# DL - L18. 11.07.2022

10-13

Correlation - linear    attention - non-linear

10-14

$a_1 \dots a_N$ is the attention value of the query $q$ wrt keys $k$.
The value $a$ gives the value by which $q_i$ and $k_i$ are close to each other

Given:

- a single query $\underline{q} \in \mathbb{R}^d$
- N key-value pairs $\underline{k}_i \in \mathbb{R}^d$, $\underline{v}_i \in \mathbb{R}^{d_v}$ $1 \leq i \leq N$

Let $\underline{\underline{K}} = [\underline{k}_1 \dots \underline{k}_N] \in \mathbb{R}^{d \times N}$, $\underline{\underline{V}} = [\underline{v}_1 \dots \underline{v}_N] \in \mathbb{R}^{d_v \times N}$

Steps

- Calculate similarities $a_i = S(\underline{q}, \underline{k}_i)$ between $\underline{q}$ and $\underline{k}_i$
- Softmax normalization

$$\alpha_i = \frac{e^{a_i}}{\sum_{i=1}^{N} e^{a_i}} \geq 0 \qquad \sum_{i=1}^{N} \alpha_i = 1$$

- Convex connection of $\underline{v}_i$:

$$\text{attention}(\underline{q}, \underline{\underline{K}}, \underline{\underline{V}}) = \sum_{i=1}^{N} \alpha_i \underline{v}_i \in \mathbb{R}^{d_v}$$

10-15

People mainly used scaled dot product

$-1 \leq$ cosine similarity $\leq 1$

Assuming $s(\underline{q}, \underline{k}_i) = \underline{k}^T \underline{q} / \sqrt{d}$ ,

$$\text{attention}(\underline{q}, \underline{\underline{K}}, \underline{\underline{V}}) = \underbrace{\underset{d_v \times N}{\underline{\underline{V}}} \cdot \underbrace{\text{softmax}(\underbrace{\underset{N \times d}{\underline{\underline{K}}^T} \cdot \underset{d \times 1}{\underline{q}}}_{N \times 1} / \sqrt{d})}_{N \times 1}}_{d_v \times 1} \in \mathbb{R}^{d_v}$$

10-16

$\underline{\underline{K}} \in \mathbb{R}^{d \times N}$ $\underline{\underline{V}} \in \mathbb{R}^{d_v \times N}$

$$\text{softmax}(\underline{\underline{Q}}, \underline{\underline{K}}, \underline{\underline{V}}) = \underbrace{\underset{d_v \times N}{\underline{\underline{V}}} \cdot \text{softmax}(\underbrace{\underset{N \times d}{\underline{\underline{K}}^T} \underset{d \times M}{\underline{\underline{Q}}}}_{N \times M} / \sqrt{d})}_{d_v \times M}$$

10-17

10-18

## 10.2.2 Transformer in Natural Language Processing (NLP)

NLP : deals with sequence of words

seq2seq model: exploit the relationship between different words in an input sequence

input sequence → $\boxed{\text{seq2seq}}$ → output sequence

- translation
- Speech recognition, speech synthesis

How to calculate with text?

→ word embedding
translate words to numbers

10 - 19

10 - 20

10 - 21

10 - 22

10 - 23

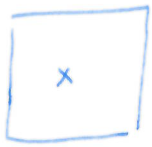
10 - 24

## 10.2.3 Transformer in Computer Vision

Apply self-attention to images

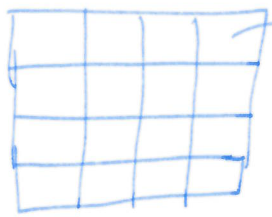
At which level ?

(a) at input pixel level :



too many pixels → not efficient

(b) at input patch level :



→ flatten layer → ≙ vec ( )

↓

column vector

≙ NLP

10 - 25

(c) at the feature level

input → CNN → transformer → .... downstream tasks

10 - 26

10 - 27

## 10.3 Networks and modules for image classification

10-28