Low Rank Adaptation(LoRA)

WHAT?

Approximate representation of a matrix

$$\begin{bmatrix} 3 & 2 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix} \approx \begin{bmatrix} 2.99 & 2.01 & 0.98 \\ 0.02 & 1.88 & 1.1 \\ -0.07 & 0.45 & 0.29 \end{bmatrix}$$

Original matrix

The numbers in this matrix are very close to the original matrix

Low Rank Adaptation (LoRA)

WHY? Why do we want to approximate a matrix?

Bigger matrix-→ Lots of numbers (or elements)→ need more storage space and more computation

How about finding a way to store the matrix with less number of elements with approximate same values?

Yes! There is a way

We can store the matrix with much less elements →Much less storage space & Much less computation power needed

Low Rank Adaptation(LoRA)

How?

Through "**Rank of a Matrix**"

Rank= No. of independent columns or rows of a matrix

Let A be a matrix of order (m x n) and its rank is 'r'. [r <= min(m,n)]
We can split the matrix A into two matrices through 'r' or even any value ranging from 1 to r.

$$A_{mxn} = B_{mxr}\, C^{T}_{rxn}$$

Example

Let m=No. of rows=5000
n=No. of columns=100
r=rank of A=20
($A_{5000x100}$)
Now Number of elements of A =5000x 100=**5,00,000**

We can find two matrices B and C^T such that
$B_{5000x20}$ and C^T_{20x100}

No. of elements of BC^T =5000x20+20x100=**1,02,000**

$$\boxed{A_{5000x100} = B_{5000x20}\, C^{T}_{20x100}}$$

So instead of storing the original matrix A with 5,00,000 elements we can store B and C^T matrix with 1,02,000 elements i.e. almost **80%** reduction in storage space required !

Low Rank Adaptation(LoRA)

Want further reduction in the storage space requirement?

Choose any lower value between 1 and 20 (i.e. lower rank of A , Full rank is 20)
Lets choose a low rank r=10

Now we can write $\mathbf{A}_{5000\times100} = \underbrace{\mathbf{B}_{5000\times10}\ \mathbf{C}^{T}_{10\times100}}_{}$

No. of elements=5000x10+10x100=**51,000**

So instead of storing the original matrix A with 5,00,000 elements we can store B and C^T matrix with 51,000 elements i.e. almost 90% reduction in storage space required !

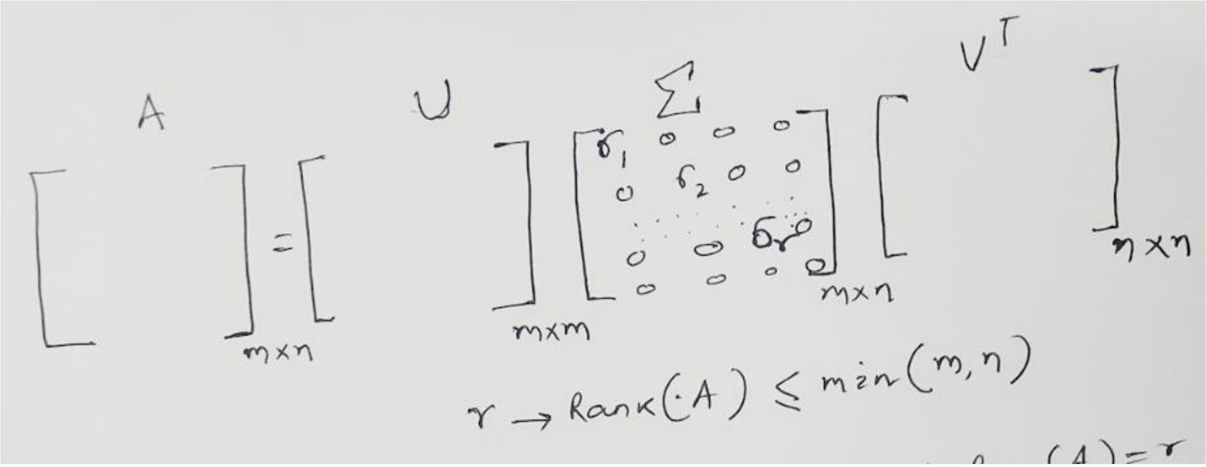
- **This is called Low rank approximation of A, because we are using a low rank of 10, instead of its full rank of 20**
- **So we can represent the same matrix A with much less number of elements which is very close to the original matrix**

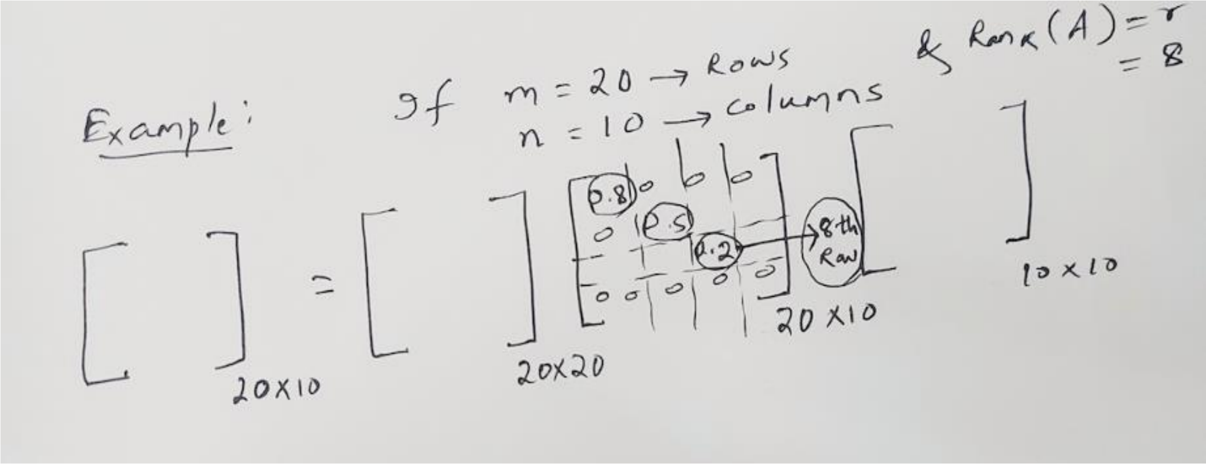
Low Rank Adaptation(LoRA)

Q:How to find B and C^T ?

A: By Singular Value Decomposition(SVD)

Any matrix of order (mxn)can be written as a matrix multiplication of 3 matrices as follows:
$A=U\Sigma V^T$ and rank of A is ‘r’
Where:
U is a mxm matrix
V is nxn matrix
Σ is a mxn matrix, the diagonal values of first ‘r’ rows are singular values of A and rest of the entries are zero.





Note: SVD can be done by any computer programing language such as python

Low Rank Adaptation(LoRA)

Example

Let A= $\begin{bmatrix} 3 & 2 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$

Here rank of A is 3 as there are three independent columns.
Let's do a low rank approximation say 2 rank approximation

Applying SVD(Here I have used python to find out U,Σ and V^T matrices), A can be split as follows:

$$\begin{bmatrix} 3 & 2 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0.91 & 0.42 & 0.02 \\ 0.41 & -0.87 & -0.26 \\ 0.09 & -0.24 & 0.97 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 4.04 & 0 & 0 \\ 0 & 1.70 & 0 \\ 0 & 0 & 0.87 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} 0.67 & 0.73 & 0.08 \\ 0.65 & -0.54 & -0.53 \\ 0.35 & -0.41 & 0.84 \end{bmatrix}^T}_{V^T}$$

LowRankAdaptation(LoRA)

Example

Rank 2 approximation of A:

$$\begin{bmatrix} 0.91 & 0.42 & 0.02 \\ 0.41 & -0.87 & -0.26 \\ 0.09 & -0.24 & 0.97 \end{bmatrix} \begin{bmatrix} 4.04 & 0 & 0 \\ 0 & 1.70 & 0 \\ 0 & 0 & 0.87 \end{bmatrix} \begin{bmatrix} 0.67 & 0.73 & 0.08 \\ 0.65 & -0.54 & -0.53 \\ 0.35 & -0.41 & 0.84 \end{bmatrix}^T$$

3×2 $\quad$ 2×2 $\quad$ 2×3

$$= \underbrace{\begin{bmatrix} 0.91 & 0.42 \\ 0.41 & -0.87 \\ 0.09 & -0.24 \end{bmatrix} \begin{bmatrix} 4.04 & 0 \\ 0 & 1.70 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} 0.67 & 0.73 & 0.08 \\ 0.65 & -0.54 & -0.53 \end{bmatrix}^T}_{C^T}$$

$$= \begin{bmatrix} 2.99 & 2.01 & 0.98 \\ 0.02 & 1.88 & 1.1 \\ -0.07 & 0.45 & 0.29 \end{bmatrix} \approx \begin{bmatrix} 3 & 2 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$ Original Matrix