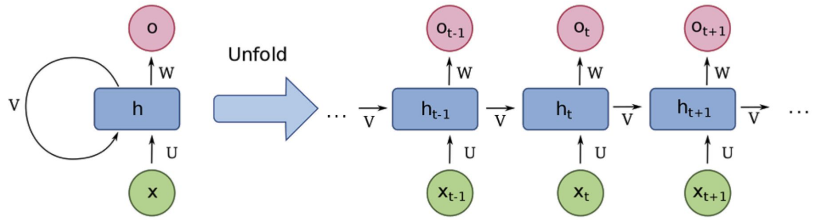
Transformers and BERT

A quick, semi-supervised tour

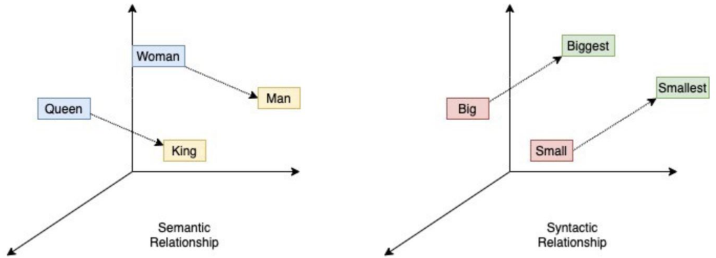
Before Transformers...

Recurrent Neural Networks



- Consider input *in order* → good idea for NLP
- Maintain some memory that gets updated at every time step
- Commonly encoder-decoder architectures

Static Word Embeddings

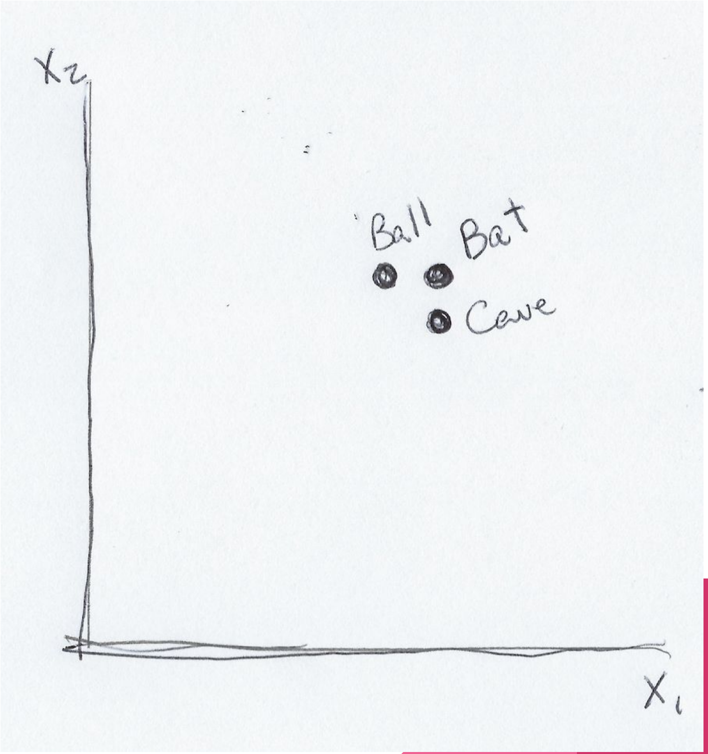


- Produce numerical representations of words that are meaningful
- Can be pre-trained, used for many tasks
- Word2vec, GLoVE

Weaknesses of Static Word Embeddings

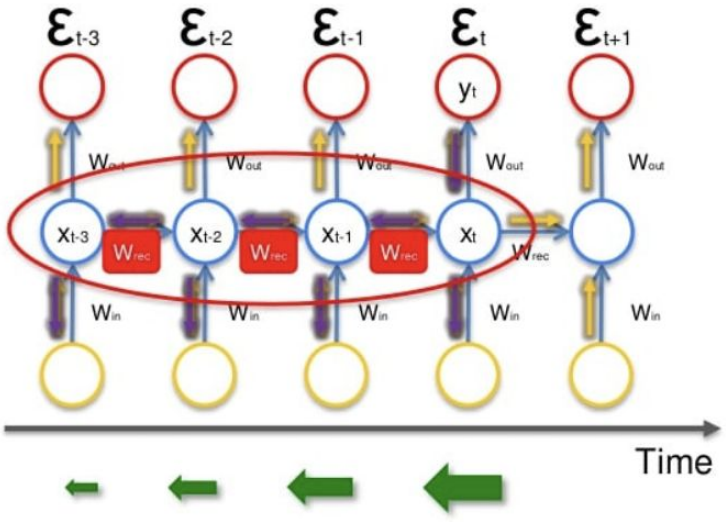
Insurmountable weakness:

- Word meanings are the same in every context
  - We need a bat and ball to play.
  - The bat returned to its cave for the day
  - “Bat” has same representation!!!



Weaknesses of RNN's

- Slow to train → have to perform back-propagation through time
- Memory cell is constantly overwritten, makes it hard to learn long-term dependencies
  - "Sita asked Rukmini to make her some tea because ***she*** was cold."
    - Who does "she" refer to?

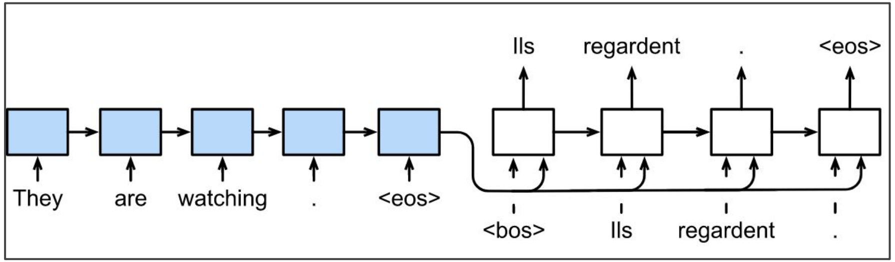


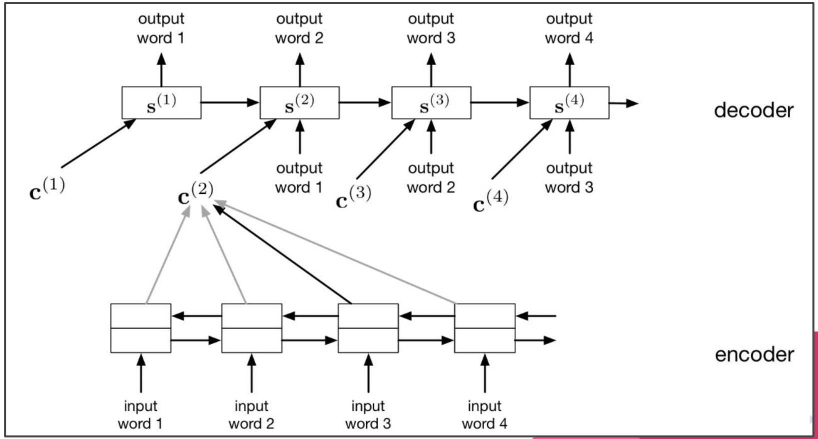
- Vanishing and exploding gradients make it very hard to train RNN's over many time-steps!!!
  - 0.9**50 = 0.005153
  - 1.1**50 = 117.4

Attention

Main Idea: Word should be represented based on context

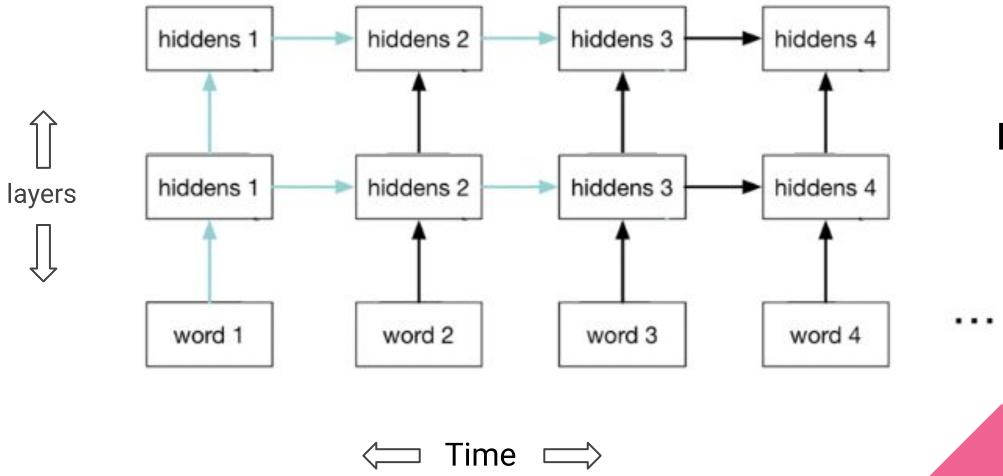
Mechanism: At every position, be able to directly access every position





Is Attention All You Need?

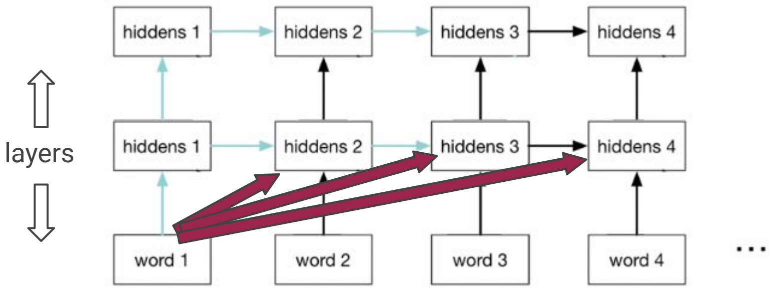
- We would like our model to have access to the other words in a sentence when building the representation of some word
- Previously we achieved this by having the recurrent connections.
- This meant we had to backprop through <u>time</u>
    - Typical sequence length = 128, 512, 1024, etc.



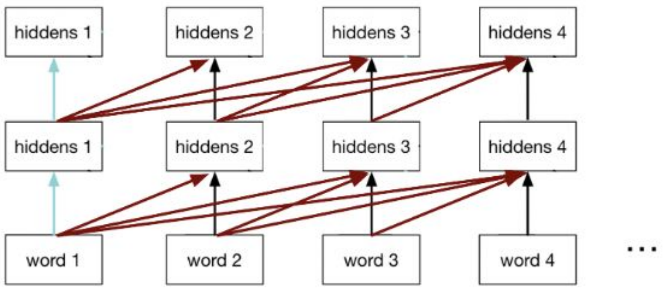
Backprop happens in opposite direction from arrows

Attention Is All You Need

Backprop happens in opposite direction from arrows



RNN

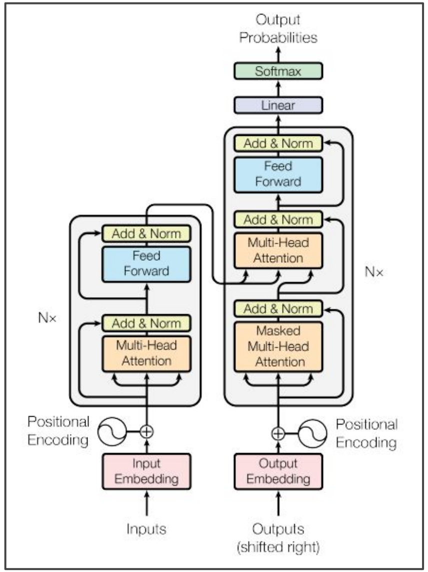


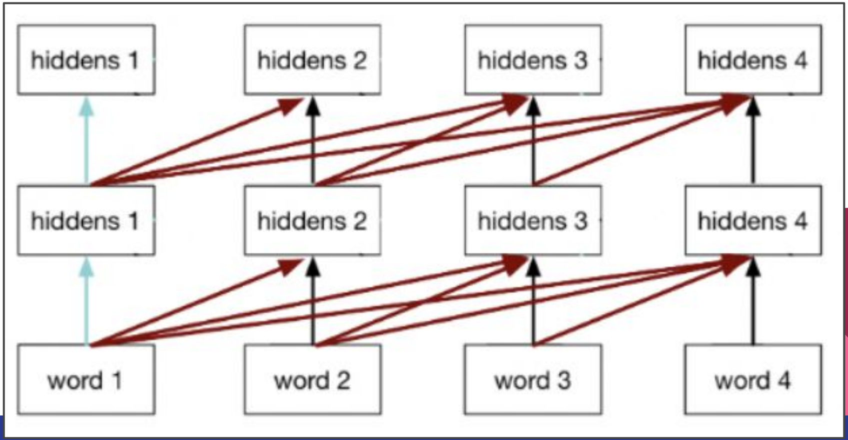
Transformer

- Transformer removes connections within a layer, gets context information from *previous* layer
- Makes training much easier, because we backprop through <u>layers instead of time</u>
  - Typical # of layers: 5-50 (compared to sequence length >= 512)

Transformers

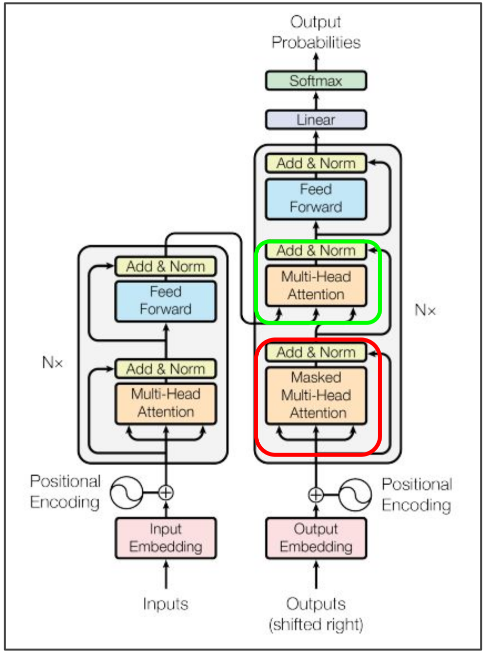
- Original paper: “Attention Is All You Need”
  - Main idea: we can throw out recurrent connections and just use attention
- Overcome RNN Weaknesses:
  - **Train faster:** no more recurrent connections
  - **Train more easily:** skip connections
- Overcome word2vec Weaknesses:
  - Words embeddings are now contextual!!!!





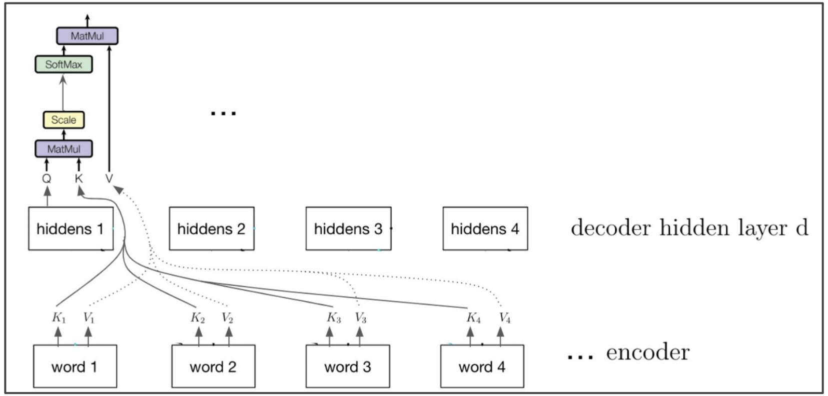
Attention in Transformers

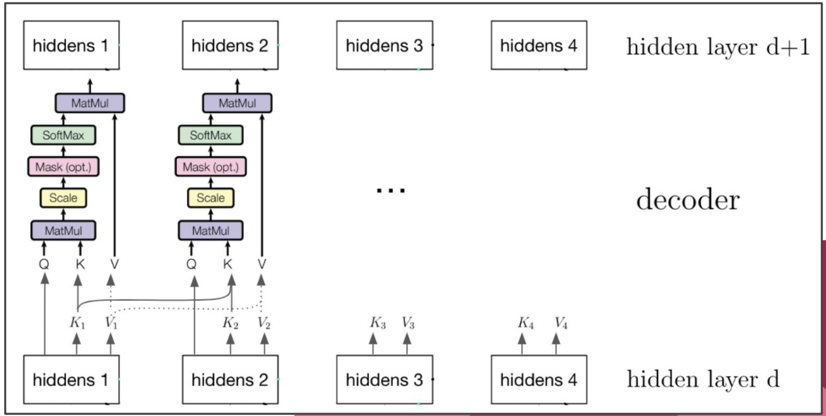
Transformer



Encoder Cross Attention

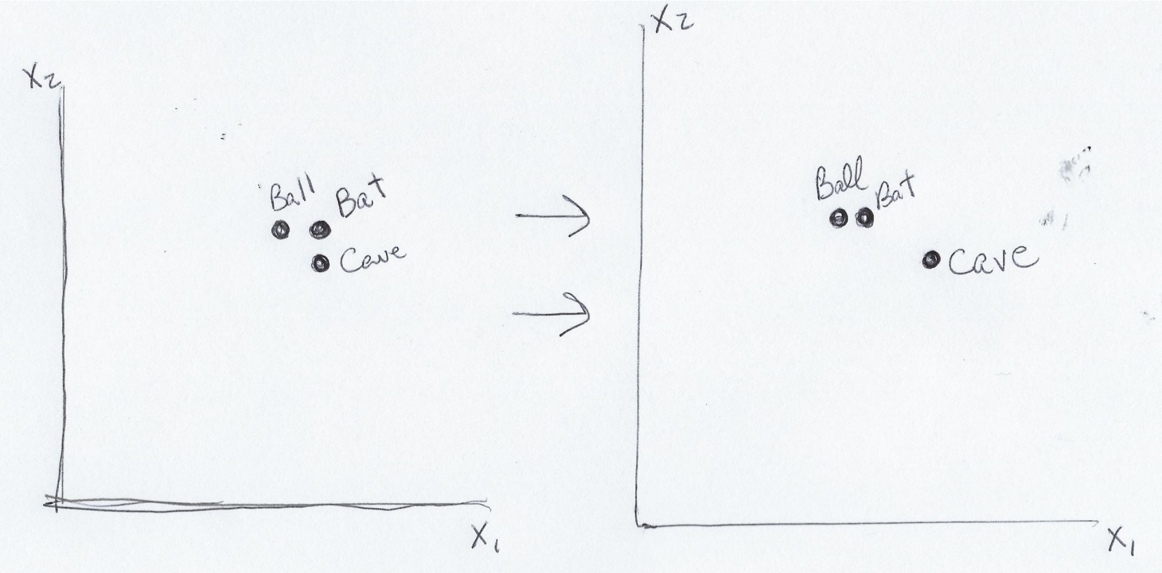
Decoder Self Attention





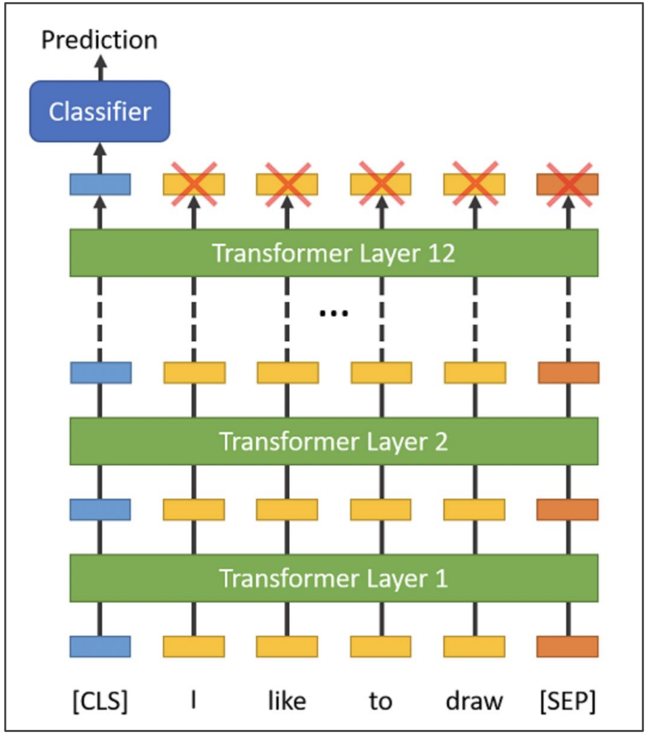
Contextual Word Embeddings

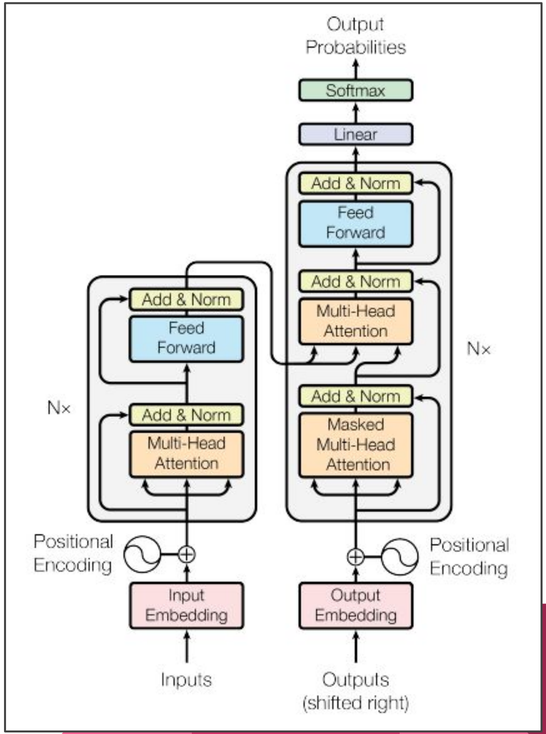
- Transformers build a representation of a word that is sensitive to surrounding context!!!!
- E.g. “You need a bat and ball to play” → meaning of bat should be closer to ball than cave



BERT Is A Transformer Encoder Only!

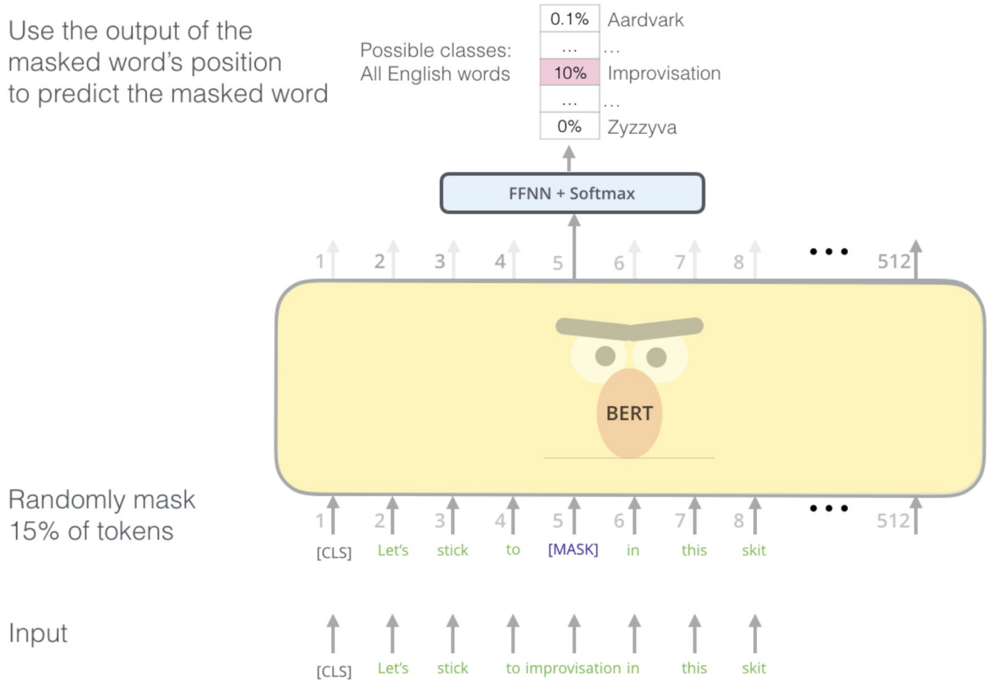
- Original Transformer: Encoder-Decoder (RIGHT)
  - Separate networks for building representations and generating response
- BERT: Encoder only (LEFT)
  - Single network for building representations
  - Bi-directional!





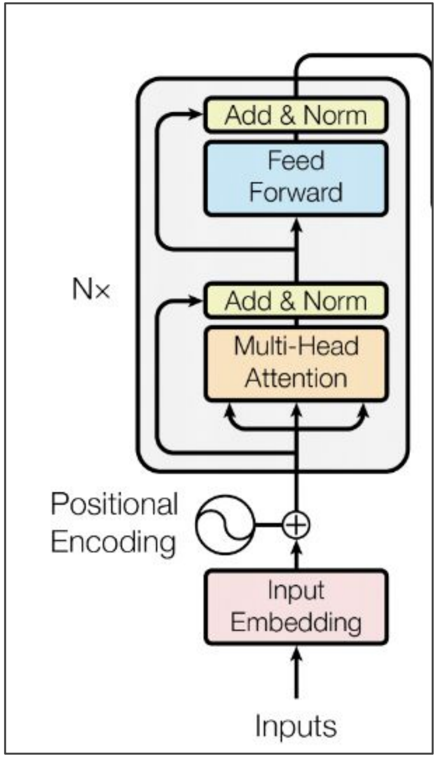
Masked Language Modeling

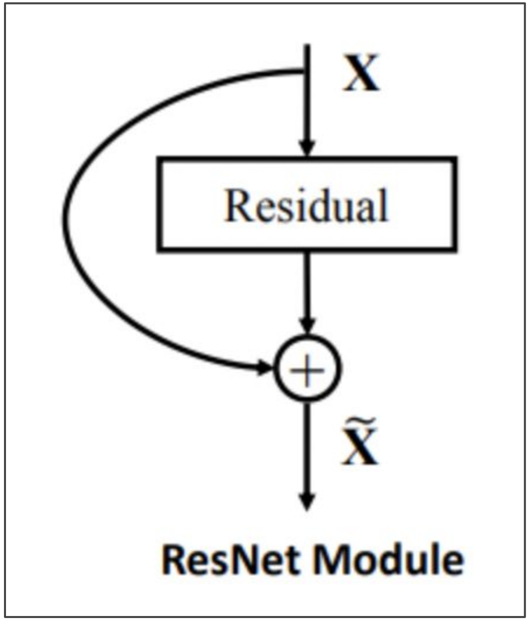
- Original Transformer was trained to predict next word → could not capture left AND right context
- BERT's training task was masked language modeling → predicting the missing word in a sentence



Residual Connections

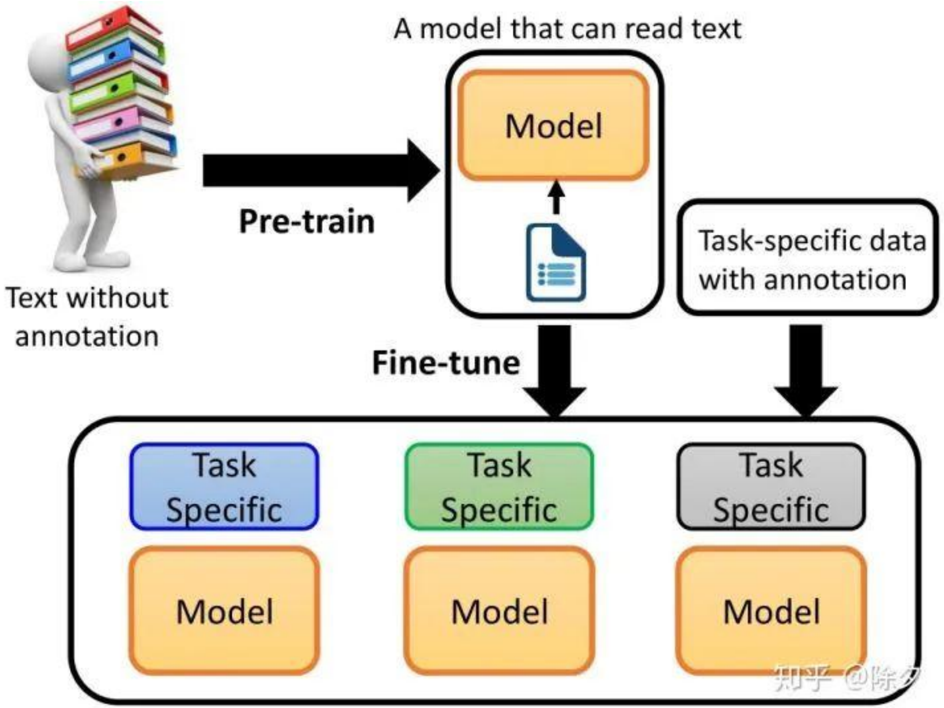
- Residual (or skip) connections are extremely popular in deep neural networks today
- Each layer adds information to input instead of generating from scratch
  - Stabilizes training
  - Allows gradients to flow more easily
- Input size = Output size!
  - Enables modularity and deep stacking





Pre-trained Language Models

- BERT is an example of transfer learning → train model on GIANT dataset, TRANSFER knowledge to many tasks
- Most useful representation of a word might be different based on your task, but pre-trained models are a great starting point for many, many tasks!!!



What Can I Use BERT For??

- Short answer = everything
  - But better at some things than others
- Best at:
  - Question Answering
  - Sentiment Analysis
  - Natural Language Understanding
  - Natural Language Inference
- Also good at:
  - Translation, Generation
- Why? Bi-directional vs. Left-to-right
- But wait for GPT if you want to generate text!
  - BERT specializes in *representing input*
- Where to start?
  - Pretrained models on Huggingface!!!

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Huggingface Demo!

HUGGING FACE

Possible Use Case

- Use semantic similarity and sentiment analysis to predict Apple stock price
- Step 1: Scrape web for each day’s news as well as Apple’s stock price
- Step 2: Use semantic similarity scores between prompts and articles to pick out news relevant to Apple
    - e.g. seed with phrases like “cell phone purchases”, “holiday shopping numbers”, etc.
- Step 3: Compute sentiment score
    - average, median, min, max?
- Create machine learning model to predict stock movement based on recent sentiment about company
    - Previous day, but maybe better as a moving average?

BERT4Rec

- BERT is a good architecture for modeling all kinds of sequences!
    - What will the next frame in a video be based on previous frames?
    - What will the next item purchased by a user be based on previous purchases?
- BERT4Rec → Instead of inputting a sequence of words, input a sequence of items
    - Discover complex patterns in the ordering of how users consume content, purchase goods, etc.
- What other types of sequences could be modelled in this way?

Thank you!

Tom Zollo

tpz2105@columbia.edu