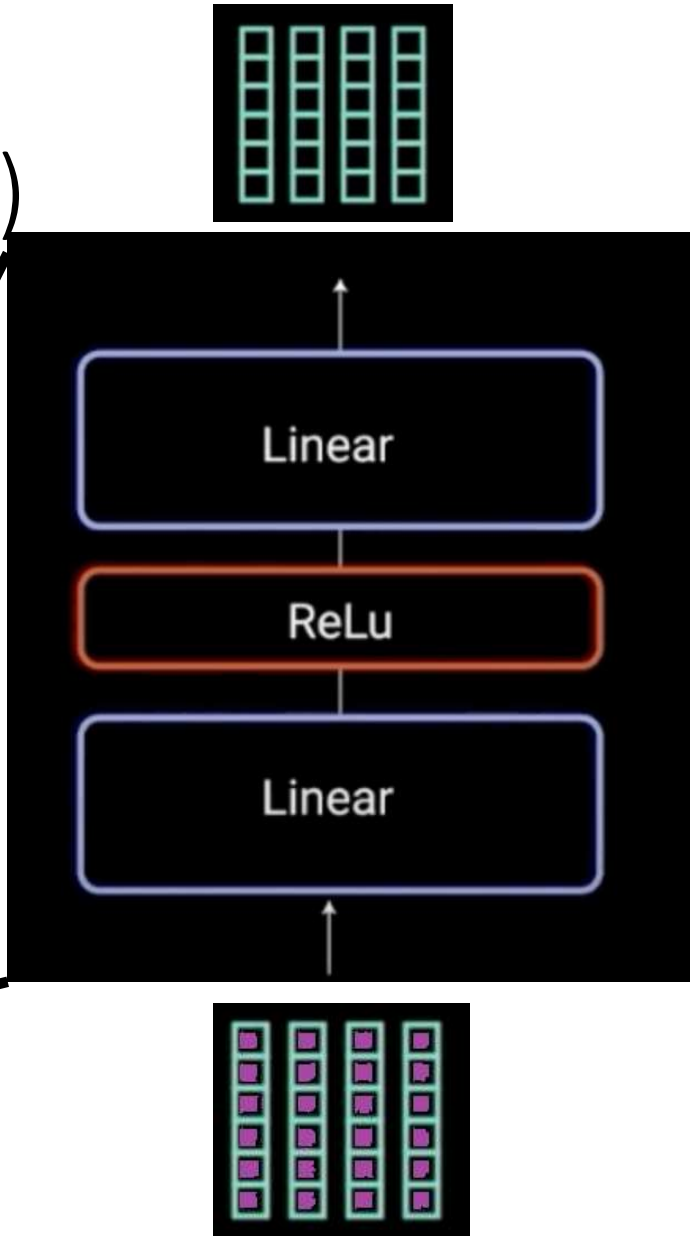
$$y = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}}$$

make mean=0, var =1


= LayerNorm(

+

)

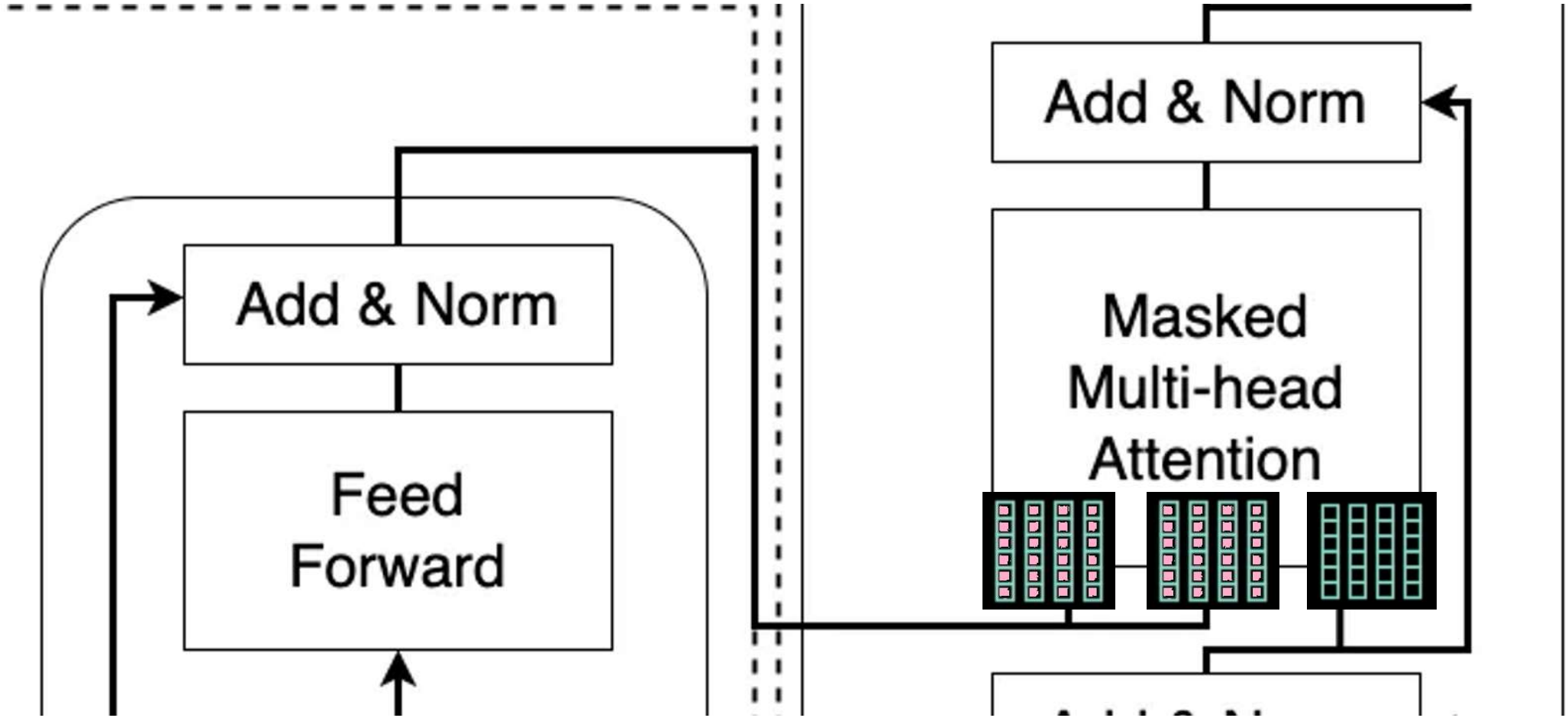
Input To Decoder for Cross Attention





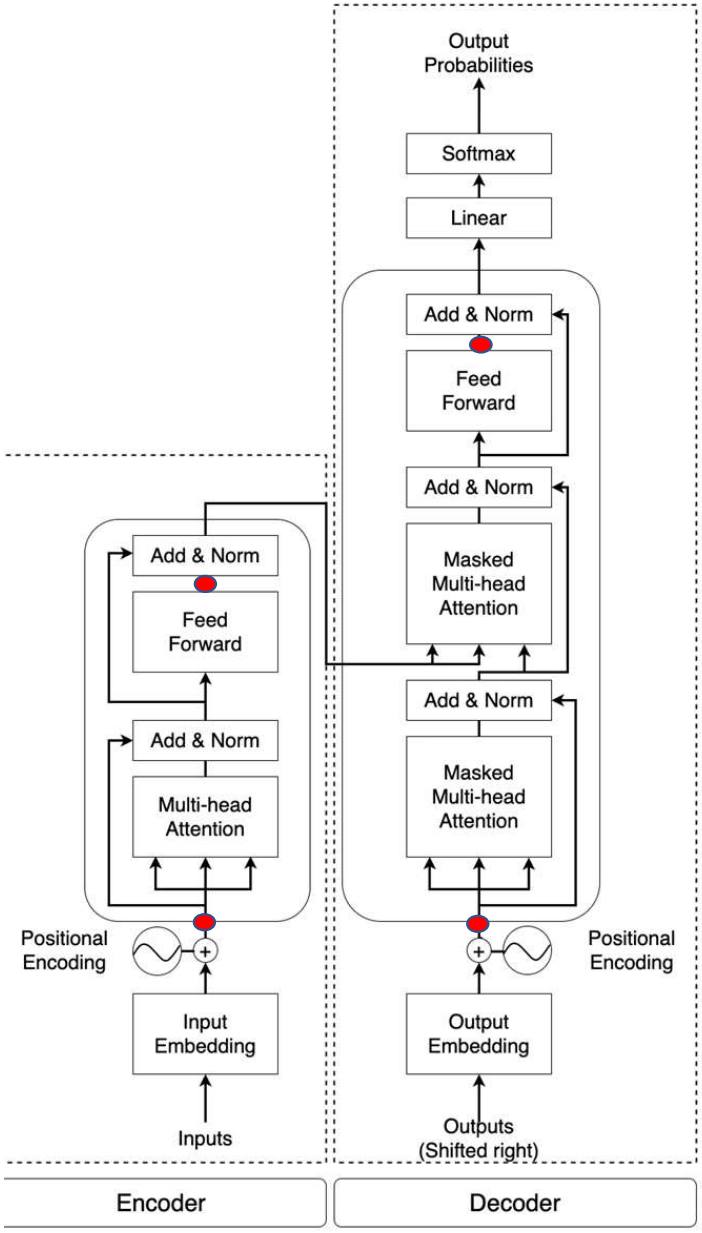
d_model =512

d_ff =2048



Encoder **Decoder**

Dropout


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

Transformer Training

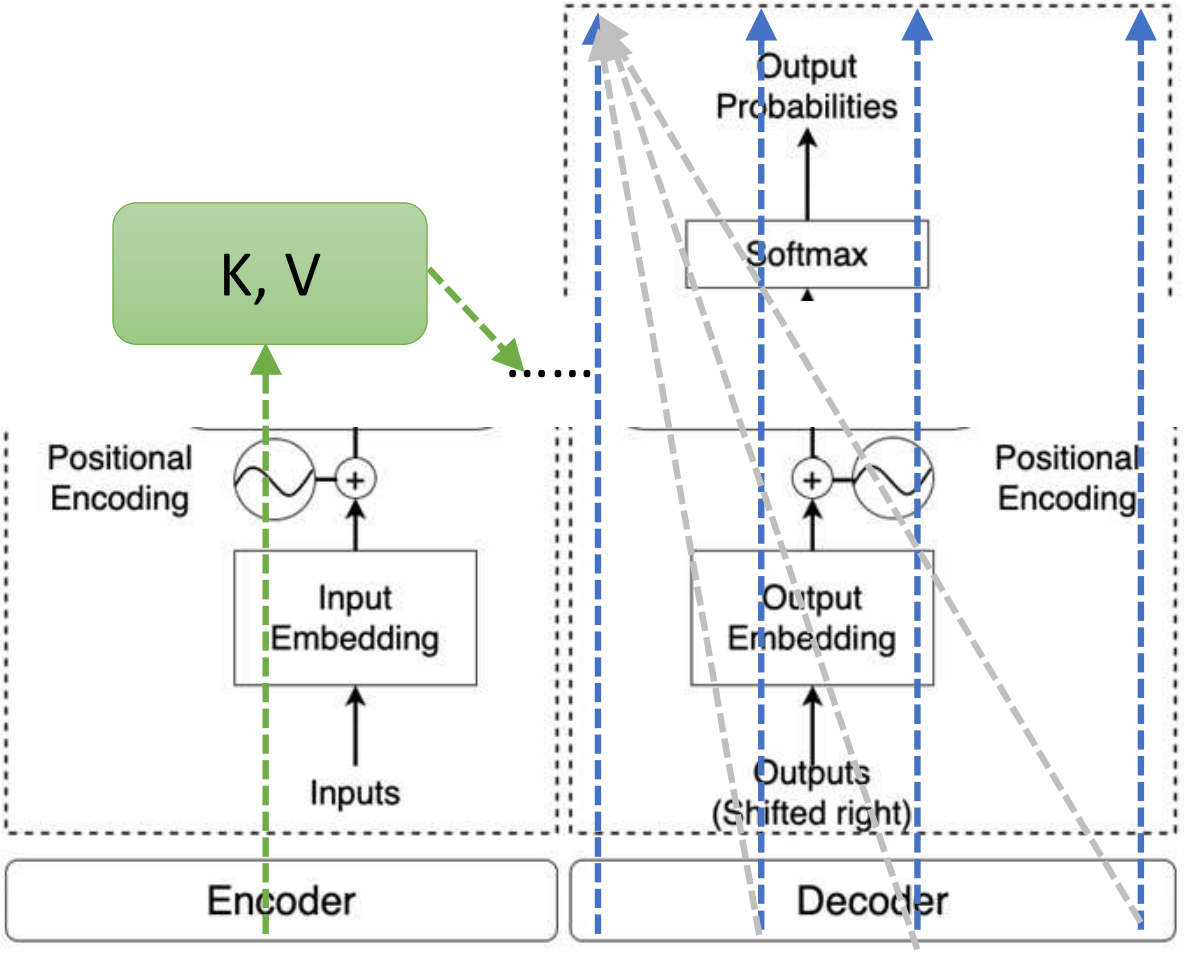
Dataset

- English-German: standard WMT 2014 English-German dataset consisting of about 4.5 million sentence pairs, Vocabulary~37000 tokens
- English-French: the significantly larger WMT 2014 English-French dataset consisting of 36M sentences and split tokens into a 32000 word-piece vocabulary
- One machine with 8 NVIDIA P100 GPUS
- Base model 12 hours, Big model 3.5 days

t=0

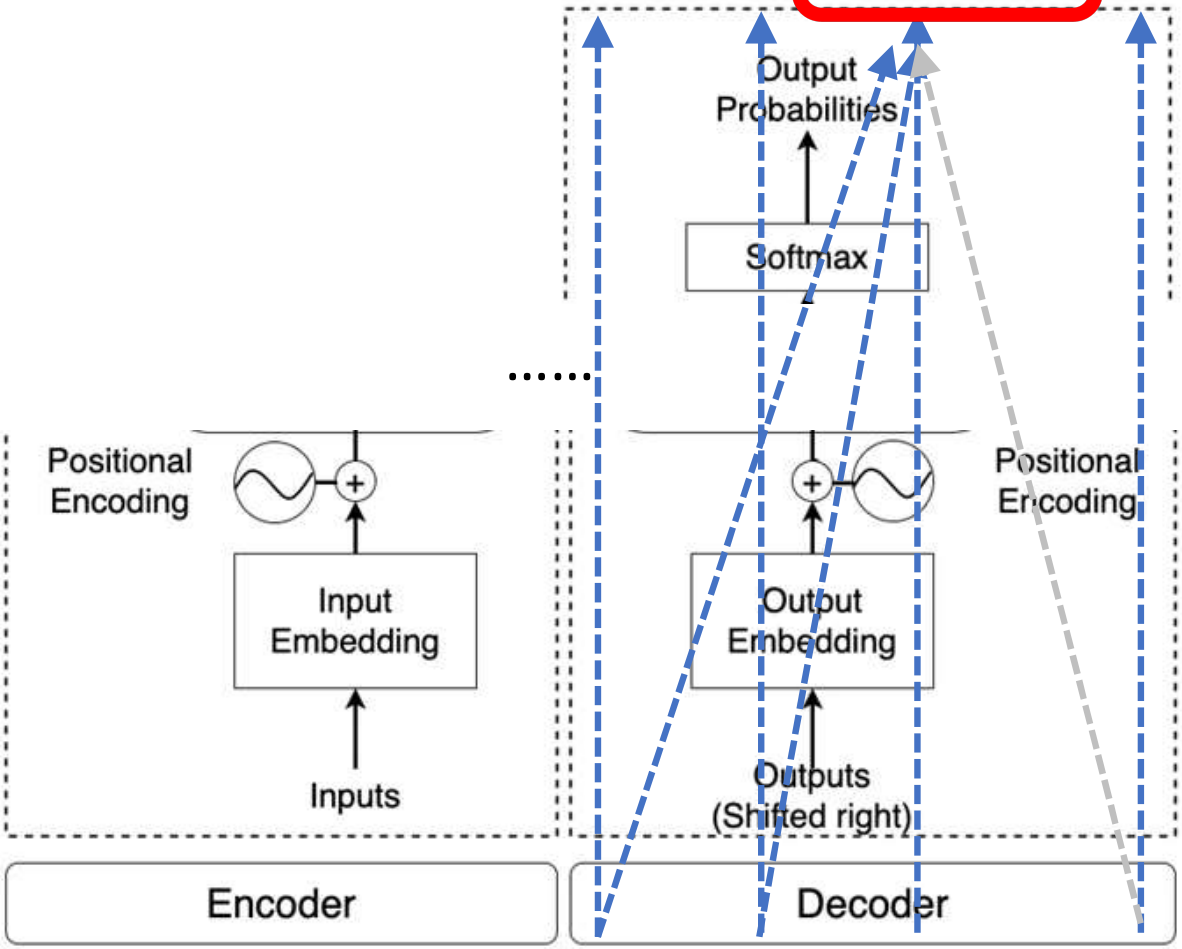
t=1

Is it interesting ? … <eos>



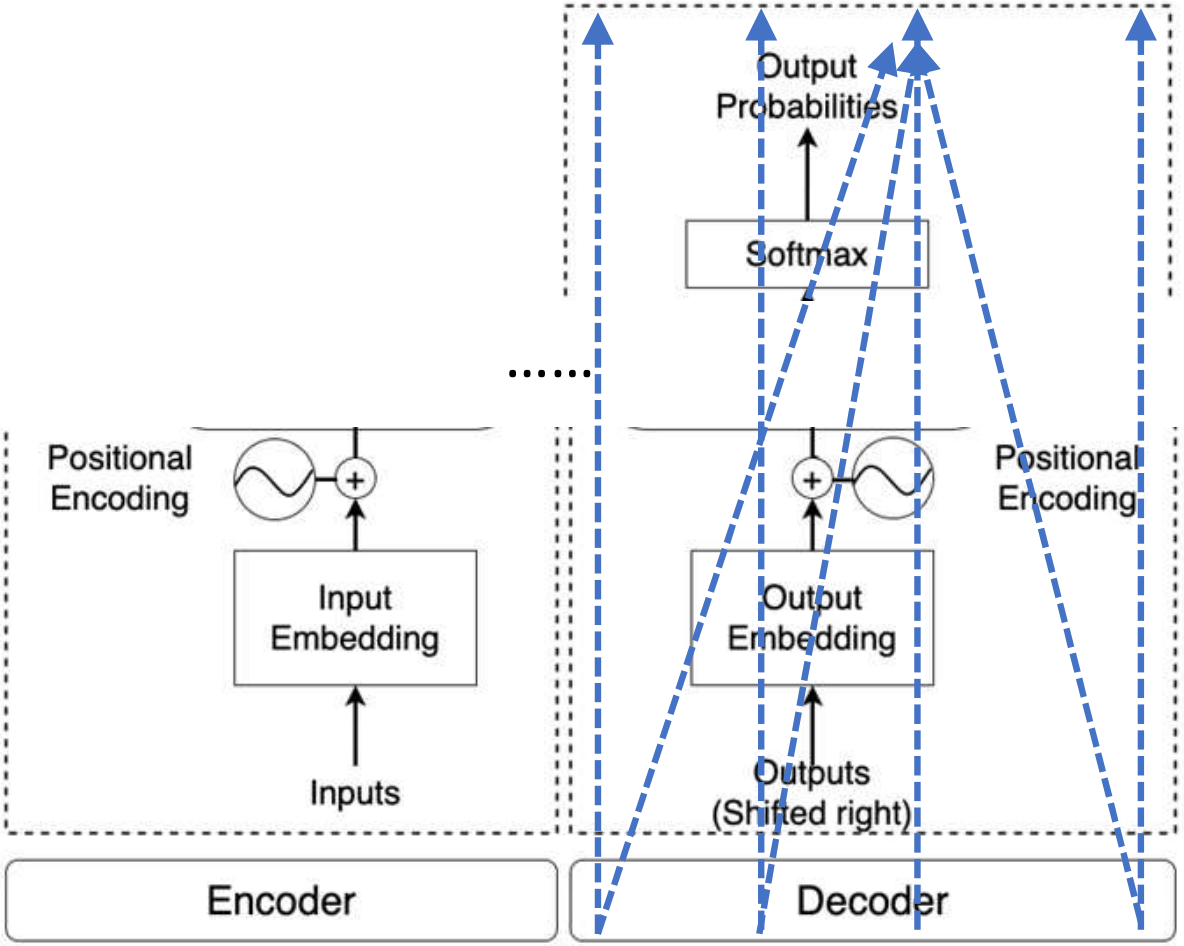
I am learning Transformer <s> Is it interesting ?

Is it interesting ? … <eos>



I am learning Transformer <s> Is it interesting ?

Is it difficult ? … <eos>



I am learning Transformer <s> Is it interesting ?

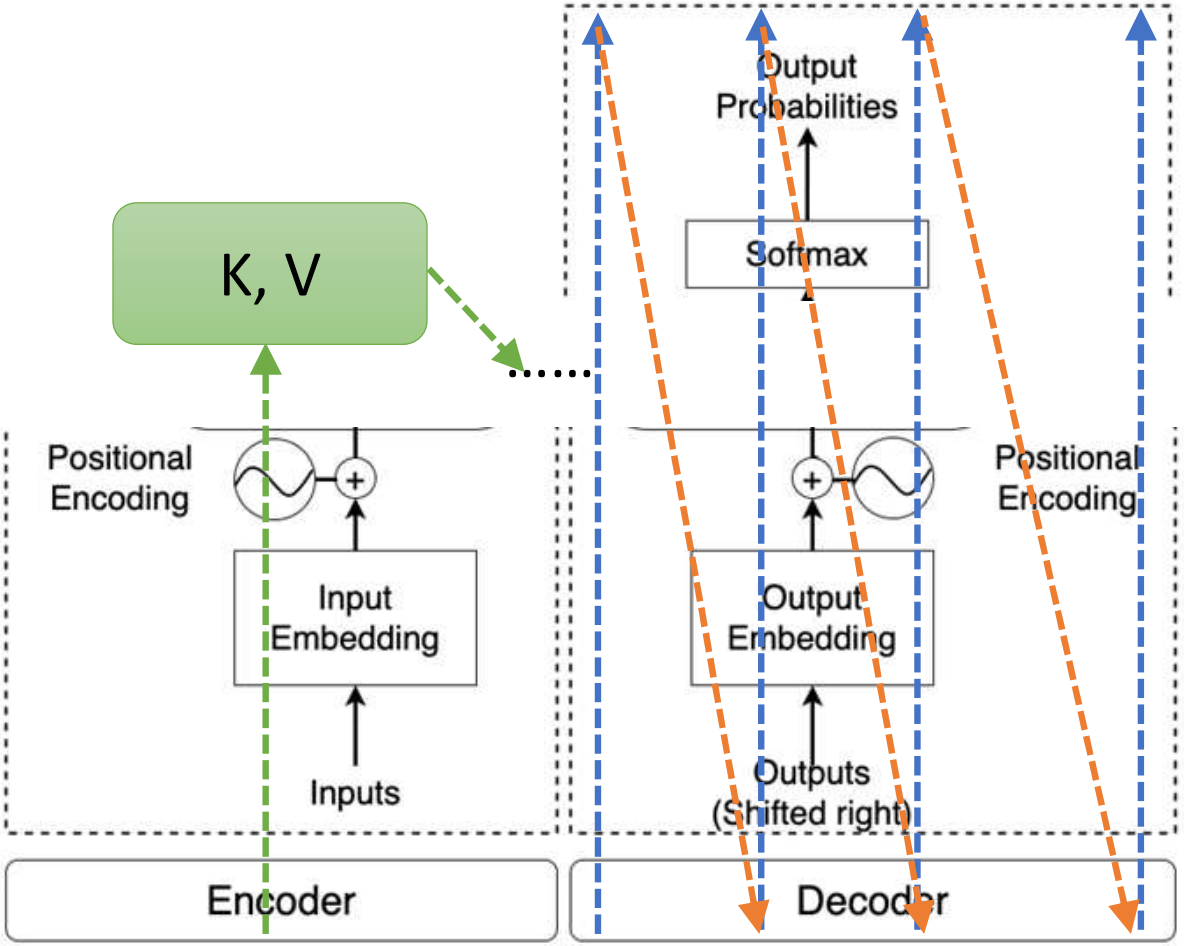
Transformer Inference

t=0

t=1~n

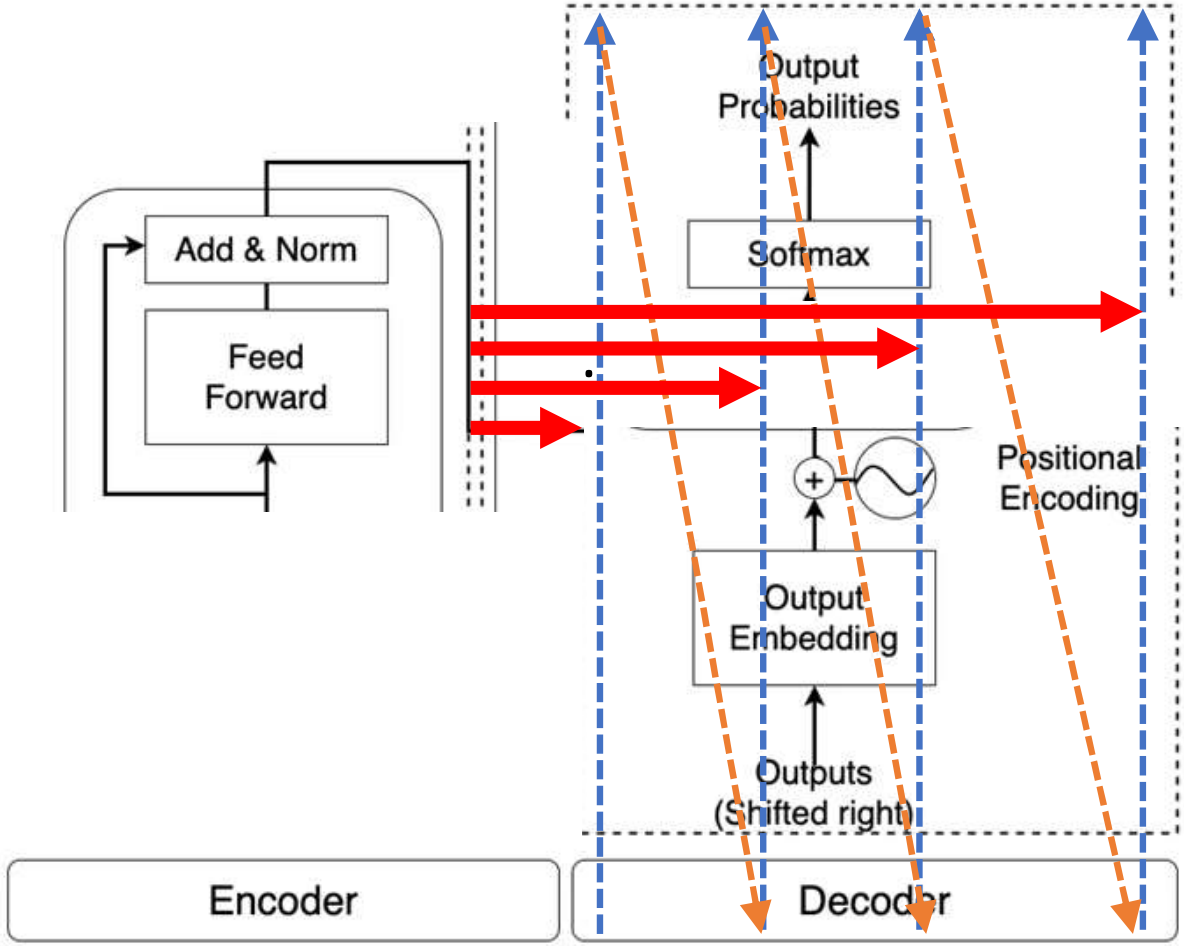
Is it interesting ? … <eos>

Is he a student … <eos>



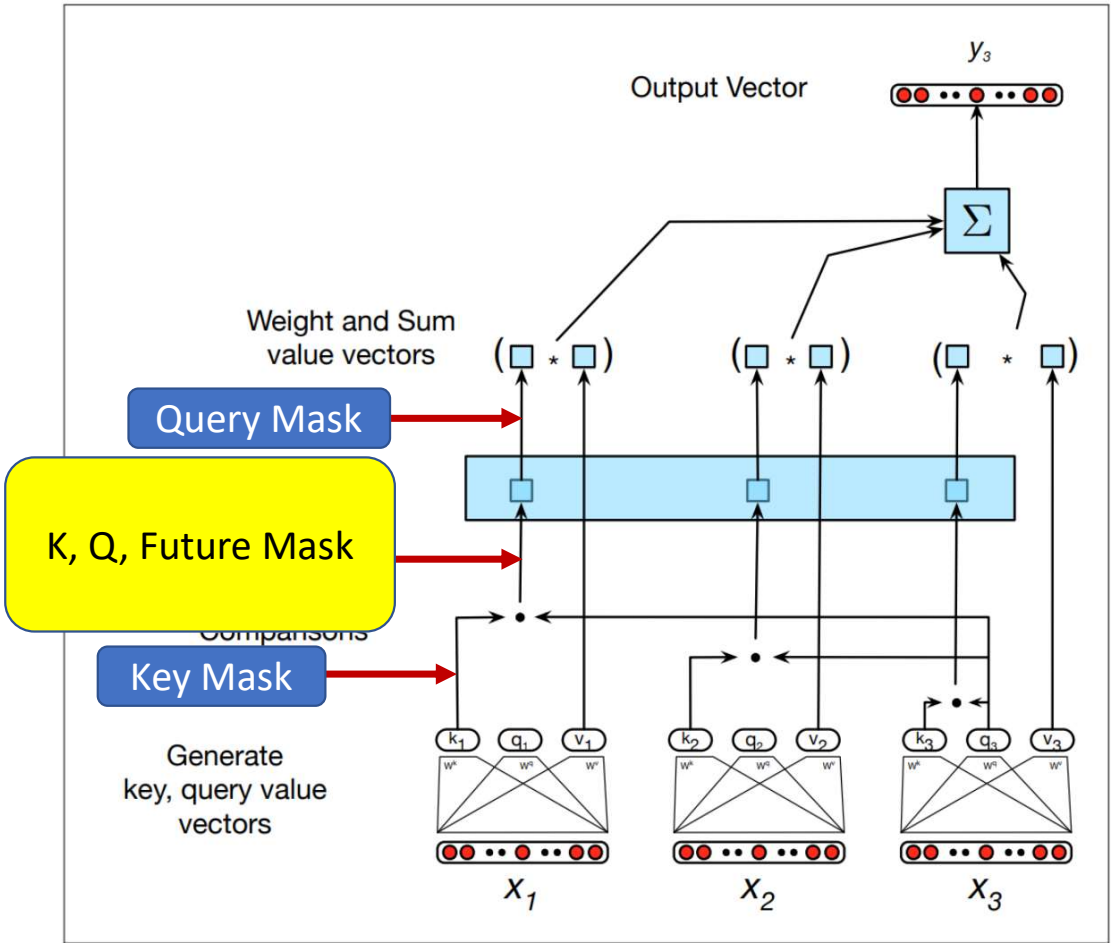
I am learning Transformer <s> Is he a

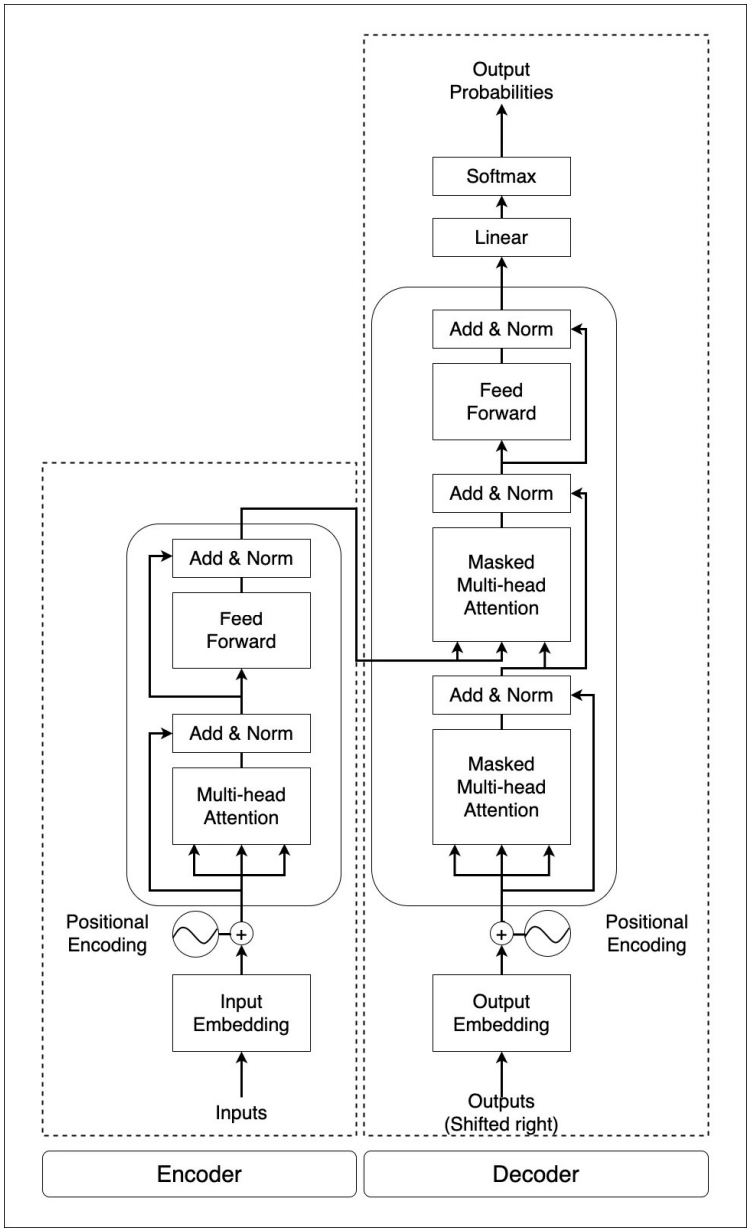
Is he a student … <eos>

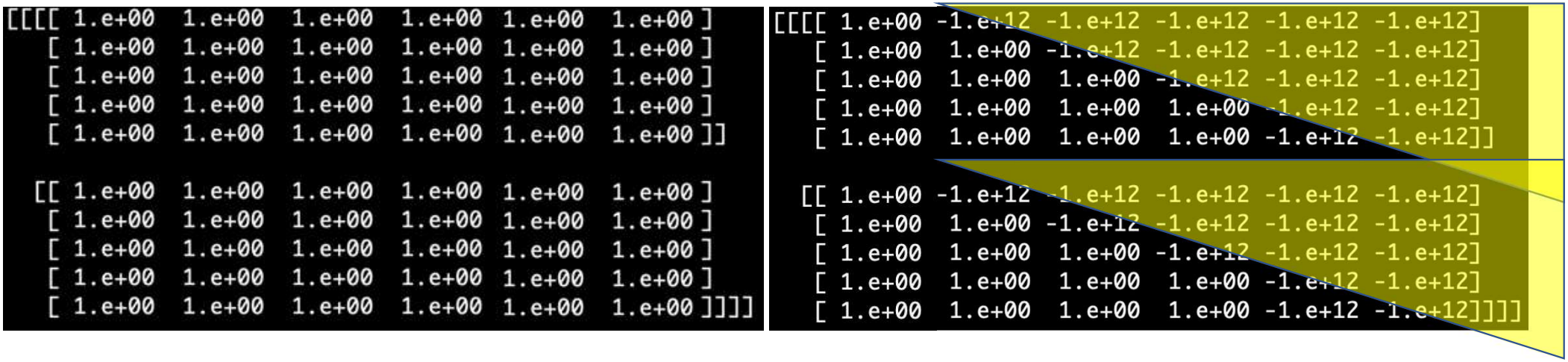


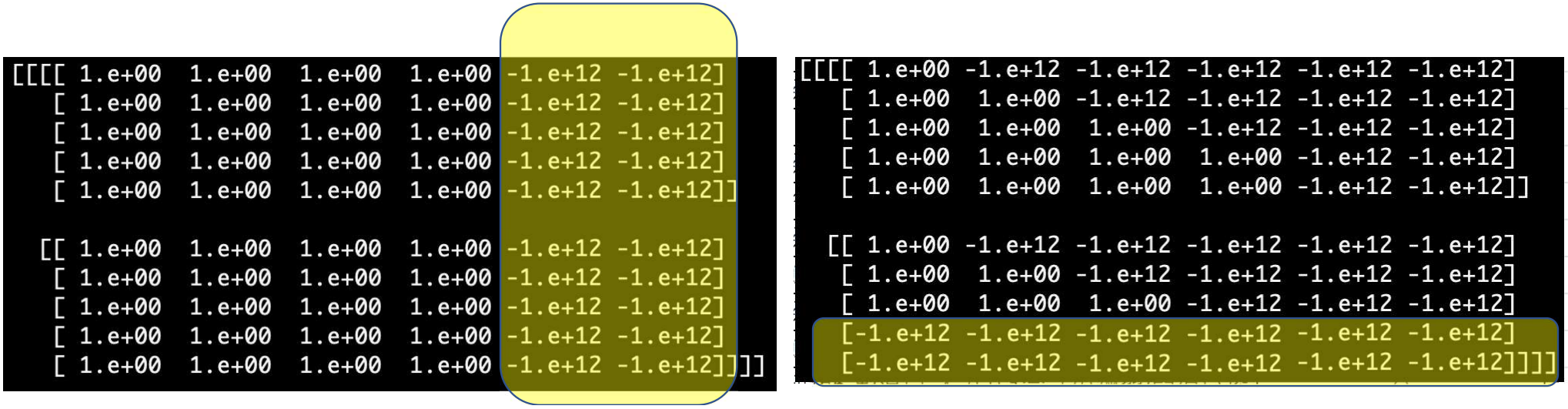
I am learning Transformer <s> Is he a

Masking in Decoder






[[[[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00]
[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00]
[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00]
[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00]
[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00]]
[[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00]
[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00]
[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00]
[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00]
[1.e+00 1.e+00 1.e+00 1.e+00 1.e+00 1.e+00]]]]
[[[[1.e+00 -1.e+12 -1.e+12 -1.e+12 -1.e+12 -1.e+12]
[1.e+00 1.e+00 -1.e+12 -1.e+12 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]]
[[1.e+00 -1.e+12 -1.e+12 -1.e+12 -1.e+12 -1.e+12]
[1.e+00 1.e+00 -1.e+12 -1.e+12 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]]]]


[[[[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]]
[[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]]]]
[[[[1.e+00 -1.e+12 -1.e+12 -1.e+12 -1.e+12 -1.e+12]
[1.e+00 1.e+00 -1.e+12 -1.e+12 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12]]
[[1.e+00 -1.e+12 -1.e+12 -1.e+12 -1.e+12 -1.e+12]
[1.e+00 1.e+00 -1.e+12 -1.e+12 -1.e+12 -1.e+12]
[1.e+00 1.e+00 1.e+00 -1.e+12 -1.e+12 -1.e+12]
[-1.e+12 -1.e+12 -1.e+12 -1.e+12 -1.e+12 -1.e+12]
[-1.e+12 -1.e+12 -1.e+12 -1.e+12 -1.e+12 -1.e+12]]]]

Masking in Encoder?

- Yes, but for different purpose
- Masked attention on padding
- Example:
  vector = [2, 0.5, 0.8, 1, 0, 0, 0, 0] (padded),
  softmax = [0.41, 0.09, 0.12, 0.15, 0.06, 0.06, 0.06, 0.06]

- Masking before softmax:
  set padded values to -inf
  masking = [2, 0.5, 0.8, 1, -1e9, -1e9, -1e9, -1e9]
  softmax = [0.53, 0.12, 0.16, 0.19, 0, 0, 0, 0]

- **Please refer to LAB codes**

Transformer for Tasks

	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32					5.01	25.4	60

$$PP(p) := 2^{H(p)} = 2^{-\sum_x p(x) \log_2 p(x)} = \prod_x p(x)^{-p(x)}$$

		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)	positional embedding instead of sinusoids									4.92	25.7	
big	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

Transformer for English Parsing

Parser	Training	WSJ 23 F1
Vinyals & Kaiser el al. (2014) [37]	WSJ only, discriminative	88.3
Petrov et al. (2006) [29]	WSJ only, discriminative	90.4
Zhu et al. (2013) [40]	WSJ only, discriminative	90.4
Dyer et al. (2016) [8]	WSJ only, discriminative	91.7
Transformer (4 layers)	WSJ only, discriminative	91.3
Zhu et al. (2013) [40]	semi-supervised	91.3
Huang & Harper (2009) [14]	semi-supervised	91.3
McClosky et al. (2006) [26]	semi-supervised	92.1
Vinyals & Kaiser el al. (2014) [37]	semi-supervised	92.1
Transformer (4 layers)	semi-supervised	92.7
Luong et al. (2015) [23]	multi-task	93.0
Dyer et al. (2016) [8]	generative	93.3

transformer

Summarization

This scheme can be used in text-to-text applications, e.g., MT, QA

$$(x_1, \ldots, x_m), (y_1, \ldots, y_n)$$
$$(x_1, \ldots, x_m, \delta, y_1, \ldots y_n)$$

Original Article

The only thing crazier than a guy in snowbound Massachusetts boxing up the powdery white stuff and offering it for sale online? People are actually buying it. For $89, self-styled entrepreneur Kyle Waring will ship you 6 pounds of Boston-area snow in an insulated Styrofoam box – enough for 10 to 15 snowballs, he says.

But not if you live in New England or surrounding states. "We will not ship snow to any states in the northeast!" says Waring's website, ShipSnowYo.com. "We're in the business of expunging snow!"

His website and social media accounts claim to have filled more than 133 orders for snow – more than 30 on Tuesday alone, his busiest day yet. With more than 45 total inches, Boston has set a record this winter for the snowiest month in its history. Most residents see the huge piles of snow choking their yards and sidewalks as a nuisance, but Waring saw an opportunity.

According to Boston.com, it all started a few weeks ago, when Waring and his wife were shoveling deep snow from their yard in Manchester-by-the-Sea, a coastal suburb north of Boston. He joked about shipping the stuff to friends and family in warmer states, and an idea was born. His business slogan: "Our nightmare is your dream!" At first, ShipSnowYo sold snow packed into empty 16.9-ounce water bottles for $19.99, but the snow usually melted before it reached its destination...

Summary

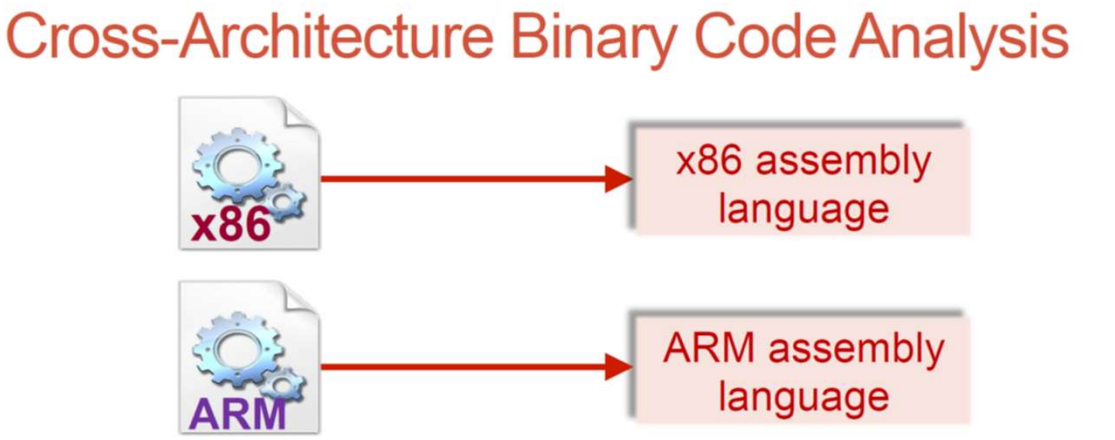
Kyle Waring will ship you 6 pounds of Boston-area snow in an insulated Styrofoam box – enough for 10 to 15 snowballs, he says. But not if you live in New England or surrounding states.

Embeddings for Codes

A Cross-Architecture Instruction Embedding Model for Natural Language Processing-Inspired Binary Code Analysis

Kimberly Redmond, Lannan (Lisa) Luo, Qiang Zeng

University of South Carolina



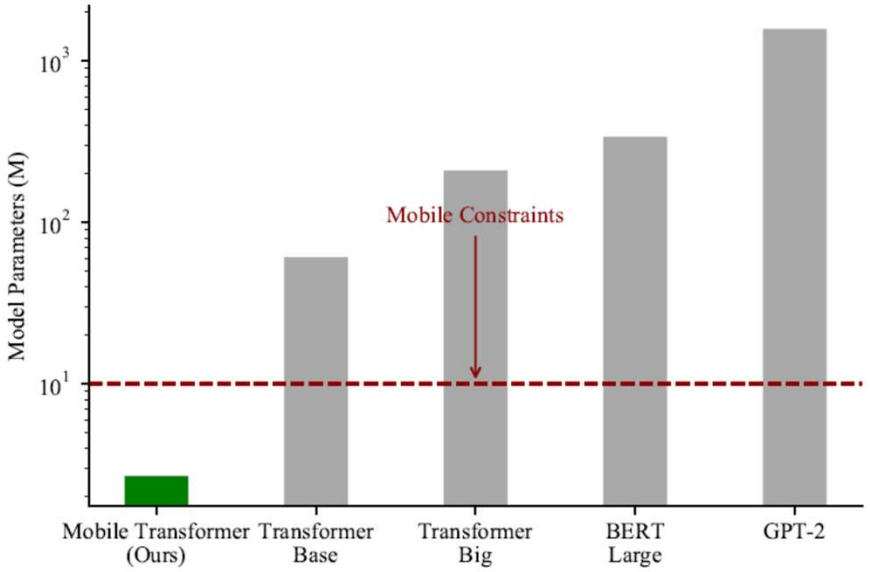
NLP-inspired binary code analysis:

- instructions are regarded as words
- instruction → instruction embeddings

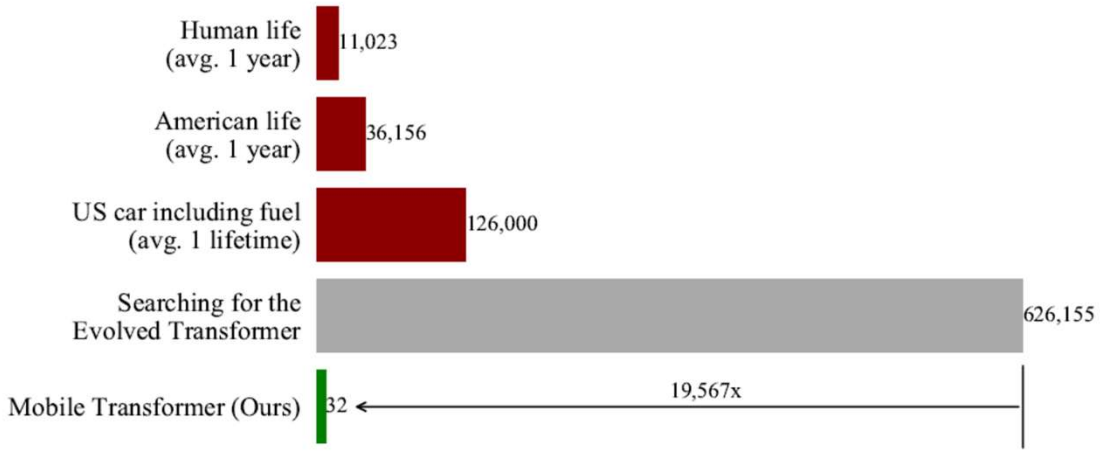
Cross-architecture binary code analysis:

- instruction → cross-architecture instruction embeddings
- similar instructions from different arch. have similar embeddings

Transformer Inference



(a) Model sizes of modern NLP models.



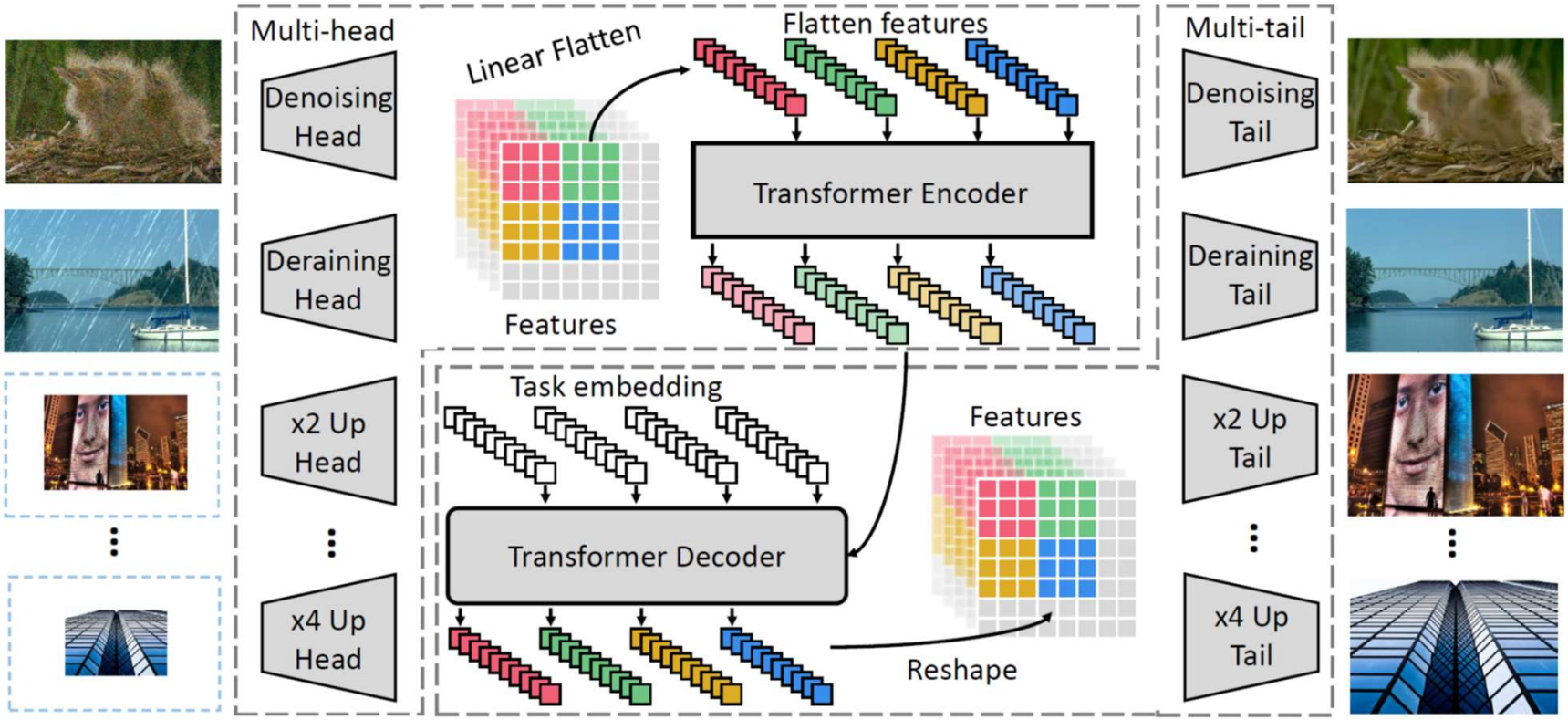
(b) The design cost measured in pounds of CO_2 emission.

- EFFICIENT TRANSFORMER FOR MOBILE APPLICATIONS,
- https://zhanghaowu.me/assets/pdf/Presentation_ENMT.pdf

Improving Inference Speeds of Transformer Models, https://medium.com/gumgum-tech/improving-inference-speeds-of-transformer-models-e03944a018aa

Below are the results of fp16 inference for RoBERTa, RoBERTa with PABEE, and DistilRoBERTa. The results shown are on a single NVIDIA Tesla V100 GPU with 16gb RAM:

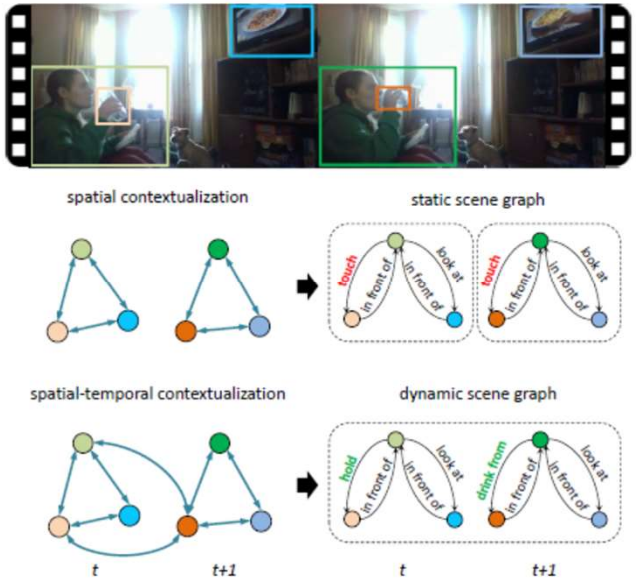
Model	Test F0.5	Test Set's Inference Speed
RoBERTa-Base	76.75%	90 messages/second
RoBERTa-PABEE	76.17%	110 messages/second
Distil-RoBERTa	76.89%	159 messages/second

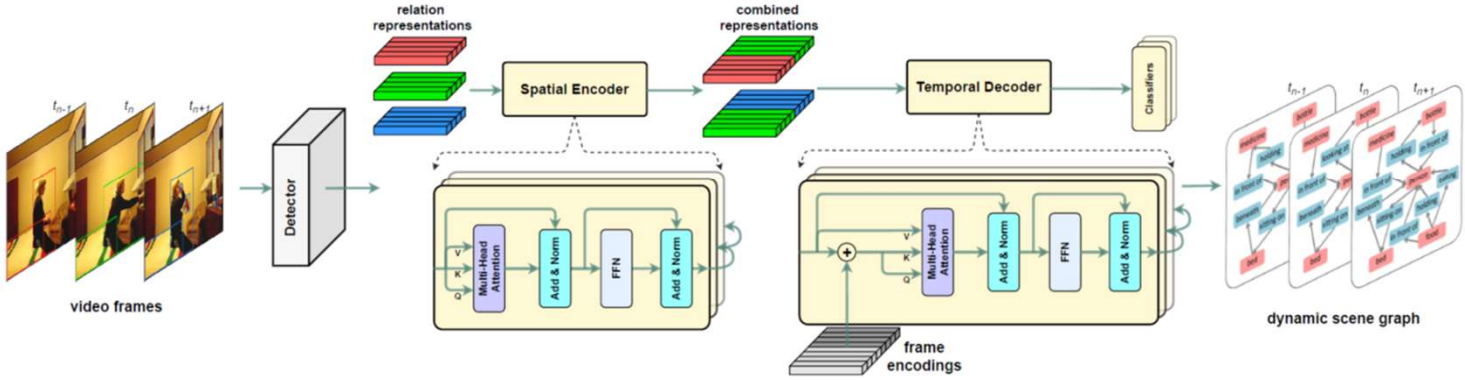


- Pre-trained Image Processing Transformer, CVPR 2021

More Transformer Applications

- Spatial-Temporal Transformer for Dynamic Scene Graph Generation (ICCV 2021)





Visual Story Telling: KG-VIST, PR-VIST