You Just Want Attention

Charlie Puth (Actually: Omar Attia)
CS886: Deep Learning and NLP
Winter 2020
Charlie Puth Vs. Google Research

- Charlie Puth released his hit “You Just Want Attention” in April 2017 (1B+ views)
- Google Research did not publish “Attention Is All You Need” until June 2017 (5K+ citations)
- Coincidence??
Agenda:

- Motivation
- What Is Attention?
- How to Compute Attention?
- Attention Functions
- The Cost of Attention
- Attention in NLP
- BERT Attention Analysis
Motivation

• First proposed in Bahdanau et al., 2014 as an alignment mechanism.

• Basically, Attention allows the model to focus on the relevant parts of the input sequence as needed (Think of how humans pay attention to only most relevant parts of a scene)
What Is Attention?

- Attention can be described as mapping a query (Q) and a set of key-value pairs (<K, V>) to an output (Z), where the query, keys, values, and output are all vectors.

- The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function (f) of the query with the corresponding key.

The K, V, Q notation is inspired by the information retrieval literature.
How to Compute Attention?

• Calculating Attention:
  1. Similarity between the query and each key to obtain scalar scores.
  2. Use a Softmax function to normalize these scores into weights.
  3. Apply weighted sum of the values.

• Major components in attention mechanisms:
  1. Define <K, V> and Q
  2. Choose a similarity function

In a lot of NLP work, the key and value are often the same, therefore key=value=word-embeddings

\[
Attention(Q, K_i, V_i) = \frac{e^{Similarity(Q,K_i)}}{\sum e^{Similarity(Q,K_i)}} V_i
\]
Additive Attention:

The attention scores are computed using a Perceptron.

Example: LSTM + Attention

Multiplicative Attention:

The attention scores are computed using (scaled/general) dot products.

Example: Transformer

Additive attention outperforms Multiplicative attention without scaling for larger dimensions, because the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients.
The Cost of Attention

• Human attention is something that’s supposed to save computational resources. By focusing on one thing, we can neglect many other things.

• However, we need to calculate an attention value for each combination of input and output objects (for long sequences, it could become prohibitively expensive).

• Essentially looks at everything in detail before deciding what to attend to.

• Intuitively that’s equivalent outputting a translated word, and then going back through all of your internal memory of the text to decide which word to produce next. That seems like a waste, and not at all what humans are doing.

• In fact, it’s more akin to memory access (Information Retrieval), not attention.
Agenda:

- Motivation
- What Is Attention?
- How to Compute Attention?
- The Cost of Attention
- Attention Functions
- Attention in NLP:
  - Self Attention
  - Multi-Head Attention
  - Hierarchical Attention
  - Co-Attention
  - Two-Way Attention
  - Key-Value-Predict Attention
- BERT Attention Analysis
Self Attention

• **Application:** Representation Learning + Machine Translation (Transformer, BERT, GPT)

• **Intuition:** When there is no external information available, relate different positions of the same sequence to compute its internal representation (i.e. $Q = K = V$)

• Seems to learn sophisticated syntactic patterns! (More on that later)
Multi-Head Attention

- **Application:**
  - Representation Learning + Machine Translation (Transformer, BERT, GPT)

- **Intuition:** The more the merrier! Allow the model to jointly attend to information from different representation subspaces at different positions.

\[
\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O
\]

where \( \text{head}_i = \text{Attention}(QW^Q_i, KW^K_i, VW^V_i) \)
Hierarchical Attention

- **Application:** Document Classification

- **Intuition:** Employ two levels of attention: one at the word level and one at the sentence level, to capture and utilize the hierarchical nature of documents (words $\rightarrow$ sentences $\rightarrow$ document)

**Word Level:**

$$x_{it} = W_e w_{it}, \quad t \in [1, T],$$

$$\overrightarrow{h}_{it} = \text{GRU}(x_{it}), \quad t \in [1, T],$$

$$\overleftarrow{h}_{it} = \text{GRU}(x_{it}), \quad t \in [T, 1].$$

$$u_{it} = \tanh(W_w h_{it} + b_w)$$

$$\alpha_{it} = \frac{\exp(u_{it}^T u_{it})}{\sum_t \exp(u_{it}^T u_{w})}$$

$$s_i = \sum_t \alpha_{it} h_{it}.$$  

**Sentence Level:**

$$\overrightarrow{h}_i = \text{GRU}(s_i), \quad i \in [1, L],$$

$$\overleftarrow{h}_i = \text{GRU}(s_i), \quad t \in [L, 1].$$

$$u_i = \tanh(W_s h_i + b_s),$$

$$\alpha_i = \frac{\exp(u_i^T u_i)}{\sum_i \exp(u_i^T u_i)},$$

$$v = \sum_i \alpha_i h_i,$$  

**Classification:**

$$p = \text{softmax}(W_c v + b_c).$$
Co-Attention

• **Application:** Question Answering

• **Intuition:** Co-attention is computed as an alignment matrix on *all* pairs of document and query words.

• **Extra:** Use an iterative procedure to select an answer span by alternating between predicting the start point and predicting the end point. This allows for recovering from initial local maxima corresponding to incorrect answer spans.
Two-Way Attention

- **Application:** Textual Entailment

- **Intuition:** Use the same model to attend over the premise conditioned on the hypothesis, as well as to attend over the hypothesis conditioned on the premise, by simply swapping the two sequences. This produces two sentence-pair representations that we concatenate for classification.

\[
M = \tanh(W^y Y + W^h h_N \otimes e_L) \quad M \in \mathbb{R}^{k \times L}
\]

\[
\alpha = \text{softmax}(w^T M) \quad \alpha \in \mathbb{R}^L
\]

\[
r = Y\alpha^T \quad r \in \mathbb{R}^k
\]

\[
M_t = \tanh(W^y Y + (W^h h_t + W^r r_{t-1}) \otimes e_L) \quad M_t \in \mathbb{R}^{k \times L}
\]

\[
\alpha_t = \text{softmax}(w^T M_t) \quad \alpha_t \in \mathbb{R}^L
\]

\[
r_t = Y\alpha_t^T + \tanh(W^t r_{t-1}) \quad r_t \in \mathbb{R}^k
\]
Key-Value-Predict Attention

- **Application:** Language Modelling

- **Intuition:** Using $K = V$ for all applications can make training difficult ($K=V$ simultaneously store information for predicting the next word, computing the attention, and encode content relevant to future steps)

\[
\begin{bmatrix}
  k_t \\
  v_t \\
  p_t
\end{bmatrix} = h_t
\]

\[
M_t = \tanh(W^Y[k_{t-L} \cdots k_{t-1}] + (W^h k_t)1^T)
\]

\[
\alpha_t = \text{softmax}(w^T M_t)
\]

\[
r_t = [v_{t-L} \cdots v_{t-1}] \alpha^T
\]

\[
h_*^t = \tanh(W^r r_t + W^x p_t)
\]
BERT Attention Analysis

• The massive success of pre-trained attention-based models (even though these models are trained in a self-supervised fashion on unlabeled data, without explicit supervision for syntax or coreference) begs the question: What specific linguistic features do they learn?

• Methodology: Collect various statistics (e.g. average entropy, average attention weight per token) and study attention maps of BERT on different datasets.

• Some Findings:
  
  • Most heads put little attention on the current token. However, there are heads that specialize in attending heavily on the next or previous token, especially in earlier layers of the network.

  • Over half of BERT’s attention in layers 6-10 focuses on the delimiter token [SEP], which could be used by the model as a sort of “no-op”.

  • Some attention heads, especially in lower layers, have very broad attention (at most 10% of their attention mass on any single word). The output of these heads is roughly a bag-of-vectors representation of the sentence.

  • Particular heads specialize to specific aspects of syntax. For example, there are heads that find direct objects of verbs, determiners of nouns, objects of prepositions, and objects of possessive pronouns with >75% accuracy.
Examples:

[CLS] Not his autograph power-hitter McGwire's . [SEP]

[CLS] Short-term interest rates fell yesterday at the government's weekly Treasury bill auction . [SEP]

[CLS] Prices of Treasury bonds tumbled in moderate to active trading . [SEP]
References:

- Attention in NLP: https://medium.com/@joealato/attention-in-nlp-734c6fa9d983
- Attention Is All You Need: https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf
- Attention and its Different Forms: https://towardsdatascience.com/attention-and-its-different-forms-7fc3674d14dc
- Illustrated: Self-Attention: https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a