Transformer for AI
Week 2: Attention Mechanisms for transformers

Stanley Liang, PhD

Research Fellow, NLM

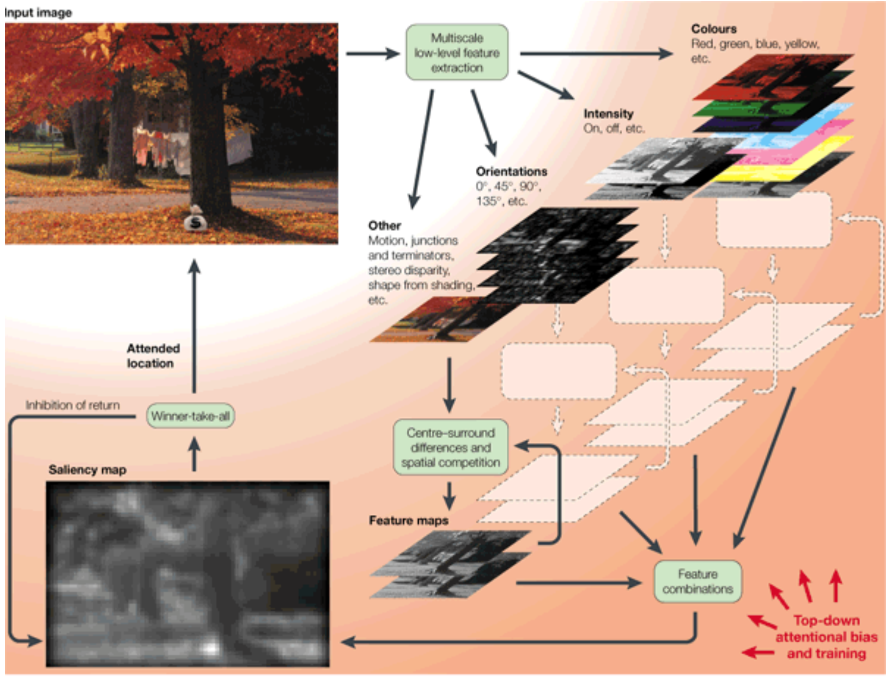
The concept of attention

Human Cognition

- *Attention is an overall level of alertness or ability to engage with surroundings*
- *Human brain has limited memory, it relies on attention to dynamically store the information it pays attention to*

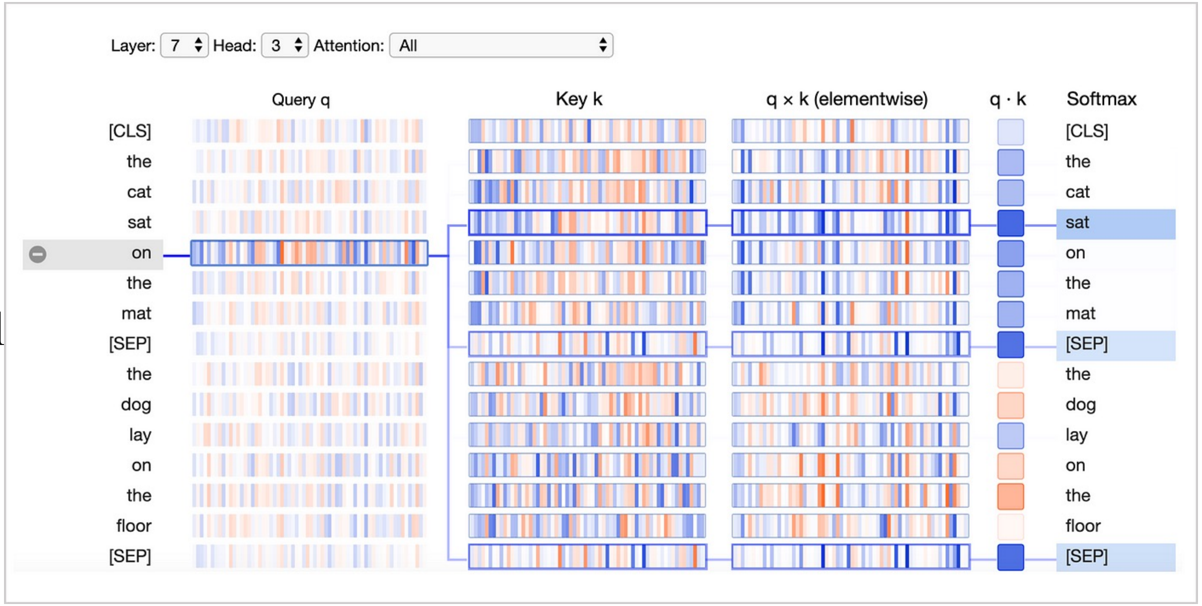
Computational cognition

- *To dynamically highlight which of the input information will be used to generate the output*
- *A mechanism to highlight the salient information across the entirety of the input*



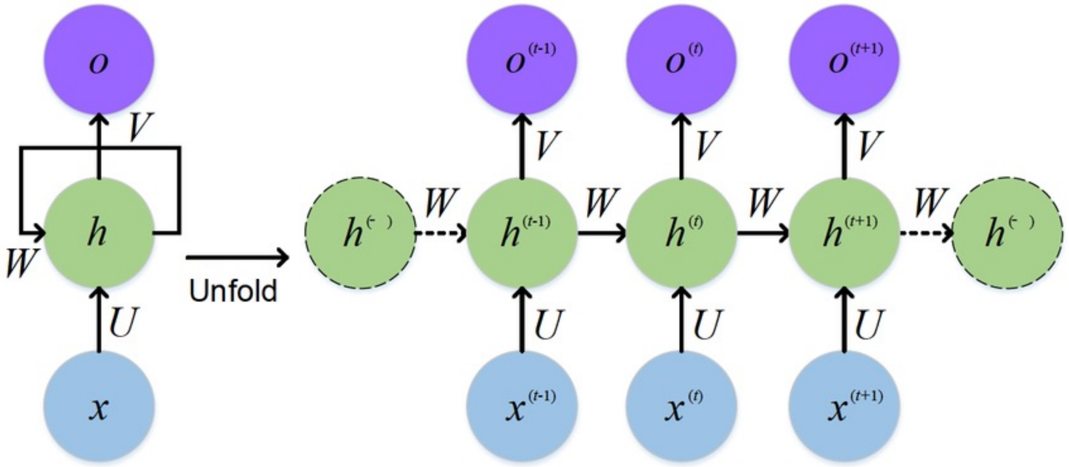
Nature Reviews | Neuroscience

Human brain attends to these salient visual features at different neuronal stages



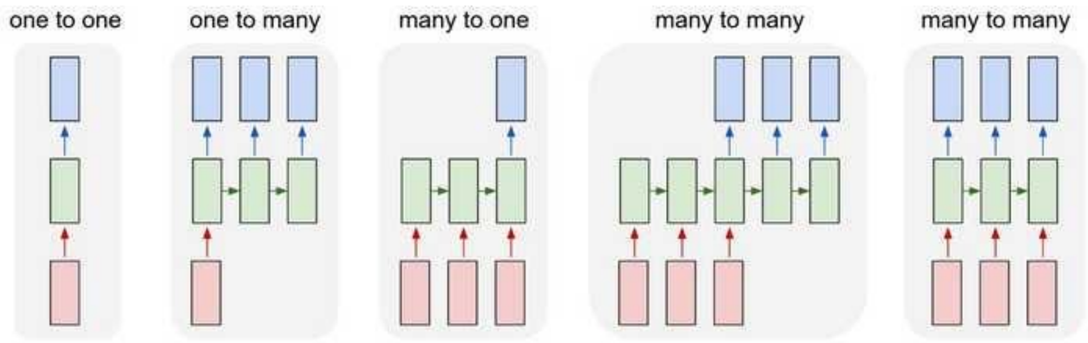
General idea of RNN

- Recurrent neural network is a special type of NN for sequence data
- In sequence data, the current data point depends upon the previous data – incorporate the dependencies between data points
- RNN use special memory to store the information states of previous inputs for the next output
- Gradient vanishing: the gradients can become increasingly small – deep architecture
- Gradient exploding: too large gradients during backpropagation – unstable training, weights -> NaN

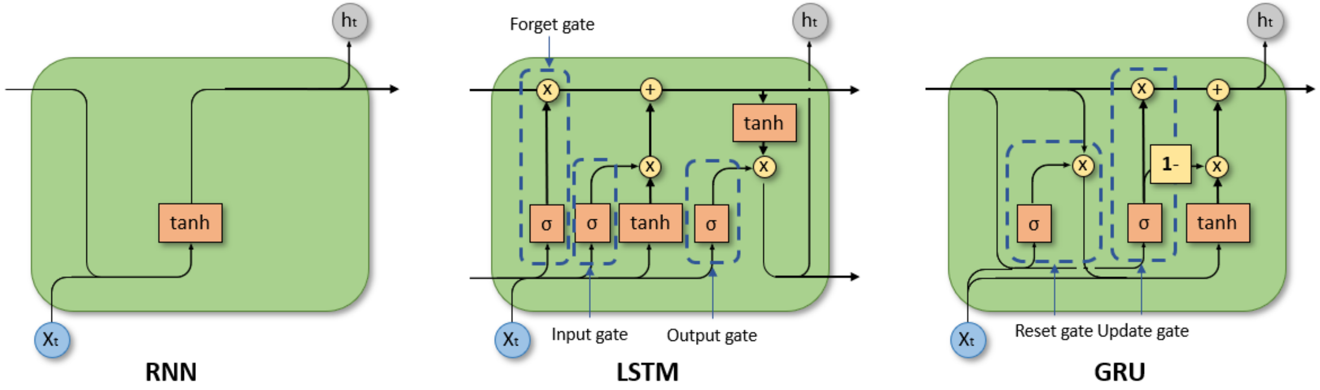


Types of RNN

- One-to-One: MLP
- One-to-Many: music generation
- Many-to-One: sentiment analysis / emotion detection
- Many to Many: translation

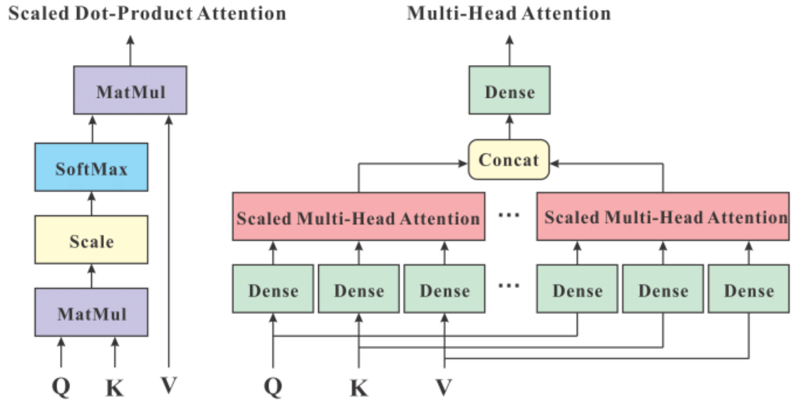


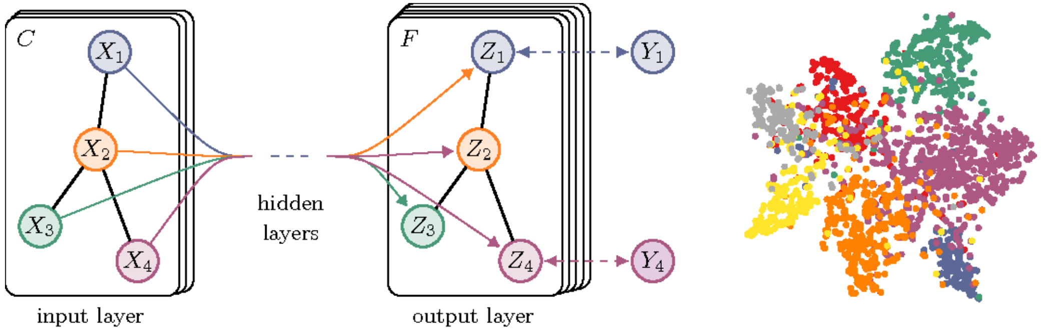
- Bidirectional recurrent neural network (BRNN): predict the middle words
- Gated Recurrent Units (GRU): reset & update gates
- Long Short-term Memory (LSTM): input, output, forget gates



Encoder-decoder architecture

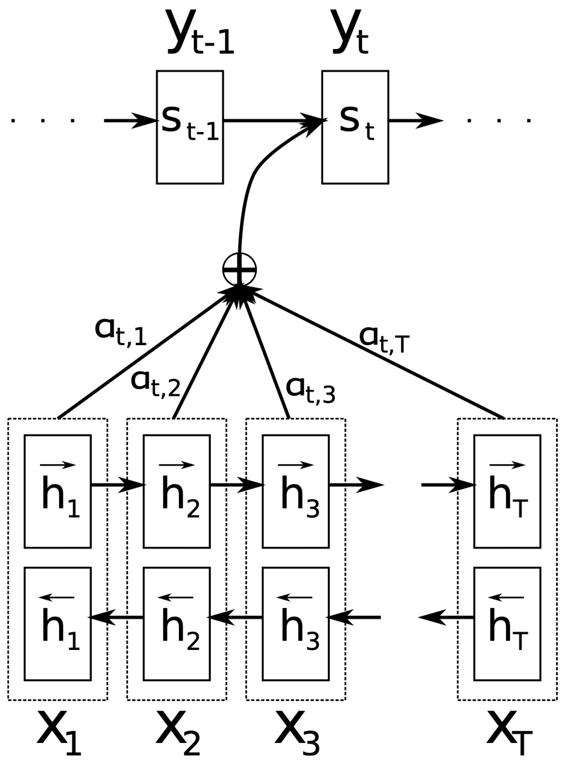
- Encoder-decoder for sequence-to-sequence (Seq2Seq) tasks
- Recurrent neural network + attention to encode long sentences.
- The transformer scaled dot-product attention
  - Recurrence → attention
- Graph Attention Networks (GAT)
  - Feature by nodes
  - relations by edges





(a) Graph Convolutional Network

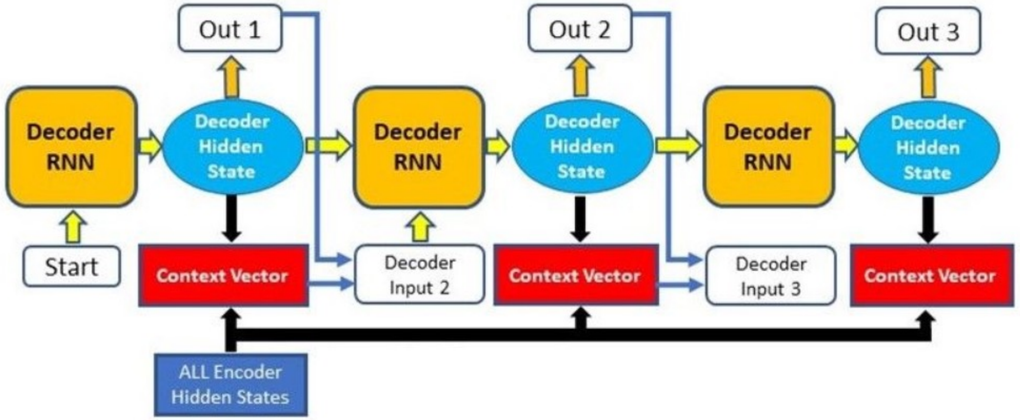
(b) Hidden layer activations



Bahdanau attention, 2014

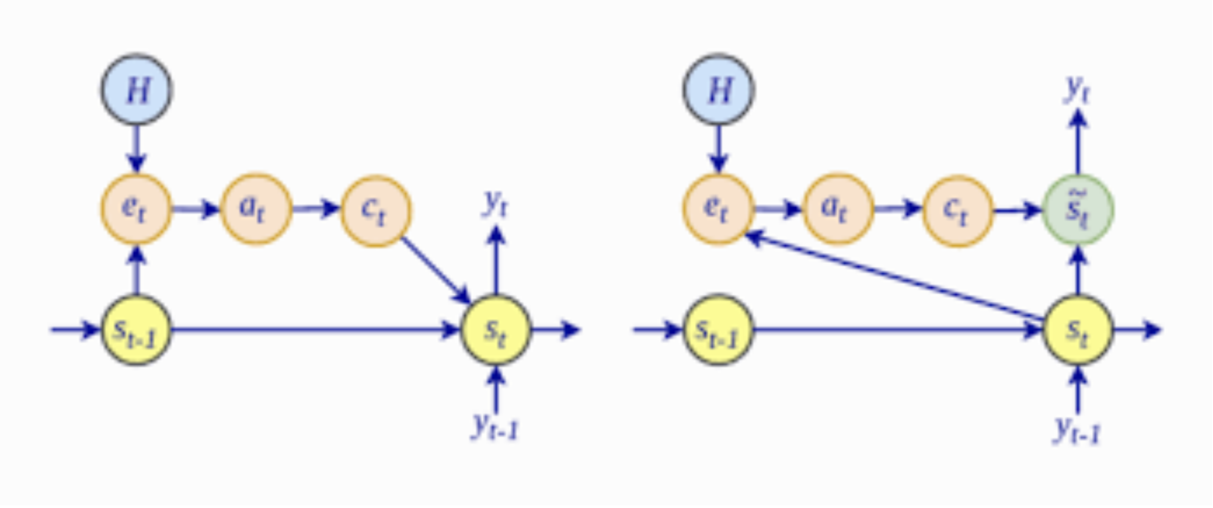
RNN with attention

- Conventional RNN encoder-decoder encodes the input sequence into a fixed-length vector → performance decrease as the sequence becomes longer
- Add Bahdanau attention to RNN
  - Can focus on relevant parts of the input sequence
  - Improve the capacity to handle long sequence
  - Explainable: can highlight which part of the input sequence being focused on for each output
- Bahdanau attention has been a foundational model for many subsequent attention mechanisms

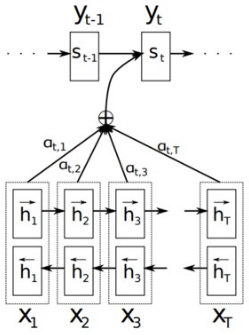


Luong Attention Mechanism
--Improvement of Bahdanau attention

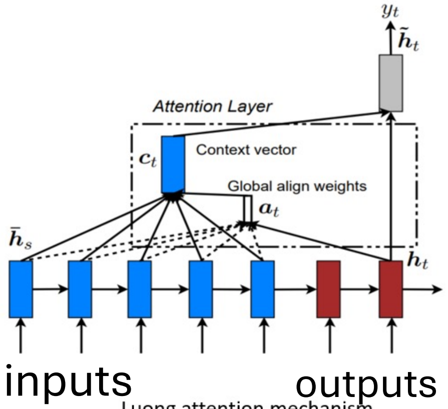
- Bahdanau
  - context vectors attached to each hidden state
  - Use bidirectional encoder
- Luong
  - Global attention – soft attention
  - Local attention – hard attention
  - Use LSTM for encoder & decoder
  - The alignment score $e_t \rightarrow current\ S_t$



Bahdanau Luong



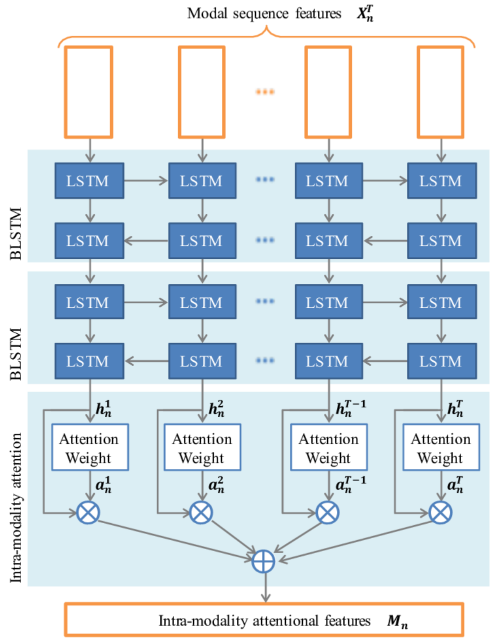
Bahdanau attention mechanism
2014



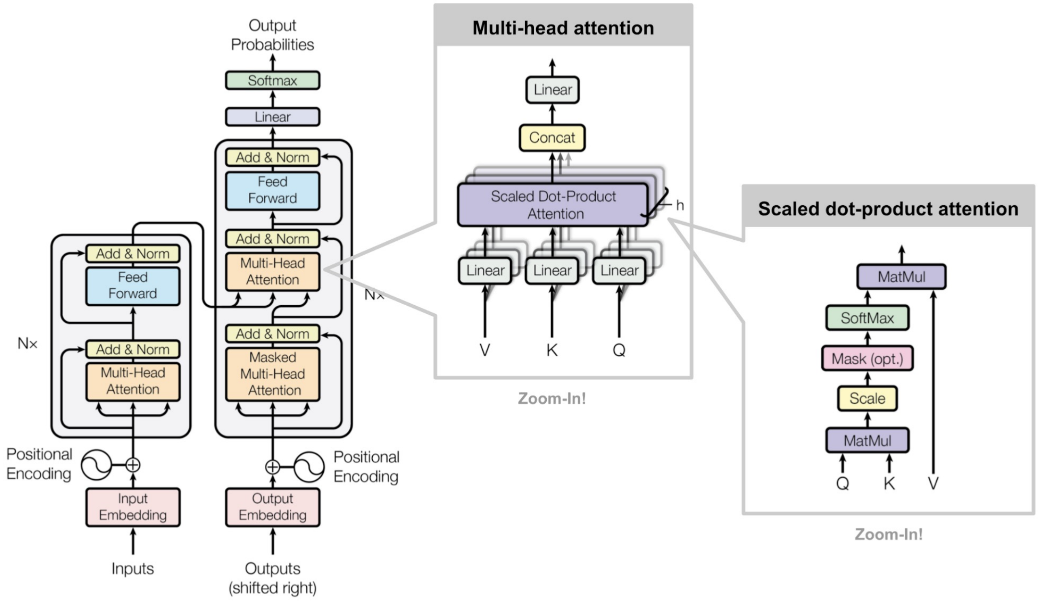
Luong attention mechanism
2015

Transformer Attention

- RNN attention (Bahdanau & Luong) – attention mechanism in conjunction with RNN
- Transformer attention – attention by dispensing with recurrence and convolution – self attention
- Transformer attention outputs a weighted sum of values based on a compatibility function of query with corresponding key



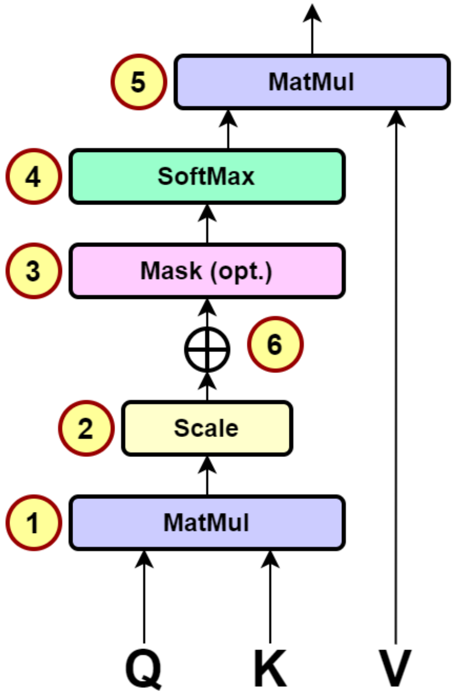
RNN attention

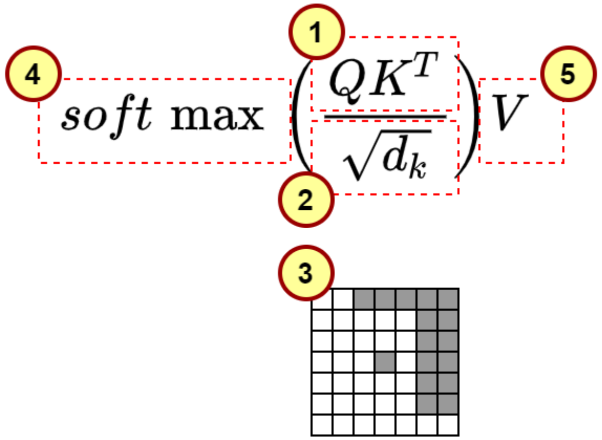


Transformer attention

Scaled dot-product attention

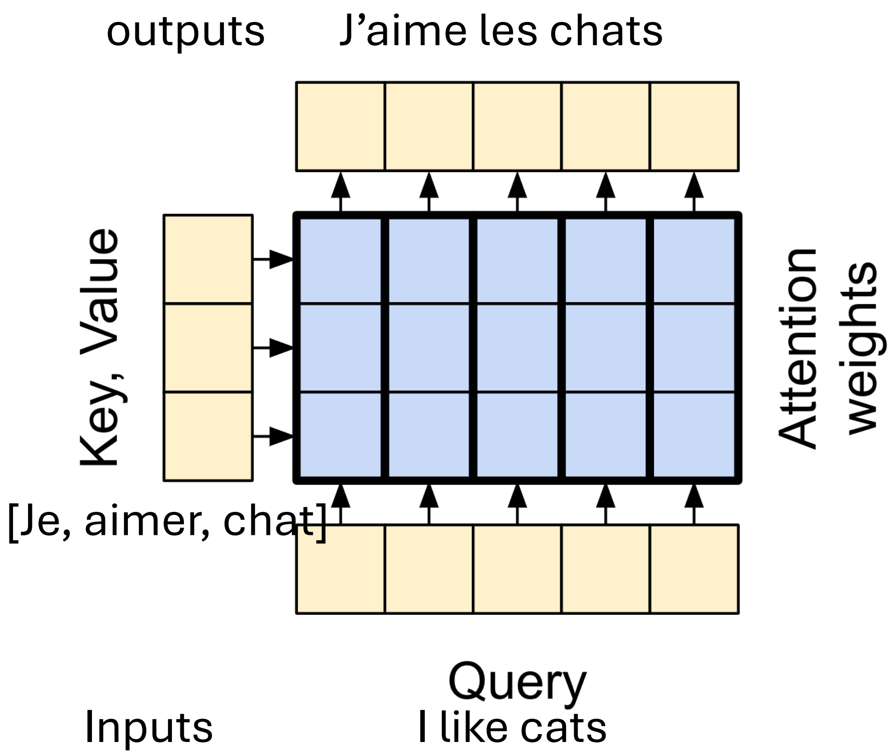
- Query (Q): the sequence being processed
- Key (K): the index pointing to the sequence being attended
- Value (V): the detailed information of the sequence being attended
- Scale: average the weight for tokens
- Mask: prevent from succeeding information





Query, Key, Value

- Query, key, and value are analogy of searching a dictionary
- The attention is like a fuzzy, differentiable, vectorized dictionary lookup
- Query – what you try to find
- Key – information in the dictionary
- Value – the information you actually want

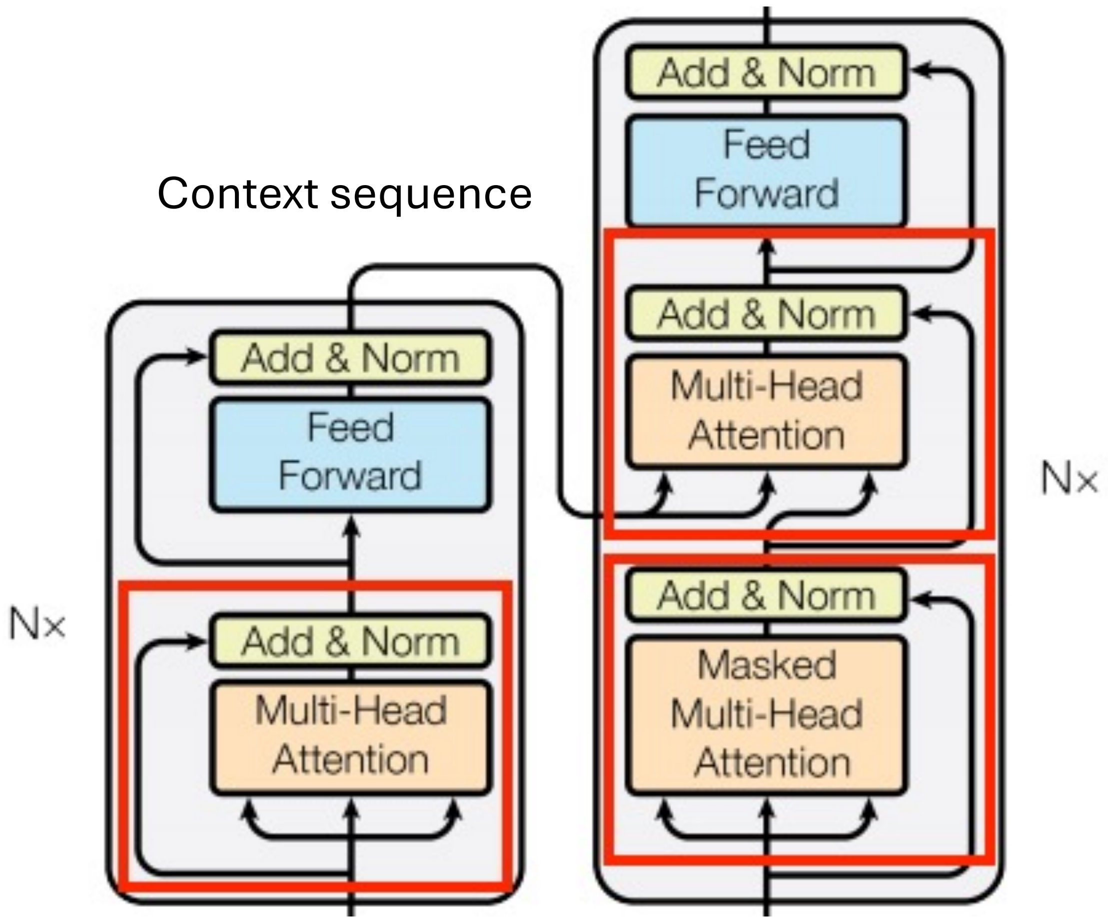


Base attention

- The query searches the key in a dictionary and returns the value
- The input sequence – query vector
- The context sequence – key, value vector
- The attention ”dot” query and key -> attention score – determine the degree of matching
- The query is what you're trying to find.
- The key is the information the dictionary has.
- The value is that information.

Predicted / output sequence

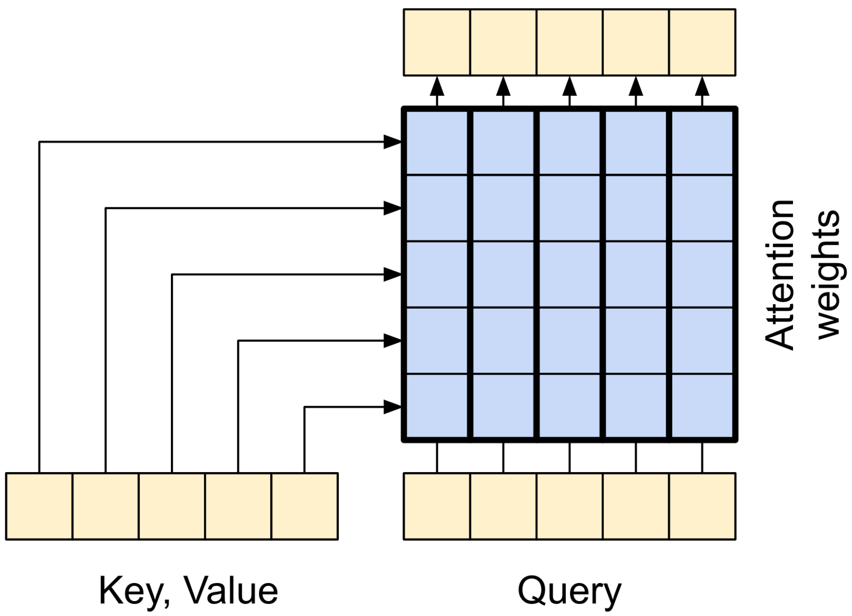
Context sequence

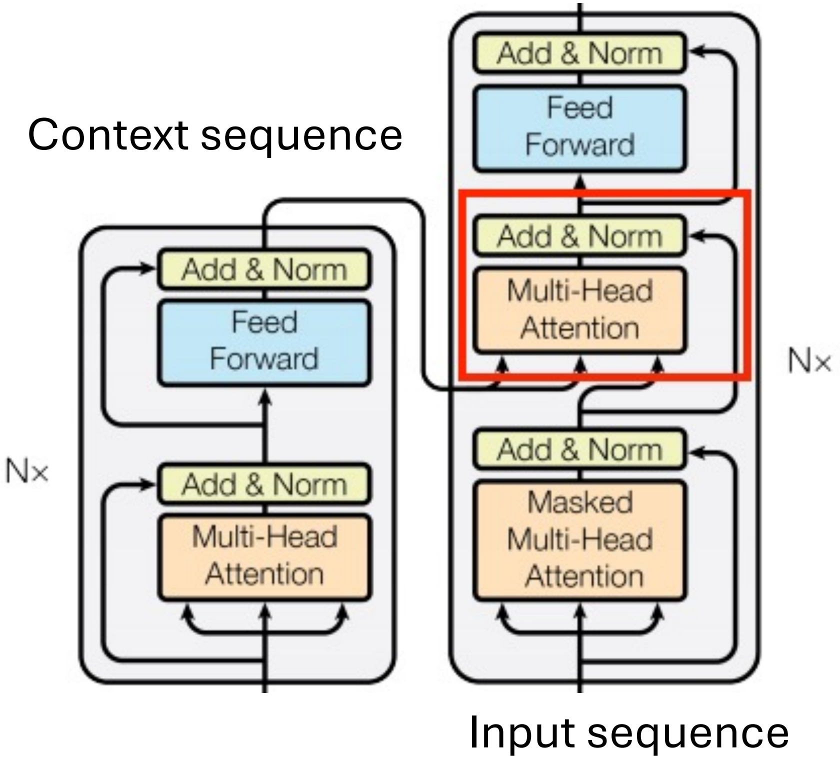


Input sequence

Cross attention

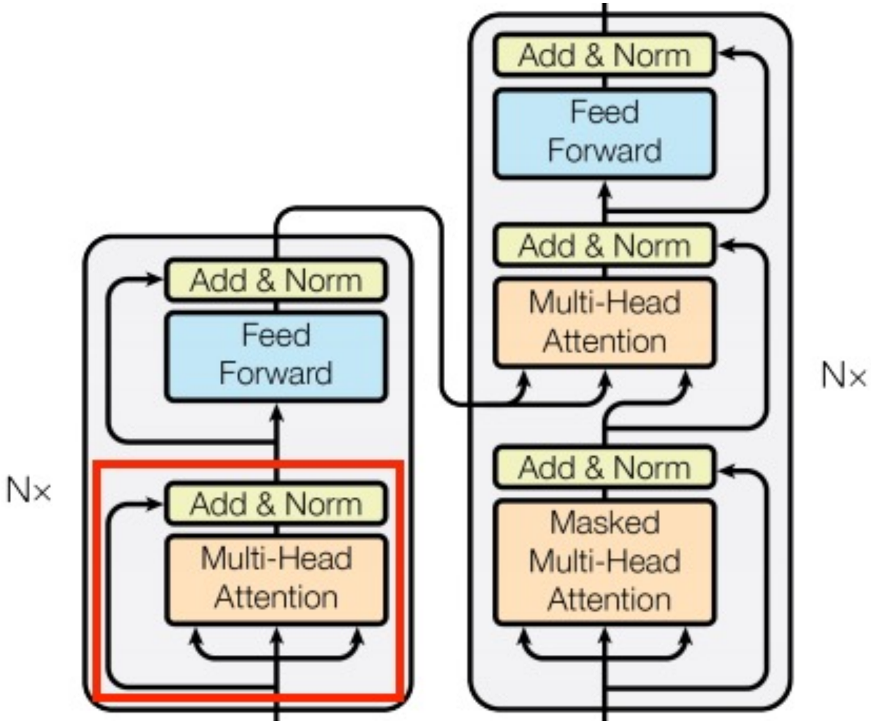
- The attention connects the encoder and the decoder
- Query – input sequence
- Key & value – context sequence

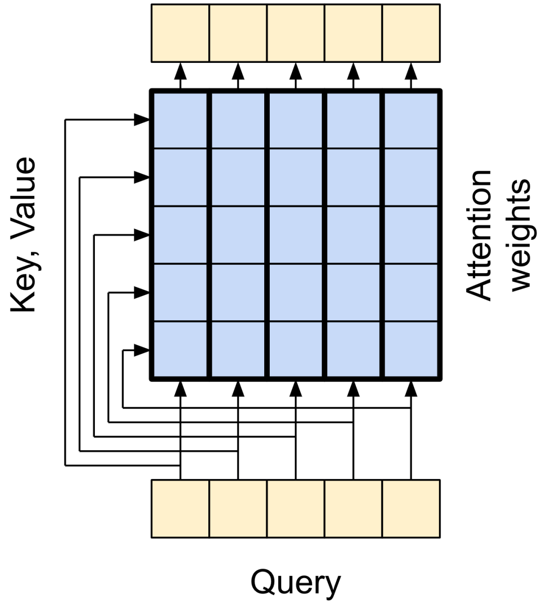




Global self-attention

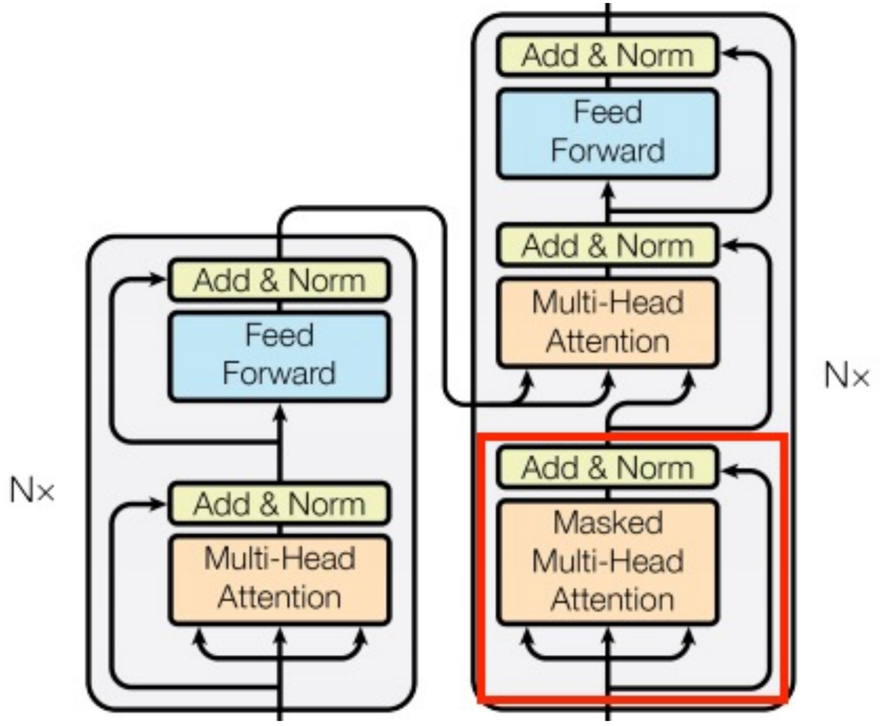
- Processing the context sequence
- propagating information along architecture
- Context sequence is fixed - bidirectional is allowed
- Query – input sequence
- Key & value – context sequence
- RNN – need to run steps sequentially, takes no advantage of parallel device
- CNN – parallel computing is feasible, but limited to linear receptive fields

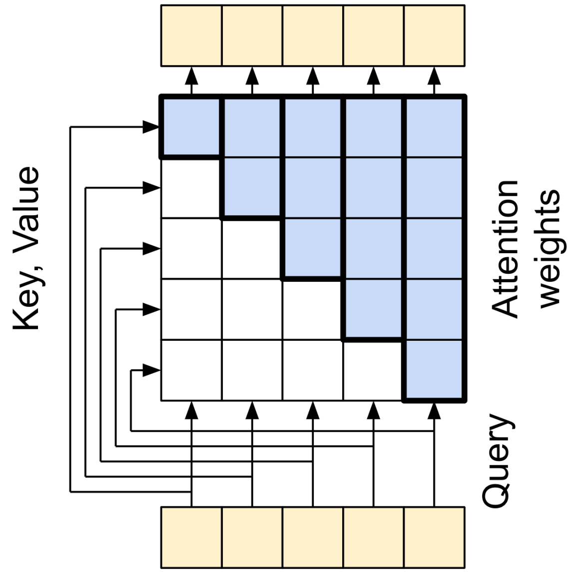




Causal self-attention

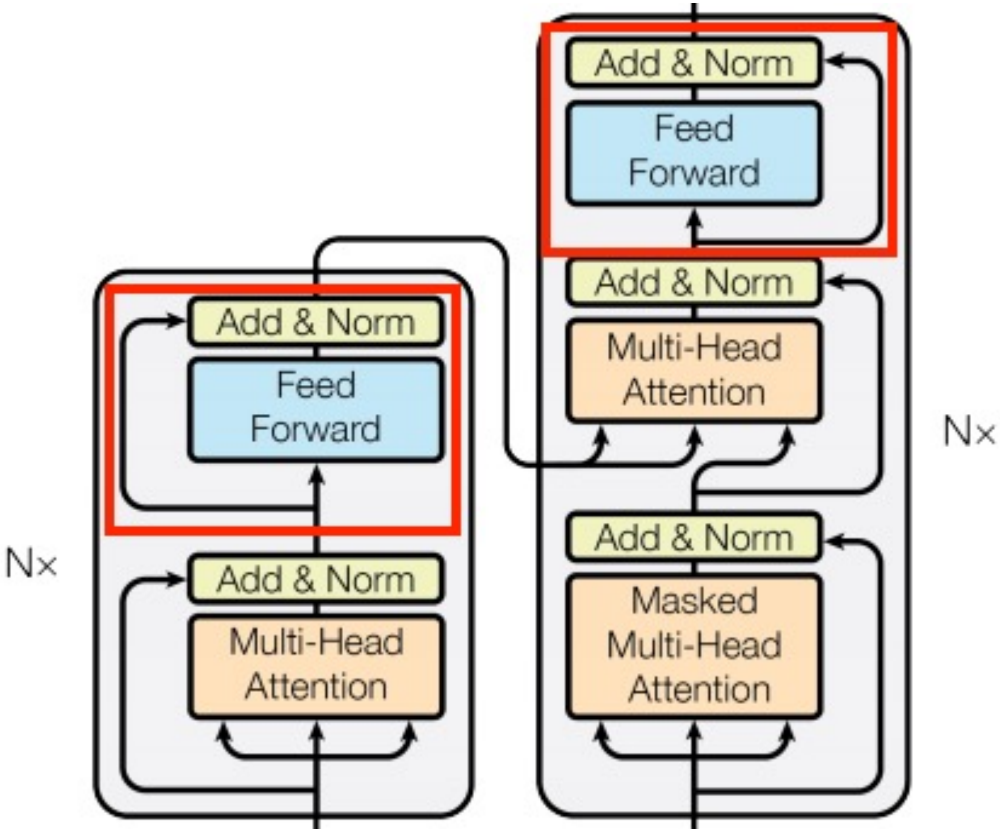
- For output sequence (decoder)
- Current sequence element is dependent on the previous elements – causal
- efficiency: compute loss of all locations in single execution
- The previous tokens can be reused for every next-token generation
- Need mask to conceal the unseen information – unidirectional
- Query, key, value – input sequence



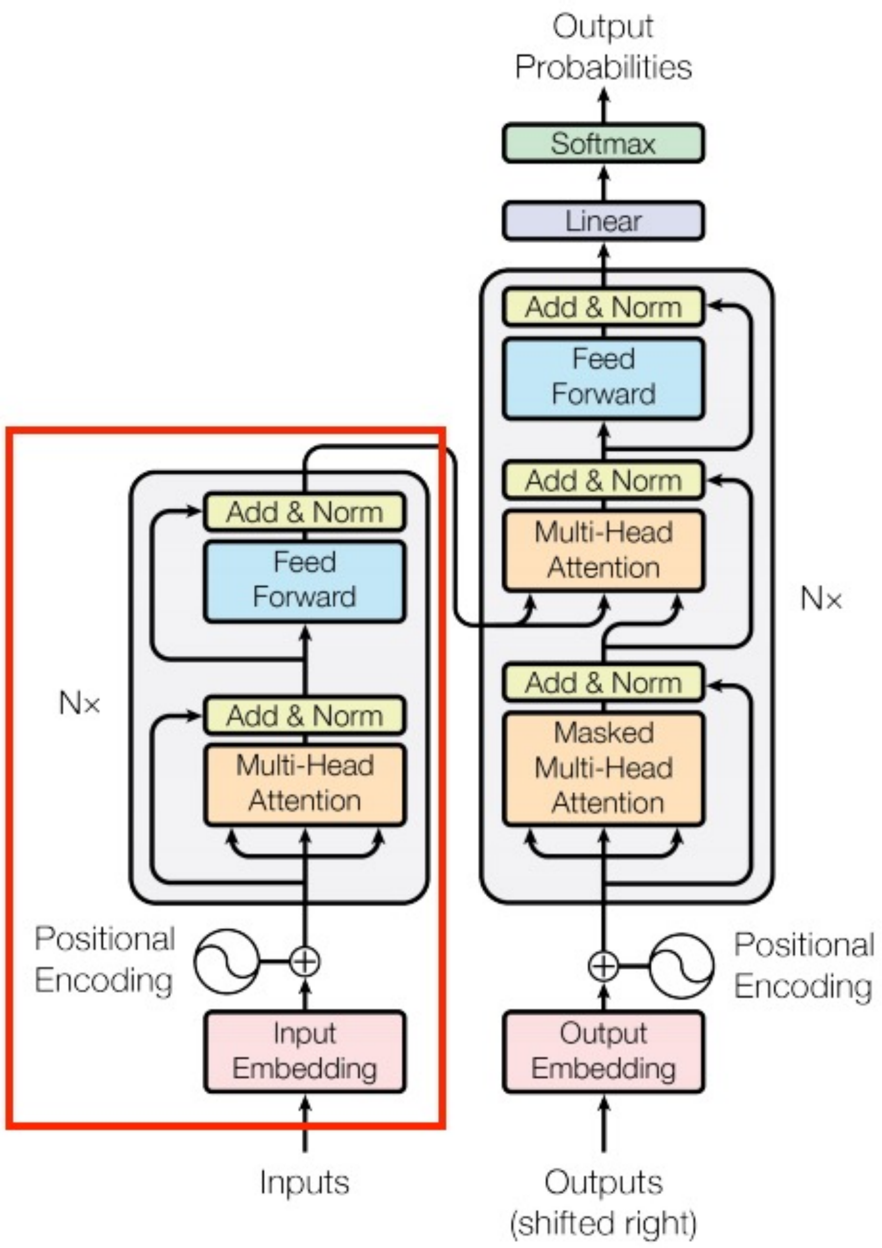


Feed forward network architecture

- Feed-forward network in both the encoder and decoder
- Consists of two linear layers with ReLU activation between them
- Include the residual connection and normalization

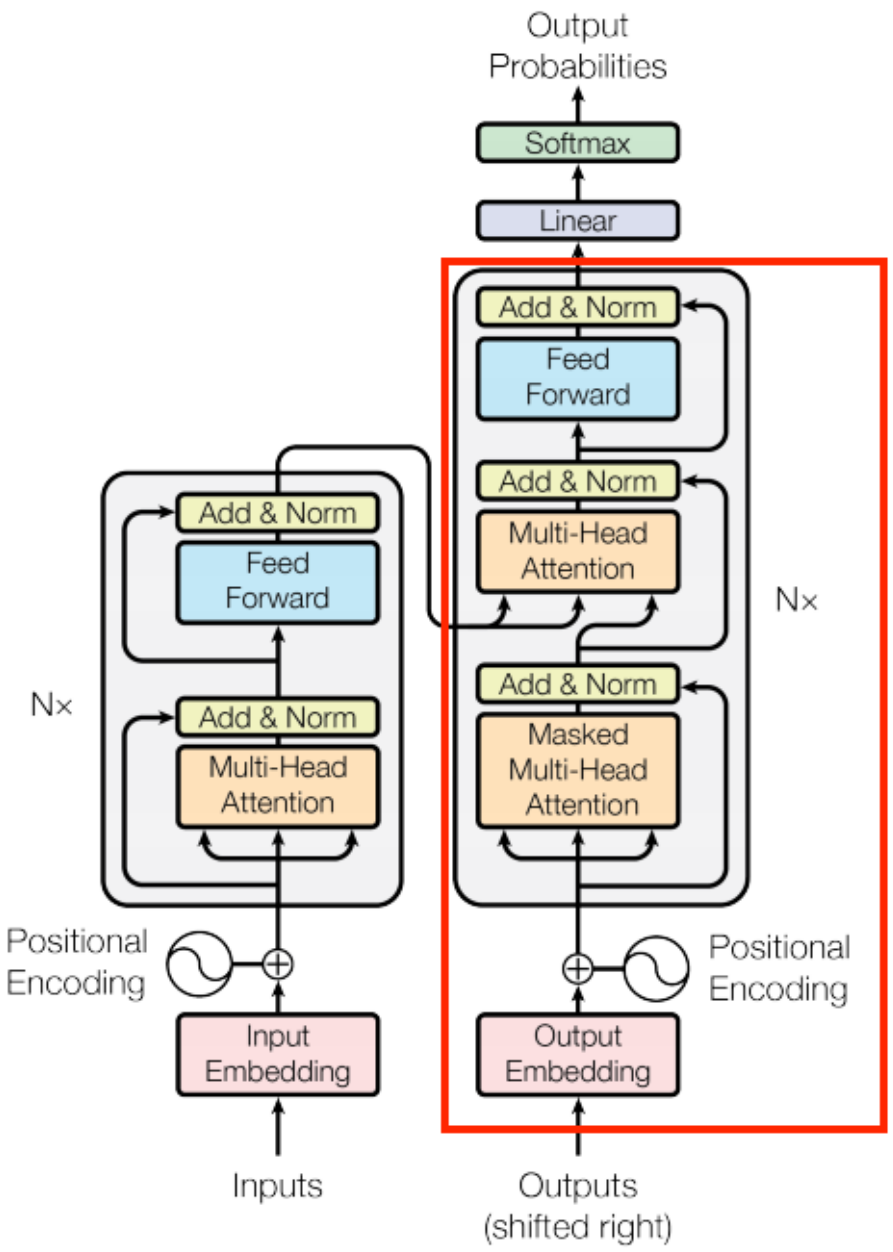


The encoder



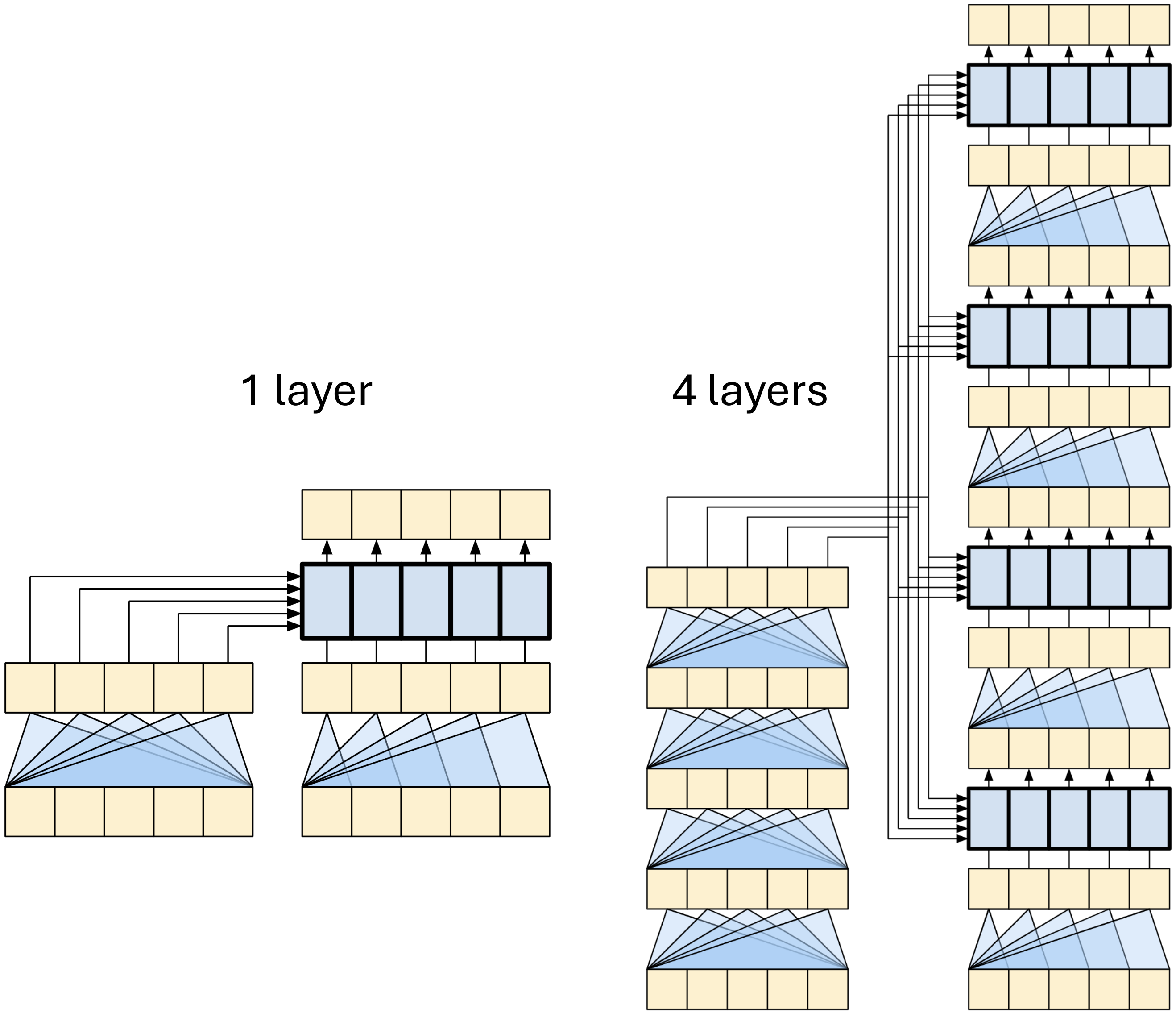
Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Add & Norm
Masked Multi-Head Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Input Embedding
Inputs
Positional Encoding
Output Embedding
Outputs (shifted right)

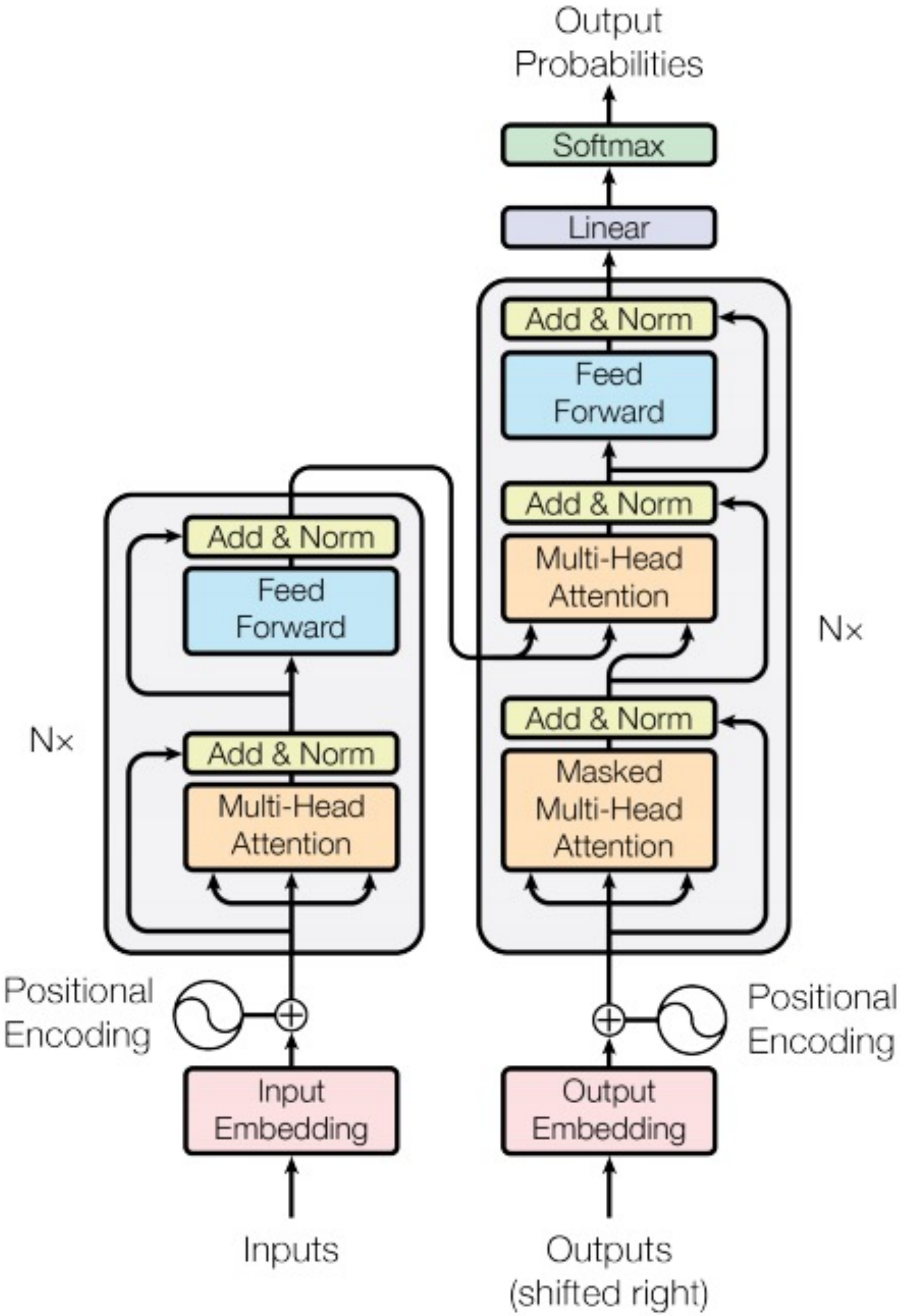
The decoder



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

The transformer architecture





This is the end

- Notebook to view
  - Implementation_self_attention
  - RNN_with_attention
  - Scaled_dot_product_attention
  - Form_transformer
- Paper to read
  - Attention is all you need
  - A survey of transformer
  - Attention in Psychology, Neuroscience, and Machine Learning