Computer Vision

CS308

Feng Zheng

SUSTech CS Vision Intelligence and Perception

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

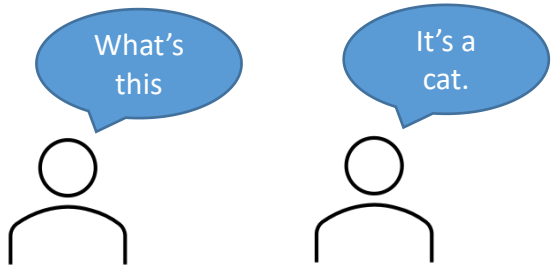
Content

- Introduction to Vision-Language Learning
- Overview of Vision-Language Research
- Sequence-to-sequence Models
  - RNNs
  - Transformers

Vision-Language Learning

Vision-Language Learning

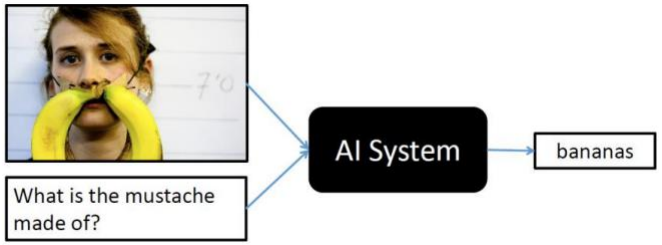
- Vision and language are two fundamental capabilities of human intelligence. Humans routinely perform tasks through the interactions between vision and language.



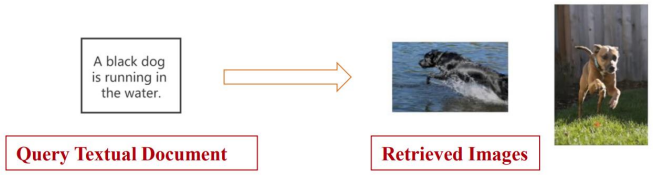
Mei, Tao, Wei Zhang, and Ting Yao. "Vision and language: from visual perception to content creation."
Miech, Antoine, et al. "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips."
Cat image is retrieved from here

Vision-Language Learning

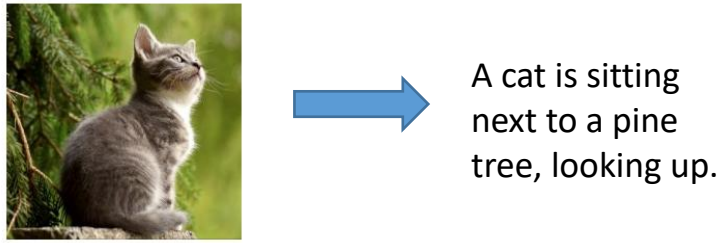
- Vision-language learning aims to build models that can process and relate information from vision and language.
- V+L research is about how to train a smart AI system that can see and talk.



Visual Question Answering



Cross-modal Retrieval



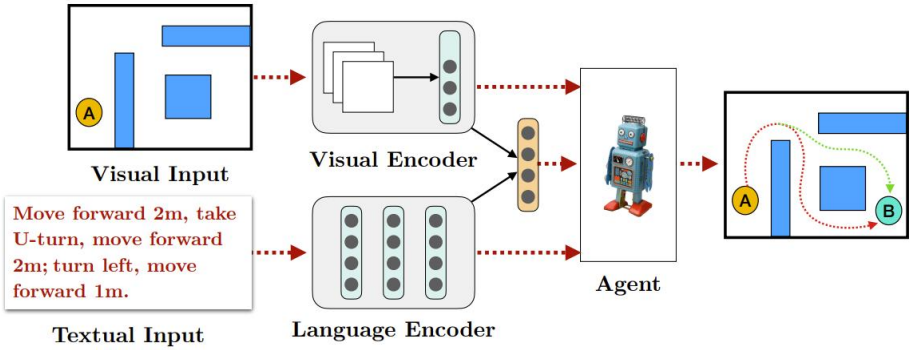
Visual Captioning

Antol, Stanislaw, et al. "Vqa: Visual question answering."

Vision-Language Learning

- Vision-language learning aims to build models that can process and relate information from vision and language.
- V+L research is about how to train a smart AI system that can see and talk.

Text-to-image Synthesis



Vision-Language Navigation

Midjourney: https://www.midjourney.com/home/
Stable Diffusion 2.0. https://stability.ai/blog/stable-diffusion-v2-release

Uppal, Shagun, et al. "Multimodal research in vision and language: A review of current and emerging trends."
Xu, Tao, et al. "Attngan: Fine-grained text to image generation with attentional generative adversarial networks."

Overview of Vision-Language Research

Visual Captioning

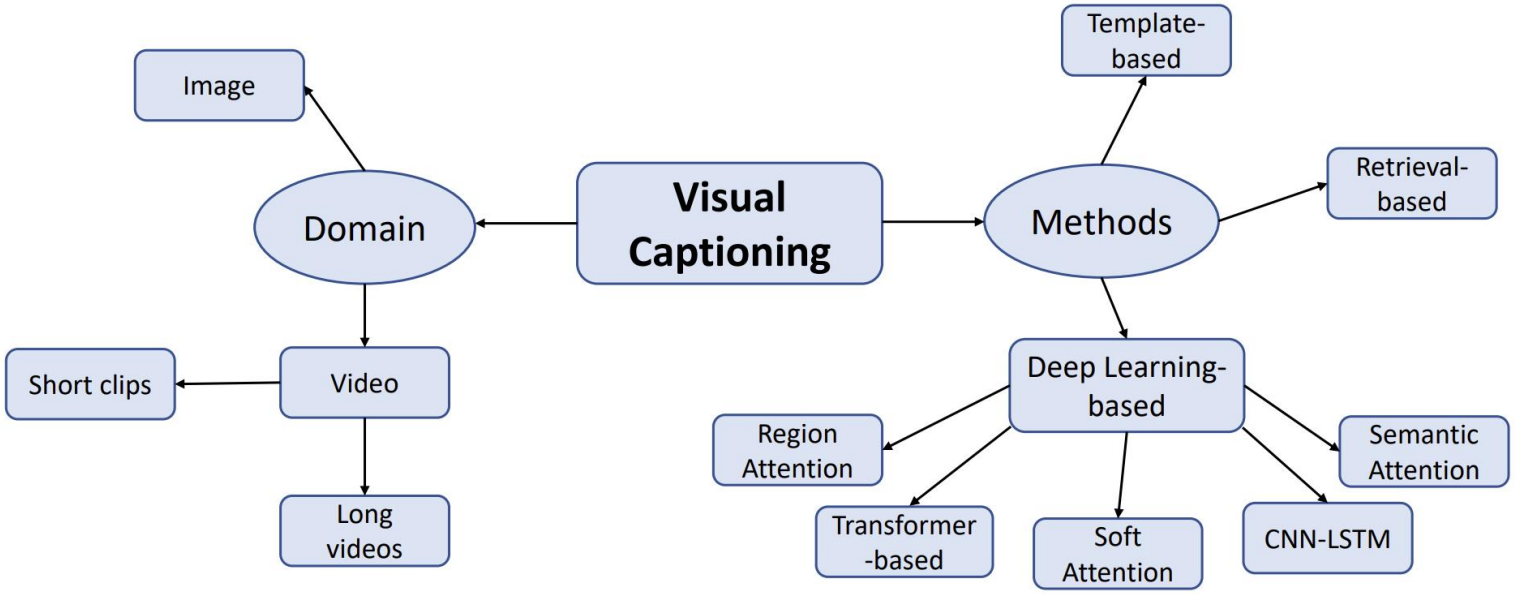
- Generating natural language description of an image/video.
- Applications:
  - Alt-text generation (from PowerPoint).
  - Generating summaries for Videos (YouTube)
  - Content-based image retrieval.
  - ……

A cat is sitting next to a pine tree, looking up.

Visual Captioning

- Datasets:
  - **MS COCO**: 12,0000 images with 5 free form sentences per image.
  - **Flickr30K**: 30k images collected from Flickr with 158k captions provided by human annotators.
  - **MSVD**: 1970 short videos collected from YouTube. Each video has about 40 English human annotations.
  - ……
- How to evaluate ?
  - There is no "correct" answer for describing an image by a sentence.
  - A predicted sentence is evaluated against a set of human annotated sentences.

Visual Captioning Taxonomy

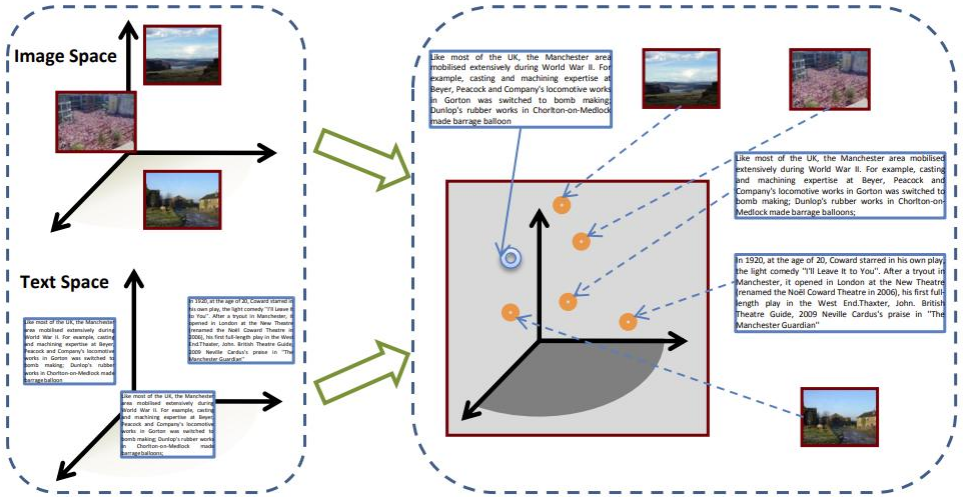


- Commonly used framework: Encoder-Decoder

Taxonomy is from "Recent Advances in Vision-and-Language Research", CVPR 2020 Tutorial

Cross-modal Retrieval

- Support similarity search for multi-modal data, e.g., the retrieval of images in response to a query textual document or vice versa



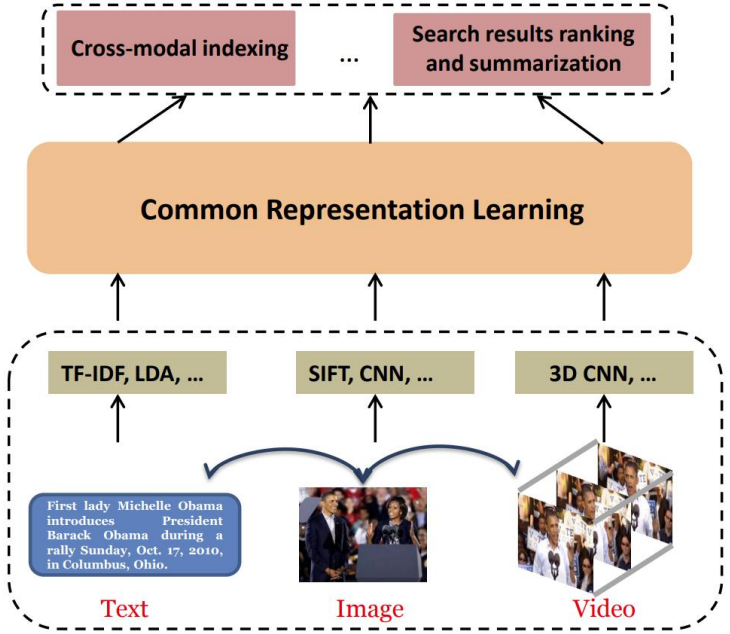


Fig. 3. The general framework of cross-modal retrieval.

Wang, Kaiye, et al. "A comprehensive survey on cross-modal retrieval."

Cross-modal Retrieval

- Datasets
  - Wiki, INRIA-Websearch, Flickr30K, Pascal-VOC, …

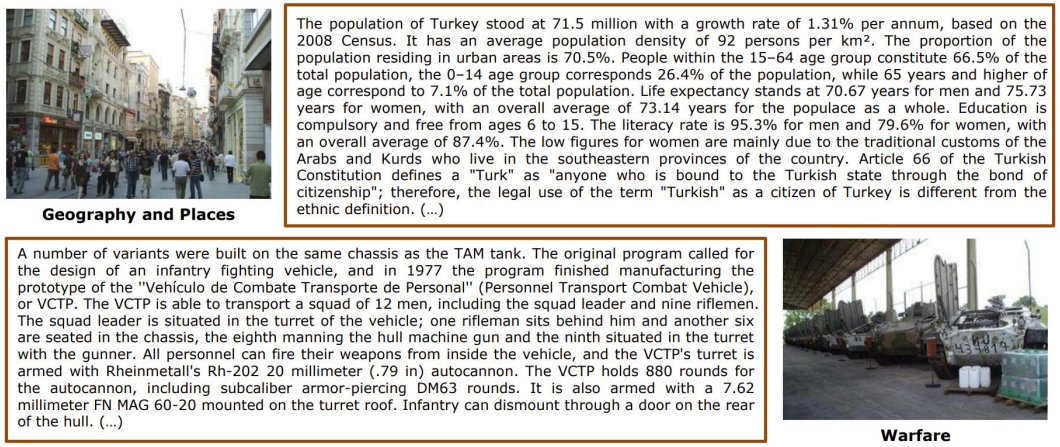


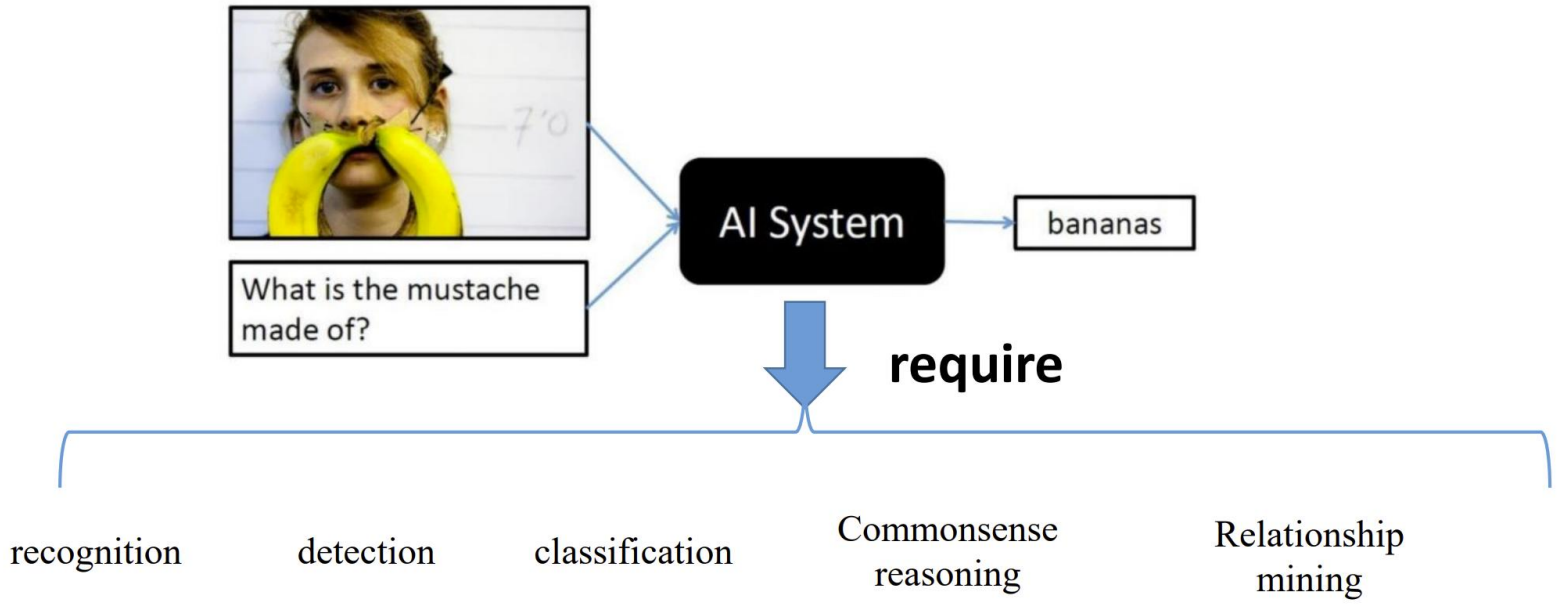
Fig. 10. Two examples on the Wiki dataset. The text is an article describing the content of an image.

Fig. 11. Examples on the Pascal VOC dataset. The tags describing the content of an image are regarded as the text modality.

Wang, Kaiye, et al. "A comprehensive survey on cross-modal retrieval."

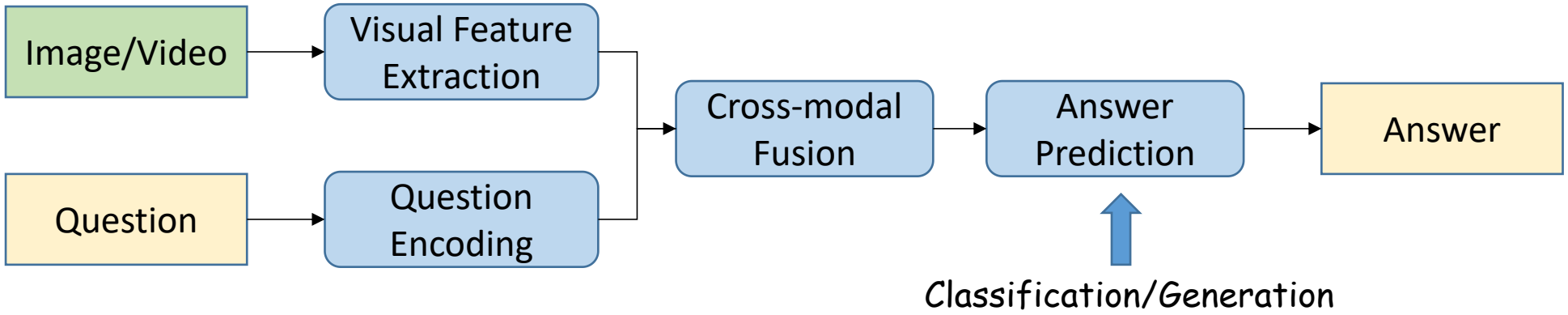
Visual Question Answering

- Given an image and a question (text) about the image/video, the model aims to provide an accurate natural language answer



Visual Question Answering

- A typical framework



From "Recent Advances in Vision-and-Language Research", CVPR 2020 Tutori

Visual Question Answering

- Datasets and Evaluation

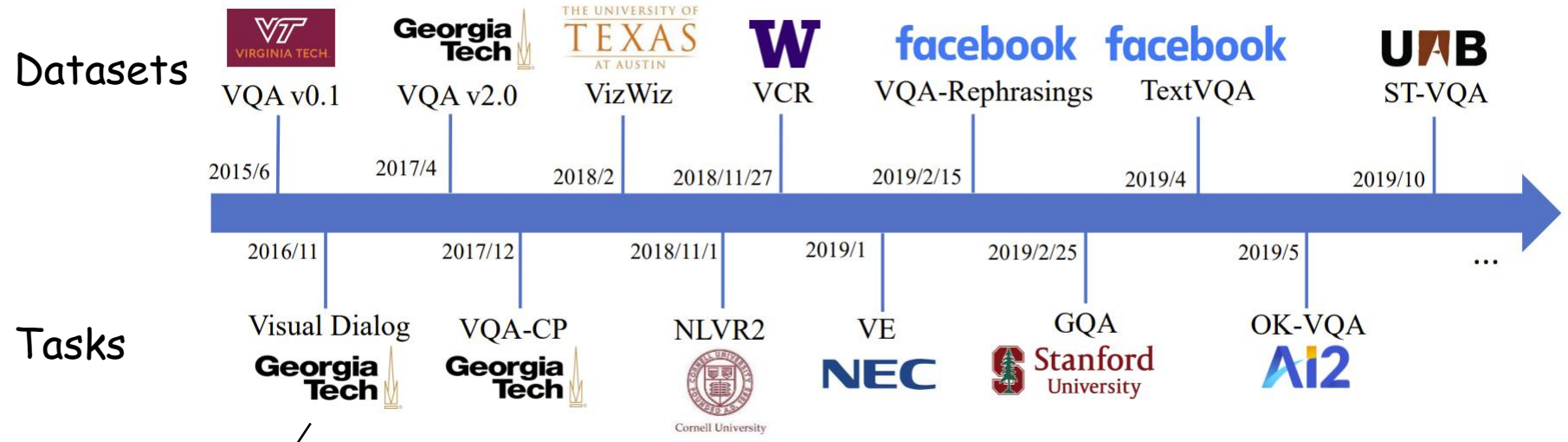
Dataset	Source of images	Number of images	Number of questions	Num. questions / num. images	Num. question categories	Question collection	Average quest. length	Average ans. length	Evaluation metrics
DAQUAR [51]	NYU-Depth V2	1,449	12,468	8.6	4	Human	11.5	1.2	Acc. & WUPS
COCO-QA [63]	COCO	117,684	117,684	1.0	4	Automatic	8.6	1.0	Acc. & WUPS
FM-IQA [22]	COCO	120,360	-	-	-	Human	-	-	Human
VQA-real [3]	COCO	204,721	614,163	3.0	20+	Human	6.2	1.1	Acc. against 10 humans
Visual Genome [41]	COCO	108,000	1,445,322	13.4	7	Human	5.7	1.8	Acc.
Visual7W [100]	COCO	47,300	327,939	6.9	7	Human	6.9	1.1	Acc.
Visual Madlibs [95]	COCO	10,738	360,001	33.5	12	Human	6.9	2.0	Acc.
VQA-abstract [3]	Clipart	50,000	150,000	3.0	20+	Human	6.2	1.1	Acc.
VQA-balanced [98]	Clipart	15,623	33,379	2.1	1	Human	6.2	1.0	Acc.
KB-VQA [78]	COCO	700	2,402	3.4	23	Human	6.8	2.0	Human
FVQA [80]	COCO & ImageNet	1,906	4,608	2.5	12	Human	9.7	1.2	Acc.

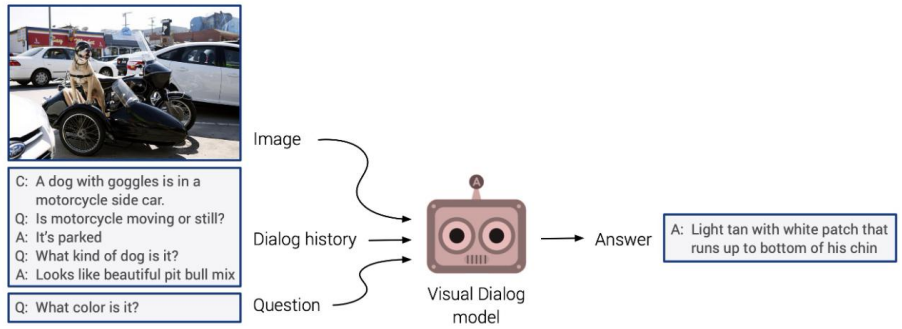
Wu, Qi, et al. "Visual question answering: A survey of methods and datasets."

Other VL Understanding tasks

Datasets

Tasks

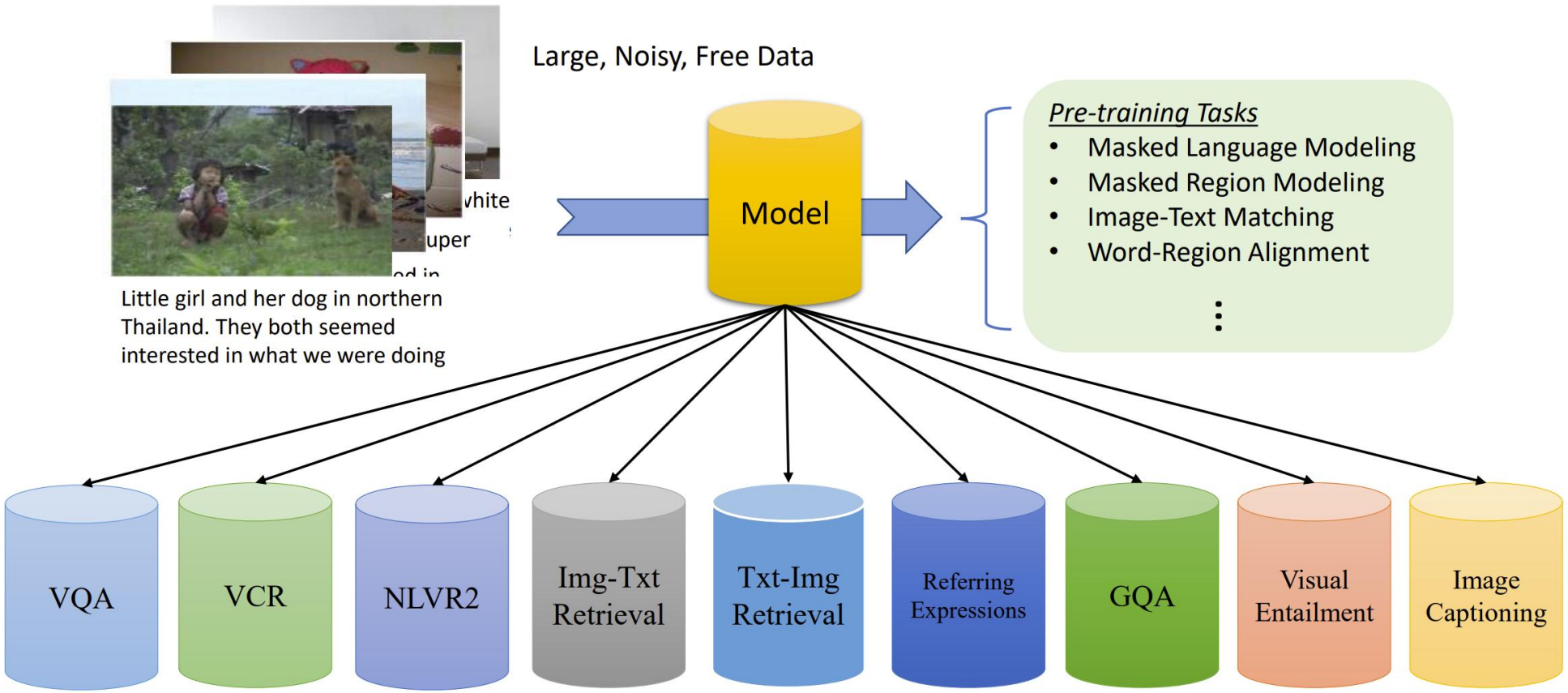




Predict if the caption is True or False

From "Recent Advances in Vision-and-Language Research", CVPR 2020 Tutorial

Self-supervised Learning for VL



From "Recent Advances in Vision-and-Language Research", CVPR 2020 Tutorial

Self-supervised Learning for VL

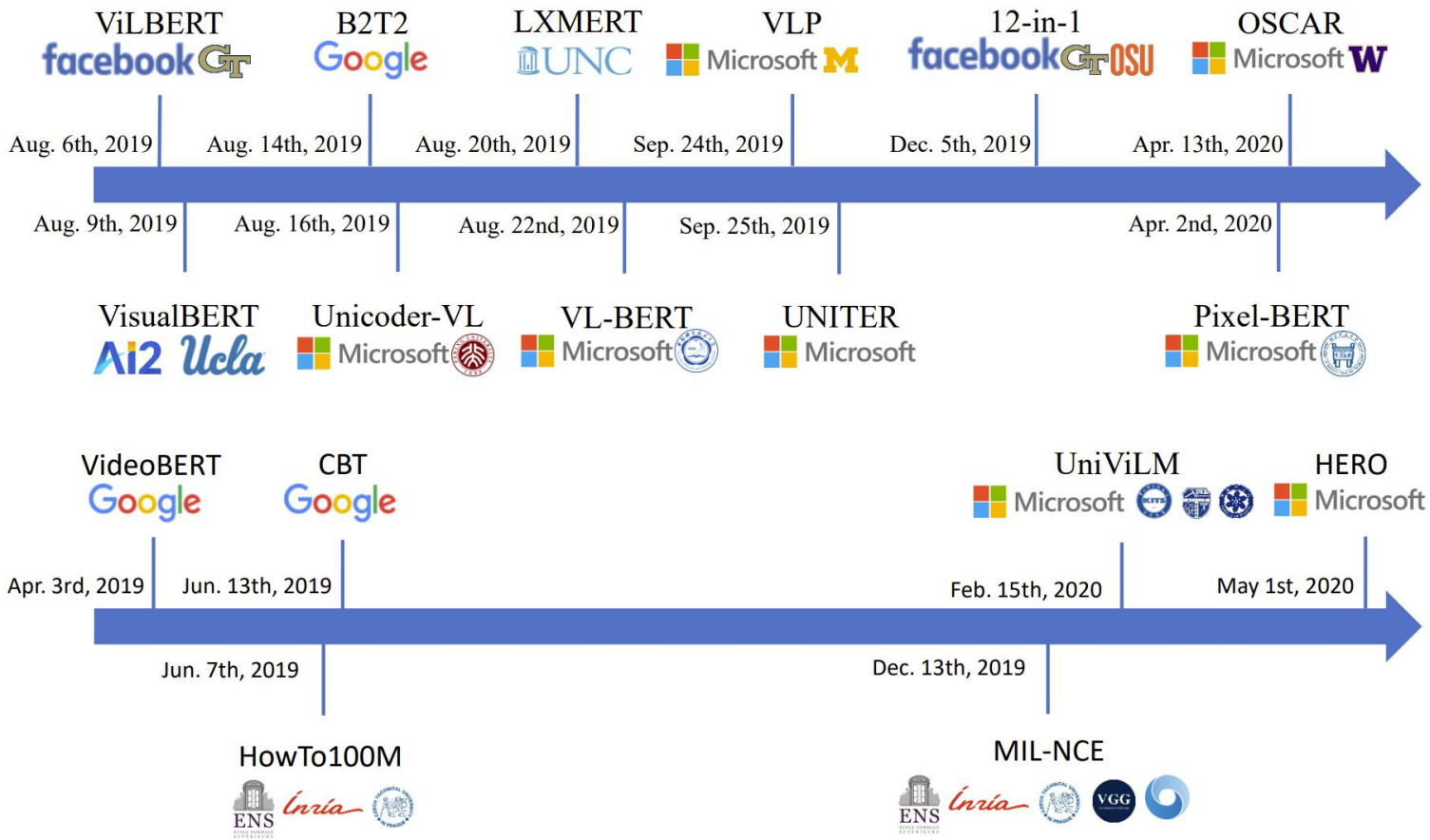


Image Downstream Tasks

- VQA
- VCR
- NLVR2
- Visual Entailment
- Referring Expressions
- Image-Text Retrieval
- Image Captioning

Video Downstream Tasks

- Video QA
- Video-and-Language Inference
- Video Captioning
- Video Moment Retrieval

From "Recent Advances in Vision-and-Language Research", CVPR 2020 Tutorial

Sequence-to-Sequence Modeling (Seq2Seq)

Sequence Data

- Text, image and video are all sequence data

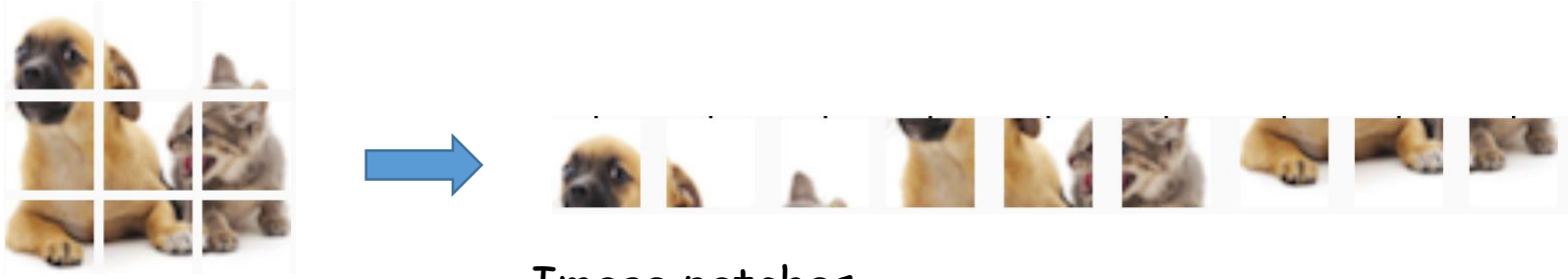


Image patches

How to model sequence data?

Video Frames

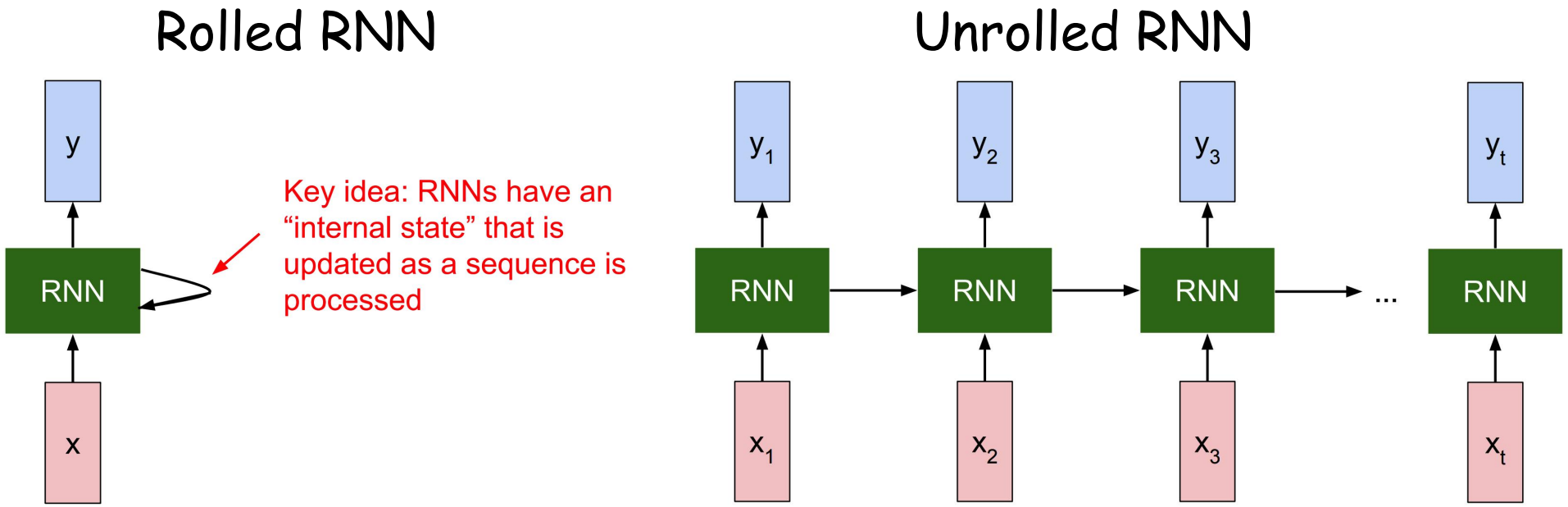
A cat is sitting next to a pine tree. → "A", "cat", "is", "sitting", "next", "to", "a", "pine", "tree", "."

Natural Language

Dog and cat image is retrieved from here

RNN for Sequence Modeling

Recurrent Neural Networks (RNN): connections between nodes form a directed or undirected graph along a temporal sequence



[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf, pp 19-42

Recurrent Neural Network

RNN Hidden State Update

We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

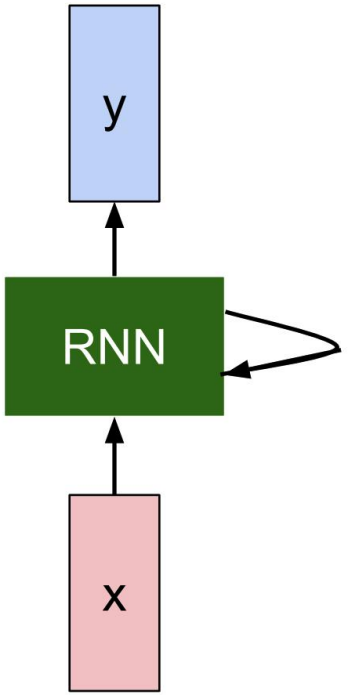
$$h_t = f_W(h_{t-1}, x_t)$$

new state

some function with parameters W

old state

input vector at some time step



[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf, pp 19-42

Recurrent Neural Network

RNN Output Generation

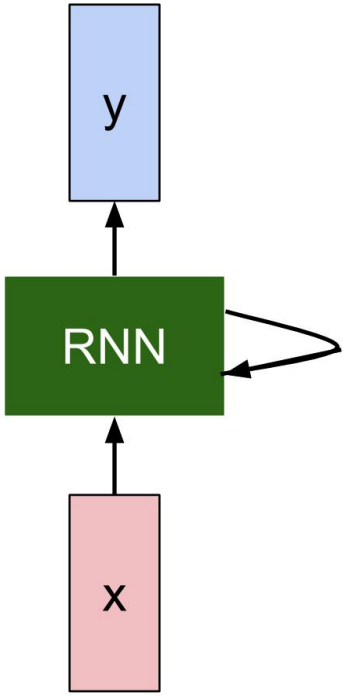
We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

$$y_t = f_{W_{hy}}(h_t)$$

output

new state

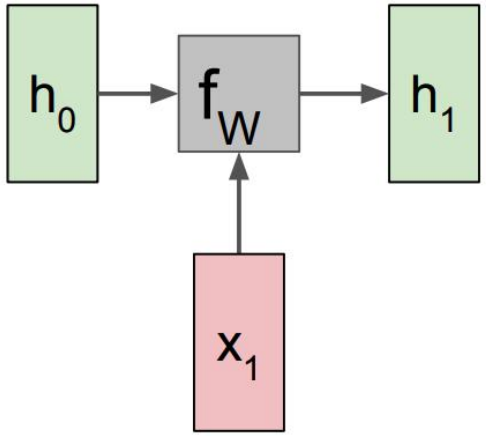
another function with parameters W_o



[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf, pp 19-42

Computational Graph

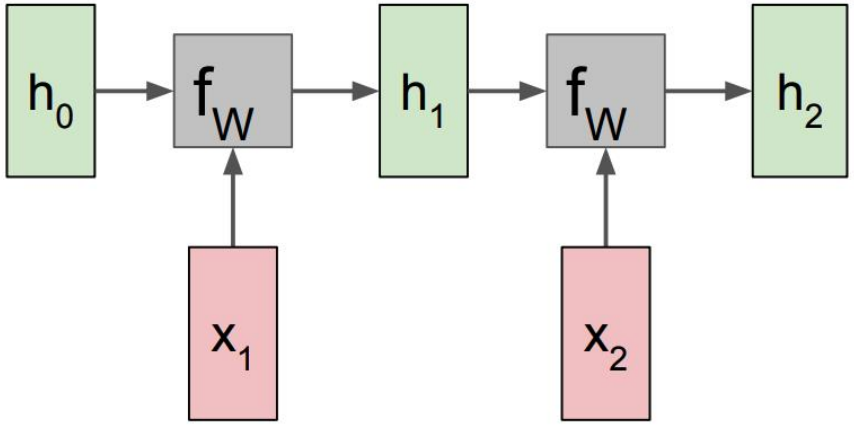
RNN Computational Graph



[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf, pp 19-42

Computational Graph

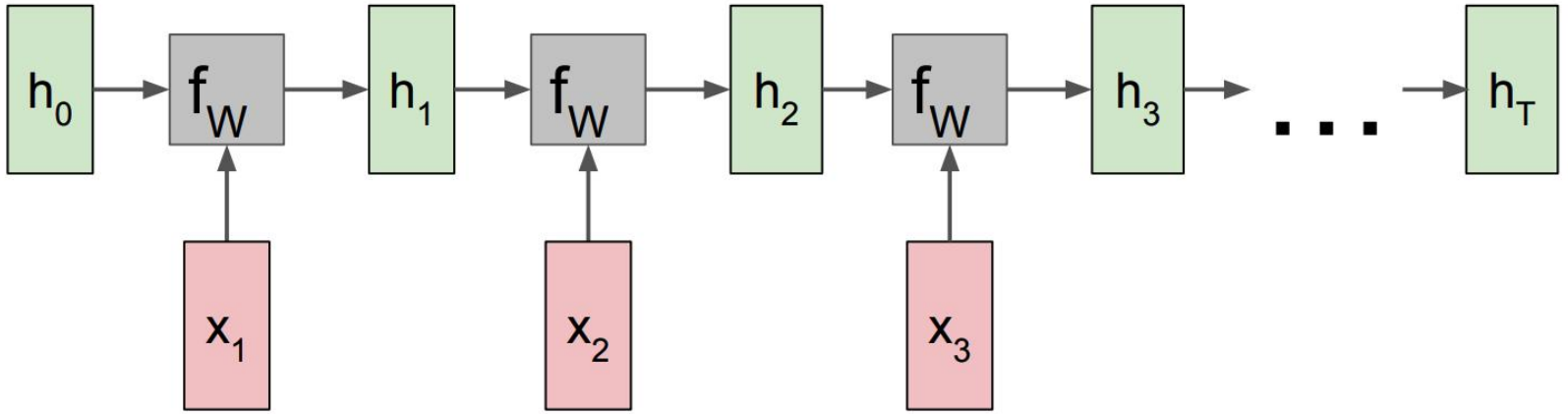
RNN Computational Graph



[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf, pp 19-42

Computational Graph

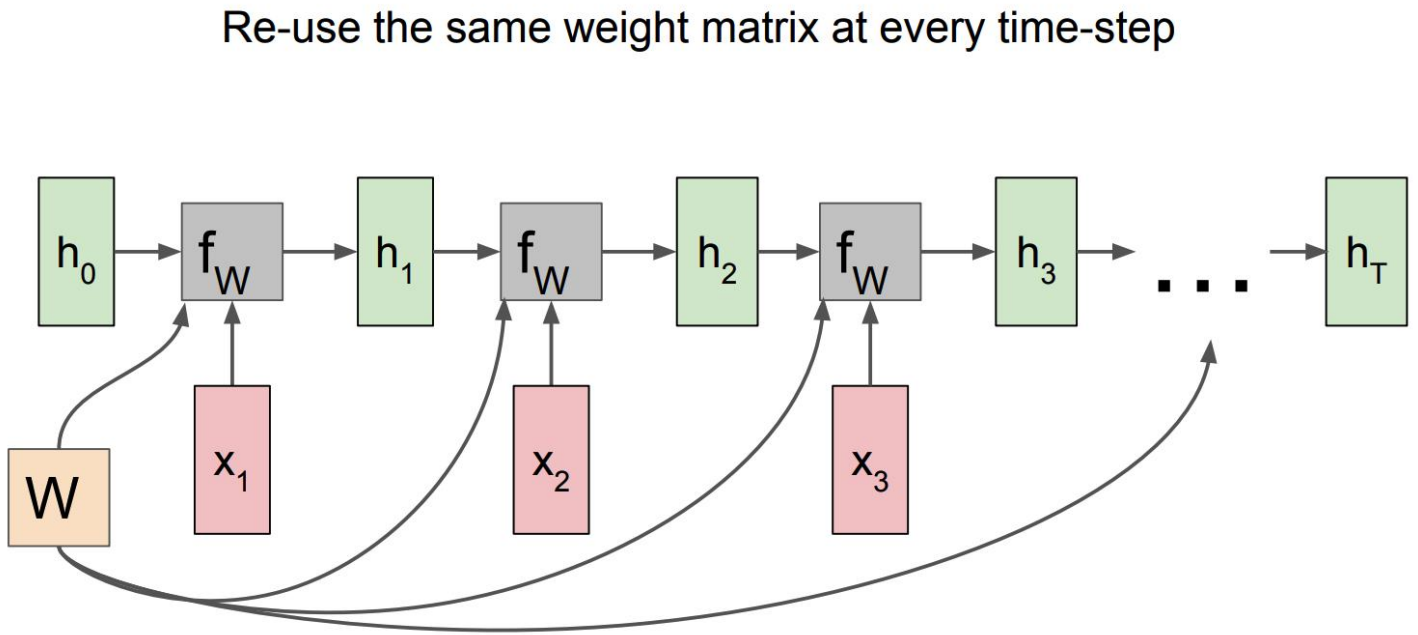
RNN Computational Graph



[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf, pp 19-42

Computational Graph

RNN Computational Graph

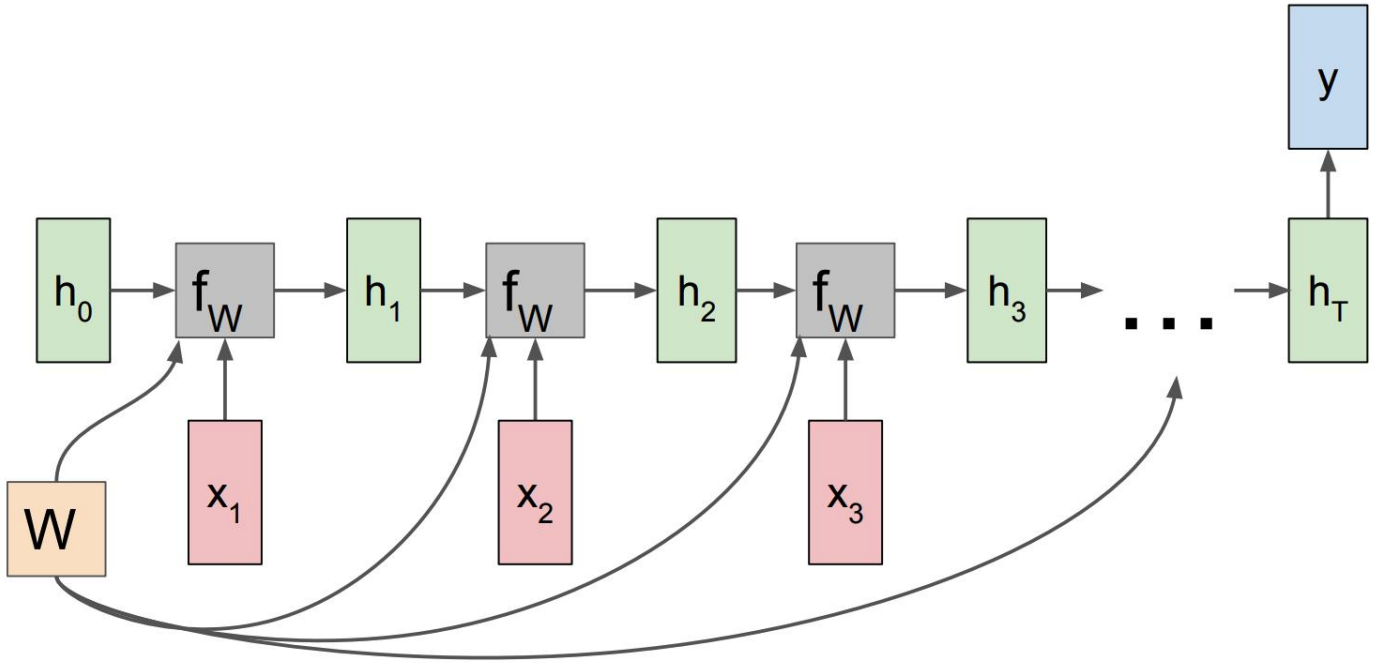


[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf, pp 19-42

Computational Graph

Many input to one output:

- Text classification
- Stock forecasting
- …

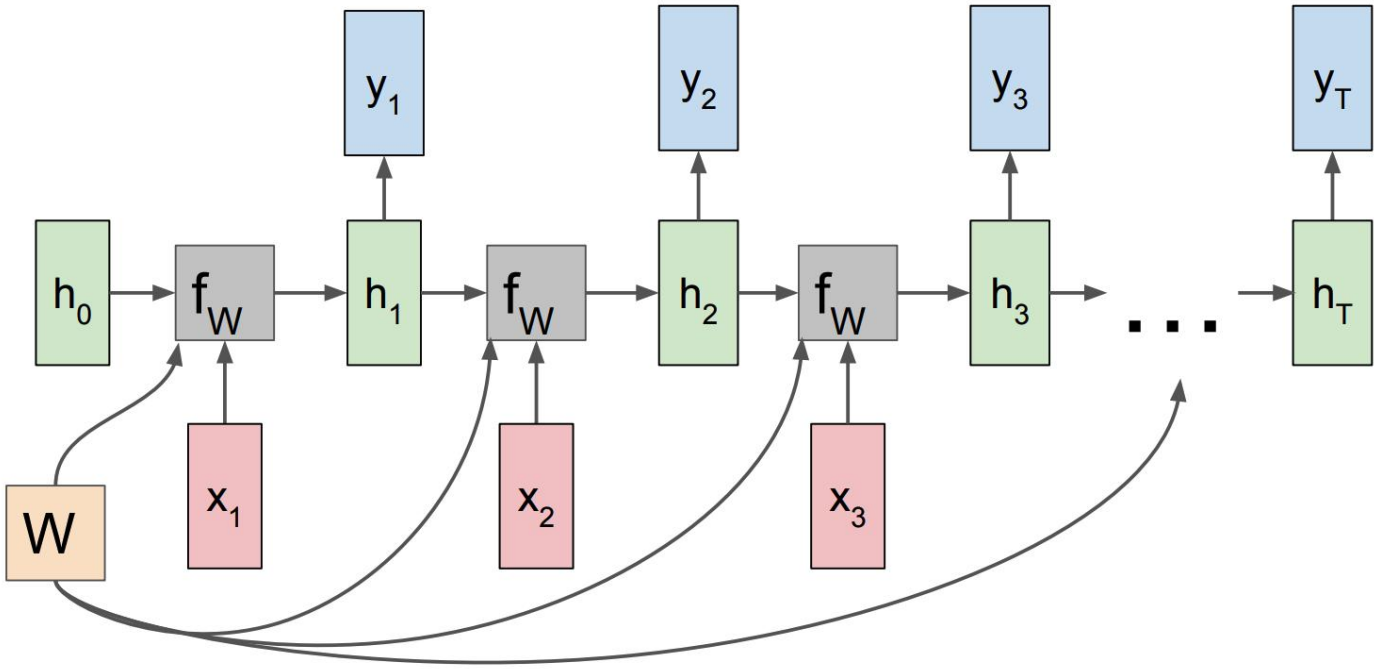


[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf, pp 19-42

Computational Graph

Many input to many output:

- Language translation

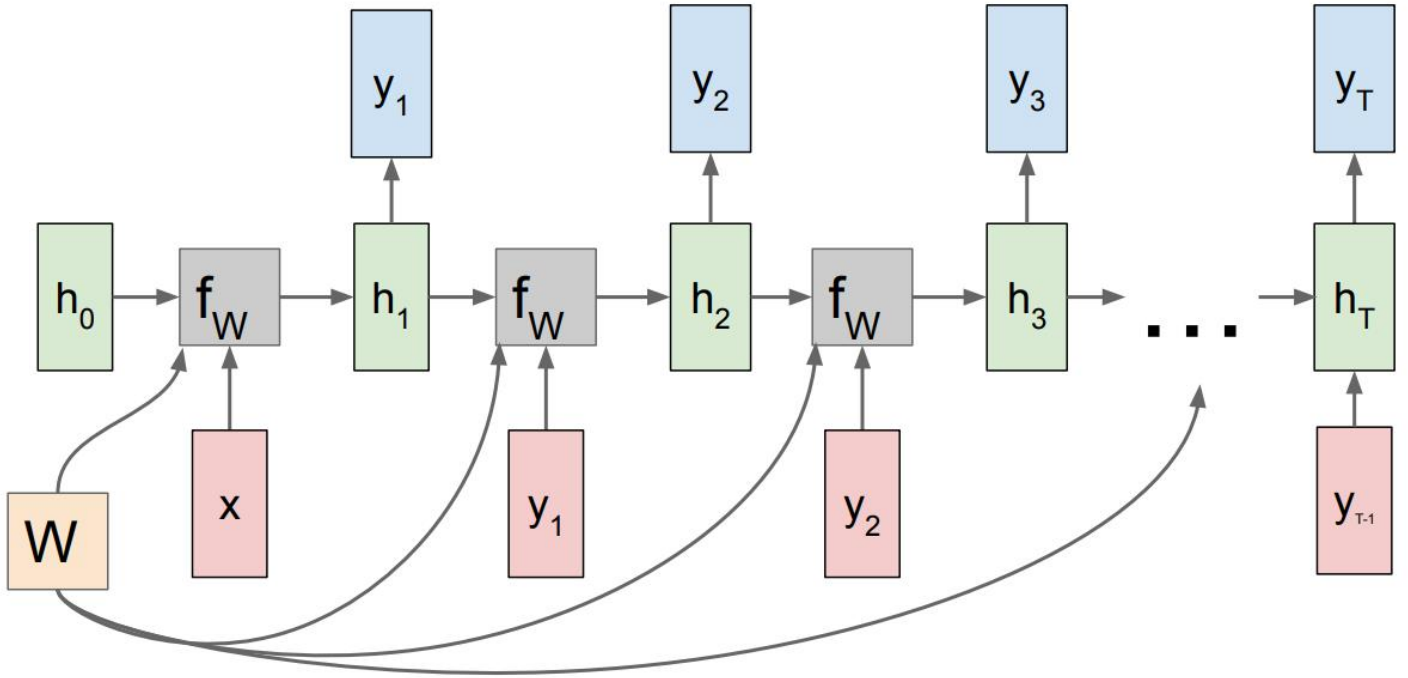


[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf, pp 19-42

Computational Graph

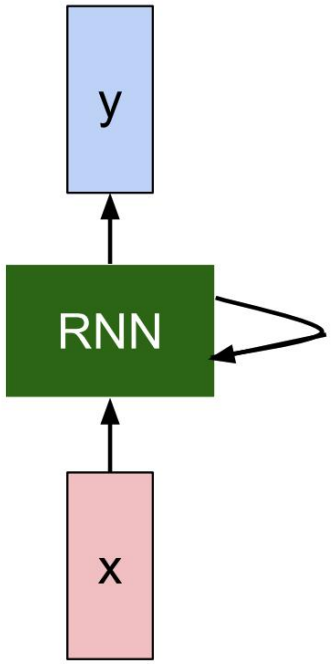
One input to many output:

- Image caption



RNN Summary

The state consists of a single *"hidden"* vector **h**:



$$h_t = f_W(h_{t-1}, x_t)$$

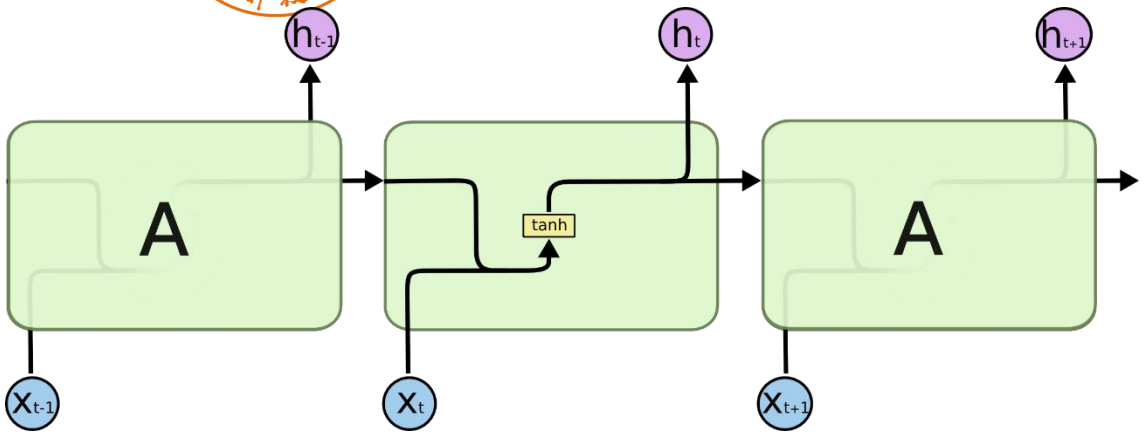
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

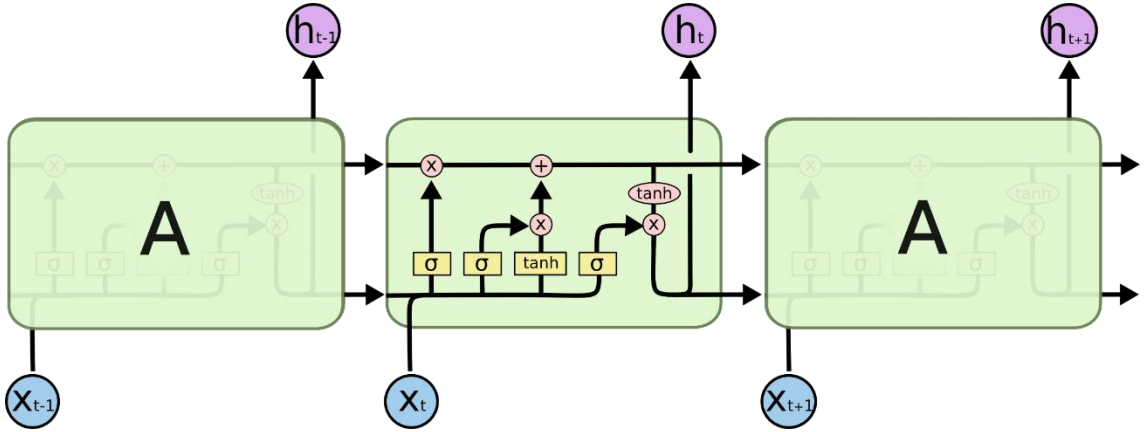
Sometimes called a "Vanilla RNN" or an "Elman RNN" after Prof. Jeffrey Elman

[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf, pp 19-42

Long Short-Term Memory (LSTM)



Repeating Module in a standard RNN



Repeating Module in a LSTM

Update hidden state:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

Forget gate: $f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$

Input gate: $i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$

Output gate: $o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$

Cell memory: $\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$

Update cell memory: $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$

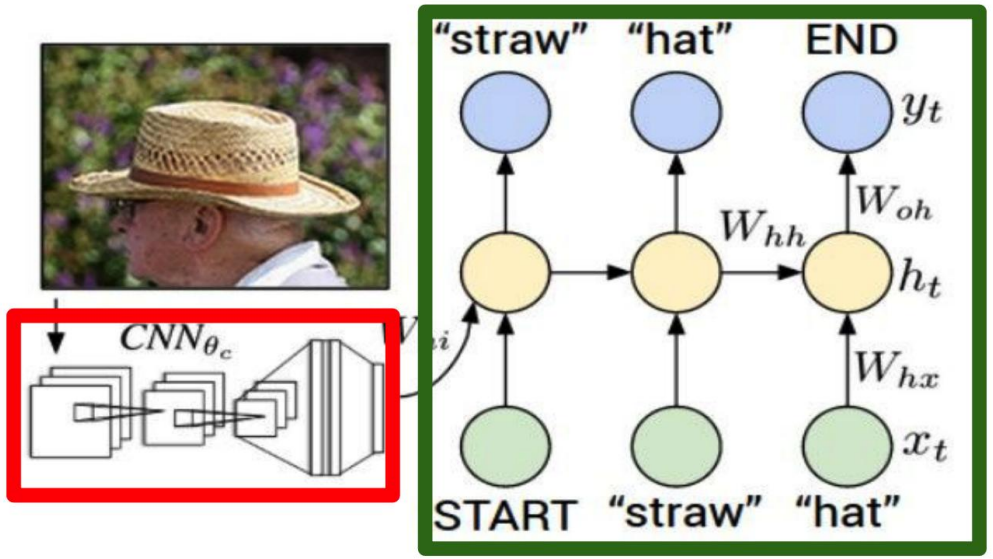
Get the hidden state: $h_t = o_t * \tanh\left(C_t\right)$

[1] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

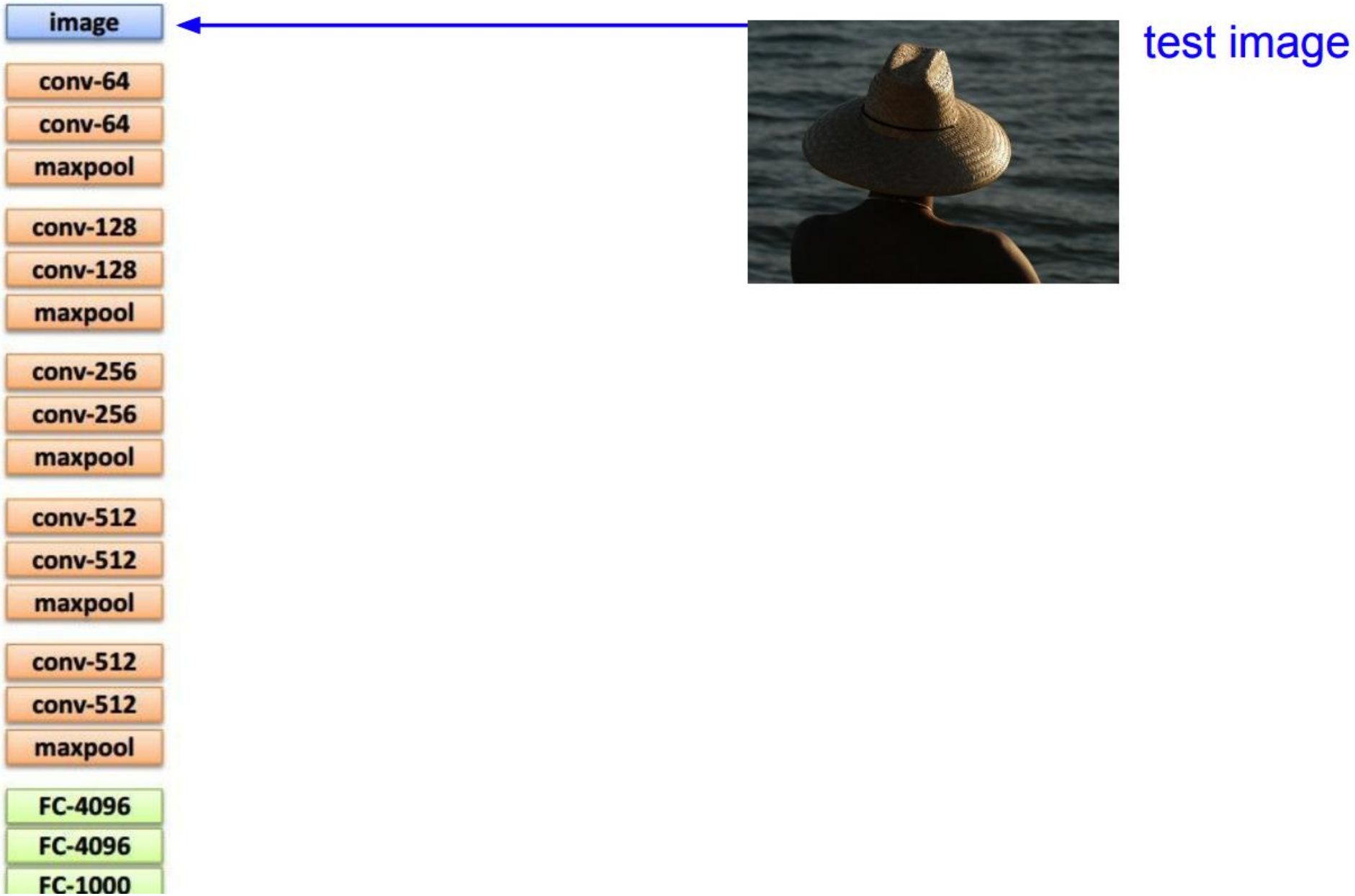
Encoder-Decoder for Image Captioning

Encoder-Decoder

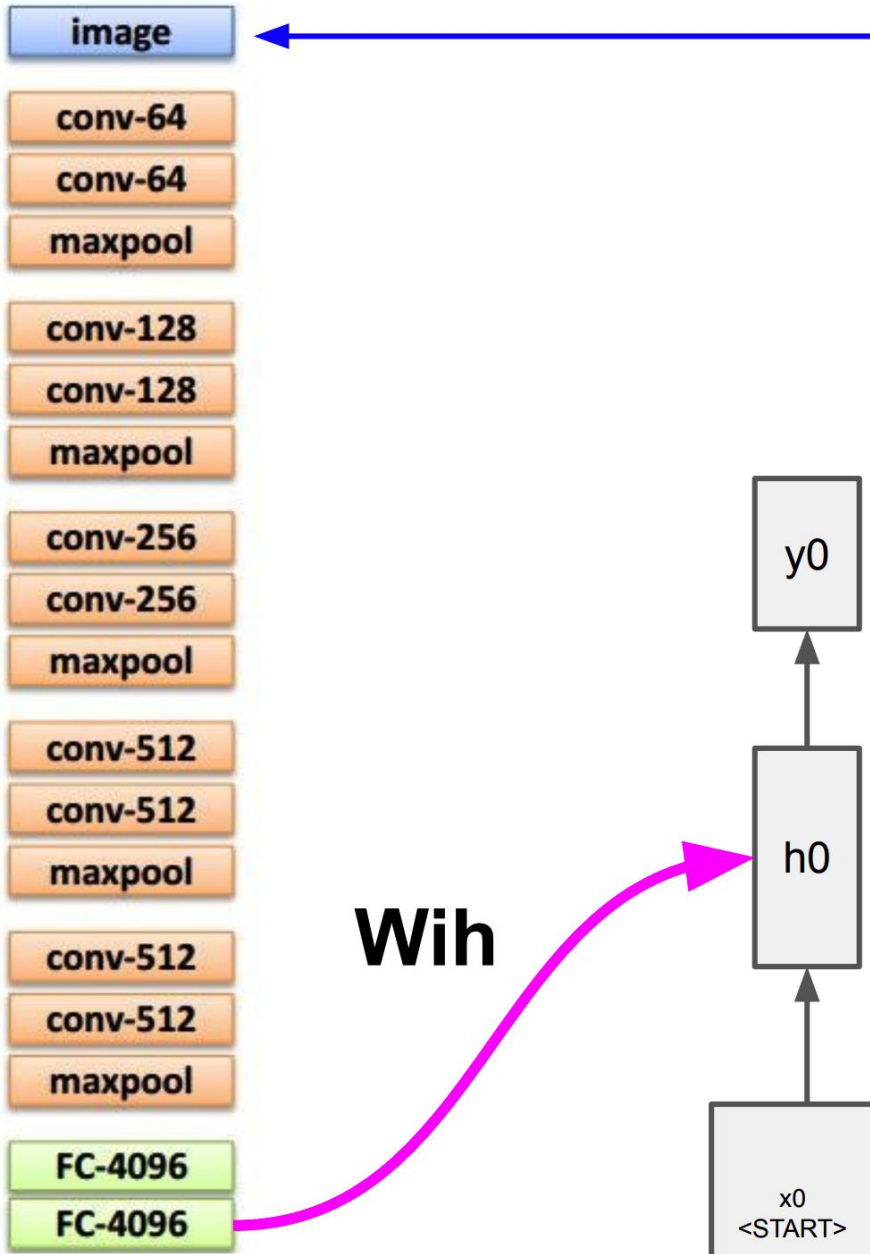
- **CNN Encoder** for summarizing the visual data into a vector
- **RNN Decoder** for generating a word sequence.



[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf



[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf



test image

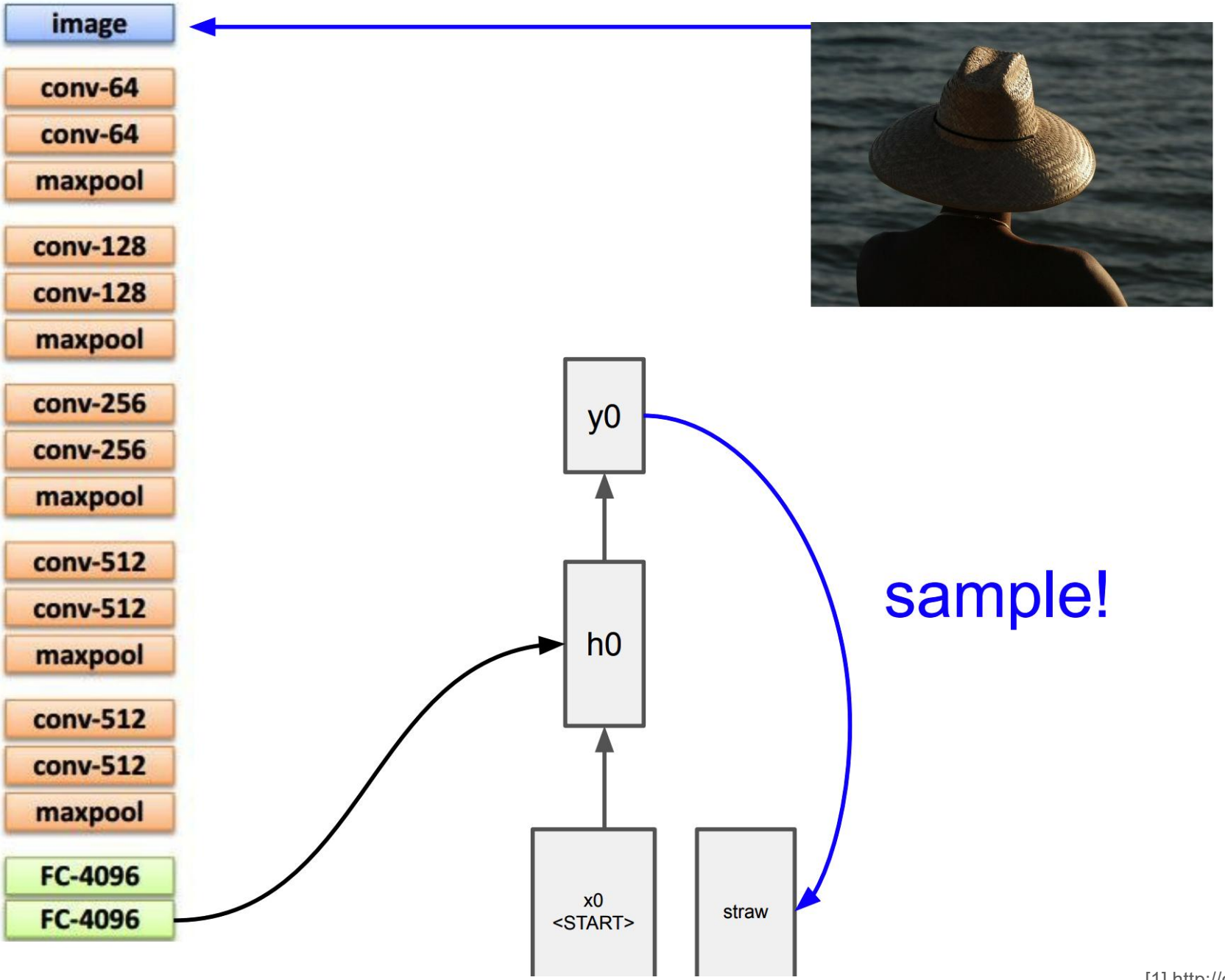
before:

$$h = \tanh(W_{xh} * x + W_{hh} * h)$$

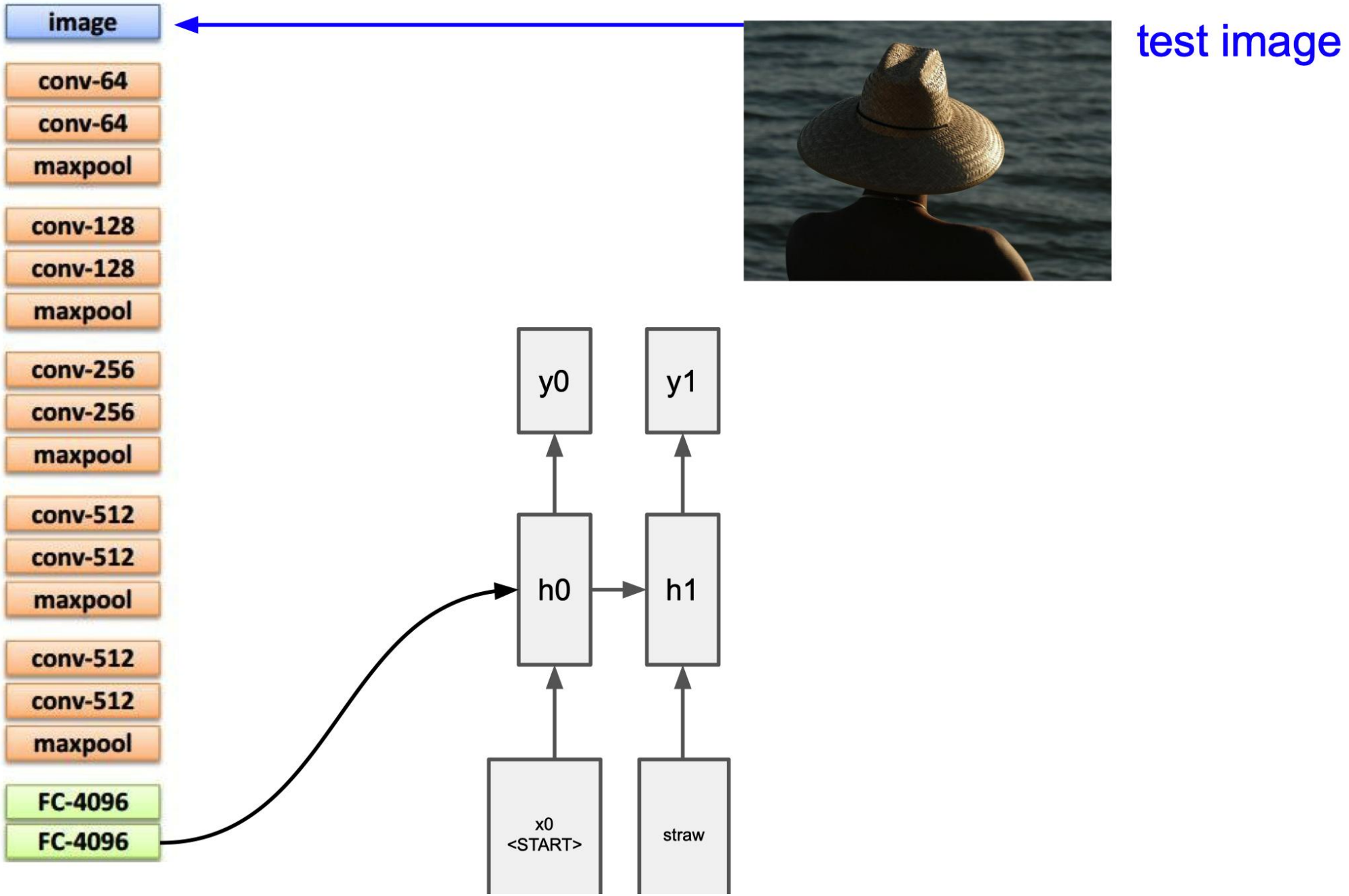
now:

$$h = \tanh(W_{xh} * x + W_{hh} * h + \mathbf{W_{ih} * v})$$

[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf

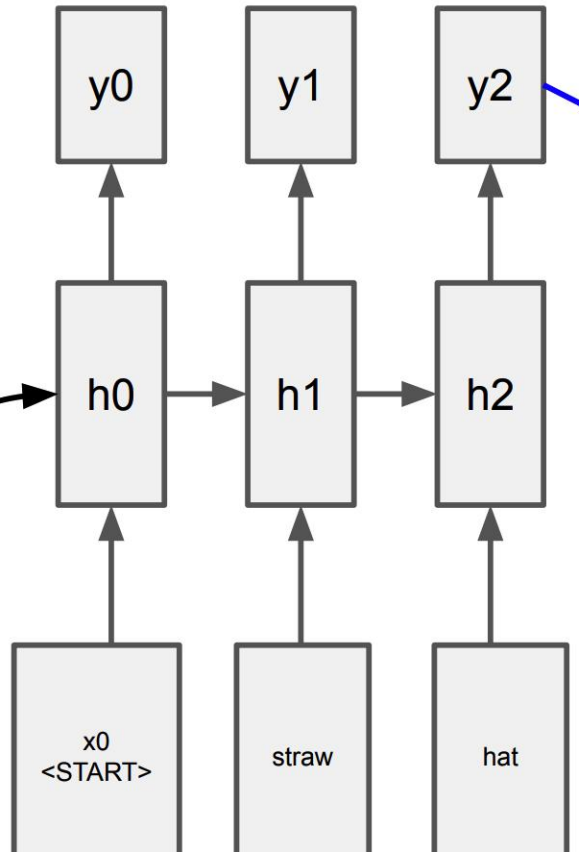


[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf



[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf

test image



sample
<END> token
=> finish.

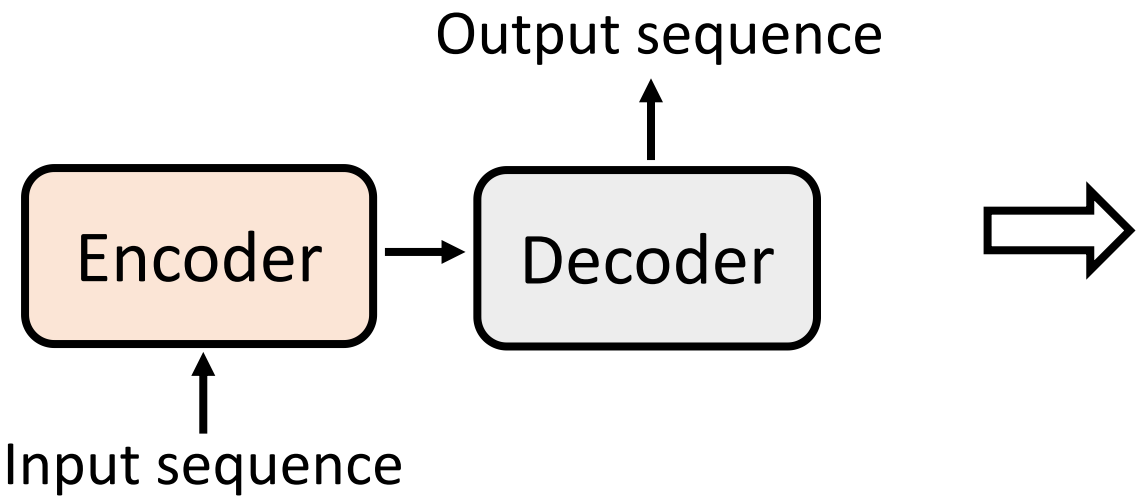
[1] http://cs231n.stanford.edu/slides/2021/lecture_10.pdf

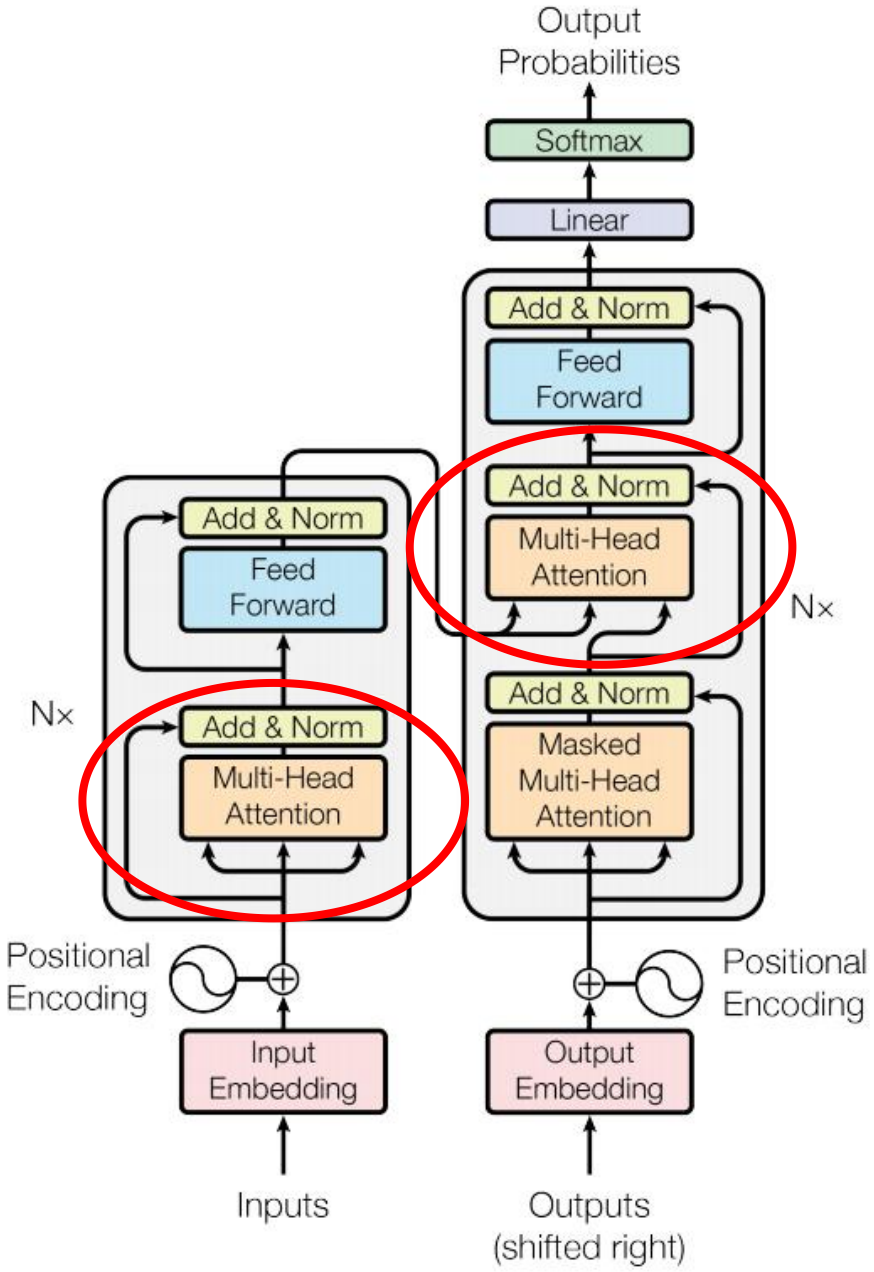
Transformer: Beyond CNNs and RNNs

- Overview
- Self-attention
- Encoder-decoder architecture
- Applications

Transformer

- **Sequence-to-sequence (Seq2seq)**
- **Encoder and Decoder Stacks**

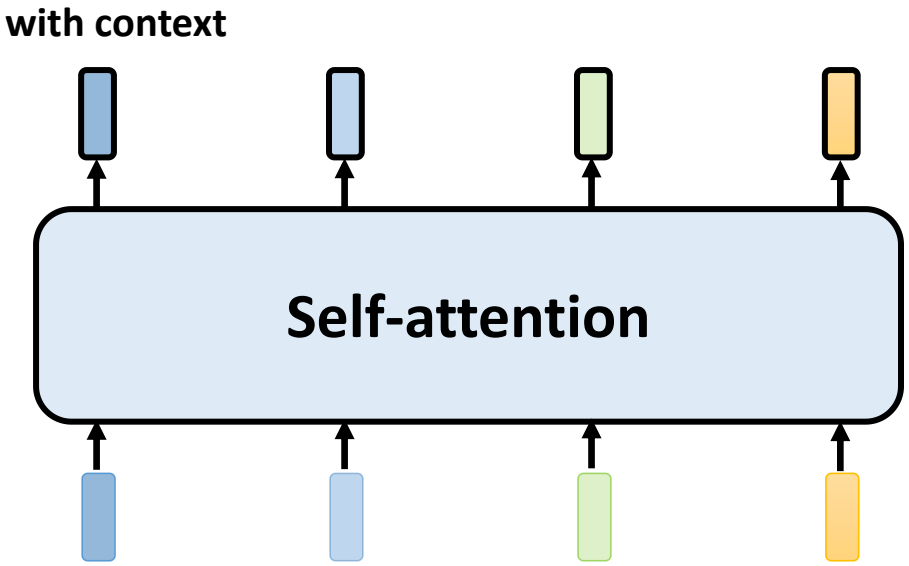


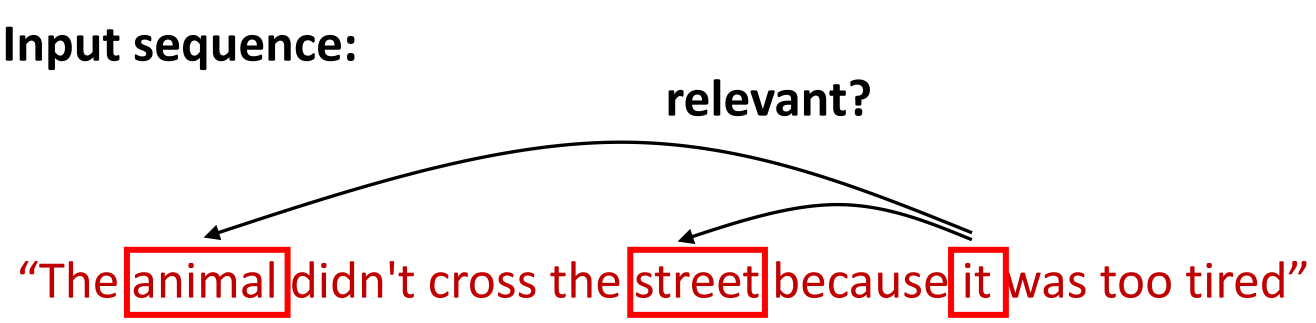


[Attention is all you need.]

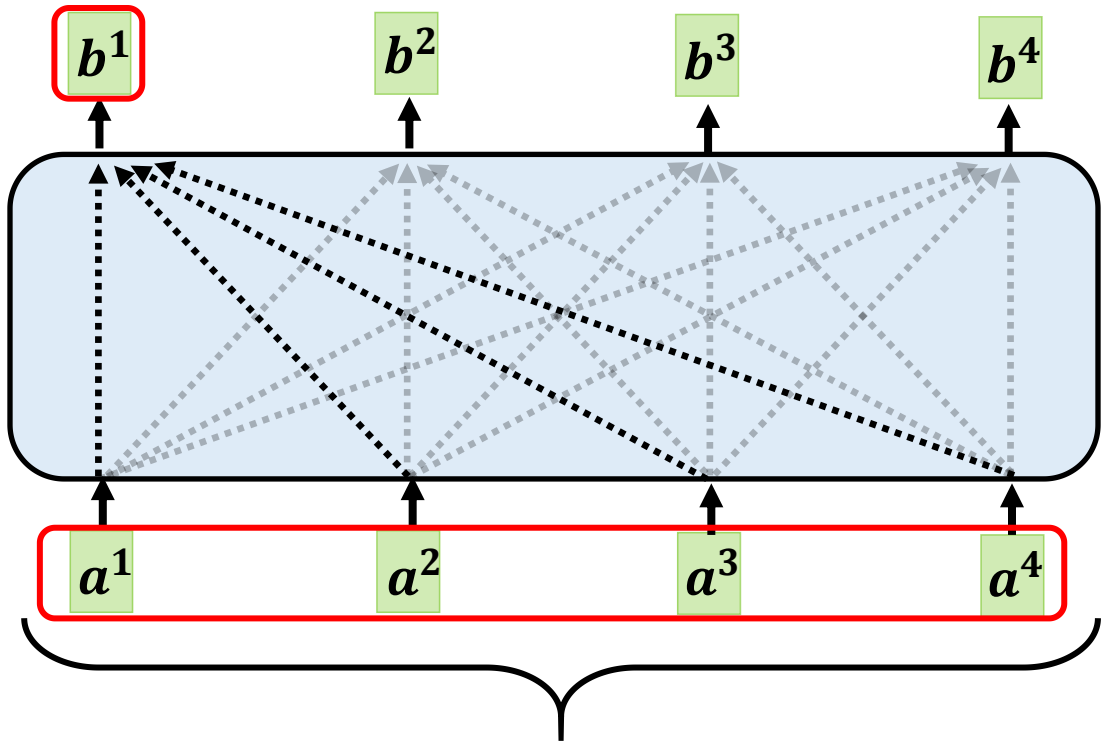
Self-Attention

- How to consider the context of the whole sequence?

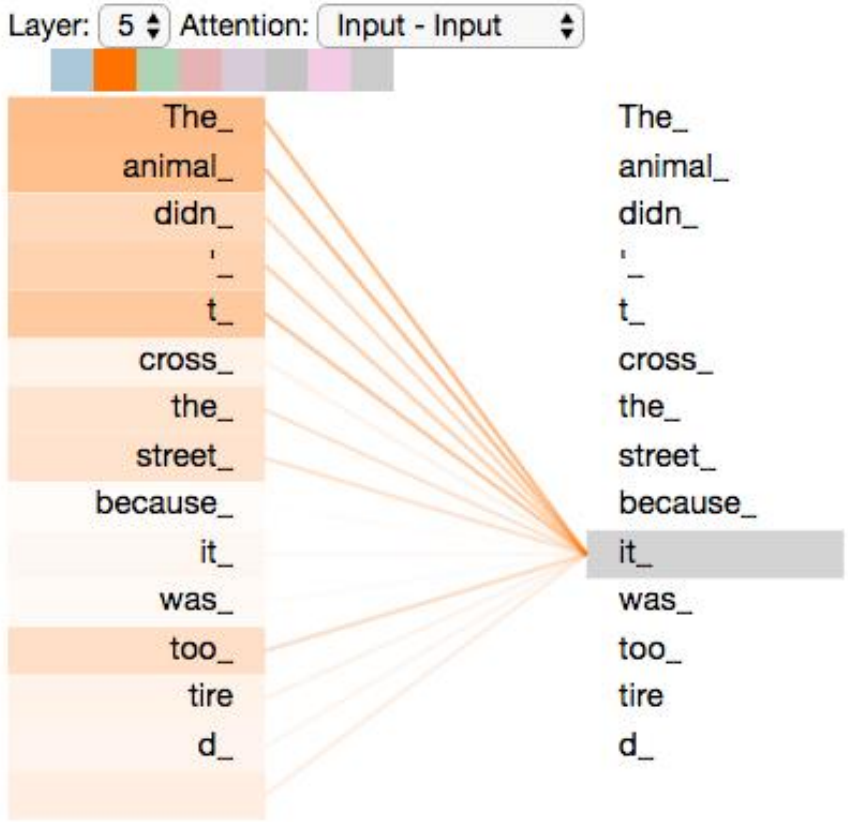




Self-Attention



Can be either input or a hidden layer



Tensor2Tensor notebook

Self-Attention

- **Calculate self-attention using vectors**

$$b^1 = \sum_i \alpha'_{1,i} v^i$$

Extract information based on attention scores

Query: $q^i = W^q a^i$

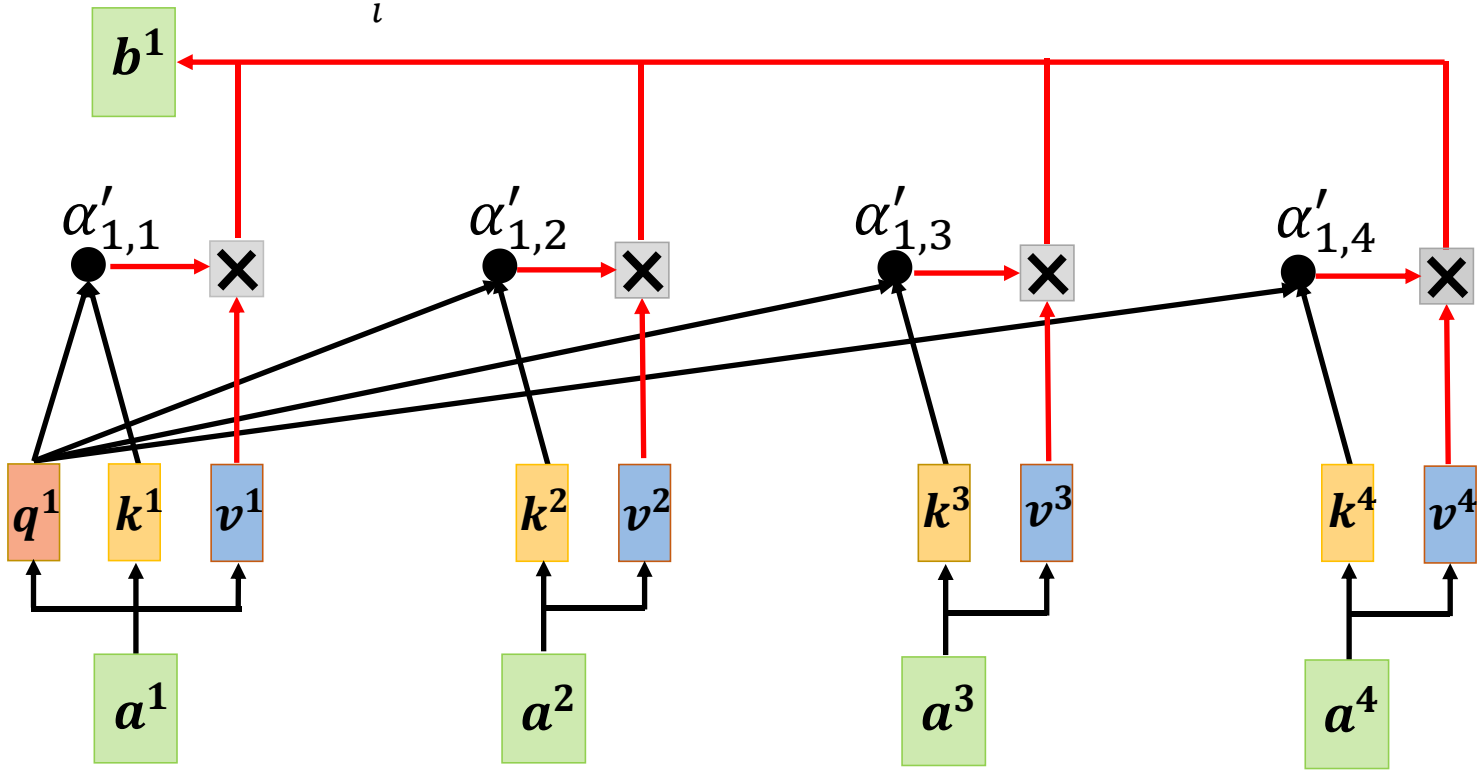
Key: $k^i = W^k a^i$

Value: $v^i = W^v a^i$

$$\alpha_{1,i} = \frac{q^1 \cdot k^i}{\sqrt{d_k}}$$

Dot-product

$$\alpha'_{1,i} = \text{Softmax}(\alpha_{1,i}) = \frac{\exp(\alpha_{1,i})}{\sum_j \exp(\alpha_{1,j})}$$

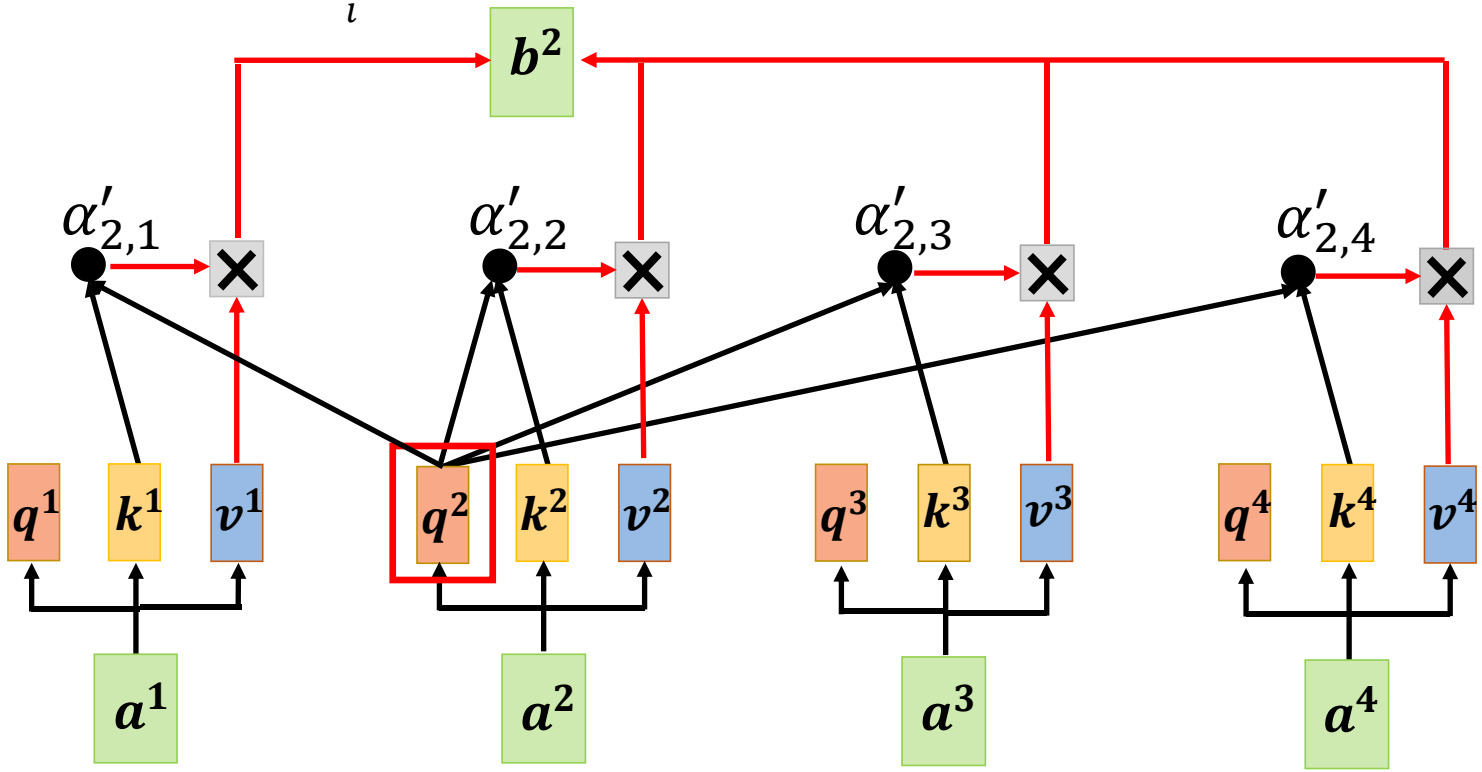


https://speech.ee.ntu.edu.tw/~hylee/ml/ml2021-course-data/self_v7.pdf

Self-Attention

- Calculate self-attention using vectors

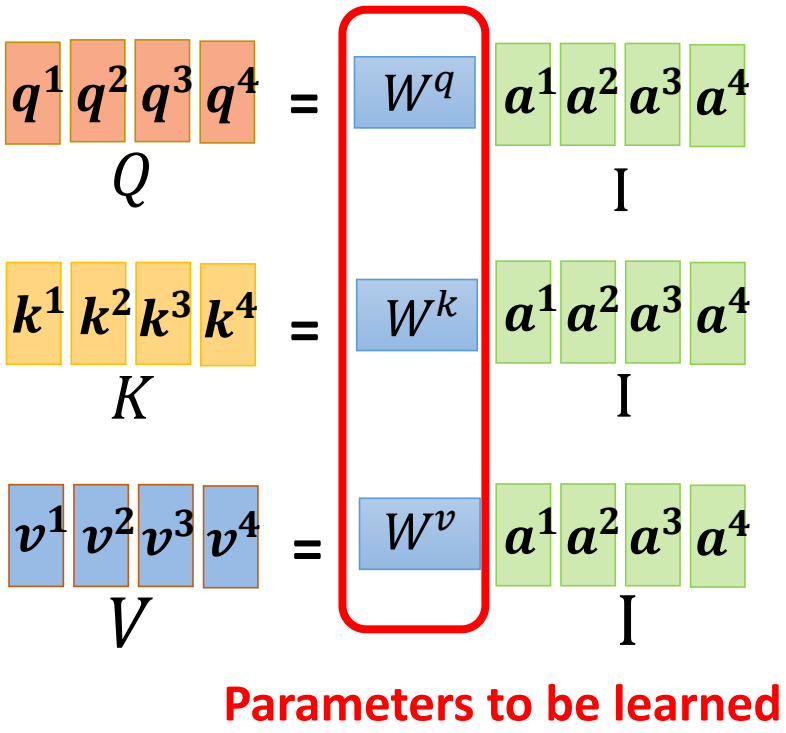
$$b^2 = \sum_i \alpha'_{2,i} v^i$$



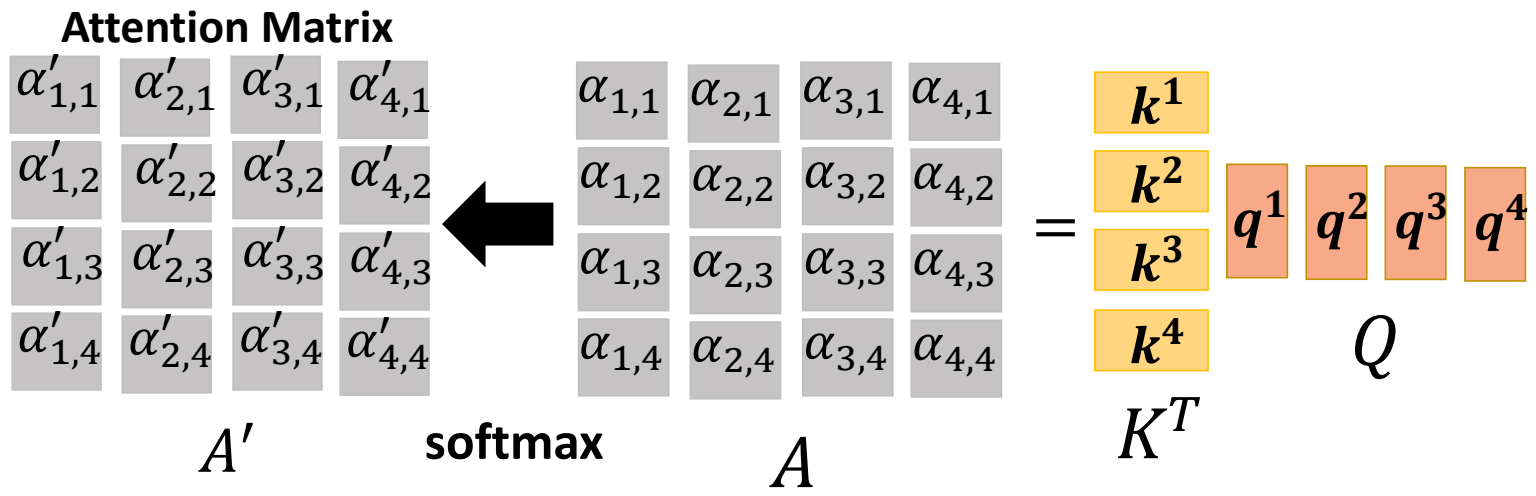
https://speech.ee.ntu.edu.tw/~hylee/ml/ml2021-course-data/self_v7.pdf

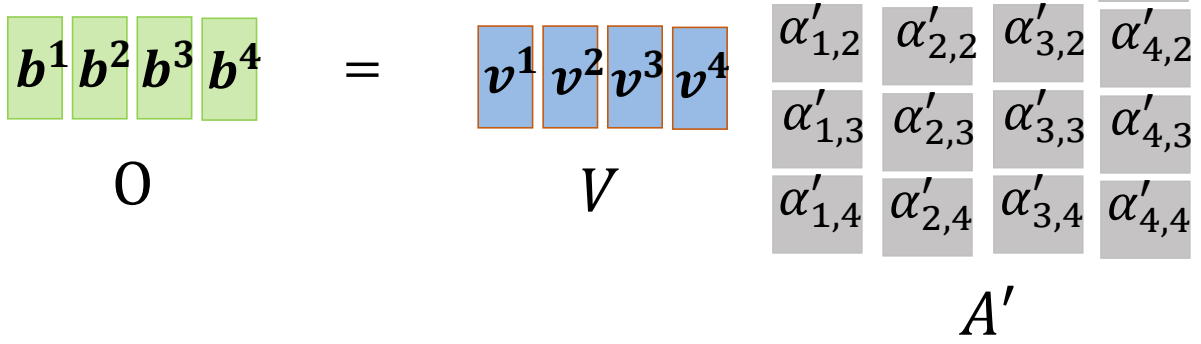
Self-Attention

- **Matrix Calculation**



Parameters to be learned





https://speech.ee.ntu.edu.tw/~hylee/ml/ml2021-course-data/self_v7.pdf

Self-Attention

- **Matrix Calculation**

$$Q = W^q I$$
$$K = W^k I$$
$$V = W^v I$$

$$A' \leftarrow \text{softmax}(A) = \frac{Q \cdot K^T}{\sqrt{d_k}}$$

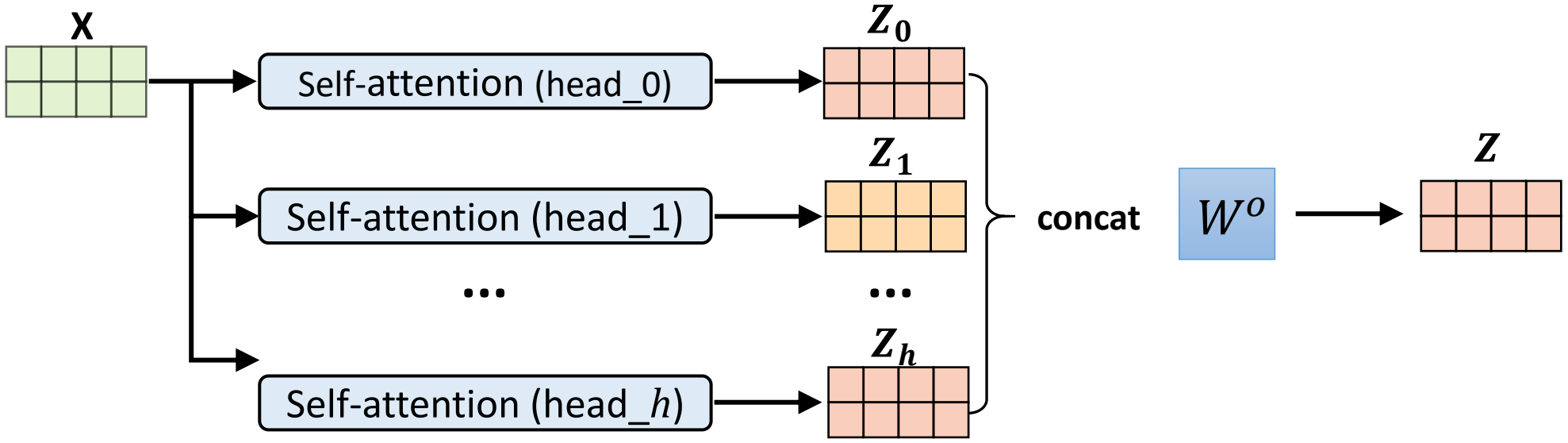
$$O = V A'$$

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

https://speech.ee.ntu.edu.tw/~hylee/ml/ml2021-course-data/self_v7.pdf

Multi-head Self-Attention

- Allows the model to jointly attend to information from different representation subspaces at different positions.
- Capture different types of relevance.

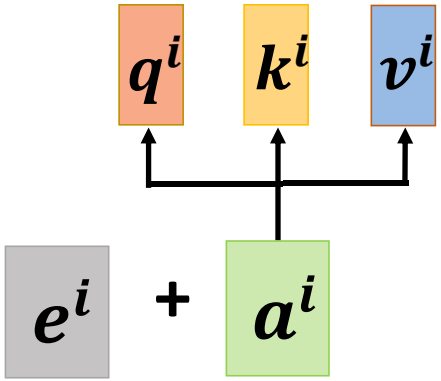


$$\mathbf{MultiHead}(Q, K, V) = \mathbf{Concat}(\mathbf{head_1}, \ldots, \mathbf{head_h})W^o$$

$$\mathbf{where\ head_i} = \mathbf{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Position Encoding

- No position information in self-attention.
- Each position has a unique positional vector e^i

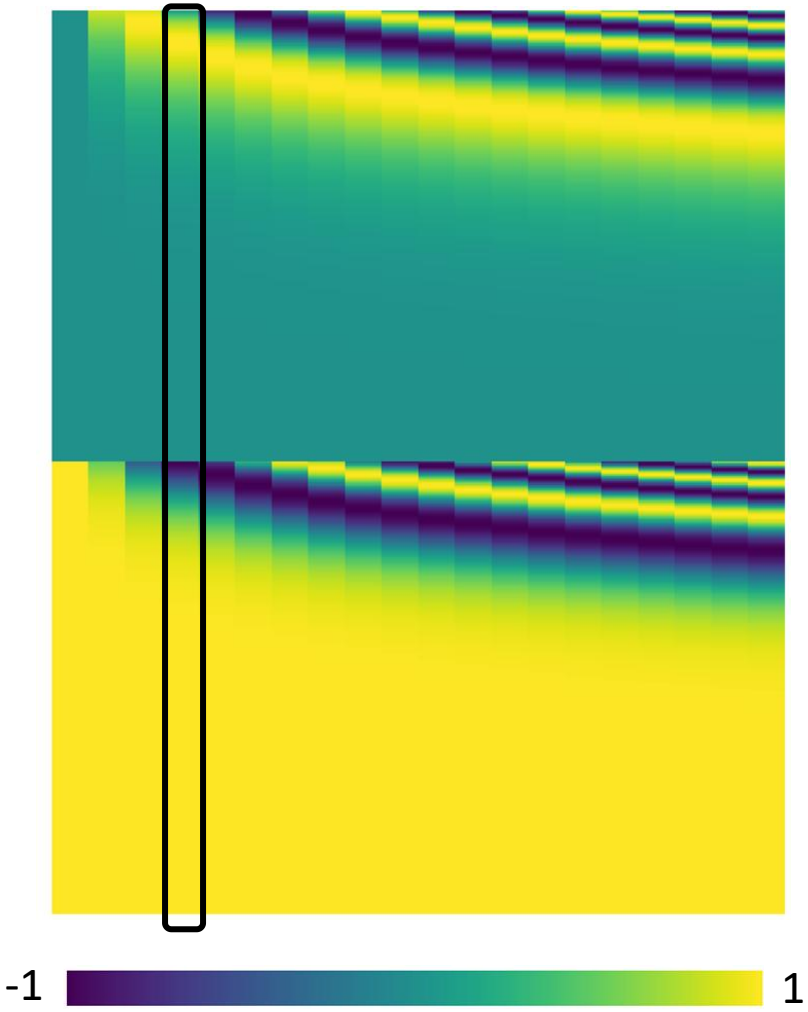


$$PE_{(pos,\,2i)} = sin(pos/1000^{2i/d_{model}})$$

$$PE_{(pos,\,2i+1)} = cos(pos/1000^{2i/d_{model}})$$

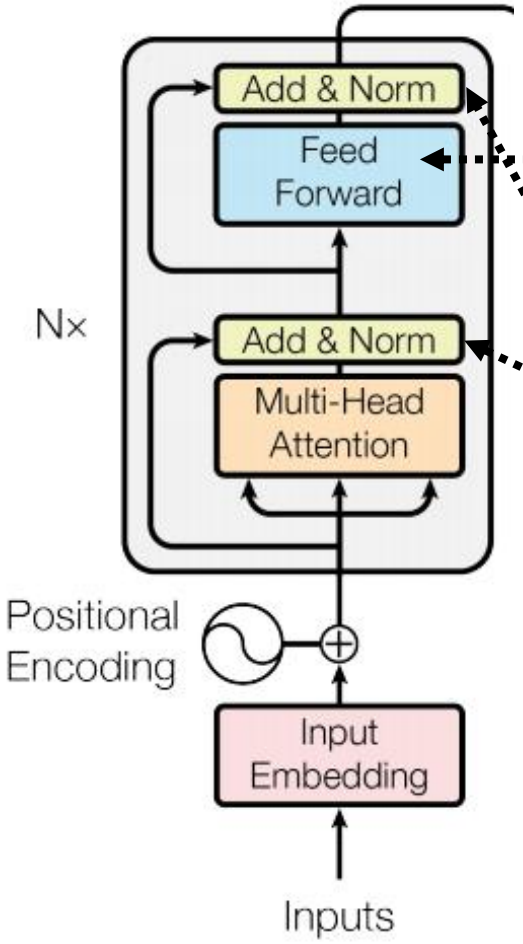
d_{model}: Dimension of the output embedding space
pos: Position of an object in the input sequence

Each column represents a positional vector e^i



https://speech.ee.ntu.edu.tw/~hylee/ml/ml2021-course-data/self_v7.pdf

Transformer Encoder



Fully connected feed-forward network

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$$

(two linear transformations with a ReLU activation)

Residual connection

+ **Layer normalization**

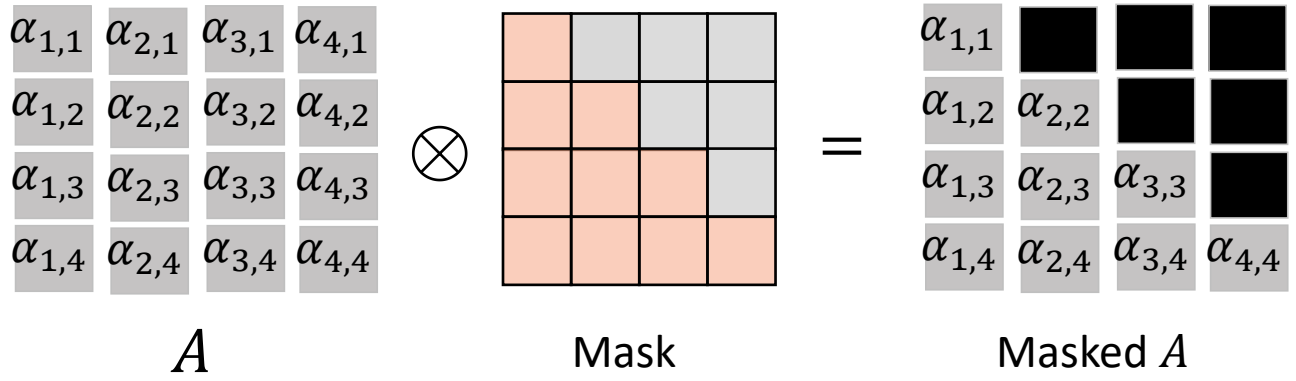
$$x'_i = \frac{x_i - m}{\sigma}$$

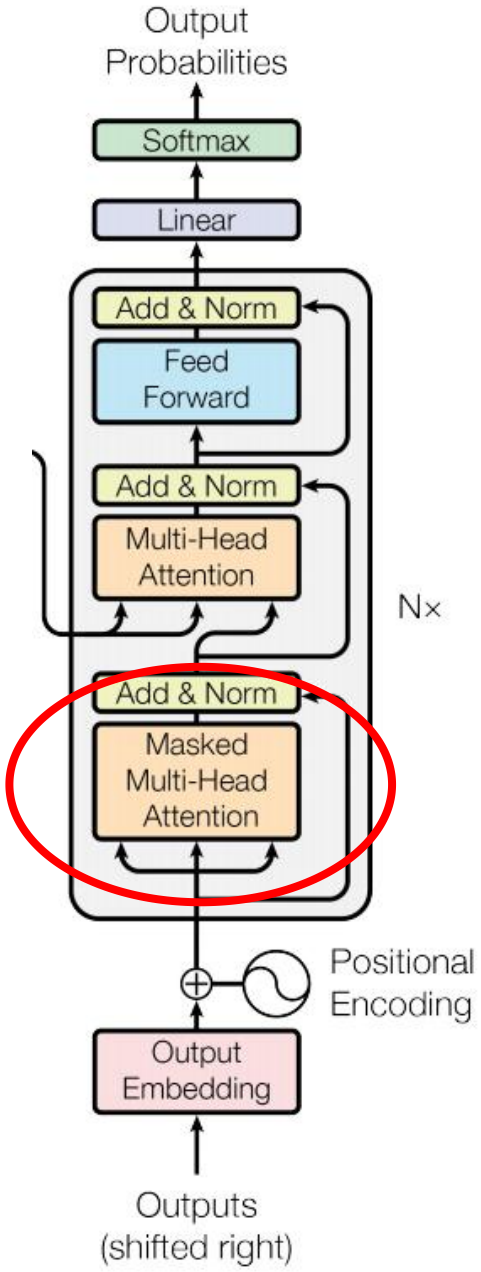
(**mean** m, standard **deviation** σ)
https://arxiv.org/abs/1607.06450

Transformer Decoder

Masked multi-head self-attention

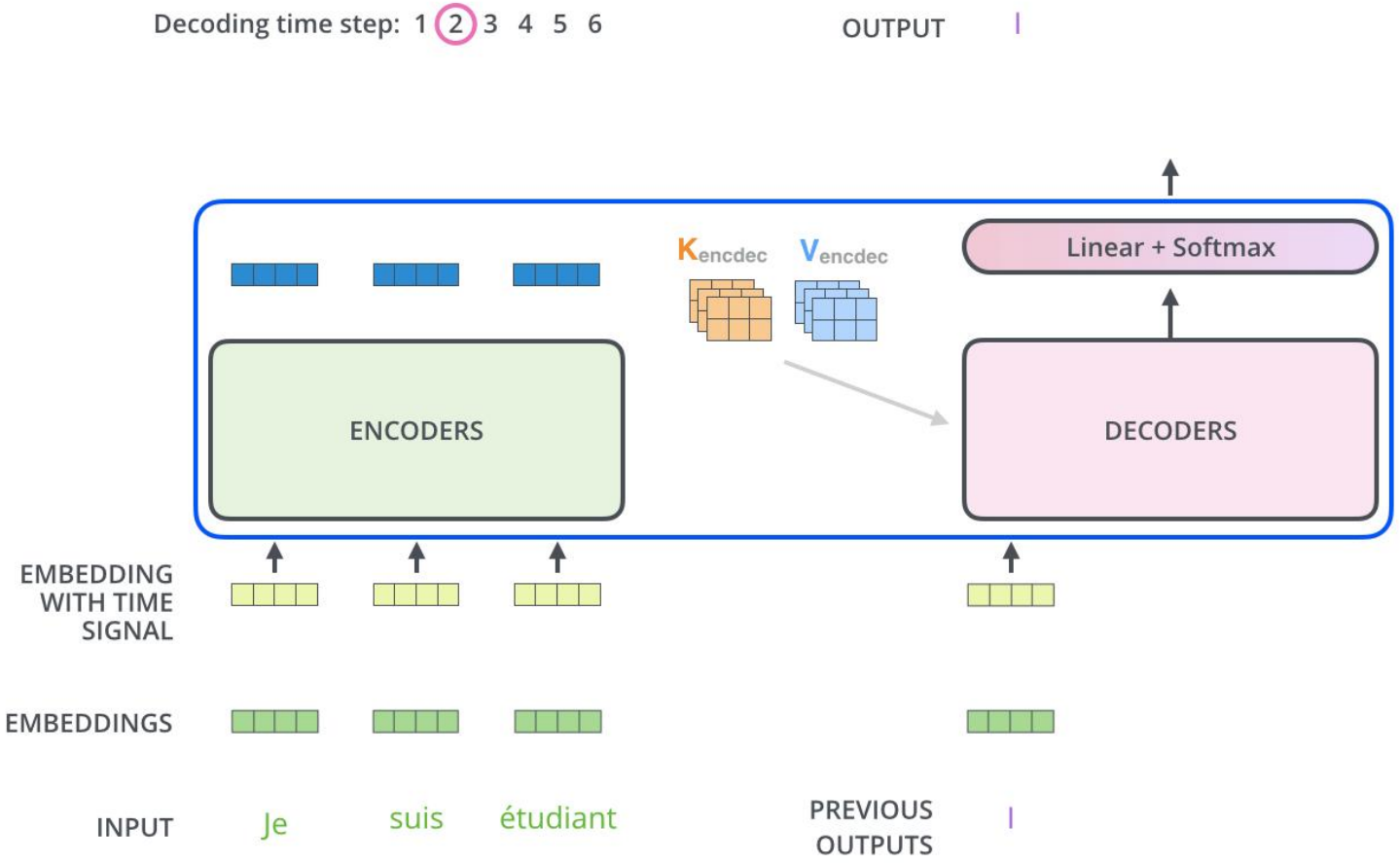
- To prevent leftward information flow in the decoder to preserve the **auto-regressive** property.

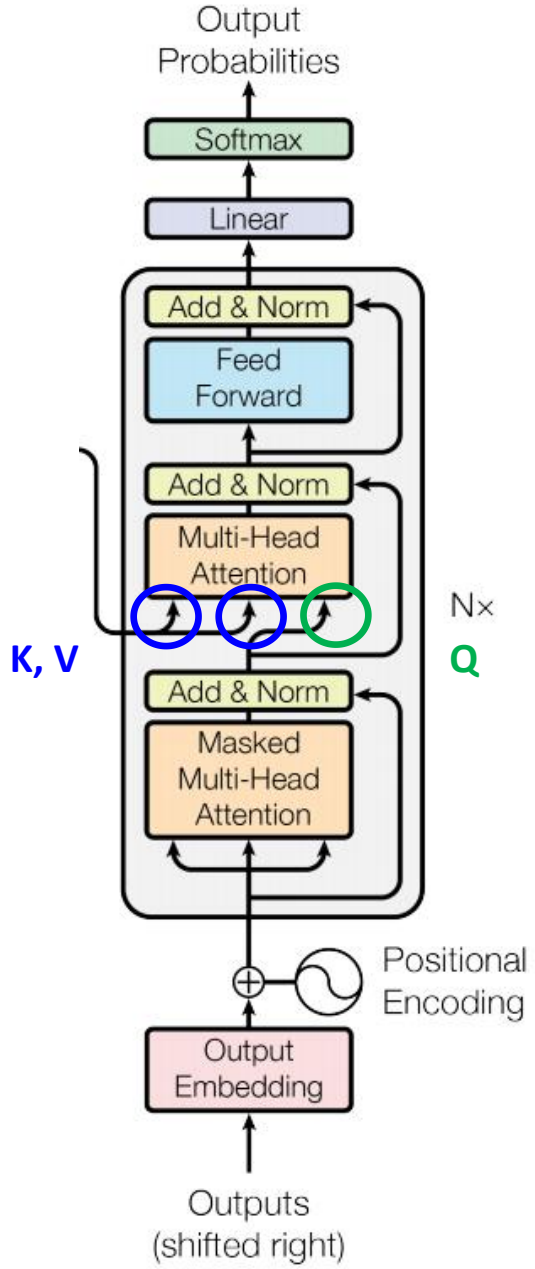




Transformer Decoder

Autoregressive

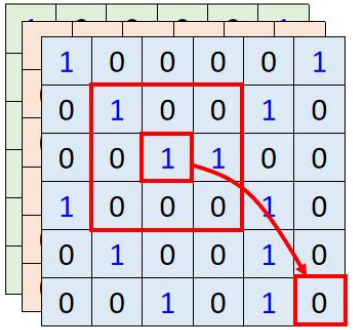




https://jalammar.github.io/illustrated-transformer/

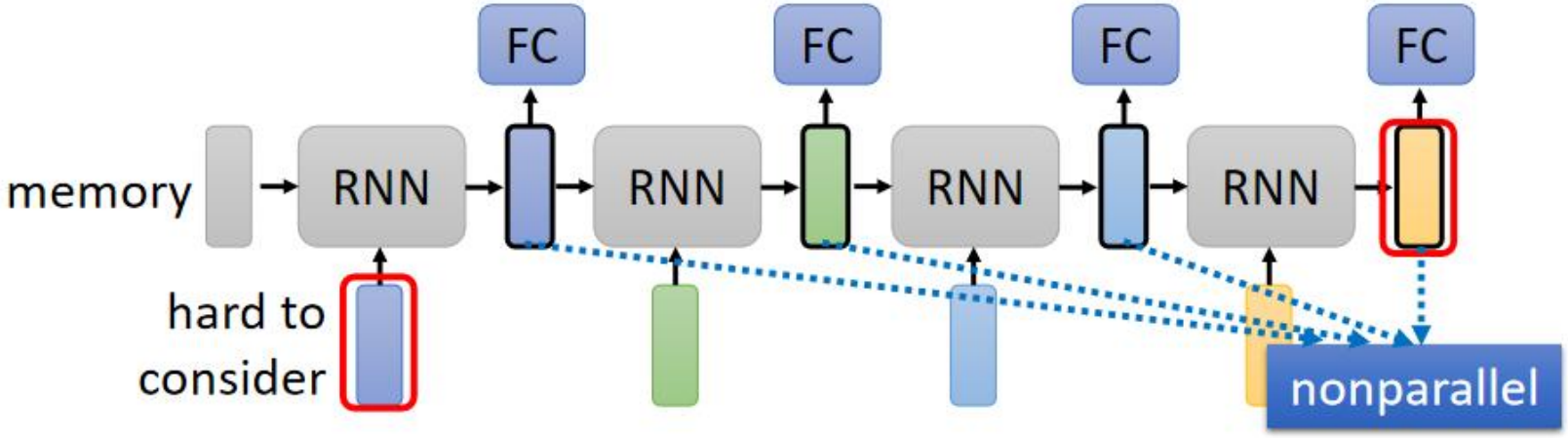
Self-Attention vs. CNN/RNN

CNN



- CNN: self-attention that can only attends in a receptive field.
- RNN: hard to learn long-range dependencies;
  nonparallel, more computation time.

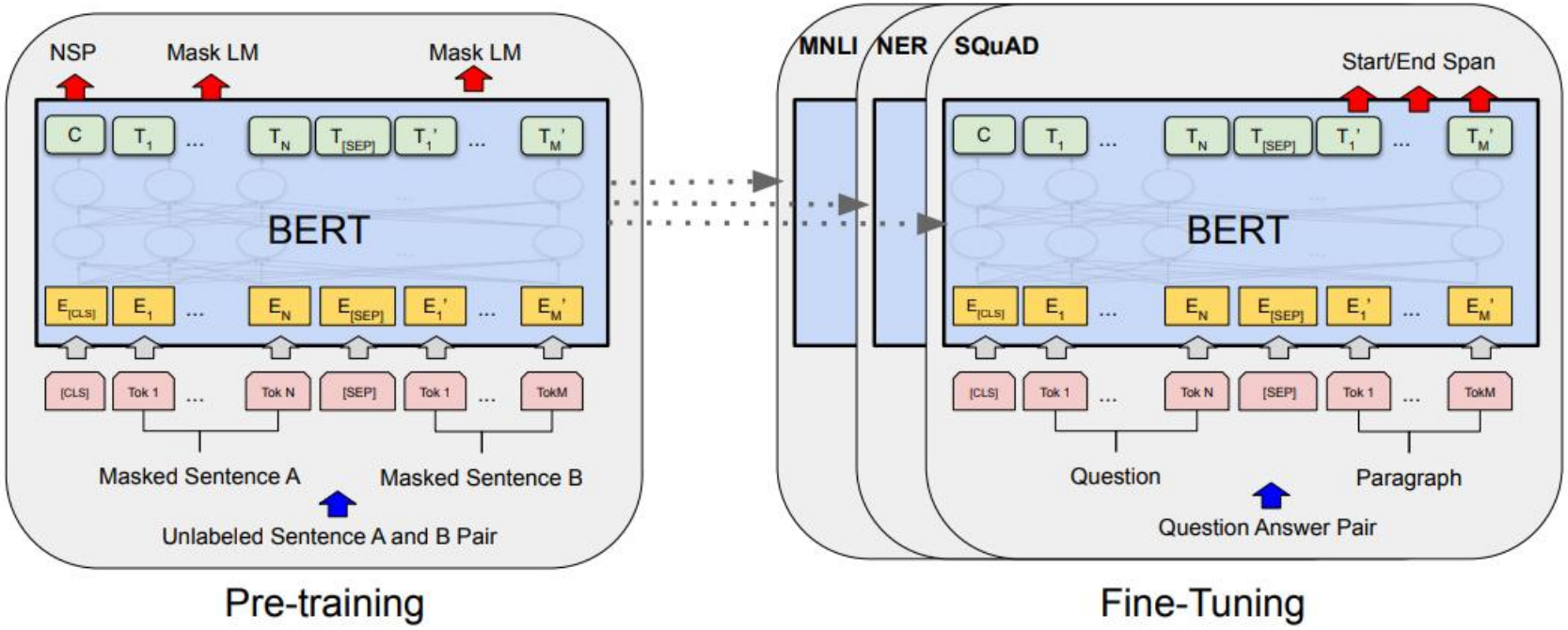
RNN



Transformer in Neural Language Processing (NLP)

- BERT: same network architecture as **transformer encoder.**

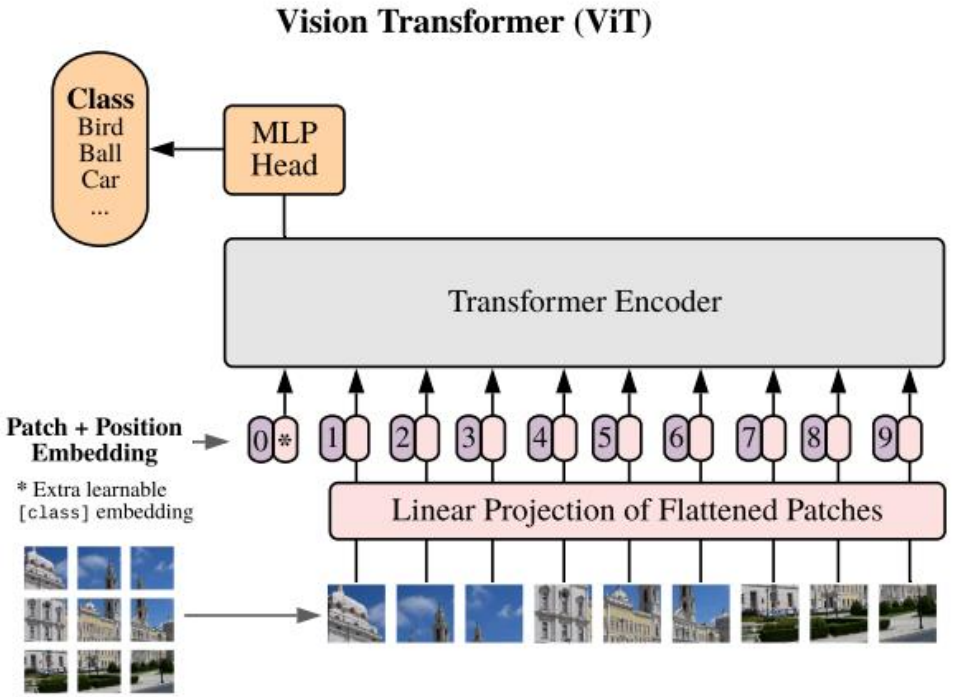
  Masked language modeling (MLM) pre-training

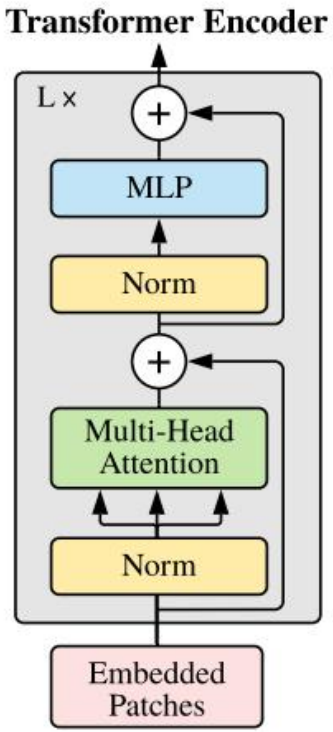


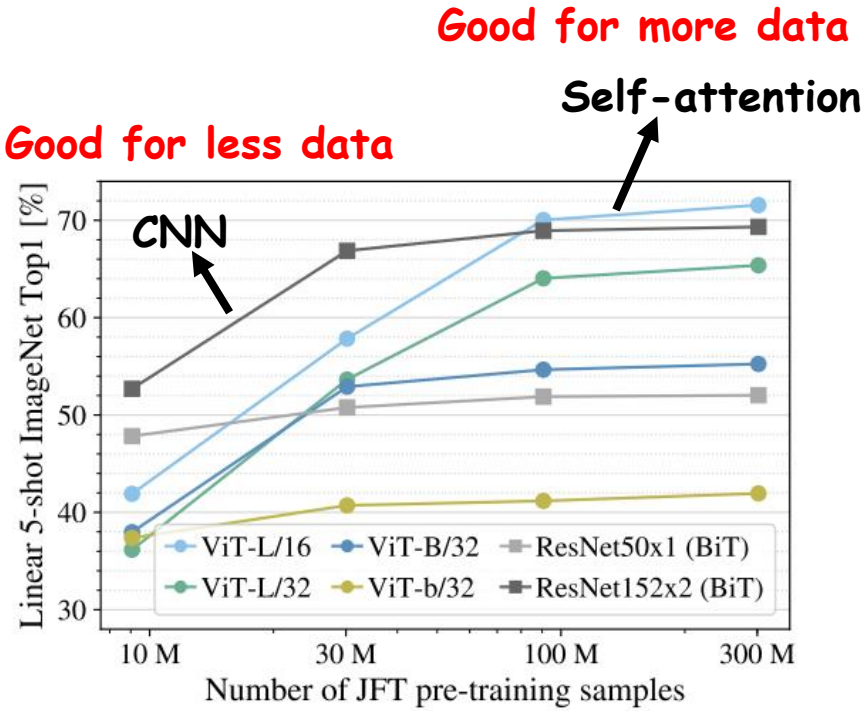
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL 2019

Transformer in Computer Vision (CV)

- **Transformers for Image Recognition**



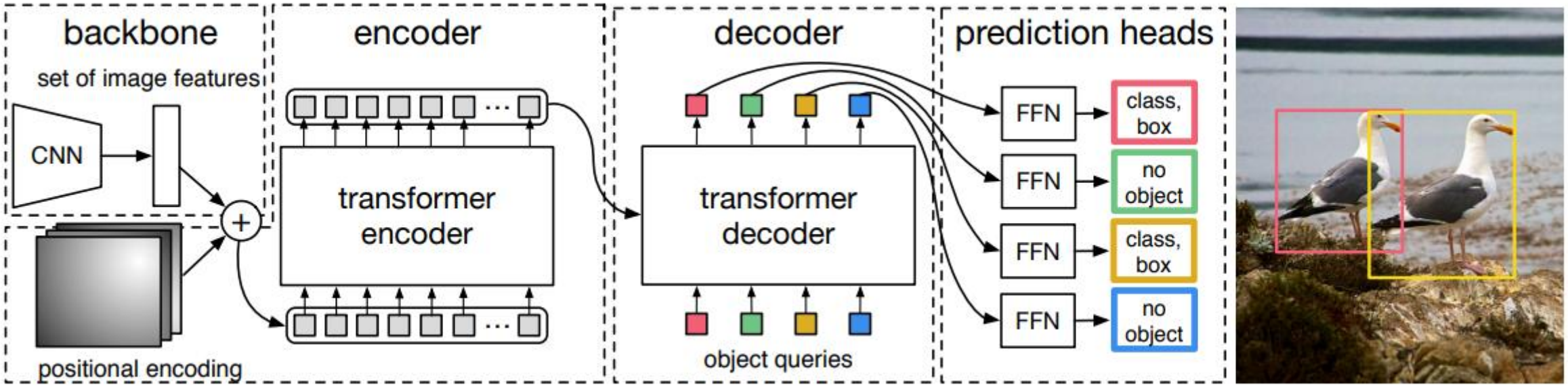




An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021

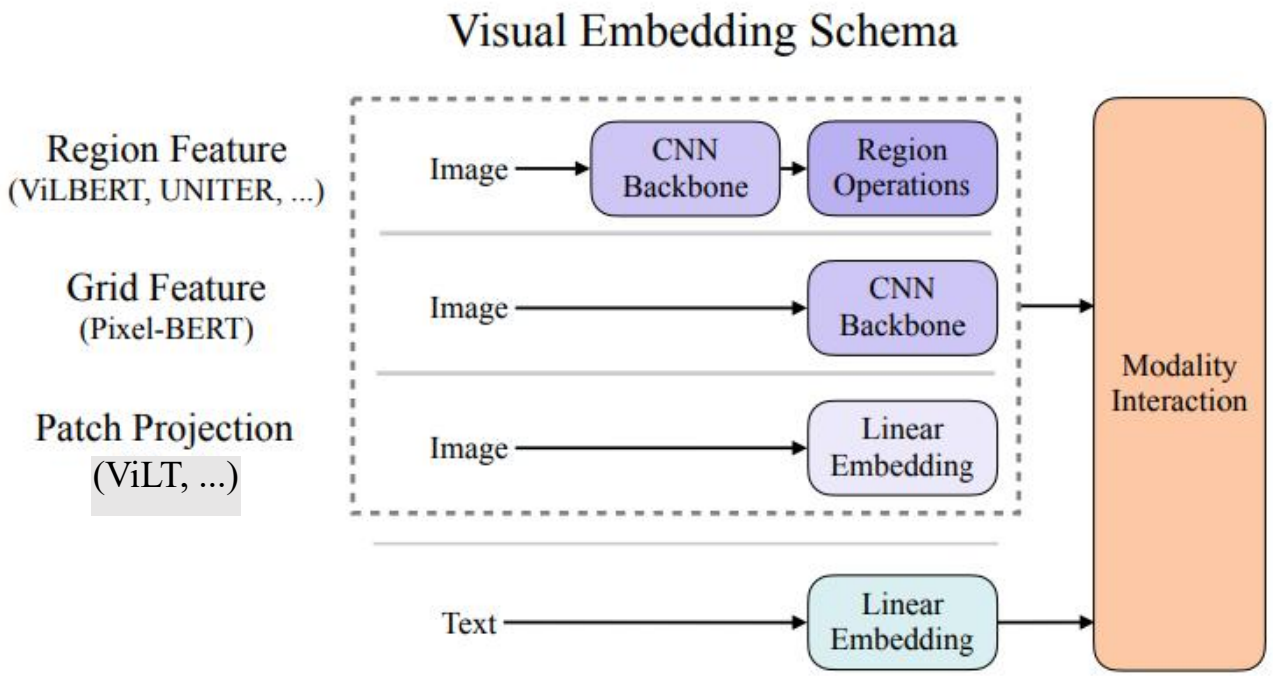
Transformer in Computer Vision (CV)

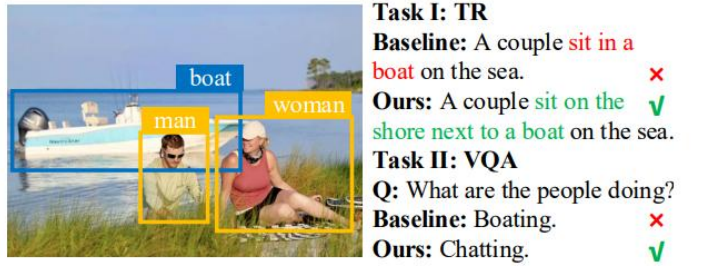
- **Transformers for Object detection**

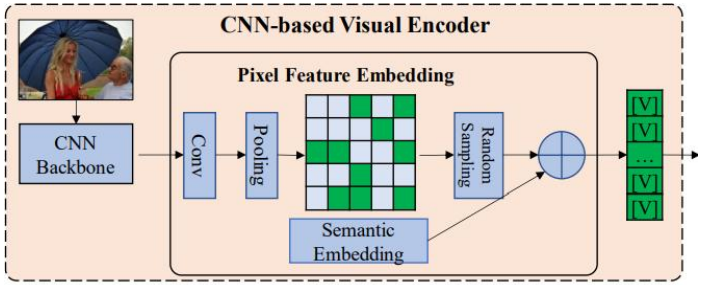


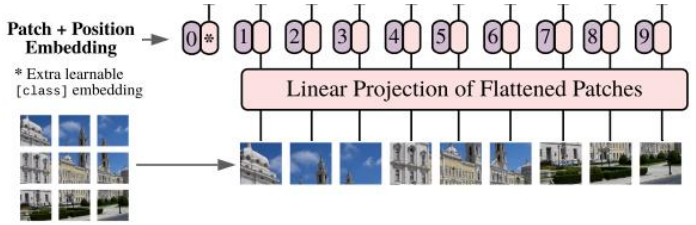
End-to-End Object Detection with Transformers. ECCV 2020

Transformer for Vison-Language (VL)



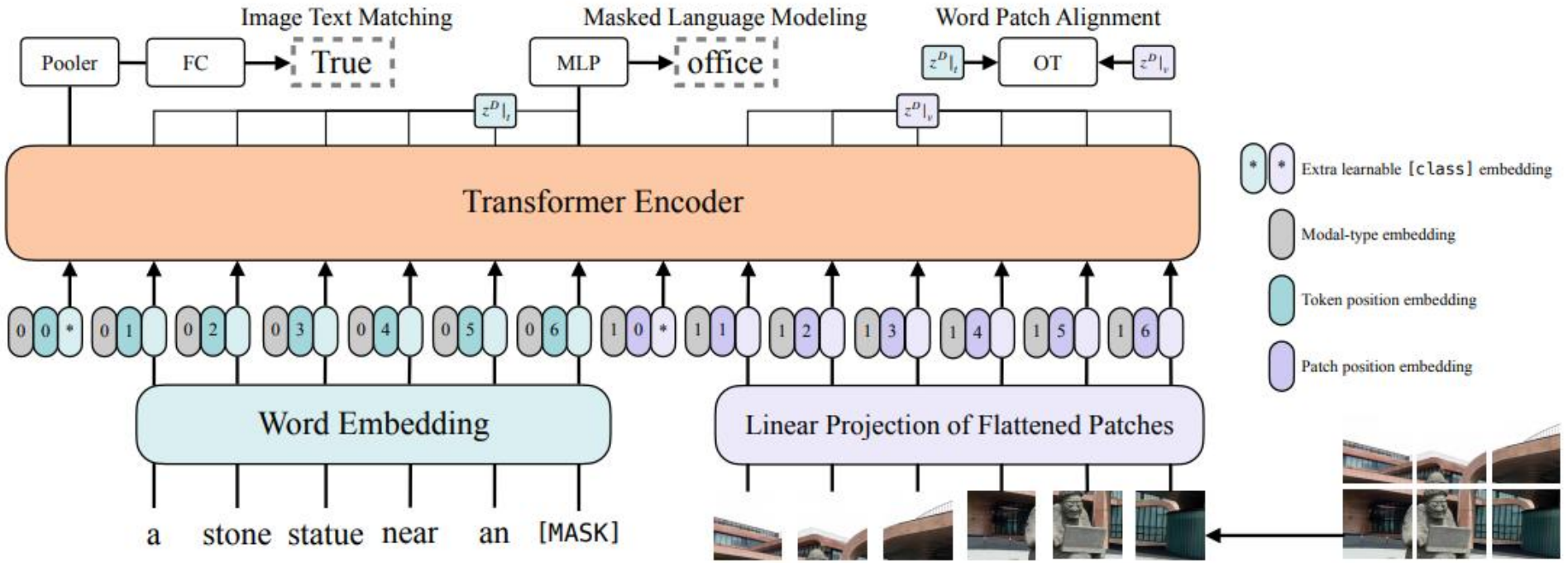




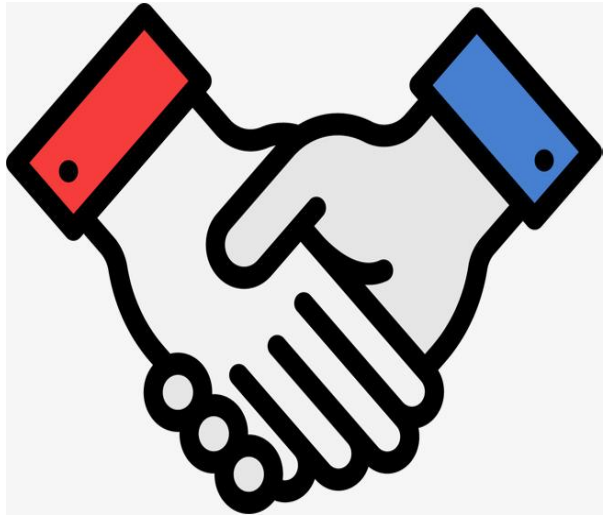


ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. NeurIPS 2019
UNITER: UNiversal Image-TExt Representation Learning. ECCV 2020
Pixel-BERT: Aligning Image Pixels with Text by Deep Multi-Modal Transformers. ArXiv:2004.00849
ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. ICML 2021

Transformer for Vison-Language (VL)



ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. ICML 2021



Thanks

zhengf@sustech.edu.cn