Attention

CISC 7026 - Introduction to Deep Learning

Steven Morad

University of Macau

Admin

Admin

Last exam next week

Admin

Last exam next week

Preliminary questions (still making exam, may change)

Admin

Last exam next week

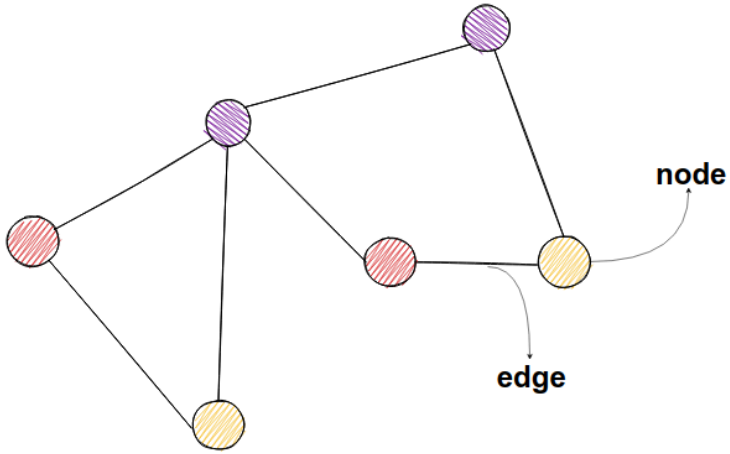
Preliminary questions (still making exam, may change)

- 1 Question perceptron autoencoder and objective
- 1 Question rewriting functions as recurrent function
- 1 Question evaluating scans
- 1 Question log likelihood derivation
- 1 Question key/query attention

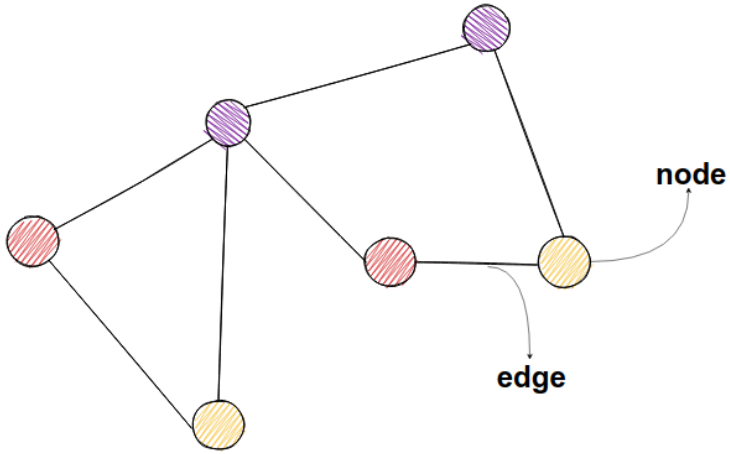
Review

Short Intro to Graph Neural Networks

Short Intro to Graph Neural Networks

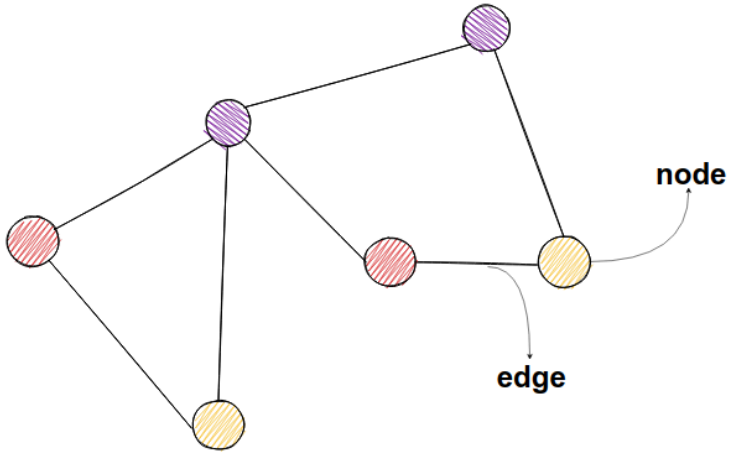


Short Intro to Graph Neural Networks



A **node** is a vector of information

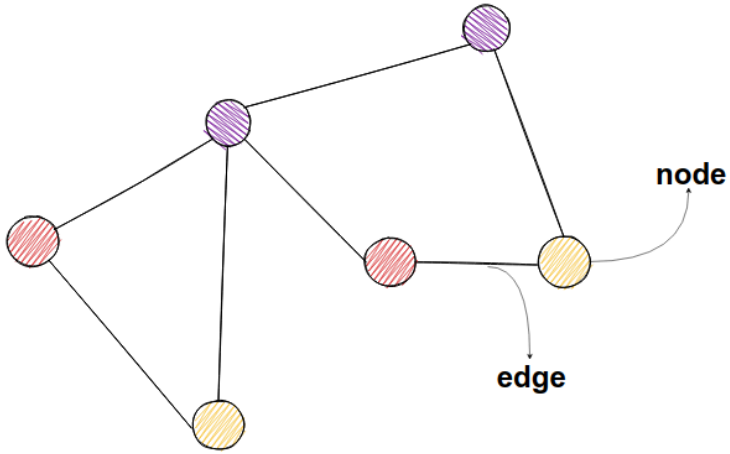
Short Intro to Graph Neural Networks



A **node** is a vector of information

An **edge** connects two nodes

Short Intro to Graph Neural Networks

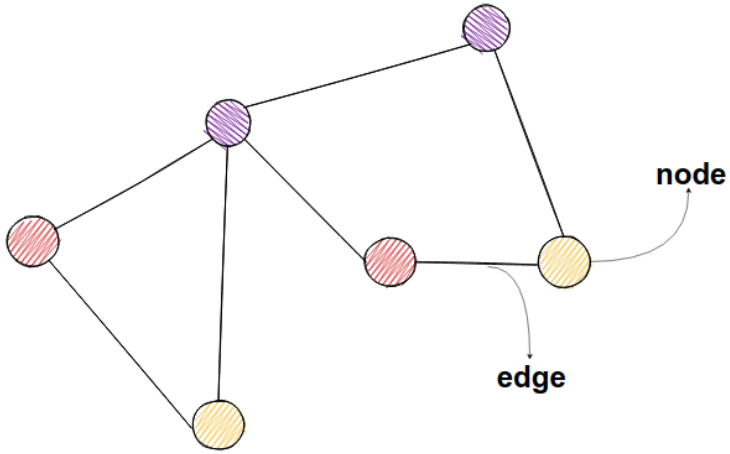


A **node** is a vector of information

An **edge** connects two nodes

If we connect nodes i and j with edge (i, j), then i and j are **neighbors**

Short Intro to Graph Neural Networks



A **node** is a vector of information

An **edge** connects two nodes

If we connect nodes i and j with edge (i, j), then i and j are **neighbors**

The **neighborhood** $\boldsymbol{N}(i)$ contains all neighbors of node i

Short Intro to Graph Neural Networks

Let us think of graphs as **signals**

Short Intro to Graph Neural Networks

Let us think of graphs as **signals**

Question: Where did we see signals before?

Short Intro to Graph Neural Networks

Let us think of graphs as **signals**

Question: Where did we see signals before?

Answer: Convolution

Short Intro to Graph Neural Networks

Let us think of graphs as **signals**

Question: Where did we see signals before?

Answer: Convolution

Rather than time t or space u, v, graphs are a function of neighborhood

Short Intro to Graph Neural Networks

Let us think of graphs as **signals**

Question: Where did we see signals before?

Answer: Convolution

Rather than time t or space u, v, graphs are a function of neighborhood

$$\text{Node } i \quad \boldsymbol{x}(i) \in \mathbb{R}^{d_x}; \quad i \in 1, ..., T$$

Short Intro to Graph Neural Networks

Let us think of graphs as **signals**

Question: Where did we see signals before?

Answer: Convolution

Rather than time t or space u, v, graphs are a function of neighborhood

$$\text{Node } i \quad \boldsymbol{x}(i) \in \mathbb{R}^{d_x}; \quad i \in 1, ..., T$$

$$\text{Neighborhood of } i \quad \boldsymbol{N}(i) = \begin{bmatrix} i \\ j \\ k \\ \vdots \end{bmatrix}; \quad \boldsymbol{N}(i) \in \mathcal{P}(i); \quad i \in 1, ..., T$$

Short Intro to Graph Neural Networks

Consider a **graph convolution layer**

Short Intro to Graph Neural Networks

Consider a **graph convolution layer**

For a node i, the graph convolution layer is:

Short Intro to Graph Neural Networks

Consider a **graph convolution layer**

For a node i, the graph convolution layer is:

$$f(\boldsymbol{x}, \boldsymbol{N}, \boldsymbol{\theta})(i) = \sigma\left(\sum_{j \in \boldsymbol{N}(i)} \boldsymbol{\theta}^{\top} \overline{\boldsymbol{x}}(j)\right)$$

Short Intro to Graph Neural Networks

Consider a **graph convolution layer**

For a node i, the graph convolution layer is:

$$f(\boldsymbol{x}, \boldsymbol{N}, \boldsymbol{\theta})(i) = \sigma\left(\sum_{j \in \boldsymbol{N}(i)} \boldsymbol{\theta}^{\top} \overline{\boldsymbol{x}}(j)\right)$$

Combine information from the neighbors of $\boldsymbol{x}(i)$

Short Intro to Graph Neural Networks

Consider a **graph convolution layer**

For a node i, the graph convolution layer is:

$$f(\boldsymbol{x}, \boldsymbol{N}, \boldsymbol{\theta})(i) = \sigma\left(\sum_{j \in \boldsymbol{N}(i)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right)$$

Combine information from the neighbors of $\boldsymbol{x}(i)$

This is just one node, we use this graph layer for all nodes in the graph

Short Intro to Graph Neural Networks

Apply graph convolution over all nodes in the graph

Short Intro to Graph Neural Networks

Apply graph convolution over all nodes in the graph

$$f(\boldsymbol{x}, \boldsymbol{N}, \boldsymbol{\theta}) = \begin{bmatrix} f(\boldsymbol{x}, \boldsymbol{N}, \boldsymbol{\theta})(1) \\ f(\boldsymbol{x}, \boldsymbol{N}, \boldsymbol{\theta})(2) \\ \vdots \\ f(\boldsymbol{x}, \boldsymbol{N}, \boldsymbol{\theta})(T) \end{bmatrix} = \begin{bmatrix} \sigma\left(\sum_{j \in \boldsymbol{N}(1)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j \in \boldsymbol{N}(2)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j \in \boldsymbol{N}(T)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Short Intro to Graph Neural Networks

Apply graph convolution over all nodes in the graph

$$f(\boldsymbol{x}, \boldsymbol{N}, \boldsymbol{\theta}) = \begin{bmatrix} f(\boldsymbol{x}, \boldsymbol{N}, \boldsymbol{\theta})(1) \\ f(\boldsymbol{x}, \boldsymbol{N}, \boldsymbol{\theta})(2) \\ \vdots \\ f(\boldsymbol{x}, \boldsymbol{N}, \boldsymbol{\theta})(T) \end{bmatrix} = \begin{bmatrix} \sigma\left(\sum_{j \in \boldsymbol{N}(1)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j \in \boldsymbol{N}(2)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j \in \boldsymbol{N}(T)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

How does this compare to regular convolution (images, sound, etc)?

Short Intro to Graph Neural Networks

Standard 1D convolution

$$\begin{bmatrix} \sigma\left(\sum_{j=1}^{k} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j=2}^{k+1} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j=T-k}^{T} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Short Intro to Graph Neural Networks

Standard 1D convolution

$$\begin{bmatrix} \sigma\left(\sum_{j=1}^{k} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j=2}^{k+1} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j=T-k}^{T} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Graph convolution

$$\begin{bmatrix} \sigma\left(\sum_{j \in \boldsymbol{N}(1)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j \in \boldsymbol{N}(2)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j \in \boldsymbol{N}(T)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Short Intro to Graph Neural Networks

Standard 1D convolution

$$\begin{bmatrix} \sigma\left(\sum_{j=1}^{k} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j=2}^{k+1} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j=T-k}^{T} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Graph convolution

$$\begin{bmatrix} \sigma\left(\sum_{j \in \boldsymbol{N}(1)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j \in \boldsymbol{N}(2)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j \in \boldsymbol{N}(T)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Question: What is the output size of standard convolution?

Short Intro to Graph Neural Networks

Standard 1D convolution

$$\begin{bmatrix} \sigma\left(\sum_{j=1}^{k} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j=2}^{k+1} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j=T-k}^{T} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Graph convolution

$$\begin{bmatrix} \sigma\left(\sum_{j \in \boldsymbol{N}(1)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j \in \boldsymbol{N}(2)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j \in \boldsymbol{N}(T)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Question: What is the output size of standard convolution?

Answer: $(T - k - 1) \times d_h$

Short Intro to Graph Neural Networks

Standard 1D convolution

$$\begin{bmatrix} \sigma\left(\sum_{j=1}^{k} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j=2}^{k+1} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j=T-k}^{T} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Graph convolution

$$\begin{bmatrix} \sigma\left(\sum_{j \in \boldsymbol{N}(1)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j \in \boldsymbol{N}(2)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j \in \boldsymbol{N}(T)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Short Intro to Graph Neural Networks

Standard 1D convolution

$$\begin{bmatrix} \sigma\left(\sum_{j=1}^{k} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j=2}^{k+1} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j=T-k}^{T} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Graph convolution

$$\begin{bmatrix} \sigma\left(\sum_{j \in \boldsymbol{N}(1)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j \in \boldsymbol{N}(2)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j \in \boldsymbol{N}(T)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Question: What is the output size of graph convolution?

Short Intro to Graph Neural Networks

Standard 1D convolution

$$\begin{bmatrix} \sigma\left(\sum_{j=1}^{k} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j=2}^{k+1} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j=T-k}^{T} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Graph convolution

$$\begin{bmatrix} \sigma\left(\sum_{j \in \boldsymbol{N}(1)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j \in \boldsymbol{N}(2)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j \in \boldsymbol{N}(T)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}$$

Question: What is the output size of graph convolution?

Answer: $T \times d_h$

Short Intro to Graph Neural Networks

We can use pooling with graph convolutions too

$$\text{SumPool}\left(\begin{bmatrix} \sigma\left(\sum_{j\in\boldsymbol{N}(1)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \sigma\left(\sum_{j\in\boldsymbol{N}(2)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \\ \vdots \\ \sigma\left(\sum_{j\in\boldsymbol{N}(T)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) \end{bmatrix}\right) =$$

$$\sigma\left(\sum_{j\in\boldsymbol{N}(1)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) + \sigma\left(\sum_{j\in\boldsymbol{N}(2)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right) + \ldots + \sigma\left(\sum_{j\in\boldsymbol{N}(T)} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}(j)\right)$$

Short Intro to Graph Neural Networks

We can write attention and transformers as a graph neural network

Short Intro to Graph Neural Networks

We can write attention and transformers as a graph neural network

- Make the neighborhood the entire graph $\boldsymbol{N}(i) = \boldsymbol{X}$

Short Intro to Graph Neural Networks

We can write attention and transformers as a graph neural network

- Make the neighborhood the entire graph $\boldsymbol{N}(i) = \boldsymbol{X}$
- Learnable continuous edge weights (key $\times$ query)

Short Intro to Graph Neural Networks

We can write attention and transformers as a graph neural network

- Make the neighborhood the entire graph $\boldsymbol{N}(i) = \boldsymbol{X}$
- Learnable continuous edge weights (key $\times$ query)

Edge weights

$$\boldsymbol{A}(i) = \text{softmax}\left(\frac{\left(\boldsymbol{\theta}_Q^\top \boldsymbol{x}_i\right)\left(\boldsymbol{\theta}_K^\top \boldsymbol{X}\right)}{\sqrt{d_z}}\right)$$

Short Intro to Graph Neural Networks

We can write attention and transformers as a graph neural network

- Make the neighborhood the entire graph $\boldsymbol{N}(i) = \boldsymbol{X}$
- Learnable continuous edge weights (key $\times$ query)

Edge weights

$$\boldsymbol{A}(i) = \text{softmax}\left(\frac{\left(\boldsymbol{\theta}_Q^\top \boldsymbol{x}_i\right)\left(\boldsymbol{\theta}_K^\top \boldsymbol{X}\right)}{\sqrt{d_z}}\right)$$

Graph convolution

$$f(\boldsymbol{X}, i) = \sum_{j \in \boldsymbol{N}(i)} \boldsymbol{\theta}_V^\top \boldsymbol{x}_j \cdot \boldsymbol{A}(i)_j$$

Short Intro to Graph Neural Networks

We can write attention and transformers as a graph neural network

- Make the neighborhood the entire graph $\boldsymbol{N}(i) = \boldsymbol{X}$
- Learnable continuous edge weights (key $\times$ query)

Edge weights

$$\boldsymbol{A}(i) = \text{softmax}\left(\frac{\left(\boldsymbol{\theta}_Q^\top \boldsymbol{x}_i\right)\left(\boldsymbol{\theta}_K^\top \boldsymbol{X}\right)}{\sqrt{d_z}}\right)$$

Graph convolution

$$f(\boldsymbol{X}, i) = \sum_{j \in \boldsymbol{N}(i)} \boldsymbol{\theta}_V^\top \boldsymbol{x}_j \cdot \boldsymbol{A}(i)_j$$

Many ways to teach attention, but I like the RNN-style approach more

Attention

Attention

Attention and transformers are the “hottest” topic in deep learning

Attention

Attention and transformers are the “hottest” topic in deep learning

People use them for almost every task (even if they shouldn’t!)

Attention

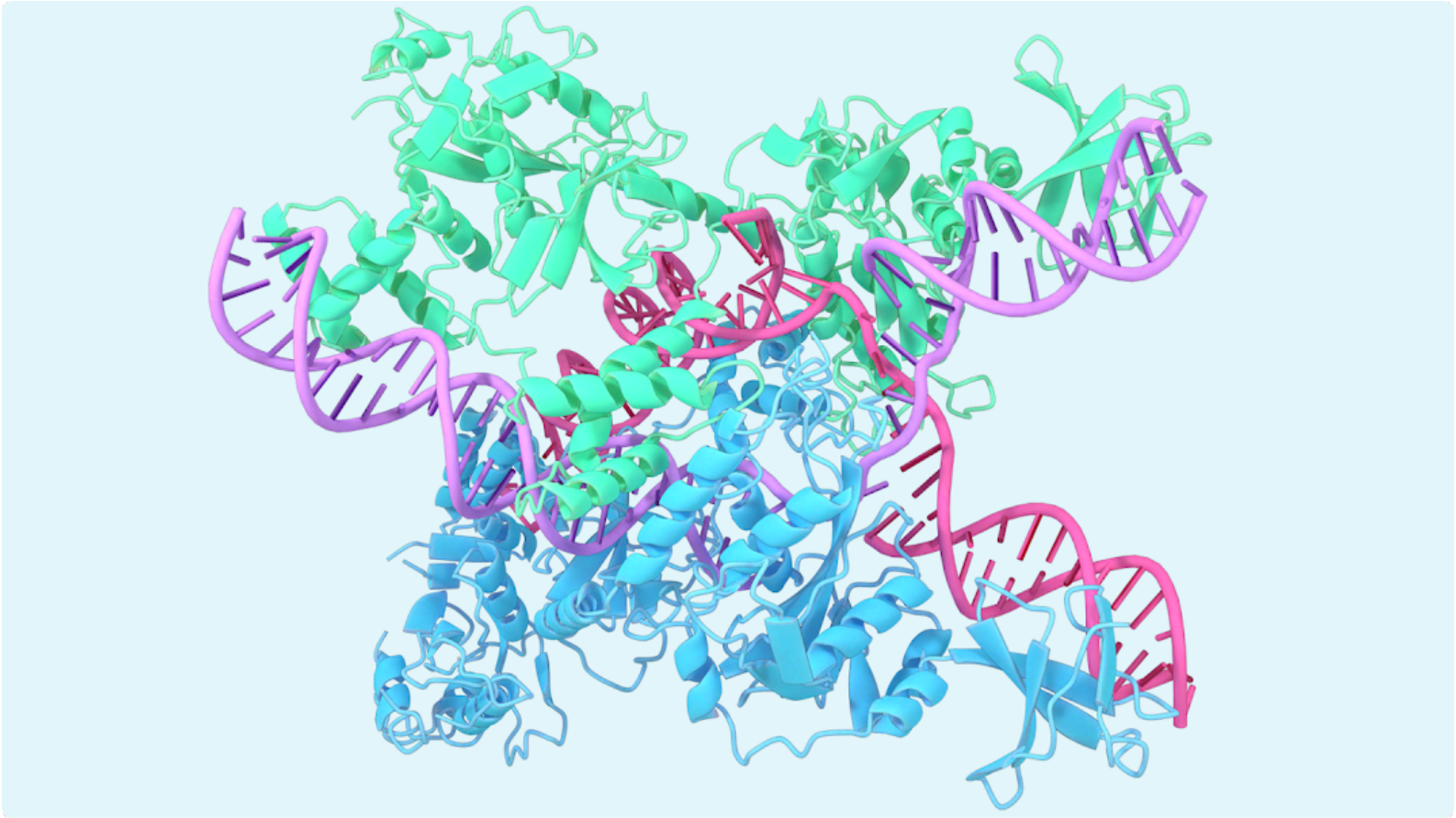
Attention and transformers are the “hottest” topic in deep learning

People use them for almost every task (even if they shouldn’t!)

Let’s review some projects based on attention

Attention

AlphaFold (Nobel prize)

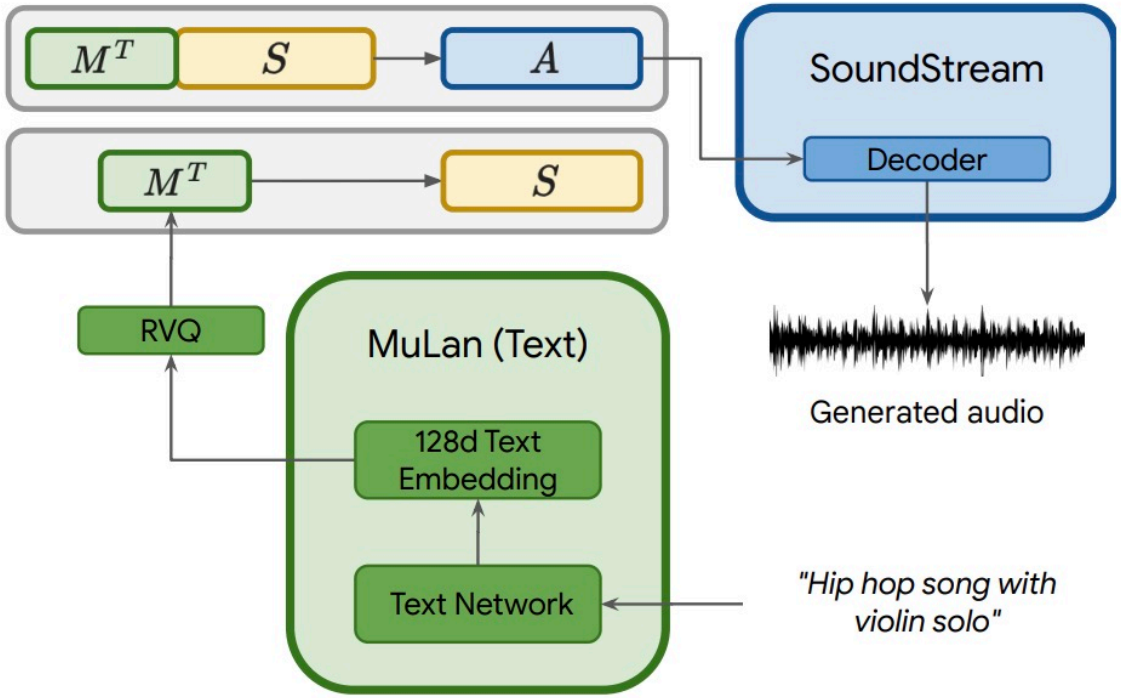


Attention

ChatGPT, Qwen, LLaMA, Mistral, Doubou, Ernie chatbots

Attention

MusicTransformer, MuLan

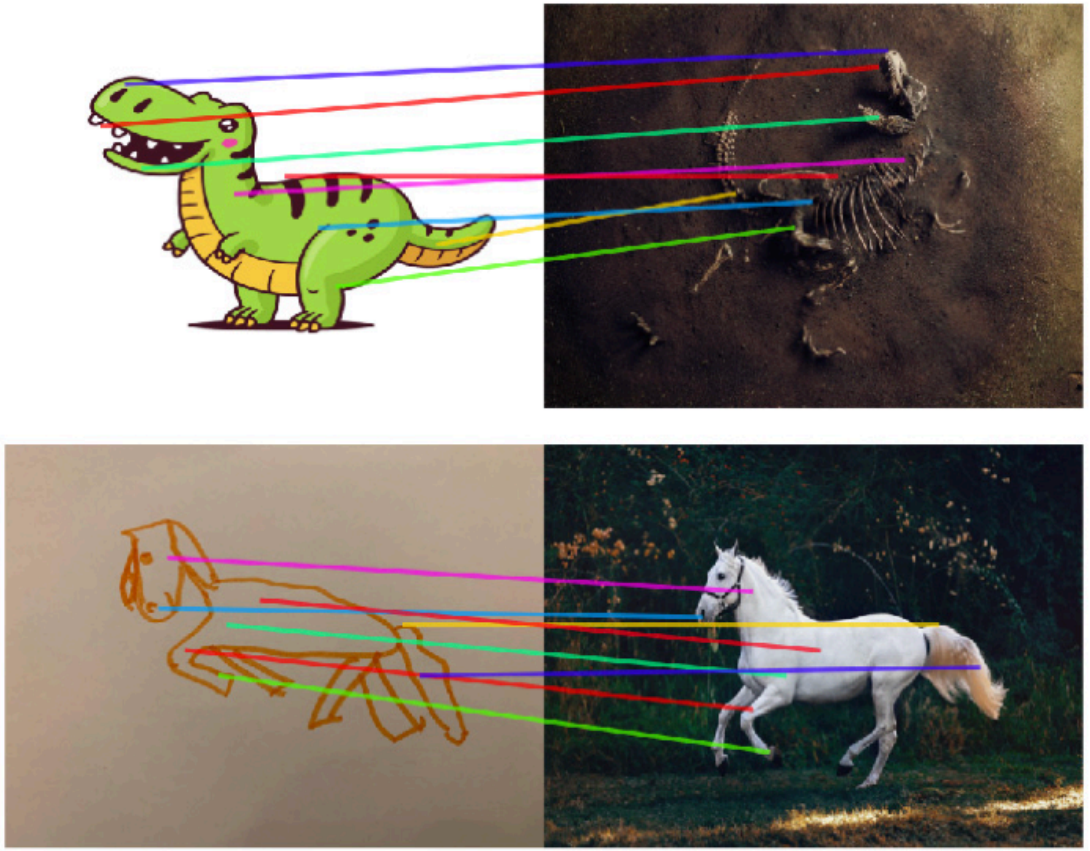


Attention

Google Translate, Baidu Translate, Apple Translate

Attention

ViT, DinoV2



Attention

All these models are **transformers**

Attention

All these models are **transformers**

At the core of each transformer is **attention**

Attention

All these models are **transformers**

At the core of each transformer is **attention**

We will write attention slightly differently than you see in textbooks

Attention

All these models are **transformers**

At the core of each transformer is **attention**

We will write attention slightly differently than you see in textbooks

- I want to stay with column-vector notation

Attention

All these models are **transformers**

At the core of each transformer is **attention**

We will write attention slightly differently than you see in textbooks

- I want to stay with column-vector notation

Textbook

$$\text{softmax}(\boldsymbol{Q}\boldsymbol{K}^\top)\boldsymbol{V}$$

Attention

All these models are **transformers**

At the core of each transformer is **attention**

We will write attention slightly differently than you see in textbooks

- I want to stay with column-vector notation

Textbook	Our representation
$\text{softmax}(\boldsymbol{Q}\boldsymbol{K}^\top)\boldsymbol{V}$	$\boldsymbol{V}\,\text{softmax}(\boldsymbol{K}^\top\boldsymbol{Q})$

Attention

All these models are **transformers**

At the core of each transformer is **attention**

We will write attention slightly differently than you see in textbooks

- I want to stay with column-vector notation

Textbook	Our representation
$\text{softmax}(\boldsymbol{Q}\boldsymbol{K}^\top)\boldsymbol{V}$	$\boldsymbol{V}\,\text{softmax}(\boldsymbol{K}^\top\boldsymbol{Q})$

Same results, but easier to write scalar/vector

Attention

We can derive attention from composite memory

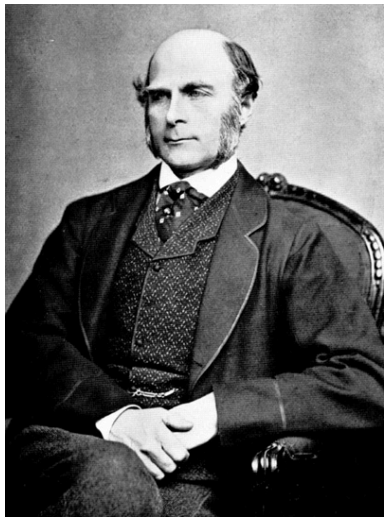
Attention

Attention

We can derive attention from composite memory

Francis Galton (1822-1911)
photo composite memory

Attention



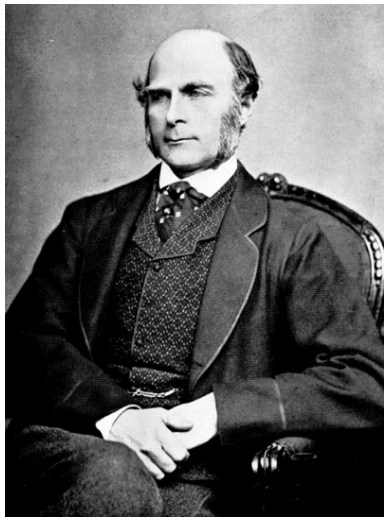
Attention

We can derive attention from composite memory

Francis Galton (1822-1911) photo composite memory

Composite photo of members of a party

Attention



Attention

Task: Find a mathematical model of how our mind represents memories

Attention

Task: Find a mathematical model of how our mind represents memories

$$X : \mathbb{R}^{h \times w} \quad \text{People you see at the party}$$

Attention

Task: Find a mathematical model of how our mind represents memories

$$X : \mathbb{R}^{h \times w} \quad \text{People you see at the party}$$

$$H : \mathbb{R}^{h \times w} \quad \text{The image in your mind}$$

Attention

Task: Find a mathematical model of how our mind represents memories

$$X : \mathbb{R}^{h \times w} \quad \text{People you see at the party}$$

$$H : \mathbb{R}^{h \times w} \quad \text{The image in your mind}$$

$$f : X^T \times \Theta \mapsto H$$

Attention

Task: Find a mathematical model of how our mind represents memories

$$X : \mathbb{R}^{h \times w} \quad \text{People you see at the party}$$

$$H : \mathbb{R}^{h \times w} \quad \text{The image in your mind}$$

$$f : X^T \times \Theta \mapsto H$$

Composite photography/memory uses a weighted sum

Attention

Task: Find a mathematical model of how our mind represents memories

$$X : \mathbb{R}^{h \times w} \quad \text{People you see at the party}$$

$$H : \mathbb{R}^{h \times w} \quad \text{The image in your mind}$$

$$f : X^T \times \Theta \mapsto H$$

Composite photography/memory uses a weighted sum

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{i=1}^{T} \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_i$$

Attention

Limited space, cannot remember everything

Attention

Limited space, cannot remember everything

Introduced forgetting term $\gamma \in [0, 1]$

Attention

Limited space, cannot remember everything

Introduced forgetting term $\gamma \in [0, 1]$

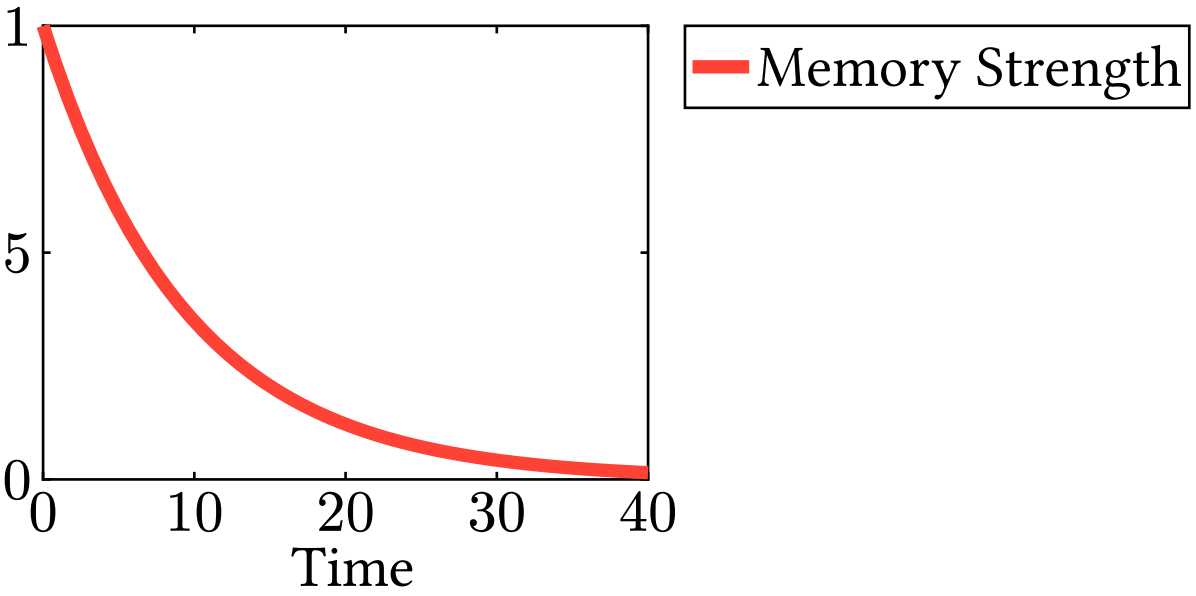
$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{i=1}^{T} \gamma^{T-i} \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_i$$

Attention

Limited space, cannot remember everything

Introduced forgetting term $\gamma \in [0, 1]$

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{i=1}^{T} \gamma^{T-i} \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_i$$

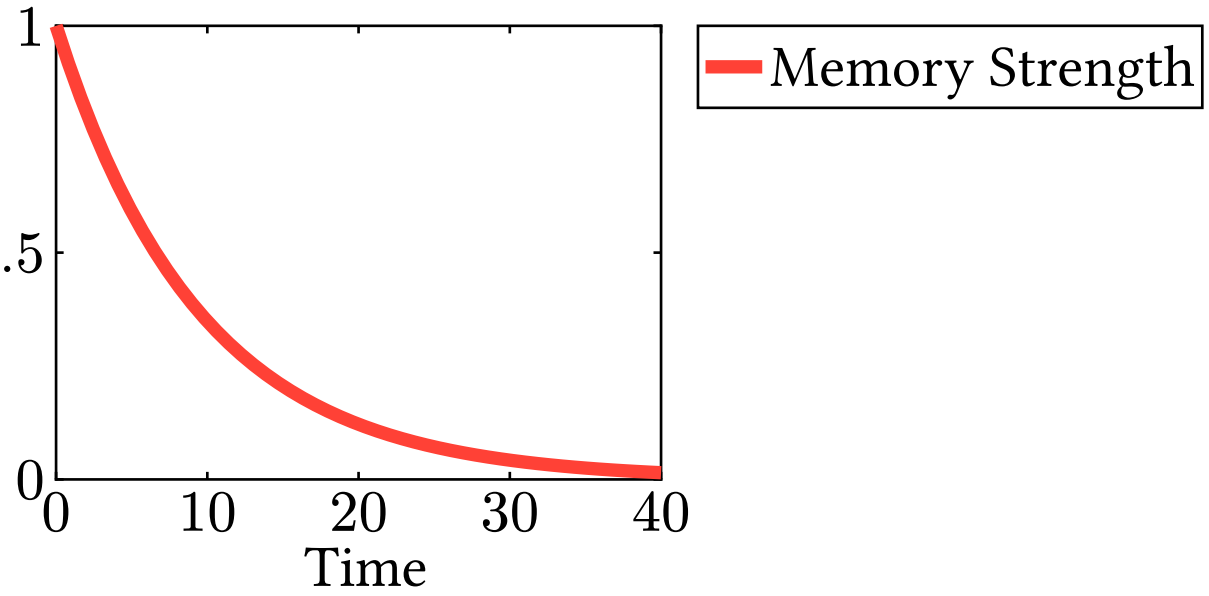


Attention

Limited space, cannot remember everything

Introduced forgetting term $\gamma \in [0, 1]$

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{i=1}^{T} \gamma^{T-i} \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_i$$



Question: Does this accurately model what **you** remember?

Attention

Example: We attend a party in 1850s

Attention

Example: We attend a party in 1850s

We talk with many people at this party

Attention

Example: We attend a party in 1850s

We talk with many people at this party

Attention

Example: We attend a party in 1850s

We talk with many people at this party

10 PM

Attention

Example: We attend a party in 1850s

We talk with many people at this party

10 PM 11 PM

Attention

Example: We attend a party in 1850s

We talk with many people at this party

10 PM 11 PM 12 AM

Attention

Example: We attend a party in 1850s

We talk with many people at this party

10 PM | 11 PM | 12 AM | 1 AM

Attention

According to forgetting, the memories should fade with time

Attention

According to forgetting, the memories should fade with time

Attention

According to forgetting, the memories should fade with time

$$\gamma^3 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1$$

Attention

According to forgetting, the memories should fade with time

$\gamma^3 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1$ $\gamma^2 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2$

Attention

According to forgetting, the memories should fade with time

$\gamma^3 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1$ $\gamma^2 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2$ $\gamma^1 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3$

Attention

According to forgetting, the memories should fade with time

$\gamma^3 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1$ $\gamma^2 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2$ $\gamma^1 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3$ $\gamma^0 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4$

Attention

According to forgetting, the memories should fade with time

$\gamma^3 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1$ $\gamma^2 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2$ $\gamma^1 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3$ $\gamma^0 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4$

Attention

Any questions before moving on?

Attention

Consider another party, with one more guest

Attention

Consider another party, with one more guest

Attention

Consider another party, with one more guest

$$\gamma^4 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad \gamma^3 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad \gamma^2 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad \gamma^1 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad \gamma^0 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Attention

Consider another party, with one more guest

$\gamma^4 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1$ $\gamma^3 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2$ $\gamma^2 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3$ $\gamma^1 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4$ $\gamma^0 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$

Question: What will happen to Taylor Swift?

Attention

$\gamma^4 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1$ $\gamma^3 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2$ $\gamma^2 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3$ $\gamma^1 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4$ $\gamma^0 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$

Attention

$\gamma^4 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1$ $\gamma^3 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2$ $\gamma^2 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3$ $\gamma^1 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4$ $\gamma^0 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$

We will forget meeting her!

Attention

$$\gamma^4 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad \gamma^3 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad \gamma^2 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad \gamma^1 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad \gamma^0 \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

We will forget meeting her!

Question: Would you forget meeting Taylor Swift?

Attention

Our model of memory is incomplete

Attention

Our model of memory is incomplete

Memories are not created equal, some are more important than others

Attention

Our model of memory is incomplete

Memories are not created equal, some are more important than others

Important memories persist longer than unimportant memories

Attention

Our model of memory is incomplete

Memories are not created equal, some are more important than others

Important memories persist longer than unimportant memories

We will **pay more attention** to certain memories

Attention

What does human memory actually look like?

Attention

What does human memory actually look like?

$$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1$$

Attention

What does human memory actually look like?

$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 0.1 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2$

Attention

What does human memory actually look like?

$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 0.1 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 0.1 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3$

Attention

What does human memory actually look like?

$$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 0.1 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 0.1 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad 0.5 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4$$

Attention

What does human memory actually look like?

$$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 0.1 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 0.1 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad 0.5 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad 0.1 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Attention

What does human memory actually look like?

$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1$ $0.1 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2$ $0.1 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3$ $0.5 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4$ $0.1 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$

Question: How can we achieve this forgetting?

Attention

In our composite model, forgetting is a function of time

Attention

In our composite model, forgetting is a function of time

Question: Any forgetting mechanism that is not a function of time?

Attention

In our composite model, forgetting is a function of time

Question: Any forgetting mechanism that is not a function of time?

Answer: Forgetting in recurrent neural network is function of input!

Attention

In our composite model, forgetting is a function of time

Question: Any forgetting mechanism that is not a function of time?

Answer: Forgetting in recurrent neural network is function of input!

$$f_{\text{forget}}(\boldsymbol{x}, \boldsymbol{\theta}) = \sigma\left(\boldsymbol{\theta}_{\lambda}^{\top} \overline{\boldsymbol{x}}\right)$$

Attention

In our composite model, forgetting is a function of time

Question: Any forgetting mechanism that is not a function of time?

Answer: Forgetting in recurrent neural network is function of input!

$$f_{\text{forget}}(\boldsymbol{x}, \boldsymbol{\theta}) = \sigma\left(\boldsymbol{\theta}_{\lambda}^{\top} \overline{\boldsymbol{x}}\right)$$

$$f(\boldsymbol{h}, \boldsymbol{x}, \boldsymbol{\theta}) = f_{\text{forget}}(\boldsymbol{x}, \boldsymbol{\theta}) \odot \boldsymbol{h} + \boldsymbol{\theta}_{x}^{\top} \overline{\boldsymbol{x}}$$

Attention

First, write our forgetting function with slightly different notation

Attention

First, write our forgetting function with slightly different notation

$$\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda) = \sigma\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}\right); \quad \boldsymbol{\theta}_\lambda \in \mathbb{R}^{(d_x+1)\times 1}$$

Attention

First, write our forgetting function with slightly different notation

$$\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda) = \sigma\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}\right); \quad \boldsymbol{\theta}_\lambda \in \mathbb{R}^{(d_x+1)\times 1}$$

Question: Shape of $\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda)$?

Attention

First, write our forgetting function with slightly different notation

$$\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda) = \sigma\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}\right); \quad \boldsymbol{\theta}_\lambda \in \mathbb{R}^{(d_x+1)\times 1}$$

Question: Shape of $\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda)$?

Answer: Scalar!

Attention

First, write our forgetting function with slightly different notation

$$\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda) = \sigma\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}\right); \quad \boldsymbol{\theta}_\lambda \in \mathbb{R}^{(d_x+1)\times 1}$$

Question: Shape of $\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda)$? **Answer:** Scalar!

Then, write our composite memory model with forgetting

Attention

First, write our forgetting function with slightly different notation

$$\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda) = \sigma\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}\right); \quad \boldsymbol{\theta}_\lambda \in \mathbb{R}^{(d_x+1)\times 1}$$

Question: Shape of $\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda)$? **Answer:** Scalar!

Then, write our composite memory model with forgetting

$$f\left(\begin{bmatrix}\boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}\right) = \sum_{i=1}^{T} \boldsymbol{\theta}^\top \boldsymbol{x}_i \cdot \lambda(\boldsymbol{x}_i, \boldsymbol{\theta}_\lambda)$$

Attention

First, write our forgetting function with slightly different notation

$$\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda) = \sigma\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}\right); \quad \boldsymbol{\theta}_\lambda \in \mathbb{R}^{(d_x+1)\times 1}$$

Question: Shape of $\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda)$? **Answer:** Scalar!

Then, write our composite memory model with forgetting

$$f\left(\begin{bmatrix}\boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}\right) = \sum_{i=1}^{T} \boldsymbol{\theta}^\top \boldsymbol{x}_i \cdot \lambda(\boldsymbol{x}_i, \boldsymbol{\theta}_\lambda)$$

Only pay attention to important inputs

Attention

We can use this simple form of attention to remember Taylor Swift

Attention

We can use this simple form of attention to remember Taylor Swift

Attention

$$\lambda(\boldsymbol{x}_1, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad \lambda(\boldsymbol{x}_2, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad \lambda(\boldsymbol{x}_3, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad \lambda(\boldsymbol{x}_4, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad \lambda(\boldsymbol{x}_5, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Attention

$$\lambda(\boldsymbol{x}_1, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad \lambda(\boldsymbol{x}_2, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad \lambda(\boldsymbol{x}_3, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad \lambda(\boldsymbol{x}_4, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad \lambda(\boldsymbol{x}_5, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Question: What do the images look like now?

Attention

$$\lambda(\boldsymbol{x}_1, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad \lambda(\boldsymbol{x}_2, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad \lambda(\boldsymbol{x}_3, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad \lambda(\boldsymbol{x}_4, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad \lambda(\boldsymbol{x}_5, \boldsymbol{\theta}_\lambda) \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Question: What do the images look like now?

Attention

This form of attention will learn to pay attention to everyone!

Attention

This form of attention will learn to pay attention to everyone!

$$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Attention

This form of attention will learn to pay attention to everyone!

$$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Attention

This form of attention will learn to pay attention to everyone!

$$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Attention

$$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Attention

$$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Why does this model of attention pay attention to everyone?

Attention

$$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Why does this model of attention pay attention to everyone?

- Honest answer is I do not know

Attention

$$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Why does this model of attention pay attention to everyone?

- Honest answer is I do not know
  - But I have tried it myself, and this usually happens

Attention

$$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Why does this model of attention pay attention to everyone?

- Honest answer is I do not know
  - But I have tried it myself, and this usually happens
  - Maybe one of you can figure out why!

Attention

$$1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad 1.0 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Why does this model of attention pay attention to everyone?

- Honest answer is I do not know
  - But I have tried it myself, and this usually happens
  - Maybe one of you can figure out why!

We need to prevent all attention weights going to 1

Attention

We should normalize $\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda)$ to model finite (human) attention span

Attention

We should normalize $\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda)$ to model finite (human) attention span

For example, normalize attention to sum to one

$$\sum_{i=1}^{T} \lambda(\boldsymbol{x}_i, \boldsymbol{\theta}_\lambda) = 1$$

Attention

We should normalize $\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda)$ to model finite (human) attention span

For example, normalize attention to sum to one

$$\sum_{i=1}^{T} \lambda(\boldsymbol{x}_i, \boldsymbol{\theta}_\lambda) = 1$$

Now the model must choose who to remember!

Attention

We should normalize $\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda)$ to model finite (human) attention span

For example, normalize attention to sum to one

$$\sum_{i=1}^{T} \lambda(\boldsymbol{x}_i, \boldsymbol{\theta}_\lambda) = 1$$

Now the model must choose who to remember!

Question: How can we ensure that the attention sums to one?

Attention

We should normalize $\lambda(\boldsymbol{x}, \boldsymbol{\theta}_\lambda)$ to model finite (human) attention span

For example, normalize attention to sum to one

$$\sum_{i=1}^{T} \lambda(\boldsymbol{x}_i, \boldsymbol{\theta}_\lambda) = 1$$

Now the model must choose who to remember!

Question: How can we ensure that the attention sums to one?

Answer: Softmax!

Attention

The softmax function maps real numbers to the simplex (probabilities)

Attention

The softmax function maps real numbers to the simplex (probabilities)

$$\text{softmax} : \mathbb{R}^k \mapsto \Delta^{k-1}$$

Attention

The softmax function maps real numbers to the simplex (probabilities)

$$\text{softmax} : \mathbb{R}^k \mapsto \Delta^{k-1}$$

$$\text{softmax}\left(\begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix}\right) = \frac{\exp(\boldsymbol{x})}{\sum_{i=1}^{k} \exp(x_i)} = \begin{bmatrix} \frac{\exp(x_1)}{\exp(x_1)+\exp(x_2)+\ldots \exp(x_k)} \\ \frac{\exp(x_2)}{\exp(x_1)+\exp(x_2)+\ldots \exp(x_k)} \\ \vdots \\ \frac{\exp(x_k)}{\exp(x_1)+\exp(x_2)+\ldots \exp(x_k)} \end{bmatrix}$$

Attention

Let us rewrite attention using softmax

Attention

Let us rewrite attention using softmax

The attention we pay to person i is

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_i = \text{softmax}\left(\begin{bmatrix}\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}_1\\ \vdots \\ \boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}_T\end{bmatrix}\right)_i = \frac{\exp\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}_i\right)}{\sum_{j=1}^T \exp\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}_j\right)}$$

Attention

Attention

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_1 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1$$

Attention

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_1 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_2$$

$$\cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2$$

Attention

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_1 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_2 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_3$$

$$\cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3$$

Attention

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_1 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_2 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_3 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_4$$

$$\cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4$$

Attention

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_1 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_2 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_3 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_4 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_5$$

$$\cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Attention

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_1 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_2 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_3 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_4 \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_5\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_5$$

$$\cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \qquad \cdot\, \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Attention

Attention

$$0.70 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1$$

Attention

$$0.70 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 0.04 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2$$

Attention

$$0.70 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 0.04 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 0.03 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3$$

Attention

$$0.70 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \qquad 0.04 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \qquad 0.03 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \qquad 0.20 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4$$

Attention

$$0.70 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \quad 0.04 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \quad 0.03 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \quad 0.20 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \quad 0.03 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

Attention

$$0.70 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_1 \quad 0.04 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_2 \quad 0.03 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_3 \quad 0.20 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_4 \quad 0.03 \cdot \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}_5$$

$$0.70 + 0.04 + 0.03 + 0.20 + 0.03 = 1.0$$

Attention

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_i = \frac{\exp\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}_i}\right)}{\sum_{j=1}^{T} \exp\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}_j\right)}$$

Attention

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_i = \frac{\exp\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}_i}\right)}{\sum_{j=1}^{T} \exp\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}_j\right)}$$

Compute attention for all inputs at once

Attention

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_i = \frac{\exp\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}_i}\right)}{\sum_{j=1}^{T} \exp\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}_j\right)}$$

Compute attention for all inputs at once

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right) = \begin{bmatrix}\frac{\exp\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}_1\right)}{\sum_{j=1}^{T} \exp\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}_j\right)}\\ \vdots \\ \frac{\exp\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}_T\right)}{\sum_{j=1}^{T} \exp\left(\boldsymbol{\theta}_\lambda^\top \overline{\boldsymbol{x}}_j\right)}\end{bmatrix}$$

Attention

This is a simple form of attention

Attention

This is a simple form of attention

Next, we will investigate the attention used in transformers

Keys and Queries

Keys and Queries

The modern form of attention behaves like a database

Keys and Queries

The modern form of attention behaves like a database

We label each person at the party with a **key**

Keys and Queries

The modern form of attention behaves like a database

We label each person at the party with a **key**

The key describes the content of each $\boldsymbol{x}$

Keys and Queries

The modern form of attention behaves like a database

We label each person at the party with a **key**

The key describes the content of each x

Keys and Queries

The modern form of attention behaves like a database

We label each person at the party with a **key**

The key describes the content of each x

Musician

Keys and Queries

The modern form of attention behaves like a database

We label each person at the party with a **key**

The key describes the content of each x

Musician Lawyer

Keys and Queries

The modern form of attention behaves like a database

We label each person at the party with a **key**

The key describes the content of each x

Musician Lawyer Shopkeeper

Keys and Queries

The modern form of attention behaves like a database

We label each person at the party with a **key**

The key describes the content of each x

Musician Lawyer Shopkeeper Chef

Keys and Queries

The modern form of attention behaves like a database

We label each person at the party with a **key**

The key describes the content of each x

Musician Lawyer Shopkeeper Chef Scientist

Keys and Queries

We can search through our keys using a **query**

Keys and Queries

We can search through our keys using a **query**

Query: Which person will help me on my exam?

Keys and Queries

We can search through our keys using a **query**

Query: Which person will help me on my exam?

Keys and Queries

We can search through our keys using a **query**

Query: Which person will help me on my exam?

Musician Lawyer Shopkeeper Chef Scientist

Keys and Queries

We can search through our keys using a **query**

Query: Which person will help me on my exam?

Musician Lawyer Shopkeeper Chef Scientist

Keys and Queries

Query: I want to have fun

Keys and Queries

Query: I want to have fun

Keys and Queries

Query: I want to have fun

Musician Lawyer Shopkeeper Chef Scientist

Keys and Queries

Query: I want to have fun

Musician Lawyer Shopkeeper Chef Scientist

Keys and Queries

Query: I want to have fun

Musician Lawyer Shopkeeper Chef Scientist

How do we represent keys and queries mathematically?

Keys and Queries

For each input, we create a key $\boldsymbol{k}$

Keys and Queries

For each input, we create a key $\boldsymbol{k}$

$$\boldsymbol{k}_j = \boldsymbol{\theta}_K^\top \boldsymbol{x}_j, \quad \boldsymbol{\theta}_K \in \mathbb{R}^{d_x \times d_h}, \quad \boldsymbol{k}_j \in \mathbb{R}^{d_h}$$

Keys and Queries

For each input, we create a key $\boldsymbol{k}$

$$\boldsymbol{k}_j = \boldsymbol{\theta}_K^\top \boldsymbol{x}_j, \quad \boldsymbol{\theta}_K \in \mathbb{R}^{d_x \times d_h}, \quad \boldsymbol{k}_j \in \mathbb{R}^{d_h}$$

Keys and Queries

Now, create a query from some $\boldsymbol{x}_q$

Keys and Queries

Now, create a query from some $\boldsymbol{x}_q$

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \boldsymbol{x}_q, \quad \boldsymbol{\theta}_Q \in \mathbb{R}^{d_x \times d_h}, \quad \boldsymbol{q} \in \mathbb{R}^{d_h}$$

Keys and Queries

Now, create a query from some $\boldsymbol{x}_q$

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \boldsymbol{x}_q, \quad \boldsymbol{\theta}_Q \in \mathbb{R}^{d_x \times d_h}, \quad \boldsymbol{q} \in \mathbb{R}^{d_h}$$

To determine if a key and query match, we will take the dot product

Keys and Queries

Now, create a query from some $\boldsymbol{x}_q$

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \boldsymbol{x}_q, \quad \boldsymbol{\theta}_Q \in \mathbb{R}^{d_x \times d_h}, \quad \boldsymbol{q} \in \mathbb{R}^{d_h}$$

To determine if a key and query match, we will take the dot product

$$\boldsymbol{q}^\top \boldsymbol{k}_i = \left(\boldsymbol{\theta}_Q^\top \boldsymbol{x}_q\right)^\top \left(\boldsymbol{\theta}_K^\top \boldsymbol{x}_i\right)$$

Keys and Queries

Now, create a query from some $\boldsymbol{x}_q$

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \boldsymbol{x}_q, \quad \boldsymbol{\theta}_Q \in \mathbb{R}^{d_x \times d_h}, \quad \boldsymbol{q} \in \mathbb{R}^{d_h}$$

To determine if a key and query match, we will take the dot product

$$\boldsymbol{q}^\top \boldsymbol{k}_i = \left(\boldsymbol{\theta}_Q^\top \boldsymbol{x}_q\right)^\top \left(\boldsymbol{\theta}_K^\top \boldsymbol{x}_i\right)$$

Question: What is the shape of $\boldsymbol{q}^\top \boldsymbol{k}_i$?

Keys and Queries

Now, create a query from some $\boldsymbol{x}_q$

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \boldsymbol{x}_q, \quad \boldsymbol{\theta}_Q \in \mathbb{R}^{d_x \times d_h}, \quad \boldsymbol{q} \in \mathbb{R}^{d_h}$$

To determine if a key and query match, we will take the dot product

$$\boldsymbol{q}^\top \boldsymbol{k}_i = \left(\boldsymbol{\theta}_Q^\top \boldsymbol{x}_q\right)^\top \left(\boldsymbol{\theta}_K^\top \boldsymbol{x}_i\right)$$

Question: What is the shape of $\boldsymbol{q}^\top \boldsymbol{k}_i$?

Answer: $(1, d_h) \times (d_h, 1) = 1$, the output is a scalar

Keys and Queries

Now, create a query from some $\boldsymbol{x}_q$

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \boldsymbol{x}_q, \quad \boldsymbol{\theta}_Q \in \mathbb{R}^{d_x \times d_h}, \quad \boldsymbol{q} \in \mathbb{R}^{d_h}$$

To determine if a key and query match, we will take the dot product

$$\boldsymbol{q}^\top \boldsymbol{k}_i = \left(\boldsymbol{\theta}_Q^\top \boldsymbol{x}_q\right)^\top \left(\boldsymbol{\theta}_K^\top \boldsymbol{x}_i\right)$$

Question: What is the shape of $\boldsymbol{q}^\top \boldsymbol{k}_i$?

Answer: $(1, d_h) \times (d_h, 1) = 1$, the output is a scalar

Large dot product $\Rightarrow$ match! Small dot product $\Rightarrow$ no match.

Keys and Queries

Example:

Keys and Queries

Example:

$$\boldsymbol{k}_i = \boldsymbol{\theta}_K^\top$$

Keys and Queries

Example:

$$\boldsymbol{k}_i = \boldsymbol{\theta}_K^\top$$ 

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \text{ Musician}$$

Keys and Queries

Example:

$$\boldsymbol{k}_i = \boldsymbol{\theta}_K^\top$$ 

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \text{ Musician}$$

$$\boldsymbol{q}^\top \boldsymbol{k}_i = \left(\boldsymbol{\theta}_Q^\top \text{ Musician}\right)^\top \left(\boldsymbol{\theta}_K^\top \right.$$  $$\left. \right) = 100$$

Keys and Queries

Example:

$$\boldsymbol{k}_i = \boldsymbol{\theta}_K^\top$$ 

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \text{ Musician}$$

$$\boldsymbol{q}^\top \boldsymbol{k}_i = \left(\boldsymbol{\theta}_Q^\top \text{ Musician}\right)^\top \left(\boldsymbol{\theta}_K^\top \right.$$  $$\left. \right) = 100$$

Large attention!

Keys and Queries

Example:

$$\boldsymbol{k}_i = \boldsymbol{\theta}_K^\top$$

Keys and Queries

Example:

$$\boldsymbol{k}_i = \boldsymbol{\theta}_K^\top$$ 

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \text{ Mathematician}$$

Keys and Queries

Example:

$$\boldsymbol{k}_i = \boldsymbol{\theta}_K^\top$$ 

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \text{ Mathematician}$$

$$\boldsymbol{q}^\top \boldsymbol{k}_i = \left(\boldsymbol{\theta}_Q^\top \text{ Mathematician}\right)^\top \left(\boldsymbol{\theta}_K^\top \right.$$  $$\left. \right) = -50$$

Keys and Queries

Example:

$$\boldsymbol{k}_i = \boldsymbol{\theta}_K^\top$$ 

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \text{ Mathematician}$$

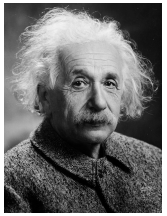
$$\boldsymbol{q}^\top \boldsymbol{k}_i = \left(\boldsymbol{\theta}_Q^\top \text{ Mathematician}\right)^\top \left(\boldsymbol{\theta}_K^\top \right.$$  $$\left. \right) = -50$$

Small attention!

Keys and Queries

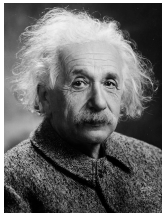
Example:

$$\boldsymbol{k}_i = \boldsymbol{\theta}_K^\top$$



Keys and Queries

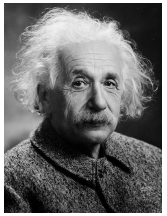
Example:

$$\boldsymbol{k}_i = \boldsymbol{\theta}_K^\top$$ 

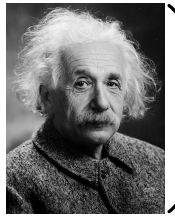
$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \text{ Mathematician}$$

Keys and Queries

Example:

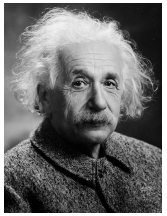
$$\boldsymbol{k}_i = \boldsymbol{\theta}_K^\top$$ 

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \text{ Mathematician}$$

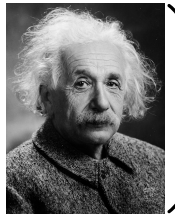
$$\boldsymbol{q}^\top \boldsymbol{k}_i = \left(\boldsymbol{\theta}_Q^\top \text{ Mathematician}\right)^\top \left(\boldsymbol{\theta}_K^\top \right.$$  $$\left. \right) = 90$$

Keys and Queries

Example:

$$\boldsymbol{k}_i = \boldsymbol{\theta}_K^\top$$ 

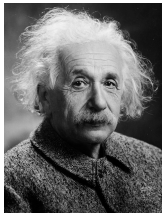
$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \text{ Mathematician}$$

$$\boldsymbol{q}^\top \boldsymbol{k}_i = \left(\boldsymbol{\theta}_Q^\top \text{ Mathematician}\right)^\top \left(\boldsymbol{\theta}_K^\top \right.$$  $$\left. \right) = 90$$

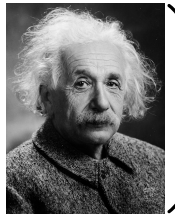
Large attention!

Keys and Queries

Example:

$$\boldsymbol{k}_i = \boldsymbol{\theta}_K^\top$$ 

$$\boldsymbol{q} = \boldsymbol{\theta}_Q^\top \text{ Mathematician}$$

$$\boldsymbol{q}^\top \boldsymbol{k}_i = \left(\boldsymbol{\theta}_Q^\top \text{ Mathematician}\right)^\top \left(\boldsymbol{\theta}_K^\top \right.$$  $$\left. \right) = 90$$

Large attention!

Remember, there are multiple inputs to pay attention to

Keys and Queries

We compute a key for each input

$$\boldsymbol{K} = [\boldsymbol{k}_1 \;\; \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{k}_T] = \left[\boldsymbol{\theta}_K^\top \boldsymbol{x}_1 \;\; \boldsymbol{\theta}_K^\top \boldsymbol{x}_2 \;\; \ldots \;\; \boldsymbol{\theta}_K^\top \boldsymbol{x}_T\right], \quad \boldsymbol{K} \in \mathbb{R}^{d_h \times T}$$

Keys and Queries

We compute a key for each input

$$\boldsymbol{K} = [\boldsymbol{k}_1 \;\; \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{k}_T] = \left[\boldsymbol{\theta}_K^\top \boldsymbol{x}_1 \;\; \boldsymbol{\theta}_K^\top \boldsymbol{x}_2 \;\; \ldots \;\; \boldsymbol{\theta}_K^\top \boldsymbol{x}_T\right], \quad \boldsymbol{K} \in \mathbb{R}^{d_h \times T}$$

Keys and Queries

Then compute attention for each key

$$\boldsymbol{q}^\top \boldsymbol{K} = \boldsymbol{q}^\top [\boldsymbol{k}_1 \;\; \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{k}_T] = [\boldsymbol{q}^\top \boldsymbol{k}_1 \;\; \boldsymbol{q}^\top \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{q}^\top \boldsymbol{k}_T]$$

Keys and Queries

Then compute attention for each key

$$\boldsymbol{q}^\top \boldsymbol{K} = \boldsymbol{q}^\top [\boldsymbol{k}_1 \;\; \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{k}_T] = [\boldsymbol{q}^\top \boldsymbol{k}_1 \;\; \boldsymbol{q}^\top \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{q}^\top \boldsymbol{k}_T]$$

$$\boldsymbol{q}^\top \in \mathbb{R}^{1,d_h}, \quad \boldsymbol{K} \in \mathbb{R}^{d_h \times T}$$

Keys and Queries

Then compute attention for each key

$$\boldsymbol{q}^\top \boldsymbol{K} = \boldsymbol{q}^\top [\boldsymbol{k}_1 \;\; \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{k}_T] = [\boldsymbol{q}^\top \boldsymbol{k}_1 \;\; \boldsymbol{q}^\top \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{q}^\top \boldsymbol{k}_T]$$

$$\boldsymbol{q}^\top \in \mathbb{R}^{1,d_h}, \quad \boldsymbol{K} \in \mathbb{R}^{d_h \times T}$$

Question: What is the shape of $\boldsymbol{q}^\top \boldsymbol{K}$?

Keys and Queries

Then compute attention for each key

$$\boldsymbol{q}^\top \boldsymbol{K} = \boldsymbol{q}^\top [\boldsymbol{k}_1 \;\; \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{k}_T] = [\boldsymbol{q}^\top \boldsymbol{k}_1 \;\; \boldsymbol{q}^\top \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{q}^\top \boldsymbol{k}_T]$$

$$\boldsymbol{q}^\top \in \mathbb{R}^{1,d_h}, \quad \boldsymbol{K} \in \mathbb{R}^{d_h \times T}$$

Question: What is the shape of $\boldsymbol{q}^\top \boldsymbol{K}$?

Answer: T

Keys and Queries

Then compute attention for each key

$$\boldsymbol{q}^\top \boldsymbol{K} = \boldsymbol{q}^\top [\boldsymbol{k}_1 \;\; \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{k}_T] = [\boldsymbol{q}^\top \boldsymbol{k}_1 \;\; \boldsymbol{q}^\top \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{q}^\top \boldsymbol{k}_T]$$

$$\boldsymbol{q}^\top \in \mathbb{R}^{1,d_h}, \quad \boldsymbol{K} \in \mathbb{R}^{d_h \times T}$$

Question: What is the shape of $\boldsymbol{q}^\top \boldsymbol{K}$?

Answer: T

Do not forget to normalize with softmax!

Keys and Queries

Normalize, only pay attention to important matches

Keys and Queries

Normalize, only pay attention to important matches

$$\begin{aligned}\text{softmax}(\boldsymbol{q}^\top \boldsymbol{K}) &= \text{softmax}(\boldsymbol{q}^\top [\boldsymbol{k}_1 \;\; \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{k}_T]) \\ &= \text{softmax}([\boldsymbol{q}^\top \boldsymbol{k}_1 \;\; \boldsymbol{q}^\top \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{q}^\top \boldsymbol{k}_T])\end{aligned}$$

Keys and Queries

Normalize, only pay attention to important matches

$$\begin{aligned}\text{softmax}(\boldsymbol{q}^\top \boldsymbol{K}) &= \text{softmax}(\boldsymbol{q}^\top [\boldsymbol{k}_1 \;\; \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{k}_T]) \\ &= \text{softmax}([\boldsymbol{q}^\top \boldsymbol{k}_1 \;\; \boldsymbol{q}^\top \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{q}^\top \boldsymbol{k}_T])\end{aligned}$$

We call this **dot-product attention**

Keys and Queries

Query: Which person will help me on my exam?

Keys and Queries

Query: Which person will help me on my exam?

$$\boldsymbol{q}^\top \boldsymbol{k}_1 \qquad \boldsymbol{q}^\top \boldsymbol{k}_2 \qquad \boldsymbol{q}^\top \boldsymbol{k}_3 \qquad \boldsymbol{q}^\top \boldsymbol{k}_4 \qquad \boldsymbol{q}^\top \boldsymbol{k}_5$$

Keys and Queries

Query: Which person will help me on my exam?

$\boldsymbol{q}^\top \boldsymbol{k}_1$	$\boldsymbol{q}^\top \boldsymbol{k}_2$	$\boldsymbol{q}^\top \boldsymbol{k}_3$	$\boldsymbol{q}^\top \boldsymbol{k}_4$	$\boldsymbol{q}^\top \boldsymbol{k}_5$
-1.71	0.60	-1.01	-0.61	2.73

softmax

Keys and Queries

Query: Which person will help me on my exam?

$\boldsymbol{q}^\top \boldsymbol{k}_1$	$\boldsymbol{q}^\top \boldsymbol{k}_2$	$\boldsymbol{q}^\top \boldsymbol{k}_3$	$\boldsymbol{q}^\top \boldsymbol{k}_4$	$\boldsymbol{q}^\top \boldsymbol{k}_5$
-1.71	0.60	-1.01	-0.61	2.73
		softmax		
0.01	0.10	0.02	0.03	0.84

Keys and Queries

Query: Which person will help me on my exam?

$\boldsymbol{q}^\top \boldsymbol{k}_1$	$\boldsymbol{q}^\top \boldsymbol{k}_2$	$\boldsymbol{q}^\top \boldsymbol{k}_3$	$\boldsymbol{q}^\top \boldsymbol{k}_4$	$\boldsymbol{q}^\top \boldsymbol{k}_5$
-1.71	0.60	-1.01	-0.61	2.73
		softmax		
0.01	0.10	0.02	0.03	0.84

Keys and Queries

Put dot product attention into our composite model

Keys and Queries

Put dot product attention into our composite model

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_i = \text{softmax}(\boldsymbol{q}^\top[\boldsymbol{k}_1 \;\; \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{k}_T])_i$$

Keys and Queries

Put dot product attention into our composite model

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_i = \text{softmax}(\boldsymbol{q}^\top [\boldsymbol{k}_1 \;\; \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{k}_T])_i$$

Then, write our composite memory model with dot product attention

Keys and Queries

Put dot product attention into our composite model

$$\lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_i = \text{softmax}(\boldsymbol{q}^\top[\boldsymbol{k}_1 \;\; \boldsymbol{k}_2 \;\; \ldots \;\; \boldsymbol{k}_T])_i$$

Then, write our composite memory model with dot product attention

$$f\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}\right) = \sum_{i=1}^{T} \boldsymbol{\theta}^\top \boldsymbol{x}_i \cdot \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_i$$

Keys and Queries

$$f\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}\right) = \sum_{i=1}^{T} \boldsymbol{\theta}^\top \boldsymbol{x}_i \cdot \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_i$$

Keys and Queries

$$f\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}\right) = \sum_{i=1}^{T} \boldsymbol{\theta}^\top \boldsymbol{x}_i \cdot \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_i$$

We relabel $\boldsymbol{\theta}$ to $\boldsymbol{\theta}_V$

$$f\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}\right) = \sum_{i=1}^{T} \boldsymbol{\theta}_V^\top \boldsymbol{x}_i \cdot \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_i$$

Keys and Queries

$$f\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}\right) = \sum_{i=1}^{T} \boldsymbol{\theta}^\top \boldsymbol{x}_i \cdot \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_i$$

We relabel $\boldsymbol{\theta}$ to $\boldsymbol{\theta}_V$

$$f\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}\right) = \sum_{i=1}^{T} \boldsymbol{\theta}_V^\top \boldsymbol{x}_i \cdot \lambda\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}_\lambda\right)_i$$

In dot-product attention, we call $\boldsymbol{\theta}_V^\top \boldsymbol{x}_i$ the **value**

Self Attention

Self Attention

Previously, we chose our own query $\boldsymbol{x}_q = \text{Musician}$

Self Attention

Previously, we chose our own query $\boldsymbol{x}_q = \text{Musician}$

We can also create queries from all the inputs

Self Attention

Previously, we chose our own query $\boldsymbol{x}_q = \text{Musician}$

We can also create queries from all the inputs

$$\boldsymbol{Q} = [\boldsymbol{q}_1 \;\; \boldsymbol{q}_2 \;\; \dots \;\; \boldsymbol{q}_T] = \left[\boldsymbol{\theta}_Q^\top \boldsymbol{x}_1 \;\; \boldsymbol{\theta}_Q^\top \boldsymbol{x}_2 \;\; \dots \;\; \boldsymbol{\theta}_Q^\top \boldsymbol{x}_T\right], \quad \boldsymbol{Q} \in \mathbb{R}^{T \times d_h}$$

Self Attention

Previously, we chose our own query $\boldsymbol{x}_q = \text{Musician}$

We can also create queries from all the inputs

$$\boldsymbol{Q} = [\boldsymbol{q}_1 \;\; \boldsymbol{q}_2 \;\; \dots \;\; \boldsymbol{q}_T] = \left[\boldsymbol{\theta}_Q^\top \boldsymbol{x}_1 \;\; \boldsymbol{\theta}_Q^\top \boldsymbol{x}_2 \;\; \dots \;\; \boldsymbol{\theta}_Q^\top \boldsymbol{x}_T\right], \quad \boldsymbol{Q} \in \mathbb{R}^{T \times d_h}$$

We call this dot-product **self** attention

Self Attention

$$\boldsymbol{Q} = [\boldsymbol{q}_1 \ \ldots \ \boldsymbol{q}_T] = \left[\boldsymbol{\theta}_Q^\top \boldsymbol{x}_1 \ \ldots \ \boldsymbol{\theta}_Q^\top \boldsymbol{x}_T\right]$$

$$\boldsymbol{K} = [\boldsymbol{k}_1 \ \ldots \ \boldsymbol{k}_T] = \left[\boldsymbol{\theta}_K^\top \boldsymbol{x}_1 \ \ldots \ \boldsymbol{\theta}_K^\top \boldsymbol{x}_T\right]$$

$$\boldsymbol{V} = [\boldsymbol{v}_1 \ \ldots \ \boldsymbol{v}_T] = \left[\boldsymbol{\theta}_V^\top \boldsymbol{x}_1 \ \ldots \ \boldsymbol{\theta}_V^\top \boldsymbol{x}_T\right]$$

Self Attention

$$\boldsymbol{Q} = [\boldsymbol{q}_1 \ \ldots \ \boldsymbol{q}_T] = \left[\boldsymbol{\theta}_Q^\top \boldsymbol{x}_1 \ \ldots \ \boldsymbol{\theta}_Q^\top \boldsymbol{x}_T\right]$$

$$\boldsymbol{K} = [\boldsymbol{k}_1 \ \ldots \ \boldsymbol{k}_T] = \left[\boldsymbol{\theta}_K^\top \boldsymbol{x}_1 \ \ldots \ \boldsymbol{\theta}_K^\top \boldsymbol{x}_T\right]$$

$$\boldsymbol{V} = [\boldsymbol{v}_1 \ \ldots \ \boldsymbol{v}_T] = \left[\boldsymbol{\theta}_V^\top \boldsymbol{x}_1 \ \ldots \ \boldsymbol{\theta}_V^\top \boldsymbol{x}_T\right]$$

$$\boldsymbol{Q}\boldsymbol{K}^\top = \begin{bmatrix} \boldsymbol{q}_1 \\ \vdots \\ \boldsymbol{q}_T \end{bmatrix} [\boldsymbol{k}_1 \ \ldots \ \boldsymbol{k}_T] = \begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \ldots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \ldots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \ldots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix}$$

Self Attention

$$\boldsymbol{V} = [\boldsymbol{v}_1 \ \ldots \ \boldsymbol{v}_T] = \left[\boldsymbol{\theta}_V^\top \boldsymbol{x}_1 \ \ldots \ \boldsymbol{\theta}_V^\top \boldsymbol{x}_T\right]$$

$$\boldsymbol{Q}\boldsymbol{K}^\top = \begin{bmatrix} \boldsymbol{q}_1 \\ \vdots \\ \boldsymbol{q}_T \end{bmatrix} [\boldsymbol{k}_1 \ \ldots \ \boldsymbol{k}_T] = \begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \ldots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \ldots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \ldots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix}$$

$$\boldsymbol{Q}\boldsymbol{K}^\top\boldsymbol{V} = \begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \ldots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \ldots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \ldots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1\boldsymbol{v}_1 + \ldots + \boldsymbol{q}_1\boldsymbol{k}_T\boldsymbol{v}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1\boldsymbol{v}_1 + \ldots + \boldsymbol{q}_2\boldsymbol{k}_T\boldsymbol{v}_T \\ \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1\boldsymbol{v}_1 + \ldots + \boldsymbol{q}_T\boldsymbol{k}_T\boldsymbol{v}_T \end{bmatrix}$$

Self Attention

$$\boldsymbol{QK^\top V} = \begin{bmatrix} \boldsymbol{q_1 k_1} & \dots & \boldsymbol{q_1 k_T} \\ \boldsymbol{q_2 k_1} & \dots & \boldsymbol{q_2 k_T} \\ \vdots & \ddots & \vdots \\ \boldsymbol{q_T k_1} & \dots & \boldsymbol{q_T k_T} \end{bmatrix} \begin{bmatrix} \boldsymbol{v_1} \\ \boldsymbol{v_2} \\ \vdots \\ \boldsymbol{v_T} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q_1 k_1 v_1} + \dots + \boldsymbol{q_1 k_T v_T} \\ \boldsymbol{q_2 k_1 v_1} + \dots + \boldsymbol{q_2 k_T v_T} \\ \vdots \\ \boldsymbol{q_T k_1 v_1} + \dots + \boldsymbol{q_T k_T v_T} \end{bmatrix}$$

Self Attention

$$\boldsymbol{QK}^\top \boldsymbol{V} = \begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \dots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \dots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \dots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1\boldsymbol{v}_1 + \dots + \boldsymbol{q}_1\boldsymbol{k}_T\boldsymbol{v}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1\boldsymbol{v}_1 + \dots + \boldsymbol{q}_2\boldsymbol{k}_T\boldsymbol{v}_T \\ \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1\boldsymbol{v}_1 + \dots + \boldsymbol{q}_T\boldsymbol{k}_T\boldsymbol{v}_T \end{bmatrix}$$

Question: Did we forget anything?

Self Attention

$$\boldsymbol{QK}^\top \boldsymbol{V} = \begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \dots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \dots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \dots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1\boldsymbol{v}_1 + \dots + \boldsymbol{q}_1\boldsymbol{k}_T\boldsymbol{v}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1\boldsymbol{v}_1 + \dots + \boldsymbol{q}_2\boldsymbol{k}_T\boldsymbol{v}_T \\ \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1\boldsymbol{v}_1 + \dots + \boldsymbol{q}_T\boldsymbol{k}_T\boldsymbol{v}_T \end{bmatrix}$$

Question: Did we forget anything? **Answer:** Softmax!

Self Attention

$$\boldsymbol{Q}\boldsymbol{K}^\top\boldsymbol{V} = \begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \dots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \dots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \dots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1\boldsymbol{v}_1 + \dots + \boldsymbol{q}_1\boldsymbol{k}_T\boldsymbol{v}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1\boldsymbol{v}_1 + \dots + \boldsymbol{q}_2\boldsymbol{k}_T\boldsymbol{v}_T \\ \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1\boldsymbol{v}_1 + \dots + \boldsymbol{q}_T\boldsymbol{k}_T\boldsymbol{v}_T \end{bmatrix}$$

Question: Did we forget anything? **Answer:** Softmax!

$$\text{softmax}(\boldsymbol{Q}\boldsymbol{K}^\top)\boldsymbol{V} = \text{softmax}\left(\begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \dots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \dots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \dots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_T \end{bmatrix}$$

Self Attention

$$\text{softmax}(\boldsymbol{Q}\boldsymbol{K}^\top)\boldsymbol{V} = \text{softmax}\left(\begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \dots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \dots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \dots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix}\right)\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_T \end{bmatrix}$$

Self Attention

$$\text{softmax}(\boldsymbol{Q}\boldsymbol{K}^\top)\boldsymbol{V} = \text{softmax}\left(\begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \dots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \dots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \dots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix}\right)\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_T \end{bmatrix}$$

Be very careful with softmax dimension

Self Attention

$$\text{softmax}(\boldsymbol{QK}^\top)\boldsymbol{V} = \text{softmax}\left(\begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \dots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \dots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \dots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix}\right)\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_T \end{bmatrix}$$

Be very careful with softmax dimension

- Writing attention different ways uses different softmax dimensions

Self Attention

$$\text{softmax}(\boldsymbol{QK}^\top)\boldsymbol{V} = \text{softmax}\left(\begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \dots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \dots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \dots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix}\right)\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_T \end{bmatrix}$$

Be very careful with softmax dimension

- Writing attention different ways uses different softmax dimensions
- When in doubt, write out full matrix

Self Attention

$$\text{softmax}(\boldsymbol{Q}\boldsymbol{K}^\top)\boldsymbol{V} = \text{softmax}\left(\begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \dots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \dots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \dots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix}\right)\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_T \end{bmatrix}$$

Be very careful with softmax dimension

- Writing attention different ways uses different softmax dimensions
- When in doubt, write out full matrix

Question: Softmax rows or columns? **Answer:** Rows

Self Attention

$$\text{softmax}(\boldsymbol{QK}^\top)\boldsymbol{V} = \text{softmax}\left(\begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \dots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \dots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \dots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix}\right)\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_T \end{bmatrix}$$

Be very careful with softmax dimension

- Writing attention different ways uses different softmax dimensions
- When in doubt, write out full matrix

Question: Softmax rows or columns? **Answer:** Rows

Only one query index $\text{softmax}(\begin{bmatrix} \boldsymbol{q}_i\boldsymbol{k}_1 & \dots & \boldsymbol{q}_i\boldsymbol{k}_T \end{bmatrix})$

Self Attention

$$\text{softmax}(\boldsymbol{QK}^\top)\boldsymbol{V} = \text{softmax}\left(\begin{bmatrix} \boldsymbol{q}_1\boldsymbol{k}_1 & \dots & \boldsymbol{q}_1\boldsymbol{k}_T \\ \boldsymbol{q}_2\boldsymbol{k}_1 & \dots & \boldsymbol{q}_2\boldsymbol{k}_T \\ \vdots & \ddots & \vdots \\ \boldsymbol{q}_T\boldsymbol{k}_1 & \dots & \boldsymbol{q}_T\boldsymbol{k}_T \end{bmatrix}\right)\begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_T \end{bmatrix}$$

Be very careful with softmax dimension

- Writing attention different ways uses different softmax dimensions
- When in doubt, write out full matrix

Question: Softmax rows or columns? **Answer:** Rows

Only one query index $\text{softmax}(\begin{bmatrix} \boldsymbol{q}_i\boldsymbol{k}_1 & \dots & \boldsymbol{q}_i\boldsymbol{k}_T \end{bmatrix})$

Self Attention

Attention paper suggests normalizing constant for faster learning

Self Attention

Attention paper suggests normalizing constant for faster learning

$$\text{attn}\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}\right) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_h}}\right)\boldsymbol{V}$$

Self Attention

Attention paper suggests normalizing constant for faster learning

$$\text{attn}\left(\begin{bmatrix}\boldsymbol{x}_1\\ \vdots \\ \boldsymbol{x}_T\end{bmatrix}, \boldsymbol{\theta}\right) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_h}}\right)\boldsymbol{V}$$

This operation powers today's biggest models

Self Attention

$$\boldsymbol{Q} \in \mathbb{R}^{T \times d_h} \quad \boldsymbol{K} \in \mathbb{R}^{T \times d_h} \quad \boldsymbol{V} \in \mathbb{R}^{T \times d_h}$$

$$\underbrace{\text{attn}\left(\begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix}, \boldsymbol{\theta}\right)}_{\mathbb{R}^{T \times d_x}} = \text{softmax}\overbrace{\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top,}{\sqrt{d_h}}\right)}^{\mathbb{R}^{T \times T}} \underbrace{\boldsymbol{V}}_{\mathbb{R}^{T \times d_h}}$$

Self Attention

$$\boldsymbol{Q} \in \mathbb{R}^{T \times d_h} \quad \boldsymbol{K} \in \mathbb{R}^{T \times d_h} \quad \boldsymbol{V} \in \mathbb{R}^{T \times d_h}$$

$$\underbrace{\text{attn}\left(\begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix}, \boldsymbol{\theta}\right)}_{\mathbb{R}^{T \times d_x}} = \text{softmax}\overbrace{\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top,}{\sqrt{d_h}}\right)}^{\mathbb{R}^{T \times T}} \underbrace{\boldsymbol{V}}_{\mathbb{R}^{T \times d_h}}$$

Self Attention

```
class Attention(nn.Module):
    def __init__(self):
        self.theta_K = nn.Linear(d_x, d_h, bias=False)
        self.theta_Q = nn.Linear(d_x, d_h, bias=False)
        self.theta_V = nn.Linear(d_x, d_h, bias=False)

    def forward(self, x):
        # Shape:  (column, row), be very careful with axis!
        A = softmax(
            self.theta_Q(x) @ self.theta_K(x).T / d_h, axis=1
        )
        V = self.theta_V(x)
        return A @ V
```

Guest Lecture - Dr. Matteo Bettini

Guest Lecture - Dr. Matteo Bettini

Dr. Matteo Bettini is a good friend of mine

Guest Lecture - Dr. Matteo Bettini

Dr. Matteo Bettini is a good friend of mine

We did our PhDs together

Guest Lecture - Dr. Matteo Bettini

Dr. Matteo Bettini

- Incoming Research Scientist at Meta's SuperIntelligence Lab

Guest Lecture - Dr. Matteo Bettini

Dr. Matteo Bettini

- Incoming Research Scientist at Meta's SuperIntelligence Lab
- Focus on agentic LLMs

Guest Lecture - Dr. Matteo Bettini

Dr. Matteo Bettini

- Incoming Research Scientist at Meta's SuperIntelligence Lab
- Focus on agentic LLMs
- PhD in Computer Science at Cambridge

Guest Lecture - Dr. Matteo Bettini

Dr. Matteo Bettini

- Incoming Research Scientist at Meta's SuperIntelligence Lab
- Focus on agentic LLMs
- PhD in Computer Science at Cambridge
- MS in Computer Science at Cambridge

Guest Lecture - Dr. Matteo Bettini

Dr. Matteo Bettini

- Incoming Research Scientist at Meta's SuperIntelligence Lab
- Focus on agentic LLMs
- PhD in Computer Science at Cambridge
- MS in Computer Science at Cambridge
- BS in Computer Engineering at Milan Polytechnic