

Maths for Machine Learning
Revision notes

Table of Contents

Contents	Page no.
Equations of a straight line	3
Parallel and perpendicular lines, hyperplane, vector form of a hyperplane	4
Vectors, halfspaces	5
Transpose, dot product, unit vector	6
Distance between two points, norm, angle b/w two vectors, projection, intersection	7
Distance b/w : Hyperplane and origin, point and hyperplane, parallel hyperplanes Circle	8
Rotating coordinate axes, limit, function	9
Important functions for ML	10
Continuity, tangent and derivative	12
Finding optima	14
Partial derivative	16
Gradient descent	17
Variants of Gradient descent	18
Constrained optimization, Method of Lagrange multipliers	19
Eigen vector and Eigen value, Principal component analysis	20

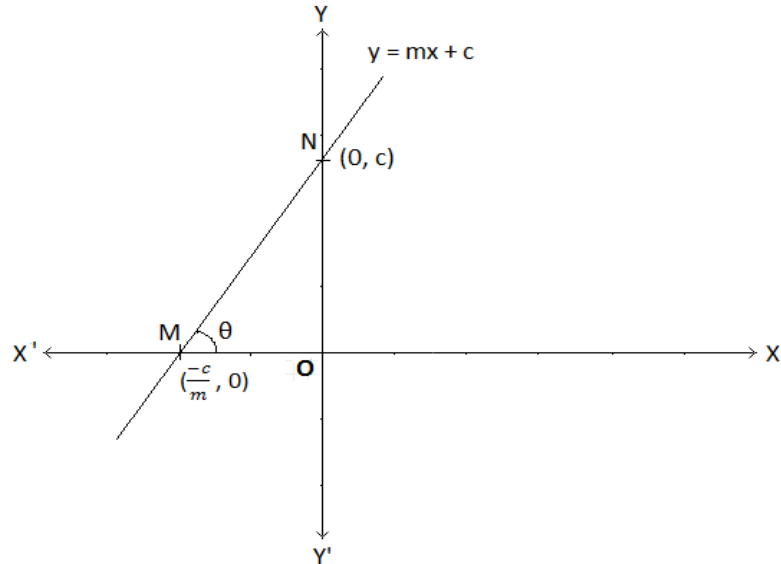
- **Equations of a straight line:**

1. **Slope-intercept form** of a straight line is given as:

$$y = mx + c$$

Where, m is the slope of the line and c is the y-intercept.

And $m = \tan\theta$ where θ is the angle which the line makes with the positive x-axis.



2. **Point-slope form:**

$$y - y_1 = m(x - x_1)$$

where m is the slope of the line and (x_1, y_1) are the coordinates of a point on the line.

3. **Two-point form:**

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

Where, (x_1, y_1) and (x_2, y_2) are the coordinates of two points on the line.

4. **Intercept form:**

$$\frac{x}{a} + \frac{y}{b} = 1$$

Where a and b are the intercepts of the line on the x-axis and y-axis respectively.

5. General form:

$$ax + by + c = 0$$

where a , b , and c are real numbers.

- Two lines are called **parallel** to each other if the values of the slope are equal.

Let's consider two lines $y = m_1x + c_1$ and $y = m_2x + c_2$.

The above two lines are parallel if $m_1 = m_2$.

The above two lines are **Perpendicular** to each other if:

$$m_1 = -\frac{1}{m_2}$$

- Hyperplane** is a linear surface in n -dimensions.

The general equation of a hyperplane is given as:

$$w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n + w_0 = 0$$

Where, $w_1, w_2, w_3, \dots, w_n$ are called the **weights/coefficients** and

$x_1, x_2, x_3, \dots, x_n$ are the **features**.

The equation of a **plane** in **3-D** is given as:

$$w_1x + w_2y + w_3z + w_0 = 0$$

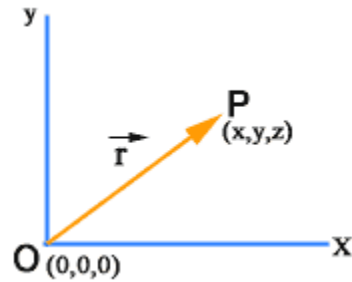
- Vector form of a hyperplane is:**

$$w^T x + w_0 = 0$$

$$\text{Where, } w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} \quad \text{and,} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

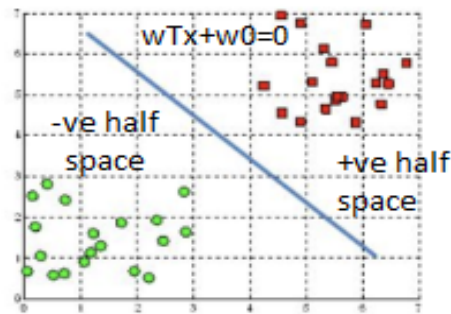
- **Vectors** can be interpreted as coordinates as well as a line segment from the origin to the coordinate.

For example, the below-given vector \vec{r} can be considered as coordinates of point $P(x,y,z)$ as well as a line segment from the origin to point $P(x,y,z)$.



- **Half Spaces:** In geometry, a half-space is either of the two parts into which a plane divides the three-dimensional Euclidean space.

Example: Let's assume that a hyperplane $w^T x + w_0 = 0$ is classifying the data points of two different classes in a space.



Let's say we got a point x_0 in the space.

Now, **if:**

$$w^T x_0 + w_0 > 0 \quad \Rightarrow \quad \text{the point is in the +ve halfspace}$$

$$w^T x_0 + w_0 < 0 \quad \Rightarrow \quad \text{the point is in the -ve halfspace.}$$

- The **transpose** operation changes a column vector into a row vector and vice versa.
For example,

$$\text{if } \vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix} \quad \text{then, } \vec{a}^T = [a_1 \ a_2 \ a_3 \ \dots \ a_n]$$

- **Dot product** of two vectors \vec{a} and \vec{b} is given as :

$$\vec{a} \cdot \vec{b} = a_1b_1 + a_2b_2 + a_3b_3 + \dots + a_nb_n$$

$$\text{Where, } \vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix} \quad \text{and } \vec{b} = [b_1 \ b_2 \ b_3 \ \dots \ b_n]$$

$$\text{Also, } \vec{a} \cdot \vec{b} = \vec{b} \cdot \vec{a}$$

Geometrically, it is the product of the magnitudes of the two vectors and the cosine of the angle between them.

$$\text{i.e. } \vec{a} \cdot \vec{b} = |\vec{a}| \cdot |\vec{b}| \cdot \cos(\theta)$$

where θ is the angle between the two vectors.

If the dot product of two vectors is **zero**, then the vectors are **perpendicular** to each other.

- **Unit vector** is a vector that has a magnitude of 1.

To convert a vector \vec{u} into a unit vector, we divide the vector by its magnitude.

$$\text{i.e. } \text{unit vector} = \hat{u} = \frac{\vec{u}}{||\vec{u}||}$$

- We can multiply any scalar value with the unit vector to get the desired magnitude (equal to that scalar value) in the same direction.
- All vectors with the same unit vector are **parallel**

- **Distance between two points** having coordinates (x_1, y_1) and (x_2, y_2) in an x-y plane is given as:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- **Norm or Magnitude** of a vector is calculated by taking the square root of dot product with itself.

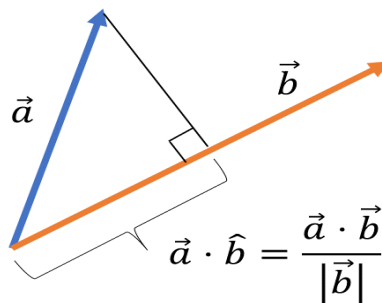
i.e. $||\vec{a}|| = \sqrt{\vec{a} \cdot \vec{a}}$

It represents the **length** of a vector or **distance** of \vec{a} coordinate from the origin.

- **Angle between two vectors** is given as :

$$\theta = \cos^{-1} \left(\frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||} \right)$$

- **Projection of vector \vec{a} on \vec{b}** = $\frac{\vec{a} \cdot \vec{b}}{||\vec{b}||}$



- At the **point of intersection** of two lines, both lines will have the same coordinates.

Example: Let's say we have two lines, $y = x+2$ and $y = 2x+1$.

We need to find the point of intersection of these two lines.

We assume that the lines intersect at a single point (a,b) .

Therefore, this point will satisfy both the line's equations.

i.e. $b = a+2$ — i)

$b = 2a+1$ — ii)

Solving above two equations, we get, $a = 1$ and $b = 3$.

Therefore, the given two lines intersect at the point (1,3).

- **Distance of a Hyperplane from the origin:**

Let's assume a hyperplane $\vec{w}^T \vec{x} + w_0 = 0$.

Its distance from the origin is given as: $d = \frac{w_0}{||\vec{w}||}$

- **Distance of a point \vec{x}_0 from a hyperplane** is given as:

$$d = \frac{|w^T x_0 + w_0|}{||w||}$$

i.e. Just put the point in the hyperplane's equation and divide by the square root of the summation of coefficients' square (or norm of the w vector)

- **Distance between two parallel hyperplanes**

Given two parallel hyperplanes, $w^T x + w_0 = 0$ and $w^T x + w_1 = 0$,

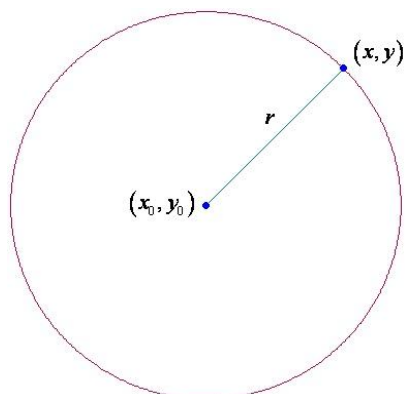
Distance between them is given as:

$$d = \frac{|w_1 - w_0|}{||w||}$$

- The **equation of a circle** in the x-y plane is given as:

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

where, (x_0, y_0) are the coordinates of the center of the circle and r is the radius of the circle.

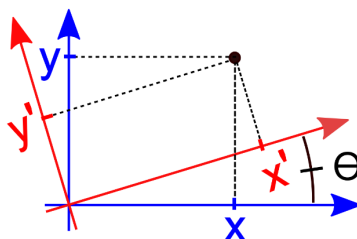


Therefore, a circle with the **center at origin** is given as:

$$x^2 + y^2 = r^2$$

Points **inside** the circle give **negative** values when substituted in the circle equation and points **outside** the circle give **positive** values.

- Let's say we have a coordinate system x-y initially and a point P(x₀, y₀) in it this system.



If the coordinate system is **rotated** by an **angle** θ in the anti-clockwise direction, then the Coordinates of point P with respect to the new coordinate system will be:

$$x_0' = x_0 \cos \theta + y_0 \sin \theta \quad \text{and} \quad y_0' = -x_0 \sin \theta + y_0 \cos \theta$$

- A **limit** is a value toward which an expression converges as one or more variables approach certain values. It is denoted as:

$$L = \lim_{x \rightarrow a} f(x)$$

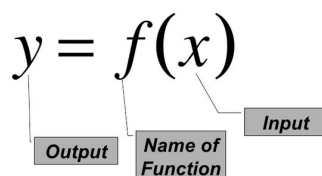
A **left-hand limit** means the limit of a function as it approaches from the left-hand side. It is denoted as:

$$LHL = \lim_{x \rightarrow a^-} f(x)$$

On the other hand, a **right-hand limit** means the limit of a function as it approaches from the right-hand side.

$$RHL = \lim_{x \rightarrow a^+} f(x)$$

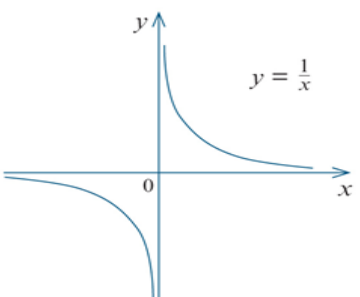
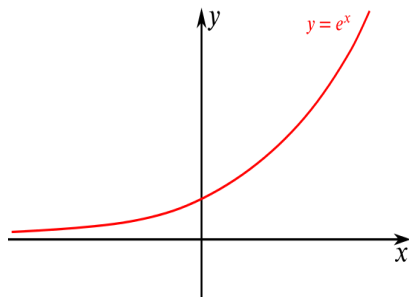
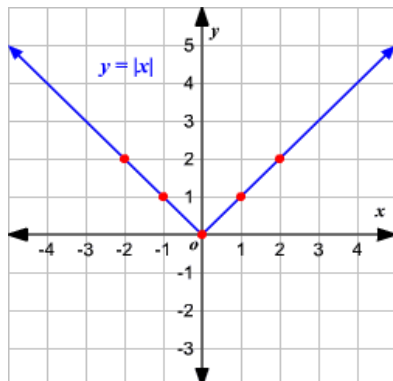
- A **Function** is a relationship between inputs and outputs where each input is related to exactly one output.

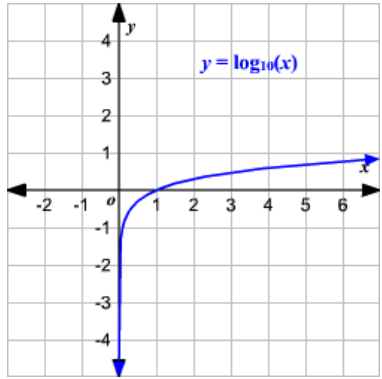
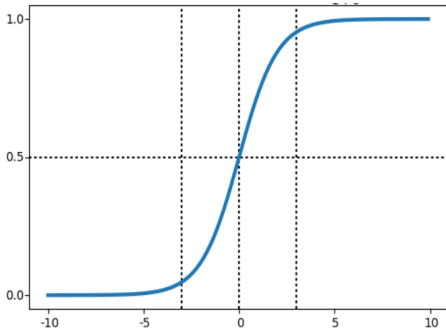
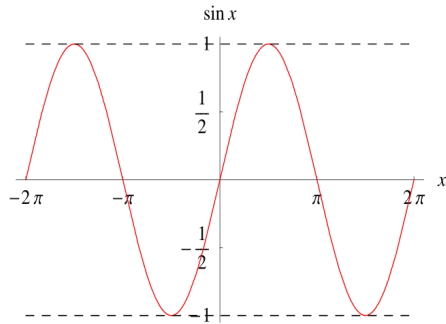
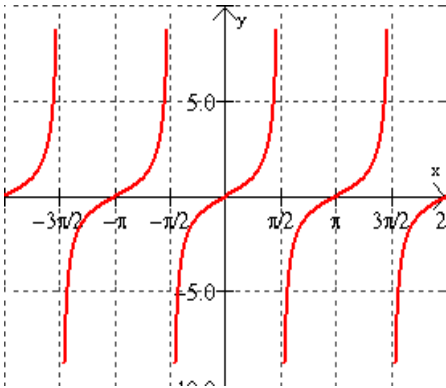


The **domain** of a function is the set of input values for f , in which the function is real and defined.

The set of all the outputs of a function is known as the **range** of the function.

- **Some important functions for Machine Learning:**

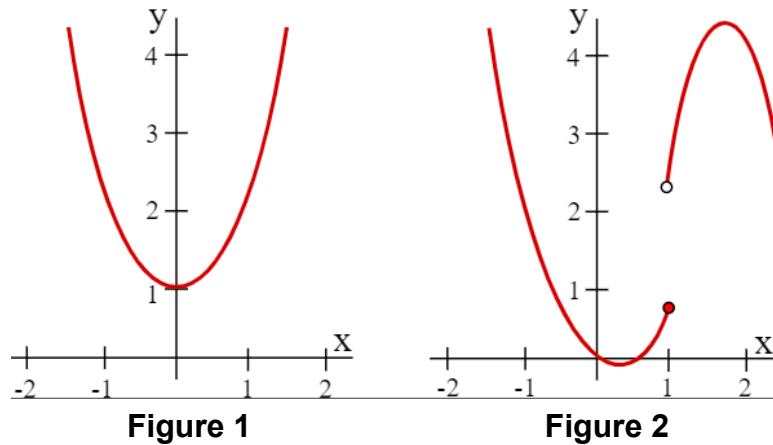
Function	Name	Domain	Range	Plot
$f(x) = \frac{1}{x}$	Hyperbola	\mathbb{R}	\mathbb{R}	
$f(x) = e^x$	Exponent	\mathbb{R}	\mathbb{R}^+	
$f(x) = x $	Modulus	\mathbb{R}	\mathbb{R}^+	

$f(x) = \log(x)$	Exponential	\mathbb{R}^+	\mathbb{R}	
$f(x) = \frac{1}{1 + e^{-x}}$	Sigmoid	\mathbb{R}	$(-1, 1)$	
$f(x) = \sin(x)$	sine	\mathbb{R}	$[-1, 1]$	
$f(x) = \tan(x)$	tangent	$\mathbb{R} - (2n + 1)\frac{\pi}{2}$	\mathbb{R}	

- $f(x)$ is **continuous** at a point $x = a$,

$$\text{if } \lim_{x \rightarrow a^+} f(x) = \lim_{x \rightarrow a^-} f(x) = \lim_{x \rightarrow a} f(x)$$

Example:



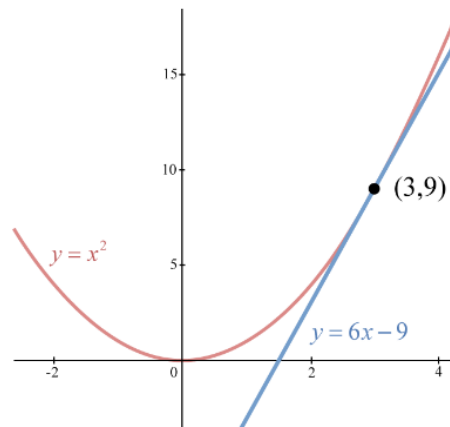
1. Figure 1 is continuous everywhere in its domain.
2. In figure 2, at $x=1$,

$$\lim_{x \rightarrow 1^-} f(x) \neq \lim_{x \rightarrow 1^+} f(x)$$

Therefore, the function given in figure 2 is not continuous at $x=1$.

- **Tangent** is a straight line that touches a graph only at one point.

Example:



In the above-given graph, line $y = 6x - 9$ is a tangent line to the curve at the point (3,9).

- The rate of change of a function with respect to a variable is called the **derivative** of the function with respect to that variable.

- **Derivative** of a function $f(x)$ with respect to variable x is denoted as:

$$f'(x) = \frac{df(x)}{dx}$$

Differentiation using the first principles:

The derivative of a function $f(x)$ at a point $x = a$ is given as:

$$f'(a) = \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h}$$

- **Common derivatives:**

$$\frac{d}{dx} c = 0 \quad \text{Constant Rule}$$

$$\frac{d}{dx} x^n = nx^{n-1} \quad \text{Power Rule}$$

$$\frac{d}{dx} \sin(x) = \cos(x) \quad \text{Trigonometric Rules}$$

$$\frac{d}{dx} \cos(x) = -\sin(x)$$

$$\frac{d}{dx} b^x = b^x \ln(b) \quad \text{Exponential Rule}$$

$$\frac{d}{dx} \ln(x) = \frac{1}{x} \quad \text{Logarithmic Rule}$$

- **Rules of differentiation:**

1. **Sum\Difference rule:** $\frac{d}{dx}[f(x) \pm g(x)] = \frac{d}{dx}f(x) \pm \frac{d}{dx}g(x)$

2. **Constant multiple rule:** $\frac{d}{dx}[k \cdot f(x)] = k \cdot \frac{d}{dx}f(x)$

3. **Product rule:** $\frac{d}{dx}[f(x) \cdot g(x)] = f(x) \cdot \frac{d}{dx}g(x) + g(x) \cdot \frac{d}{dx}f(x)$

4. **Quotient rule:** $\frac{d}{dx} \left[\frac{f(x)}{g(x)} \right] = \frac{g(x) \cdot f'(x) - f(x) \cdot g'(x)}{[g(x)]^2}$

5. Chain rule: $\frac{d}{dx}f(g(x)) = f'(g(x)).g'(x)$

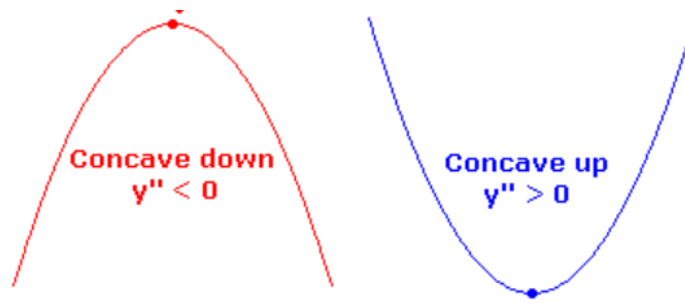
- The derivative of a function gives us the **slope** of the **tangent** line to the function at any point on the graph.

If the derivative/slope of the tangent at a certain point is **positive**, then the function is **increasing**.

If the derivative/slope of the tangent at a certain point is **negative**, then the function is **decreasing**.

If the slope of the tangent is **zero**, then the function is neither decreasing nor increasing at that point.

- The **Second derivative** of a function represents its concavity.
If the second derivative is **positive**, then the function is **concave upwards**.
If the second derivative is **negative**, then the function is **concave downwards**.



- Steps to find the optima:**
 - Given a function $f(x)$, firstly calculate its derivative. i.e. $f'(x)$
 - Put $f'(x) = 0$ to obtain the stationary points $x = c$.
 - calculate $f''(x)$ at each stationary points $x = c$ (i.e $f''(c)$)
 - We get the following situations:
 - If $f''(c) > 0$, then $f(x)$ has a **minimum value** at $x = c$.
 - If $f''(c) < 0$, then $f(x)$ has a **maximum value** at $x = c$.
 - If $f''(c) = 0$, then $f(x)$ may or may not have a maxima or minima at $x = c$.

Example: Let's find the optima of the function $f(x) = 2x^2 - 4x + 1$

Step 1: Calculate the first derivative.

$$f'(x) = 4x - 4$$

Step 2: Put $f'(x) = 0$ to obtain the stationary points.

$$4x - 4 = 0 \Rightarrow x = 1$$

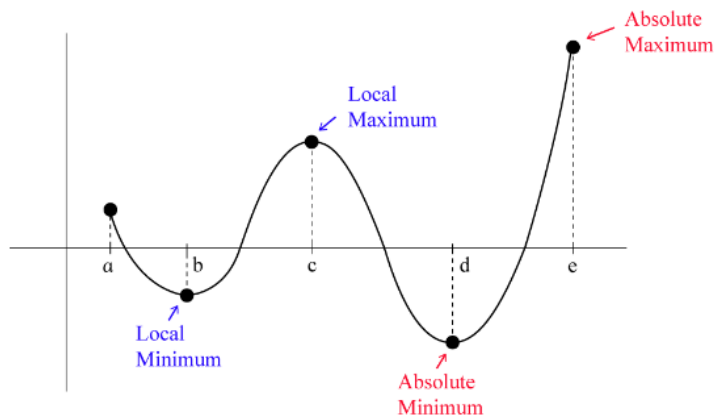
Step 3: Calculate $f''(1)$

$$f''(1) = 4 > 0$$

Step 4: Since, $f''(1) > 0$
therefore, there exists minima at $x = 1$.

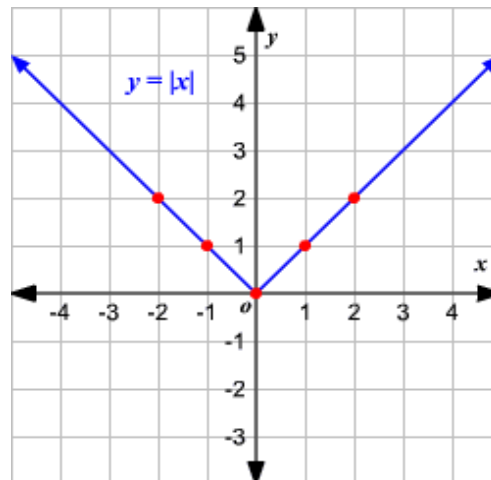
- **Local minima/maxima** can be defined as a point where the function has minimum/maximum value with respect to its vicinity/surrounding. We have marked it as Lmin and Lmax in the image above.

Global minima/maxima can be defined as the minimum/maximum value across the whole domain. It is also called **absolute maxima/minima**.



- A function $f(x)$ is said to be differentiable if it satisfies the following conditions:
 1. $f(x)$ should be **smooth** in its domain.
 2. $f(x)$ is **continuous** in its domain and
 3. $f'(x)$ is **continuous**.

Example: $f(x) = |x|$ is not differentiable at $x = 0$ as it has a sharp point (not smooth) at this point.



- We can also have functions that have more than one variable.

Example: $f(x, y) = x^2 + y^2$

A **partial derivative** of a function of several variables is its derivative with respect to one of those variables, with the others held constant.

For example, the partial derivatives of $f(x, y)$ with respect to x and y are given as:

$$\frac{\partial f}{\partial x} = 2x + 0 = 2x \quad \text{and} \quad \frac{\partial f}{\partial y} = 0 + 2y = 2y$$

- ∇ is called a **delta operator**. It is a 2D vector that consists of derivatives w.r.t single variables also called partial derivatives.

Let us assume a function $f(\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n)$ that has more than one input variable.

Then,

$$\nabla f(w_0, w_1, w_2, \dots, w_n) = \begin{bmatrix} \frac{\partial f}{\partial w_0} \\ \frac{\partial f}{\partial w_1} \\ \frac{\partial f}{\partial w_2} \\ \vdots \\ \frac{\partial f}{\partial w_n} \end{bmatrix}$$

Optima for f can be found by putting ∇f equal to a **Null matrix** of the same dimensions as ∇f .

We can have points where $\nabla f = 0$, but those points may not be maxima or minima. Those are called **Saddle points**.

- **Gradient descent** is an iterative algorithm to reach the optima of a function. Let's say we have a function $z = f(x, y)$ and are trying to find the minimum of this function.

We will start by initializing x_0 and y_0 randomly. Then we will keep updating x and y till the point where the partial derivatives are very close to 0 or some fixed number of iterations.

Algorithm:

Step 1: Initially, pick x_0 and y_0 randomly.

Step 2: Compute $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ at $x = x_0$ and $y = y_0$ respectively.

Step 3: The new values of x_0 and y_0 which are closer to the optima are given as:

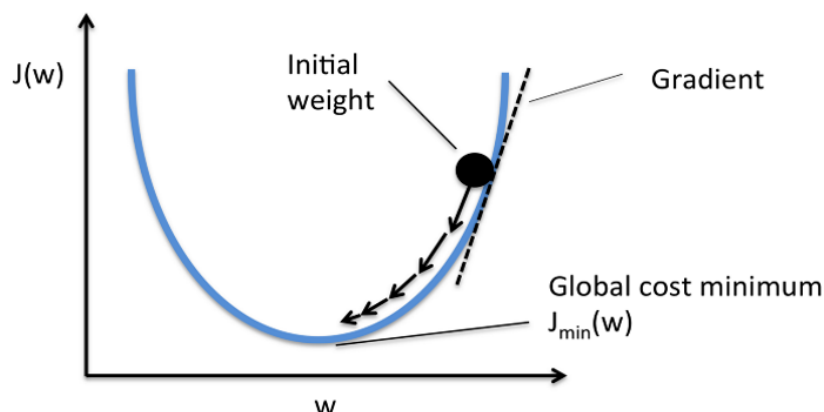
$$x_1 = x_0 - \eta \cdot \frac{\partial f}{\partial x} \quad \text{and} \quad y_1 = y_0 - \eta \cdot \frac{\partial f}{\partial y}$$

Step 4: Repeat the step 3 either till some k (constant) iterations or till a point where $\delta f / \delta x \approx 0$ and $\delta f / \delta y \approx 0$.

Here, η (eta) is the **learning rate** and decides the step size of our iterations. If we set its value to very small, then the updates will happen **very slowly**. If we set it to a large value, it may **overshoot** the minima.

Therefore, the **update rule** of the Gradient descent algorithm is:

$$x_{i+1} = x_i - \eta \frac{\partial f}{\partial x} \Big|_{x=x_i}$$



If we want to **maximize** some function. Then we can convert the maximum function into a minimum function by adding a **negative** sign..

i.e. $\max f(x, y) = \min -f(x, y)$

- **Variants of Gradient descent:**

Let's assume that we are minimizing a loss function (f) while training a model.

1. **Batch Gradient descent** calculates the partial derivative using the full training set at each step.

The update using Batch Gradient descent is given by:

$$\theta^{t+1} = \theta^t - \eta \sum_{i=1}^n \frac{\partial f(x_i)}{\partial \theta}$$

We use all the data points for one update, which leads to a high computation time if our dataset is very large.

2. **Mini - Batch Gradient descent** calculates the partial derivative using only a few data points from our data set randomly while performing many updates.

i.e.
$$\theta^{t+1} = \theta^t - \eta \sum_{i \in B} \frac{\partial f(x_i)}{\partial \theta}$$

where, B is a **random sample** of our data points.

We get very high-speed improvement while training our model with almost a similar accuracy.

3. **Stochastic Gradient descent** updates the parameters for each training example one by one.

i.e.
$$\theta^{t+1} = \theta^t - \eta \cdot \frac{\partial f(x_k)}{\partial \theta}$$

where, k is a random number from 1 to n .

It is comparatively faster than Batch GD but the number of updates needed to reach the minima is large.

- For a **constrained optimization** problem, we have an objective function that we are trying to optimize (say, $\min_{x,y} f(x, y)$) and this objective function will be subjected to some constraints.

The constraint may be an **equality constraint** ($g(x, y) = 0$) or we can also have **inequality constraints** like $g(x, y) < c$.

- The **method of Lagrange multipliers** is a method of finding the local minima or local maxima of a function subject to equality or inequality constraints.

We want to solve the problem, $x^*, y^* = \min_{x,y} f(x, y)$

subjected to the constraint $g(x, y) = c$

In order to solve the above problem, we **combine** both the constraint and the objective function. We can write the constraint as $g(x, y) - c$ and then rewrite our problem as:

$$x^*, y^* = \min_{x,y} f(x, y) + \lambda(g(x, y) - c) = L(x, y, \lambda)$$

Here λ is called a **Lagrange multiplier** ($\lambda \geq 0$) and the function $L(x, y, \lambda)$ is called the **Lagrangian function**.

Example: $\min_{x,y} \sum_{i=1}^n -y_i(w^T x_i + w_0)$, subjected to the constraint $\|w\|^2 = 1$

We can rewrite the constraint as $\|w\|^2 - 1 = 0$.

Using Lagrange multiplier, we can convert it into an unconstrained optimization problem.

i.e.
$$L = \min_{x,y} \sum_{i=1}^n -y_i(w^T x_i + w_0) + \lambda(\|w\|^2 - 1), \quad \lambda \geq 0$$

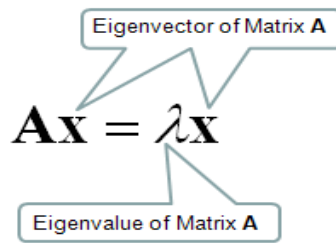
We can solve for the optimal value using the Gradient Descent algorithm.

- **Eigen vector and Eigen value:**

For any matrix **A**, there exists a vector \vec{x} such that when this vector is multiplied with the matrix A, we get a new vector in the same direction having a different magnitude.

The vector \vec{x} is called the **Eigen vector** and the length is called as **Eigen value**.

i.e.



The diagram shows the equation $\mathbf{Ax} = \lambda \mathbf{x}$. A callout box labeled "Eigenvector of Matrix A" points to the vector \mathbf{x} on both sides of the equation. Another callout box labeled "Eigenvalue of Matrix A" points to the scalar λ .

There can be multiple eigen vectors, which are always **orthogonal** to each other.

The eigenvector associated with the **largest eigenvalue** indicates the direction in which the data has the most variance.

Therefore, we can select our **principal components** in the direction of the eigenvectors having large eigenvalues and drop the principal components having relatively small eigenvalues.

- **Dimensionality reduction techniques** help to convert a high dimensional data to fewer dimensions which can be then visualized using simpler plots or it can be used when we want to preserve the information of our feature columns.

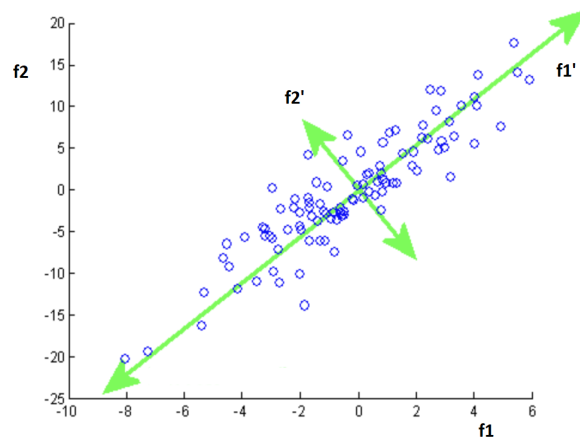
- **Principal Component Analysis (PCA)**

Principal component analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets.

PCA is an act of finding a new axis to represent the data so that a few principal components may contain most of the **information**.

The main **objective** here is whenever we are going from a higher dimension(d) to a lower dimension(d') we want to preserve those dimensions which have **high variance** (or high information.)

For example: Imagine we have two features f_1 and f_2 , and the data is spread as shown in the image below. We want to project the given 2-D data to 1-D.

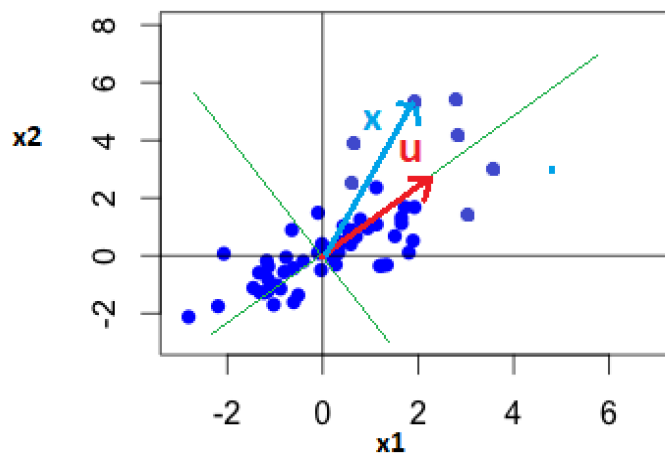


One thing we can observe here is that both axes have a good amount of variance. It will be difficult to decide which feature to drop.

Therefore, we rotated the entire coordinate axes. It is shown in green.

Let's call these new dimensions as f_1' and f_2' . Now we can see that variability on f_1' is more than f_2' and we can now drop the f_1' axis.

Consider a vector \mathbf{x} in the space representing one of the points in our data and a unit vector \mathbf{u} representing the direction of the new axis.



The projection of a vector \mathbf{x} on vector \mathbf{u} is given as: $\frac{\vec{u} \cdot \vec{x}}{\|\vec{u}\|}$

The best \mathbf{u} will be where the summation of the length of projections of all such points (\mathbf{x}_i) on the vector \mathbf{u} is maximum. This is equivalent to maximizing the variance of new coordinates along \mathbf{u} .

i.e.
$$\max_{\vec{u}} \frac{1}{n} \sum_{i=1}^n \frac{\vec{u} \cdot \vec{x}_i}{\|\vec{u}\|}$$

Variance Explained

Assume we have N columns. when we perform PCA, we are attempting to reduce N to n such that $n < N$.

When choosing n, an important consideration is the amount of variance it explains of the original set.

For example:

If $N = 3$, then,

- With the first 3 Principal components, we'll be getting 100% variance explained.
- With the first 2 Principal components, we'll be getting 90% variance explained, and
- The first 3 PCs would capture the 60% variance.