Transformers – Part 1

Summary of Chapter 10 from
Speech and Language Processing,
Jurafsky and Martin, August 20, 2024 draft
Michael Wollowski

Next Word Prediction

- It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a …

Next Word Prediction

- It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a **wife**.
  - Jane Austen: Pride and Prejudice
- In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever …

Next Word Prediction

- It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a **wife**.
  - Jane Austen: Pride and Prejudice
- In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever **since**.
  - F. Scott Fitzgerald, *The Great Gatsby*
- All this happened, more or …

Next Word Prediction

- It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a **wife**.
  - Jane Austen: Pride and Prejudice
- In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever **since**.
  - F. Scott Fitzgerald, *The Great Gatsby*
- All this happened, more or **less**.
  - Kurt Vonnegut*, Slaughterhouse-Five*

Transformers: The Basics

- The transformer is the standard architecture for building large language models.
- Left-to-right (autoregressive) language modeling:
  - Given a sequence of input tokens,
  - Predict output tokens one by one,
  - Conditioned on the prior context.
- Key component of a transformer:
  - self-attention also called multi-head attention.

Quick Review of Attention

- Build contextual representations of a token's meaning.
- Attending to and integrating information from surrounding tokens.
- Helping the model learn how tokens relate to each other over large spans.

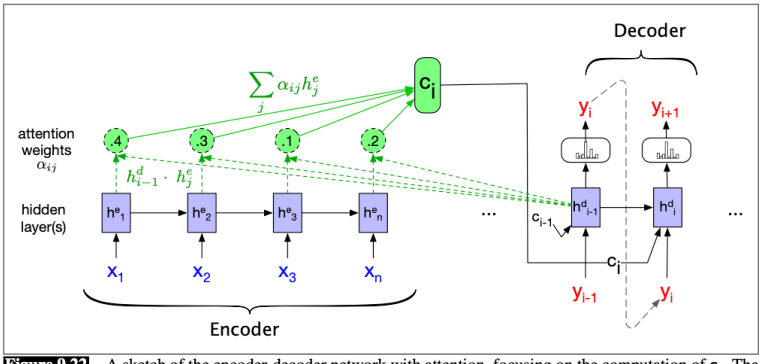


Figure 9.22 A sketch of the encoder-decoder network with attention, focusing on the computation of $\mathbf{c}_i$. The context value $\mathbf{c}_i$ is one of the inputs to the computation of $\mathbf{h}_i^d$. It is computed by taking the weighted sum of all the encoder hidden states, each weighted by their dot product with the prior decoder hidden state $\mathbf{h}_{i-1}^d$.

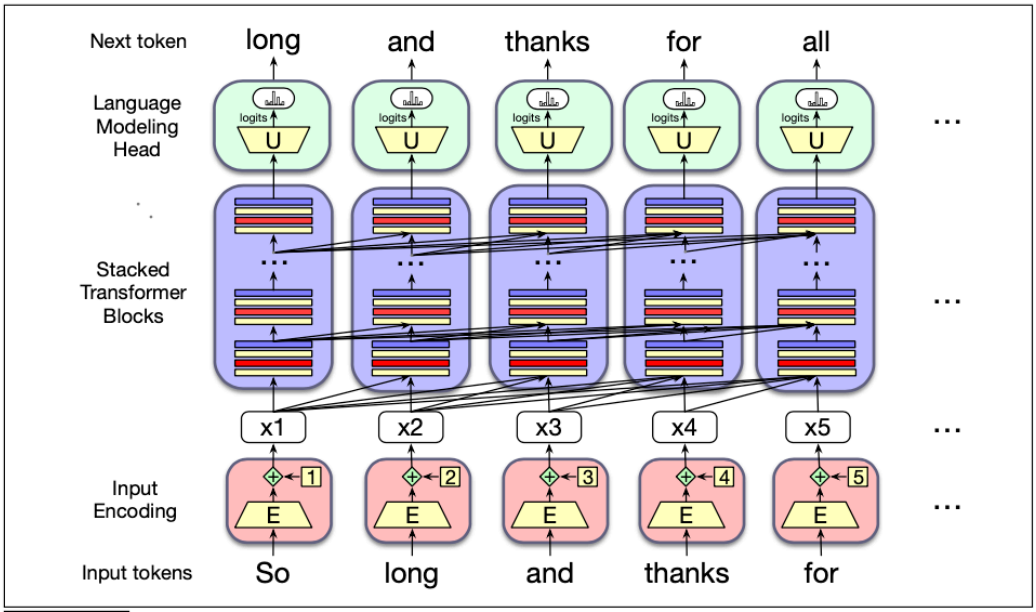


Figure 9.1 The architecture of a (left-to-right) transformer, showing how each input token get encoded, passed through a set of stacked transformer blocks, and then a language model head that predicts the next token.

Transformers: The Basic Architecture

- Unlike an RNN, a transformer processes several tokens.
- They are called the context window.
- A set of *n* blocks maps an entire input vector (x_1, ..., x_n) to an output vector (h_1, ..., h_n) of the same length.
- Typically the "blocks" are several blocks stacked on top of each other.

Transformer Blocks

- Each block is a multilayer network, consisting of:
  - a multi-head attention layer,
  - feedforward networks and
  - layer normalization steps.
- Lot's of weights!
- We will investigate those in detail.

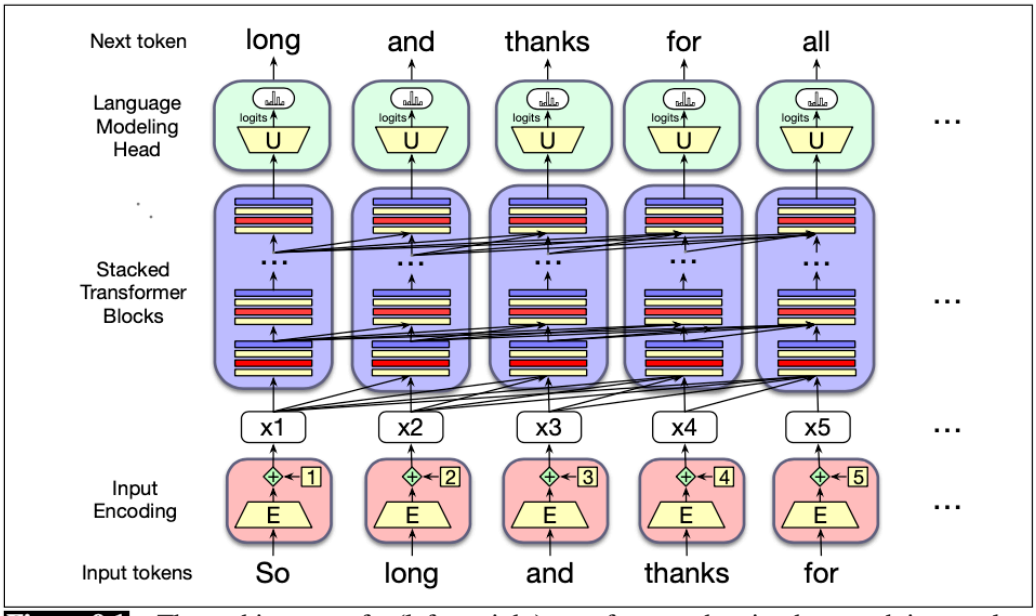


Figure 9.1 The architecture of a (left-to-right) transformer, showing how each input token get encoded, passed through a set of stacked transformer blocks, and then a language model head that predicts the next token.

Transformers: The Basics

- **Input encoding** though embedding matrix E
- **Language modeling head** through unembedding matrix U.
- Number of stacked blocks: 12 to 96.
- GPT-4: 120 blocks

GPT-4

- Standard GPT-4 model offers 8,000 tokens for the context*).
- 8000 tokens amount to about 26 pages of a novel**).

*) Source: Maximum Token length in GPT-4. https://community.openai.com/t/maximum-token-length-in-gpt-4/385914
**) Assuming 250-300 words per book page. Source: https://hotghostwriter.com/blogs/blog/novel-length-how-long-is-long-enough It should be noted that the token count is typically larger than the word count.

GPT-4

- An extended 32,000 token context-length model is available*).
- 32000 tokens amount to about 106 pages of a novel**).
- Suddenly, next word prediction does not seem to be such a hard problem any longer.

*) Source: Maximum Token length in GPT-4. https://community.openai.com/t/maximum-token-length-in-gpt-4/385914
**) Assuming 250-300 words per book page. Source: https://hotghostwriter.com/blogs/blog/novel-length-how-long-is-long-enough It should be noted that the token count is typically larger than the word count.

Attention

- Embeddings represent a word's meaning by a fixed vector.
- The word "chicken" has a specific vector.
- And so does the pronoun "it."
- The vector for "it" might somehow encode that this it is a pronoun.
- However, a pronoun refers to some noun.
- This noun appears in the sentence or surrounding sentences.
- This sentences are the context.

Attention

- Consider the following examples.
  - The chicken didn't cross the road because **it** was too tired.
  - The chicken didn't cross the road because **it** was too wide.

Language and World Knowledge

- 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.
- 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.
- So, read more!

Language and World Knowledge

- Word meanings can be learned even without any grounding in the real world.
- They can be learned solely based on the content of the texts we encounter.
- So, yes, don't go out in the world! Stay in your rooms!
- This knowledge is based on the complex association of words with the words they co-occur with.

Transformers and World Knowledge

- The stacked layers in a transformer: used to build up richer and richer contextualized representations of the words in a sentence.
- The goal is to produce a contextualized representation for each word at each position.

Back to Chickens Though

- Consider:

  The chicken didn't cross the road because **it** …
- At this point we do not yet know which thing "it" is going to end up referring to.
- A representation of the input must be such that "it" can be resolved to "chicken" or "road."

Back to Chickens Though

- The self-attention weight distribution α that is part of the computation of the representation for the word "it" at layer k + 1.
- In computing the representation for it, we attend differently to the various words at layer k.
- Darker shades indicate higher self-attention values.

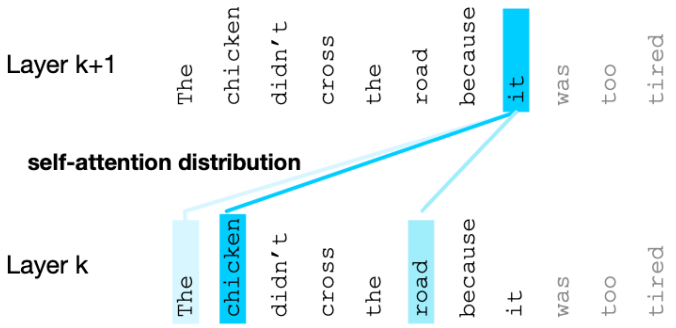
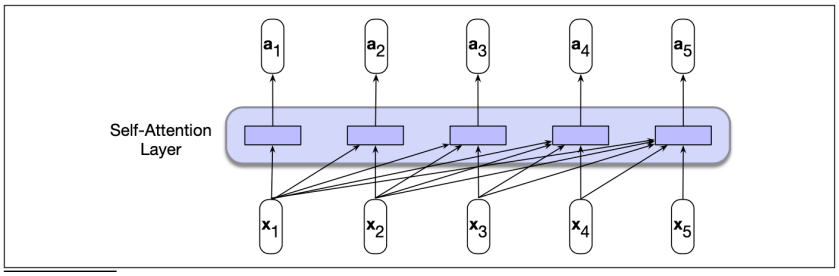


Image source: Speech and Language Processing, Jurafsky and Martin, Aug. 20, 2024 draft

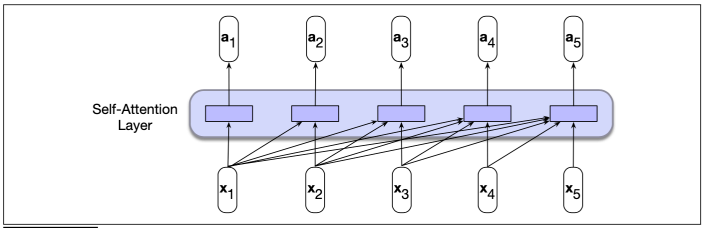
Causal or Backward-looking Self-attention



- 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

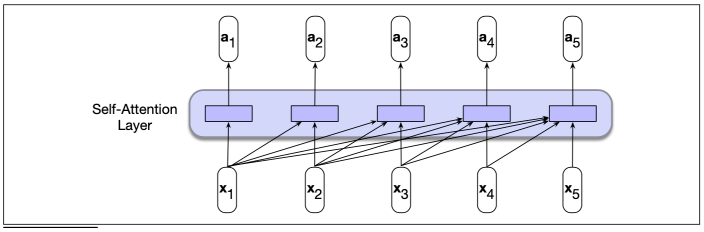
Causal or backward-looking self-attention



- 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.
- 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.

Attention on Steroids

- Let's compare a Transformer's attention mechanism to that of an RNN

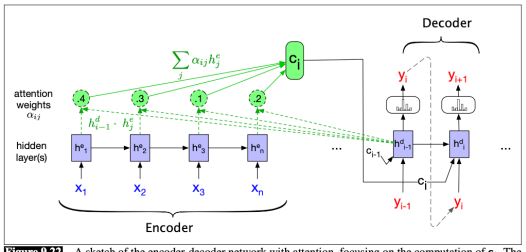
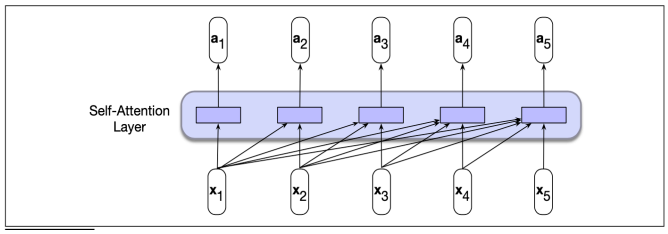


Figure 9.22 A sketch of the encoder-decoder network with attention, focusing on the computation of $\mathbf{c}_i$. The context value $\mathbf{c}_i$ is one of the inputs to the computation of $\mathbf{h}_i^d$. It is computed by taking the weighted sum of all the encoder hidden states, each weighted by their dot product with the prior decoder hidden state $\mathbf{h}_{i-1}^d$.



Simplified Version of Attention

- At its heart, attention is a weighted sum of context vectors
- With some complexity added to how the weights are computed and what gets summed.
- For now, let's look at simplified version of attention:
- Attention output $\mathbf{a}_i$ at token position *i* is simply the weighted sum of all the representations $\mathbf{x}_j$, for all j ≤ i.
- We use α_{ij} to represent how much $\mathbf{x}_i$ should contribute to $\mathbf{a}_i$:
- Simplified version: $\mathbf{a}_i = \sum_{j \le i} \alpha_{ij} x_j$

Simplified Version of Attention

- Each α_{ij} is a scalar.
- It is used for weighing the value of input $\mathbf{x}_j$ when summing up the inputs to compute $\mathbf{a}_i$.
- How shall we compute this α weighting?
- We weight each prior embedding proportionally to how similar it is to the current token *i*.
- The output of attention is a sum of the embeddings of prior tokens weighted by their similarity with the current token embedding.

Simplified Version of Attention

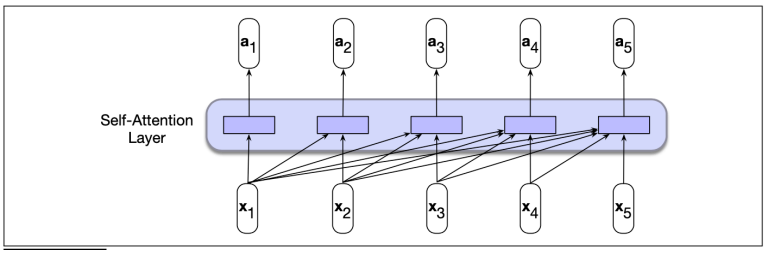
- We compute similarity scores via dot product, which maps two vectors into a scalar value ranging from $-\infty$ to ∞.
- The larger the score, the more similar the vectors that are being compared.
- We'll normalize these scores with a softmax to create the vector of weights α_{ij}, $j \leq i$.
- Simplified version:

$$\text{score}(x_i, x_j) = x_i \cdot x_j$$

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

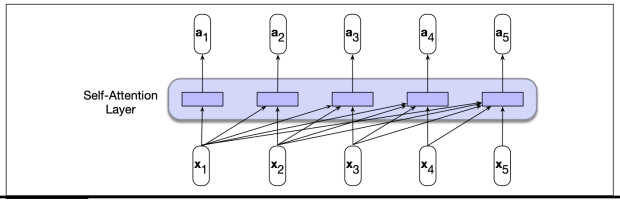
Simplified Version of Attention

- In the example from the figure, the first step in computing a_3 would be to compute three scores:
  1. $x_3 \cdot x_1$,
  2. $x_3 \cdot x_2$,
  3. $x_3 \cdot x_3$.



Simplified Version of Attention

- The resulting values are treated as weights
- They indicate the proportional relevance of the prior token to the current token at position *i.*
- The softmax value will likely be highest for $\mathbf{x}_i$, since it is very similar to itself.
- However, other context words may also be similar to *i*, and softmax will also assign some weight to those words.



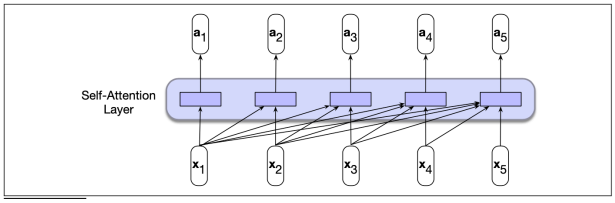
Simplified Version of Attention

- Putting everything together, we get attention $\mathbf{a}_i$:

$$\text{score}(x_i, x_j) = x_i \cdot x_j$$

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

$$\mathbf{a}_i = \sum_{j \le i} \alpha_{ij} x_j$$



Detour: Locating the Attention Head

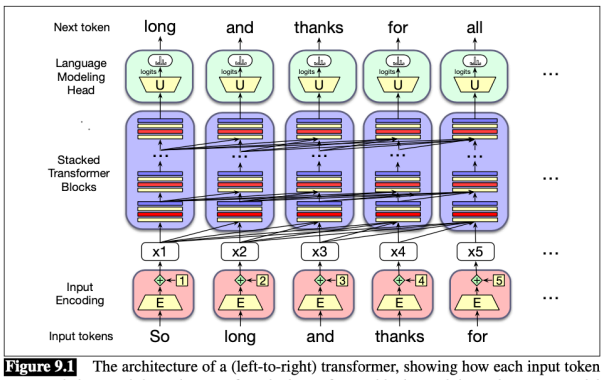
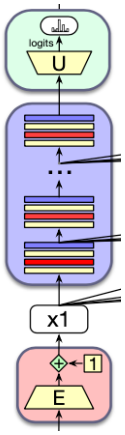
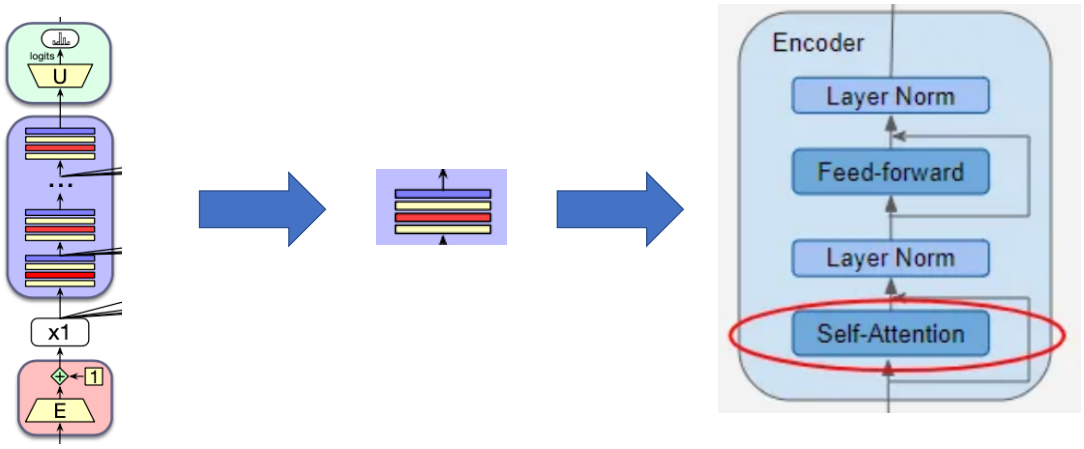


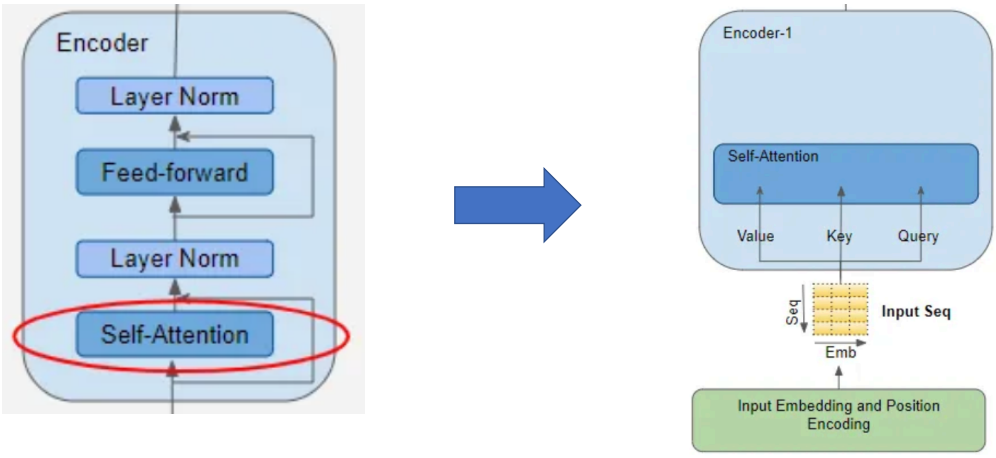
Figure 9.1 The architecture of a (left-to-right) transformer, showing how each input token get encoded, passed through a set of stacked transformer blocks, and then a language model head that predicts the next token.



Detour: Locating the Attention Head

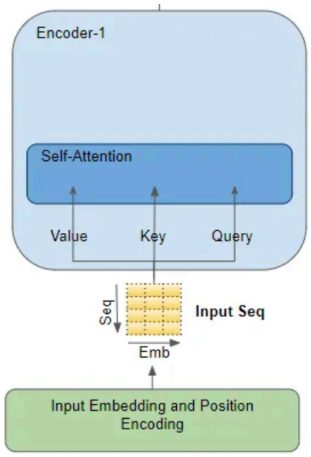


Detour: Locating the Attention Head



Less Simplified Version of Attention

- There are three different roles that each input embedding plays during the course of the attention process:
  - **Query**: As *the current element* being compared to the other preceding inputs.
  - **Key**: In its role as *a preceding input* being compared to the current element to determine a similarity weight.
  - **Value**: As a value of a preceding element that gets weighted and summed up to compute the output for the current element.



Less Simplified Version of Attention

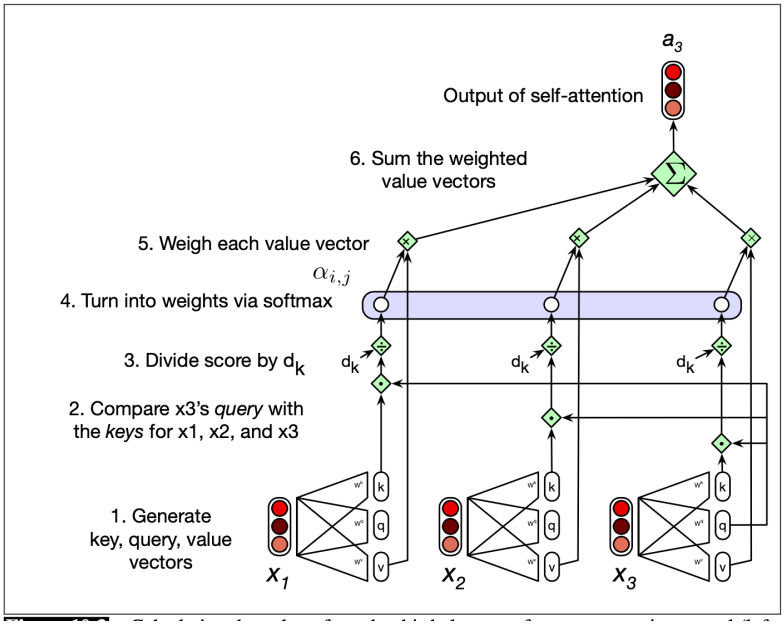


Image source: Speech and Language Processing, Jurafsky and Martin, Feb. 3, 2024 draft

Less Simplified Version of Attention

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

$$\mathbf{q}_i = \mathbf{x}_i\mathbf{W}^Q$$
$$\mathbf{k}_i = \mathbf{x}_i\mathbf{W}^K$$
$$\mathbf{v}_i = \mathbf{x}_i\mathbf{W}^V$$

- Hmmm, more weights…