

CS 4824/ECE 4424: Attention and Transformers

Acknowledgement:

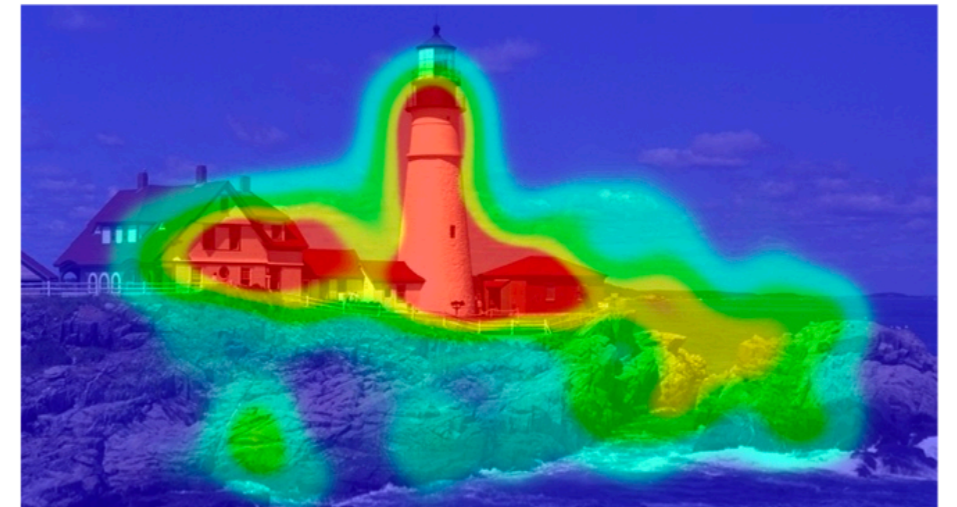
Many of these slides are derived from Tom Mitchell, Pascal Poupart, Pieter Abbeel, Eric Eaton, Carlos Guestrin, William Cohen, and Andrew Moore.

Attention

- **Key idea:** highlight important parts of the inputs
- Mechanism for alignment in machine translation, image captioning, etc.
- Attention in machine translation: align each output word with relevant input words by computing a softmax of the inputs

Attention

- Attention in Computer Vision
 - 2014: Attention used to highlight important parts of an image that contribute to a desired output



- Attention in NLP
 - 2015: machine translation
 - 2017: Language modeling with **Transformer networks**

Sequence Modeling

- **Challenges with RNNs**

- Long range dependencies
- Gradient vanishing (and explosion)
- Large # of training steps
- Recurrence prevents parallel computation

vs

- **Transformer Networks**

- Facilitate long range dependencies
- No gradient vanishing (and explosion)
- Fewer training steps
- No recurrence that facilitate parallel computation

Attention Mechanism

- Mimics the retrieval of a **value** v_i for a **query** q based on a **key** k_i in database
- Retrieval:

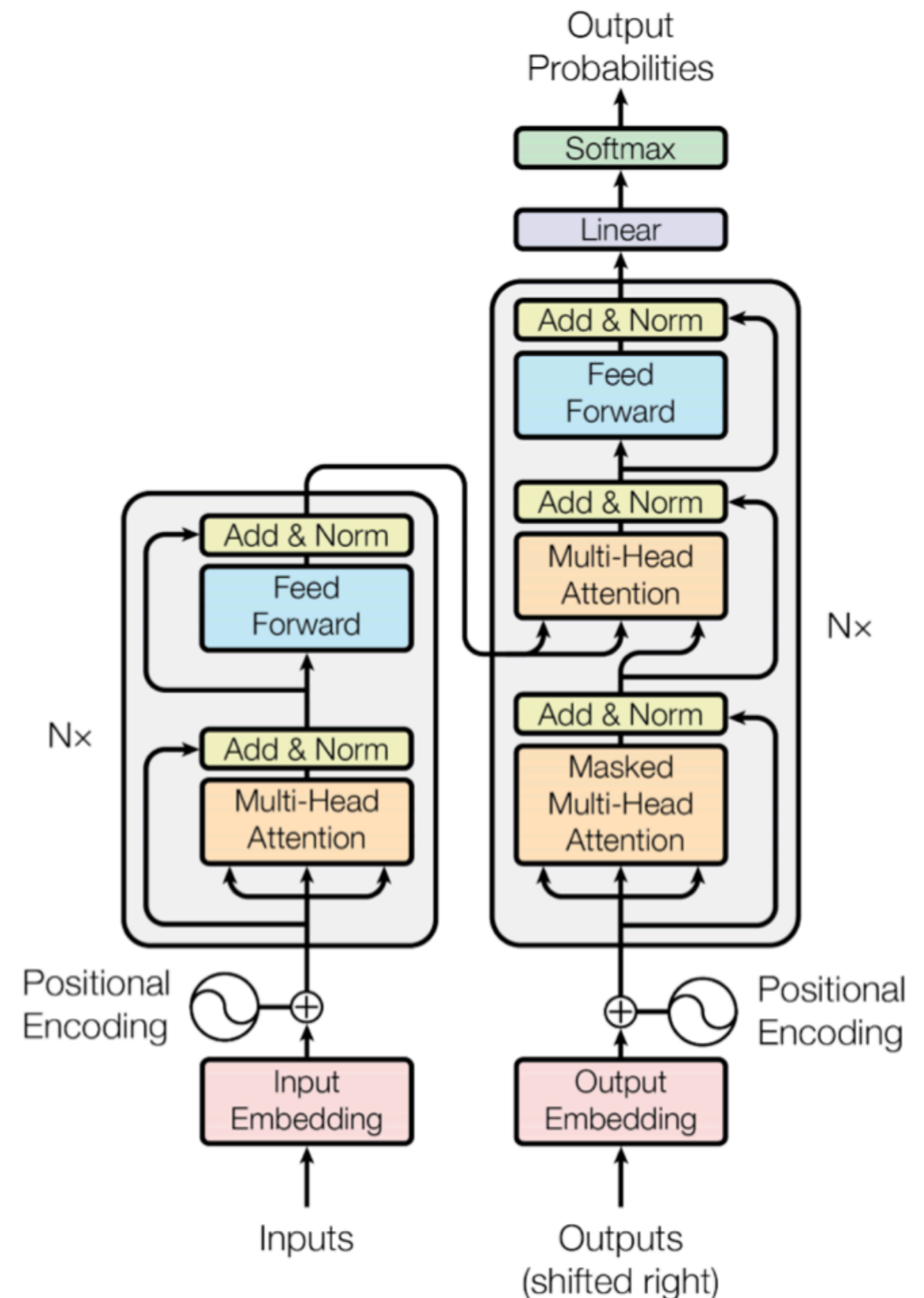
$$attention(q, \mathbf{k}, \mathbf{v}) = \sum_i similarity(q, k_i) \times v_i$$

Attention Mechanism

- Neural architecture

“Attention is all you need”

- Vaswani et al. (2017) – Transformer Network
- Encoder-decoder based on attention (no recurrence)



Multihead attention

- **Key idea:** compute multiple attentions per query with different weights
- Schematic

$$\text{multihead}(Q, K, V) = W^0 \text{contact}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)$$

$$\text{head}_i = \text{attention}(W_i^Q Q, W_i^K K, W_i^V V)$$

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

Masked Multi-head attention

- **Key idea:** multi-head where some values are masked (i.e., probabilities of masked values are nullified to prevent them from being selected)
- When decoding, an output value should only depend on previous outputs (not future outputs). Hence we mask future outputs

$$attention(Q, K, V) = softmax \left(\frac{q^T K}{\sqrt{d_k}} \right) V$$

$$MaskedAttention(Q, K, V) = softmax \left(\frac{q^T K + M}{\sqrt{d_k}} \right) V$$

where M is a mask matrix of 0's and $-\infty$'s

Layer normalization and positional embedding

- Layer normalization
 - Normalize values in each layer to have 0 mean and 1 variance
- Positional embedding
 - Embedding to distinguish each position