Why Deep Language Understanding (NLP) cannot be solved as easily as image recognition, but there is hope.

Tomas Vykruta Follow Feb 7

The second AI winter came to a sudden finale in 2012 when AlexNet took first prize in the ImageNet competition. **Deep learning was born**.

Yet, it took many more years before similar deep learning techniques were effectively applied to language. Words and characters, surely, can't be as complex to interpret as images or pixels which consume thousands of times more memory? Au contraire, language is a **much** more complex problem than vision.

Vision is child's play

Vision is an easy to solve problem. Animals and babies are born with this ability, it is not learned. Even simple organisms like insects have basic vision ability. We can borrow from physics and say that visual information is spatially and temporally coherent: when a tree is observed, all the parts are connected. When the wind moves branches, the movements trace a deterministic pattern.

Images are not abstract or symbolic. Unlike the word “tree”, which is an abstract symbol for a tree, the visual image of a tree we observe is in fact the tree itself and not open to interpretation.

Images are already encoded in a perfect statistical tokenized schema of columns and rows with a 3 number representation (RGB). They don’t need to transformed or tokenized, and they can be scaled and rotated. It’s a linear problem, perfect for machine learning. Video just adds a temporal dimension.

Human language is a modern technology

Human language is a very difficult non linear problem and a recent invention (few 1000 years old). Animals can never learn it, and it takes humans a few years of practice to become proficient. Language is not spatially nor temporally coherent. I could give you advice like “That was the best meal I ever had, but I got food poisoning later, the staff was rude, and they brought the food 30 minutes late.” There is incredible complexity in that sentence. The latter statements modify the sentiment of the first statement. The worde “They” references staff, and “30 minutes late” refers to the food mentioned in the first phrase.

Language encodes different types of concepts: categorical (SUVs vs sports cars), continuous (the crash occurred at **70mph**), abstract, and so fourth. There is also context and sentiment. These concepts all require specialized processing, both by humans and by computers.

Attention Mechanisms are the new frontier

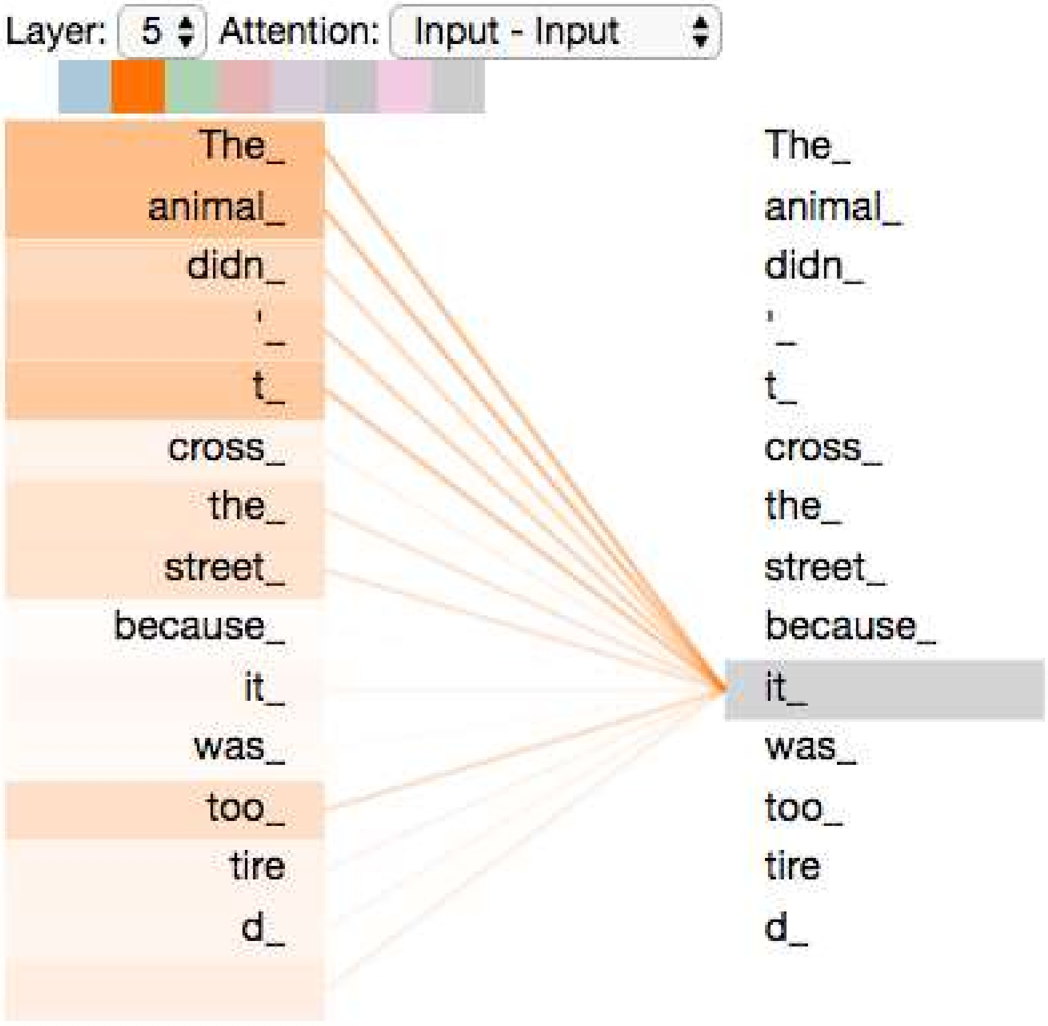
RNN’s and CNN’s showed early promise of deep learning applied to language, but both suffer from a fundamental flaw: insufficient long term memory. Very briefly let’s review why. The RNN architecture copies state from the data set inside its neurons. You can immediately see the problem. If your RNN is made of 1M neurons and your data set is 100M tokens, the best you can fit is about 1% in tne NN “memory”. LSTM’s and GRU’s improve on basic RNN memory by being selective about *what* to remember but don’t fundamentally solve the problem. No matter how large the hidden state becomes, eventually, permanent memory loss occurs.

What if we fundamentally restructure this problem such that we *didn’t have to remember anything at all?* Let’s illustrate with a human analogy first. Instead of reading War And Peace and later answering questions entirely from memory, you leave sticky notes in the pages as you read through, and use them to find answers to question. Since you’re a computer, you’re able to find the sticky notes and read the relevant text almost *instantly*. It further helps to know what *types of question will be asked* when you read through so you know what types of sticky notes to leave.

Attention Mechanism

Attention is deep learning’s answer to sticky notes: a weighted map that references previously observed parts of the document. Unlike RNN’s which are limited by hidden state memory in remembering previously observed parts, the attention map has no such limit as it doesn't actually remember anything.

note: Like many concepts in machine learning, Attention is poorly named. It has a lot to do with memory, little to do with paying attention.



Layer: 5 Attention: Input - Input

Attention is implemented as a Searchable Map, which works a lot like a regular hash map with a few subtle differences:

- You don't need an exact key for lookup, you can use a loose search query
- The result can include multiple entries which are attention-weighted

Another notable advantage is attention has over RNN is interpretability, or attribution. RNN hidden state is very hard to interpret, while attention maps are relatively easy to interpret as they behave a lot more like a simple logistic regression net with direct connections and weights.

A lot of details were intentionally skipped here, but hopefully you can walk away with an intuitive understanding of why attention is such an important breakthrough in neural networks.

For a deeper understanding, learn about the evolution of attention networks:

- Recurrent Attention
- Self attention
- Learned Self Attention
- Memory Networks

Inspired by work by Paul Dubs from Google Brain.