Member-only story

Attention models in NLP a quick introduction

Manish Chablani · Follow
Published in Towards Data Science · 5 min read · Aug 11, 2017

194

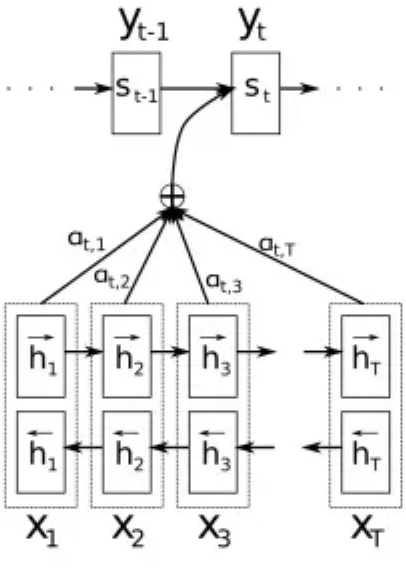
Credits: Here is abridged version of wildml article: http://www.wildml.com/2016/01/attention-and-memory-in-deep-learning-and-nlp/

Typical seq2seq models usually are of the form explained in my blog: https://medium.com/towards-data-science/sequence-to-sequence-model-introduction-and-concepts-44d9b41cd42d

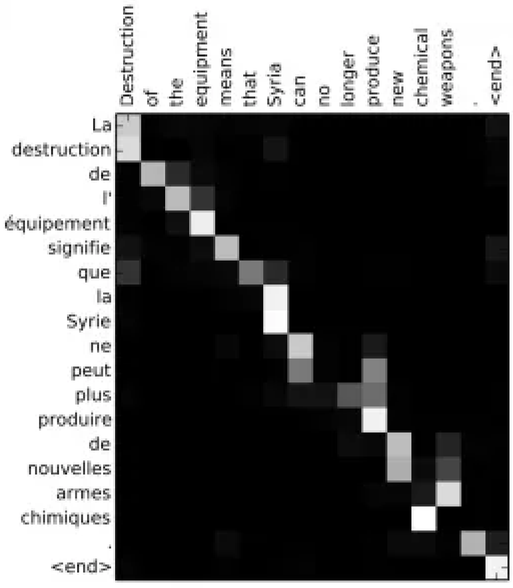
When assuming language to language translation example: Decoder is supposed to generate a translation solely based on the last hidden state from the encoder. This vector must encode everything we need to know about the source sentence. It must fully capture its meaning. In more technical terms, that vector is a sentence *embedding*. In fact, if you plot the embeddings of different sentences in a low dimensional space using PCA or t-SNE for dimensionality reduction, you can see that semantically similar phrases end up close to each other. That's pretty amazing.

Still, it seems somewhat unreasonable to assume that we can encode all information about a potentially very long sentence into a single vector and then have the decoder produce a good translation based on only that. Let's say your source sentence is 50 words long. The first word of the English translation is probably highly correlated with the first word of the source sentence. But that means decoder has to consider information from 50 steps ago, and that information needs to be somehow encoded in the vector. Recurrent Neural Networks are known to have problems dealing with such long-range dependencies. In theory, architectures like LSTMs should be able to deal with this, but in practice long-range dependencies are still problematic. For example, researchers have found that reversing the source sequence (feeding it backwards into the encoder) produces significantly better results because it shortens the path from the decoder to the relevant parts of the encoder. Similarly, feeding an input sequence twice also seems to help a network to better memorize things. It makes things work better in practice, but it's not a principled solution. Most translation benchmarks are done on languages like French and German, which are quite similar to English (even Chinese word order is quite similar to English). But there are languages (like Japanese) where the last word of a sentence could be highly predictive of the first word in an English translation. In that case, reversing the input would make things worse. So, what's an alternative? Attention Mechanisms.

With an attention mechanism we no longer try encode the full source sentence into a fixed-length vector. Rather, we allow the decoder to "attend" to different parts of the source sentence at each step of the output generation. Importantly, we let the model **learn** what to attend to based on the input sentence and what it has produced so far. So, in languages that are pretty well aligned (like English and German) the decoder would probably choose to attend to things sequentially. Attending to the first word when producing the first English word, and so on. That's what was done in Neural Machine Translation by Jointly Learning to Align and Translate and look as follows:



Here, The y's are our translated words produced by the decoder, and the x's are our source sentence words. The above illustration uses a bidirectional recurrent network, but that's not important. The important part is that each decoder output word yt now depends on a **weighted combination of all the input states**, not just the last state. The a's are weights that define in how much of each input state should be considered for each output. So, if a32 is a large number, this would mean that the decoder pays a lot of attention to the second state in the source sentence while producing the third word of the target sentence. The a's are typically normalized to sum to 1 (so they are a distribution over the input states). A big advantage of attention is that it gives us the ability to interpret and visualize what the model is doing. For example, by visualizing the attention weight matrix a when a sentence is translated, we can understand how the model is translating:



Here we see that while translating from French to English, the network attends sequentially to each input state, but sometimes it attends to two words at time while producing an output, as in translation “la Syrie” to “Syria” for example.

The Cost of Attention

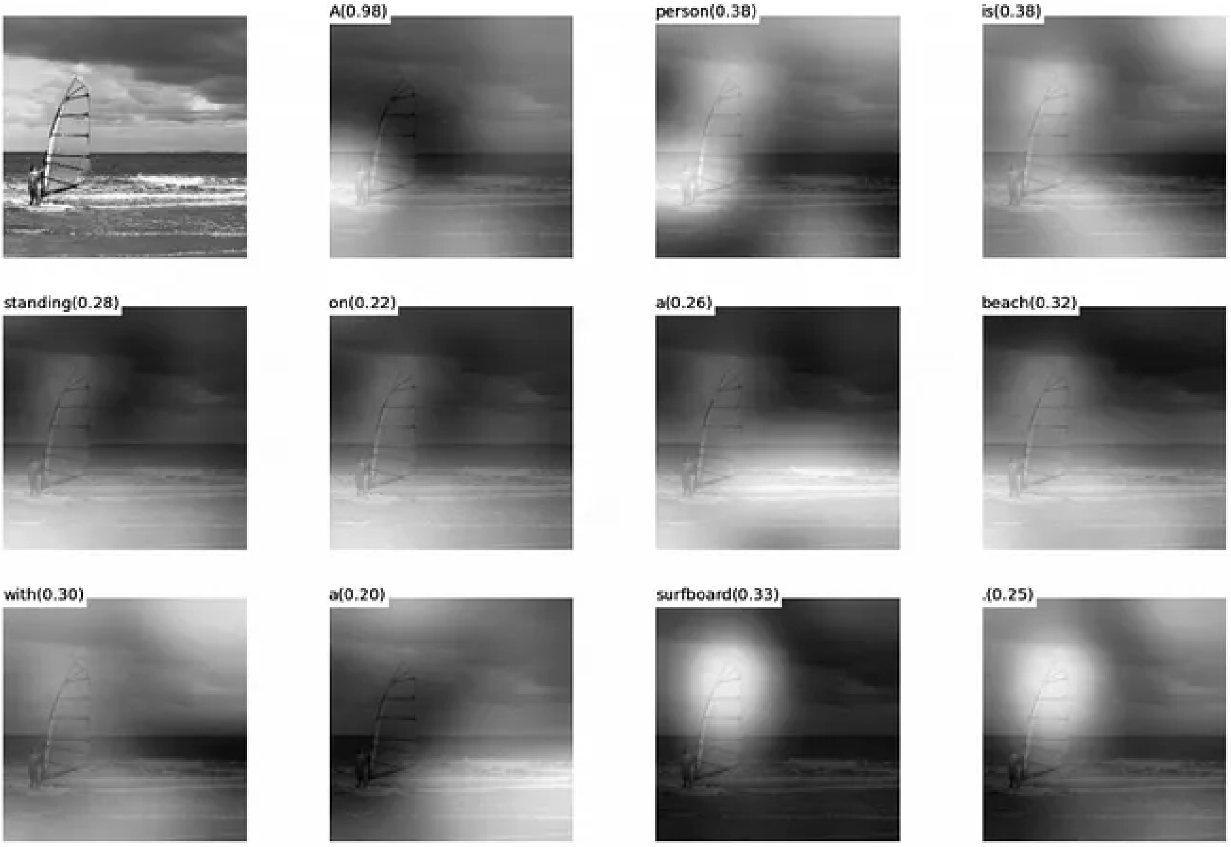
If we look a bit more look closely at the equation for attention we can see that attention comes at a cost. We need to calculate an attention value for each combination of input and output word. If you have a 50-word input sequence and generate a 50-word output sequence that would be 2500 attention values. That’s not too bad, but if you do character-level computations and deal with sequences consisting of hundreds of tokens the above attention mechanisms can become prohibitively expensive. That seems like a waste, and not at all what humans are doing. In fact, it’s more akin to memory access, not attention, which in my opinion is somewhat of a misnomer (more on that below). Still, that hasn’t stopped attention mechanisms from becoming quite popular and performing well on many tasks.

An alternative approach to attention is to use Reinforcement Learning to predict an approximate location to focus to. That sounds a lot more like human attention, and that's what's done in Recurrent Models of Visual Attention.

Attention beyond Machine Translation

Attention mechanism from above can be applied to any recurrent model.

In Show, Attend and Tell the authors apply attention mechanisms to the problem of generating image descriptions. They use a Convolutional Neural Network to "encode" the image, and a Recurrent Neural Network with attention mechanisms to generate a description. By visualizing the attention weights (just like in the translation example), we interpret what the model is looking at while generating a word:



(b) A person is standing on a beach with a surfboard.

Machine Learning NLP Rnn Attention Deep Learning

More from the list: "NLP"

Curated by Himanshu Birla

Jon Gi... in Towards Data ...

Jon Gi... in Towards Data ...

Jon Gi... in

Characteristics of Word Embeddings

✦ · 11 min read · Sep 4, 2021

The Word2vec Hyperparameters

✦ · 6 min read · Sep 3, 2021

The Word2ve

✦ · 15 min rea

View list

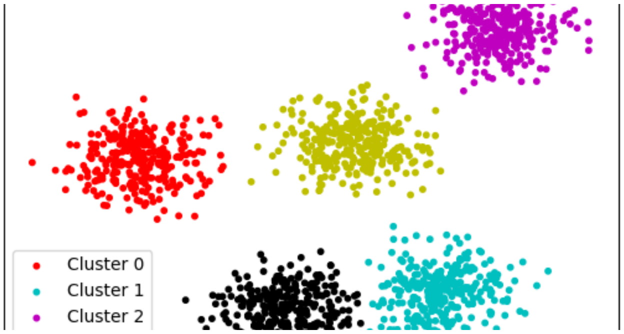
Written by Manish Chablani

Follow

1.92K Followers · Writer for Towards Data Science

Head of AI @EightSleep , Marathoner. (Past: AI in healthcare @curaiHQ , DL for self driving cars @cruise , ML @Uber , Early engineer @MicrosoftAzure cloud

More from Manish Chablani and Towards Data Science



Manish Chablani in Towards Data Science

Semantic similarity classifier and clustering sentences based on...

Recently we have been doing some experiments to cluster semantically similar...

Antonis Makropoulos in Towards Data Science

How to Build a Multi-GPU System for Deep Learning in 2023

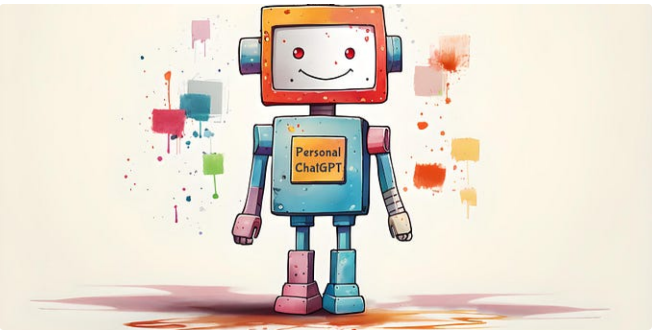
This story provides a guide on how to build a multi-GPU system for deep learning and...

✦ · 2 min read · Jun 30, 2019

136 6

10 min read · Sep 17

549 11



Robert A. Gonsalves in Towards Data Science

Your Own Personal ChatGPT

How you can fine-tune OpenAI's GPT-3.5 Turbo model to perform new tasks using you...

✦ · 15 min read · Sep 8

595 7

Manish Chablani

Building a Q&A bot with LangChain and OpenAI API

Its a well know that LLM's hallucinate more so specifically when exposed to adversarial...

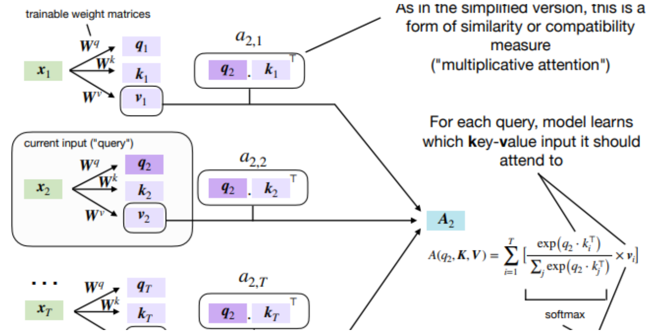
6 min read · Jun 13

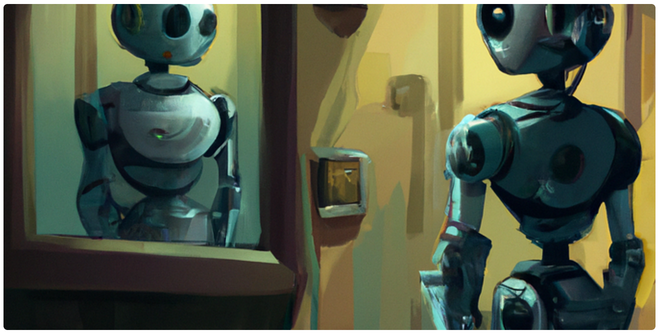
1 1

See all from Manish Chablani

See all from Towards Data Science

Recommended from Medium


trainable weight matrices
W^q
W^k
W^v
x_1
q_1
k_1
v_1
$a_{2,1}$
$q_2 \cdot k_1^T$
As in the simplified version, this is a form of similarity or compatibility measure ("multiplicative attention")
current input ("query")
x_2
q_2
k_2
v_2
$a_{2,2}$
$q_2 \cdot k_2^T$
For each query, model learns which key-value input it should attend to
A_2
$A(q_2, K, V) = \sum_{i=1}^{T} \left[\frac{\exp(q_2 \cdot k_i^T)}{\sum_j \exp(q_2 \cdot k_j^T)} \times v_i \right]$
softmax
x_T
q_T
k_T
$a_{2,T}$
$q_2 \cdot k_T^T$



Zain ul Abideen

Attention Is All You Need: The Core Idea of the Transformer

An overview of the Transformer model and its key components.

6 min read · Jun 26

144

Thomas van Dongen in Towards Data Science

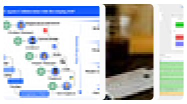
Demystifying efficient self-attention

A practical overview

20 min read · Nov 7, 2022

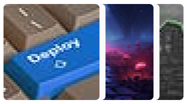
477 2

Lists



Natural Language Processing

669 stories · 283 saves



Predictive Modeling w/ Python

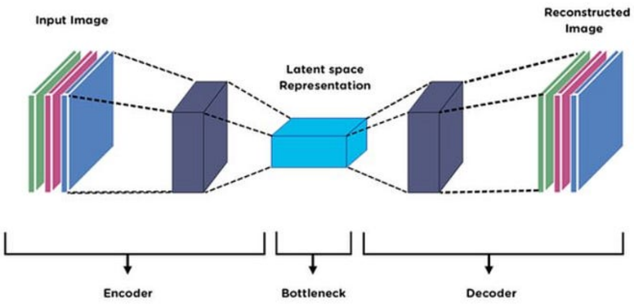
20 stories · 452 saves

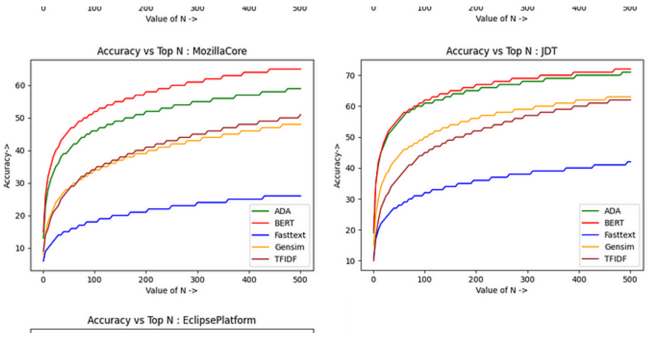
Practical Guides to Machine Learning

10 stories · 519 saves

The New Chatbots: ChatGPT, Bard, and Beyond

13 stories · 133 saves





Ahmadsabry

A Perfect guide to Understand Encoder Decoders in Depth with...

Introduction

6 min read · Jun 24

Avinash Patil

Embeddings: BERT better than ChatGPT4?

In this study, we compared the effectiveness of semantic textual similarity methods for...

4 min read · Sep 19

3 1

Tiya Vaj

The key difference between Transformers architecture and th...

The key difference between Transformers architecture and the GPT architecture used i...

2 min read · May 2

Tomas Vykruta

Understanding Causal LLM’s, Masked LLM’s, and Seq2Seq: A...

In the world of natural language processing (NLP), choosing the right training approach i...

7 min read · Apr 30

20

See more recommendations