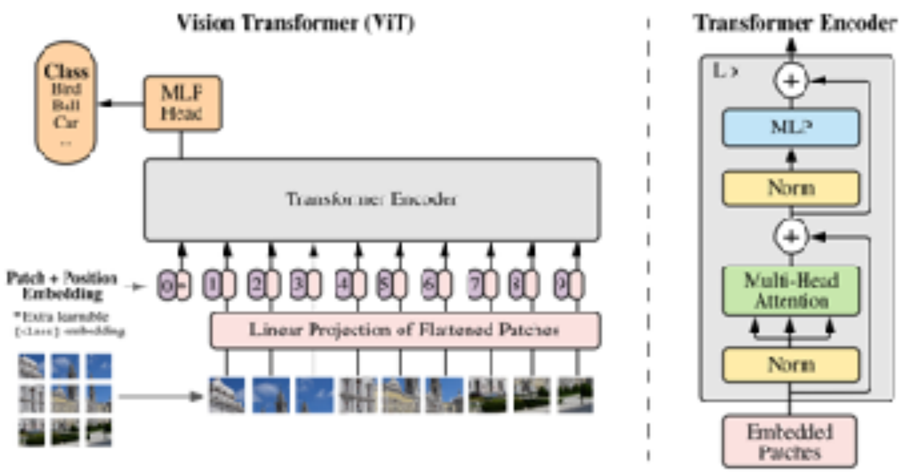
COMP 590/790:
Visual Recognition with Transformers



https://www.gedasbertasius.com/comp790-24s

Course Introduction

Gedas Bertasius

About Me

- Originally from Lithuania.

- Came to the US to play basketball.

- Got a PhD from UPenn.
- Spent 2 years at Facebook AI Research.
- Joined UNC last summer.

Introductions in Canvas

- Name?
- BA / BS / MS / PhD?
- Year?
- What are you excited about in computer vision and AI in general?
- Why are you taking this course?

Plan for Today

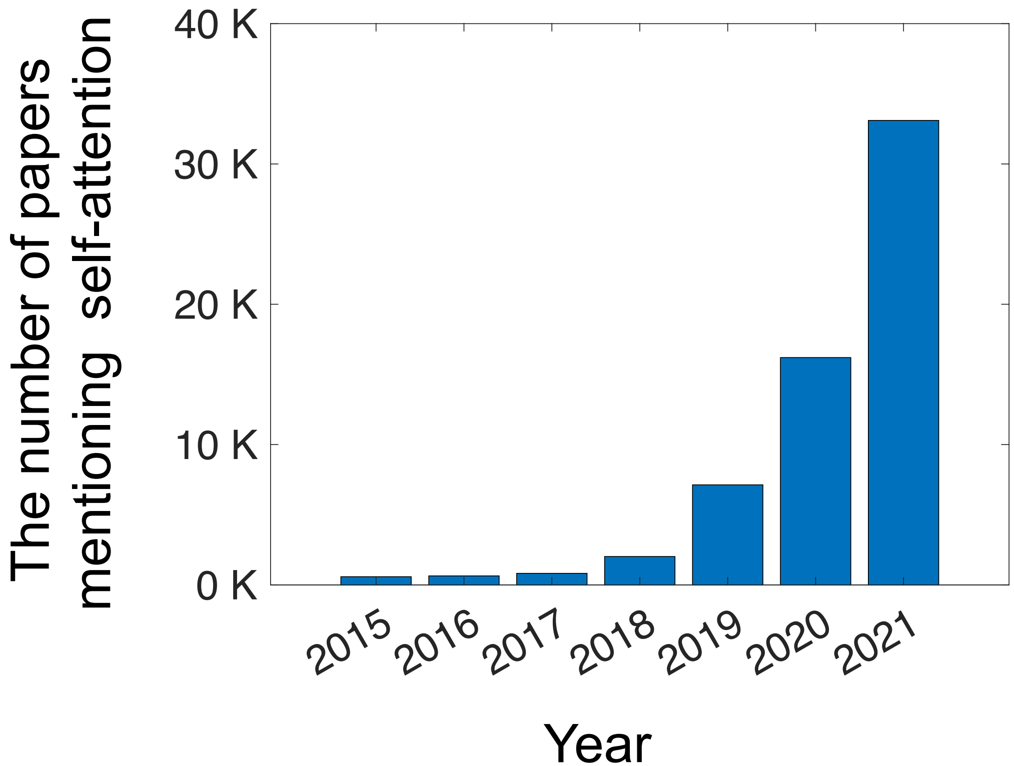
- Motivation for the course
- Course overview
- Overview of self-attention & transformers

Plan for Today

- Motivation for the course
- Course overview
- Overview of self-attention & transformers

Why Transformers?

- In the recent years, transformers have revolutionized the field of natural language processing (NLP).



*data obtained from Google Scholar

Why Transformers?

Attention is all you need
A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones… - Advances in neural information processing systems, 2017
Cited by 103837 Related articles All 62 versions

Bert: Pre-training of deep bidirectional transformers for language understanding
J Devlin, MW Chang, K Lee, K Toutanova - arXiv preprint arXiv:1810.04805, 2018
Cited by 88345 Related articles All 46 versions

Why Transformers?

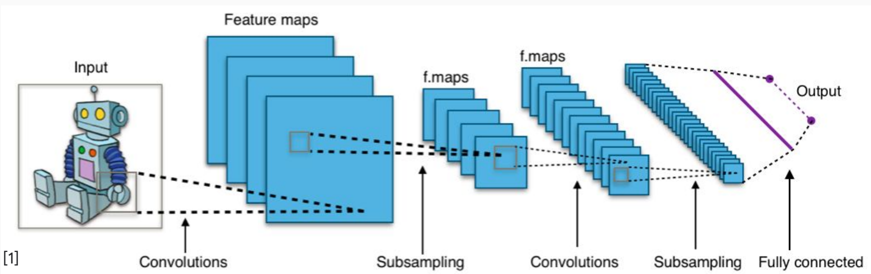
An image is worth 16x16 words: Transformers for image recognition at scale
A Dosovitskiy, L Beyer, A Kolesnikov, D Weissenborn… - arXiv preprint arXiv:2010.11929, 2020
Cited by 27499 Related articles All 12 versions

End-to-end object detection with transformers
N Carion, F Massa, G Synnaeve, N Usunier, A Kirillov… - European conference on computer vision, 2020
Cited by 9374 Related articles All 13 versions

Before Transformers

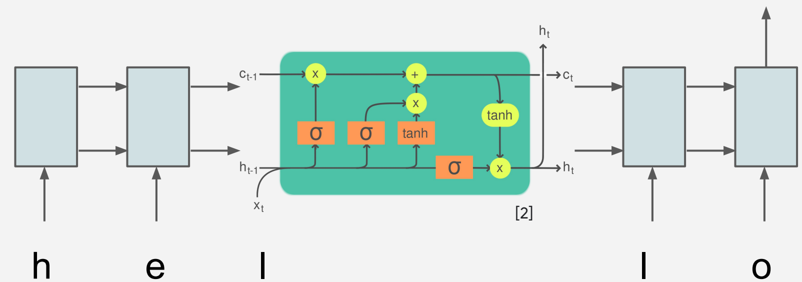
Computer Vision

Convolutional NNs (+ResNets)



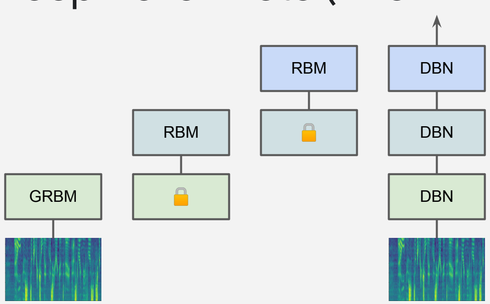
Natural Lang. Proc.

Recurrent NNs (+LSTMs)



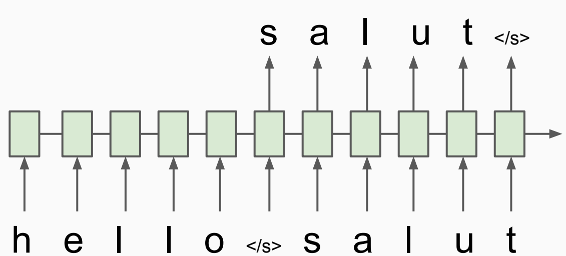
Speech

Deep Belief Nets (+non-DL)



Translation

Seq2Seq



RL

BC/GAIL

Algorithm 1 Generative adversarial imitation learning

Input: Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
for $i = 0, 1, 2, \ldots$ **do**
  Sample trajectories $\tau_i \sim \pi_{\theta_i}$
  Update the discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \quad (17)$$

  Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

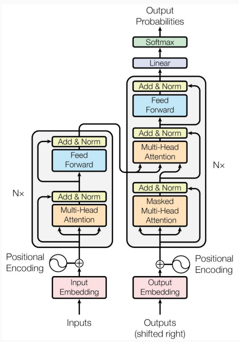
$$\hat{\mathbb{E}}_{\tau_i}[\nabla_\theta \log \pi_\theta(a|s)Q(s,a)] - \lambda \nabla_\theta H(\pi_\theta),$$
$$\text{where } Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \mid s_0 = \bar{s}, a_0 = \bar{a}] \quad (18)$$

end for

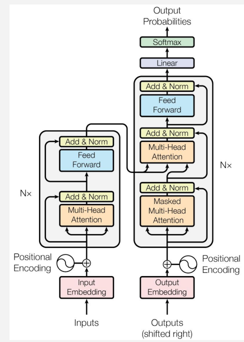
*Lucas Beyer's Slides

After Transformers

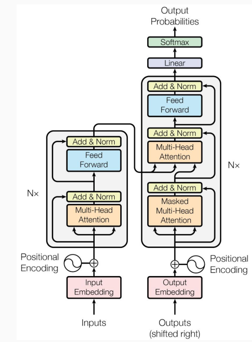
Computer Vision



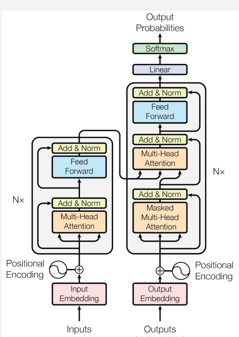
Natural Lang. Proc.



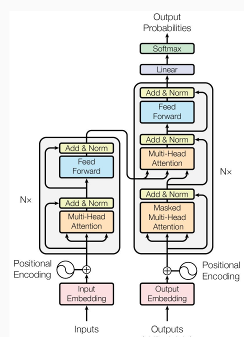
Reinf. Learning



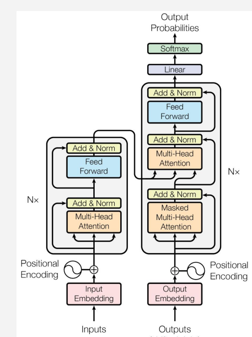
Speech



Translation



Graphs/Science

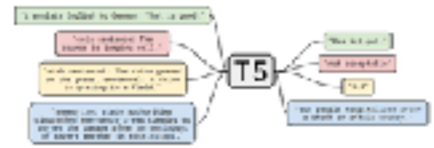


Transformer image source: "Attention Is All You Need" paper

*Lucas Beyer's Slides

Applications of Transformers

- Natural Language Processing



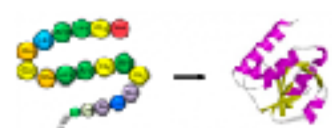
- Computer Vision

- Audio Analysis

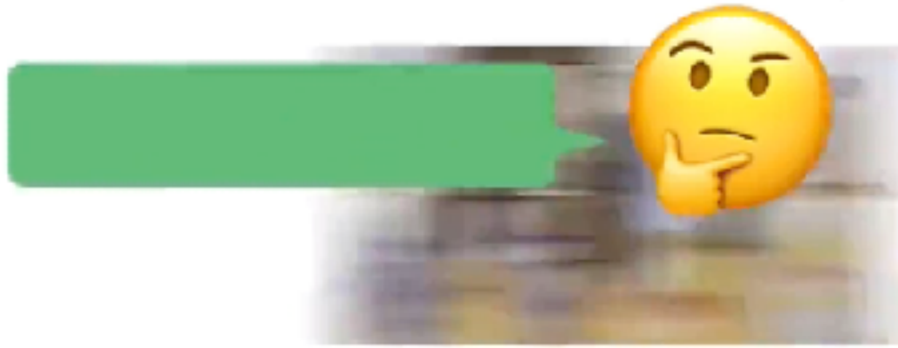
- Speech Processing

- Multimodal Understanding

- Many Others



Video Captioning



Text-to-Video Generation

Prompts used:

Lots of traffic in futuristic city. An alien spaceship arrives to the futuristic city. The camera gets inside the alien spaceship. The camera moves forward until showing an astronaut in the blue room. The astronaut is typing in the keyboard. The camera moves away from the astronaut. The astronaut leaves the keyboard and walks to the left. The astronaut leaves the keyboard and walks away. The camera moves beyond the astronaut and looks at the screen. The screen behind the astronaut displays fish swimming in the sea. Crash zoom into the blue fish. We follow the blue fish as it swims in the dark ocean. The camera points up to the sky through the water. The ocean and the coastline of a futuristic city. Crash zoom towards a futuristic skyscraper. The camera zooms into one of the many windows. We are in an office room with empty desks. A lion runs on top of the office desks. The camera zooms into the lion's face, inside the office. Zoom out to the lion wearing a dark suit in an office room. The lion wearing looks at the camera and smiles. The camera zooms out slowly to the skyscraper exterior. Timelapse of sunset in the modern city

Referred Video Object Segmentation

- Given a text query and a video, the proposed model segments object instances referred to in the textual query.

Botach et al. “End-to-End Referring Video Object Segmentation with Multimodal Transformers”, arXiv 2021

Plan for Today

- Motivation for the course
- Course overview
- Overview of self-attention & transformers

Course Objectives

- Provide a thorough overview of state-of-the-art in this emerging area of research.
- Learn how to analyze and present research papers.
- Discuss cutting edge research and speculate about future research directions.
- Carry out a semester-long project culminating in a paper of nearly publishable quality.

Prerequisites

- Understanding of fundamental machine learning / deep learning concepts.

- The ability to analyze research papers published in major machine learning and computer vision conferences.

- Check the papers listed on the course website, and see if you would be comfortable presenting them.

- **Undergraduates:** one of machine learning, deep learning or computer vision courses taken at UNC. Upload a brief statement in the Assignments section of Canvas stating which prerequisite you've met by **Sunday, January 14th, 11:59 pm**.

Grading

- Class Participation: 10%
- Written Paper Critiques: 20%
- Paper Presentations: 30%
- Course Project: 40%

Paper Presentations

- Everybody will give two types of presentations:
  1. One 30min or one 45min paper presentation (presented solo or in pairs).
  2. One 20min paper + discussion for a paper battle (presented in groups of ~3).

- Rehearse your talk to make sure it fits within the time limit listed in the Schedule next to each paper.

- If you need help understanding the paper, send me an email and we'll set up a meeting.

Paper Presentations

- Focus on the most important high-level concepts. No need to present every single technical detail / experiment.

- Your audience should understand:
    1. The research problem.
    2. The motivation of the proposed research.
    3. Any necessary background info.
    4. The main technical details.
    5. The key experimental results.

- Spend time on description of the experiments.

Written Paper Critiques

- The goal is to provide a critical analysis of the paper (positive or negative).
- You will need to submit paper critiques for 9 of my selected papers (each critique worth 2% of the total course grade).
- Each critique graded as Pass or Fail (no detailed feedback).
- Use the template here (also provided in Canvas).
- Please write your critiques independently.
- Upload the critiques in a PDF format on Canvas by 11:59 PM on the day before the class.

QA Prompts for a Paper Discussion

- With each paper critique, you will also include one paper discussion question and your answer to that question.
- We will use your submitted discussion questions for detailed ~30min paper discussions.
- Check out some general info here on how to come up with good questions for a discussion.
- I will read every single one of these so you should come up with meaningful questions.

Detailed Paper Discussions

- For 9 of the selected papers, we will have detailed ~30min paper discussions.
- I will use your submitted QA discussion prompts (from your paper critiques) to compile a set of 6-8 discussion questions.
- We will then break out into small groups where each group will discuss one of the questions among themselves.
- Afterward, we will reconvene to discuss all of the questions together.

Paper Battles

- For 6 paper pairs, we will have paper battles, i.e., detailed head-to-head paper comparisons between two groups of students.
- Assume that the two given papers represent two conference paper submissions.
- However, only one of those papers can be accepted.
- Two groups of students will try to convince the audience that their presented paper should be “accepted”.

Paper Battles (Continued)

- Each group will give a brief 20min overview of their assigned paper (time limit will be strictly enforced).
- Following both presentations, each group will present slides with 3 main reasons why their presented paper is better.
- Afterward, we will have a brief discussion allowing each group to rebut another group's points.
- Lastly, the students in the class who were not presenting will vote on which paper is better (and provide a justification for their vote).

Course Projects

- There will be two tracks for course projects: (1) graduate and (2) undergraduate.
- Projects should be completed in groups of 2-4.
- If you want to pursue a project individually (e.g., for your dissertation, etc.), please talk to me before doing so.
- If you do not have access to GPUs, send me an email or talk to me after class.
- Start thinking early!

Graduate Track

- Aimed at graduate students pursuing research requiring visual analysis.
- You can propose any project involving visual transformers in any area of interest to you.
- Review the paper list for inspiration, or come talk to me for topic related suggestions.
- Undergraduate students interested in visual transformers research are welcome to pursue this track but should talk to me before doing so.

Undergraduate Track

- Aimed at undergraduate students who want to gain more experience with existing visual transformer tools.
- The project will require students applying existing transformer models to 5 of their selected CV applications.
- For each application, the students will need to identify 3 failure and 3 success instances and document their experiences/insights about the model’s shortcomings, etc.
- Graduate students are also welcome to pursue this track if it fits them better.

Undergraduate Track

Potentially useful pointers of existing tools/demos:

- Image Classification
- Image Captioning and Question Answering
- Video Captioning / Question Answering
- Text-to-Motion Generation
- Object Detection and Segmentation
- Audiovisual Video Question Answering

Project Submissions

- Proposal (10% of the total grade):
  - Presentations on **02/19/24** & **02/21/24**.
  - Write-up due **02/25/24.**
  - Overview of your project plan.

- Milestone (10% of the total grade):
  - Presentations on **03/25/24** & **03/27/24**.
  - Write-up due **03/31/24.**
  - A checkpoint to make sure you are making progress.

- Final (20% of the total grade):
  - Presentations on **04/22/24** & **04/24/24**.
  - Write-up due **04/29/24.**
  - Final findings of the project.

- Use templates here for your project write-up (also available on Canvas).
- Upload your slides & write-ups on Canvas in the Assignment section.

Zoom Attendance

- In case you cannot attend in person, you can join remotely via the following Zoom link (passcode: vit24s)
- I expect most of you to join in person as it will facilitate more effective class discussions.

Office Hours

- Office hours are available by appointment.
- Please sign up for a meeting at https://calendly.com/gedasb
- If none of the time slots work, send me an email to schedule an appointment separately.

Canvas

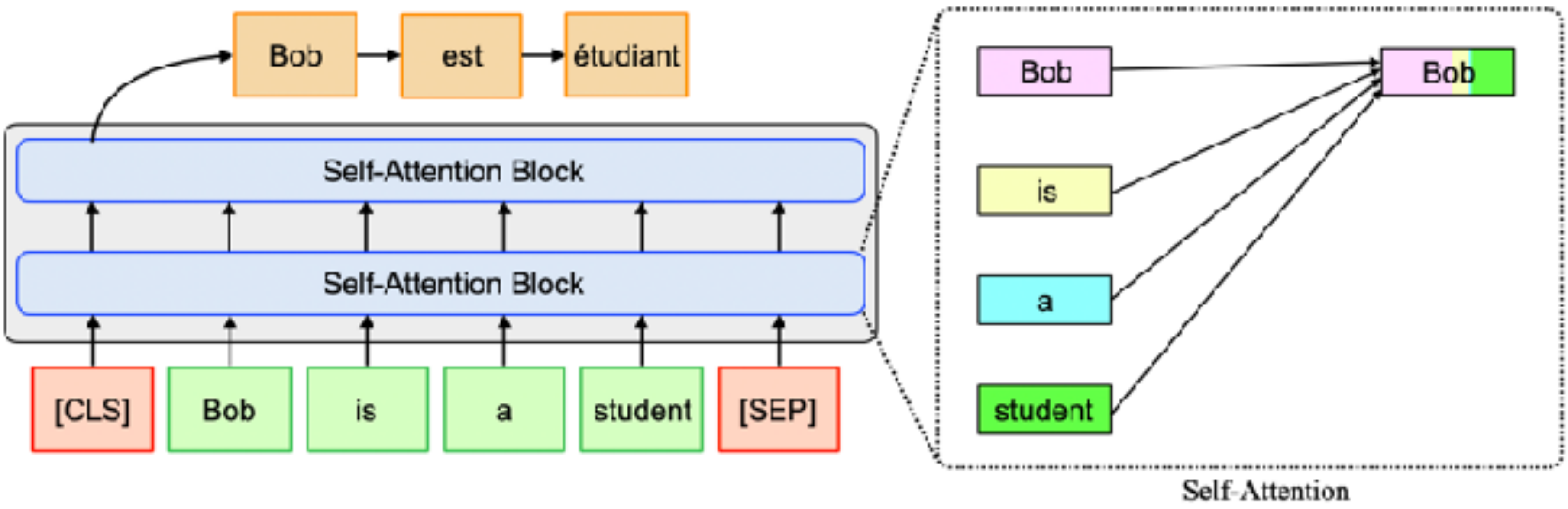
- We will use Canvas for many course-related activities.
- All the announcements will be made on Canvas so please check it regularly.
- You will need to upload your assignments on Canvas.
- The discussion and collaboration pages are enabled on Canvas. Please share any interesting papers, blog posts, or general ideas in the discussion page.
- You can find collaborators for project on Canvas as well.

Plan for Today

- Motivation for the course
- Course overview
- Overview of self-attention & transformers

Self-Attention

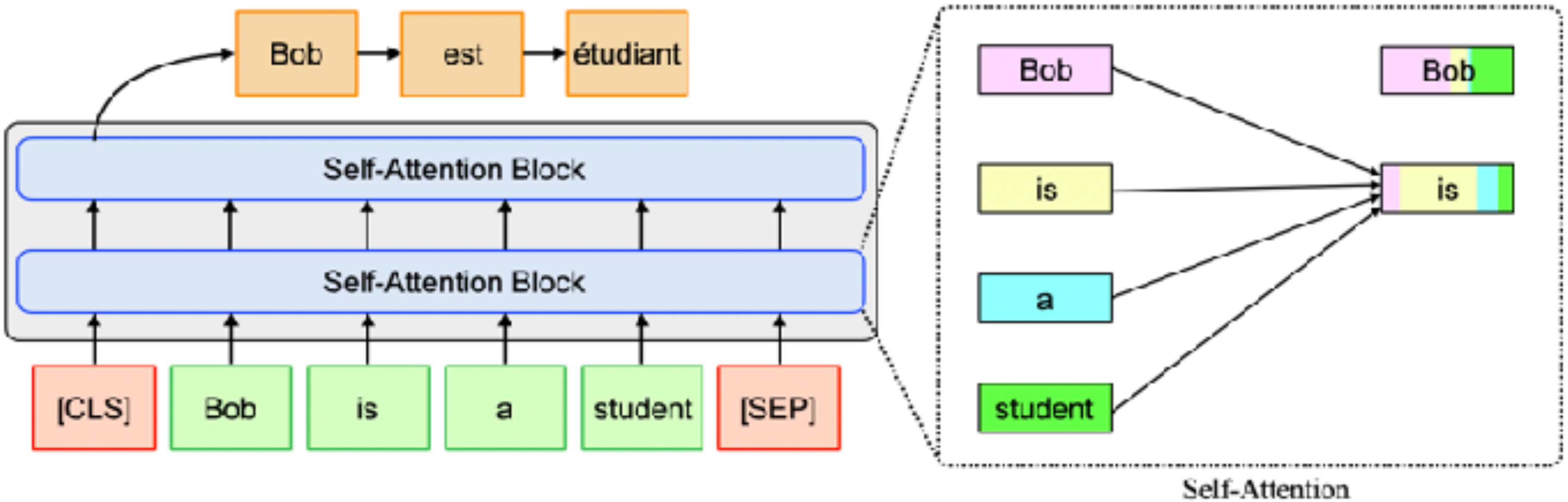
- Self-attention enables capturing long-range dependencies among words.



Vaswani et al., ”Attention is All You Need”, NIPS 2017

Self-Attention

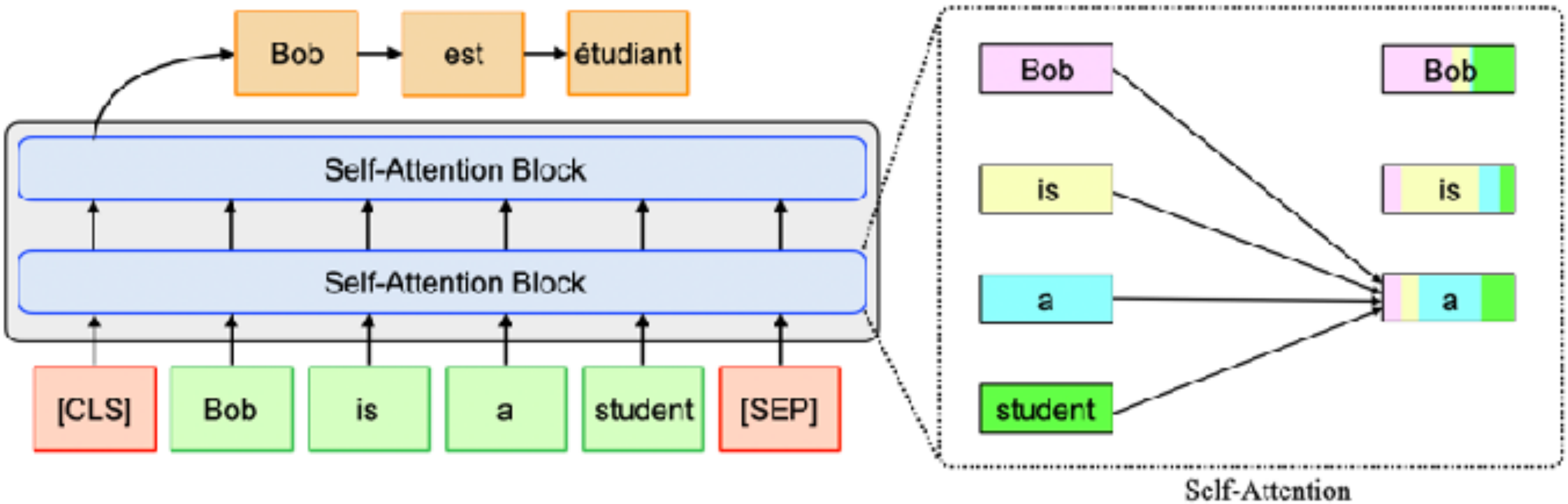
- Self-attention enables capturing long-range dependencies among words.



Vaswani et al., "Attention is All You Need", NIPS 2017

Self-Attention

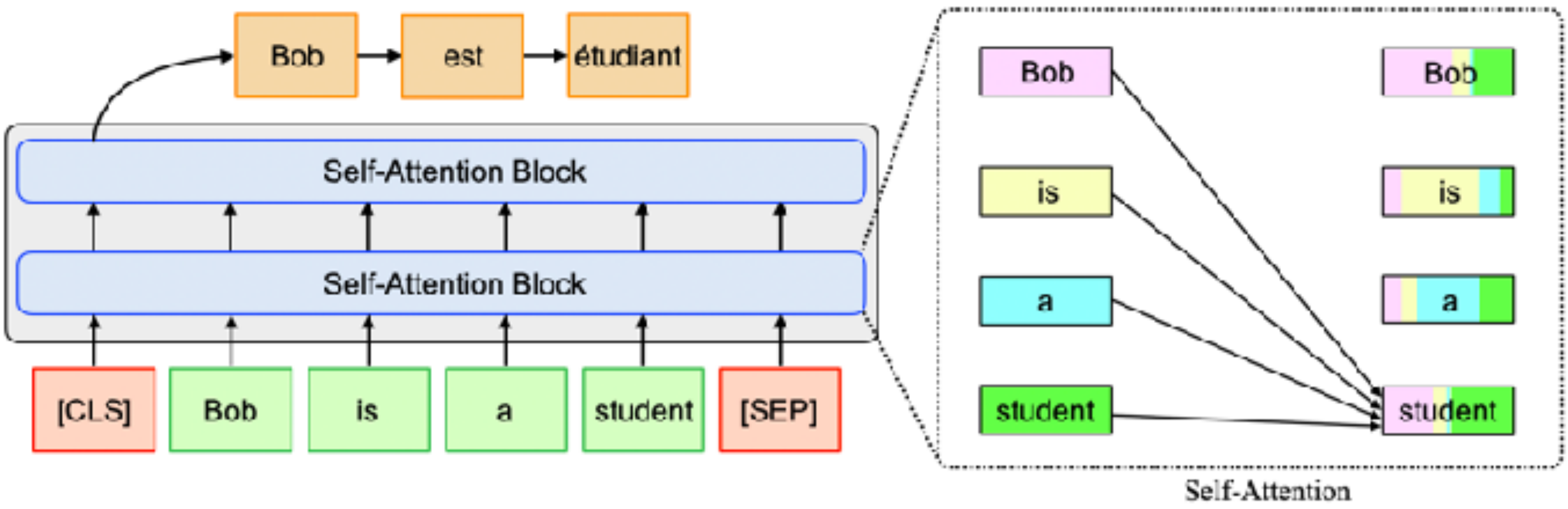
- Self-attention enables capturing long-range dependencies among words.



Vaswani et al., "Attention is All You Need", NIPS 2017

Self-Attention

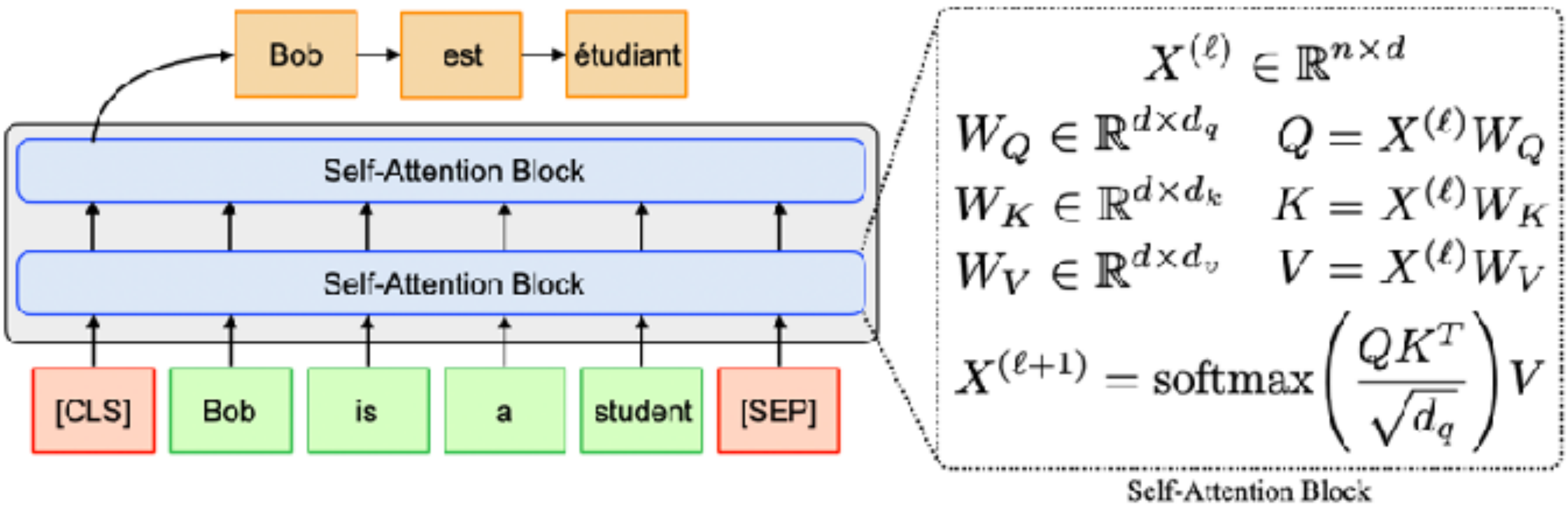
- Self-attention enables capturing long-range dependencies among words.



Vaswani et al., "Attention is All You Need", NIPS 2017

Self-Attention

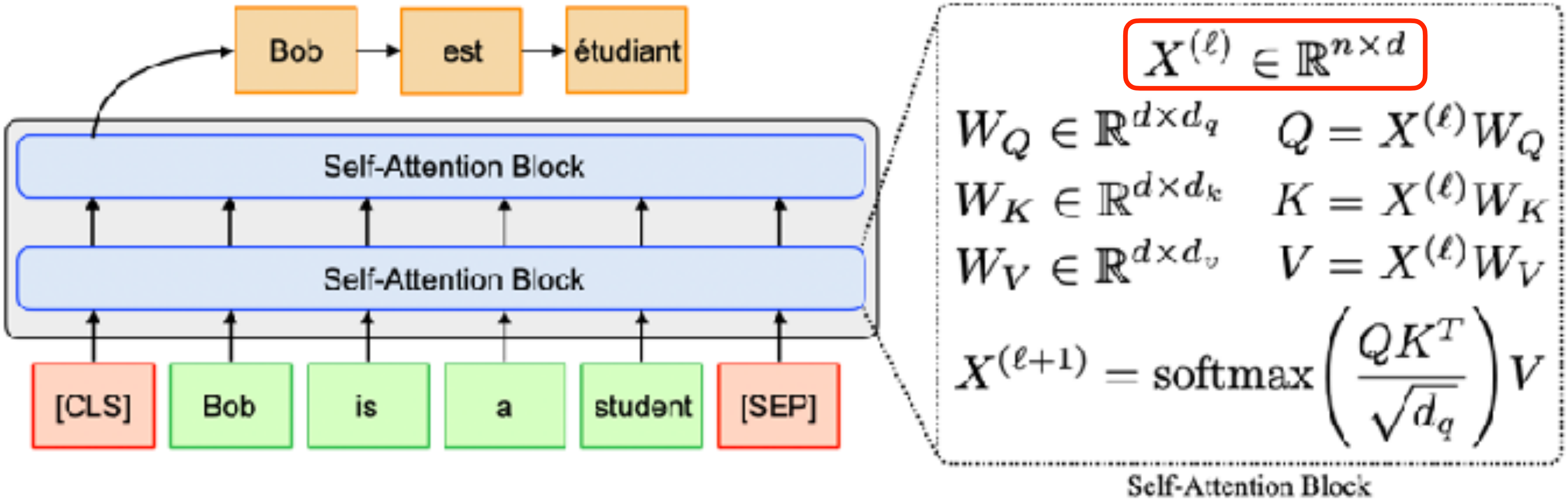
- Self-attention enables capturing long-range dependencies among words.



Vaswani et al., "Attention is All You Need", NIPS 2017

Self-Attention

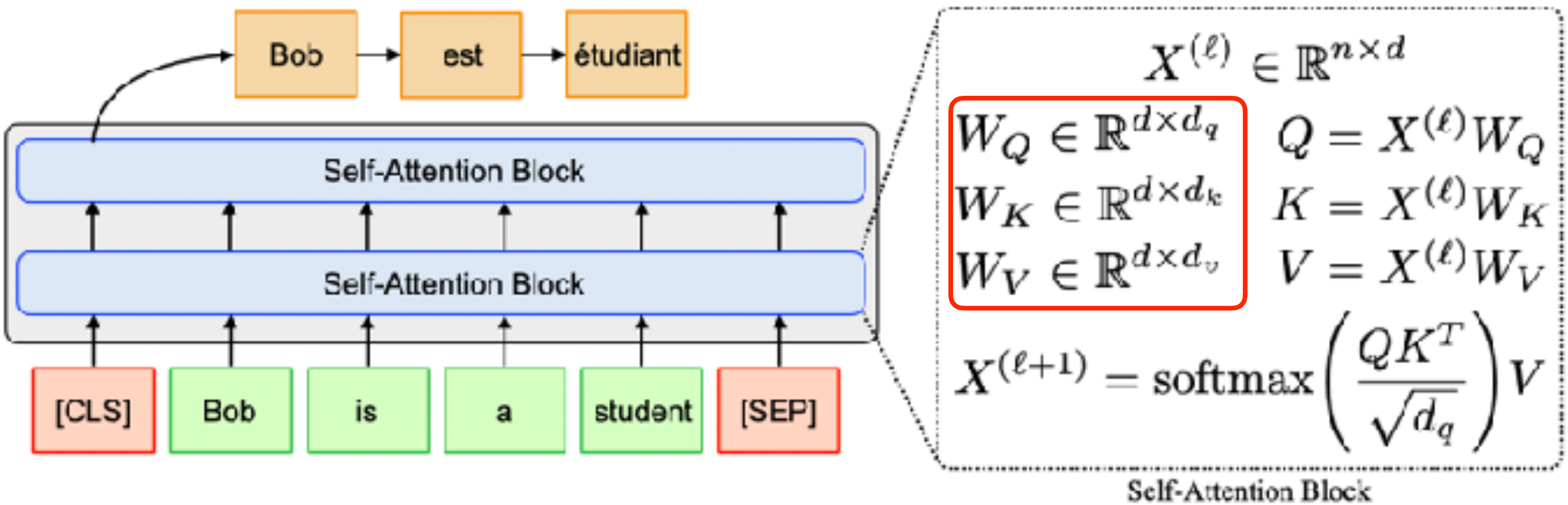
- Self-attention enables capturing long-range dependencies among words.



Vaswani et al., "Attention is All You Need", NIPS 2017

Self-Attention

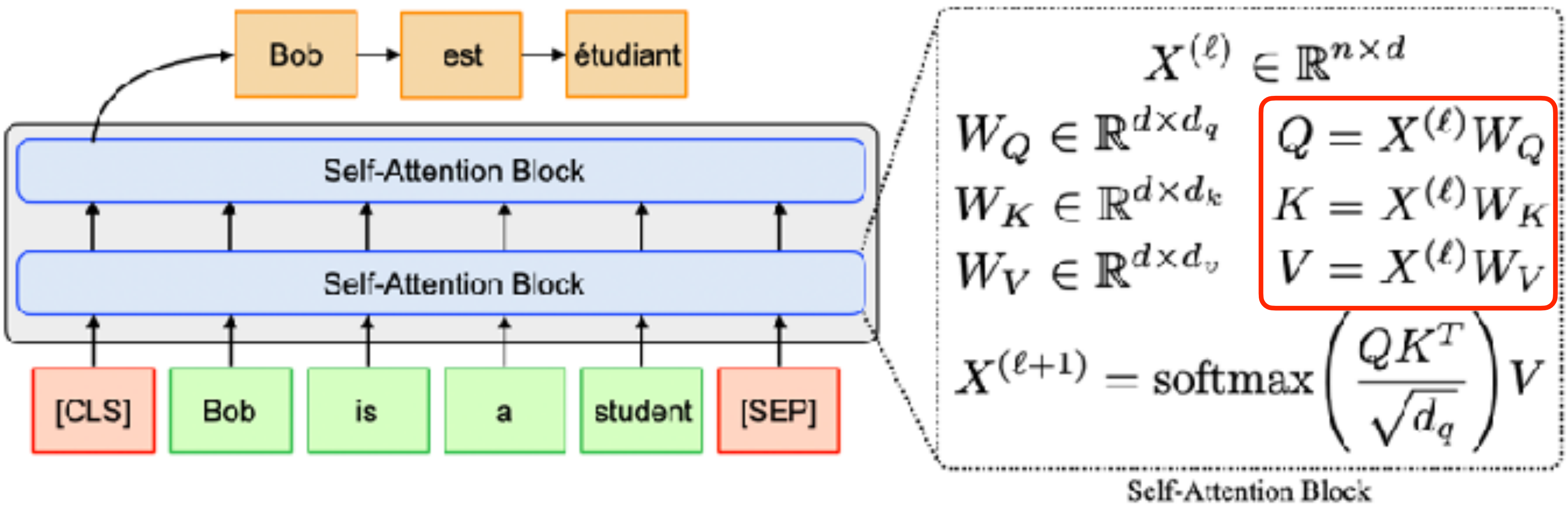
- Self-attention enables capturing long-range dependencies among words.



Vaswani et al., "Attention is All You Need", NIPS 2017

Self-Attention

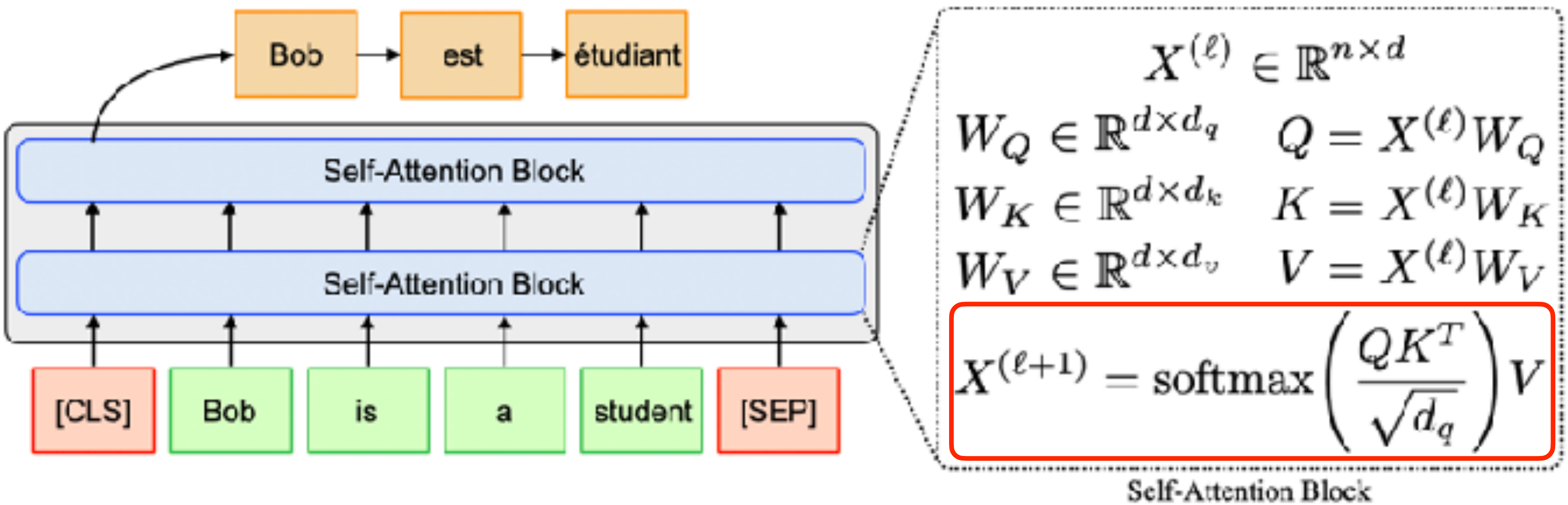
- Self-attention enables capturing long-range dependencies among words.



Vaswani et al., "Attention is All You Need", NIPS 2017

Self-Attention

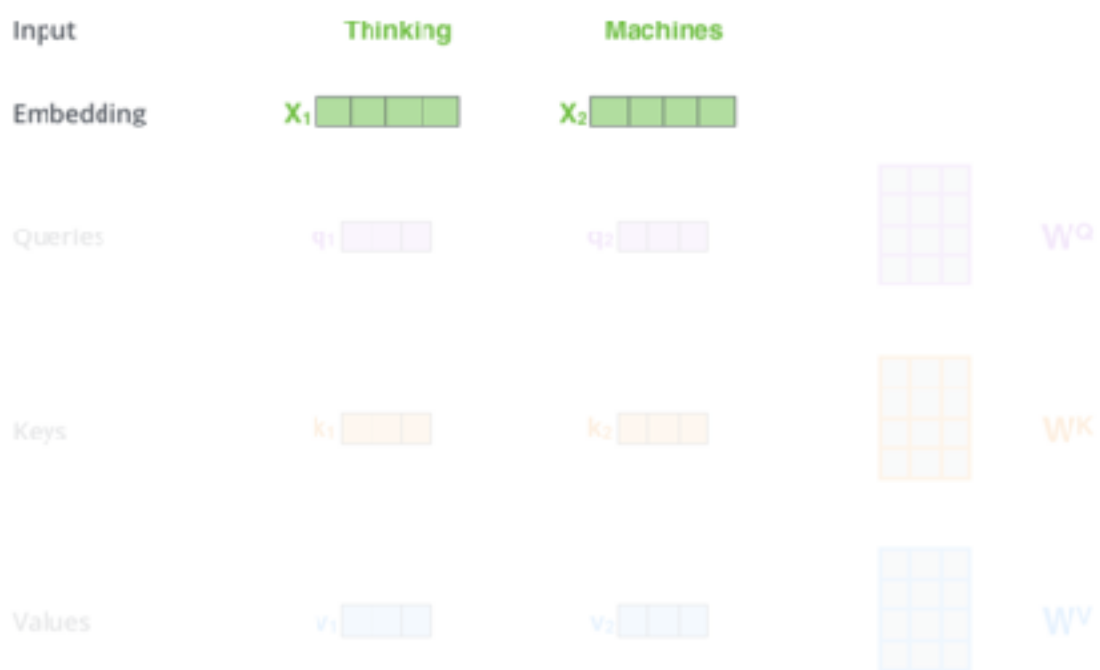
- Self-attention enables capturing long-range dependencies among words.



Vaswani et al., "Attention is All You Need", NIPS 2017

Self-Attention

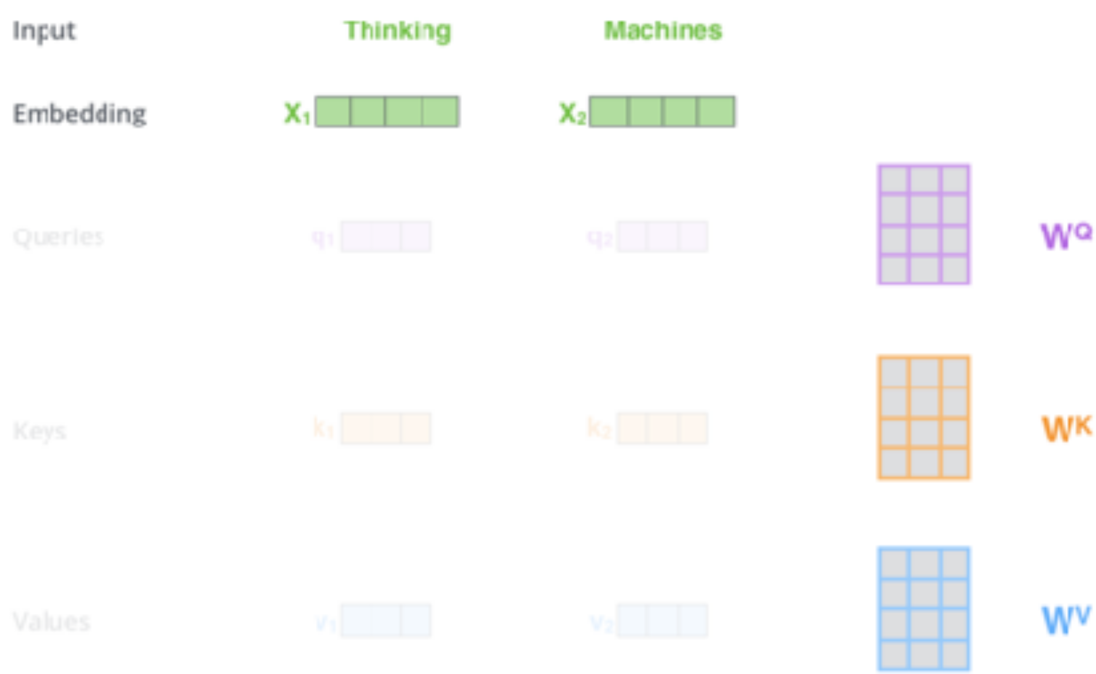
- Multiplying the embedding x_1 with query, key and value weight matrices (i.e., W^Q, W^K, W^V) produces the query, key and value vectors (i.e., q_1, k_1, v_1) associated with that word.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Self-Attention

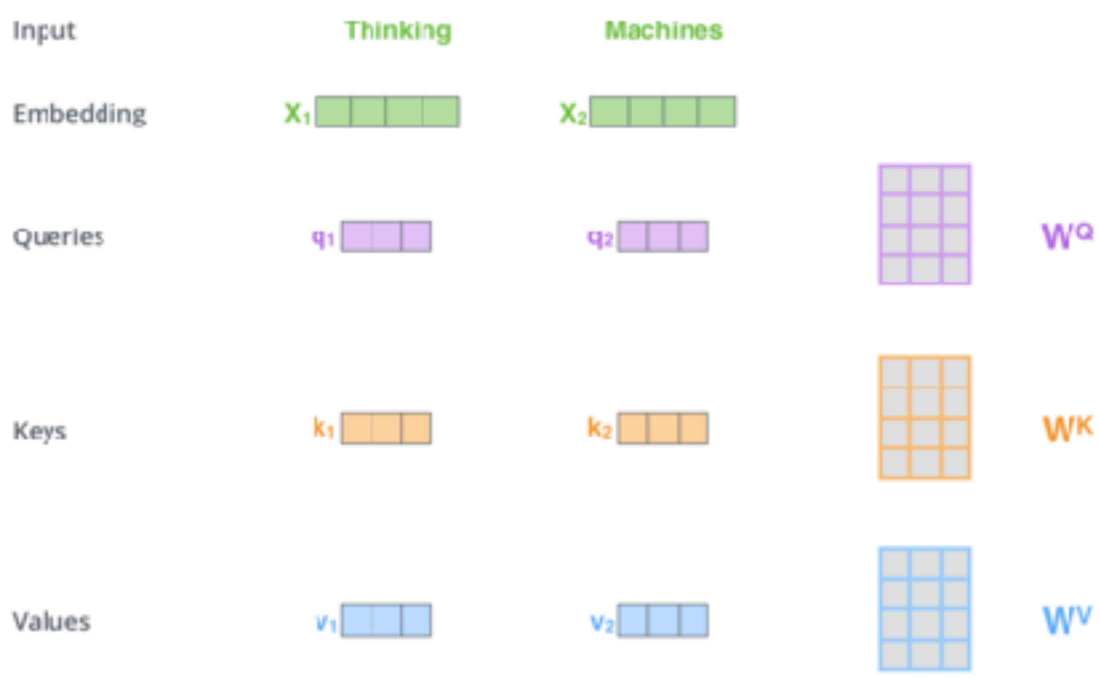
- Multiplying the embedding x_1 with query, key and value weight matrices (i.e., W^Q, W^K, W^V) produces the query, key and value vectors (i.e., q_1, k_1, v_1) associated with that word.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Self-Attention

- Multiplying the embedding x_1 with query, key and value weight matrices (i.e., W^Q, W^K, W^V) produces the query, key and value vectors (i.e., q_1, k_1, v_1) associated with that word.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

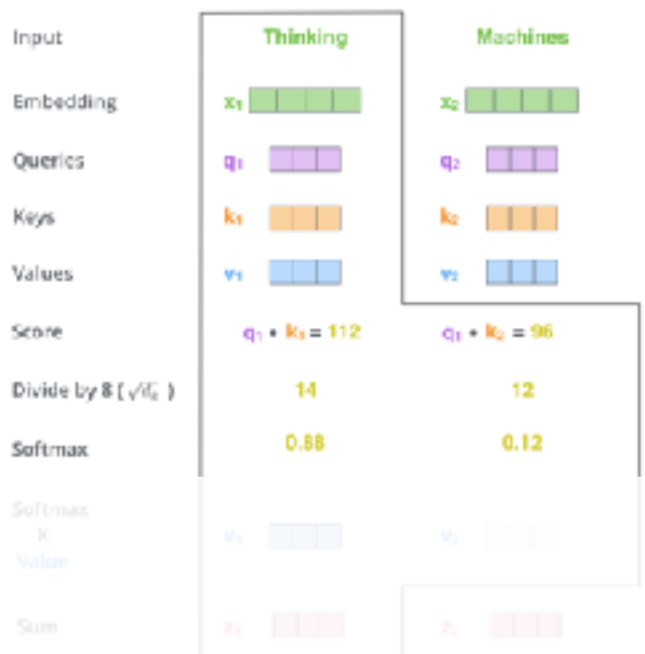
Self-Attention

- Next, we compute a pairwise similarity score for each pair of tokens in the input.
- The similarity scores are used to aggregate contextual information from the other words in a given sequence.

*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Self-Attention

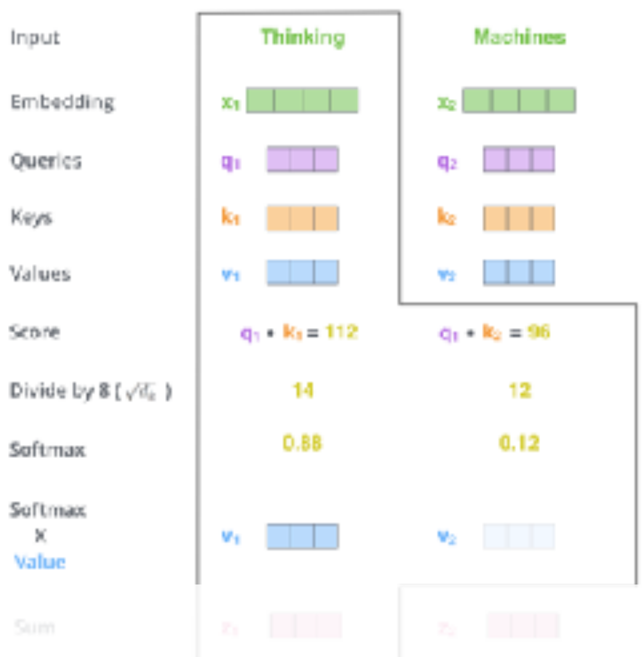
- Next, we compute a pairwise similarity score for each pair of tokens in the input.
- The similarity scores are used to aggregate contextual information from the other words in a given sequence.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Self-Attention

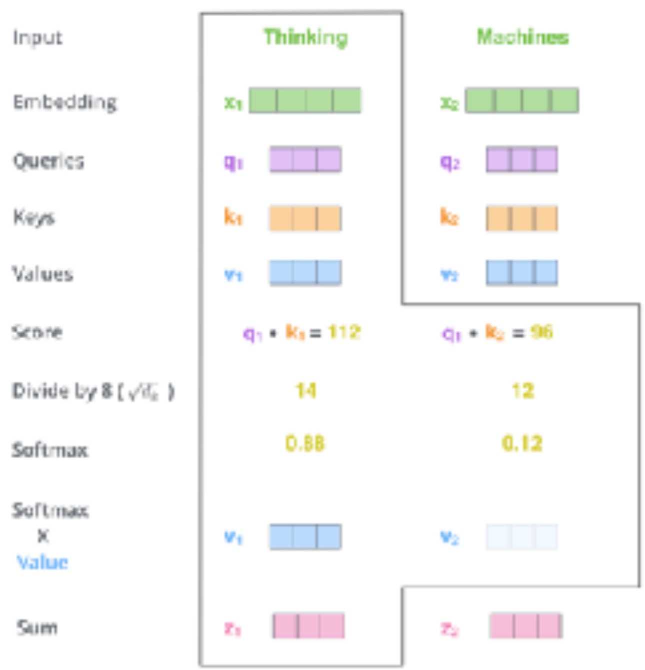
- Next, we compute a pairwise similarity score for each pair of tokens in the input.
- The similarity scores are used to aggregate contextual information from the other words in a given sequence.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Self-Attention

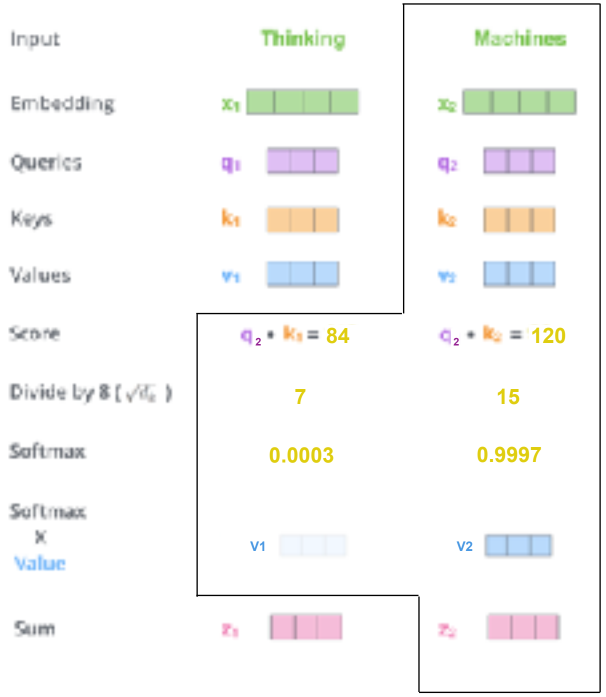
- Next, we compute a pairwise similarity score for each pair of tokens in the input.
- The similarity scores are used to aggregate contextual information from the other words in a given sequence.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Self-Attention

- Next, we compute a pairwise similarity score for each pair of tokens in the input.
- The similarity scores are used to aggregate contextual information from the other words in a given sequence.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Matrix Calculation of Self-Attention

- Every row in the X matrix corresponds to a word in the input sentence.
- The updated representation Z is computed using a scaled dot product attention.

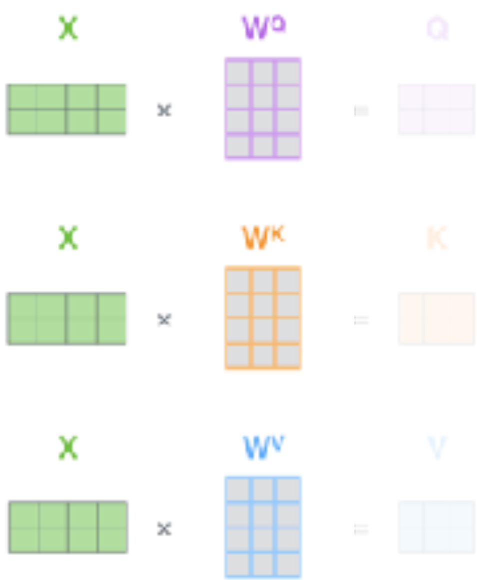
b) Scaled Dot-Product Attention

a) Computing Query, Key & Value Matrices

*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Matrix Calculation of Self-Attention

- Every row in the X matrix corresponds to a word in the input sentence.
- The updated representation Z is computed using a scaled dot product attention.



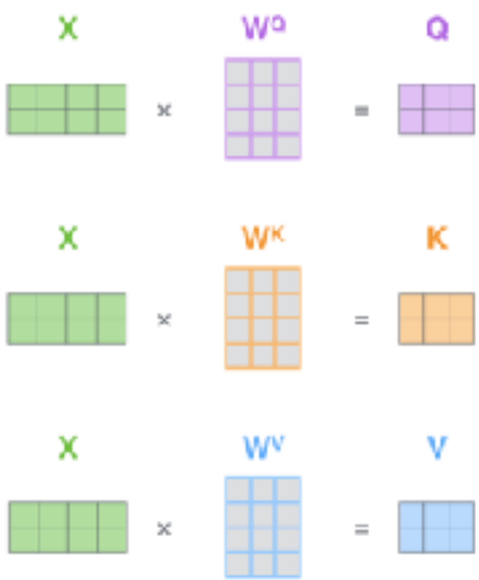
b) Scaled Dot-Product Attention

a) Computing Query, Key & Value Matrices

*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Matrix Calculation of Self-Attention

- Every row in the X matrix corresponds to a word in the input sentence.
- The updated representation Z is computed using a scaled dot product attention.



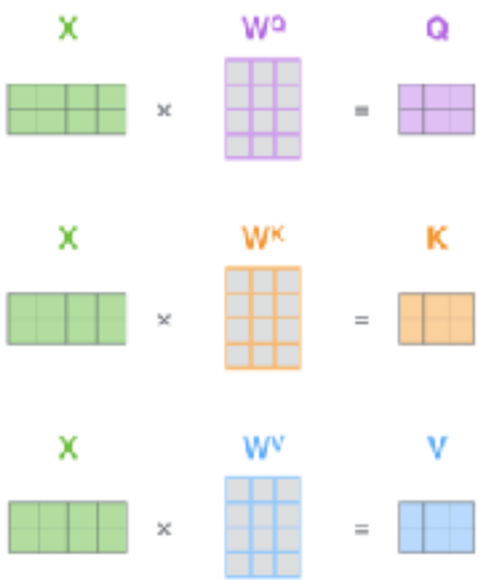
a) Computing Query, Key & Value Matrices

b) Scaled Dot-Product Attention

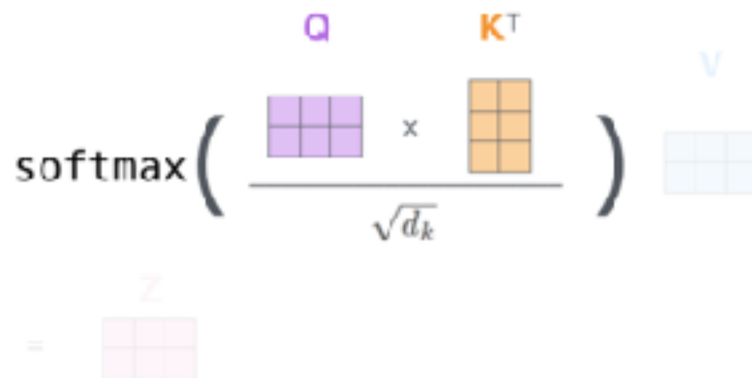
*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Matrix Calculation of Self-Attention

- Every row in the X matrix corresponds to a word in the input sentence.
- The updated representation Z is computed using a scaled dot product attention.



a) Computing Query, Key & Value Matrices

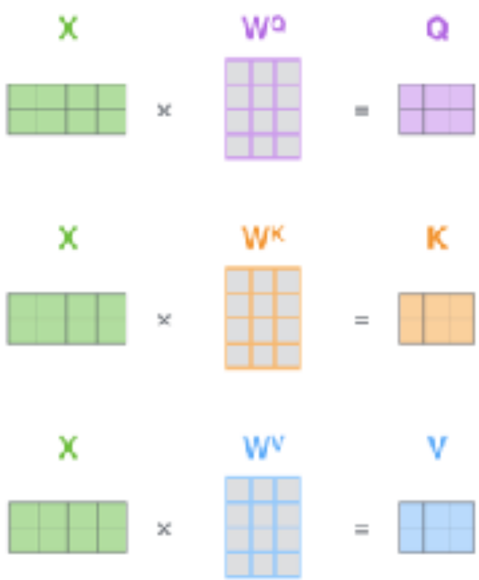


b) Scaled Dot-Product Attention

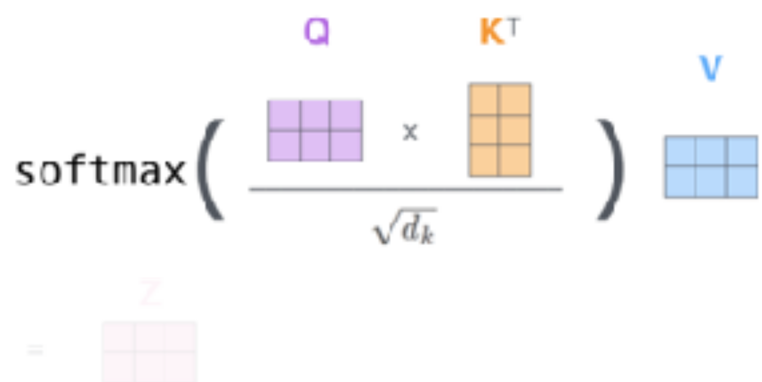
*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Matrix Calculation of Self-Attention

- Every row in the X matrix corresponds to a word in the input sentence.
- The updated representation Z is computed using a scaled dot product attention.



a) Computing Query, Key & Value Matrices

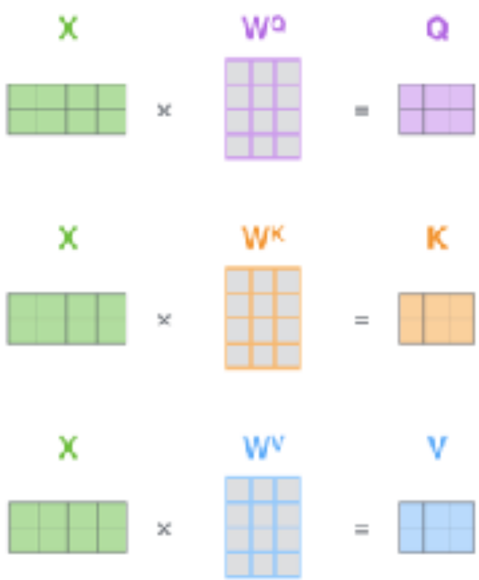


b) Scaled Dot-Product Attention

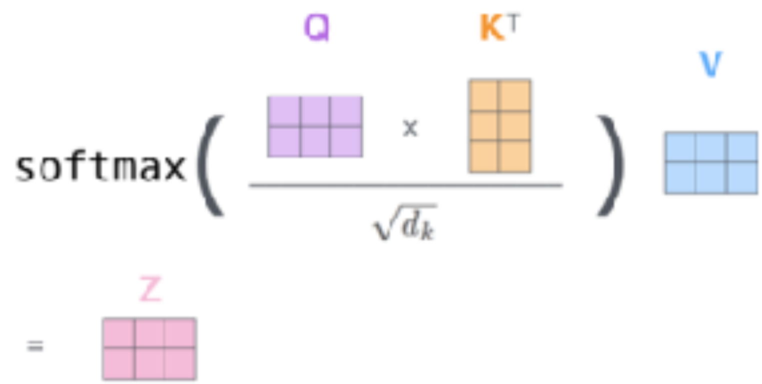
*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Matrix Calculation of Self-Attention

- Every row in the X matrix corresponds to a word in the input sentence.
- The updated representation Z is computed using a scaled dot product attention.



a) Computing Query, Key & Value Matrices

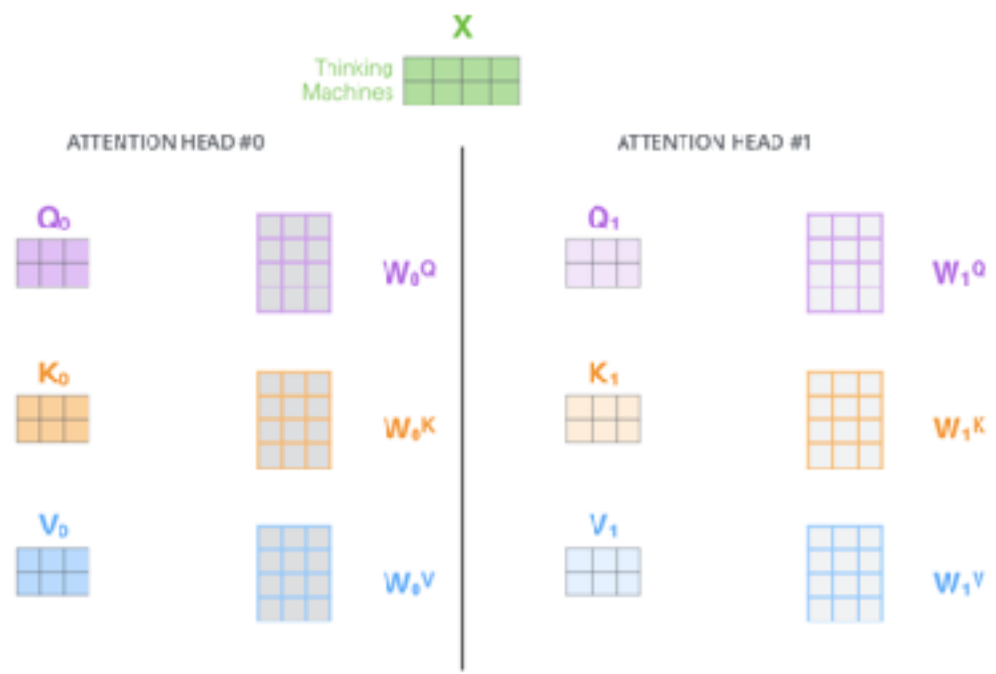


b) Scaled Dot-Product Attention

*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention

- Multi-head attention enables to learn multiple “representation subspaces”, improving the expressivity of the model.
- Computational cost is similar to standard self-attention.

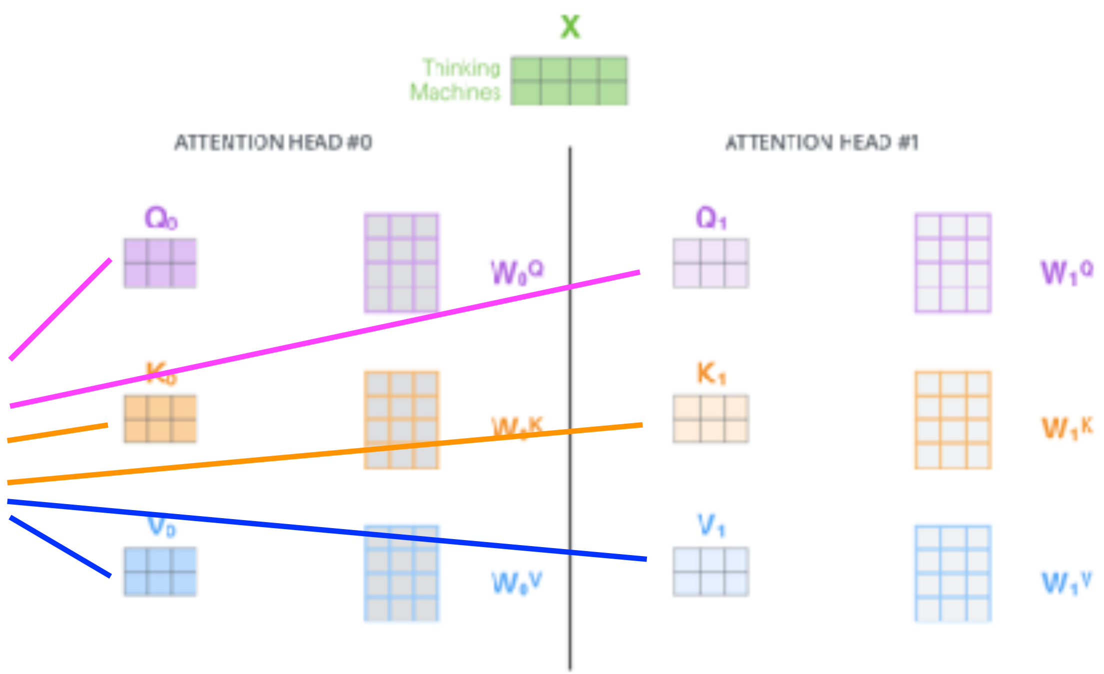


*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention

- Multi-head attention enables to learn multiple “representation subspaces”, improving the expressivity of the model.
- Computational cost is similar to standard self-attention.

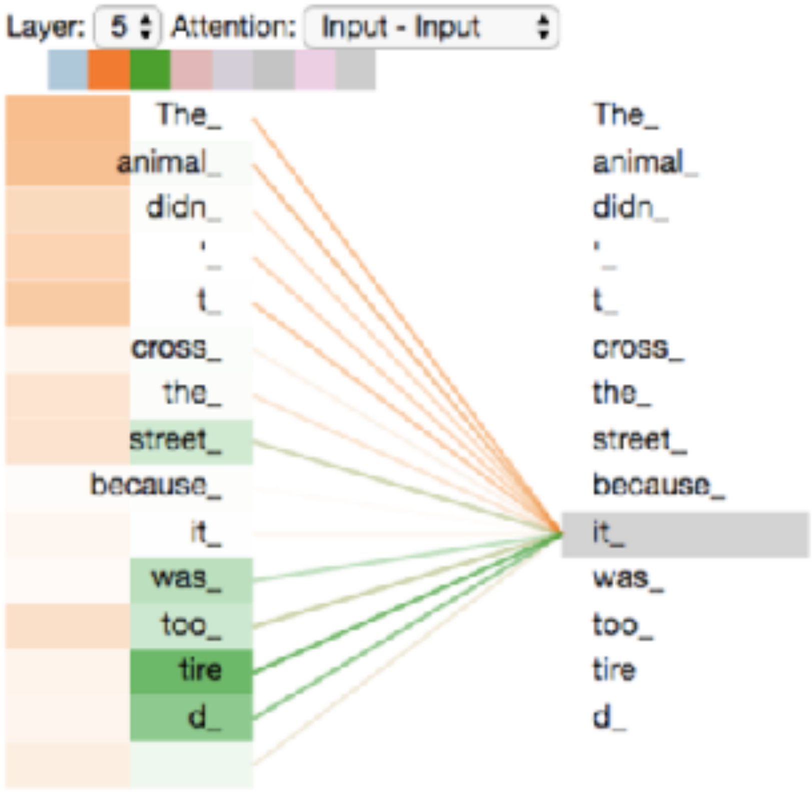
Reduced dimensionality for each head



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention

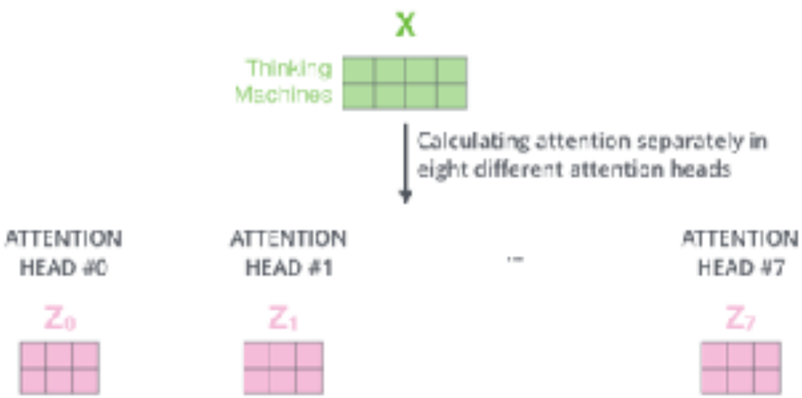
- Different attention heads learn to focus on different contextual information.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention

- Multi-head attention enables to learn multiple “representation subspaces”, improving the expressivity of the model.
- Computational cost is similar to standard self-attention.



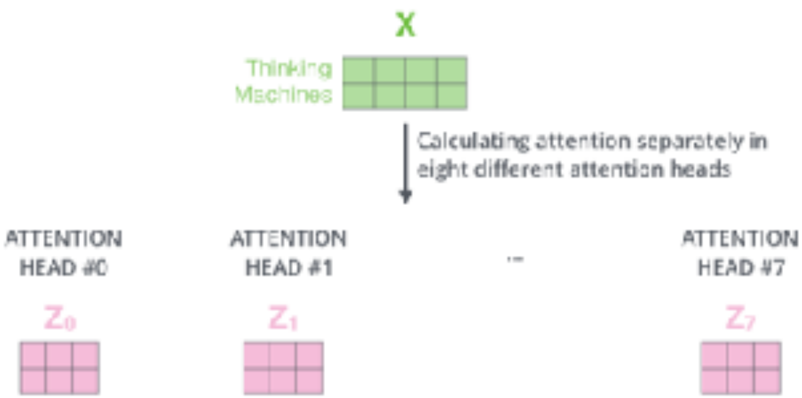
a) Calculating attention separately for each head

b) Concatenating the attention heads and multiplying by another weight matrix W^O

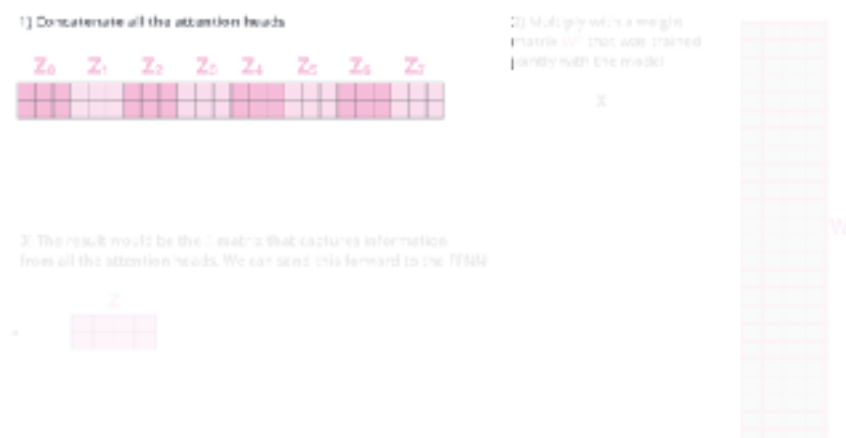
*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention

- Multi-head attention enables to learn multiple “representation subspaces”, improving the expressivity of the model.
- Computational cost is similar to standard self-attention.



a) Calculating attention separately for each head

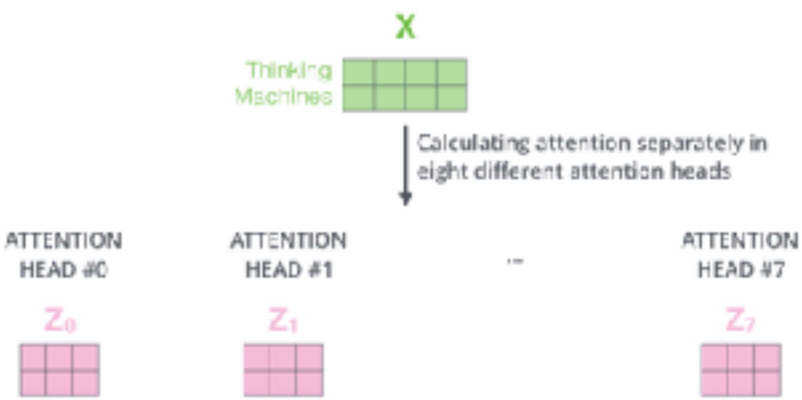


b) Concatenating the attention heads and multiplying by another weight matrix W^O

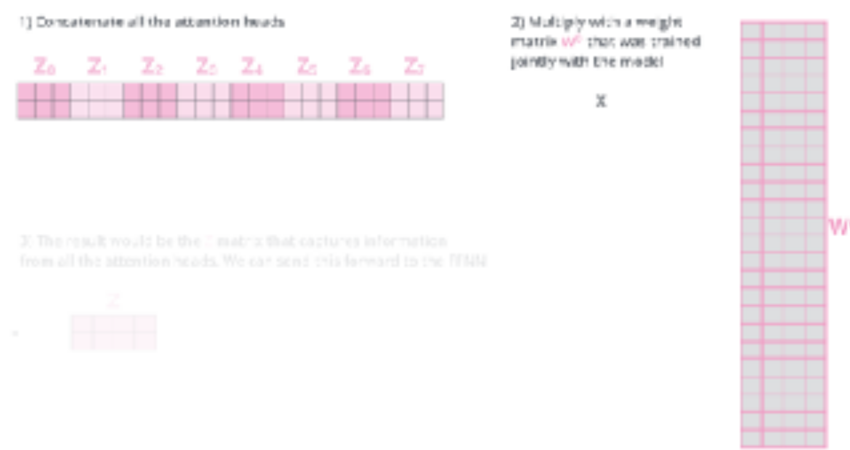
*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention

- Multi-head attention enables to learn multiple “representation subspaces”, improving the expressivity of the model.
- Computational cost is similar to standard self-attention.



a) Calculating attention separately for each head

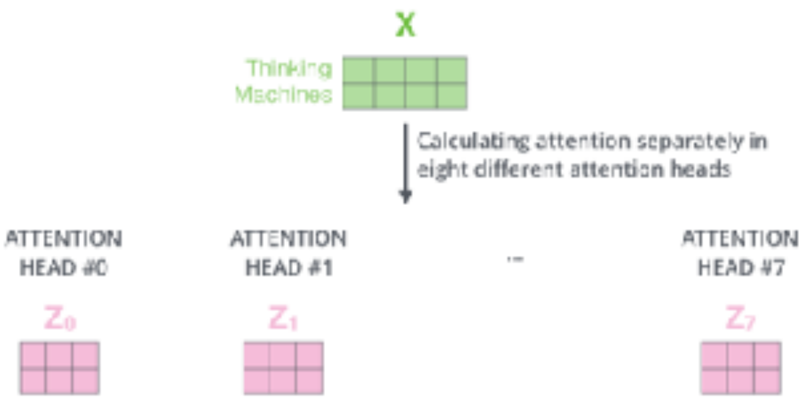


b) Concatenating the attention heads and multiplying by another weight matrix W^O

*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention

- Multi-head attention enables to learn multiple “representation subspaces”, improving the expressivity of the model.
- Computational cost is similar to standard self-attention.

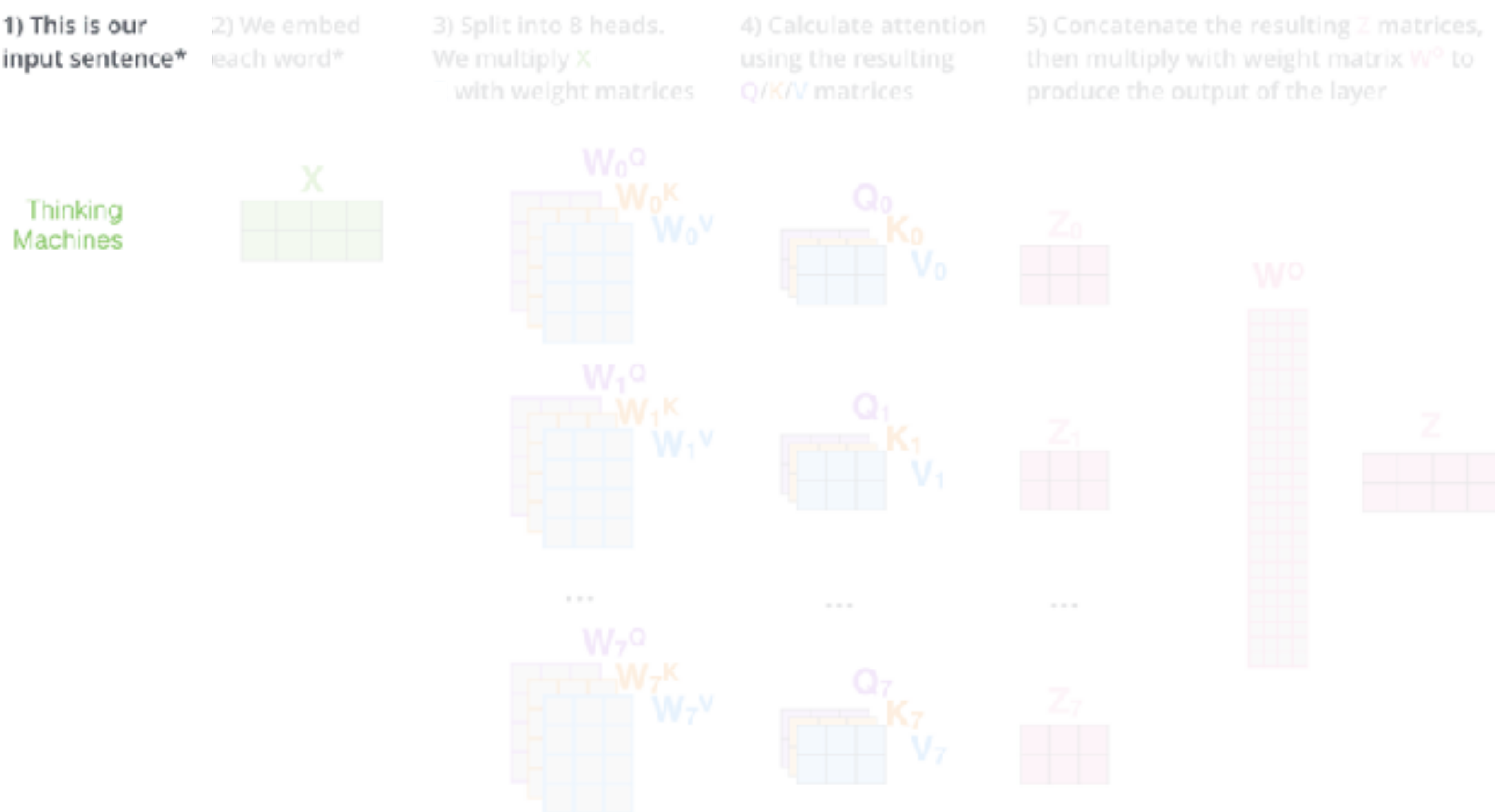


a) Calculating attention separately for each head

b) Concatenating the attention heads and multiplying by another weight matrix W^O

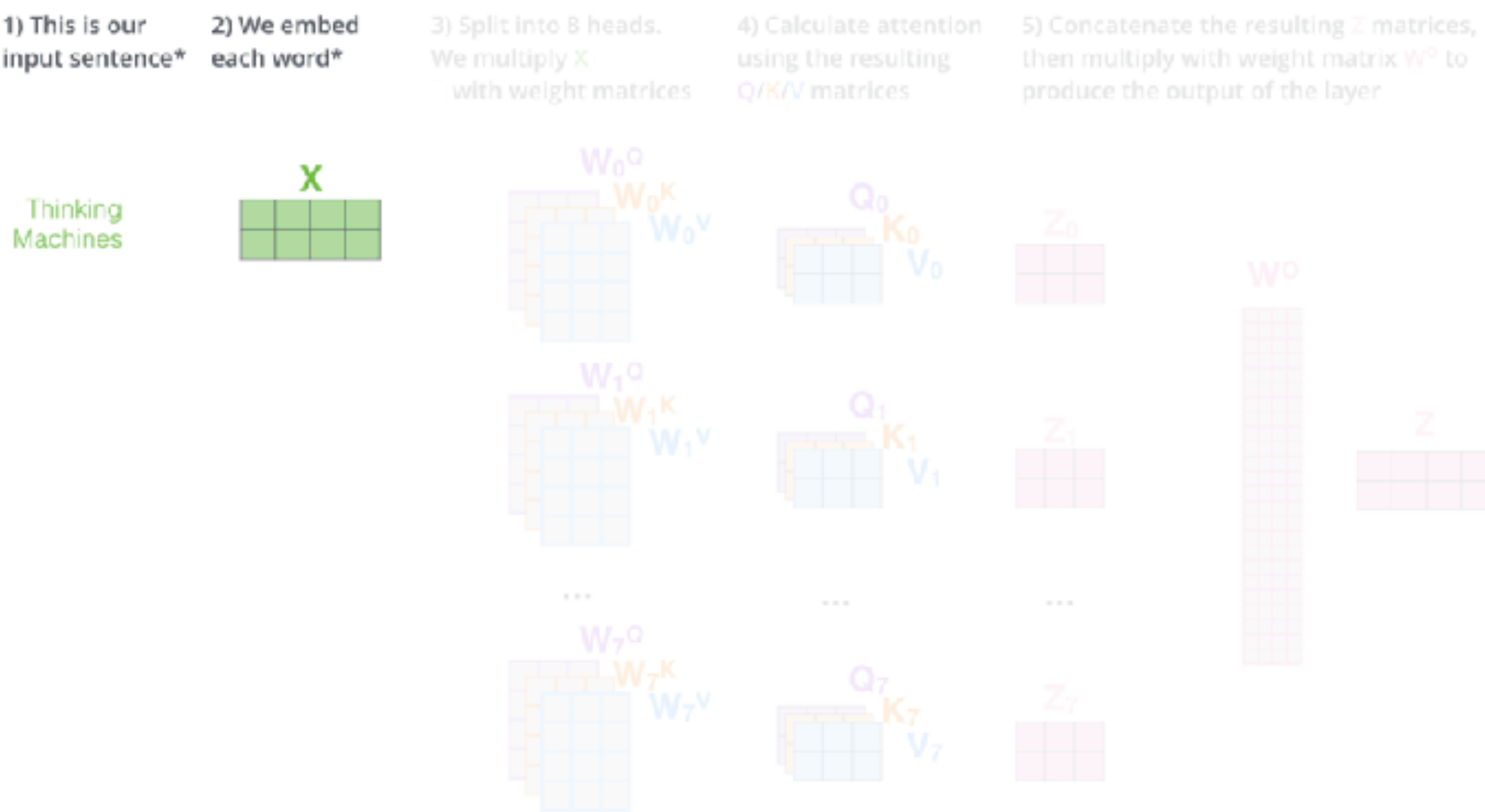
*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention



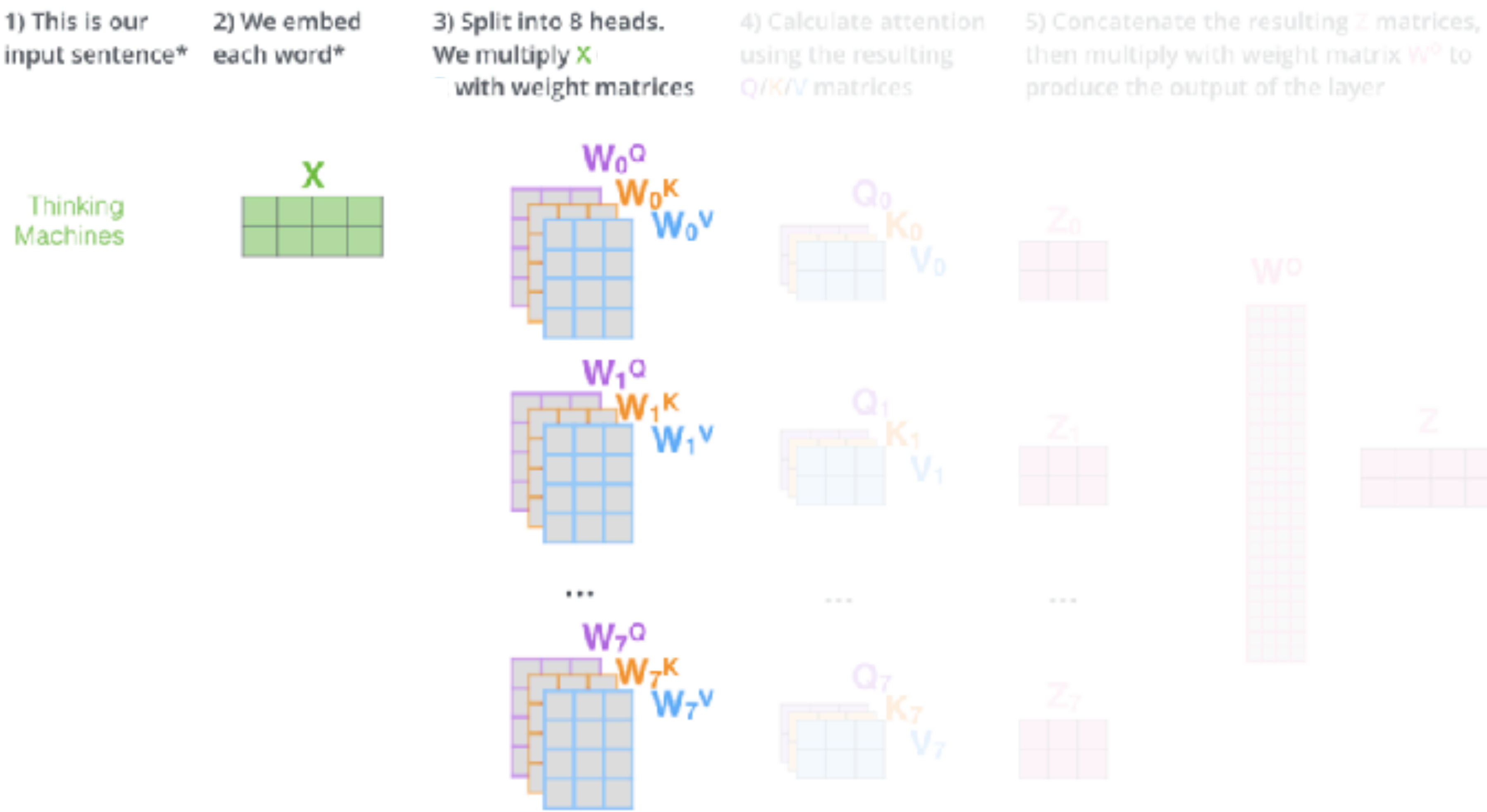
*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention



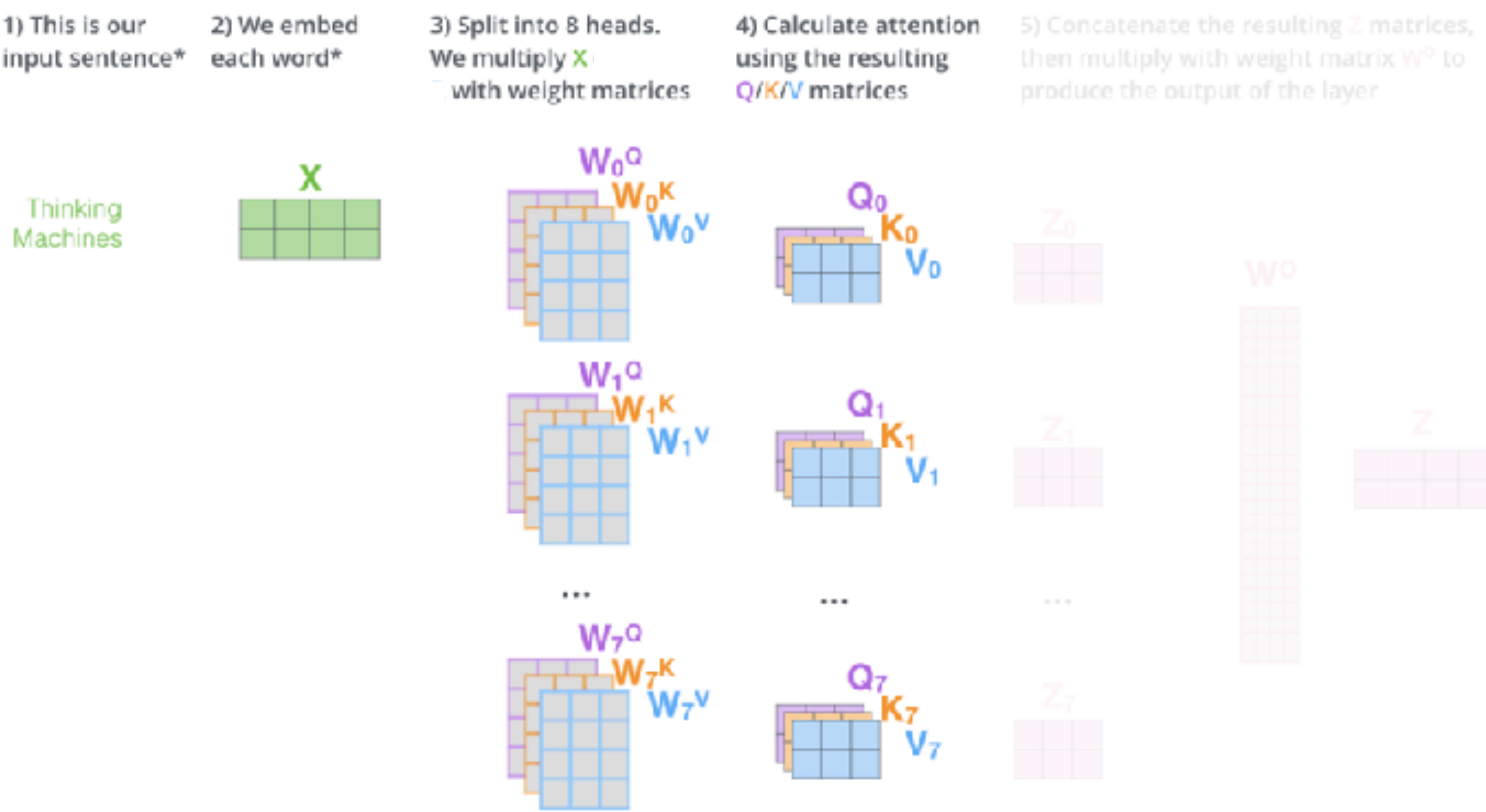
*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention



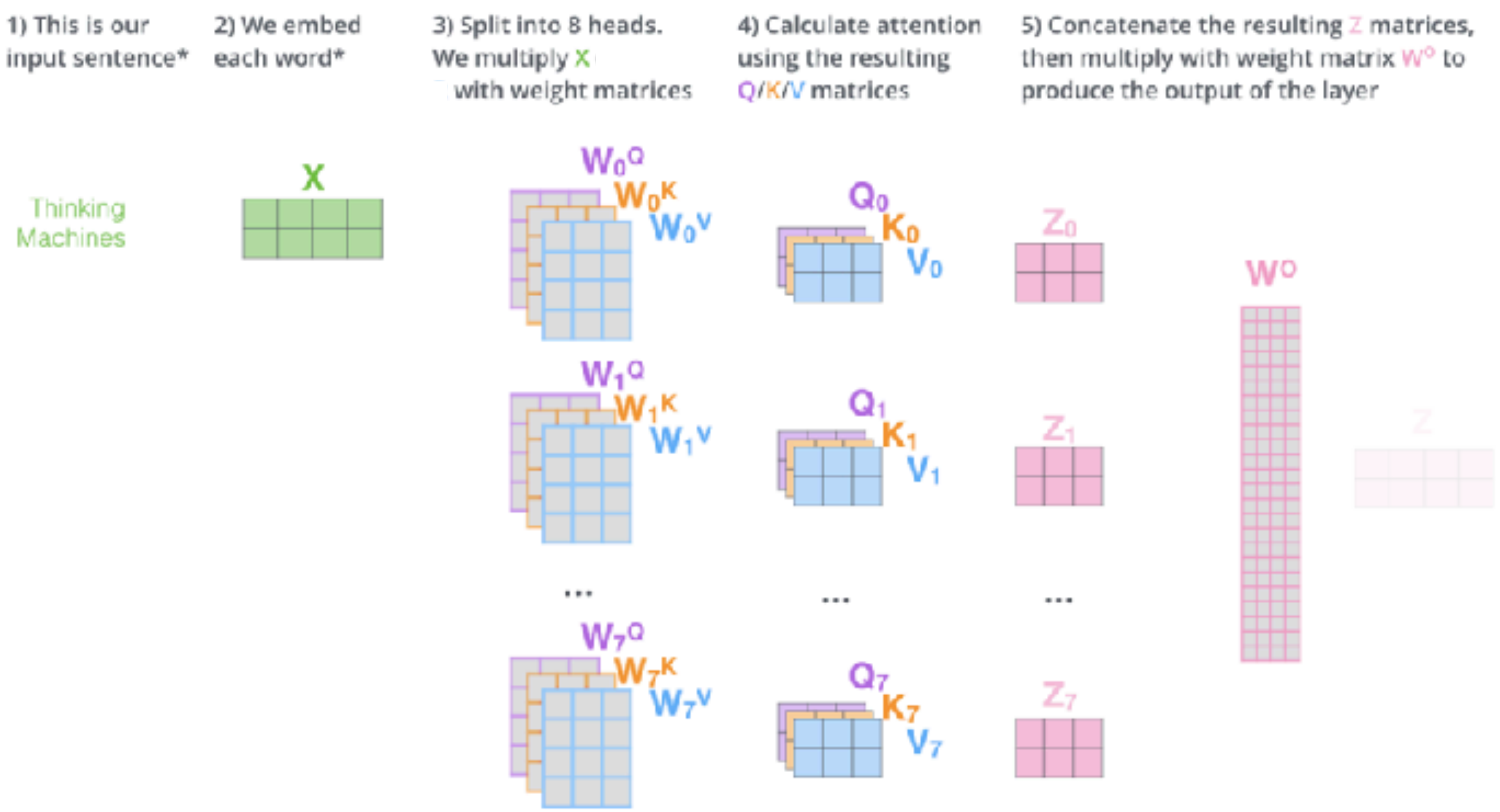
*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention



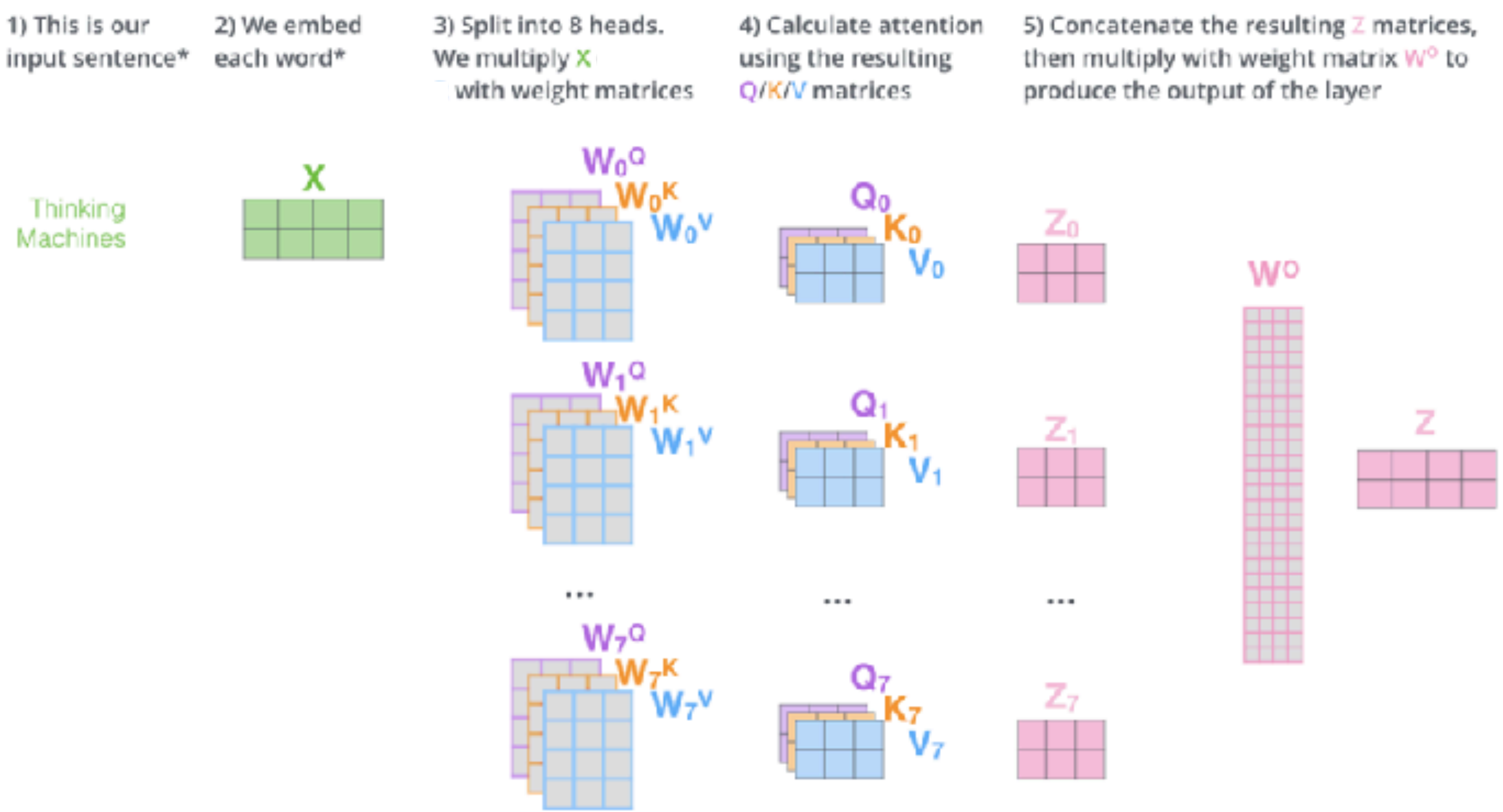
*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

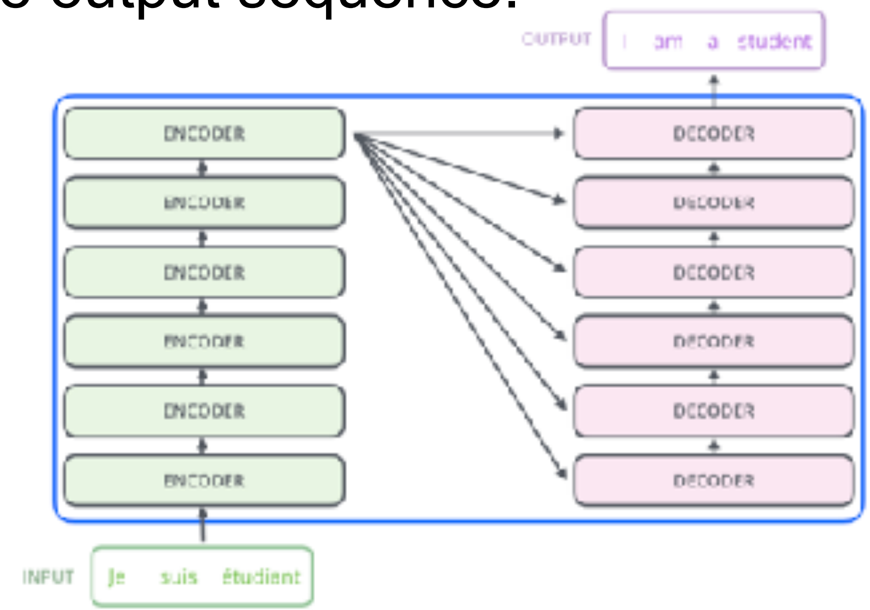
Multi-Head Self-Attention



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

The Transformer Model

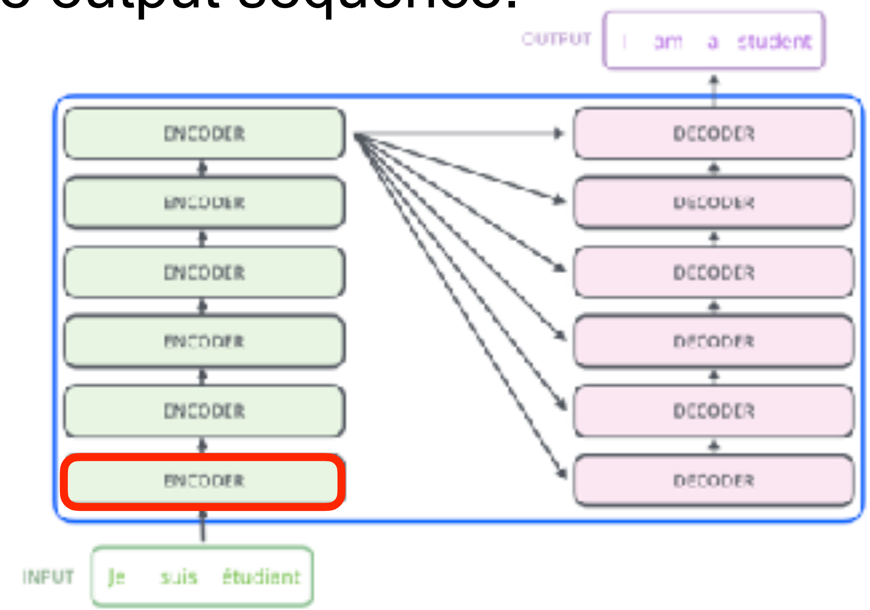
- Built using either an encoder-only (e.g., BERT), decoder-only (e.g., GPT) or encoder-decoder (e.g., T5) architectures.
- The encoder consists of N encoder layers used to process an input sequence.
- The decoder is comprised of N decoder layers used to generate the output sequence.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

The Transformer Model

- Built using either an encoder-only (e.g., BERT), decoder-only (e.g., GPT) or encoder-decoder (e.g., T5) architectures.
- The encoder consists of N encoder layers used to process an input sequence.
- The decoder is comprised of N decoder layers used to generate the output sequence.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

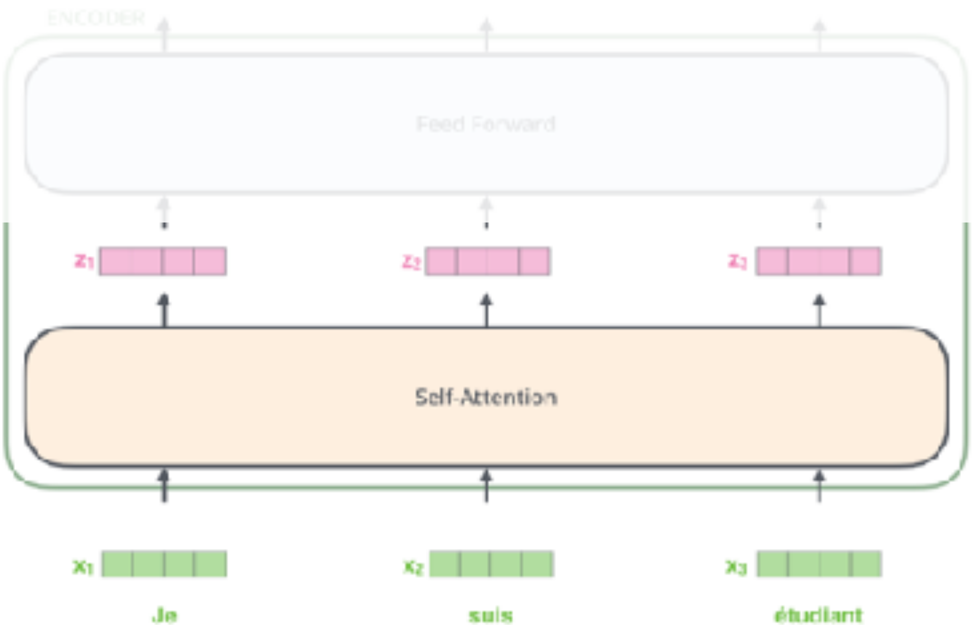
Encoder Layer

- The encoder's inputs first flow through a self-attention layer.
- The outputs of the self-attention layer are then fed into a feed-forward neural network, which is independently applied to each token (e.g., word).

*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Encoder Layer

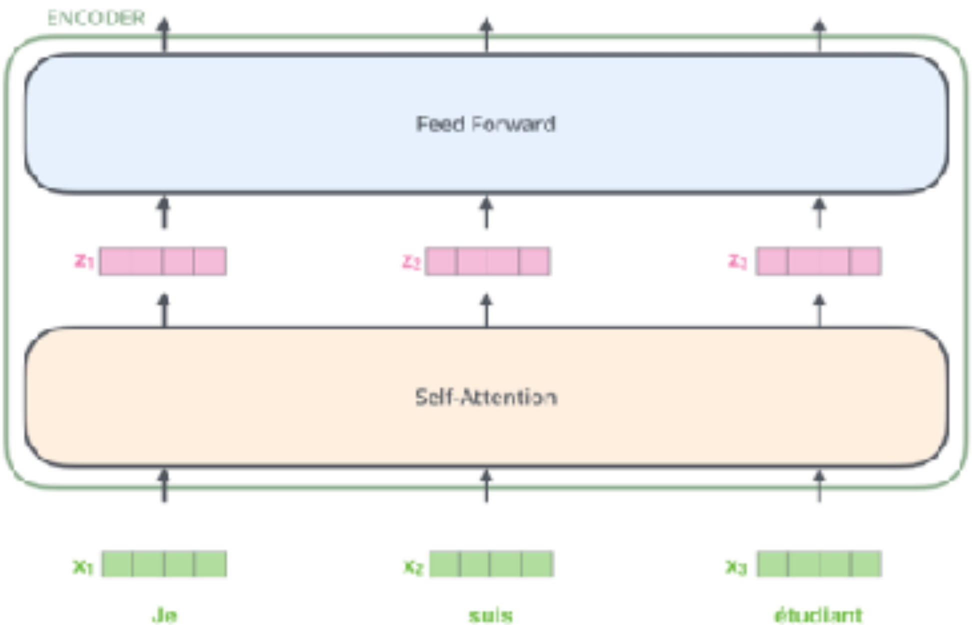
- The encoder's inputs first flow through a self-attention layer.
- The outputs of the self-attention layer are then fed into a feed-forward neural network, which is independently applied to each token (e.g., word).



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Encoder Layer

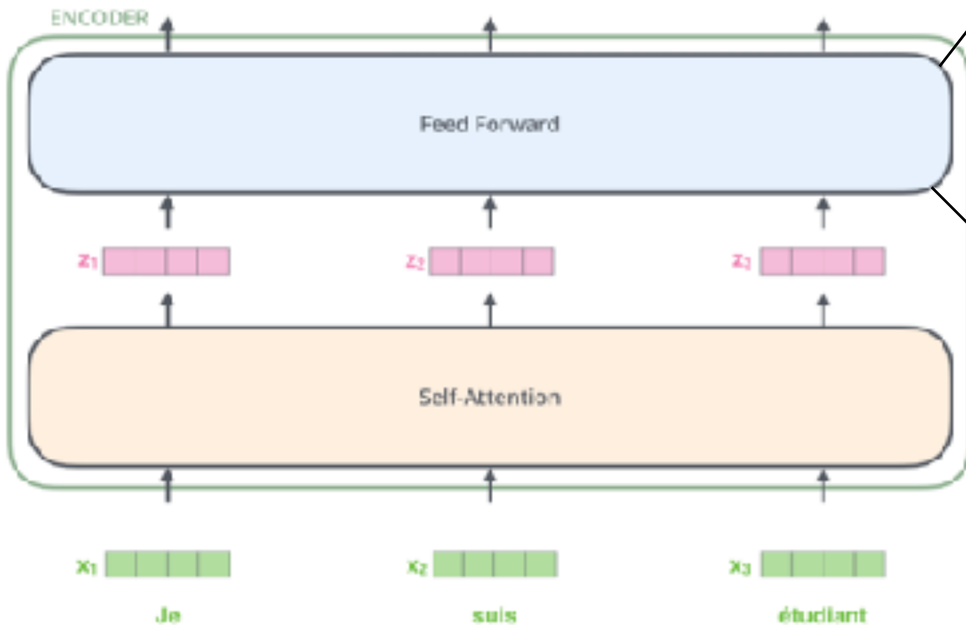
- The encoder's inputs first flow through a self-attention layer.
- The outputs of the self-attention layer are then fed into a feed-forward neural network, which is independently applied to each token (e.g., word).

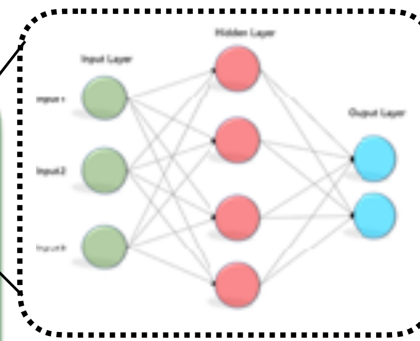


*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Encoder Layer

- The encoder's inputs first flow through a self-attention layer.
- The outputs of the self-attention layer are then fed into a feed-forward neural network, which is independently applied to each token (e.g., word).



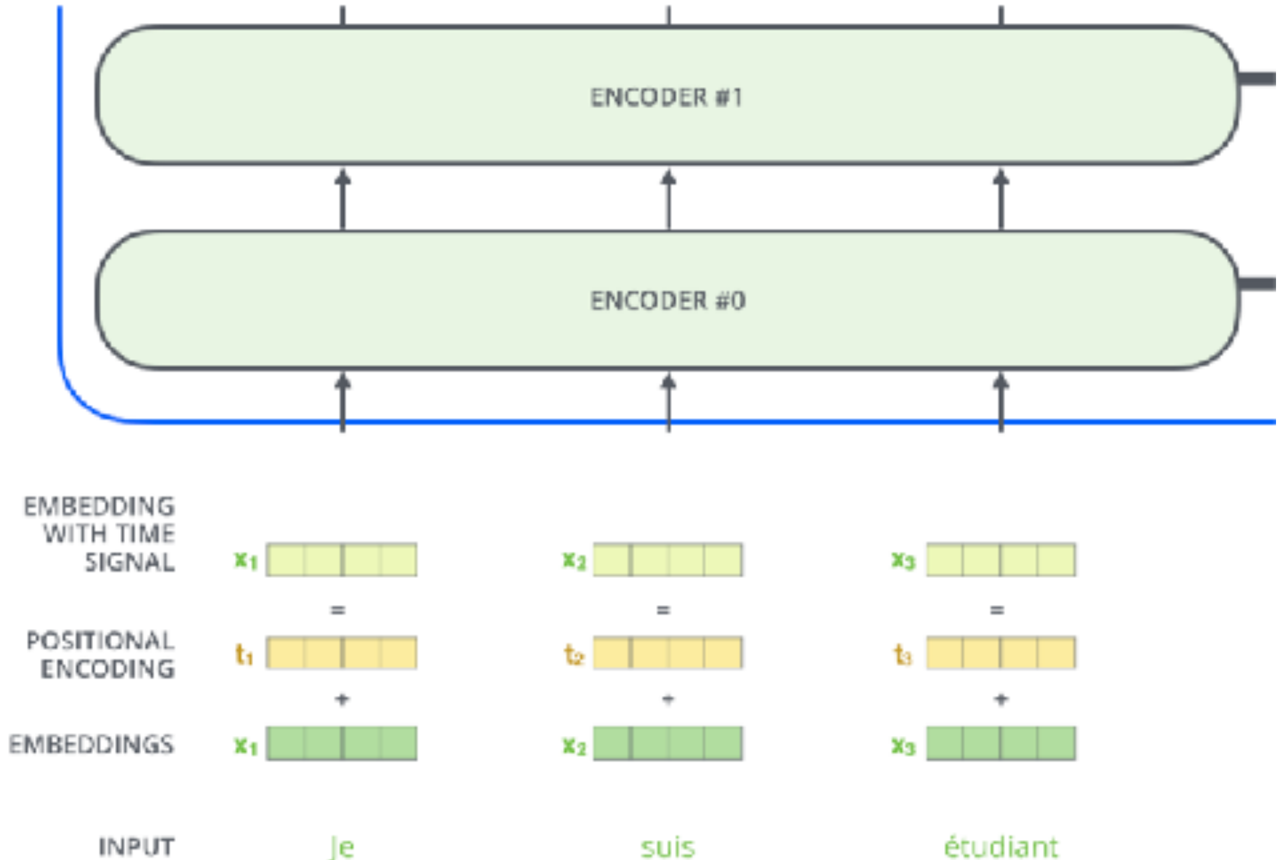


2 Layer MLP

*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Positional Encoding

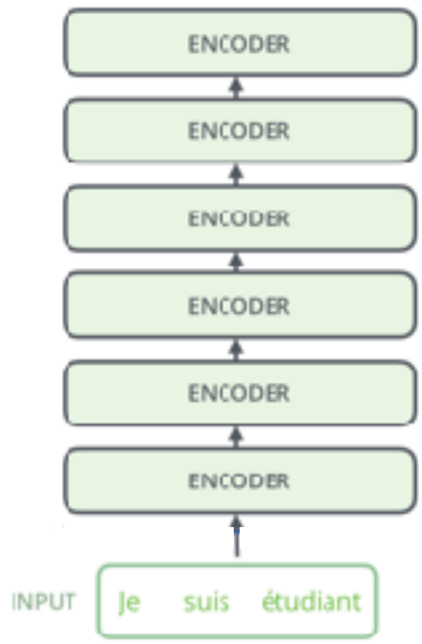
- How do we account for the order of the words in the input sequence?



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Stacking Encoder Layers

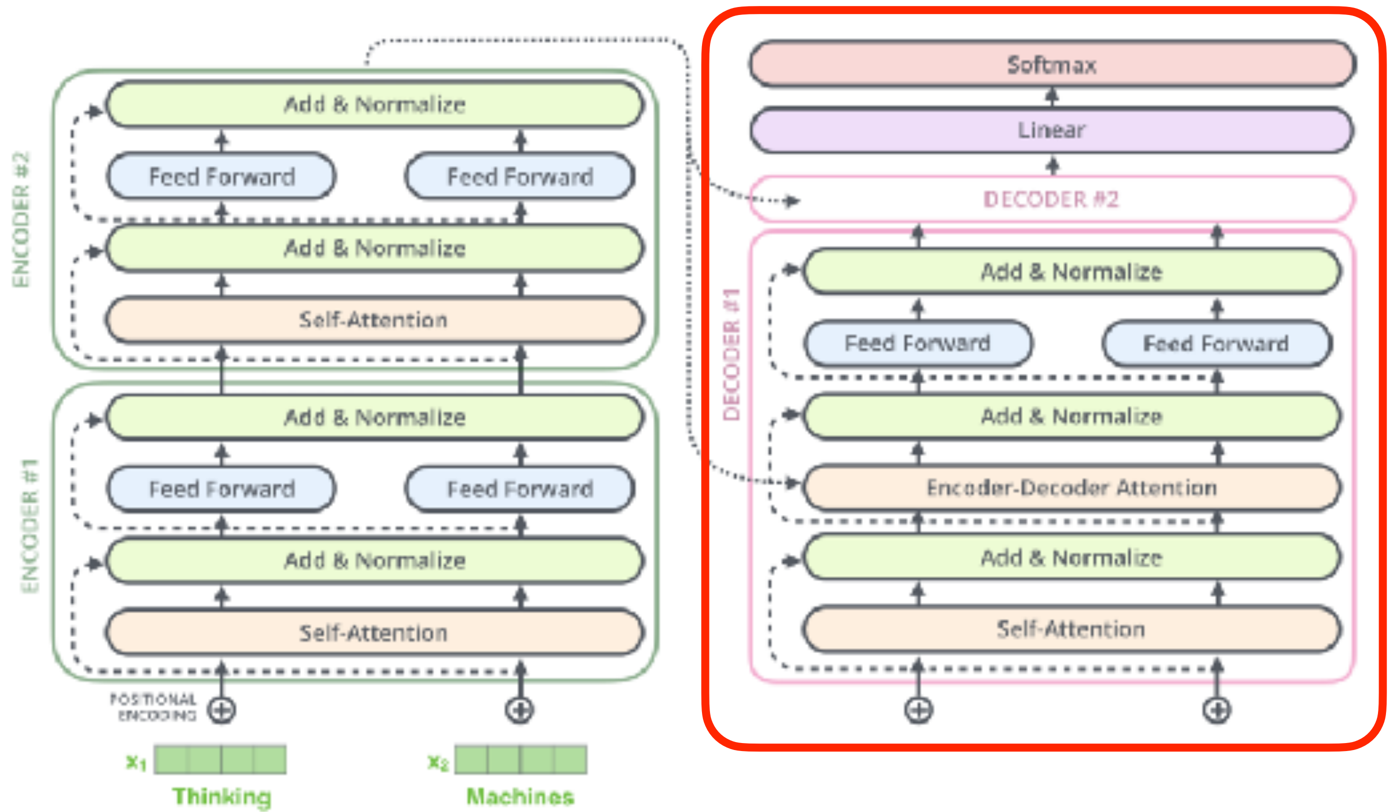
- The output of one encoder layer is then used as input to another encoder layer.
- Most standard transformer architectures consist of 6 sequential encoder layers.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Decoder

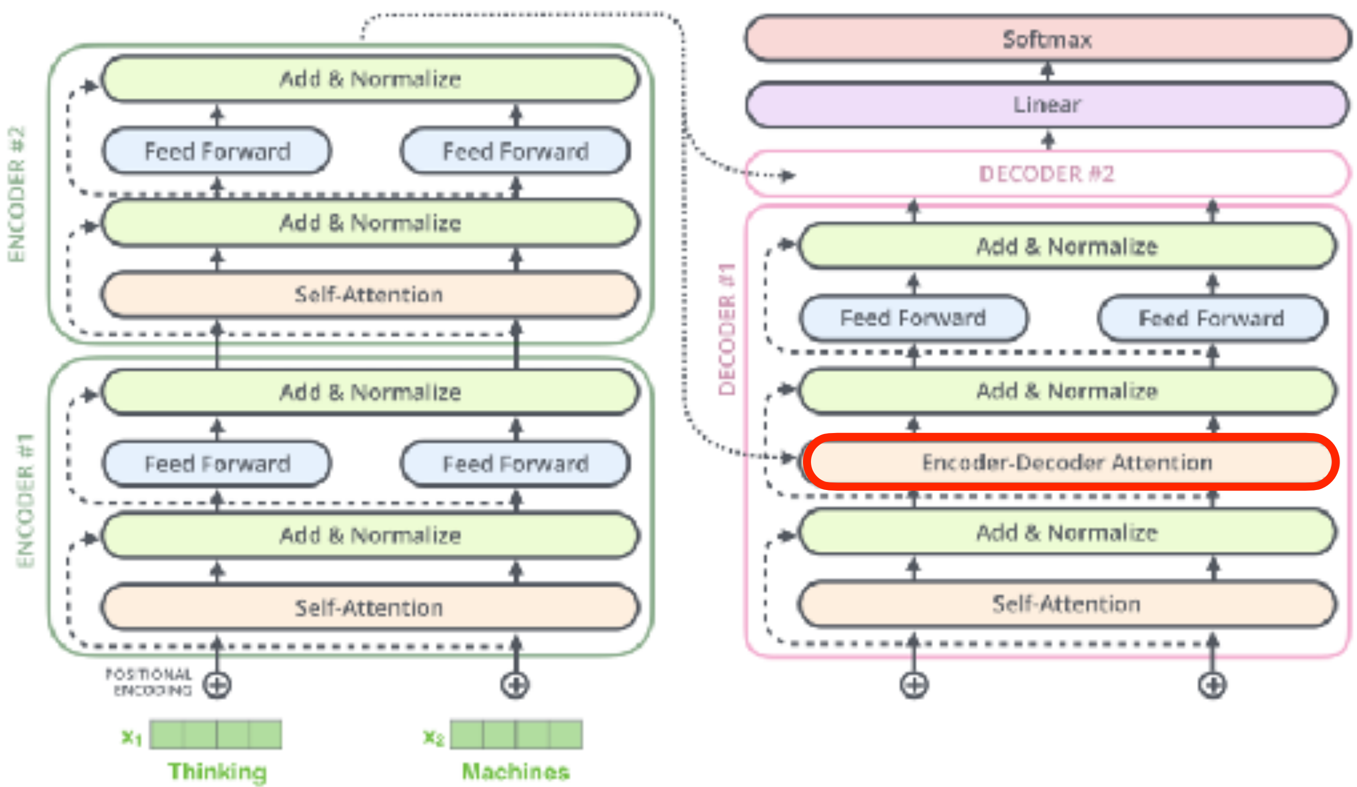
- The decoder shares a similar architecture as the encoder but it additionally has an encoder-decoder attention layer.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Decoder

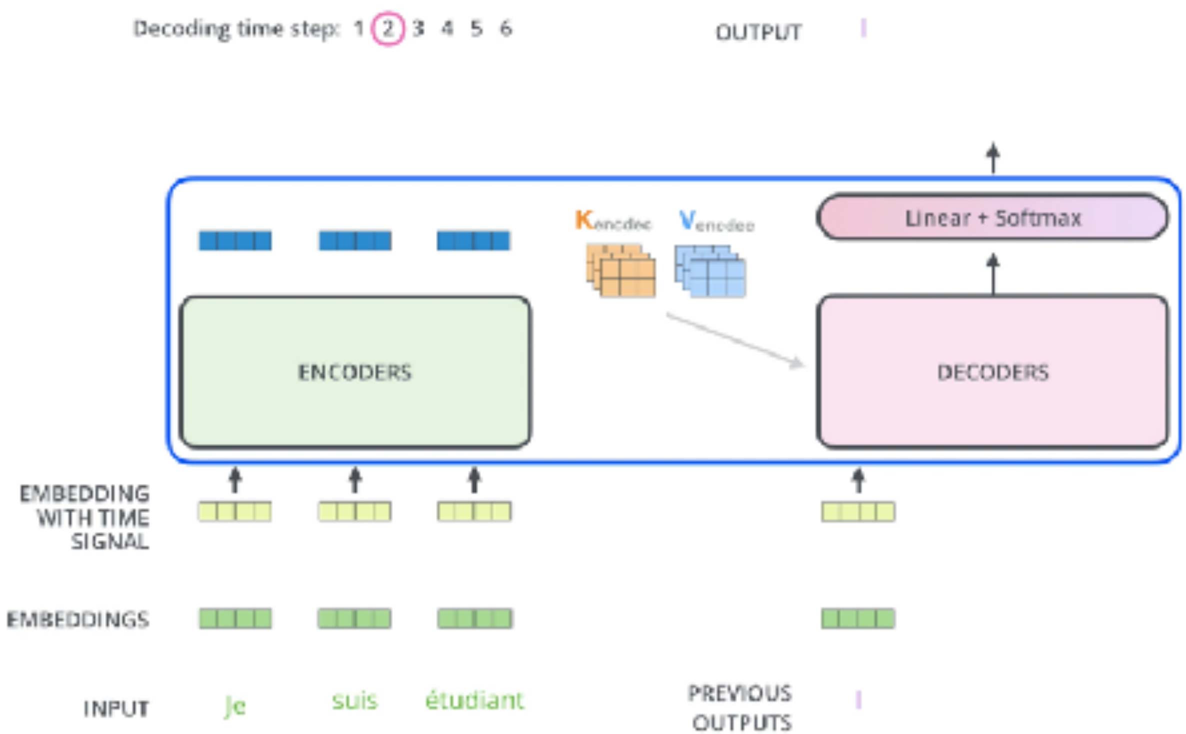
- The decoder shares a similar architecture as the encoder but it additionally has an encoder-decoder attention layer.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Decoder

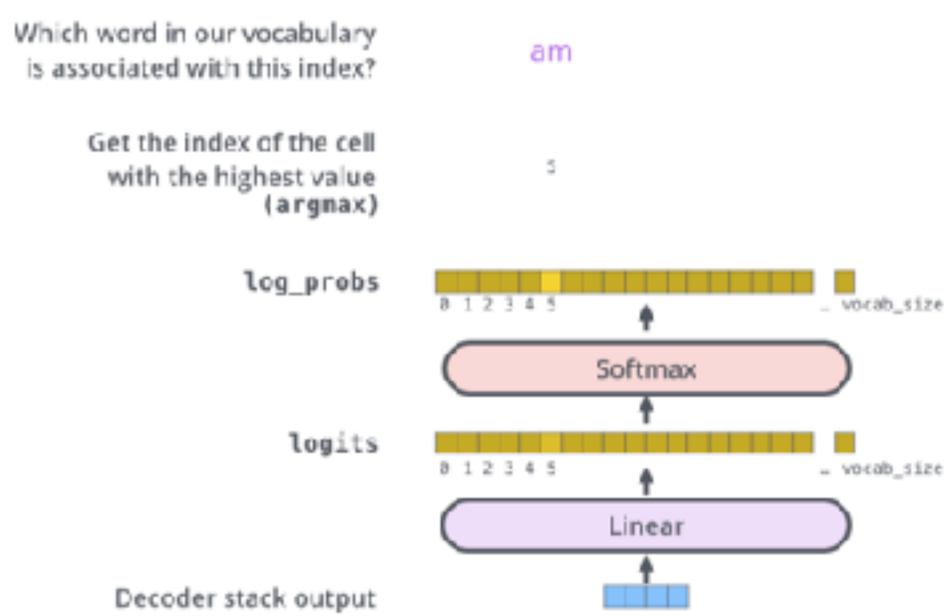
- The decoder uses previously generated tokens (e.g., words) to decide, which tokens it should generate next.



*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

The Final Layer

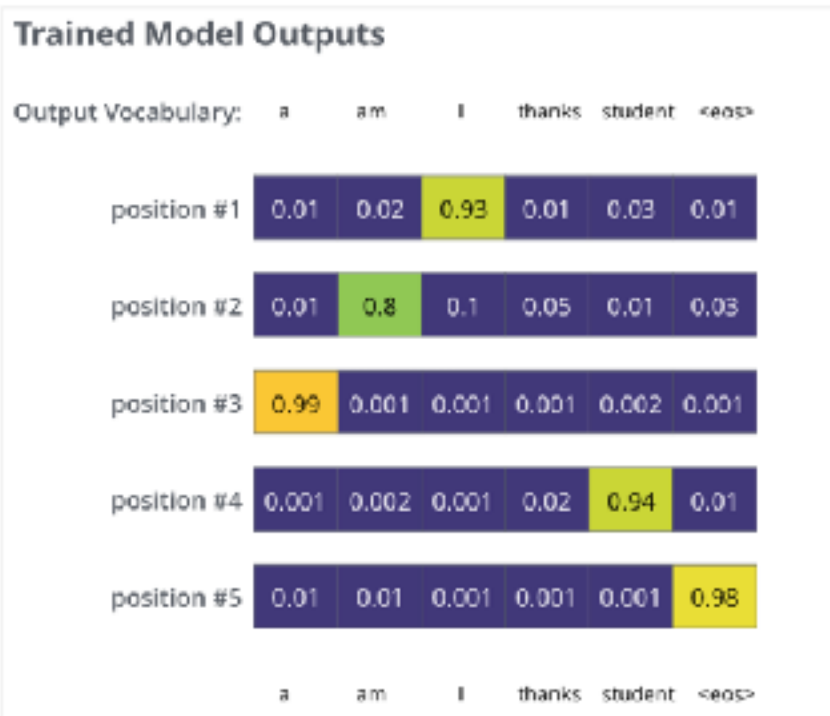
- The linear layer projects the vector produced by the stack of decoders, into a logits vector.
- Each entry in the logits vector corresponds to a score associated with a unique word.



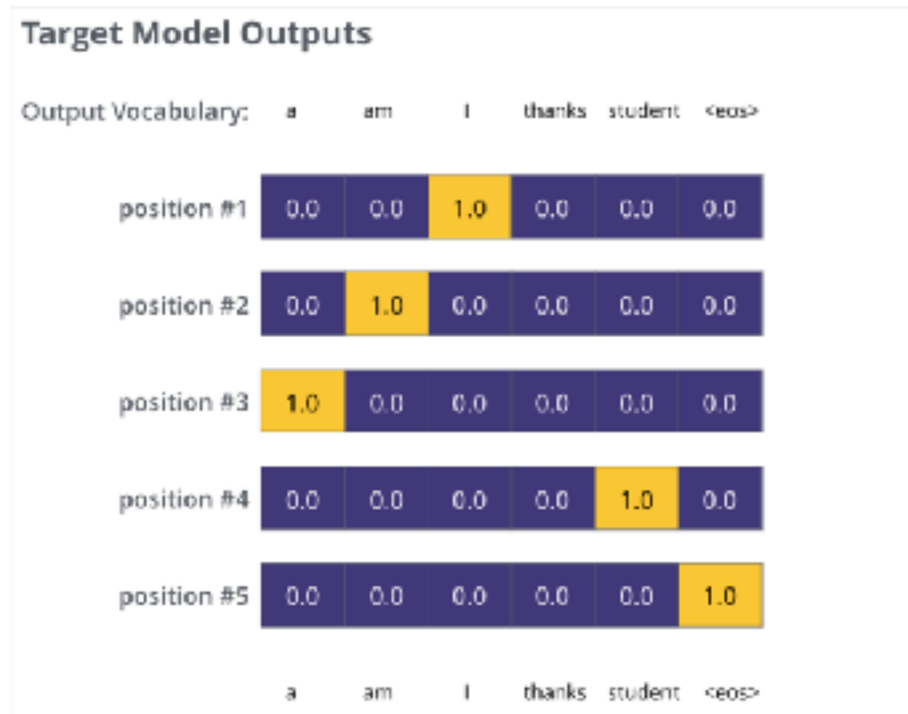
*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Loss Function

- We can use a cross-entropy loss to optimize our transformer model end-to-end.



a) Trained Model Predictions

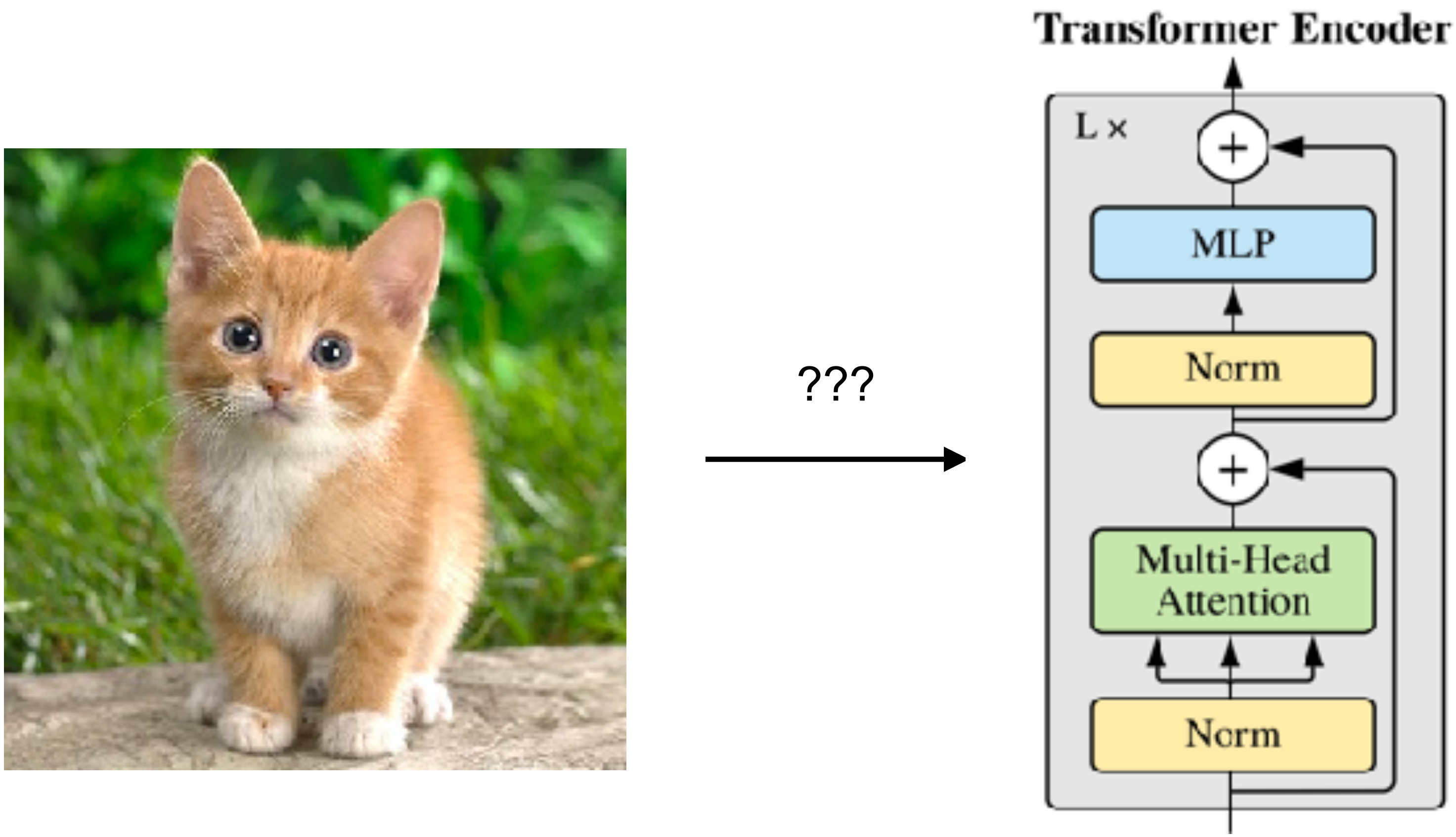


b) Ground Truth

*illustrations adopted from https://jalammar.github.io/illustrated-transformer/

Transformers for Visual Data

- How do we apply transformers on visual data (e.g., images or videos)?



First Assignment

- **Undergraduates:** Please upload a brief statement in the Assignments section of Canvas stating which prerequisite you've met by **Sunday, January 14th, 11:59 pm**.

Second Assignment

- The reading list is posted here.
- Select the following:
    1. Seven 30min or 45min papers for standard paper presentations (marked **red** and **purple** in the schedule). Any combo of the papers suffice (e.g., five 30min & two 45min papers, all 30min papers, etc.)
    2. Three 20min papers for paper battles (marked **green** in the schedule).
- Make sure that the papers that you selected will **NOT** be presented by me.
- Rank the papers in each of these lists in descending order of preference (from highest to lowest) and upload them to Canvas **by Wednesday, Jan 17th, 11:59 PM** (please include paper IDs in your lists!!).
- I will then update the website with the paper assignments.

Third Assignment

- Complete the paper critique for paper [2] An image is worth 16x16 words: Transformers for image recognition at scale.

- Upload it to Canvas by **11:59 PM on Tuesday, Jan 16th**.