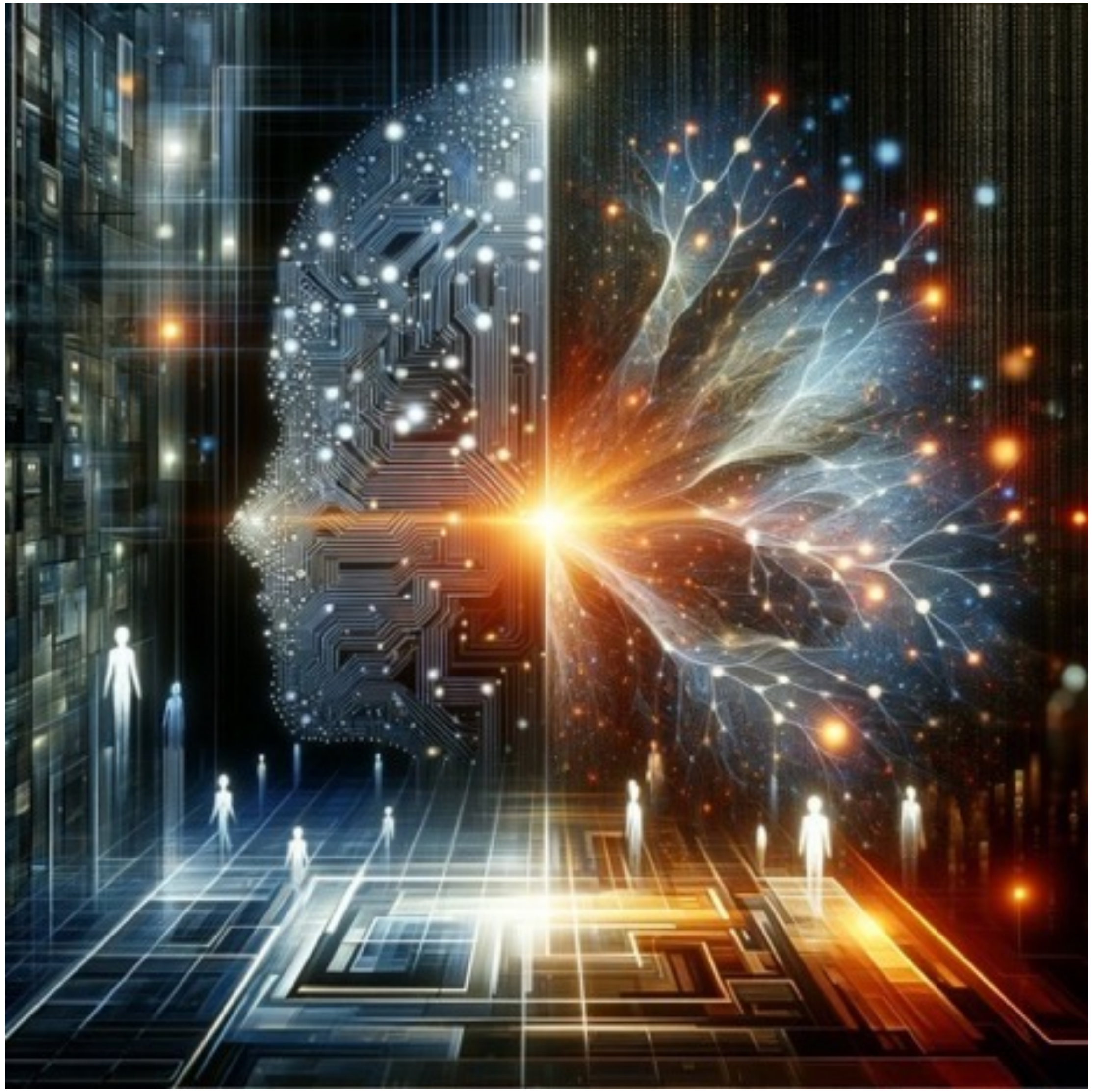
Understanding Large Language Models

Marco Giunti

Università di Cagliari

giunti@unica.it

FAIR-QC 2025 | Cagliari, April 4, 2025

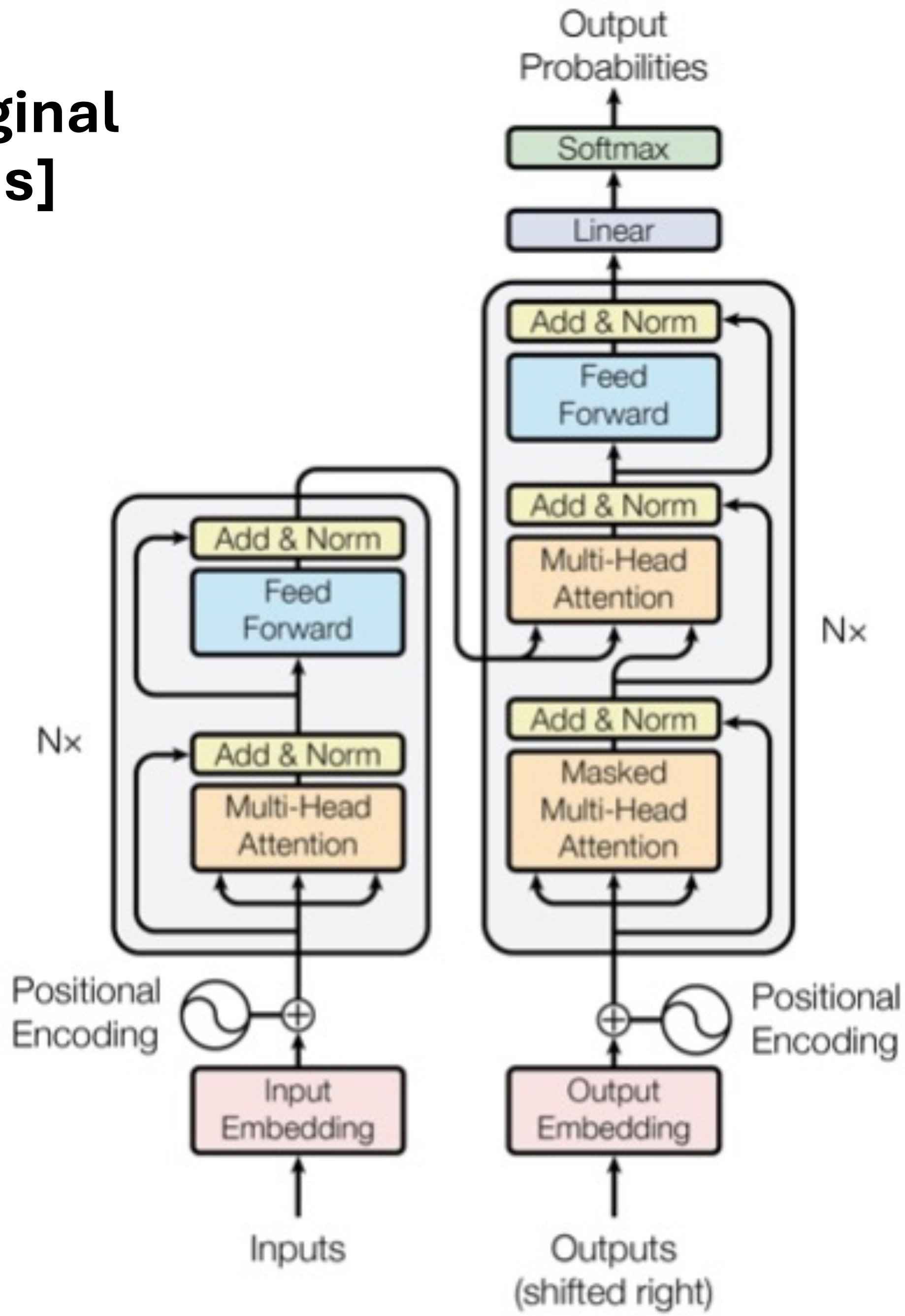


Summary

- Transformer Architecture (2017) ⟾ Large Language Models (LLMs) ⟾ A new era for AI: unprecedented achievements in
  - linguistic competence, logical reasoning, creative problem-solving, contextual understanding, synthesis of complex knowledge.
- Focus on decoder-only Transformers during the inference phase (i.e., generation of responses in normal operation):
  - ➢They are not purely statistical systems.
- A better interpretation of the model's structure and functioning:
  - ➢A Transformer is a **complex system of simple neural networks**, **interconnected in a non-standard way**, which **transforms concepts through concepts**;
  - ➢the **transforming concepts** are the **simple networks**;
  - ➢the **non-standard interconnections implement dynamic transforming concepts**, which are **not determined only by training**.

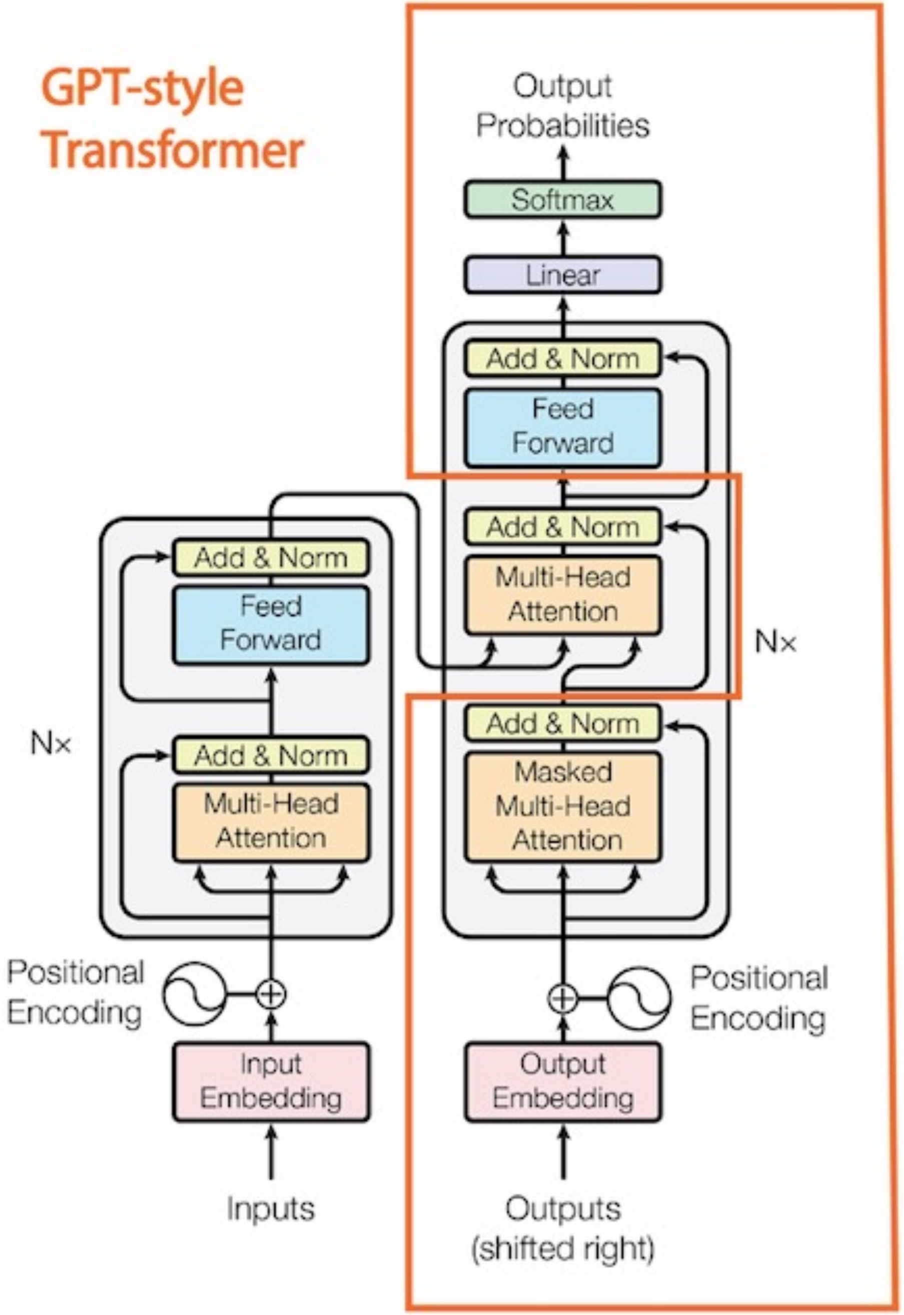
The Encoder-Decoder Architecture of the Original Transformer (Vaswani et al., 2017) [Google Labs]

- The **encoder** is on the left side of the figure, while the **decoder** is on the right.
- The large gray section in both the encoder and decoder represents the **transformer stack**, composed of **N** identical **transformer layers** connected in series.



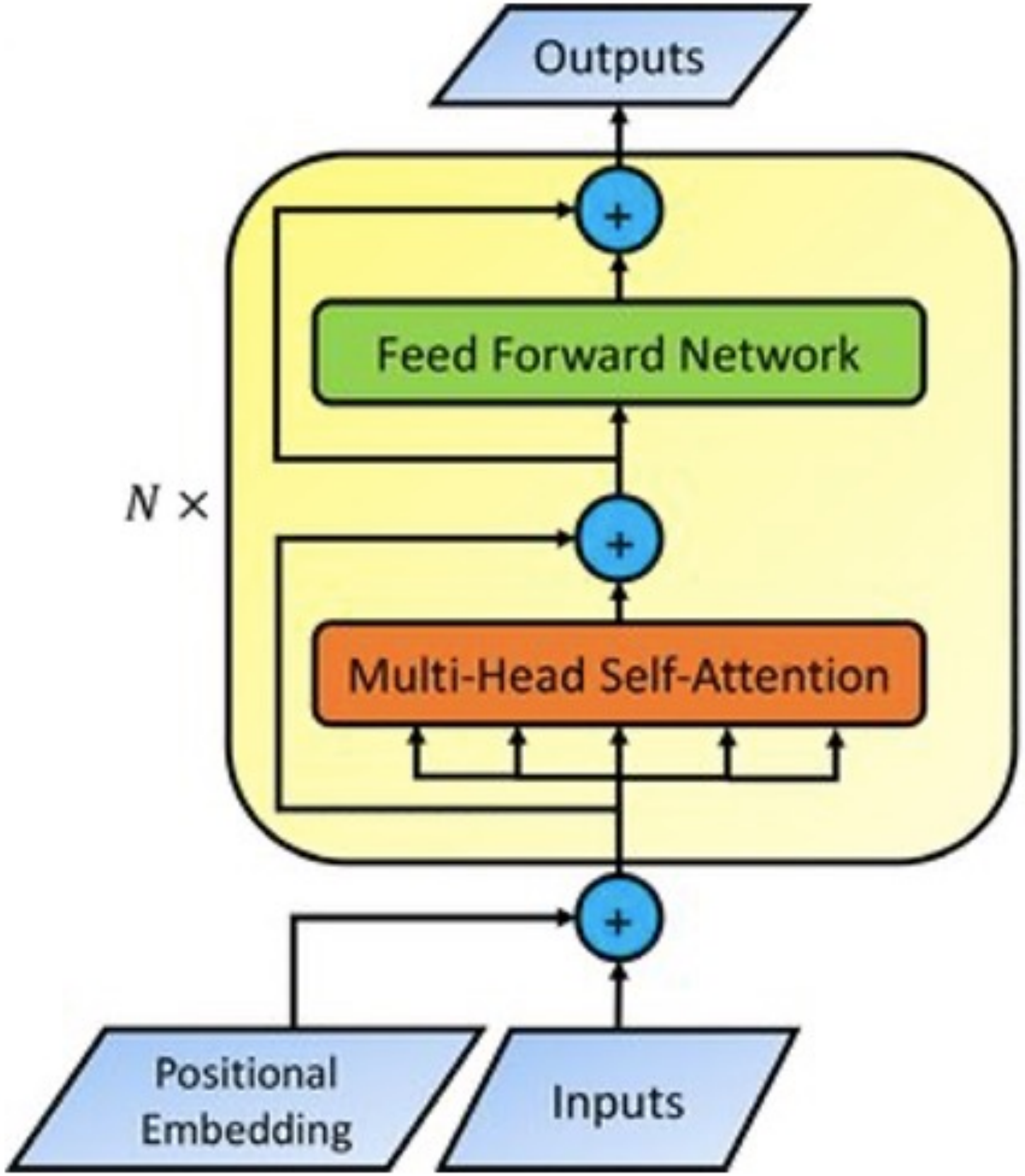
The Decoder-Only Architecture of GPT-Series Transformers and Similar Models

- The grey section in the orange box is the **transformer stack**, composed of **N** identical **transformer layers**.
- The section before the transformer stack is the **input module**, while the two boxes after the transformer stack form the **output module**.



GPT-3-175B

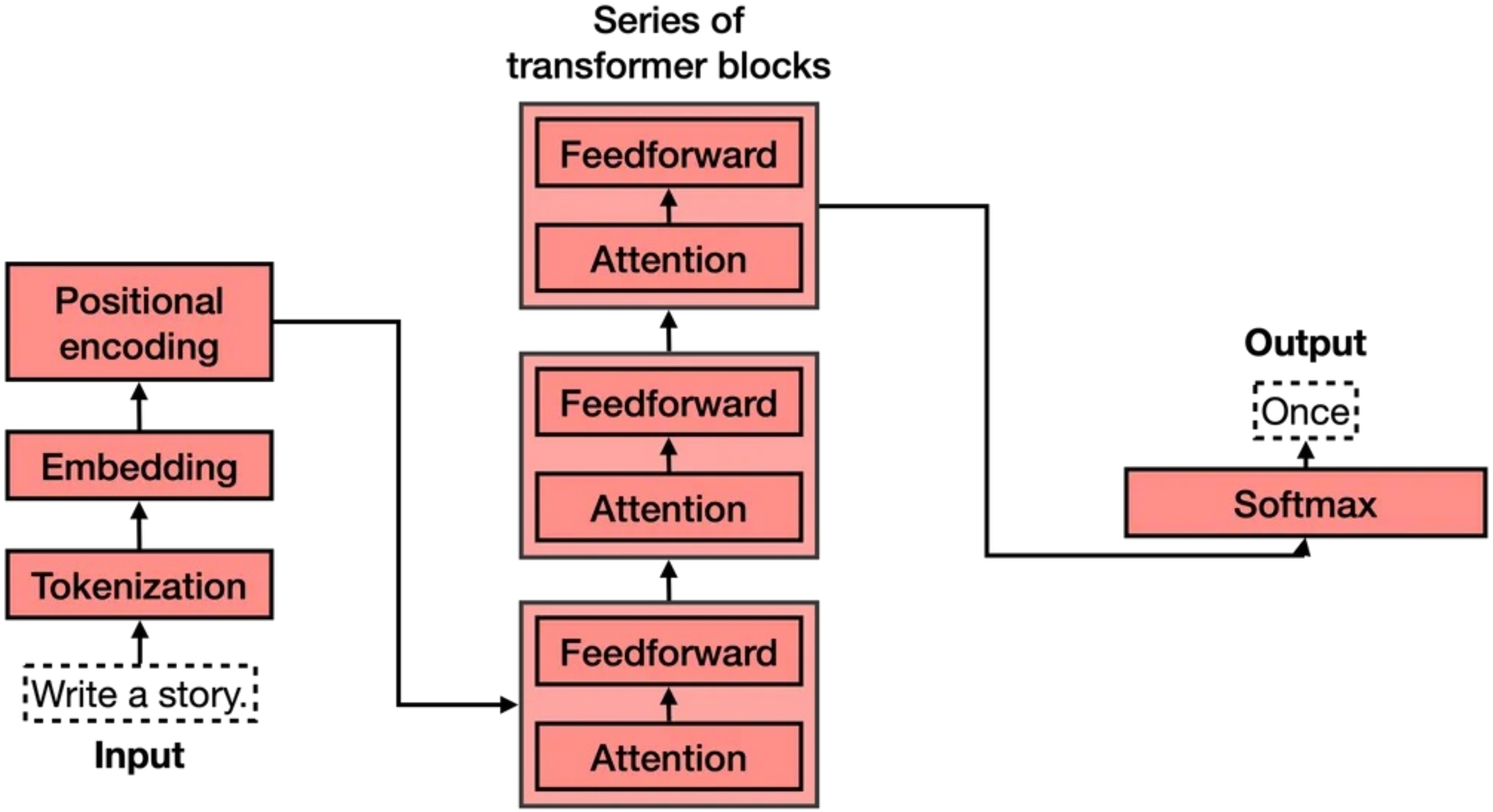
- d_{voc} = 50257 [# tokens in vocabulary]
- d_{model} = 12288 [token vector dim]
- $d_k = d_v$ = 128 [key and query vector dim = value vector dim]
- a = 4, $a d_{model} = d_f$ [hidden dim]
- N = 96 [# transformer layers]
- h = 96 [# attention-heads in each layer]   h = N, $h = d_{model} / d_v$
- n_{max} = 2k tokens, where k = 1024 [context window]   GPT4: n_{max} = 8k – 128k tokens
- max batch ≈ 3,2 million tokens [about 1500 prompts in parallel]
- # trainable parameters ≈ 175 billion [matrix weights + embedding vector elements + scaling and bias vector elements in norm and feed forward modules]
- training texts ≈ 500 billion words



A Transformer is designed to generate text by continuing from a given text called «a prompt» 1/2

- The **prompt** is initially transformed into a **sequence of token-vectors** (a token corresponds to a word or a piece of a word) that constitutes the model's initial input.
- The model **processes all the input in parallel** and then **generates the next token** based on the entire token sequence that makes up the input.
- The **multi-head attention mechanism** allows these models to consider the entire input sequence to generate the token that, according to the model, represents the best continuation of that input.

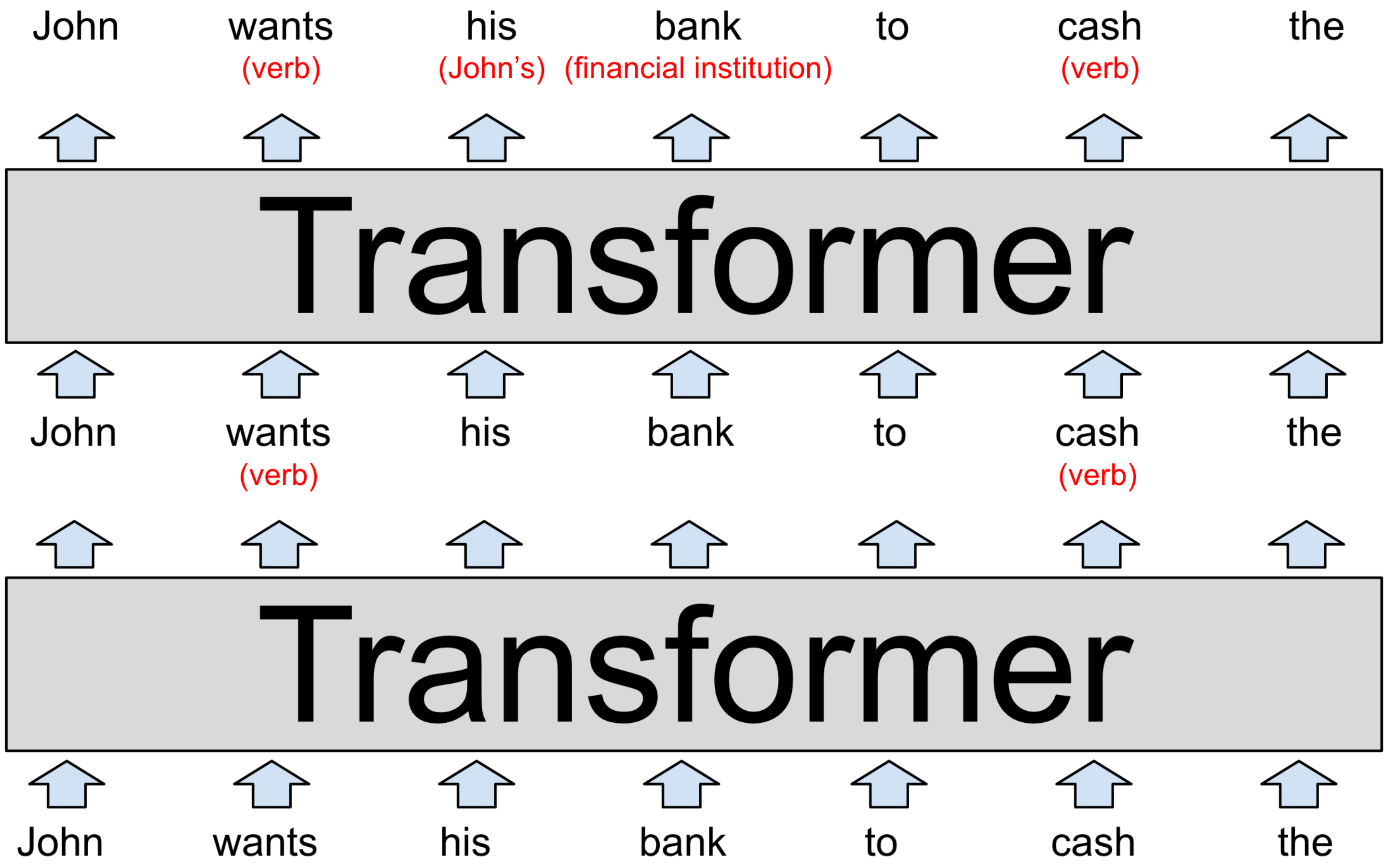
Basic processing cycle of a Transformer system



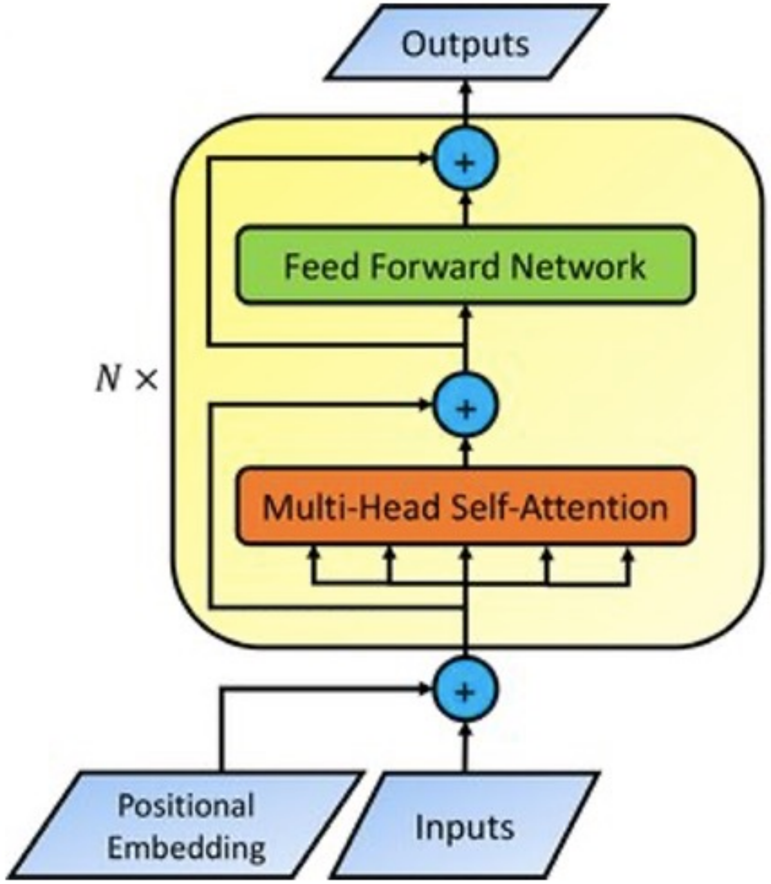
A Transformer is designed to generate text by continuing from a given text called «a prompt» 2/2

- The process is then **iterated**, feeding the model the previous input with the added generated token.
- **Generation ends** when the model produces the special token <**EOS**> (End Of Sequence). Alternatively, an external system to the Transformer may be in place, evaluating the generated sequence at each iteration cycle and deciding whether to continue or stop the generation.
- **The final output** of the entire generative process **is then transformed back into text**, which constitutes the response to the initial prompt.

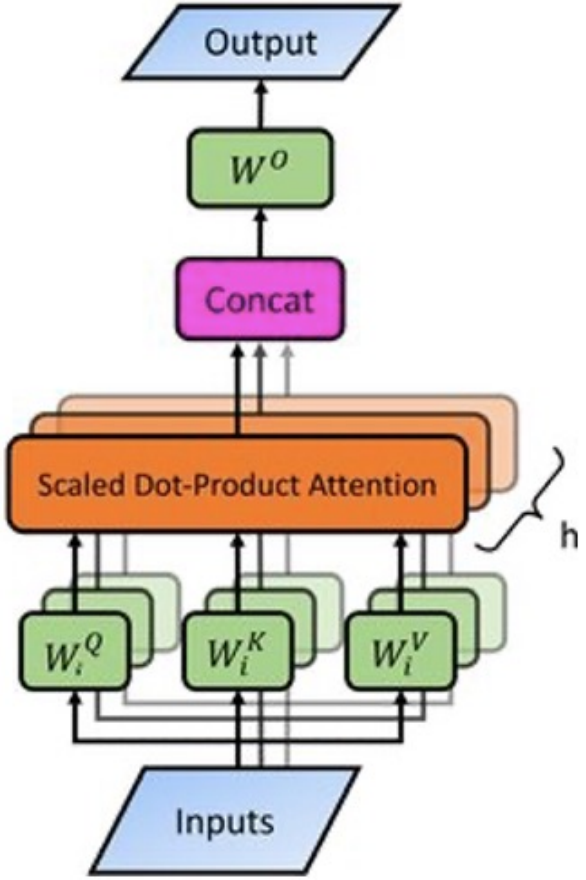
Transforming token (≈ word) vectors into predictors of the next word by adding context



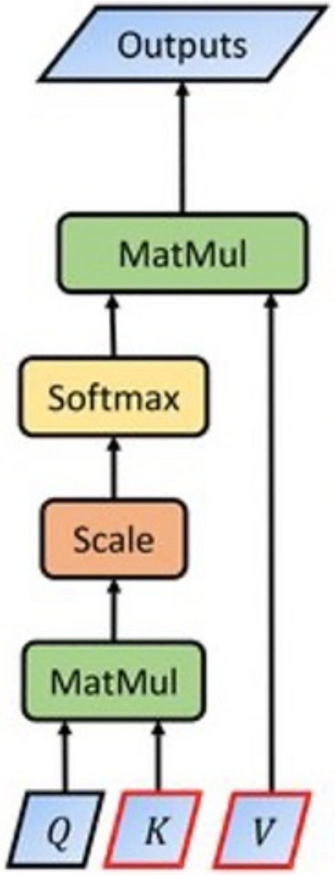
a) Transformer system



b) Multi-Head Self-Attention



c) Scaled Dot Product Attention



The concepts used and transformed by a Transformer system

- The best way to interpret the overall functioning of a Transformer is to think of it as a system that **transforms concepts through concepts**.
- The **transformee concepts** are the **vector representations of each token** that are **transformed by other concepts**; they are
  - **trained** := the initial **embedding vectors** (made of **trained parameters** of the model)
  - **dynamic** := those gradually **generated** during the processing.
- The **transforming concepts** modify, enrich, and refine these representations through successive transformations; they are
  - **trained** := the **weight matrices** and the **positional encoding, bias, and scaling vectors** of the Transformer (all these are made of **trained parameters**)
  - **dynamic** := other **weight matrices** and **vectors generated** during the processing, namely: the **value matrices** and **transposed key matrices** of each **attention head**, and the **output vectors** of each **multi-head attention module** and **feed forward** module.

The three types of vector transformations operated by transforming concepts

- **Transforming concepts** are of two types:
  a) simple $:=$ **vector**
  b) complex $:=$ **matrix**

a) vector $\boldsymbol{v} = (v_1, \dots, v_n)$ transformed by **simple concept** $\boldsymbol{c} = (c_1, \dots, c_n)$

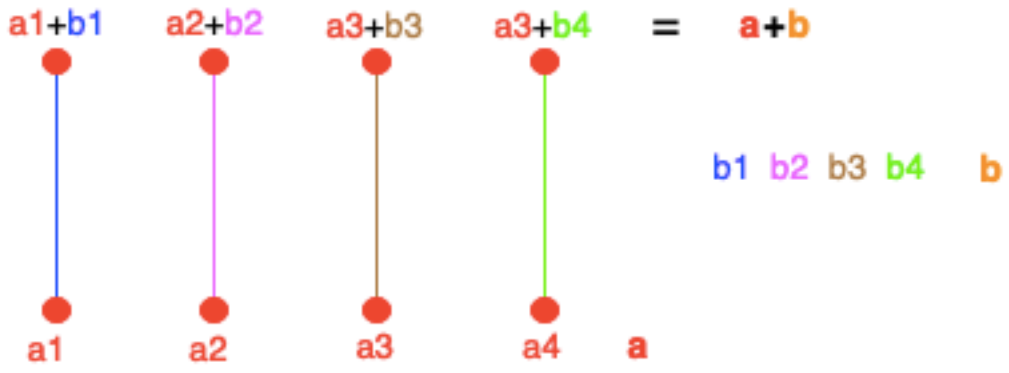
1. **VECTOR SUM:** $\boldsymbol{v} + \boldsymbol{c} \coloneqq (v_1 + c_1, \dots, v_n + c_n)$
2. **HADAMARD VECTOR PRODUCT:** $\boldsymbol{v} \odot \boldsymbol{c} \coloneqq (v_1 c_1, \dots, v_n c_n)$

b) vector $\boldsymbol{v} = (v_1, \dots, v_n)$ transformed by **complex concept** $\boldsymbol{C} = \begin{pmatrix} c_{11}, \dots, c_{1m} \\ c_{21}, \dots, c_{2m} \\ . \ , \dots , \ . \\ . \ , \dots , \ . \\ . \ , \dots , \ . \\ c_{n1}, \dots, c_{nm} \end{pmatrix}$

3. **VECTOR-MATRIX PRODUCT:** $\boldsymbol{vC} \coloneqq (u_1, \dots, u_m)$, where u_i is the dot product of $\boldsymbol{v}$ with column i of $\boldsymbol{C}$, that is:

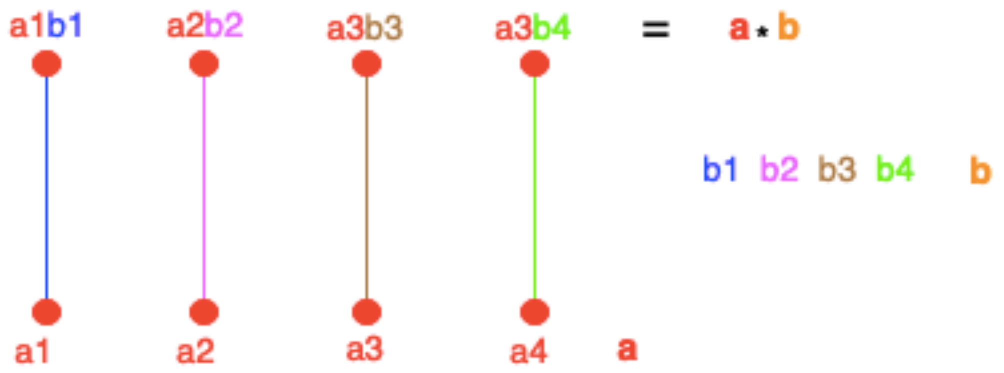
$$\boldsymbol{vC} \coloneqq (u_1, \dots, u_m) = \begin{pmatrix} v_1 c_{11} +, \dots, v_1 c_{1m} + \\ v_2 c_{21} +, \dots, v_2 c_{2m} + \\ . \quad +, \dots, \quad . \quad + \\ . \quad +, \dots, \quad . \quad + \\ . \quad +, \dots, \quad . \quad + \\ v_n c_{n1} \quad , \dots, v_n c_{2m} \end{pmatrix} = \begin{array}{l} (v_1 c_{11}, \dots, v_1 c_{1m}) + \\ (v_2 c_{21}, \dots, v_2 c_{2m}) + \\ (\quad . \quad , \dots, \quad . \quad) + \\ (\quad . \quad , \dots, \quad . \quad) + \\ (\quad . \quad , \dots, \quad . \quad) + \\ (v_n c_{n1}, \dots, v_n c_{nm}) \end{array} = \begin{array}{l} v_1 (c_{11}, \dots, c_{1m}) + \\ v_2 (c_{21}, \dots, c_{2m}) + \\ . \ (\ . \ , \dots, \ . \ ) + \\ . \ (\ . \ , \dots, \ . \ ) + \\ . \ (\ . \ , \dots, \ . \ ) + \\ v_n (c_{n1}, \dots, c_{nm}) \end{array}$$

One-to-one additive network = Vector sum



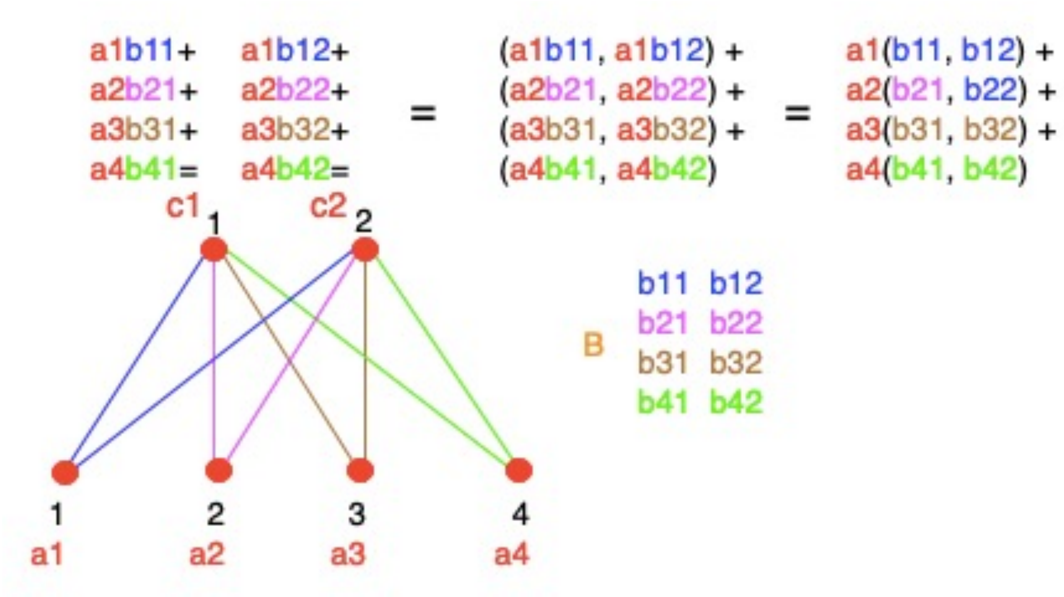
- Trained transforming concepts:
  - **positional encoding vectors** in input module
  - **bias vectors** in **feed forward** modules
  - **bias vectors** in **norm** modules
- Dynamic transforming concepts:
  - **output vectors** of each **multi-head attention module** (residual connection)
  - **output vectors** of each **feed forward** module (residual connection)

One-to-one multiplicative network = Hadamard vector product



- Trained transforming concepts:
  - **scaling vectors** in **norm** modules

Densely connected linear network = Vector-matrix product = Linear transformation



- Trained transforming concepts:
  - for each attention head i, the three weight matrices $\mathbf{W}_i{}^{Q}$, $\mathbf{W}_i{}^{K}$, $\mathbf{W}_i{}^{V}$
  - for each multi-head attention module, the projection matrix $\mathbf{W}^{O}$
  - for each feed-forward module the two matrices $\mathbf{W}^{1}$ and $\mathbf{W}^{2}$
  - the matrix $\mathbf{W}^{L}$ of the output module
- Dynamic transforming concepts:
  - the value matrix V_i of each attention head i
  - the transposed key matrix $K_i^{\mathrm{T}}$of each attention head i

How can you get new transforming concepts without new training?

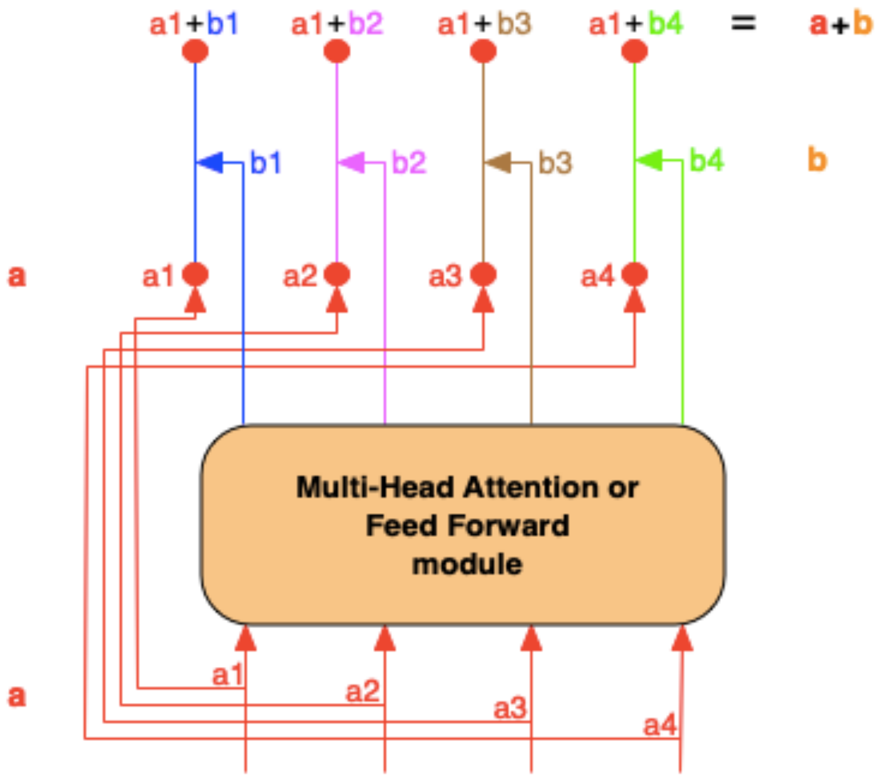
1. Any transforming concept **C** is either a vector (simple) or a matrix (complex).
2. If **C** is a vector, its components are the weights of a one-to-one network, either additive or multiplicative.
3. If **C** is matrix, its components are the weights of a densely connected linear network.
4. In neural networks weights are changed only by training.

➢ Therefore, you cannot get new transforming concepts without new training. Or, in other words: dynamic transforming concepts are impossible.

Premise 4 is false in Transformers

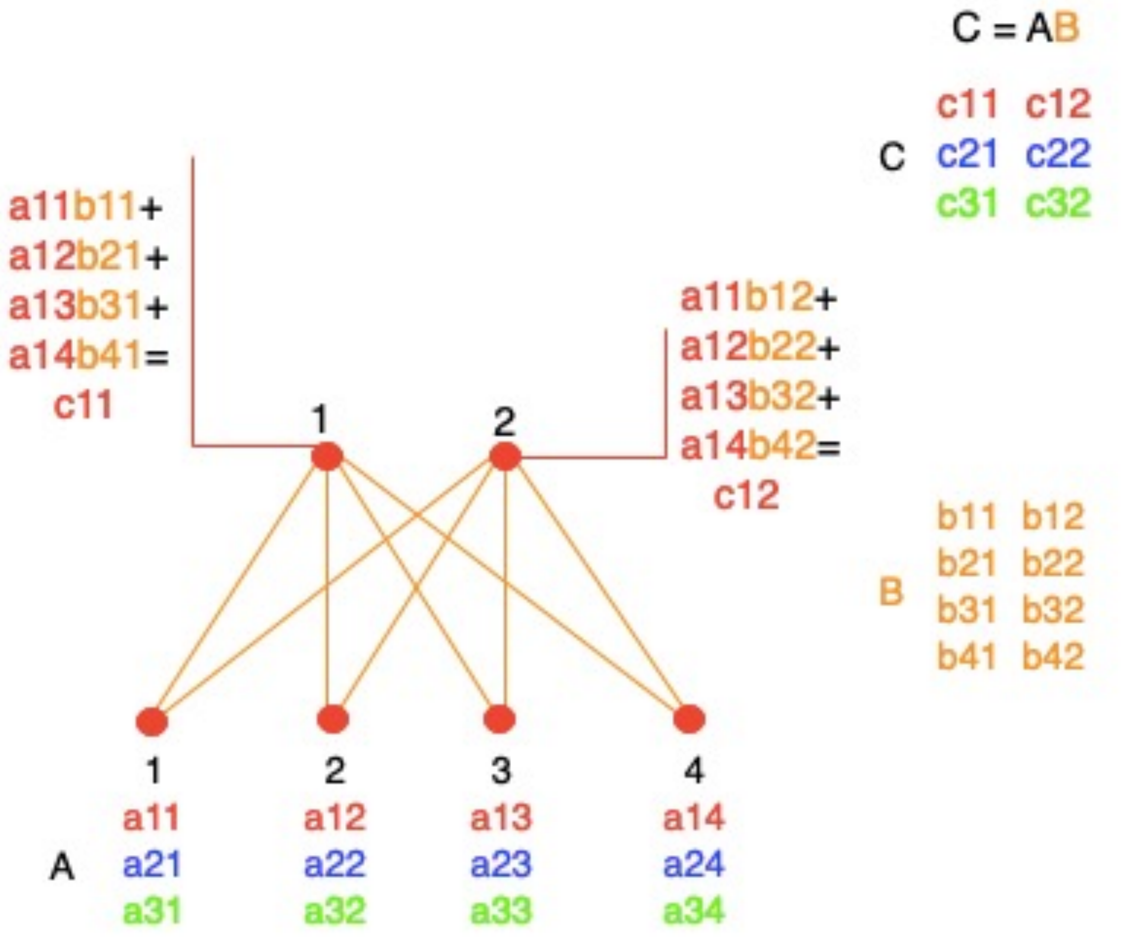
- In **standard** neural network systems, the only way of **connecting two networks** is by using the **outputs** of a network as **inputs** of the other.
- But in Transformers the **outputs** of some networks are used as **weights** of other networks.
- Thus, these **non-standard connections** between different networks allow for **weight change without training**, making **dynamic transforming concepts** possible.
- There are **two types** of non-standard connections:
  - **residual connections:** each output vector of a multi-head attention or feed-forward module becomes the weight vector of a corresponding one-to-one additive network;
  - **cross connections** in each attention head implement complex dynamic transforming concepts.

Residual connections implement simple dynamic transforming concepts



Vector b is one of the outputs of either a multi-head attention module or a feed-forward module; a is the corresponding input vector of that module.

Matrix multiplication AB = C ≈ Application of linear transformation B to each row of matrix A



C = AB

$$C \quad \begin{array}{cc} c11 & c12 \\ c21 & c22 \\ c31 & c32 \end{array}$$

a11b11+
a12b21+
a13b31+
a14b41=
c11

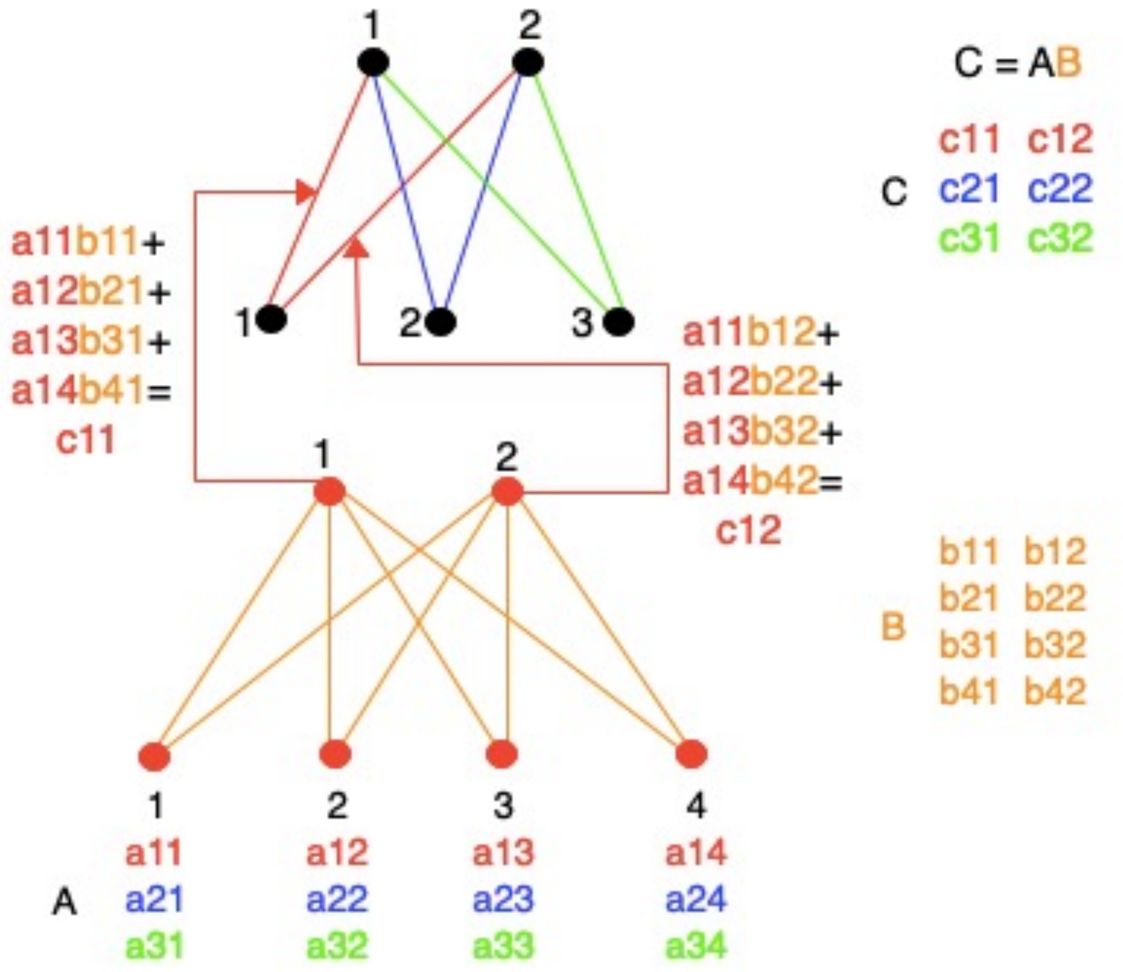
a11b12+
a12b22+
a13b32+
a14b42=
c12

$$B \quad \begin{array}{cc} b11 & b12 \\ b21 & b22 \\ b31 & b32 \\ b41 & b42 \end{array}$$

$$A \quad \begin{array}{cccc} 1 & 2 & 3 & 4 \\ a11 & a12 & a13 & a14 \\ a21 & a22 & a23 & a24 \\ a31 & a32 & a33 & a34 \end{array}$$

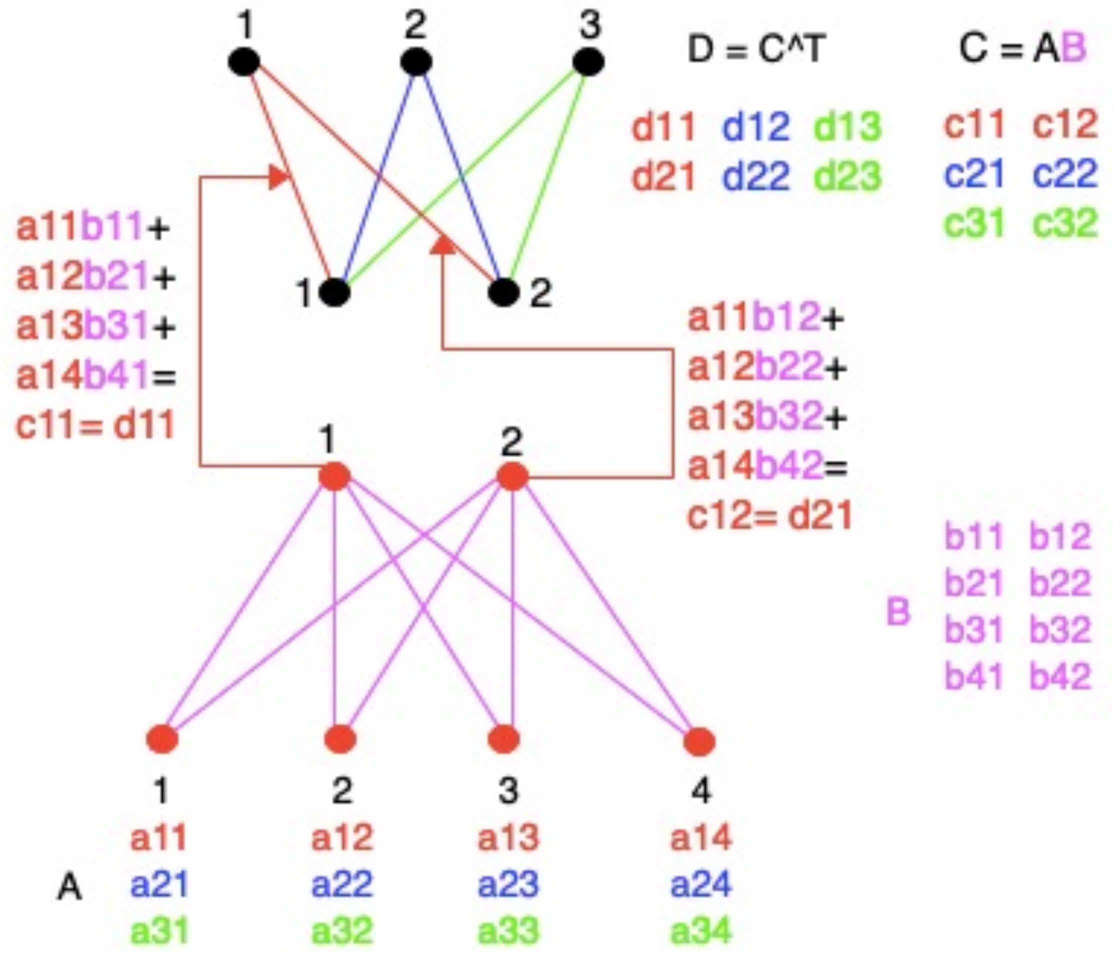
Matrix C ≈ New linear transformation whose rows depend on the corresponding rows of A

To get all weights of network C we need 2 more copies of network B, with output links to blue and green connections in C, respectively.



Matrix D = $C^T \approx$ New linear transformation whose columns depend on the corresponding rows of A

To get all weights of network D we need 2 more copies of network B, with output links to blue and green connections in D, respectively.



D = C^T

d11 d12 d13
d21 d22 d23

C = AB

c11 c12
c21 c22
c31 c32

a11b11+
a12b21+
a13b31+
a14b41=
c11= d11

a11b12+
a12b22+
a13b32+
a14b42=
c12= d21

B

b11 b12
b21 b22
b31 b32
b41 b42

A

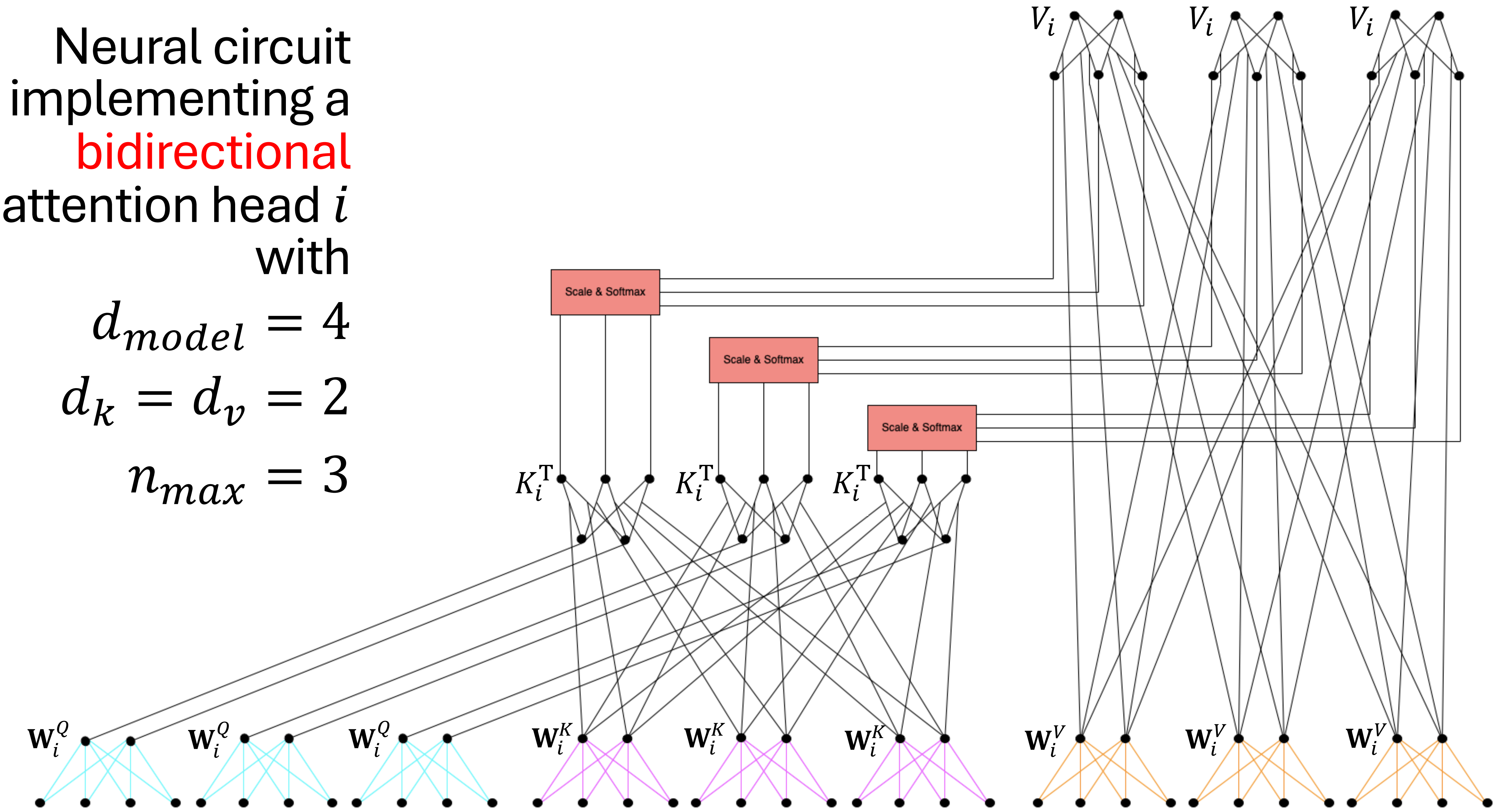
a11 a12 a13 a14
a21 a22 a23 a24
a31 a32 a33 a34

Neural circuit implementing a bidirectional attention head i with

$$d_{model} = 4$$

$$d_k = d_v = 2$$

$$n_{max} = 3$$

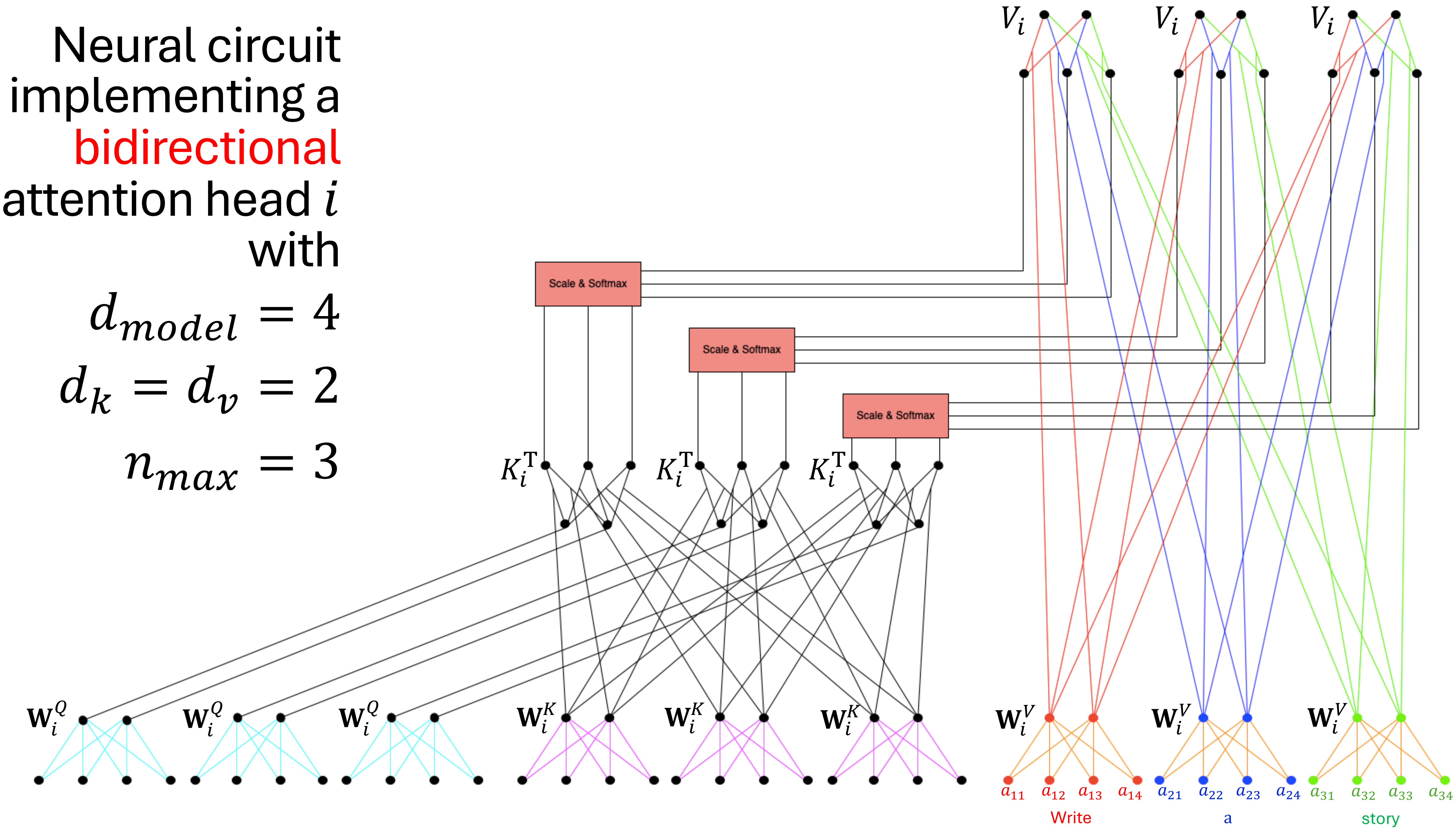


Neural circuit implementing a bidirectional attention head i with

$$d_{model} = 4$$

$$d_k = d_v = 2$$

$$n_{max} = 3$$

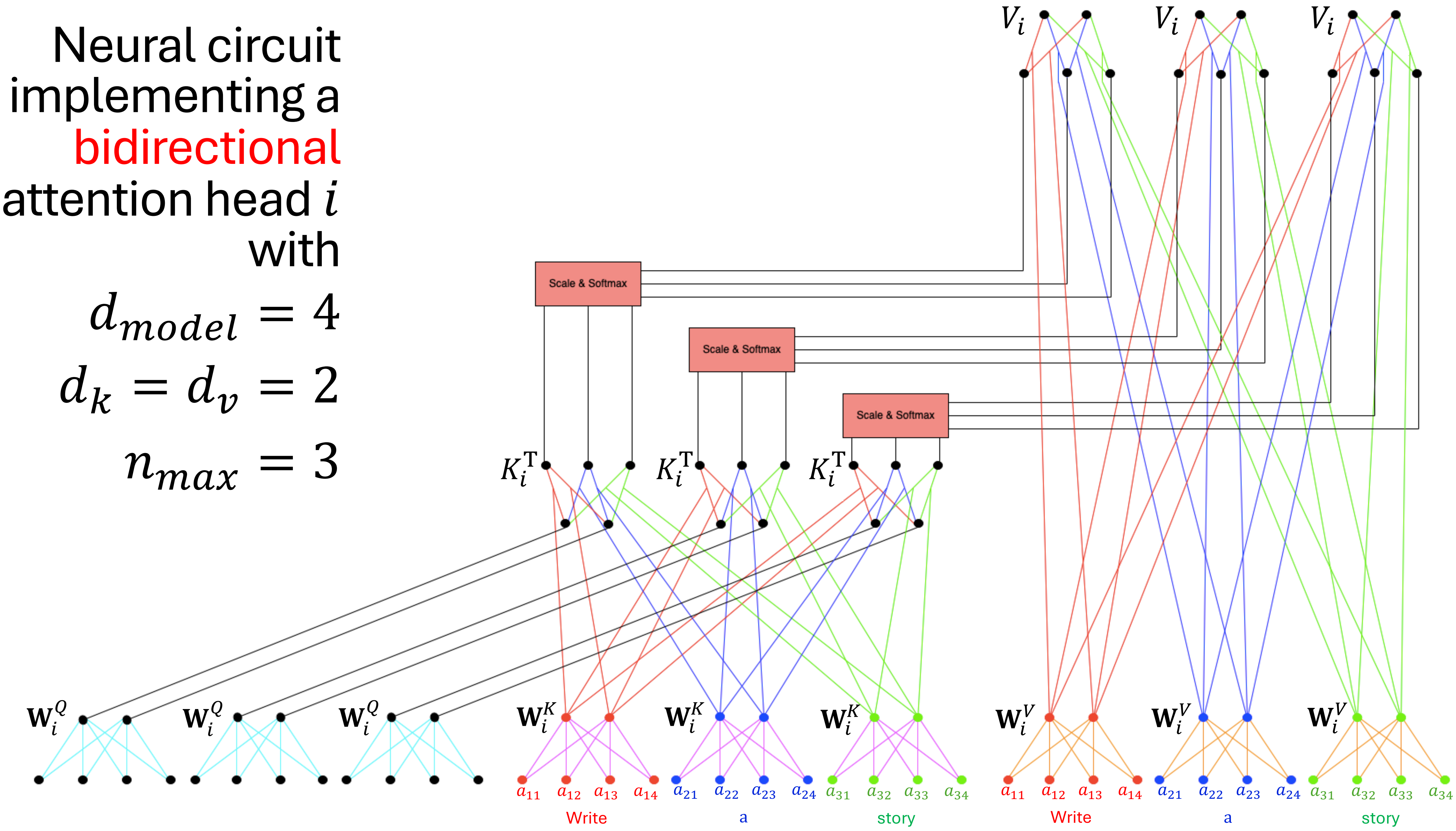


Neural circuit implementing a bidirectional attention head i with

$$d_{model} = 4$$

$$d_k = d_v = 2$$

$$n_{max} = 3$$

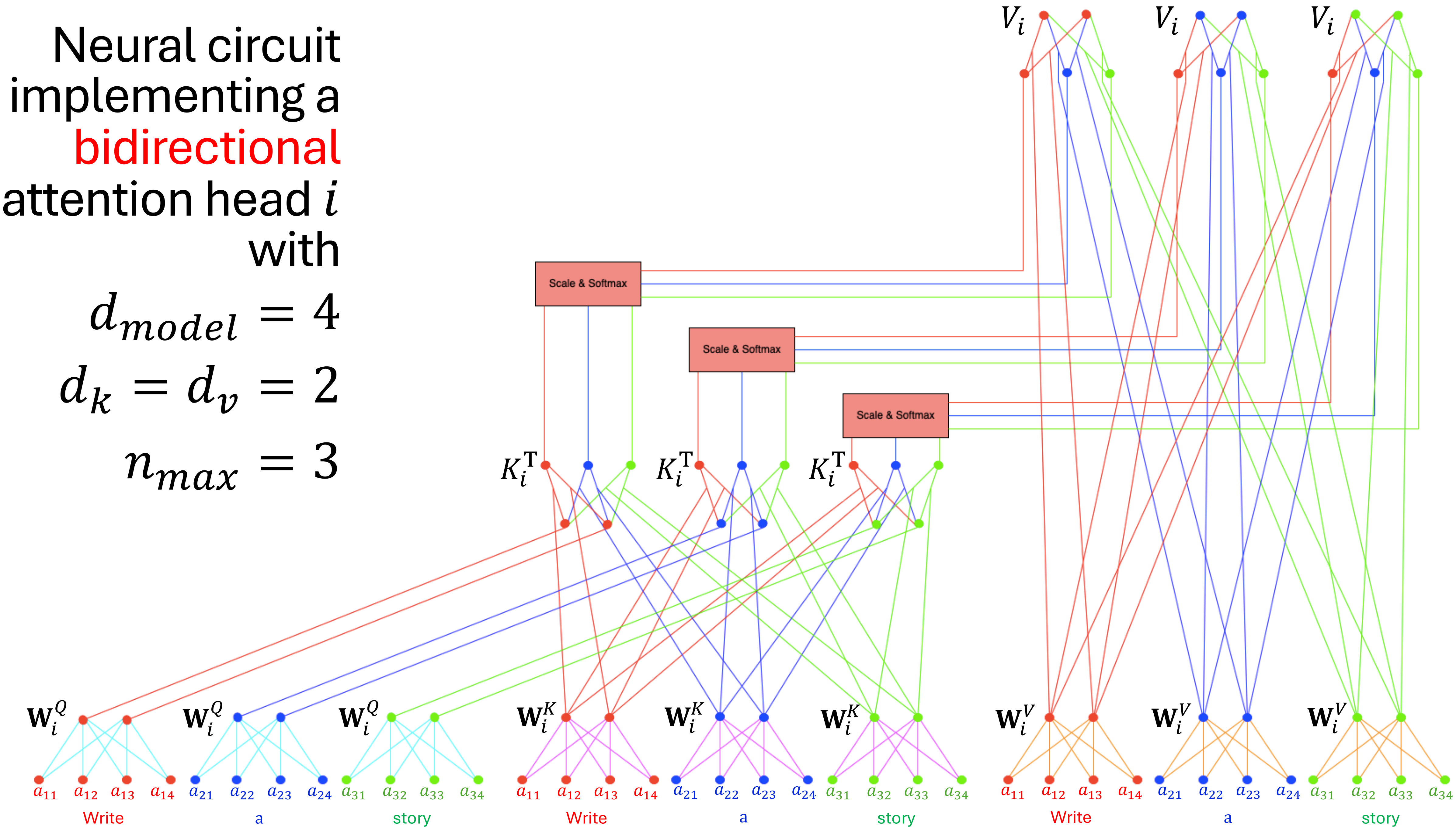


Neural circuit implementing a bidirectional attention head i with

$$d_{model} = 4$$

$$d_k = d_v = 2$$

$$n_{max} = 3$$



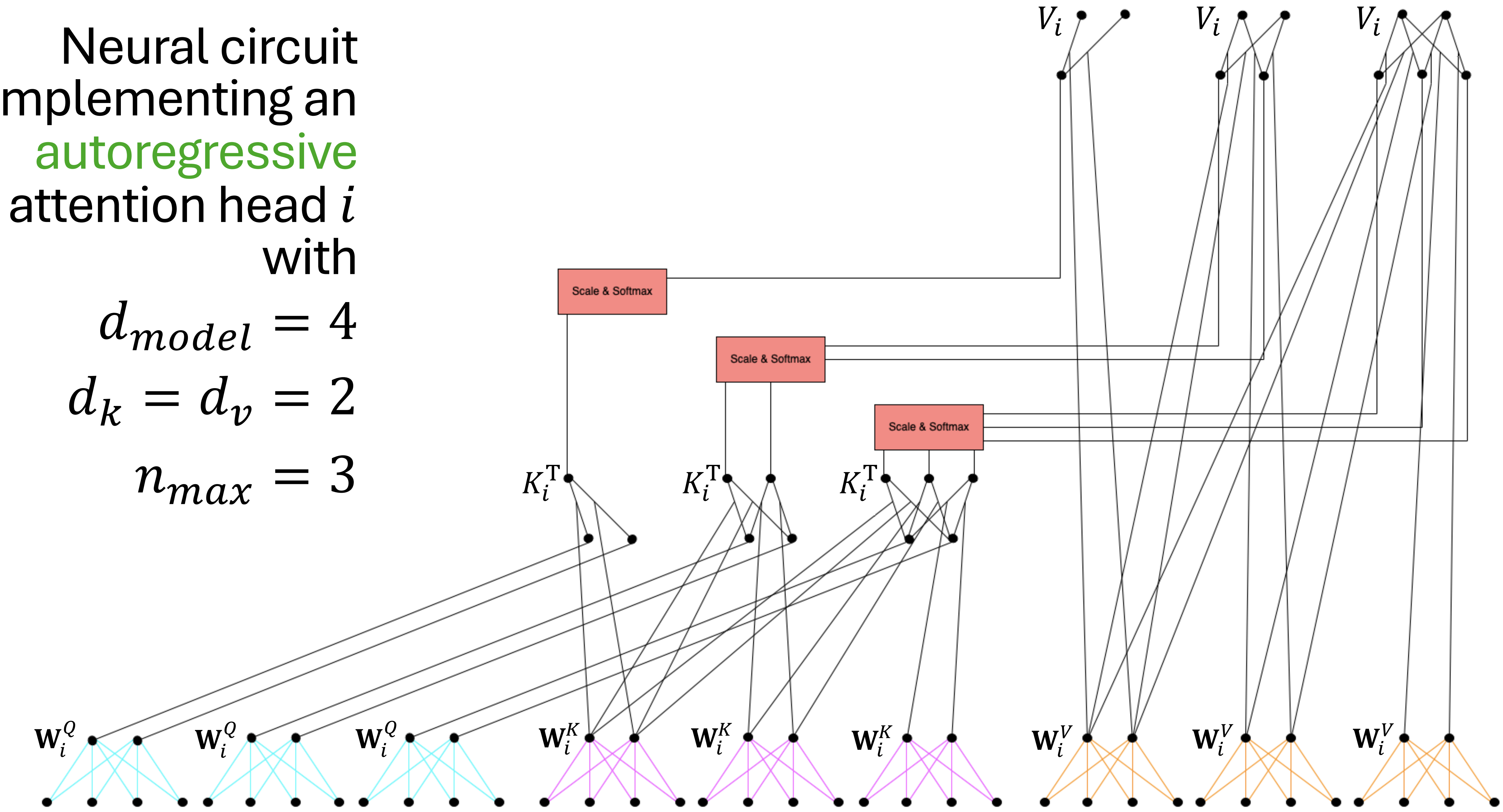
V_i
V_i
V_i
Scale & Softmax
Scale & Softmax
Scale & Softmax
K_i^{T}
K_i^{T}
K_i^{T}
$\mathbf{W}_i^Q$
$\mathbf{W}_i^Q$
$\mathbf{W}_i^Q$
$\mathbf{W}_i^K$
$\mathbf{W}_i^K$
$\mathbf{W}_i^K$
$\mathbf{W}_i^V$
$\mathbf{W}_i^V$
$\mathbf{W}_i^V$
a_{11} a_{12} a_{13} a_{14}
Write
a_{21} a_{22} a_{23} a_{24}
a
a_{31} a_{32} a_{33} a_{34}
story
a_{11} a_{12} a_{13} a_{14}
Write
a_{21} a_{22} a_{23} a_{24}
a
a_{31} a_{32} a_{33} a_{34}
story
a_{11} a_{12} a_{13} a_{14}
Write
a_{21} a_{22} a_{23} a_{24}
a
a_{31} a_{32} a_{33} a_{34}
story

Neural circuit implementing an autoregressive attention head i with

$$d_{model} = 4$$

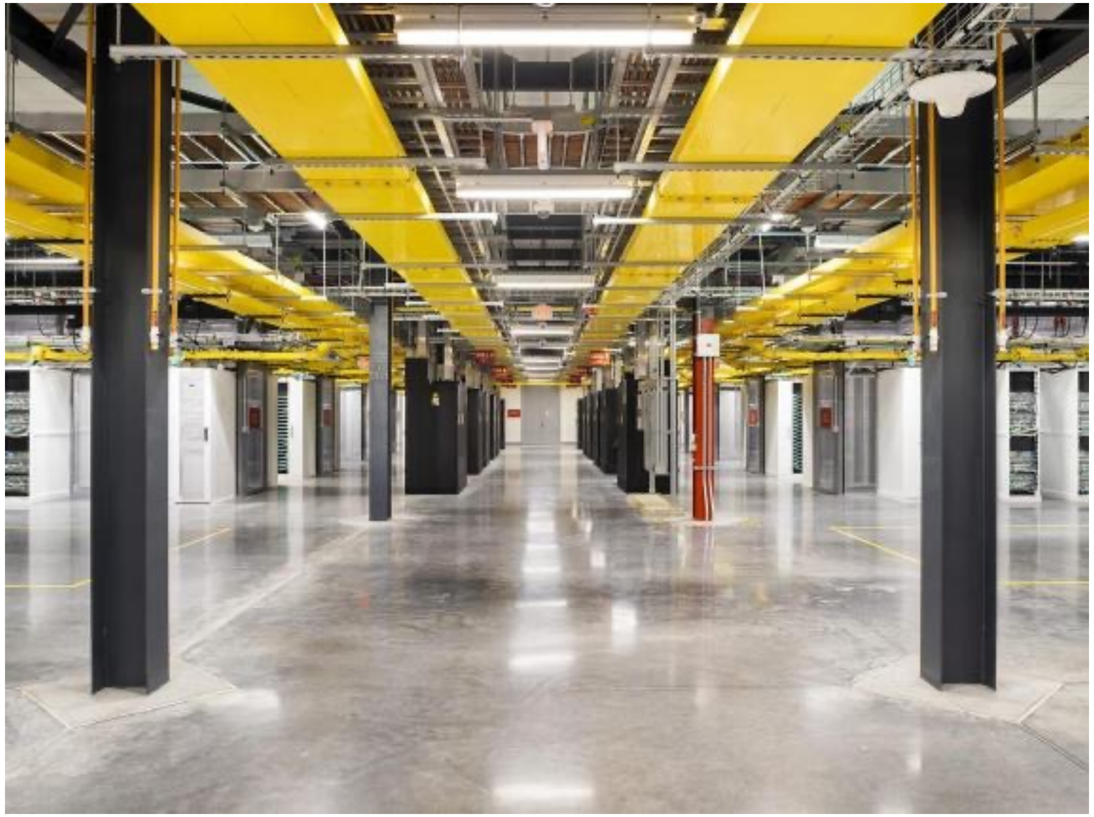
$$d_k = d_v = 2$$

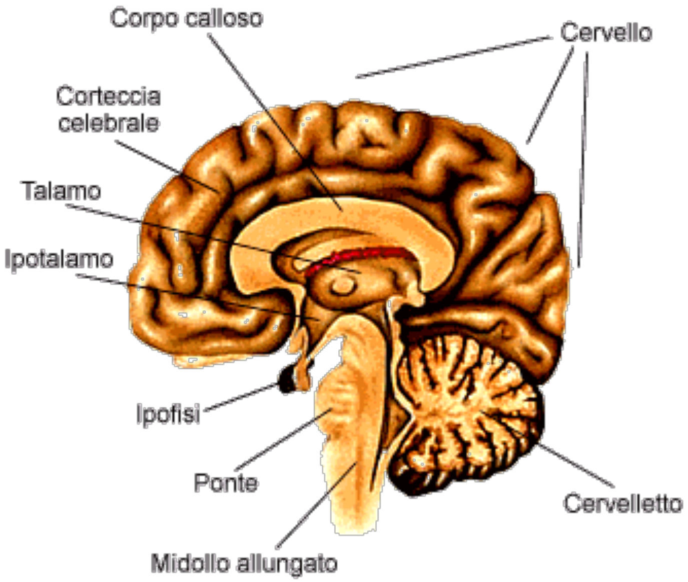
$$n_{max} = 3$$



Large Language Model: A system of neural nets SIMULATED by clusters of parallel computers in huge data centers

Human Cognitive Model: A system of neural nets HARDWIRED in our brain and body





GPT-3-175B	Humans
d_{voc} = 50257 [# tokens in vocabulary]	words known by average adult native speaker ≈ 50000
d_{model} = 12288 [token vector dim]	not comparable
$d_k = d_v$ = 128 [key and query vector dim = value vector dimension]	not comparable
N = 96 [# transformer layers]	not comparable
h = 96 [# attention-heads in each layer] h = N, h = d_{model} / d_v	simultaneous attention processes on the same input 1 – 10?
n_{max} = 2k tokens, where k = 1024 [context window] GPT4: 8k – 128k tokens	words attended to together by each attention process 1 – 100? 1000? More?
max batch ≈ 3,2 million tokens [about 1500 prompts in parallel]	simultaneous linguistic inputs 1-5? (Just one?)
# trainable parameters ≈ 175 billion (1:500 – 1:5000) [mostly neural network weights in transformer modules]	# synapses in the entire brain ≈ 100,000 – 1,000,000 billion (×500 – ×5000) # neurons in the entire brain ≈ 100 billion (1:2)
training texts ≈ 500 billion words (×5000)	a 10-year-old child encounters ≈ 100 million words (1:5000)

Microsoft Data Center for ChatGPT-4 (Des Moines, Iowa)

Energy and Environmental Impact
(Corriere della Sera, Jan 16, 2024)

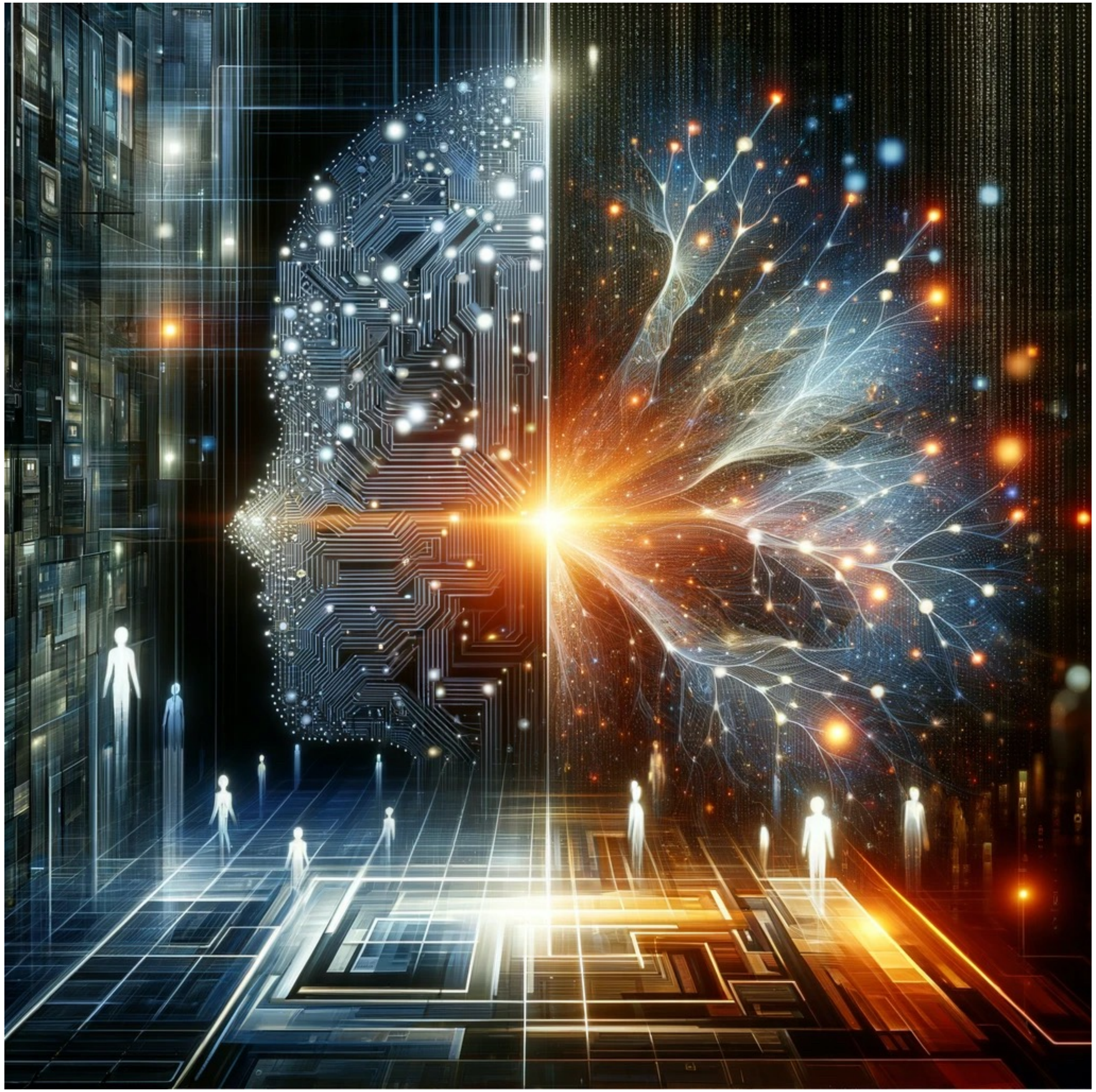
For cooling:

1 liter of potable water per
20 – 100 questions

100 mil queries per week
1 mil – 5 mil liters per week
(≈ 2 olympic-sized swimming pools)

2027: Estimates predict between 4.2 and 6.6 billion cubic meters per year, **≈ 4 times Denmark's annual water consumption.**

What to do?



A Look into the Future

- Reducing energy needs, environmental impact, and physical dimensions:
  - from SIMULATED systems of interconnected neural nets
    - huge centralized systems
  - to HARD-WIRED architectures of interconnected neural nets
    - small autonomous agents
- Developing a UNIFIED UNDERSTANDING of the PRINCIPLES at work in **complex cognitive systems of interconnected neural nets**, both ARTIFICIAL and NATURAL:
  - towards a **new alliance** between **Artificial Intelligence**, **Cognitive Science**, **Neuroscience**, **Philosophy**.

A Look into the Future

- Reducing energy needs, environmental impact, and physical dimensions:
  - from SIMULATED systems of interconnected neural nets
    - ➢ huge centralized systems
  - to HARD-WIRED architectures of interconnected neural nets
    - ➢ small autonomous agents
- Developing a UNIFIED UNDERSTANDING of the PRINCIPLES at work in **complex cognitive systems of interconnected neural nets**, both ARTIFICIAL and NATURAL:
  - towards a **new alliance** between **Artificial Intelligence**, **Cognitive Science**, **Neuroscience**, **Philosophy**.

Thanks!

Thanks!

Special thanks to ChatGPT, without whose valuable support this talk would not have been possible.

References

- Giunti M. (2025), Tutto quello che avreste voluto sapere su ChatGPT ma non avete mai osato chiedere. http://dx.doi.org/10.13140/RG.2.2.31444.62084/
- Lee T. B., Trott S. (2023). A jargon-free explanation of how AI large language models work. https://arstechnica.com/science/2023/07/a-jargon-free-explanation-of-how-ai-large-language-models-work/
- Saleh M., Paquelet S. (2024). Anatomy of neural language models. https://doi.org/10.48550/arXiv.2401.03797
- Stollnitz B. Blog. https://bea.stollnitz.com/blog/
- Stollnitz B. (2023). The Transformer architecture of GPT models. https://bea.stollnitz.com/blog/gpt-transformer/
- Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. (2017). Attention is all you need. In: *Advances in Neural Information Processing Systems*, 30 (NIPS 2017), pp. 5998-6008. https://papers.nips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf