Transformers

Summary of Chapter 10 from
Speech and Language Processing,
Jurafsky and Martin, Feb. 3, 2024 draft

Michael Wollowski

Introduction

- Fluent speakers of a language bring an enormous amount of knowledge to bear during comprehension and production.
- This knowledge is embodied in many forms, perhaps most obviously in the vocabulary.
- It is estimated that young adult speakers of American English have a vocabulary size ranging from 30,000 to 100,000
- Children have to learn about 7 to 10 words a day, to arrive at observed vocabulary levels by the time they are 20 years of age.

Introduction

- Most of this growth is not happening through direct vocabulary instruction in school.
- The bulk of this knowledge acquisition happens as a by-product of reading, as part of the rich processing and reasoning that we perform when we read.
- Word learning appears to be based on the *distributional hypothesis*.
- Word meanings can be learned even without any grounding in the real world, solely based on the content of the texts we encounter over our lives.
- This knowledge is based on the complex association of words with the words they co-occur with (and with the words that those words occur with).

Introduction

- **Pretraining** is defined as learning knowledge about language and the world from vast amounts of text
- The resulting pre-trained language models are called **large language models**.
- The standard architecture for building large language models is the **transformer**.
- The transformer makes use of a novel mechanism called **self-attention**
- It developed out of the idea of *attention*.

Introduction

- Self-attention can be thought of a way to build *contextual representations* of a word's meaning.
- They integrate information from surrounding words, helping the model learn how words relate to each other over large spans of text.

The Transformer: A Self-Attention Network

- The input to a transformer is a sequence of words.
- The output is a prediction for what word comes next, as well as a sequence of contextual embedding that represents the contextual meaning of each of the input words.
- Transformers are made up of stacks of transformer blocks.
- Each block is a multilayer network that maps sequences of input vectors (x_1,...,x_n) to sequences of output vectors (z_1,...,z_n) of the same length.

The Transformer: A Self-Attention Network

- The blocks are made by combining simple linear layers, feedforward networks, and self-attention layers.
- Self-attention allows a network to directly extract and use information from arbitrarily large contexts

Transformers: The Intuition

- Across a series of layers, we build up richer and richer contextualized representations of the meanings of input words or tokens (pattern recognition)
- To compute the representation of a word *i* we combine information from the representation of *i* at the previous layer with information from the representations of the neighboring words.
- The goal is to produce a contextualized representation for each word at each position.
- In other words, they represent what a word means in the particular context in which it occurs.

Transformers: The Intuition

- We need a mechanism that tells us how to weigh and combine the representations of the different words from the context at the prior level in order to compute our representation at this layer.
- This mechanism must be able to look broadly in the context, since words have rich linguistic relationships with words that can be many sentences away.
- Even within the sentence, words have important linguistic relationships with contextual words.

Transformers: The Intuition

- Consider the following example:
  - The keys to the cabinet are on the table.
- The phrase *The keys* is the subject of the sentence.
- In English it must agree in grammatical number with the verb *are.*
- In the example, both are plural.

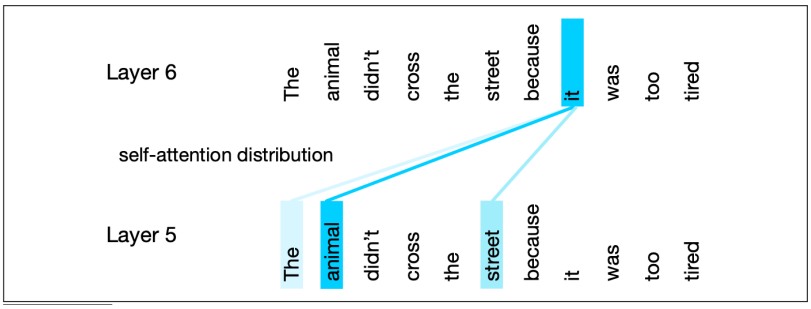
Transformers: The Intuition

- Consider the example below.
  - The chicken crossed the road because it wanted to get to the other side.
- The pronoun *it* co-refers to the chicken.

Transformers: The Intuition

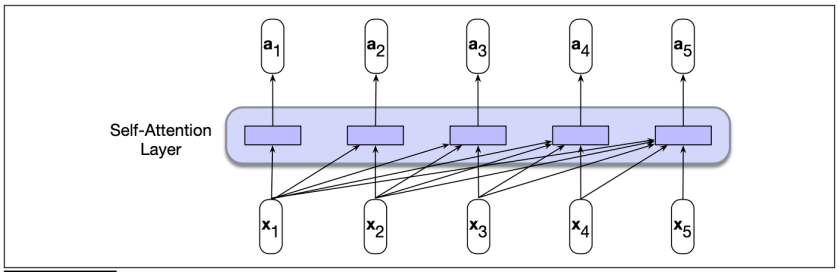
- Consider the example below.
  - I walked along the pond, and noticed that one of the trees along the bank had fallen into the water after the storm.
- The way we know that *bank* refers to the side of a pond or river and not a financial institution is from the context, including words like *pond* and *water*.
- Contextual words can be quite far way in the sentence or paragraph.
- We need a mechanism that can look broadly in the context to help compute representations for words.
- Self-attention is just such a mechanism: it allows us to look broadly in the context and tells us how to integrate the representation from words in that context from layer $k - 1$ to build the representation for words in layer k.

Transformers: The Intuition



- The figure shows the self-attention weight distribution that is part of the computation of the representation for the word *it* at layer 6.
- In computing the representation for *it*, the system attends differently to the various words at layer 5, with darker shades indicating higher self-attention values.
- The transformer is attending highly to *animal*, a sensible result, since in this example *it* co-refers with the animal.

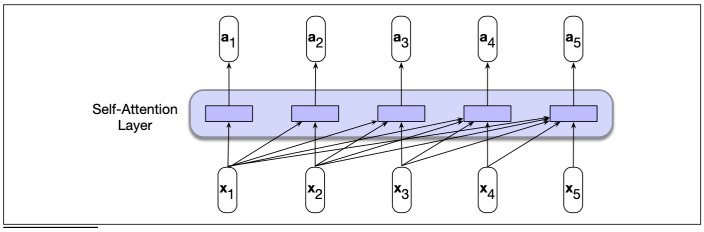
Causal or backward-looking self-attention



- The concept of *context* can be used in two ways in self-attention.
- In causal, or backward looking self-attention, the context is any of the prior words.
- In general bidirectional self-attention, the context can include future words.

Image source: Speech and Language Processing, Jurafsky and Martin, Feb. 3, 2024 draft

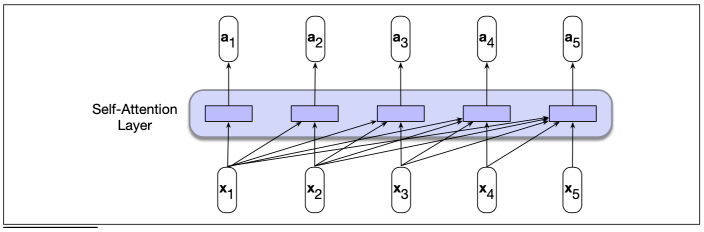
Causal or backward-looking self-attention



- The figure shows a single causal, or backward looking, self-attention layer.
- A self-attention layer maps input sequences (x_1,...,x_n) to output sequences of the same length (a_1,...,a_n).
- When processing each item in the input, the model has access to all of the inputs up to and including the one under consideration.
- It does not have access to information about inputs beyond the current one.
- The computation performed for each item is independent of all the other computations.

Image source: Speech and Language Processing, Jurafsky and Martin, Feb. 3, 2024 draft

Self-attention more formally



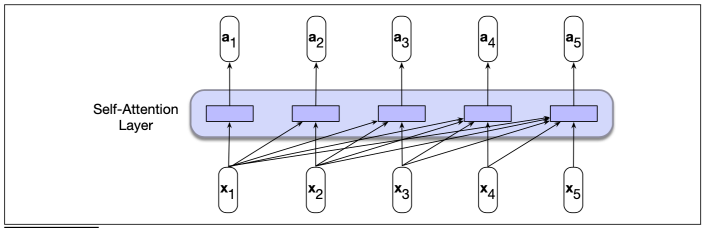
- The core intuition of attention is the idea of *comparing* an item of interest to a collection of other items in a way that reveals their relevance in the current context.
- In the case of self-attention for language, the set of comparisons are to other words (or tokens) within a given sequence.
- The result of these comparisons is then used to compute an output sequence for the current input sequence.
- For example, in the figure the computation of a_3 is based on a set of comparisons between the input x_3 and its preceding elements x_1 and x_2, and to x_3 itself.

Self-attention more formally

- How shall we compare words to other words?
- Since our representations for words are vectors, we'll make use of our old friend the dot product that we used for computing word similarity.
- Let's refer to the result of this comparison between words *i* and *j* as a score:
  - score(x_i, x_j) = $x_i \cdot x_j$
- The result of a dot product is a scalar value ranging from $-\infty$ to ∞.
- The larger the value, the more similar the vectors that are being compared.

Self-attention more formally

- In the example from the figure, the first step in computing y_3 would be to compute three scores:
  1. $x_3 \cdot x_1$,
  2. $x_3 \cdot x_2$ and
  3. $x_3 \cdot x_3$.



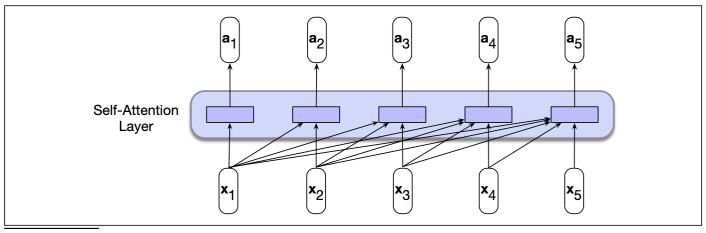
Self-attention more formally

- To make effective use of these scores, we will normalize them with softmax.
- This creates a vector of weights, αij, that indicates the proportional relevance of each input to the input element *i* that is the current focus of attention.

$$\alpha_{ij} = \text{softmax}(\text{score}(x_i, x_j)) \ \forall j \leq i$$

- Given the proportional scores in α, we then generate an output value y_i by taking the sum of the inputs seen so far, weighted by their respective α value.

$$a_i = \sum_{j \leq i} \alpha_{ij} x_j$$



Self-attention more formally

- The three steps represent the core of an attention-based approach:
  - a set of comparisons to relevant items in some context,
  - a normalization of those scores to provide a probability distribution,
  - followed by a weighted sum using this distribution.

Self-Attention

- Transformers allow us to create a more sophisticated way of representing how words can contribute to the representation of longer inputs.
- Consider the three different roles that each input embedding plays during the course of the attention process:
  - **Query**: As *the current focus of attention* when being compared to all of the other preceding inputs.
  - **Key**: In its role as *a preceding input* being compared to the current focus of attention.
  - **Value**: As a value used to compute the output for the current focus of attention.

Self-Attention

- To capture these three different roles, transformers introduce weight matrices $\mathbf{W^Q}$, $\mathbf{W^K}$, and $\mathbf{W^V}$.
- These weights will be used to project each input vector x_i into a representation of its role as a query, key, or value.

$$\mathbf{q}_i = \mathbf{W^Q x}_i$$

$$\mathbf{k}_i = \mathbf{W^K x}_i$$

$$\mathbf{v}_i = \mathbf{W^V x}_i$$

Self-Attention

- The score between a current focus of attention, x_i, and an element in the preceding context, x_j, consists of a dot product between its query vector q_i and the preceding element's key vectors k_j.
- Let's update our previous comparison calculation to reflect this:
  - $\text{score}(x_i, x_j) = x_i \cdot x_j$ -> $\text{score}(x_i, x_j) = q_i \cdot k_j$

Self-Attention

- The ensuing softmax calculation resulting in α_{ij}, j remains the same, but the output calculation for a_i is now based on a weighted sum over the value vectors v.

$$a_i = \sum_{j \le i} \alpha_{ij} x_j \quad \text{->} \quad a_i = \sum_{j \le i} \alpha_{ij} v_j$$

The softmax weight α_{ij} will likely be highest for the current focus element i, and so the value for y_i will be most influenced by v_i.

However, the model will also pay attention to other contextual words if they are similar to i, allowing their values to also influence the final value of v_j.

Context words that are not similar to i will have their values down-weighted and won't contribute to the final value.

Self-Attention

- The result of a dot product can be an arbitrarily large (positive or negative) value.
- Exponentiating large values can lead to numerical issues and to an effective loss of gradients during training.
- To avoid this, we scale down the result of the dot product, by dividing it by a factor related to the size of the embeddings.
- A typical approach is to divide by the square root of the dimensionality of the query and key vectors (d_k).

Overview

Query: As *the current focus of attention* when being compared to all of the other preceding inputs.
Key: In its role as *a preceding input* being compared to the current focus of attention.
Value: As a value used to compute the output for the current focus of attention.

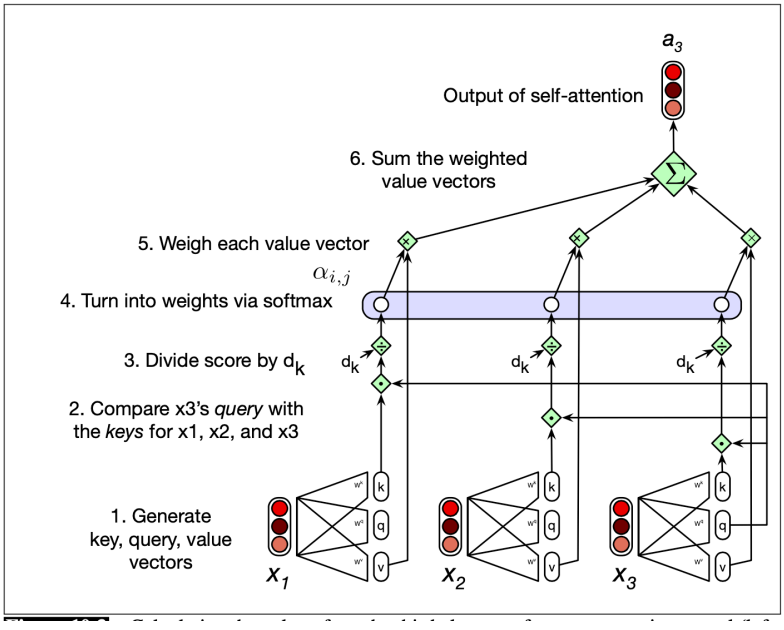


Image source: Speech and Language Processing, Jurafsky and Martin, Feb. 3, 2024 draft