Artificial Intelligence Algorithms and Mathematics

CSCN 8000

Motivation

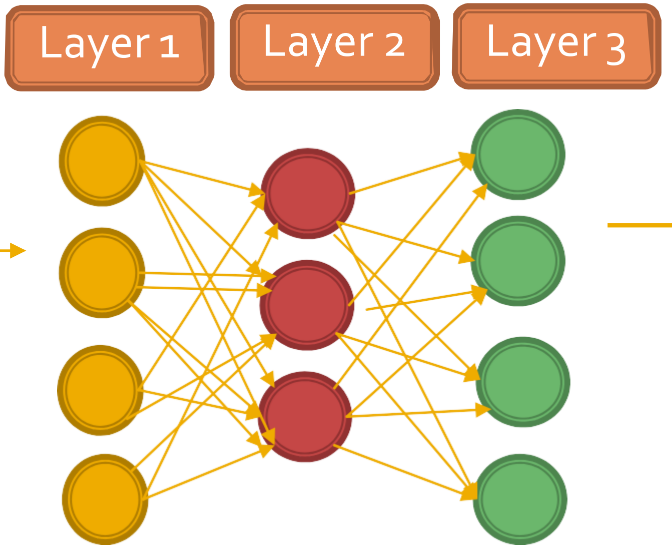
- Vectors, Matrices, Gradients, Optimization, and Probability Distribution occurs a lot in Machine Learning.
- How and why do machine learning algorithms work?
- To build customized machine learning models.
- Debugging the machine learning algorithms.
- Understanding the maths will help you to improve the performance of the machine learning algorithms.

Linear Algebra

- Vectors
- Matrix
  - Inverse
  - Rank
- Linear Transformation
- Eigenvalues
- Eigenvectors
- These are used in machine learning to store and compute on data.

Motivation to Machine Learning

- Linear Algebra most useful in machine learning
- Most popular application –Neural Networks –image recognition –using matrix operations.



Output Prediction

It is a Cat

Pixels- Matrix

Node-Matrix

Node-Matrix

Node -Matrix

Prediction -Matrix

Linear Algebra

Motivation to Machine Learning

- To perform all the vector or matrix operations to go from the image on the left to the prediction on the right, we use linear algebra.
- The pixels in the image are simply values that serve as an input where the algorithm(neural networks) performs mathematical operations to determine the output prediction.

Vector

- A vector is a quantity defined by a magnitude and a direction. For example, a rocket's velocity is a 3-dimensional vector: its magnitude is the rocket’s speed, and its direction is (hopefully) up.
- A vector can be represented by an array of numbers called *scalars*.
- Each scalar corresponds to the magnitude of the vector with regard to each dimension.
- For example, say the rocket is going up at a slight angle: it has a vertical speed of 5,000 m/s, and also a slight speed towards the East at 10 m/s, and a slight speed towards the North at 50 m/s.

$$\textbf{velocity} = \begin{pmatrix} 10 \\ 50 \\ 5000 \end{pmatrix}$$

Purpose

- To represent observations and predictions.
- For example, say we built a Machine Learning system to classify videos into 3 categories (good, spam, clickbait) based on what we know about them.
- For each video, we would have a vector representing what we know about it, such as:

$$\text{video} = \begin{pmatrix} 10.5 \\ 5.2 \\ 3.25 \\ 7.0 \end{pmatrix}$$

This vector could represent a video that lasts 10.5 minutes, but only 5.2% viewers watch for more than a minute, it gets 3.25 views per day on average, and it was flagged 7 times as spam

Purpose

- Based on this vector our Machine Learning system may predict that there is an 80% probability that it is a spam video, 18% that it is clickbait, and 2% that it is a good video.
- This could be represented as the following vector:

$$\text{class\_probabilities} = \begin{pmatrix} 0.80 \\ 0.18 \\ 0.02 \end{pmatrix}$$

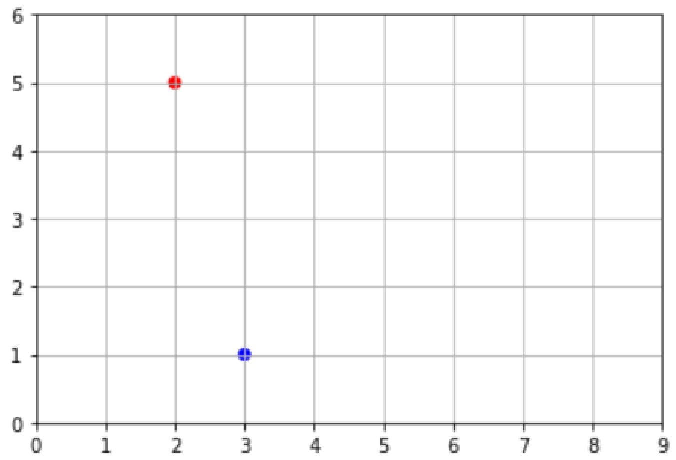
Vectors in python

- In python, a vector can be represented in many ways
  - import numpy as np
  - video = np.array([10.5, 5.2, 3.25, 7.0])
  - Video

Output : array([10.5 , 5.2 , 3.25, 7.])

Plotting vectors

- To plot vectors, we will use matplotlib, so let's start by importing
- import matplotlib.pyplot as plt
- Let's create a couple of very simple 2D vectors to plot:
  - u = np.array([2, 5])
  - v = np.array([3, 1])
- These vectors each have 2 elements, so they can easily be represented graphically on a 2D graph, for example as points:
  - x_coords, y_coords = zip(u, v)
  - plt.scatter(x_coords, y_coords, color=["r","b"])
  - plt.axis([0, 9, 0, 6])
  - plt.grid()
  - plt.show()



Arrays

- The array object in NumPy is called **ndarray** meaning 'n-dimensional array’.
- To begin with, you will use one of the most common array types: the one-dimensional array (‘1-D’).
- A 1-D array represents a standard list of values entirely in one dimension.
- **Remember that in NumPy, all of the elements within the array are of the same type.**

Arrays -Examples

- Create and print a NumPy array 'a' containing the elements 1, 2, 3.
  - a = np.array([1, 2, 3])
  - print(a)

Output: [1 2 3]

- Create an array with 3 integers, starting from the default integer 0.
  - b = np.arange(3)
  - print(b)

Output: [0 1 2]

- NumPy function np.linspace is a floating point (np.float64).
  - l= np.linspace(0, 100, 5)
  - print(l)

Output: [0. 25. 50. 75. 100.]

- Converting and displaying as int
- l_int= np.linspace(0, 100, 5, dtype=int)
- print(l_int)

Output: [0 25 50 75 100]

Arrays

- One of the advantages of using NumPy is that you can easily create arrays with built-in functions such as:
  - np.ones() - Returns a new array setting values to one.
  - np.zeros() - Returns a new array setting values to zero.
  - np.empty() - Returns a new uninitialized array.
  - np.random.rand() - Returns a new array with values chosen at random.

Vector Operations

- Vector Addition:

$$\begin{bmatrix}1\\2\\3\end{bmatrix} + \begin{bmatrix}1\\2\\3\end{bmatrix} = \begin{bmatrix}2\\4\\6\end{bmatrix}$$

- Vector Multiplication:

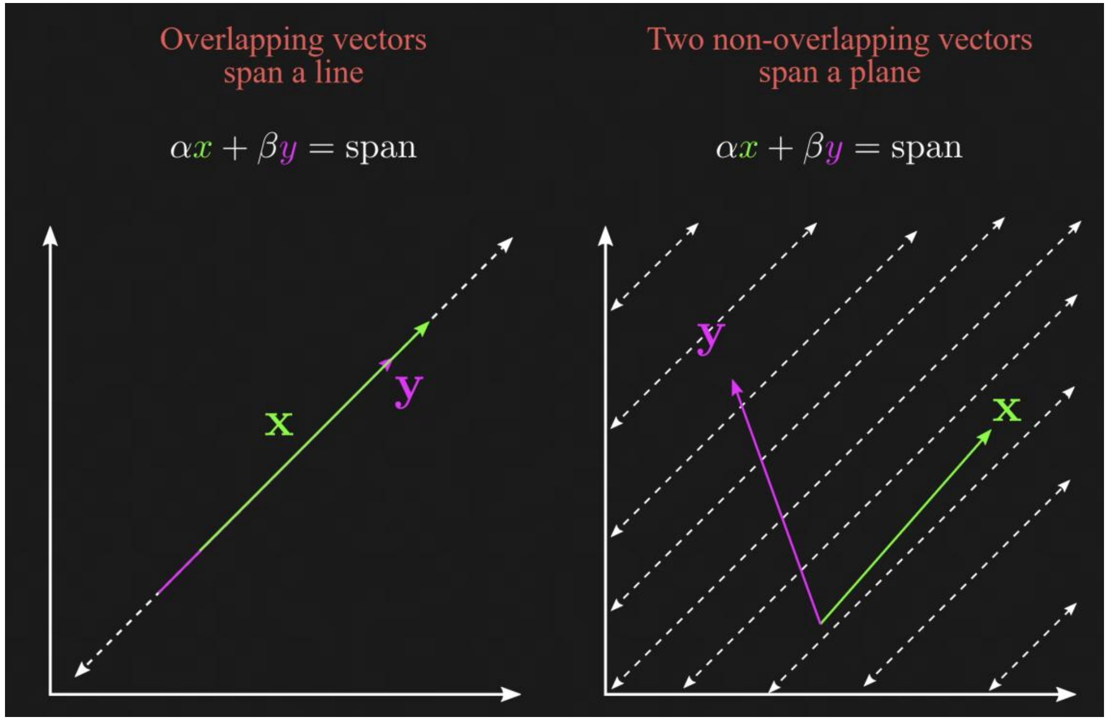
$$\begin{bmatrix}1\\2\\3\end{bmatrix} * \begin{bmatrix}1\\2\\3\end{bmatrix} = \begin{bmatrix}1\\4\\9\end{bmatrix}$$

- Vector Dot Product:

$$\begin{bmatrix}1 & 2 & 3\end{bmatrix}.\begin{bmatrix}1\\2\\3\end{bmatrix} = (1 * 1) + (2 * 2) + (3 * 3) = 14$$

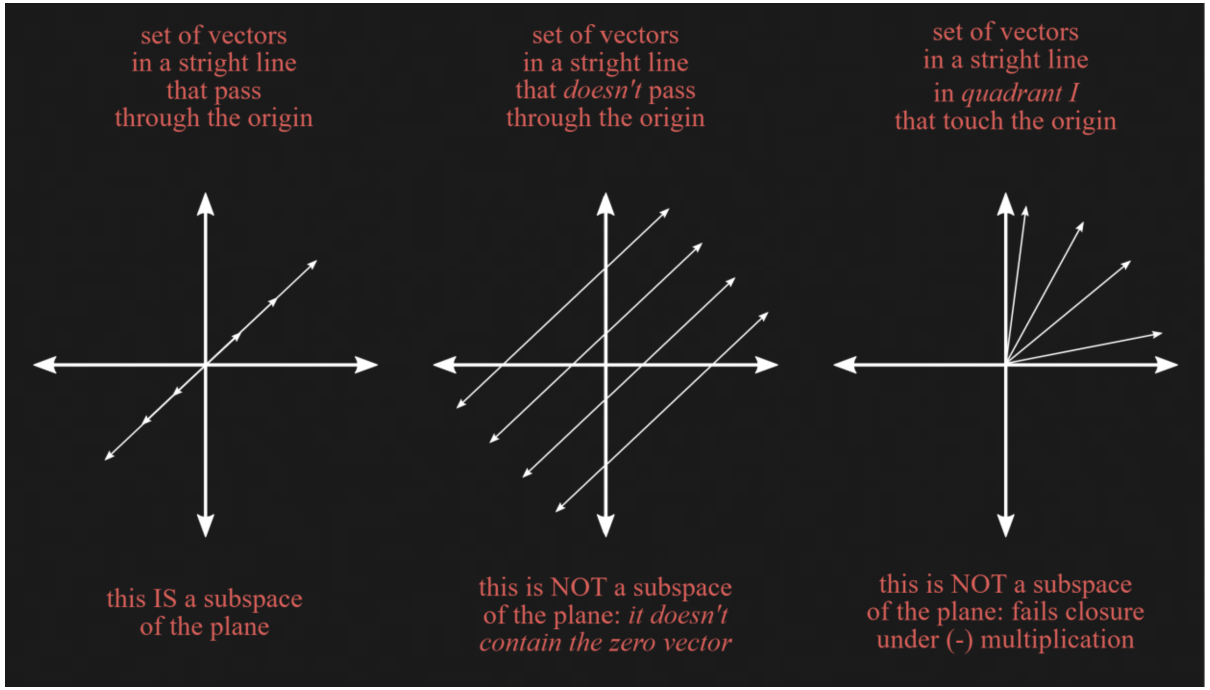
Vector Space & Span

- Vector Space:
  - In its more general form, a vector space, also known as linear space, is a collection of objects that follow the rules defined for vectors in R^n.
- Vector Span:
  - Consider the vectors $x \; and \; y$ and the scalars α and β. If we take all possible linear combinations of $\alpha x + \beta y$ we would obtain the ***span*** of such vectors.



Vector Subspace

- Vector Subspace:
  - A vector subspace is a vector space that lies within a larger vector space. These are also known as linear subspaces.
  - Consider a subspace S. For a vector to be in the valid subspace it has to meet three conditions:
    - Contains the zero vector, $\mathbf{0} \in S$
    - Closure under multiplication, $\forall \alpha \in R \rightarrow \alpha \times s_i \in S$.
    - Closure under addition, $\forall s_i \in S \rightarrow s_1 + s_2 \in S$.



Is the span of $x = [1\ 1]$ a valid subspace of R^2?

Basis of Vector Space

- The basis of a vector space is a set of vectors that satisfies two critical criteria:
  - **Linear Independence:** The vectors in the basis set are linearly independent. This means that no vector in the basis set can be written as a linear combination of the others.
  - **Spanning the Space:** The basis set of vectors spans the vector space. This means that any vector in the vector space can be expressed as a linear combination of the vectors in the basis set.
- The number of vectors in a basis set of a vector space is equal to the number of dimensions.
- What is the basis set of R^3?

Matrices

- With matrices, we can represent sets of variables. In this sense, a matrix is simply an ordered collection of vectors.
- Matrix-Matrix Addition:

$$A + B = \begin{bmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{bmatrix}$$

- Matrix-Matrix Dot Product:
  - $A \in \mathbb{R}^{m\times \boldsymbol{n}}, B \in \mathbb{R}^{\boldsymbol{n}\times p}$
  - $A.B = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} . \begin{bmatrix} b_{11} & \cdots & b_{1p} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{np} \end{bmatrix} = \begin{bmatrix} c_{11} & \cdots & c_{1p} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mp} \end{bmatrix}$
  - $c_{ij} = \sum_{l=1}^{n} a_{il} b_{lj} \;\; where\ i = 1, \dots, m\ , and, j = 1, \dots, p$

The Determinant

- The determinant is a scalar value that can be computed from the elements of a square matrix. It is denoted as $\det(A)$ or $|A|$ for a matrix A.
- Consider Matrix: $\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$
- $Determinant = a.d - b.c = (1 * 4) - (3 * 2) = -2$
- Applications:
  - **Geometry:** Used in transformations, rotations, and scaling of geometric figures.
  - **Physics and Engineering:** In solving systems of linear equations which frequently arise in physics and engineering problems.
  - **Computer Graphics:** In transformations and animations.
  - **Economics:** In solving linear models and systems of equations in economic analysis.

Matrices in python

- In python, a matrix can be represented in various ways. The simplest is just a list of python lists:
  - [10, 20, 30],
  - [40, 50, 60]

[[10, 20, 30], [40, 50, 60]]

Matrices

- Use the NumPy library which provides optimized implementations of many matrix operations:
  - A = np.array([[10,20,30],[40,50,60]])
  - A

```
array([[10, 20, 30],
       [40, 50, 60]])
```

Size

- The size of a matrix is defined by its number of rows and number of columns.
- For example, consider a2×3 matrix: 2 rows, 3 columns. Caution: a 3×2 matrix would have 3 rows and 2 columns.
- To get a matrix's size in NumPy:
  - A.shape

    (2, 3)

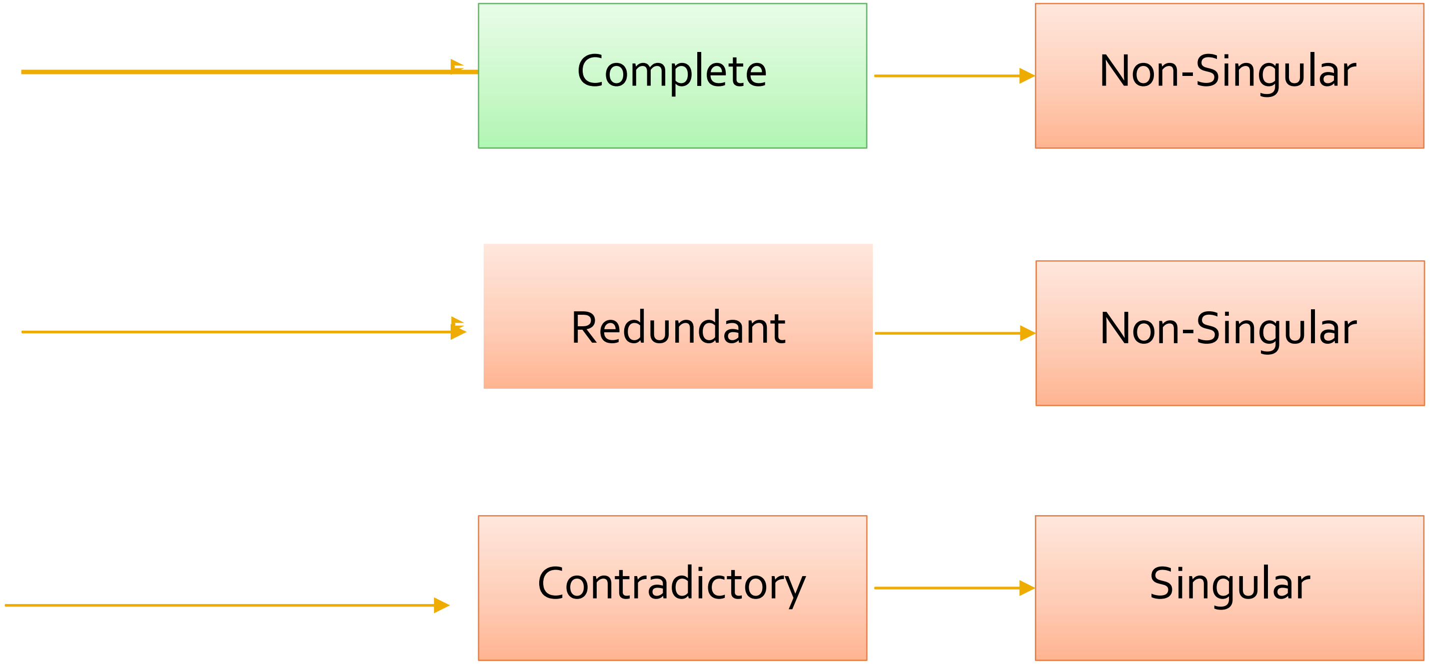
  - Caution: the size attribute represents the number of elements in the ndarray, not the matrix's size:

  - A.size

    6

Introduction to System of Linear Equations

- System 1
  - The apple is red
  - The pear is green
- System 2
  - The apple is red
  - The apple is red
- System 2
  - The apple is red
  - The apple is green



System of Equations

- The system of equations can also be singular and non-singular, very similar to sentences.

System of Linear Equations

- Linear Equations
  - $x + y = 20$
  - $x + 2y = 22$

- Non Linear Equations-complicated
  - $x^2 + y^2 = 20$
  - $x + 2y = 22$
  - $\sin(a) + 2y = 35$
  - $xy + \log(z) = 5$

System of Equations as Lines , planes

- Plotting the different solutions as the coordinates visually on the graph.
- Example: Consider the linear equation $3x + 4y = 10$, $2x - 2y = 2$;
- Since the points cross at a unique solution, it is said to be nonsingular.
- If there are infinite solutions or no solutions then it is said to be singular.

System of Linear Equations as Matrices

- Coefficients of the Linear Equations are considered elements of the matrix.
- Example : $x+y = 0$ $\quad$ $x+2y=0$ $\quad$ matrix = $\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$
- Example : $x+y = 0$ $\quad$ $2x+2y=0$ $\quad$ matrix = $\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$

- Matrices can also be Singular and Non Singular

Singular Vs Nonsingular Matrix

- **Linear dependence** between rows – Second row is multiple of the first row. Here they are linearly dependent
  - For Example : x+y = 0

    2x+2y=0

1	1	First Row
2	2	Second Row/multiple of the first row

- Otherwise, they are referred to be **linearly independent**
  - For Example : x+y = 0

    x+2y=0

1	1	First Row
1	2	Second Row/not a multiple of the first row

Singular vs Non-Singular Matrix

- To determine whether a system if linear equations is singular vs non-singular, one could use ***The Determinant:***
  - If the determinant of the matrix of the system is equal to zero → **Singular System**
  - If the determinant of the matrix of the system is not equal to zero → **Non-Singular System**

Solving Systems of Linear Equations using Matrices

- NumPy linear algebra package provides a quick and reliable way to solve the system of linear equations using the function **np.linalg.solve(A, b).**
  - A is a matrix, each row of which represents one equation in the system and each column corresponds to the variable $x1$, $x2$
  - b is a 1-D array of the free (right side) coefficients.

- More about the Package:https://numpy.org/doc/stable/reference/generated/numpy.linalg.solve.html

References

- https://learning.oreilly.com/library/view/hands-on-machine-learning/9781098125967/ch01.html#what_is_machine_learning

Thank you!

- Any questions?

Thank You

Youssef Abdelkareem

yabdelkareem@conestogac.on.ca

**** Credits for Dr. Anasuya Bhima for assisting in the slides content***