

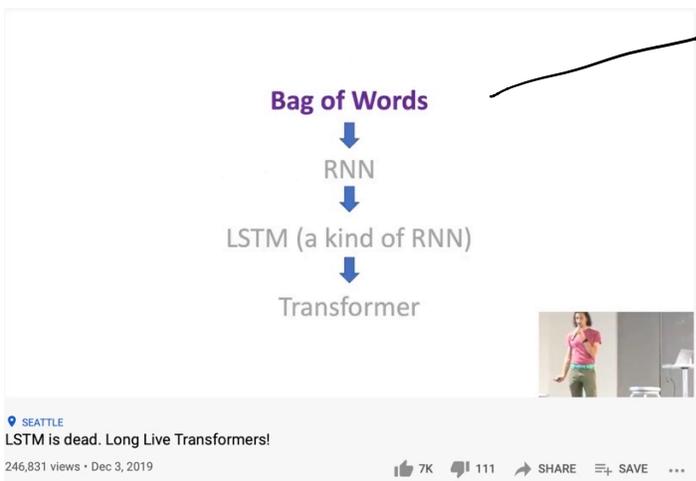
Challenges in working with natural language and sequential data

Challenges - words \rightarrow vectors / numerical values / differentiate
 pay attn to position
 variable length input



- Text generation / story generation
- Translation
- Creating Summaries
- Question Answering
- Text autocompletion
- Irony/sarcasm detection
- Grammar/spelling correction
- Spam detection

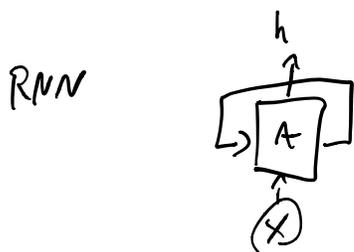
Google: GPT-3, DALL-E



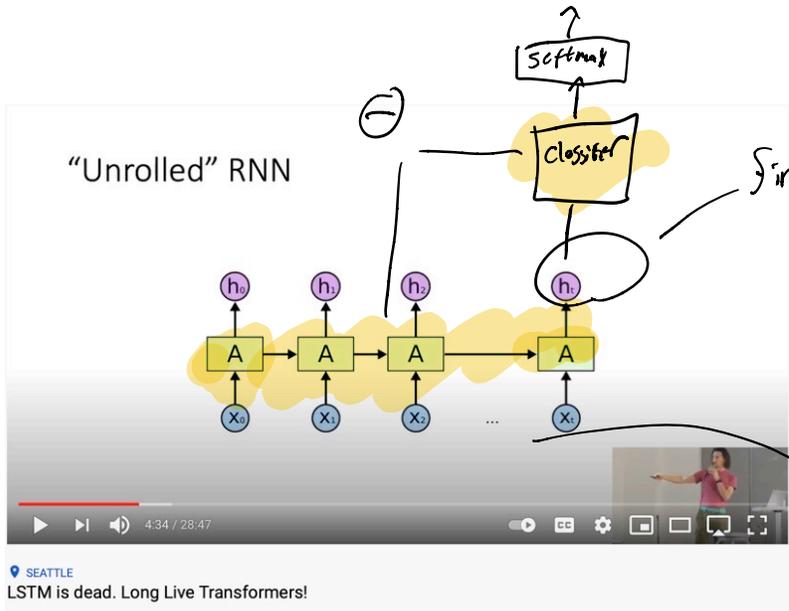
Input^s
 Vector of counts
 of each word in dict

$$\begin{matrix} \text{the} & 1 \\ \text{cat} & 0 \\ \text{happy} & 1 \\ \vdots & \vdots \\ & \text{word} \\ & 1 \end{matrix} \in \mathbb{R}^{10000}$$

“Work to live” vs “live to work”



distrib over classes



Data - {reviews, labels} $\leftarrow y$

Final state $\in \mathbb{R}^d$
 put it into
 a classifier
 (FCN)

$$f_{\theta}(x) = \hat{y}$$

$$\min_{\theta} \sum_i \mathcal{L}(\hat{y}_i, y_i)$$

SEATTLE
 LSTM is dead. Long Live Transformers!

Vanishing & Exploding Gradients

$$H_{i+1} = A(H_i, x_i)$$

$$H_3 = A(A(A(H_0, x_0), x_1), x_2)$$

$$A(H, x) := \mathbf{W}x$$

$$H_N = \mathbf{W}^N x_0 + \mathbf{W}^{N-1} x_1 + \dots$$

*Leaving out the nonlinear activation in A for clarity. The idea holds.

SEATTLE
 LSTM is dead. Long Live Transformers!

\mathbf{W}^N when $N \rightarrow \infty$

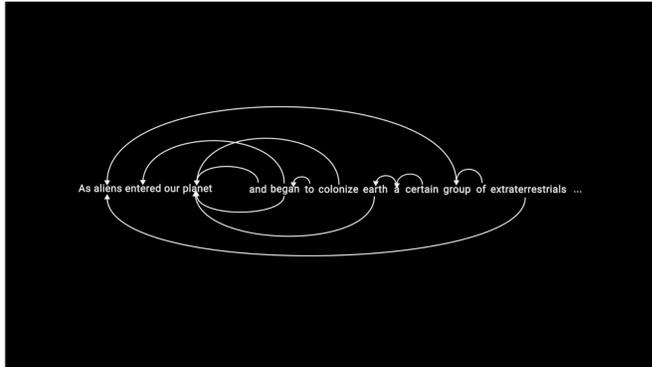
if $w \in \mathbb{R}$,

$w > 1, w^N \rightarrow \infty$ "fast"

$w < 1, w^N \rightarrow 0$

RNNs have a limited window of past times/positions that they can pay attention to

Attention



Illustrated Guide to Transformers Neural Network: A step by step explanation

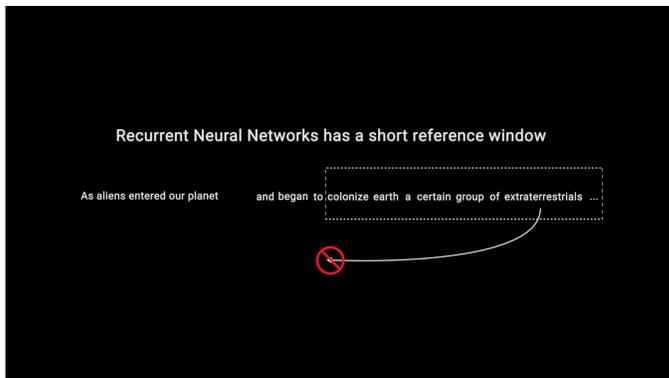
63,343 views · Apr 27, 2020

2.9K 18 SHARE SAVE ...



The A.I. Hacker - Michael Phi
16.2K subscribers

SUBSCRIBE



Illustrated Guide to Transformers Neural Network: A step by step explanation

63,343 views · Apr 27, 2020

2.9K 18 SHARE SAVE ...



The A.I. Hacker - Michael Phi
16.2K subscribers

SUBSCRIBE

Visualizing Attention

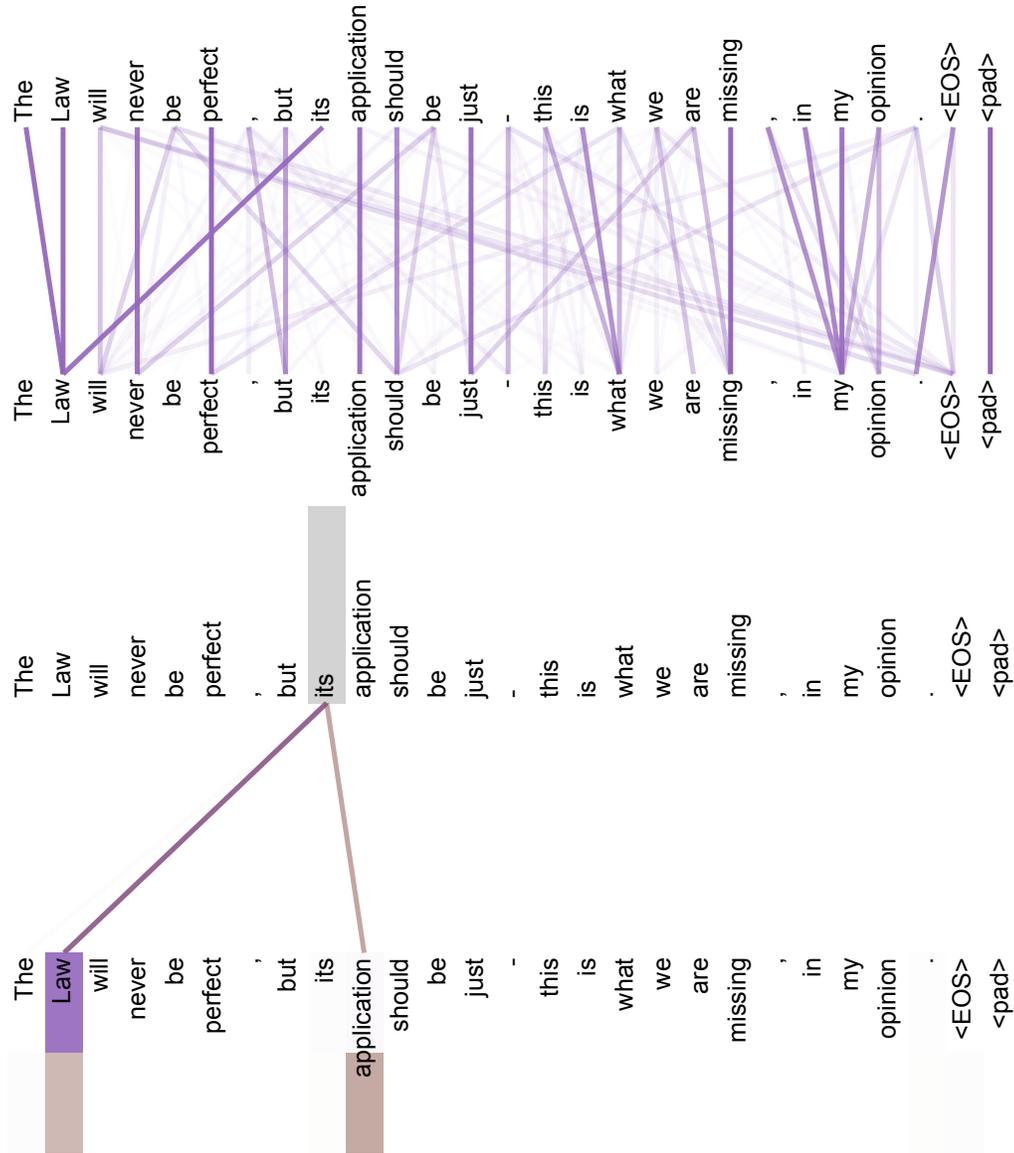


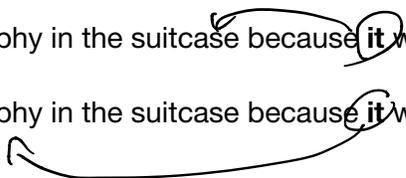
Figure 4: Two attention heads, also in layer 5 of 6, apparently involved in anaphora resolution. Top: Full attentions for head 5. Bottom: Isolated attentions from just the word 'its' for attention heads 5 and 6. Note that the attentions are very sharp for this word.

Example of challenges that attention can help with:

Machine Translation

I didn't put the trophy in the suitcase because **it** was too **small**

I didn't put the trophy in the suitcase because **it** was too **big**



Transformer Architecture - Application to Machine Translation

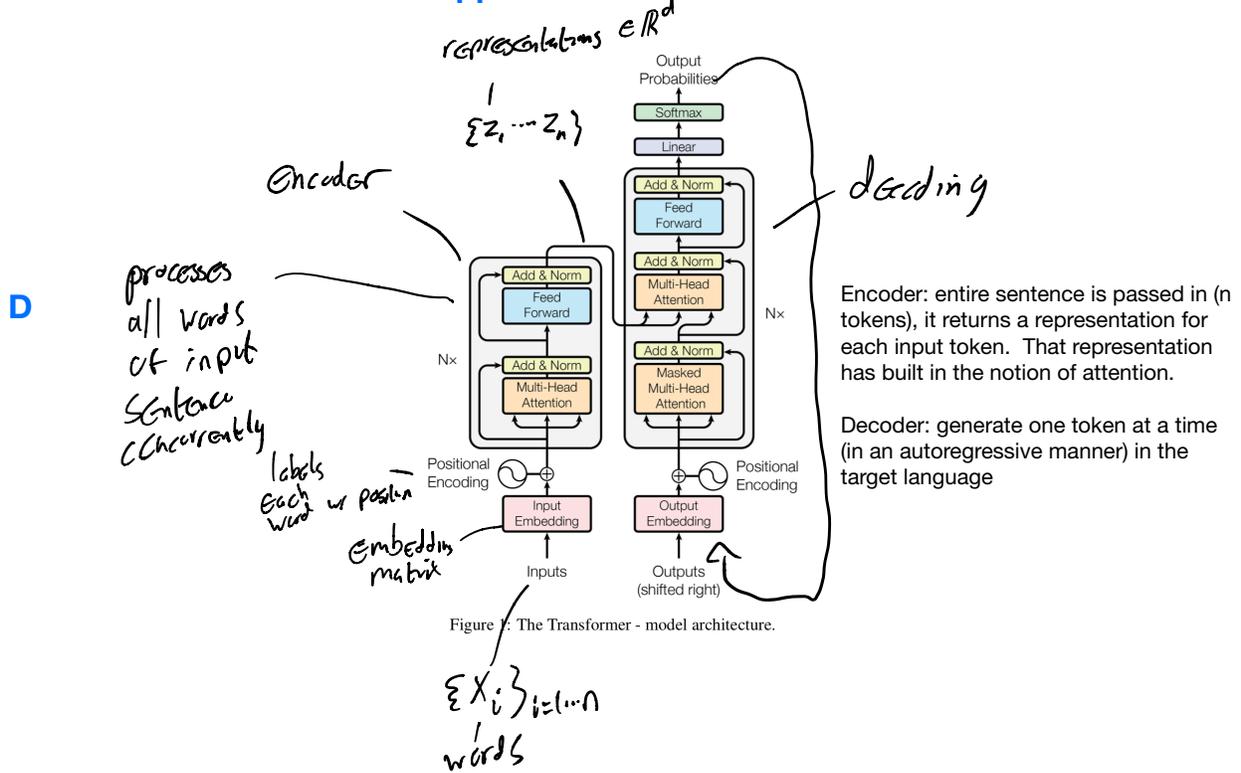


Figure 1: The Transformer - model architecture.

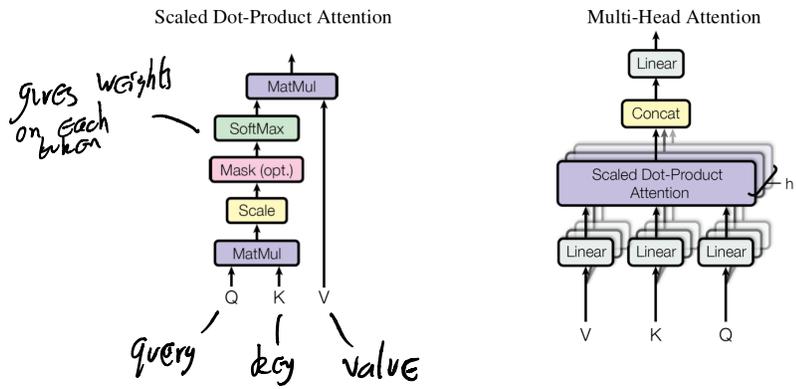


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

k_i	v_i

Soft query $q \cdot k$ - how aligned is w/ each key - $v \cdot \text{score}$

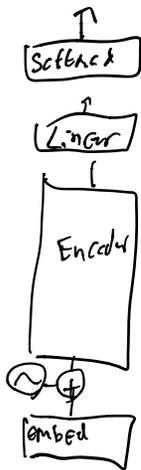
$\frac{q \cdot k}{\sqrt{d}}$

$\text{softmax} \left(\frac{q \cdot k}{\sqrt{d}} \right)$ - prob dist over keys

$\text{softmax} \left(\frac{q \cdot k}{\sqrt{d}} \right) \cdot v$ - weight avg value as per the dist.

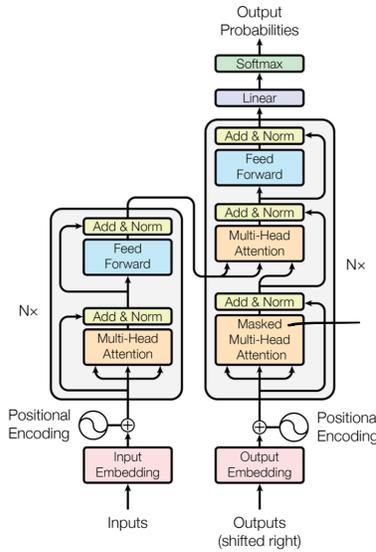
Application to other NLP tasks

If you wanted to do sentiment analysis how would you modify this architecture?



What architecture would I use for unconditional text generation?

use the decoder

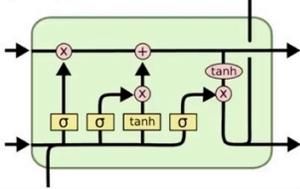


can only attend to words gen. already

Figure 1: The Transformer - model architecture.

Why Transformers are awesome

- All-to-all comparisons can be done fully parallel
 - GPUs change the game for compute
 - N^2 but extra parallel operations can be "free"
 - (RNN/LSTM must be computed in serial per token.)
- Sigmoid / tanh activations are tough



SEATTLE
LSTM is dead. Long Live Transformers!

have to process tokens serially

Sequence Modeling

Challenges with RNNs	Transformer Networks
<ul style="list-style-type: none"> • Long range dependencies • Gradient vanishing and explosion • Large # of training steps • Recurrence prevents parallel computation 	<ul style="list-style-type: none"> • Facilitate long range dependencies • No gradient vanishing and explosion • Fewer training steps • No recurrence that facilitate parallel computation

CS480/680 Lecture 19: Attention and Transformer Networks
68,754 views · Jul 16, 2019

Pascal Poupart
6K subscribers

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)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$