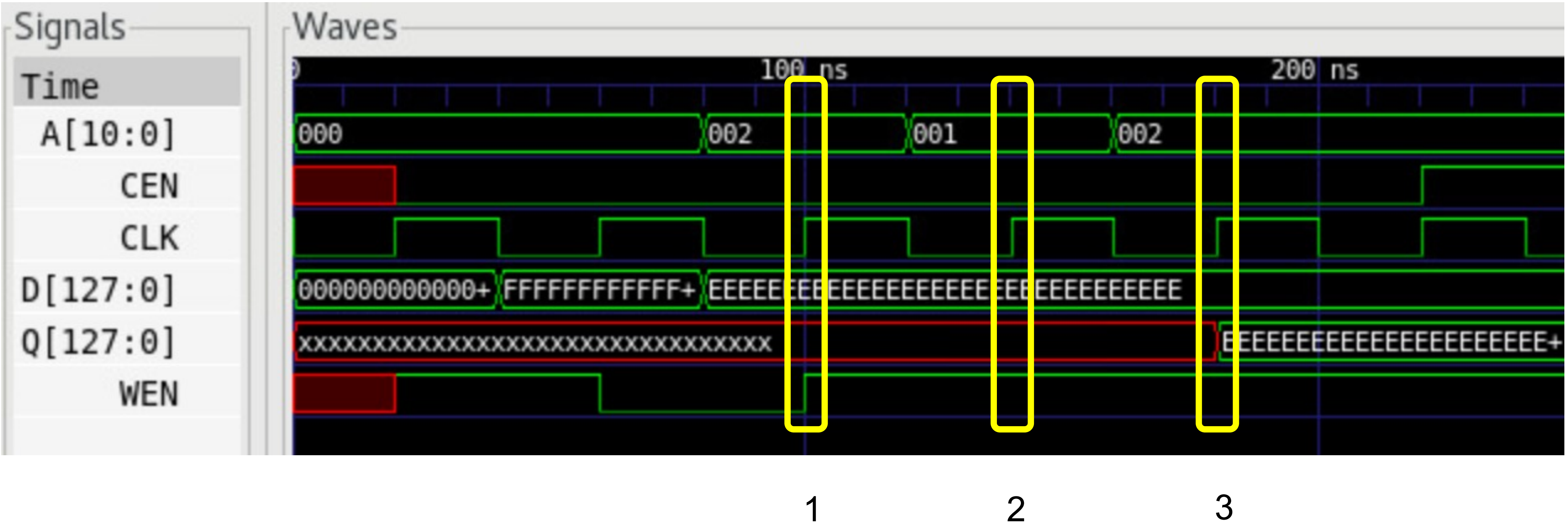
# ECE284 Fall 21 W6S1

## Low-power VLSI Implementation for Machine Learning

Prof. Mingu Kang

UCSD Computer Engineering

# [Example2] Memory Write and Read




1. "WEN=0 & ADD = 002 & D = EEEE.." are received by SRAM
2. "WEN=1 & ADD = 001" are received by SRAM
3. "WEN=1 & ADD = 002" are received by SRAM

From tb, it is good to apply new input at CLK = 0, or pass through CEN_EXT

# [HW_ prob2] Memory Write and Read with VGG Model

- Open [HW6_Prob2]_Memory_Write_Read.ipynb
- Assume 2D systolic array size: 8 x 8
- For tile_ic = 0, tile_oc = 0, kij = 0, nij = starting from 200 to 264

- Print the weight contents in ‘weight.txt’ file such as ‘activation.txt’
- But, please note weight can be negative number and you need to follow 2’s complement number system. e.g., -7 = -7 + 16 = 1001
- Print the expected psum (16b) result in ‘psum.txt’ file

- Then, try “w6/hw1” with your own ‘activation.txt’ file to write the data into the memory at the address of A = 0 – 63.
- Then, read the memory from the address of A = 0 – 63.
- Compare the read data is the same as expected in the ‘activation.txt’ file

# NLP Applications (GLUE data set)

| Dataset | Description | Data example | Metric |
|---|---|---|---|
| CoLA | Is the sentence grammatical or ungrammatical? | "This building is than that one."<br>= **Ungrammatical** | Matthews |
| SST-2 | Is the movie review positive, negative, or neutral? | "The movie is funny , smart , visually inventive , and most of all , alive ."<br>= **.93056 (Very Positive)** | Accuracy |
| MRPC | Is the sentence B a paraphrase of sentence A? | A) "Yesterday , Taiwan reported 35 new infections , bringing the total number of cases to 418 ."<br>B) "The island reported another 35 probable cases yesterday , taking its total to 418 ."<br>= **A Paraphrase** | Accuracy / F1 |
| STS-B | How similar are sentences A and B? | A) "Elephants are walking down a trail."<br>B) "A herd of elephants are walking along a trail."<br>= **4.6 (Very Similar)** | Pearson / Spearman |
| QQP | Are the two questions similar? | A) "How can I increase the speed of my internet connection while using a VPN?"<br>B) "How can Internet speed be increased by hacking through DNS?"<br>= **Not Similar** | Accuracy / F1 |
| MNLI-mm | Does sentence A entail or contradict sentence B? | A) "Tourist Information offices can be very helpful."<br>B) "Tourist Information offices are never of any help."<br>= **Contradiction** | Accuracy |
| QNLI | Does sentence B contain the answer to the question in sentence A? | A) "What is essential for the mating of the elements that create radio waves?"<br>B) "Antennas are required by any radio receiver or transmitter to couple its electrical connection to the electromagnetic field."<br>= **Answerable** | Accuracy |
| RTE | Does sentence A entail sentence B? | A) "In 2003, Yunus brought the microcredit revolution to the streets of Bangladesh to support more than 50,000 beggars, whom the Grameen Bank respectfully calls Struggling Members."<br>B) "Yunus supported more than 50,000 Struggling Members."<br>= **Entailed** | Accuracy |
| WNLI | Sentence B replaces sentence A's ambiguous pronoun with one of the nouns - is this the correct noun? | A) "Lily spoke to Donna, breaking her concentration."<br>B) "Lily spoke to Donna, breaking Lily's concentration."<br>= **Incorrect Referent** | Accuracy |

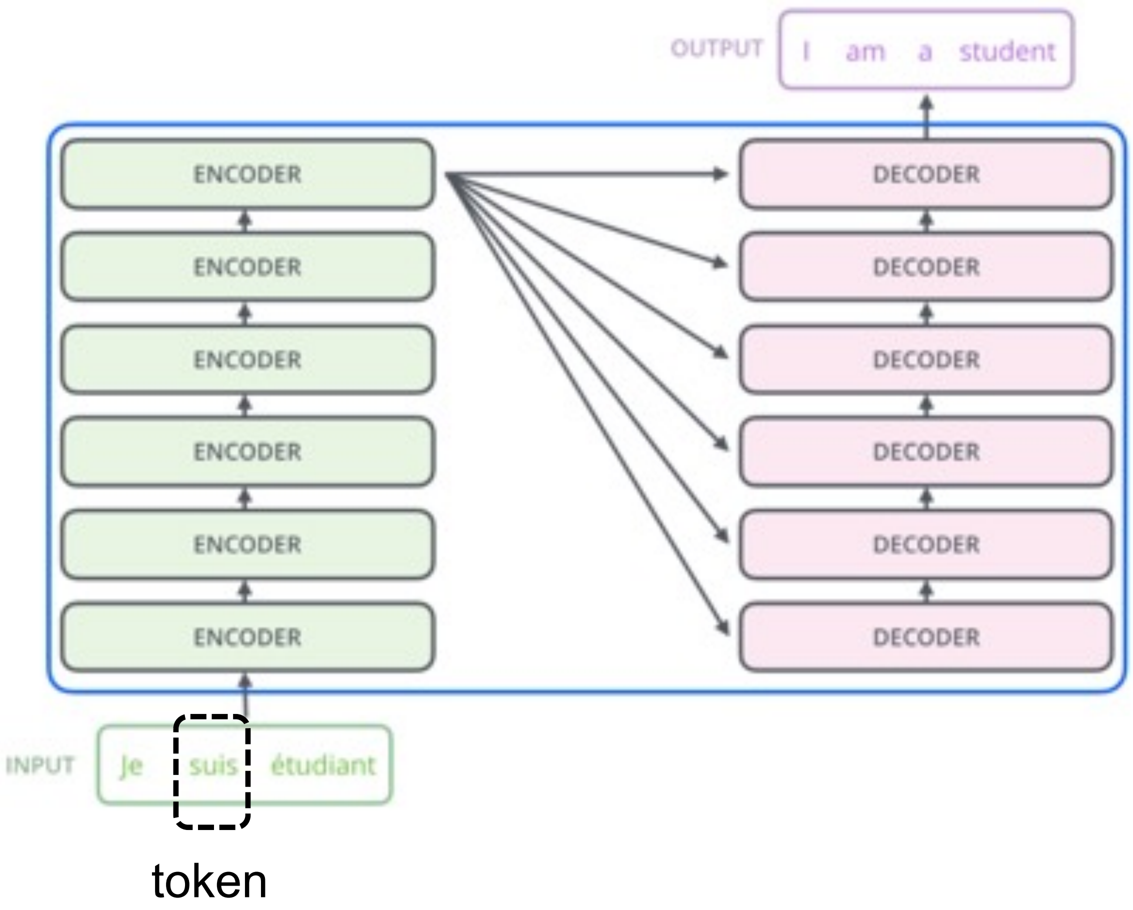
https://mccormickml.com/2019/11/05/GLUE/

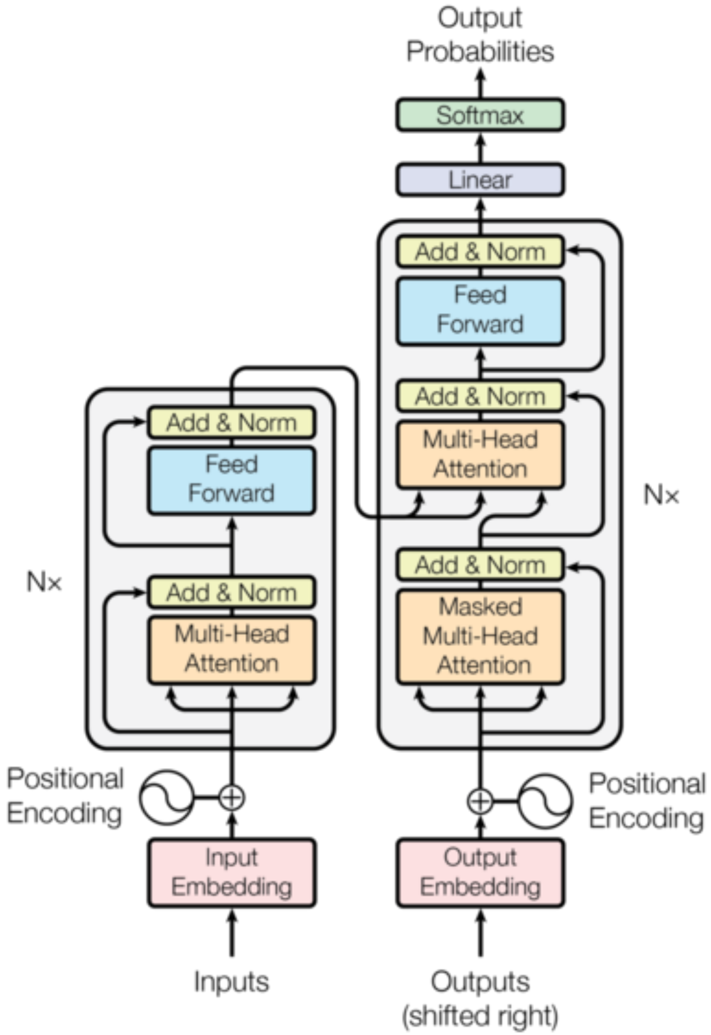
# Data Sets

- GLUE data set (https://gluebenchmark.com/)
- Facebook bAbI 20 QA tasks (https://research.fb.com/downloads/babi/)
- Stanford Natural Language Inference (SNLI) Corpus (https://nlp.stanford.edu/projects/snli/)
- SQUAD dataset (https://rajpurkar.github.io/SQuAD-explorer/)
- ….

- Above data sets are generally the collection of multiple tasks such as GLUE.

# Transformer




A. Vaswani, "*Attention is all you need*", Neurips17

# Data Preparation

- Token: word or sometimes sentence
- Embedding: conversion from token to a vector,

  - e.g., nn.Embedding(1000, 512): maps 1000 different types of embedding to a unique vectors with length 512

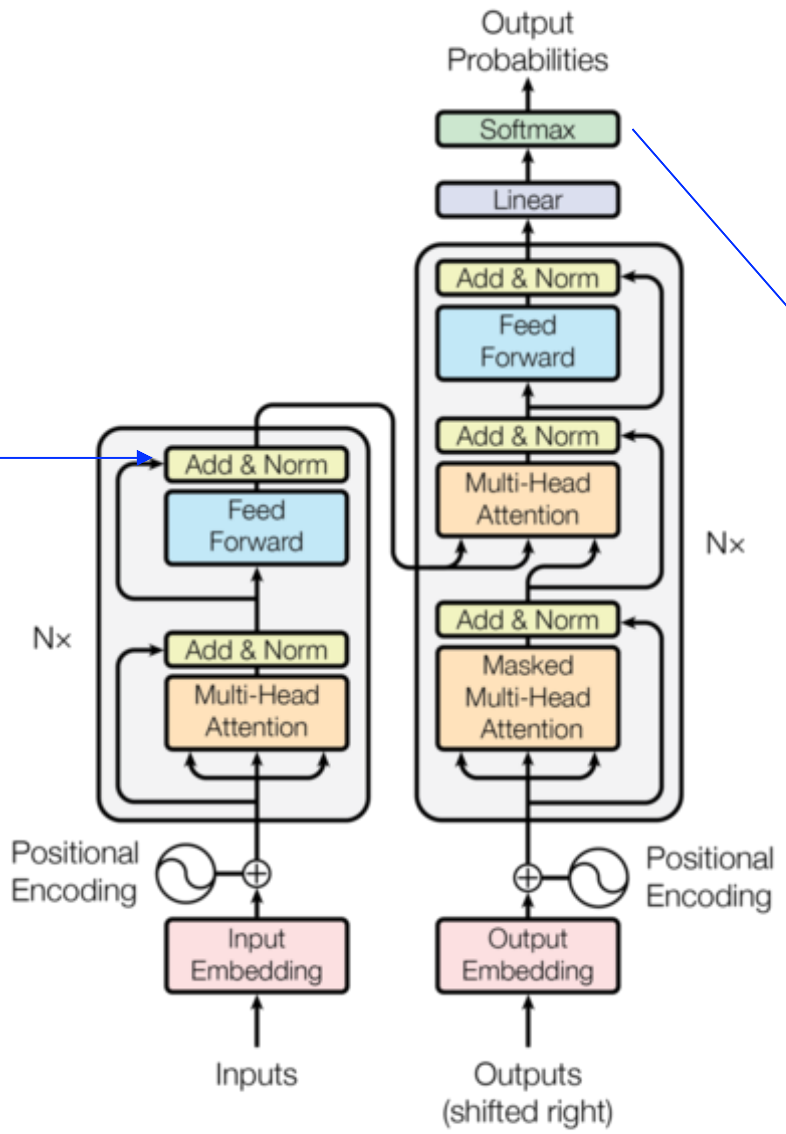
- Positional encoding:

  - provide the position information, e.g., the location of word in the sentence
  - simple sinusoidal signals e.g., $PE(p,i) = \sin(p/10000^{i/512})$, where $i = 1 – 512$, and p is position
  - Based on the position, unique 512 length positional vector is generated and added to the embedding

# [Example1] Embedding for Token

- nn.Embedding(vaca #, length of vector)
- try more number of vocabularies than your “voca #” in nn.Embedding

# Transformer




($\vec{x}$ - mean) / std

$$\sigma(\vec{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$$

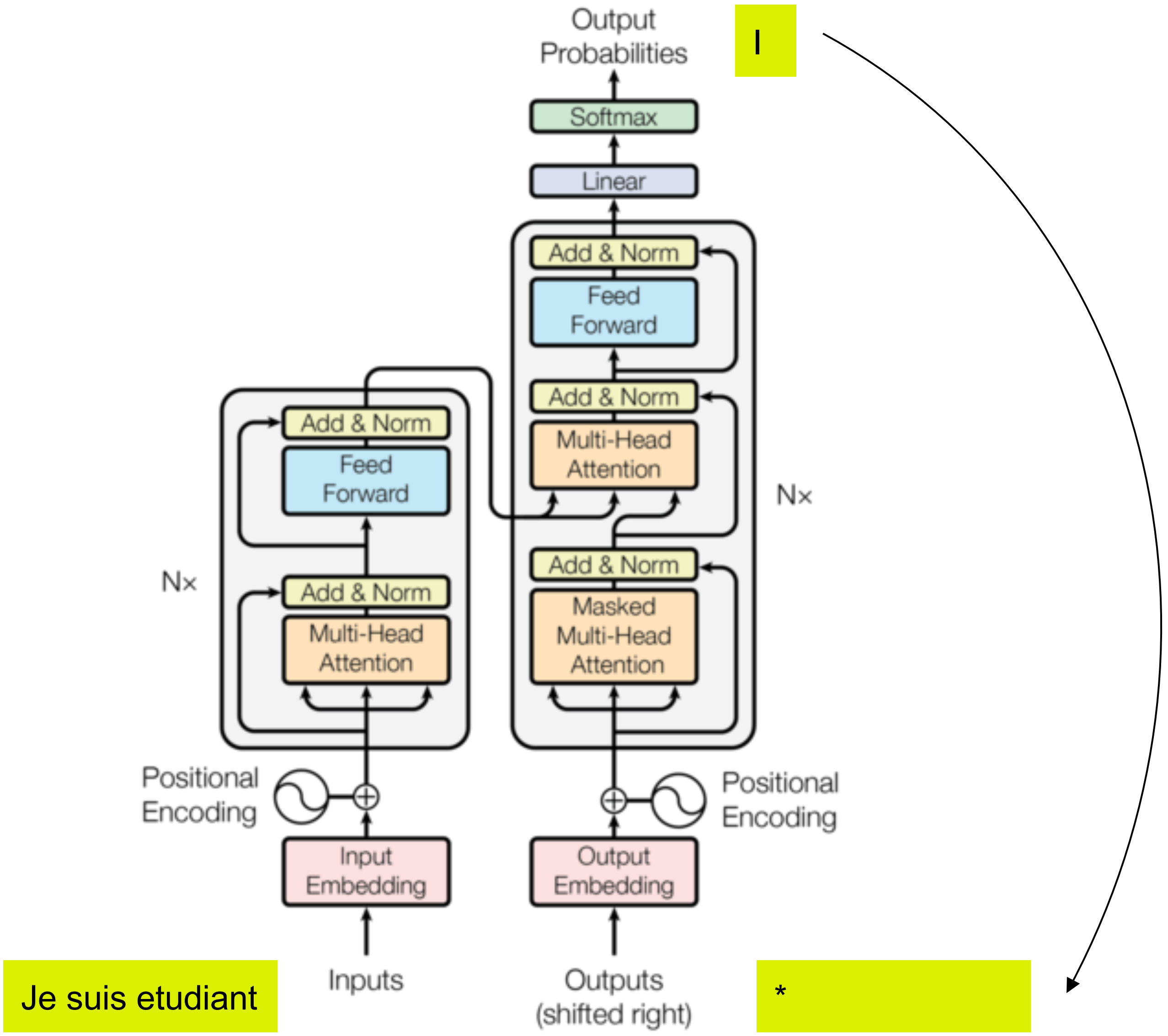
$\vec{x}$: is a vector $\{x_1, \ldots, x_K\}$

$\sigma(\vec{x})_i$: softmax for $x_i$

A. Vaswani, "*Attention is all you need*", Neurips17

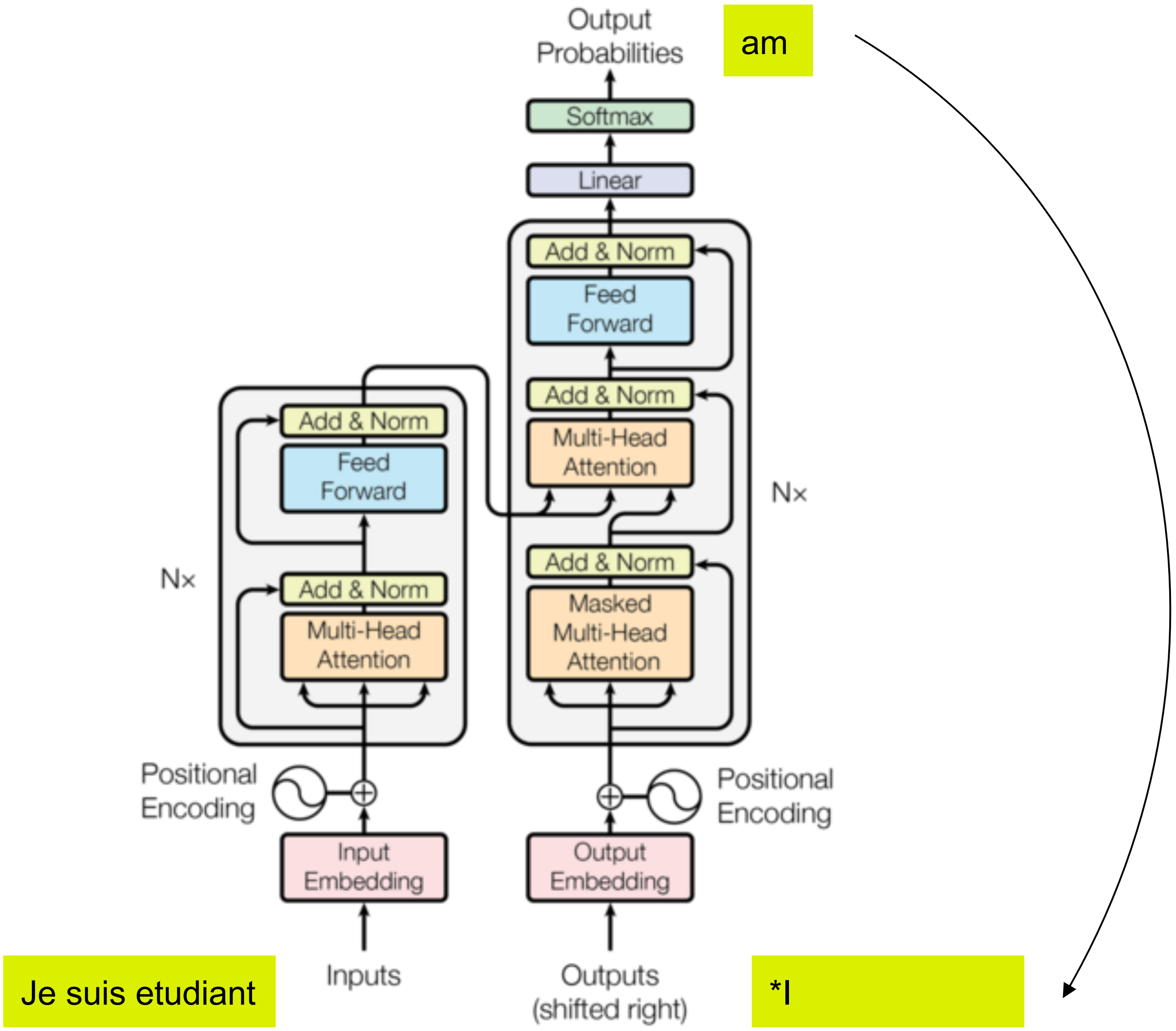
# Inference of Transformer (1st iteration)

*: start of sentence
#: end of sentence

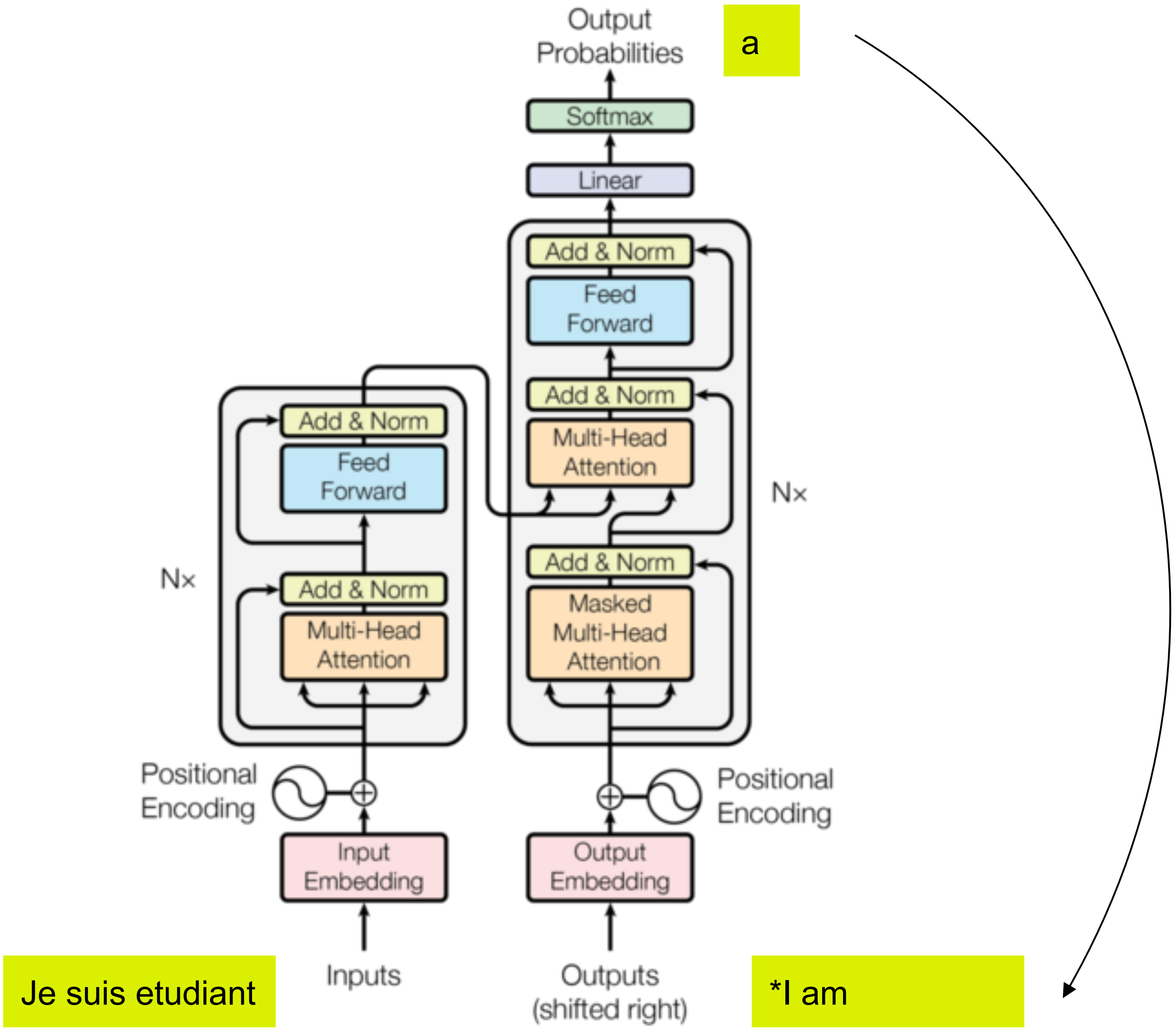


# Inference of Transformer ($2^{nd}$ iteration)

*: start of sentence
#: end of sentence

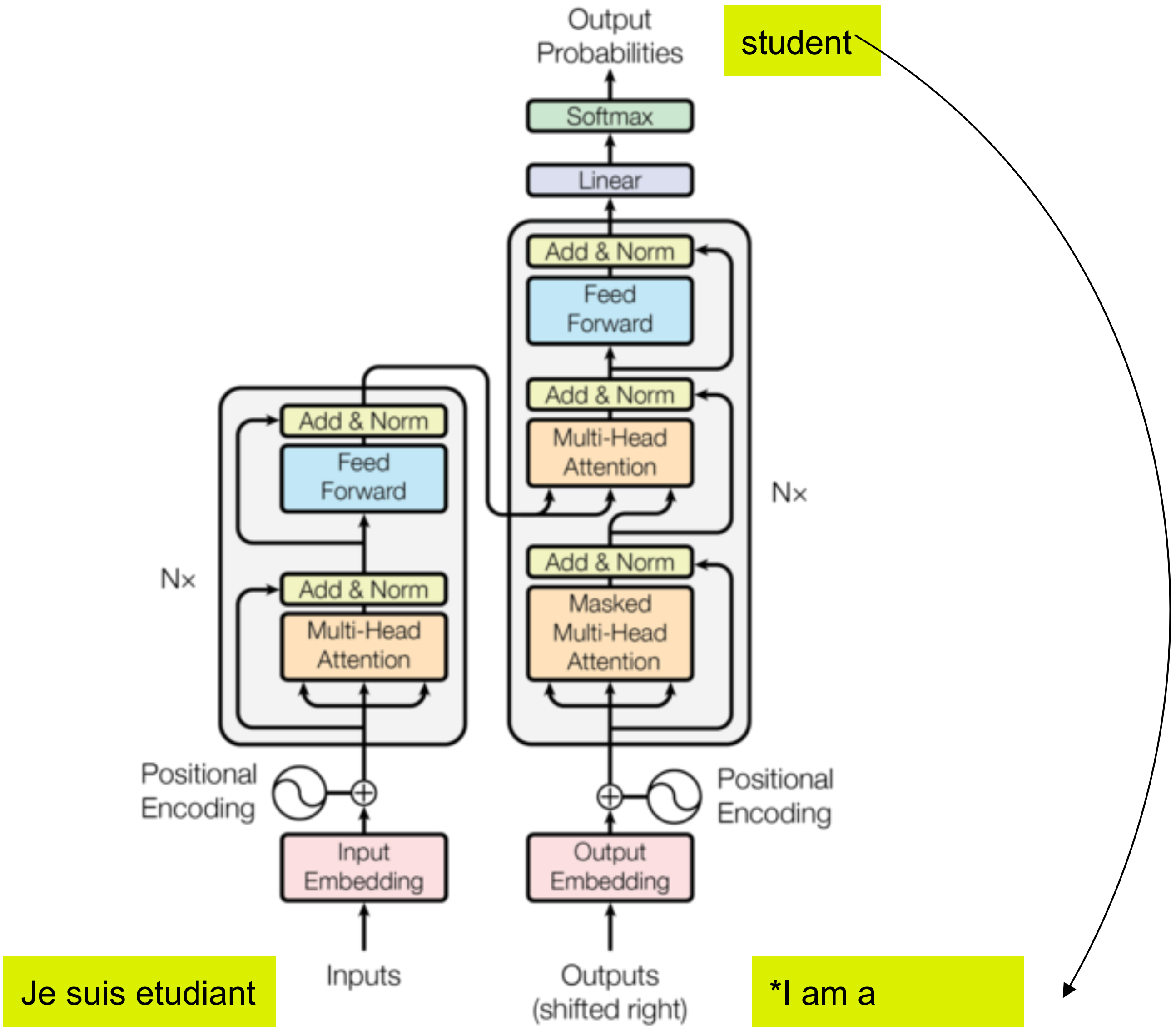


# Inference of Transformer (3rd iteration)




*: start of sentence
#: end of sentence

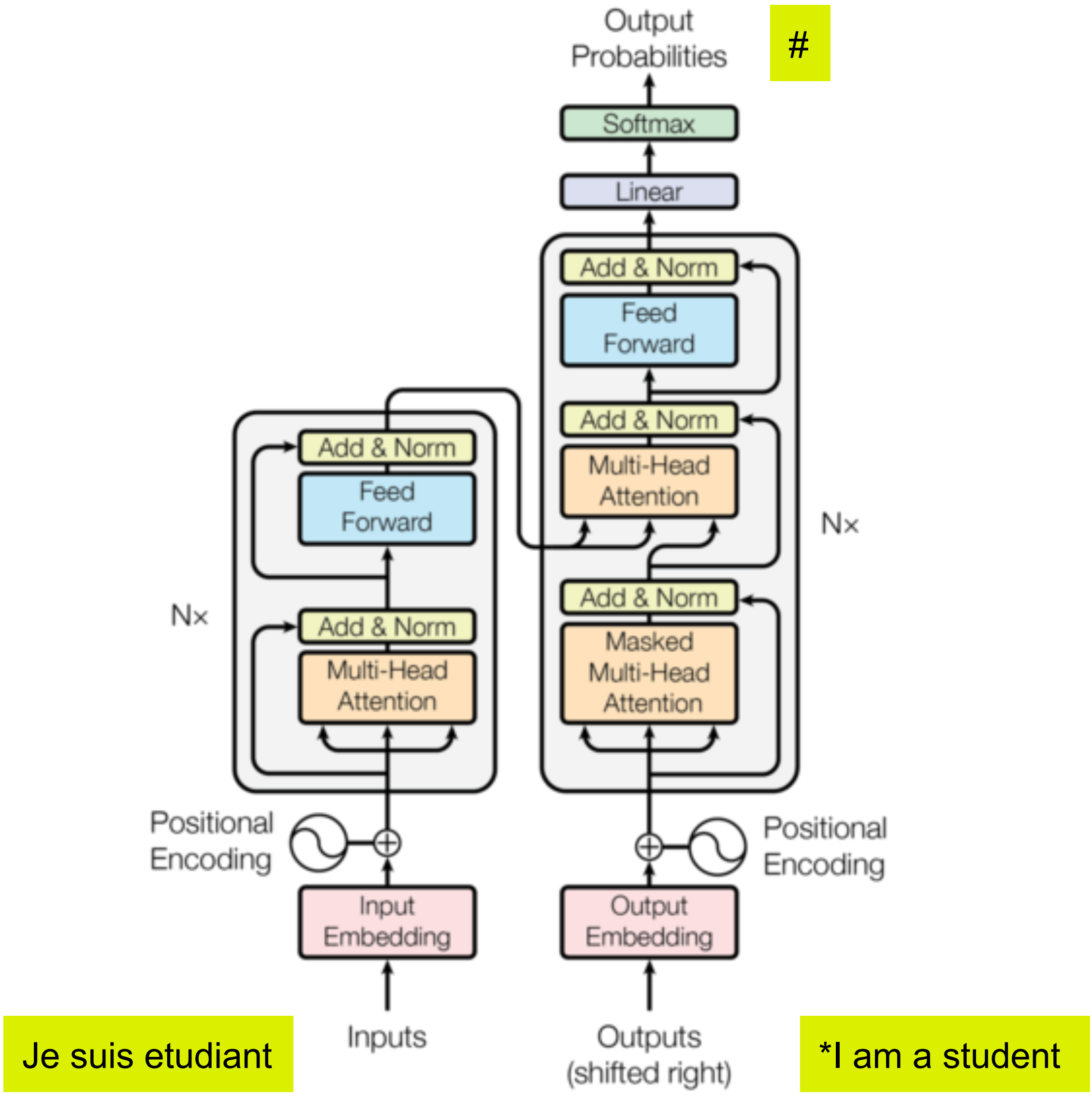
# Inference of Transformer ($4^{th}$ iteration)




*: start of sentence
#: end of sentence

# Inference of Transformer (5th iteration)

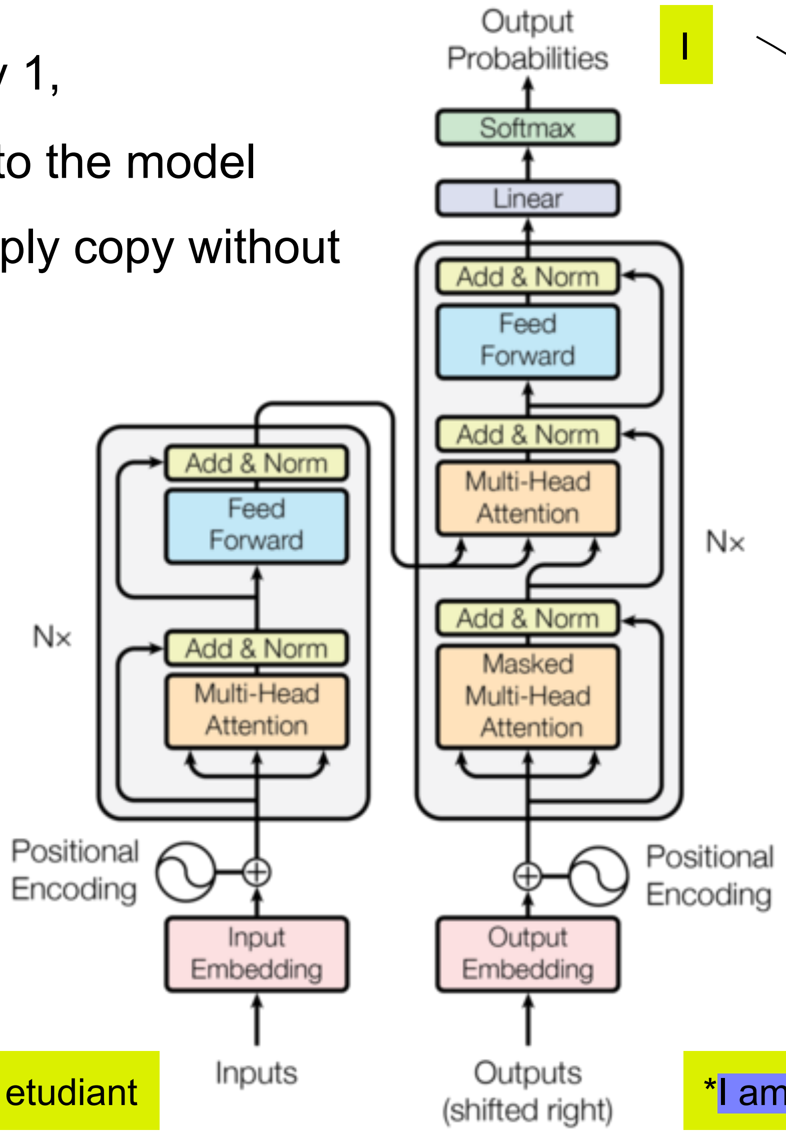
*: start of sentence
#: end of sentence



# Training of Transformer (1st iteration)

- By applying mask + right shift by 1, do not show the correct answer to the model
- Otherwise, the model would simply copy without trying to predict the next word

*: start of sentence
#: end of sentence




I

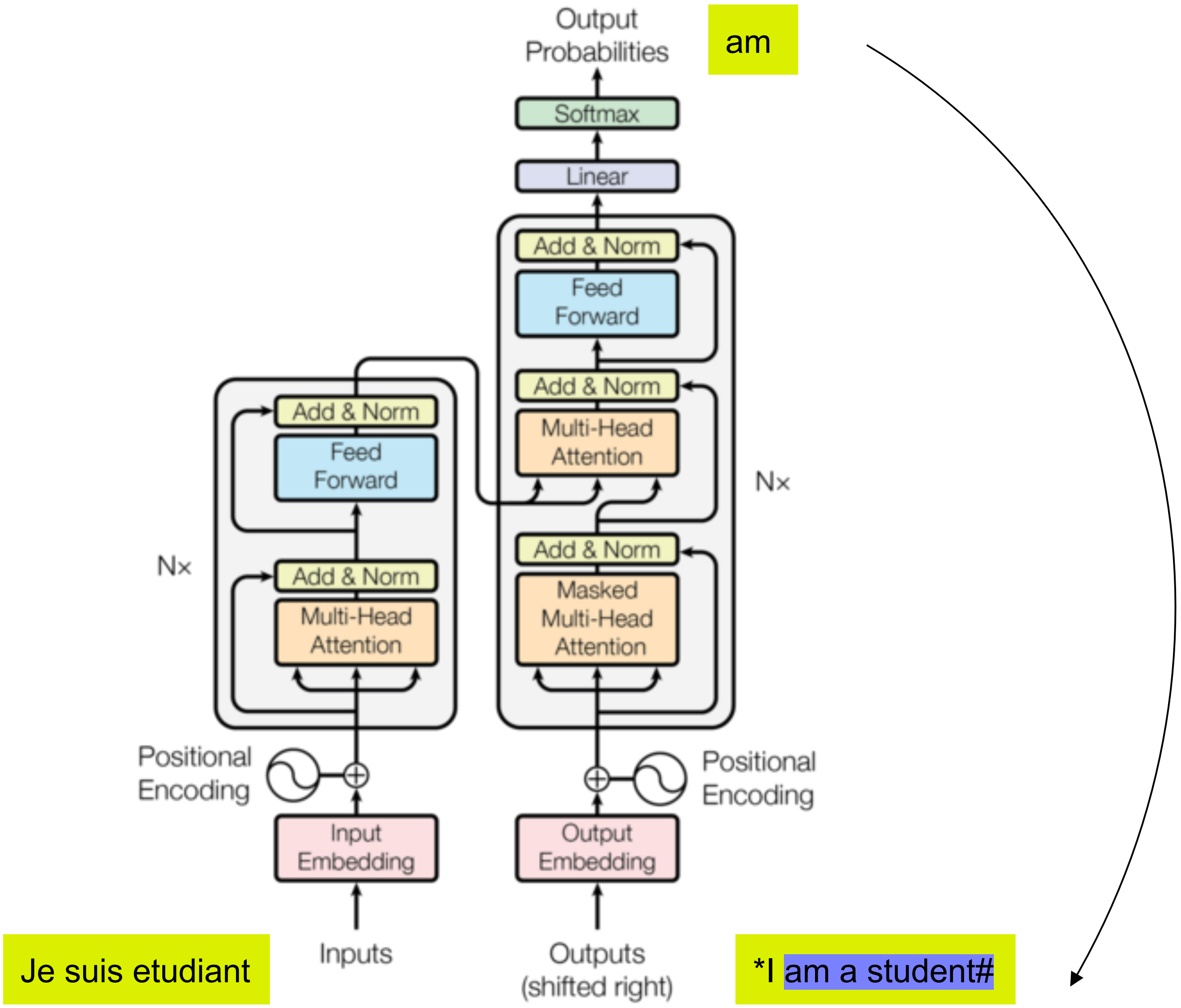
Je suis etudiant

*I am a student#

Masked area not visible to decoder

Shifted by 1 to right due to *

# Inference of Transformer (2nd iteration)

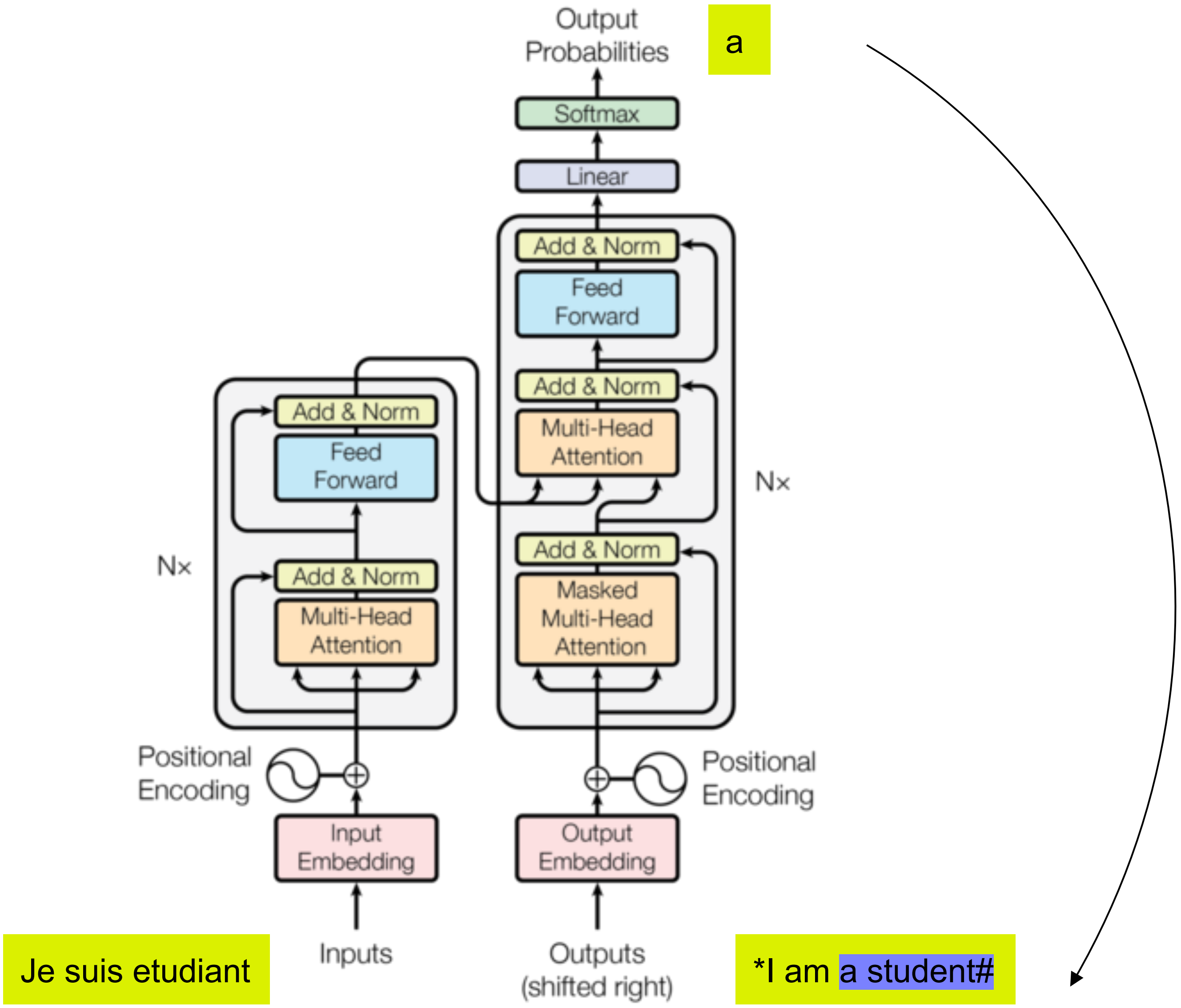



*: start of sentence
#: end of sentence

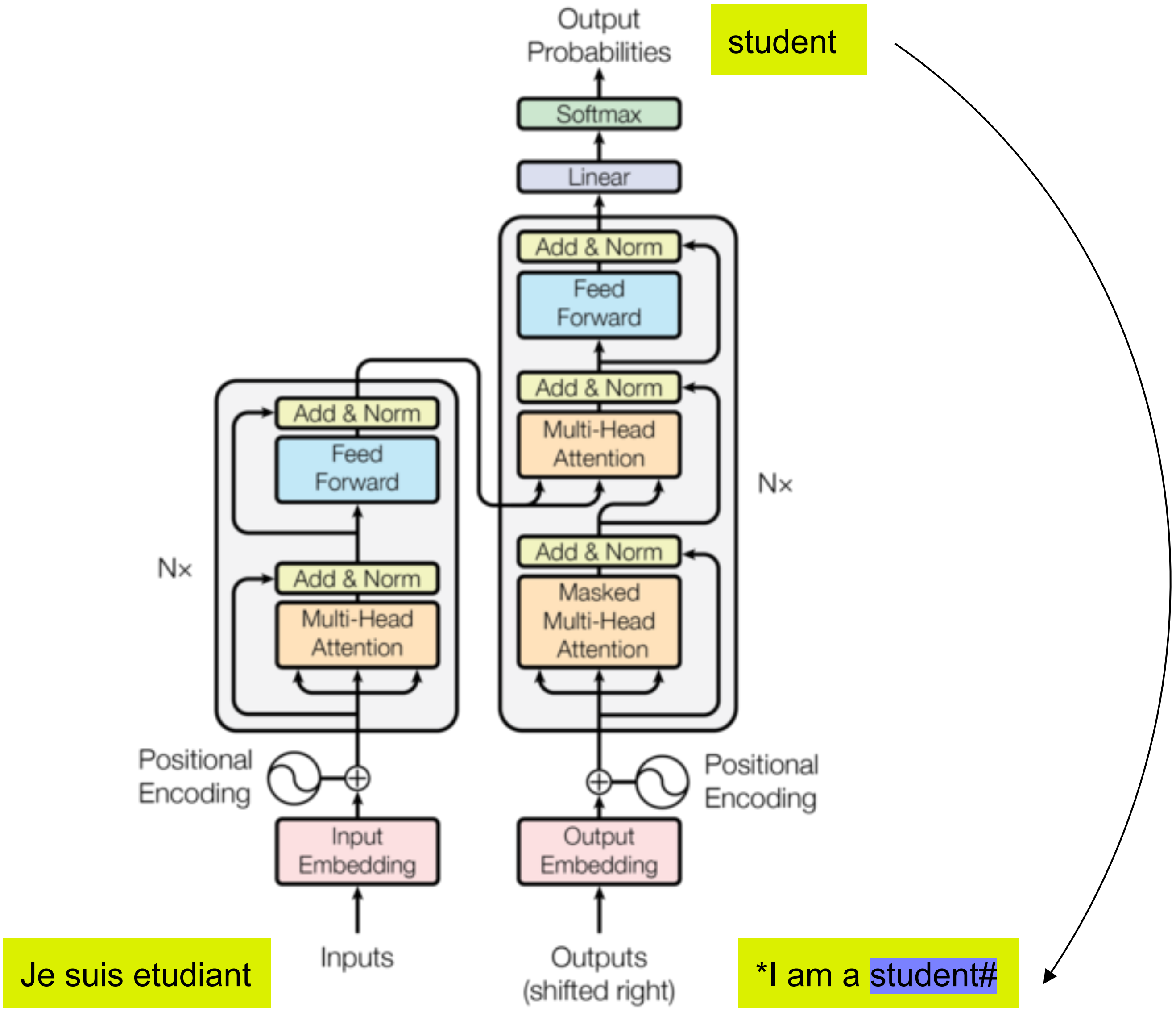
- Now, we can see the target word was “I”.
- So, we can compute the loss now.

# Inference of Transformer (3rd iteration)

*: start of sentence
#: end of sentence



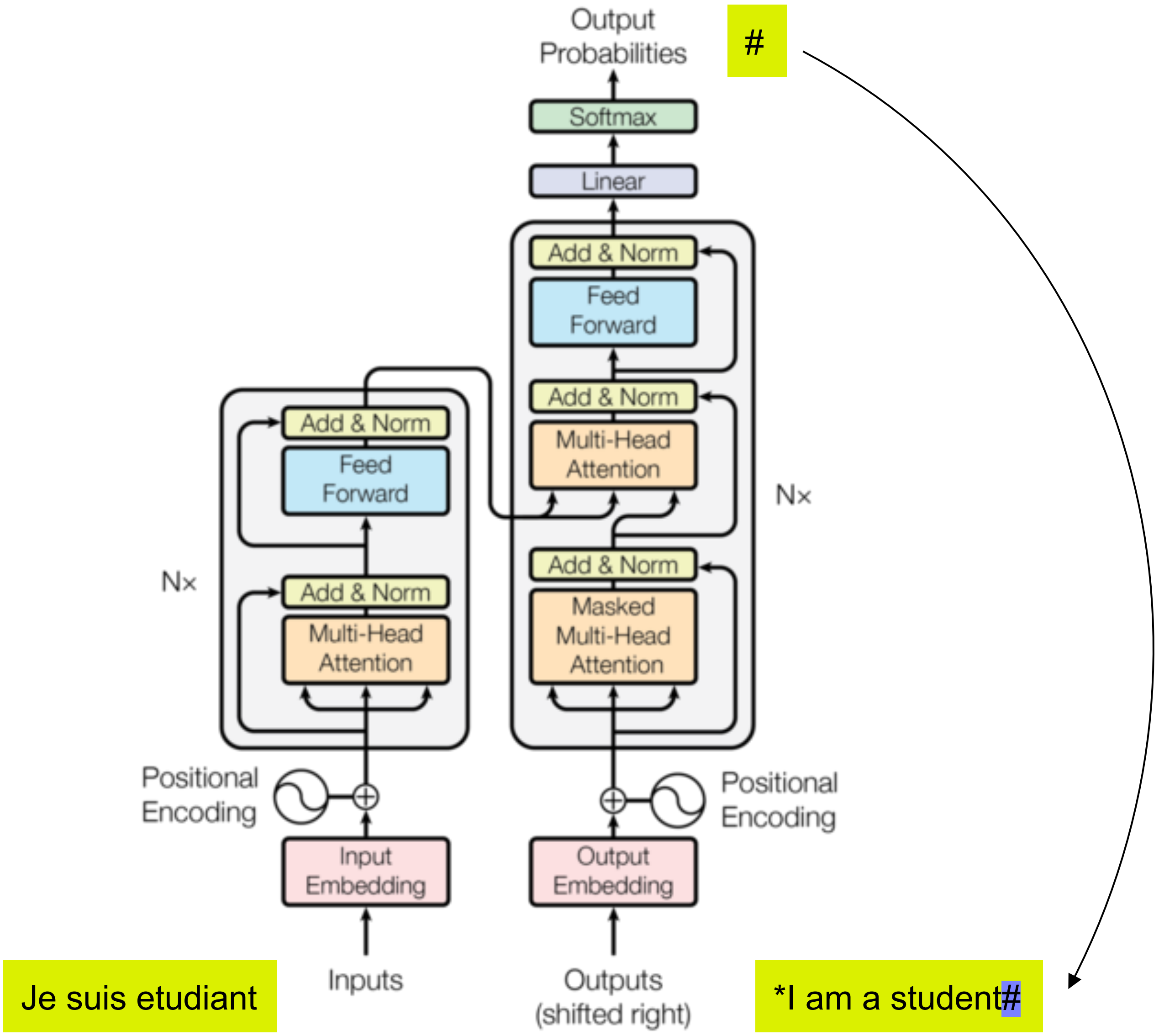
# Inference of Transformer (4th iteration)




*: start of sentence
#: end of sentence

# Inference of Transformer (5th iteration)

*: start of sentence
#: end of sentence



# One hot Encoded Vocabulary and Training

Output Vocabulary

| WORD | a | am | I | thanks | student | <eos> |
|---|---|---|---|---|---|---|
| INDEX | 0 | 1 | 2 | 3 | 4 | 5 |

One-hot encoding of the word “am”

| | | | | | |
|---|---|---|---|---|---|
| 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |

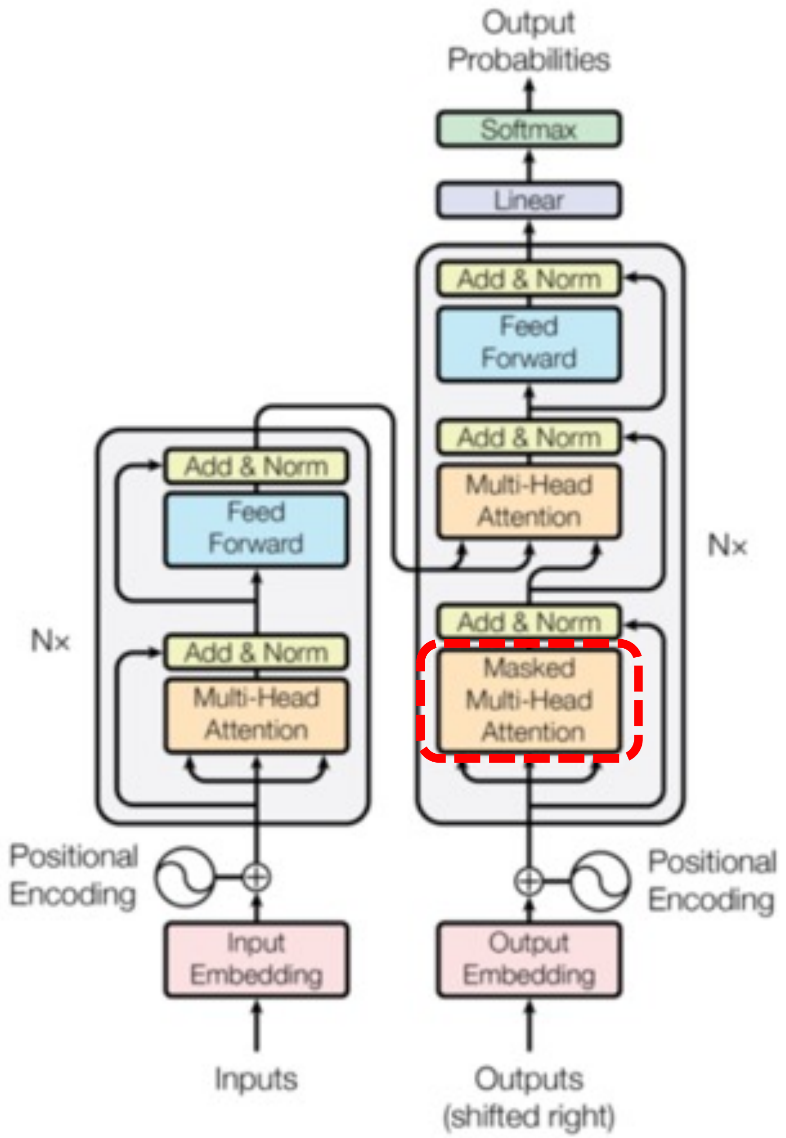
## Trained Model Outputs

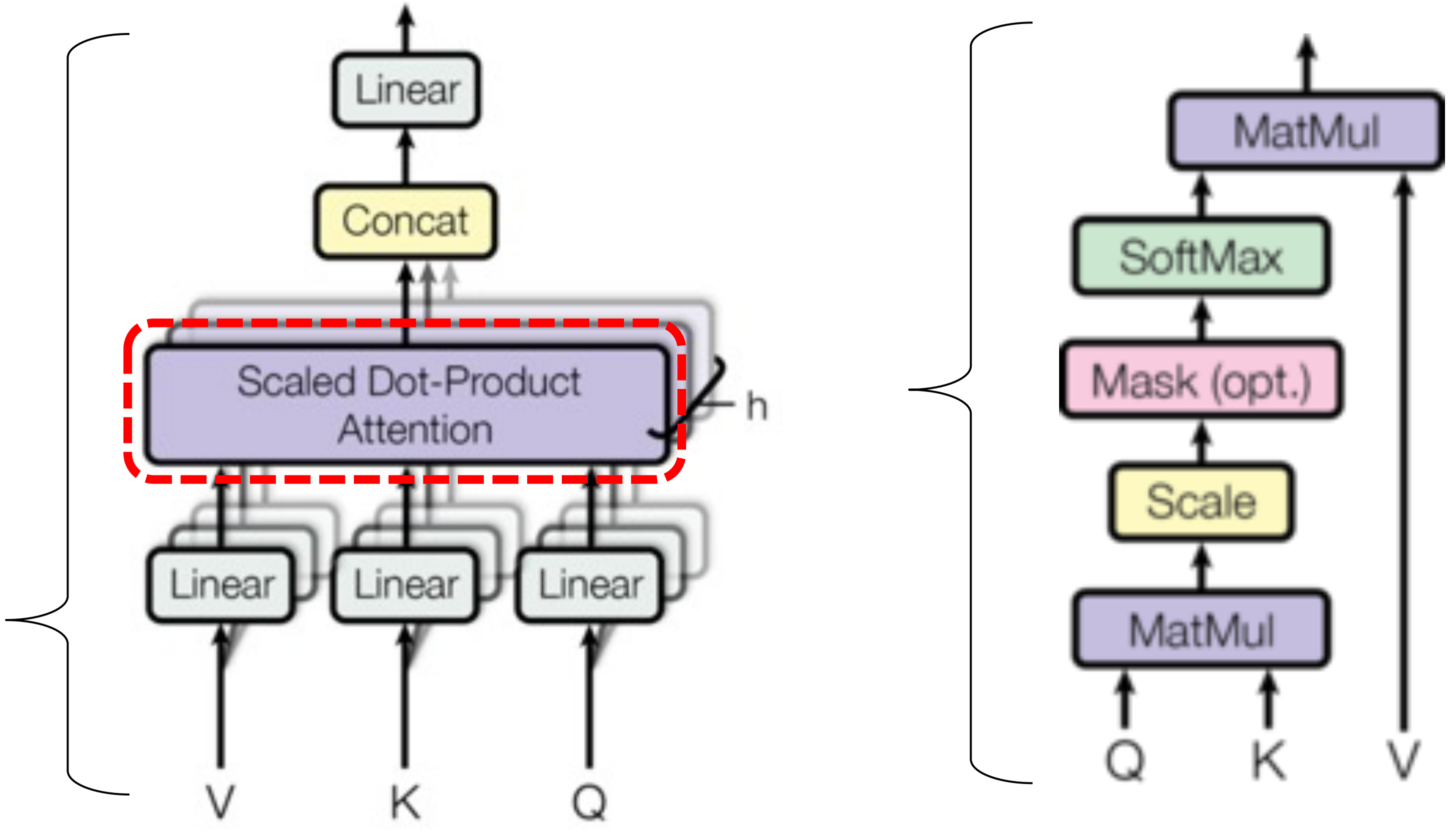
| Output Vocabulary: | a | am | I | thanks | student | <eos> |
|---|---|---|---|---|---|---|
| position #1 | 0.01 | 0.02 | 0.93 | 0.01 | 0.03 | 0.01 |
| position #2 | 0.01 | 0.8 | 0.1 | 0.05 | 0.01 | 0.03 |
| position #3 | 0.99 | 0.001 | 0.001 | 0.001 | 0.002 | 0.001 |
| position #4 | 0.001 | 0.002 | 0.001 | 0.02 | 0.94 | 0.01 |
| position #5 | 0.01 | 0.01 | 0.001 | 0.001 | 0.001 | 0.98 |
| | a | am | I | thanks | student | <eos> |

## Target Model Outputs

| Output Vocabulary: | a | am | I | thanks | student | <eos> |
|---|---|---|---|---|---|---|
| position #1 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| position #2 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position #3 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| position #4 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 |
| position #5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| | a | am | I | thanks | student | <eos> |

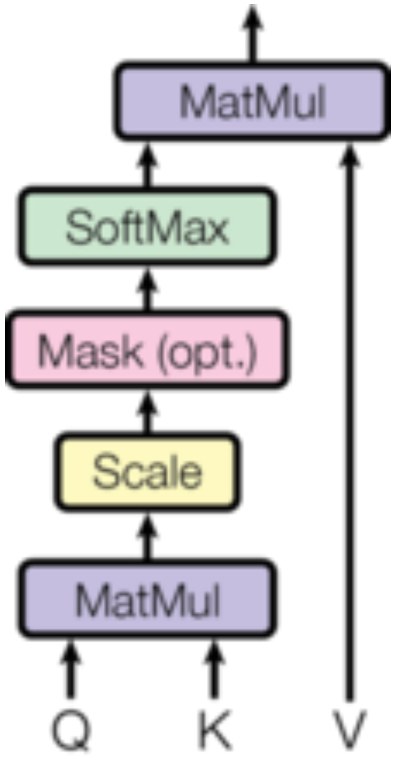
# Attention Layer



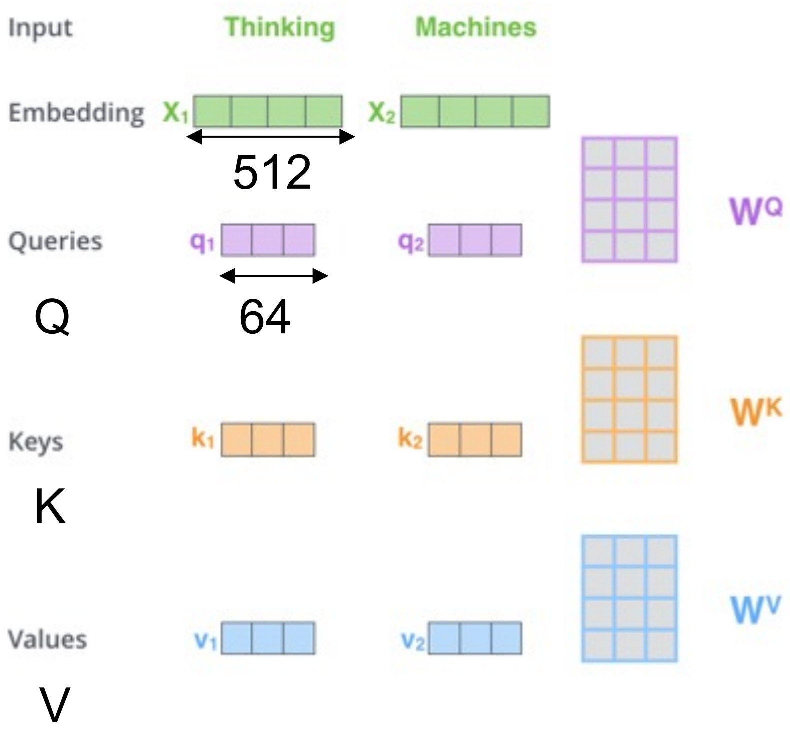




$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

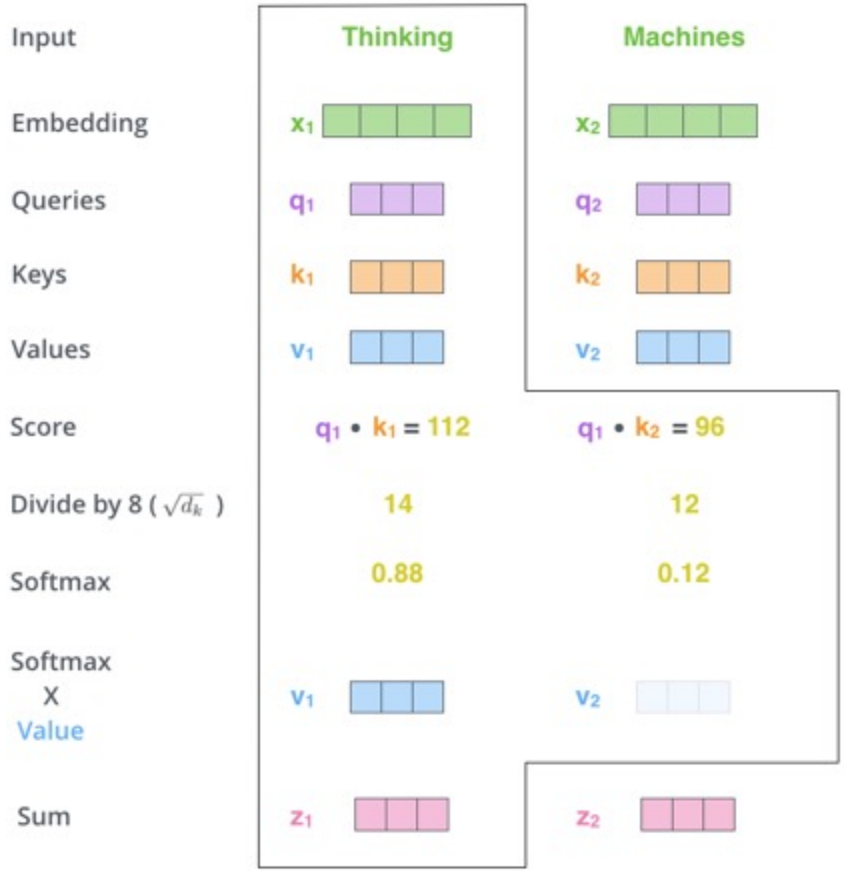
# Attention Layer (Single head)




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



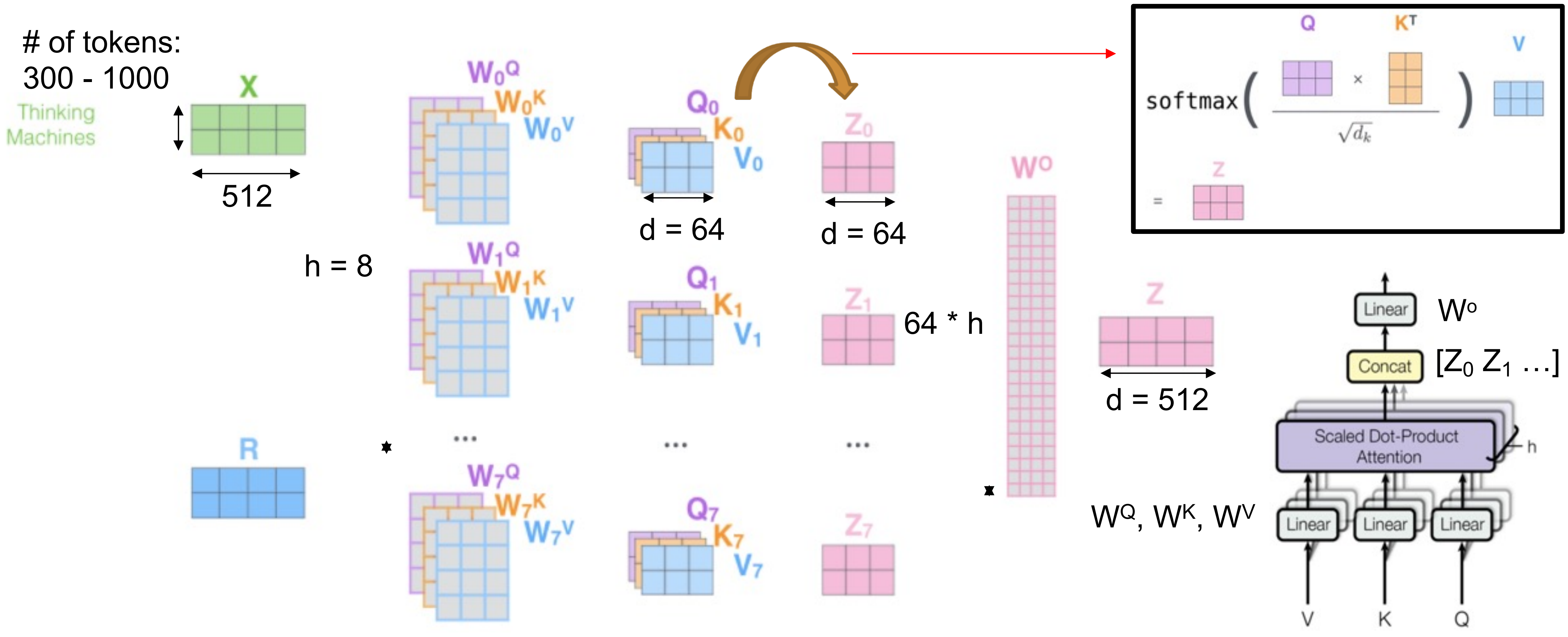

Q,V,K generation



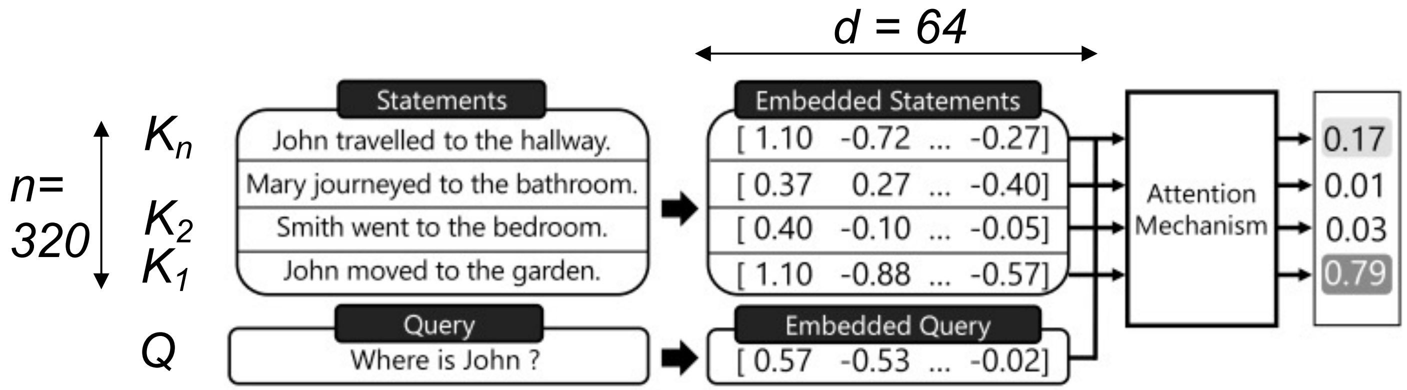

https://jalammar.github.io/illustrated-transformer/

# Attention Layer (Multi-head)

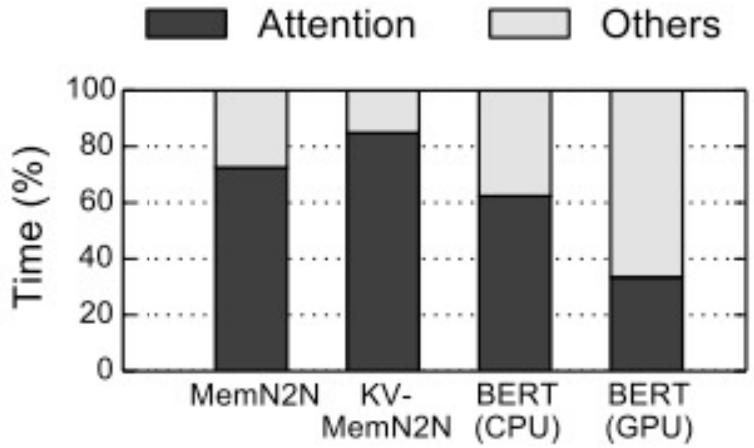
- Matrix processing for multi-head parallelism



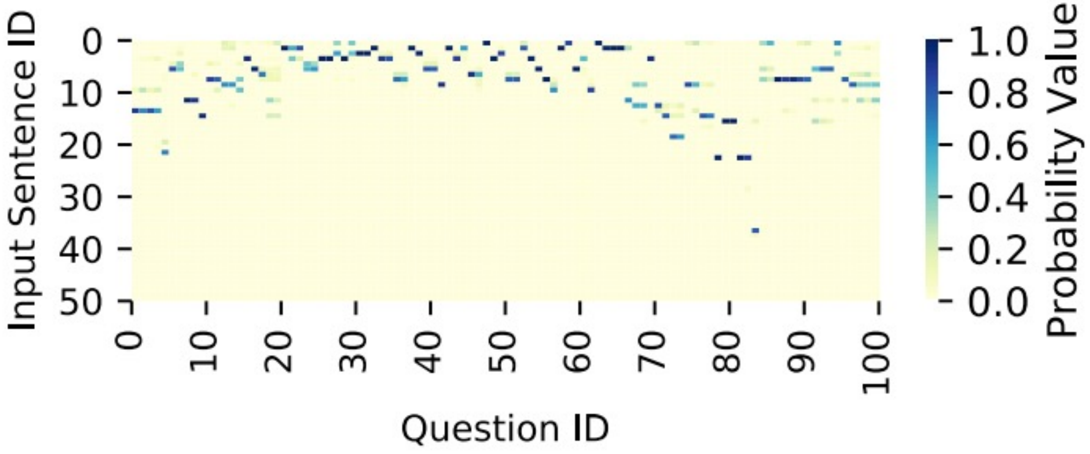
# Hardware Accelerator Example




Facebook bAbi QA task processing




**Question-Answering Time**




H. Jang, “MnnFast: A Fast and Scalable System Architecture for Memory-Augmented Neural Networks”, ISCA19

TJ. Ham, “A3: Accelerating Attention Mechanisms in Neural Networks with Approximation”, HPCA20

# Data Flow and Bit Precision

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

*V

Preventing overflow

Q*K

(4bit, 8bit)

(4bit, 8bit)

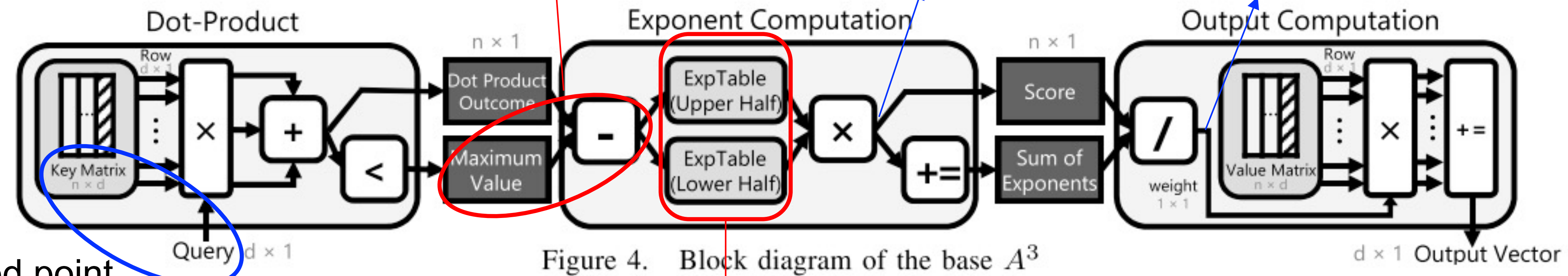


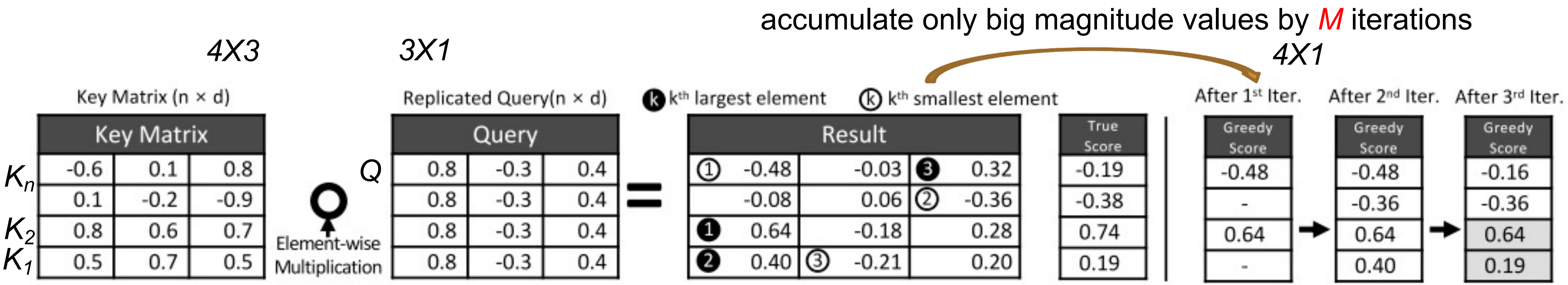

Figure 4. Block diagram of the base $A^3$

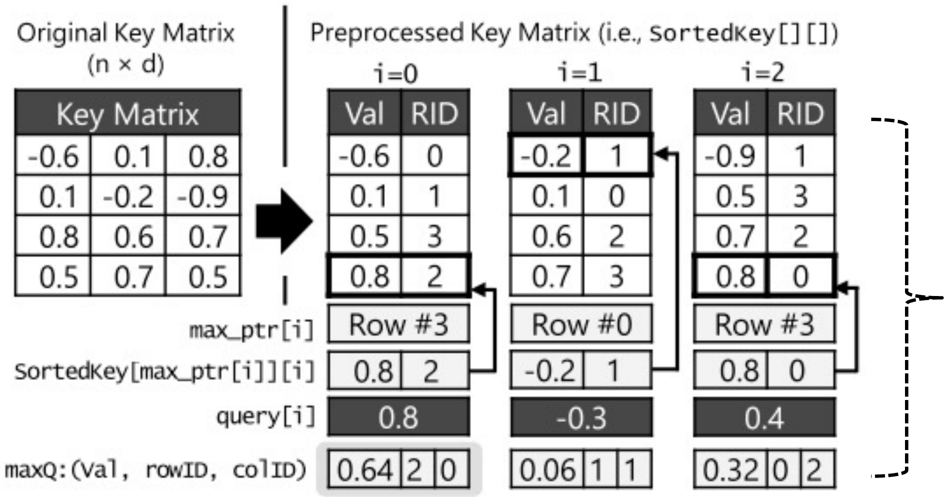
8 bit fixed point
(4bit integer, and 4bit fraction)

(4bit, 12bit)

$$e^{0.\underbrace{10101111}_{\text{8bit lut}}{}_2} = e^{0.\underbrace{1010}_{\text{4bit lut}}0000_2} \times e^{0.0000\underbrace{1111}_{\text{4bit lut}}{}_2}$$

# Approximated Computation

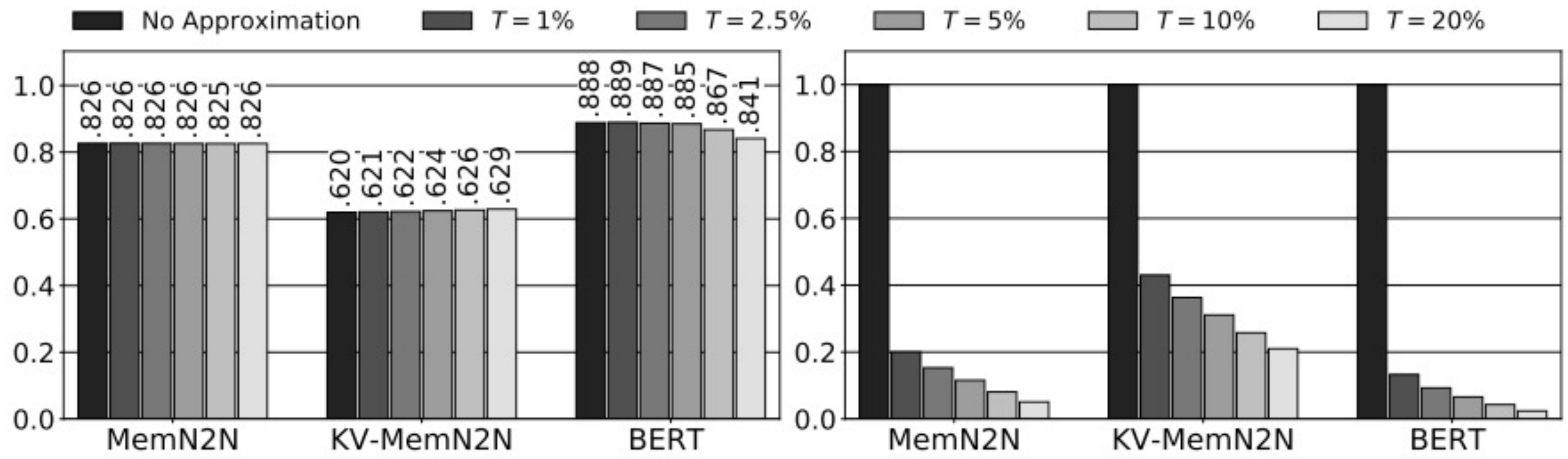


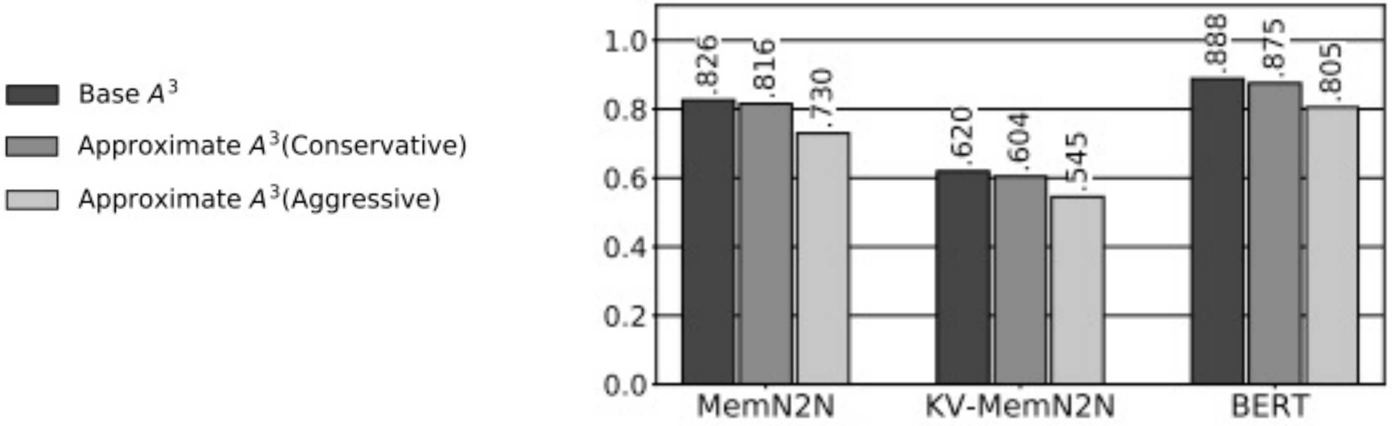

To minimize unnecessary "Result" computation, use sorted column

Most will be near zero after softmax, thus sort only a few important rows. Here, sort only the row's score > *T* % of max score based on post-softmax score

# Accuracy vs. Threshold



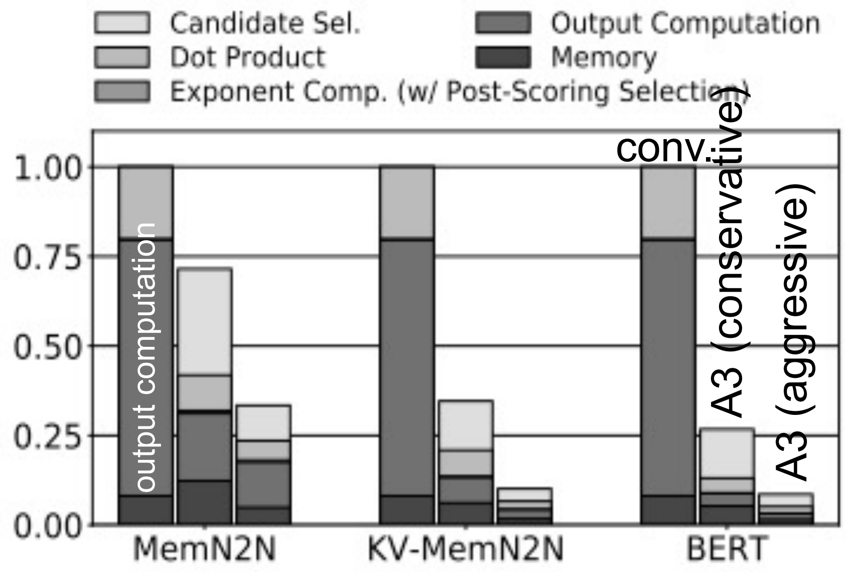

(a) End-to-End Accuracy (b) Normalized # of Selected Entries




(a) End-to-End Accuracy

Conservative: M = 1/2n and T = 5%
Aggressive: M = 1/8n and T = 10%

# Energy Breakdown




(b) Normalized Energy Breakdown

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

- candidate selection: the M iterative stage to sort big products

```
// key: n × d, value: n × d, query, output: d
float[] attention_mechanism (float key[][],
  float value[][], float query[]) {
  /* Step 1 : Dot-Product Computation */
  for i = 0 to n:
    sum = 0
    for j = 0 to d:
      sum += key[i][j] * query[j]
    dot_product[i] = sum
  /* Step 2 : Softmax Computation */
  score = softmax(dot_product)
  /* Step 3 : Output Computation */
  for j = 0 to d:
    sum = 0
    for i = 0 to n:
      sum += score[i] * value[i][j]
    output[j] = sum
  return output
}
float[] softmax(float input[]) {
  sum = 0
  for i = 0 to n:
    sum += exp(input[i])
  for i = 0 to n:
    output[i] = exp(input[i]) / sum
  return output
}
```