Attention Mechanisms

Bora Deren, 438377
Tuğçe Kul, 465762

Supervisor: Pooja Babu

Outline

Outline

Biological Inspiration: From Brains to Networks

- **Human cognition**
  - Neurons and synapses form distributed networks
  - Information is processed in parallel
  - Selective attention prioritizes relevant signals
- **Artificial neural networks**
  - Distributed representations
  - Parallel computation
  - Weighted aggregation of inputs

Biological Attention Networks

- Attention emerges from distributed brain networks.
- Frontal-parietal regions modulate sensory processing.
- Relevance is encoded via differential signal weighting.

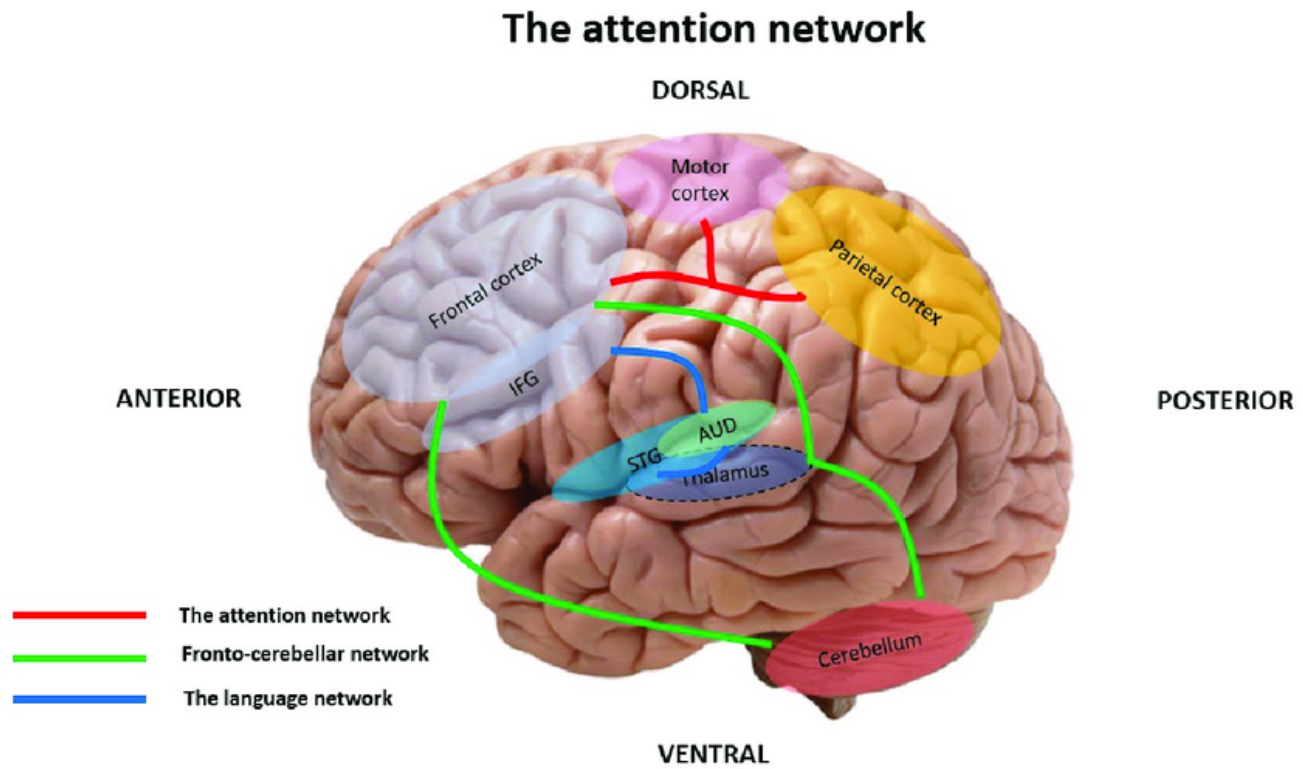


Figure 1: Biological attention network in the human brain, source 1

Basics of Neural Networks

- Artificial neuron:

$$y = \sigma\left(\sum_i w_i x_i + b\right)$$

- Neural networks combine such units across multiple layers.
- Learning updates the weights based on data.

Deep Learning: Why It Works Well

- Deep learning builds hierarchical representations.
- Many stacked layers enable increasing abstraction.
- Features are learned automatically from data.

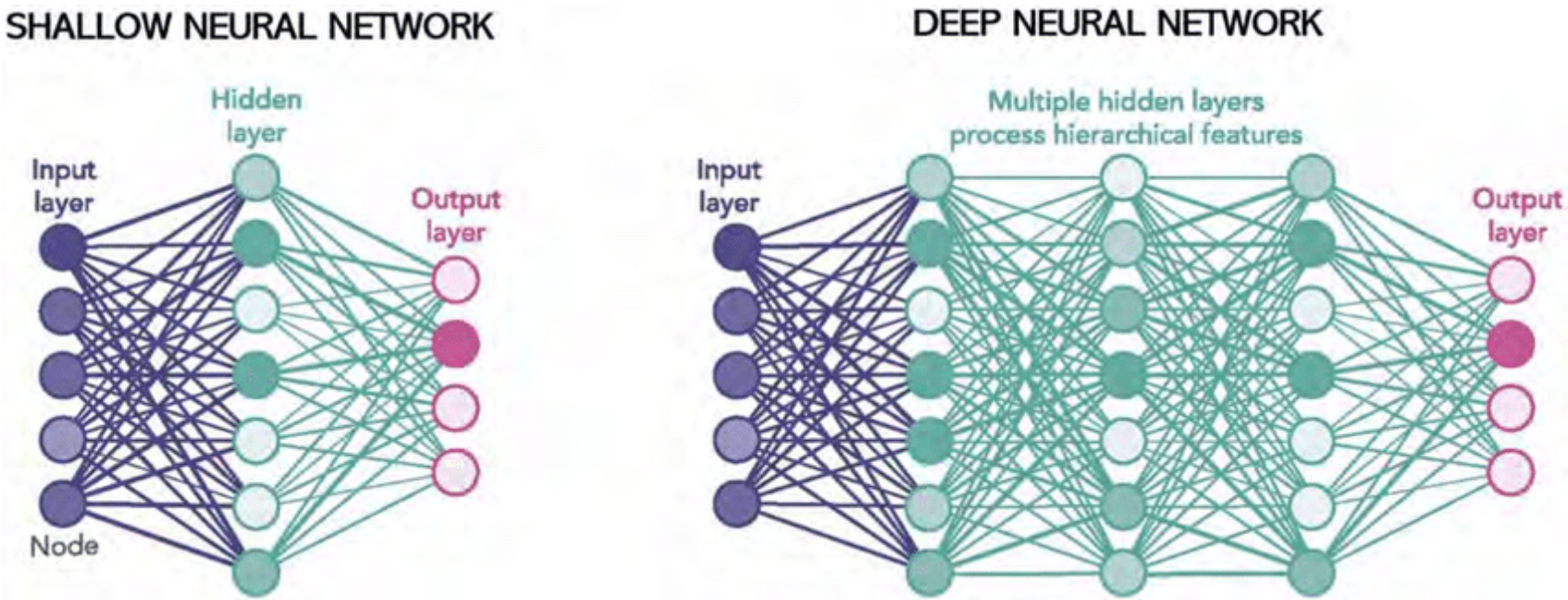


Figure 2: Comparison of shallow and deep neural networks, source 2

Outline

What Is NLP?

- Interaction between computers and human language
- Core tasks:

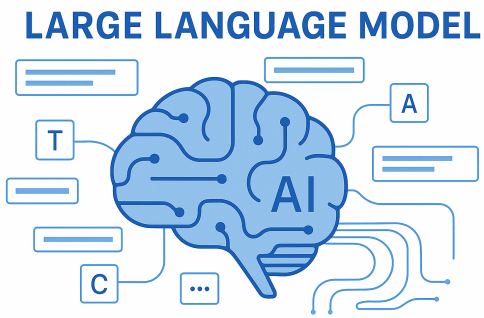


Figure 3: Language modeling task

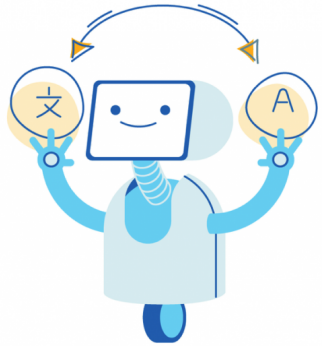


Figure 4: Machine translation

Figure 5: Question answering

Figure 6: Text summarization

Early NLP: n-grams

- Next-token prediction probability:

$$P(w_t \mid w_{t-1}, \ldots, w_{t-n+1})$$

- Strengths:
  - Simple
  - Interpretable
- Limitations:
  - Sparse statistics
  - Short context
  - No semantic understanding

Tokens and Embeddings

- Text is first converted into **tokens**
- Token granularity:
  - Words
  - Subwords (BPE, WordPiece)
  - Characters

Text	$\rightarrow$	**Tokens**
"I love NLP"		[I, love, NLP]

$$\Downarrow$$

$$w \rightarrow \mathbf{e}_w \in \mathbb{R}^d$$

Why Embeddings Matter

- One-hot representations do not generalize
- Embeddings capture similarity
- Enable semantic reasoning:

king − man + woman ≈ queen

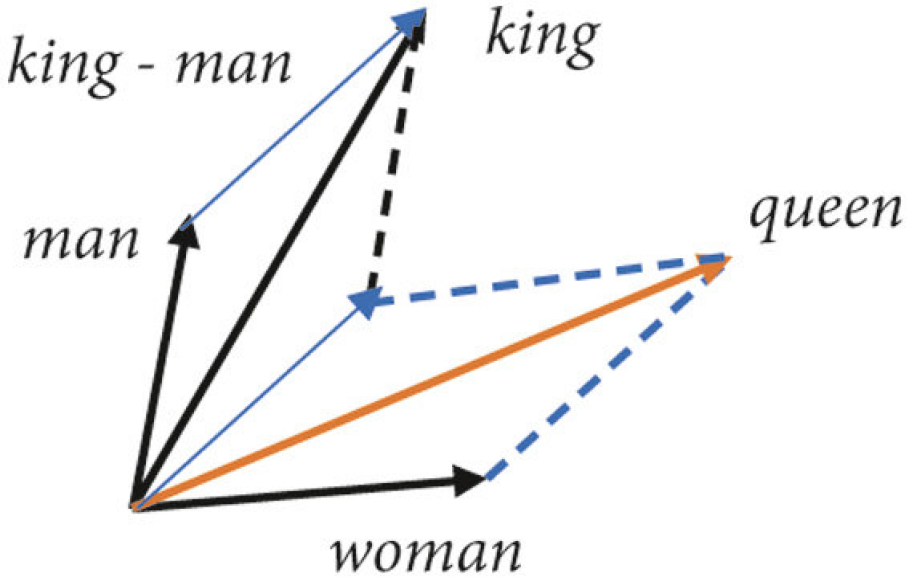


Figure 7: source 7

Outline

Neural Models for Language

- **Feedforward Neural Networks:**
  - Fixed context window
  - No sequence memory
- **Convolutional Neural Networks:**
  - Local pattern detection
  - Limited long-range context

Recurrent Neural Networks

- Designed for sequential data
- Processes inputs step by step
- Hidden state carries past information
- Limitations:
  - Vanishing gradients
  - Sequential computation

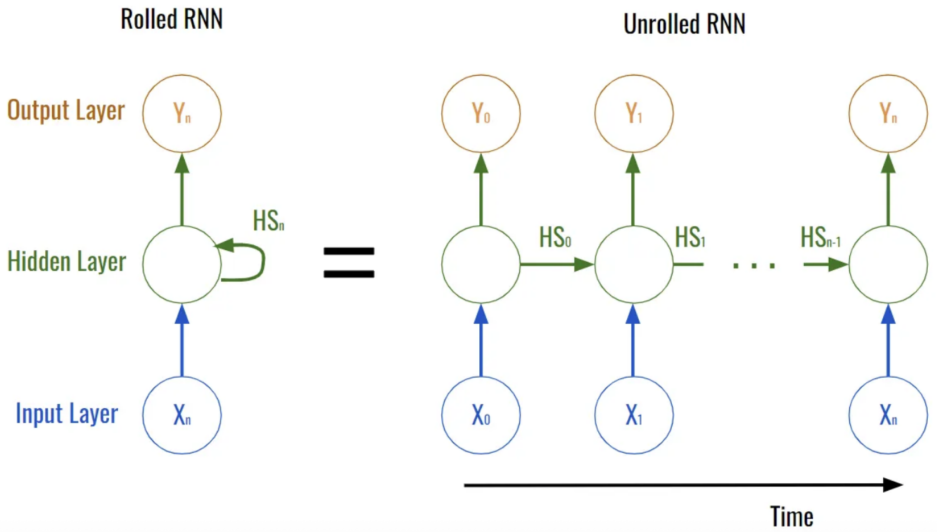


Figure 8: RNN architecture, source 8

LSTMs and GRUs

- Extensions of RNNs with gating mechanisms
- Better handling of long-term dependencies
- Remaining issues:
  - Sequential computation
  - Hard to parallelize

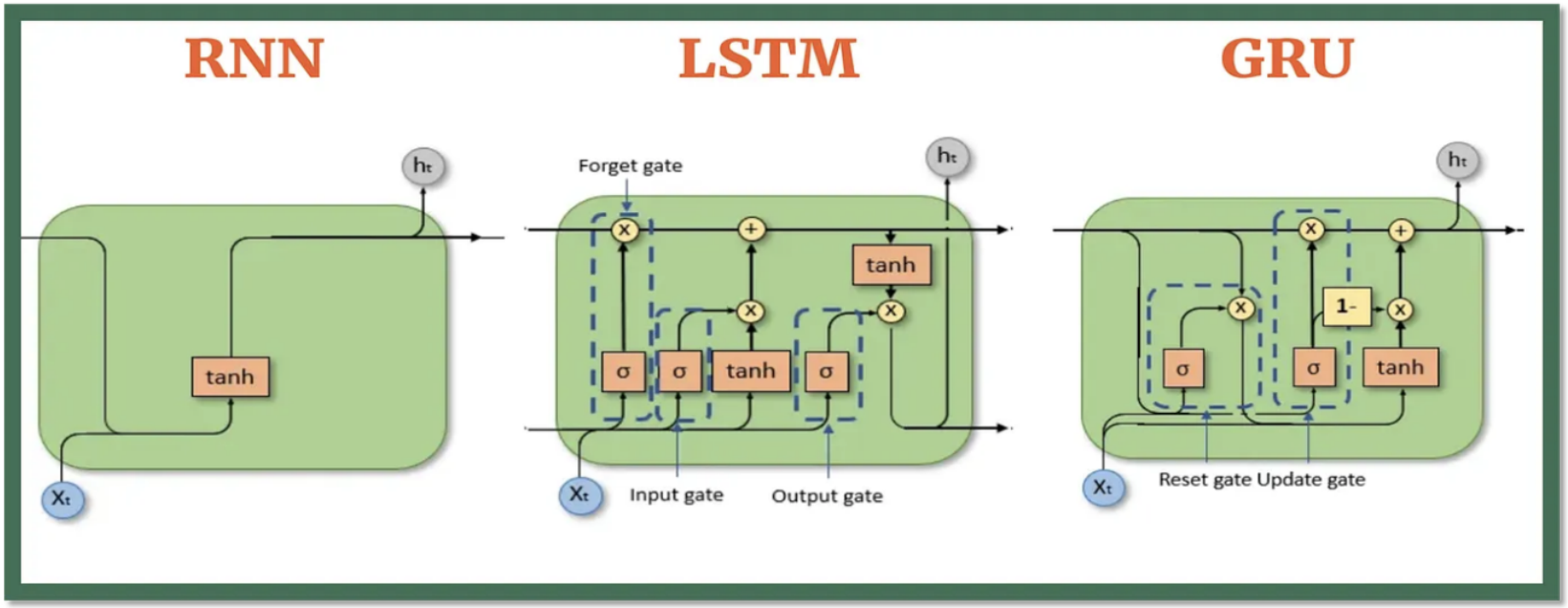


Figure 9: Comparison of RNN, LSTM and GRU architectures, source 9

Attention in Encoder-Decoder Models

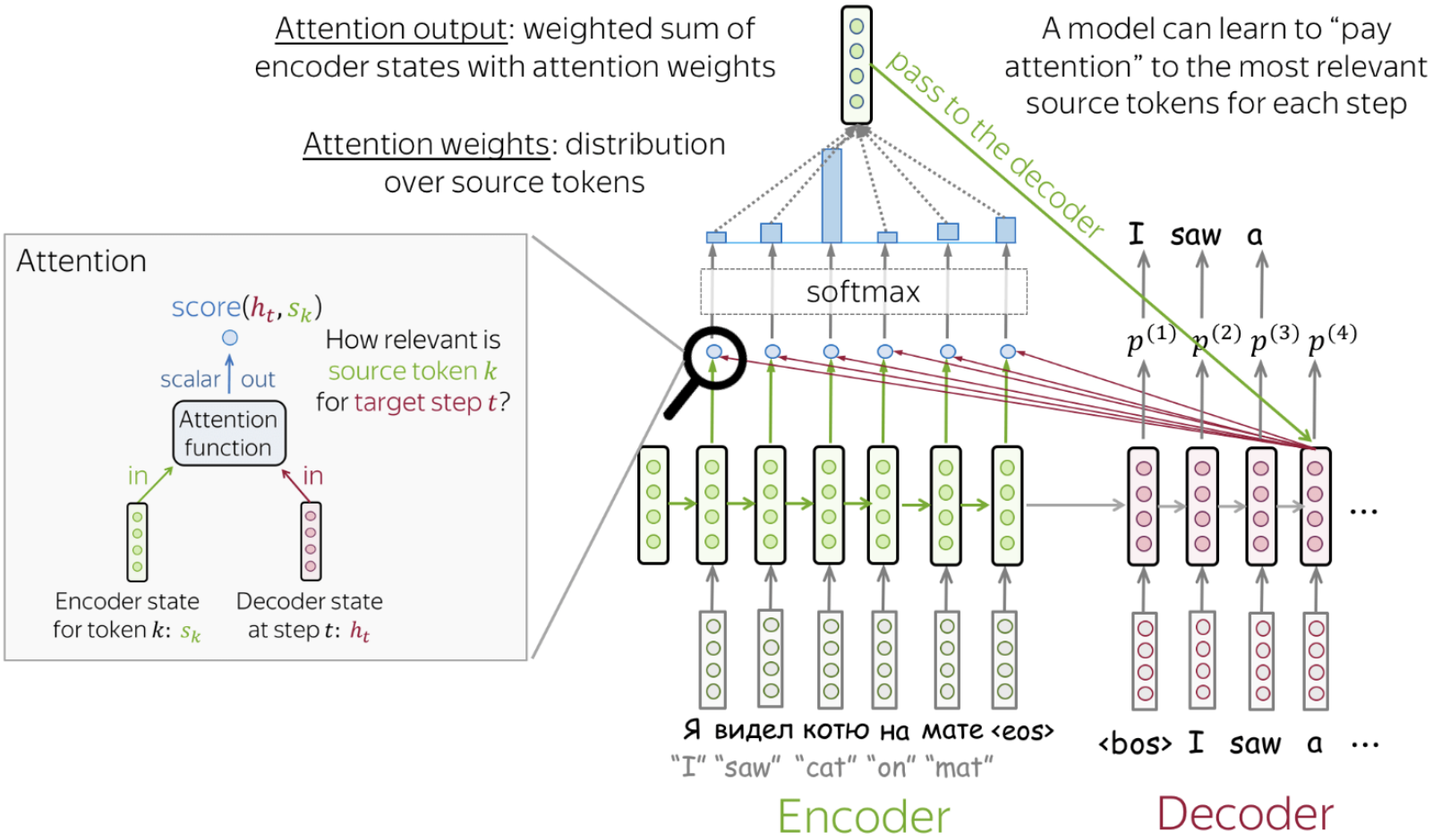


Figure 10: Attention mechanism in encoder–decoder models, source 10

Key Motivation for Attention

- Modeling long-range dependencies
  - $\rightarrow$ Meaning depends on distant tokens
- Efficient parallel computation
  - $\rightarrow$ Sequential processing limits scalability
- Interpretable relevance modeling
  - $\rightarrow$ Explicit relationships between tokens

Outline

Transformers

- Introduced in 2017 by Vaswani et al. (*"Attention Is All You Need"*)
- Sequence-to-sequence encoder–decoder architecture
- Eliminates recurrence (RNNs) and convolution (CNNs)
- Core building block: **self-attention**
- Processes entire sequences in parallel
- Uses:
  - Multi-head attention
  - Position-wise feed-forward networks
  - Residual connections and layer normalization
- Requires positional encoding to model word order

Vanilla Transformer

- Encoder–decoder architecture
- Introduced by Vaswani et al. (2017)
- Encoder:
  - Bidirectional self-attention
  - Produces contextual representations
- Decoder:
  - Masked (causal) self-attention
  - Cross-attention over encoder output
- Used for sequence-to-sequence tasks

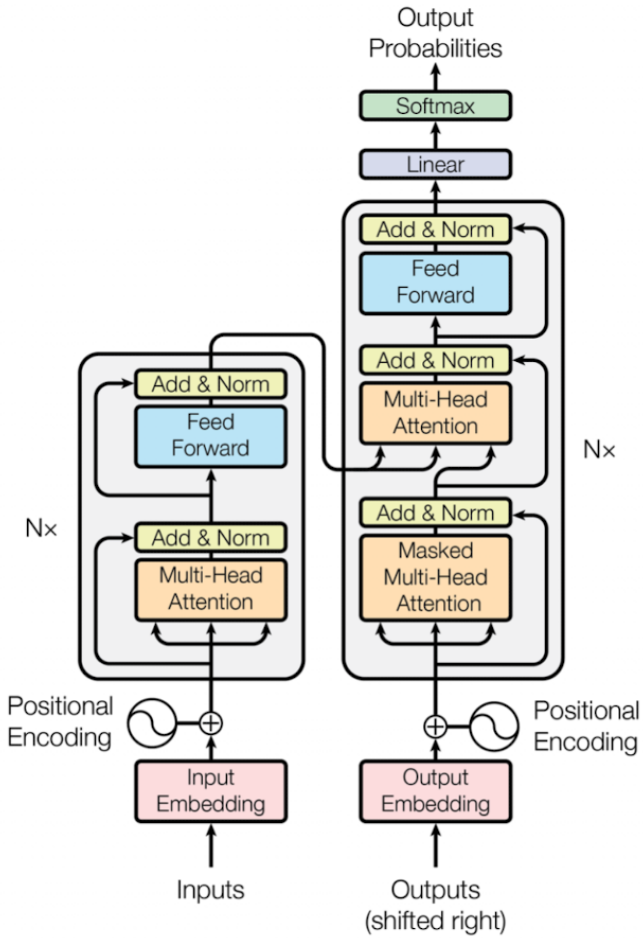


Figure 11: The Vanilla Transformer, source 14

Encoder-Only Transformer

- Stack of encoder blocks only
- Fully bidirectional self-attention
- Each token attends to all other tokens
- Learns rich contextual embeddings
- Pretrained with masked language modeling
- Common tasks:
  - Text classification
  - Question answering
  - Named entity recognition

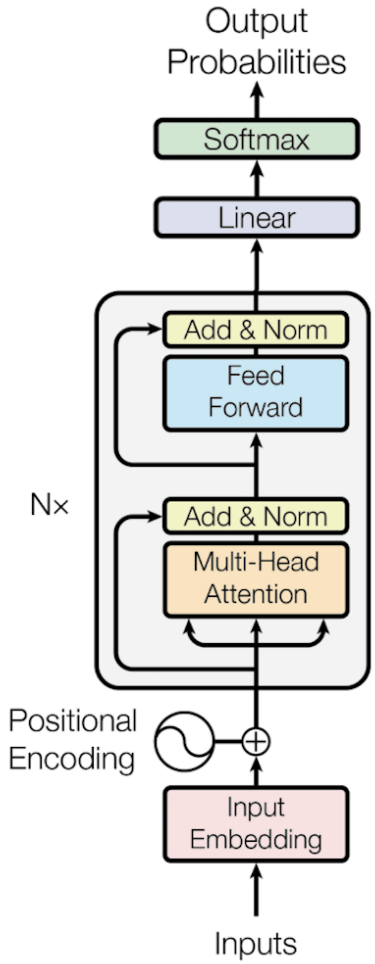


Figure 12: Encoder-only transformer, source 14

Decoder-Only Transformer

- Stack of decoder blocks only
- Uses causal (masked) self-attention
- Each token attends only to past tokens
- Autoregressive generation:
  - Predicts next token step-by-step
- No encoder or cross-attention
- Used for language modeling and text generation

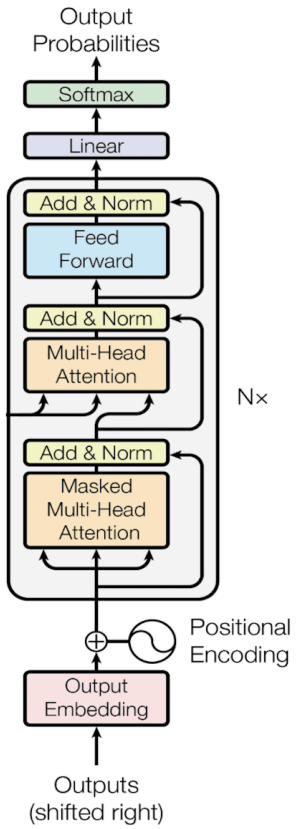


Figure 13: Decoder-only Transformer, source 14

Embeddings

- Sequence: "King likes queen"
- Tokenization: [King, likes, queen]
- Token IDs: [3, 7, 12]
- Embedded sequence:

$$\mathbf{X}_{emb} = \begin{bmatrix} 0.2 & 0.1 \\ 0.6 & 0.4 \\ 0.9 & 0.8 \end{bmatrix} \in \mathbb{R}^{T \times 2}$$

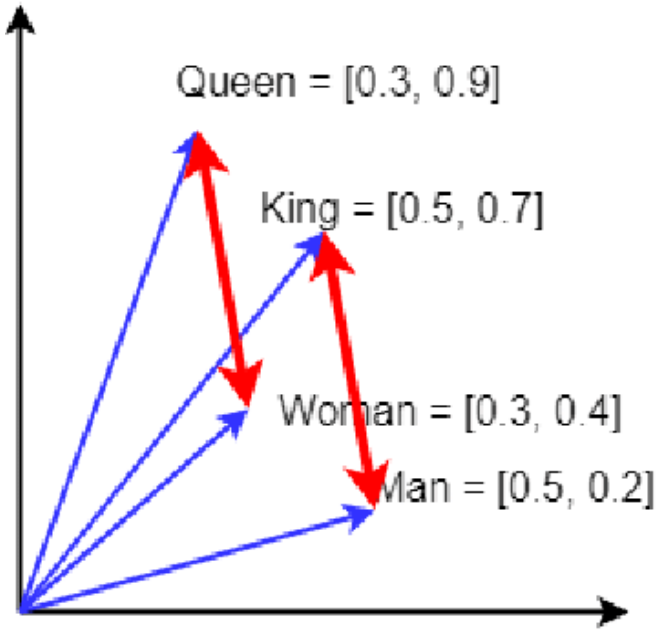


Figure 14: Embedding space, source 13

Positional Encodings

- Self-attention is permutation-invariant
- Position information must be added explicitly
- Positional encodings can be learned or fixed
- In vanilla Transformers, positional encodings are fixed sinusoidal functions:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right)$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d}}\right)$$

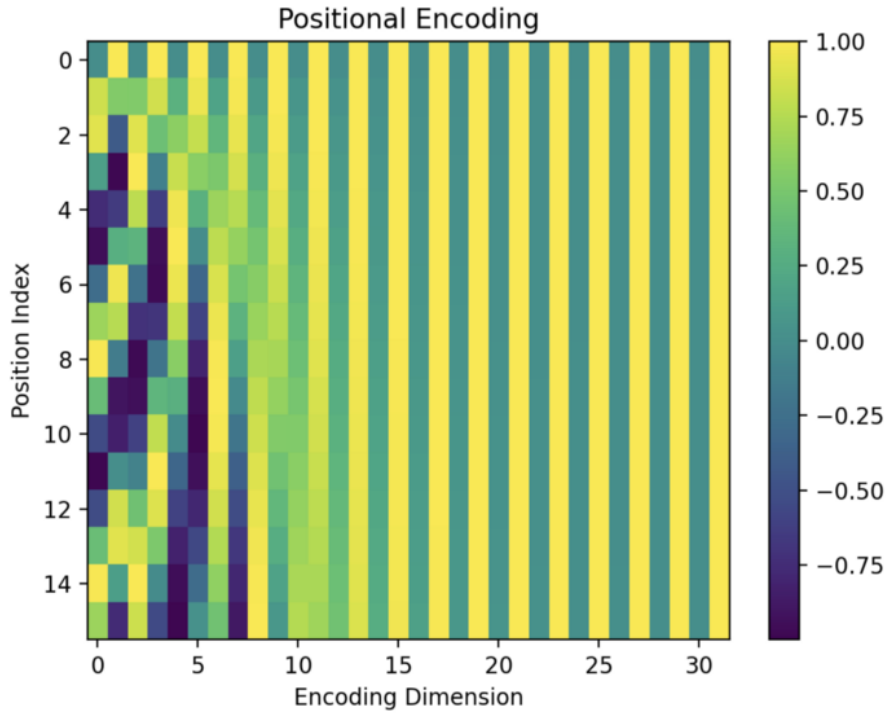


Figure 15: Sinusoidal encodings, source 11

Positional Embeddings

- The sequence "King likes queen" has to have a different embedding than the sequence "Queen likes king"
- Tokens: [King, likes, queen]
- Positional encodings:

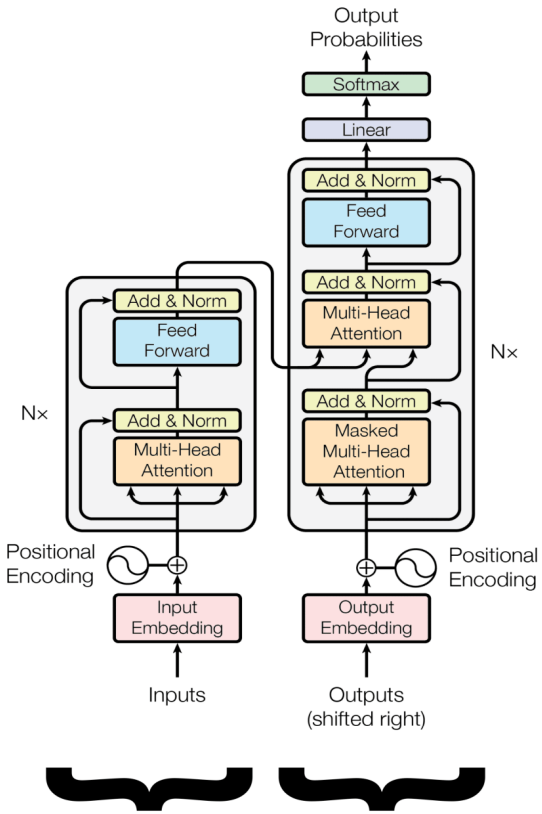
$$\mathbf{P} = \begin{bmatrix} 0.0 & 0.0 \\ 0.1 & 0.0 \\ 0.2 & 0.1 \end{bmatrix}$$

- Input to attention:

$$\mathbf{H}_0 = \mathbf{X}_{emb} + \mathbf{P} = \begin{bmatrix} 0.2 & 0.1 \\ 0.7 & 0.4 \\ 1.1 & 0.9 \end{bmatrix} \in \mathbb{R}^{T \times 2}$$

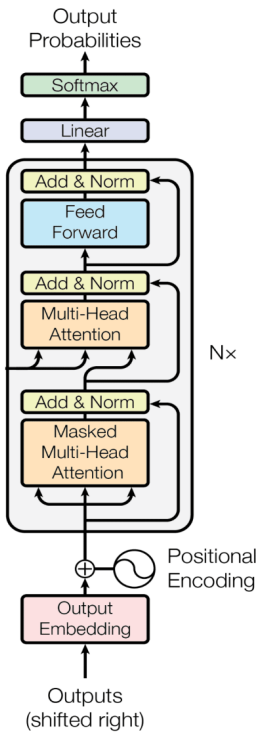
Overview

Transformer



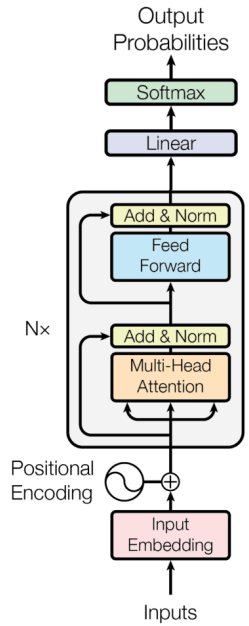
Encoder **Decoder**

GPT*



Decoder-only

BERT*



Encoder-only

*Illustrative example, exact model architecture may vary slightly

Why Transformers Work Well

- Constant path length between tokens
- Fully parallelizable
- Strong scaling with data and compute

Outline

Biological Attention

- The brain receives more information than it can process
- Attention helps select what is most relevant at a given moment
- Examples:
  - Focusing on one voice in a noisy room
  - Watching a single moving object in a crowd
  - Paying attention to a traffic light while ignoring other cars

Attention: Core Idea

- Each element decides what to focus on
- Queries match Keys to retrieve Values
- Allows direct interaction between all tokens

Attention Formula

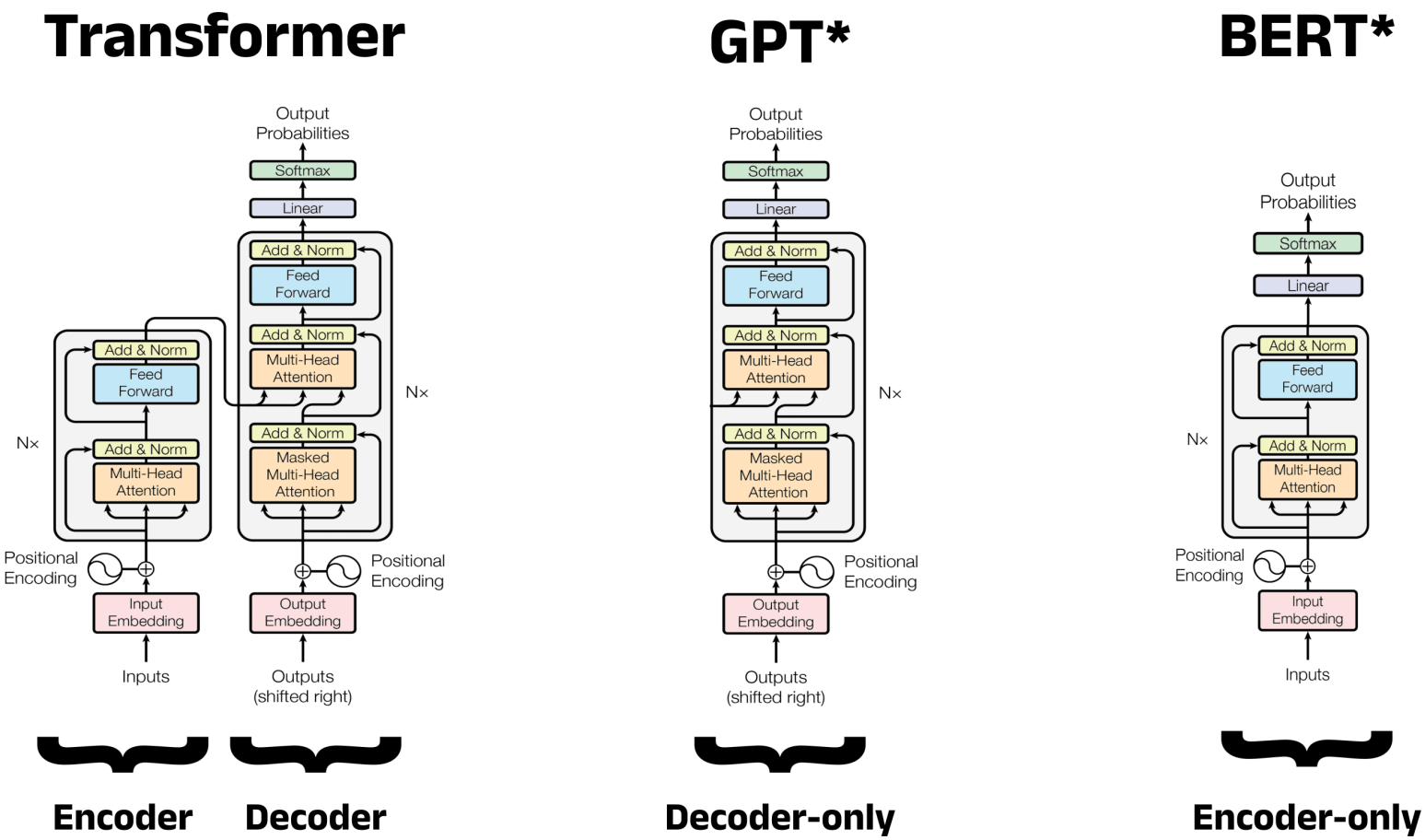
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- Q: Query
- K: Key
- V: Value

Types of Attention Mechanisms: Overview

- **By scope (who attends to whom)**
  - Self-attention: queries, keys, values from the same sequence
  - Cross-attention: queries from one sequence, keys/values from another
- **By direction (information flow)**
  - Bidirectional: attend to past and future tokens (e.g. encoders, BERT)
  - Causal (masked): attend only to past tokens (e.g. decoders, GPT)
- **By scoring function**
  - Dot-product attention: similarity via inner product
  - Scaled dot-product attention: dot-product scaled by $\sqrt{d_k}$
- **By number of heads**
  - Single-head attention: one attention distribution
  - Multi-head attention: multiple parallel attention heads

Overview



*Illustrative example, exact model architecture may vary slightly

Figure 16: Transformers, source 14

Bidirectional Attention (1)

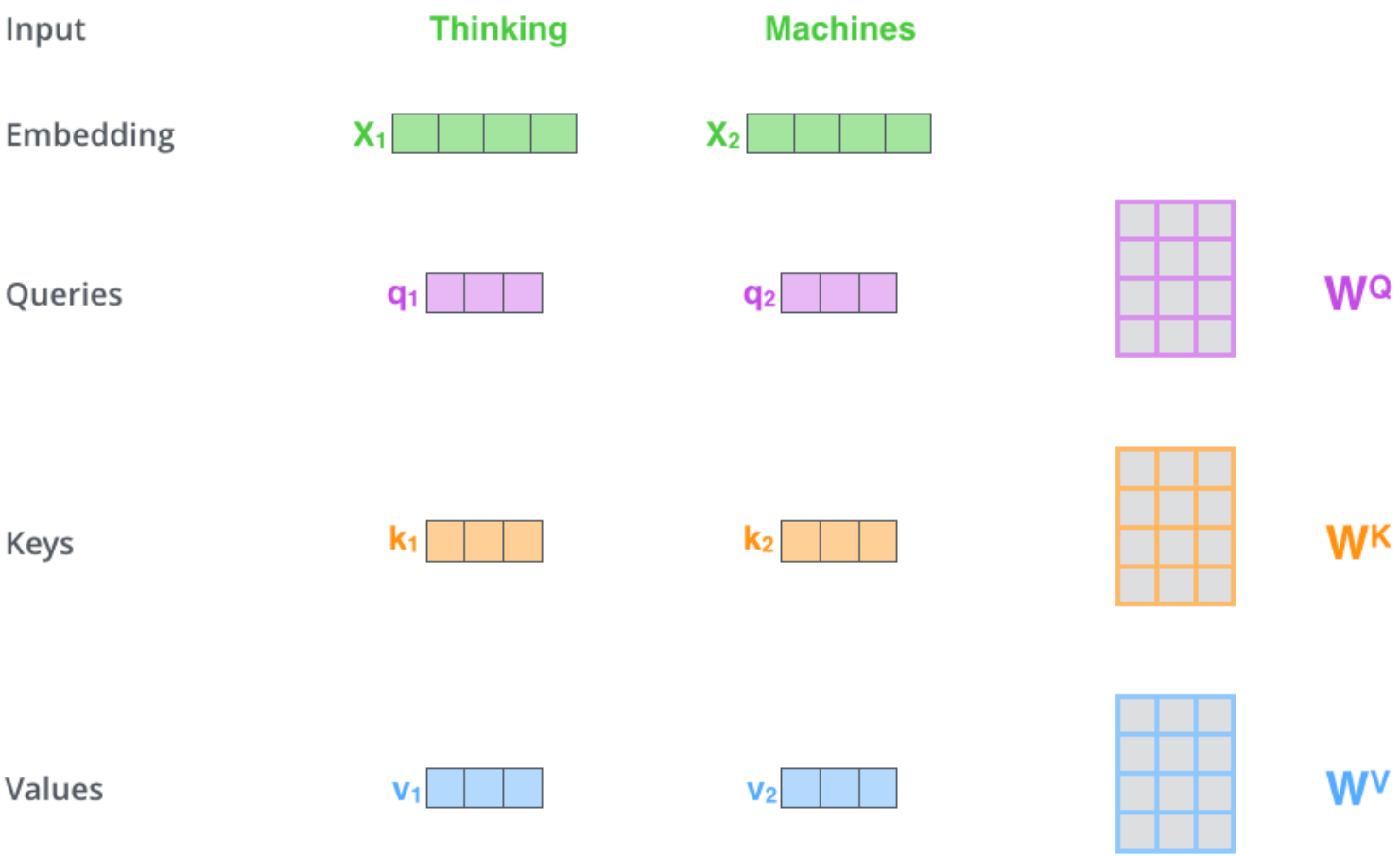


Figure 17: Here the head dimension is 3, source 12

Bidirectional Attention (2)

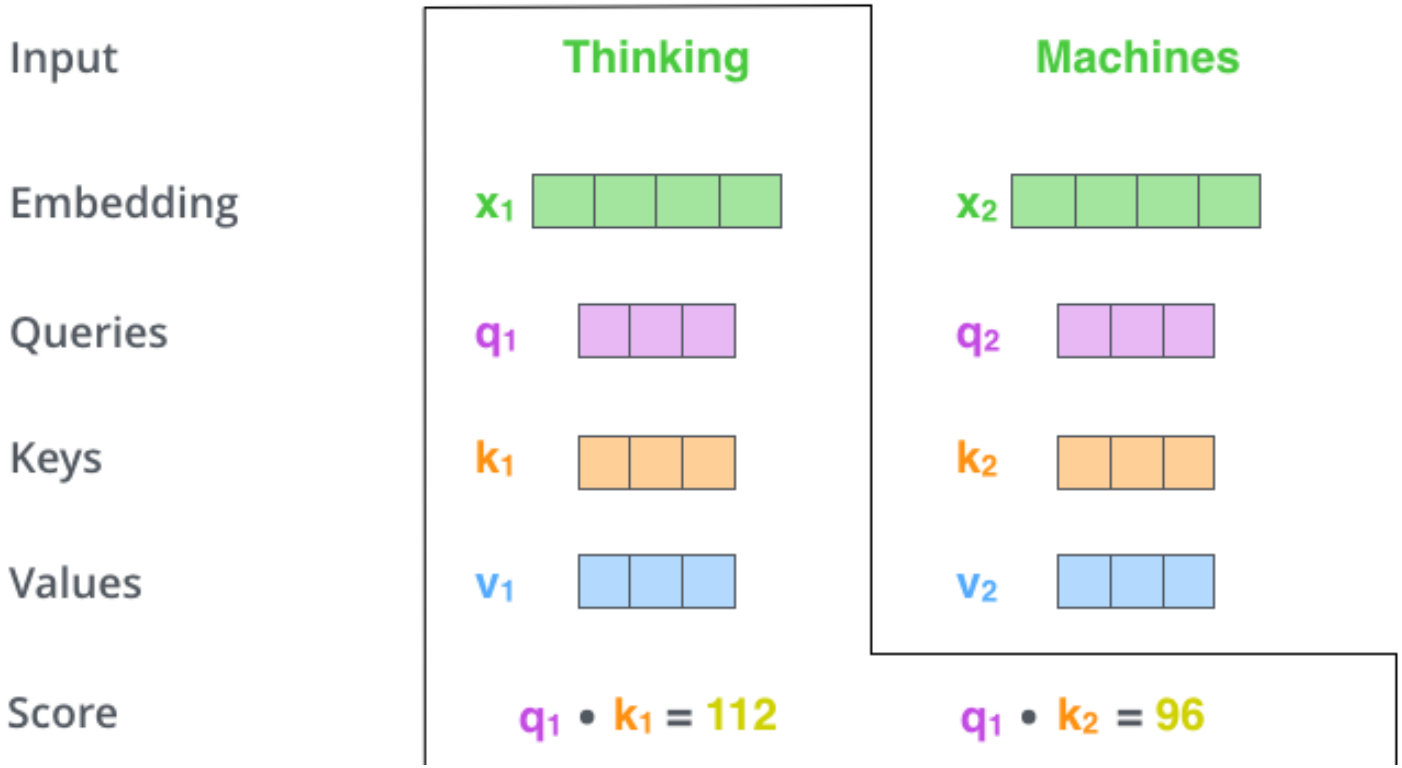


Figure 18: Attention scores of "Thinking" to other words, source 12

Bidirectional Attention (3)

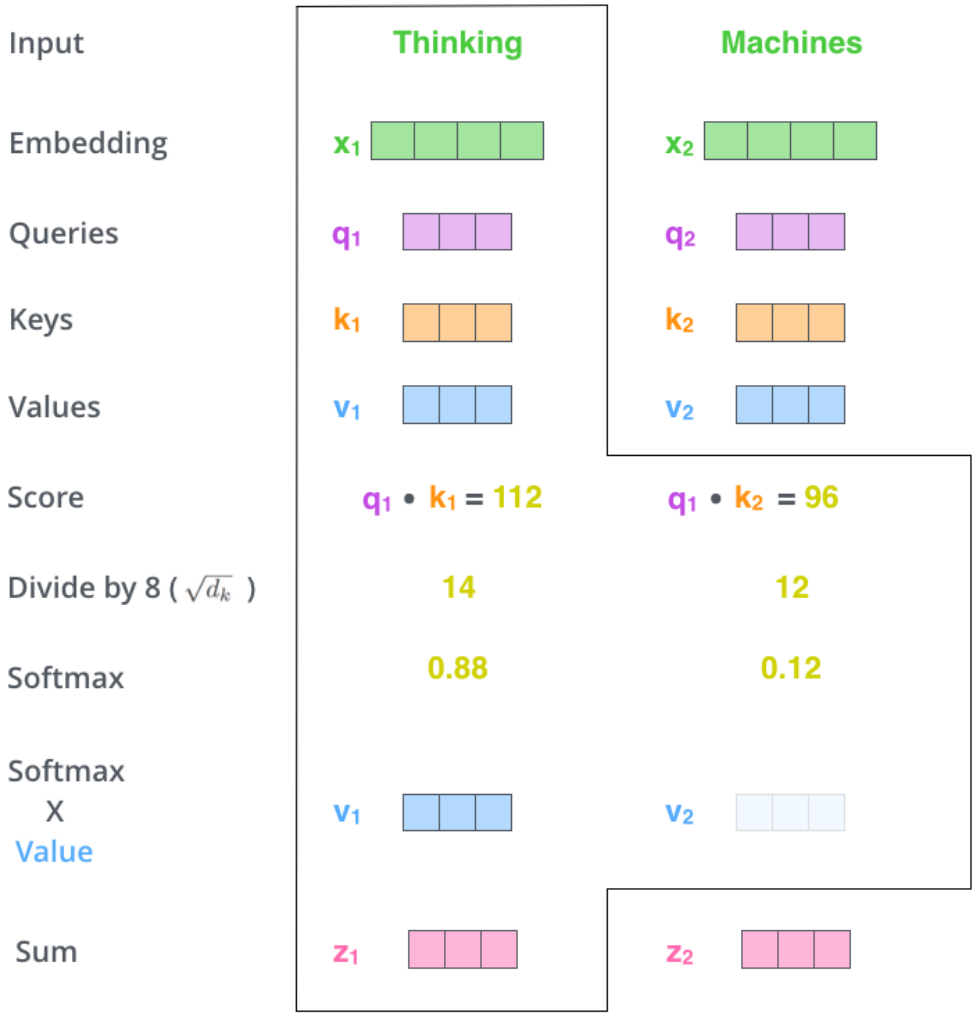


Figure 19: $z_1 = 0.88\, v_1 + 0.12\, v_2$ is an internal 3-dimensional representation of token the "Thinking", source 12

Attention with matrices

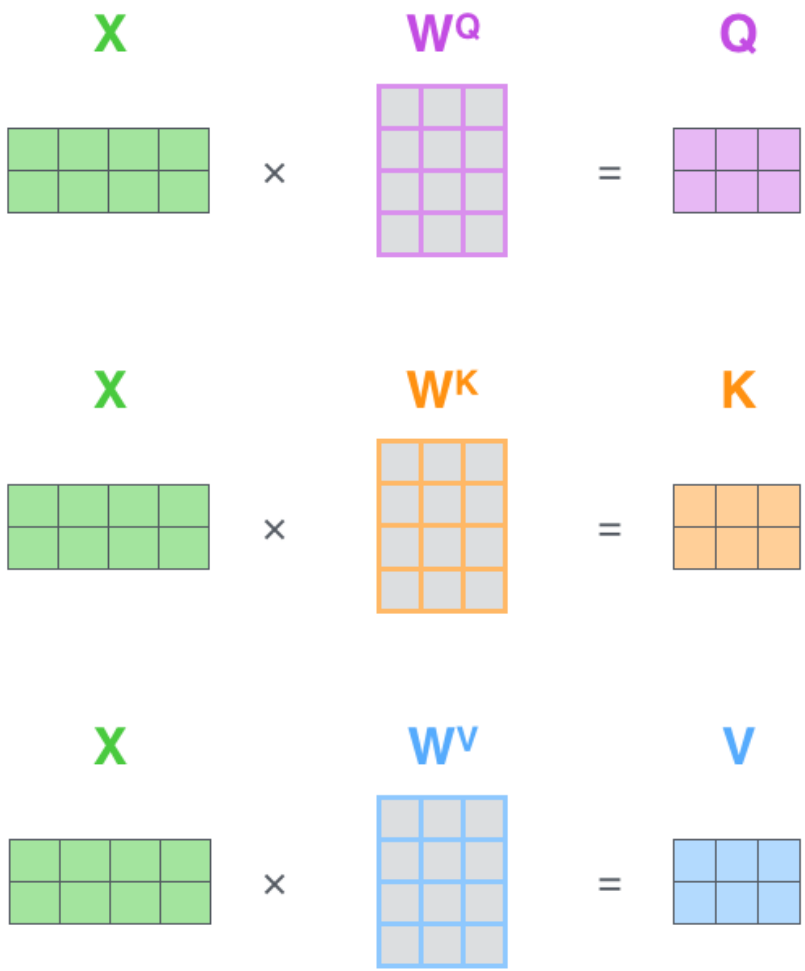


Figure 20: Here the head dimension is 3, source 12

Attention with matrices

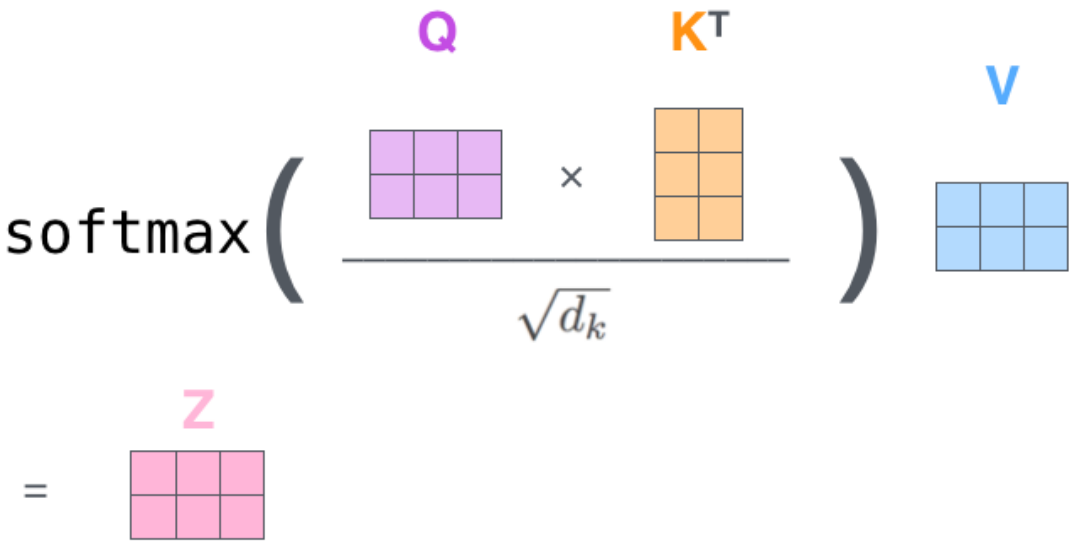


Figure 21: Z is the internal representations for all tokens in the sequence source 12

Attention with matrices

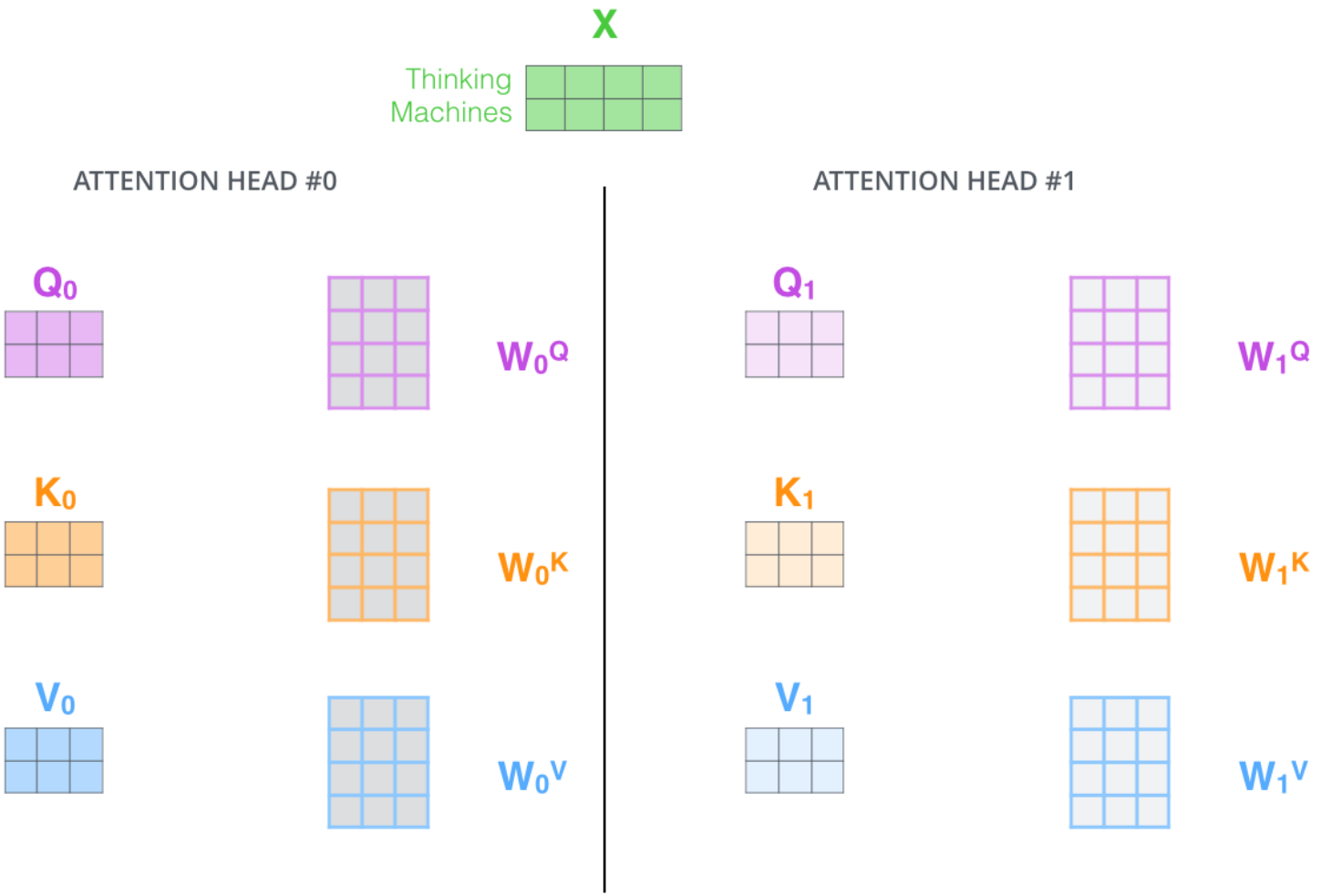


Figure 22: Different projection matrices for different heads are used, source 12

Multi-Head Self-Attention

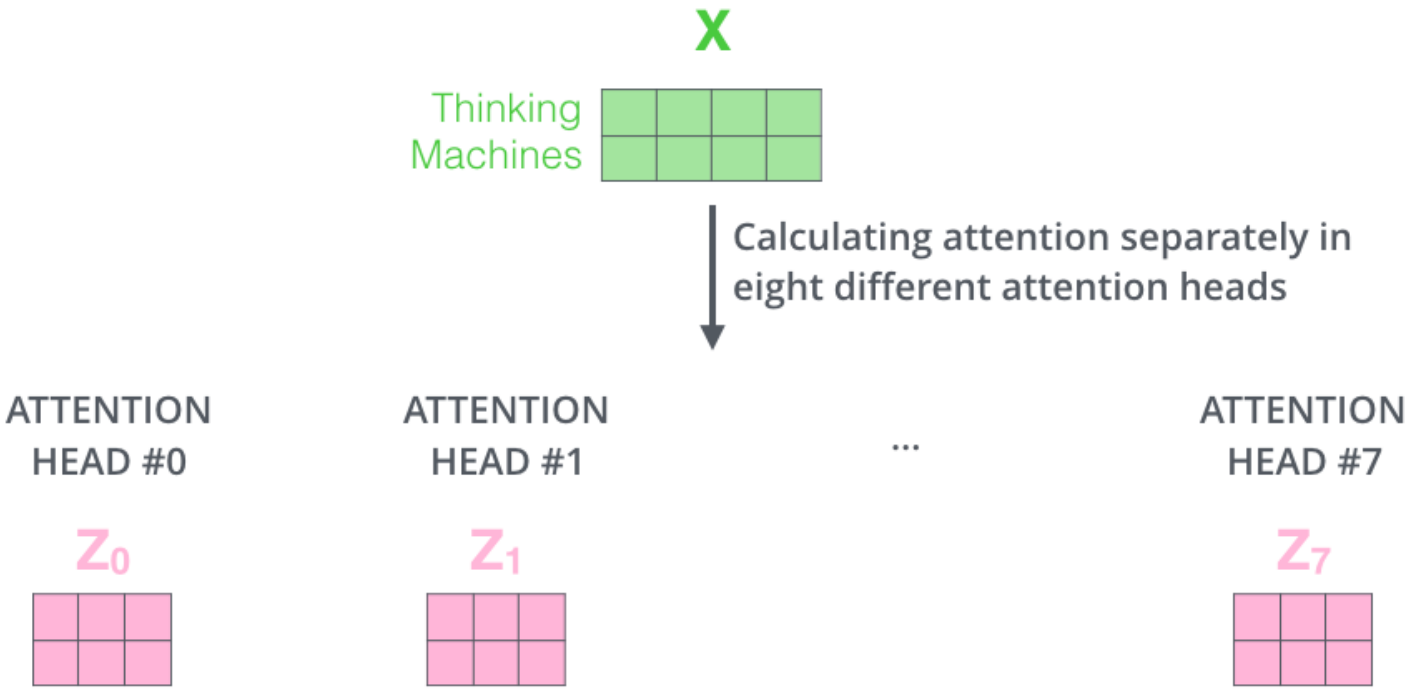


Figure 23: Z_i is the internal representation of all tokens after head i, source 12

Concatenation and projection to *emb_dim*

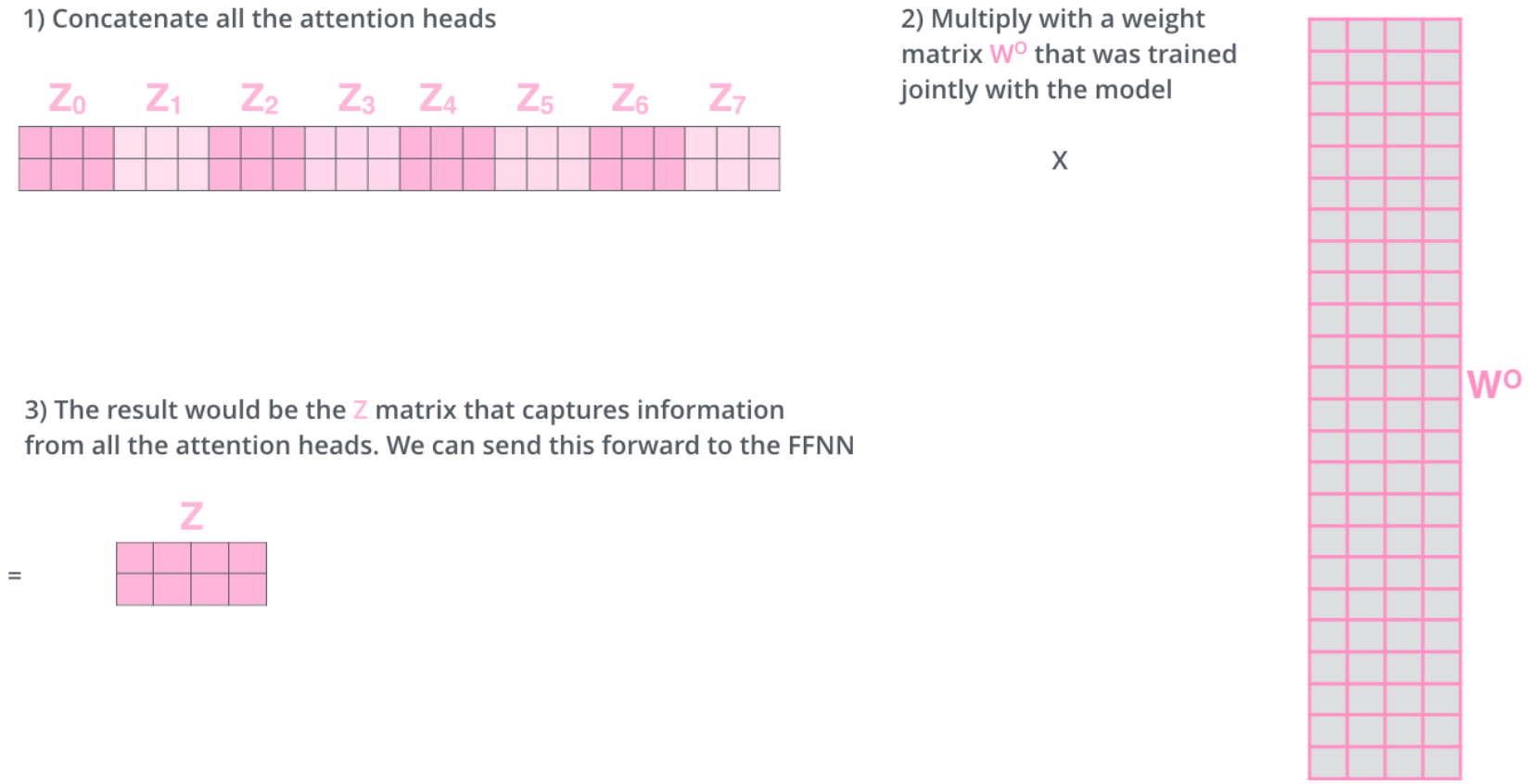


Figure 24: Concatenation of heads and projection back to *emb_dim*, source 12

Overview

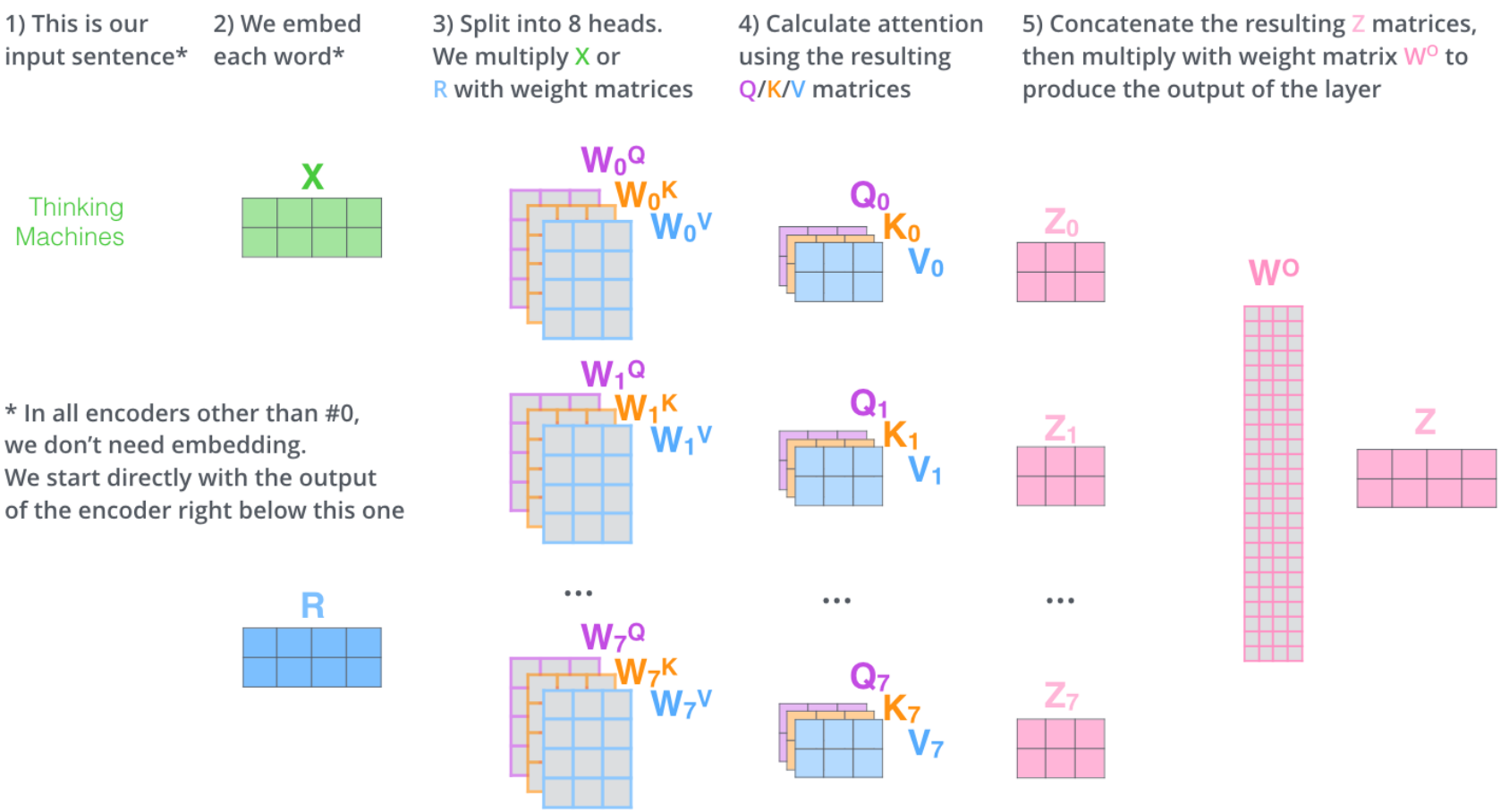


Figure 25: Overview of self-attention, source 12

Notes

- Z_i has now the contextual "enriched" embeddings of the tokens
- The input and output to the self-attention blocks are of same shape
- The vanilla transformer uses multi-head self-attention with

$$head\_dim = \frac{emd\_dim}{num\_heads}$$

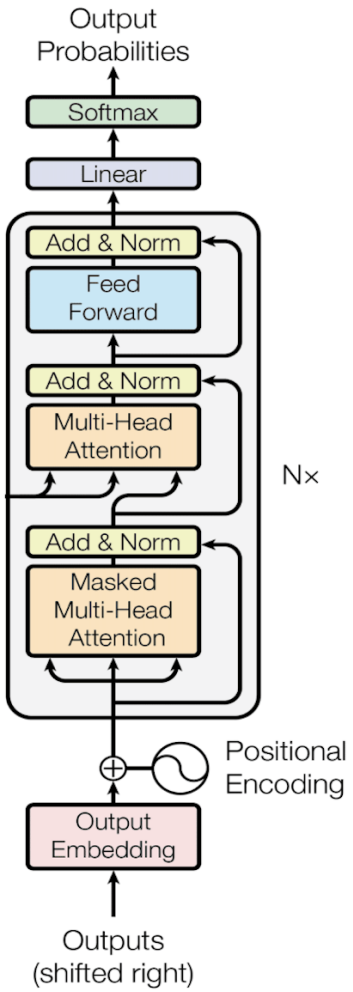


Figure 26: Decoder-only Transformer, source 14

Masked Single-Head Self-Attention

- Input positional embeddings:

$$\mathbf{H}_0 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \quad (T = 3,\ d = 2)$$

- Linear projections:

$$\mathbf{W}_Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{W}_K = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{Q} = \mathbf{H}_0\mathbf{W}_Q = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{K} = \mathbf{H}_0\mathbf{W}_K = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 0 & 1 \end{bmatrix}$$

- Unscaled attention scores:

$$\mathbf{Q}\mathbf{K}^\top = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 3 & 1 \\ 1 & 2 & 1 \end{bmatrix}$$

Masked Single-Head Self-Attention

- Apply causal mask:

$$\begin{bmatrix} 1 & -\infty & -\infty \\ 2 & 3 & -\infty \\ 1 & 2 & 1 \end{bmatrix}$$

- Scale by $\sqrt{d} = \sqrt{2}$ and apply softmax (row-wise):

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0.33 & 0.67 & 0 \\ 0.21 & 0.58 & 0.21 \end{bmatrix}$$

Masked Single-Head Self-Attention

Value projection:

$$\mathbf{W}_V = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{V} = \mathbf{H}_0 \mathbf{W}_V = \begin{bmatrix} 1 & 0 \\ 2 & 1 \\ 1 & 1 \end{bmatrix}$$

Attention output:

$$\mathbf{H}_1 = \mathbf{AV} = \begin{bmatrix} 1.00 & 0.00 \\ 1.67 & 0.67 \\ 1.58 & 0.79 \end{bmatrix}$$

Each token attends only to itself and past tokens.

Residuals and LayerNorm

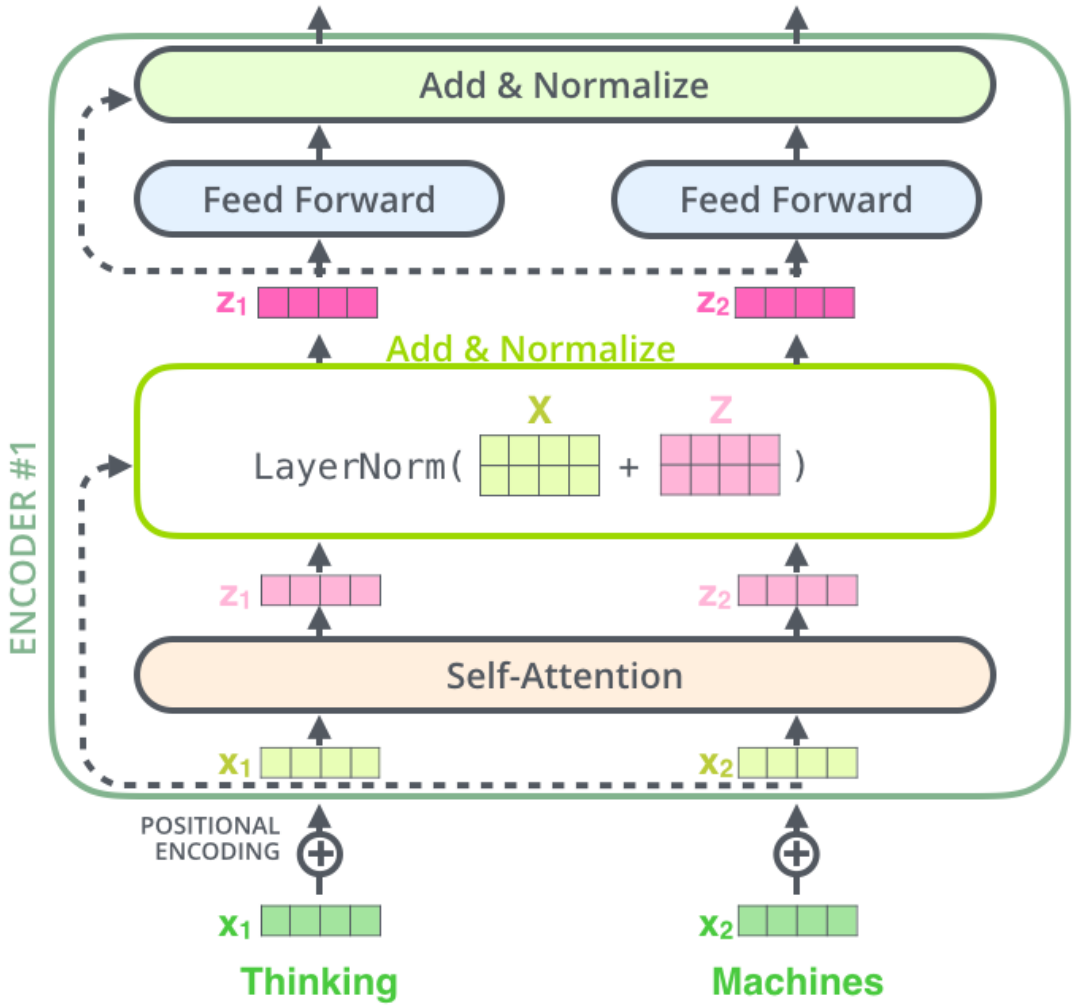


Figure 27: Residuals and LayerNorm, source 12

Output

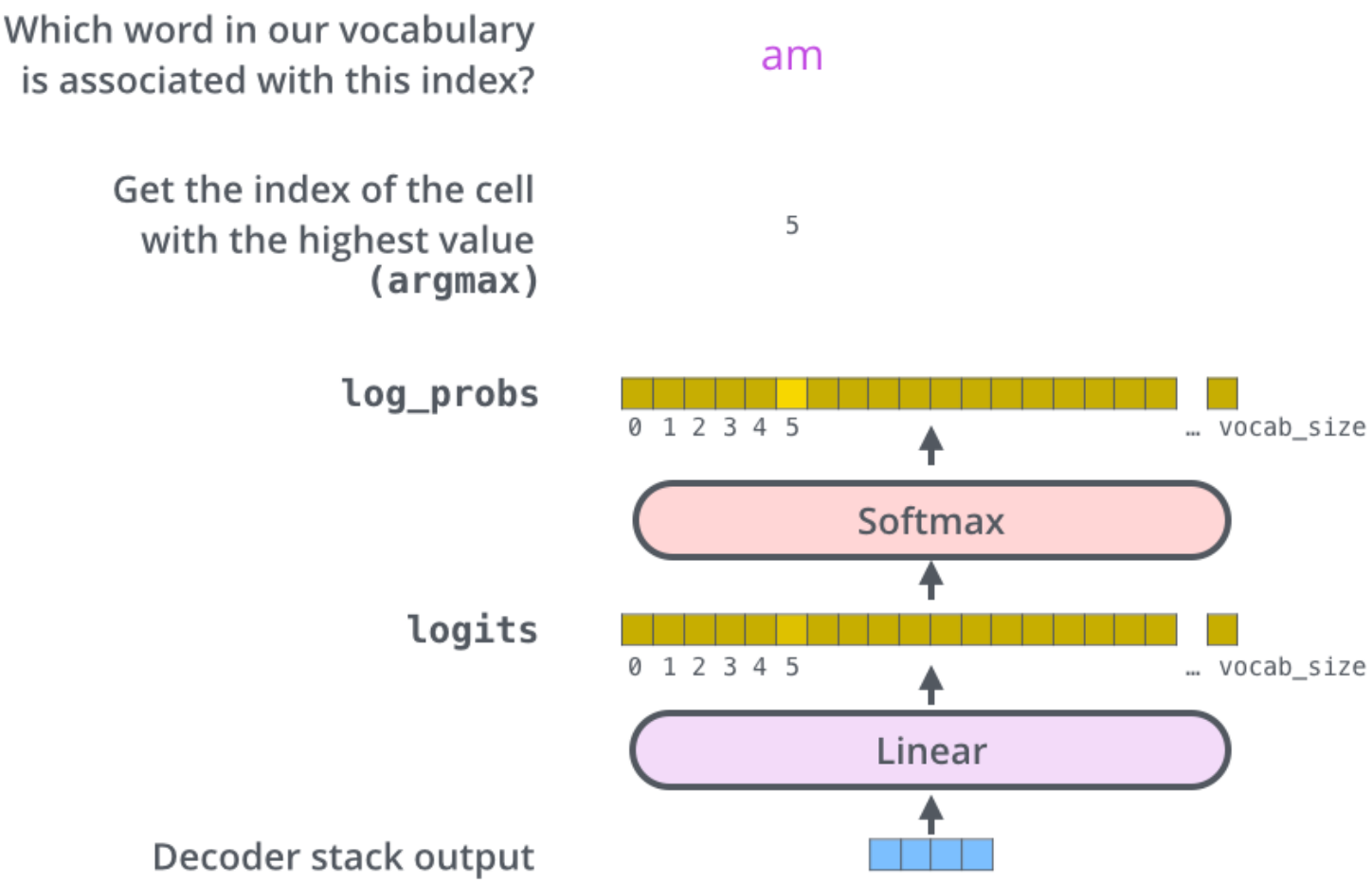


Figure 28: Reading the output, source 12

Training and Loss

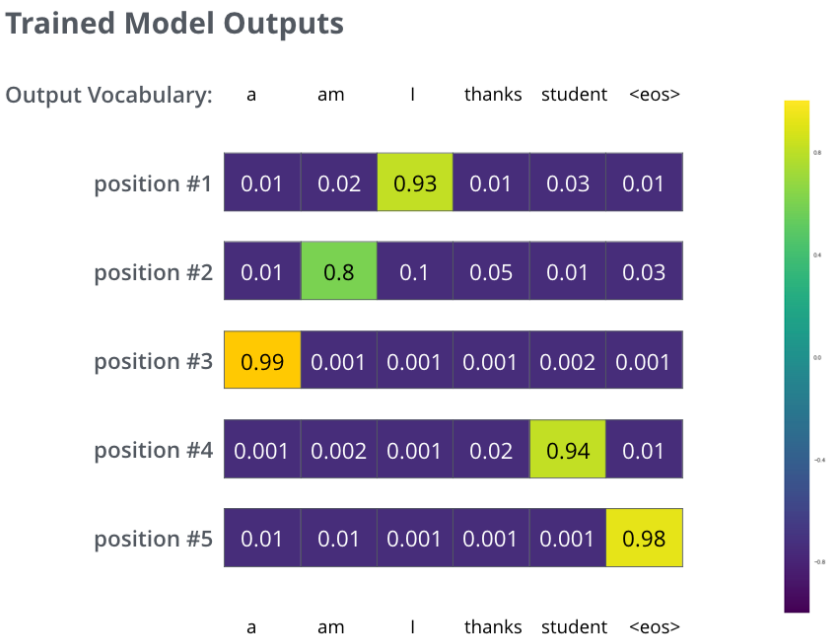


Figure 29: Probabilites assigned to each token in vocabulary, source 12

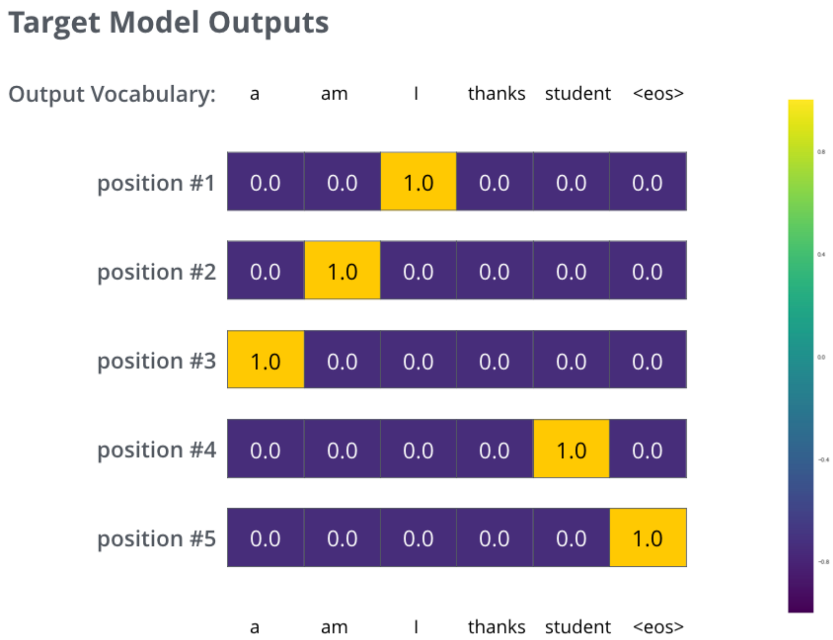


Figure 30: One-hot coded targets of each token in vocabulary, source 12

Outline

Encoder–Decoder Transformers

Architecture

- Encoder: bidirectional self-attention
- Decoder: autoregressive self-attention with causal masking
- Cross-attention aligns input/output

Typical Applications

- Machine translation
- Text summarization

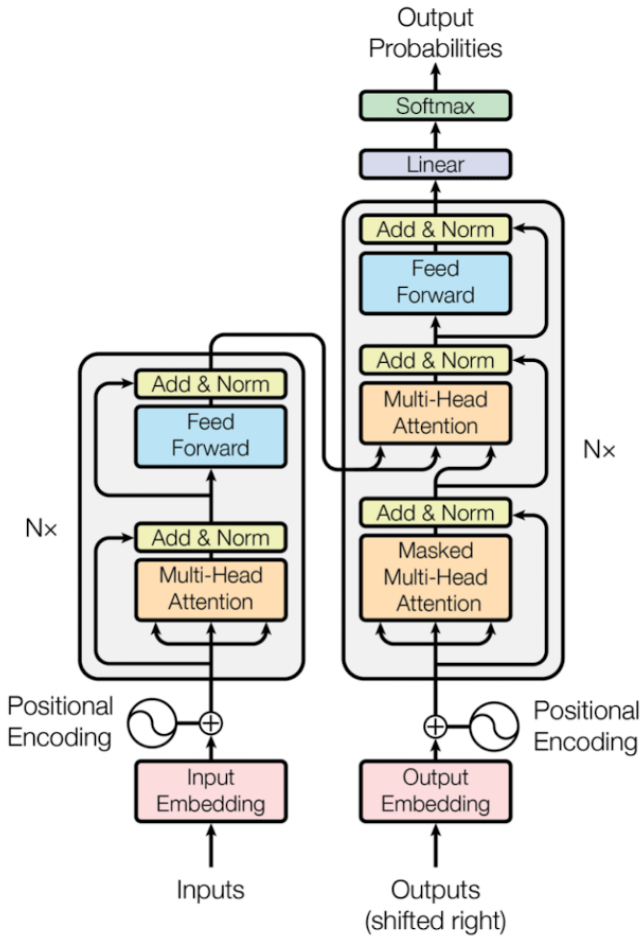


Figure 31: Encoder-Decoder, source 14

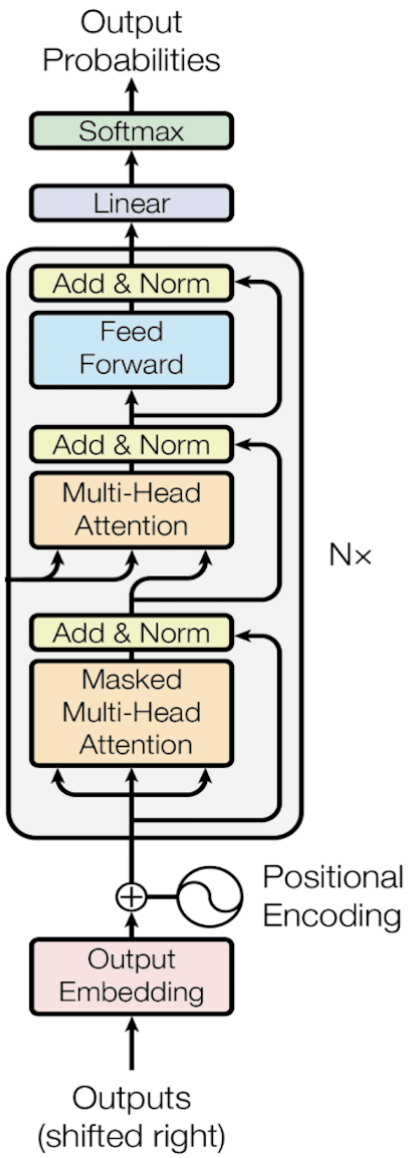
Decoder-Only Transformers

Architecture

- Stack of masked self-attention layers
- No encoder; attends only to past
- Trained with next-token prediction

Typical Applications

- Language modeling (GPT)
- Text generation
- Chatbots



Encoder-Only Transformers

Architecture

- Stack of bidirectional self-attention layers
- No causal masking
- Contextualized representations

Typical Applications

- Text classification (Sentiment)
- NER
- Extractive QA

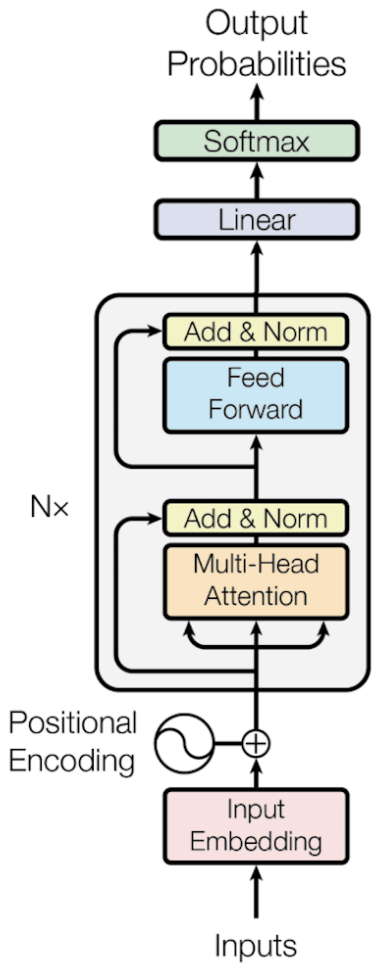


Figure 33: Encoder-Only, source 14

Vision Transformers (ViT)

- Images are split into fixed-size patches
- Each patch is embedded as a token
- A Transformer encoder models global relations

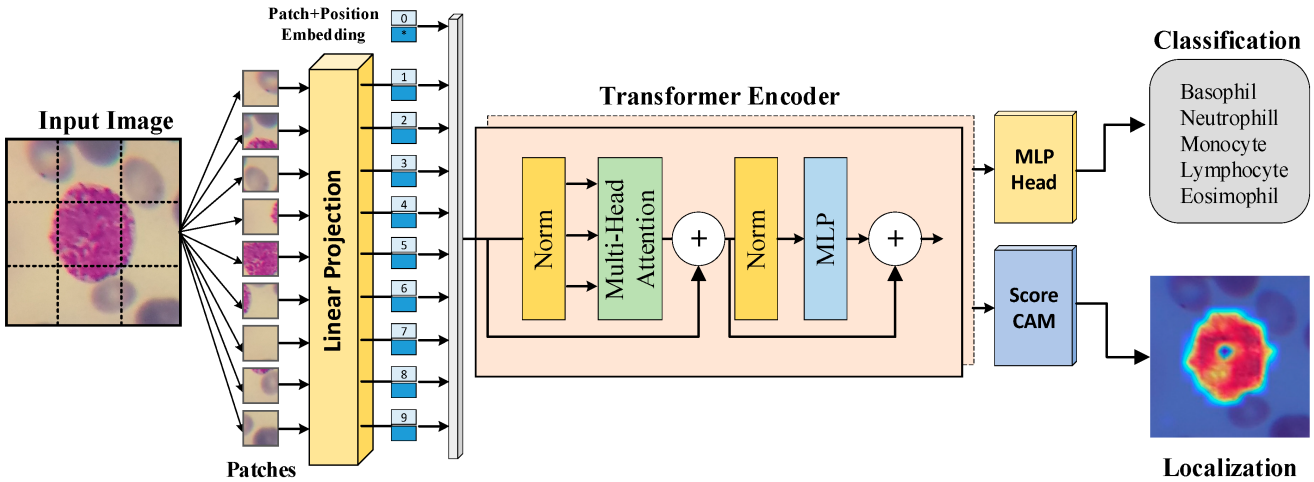


Figure 34: Vision Tranformer, source 15

Application: Character-Level Language Modeling (Shakespeare)

- Task: learn $p(x_t \mid x_{<t})$ and generate Shakespeare-like text, source 16
- Same dataset, same sampling procedure
- Comparison of model capacity and context modeling

Bigram Model

Who mishef dr'tof VA:
OMIAs.

BE:
3qud fallelibre cowe ur tous f th polas ay iltur
TIf waned t;

AUTIt
WI:
Ju thecklmy n odshareatave tsove aces mungot beenougr tourer; qu sot corowof nqunt th lathale lly te outill I de was al aly:

LIld PHurerga bes, someato labre s h, inomeyoom. fofat his th n whir juthit ino ad otyf ant bes mand

Decoder-Only Transformer

ERUS:
VIful to furth herve lieng. Fruissol, cathone theser
But bouth Say ally prot mon, bluth thent hats and,
Un if thouch't with beth, of willt?

LUENIAUS:
Why you thou colt the caurds.

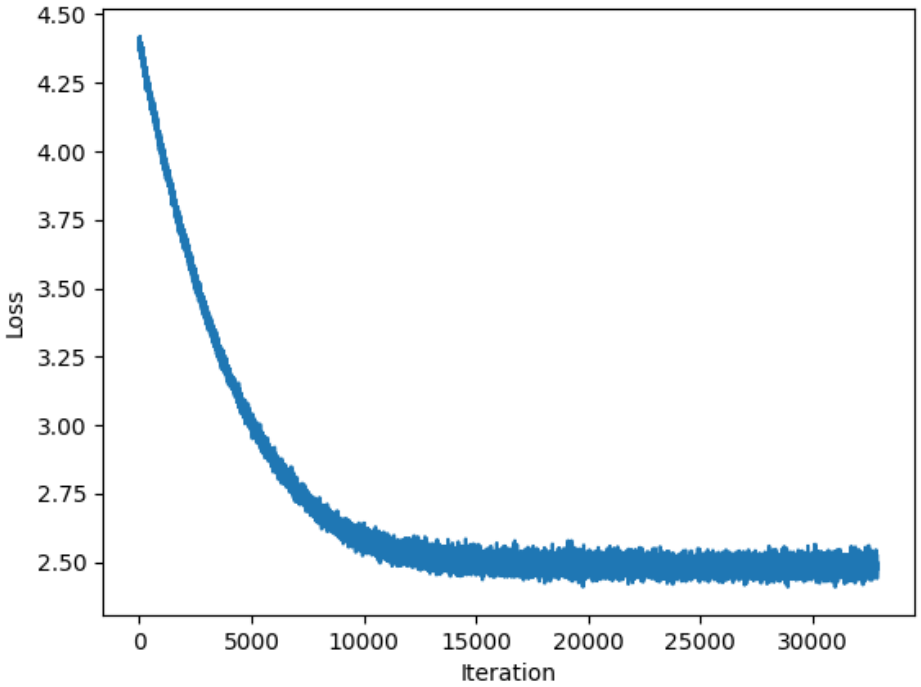
ADWARGBET:
Comie Your how is malth beacars?
Bo henthe shest in and beter heat plis theen
p sho mim who dritheed, in I.

JULETET:
I'd weirl, have ne beeth- who at mus'de
Trigh.

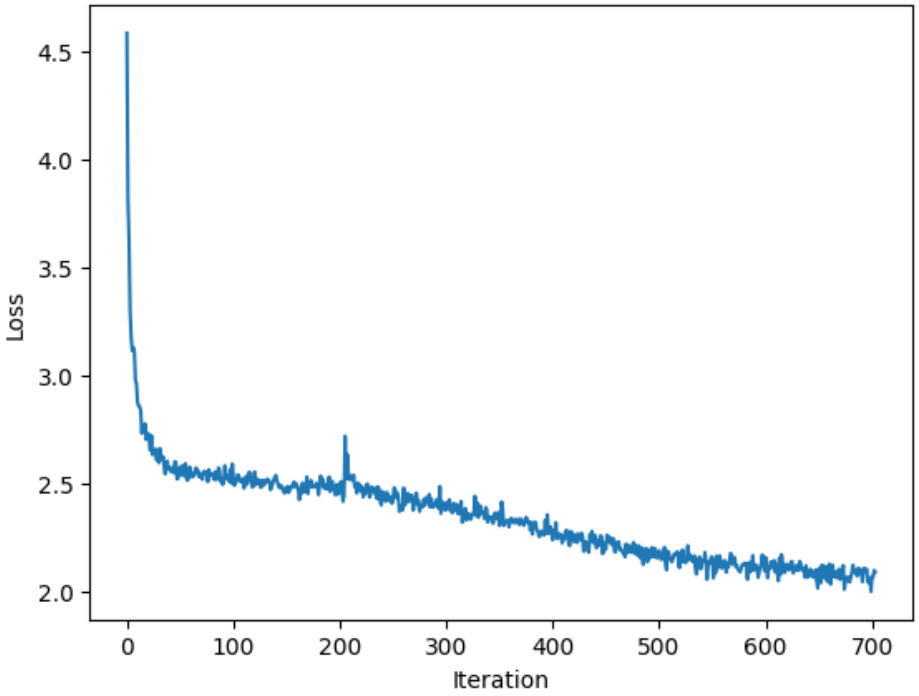
Model Training & Comparison

- **Bigram Model**:
  - Parameters: 4,225
- **Decoder-Only Transformer**:
  - Parameters: 7,956,033
  - Embedding Dimension: 256

Bigram Loss



Transformer Loss



Outline

Conclusion

- Language modeling has evolved from local context to global dependency modeling.
- Attention enables models to dynamically focus on what matters most.
- Transformers fully embrace attention, replacing recurrence with parallel computation.
- Today, attention forms the foundation of state-of-the-art models across domains.

Thank you for listening!

Questions?

Image Sources

1. https://www.researchgate.net/profile/Najla-Azaiez/publication/373603670/figure/fig1/
2. https://www.researchgate.net/profile/Dongkwon-Han/publication/346219836/figure/fig1/
3. https://dataaspirant.com/wp-content/uploads/2025/04/1-2.png
4. https://www.aisa.digital/wp-content/uploads/2020/09/multilingual-chatbot-700x613.png
5. https://www.equalexperts.com/wp-content/uploads/2021/08/Haystack-TN.png.webp
6. https://www.linkedin.com/pulse/introduction-text-summarization-nlp-souvik-bose
7. https://www.researchgate.net/figure/Analogical-reasoning-on-vectors-a-king-man-womanqueen-and-b_fig1_358432453
8. https://medium.com/@CallMeTwitch/building-a-neural-network-zoo-from-scratch-the-recurrent-neural-network-9357b43e113c
9. https://miro.medium.com/v2/resize:fit:1400/format:webp/1*I5iwCL8zDo9OppBPsprwTw.png
10. https://lena-voita.github.io/resources/lectures/seq2seq/attention/general_scheme-min.png
11. https://naokishibuya.github.io/blog/2021-10-31-transformers-positional-encoding/images/positional-encoding-rough.png
12. https://jalammar.github.io/illustrated-transformer/
13. https://www.researchgate.net/profile/Peter-Sutor/publication/332679657/figure/fig1/AS:809485488640000@1570007788866/The-classical-king-woman-man-queen-example-of-neural-word-embeddings-in-2D-It.png
14. https://arxiv.org/abs/1706.03762
15. https://www.mdpi.com/2075-4418/13/14/2459
16. https://www.youtube.com/@AndrejKarpathy