# Math for Deep Learning

## End-term Assignment

**Q(1)** An image can be viewed as a matrix.

A grayscale image can be modelled as a $m \times n$ matrix $A$.

For any $A \in M_{m \times n}(\mathbb{R})$, $\exists$ orthogonal matrices $U$ and $V$ and a diagonal matrix $\Sigma$ s.t.

$$A = U \Sigma V^T$$

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$$

where $\sigma_i$ = singular values
$r = \text{rank}(A)$.

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{r-1} \geq \sigma_r .$$

Firstly, we have reduced the number of entities we are storing from $mn$ to $r(m+n+1)$.

Secondly, in case of large images, we may store only the initial larger $\sigma_i$'s rather than storing entire matrix $\Sigma$, thereby leading to image compression.

Moreover, for images we have $\sigma_1 \gg \sigma_2 \gg \sigma_3 \gg \cdots \gg \sigma_r$.

[An elaborate mathematical proof exists.]

## Another reasoning

Let B be the obtained ~~img~~ image matrix after compression. Clearly, $\text{rank}(B) \le \text{rank}(A)$.

Problem: minimise $\|A - B\|_F$ where $\text{rank}(B) \le k$ (~~k~~ k is fixed)

$\hookrightarrow$ (Frobenius norm)

Eckart-Young-Mirsky theorem guarantees that SVD ~~provides an optimal soln~~ of A by taking only k $\sigma_i$'s provides an optimal soln.

### Example

Let $A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$.

For A, $\Sigma = \begin{bmatrix} 18.2 & 0 & 0 \\ 0 & 1.16 & 0 \\ 0 & 0 & 6.21 \times 10^{-16} \end{bmatrix}$

Clearly, ~~third~~ $\Sigma_{33}$ is negligibly small in comparison to $\Sigma_{11}$ and $\Sigma_{22}$ and may thus be neglected for practical purposes.