CHULA ENGINEERING
Foundation toward Innovation
COMPUTER

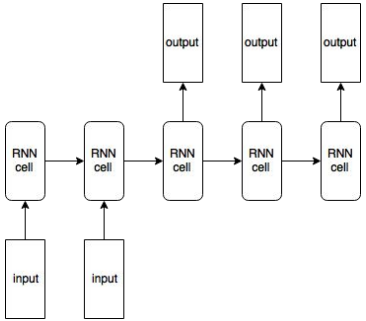
# Attention and Transformer

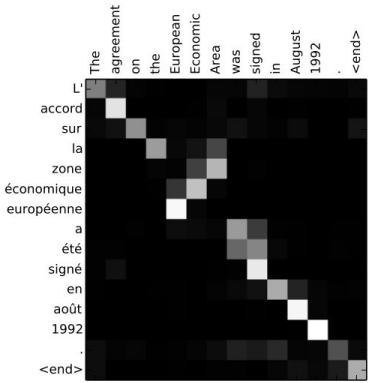
Ekapol Chuangsuwanich

For NIDA

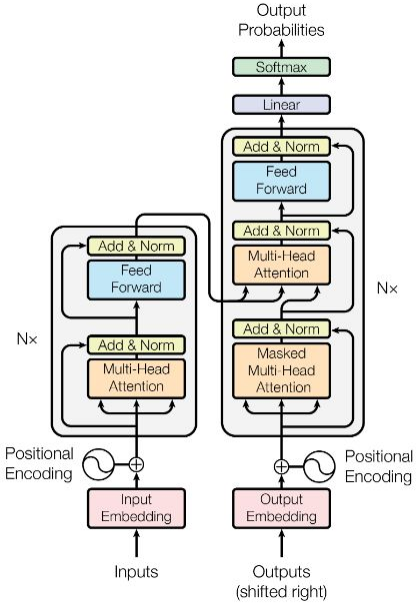
# Outline

## Text Generation & Attention Mechanism



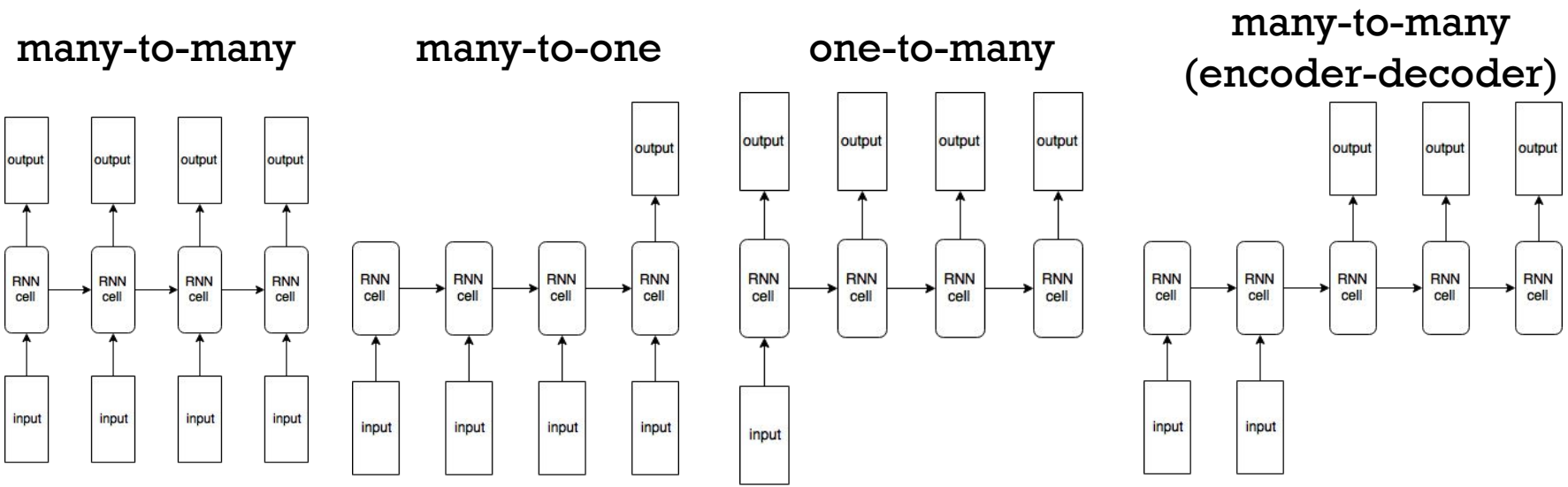

## Transformer



+

# Text Generation & Attention Mechanism

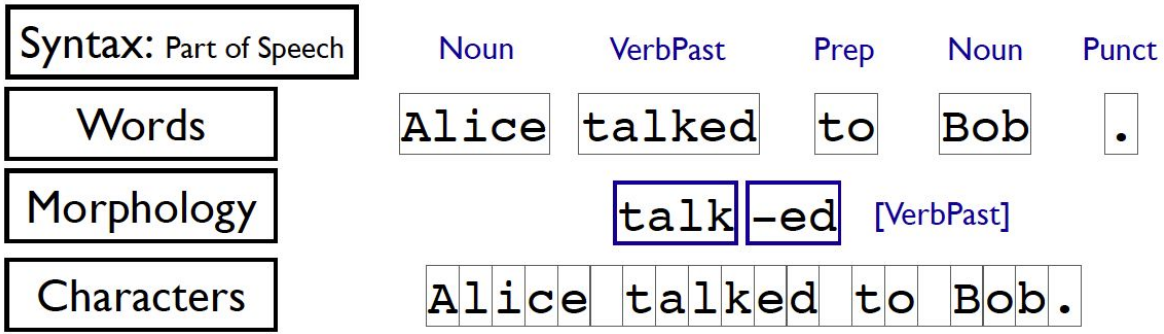
# Different types of RNN architectures

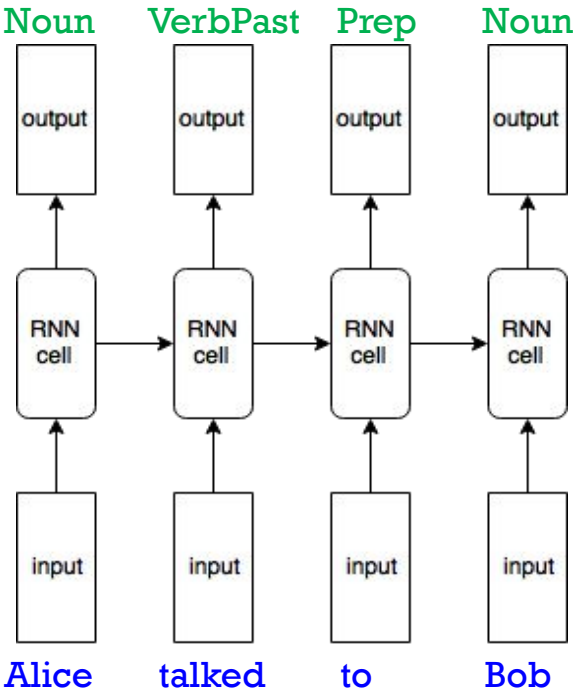



Reference: The Unreasonable Effectiveness of Recurrent Neural Networks, http://karpathy.github.io/2015/05/21/rnn-effectiveness/

# Many-to-many

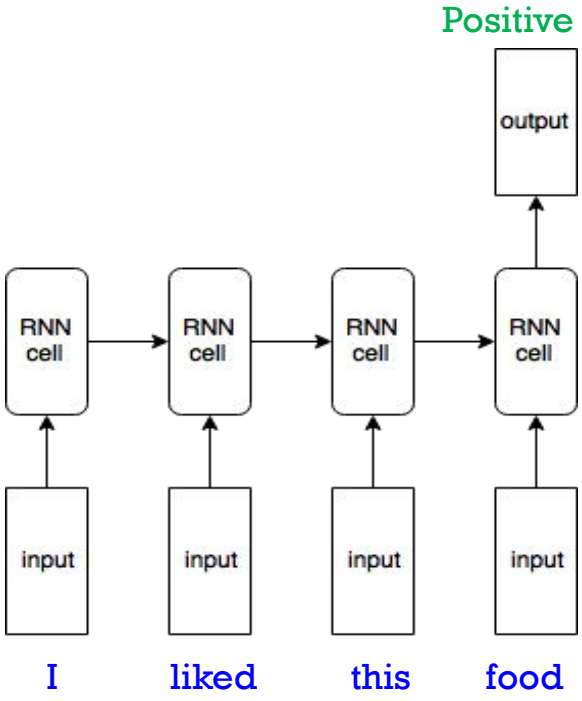
- You have seen and implemented this type of RNN architecture in your homework already.
- E.g. Tokenization, POS tagging
- Sequence Input, Sequence Output



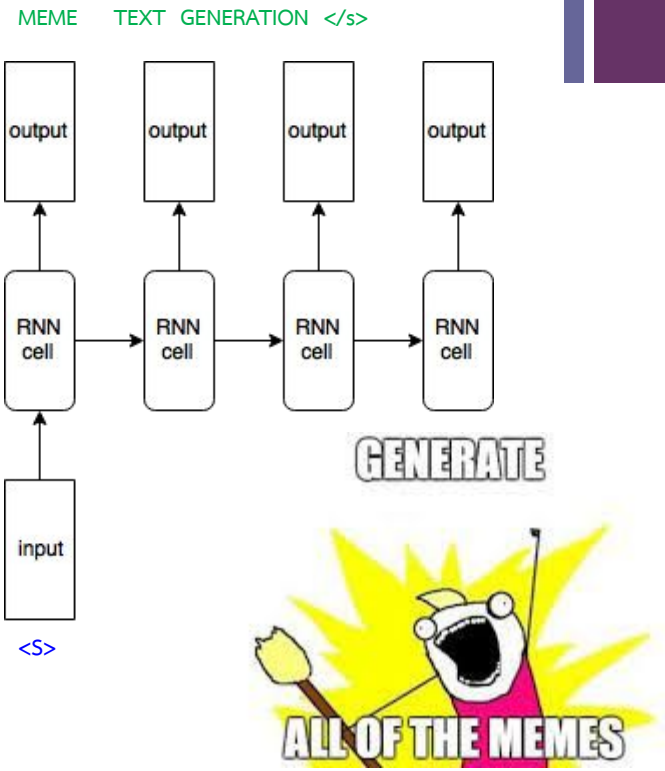
# Many-to-one

- You probably just implemented this type of RNN for your take-home exam.
- E.g. Sentiment Analysis, Text classification
- Sequence input



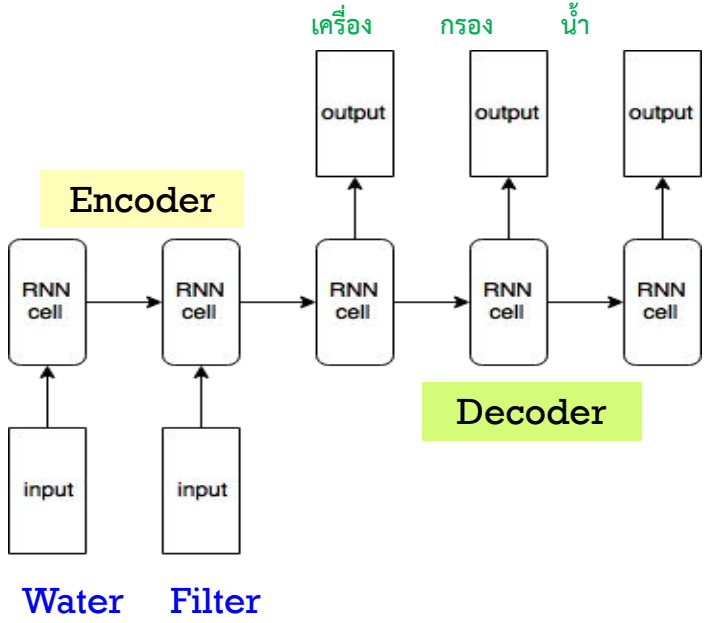
# One-to-many

- Sequence output
- E.g. Music Generation, Image caption generation
- Music **generation**
  - Input: Initial seed
  - Output: Sequence of music notes
- Image caption **generation**
  - Input: Image features extracted by CNN
  - Output: Sequence of text



# Many-to-many (encoder-decoder)

- Sequence Input, Sequence output
- These two sequences can be of different length
- E.g. Machine Translation
  - Input: English Sentence
  - Output: Thai Sentence
- Machine Translation is also a **text generation task**



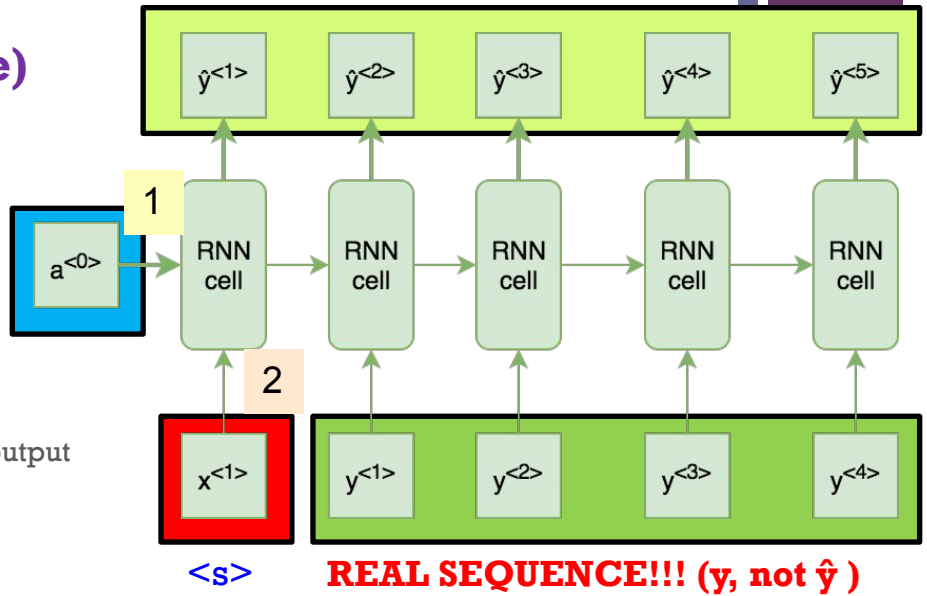
# Text generation model (training)

Training | Inference

## One-to-Many RNN (autoregressive)

- The only real input is $x^{<1>}$
- $a^{<0>}$ is the initial hidden state.
- $\hat{y}$ is the predicted output.
- $y$ is an actual output.
- During **the training phase**, instead of using the predicted output to feed into the next time-step, we use the actual output.

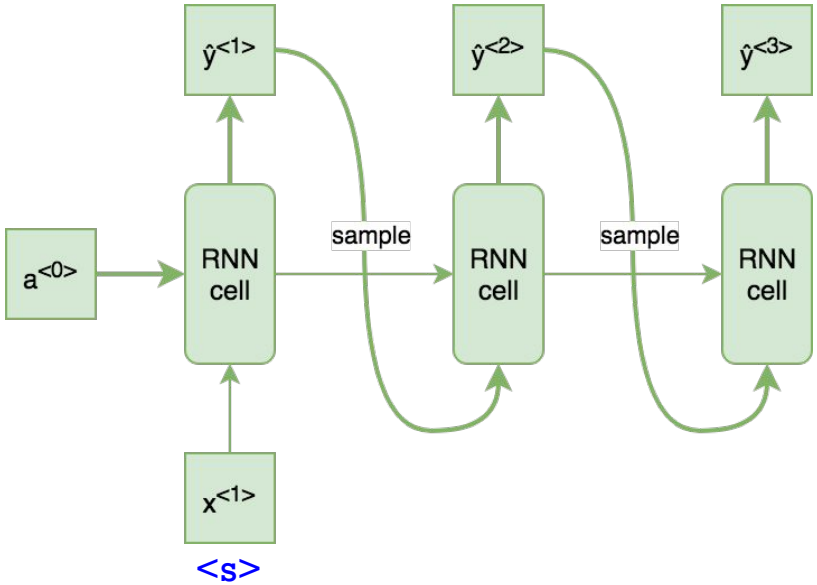
$$a^{<t>} = \underbrace{Wa^{<t-1>}}_{1} + \underbrace{Wx^{<t>}}_{2} + b$$



# Text generation model (inference; testing)

Training | Inference

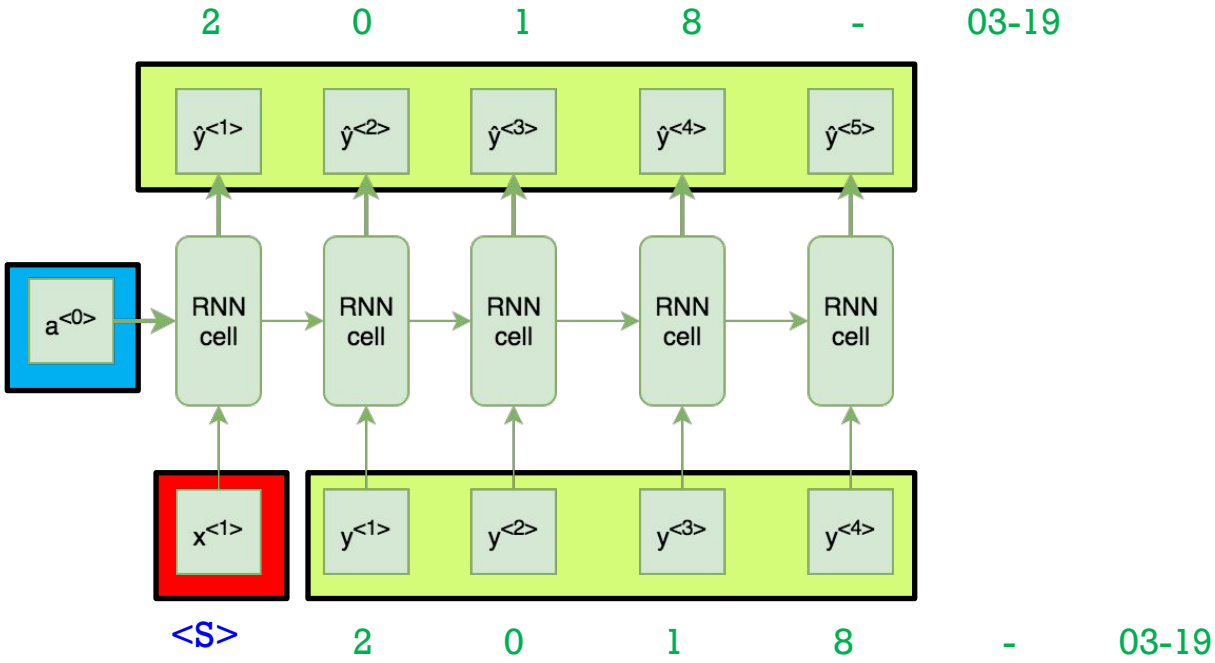
- To generate a novel sequence, the inference model (testing phase) randomly samples an output from a softmax distribution.



# In class demo: Text generation

In-class demo: Generating a piece of text using RNN; Random Date Generation "2018-03-19"

https://github.com/ekapolc/NLP_2021/blob/main/HW8/Demo1_text_generation.ipynb

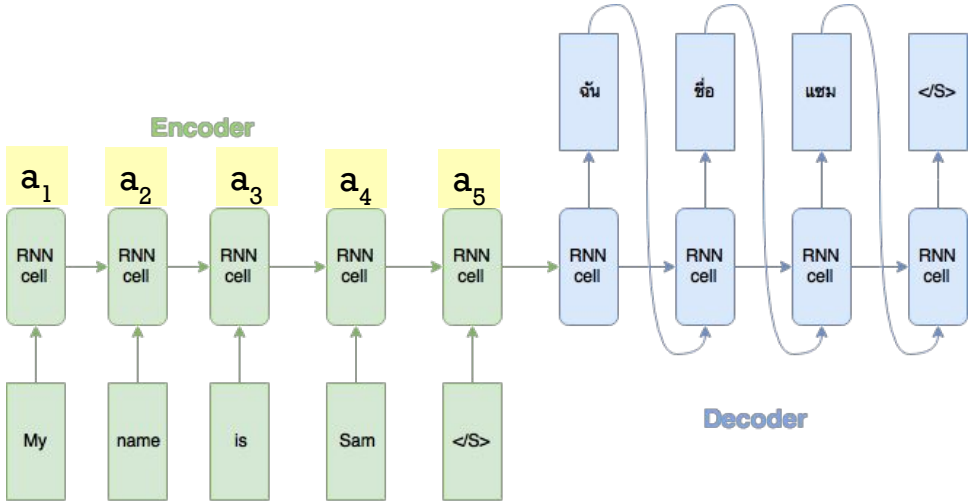


# Attention Mechanism (Many-to-Many)

**Attention** is commonly used in sequence-to-sequence model, it allows **the decoder part** of the network to focus/**attend** on a different part of **the encoder outputs** for every step of the decoder's own outputs.

Why attention?

This is what we want you to think about: How can information travel from one end to another in neural networks?

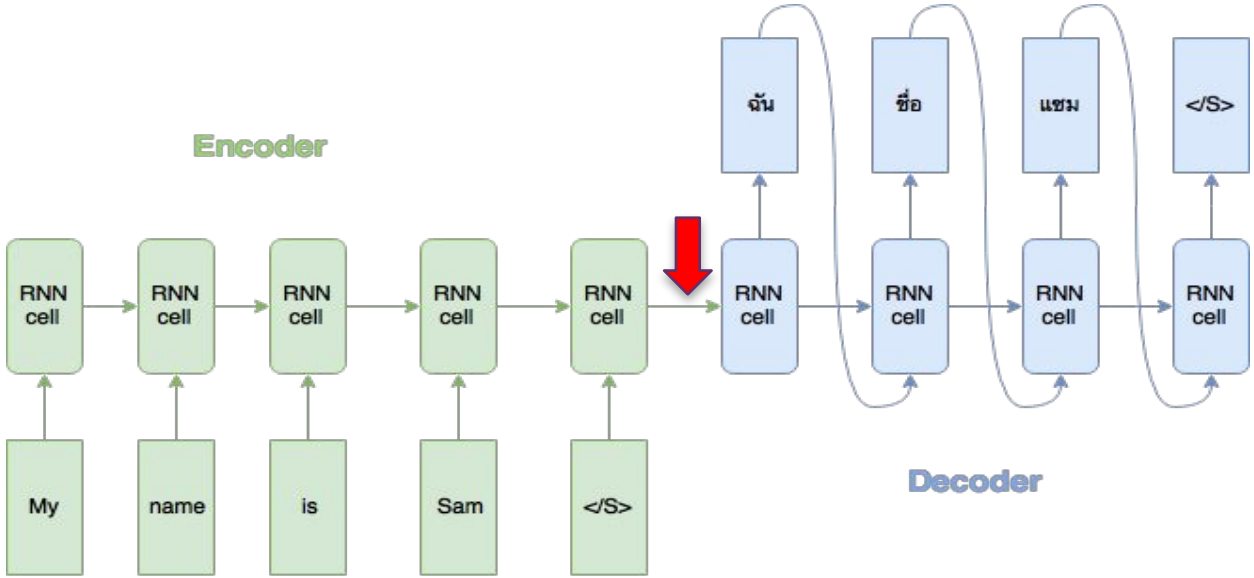



Machine Translation Problem: English to Thai

# Attention Mechanism (cont.)

Why attention?

"You can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector!" - Raymond Mooney (2014)




Reference:http://yoavartzi.com/sp14/slides/mooney.sp14.pdf

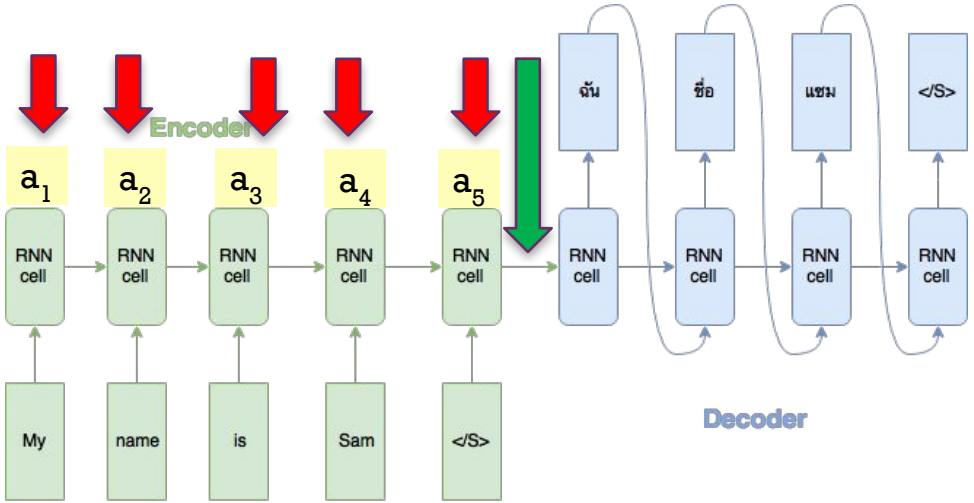
Machine Translation Problem: English to Thai

# Attention Mechanism (cont.)

Why attention?

Main idea: We can use multiple vectors based on the length of the sentence instead of one.

Attention mechanism = Instead of encoding all the information into a fixed-length vector, the decoder gets to decide parts of the input source to pay attention.

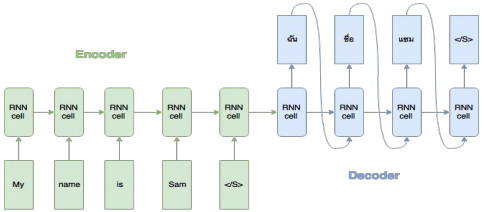



Machine Translation Problem: English to Thai

# Graphical Example: English-to-Thai machine translation

- This is a rough estimate of what might occur for English-to-Thai translation

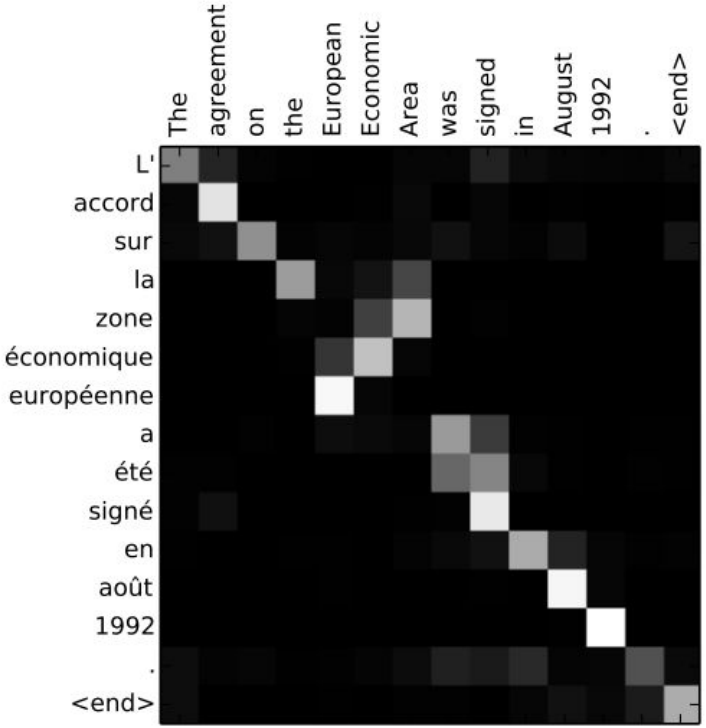
| | My | name | is | Sam | encoder |
|---|---|---|---|---|---|
| ฉัน | | | | | |
| ชื่อ | | | | | |
| แซม | | | | | |
| decoder | | | | | |




min max

Machine Translation Problem: English to Thai

# Graphical Example: English-to-French machine translation





Reference: Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." ICLR(2015).

# Attention Mechanism: Recap Basic Idea

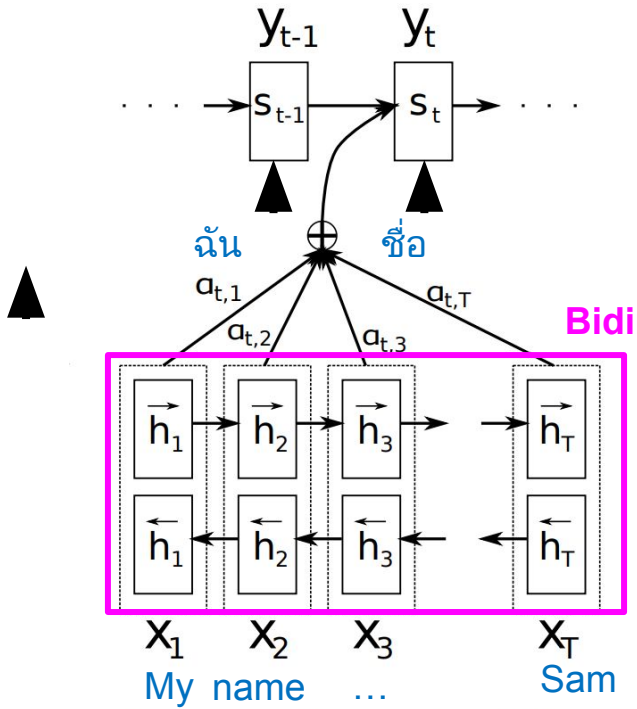
- Encode each word in the sequence into a vector
- When DECODING, perform a linear combination of these encoded vectors from the encoding step with their corresponding "attention weights".
  - (scalar 1)(encoded vector1) + (scalar 2)(encoded vector 2) + (scalar 3)(encoded vector 3)

$$\mathbf{c}_i = \sum_j a_{ij}\mathbf{h}_j$$

j = each encoder's input
i = each decoder's input

- A vector formed by this linear combination is called "context vector"
- Use context vectors as inputs for the decoding step

Reference: Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." ICLR(2015).




Decoder

Bidirectional LSTM

Encoder

Figure 1: The graphical illustration of the proposed model trying to generate the $t$-th target word $y_t$ given a source sentence $(x_1, x_2, \ldots, x_T)$.

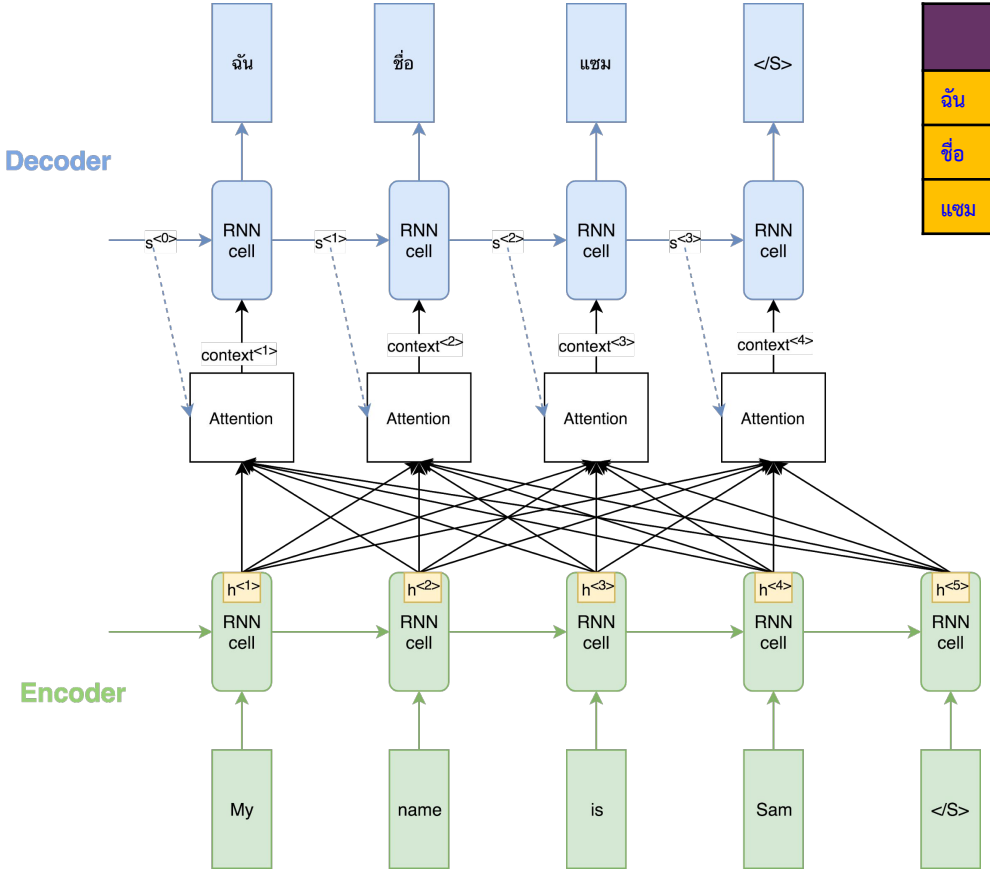
source = encoder

target word = decoder



Reference: Bahdanau, D., Cho, K., & Bengio, Y.. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015

# RNN and attention mechanism




| | My | name | is | Sam |
|---|---|---|---|---|
| ฉัน | | | | |
| ชื่อ | | | | |
| แซม | | | | |

# Attention Mechanism (1): $C_i$

We want to calculate a context vector ***c*** based on hidden states $\mathbf{s}_0 \ldots \mathbf{s}_{m-1}$ that can be used with the current state $\mathbf{h}_j$ for prediction. The context vector $\mathbf{c}_i$ at position "i" is calculated as an average of the previous states weighted with the attention scores $\mathbf{a}_i$.

$$\mathbf{c}_i = \sum_j a_{ij} \mathbf{h}_j$$

context vector

attention score

**encoder** state at index j

i = decoder index
j = encoder index

$$a_{ij} = softmax(f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j))$$

previous hidden state/ **decoder** state

attention score(weight vector) of encoder state at index j

**encoder** state at index j

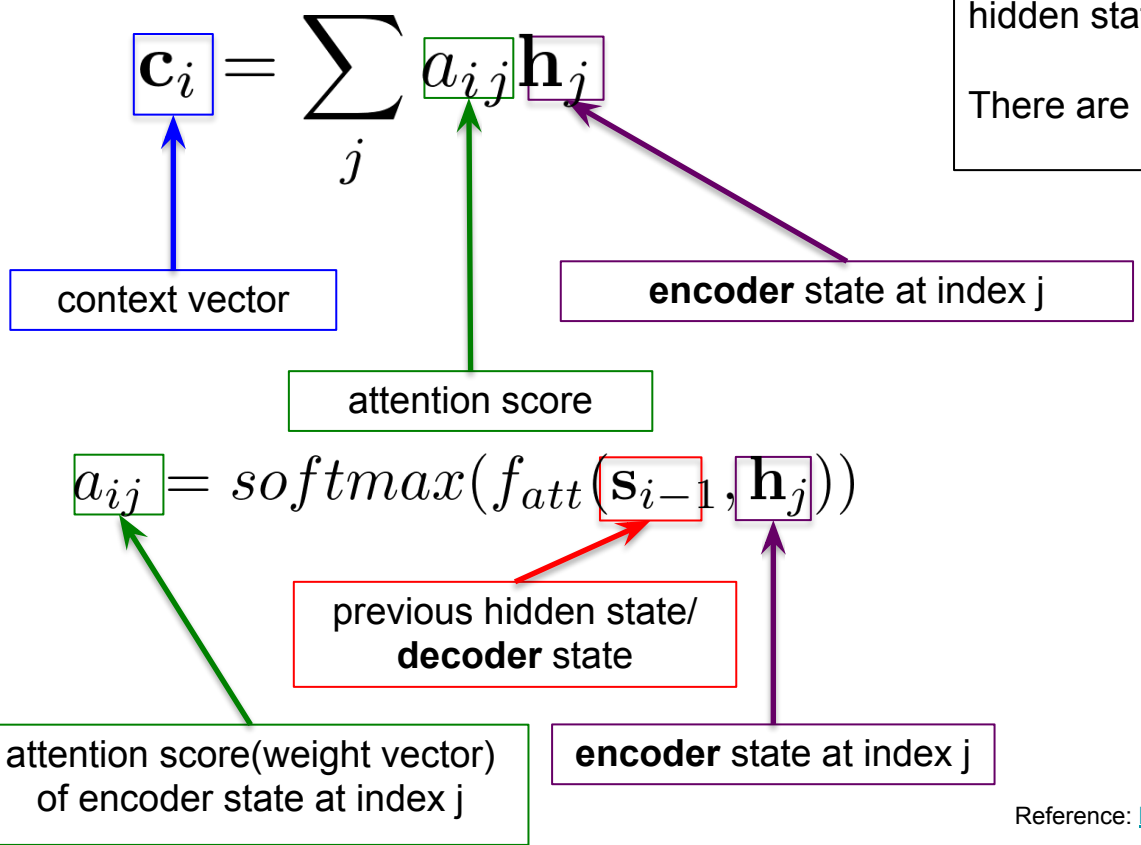


Reference: http://ruder.io/deep-learning-nlp-best-practices/index.html#attention

# Attention Mechanism (2): $f_{att}$

The attention function $f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j)$ calculates an unnormalized alignment score between the current hidden state $\mathbf{s}_{i-1}$ and the previous hidden state $\mathbf{h}_j$.

There are many variants of the attention function $f_{att}$.



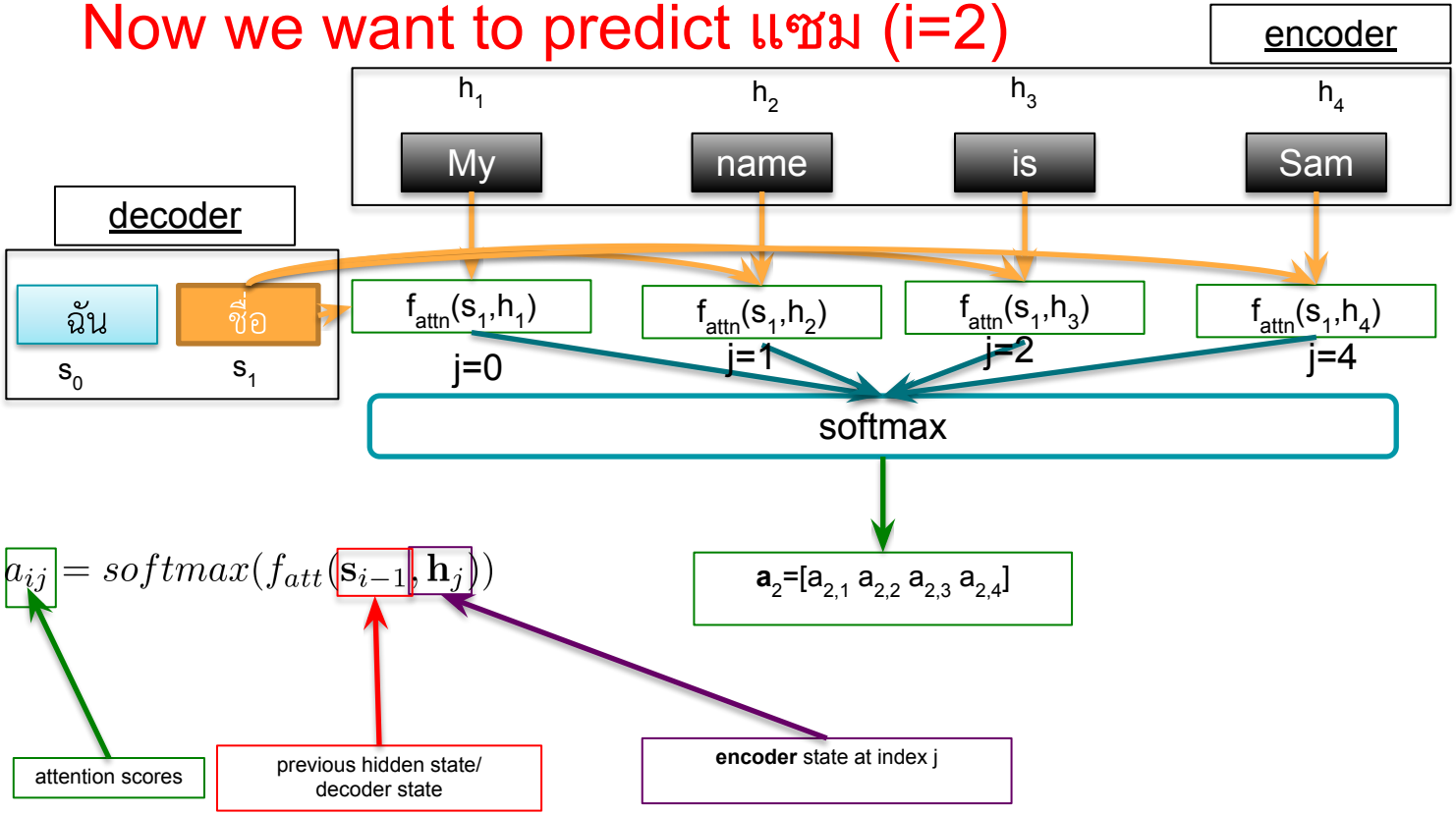

i = decoder index
j = encoder index



Reference: http://ruder.io/deep-learning-nlp-best-practices/index.html#attention

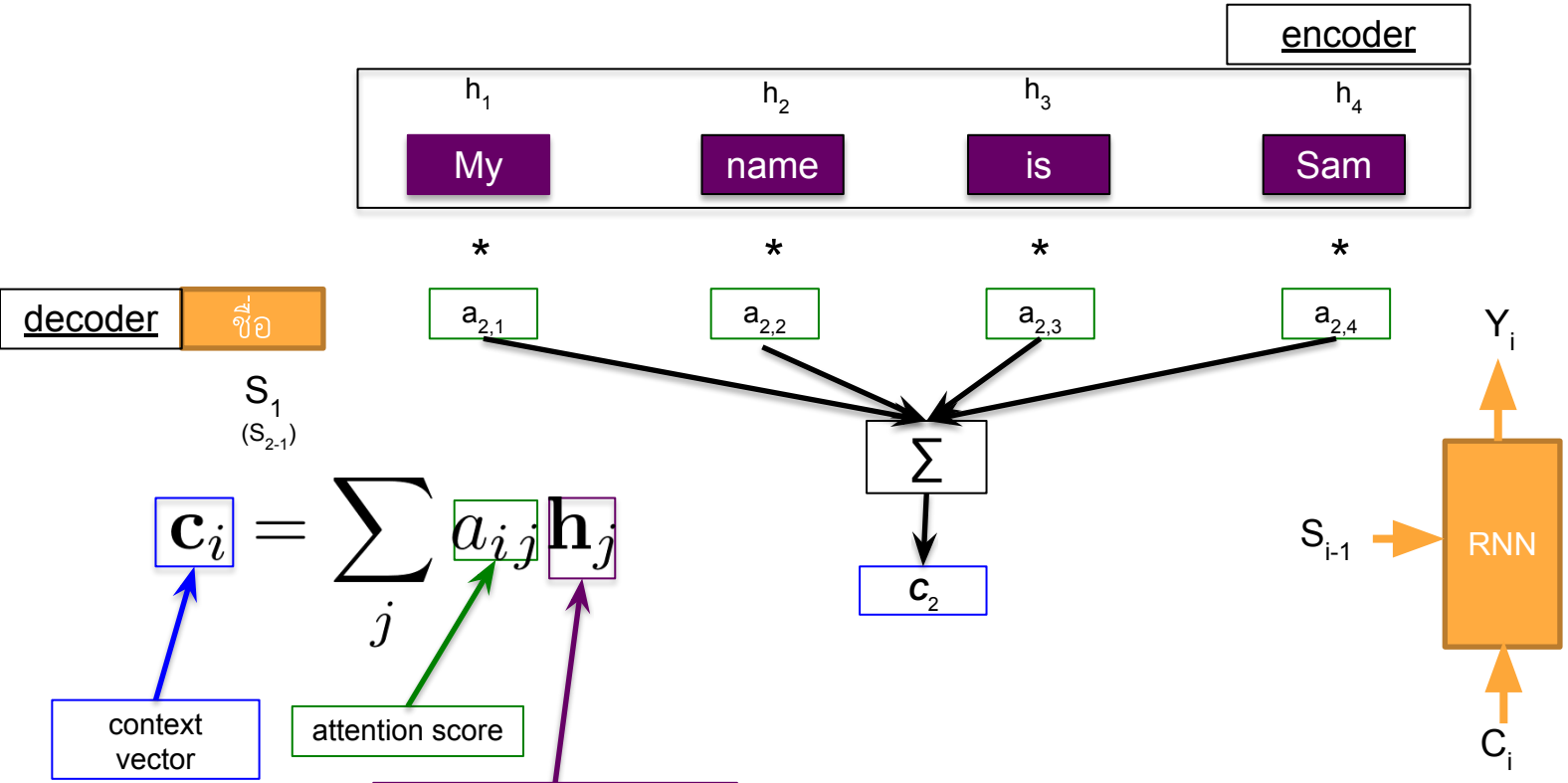
# Attention Calculation Example (1): Attention Scores

## Now we want to predict แซม (i=2)



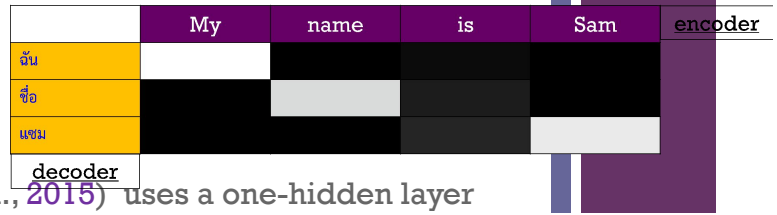

i = decoder index
j = encoder index

# Attention Calculation Example (2): Context Vector



$$a_{ij} = softmax(f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j))$$

# Type of Attention mechanisms




(Remember that there are many variants of attention function $\mathbf{f}_{attn}$ )

**Additive attention:** The original attention mechanism (Bahdanau et al., 2015) uses a one-hidden layer feed-forward network to calculate the attention alignment:

$$f_{attn}(\mathbf{s}_{i-1}, \mathbf{h}_j) = tanh(\mathbf{W}_a[\mathbf{s}_{i-1}; \mathbf{h}_j])$$

**Multiplicative attention:** Multiplicative attention (Luong et al., 2015) simplifies the attention operation by calculating the following function:

$$f_{attn}(\mathbf{s}_{i-1}, \mathbf{h}_j) = \mathbf{s}_{i-1}^{\top} \mathbf{W}_a \mathbf{h}_j$$

**Self-attention:** Without any additional information, however, we can still extract relevant aspects from the sentence by allowing it to attend to itself using self-attention (Lin et al., 2017)

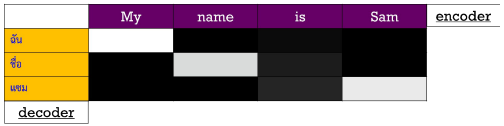
$$\mathbf{a} = softmax(\mathbf{w}_{s_2} tanh(\mathbf{W}_{s_1} \mathbf{H}^T))$$

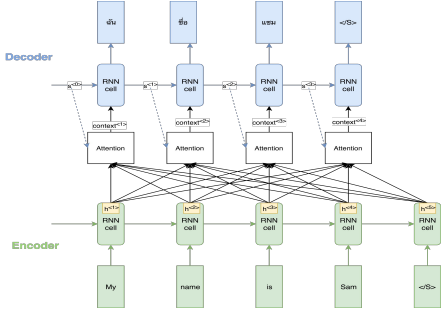
**Key-value attention:** key-value attention (Daniluk et al., 2017) is a recent attention variant that separates form from function by keeping separate vectors for the attention calculation.

Reference: http://ruder.io/deep-learning-nlp-best-practices/index.html#attention

$$a_{ij} = softmax(f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j))$$

# Additive Attention




- The original attention mechanism (Bahdanau et al., 2015) uses a one-hidden layer feed-forward network to calculate the attention alignment:

concatenation

$$f_{attn}(\mathbf{s}_{i-1}, \mathbf{h}_j) = tanh(\mathbf{W}_a[\mathbf{s}_{i-1}; \mathbf{h}_j])$$

One-hidden layer (Dense)

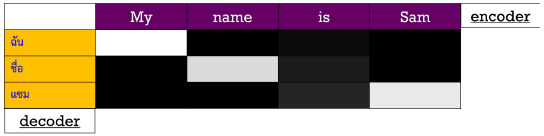
- Where $\mathbf{W}_a$ are learned attention parameters. Analogously, we can also use matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ to learn separate transformations for $\mathbf{s}_{i-1}$ and $\mathbf{h}_j$ respectively, which are then summed (hence the name additive):

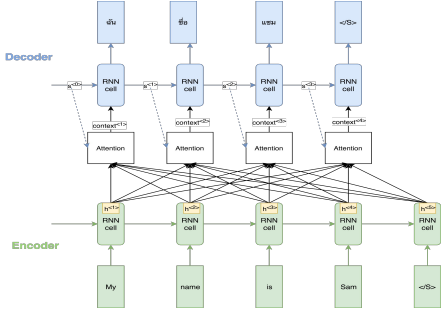
$$f_{attn}(\mathbf{s}_{i-1}, \mathbf{h}_j) = tanh(\mathbf{W}_1\mathbf{s}_{i-1} + \mathbf{W}_2\mathbf{h}_j)$$

Reference: http://ruder.io/deep-learning-nlp-best-practices/index.html#attention

$$a_{ij} = softmax(f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j))$$

# Multiplicative Attention




- Multiplicative attention (Luong et al., 2015) [16] simplifies the attention operation by calculating the following function:

$$f_{attn}(\mathbf{s}_{i-1}, \mathbf{h}_j) = \mathbf{s}_{i-1}^{\top} \mathbf{W}_a \mathbf{h}_j$$

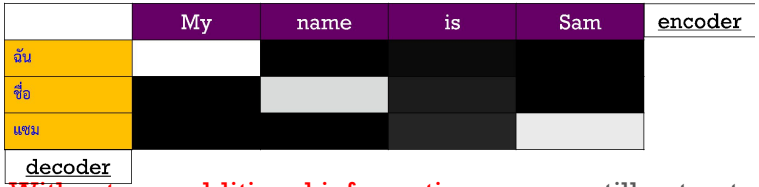
- Faster, more efficient than additive attention BUT additive attention performs better for larger dimensions
- One way to mitigate this is to scale $f_{attn}$ by $\frac{1}{\sqrt{d_s}}$

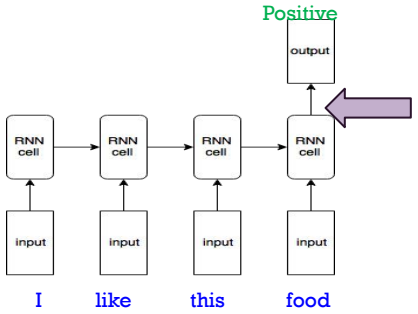
$d_s$ = #dimensions of hidden states in LSTM (context vector; latent factors)

  - Dot product of high dimensional vectors has high variance -> softmax is peaky -> small gradient -> harder to train

# Self Attention (1)

$$a_{ij} = softmax(f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j))$$




- Without any additional information, we can still extract relevant aspects from the sentence by allowing it to attend to itself using self-attention (Lin et al., 2017)

$$H = (\mathbf{h_1}, \mathbf{h_2}, \cdots \mathbf{h_n})$$

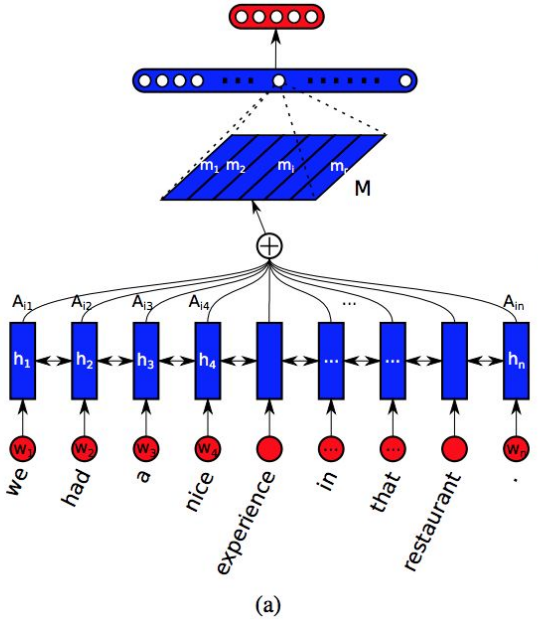
Fully connected layer

$$\mathbf{a} = softmax(\mathbf{w}_{s_2} tanh(\mathbf{W}_{s_1} \mathbf{H}^T))$$

One-hidden layer (Dense)

- $w_{s1}$ is a weight matrix, $w_{s2}$ is a vector of parameters. Note that these parameters are tuned by the neural networks.

- **The objective is to improve a quality of embedding vector by adding context information.**



# Self-attention (2)

- if I can give this restaurant a 0 I will we be just ask our waitress leave because someone with a reservation be wait for our table my father and father-in-law be still finish up their coffee and we have not yet finish our dessert I have never be so humiliated do not go to this restaurant their food be mediocre at best if you want excellent Italian in a small intimate restaurant go to dish on the South Side I will not be go back

- this place suck the food be gross and taste like grease I will never go here again ever sure the entrance look cool and the waiter can be very nice but the food simply be gross taste like cheap 99cent food do not go here the food shot out of me quick then it go in

- everything be pre cook and dry its crazy most Filipino people be used to very cheap ingredient and they do not know quality the food be disgusting I have eat at least 20 different Filipino family home this not even mediocre

- seriously f *** this place disgust food and shitty service ambience be great if you like dine in a hot cellar engulf in stagnate air truly it be over rate over price and they just under deliver forget try order a drink here it will take forever get and when it finally do arrive you will be ready pass out from heat exhaustion and lack of oxygen how be that a head change you do not even have pay for it I will not disgust you with the detailed review of everything I have try here but make it simple it all suck and after you get the bill you will be walk out with a sore ass save your money and spare your self the disappointment

- i be so angry about my horrible experience at Medusa today my previous visit be amaze 5/5 however my go to out of town and I land an appointment with Stephanie I go in with a picture of roughly what I want and come out look absolutely nothing like it my hair be a horrible ashy blonde not anywhere close to the platinum blonde I request she will not do any of the pop of colour I want and even after specifically tell her I do not like blunt cut my hair have lot of straight edge she do not listen to a single thing I want and when I tell her I be unhappy with the colour she basically tell me I be wrong and I have do it this way no no I do not if I can go from Little Mermaid red to golden blonde in 1 sitting that leave my hair fine I shall be able go from golden blonde to a shade of platinum blonde in 1 sitting thanks for ruin my New Year's with 1 the bad hair job I have ever have

(a) 1 star reviews

- I really enjoy Ashley and Ami salon she do a great job be friendly and professional I usually get my hair do when I go to MI because of the quality of the highlight and the price the price be very affordable the highlight fantastic thank Ashley i highly recommend you and ill be back

- love this place it really be my favorite restaurant in Charlotte they use charcoal for their grill and you can taste it steak with chimichurri be always perfect Fried yucca cilantro rice pork sandwich and the good tres lech I have had.The desert be all incredible if you do not like it you be a mutant if you will like diabeetus try the Inca Cola

- this place be so much fun I have never go at night because it seem a little too busy for my taste but that just prove how great this restaurant be they have amazing food and the staff definitely remember us every time we be in town I love when a waitress or waiter come over and ask if you want the cab or the Pinot even when there be a rush and the staff be run around like crazy whenever I grab someone they instantly smile acknowlegde us the food be also killer I love when everyone know the special and can tell you they have try them all and what they pair well with this be a first last stop whenever we be in Charlotte and I highly recommend them

- great food and good service .... what else can you ask for everything that I have ever try here have be great

- first off I hardly remember waiter name because its rare you have an unforgettable experience the day I go I be celebrate my birthday and let me say I leave feel extra special our waiter be the best ever Carlos and the staff as well I be with a party of 4 and we order the potato salad shrimp cocktail lobster amongst other thing and boy be the food great the lobster be the good lobster I have ever eat if you eat a dessert I will recommend the cheese cake that be also the good I have ever have it be expensive but so worth every penny I will definitely be back there go again for the second time in a week and it be even good ..... this place be amazing

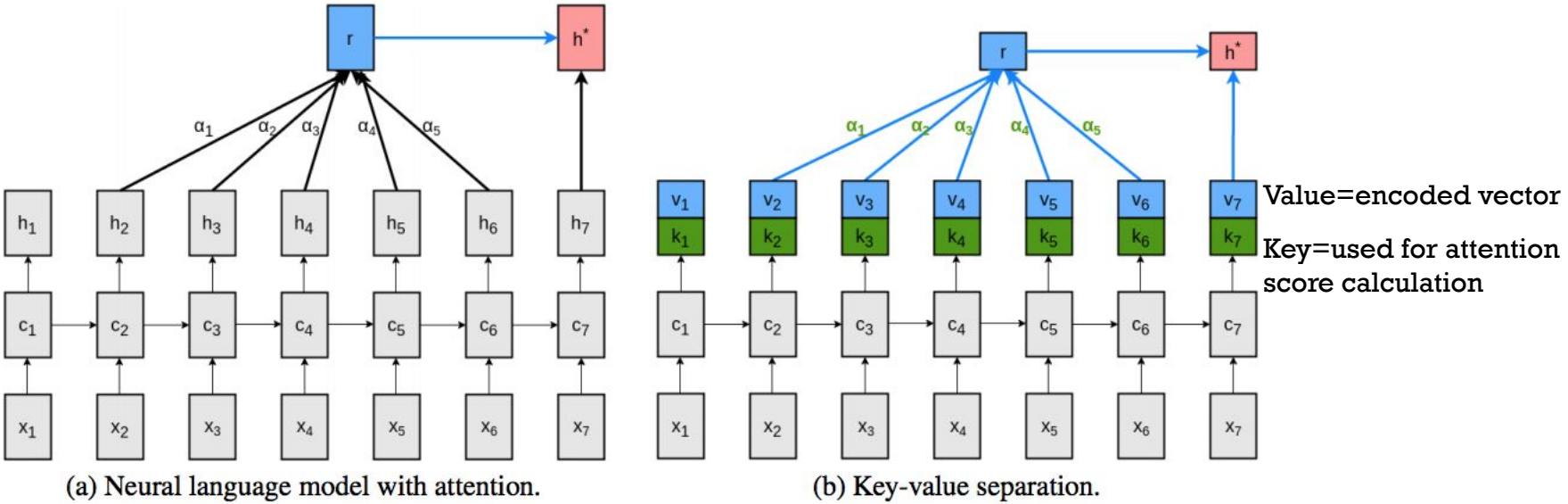
(b) 5 star reviews

Figure 2: Heatmap of Yelp reviews with the two extreme score.

# Key-value attention (1)

$$a_{ij} = softmax(f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j))$$

$$\mathbf{c}_i = \sum_j a_{ij}\mathbf{h}_j$$




(a) Neural language model with attention.

(b) Key-value separation.

Reference: Daniluk, M., Rockt, T., Welbl, J., & Riedel, S. (2017). Frustratingly Short Attention Spans in Neural Language Modeling. In ICLR 2017.

# Key-value attention (2)

previous outputs (memory) **with window L**

current output

*k - output dimension*
*L - output length*

key

value

$$\begin{bmatrix} \boldsymbol{k}_t \\ \boldsymbol{v}_t \end{bmatrix} = \boldsymbol{h}_t \quad \in \mathbb{R}^{2k}$$

$$\boldsymbol{M}_t = \tanh(\boldsymbol{W}^Y [\boldsymbol{k}_{t-L} \cdots \boldsymbol{k}_{t-1}] + (\boldsymbol{W}^h \boldsymbol{k}_t)\mathbf{1}^T) \quad \in \mathbb{R}^{k \times L}$$

Attention score

$$\boldsymbol{\alpha}_t = \text{softmax}(\boldsymbol{w}^T \boldsymbol{M}_t) \quad \in \mathbb{R}^{1 \times L}$$

Context vector ($v_t$ not included)

$$\boldsymbol{r}_t = [\boldsymbol{v}_{t-L} \cdots \boldsymbol{v}_{t-1}]\boldsymbol{\alpha}^T \quad \in \mathbb{R}^k$$

The final representation

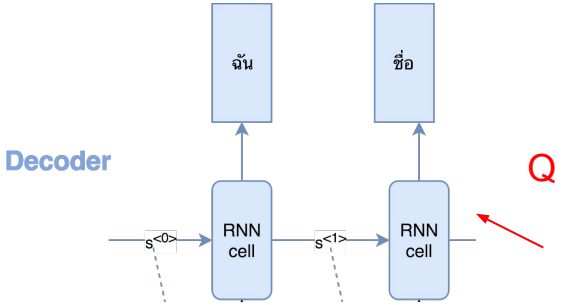
$$\boldsymbol{h}_t^* = \tanh(\boldsymbol{W}^r \boldsymbol{r}_t + \boldsymbol{W}^x \boldsymbol{v}_t) \quad \in \mathbb{R}^k$$

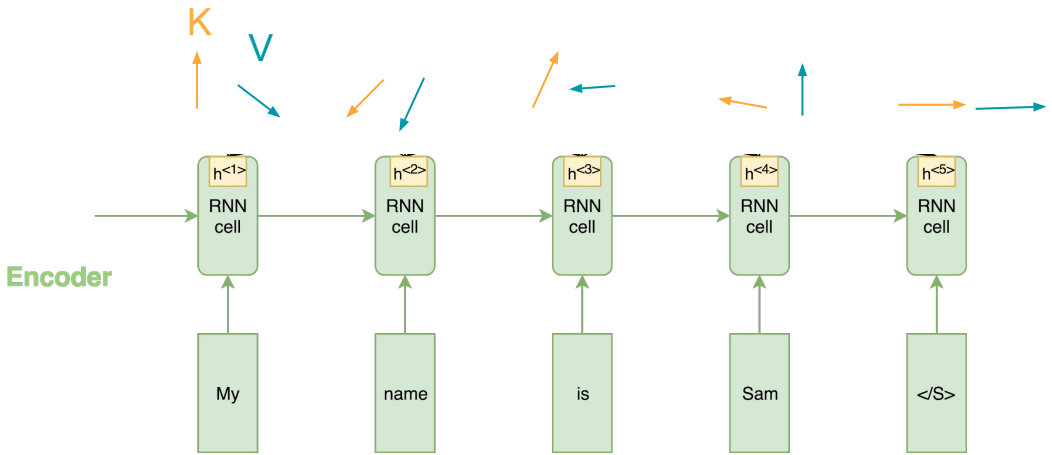
Context vector ($v_t$ not included)

Original vector at index t

$$\mathbf{c}_i = \sum_j a_{ij} \mathbf{h}_j$$

# Pictorial view of KV attention



# Demo: Neural Machine Translation with Attention (Additive Attention)

- Translate: various date formats to ISO date format
  - 27 January 2018 2018-01-27
  - 27 JAN 2018 2018-01-27

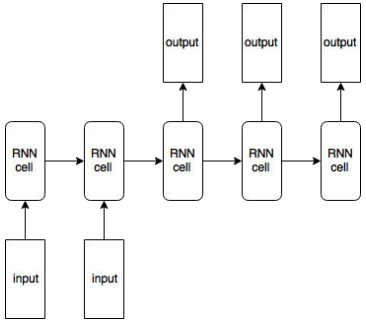
https://github.com/ekapolc/NLP_2021/blob/main/HW8/Demo2_attention_mechanism.ipynb

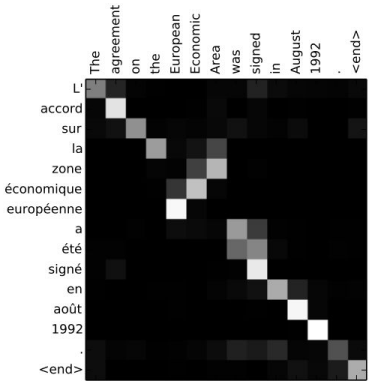
# Attention vs Convolution

- Convolution: fixed weights, limited width
- Attention: weights change with context, unlimited width

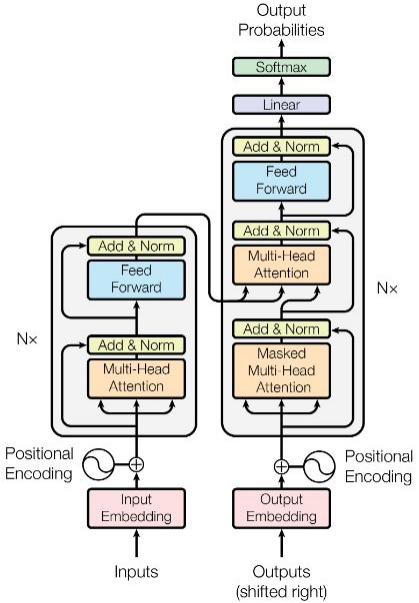
# Outline

## Text Generation & Attention Mechanism




## Transformer

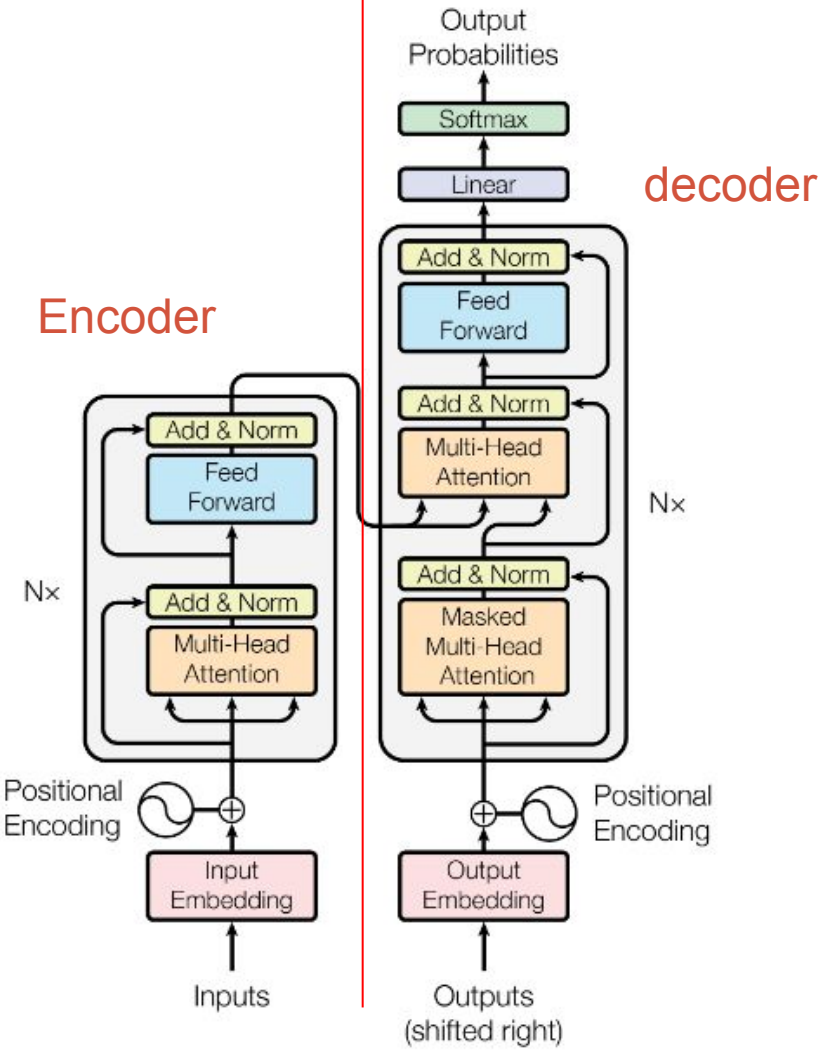


# Attention is all you need


## Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

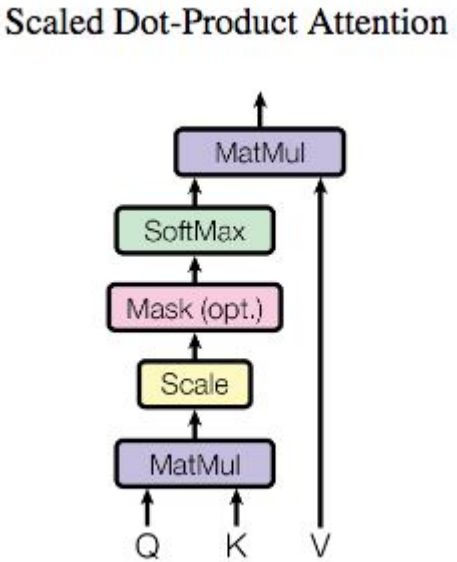

https://arxiv.org/pdf/1706.03762.pdf

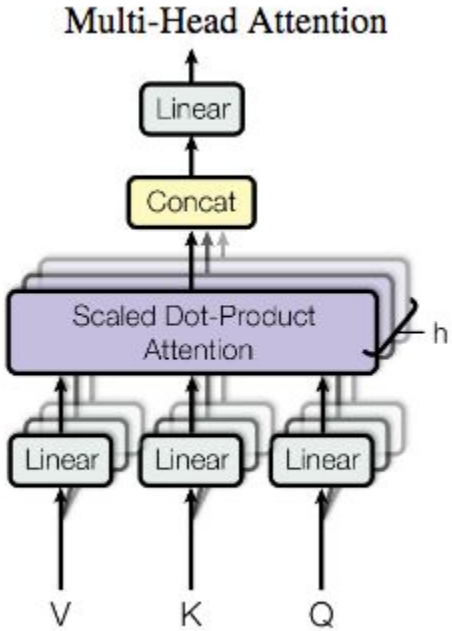

Output Probabilities
Softmax
Linear
decoder
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Encoder
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted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

# Attention drawback

- Convolution: weights * input. Each weights are different. So position is encoded.
- Self-attention: a weighted average. Position information is lost at the output

# Multi-head attention




What's this????

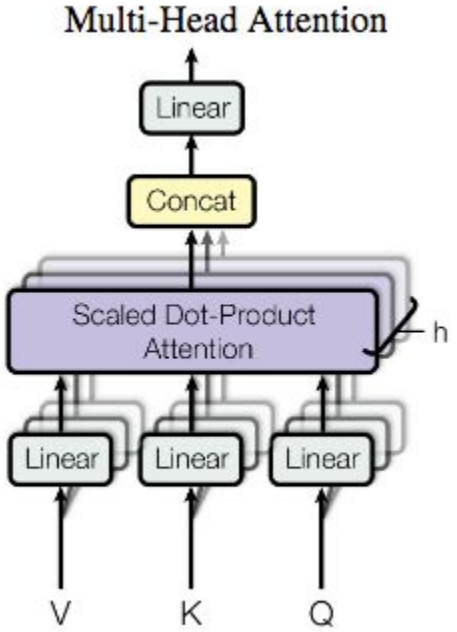
Query – used with Key to determine the position
Value – used as the information after determining the position

# Attention drawback

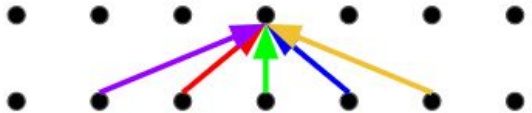
- Convolution: weights * input. Each weights are different. So position is encoded.
- Self-attention: a weighted average. Position information is lost at the output

# Multi-head attention

- Multiple attention layers (heads) that run in parallel
- Each head use different weights
- Each head can learn different relationship

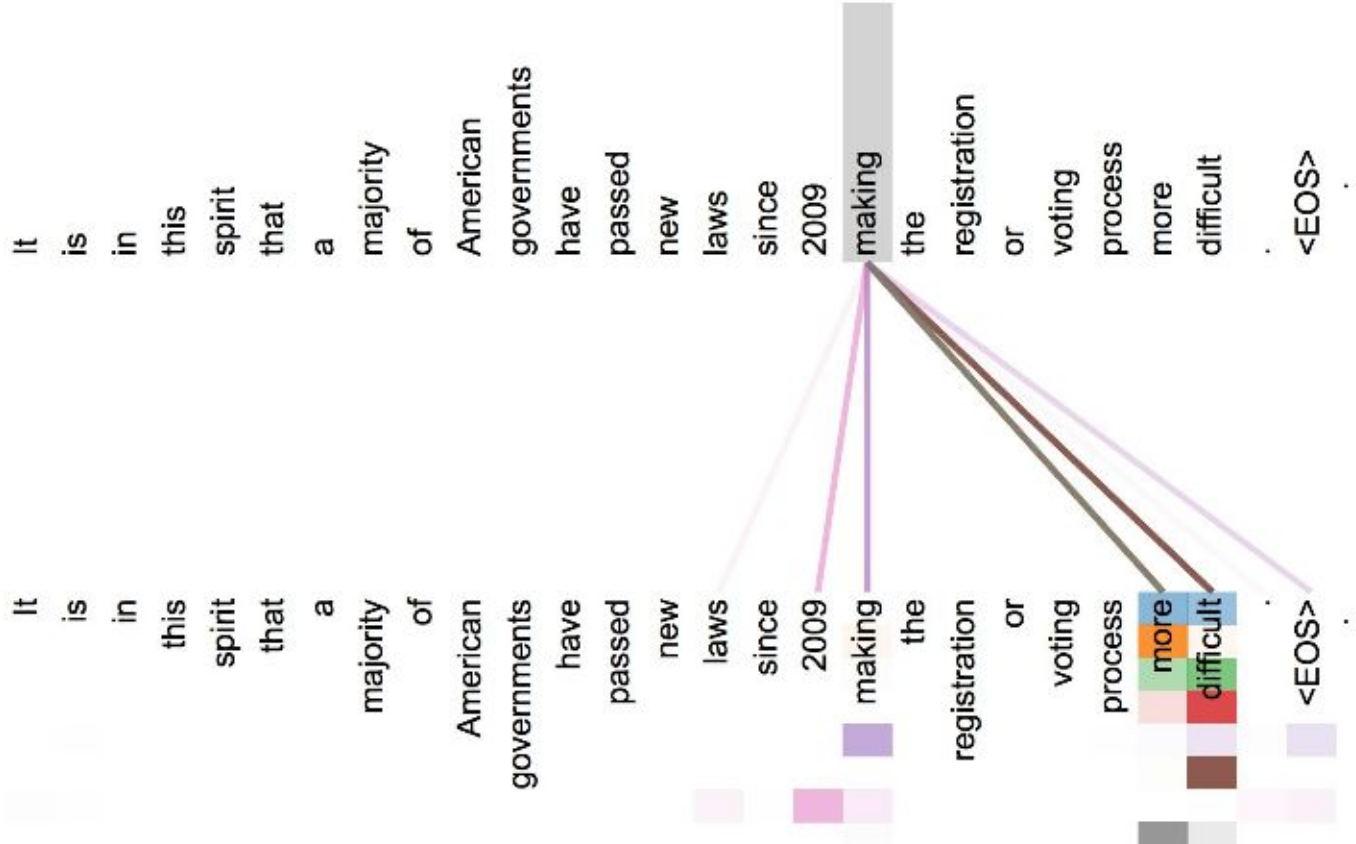


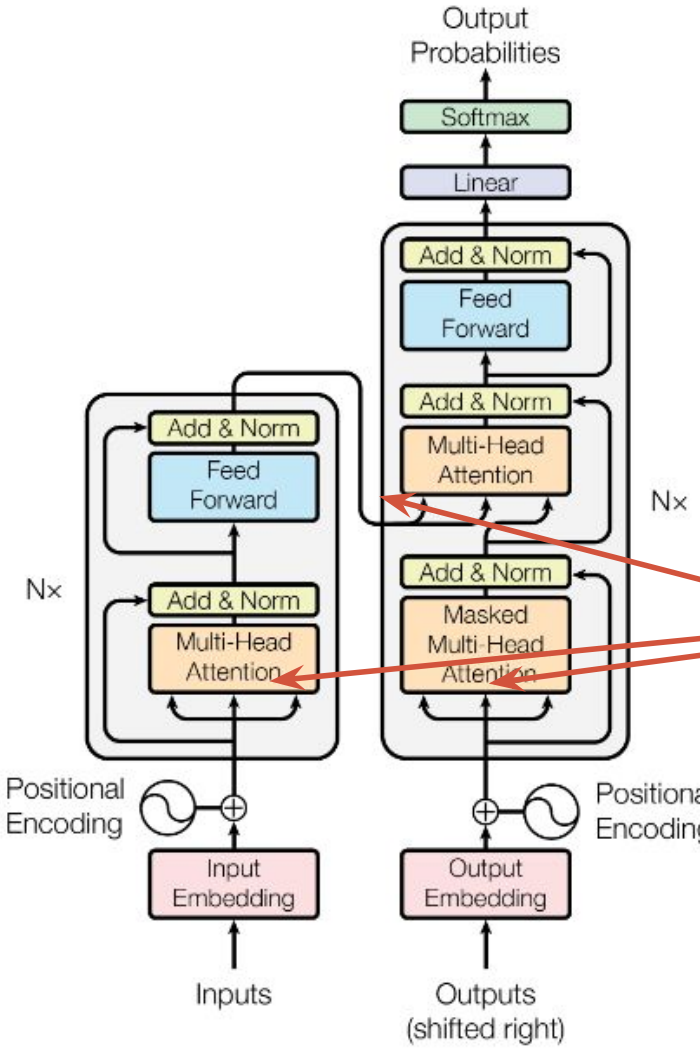

Convolution

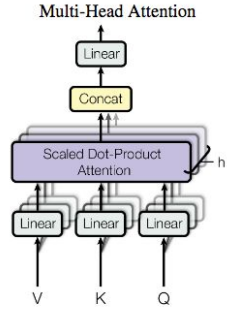


Multi-Head Attention

# Multi-head visualization

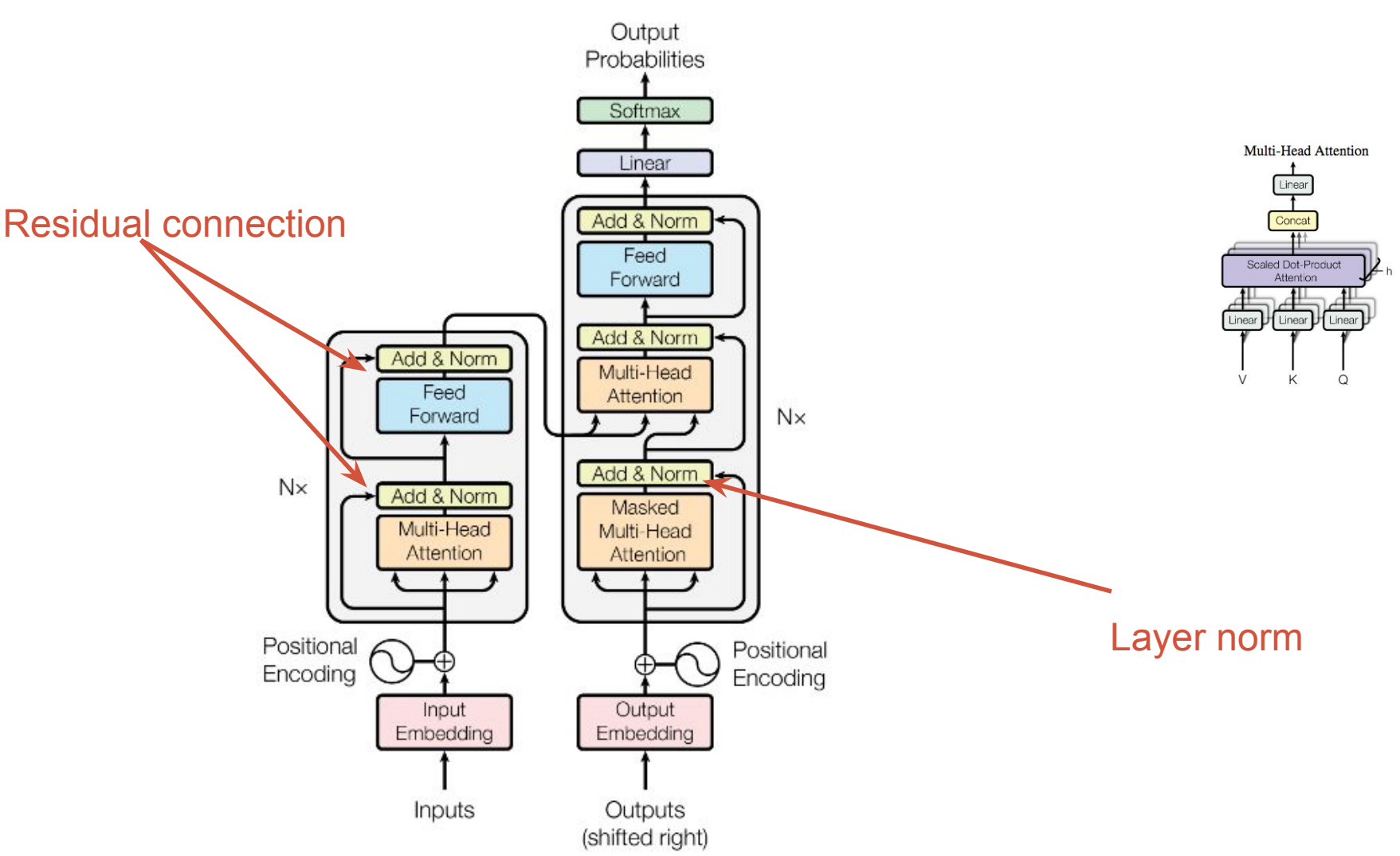





Self attention

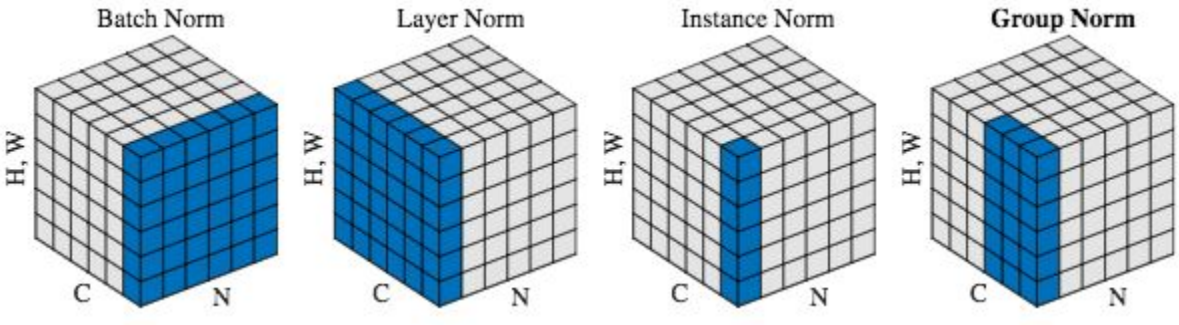
Encoder gives Value and Key
The decoder gives Query


Residual connection
Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)
Layer norm
Multi-Head Attention
Linear
Concat
Scaled Dot-Product Attention
h
Linear
Linear
Linear
V
K
Q

# Layer norm

- Normalize the mean and SD
- Batch norm vs layer norm vs Instance norm vs group norm

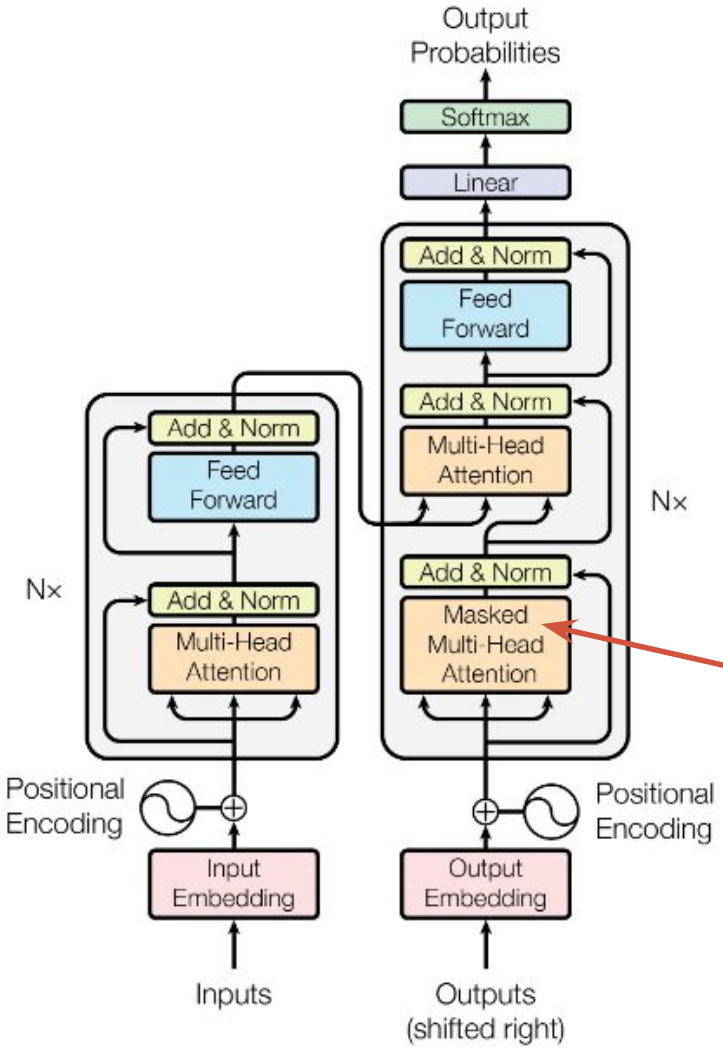
Group is used to distributed models into multiple GPUs

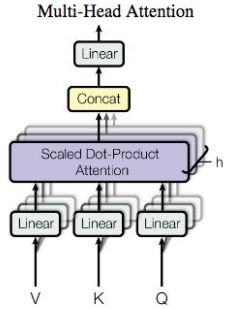



N – example in mini batch
C – Channel output
H,W – spatial coordinates (x,y)

Box is output tensor from CNN

BN and GN are usually best, GN is better when batch size is small (Vision task)
https://arxiv.org/abs/1803.08494




Prevent looking ahead (cheating)

# MT results

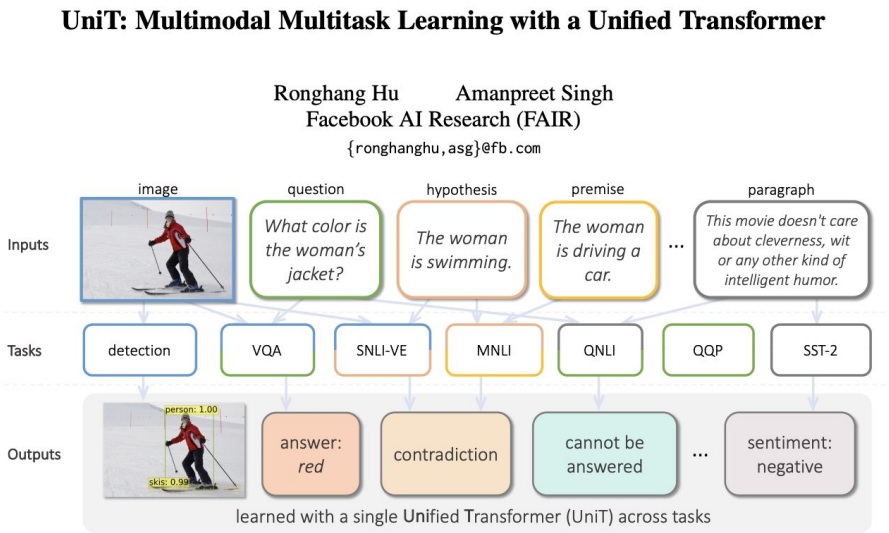
Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

| Model | BLEU | | Training Cost (FLOPs) | |
|---|---|---|---|---|
| | EN-DE | EN-FR | EN-DE | EN-FR |
| ByteNet [18] | 23.75 | | | |
| Deep-Att + PosUnk [39] | | 39.2 | | $1.0 \cdot 10^{20}$ |
| GNMT + RL [38] | 24.6 | 39.92 | $2.3 \cdot 10^{19}$ | $1.4 \cdot 10^{20}$ |
| ConvS2S [9] | 25.16 | 40.46 | $9.6 \cdot 10^{18}$ | $1.5 \cdot 10^{20}$ |
| MoE [32] | 26.03 | 40.56 | $2.0 \cdot 10^{19}$ | $1.2 \cdot 10^{20}$ |
| Deep-Att + PosUnk Ensemble [39] | | 40.4 | | $8.0 \cdot 10^{20}$ |
| GNMT + RL Ensemble [38] | 26.30 | 41.16 | $1.8 \cdot 10^{20}$ | $1.1 \cdot 10^{21}$ |
| ConvS2S Ensemble [9] | 26.36 | **41.29** | $7.7 \cdot 10^{19}$ | $1.2 \cdot 10^{21}$ |
| Transformer (base model) | 27.3 | 38.1 | **$3.3 \cdot 10^{18}$** | |
| Transformer (big) | **28.4** | **41.8** | $2.3 \cdot 10^{19}$ | |

Can use for other tasks, like ASR, parsing, etc.

# Trends

- Transformer is expensive
- Transformer replaces CNN-based architecture
- Transformer is all you need




UniT: Multimodal Multitask Learning with a Unified Transformer, 2021

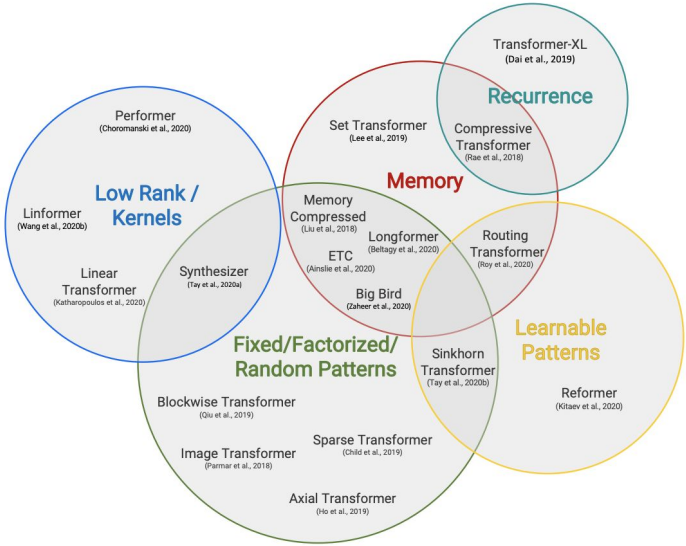
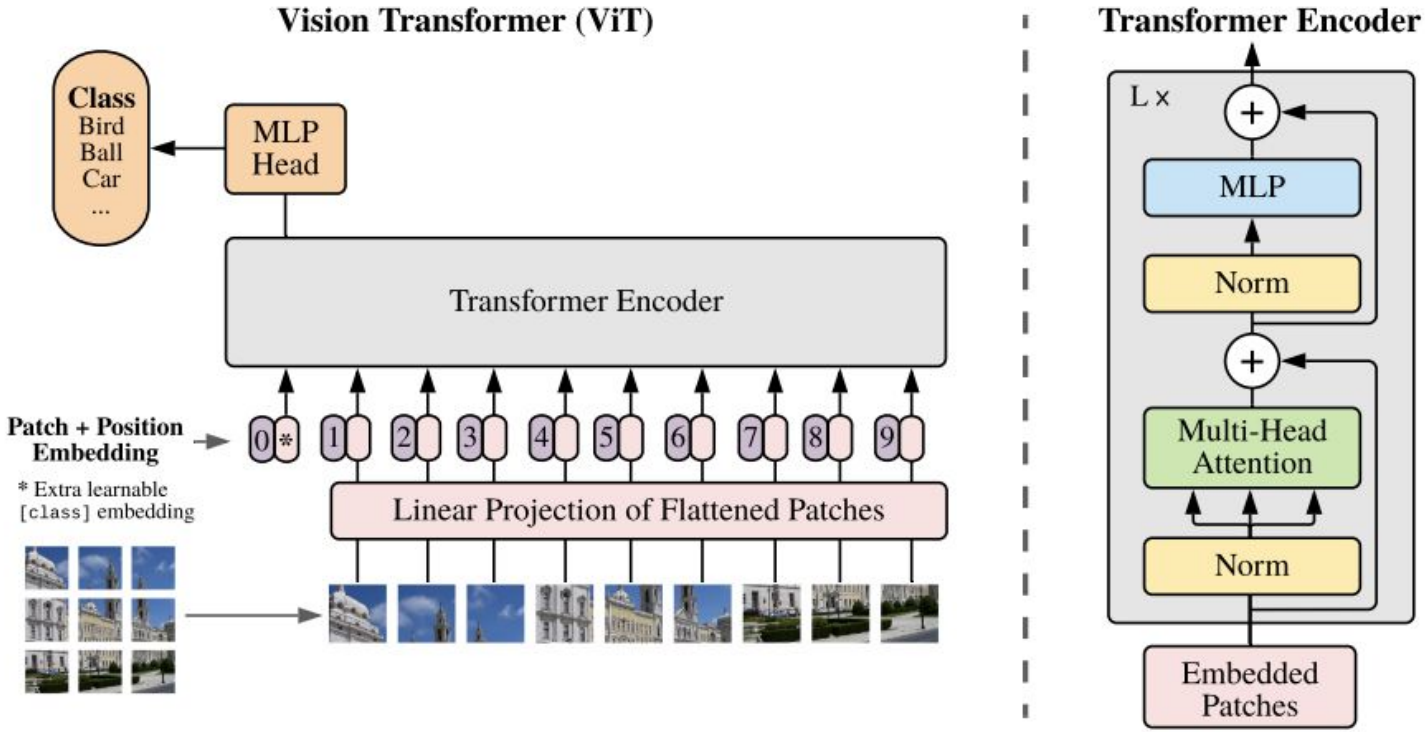



Figure 2: Taxonomy of Efficient Transformer Architectures.

Efficient Transformers: A Survey, 2020

# Vision transformer




AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE
https://arxiv.org/pdf/2010.11929v1.pdf

# Dall E

GPT3 for images
Cluster image patches into tokens and combined with text to train an autoregressive model


(a) a tapir made of accordion. a tapir with the texture of an accordion.

(b) an illustration of a baby hedgehog in a christmas sweater walking a dog

(c) a neon sign that reads "backprop". a neon sign that reads "backprop". backprop neon sign

(d) the exact same cat on the top as a sketch on the bottom

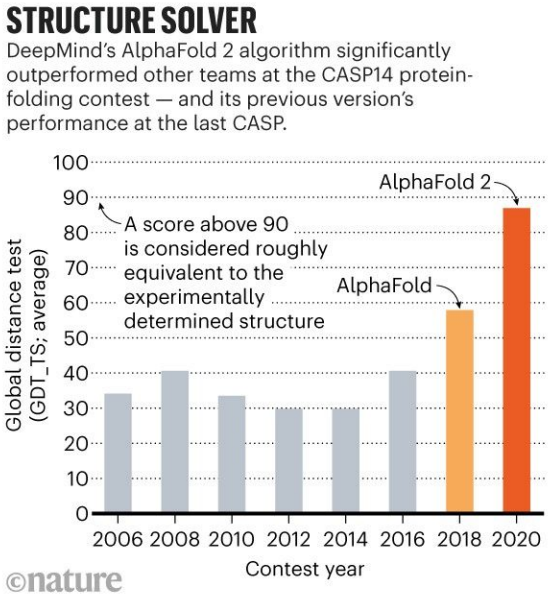
*Figure 2.* With varying degrees of reliability, our model appears to be able to combine distinct concepts in plausible ways, create anthropomorphized versions of animals, render text, and perform some types of image-to-image translation.

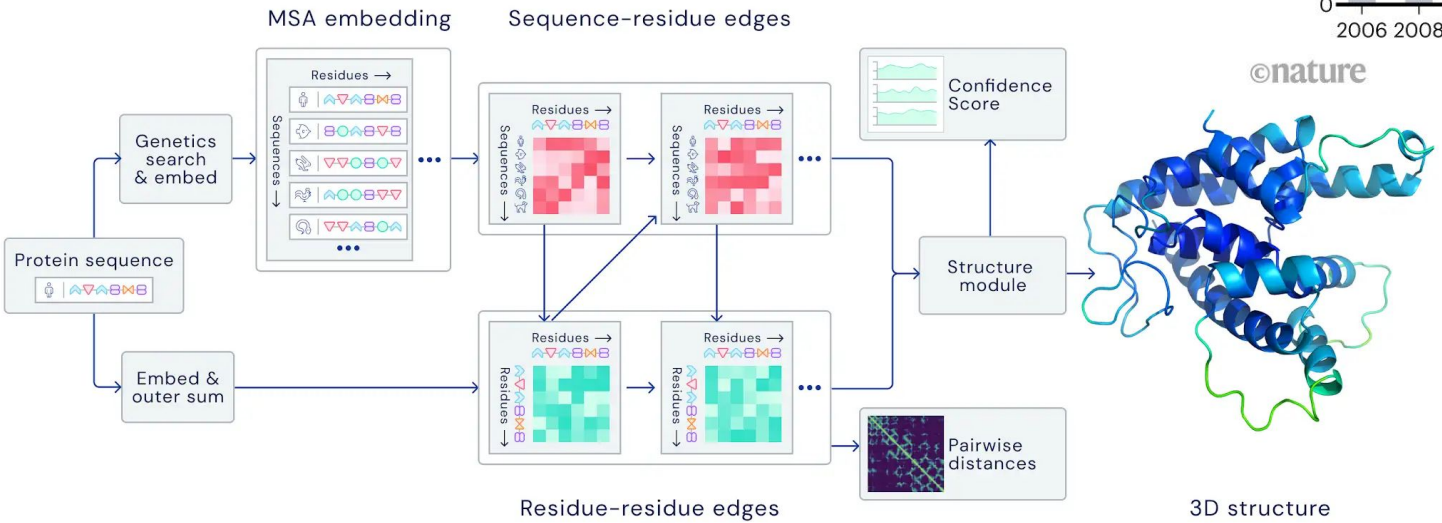
https://openai.com/blog/dall-e/
https://arxiv.org/pdf/2102.12092.pdf

# Alphafold2

https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology

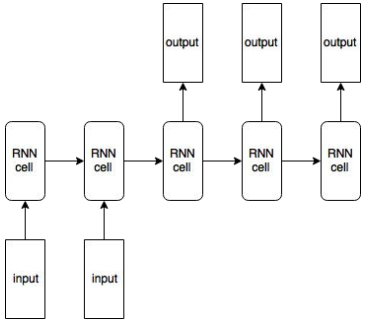
https://www.nature.com/articles/d41586-020-03348-4

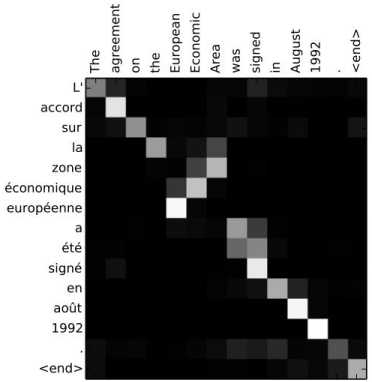


# Outline

## Text Generation & Attention Mechanism




## Transformer