CAI 4104/6108 – Machine Learning Engineering:

Transformers & Language Models

Prof. Vincent Bindschaedler

Spring 2024

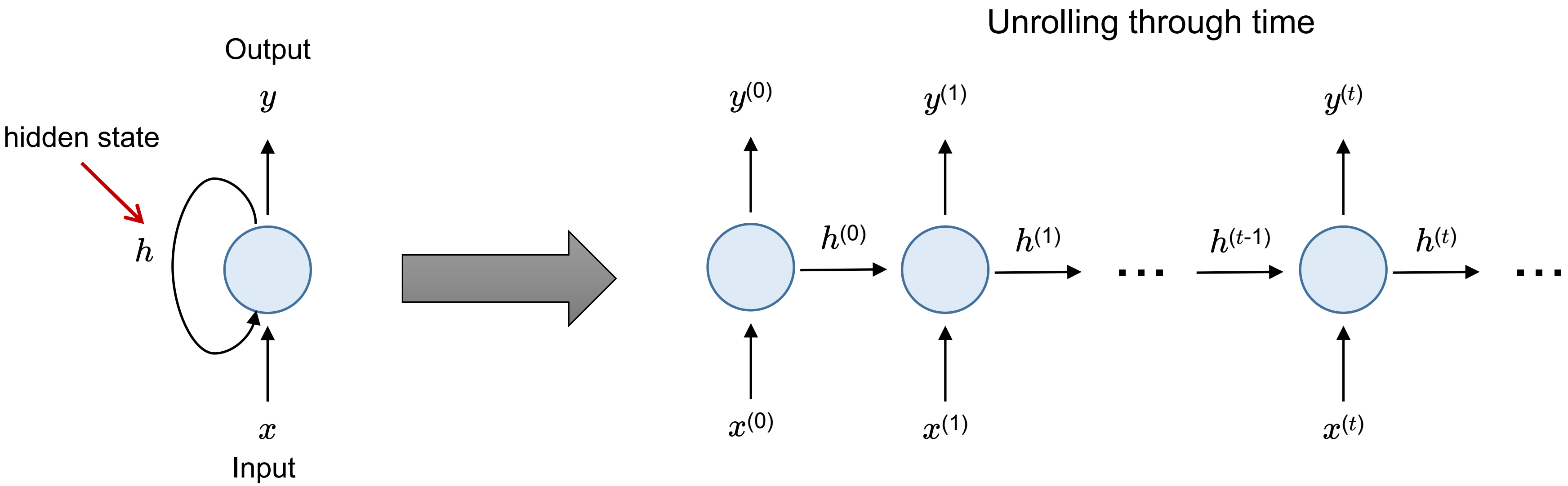
Administrivia

- Project Proposals
  - I have reviewed the proposal (see comments on Canvas)
  - Please go ahead and start working on your project

- Homework 4
  - Topic: (debugging) training neural networks (& some CNNs)
  - Due Wednesday 4/3

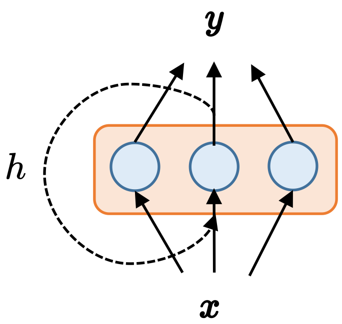
Reminder: Recurrent Neurons/Units

- Recurrent Layers:
  - Made up of recurrent neurons/units which keep state
  - State at time t: $h^{(t)}$ is a function g of the previous state $h^{(t\text{-}1)}$ and the current input $x^{(t)}$
    - $h^{(t)} = g(h^{(t\text{-}1)}, x^{(t)})$



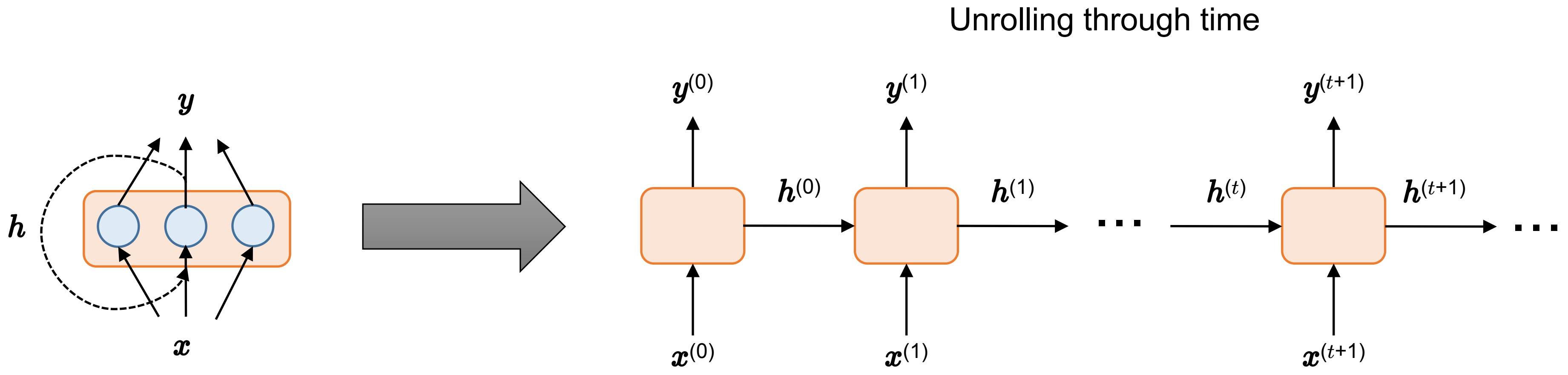
Reminder: Recurrent Layers

- Recurrent Layers:
  - Weight matrices: $\boldsymbol{W}$ (m × k) and $\boldsymbol{V}$ (k × k) Bias vector: $\boldsymbol{b}$ (k × 1)
    - k is the number of units/neurons
  - Activation function: f (e.g., tanh)
  - Hidden state vector: $\boldsymbol{h}^{(t)} = \boldsymbol{V}\ \boldsymbol{y}^{(t\text{-}1)}$ (e.g., we can set $\boldsymbol{h}^{(0)} = 0$)
  - Output vector: $\boldsymbol{y}^{(t)} = f\,(\boldsymbol{W}^T \boldsymbol{x}^{(t)} + \boldsymbol{h}^{(t)} + \boldsymbol{b})$



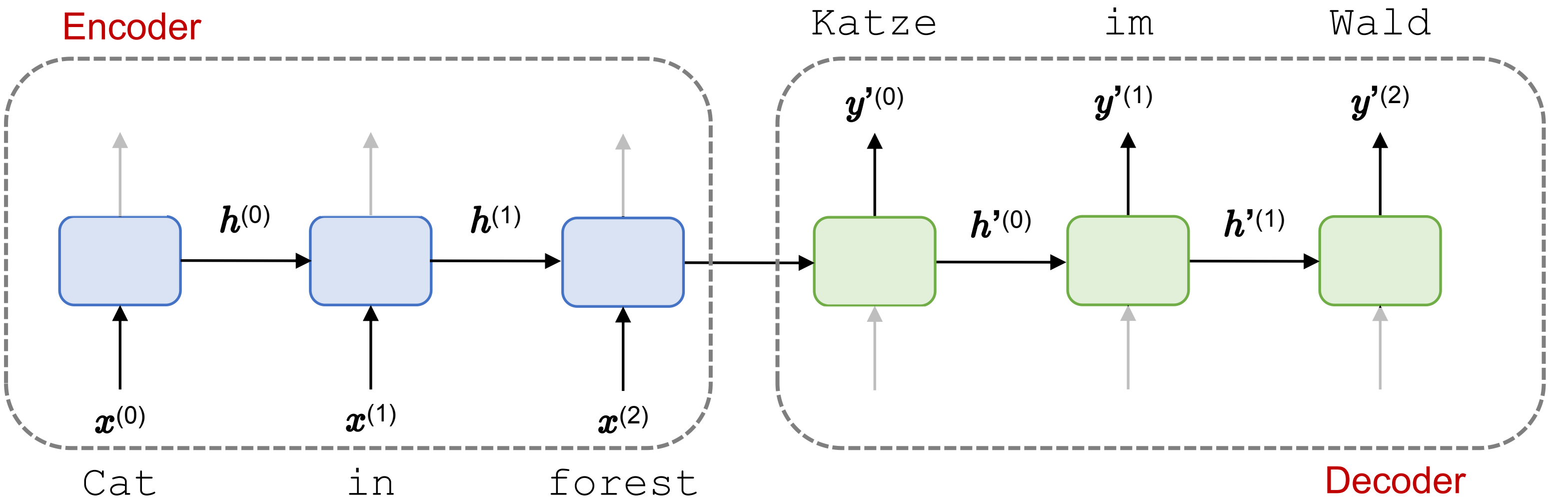
Reminder: Processing Sequences

- Architecture & Tasks:
  - Sequence-to-sequence: from an input sequence produce a sequence as output
  - Vector-to-sequence: from a fixed length input produce a sequence as output
  - Sequence-to-vector: from an input sequence produce a fixed length output
  - Encoder-decoder networks: sequence-to-vector followed by vector-to-sequence



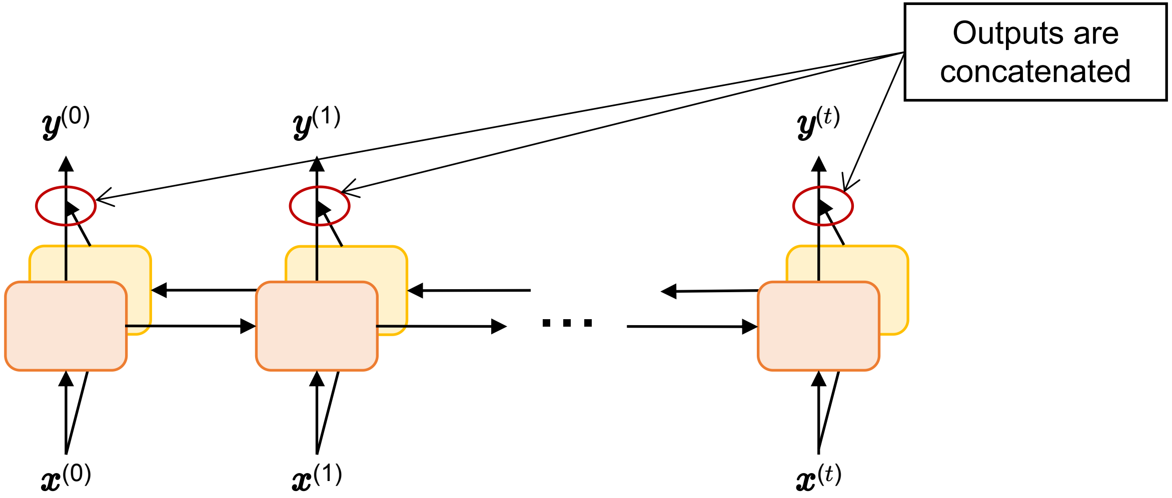
Reminder: Processing Sequences

- Architecture & Tasks:
  - Encoder-decoder networks: sequence-to-vector followed by vector-to-sequence
  - Example: Language translation
    - Translate a sentence from one language to another

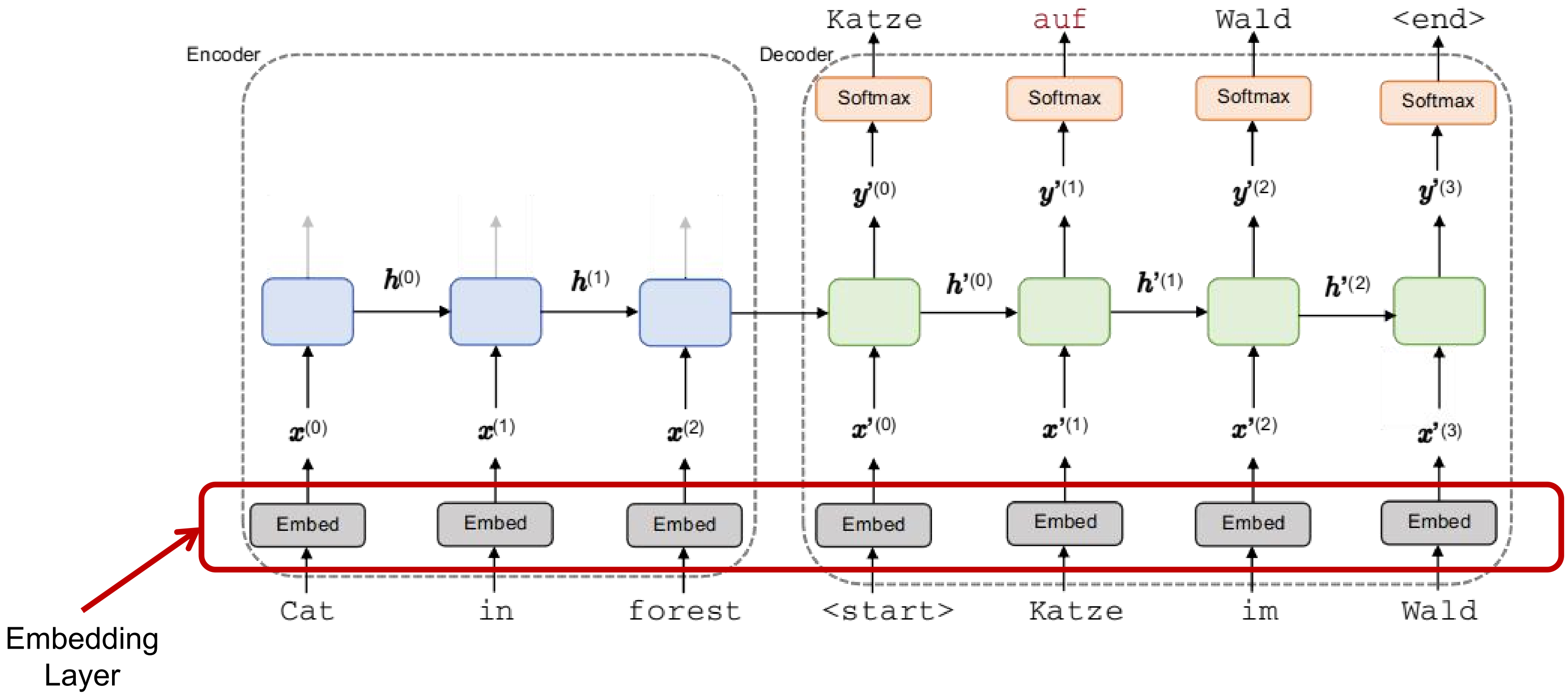


Reminder: Deep RNNs & Bidirectionality

- Deep RNNs:
  - RNNs with multiple recurrent layers
- Bi-directional RNNs:
  - Takes in sequence forwards and also backwards

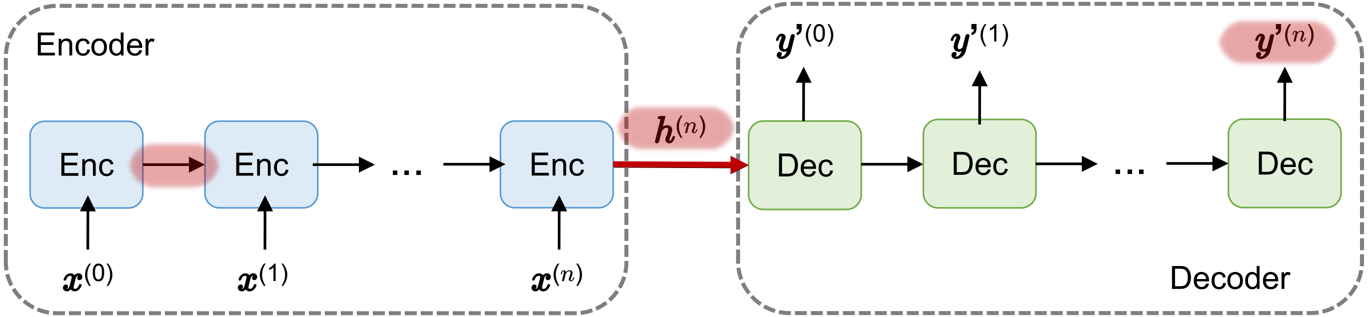


Reminder: Training Encoder-Decoders



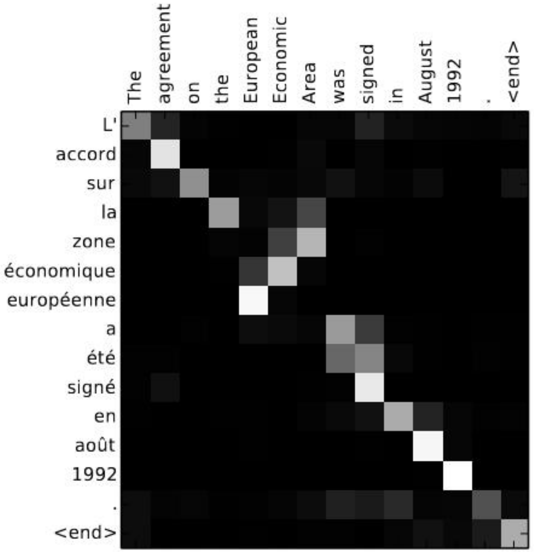
Reminder: Long Paths & Attention

- *What if the decoder could **focus** on the **most important words** at each time step?*
  - Seminal paper introducing attention (called Bahdanau attention)
    - Bahdanau, Dzmitry, Kyung Hyun Cho, and Yoshua Bengio. "*Neural machine translation by jointly learning to align and translate*." In 3rd International Conference on Learning Representations, ICLR 2015.
  - Note: there are multiple types of attention (e.g., Dot-product attention, multiplicative attention, self-attention, visual attention etc.)

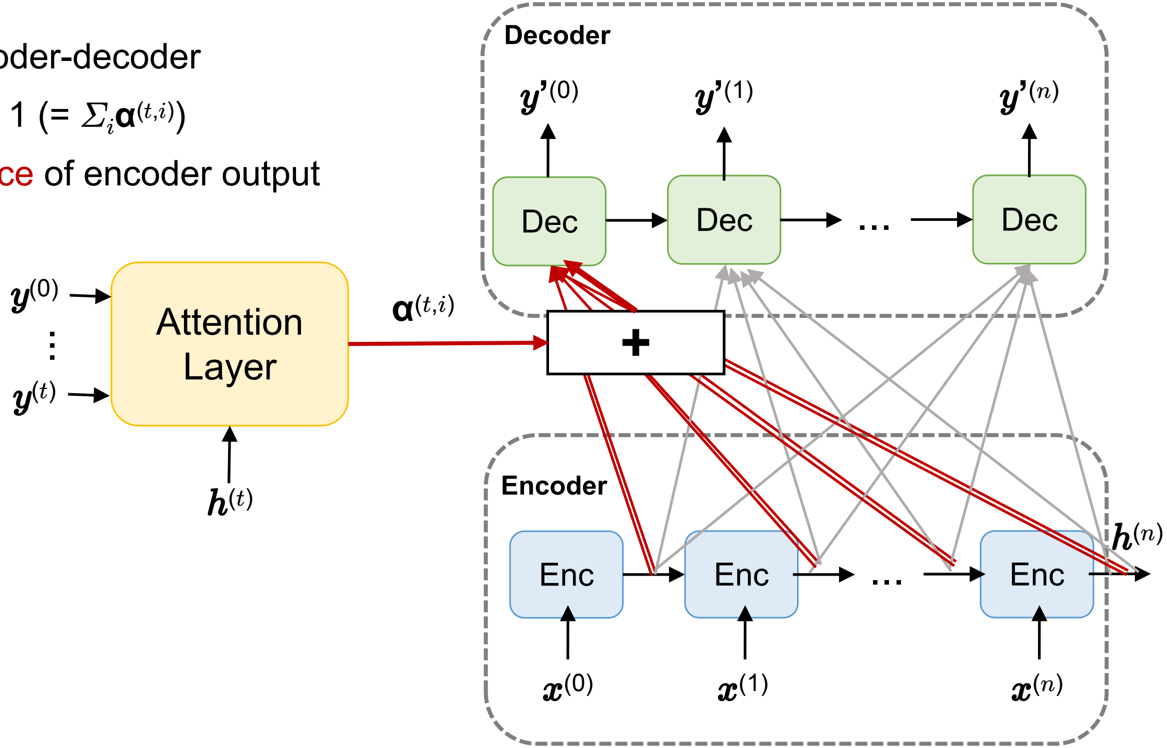


Reminder: Attention

- **Attention** "network" (or layer)
  - Trained at the same time at the encoder-decoder
  - Produces weights $\boldsymbol{\alpha}^{(t,i)}$ which sum to 1 (= $\Sigma_i \boldsymbol{\alpha}^{(t,i)}$)
  - Weight $\boldsymbol{\alpha}^{(t,i)}$ represents the importance of encoder output



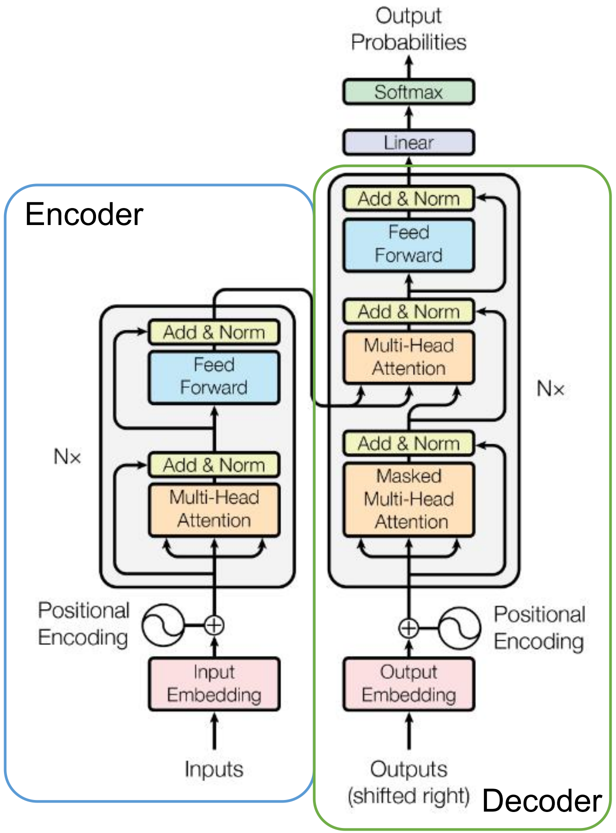
Source: Bahdanau et al. ICLR 2015.



Transformer

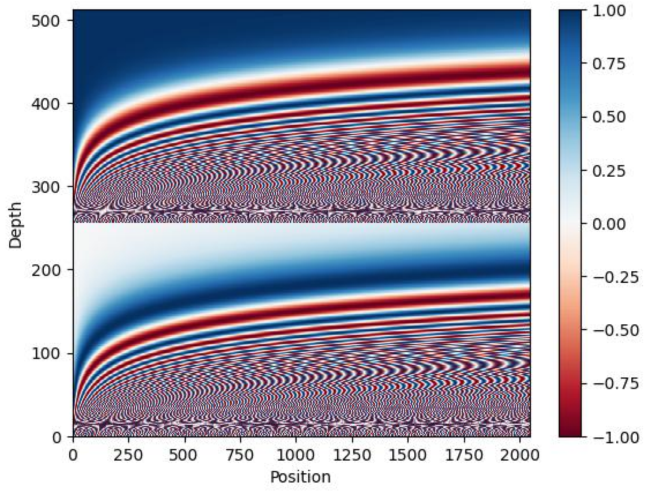
- Seminal paper:
  - Vaswani et al. *"Attention is all you need."* NeurIPS, 2017.
  - Proposes the Transformer architecture
  - Main claim: we don't need recurrent neural networks, we just need attention

- Transformer architecture
  - Encoder - decoder (**no recurrence**)
  - Key ideas:
    - Positional Encoding
    - Scaled-dot product attention with multiple heads
    - Cross-attention between encoder and decoder
    - Layer normalization



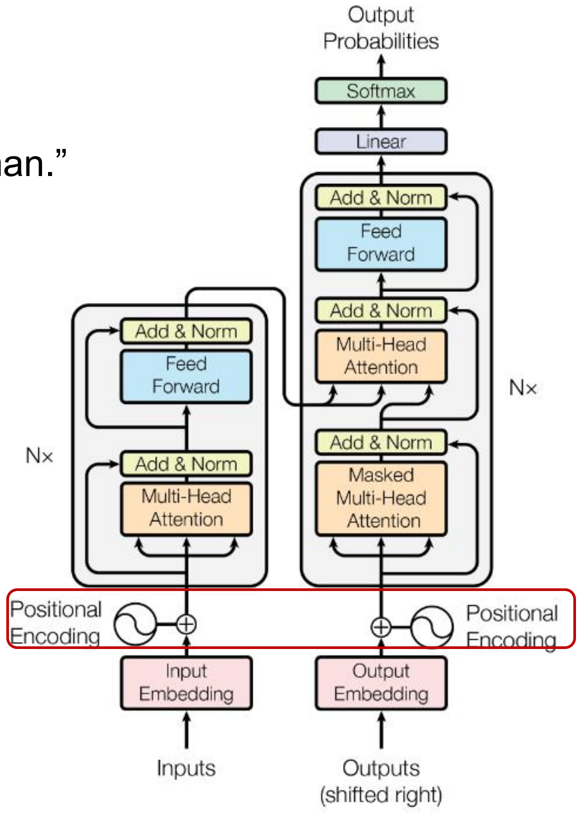
Positional Encoding

- The model is not an RNN
  - So we need a way to **encode** the **position** of tokens/words
  - Model must be able to distinguish “man bites dog” from “dog bites man.”
  - pos is the position in the sequence and i is the depth.



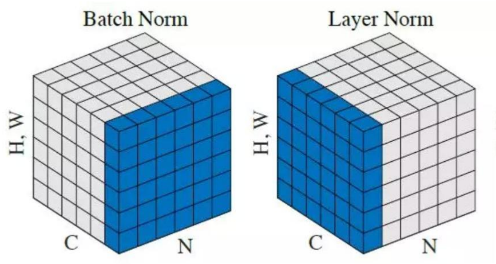
source: tensorflow transformer tutorial

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

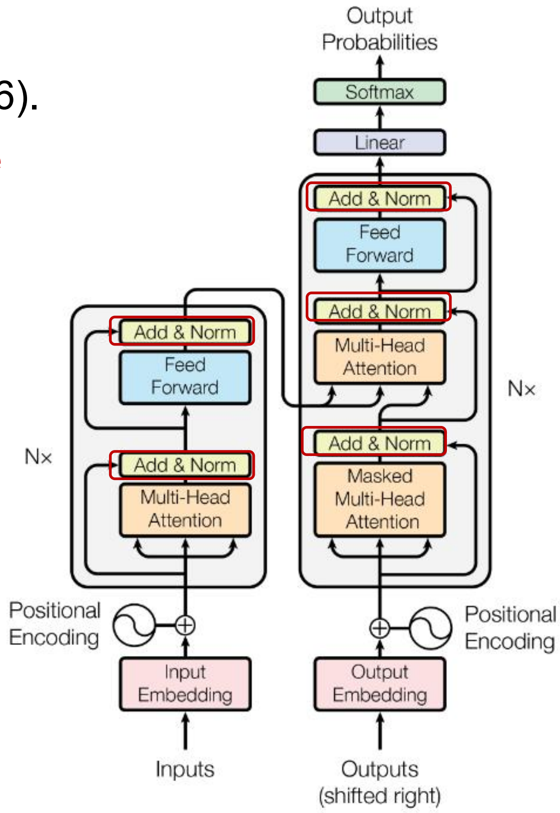


Layer Normalization

- A way to normalize a mini-batch
  - Proposed in Ba et al. “Layer Normalization.” stat, 1050, 21 (2016).
  - Compute the mean μ and standard deviation σ of each example over the hidden units (in each layer)
    - $x_{\text{normalized}} = \gamma\,(x - \mu) / (\sigma^2 + \varepsilon)^{1/2} + \beta$,
    - where γ, β are learnable parameters

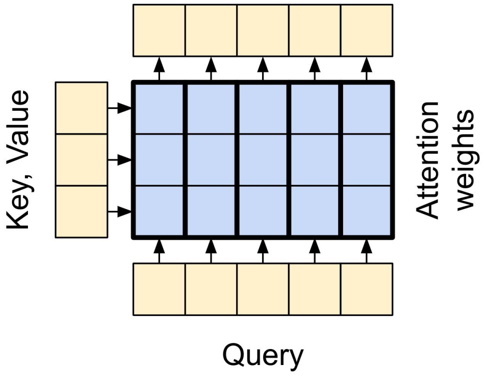


source: *https://paperswithcode.com/method/layer-normalization*

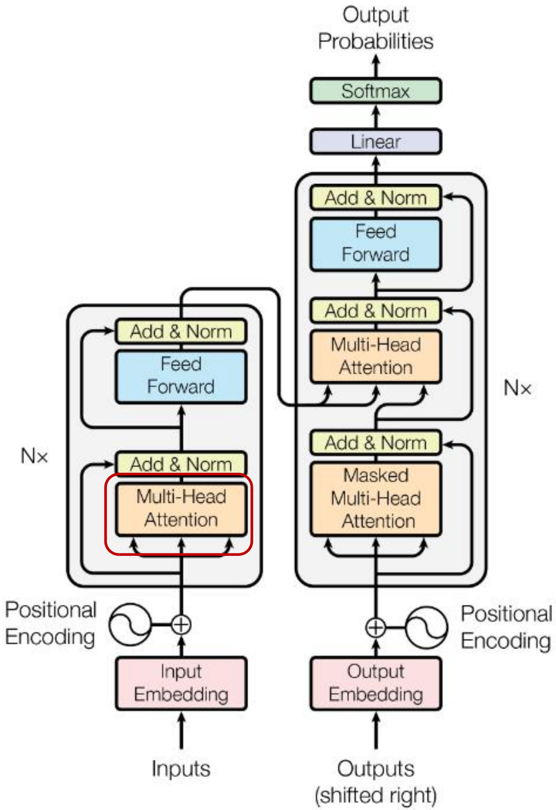


Attention Layers

- Query, Key, Value Attention Layer
  - Think of it as “fuzzy” dictionary lookup using vectors
  - query: the sequence being processed
  - key, value: sequence attended to

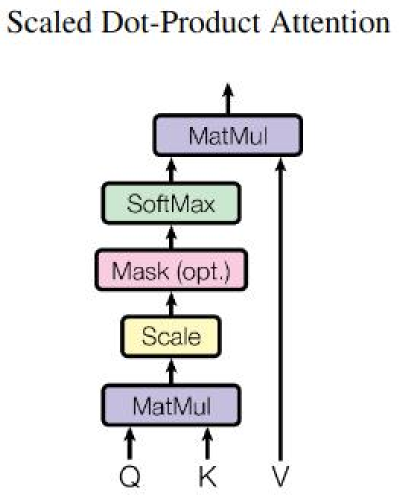


source: tensorflow transformer tutorial



Scaled Dot-Product Attention

- Inputs:
  - Queries and keys of dimension d_k
  - Values of dimension d_v



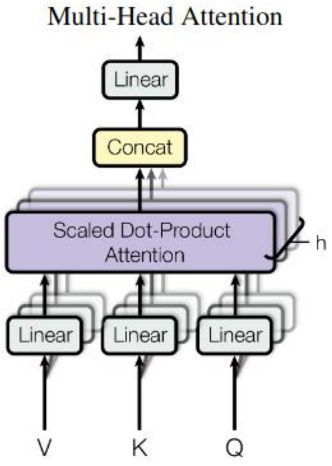
source: Vaswani *et al.*

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

- Here Q, K, V are matrices:
  - We compute attention on all queries at the same time (all packed into matrix Q)
  - Q and K have shape (seq, d_k)
  - V has shape (seq, d_v)

Multi-Head Attention

- Multi-head attention = multiple attention layers in parallel
  - The original transformer paper suggests using 8 heads (h = 8)
    - Note: BERT has 12 heads, GPT-3 has 96 heads



source: Vaswani *et al.*

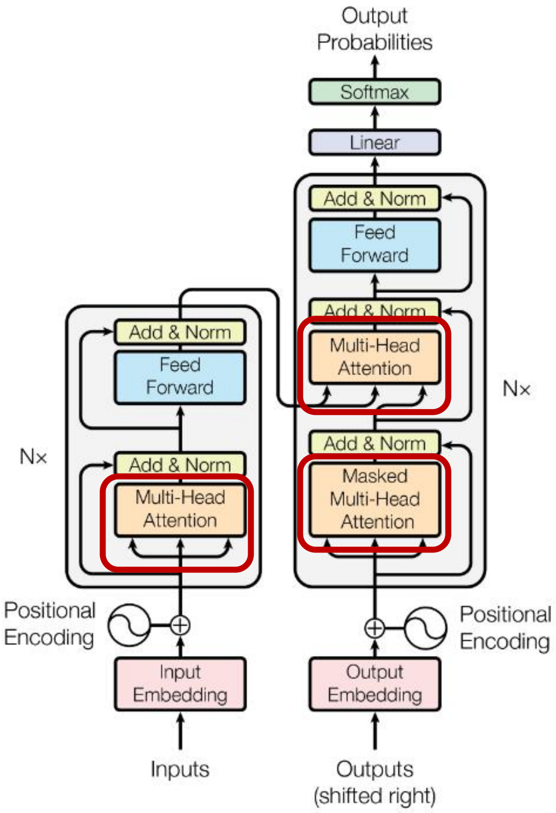
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

- Parameter matrices:
  - W_i^Q, W_i^K, W_i^V, W^O
  - Learned during training

Attention in the Transformer

- Different kinds of attention

- In the encoder:
  - Global self-attention layer
- In the decoder:
  - Cross-attention layer
    - Query comes from decoder
    - Key, value come from encoder
  - Causal attention layer
    - A mask is applied to ensure the model is causal
    - The mask ensures model does not need see the “future”
    - This makes the model autoregressive (it only depends on previous elements of the sequence)



Transformer Architecture

- Transformer Architecture
  - Advantages
    - Faster training than RNNs and parallelizable
    - Much lower maximum path length
      - Attention helps overcome long-range dependencies
  - Disadvantage
    - Self-attention is inherently $O(n^2)$

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

source: Vaswani *et al.*

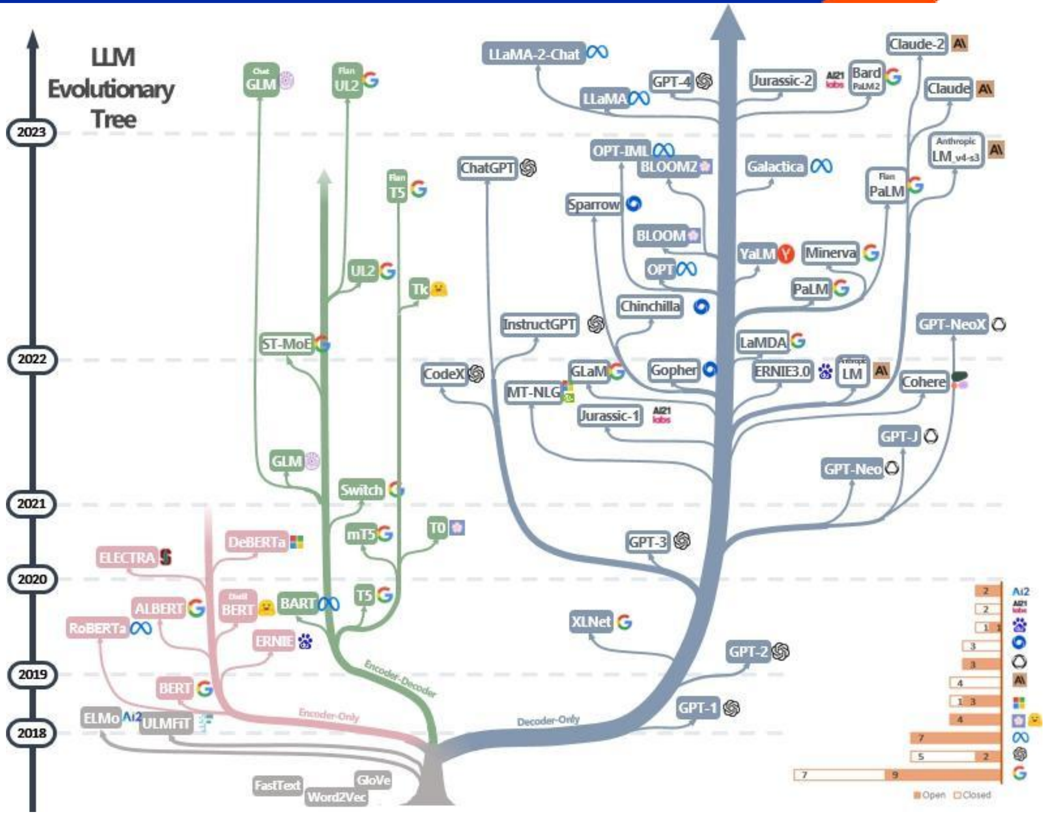
Transformer Architecture: Notes

- Transformer ≠ LLM
  - The Transformer architecture is a general architecture
  - It can be used for tasks other than natural language
  - For example:
    - To process images (ViT - Vision Transformer)
    - To deal with graphs (Graph Transformer)

- LLM using a Transformer
  - Input text (a string) is tokenized into tokens from vocabulary using a tokenizer
  - The vocabulary is learned from data through Byte Pair Encoding (BPE)
    - Common words **and subword units** become tokens
    - Some tokens consist of multiple words and many words consist of multiple tokens
    - For example (GPT-2 vocab): `'learning'` and `' learning'` are tokens.

Transformer & LLM Evolutionary Tree

- Types of Transformers:
  - Decoder-only causal models (e.g., GPTs)
  - Encoder-only models (e.g., BERT)
  - Encoder + decoder



source: Yang et al. “Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond.” arXiv, 2023.

Large Language Models (LLMs)

- What is an LLM?
  - A distribution $p(\cdot|x)$ over token sequences conditional on some input prompt x (a string)
  - We can use it to sample text by repeatedly sampling from p
    - Given initial prompt x, we sample $y_1 \sim p(\cdot|x)$, $y_2 \sim p(\cdot|xy_1)$, $y_3 \sim p(\cdot|xy_1y_2)$, ... $y_n \sim p(\cdot|xy_1y_2...y_{n\text{-}1})$
    - Output: $y = y_1y_2y_3...y_n$
- How are LLMs trained?
  - Step 1: Pretraining — LLM trained using a ***large*** corpus of text documents
    - Task: **predict the next token** given a prefix
  - Step 2: Supervised fine-tuning — model is trained using pairs of prompts and responses
  - Step 3: Reinforcement Learning with Human Feedback (RLHF)
    - i) A set of (human) labelers rate answers from the model
    - ii) These ratings are used to train a **reward model**
    - iii) The reward model is used to further optimize the LLM using reinforcement learning

Can We Trust LLMs?

- Large Language Models (LLMs) are known to produce ***hallucinations***
  - Why does this happen? **How are LLMs trained?**
    - LLMs are trained to ***predict the next (or nearby) token/words***; not to produce factual information!
    - **Note:** systems like ChatGPT are also fine-tuned using Reinforcement Learning with Human Feedback (RLHF)
  - Types of Transformers/LLMs:
    - Decoder-only causal models (e.g., GPTs) are inherently left-to-right -> trained to **predict the next token**
    - Encoder-only models (e.g., BERT) usually trained to ***predict randomly masked tokens***
    - Encoder + decoder
- Interesting references:
  - Z. Ji et al. "*Survey of hallucination in natural language generation.*" ACM Computing Surveys 55, no. 12 (2023): 1-38.
  - N. Dziri et al. "*On the origin of hallucinations in conversational models: Is it the datasets or the models?*." arXiv preprint arXiv:2204.07931 (2022).

Examples of Hallucinations

Table 1. Examples of Each Category of Hallucinations for Each Task

Task	Sub-Task	Type	Source	Output
Abstractive Summarization [99]		Intrinsic	The first vaccine for Ebola was approved by the FDA in 2019 in the US, five years after the initial outbreak in 2014. To produce the vaccine, scientists had to sequence the DNA of Ebola, then identify possible vaccines, and finally show successful clinical trials. Scientists say a vaccine for COVID-19 is unlikely to be ready this year, although clinical trials have already started.	The first Ebola vaccine was approved in **2021**.
		Extrinsic		**China** has already started clinical trials of the COVID-19 vaccine.
Dialogue	Task-Oriented [82]	Intrinsic	inform (NAME = pickwick hotel, PRICERANGE = moderate)	the hotel named pickwick hotel is in a **high** price range.
		Extrinsic		the pickwick hotel **in san diego** is a moderate price range
	Open-Domain [116]	Intrinsic	Dialog History: Speaker 1: What do you think about Murray? Speaker 2: I think Murray is a great player he just needs to stay healthy in order to compete more. Who do you like best? Knowledge: In tennis, the term Big Four refers to the quartet of men's singles players comprising Roger Federer, Rafael Nadal, Novak Djokovic, and Andy Murray.	Speaker 1: I like **Roger Nadal** the most!
		Extrinsic		Speaker 1: I like Djokovic. He has played **in the top ten singles players of the world**.
GQA [96]		Intrinsic	Question: dow jones industrial average please?	Answer: The Dow Jones Industrial Average(DJIA) is an index of **30 major U.S.stock indexes**. The DJIAwas created in 1896 by Charles Dow, a statistician and editor of The Wall Street Journal. The index was first published in the mid-1880s, and **was last updated on May 26,1896.**
		Extrinsic	Question: definition of sadduction	Answer: Sadducees. noun. **The definition of a sadducee is a person who acts in a deceitful or duplicitous manner. An example of a Sadduceee is a politician who acts deceitfully in order to gain political power**.1 a member of a Jewish sect that was active during the Second Temple.
Data2Text [149]		Intrinsic	TEAM / CITY / WIN / LOSS / PTS / FG_PCT / BLK Rockets / Houston / 18 / 5 / 108 / 44 / 7 Nuggets / Denver / 10 / 13 / 96 / 38 / 7	The Houston Rockets **(18-4)** defeated the Denver Nuggets (10-13) 108-96 on Saturday.
		Extrinsic		**Houston has won two straight games and six of their last seven.**
Translation [168]		Intrinsic	迈克周四去书店。(Michael went to the bookstore on Thursday.)	**Jerry didn't go** to the bookstore.
		Extrinsic	迈克周四去书店。(Michael went to the bookstore on Thursday.)	Michael **happily** went to the bookstore **with his friend.**

In the Data2Text task: H/A, H/A, home/away; MIN, minutes; PTS, points; REB, rebounds; AST, assists; BLK, blocks; FG_PCT, field goals percentage.

Source: Z. Ji et al. "Survey of hallucination in natural language generation." ACM Computing Surveys 55, no. 12 (2023): 1-38.

Next Time

- Wednesday (4/3): Guest Lecture (Scott Siegel)
  - Topic: Reinforcement Learning

- Upcoming:
  - Homework 4 due 4/3