Discussion Mini Lecture 10

Transformers

Why is Attention All You Need?

CS 189/289A, Fall 2025 @ UC Berkeley
Sara Pohland

Concepts Covered

1. Transformer Architecture
2. Modeling Attention

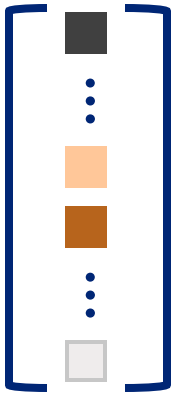
Overview of a Transformer

1. Transformer Architecture
2. Modeling Attention

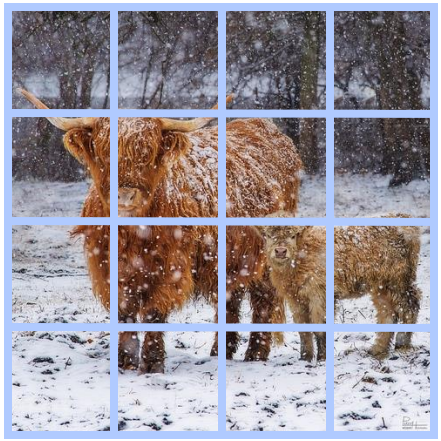
How do we represent our data?

Image

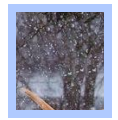
Naïve: Vector of Pixels



Better: Patches



Token Embeddings



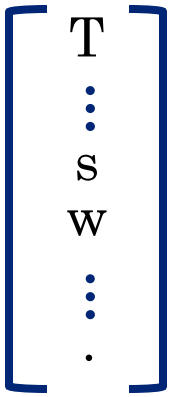
D-dimensional vector

N = 16 embeddings

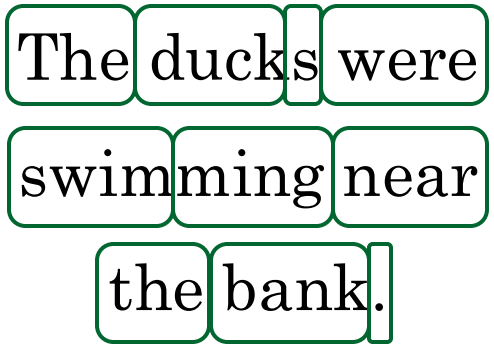
Text

The ducks were swimming near the bank.

Naïve: Vector of Characters



Better: Word Parts



Token Embeddings

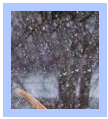
D-dimensional vector

N = 10 embeddings

How do we represent our data?

Image

Token Embeddings

 → D-dimensional vector

N = 16 embeddings

$$\mathbf{X} = \begin{bmatrix} -\boldsymbol{x}_1^\top - \\ \vdots \\ -\boldsymbol{x}_N^\top - \end{bmatrix} \in \mathbb{R}^{N \times D}$$

$\boldsymbol{x}_n \in \mathbb{R}^D$ – nth token embedding

Text

The ducks were swimming near the bank.

Token Embeddings

The → D-dimensional vector

N = 10 embeddings

$$\mathbf{X} = \begin{bmatrix} -\boldsymbol{x}_1^\top - \\ \vdots \\ -\boldsymbol{x}_N^\top - \end{bmatrix} \in \mathbb{R}^{N \times D}$$

$\boldsymbol{x}_n \in \mathbb{R}^D$ – nth token embedding

What do we lose in this representation?

→ What are we looking at here?

bank → What type of bank are we talking about?

Context is important – in both image and text processing.

How do we capture context?

$\boldsymbol{x_n} \in \mathbb{R}^D$ – nth token embedding

$\boldsymbol{v_n} = (\mathbf{W}^{(v)})^\top \boldsymbol{x_n} \in \mathbb{R}^{D_v}$ – context provided by nth token

$\mathbf{W}^{(v)} \in \mathbb{R}^{D \times D_v}$ is a learned weight matrix

$$\boldsymbol{y_n} = \boldsymbol{x_n} + \sum_{i=1}^{N} \alpha_{ni} \boldsymbol{v}_i$$ – nth token embedding with context

$\alpha_{ni} \in \mathbb{R}$ are learned attention weight

α_{ni} indicates the relevance of token $i \in \{1, \ldots, N\}$ to token $n \in \{1, \ldots, N\}$

$\boldsymbol{y_n}$ is a token plus a *weighted average* of the context of other tokens

How do we capture context?

nth token embedding with context:

$$\boldsymbol{y_n} = \boldsymbol{x_n} + \sum_{i=1}^{N} \alpha_{ni} \boldsymbol{v}_i = \boldsymbol{x_n} + \sum_{i=1}^{N} \alpha_{ni} (\mathbf{W}^{(v)})^\top \boldsymbol{x_n}$$

nth token embedding

learned attention weight (relevance of token i to token n)

context provided by ith token

learned context matrix

How do we learn appropriate attention weights?

How to Model Attention

1. Transformer Architecture
2. Modeling Attention

Capturing Similarity via Inner Product

Token Keys

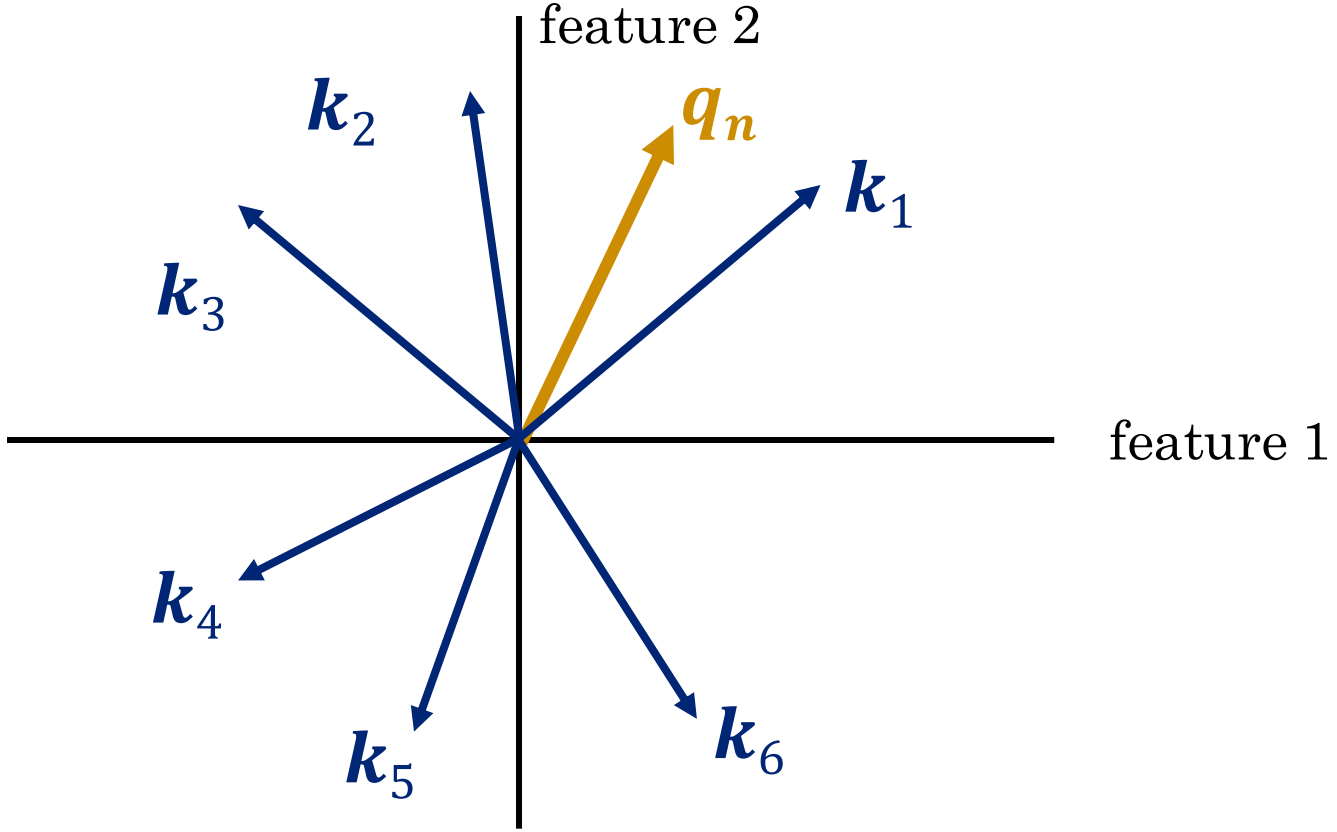
For a weight matrix $\mathbf{W}^{(k)} \in \mathbb{R}^{D \times D_k}$, we can define the key vector of the nth token embedding as

$$\boldsymbol{k}_n = (\mathbf{W}^{(k)})^\top \boldsymbol{x}_n \in \mathbb{R}^{D_k}$$

Token Queries

For a weight matrix $\mathbf{W}^{(q)} \in \mathbb{R}^{D \times D_k}$, we can define the query vector of the nth token embedding as

$$\boldsymbol{q}_n = (\mathbf{W}^{(q)})^\top \boldsymbol{x}_n \in \mathbb{R}^{D_k}$$



Similarity between $\boldsymbol{q}_n$ and $\boldsymbol{k}_i$: $z_{n,i} = \boldsymbol{q}_n^\top \boldsymbol{k}_i$*

$$z_{n,1} > z_{n,2} > z_{n,3} > z_{n,6} > z_{n,4} > z_{n,5}$$

*Assume vectors have unit norm.

Capturing Similarity via Inner Product

Token Keys

$$\mathbf{K} = \begin{bmatrix} -\boldsymbol{k}_1^\top - \\ \vdots \\ -\boldsymbol{k}_N^\top - \end{bmatrix} = \mathbf{X}\,\mathbf{W}^{(k)} \in \mathbb{R}^{N \times D_\mathrm{k}}$$

$$\boldsymbol{k}_n = (\mathbf{W}^{(k)})^\top \boldsymbol{x}_n \in \mathbb{R}^{D_\mathrm{k}}$$

Token Queries

$$\mathbf{Q} = \begin{bmatrix} -\boldsymbol{q}_1^\top - \\ \vdots \\ -\boldsymbol{q}_N^\top - \end{bmatrix} = \mathbf{X}\,\mathbf{W}^{(q)} \in \mathbb{R}^{N \times D_\mathrm{k}}$$

$$\boldsymbol{q}_n = (\mathbf{W}^{(q)})^\top \boldsymbol{x}_n \in \mathbb{R}^{D_\mathrm{k}}$$

Token Similarities

$$\mathbf{Z} = \mathbf{Q}\,\mathbf{K}^\top \in \mathbb{R}^{N \times N}$$

Similarity between $\boldsymbol{q}_n$ and $\boldsymbol{k}_i$:

$$z_{ni} = (\mathbf{Q}\,\mathbf{K}^\top)_{ni} = \boldsymbol{q}_n^\top \boldsymbol{k}_i$$

We'll review these calculations in Problem 1 on the Disc. 10 worksheet.

Transforming Similarity into Probability

nth token embedding with context:

$$\boldsymbol{y_n} = \boldsymbol{x_n} + \sum_{\mathrm{i=1}}^{\mathrm{N}} \alpha_{ni} \boldsymbol{v}_i = \boldsymbol{x_n} + \sum_{\mathrm{i=1}}^{\mathrm{N}} \alpha_{ni} (\mathbf{W}^{(v)})^\top \boldsymbol{x_n}$$

We want the attention weights to satisfy $\alpha_{ni} \geq 0$ and $\sum_{i=1}^{N} \alpha_{ni} = 1$.

This is not guaranteed using the inner product for similarity, but we can define

$$\boldsymbol{\alpha} = \mathrm{SoftMax}_{\mathrm{row}}[\mathbf{Z}] \in \mathbb{R}^{N \times N}$$

→ attention matrix

$$\alpha_{ni} = \frac{\exp(z_{ni})}{\sum_{i=1}^{N} \exp(z_{ni})}$$

→ relevance of token i to token n

Ensuring Unit Variance Distributions

To ensure the variance of these attention scores stay close to one, we will actually scale the inner product terms by the square root of the key/query dimension, D_k, before taking the softmax:

$$\boldsymbol{\alpha} = \text{SoftMax}_{\text{row}}\left[\frac{\mathbf{Z}}{\sqrt{D_k}}\right]$$

$$\boldsymbol{\alpha}_{ni} = \exp\left(\frac{z_{ni}}{\sqrt{D_k}}\right)\left(\sum_{j=1}^{N}\exp\left(\frac{z_{nj}}{\sqrt{D_k}}\right)\right)^{-1}$$

We'll discuss this scaling in Problem 2 on the Disc. 10 worksheet.

The Self Attention Layer

$$\text{Attention}[\mathbf{K}, \mathbf{Q}, \mathbf{V}] = \boldsymbol{\alpha}\,\mathbf{V} = \sum_{i=1}^{N} \alpha_{ni} \boldsymbol{v_i}$$

$\mathbf{V} = \mathbf{X}\,\mathbf{W}^{(v)} \in \mathbb{R}^{N \times D_v}$ – value (context) matrix

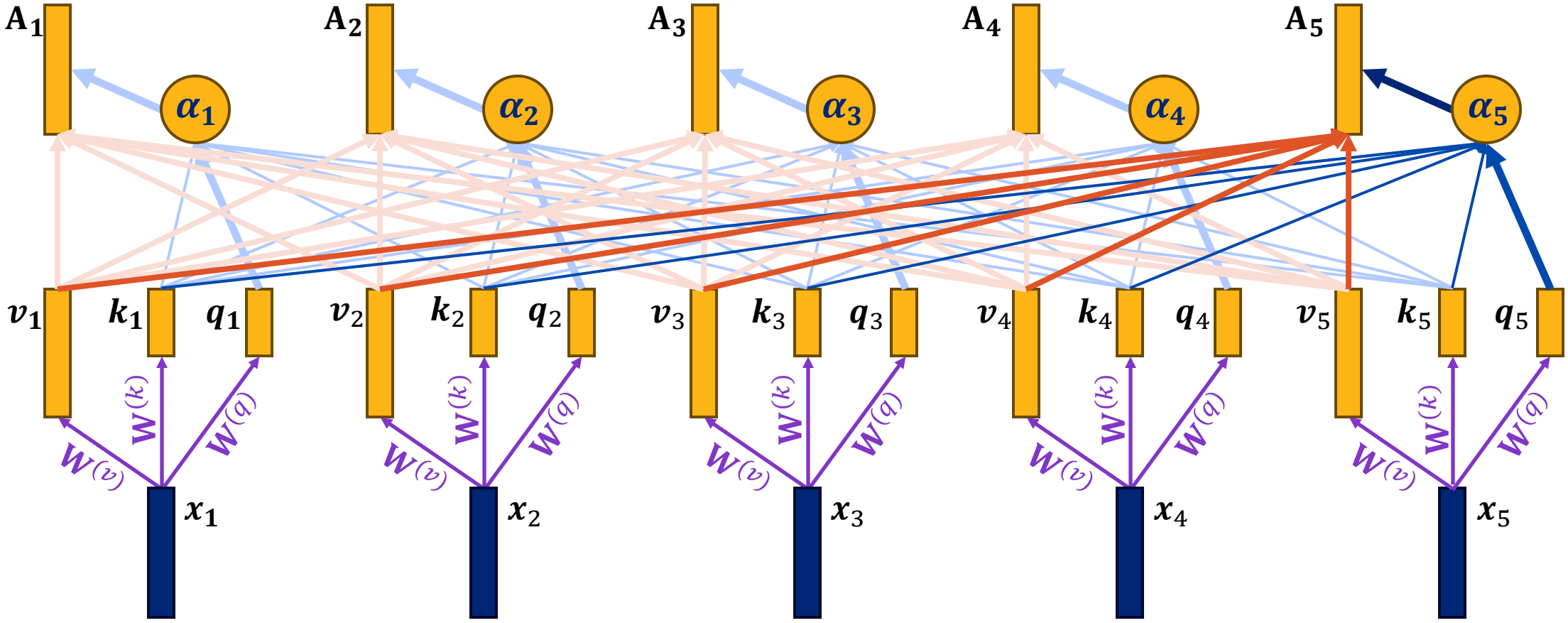
$\boldsymbol{\alpha} = \left(\text{SoftMax}_{\text{row}} \left[\frac{\mathbf{Q}\,\mathbf{K}^{\top}}{\sqrt{D_k}} \right] \right) \in \mathbb{R}^{N \times N}$ – attention matrix

$\mathbf{K} = \mathbf{X}\,\mathbf{W}^{(k)} \in \mathbb{R}^{N \times D_{\text{k}}}$ – key matrix

$\mathbf{Q} = \mathbf{X}\,\mathbf{W}^{(q)} \in \mathbb{R}^{N \times D_{\text{k}}}$ – query matrix

The Self Attention Layer

$$\text{Attention}[\mathbf{K}, \mathbf{Q}, \mathbf{V}] = \boldsymbol{\alpha}\,\mathbf{V} = \sum_{i=1}^{N} \alpha_{ni} \boldsymbol{v_i}$$



Discussion Mini Lecture 10

Transformers

Contributors: Sara Pohland

Additional Resources

1. Transformers
   - Deep Learning Foundations and Concepts – Chapter 12.1
   - Umar Jamil – Attention is all you need (Transformer) - Model explanation (including math), Inference and Training