Week 4: Determinants and Eigenvectors

As seen, the grid after linear transformation we got another 2D grid, because the transformation happened on a non-singular matrix. Linear transformation on singular matrices gives lesser dimensions than the original grid's. The number of dimensions in the output grid represent the matrice's ranks.

For example:

→ Redundunt (2X2) matrix gets linearly transformed into a line (1D), which coresponds to Rank 1.

$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ → Zero (2X2) matrix gets linearly transformed into a point (0D), which corresponds to Rank 0.

The absolute determinant represent the area of the basis in the linearly tranformed grid. This corresponds to the two mentioned examples; determinant of the singular matrices is equal to zero (area of line equal zero & area of a point equal to zero).

In case of matrix multiplication, the determinant of the product matrix is the product of the two determinants.

$$\det(A \cdot B) = \det(A) \cdot \det(B)$$

The determinant of an inversed matrix equals to the multiplicative inverse on the original matrix determinant.

$$\det(A^{-1}) = \frac{1}{\det(A)}$$

The determinant of the identity matrix is always 1.

$$\det(I) = 1$$

A basis is any two vector that are not multiples of each other in a grid. A vector in a basis and its opposite (negative) vector are considered the same vector.

A basis can be used to reach any point on a grid, by multiplying any of the two vectors of the basis by a constant (can be decimal) and add it to the other vector multiplied by a constant.

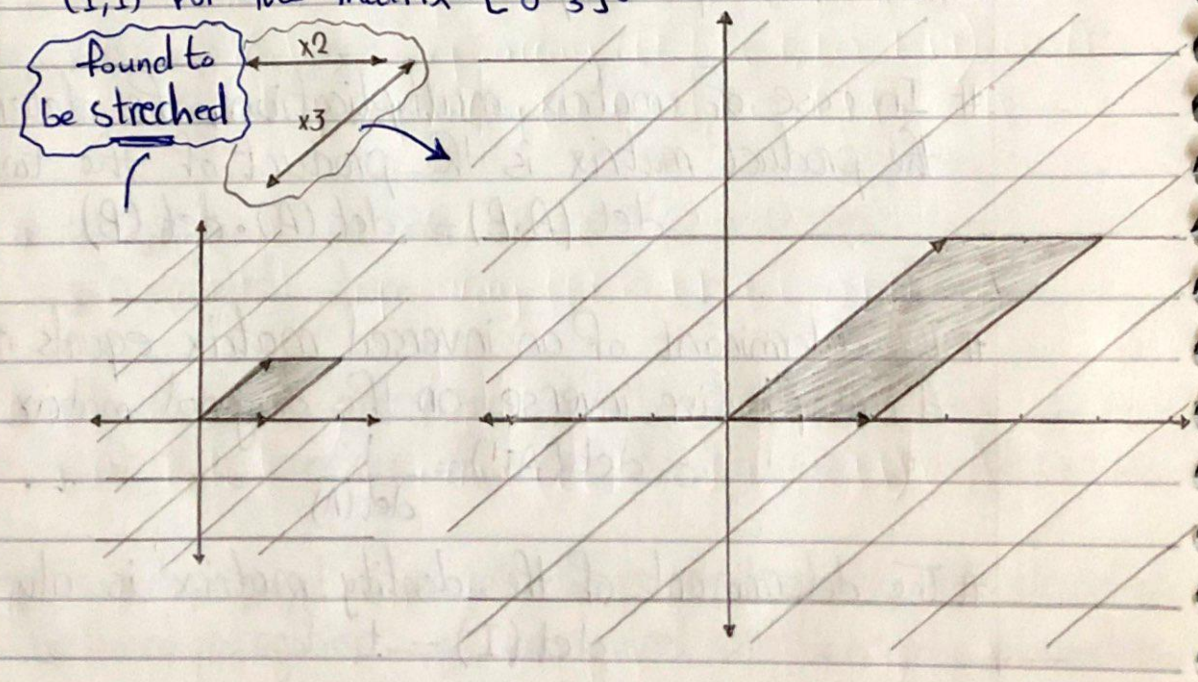
Two vector which are opposite or multiples of each other can't be considered a basis.

→ A basis for 2D grid is two vectors. (can't be more)
→ A basis for 1D line is one vector. (can't be 2)

A basis is a minimal spanning set.

Eigenbasis is the most important basis in machine learning applications.

Let's change the basis from (0,1) & (1,0) to (1,0) & (1,1) for the matrix $\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$:

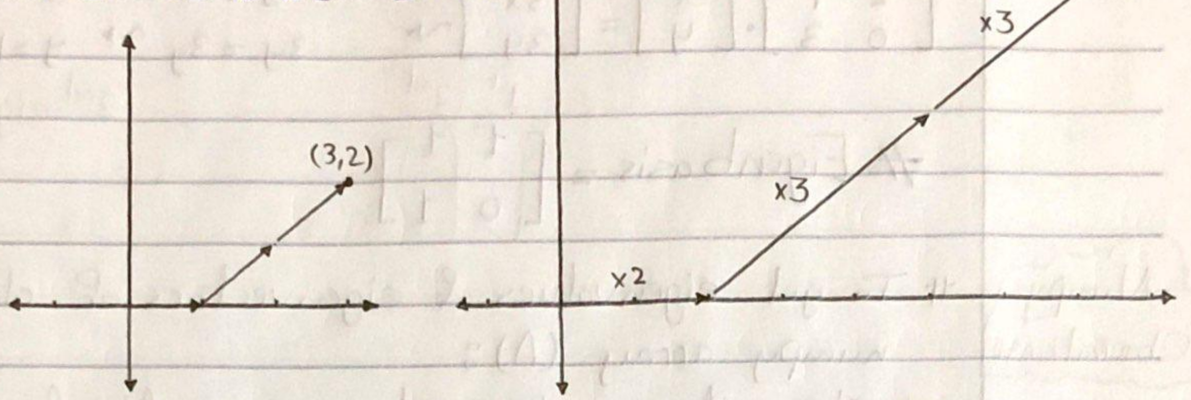


very important

since the basis vectors and the transformed vectors are parallel, but stretched, it's called Eigenbasis: → two basis vectors are called Eigenvectors
→ streching factors are called Eigenvalues.

In the previous example, let's consider the point (3,2) for the same matrix $\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$:

$$\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$



Eigenvectors: (0,1) & (1,1)
Eigenvalues: 2 & 3

Eigenvalue of an inversed matrix is equal to the multiplicative inverse of the eigenvalue of the original matrix:
→ If eigen-value of matrix A is i, then $\frac{1}{i}$ is the eigenvalue of matrix A^{-1}.

let (x,y) represent the coordinates of any point on a specific straight line

If λ is an eigen-value for the matrix $\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$, then:

$$\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$; for infinitely many (x,y)

Then: $$\begin{bmatrix} (2-\lambda) & 1 \\ 0 & (3-\lambda) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$; has infinitely many solutions (redundant singular)

Therefore: $$\det\left(\begin{bmatrix} (2-\lambda) & 1 \\ 0 & (3-\lambda) \end{bmatrix}\right) = 0$$

very important

$(2-\lambda)(3-\lambda) - 1 \cdot 0 = 0$ → Characteristic Polynomial solving the characteristic polynomial, gives the eigenvalues:

$\lambda = 2$ & $\lambda = 3$

then substitute in $\begin{bmatrix} a & b \\ 6 & d \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \end{bmatrix}$ and solve for eigenvectors:

$\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix} \rightsquigarrow \begin{matrix} 2x + y = 2x \\ 3y = 2y \end{matrix} \rightsquigarrow \begin{matrix} x = 1 \\ y = 0 \end{matrix} \rightsquigarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

1st eigenvector

$\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3x \\ 3y \end{bmatrix} \rightsquigarrow \begin{matrix} 2x + y = 3x \\ 3y = 3y \end{matrix} \rightsquigarrow \begin{matrix} x = 1 \\ y = 1 \end{matrix} \rightsquigarrow \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

2nd eigenvector

$$\text{\# Eigenbasis} = \begin{array}{cc} 1^{st} & 2^{nd} \\ \end{array} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Numpy Observations:

To get eigenvalues & eigenvectors of already defined numpy array (A):
→ eigenvalues, eigenvectors = np.linalg.eig(A)

To get the reflection about y-axis & x-axis:
→ A_reflected_yaxis = np.array([[-1,0],[0,1]]) @ A
→ A_reflected_xaxis = np.array([[1,0],[0,-1]]) @ A

To get it sheared positive constant c in x & in y:
→ A_sheared_x = np.array([[1, c],[0, 1]]) @ A
→ A_sheared_y = np.array([[1, 0],[c, 1]]) @ A

To rotate A 90° clockwise:
→ A_rotated = np.array([[0,1],[-1,0]]) @ A

To scale A by value k:
→ A_scaled = np.array([[k, 0],[0, k]]) @ A

To get the projection of A on x-axis:
→ A_projected_x = np.array([[1,0],[0,0]]) @ A