Last time

- "auxiliary" layers (Batch Normalization, Dropout, Residual Layer)

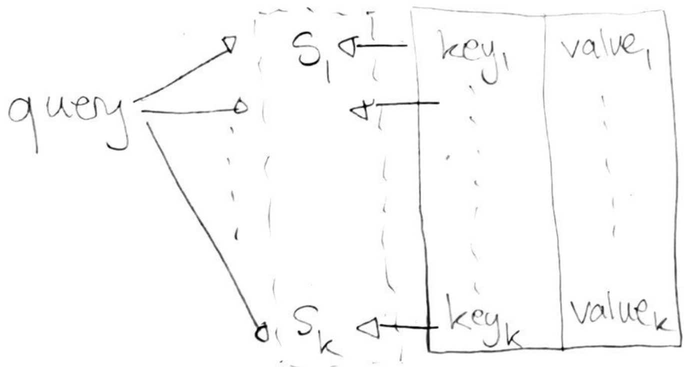
Today

- Self - Attention Layer
- Wrap Up

Self Attention Layer

- the attention mechanism is the key innovation underpinning the very successful Transformer architectures that have been used to build chatbots like Chat GPT (generative pre-trained transformer)
- Key paper: "Attention is all you need", Vaswani et al 2017
- Idea: The templates being checked for are dependent on the content of each observation

Intuition: In a relational database, a select operation checks similarity between the query q and a set of keys k_j, returning the value v_j corresponding to the key most similar to the query.



$$\text{select}(q, K, V) = \sum_{j=1}^{k} \text{sim}(q, k_j) \cdot v_j$$

where $\text{sim}(q, k_j) = 1$ for exactly one k_j and is zero for the rest.

$$\text{attention}(q, K, V) = \sum_{j=1}^{k} \text{sim}(q, k_j) \cdot v_j$$

where $\text{sim}(q, k_j) \in [0, 1]$ and $\sum_{j=1}^{k} \text{sim}(q, k_j) = 1$

$\rightarrow$ is a "soft" select

In more detail:

One observation X_i ([$1 \times p$]) as input

Divide observation into patches $X_{i1}, \ldots, X_{im}$ ([$1 \times d$])

($m \times d = p$)

$$Z_i = \begin{bmatrix} X_{i1} \\ \vdots \\ X_{im} \end{bmatrix} \quad ([m \times d])$$

Define:

$$Q = Z_i W_q + B_q \qquad [m \times e]$$

$$K = Z_i W_k + B_k \qquad [m \times e]$$

$$V = Z_i W_v + B_k \qquad [m \times d_2]$$

Next define similarity:

(Scaled dot product similarity)

$$\text{sim}(Q, K) = \text{softmax}\left(\frac{Q K^T}{\sqrt{e}}\right) \leftarrow [m \times m]$$

each row is a discrete probability distribution

$$\text{e.g. } \text{sim}(Q, K) = \begin{bmatrix} 0 & 1 & 0 \\ .7 & .3 & 0 \\ .5 & .45 & 0.5 \end{bmatrix} \leftarrow m = 3$$

→ The $\text{sim}(Q, K)$ matrix defines how each patch x_{ij} relates to any other patch x_{ik}.

→ in NLP: patches are tokens ≈ words

→ in CV: patches are... patches (of images)

Then:

$$\text{(self) attention}(X_i) = \text{softmax}\left(\frac{QK^T}{\sqrt{e}}\right) V \leftarrow [m \times d_2]$$

→ producing a new representation of X_i which combines information from across patches via a weighting that is itself a function of the input itself.

Wrap Up:

$$\hat{f} = \underset{f \in \mathcal{F}}{\text{opt}} \; L(f, D)$$

- losses are used to encode / define task performance on dataset D by candidate f
- optimization is used to search the candidate function space $\mathcal{F}$ for f's with low L
  - gradient descent
  - SGD
  - momentum
  - RMSProp
  - Adam
- the family of neural networks are a very rich $\mathcal{F}$.
  - Dense
  - Convolutional
  - Batch Normalization
  - Dropout
  - Residual
  - Attention