Tasks with Long Sequences

Long Text Sequences
* Books, Chatbots, Stories
* GPT3
* Context windows are really long
* context based Q and A needs both a question and relevant text from where it's going to retrieve an answer
* Closed loop Q and A, doesn't need extra text to go along with a question or prompt from a human. All the knowledge is stored in the weights of the model itself during training.

Optional AI Storytelling

Dragon model for Dungeon is based on GPT-3. It generates an interactive story based on all previous turns as inputs. That makes for a task that uses very long sequences. Check it out!

1. https://play.aidungeon.io/main/landing

Jukebox - A neural network that generates music!

https://openai.com/blog/jukebox/

GPT-3 Can also help with auto-programming!

https://beta.openai.com/?app=productivity&example=4_2_0

Transformer Complexity

Transformer Issues

- Attention on sequence of length L takes L^2 time and memory

L=100	L^2 = 10K	(0.001s at 10M ops/s)
L=1000	L^2 = 1M	(0.1s at 10M ops/s)
L=10000	L^2 = 100M	(10s at 10M ops/s)
L=100000	L^2 = 10B	(1000s at 10M ops/s)

- N layers take N times as much memory

  GPT-3 has 96 layers and new models will have more

Attention Complexity

- Attention: softmax(QK^T)V
- Q, K, V are all [L, d_model]
- QK^T is [L, L]
- Save compute by using area of interest for large L

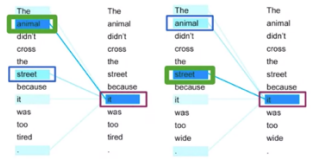
Memory with N Layers

- Activations need to be stored for backprop
- Big models are getting bigger
- Compute vs memory tradeoff

LSH Attention

What does Attention do?

Select Nearest Neighbors (K,Q) and return corresponding V



Nearest Neighbors

Course:

Natural Language Processing with Classification and Vector Spaces

Lessons:

- KNN
- Hash Tables and Hash Functions
- Locality Sensitive Hashing
- Multiple Planes

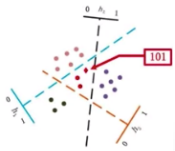
LSH Attention

Standard Attention:

$$A(Q, K, V) = \text{softmax}(QK^T)V$$

LSH Attention:

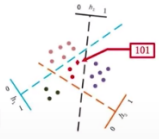
- Hash Q and K
- Standard attention within same-hash bins
- Repeat a few times to increase probability of key in the same bin



Nearest Neighbors

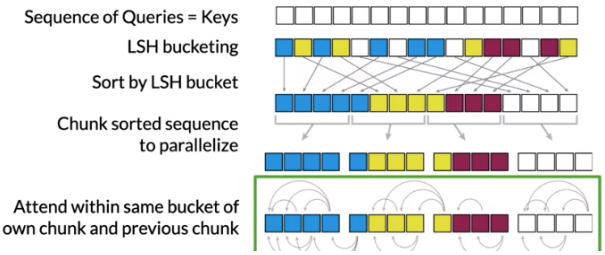
Compute the nearest neighbor to q among vectors $\{k_1, ..., k_n\}$

- Attention computes $d(q, k_i)$ for i from 1 to n which can be slow
- Faster *approximate* uses locality sensitive hashing (LSH)
- Locality sensitive: if q is close to k_i: $\text{hash}(q) == \text{hash}(k_i)$
- Achieve by randomly cutting space $\text{hash}(x) = \text{sign}(xR)$ R: [d, n_hash_bins]



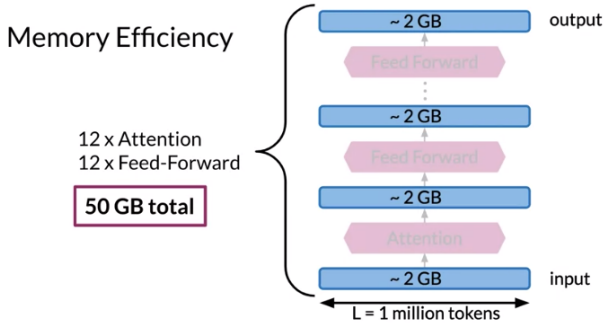
The Efficient Transformer)

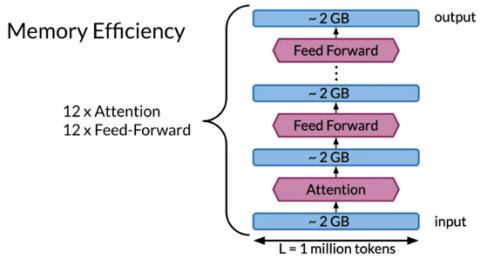
LSH Attention



LSH is a probabilistic not deterministic model, because of the inherent randomness within the LSH algo. Hash can change along with buckets a vector finds itself mapped to.

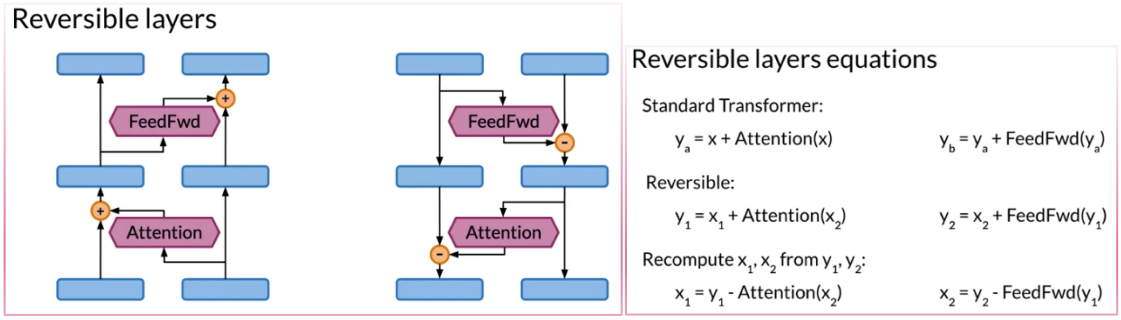
Motivation for Reversible Layers: Memory

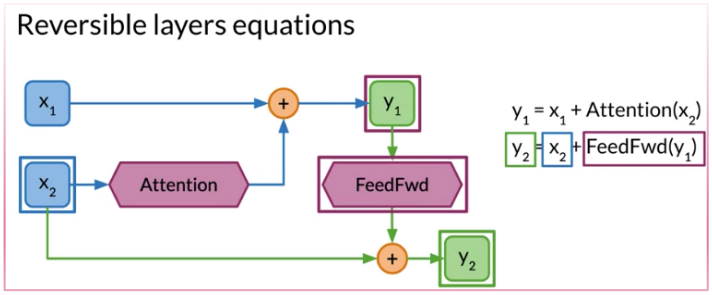


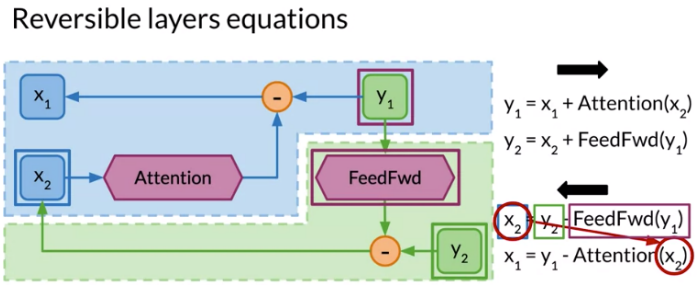


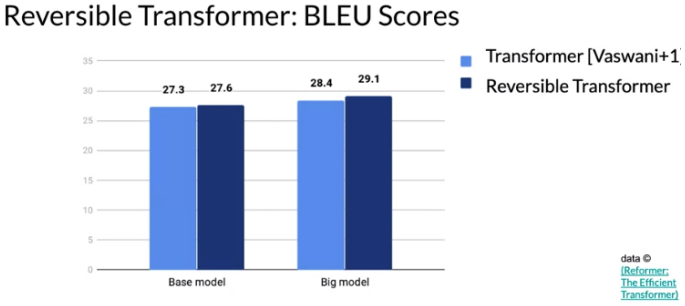
Reversible Residual Layers

The transformer network precedes by repeatedly adding residuals to the hidden states.
To run it in reverse, you can subtract the residuals in the opposite order, starting with the outputs of the model. But in order to save memory otherwise used to store the residuals, you need to be able to recompute them quickly instead, and this is where reversible residual connections come in. The key idea is that you start with two copies of the model inputs, then at each layer you only update one of them. The activations that you don't update will be the ones used to compute the residuals, where this configuration you can now run the network in reverse. Layer 1 is attention and layer 2 is feedforward. The activations in the model are now twice as big, but you don't have to worry about caching for the backwards pass









REFORMER – The Reversible Transfomer

Reformer model is a transformer model designed to be memory efficient so it can run on smaller m/c. MultiWOZ datasets using the Trax framework from the Google Brain Team. MultiWOZ is a very large datasets of human conversations, covering multiple domains and topics.

Reformer

- LSH Attention
- Reversible Layers

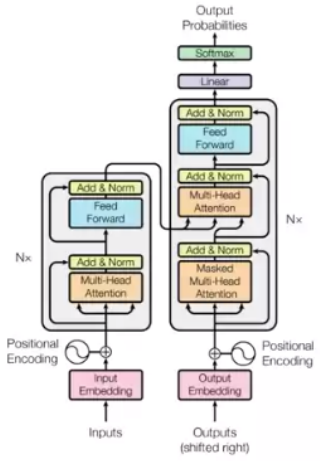


image © [(Attention Is All You Need)]

KNN

https://www.coursera.org/learn/classification-vector-spaces-in-nlp/lecture/d13tm/k-nearest-neighbors

Hash Tables and Hash Functions

https://www.coursera.org/learn/classification-vector-spaces-in-nlp/lecture/OpheJ/hash-tables-and-hash-functions

Locality Sensitive Hashing

https://www.coursera.org/learn/classification-vector-spaces-in-nlp/lecture/HhTQF/locality-sensitive-hashing

Multiple Planes

https://www.coursera.org/learn/classification-vector-spaces-in-nlp/lecture/wdPgw/multiple-planes

Optional AI Storytelling

Dragon model for Dungeon is based on GPT-3. It generates an interactive story based on all previous turns as inputs. That makes for a task that uses very long sequences. Check it out!

2. https://play.aidungeon.io/main/landing

Jukebox - A neural network that generates music!

https://openai.com/blog/jukebox/

GPT-3 Can also help with auto-programming!

https://beta.openai.com/?app=productivity&example=4_2_0