Lecture 17: Attention, Transformers, Foundation Models.

CS4787/5777 — Principles of Large-Scale ML Systems

Sequence networks. How to process examples where the input is of varying size? Or what if the output needs to be of varying size? Very common in NLP.

How could we handle inputs and outputs which might be of varying lengths?

Review: Recurrent neural networks. Consume a sequence one-token-at-a-time, like a DFA.

Multiple ways to produce output:

- single output at the end of the sequence
- one output per sequence element
- output that might vary in length

Popular kind of RNN is the LSTM (long short term memory network), which has extra features that discourage ”forgetting” of information across the sequence.

Spill-over from last time pushed to the end of the lecture.

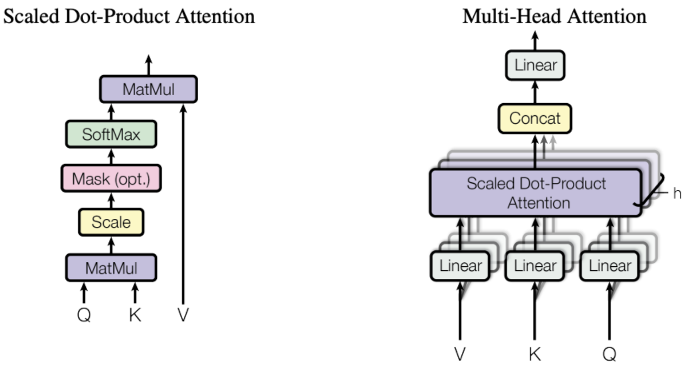
Attention/Transformers. Transformers give an alternate way of approaching sequence models with deep learning. Main ideas:

- positional encoding for each sequence element (called a “token”)
- process tokens mostly in parallel
- only layer which mixes information across tokens is the *self-attention layer*

For queries and keys in dimension d_k and values in dimension d_v, if n is the sequence length, then for $Q \in \mathbb{R}^{n \times d_k}$, $K \in \mathbb{R}^{n \times d_k}$, and $V \in \mathbb{R}^{n \times d_v}$,

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

Typical to use *multi-head attention*, which does this in parallel across multiple *heads* each of which has their own attention matrix. Then the outputs are concatenated.

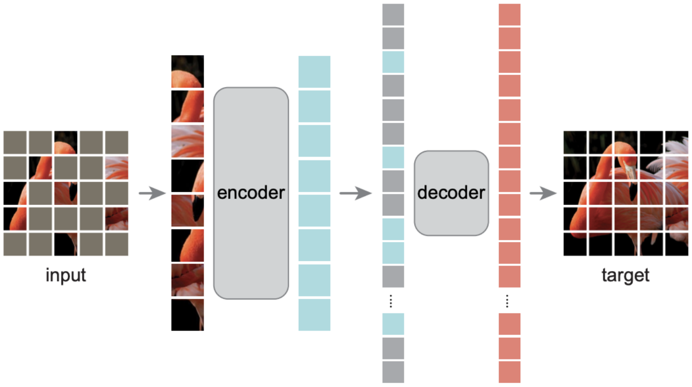


An illustrative figure from “Attention Is All You Need” by Vashwani et al, NeurIPS 2017.

Can even use it for vision! ViT. How?

Self-supervised learning. Extract a supervision signal from the data itself, usually leveraging domain-specific structure we know the data has.

Example: fill-in-the-blanks for computer vision. Take a image. Remove patches from the image. Train a DNN to recover the original image from the version with the patches removed. Then use part of this network as an initialization for a downstream supervised task.



An illustrative figure from “Masked Autoencoders Are Scalable Vision Learners” by He et al, CVPR 2022.

Transfer learning. Train a model on one task. Apply it to another task.

When might transfer learning be useful? When might it be a bad idea?