5CS037 – Concepts and Technology of AI.
Lecture – 01
Review of Mathematics for Machine Learning.
Linear Algebra for Machine Learning.

Siman Giri {Module Leader – 5CS037}

Learning Outcome:

- Review and revise some fundamental concepts from mathematics we will be using through out the course.
- We will discuss:
  - **Why do we need Linear Algebra for Machine Learning?**
    - {**Almost**} Everything you need to know about **vector and matrices** for Machine Learning.
  - **A very big picture on Definition of Derivative.**

A. Preliminaries – Set Theory.

{Intuitive Set Theory and Set Algebra}

A.1 Sets

- A **set** is well defined **collection of {unique or distinct} objects**:
  - Examples: A pack of wolves, A deck of cards, A flock of Pigeons.
- Sets are denoted by **capital letters** such as **A or X** and when listing the elements of a set, curly brackets {.} are used:
  - Sets have **elements** or **members**!!
    - $x \in A$**(read: x is an element of A or x belongs to A).**
      - Elements/ members: Objects that belongs to set.
      - **Caution!!!** Set it self can be member or elements of other sets.
- **Sets Notations:**
  - There are several different notations for defining sets; the three most common ones are listed below:
    - **Verbal notation**: Let **X** be the set of letters in the word **"English".**
    - **Roster notation**: $\mathbf{X = \{x_1, \ldots, x_n\}}$ [List all the elements of the set.]
    - **Set – Builder notation**: $\mathbf{X = \{x : x \text{ is an even integer and } 0 \leq x \leq 6\} \equiv \{0, 2, 4, 6\}}$
      [we will follow this notation]

A.2 Sets: Algebra.

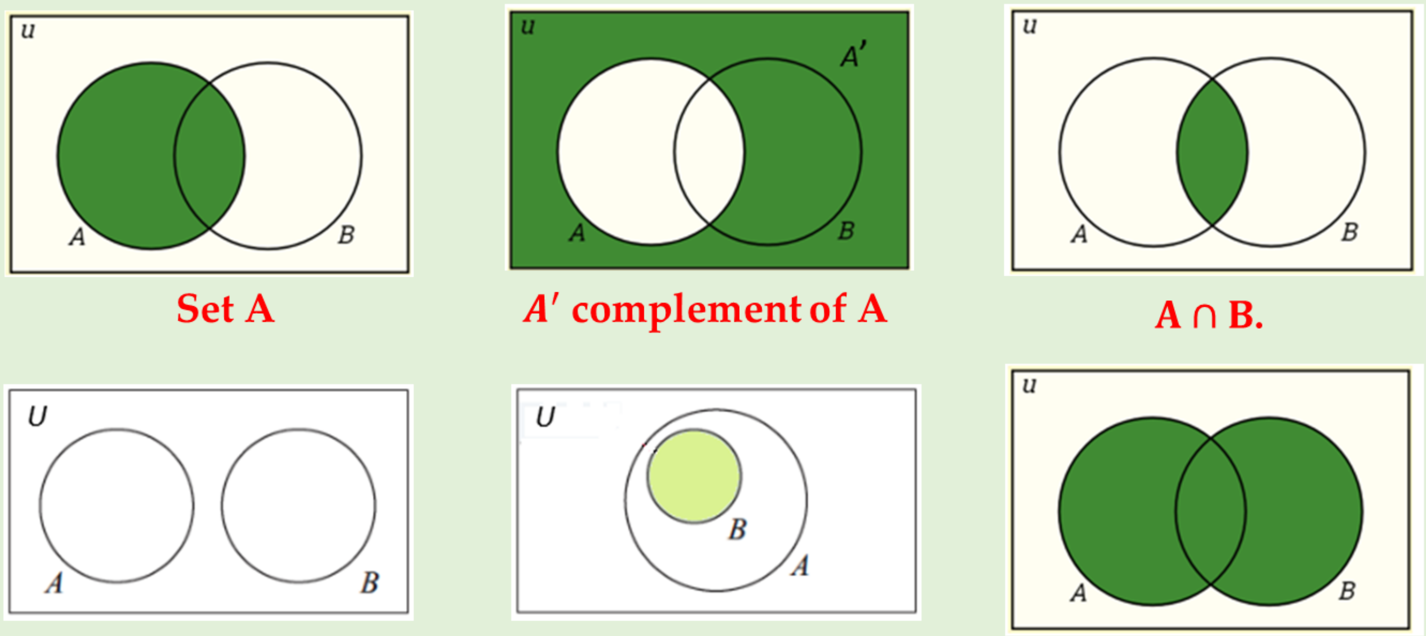
- **Some of the relations**:
  - **Set A is subset of B**: every element of A is also an element of B, {aka A is contained in B}.
    - **Notations:** $\boldsymbol{A \subseteq B}$**.**
      Example: $\{\mathbf{4,5,8}\} \subseteq \{\mathbf{2,3,4,5,6,7,8,9}\}$
  - **Set A and Set B are equal**: every element of B is in A.
    - **Notations:** $\boldsymbol{A = B}$**.**
  - **Empty sets**: Set with no elements are called empty sets and is denoted by **ϕ**.
  - **Universal Set (denotes with U)** : It is a set which has elements of all the related sets, without any repetition of elements.
- Some **example of important and universally accepted sets** are:

$$\mathbb{N} = \{n: n \; is \; a \; natural \; numbers\} = \{1,2,3 \ldots\}$$
$$\mathbb{Z} = \{n: n \; is \; an \; integer\} = \{-1,0,1, \ldots\}$$
$$\mathbb{R} = \{x: x \; is \; a \; real \; number\}$$
$$\mathbb{C} = \{z: z \; is \; a \; complex \; number\}.$$

- **Some Set Operations For two sets A and B:**
- **Intersections:**
  - The intersection $\{\mathbf{A \cap B}\}$ is the set of all elements contained in both A and B.
    - $\mathbf{A \cap B = \{x: x \in A \text{ and } x \in B\}}$
- **Union:**
  - The union $\{\mathbf{A \cup B}\}$ is the set of all elements either in A or B.
    - $\mathbf{A \cup B = \{x: x \in A \text{ or } x \in B\}}$
- **Disjoint:**
  - Set A and B are disjoint set if they do not have any element in common i.e. $\mathbf{A \cap B = \emptyset}$
- **Complement of A:**
  - For all $\mathbf{A \subset U}$: **U universal set**. Complement of A is :
    - $\mathbf{A' = \{x: x \in U \text{ and } x \notin A\}.}$
- **Difference of two sets A and B is:**
  - $\mathbf{A \backslash B = A \cap B' = \{x: x \in A \text{ and } x \notin B\}.}$

A.3 Sets: Venn Diagrams.

- Venn diagrams are very useful tool for **visualizing** various set operations.



Set A

A' complement of A

A ∩ B.

A and B are Disjoint.

$B \subseteq A$.

A ∪ B.

Fig: Sets and Venn Diagrams

A.4 Sets: Cartesian Products.

- Given sets A and B, we can define a new set $A \times B$ **(called Cartesian product of A and B)** are a set of **ordered pairs** i.e. $\mathbf{A \times B = \{(a, b) : a \in A \text{ and } b \in B\}}$

Example:
If $A = \{x, y\}, B = \{1, 2, 3\}, and\ C = \emptyset$, then
$A \times B = \{(x, 1), (x, 2), (x, 3), (y, 1), (y, 2), (y, 3)\}$.
and,

$$A \times C = \emptyset$$

- Cartesian product of **n sets** can be defined as:
  - $\mathbf{A_1 \times \cdots \times A_n = \{(a_1, \dots, a_n) : a_i \in A_i \text{ for } i = 1, \dots, n\}}$.

A.5 Cartesian Products: Mappings.

- Mappings a.k.a **relations** or **function** are the **subsets** of **Cartesian products** i.e. for Cartesian set $\mathbf{A \times B}$ function $\mathbf{f}$ is
    - $\mathbf{f \subset A \times B}$.
    - This represents a special type of relation where $(\mathbf{a}, \mathbf{b}) \in \mathbf{f}$ if every element $\mathbf{a} \in \mathbf{A}$ there exists a **unique element** $\boldsymbol{b} \in \boldsymbol{B}$. (**for every element in A, f assigns a unique element in B**).
    - Notations: **for functions**: $\mathbf{f}: \mathbf{A} \to \mathbf{B}$ and for **ordered pairs**
        - $(\mathbf{a}, \mathbf{b}) \in \mathbf{A \times B}; \mathbf{f}(\mathbf{a}) = \mathbf{b}$ **or** $\mathbf{f}: \mathbf{a} \to \mathbf{b}$.
- The set **A** is called the **domain** of $\mathbf{f}$ and $\mathbf{f(A)} = \{\mathbf{f(a)}: \mathbf{a} \in \mathbf{A}\} \subset \mathbf{B}$ is called the **range(co-domain)** of $\mathbf{f}$ **and its elements are called image under f.**
    - The elements of the function's domain as **input values** and the elements in the function's range as **output values**.

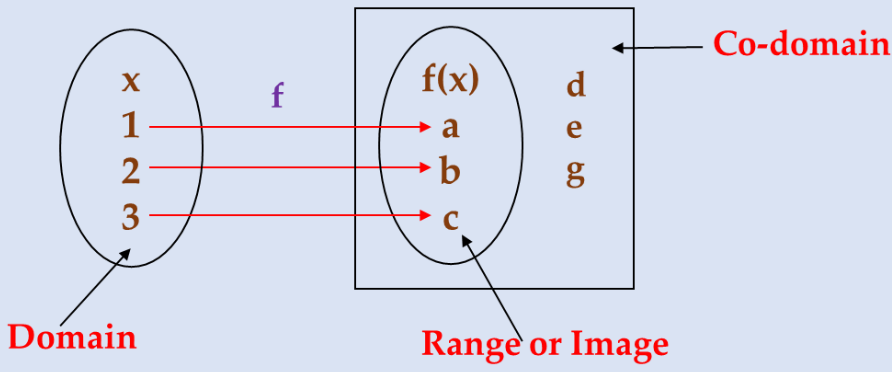


Fig: Concepts of Function.

A.5.1 A Valid Function

- Let **f : A → B** be a function from the domain *A* to the codomain *B*.
  - **well-defined**: A **relation/function** is well-defined if **each element** in the **domain** is assigned to a **unique element** in the **range**.
  - i.e. a function must always return the same value for specific input {**one-to-one**} or collection of input {**many-to-one**}.
- **Not a functions**:

  i) domain has **no image** associated with it.

  ii) one of the elements in the domain has **two images** assigned to it {**one-to-many**}.

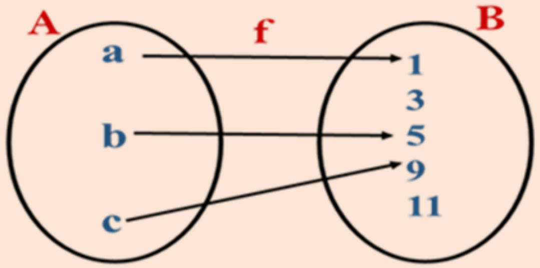


Fig: A well defined function.

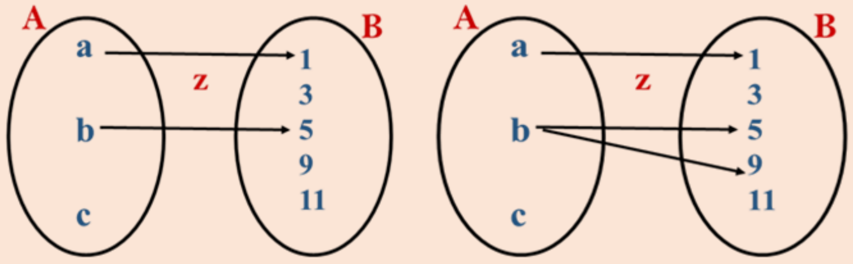


Fig: Not a Functions

A.5.2 Composite Function

- A **composite function** is a function that is created by **combining two functions**.
  - If you have **two functions**, say **fand g**,
    - then the composite function $(\mathbf{f} \circ \mathbf{g})(\mathbf{x})$ means you
      - first apply **function g** to the **input x** and
      - then apply **function f** to the result of **g(x).**
        - **Mathematically:** $(\mathbf{f} \circ \mathbf{g})(\mathbf{x}) = \mathbf{f}(\mathbf{g}(\mathbf{x}))$**.**
    - In general: $(\mathbf{f} \circ \mathbf{g})(\mathbf{x}) \neq (\mathbf{g} \circ \mathbf{f})(\mathbf{x})$

Example:
Suppose → $\mathbf{f}(\mathbf{x}) = \mathbf{2x + 3}$ **and** $\mathbf{g}(\mathbf{x}) = \mathbf{x}^2$
Then composite function would be:

$$(\mathbf{f} \circ \mathbf{g})(\mathbf{x}) = \mathbf{f}\big(\mathbf{g}(\mathbf{x})\big) = \mathbf{f}\big(\mathbf{x}^2\big) = \mathbf{2x^2 + 3}$$

Similarly:

$$(\mathbf{g} \circ \mathbf{f})(\mathbf{x}) = \mathbf{g}\big(\mathbf{f}(\mathbf{x})\big) = \mathbf{g}(\mathbf{2x + 3}) = (\mathbf{2x + 3})^2$$

A.5.3 Inverse Function

- The **inverse of a function** reverses the effect of the original function. If you have a **function f**, its inverse, denoted as $\mathbf{f^{-1}}$, will "undo" the action of **f.**
- Mathematically, if: $\mathbf{y = f(x)}$ then the **inverse function** $\mathbf{f^{-1}}$will satisfy: $\mathbf{x = f^{-1}(y)}$
- Key Properties:
  - If $\mathbf{f \circ f^{-1} = I}$ also $\mathbf{f^{-1} \circ f = I}$:
    - This means that **applying** f and then f^{-1} or vice versa will give you the original input x , where **I** represents the **identity function**.
  - $\mathbf{f^{-1}(x) \neq \frac{1}{f(x)}}$
- For a **function to have an inverse**, it must be both **one-to-one** and **on-to**. Essentially, **every output** of **f** must **correspond** to **exactly one input**.

A.5.3.1 Inverse Function : Example

- A good example of **inverse functions** are **exponential and logarithmic functions**:
  - **logarithms are the inverse of exponential functions and vice versa.**
    - we can say that:
      - the **domain of logarithmic functions** is **the range of exponential functions (0, ∞),**
      - and **the range of logarithmic functions** is **the domain of exponential functions (−∞, ∞):**

For instance:
Consider the following exponential function

$$\mathbf{f(x) = e^x}$$

Where its inverse is the logarithm defined as:

$$\mathbf{f^{-1}(x) = \log(x)}$$

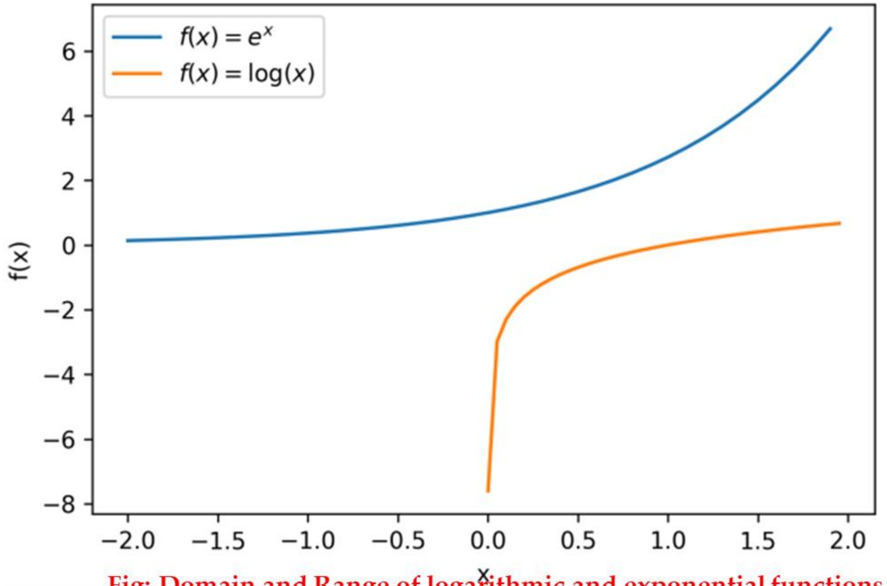


Fig: Domain and Range of logarithmic and exponential functions

A. Summary: Function and Machine Learning.

- The ultimate goal of **machine learning** is **learning** a **functions** from **data**, i.e. mappings from domain **(feature vector space(set))** onto the range **(target variable)** of a function.
- The **objective** of **5CS037** is to be able to **understand** all the highlighted **terms** in above **statement**.

B. Introduction to Linear Algebra.

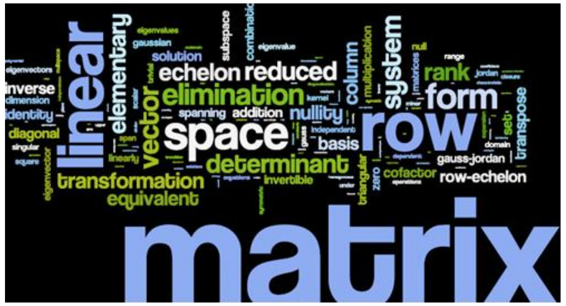
{Why to study Linear Algebra for Machine Learning?}

B.1 What is Linear Algebra?

- **Linear Algebra** is the branch of **mathematics** concerning **linear equations** such as:
  - $\mathbf{a_1x_1 + \ldots\ldots + a_nx_n = b}$;
  - **linear maps** such as:
    - $(\mathbf{x_1, \ldots\ldots, x_n}) \mapsto \mathbf{a_1x_1 + \cdots + a_nx_n}$;
  - and **there representations** in **vector spaces and through matrices**.

--Wikipedia.

- **Linear algebra** is a branch of mathematics that deals with **vectors, vector spaces** (also **known as linear spaces**),
  - and **linear transformations** between these spaces.
    - It involves operations on matrices and vectors, solving **systems of linear equations**, and understanding geometric concepts like **lines, planes, and subspaces**. – "chatgpt."



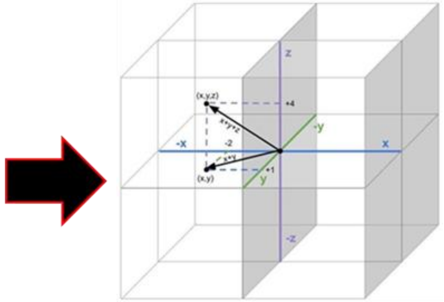


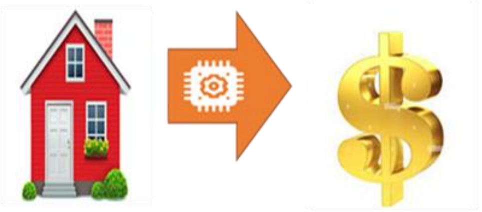
Fig: What is Linear Algebra?

Image: somewhere from web compiled by siman

B.2 Why Linear Algebra for Machine Learning?

- **Representation of Data:**
  - In machine learning, data is typically **represented** as **vectors** and **matrices**. For example, a dataset might be **stored as a matrix** where each row is a data point (vector), and each column is a feature.

Task: House Price Prediction.



Data: Features/Descriptor of House

Area	Rooms	Price
1080	8	1,00,000.00
1200	10	1,50,000.00

How would you represent this, for computer?

Matrix.

$$\begin{bmatrix} 1080 & 8 \\ 1200 & 10 \end{bmatrix} \begin{bmatrix} 1,00,000 \\ 1,50,000 \end{bmatrix}$$

B.2.1 Why Linear Algebra for Machine Learning?

- **Efficient Computing:**
  - Matrix operations allow for efficient computations on large datasets. Libraries like **NumPy**, **TensorFlow**, and **PyTorch** leverage **linear algebra** for operations on large matrices and tensors {**Vectorizations**}, which makes **machine learning models faster** and more **scalable**.

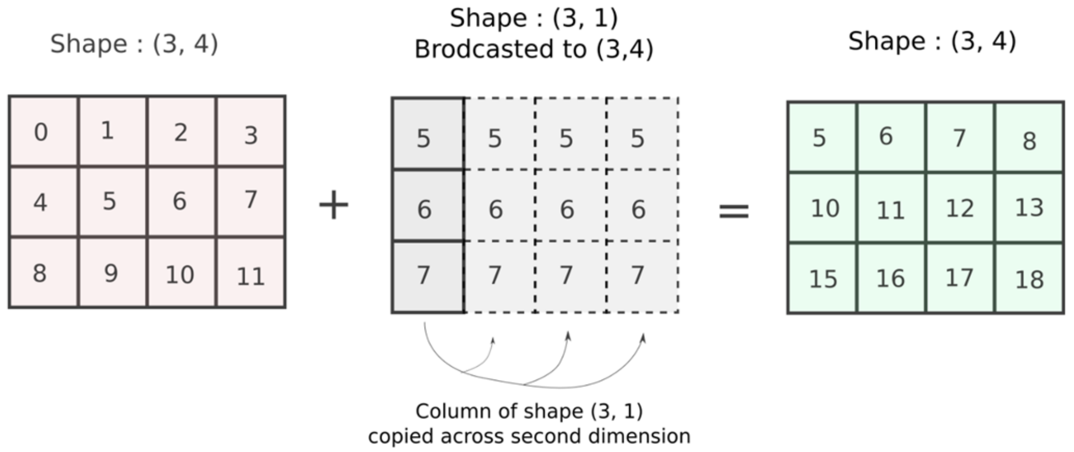
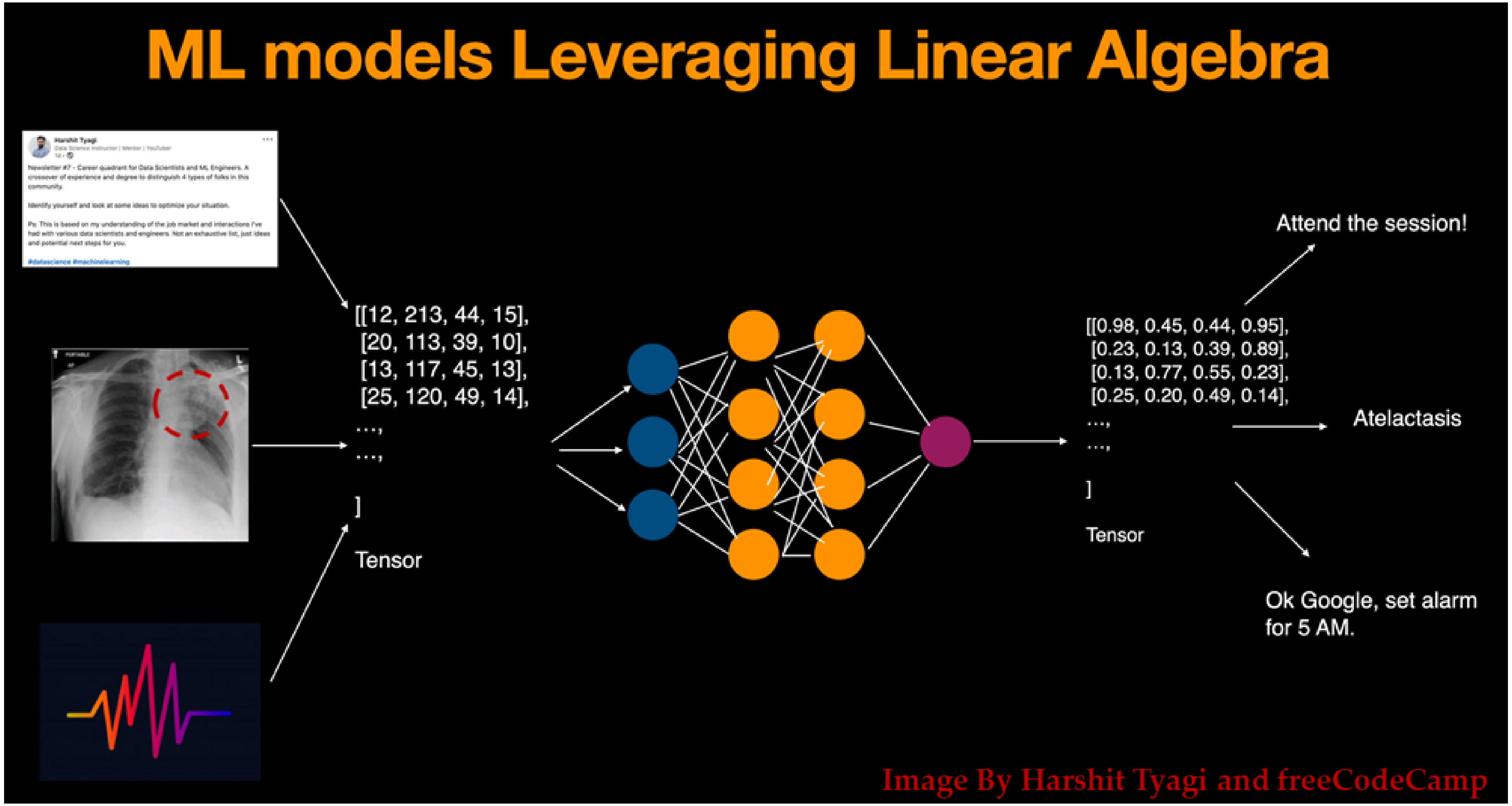


Fig: Idea of Vectorizatons.

B.2.2 Why Linear Algebra for Machine Learning?

- **Understanding {Machine Learning} Algorithms:**
  - Training **machine learning models** often involves **solving systems of linear equations**.
    - Linear algebra provides the **necessary tools** to solve these systems efficiently.
  - Many machine learning algorithms are based on linear algebra concepts.
  - For instance:
    - **Linear Regression** involves finding a line (or hyperplane) that best fits the data.
    - **Support Vector Machines (SVM)** utilize dot products to measure similarity.
    - **Neural Networks** use matrix multiplication for forward and backward propagation.

B. Summary : Linear Algebra for Machine Learning.



{Almost} Everything about Math for ML.

{1. Review → What are Vectors?}

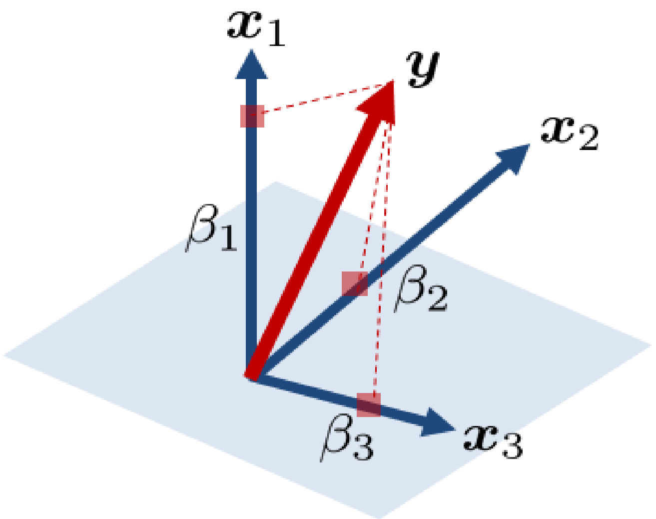


Image from Stanley Chan Book: Introduction to Probability for Data Science.

1.1 What are Vectors?

- **Interpretation – 1 – Direction in Space:**
  - E.g., the vector $\vec{\mathbf{v}} = [\mathbf{3}, \mathbf{2}]^{\mathbf{T}}$ has a **direction** of **3 steps to the right** and **2 steps up**
  - The **notation** $\vec{\mathbf{v}}$ is sometimes used to indicate that the **vectors have a direction**
  - All **vectors in the figure** have the **same direction**

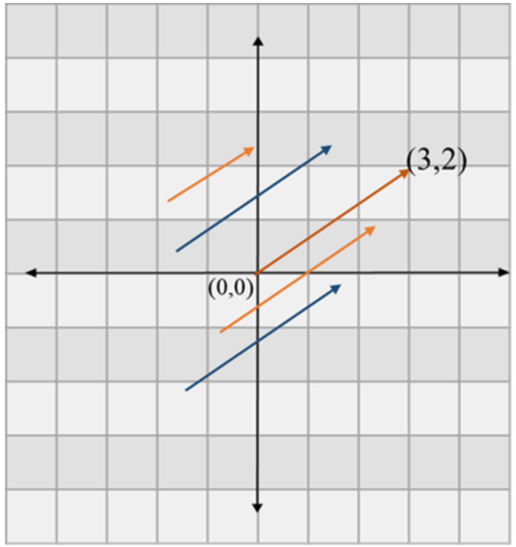


Fig: Vector as Direction

1.1.1 What are Vectors?

- **Interpretation – 2 – point in space:**
  - E.g., in 2D{dimension} we can visualize **the data points** with respect to a coordinate origin

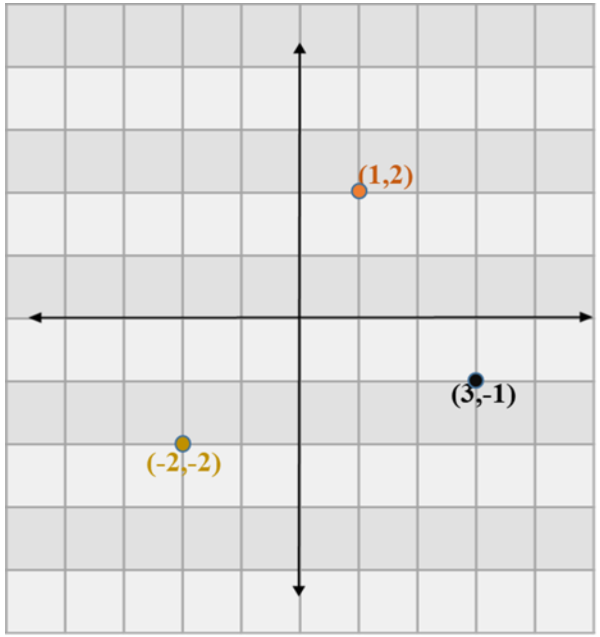


Fig: Vector as a point

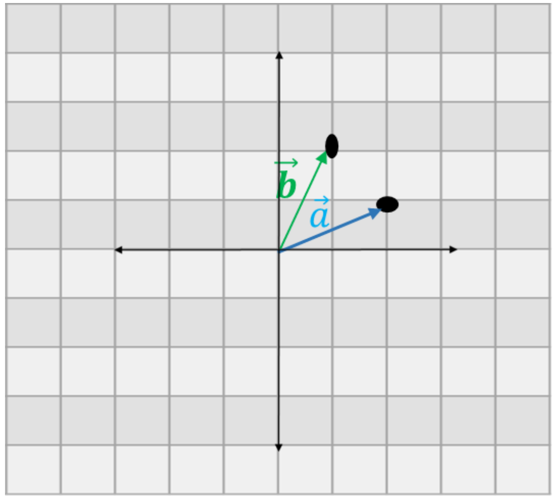
1.2 Vector : General Definition.

- **For Math/Physics:**
  - Vectors are quantity having both **direction and magnitude** written as $\vec{\mathbf{v}}$ (**arrow** represents the **direction** and **v: magnitude** which is proportional to length)
- **For Computer science:**
  - Vectors are **one dimensional ordered array** of **real value numbers** (**scalars**).
  - Denoted by **bold-font lower case**, can be written in **column** or **row** form.

$$\boldsymbol{v} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{5} \end{bmatrix} \textbf{ or; } \boldsymbol{v} = [\mathbf{1} \quad \mathbf{0} \quad \mathbf{5}]$$

    - length of an array defines the **dimension of vector** i.e. how many **axis** are required to represent the **vector in graph**.
  - To generalize: we can write $\boldsymbol{v} \in \mathbb{R}^{\boldsymbol{n}}$. Here $\boldsymbol{n}$ signifies the dimension.

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{v_1} \\ \boldsymbol{v_2} \\ \vdots \\ \boldsymbol{v_n} \end{bmatrix}$$



In Figure Above we can observe **two vector**:

$\vec{\mathbf{a}} = [\mathbf{2}, \mathbf{1}]$ **and** $\vec{\mathbf{b}} = [\mathbf{1}, \mathbf{2}]$ in **2 dimensional space**

i.e. requires two elements or position or axis to describe each vector.

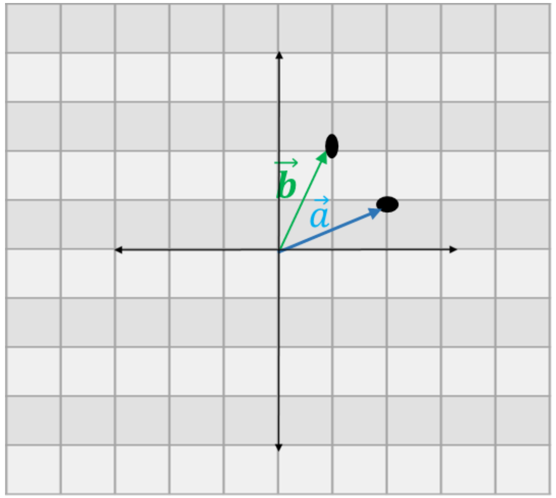
1.2 Vector : General Definition.

- **For Math/Physics:**
  - Vectors are quantity having both **direction and magnitude** written as $\vec{\mathbf{v}}$ (**arrow** represents the **direction** and **v: magnitude** which is proportional to length)
- **For Computer science:**
  - Vectors are **one dimensional ordered array** of **real value numbers** (**scalars**).
  - Denoted by **bold-font lower case**, can be written in **column** or **row** form.

$$v = \begin{bmatrix} 1 \\ 0 \\ 5 \end{bmatrix} \text{ or; } v = \begin{bmatrix} 1 & 0 & 5 \end{bmatrix}$$

    - length of an array defines the **dimension of vector** i.e. how many **axis** are required to represent the **vector in graph**.
  - To generalize: we can write $v \in \mathbb{R}^n$. Here n signifies the dimension.

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \ddots \\ v_n \end{bmatrix}$$



In Figure Above we can observe **two vector**:
$\vec{a} = [2, 1]$ **and** $\vec{b} = [1, 2]$ in **2 dimensional space**
i.e. requires two elements or position or axis to describe each vector.

- **Question:**
  - $\vec{c} = [1, 2, 3]$: **In which dimensional space is vector c on?**
  - **Is $\vec{a} == \vec{b}$? And can we $\vec{a} + \vec{c}$?**
  - **What significance does dimension hold?**

1.3 Vector in Space.

- **Vector in n-dimensional space:**
  - If **n** is a positive integer, then an ordered n-tuple is a sequence of n real numbers $[\mathbf{n_1}, \mathbf{n_2}, \dots, \mathbf{n_n}]$ .The set of all **ordered n-tuples** is called **n-space** and is denoted by $\mathbb{R}^n$.
    - **Vectors in $\mathbb{R}^n$:**
      - Let $\mathbb{R}^n = \{(\mathbf{x_1}, \dots, \mathbf{x_n}) : \mathbf{x_j} \in \mathbb{R} \text{ for } \mathbf{j} = 1, \dots, \mathbf{n}\}$. Then,
        - $\vec{\mathbf{x}} = [\mathbf{x_1}, \dots, \mathbf{x_n}]$ is called a **vector**.
      - {The number $\mathbf{x_j}$ are called the components of $\vec{\mathbf{x}}$.}

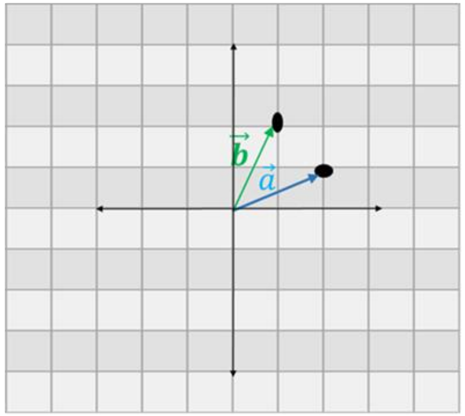


Fig: 2 dimensional vector space

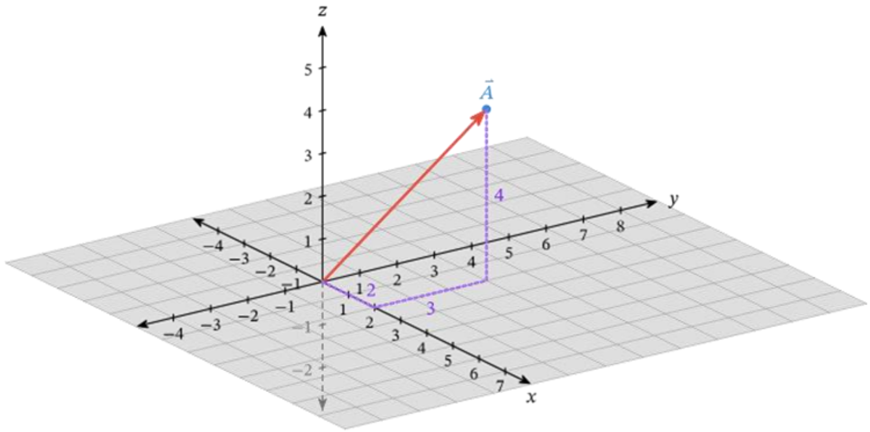


Fig: 3 dimensional vector space

1.4 Vector in Vector – Space.

- **Vector Space:**
  - A set **V** of ***n*-dimensional vectors** (with a corresponding set of scalars) such that the set of vectors is:
    - **"closed" under vector addition.**
    - **"closed" under scalar multiplication.**
    - **Origins are defined and fixed {0 vector must exist}**
  - In other words:
    - **For addition of two vectors:**
      - takes **two vectors** $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$, and it produces the **third vector** $\mathbf{u} + \mathbf{v} \in \mathbb{R}^2$.
      - (addition of vectors – gives another vector in the same set)
    - **For scalar Multiplication:**
      - Takes **a scalar c** $\in$ **F** and a vector $\mathbf{v} \in \mathbb{R}^n$ produces a **new vector cv** $\in \mathbb{R}^n$.
      - (multiplying a vector by a scalar – gives another vector in the same set)

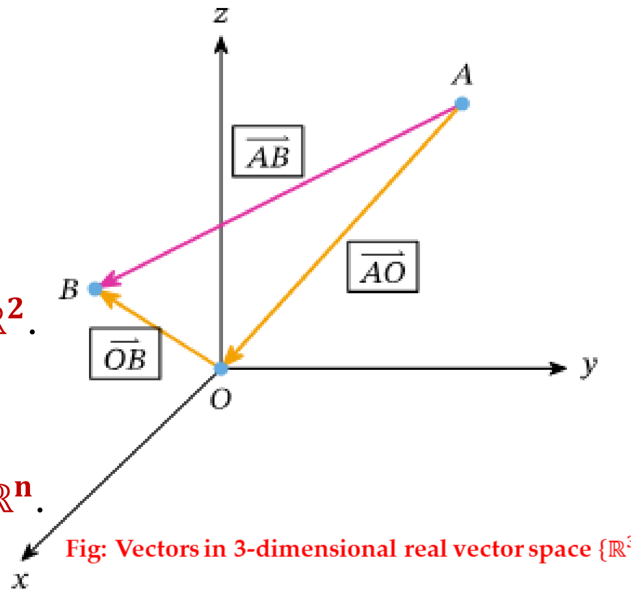


Fig: Vectors in 3-dimensional real vector space $\{\mathbb{R}^3\}$

1.4.1 Axioms of Vector – Space.

- If **V** is a set of vectors satisfying the above definition of a vector space then it satisfies the following axioms:
  - **Existence of an Additive Identity**: any vector space **V** **must have a zero vector**.
  - **Existence of Negative Vector**: for any vector **v** in **V** its **–ve** must also be in **V**.
  - Has **Athematic/Algebraic Properties** – We can perform valid mathematical operations.

{details in next slide}

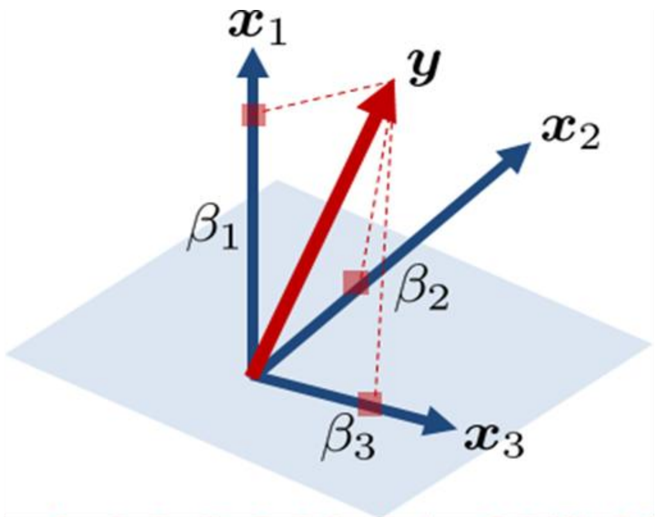


Image from Stanley Chan Book: Introduction to Probability for Data Science.

1.5 Vector Algebra in Vector Space.

- If **u, v,** and **w** are **vectors** in $\mathbb{R}^n$, and if **k** and **m** are **scalars**, then:
  - **Commutative properties**:
    - **u + v = v + u**
    - **ku = uk**
  - Associative Properties:
    - **(u + v) + w = u + (v + w)**
    - **k(mu) = (km)u**
  - Distributive Properties:
    - **k(u + v) = ku + kv**
    - **(k + m)u = ku + mu**
  - **Scalar unity and Scalar zero:**
    - **1u = u**
    - **u + 0 = 0 + u = u**
    - **u + (−u) = 0**
    - **0u = 0 (zero vector).**

1.6 Vector Operations : Addition.

- **Parallelogram Rule for Vector Addition:**
  - If **v** and **w** are vectors in **n-space{ℝ^*n*}** that are positioned so their **initial points coincide**,
    - then the **two vectors form adjacent sides of a parallelogram**, and
    - the **sum v + w** is the vector represented by the arrow from the common initial point of v and w to the opposite vertex of the parallelogram :

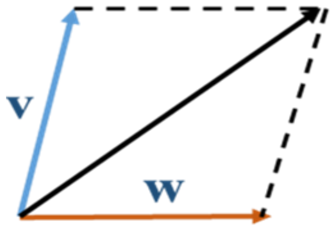


Fig: a

- **Triangle Rule for Vector Addition:**
  - If **v** and **w** are vectors in **n-space{ℝ^*n*}** that are positioned so the initial point of w is at the terminal point of v,
  - then the **sum v + w** is represented by the arrow from the initial point of v to the terminal point of w :

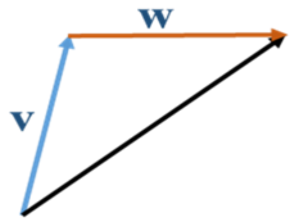


Fig: b

1.6.1 Vector Operations : Subtraction.

- The negative of a **vector v**, denoted by **−v**, is the **vector** that has the **same length** as **v** but is **oppositely directed** (**Fig: a**), and the difference of **v** from **w**, denoted by **w − v**, is taken to be the **sum w − v = w + (−v).**

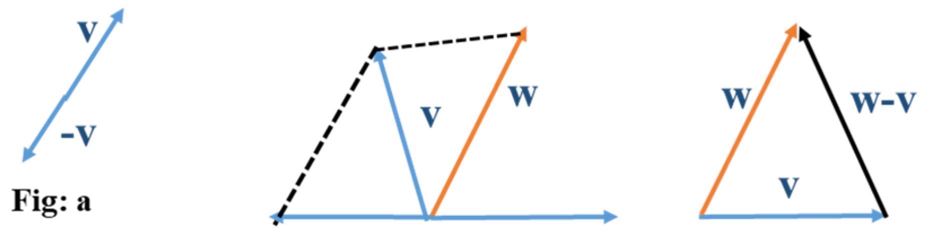


Fig: a Fig: b Fig: c

- **Vector Operations with Computer:**
  - In computer addition and/or subtraction in computer are done as **element wise operation** as shown below:

Addition or Subtraction:

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_n \end{bmatrix} \in \mathbb{R}^n; \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \cdots \\ v_n \end{bmatrix} \in \mathbb{R}^n \text{then:}$$

$$\mathbf{w} \pm \mathbf{v} = \begin{bmatrix} w_1 \pm v_1 \\ w_2 \pm v_2 \\ \cdots \pm \cdots \\ w_n \pm v_n \end{bmatrix} \in \mathbb{R}^n$$

1.6.2 Vector Operations : Scalar – Vector Multiplication.

- **What is Scalar?**
  - It is a number real or complex.
    - Vectors of interest are real then the set of scalars are also real.
  - **Why the name scalars?**
    - It scales the vector by given number.
- **Scalar Multiplication**:
  - If v is a nonzero vector in **n-space**{$\mathbb{R}^n$} , and if k is a **nonzero scalar (number)**, then we define the **scalar product (multiplication)** of v **by** k to be the **vector** whose **length is |k|** times the **length of v** and **whose direction** is the **same** as that of v if k is **positive**
  - and **opposite** to that of v if k is **negative**.
  - If **k = 0** or **v = 0**, then we define **kv to be 0.**

i.e. For $v \in \mathbb{R}^n$, and $k \in \mathbb{R}$ scalar multiplication is:

$$k\,.\,v = \begin{bmatrix} k.v_1 \\ k.v_2 \\ \dots \\ k.v_n \end{bmatrix} \in \mathbb{R}^n$$

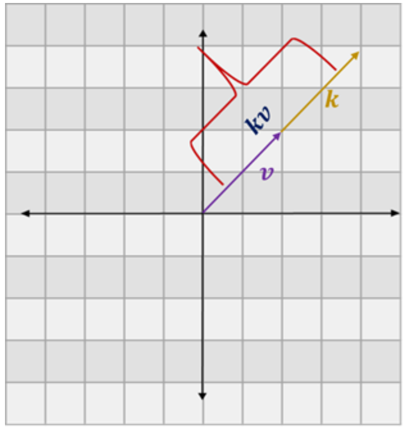


Fig: Scaling the Vector.

Scalar Multiplication – Properties:
{j,k:= scalar and v,u:= vectors}

- Associativity: (**jk**)**v** = **j**(**kv**)
- Distributive property (Left and Right): (**j+k**)**v** = **jv** + **kv** ~ **v**(**j+k**) = **vj** + **vk**
- Distributive property vector addition: **j**(**u + v**) = **ju** + **kv**

1.6.3 Vector Operations : Vector – Vector Multiplication.

- **aka dot or inner product:**
  - Given two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, the quantity $\mathbf{u}^{\mathbf{T}}\mathbf{v}$, sometimes called the **inner product** or **dot product** of the vectors, is a real number given by:
  - $\mathbf{u}^{\mathbf{T}}\mathbf{v} \in \mathbb{R} = [\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_n}] \cdot \begin{bmatrix} \mathbf{v_1} \\ \mathbf{v_2} \\ \ldots \\ \mathbf{v_n} \end{bmatrix} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{u_i} \times \mathbf{v_i}$
  - **dot product: properties:**
    - If u, v, and w are vectors in $\mathbb{R}^n$ and k a scalar then: **{u, v := vector & k := scalar}**
      - **0.v = v.0 = 0 [0 vector]**
      - **(u+v).w = u.w + v.w**
      - **u.(v-w) = u.v – u.w**
      - **(u-v).w = u.w – v.w**
      - **k(u.v) = u.(kv)**

Disclaimer: Dot Product Vs. Inner Product – Key Difference:

Dot Product:
- Specific to **real-valued vectors** in **Euclidean space**.
- Yields a **scalar** that measures the **similarity or projection of vectors**.

Inner Product:
- Generalized to **complex vectors** and **other vector spaces** (**like function spaces**).
- Includes the concept of taking the **complex conjugate** in **complex spaces**.
- Must satisfy additional properties like **conjugate symmetry and linearity**.

In essence, the **dot product** is a **special case** of the **inner product** for **real-valued vectors**

1.7 Magnitude of a Vector – Vector Norm.

- **Vector Norm: Definition**
  - For a vector:= $\mathbf{v} = [\mathbf{v_1} \quad \cdots \quad \mathbf{v_n}] \in \mathbb{R}^\mathbf{n}$, then the **norm of v** (also called the **length of v** or the **magnitude of v**) is denoted by $\|\mathbf{v}\|$.
  - A **norm** can be any function $\mathbf{f}: \mathbb{R}^\mathbf{n} \to \mathbb{R}$ that satisfies following properties:
    - For all $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{f(v) \geq 0}$ (**non-negativity**).
    - $\mathbf{f(v) = 0}$ if and only if $\mathbf{v = 0}$ (**definiteness**).
    - For all **vector** $\mathbf{v} \in \mathbb{R}^n$,and **scalar** $\mathbf{t} \in \mathbf{R}$, $\mathbf{f(tv) = |t| f(v)}$ (**homogeneity**).
    - For all **vector** $\mathbf{v, w} \in \mathbb{R}^n$: $\mathbf{f(v + w) \leq f(v) + f(w)}$ (**triangle inequality**).

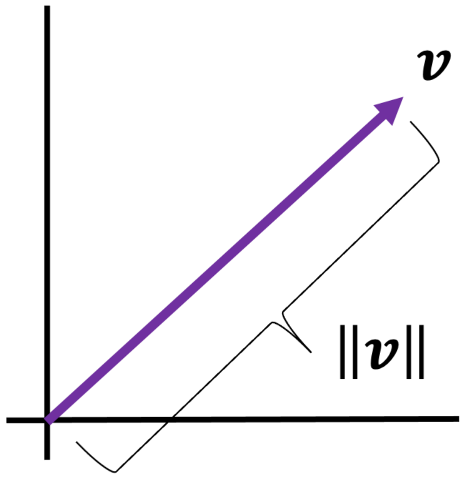


Fig: Norm is a length of a Vector.

1.7.1 Finding : Norm.

- Let $\mathbf{X} = [\mathbf{x_1}, \dots \dots, \mathbf{x_N}]^\mathbf{T}$ be a vector, Then ℓ_p **- norm** of $\mathbf{X}$ is :

$$\|\mathbf{X}\|_\mathbf{p} = \left(\sum_{\mathbf{i=1}}^{\mathbf{N}} |\mathbf{x_i}|^\mathbf{p}\right)^{\frac{\mathbf{1}}{\mathbf{p}}} \text{ for any } \mathbf{p} \geq \mathbf{1}.$$

- **Some popular norms are**:
  - **For $p = 2$, we have ℓ_2 norm**
    - Also called **Euclidean norm**
    - It is the most often used norm
    - ℓ_2 norm is often denoted just as $\|\mathbf{x}\|$ with the subscript 2 omitted
  - **For $p = 1$, we have ℓ_1 norm**
    - Uses the absolute values of the elements
    - Discriminate between zero and non-zero elements
  - **For $p = \infty$, we have ℓ_∞ norm**
    - Known as **infinity norm**, or **max norm**
    - Outputs the absolute value of the largest element

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{\mathbf{x}^T\mathbf{x}}$$

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

$$\|\mathbf{x}\|_\infty = \max_i |x_i|$$

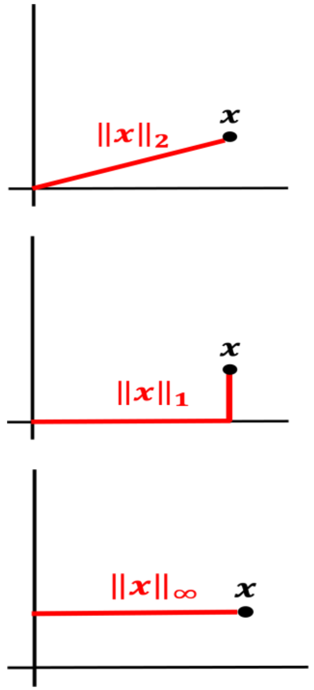


Image from M.N. Bernstein's Blog.

1.8 Vector Projections.

- **Vector projection** is a way of **projecting one vector onto another** to see **how much of one vector lies in the direction** of the other.
  - It helps to understand the component of a vector that aligns with a given direction, and it is widely used in geometry, physics, and machine learning for understanding vector relations, decomposing vectors, and more.

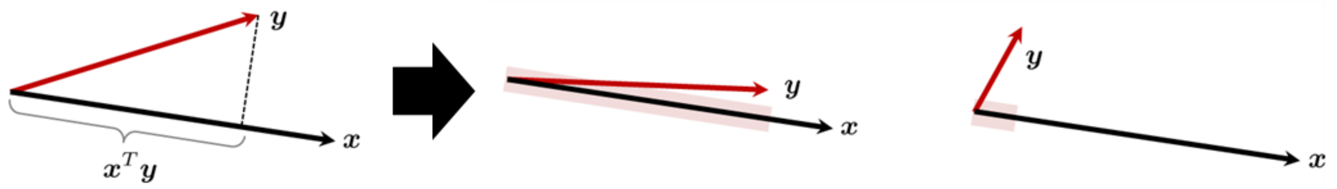


Fig: Inner Product and Vector Projections.

Image from Stanley Chan Book: Introduction to Probability for Data Science.

- **Inner Product and Vector Projections:**
  - When one vector is projected onto the other vector, The **projected distance** is the **inner product**.
  - If **two vectors are correlated** i.e. **nearly parallel**, then the **inner product** will give us **a large value**.
    - Conversely, if the two vectors are **close to perpendicular**, then the **inner product** will **be small**.
  - Therefore, the **inner product** provides a **measure of closeness/similarity** between **two vectors**.

1.8.1 Vector Projections: Types.

- Scalar Projection:
  - The scalar projection of a **vector a onto a vector b** gives a **scalar value** that represents **the magnitude** of **a** in the **direction** of **b**. It is calculated using:

$$\mathbf{proj_b(a)} = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{b}\|}$$

Where:
- $\mathbf{a} \cdot \mathbf{b}$ is the **dot product** of **a and b**.
- $\|\mathbf{b}\|$ is the **magnitude** (length) of **vector b**.

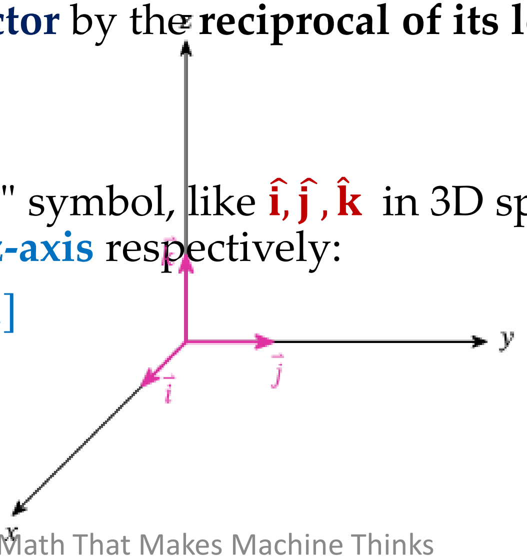
  - This scalar value can be **positive** (when **a** points in the **same direction** as **b**) or **negative** (when they point in opposite **directions**).
- Vector Projection:
  - The **vector projection** of **vector a onto vector b** is a **vector** that points in the direction of b and has **a magnitude** equal to the **scalar projection**.
  - It tells you how much **of vector a "points along" vector b**. It is calculated as:

$$\mathbf{proj_b(a)} = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{b}\|^2}\mathbf{b}$$

Where:
- $\mathbf{a} \cdot \mathbf{b}$ is the **dot product** of **a and b**.
- $\|\mathbf{b}\|^2$ is the **magnitude** (length) of **vector b**.

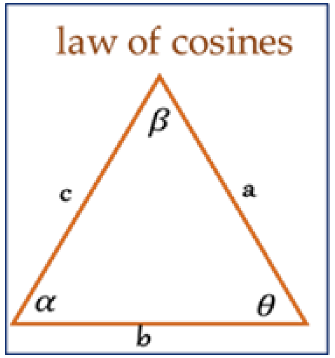
1.9 a Unit Vector.

- A vector of **norm 1** is called a **unit vector**.
  - Such vectors are useful for **specifying a direction** when **length is not relevant** to the problem at hand.
  - “You can obtain a unit vector in a desired direction by choosing any nonzero vector v in that direction and multiplying v by the reciprocal of its length.”
  - Mathematically, if **v** is any **nonzero vector** in $\mathbb{R}^{\mathbf{n}}$, then **unit vector of v** is:
    - $\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$
- **Normalizing a Vector**:
  - The process of **multiplying** a **nonzero vector** by the **reciprocal of its length** to obtain a **unit vector** is called **normalizing v**.
- **Notation:**
  - Unit vectors are often denoted with a "hat" symbol, like $\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}$ in 3D space, which represent the standard unit vectors along the **x-axis**, **y-axis**, and **z-axis** respectively:
    - $\hat{\mathbf{i}} = [1, 0, 0]$; $\hat{\mathbf{j}} = [0, 1, 0]$; $\hat{\mathbf{k}} = [0, 0, 1]$



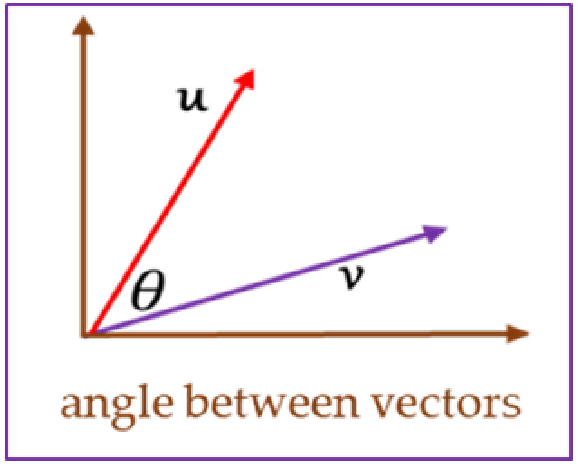
1.10 Angle between two Vector.

- The definition of the angle between vectors can be thought as a generalization of the law of cosines in trigonometry, which defines for a triangle with sides a, b, and c, and angle θ are related as:



$$c^2 = a^2 + b^2 - 2ab\cos\theta$$

- For angel between two vectors **u and v**:
  - We can replace above expression with vector lengths:
    - $\|\mathbf{u}-\mathbf{v}\|^2 = \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 - 2\|\mathbf{u}\|\|\mathbf{v}\|\cos\theta.$
  - With a bit of algebraic manipulation:
    - $\cos\theta = \langle \mathbf{u},\mathbf{v}\rangle / \|\mathbf{u}\|\|\mathbf{v}\|$ then; $\theta = \arccos\left(\frac{\langle \mathbf{u},\mathbf{v}\rangle}{\|\mathbf{u}\|\|\mathbf{v}\|}\right)$



What happens if:
inner product $\rightarrow \langle \mathbf{u},\mathbf{v}\rangle = 0$

1.11 Orthogonal Vectors

- A pair of vectors **u and v** are **orthogonal** if their **inner product** is zero
  - i.e. $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{0}.$
- Notation for a pair of orthogonal vectors is $\mathbf{u} \perp \mathbf{v}$ {i.e. Vector are perpendicular to each other}.
- In the $\mathbb{R}^{\mathbf{n}}$; this is equal to pair of vector forming a $\mathbf{90^0}$ angle.

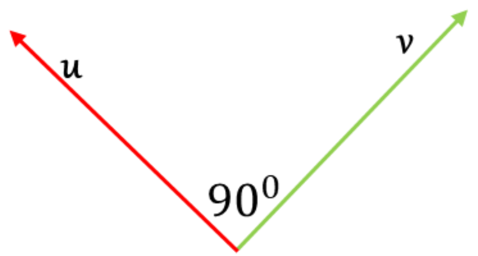


Fig: Orthogonal Vectors

{Almost} Everything about Math for ML.

{2. Review → What are Matrix?}

$$
\begin{array}{c c}
 & \begin{array}{cccc} 1 & 2 & \cdots & n \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ \vdots \\ m \end{array} &
\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
a_{31} & a_{32} & \cdots & a_{3n} \\
\vdots & \vdots & \vdots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{bmatrix}
\end{array}
$$

Image from Wikipedia

2.1 Matrices: Introduction.

- In general: A **matrix** is a **rectangular array** of numbers. The **numbers in the array** are called **the entries** in the **matrix**.
  - Array of numbers are an ***ordered collection of vectors***.
  - Like vectors matrices are also fundamentals in machine learning/AI, as matrices are the way computer ***interact with data*** in practice.
- A **matrix** is represented with a *italicized* upper-case letter like **A**.
  - For two dimensions: we say the matrix **A** has **m rows and n columns**. Each entry of **A** is defined as $\mathbf{a_{ij}}$.
  - Thus a matrix $\mathbf{A^{m\times n}}$ is define as:

$$\boxed{A_{m\times n} \coloneqq \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, a_{ij} \in \mathbb{R}}$$

2.2 Special Matrices.

- Rectangular Matrix:
  - Matrices are said to be rectangular when the number of rows is $\neq$ to the number of columns, i.e. $\mathbf{A}^{\mathbf{m}\times\mathbf{n}}$ with $\mathbf{m} \neq \mathbf{n}$. For instance:

$$\boxed{A_{2\times3} := \begin{bmatrix} 1 & 2 & 3 \\ 5 & 5 & 4 \end{bmatrix}}$$

- Square Matrix:
  - Matrices are said to be square when the number of rows = the number of columns, i.e. $\mathbf{A}^{\mathbf{m}\times\mathbf{n}}$. For instance:

$$\boxed{A_{2\times2} := \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}}$$

- Diagonal Matrix:
  - Square matrices are said to be diagonal when each of its non-diagonal elements is zero, i.e. for
    - $\mathbf{D} = (\mathbf{d_{ij}})$,we have $\forall \mathbf{i}, \mathbf{j} \in \mathbf{n}\ \mathbf{i} \neq \mathbf{j} \Rightarrow \mathbf{d_{ij}} = \mathbf{0}$.
  - For instance:

$$\boxed{A_{3\times3} := \begin{bmatrix} 9 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 5 \end{bmatrix}}$$

- Upper triangular matrix:
  - Square matrices are said to be upper triangular when the elements below the main diagonal are zero i.e. For $\mathbf{D} = (\mathbf{d_{ij}})$, we have $\mathbf{d_{ij}} = \mathbf{0}$, for $\mathbf{i} > \mathbf{j}$. For instance:

$$\boxed{A_{3\times3} := \begin{bmatrix} 9 & 8 & 4 \\ 0 & 1 & 3 \\ 0 & 0 & 5 \end{bmatrix}}$$

- Lower triangular matrix:
  - Square matrices are said to be lower triangular when the elements above the main diagonal are zero . i.e. $\mathbf{D} = (\mathbf{d_{ij}})$, we have $\mathbf{d_{ij}} = \mathbf{0}$, for $\mathbf{i} < \mathbf{j}$. For instance:

$$\boxed{A_{3\times3} := \begin{bmatrix} 9 & 0 & 0 \\ 8 & 1 & 0 \\ 4 & 2 & 5 \end{bmatrix}}$$

- Identity Matrix:
  - A diagonal matrix is said to be the identity when the elements along its main diagonal are equal to one. For instance:

$$\boxed{A_{3\times3} := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

2.2.1 Special Matrices.

- **Symmetric Matrix:**
  - Square matrices are said to be symmetric its equal to its transpose, i.e. $\mathbf{A} = \mathbf{A}^{\mathbf{T}}$. For instance:

$$A_{3\times3} \coloneqq \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 6 \\ 3 & 6 & 1 \end{bmatrix}$$

- **Scalar Matrix:**
  - Diagonal matrices are said to be scalar when all the elements along its main diagonal are equal, i.e. $\mathbf{D} = \boldsymbol{\alpha}\mathbf{I}$. For instance:

$$A_{3\times3} \coloneqq \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

- **Null or Zero Matrix:**
  - Matrices are said to be null or zero matrices when all its elements equal to zero, which is denoted as $\mathbf{0}_{\mathbf{m}\times\mathbf{n}}$. For instance:

$$A_{3\times3} \coloneqq \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- **Equal Matrix:**
  - Two matrix are said to be equal if
    - $\mathbf{A}(\mathbf{a_{ij}}) = \mathbf{B}(\mathbf{b_{ij}})$.
  - For instance:

$$B_{2\times2} \coloneqq \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}$$

$$A_{2\times2} \coloneqq \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}$$

2.2.2 Special Matrices → Design Matrix.

- **Design Matrix:**
  - A design matrix is a matrix containing **data** about **multiple characteristics** of **several individuals or objects**. **Each row corresponds to an individual** and **each column to a characteristic**. For instances:
    - If we measure the **height and weight** of **five individuals**, we can collect the measurements in a design matrix having five rows and two columns.
    - Each **row corresponds** to one of the ten individuals, the first column contains the height measurements and the second one reports the weights:

S.NO	Height	Weight
1	$\mathbf{h_1}$	$\mathbf{w_1}$
2	$\mathbf{h_2}$	$\mathbf{w_2}$
3	$\mathbf{h_3}$	$\mathbf{w_3}$
4	$\mathbf{h_4}$	$\mathbf{w_4}$
5	$\mathbf{h_5}$	$\mathbf{w_5}$

Data in Tabular Format

$$\textbf{Data} \coloneqq \begin{bmatrix} \mathbf{h_1} & \mathbf{w_1} \\ \mathbf{h_2} & \mathbf{w_2} \\ \mathbf{h_3} & \mathbf{w_3} \\ \mathbf{h_4} & \mathbf{w_4} \\ \mathbf{h_5} & \mathbf{w_5} \end{bmatrix} \textbf{ or } \coloneqq \begin{bmatrix} \mathbf{h_1} & \mathbf{h_2} & \mathbf{h_3} & \mathbf{h_4} & \mathbf{h_5} \\ \mathbf{w_1} & \mathbf{w_2} & \mathbf{w_3} & \mathbf{w_4} & \mathbf{w_5} \end{bmatrix}$$

Representing Data in Matrix .

Fig: Design Matrix Representing a Data.

2.3 Matrix Operation: Arithmetic.

Matrix – Matrix Sum:

- Matrices are added or subtracted in a element wise fashion.
- The sum **(+ or-)** of $\mathbf{A} \in \mathbb{R}^{\mathbf{m \times n}}$ and $\mathbf{B} \in \mathbb{R}^{\mathbf{m \times n}}$ and is defined as:

$$\boldsymbol{A \pm B} := \begin{bmatrix} a_{11} \pm b_{11} & \cdots & a_{1n} \pm b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} \pm b_{m1} & \cdots & a_{mn} \pm b_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}$$

Matrix – Scalar Multiplication:

- Matrix-scalar multiplication is an element-wise operation.
- Each element of the **matrix A** is multiplied by the **scalar α** is defined as:

$$\mathbf{a_{ij} \times \alpha, \text{ such that } (\alpha A)_{ij} = \alpha(A)_{ij}}.$$

$$\alpha = 2 \text{ and } A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$\alpha A = 2 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$$

2.3.1 Matrix Operation: Arithmetic.

Matrix – vector Multiplication:

- Matrix-vector multiplication equals to taking **the dot product** of **each column n** of **matrix-A** with **each element** of **vector-x** resulting in **vector y** and is defined as:

$$A.X := \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1 \begin{bmatrix} a_{11} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ \vdots \\ a_{m2} \end{bmatrix} + x_n \begin{bmatrix} a_{1n} \\ \vdots \\ a_{mn} \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_{mn} \end{bmatrix}$$

- Example:

Given Matrix A and column vector x ; Compute $A \times x$:

$$A = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 3 & -1 \\ 1 & 2 & 1 \end{bmatrix} \in \mathbb{R}^{3\times3} \text{ and } x = \begin{bmatrix} 3 \\ -1 \\ 4 \end{bmatrix} \in \mathbb{R}^{3\times1}$$

$$Ax = 3 \times \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + (-1) \times \begin{bmatrix} 0 \\ 3 \\ 2 \end{bmatrix} + 4 \times \begin{bmatrix} -2 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \times 1 + 3 \times 0 + 3 \times 1 \\ -1 \times 0 - 1 \times 3 - 1 \times 2 \\ 4 \times -2 + 4 \times -1 + 4 \times 1 \end{bmatrix}$$

$$= \begin{bmatrix} -5 \\ -7 \\ 5 \end{bmatrix} \in \mathbb{R}^{3\times1}$$

Hadamard Product:

- It is tempting to think in **matrix-matrix multiplication** as an **element-wise operation**, as multiplying each overlapping element of **A and B**.
- Such operation is called **Hadamard product ; defined as:**

$$A^{\circ}B = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m\times n} \circ \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{bmatrix} \in \mathbb{R}^{m\times n} = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mn} \end{bmatrix} \in \mathbb{R}^{m\times n}$$

Where:

$$a_{mn} \times b_{mn} := c_{mn}$$

- **Example:**

$$A.B = \begin{bmatrix} 0 & 2 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 0 \times 1 & 2 \times 3 \\ 1 \times 2 & 4 \times 1 \end{bmatrix} = \begin{bmatrix} 0 & 6 \\ 2 & 4 \end{bmatrix}$$

2.3.2 Matrix Operation: Arithmetic.

- **Matrix – Matrix Multiplication:**
  - Matrix multiplication between $\mathbf{A} \in \mathbb{R}^{\mathbf{n \times p}}$ and $\mathbf{B} \in \mathbb{R}^{\mathbf{n \times p}}$ with resultant matrix $\mathbf{C} \in \mathbb{R}^{\mathbf{m \times p}}$ can be defined as:

$$A.B := \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} b_{11} & \cdots & b_{1p} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{np} \end{bmatrix} = \begin{bmatrix} c_{11} & \cdots & c_{1p} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mp} \end{bmatrix}$$

  - Where: $c_{ij} := \sum_{l=1}^{n} a_{il} b_{lj}$; with i=1,...m; and j=1,...,p

  - Matrix – Matrix Multiplication Properties:
    - Associativity:
      $$(AB)C = A(BC)$$
      Associativity with scalar multiplication:
      $$\alpha(AB) = (\alpha A)B$$
    - Distributive with addition:
      $$A(B \pm C) = AB \pm AC$$
    - Caution! In matrix-matrix multiplication orders matter, it is not commutative i.e.
      $$AB \neq BA.$$

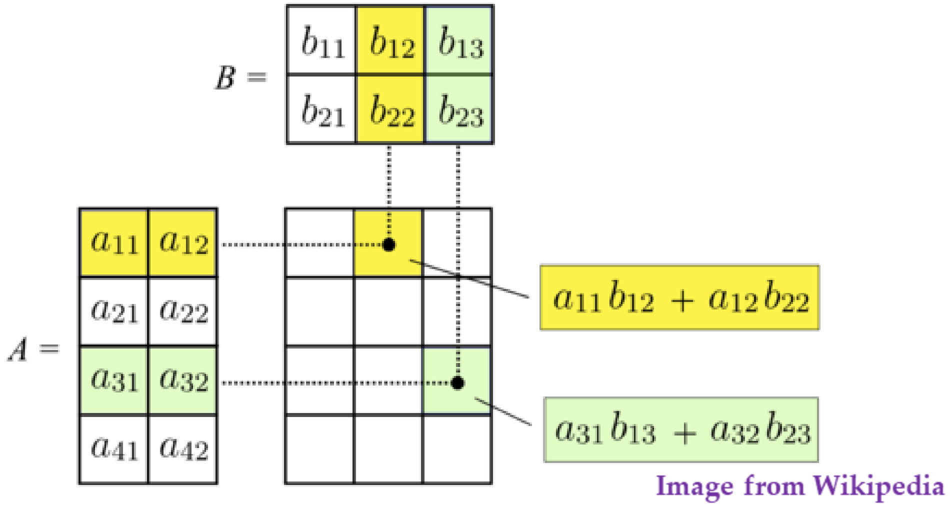


Fig: Schematic representation of Matrix product

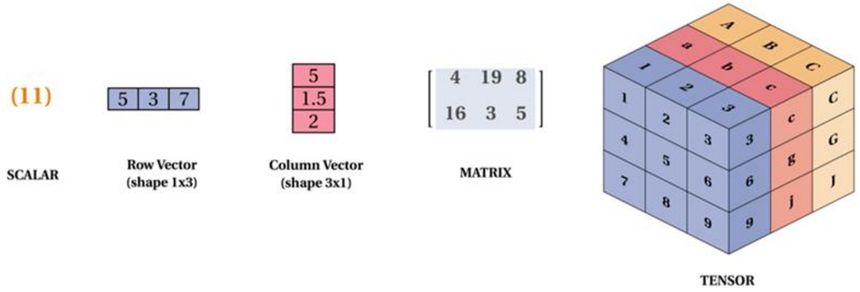
{Almost} Everything about Math for ML.

{3. Review → What are Tensors?}

?

3.1 What are Tensors?

- A tensor is a **multidimensional array** and a **generalization** of the concepts of a **vector and a matrix.**



- Tensors can have many axes, here is a tensor with three axes:

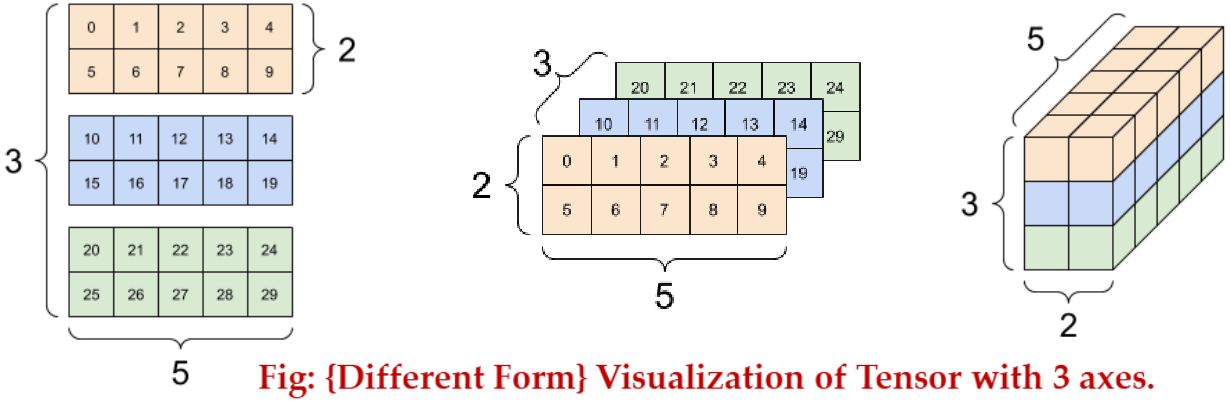
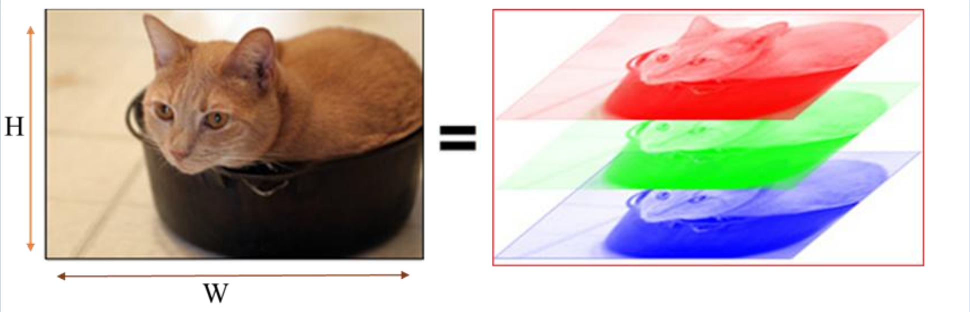


Fig: {Different Form} Visualization of Tensor with 3 axes.

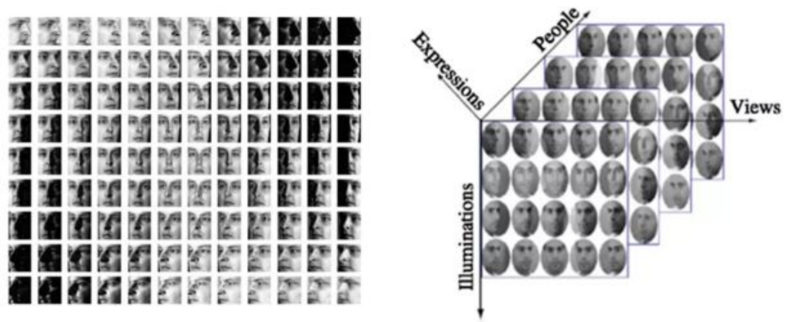
3.2 Tensor → Example.

- Tensors in DL are Used to represent an image.
  - image_shape := Height × Width × Color Channel (RGB)



A color image is composed of 3 channels and is 3rd rank tensors.

facial images database is 6th-order tensor



color video is 4th-order tensor

{Almost} Everything about Math for ML.

{4. Review → Derivative of a {univariate} Function.}

4.1 What is Derivative?

- The **derivative of a function** measures **how** the **output value** of the function changes as we make **small adjustments** to its **input**.
- Notations:
  - The derivative of a function $\mathbf{f(x)}$ is represented by $\mathbf{\frac{d}{d(x)}(f(x))}$ or $\mathbf{\frac{df(x)}{d(x)}}$ or $\mathbf{f'(x)}$ and is defined as:
- If we have a function $\mathbf{f(x)}$, the derivative $\mathbf{f'(x)}$ at a point $\mathbf{x}$ tell us the rate of change of function $\mathbf{f}$ at that point.
- This rate of change is crucial for optimization techniques,
  - such as finding maxima or minima, which are frequently used in training machine learning models.

4.2 Derivatives: Scalar Function.

- **Most popular:**
  - **Derivative of a Scalar function i.e. Scalar derivatives $\mathbf{f}: \mathbb{R} \to \mathbb{R}$**
    - A scalar function is a function that maps a **real number $\mathbf{x}$** to another **real number $\mathbf{f(x)}$.**
      - $\mathbf{f(x) = x^2}$
      - Here x: a real number and $f(x)$: also a real number.
    - We are interested in the **rate** at which $\mathbf{f(x)}$ **changes** as $\mathbf{x}$ **changes**.
    - The derivative is the heart of calculus, buried inside this definition:
      - $\mathbf{f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}}$ **when the limit exists.**
      - popularly known as the "**limit definition of the derivative**" or "**derivative by using the first principle**"
    - But what does it mean?

4.2.1 Derivative First Principle: Interpretation.

- **Derivative of a function is a measure of local slope.**
  - **1st Example: For Linear Function $y = f(x) = 2x$.**

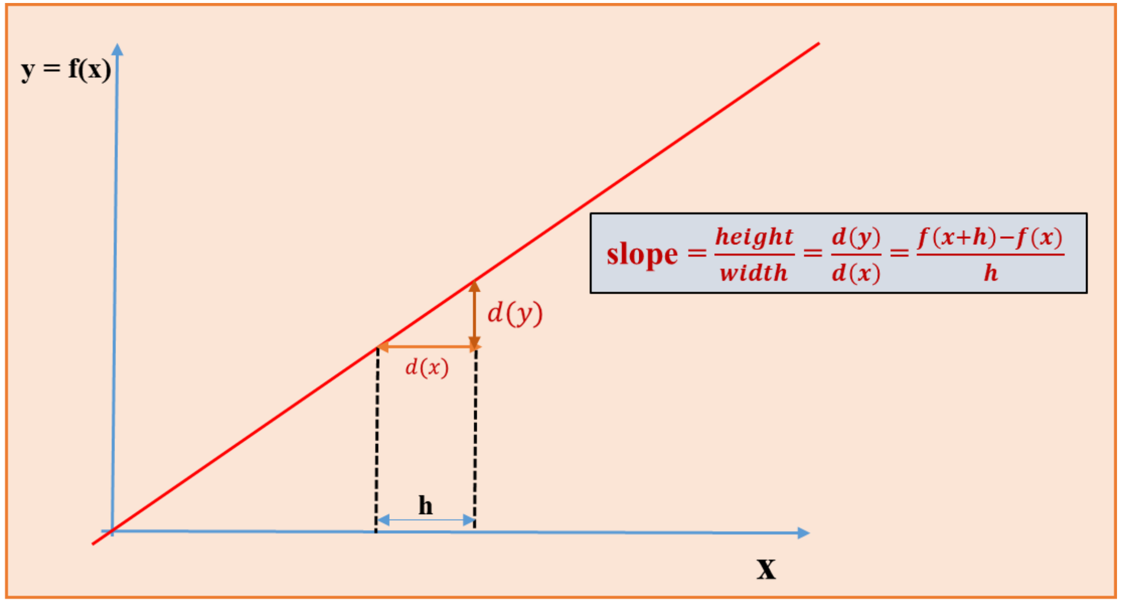
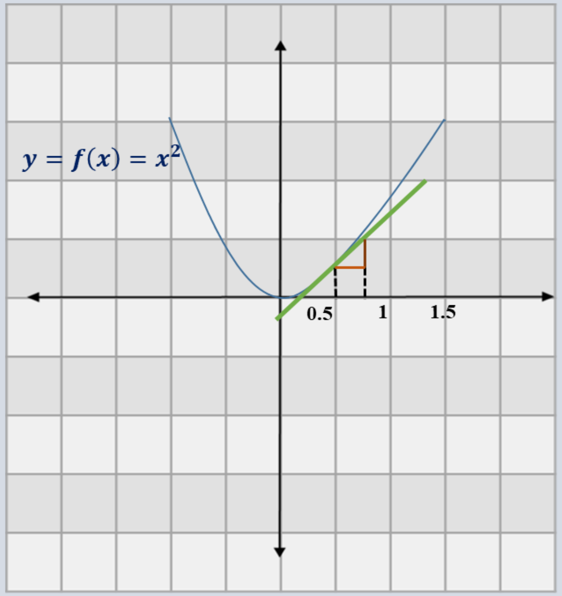


Fig: Derivative → Interpretation.

What for non linear function?

4.2.2 Derivative First Principle: Interpretation.

- **2nd Example: For Non - Linear Function $y = f(x) = x^2$.**



I want to find the slope at point 0.5. What does this mean?
You draw a tangent at x = 0.5 and find the slope of that tangent line.

$$\text{Slope} = \frac{height}{width} = \frac{d(y)}{d(x)}\Big|_{x=0.5} = 1$$

- The derivative of a function at a **point is the slope of the tangent drawn to that curve** at that point.
  - (slope) derivative of a linear function (straight line) is constant at all the point not for the non-linear function.
- It also represents the **instantaneous rate of change** at a point on the function.

4.3 Some Common Rules for determining Derivative.

Rule	Function	Derivative
Sum – Difference Rule	$f(x) \pm g(x)$	$f'(x) \pm g'(x)$
Multiplication by Constant	$c.f(x)$	$c.f'(x)$
Product Rule	$f(x).g(x)$	$f'(x).g(x) + f(x).g'(x)$
Quotient Rule	$f(x)/g(x)$	$\frac{f'(x).g(x) - f(x).g'(x)}{(g(x))^2}$
Chain Rule	$f(g(x))$	$f'(g(x)).g'(x)$

!!! Hands on practice in Tutorial.

4.4 Derivative of some common function.

Function - Type	Function - Notation	Derivative
Constant function	$f(x) = c$; where c is real constant.	$f'(x) = (c)' = 0$.
Identity function	$f(x) = x$	$f'(x) = (x)' = 1$.
Linear function	$f(x) = mx$	$f'(x) = (mx)' = m$.
Function of the form	$f(x) = x^n$	$f'(x) = (x^n)' = nx^{n-1}$.
Exponential function of the form	$f(x) = a^x$; where $a > 0$	$f'(x) = (a^x)' = a^x \ln(a)$.
Exponential function	$f(x) = e^x$	$f'(x) = (e^x)' = e^x$.
Logarithmic function	$f(x) = \ln(x)$	$f'(x) = (\ln(x))' = \frac{1}{x}$.
Sinusoidal function	$f(x) = \sin(x)$	$f'(x) = (\sin(x))' = \cos(x)$.
Cosine function	$f(x) = \cos(x)$	$f'(x) = (\cos(x))' = -\sin(x)$.
Tangent function	$f(x) = \tan(x)$	$f'(x) = (\tan(x))' = \sec^2(x)$.

{Almost} Everything about Math for ML.

{5. Review → Derivative of a {multivariate} Function.}

5.1 Derivative of a Multivariate Function.

- (**scalar derivative of**)Multivariate function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ are in the form $f(x, y) = x^2 y$.
- **Partial Derivative:**
  - In mathematics, a **partial derivative** of a function of several variables is its derivative with respect to one of those variables, with the others held constant (as opposed to the total derivative, in which all variables are allowed to vary).
  - This swirly-d symbol,∂, often called "del", is used to distinguish partial derivatives from ordinary single-variable (regular) derivatives.
    - For Example: $f(x, y) = x^2 y$.

$$\frac{\partial}{\partial x} = \underbrace{\frac{\partial}{\partial x} x^2 y}_{\text{Treat } y \text{ as a constant, then take regular derivative.}} = y \frac{\partial}{\partial x} x^2 = 2xy$$

Treat y as a constant, then take regular derivative.

$$\frac{\partial}{\partial y} = \underbrace{\frac{\partial}{\partial y} x^2 y}_{\text{Treat } x \text{ as a constant, then take regular derivative.}} = x^2 \frac{\partial}{\partial y} y = x^2 . 1$$

Treat x as a constant, then take regular derivative.

Derivative of $f(x, y) = x^2 y$ are $2xy; x^2$

- Partial derivatives are used in vector calculus and differential geometry.

5.2 {some popular} Nomenclature of Derivative.

- Derivative of a vector/matrix a.k.a Matrix/Vector Calculus is an extension of ordinary scalar derivative to higher dimensional settings.
- Overview of some extended derivative style:

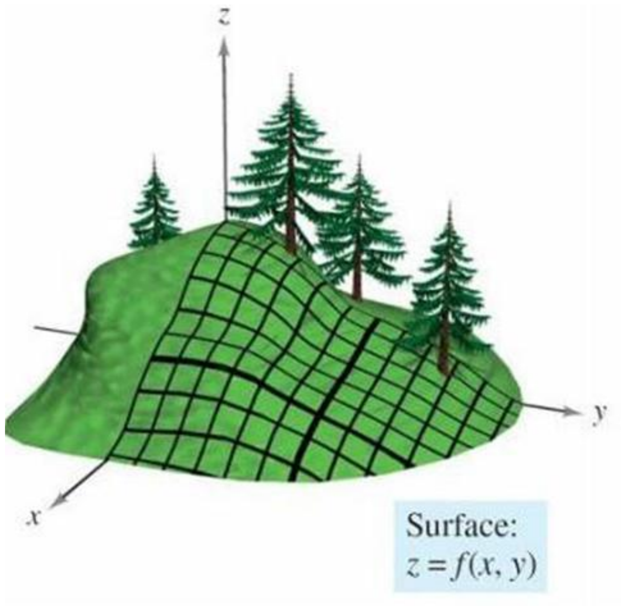
Setting	Derivative	Notation
$\boldsymbol{f}: \mathbb{R} \to \mathbb{R}$	Scalar Derivative	$\boldsymbol{f}'(\boldsymbol{x})$
$\boldsymbol{f}: \mathbb{R}^{\boldsymbol{n}} \to \mathbb{R}$	Gradient	$\nabla \boldsymbol{f}(\boldsymbol{x})$
$\boldsymbol{f}: \mathbb{R}^{\boldsymbol{n} \times \boldsymbol{m}} \to \mathbb{R}$	Gradient	$\nabla \boldsymbol{f}(\boldsymbol{x})$
$\boldsymbol{f}: \mathbb{R}^{\boldsymbol{n}} \to \mathbb{R}^{\boldsymbol{m}}$	Jacobian	$\boldsymbol{J}_{\boldsymbol{f}}$

5.3 Gradient.

- Gradient:
  - The gradient of a function of multiple variables is the vector of partial derivatives of the function with respect to each variable.
  - Scalar-by-vector $\{f: \mathbb{R} \to \mathbb{R}^n\}$:
    - The derivative of a scalar function y with respect to a vector $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ is written as:
      - gradients of y: $\nabla y = \frac{\partial y}{\partial x} = \left[\frac{\partial y}{\partial x_1} \;\; \frac{\partial y}{\partial x_2} \;\; \cdots \;\; \frac{\partial y}{\partial x_n}\right]^T. \longrightarrow$ gradients.
        - **{Stack the partial derivative against all the element of vector $\boldsymbol{x}$}**
  - Scalar-by-Matrix $\{f: \mathbb{R} \to \mathbb{R}^{n \times m}\}$:
    - The derivative of a scalar function y with respect to a $n \times m$ matrix X is written as:
      - gradients of y: $\nabla y = \frac{\partial y}{\partial X} = \begin{bmatrix} \frac{\partial y}{\partial x_{11}} & \cdots & \frac{\partial y}{\partial x_{n1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial x_{1m}} & \cdots & \frac{\partial y}{\partial x_{nm}} \end{bmatrix}$
      - **{Stack the partial derivative against all the element of Matrix X.}**

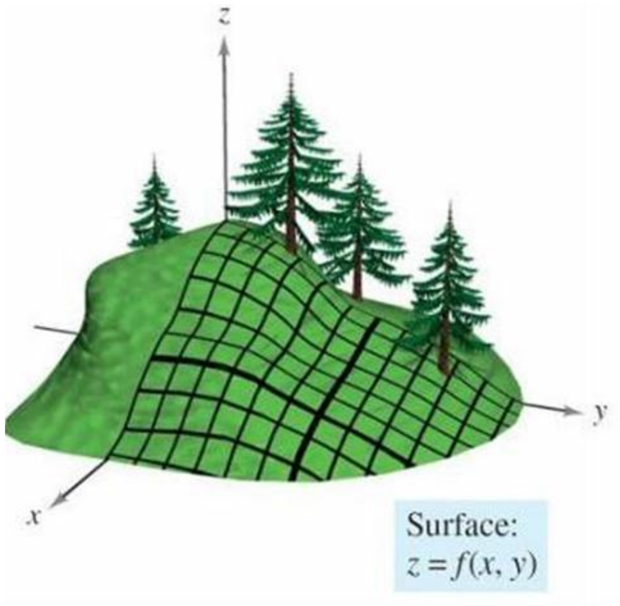
gradient is also the direction of steepest ascent, What does that mean?

5.4 Gradient: Geometric Interpretation.



He want's to scale the hill:
Let's assume he can take two routes:
  One through x→ co-ordinate(direction)
  One through y → co-ordinate(direction)
Which route would be fastest?

5.4 Gradient: Geometric Interpretation.



He want's to scale the hill:
Let's assume he can take two routes:
One through x→ co-ordinate(direction)
One through y → co-ordinate(direction)
Which route would be fastest?

Whichever direction has highest slope(gradient) i.e.
Find the gradient of the surface:

$$z = f(x, y)$$

gradient is a partial derivative of z against x and y stack in the vector.

This is read as: "grad. of z" or "grad z" ← $\nabla z = [\frac{\partial f(x,y)}{\partial x} \quad \frac{\partial f(x,y)}{\partial y}]$

5.5 Gradient: Example 1.

- **$z = f(x, y) = 3x^2y$ find the gradient of z at $[1, 1]$.**
- We know gradient of z is :
  - $\nabla z = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} & \frac{\partial f(x,y)}{\partial y} \end{bmatrix}$
- Finding: $\frac{\partial f(x,y)}{\partial x}$ i.e. y is constant.

$$\boxed{\frac{\partial f(x,y)}{\partial x} = \frac{\partial 3yx^2}{\partial x} = \frac{3y\partial x^2}{\partial x} = 3y2x = 6yx}$$

- Finding $\frac{\partial f(x,y)}{\partial y}$ i.e. x is constant.

$$\boxed{\frac{\partial f(x,y)}{\partial y} = \frac{\partial 3yx^2}{\partial y} = \frac{3x^2\partial y}{\partial y} = 3x^2 \times 1 = 3x^2}$$

- ∇z is: $\boxed{\nabla z = \begin{bmatrix} 6yx & 3x^2 \end{bmatrix}}$
- ∇z at $\begin{bmatrix} 1 & 1 \end{bmatrix}$: $\boxed{\nabla z = \begin{bmatrix} 6 \times 1 \times 1 & 3 \times 1^2 \end{bmatrix} = \begin{bmatrix} 6 & 3 \end{bmatrix}}$

5.6 A Vector – Valued Function : Definition.

- **Vector – Valued Function:**
  - When dealing with **vector-valued functions**, we consider functions that take **one or more variables** as input and produce a vector as output.
  - Unlike scalar-valued functions, where the output is a single value, vector-valued functions yield a vector that has multiple components.

Definition of Vector – Valued Function:

A vector valued function can be represented as:

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \mathbf{f}_1(\mathbf{x}) \\ \mathbf{f}_2(\mathbf{x}) \\ \vdots \\ \mathbf{f}_n(\mathbf{x}) \end{pmatrix}$$

where $\mathbf{f}: \mathbb{R}^m \to \mathbb{R}^n$ and each component $\mathbf{f}_i(\mathbf{x})$ is a scalar function of $\mathbf{x}$.

For example, a **function** $\mathbf{r}$ **that takes a scalar** $\mathbf{t}$ often representing time and maps it to a position vector in three-dimensional space could look like this:

$$\mathbf{r}(\mathbf{t}) = \begin{pmatrix} \mathbf{x}(\mathbf{t}) \\ \mathbf{y}(\mathbf{t}) \\ \mathbf{z}(\mathbf{t}) \end{pmatrix}$$

5.6.1 Gradient of a Vector – Valued Function : Jacobian.

- **vector-by-vector $\{\boldsymbol{f}: \mathbb{R}^n \to \mathbb{R}^m\}$:**
  - The derivative of a vector function : $\boldsymbol{y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n]^T \in \mathbb{R}^n$ with respect to an input vector $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]^T \in \mathbb{R}^m$ is written as:

$$\frac{\partial \mathrm{y}}{\partial \mathrm{x}} = \begin{bmatrix} \frac{\partial \mathrm{y}}{\partial \mathrm{x}_1} \\ \frac{\partial \mathrm{y}}{\partial \mathrm{x}_2} \\ \cdots \\ \frac{\partial \mathrm{y}}{\partial \mathrm{x}_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathrm{y}_1}{\partial \mathrm{x}_1} & \frac{\partial \mathrm{y}_1}{\partial \mathrm{x}_2} & \cdots & \frac{\partial \mathrm{y}_1}{\partial \mathrm{x}_m} \\ \frac{\partial \mathrm{y}_2}{\partial \mathrm{x}_1} & \frac{\partial \mathrm{y}_2}{\partial \mathrm{x}_2} & \cdots & \frac{\partial \mathrm{y}_2}{\partial \mathrm{x}_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathrm{y}_n}{\partial \mathrm{x}_1} & \frac{\partial \mathrm{y}_n}{\partial \mathrm{x}_2} & \cdots & \frac{\partial \mathrm{y}_n}{\partial \mathrm{x}_m} \end{bmatrix} = \mathrm{J}_{\mathrm{y}}$$

  - **J_{y}: called Jacobian matrix** is a matrix **which contains all the partial derivatives** of each output component with respect to each input variable, providing a full picture how the vector-valued function changes as each input variable changes.

5.6 Derivative: Key Point

- The **derivative** of a **univariate function** is a **scalar**,
  - When the **derivative** of a **multivariate function** is organized and stored in a **vector**, the so-called **gradient**.
    - we denote the **derivative of a multivariable function f** using the **gradient symbol** ***Δ*** **{read "del" or "nabla"}**
      - $\Delta f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \\ \vdots \end{bmatrix}$
- The **gradient is simply a vector listing** the **derivatives of a function** with respect to **each argument of a function**.

Thank You!!!