Deep Learning II

The restaurant refused to serve me a ham sandwich because it only cooks vegetarian food. In the end, they just gave me two slices of bread. Their ambiance was just as good as the food and service.

Quote from Prince, Understanding Deep Learning, 2023

The restaurant refused to serve me a ham sandwich because **it** only cooks vegetarian food. In the end, **they** just gave me two slices of bread. **Their** ambiance was just as good as the food and service.

What does each **bolded** word refer to?

Is this a positive review or a negative one?

What kind of food do they serve at this restaurant?

Quote from Prince, Understanding Deep Learning, 2023

Types of NLP problems

Text Classification

- Sentiment Analysis
- Topic categorization
- Extractive question answering

Sequence-to-sequence

- Machine translation (text-to-text)
- Summarization (text-to-text)
- Speech-to-text, text-to-speech

- ## Generative models
  - Chatbots
  - Generative question answering
  - Code generation

Encoder models

BERT (Bidirectional Encoder Representations from Transformers)

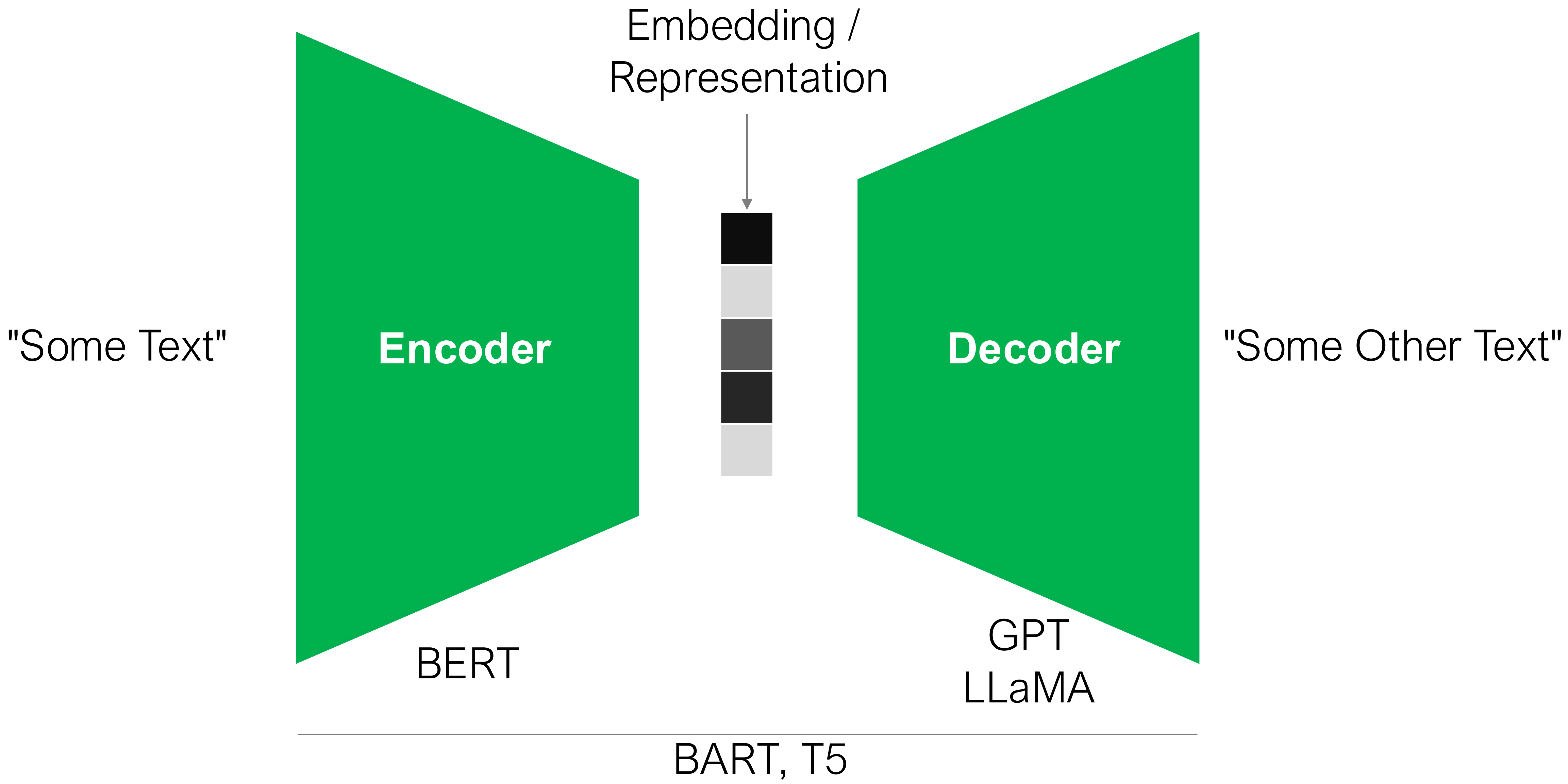
Encoder-decoder models

BART (Bidirectional and Auto-Regressive Transformers)
T5 (Text-to-Text Transfer Transformer)

Decoder models

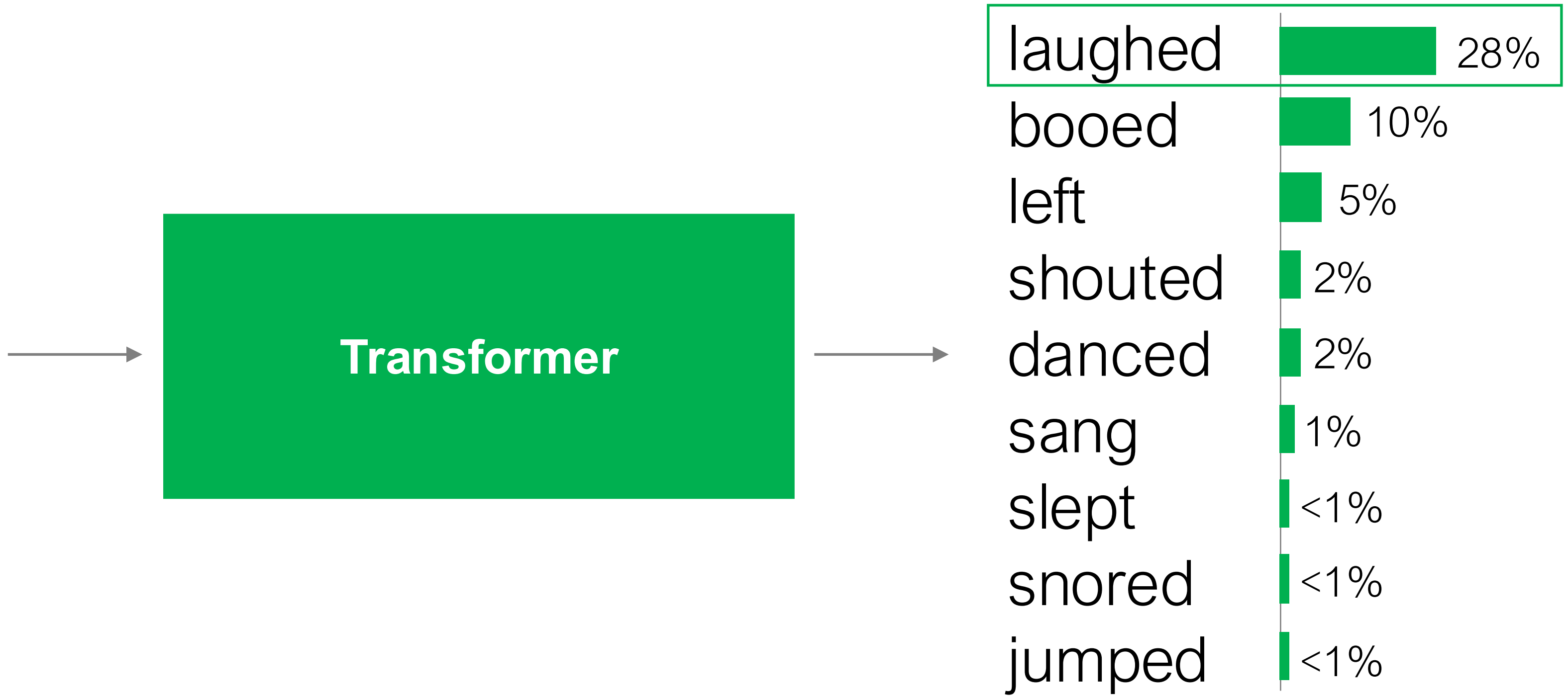
GPT (Generative Pre-trained Transformer)
LLaMA (Large Language Model Meta AI)

Types of Transformers



Example: Predict the next word

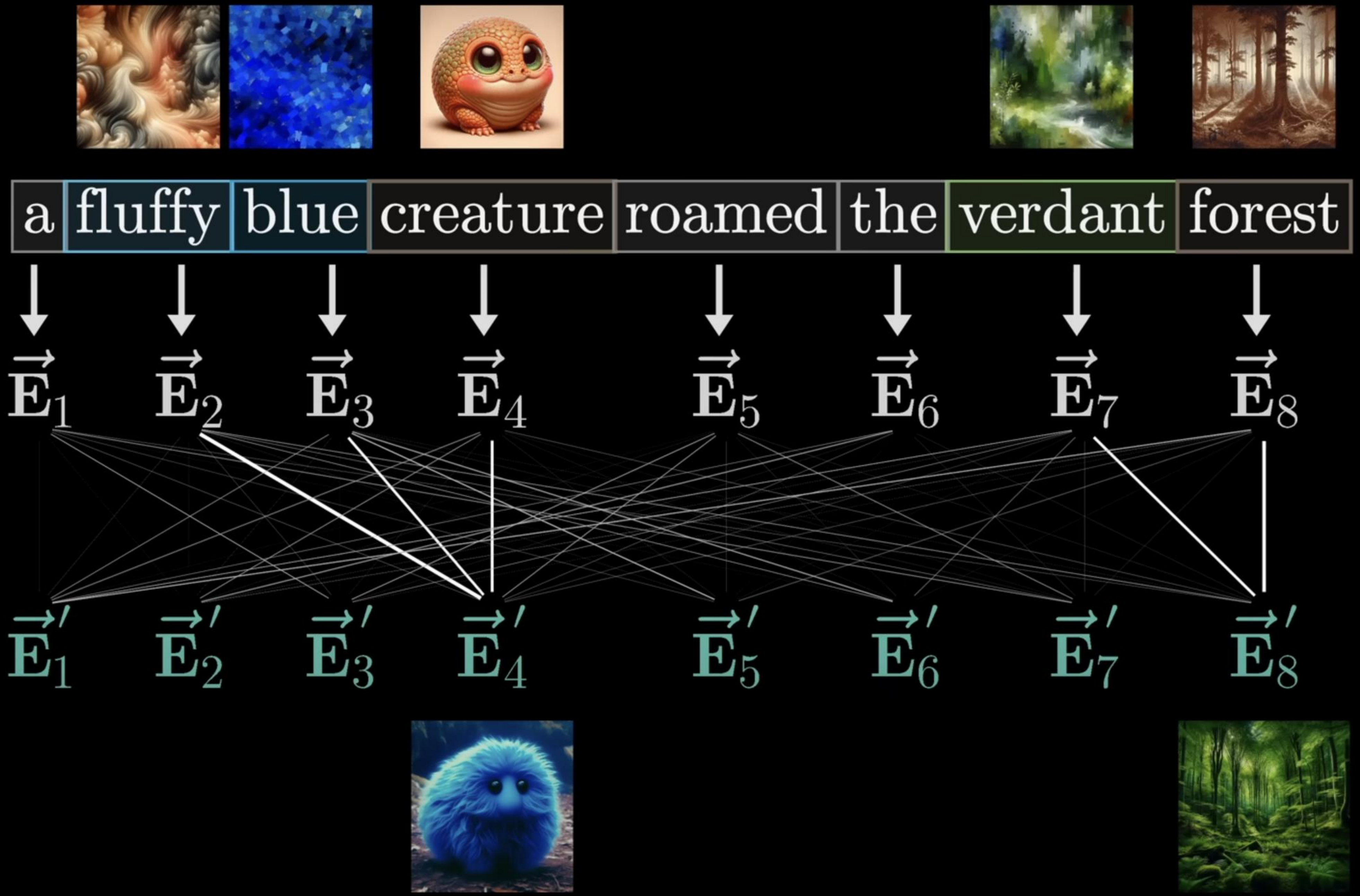
After hearing the terrible pun, the data science students, to the surprise of their instructor, all suddenly ________



If we repeat this again and again, we can generate text

Transformer intuition

The quick brown fox jumped over the lazy dog



Create improved embeddings by incorporating meaning from other words

Image from 3Blue1Brown (link)

Transformer steps and components

1. Tokenization – convert text to numbers
2. Position embedding – encode the order of the tokens
3. Self-attention – enable learning from context
    1. Queries
    2. Keys
    3. Values
4. Layer normalization
5. Multilayer perceptron

Tokenization

Converts text to numerical IDs for each token

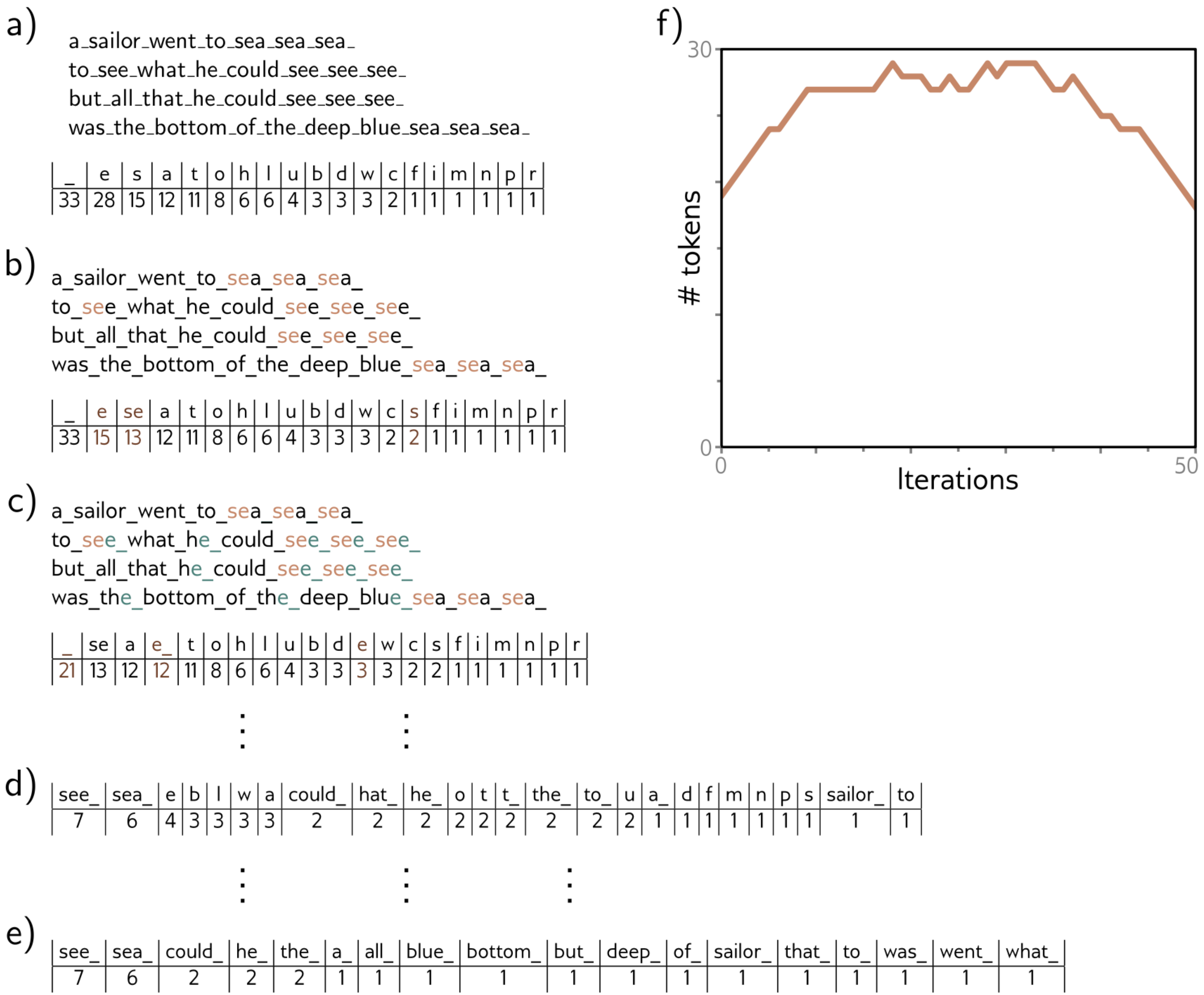
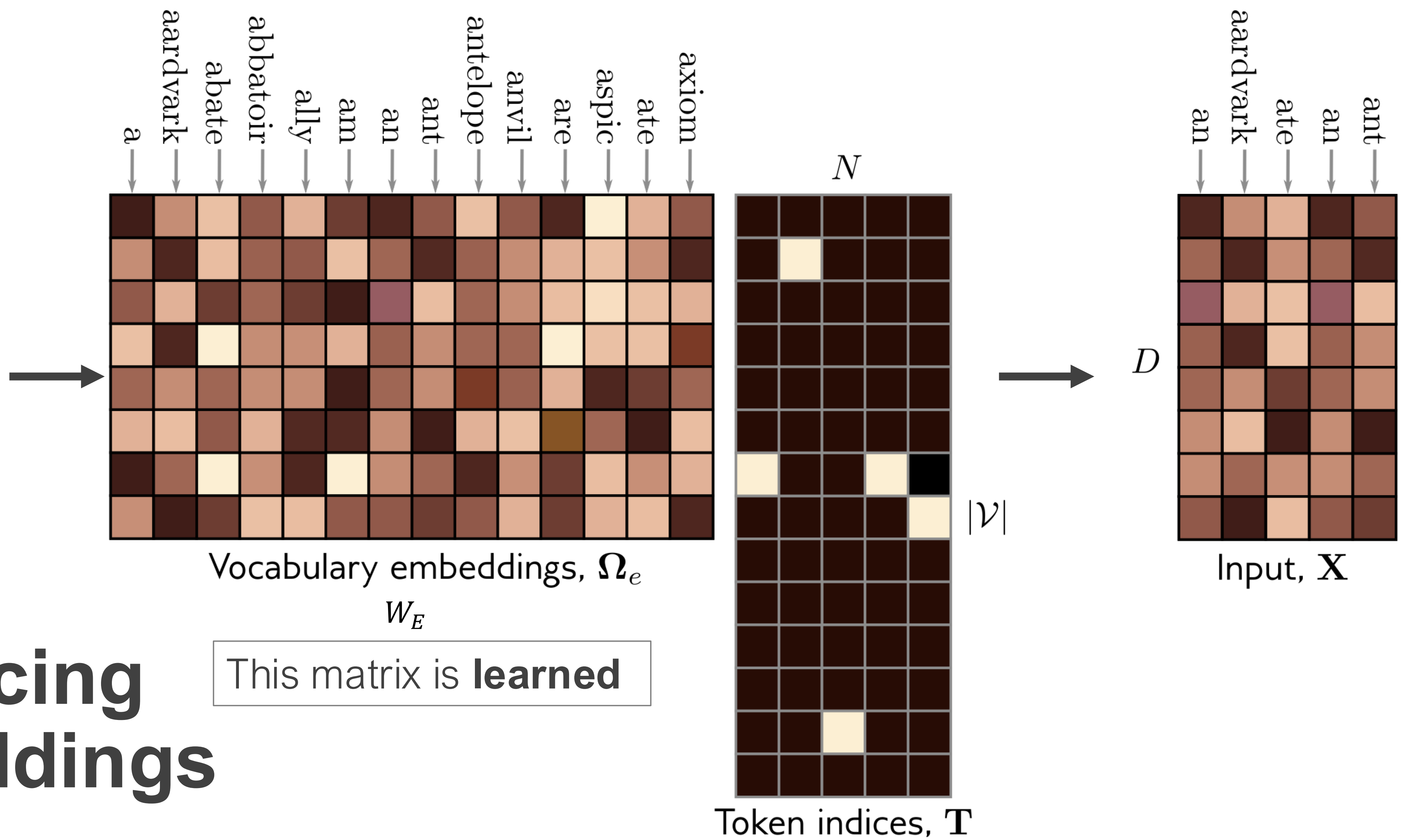


Image from Prince, Understanding Deep Learning, 2023

Token indices are transformed into a sequence of embeddings (the embeddings are learned)

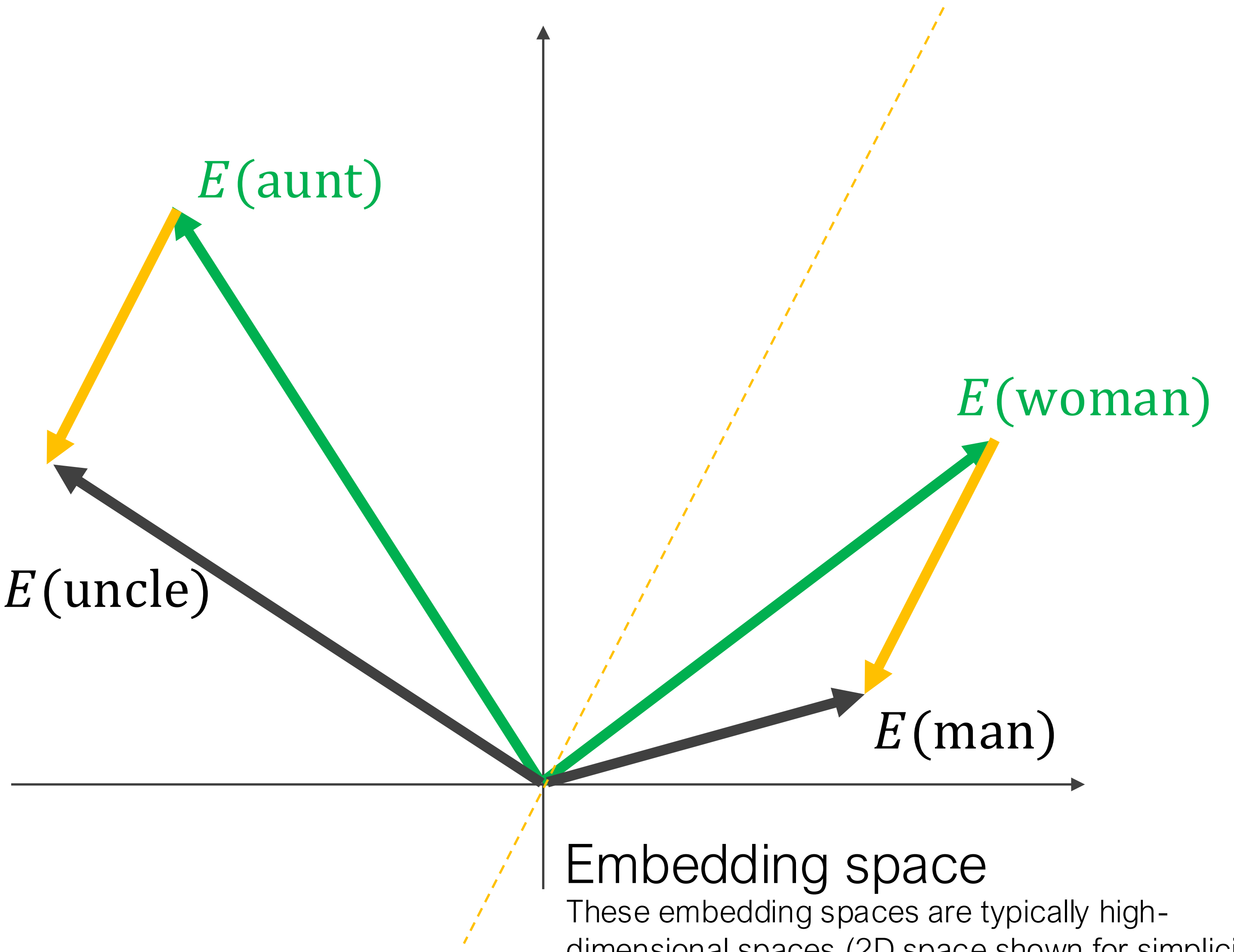
"an aardvark ate an ant"



This matrix is **learned**

Producing embeddings

Image from Prince, Understanding Deep Learning, 2023



Embedding space

These embedding spaces are typically high-dimensional spaces (2D space shown for simplicity)

These embeddings spaces start off random and through training gain semantic meaning

Directions in these high dimensional spaces may begin to represent concepts

Positional Encoding

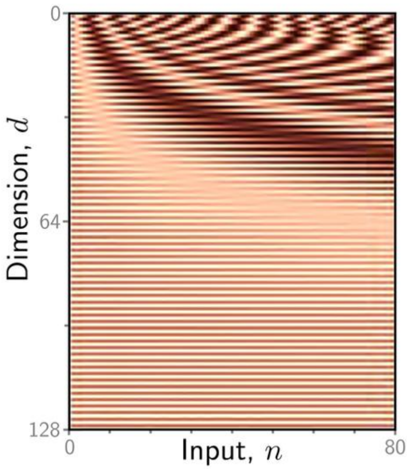
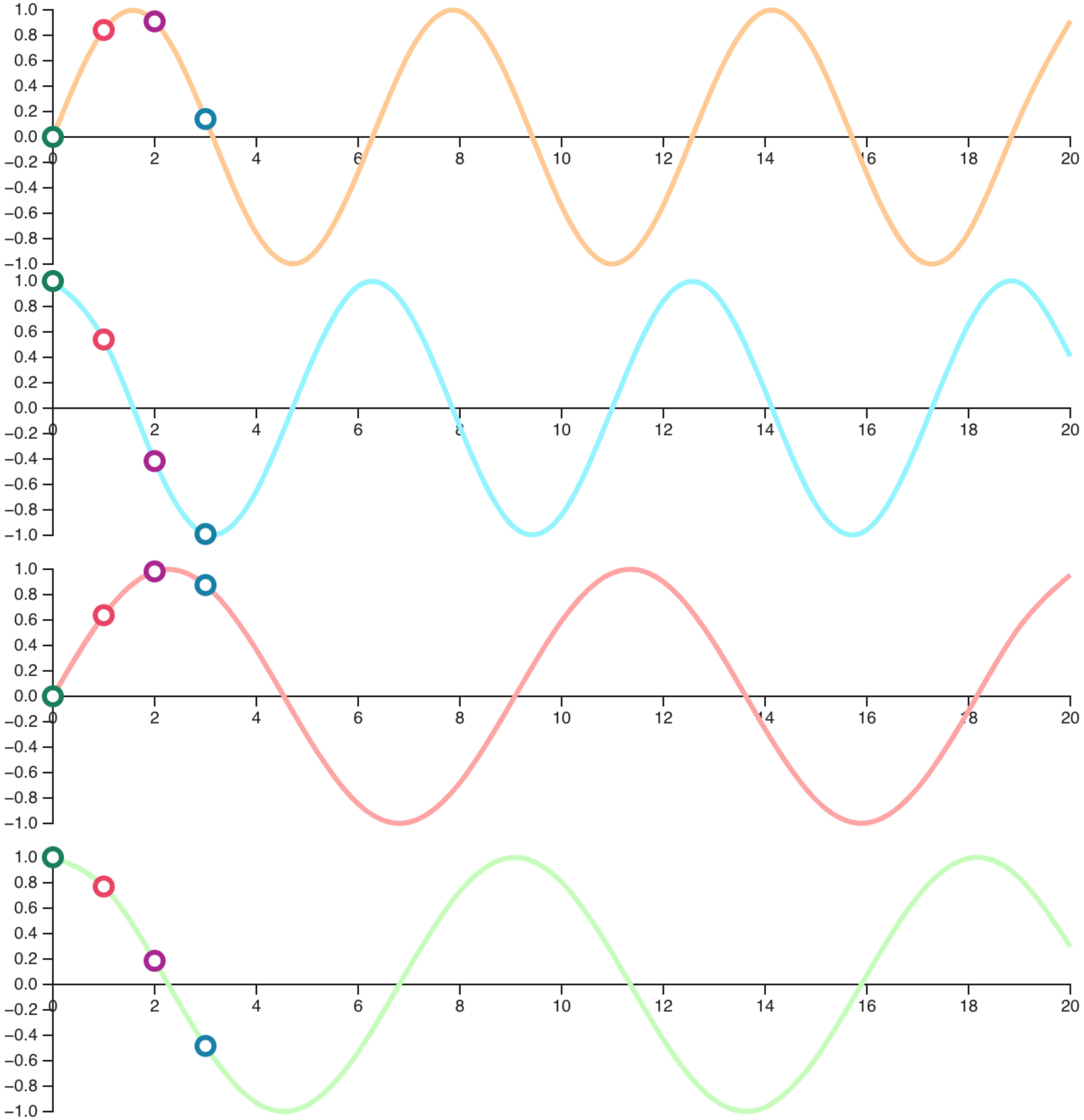


Image from Prince, Understanding Deep Learning, 2023



p0	p1	p2	p3	
0.000	0.841	0.909	0.141	i=0
1.000	0.540	-0.416	-0.990	i=1
0.000	0.638	0.983	0.875	i=2
1.000	0.770	0.186	-0.484	i=3

Positional Encoding

$$PE_{(pos,2i)} = sin(\frac{pos}{10000^{2i/d_{\text{model}}}})$$

$$PE_{(pos,2i+1)} = cos(\frac{pos}{10000^{2i/d_{\text{model}}}})$$

Settings: d = 50

The value of each positional encoding depends on the *position* (*pos*) and *dimension* (*d*). We calculate result for every *index* (*i*) to get the whole vector.

Image source: https://erdem.pl/2021/05/understanding-positional-encoding-in-transformers "quick brown fox"

Tokenization

[423,23,342]

Token (word) embeddings

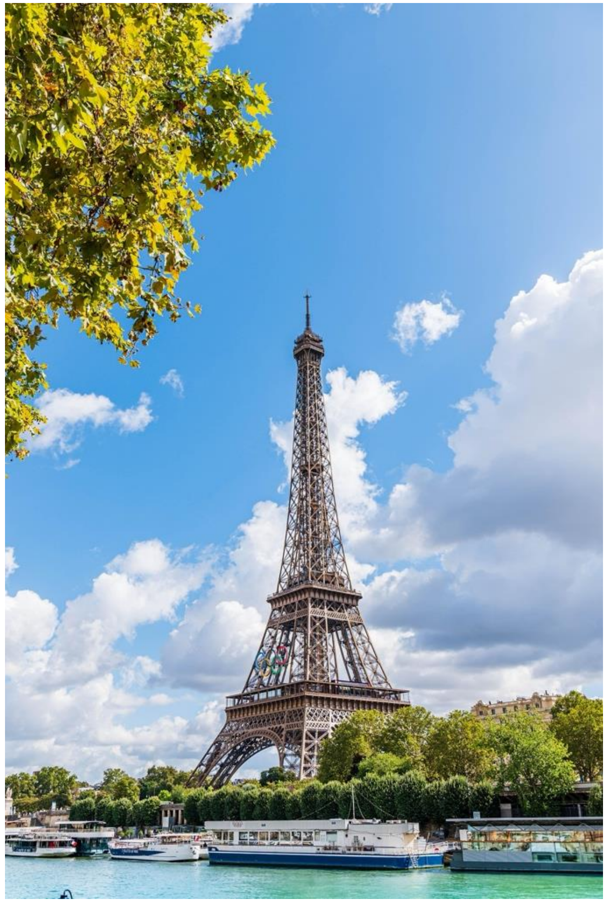
Token (word) embeddings		Positional encoding		Token and position embedding
1.6		-1.0		0.6
3.2		0.50		3.7
-0.4		-0.10		-0.50
-4.9	+	0.0	=	-4.9
0.42		0.35		0.77
1.6		-0.21		1.39

What it is and where it is

Context typically adds meaning

tower

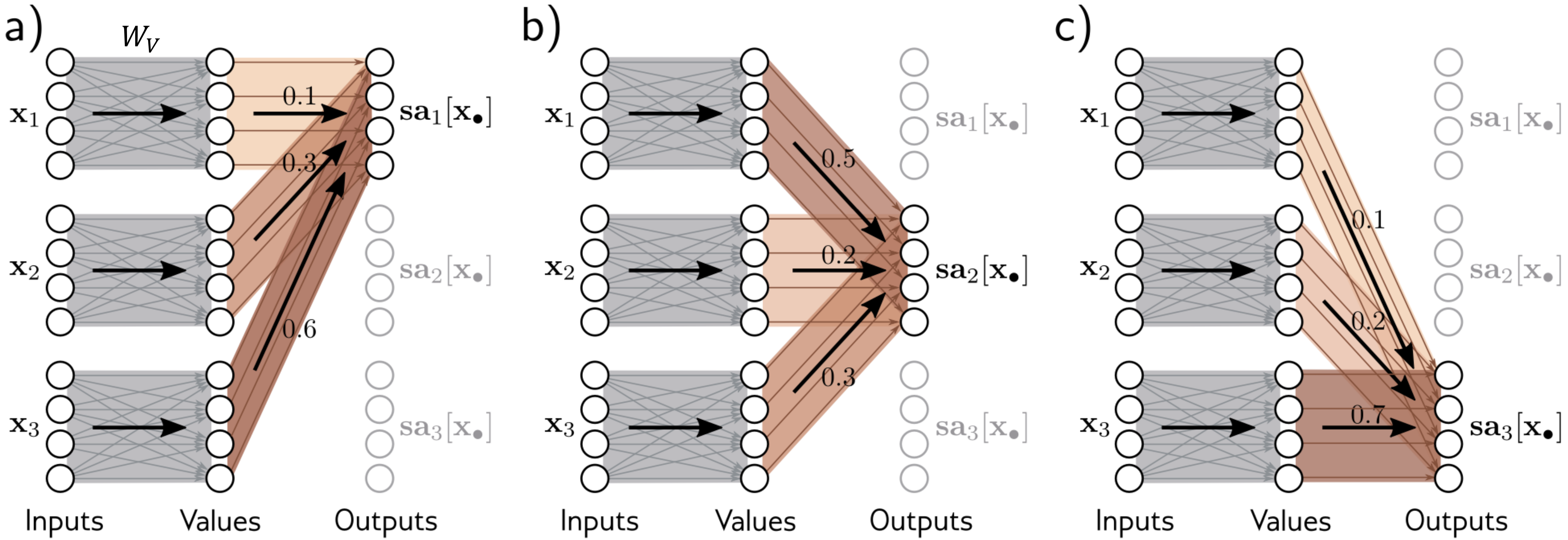
Eiffel tower



miniature Eiffel tower

Concept from 3Brown1Blue
Images from Pixabay

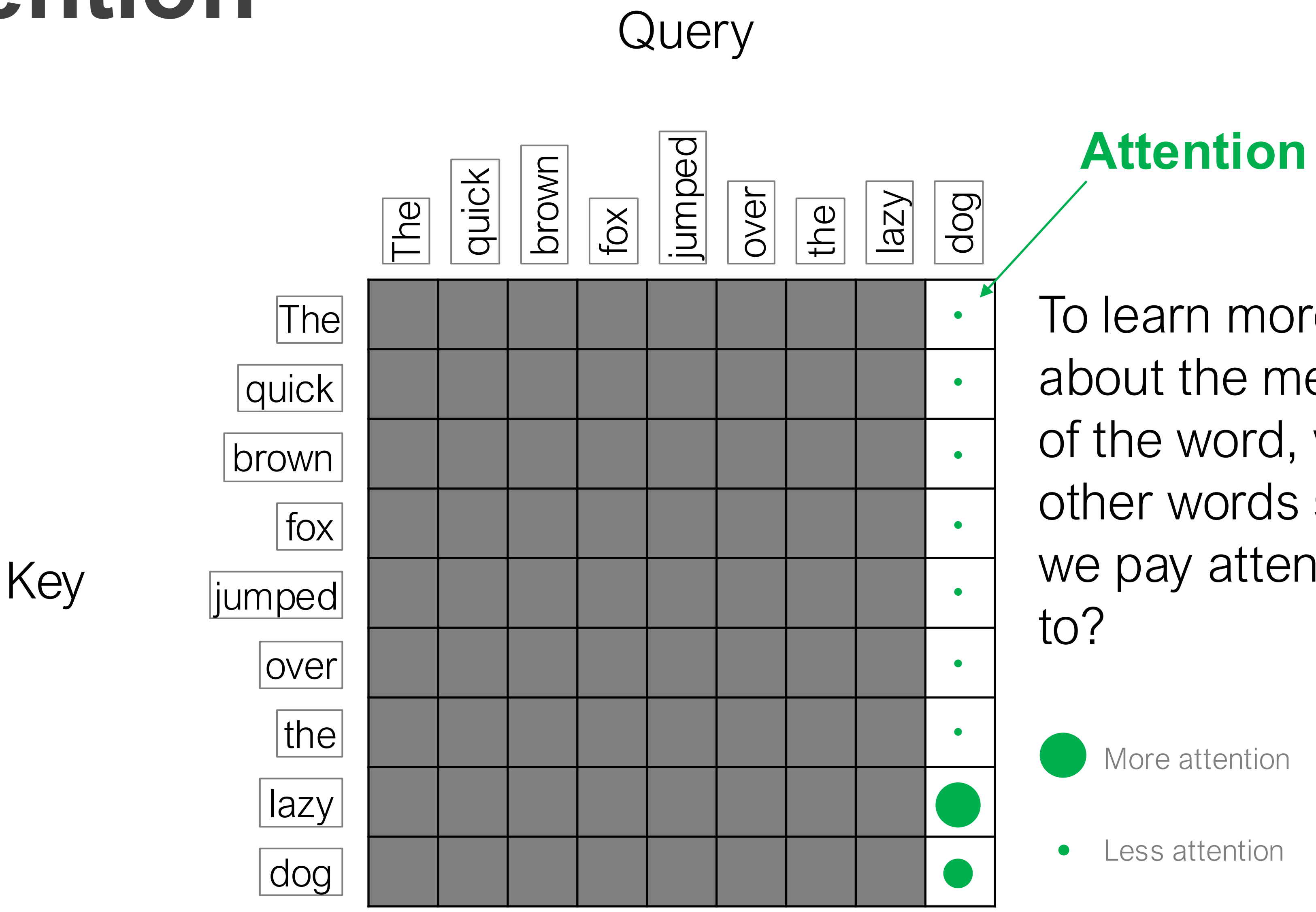
Self-attention



Inputs are the word/token embeddings

Image from Prince, Understanding Deep Learning, 2023

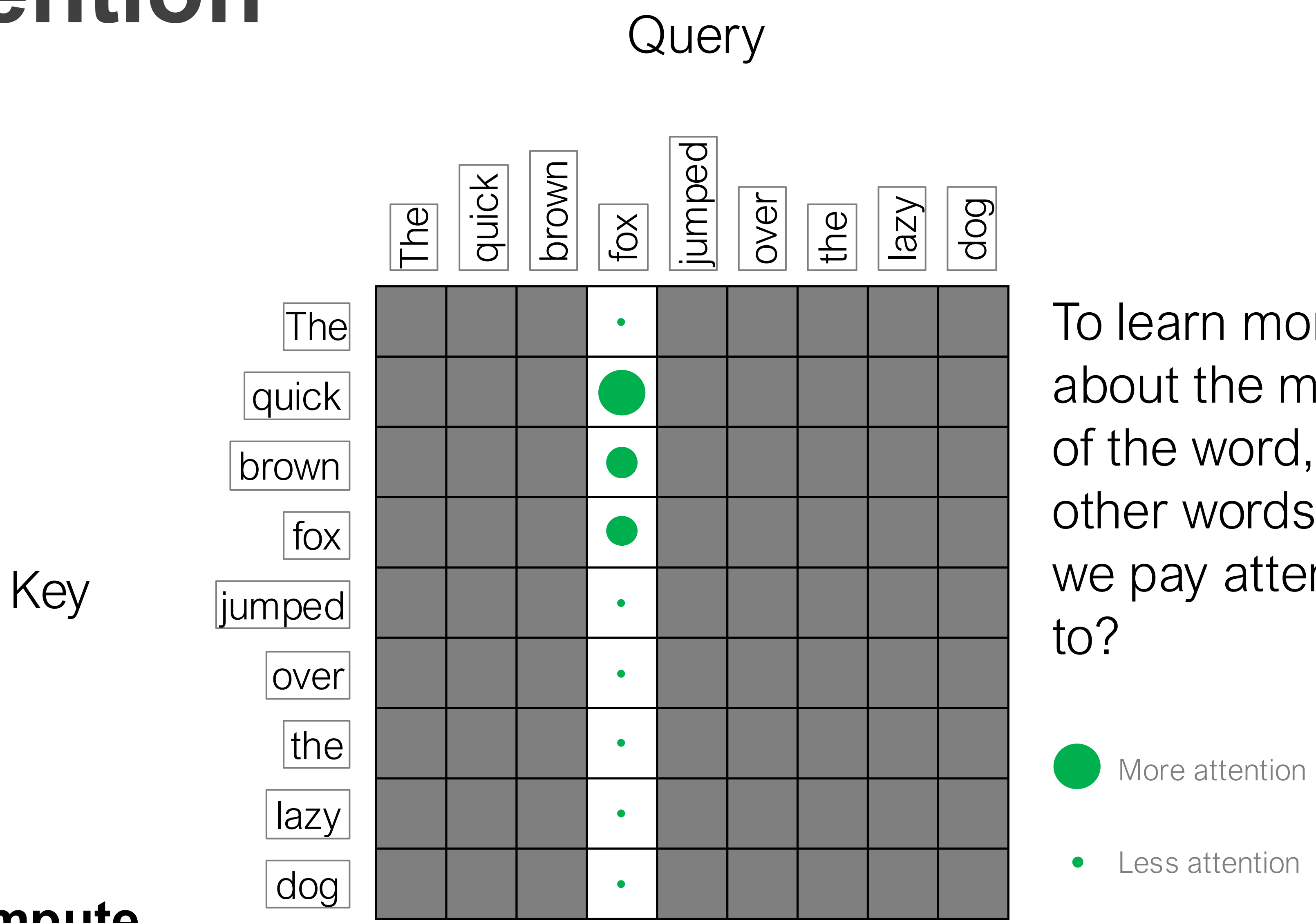
Self-attention



To learn more about the meaning of the word, which other words should we pay attention to?

Figure adapted from 3Blue1Brown (link)

Self-attention



To learn more about the meaning of the word, which other words should we pay attention to?

How do we compute self-attention?

Figure adapted from 3Blue1Brown (link)

Query

	The	quick	brown	fox	jumped	over	the	lazy	dog
	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$
	$\downarrow W_q$	$\downarrow W_q$	$\downarrow W_q$	$\downarrow W_q$	$\downarrow W_q$	$\downarrow W_q$	$\downarrow W_q$	$\downarrow W_q$	$\downarrow W_q$
	$\mathbf{q}_1$	$\mathbf{q}_2$	$\mathbf{q}_3$	$\mathbf{q}_4$	$\mathbf{q}_5$	$\mathbf{q}_6$	$\mathbf{q}_7$	$\mathbf{q}_8$	$\mathbf{q}_9$
The $\mathbf{x}_1 \xrightarrow{W_k} \mathbf{k}_1$	$\mathbf{k}_1^T\mathbf{q}_1$	$\mathbf{k}_1^T\mathbf{q}_2$	$\mathbf{k}_1^T\mathbf{q}_3$						
quick $\mathbf{x}_2 \xrightarrow{W_k} \mathbf{k}_2$	$\mathbf{k}_2^T\mathbf{q}_1$	$\mathbf{k}_2^T\mathbf{q}_2$	$\mathbf{k}_2^T\mathbf{q}_3$						
brown $\mathbf{x}_3 \xrightarrow{W_k} \mathbf{k}_3$	$\mathbf{k}_3^T\mathbf{q}_1$	$\mathbf{k}_3^T\mathbf{q}_2$	$\mathbf{k}_3^T\mathbf{q}_3$						
fox $\mathbf{x}_4 \xrightarrow{W_k} \mathbf{k}_4$				$\ddots$					
jumped $\mathbf{x}_5 \xrightarrow{W_k} \mathbf{k}_5$									
over $\mathbf{x}_6 \xrightarrow{W_k} \mathbf{k}_6$									
the $\mathbf{x}_7 \xrightarrow{W_k} \mathbf{k}_7$									
lazy $\mathbf{x}_8 \xrightarrow{W_k} \mathbf{k}_8$									
dog $\mathbf{x}_9 \xrightarrow{W_k} \mathbf{k}_9$									

Keys $\mathbf{k}_i = W_k \mathbf{x}_i$

$\mathrm{K} = [\mathbf{k}_1, \mathbf{k}_2, \dots, \mathbf{k}_9]$

Queries $\mathbf{q}_i = W_q \mathbf{x}_i$

$\mathrm{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_9]$

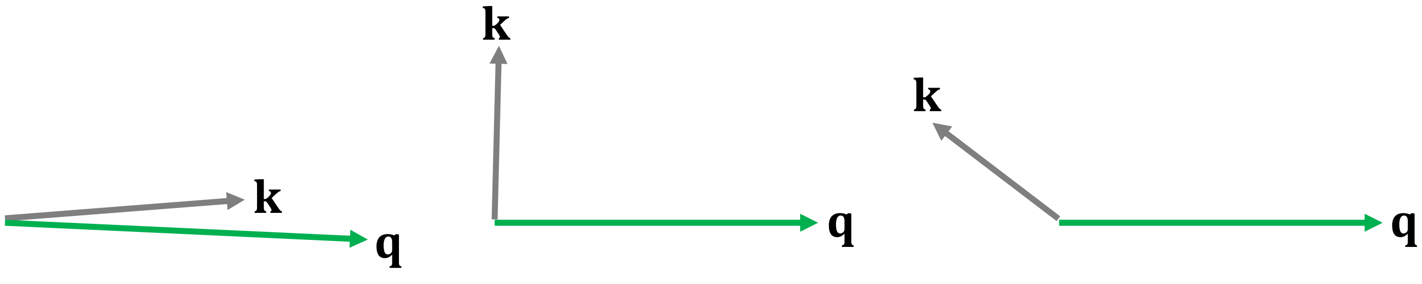
Figure adapted from 3Blue1Brown (link)

Queries and Keys

The quick brown fox jumped over the lazy dog

We measure the similarity between the key ($\mathbf{k}$) and the query ($\mathbf{q}$) through the dot product:

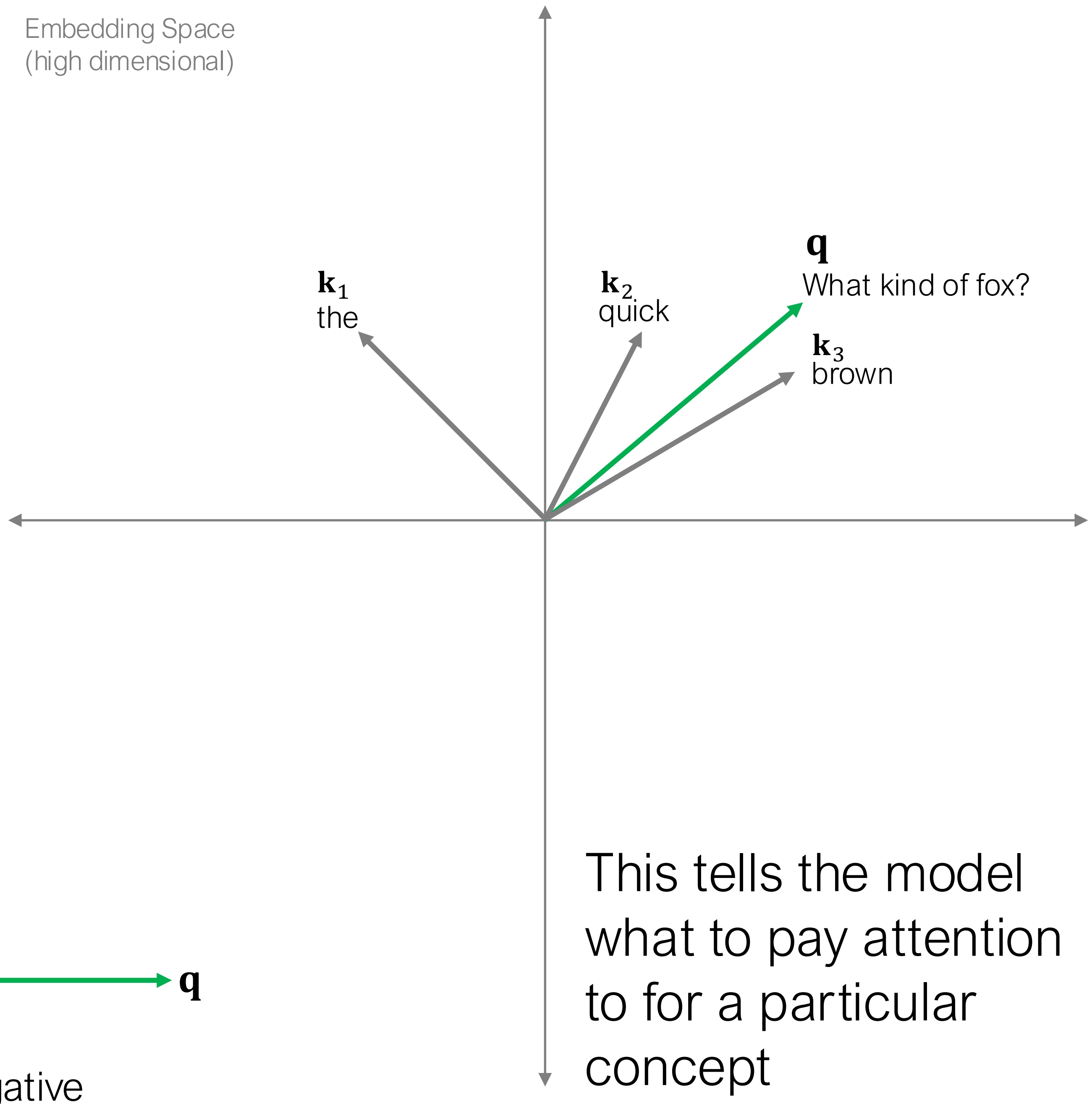
$$\mathbf{k} \cdot \mathbf{q} = \mathbf{k}^T \mathbf{q}$$



$\mathbf{k}^T \mathbf{q}$ positive

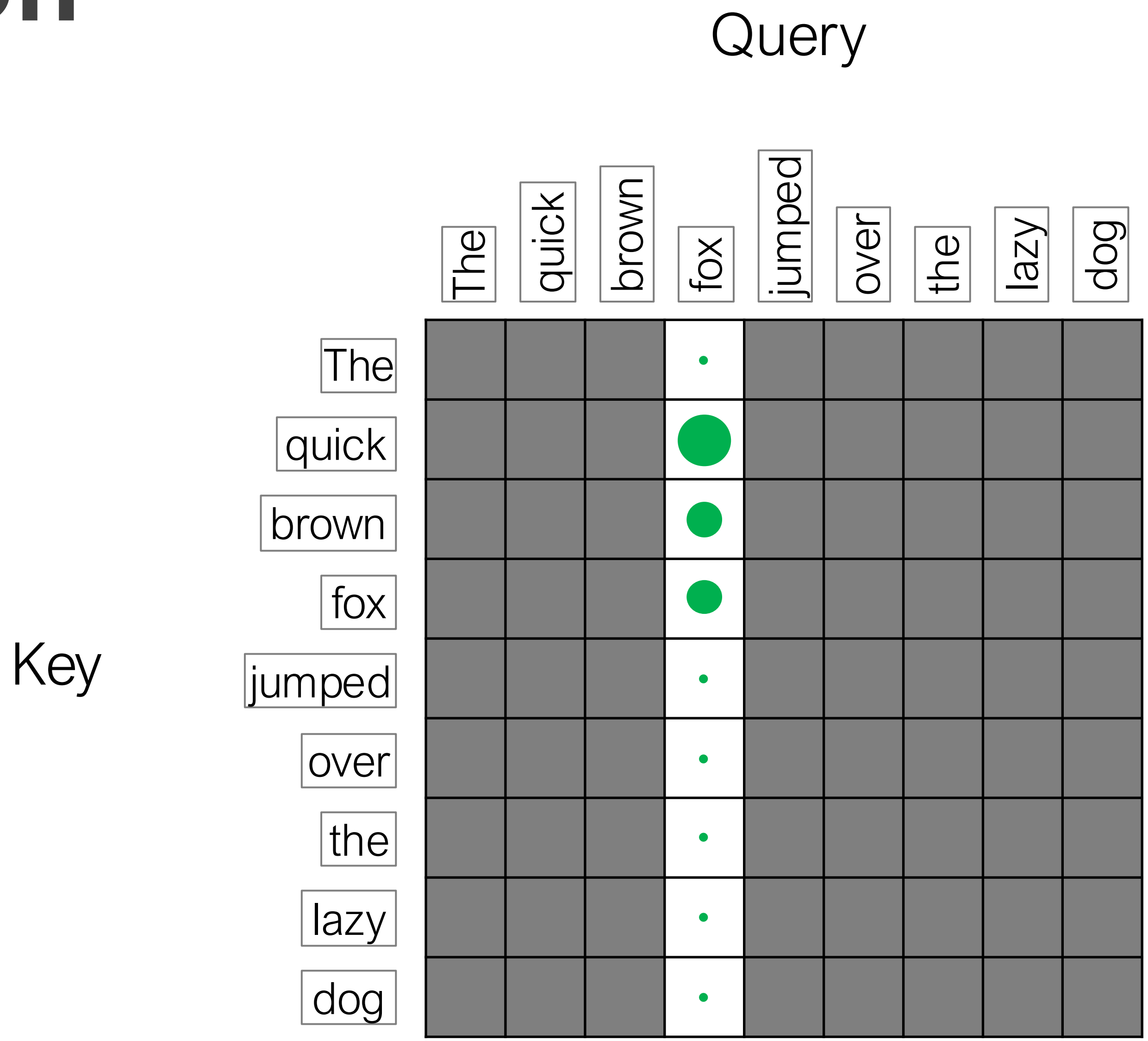
$\mathbf{k}^T \mathbf{q}$ zero

$\mathbf{k}^T \mathbf{q}$ negative



This tells the model what to pay attention to for a particular concept

Attention



To learn more about the meaning of the word, which other words should we pay attention to?

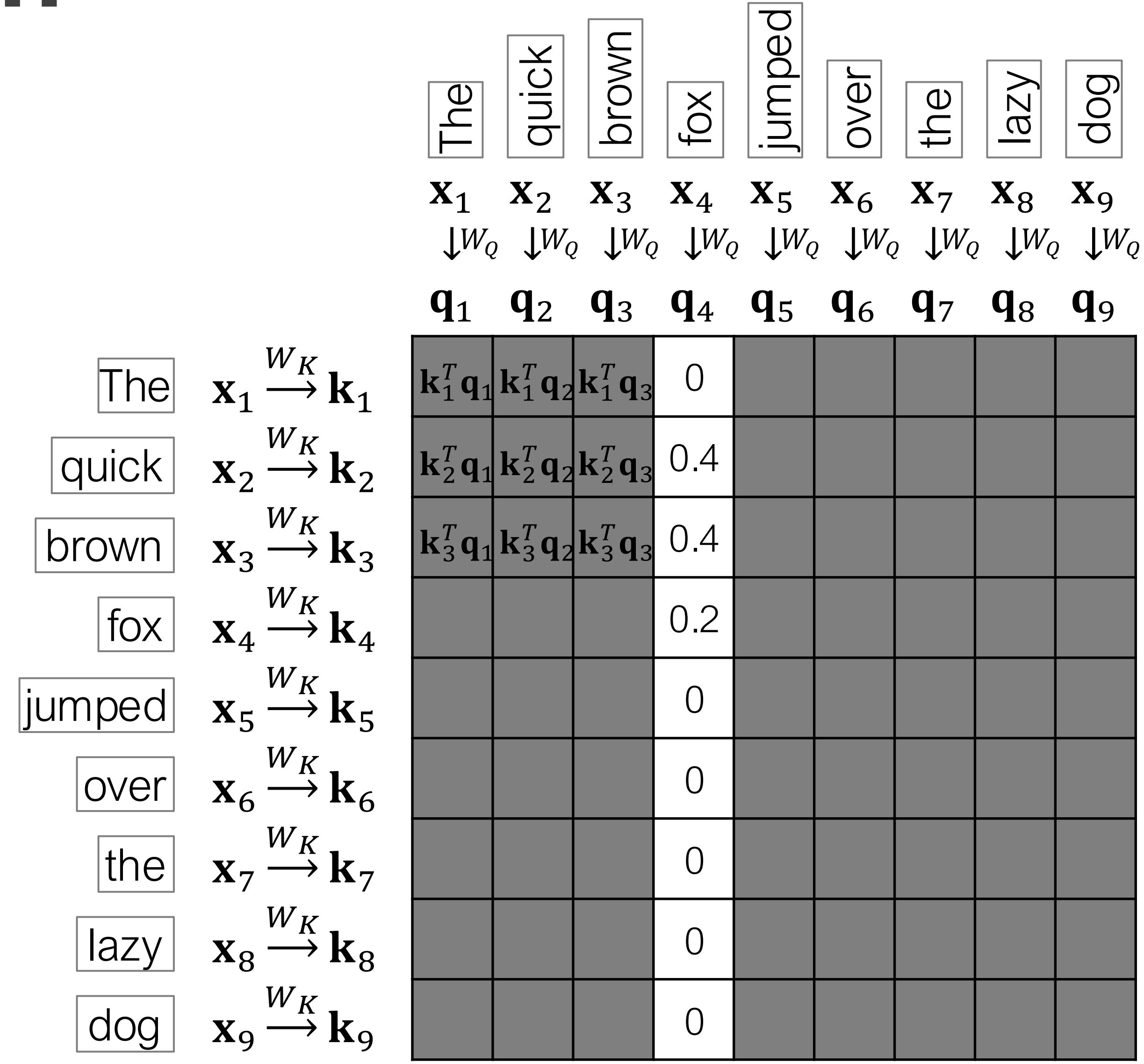
More attention

Less attention

Figure adapted from 3Blue1Brown (link)

Attention

Query



$K^T\mathbf{q}_j$

Queries $\mathbf{q}_i = W_Q\mathbf{x}_i$

$Q = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_9]$

Keys $\mathbf{k}_i = W_K\mathbf{x}_i$

$K = [\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_9]$

Attention weights

$a[\mathbf{x}_i, \mathbf{x}_i] = \text{softmax}_i(K^T\mathbf{q}_j)$

Note: we apply softmax to each column so it sums to 1

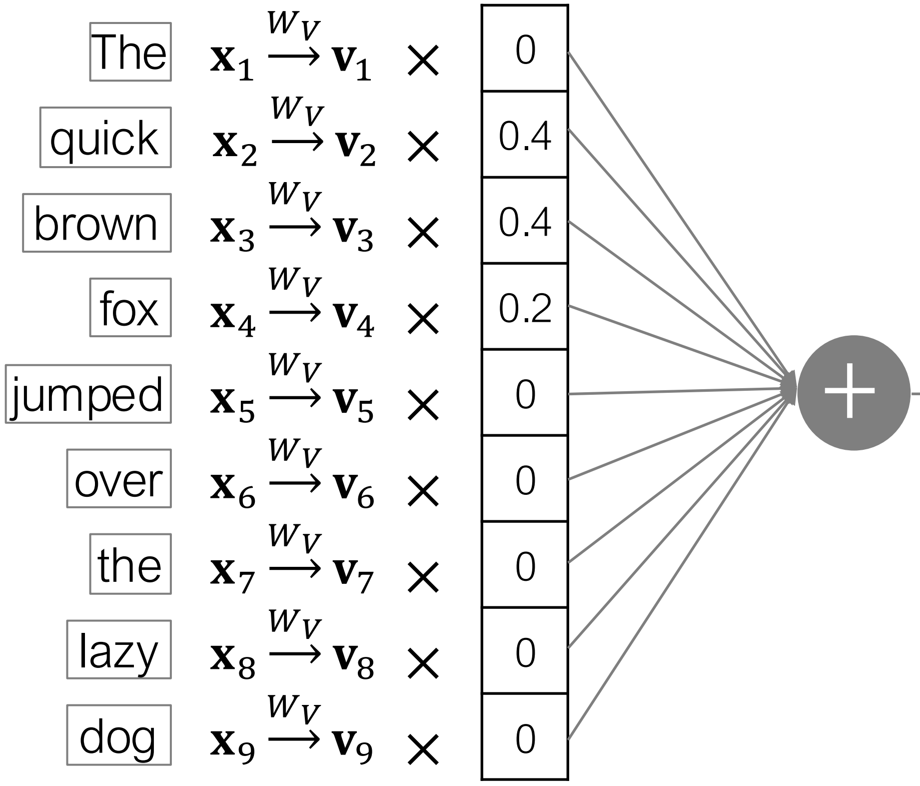
Figure adapted from 3Blue1Brown (link)

Self-attention

Weighted sum of values

(based on self-attention weights)

$a[\mathbf{x}_i, \mathbf{x}_i]$



Values

$$\mathbf{v}_i = W_V \mathbf{x}_i$$

$$\mathrm{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_9]$$

This simply says – weight the embeddings based on the self-attention weights

$$\mathbf{sa}_n[x_1, \dots, x_N] = \sum_{m=1}^{N} a[\mathbf{x}_m, \mathbf{x}_n]\mathbf{v}_m$$

Figure adapted from 3Blue1Brown (link)

Self-attention in matrix form

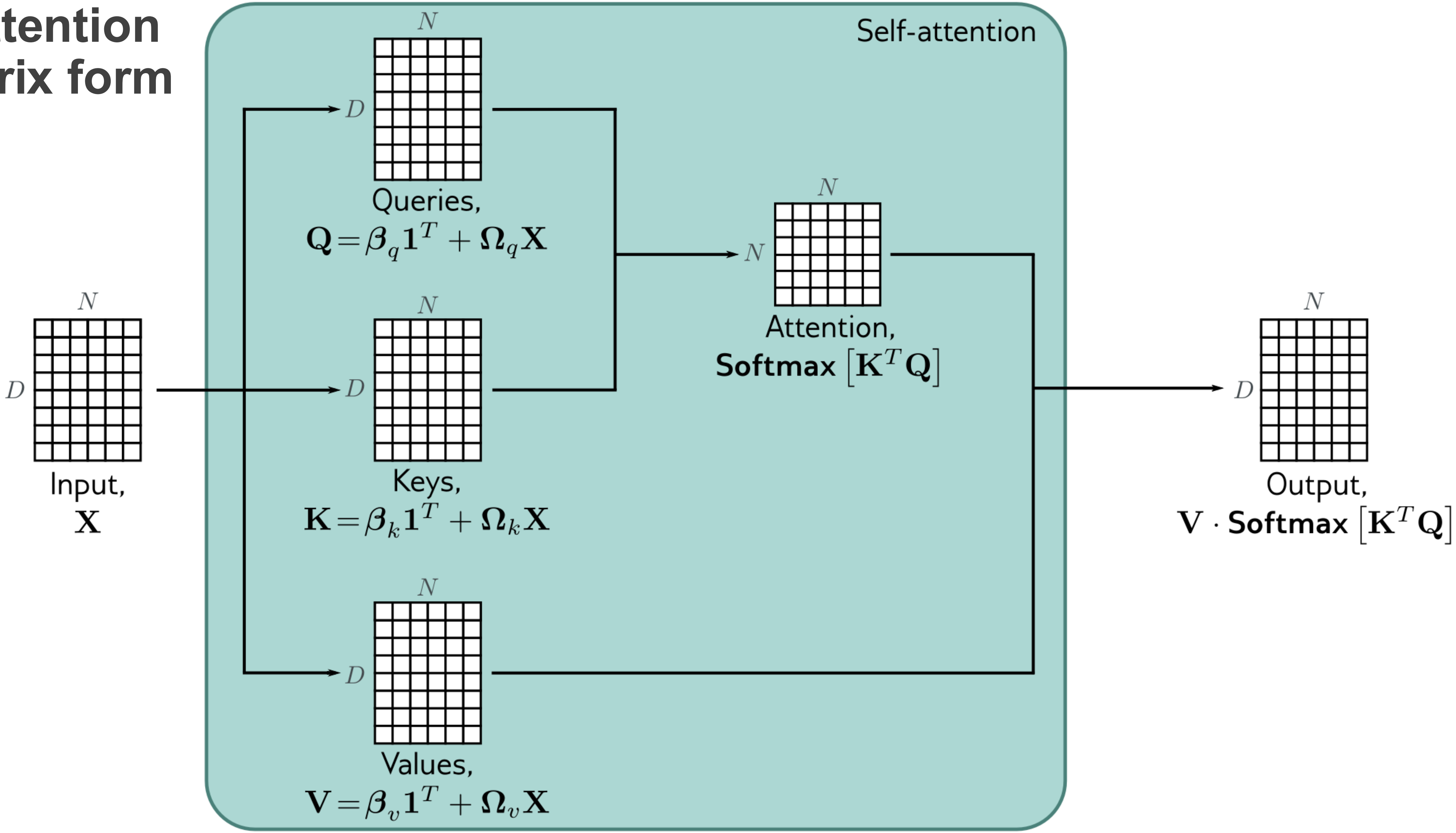


Image from Prince, Understanding Deep Learning, 2023

Multiheaded attention

Each head may explore different types of relevant information in the rest of the text

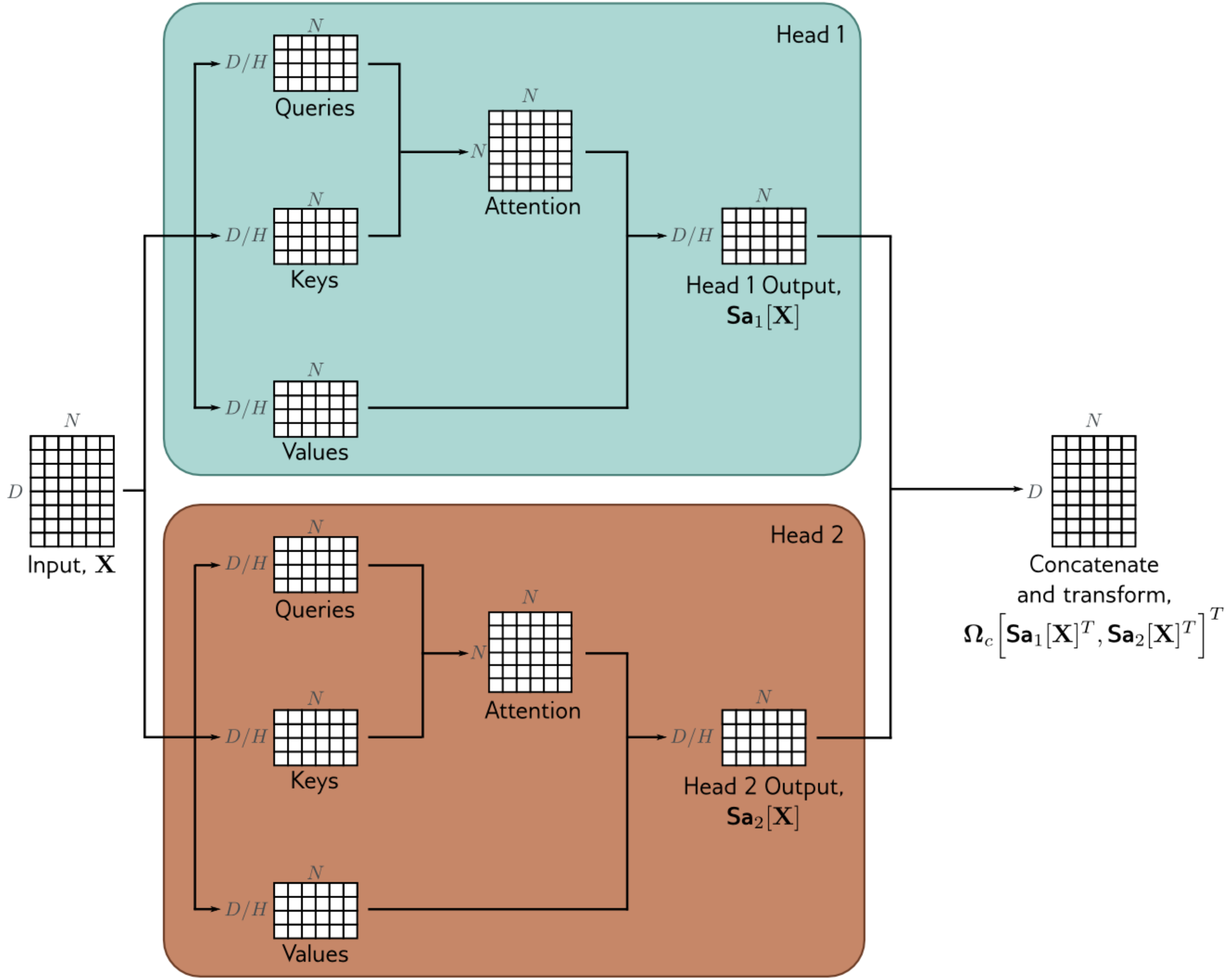
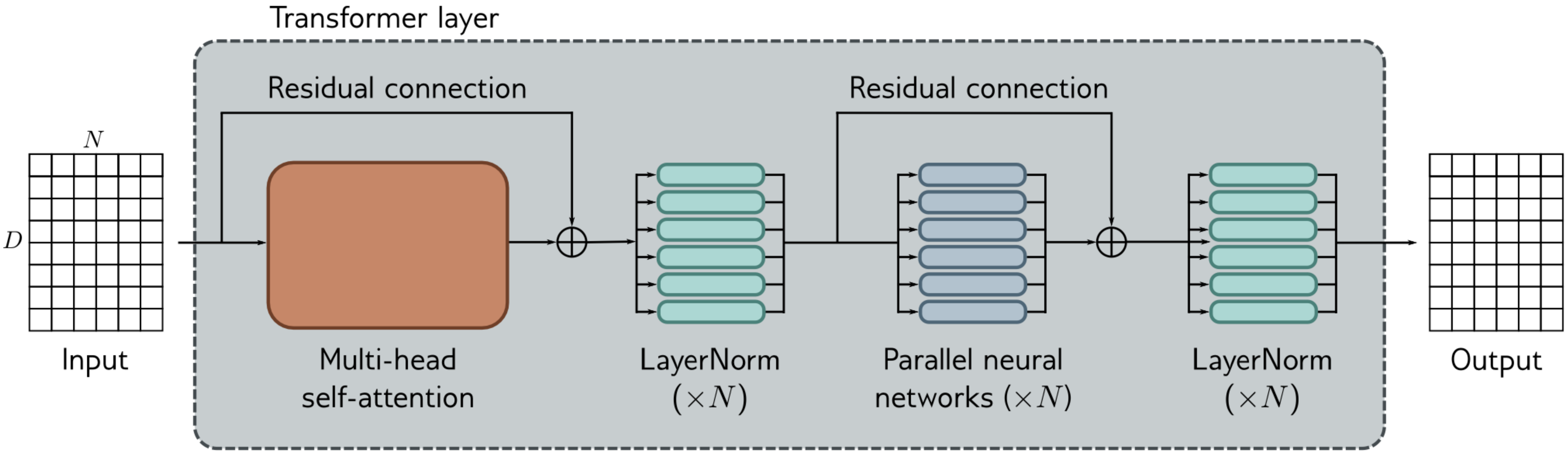


Image from Prince, Understanding Deep Learning, 2023

The Transformer



Residual connections can be thought of as:

new representation = old representation + correction/improvement

(token + position embedding) (self-attention output)

Image from Prince, Understanding Deep Learning, 2023

Vision Transformer (ViT)

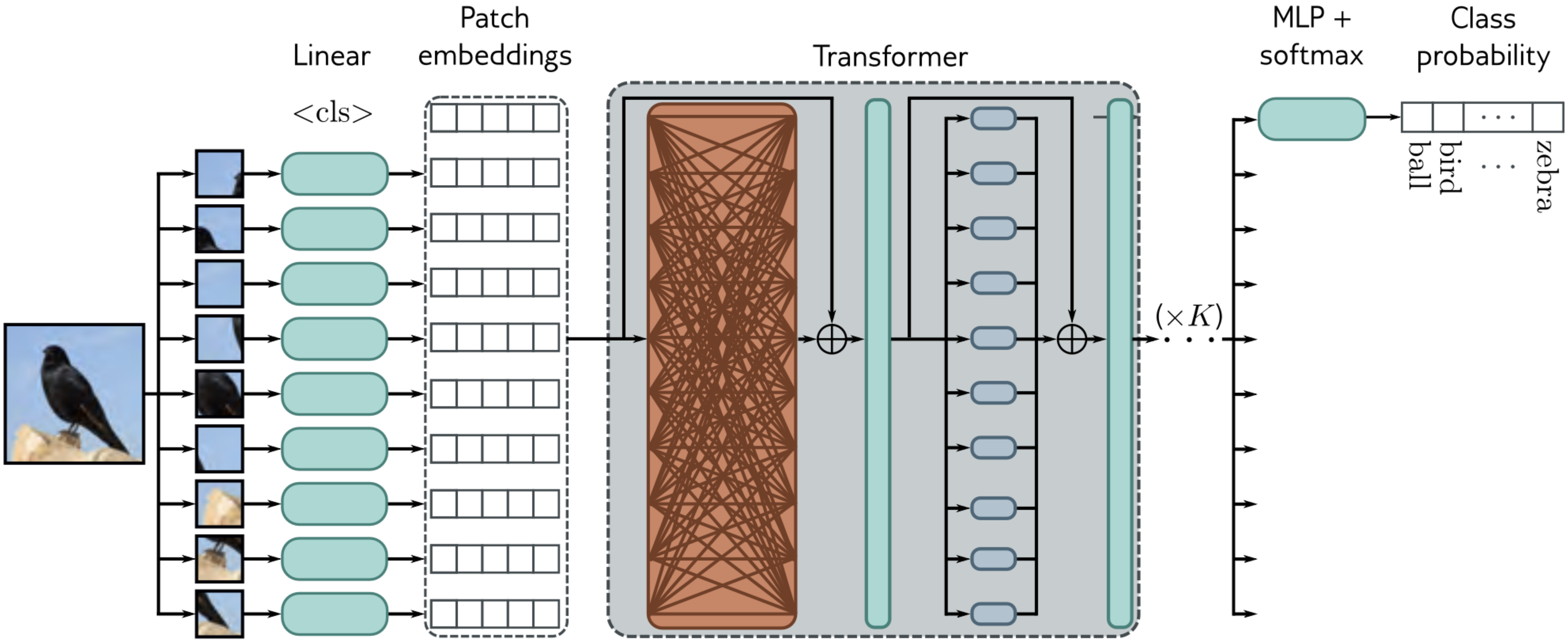


Image from Prince, Understanding Deep Learning, 2023

Supervised Learning Techniques

Covered so far

- Linear Regression
- K-Nearest Neighbors
- Logistic Regression
- Linear/Quadratic Discriminant Analysis
- Naïve Bayes
- Decision Trees
- Random Forests
- Gradient Boosted Decision Trees
- Neural networks and deep learning

Ensemble approaches, including bagging, boosting, and stacking, can be used with numerous machine learning techniques, often CART

Appropriate for:
- Classification
- Regression

Supervised learning in practice

Preprocessing
Explore & prepare data

Data Visualization and Exploration

Identify patterns that can be leveraged for learning

Data Cleaning

- Missing data
- Noisy data
- Erroneous data

Scaling (Standardization)

Prepare data for use in scale-dependent algorithms.

Feature Extraction

Dimensionality reduction eliminates redundant information

Model training

1 Supervised Learning Models: Linear models and KNN
(enough to get started using supervised learning)

5 Other algorithms and concepts:

- Generative vs discriminative models
- Parametric vs nonparametric models
- Model ensembles
- Feature/representation learning (neural networks, deep learning)

4 How to control model overfit: regularization strategies for model refinement

Iteratively fine tune the model

Performance evaluation

Make a prediction on validation data

2 Evaluating model performance and comparing models

Classification

3 How to make decisions using models

Regression

MSE, explained variance, R^2