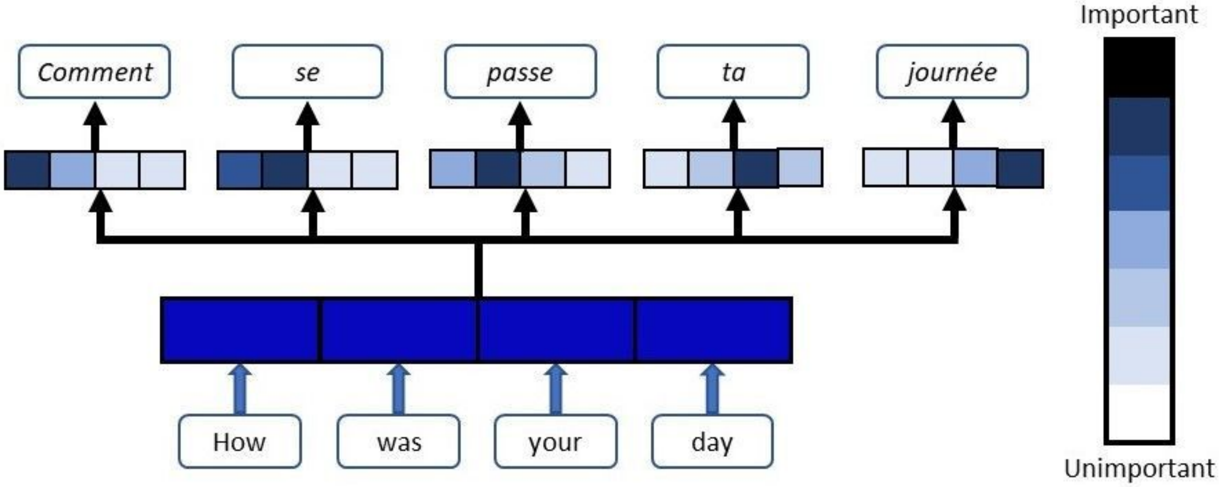
Attention

By Vladislav Ruskovoloshin, Helly Patel, Gnaneshwar Mandava (Group 6)

How Attention Began

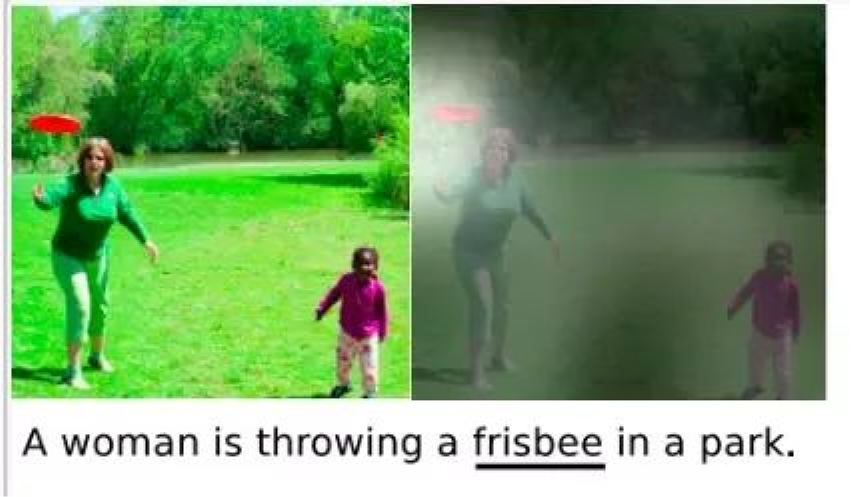
The concept of attention was first used in conjunction with encoder/decoder neural networks. There is an issue with LSTM where it was difficult for them to keep track of long term time dependencies. Given a long string of words to translate to a different language LSTM's struggled the longer the sentence got and in certain scenarios. To combat this attention was created. For every word an attention vector was calculated in reference to all other words. The vector would rate the importance of every other word in reference to the current word. This vector was calculated for every word and captured that long time dependency that LSTMs struggled with.

The idea is to have an attention vector for every word. Each attention vector would get passed to the encoder which would have different weights for each of the attention vectors depending on which part it was trying to translate. The weights add up to one for each input and they correspond to probabilities. The weights are calculated using a softmax function. By adding up the different attention vectors multiplied by their corresponding weights you get a context vector which provides the information needed to produce a result. This information is combined with the hidden state of the decoder network and the previous output. The results when compared to standard LSTM encoder/decoder networks are improved.



How Attention is Applied to Computer Vision

When dealing with nonvisual data the dependence happens over time. In vision that dependency is space. The goal is to apply attention to specific parts of the image and pass that information to a neural network and do this several times until you have gathered all the necessary information to get some form of output. This works because it can be faster to process small parts of the image rather than looking at all of it at once. The training process can be more complicated though, but the results outperform non-attention models.



As you can see in the image there is a focus on the people and the frisbee. The image on the right shows what the attention mechanism focused on to create the caption. Each pixel has a weight corresponding to how useful it was to generating the caption and the lighter it is the more useful it was. A good example of how attention looks. This is an example of soft attention. There are many different ways to apply the concept of attention to computer vision.

Soft Attention vs Hard Attention

The difference between soft attention and hard attention is how the glimpse is stored. In soft attention the glimpse is the entire image but with more detail in the selected sub area. In hard attention the glimpse is just the sub area. Hard attention is faster however it is harder to compute since it is not differentiable. It can be trained using a Reinforcement algorithm.

Reinforcement learning is a cool concept. It is an alternative to supervised learning. A neural network trained on reinforcement learning can learn by exploring randomly with the help of a reward function. Eventually it will learn to play a game or in this scenario focus its attention on the correct areas of the screen (There are other applications). You can also separate the image into parts and randomly select regions to test but that is not as effective.

Soft attention is differentiable because it contains the entire image. It is differentiable because based on the current focused location you can determine where to go to next since the entire image is still being stored and is visible. It is trained slower and is often slower than a regular neural network but easier to train.

Squeeze Excite

A form of attention that is applied to the channels of a convolutional layer in the form of weights. Seems to improve accuracy of vision based tasks while decreasing complexity of network.

Transformer Network

This is the network of the future. It uses attention blocks called transformers and it is the next step in the evolution of neural networks that work with sequences. It performs better then LSTMs because it can be computed faster with less training steps and performs better with long range dependencies. As a result LSTMs might be replaced by transformer networks which use the idea of attention. Because of the way attention neural networks work, long range dependencies have the same likelihood of being found as short term ones which is a significant improvement over LSTMs.

Transformer networks work similar to databases. There is a query, keys and values. The query is an input that gets compared to keys and the output is the value attached to the keys. Unlike a database the value that is output is not just the value associated with the key greatest similarity but a combination of all the values multiplied by weights determining how similar the key was to the query. So the similarities of each key in comparison with the query is multiplied by the value and all the values are added up and an attention vector is the final product.

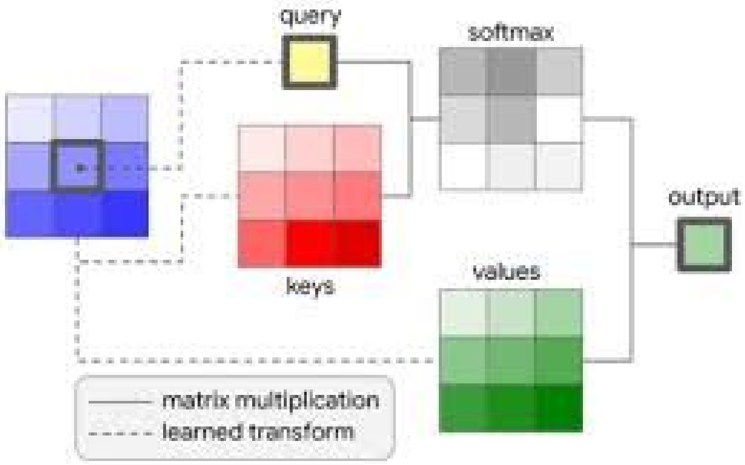
$$a = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

Here is an example of this where the query and the key are compared (via dot product). Both Q and K are vector representations of some information where Q is the query and K is the key. The value a is then multiplied by the value corresponding to that key. The weights for each key are calculated and the resulting vectors multiplied by the weights are added up. This is similar to a hidden state in a LSTM except it has the ability to capture long term dependencies better.

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Full Attention

The concept of self attention is applied to convolutional neural networks. It works similar to the convolutional layer in a CNN except for some slight differences. Like in a transformer block in a transformer neural network there is a query, keys, and values. The attention output will be calculated for every individual pixel. There will also be a window of kxk size that will be used around that pixel to determine its value. Values and keys are calculated similarly to the way a convolutional filter in a CNN would be calculated by updating through back propagation.



It would be possible to construct a CNN using attention and pooling without convolutional filters. In the research paper “Stand-Alone Self-Attention in Vision Models” the researchers developed a full attention neural network. They produced better results to a regular CNN in classification tasks for all the different depths that they tried. Although it is important to note the results were not better by a significant margin, only by about a percent.

Sources

https://mcneela.github.io/math/2018/04/18/A-Tutorial-on-the-REINFORCE-Algorithm.html
https://medium.com/@shairozsohail/a-survey-of-visual-attention-mechanisms-in-deep-learning-1043eb25f343
https://papers.nips.cc/paper/8302-stand-alone-self-attention-in-vision-models.pdf
https://www.youtube.com/watch?v=SysgYptB198
https://www.youtube.com/watch?v=quoGRI-1l0A
https://www.youtube.com/watch?v=JgvyzIkgxF0
https://www.youtube.com/watch?v=W2rWgXJBZhU
https://www.youtube.com/watch?v=OyFJWRnt_AY