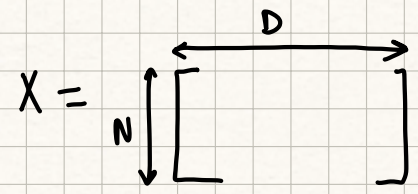
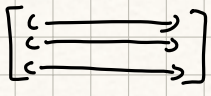
PCA ANALYSIS: EFFECTIVE BUFFER DIMENSIONALITY



GRAM MATRIX : the matrix of scalar products

$$\begin{cases} G_N := \underset{(N\times D)\,(D\times N)}{X \cdot X^T} \\ G_D := \underset{(D\times N)\,(N\times D)}{X^T \cdot X} \end{cases} \qquad G_D(i,j) = \mathrm{Cov}(x_i, x_j)$$

$$0 \le rk(X) \le N$$

$X =$ 

$\mathrm{span}\{rows(X)\}$ is a subspace of $\mathbb{R}^D$

the maximum dimension is N.

we want to find its dimension = the rank of matrix X.

STRATEGY: 1) center X: $X = X - X_{mean}$ (-1 degree of freedom now)

2) we apply PCA on the (normalized) Gram matrix $G_D := \left(\frac{1}{N-1}\right) \underset{(N,D)\,(D,N)}{X^T X}$. This has max rank (N-1). The number of non-null eigenvalues of $\Lambda = U G_N V^T$ gives the true dimension of X.

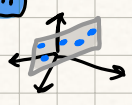
Max dim of N points is N-1 GO

$$\begin{cases} \max\dim(\vec{x}) = 0 \quad \text{(one point)} \\ \max\dim(\{\vec{x_1}, \vec{x_2}\}) = 1 \quad \text{(a straight line)} \\ \max\dim(\{\vec{x_1}, \vec{x_2}, \vec{x_3}\}) = 2 \quad \text{a plane} \end{cases}$$

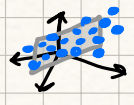
CAVEATS OF THE FINITE SAMPLE

- Λ has N eigenvalues → discrete spectrum.
- for data $N \to +\infty$, Λ has a continuous spectrum
- for data $N \to +\infty$, in particular, at some point $N > D$: therefore, the matrix X can reach rank D, which would mean that the true dimensionality of the buffer is really D and not lower

CASE A

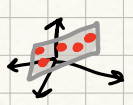


for moderate N there seems to exist a subspace D<N where points live

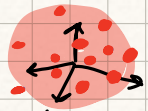


adding more and more points, it stays the same dimension

CASE B



for moderate N there seems to exist a subspace D<N where points live



but it disappears adding more and more points

PCA: an essential RECAP

$X = \left(\begin{array}{c} \\ \\ \\ \\ \end{array}\right)$ N indep. observations of variable $\vec{x}^T = (x_1, \dots x_D)$

we want to study the COVARIANCE between pairs of random variables. Easily show that this is given by

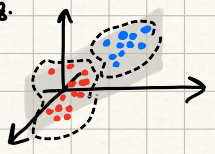
$$\Sigma := \widehat{Cov}(X_i, X_j) = \frac{(X - \bar{X})(X - \bar{X})^T}{N-1}$$

(sample estimate)

EXAMPLE:
features that don't vary or vary little are features we should care about (e.g. if we want to do clustering)

$$\begin{cases} X_1 = \mathcal{N}(0,1) \\ X_2 = \mathcal{N}(0,3) + X_1 \\ X_3 = 2X_1 + X_2 \end{cases}$$

X_3 determined completely from the other two



Σ is the NxN Gram matrix: $\Sigma = \left(\frac{1}{N-1}\right) \underset{(DxN)}{X^T} \underset{(NxD)}{X}$

Being square and sym, it can be diagonalized:

$\Lambda = \underset{T_{\mathcal{E}}^{\mathcal{B}}}{Q^T} \cdot \Sigma \cdot Q$. Λ is the covariance matrix of the transfored data $X' = X \cdot Q$. In the new basis $\mathcal{B}$ ($T_{\mathcal{E}}^{\mathcal{B}} = Q^T$) the random variables are independent because Λ is diagonal.

In the rotated frame, $\sigma^2(X_3') = 0$ (one coordinate is always zero)

The number of non-null eigenvalues gives the EFFECTIVE dimensionality of the dataset.

EIGENVALUE DENSITY DISTRIBUTION

$\mathbb{1}_N$ identity matrix

$\lambda = 1$ identity

all N eigenvalues are 1

Let's estimate $p(\lambda)$ from the observed empirical frequencies of each eigval

we could find an histogram like this...

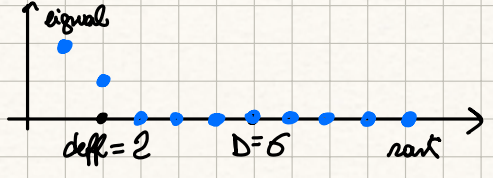
most eigvals are null

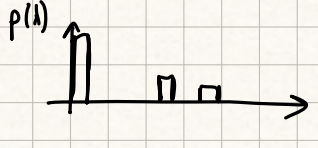
we should ask ourselves:

what would it become like in the large-N limit?

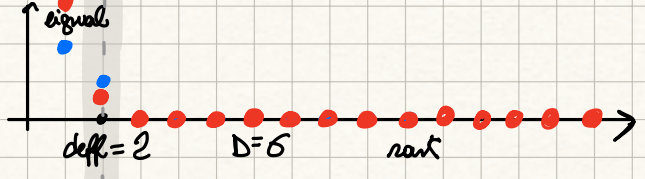
EIG. VS. RANK PLOT IF THE BUFFER REALLY HAS AN EFFECTIVE DIMENSIONALITY $d_{eff} < D$

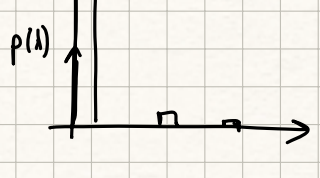
$N_1 = 10$





$N_2 = 15$



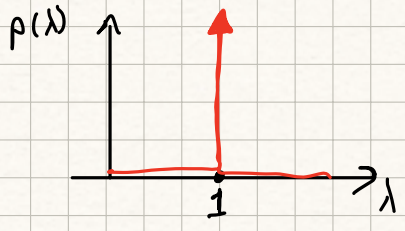


limit $p(\lambda) = \delta(0)$

(all last d_{eff} are ZERO)

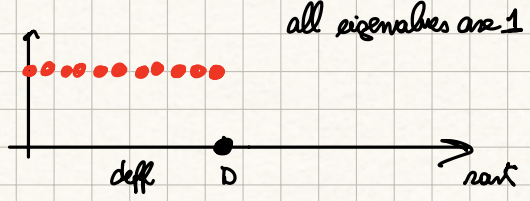
QUESTION: Why check $X^T X$ against $\mathbb{1}_D$?

$(D \times N)(N \times D)$



$p(\lambda) = \delta(\lambda - 1)$

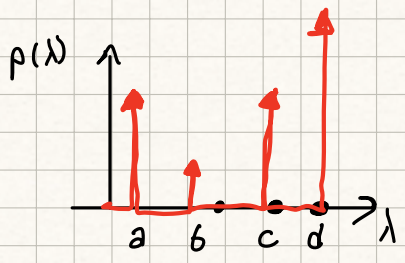
$$\mathbb{1}_D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



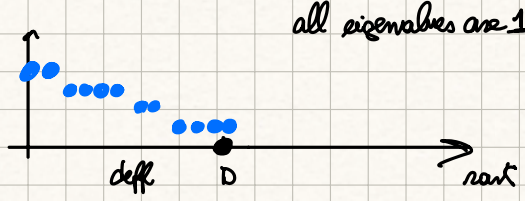
equal variance in all directions ⇒ max dimension, D

ball of noise (with FINITE variance! - doesn't span over all the available space)

other possibilities ...



$$\mathbb{1}_D = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{pmatrix}$$



equal variance in all directions ⇒ max dimension, D

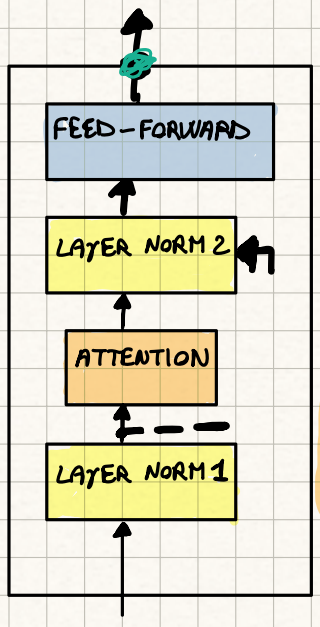
$p(\lambda) = \alpha \cdot \delta(\lambda - a) + \beta \cdot \delta(\lambda - b) + \gamma \cdot \delta(\lambda - c) + \zeta \cdot \delta(\lambda - d)$

I think all we said doesn't really apply!

we need to look at $X^T X$ for dimensionality / not the other!

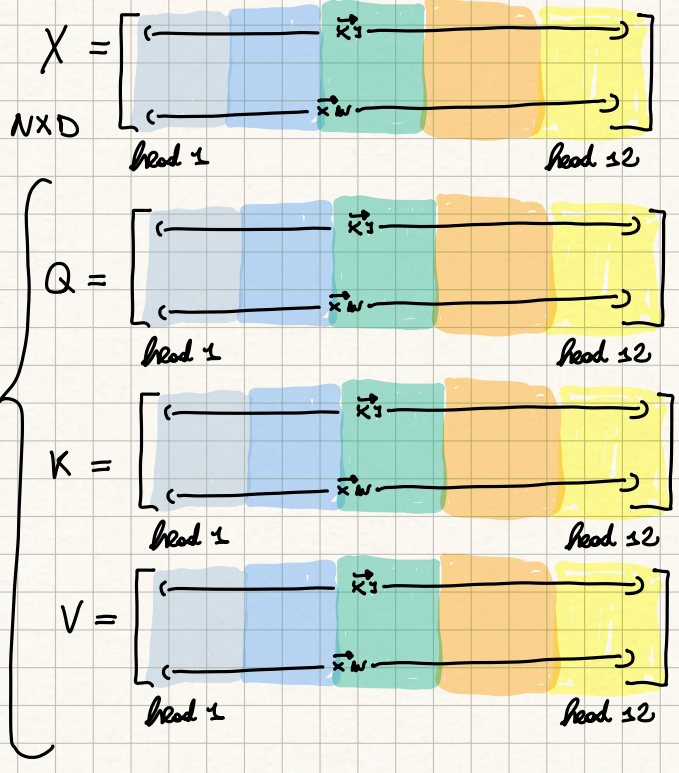
EQUATIONS: WHAT IS HAPPENING EXACTLY, STEP BY STEP

THE ATTENTION LAYER



c_proj_bias	D
c_proj_weight	DxD
c_attn_bias	3D
c_attn_weight	Dx3D

$n_{HEAD} = 12, \quad d_k = \frac{D}{n_{HEAD}} = 64$



STEP 1: create queries, values and keys

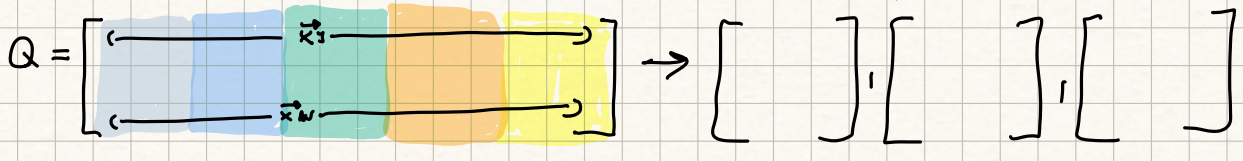
$(c\_attn\_weight, c\_attn\_bias)$
$D \times 3D \quad 3D$

$[Q, K, V] = X \cdot c\_attn\_weight + \begin{bmatrix} c\_attn\_bias \\ c\_attn\_bias \\ c\_attn\_bias \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} X \cdot \begin{bmatrix} c\_attn\_bias \\ c\_attn\_weight \end{bmatrix}$

stacked horizontally — equivalent

STEP 2: Split into heads

NOTICE: each chunk still carries inside the information coming from the WHOLE buffer X.



$Q_h = Q_h(X)$ and not $Q_h(X_h)$. It is a NON LINEAR operation.

STEP 3: compute attention weights and stack heads together

$$head\_i = \underset{N \times d_k}{} \; e^{\left(\frac{Q_h \cdot K_h^T}{\sqrt{d_k}}\right)} \cdot V_h \longrightarrow [head\_1, head\_2, \dots head_{n_H}]$$

$N \times N \quad N \times d_k \qquad N \times (d_k \times n_H) = N \times D$

STEP 4: "project back"?

$$Y = [\vec{1}, head\_1, head\_2, \dots head_{n_H}] \cdot \begin{bmatrix} c\_proj\_bias \\ c\_proj\_weight \end{bmatrix}$$

$N \times D+1 \qquad (D+1) \times D$

PROPOSALS OF ANALYSIS

1) QUESTION

Is a $Q = W_Q X$ a PROJECTION to a lower dim. space?

like in... $Y = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix}$ or isn't it?

to understand, we should check if the matrices Q, K and V are FULL RANK or not.

MY GUESS: They are NOT full rank.

2) $Q \cdot K^T = X W_Q W_K^T X$. How different from a symmetric matrix? How different from the identity?

If it was symmetric, the machinery would be over-complicated, because all would reduce to first transform $X' := XA$ and then take the NxN Gram matrix: $(X')(X')^T = (XA)(XA)^T = XAA^TX = XMX$ (Cholevski decomposition)

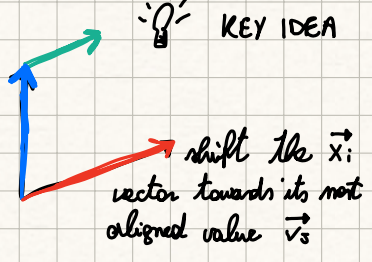
Even more, if it was $M \cong \mathbb{1}$, then attention weights would be nothing but scalar products of X on itself.

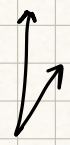
- GUESS Since $(QK^T)^T = (XMX^T)^T = XM^TX^T$, QK^T is symmetric $\Leftrightarrow$ M is symmetric.
- Make some prompts and show the attention weights: they usually are very asimmetric. Then also M is asymmetric, otherwise we would observe QK^T symmetric ALWAYS.

SPOILER: Indeed, M is ASYMMETRIC and differs from $\mathbb{1}$. Therefore, we know at least that the attention stage is doing something (non-trivial), even if we don't yet understand WHAT.

3) STEP-BY STEP DIMENSION ANALYSIS (?)

- $[Q, K, V] = X \cdot$ c_attn_weight dimension of Q, K, V?
- $\text{head}_h = \underbrace{\frac{1}{Z} \exp\left(\frac{Q_h \cdot K_h^T}{\sqrt{d_K}}\right)}_{\text{normalization to 1}} V_h = ?$ not decomposable more than this
- $Y = [\ ] \cdot W_O$



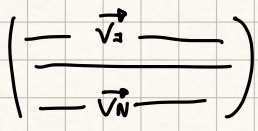


QK^T alignment and magnitude both matter (angle)

$$V \longrightarrow H = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V \qquad \vec{h_i} = \sum_j \text{weight}_{ij}\, \vec{V_j}$$

where weights sum up to 1

from



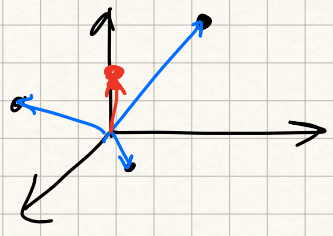
to

$$\begin{pmatrix} \vec{h_1}(\vec{V_1}, \vec{V_2}, \dots \vec{V_N}) \\ \vec{h_N}(\vec{V_1}, \vec{V_2}, \dots \vec{V_N}) \end{pmatrix}$$

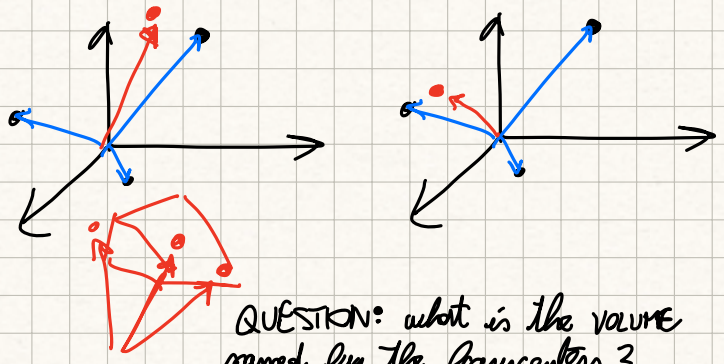
mixing all vectors together

BARYCENTER

each $\vec{h_i}$ is a differently weighted barycenter of the vectors $\{\vec{V}\}$

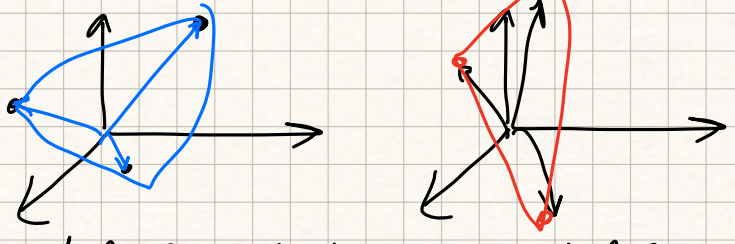


$$B_x = \sum m_i \cdot \vec{V}_{i,x}$$
$$B_y = \sum m_i \cdot \vec{V}_{i,y}$$
$$B_z = \sum m_i \cdot \vec{V}_{i,z}$$



QUESTION: what is the volume spanned by the barycenters?

smaller or larger?



set of values gets transformed in set of barycenters

ASYMMETRY OF THE WEIGHT MATRIX and DIRECTION OF MOVEMENTS

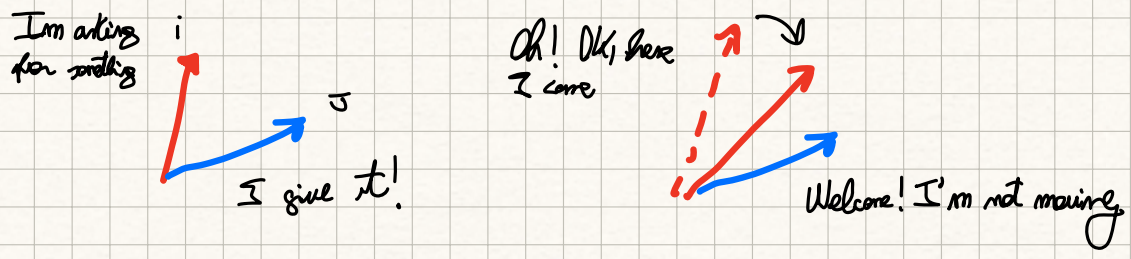
$$\vec{h_i} = \sum_j \text{weight}_{ij} \cdot \vec{V_j}$$

Notice that the fact that the matrix QK^T is asymmetric has a strong consequence:

The magnitude of weight_{ij} depends on how well the query of $\vec{x_i}$, $\vec{q_i}$ matched against the KEY of $\vec{x_j}$ (what it had to offer), not viceversa!

$$(QK^T)_{ij} = \langle \vec{q_i}, \vec{K_j} \rangle$$

EXAMPLE $\begin{cases} (QK^T)_{ij} = \langle \vec{q}_i, \vec{k}_j \rangle \neq 0 \\ \text{but } (QK)^T_{ji} = 0 \end{cases}$



Vectors move towards those vectors that have something to offer.

In MASKED attention, query of i cannot match key of j if j comes after, because i doesn't see j.

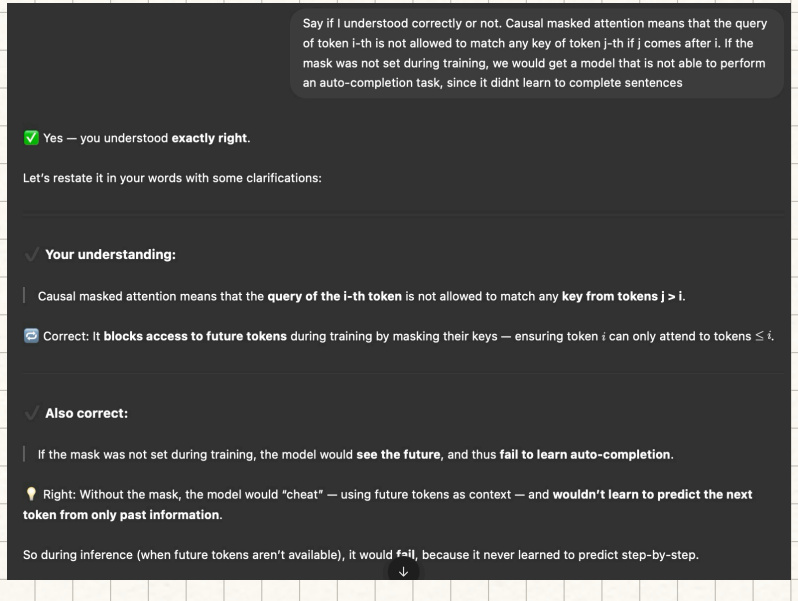
If we trained the model without restricting what it can see, it would not be able to auto-complete text when used in evaluation mode.

$$QK^T = \begin{matrix} \\ i \end{matrix} \begin{bmatrix} & 0 \\ & \\ & \end{bmatrix} \quad \begin{matrix} \\ \\ j \end{matrix}$$

$\vec{x}_i$ cannot ask anything to the vectors that come AFTER it.

I swam across the river learning

I swam across the ... auto-completion task



Say if I understood correctly or not. Causal masked attention means that the query of token i-th is not allowed to match any key of token j-th if j comes after i. If the mask was not set during training, we would get a model that is not able to perform an auto-completion task, since it didnt learn to complete sentences

✅ Yes — you understood **exactly right**.

Let's restate it in your words with some clarifications:

✓ **Your understanding:**

> Causal masked attention means that the **query of the i-th token** is not allowed to match any **key from tokens j > i**.

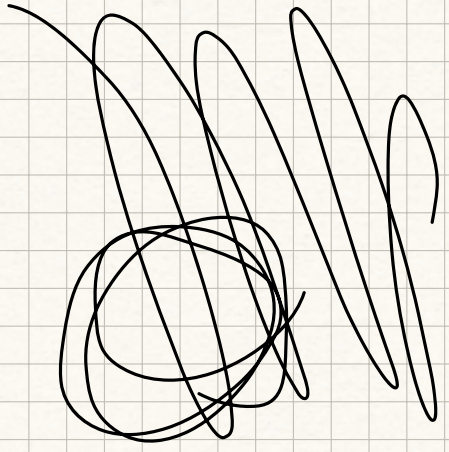
🔁 Correct: It **blocks access to future tokens** during training by masking their keys — ensuring token i can only attend to tokens $\leq i$.

✓ **Also correct:**

> If the mask was not set during training, the model would **see the future**, and thus **fail to learn auto-completion**.

💡 Right: Without the mask, the model would "cheat" — using future tokens as context — and **wouldn't learn to predict the next token from only past information**.

So during inference (when future tokens aren't available), it would fail, because it never learned to predict step-by-step.

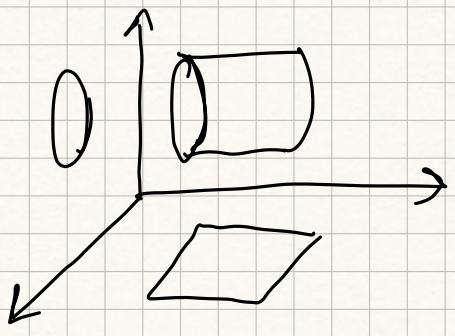


But then, do we also need the mask during inference?

MULTI-HEAD ATTENTION heads partition elements in different lower-dimensional subspaces

$$Q_h = X \cdot W_{Q,h}$$
$(N \times D)$ (D, d_k)

max rank is d_k



its like looking at an object FROM DIFFERENT POINTS OF VIEW

like in technical drawings at school

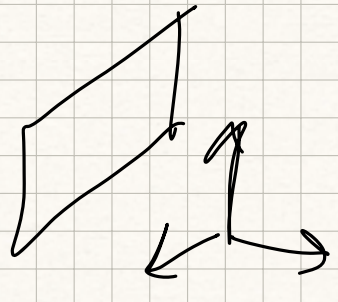
here projections are not ortogonal because there is just a change of basis (one is it a projection?)

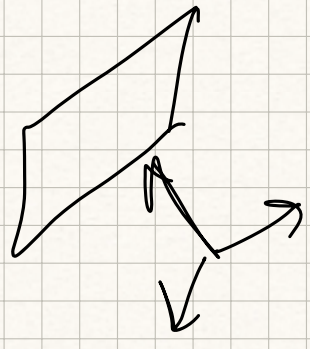
CHECK RANK of W_Q, W_K, W_V

What is

$$\underset{N \times D}{X} \cdot \underset{D \times D}{A}$$ doing on X ?

- When A is full rank, a change of basis
- when A is not, a projection.





if points are lying on a plane, and you change the basis, they are still lying on a plane.

So

$Q_h = \underset{N \times D}{X} \cdot \underset{D \times d_K}{W_{Q,h}}$ $\quad X \cdot [W_{Q,1}, W_{Q,2}, \dots W_{Q,n_h}]$

$\Downarrow$

$(Q_1)\ (Q_2) \quad (Q_h)$

$N \times d_K \quad N \times d_K \quad N \times d_K$

$W_{Q,i} \cdot W_{K,i}^T$

$W_Q \cdot W_K^T = [W_{Q,1}, W_{Q,2}, \dots W_{Q,n_h}] \cdot \begin{bmatrix} W_{K,1}^T \\ \vdots \\ W_{K,n_h}^T \end{bmatrix} = \sum W_{Q,i} \cdot W_{K,i}^T$

cambiare base e poi splittare è uguale a proiettare (con matrice) direttamente.