



Deep Learning

Maths Refresher

Tuesday 29th November 2019

Dr. Nicholas Cummins



- Linear Algebra
- Probability
- Differential Calculus
- Gradient Descent



- **Linear Algebra**
- Probability
- Differential Calculus
- Gradient Descent

Why Linear Algebra?

- It is a key **foundation** to the field of machine learning
 - Present from notations used to describe the operation of algorithms to the implementation of algorithms in code
- Also needed to **understanding** the calculus and statistics used in machine learning
- Enables a deeper **intuition** in algorithms
 - Implement algorithms from scratch
 - Devise new algorithms

- **Scalar**

- A one-dimensional vector, i.e. 1×1

- **Vector**

- A single-dimensional array of numbers
- i.e. $1 \times m$

- **Matrix**

- A two-dimensional array of numbers
- An $m \times n$ matrix has m rows and n columns

- **Tensor**

- A multidimensional array of numbers

Scalar:

a

Vector:

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$$

Matrix:

n columns \xrightarrow{j}

m rows $\downarrow i$

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ a_{m1} & \dots & \dots & a_{mn} \end{bmatrix}$$

- **Norm of a vector**

- The norm is a measure of magnitude
 - The l^p norm is given by:

$$l^p = (|x_1|^p + |x_2|^p + \dots + |x_n|^p)^{1/p}$$

- Machine learning generally uses the l^1 and l^2 norms
 - The least squares cost function is the l^2 norm of an error vector
 - Norm of a parameter vector can be used in regularization

- **Norm of a vector**

- l^1 norm example

$$v = \begin{bmatrix} 1 \\ -4 \\ 5 \end{bmatrix}, \|v\|_1 = |1| + |-4| + |5| = 10$$

- l^2 norm example

$$v = \begin{bmatrix} 1 \\ -4 \\ 5 \end{bmatrix}, \|v\|_2 = \sqrt{|1|^2 + |-4|^2 + |5|^2} = \sqrt{42}$$

- **Dot product**

- The dot product of two vectors, $v_1 \in \mathbb{R}^{n \times 1}$ and $v_2 \in \mathbb{R}^{n \times 1}$, is the sum of the product of the corresponding elements:

$$v_1 \cdot v_2 = v_1^T v_2 = v_2^T v_1 = v_{1_1} v_{2_1} + v_{1_2} v_{2_2} + \cdots + v_{1_n} v_{2_n} = \sum_{k=1}^n v_{1_k} v_{2_k}$$

- **Dot product**

- Example:

$$v_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, v_2 = \begin{bmatrix} 3 \\ 5 \\ -1 \end{bmatrix}$$

$$v_1 \cdot v_2 = v_1^T v_2 = 1 \times 3 + 2 \times 5 - 3 \times 1 = 10$$

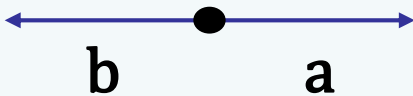
- **Why is the dot product important?**
 - The dot product encodes information about the angle between two vectors

$$\mathbf{v}_1 \cdot \mathbf{v}_1 = \|\mathbf{v}_1\| \|\mathbf{v}_1\| \cos \theta$$

$$\theta = \arccos \left(\frac{\mathbf{v}_1 \cdot \mathbf{v}_1}{\|\mathbf{v}_1\| \|\mathbf{v}_1\|} \right)$$

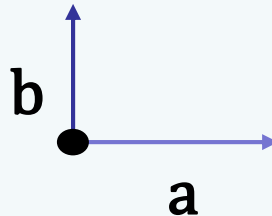
- Why is the dot product important?

$$\cos \pi = -1$$



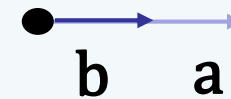
Dot product is negative

$$\cos \frac{\pi}{2} = 0$$



Dot product is zero

$$\cos 0 = 1$$



Dot product is positive

– The dot product measures how similar two vectors are

- **Matrix multiplication**

- For $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$ to be multipliable n must equal p and the resulting matrix is $C \in \mathbb{R}^{m \times q}$

$$C_{i,j} = \sum_{k=1}^n a_{i,k} b_{k,j}$$

$$\forall i \in \{1, 2, \dots, m\}$$

$$\forall j \in \{1, 2, \dots, q\}$$

- **Matrix multiplication**

– Example:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}, C = A \times B$$

$$C_{11} = \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 5 \\ 7 \end{bmatrix}, 1 \times 5 + 2 \times 7 = 19, C_{12} = \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 6 \\ 8 \end{bmatrix}, 1 \times 6 + 2 \times 8 = 22$$

$$C_{21} = \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} 5 \\ 7 \end{bmatrix}, 3 \times 5 + 4 \times 7 = 43, C_{22} = \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} 6 \\ 8 \end{bmatrix}, 3 \times 6 + 4 \times 8 = 50$$

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

- **Matrix transpose**

- The transpose of a matrix $A \in \mathbb{R}^{m \times n}$ is generally represented as $A^T \in \mathbb{R}^{n \times m}$
- This is performed by transposing the column vectors as row vectors

$$a'_{ji} = a_{i,j}, \forall i \in \{1, 2, \dots, m\}, \forall j \in \{1, 2, \dots, n\}$$

$$\text{where } a'_{ji} \in A^T \text{ and } a_{i,j} \in A$$

- Transpose is important as it relates in inner products
 - The dot product of two column vectors a and b can be computed as the single entry of the matrix product: $[a \cdot b] = a^T b$

- **Matrix transpose**

- Examples:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ then } A^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 4 & 3 \\ 8 & 2 & 6 \\ 7 & 8 & 3 \\ 4 & 9 & 6 \\ 7 & 8 & 1 \end{bmatrix} \text{ then } A^T = \begin{bmatrix} 1 & 8 & 7 & 4 & 7 \\ 4 & 2 & 8 & 9 & 8 \\ 3 & 6 & 3 & 6 & 1 \end{bmatrix}$$

- **Determinant of a matrix**

- The determinant of a square matrix $A \in \mathbb{R}^{n \times n}$ is a number denoted by $|A|$ or $\det(A)$ and is given by:

$$\det(A) = \pm \prod a_{1j_1} a_{2j_2}, \dots, a_{nj_n}$$

- where the column indices j_1, j_2, \dots, j_n are taken from the set $\{1, 2, \dots, n\}$, with no repetitions allowed. The plus (minus) sign is taken if the permutation (j_1, j_2, \dots, j_n) is even (odd)

- **Determinant of a matrix**

- Examples:

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

$$= a_{11}(a_{22}a_{33} - a_{32}a_{23}) - a_{21}(a_{12}a_{33} - a_{32}a_{13}) + a_{31}(a_{12}a_{23} - a_{22}a_{13})$$

- **Inverse of a matrix**

- For a square matrix $A \in \mathbb{R}^{n \times n}$, the inverse is denoted as A^{-1} and produces the identity when multiplied by A

$$AA^{-1} = A^{-1}A$$

$$A^{-1} = \frac{(\text{cofactor matrix of } A)^T}{\det(A)}$$

- Cofactor for $a_{i,j} = (-1)^{i+j} d_{ij}$
 - Where d_{ij} is the determinate of the matrix formed by deleting row i and column j from A

- **Inverse of a matrix**

- Examples

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{1}{a_{11} \times a_{22} - a_{21} \times a_{12}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

$$\begin{bmatrix} 4 & 7 \\ 2 & 6 \end{bmatrix}^{-1} = \frac{1}{4 \times 6 - 2 \times 7} \begin{bmatrix} 6 & -7 \\ -2 & 4 \end{bmatrix} = \frac{1}{10} \begin{bmatrix} 6 & -7 \\ -2 & 4 \end{bmatrix} = \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.4 \end{bmatrix}$$

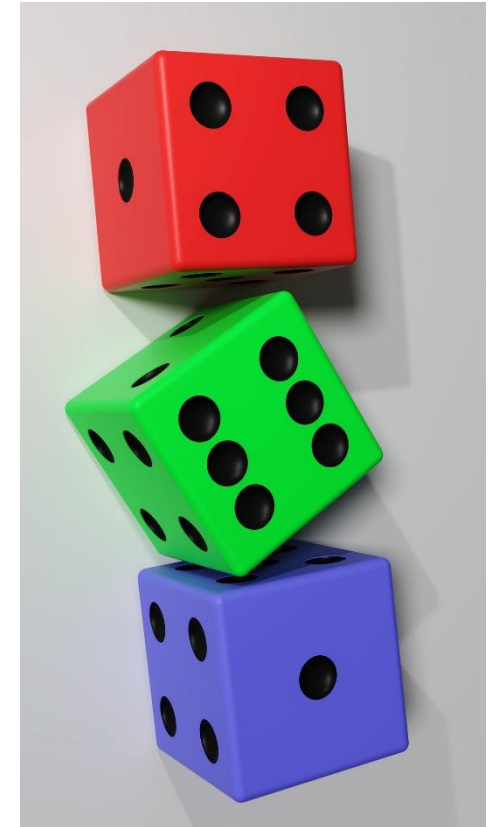
$$\begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix}^{-1} = \frac{1}{3 \times 8 - 6 \times 4} \begin{bmatrix} 8 & -4 \\ -6 & 3 \end{bmatrix} \rightarrow \text{inverse does not exist}$$



- Linear Algebra
- **Probability**
- Differential Calculus
- Gradient Descent

- **Why probability?**

- Machine learning must always deal with uncertain quantities
- Laws of probability govern how a machine learning algorithm should reason
 - We design machine learning algorithms to approximate expressions derived from probability theory
- Analyse the behaviour of a proposed approach



- **Definitions**

- **Experiment:** any process of observation
- **Random experiment:** An experiment in which the outcomes cannot be precisely predicted
- **Sample space:** set of all possible outcomes
- **Probability measure P :** an assignment of a number between 0 and 1 to a particular event in the sample space

$P(A)$: the probability that an event A will occur

$$0 \leq P(A) \leq 1$$

- **Rules of Probability**

- **Intersection of events**

- The probability that Events *A and B* occur, denoted $P(A \cap B)$

- **Mutually exclusive events**

- Cannot occur at the same time i.e. $P(A \cap B) = 0$

- **Union of events**

- The probability that events *A or B* occur, denoted $P(A \cup B)$

- **Conditional probability**

- The probability that Event *A* occurs, given that Event *B* has occurred
 - Denoted $P(A|B)$

- **Rules of probability**

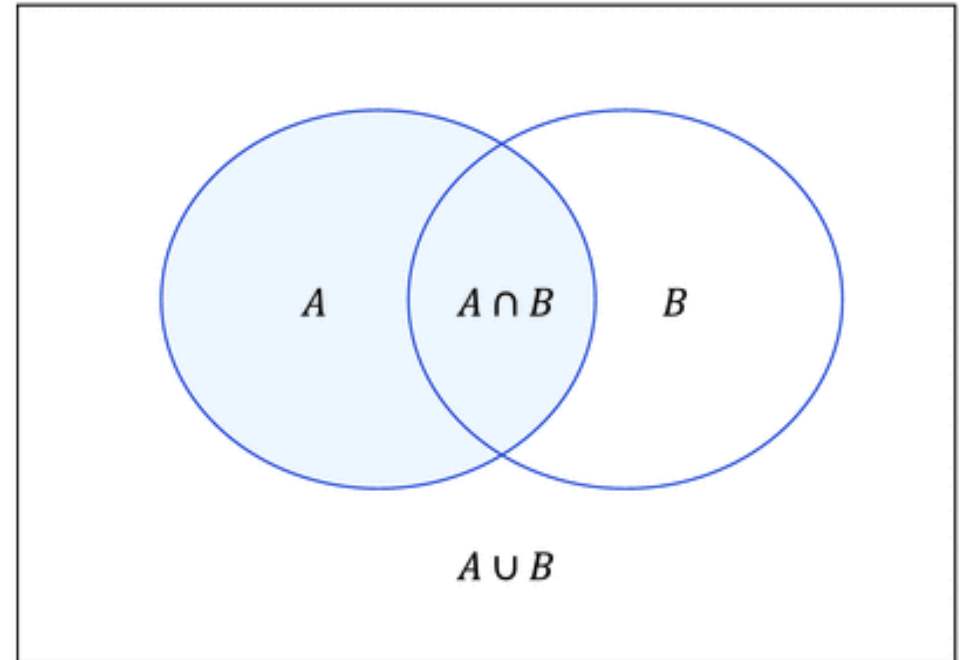
- **Rule of multiplication**

- $P(A \cap B) = P(A) P(B|A)$
 $= P(B)P(A|B)$

- Note, $P(A \cap B)$ is denoted as $P(AB)$

- **Rule of addition**

- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$



- **Rules of probability**

- **Chain rule of probability**

- Extension of the rule of multiplication

$$\begin{aligned} P(A_1 A_2 A_3 \dots A_n) &= P(A_1) P(A_2 | A_1) P(A_3 | A_1 A_2) P(A_n | A_1 A_2 A_3 \dots A_{n-1}) \\ &= P(A_1) \prod_{i=2}^n P(A_i | A_1 A_2 A_3 \dots A_{i-1}) \end{aligned}$$

- **Mutually exclusive events $P(AB) = 0$**

$$\begin{aligned} P(A_1 \cup A_2 \cup A_3 \cup \dots \cup A_n) &= P(A_1) + P(A_2) + P(A_3) + \dots + P(A_n) \\ &= \sum_{i=1}^n P(A_i) \end{aligned}$$

- **Rules of probability**

- **Independence of events**

$$P(AB) = P(A)P(B)$$

- **Bayes' rule**

- From multiplication rule

$$P(AB) = P(A)P(B|A) = P(B)P(A|B)$$

- Therefore

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

- **Random experiment**

- An experiment or a process for which the outcome cannot be predicted with absolute certainty
 - However, we have knowledge of the *sample space*, the set of all possible outcomes

- **Random variable**

- Individual outcomes of a random experiment
 - A function that maps the outcomes of random experiment (the sample space) to a subset of real numbers (i.e. \mathbb{R}).
 - E.g. A random variable can be used to describe the process of rolling a fair die and the possible outcomes $\{1, 2, 3, 4, 5, 6\}$

Probability (Refresher)

- **Probability Mass Function (PMF)**

- Let X be a random variable with domain D
- The probability mass function is then defined as the probability that X is equal to some value x

$$\sum_{x \in D} P(X = x) = 1$$

- To be a PMF, P must satisfy
 - P must be the sets of all possible states of X
 - $0 \leq P(X) \leq 1$
 - $\sum_{x \in D} P(X) = 1$

Probability (Refresher)

- **Expectation**

- The average value that some function takes when x is drawn from P

$$\mathbb{E}_{x \sim P}[f(x)] = \mu = \sum_x P(x)f(x)$$

- **Variance**

- Variation in different sample values of x when drawn from its probability distribution

$$\text{Var}[f(x)] = \sigma^2 = \mathbb{E}[f(x) - \mathbb{E}[f(x)]]^2$$

- **Covariance**

- Measure joint variability between two random variables

$$\text{Cov}(f(x), g(y)) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])(g(x) - \mathbb{E}[g(x)])]$$

- **Random processes**

- A collection of random variables defined over a common PMF

- Consider a random process η with N observed values

- **Mean**

$$\mu_{\eta} = \frac{1}{N} \sum_{n=0}^{N-1} \eta(n)$$

- **Mean-Square**

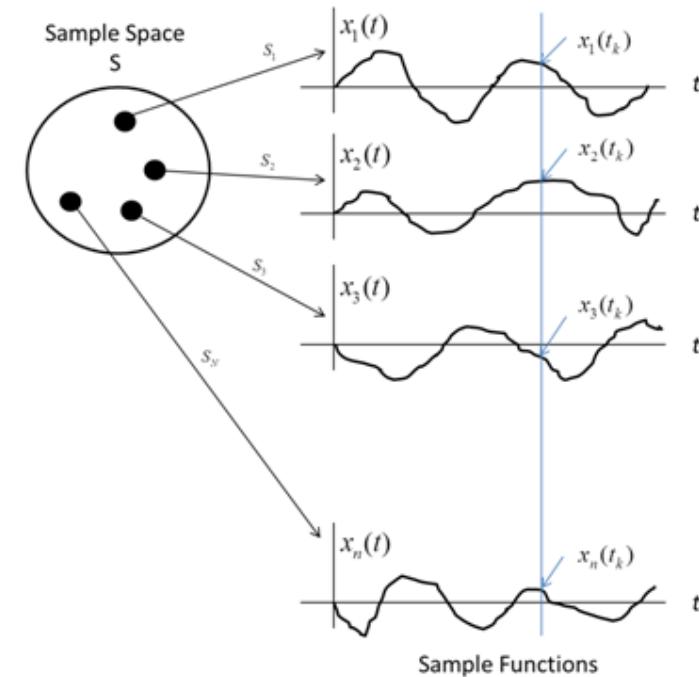
- The ‘power’ of the process

$$MS_{\eta} = \frac{1}{N} \sum_{n=0}^{N-1} (\eta(n))^2$$

- **Variance**

$$\sigma_{\eta}^2 = \frac{1}{N} \sum_{n=0}^{N-1} (\eta(n) - \mu_{\eta})^2$$

- **Random signal**
 - When the values of a random process η form a time series
 - Also known as a stochastic process
 - Denoted $\eta(t)$
- **Key properties**
 - μ_η represents the DC component
 - DC component is the amplitude signal fluctuates around
 - Assumed to be zero for random noise
 - MS_η represents the average power
 - If μ_η is zero $\sigma_\eta^2 = MS_\eta$



- **Information Theory**

- Quantifying how much information is present in a signal
 - Likely events should have low information content
 - Likely events are uninformative
 - Less likely events should have higher information content
 - Unlikely events are more informative

- **Self-Information:**

$$I(x) = -\log P(x)$$

- **Entropy:**

$$H(x) = E_{x \sim P}[I(x)] = -E_{x \sim P}[\log P(x)]$$

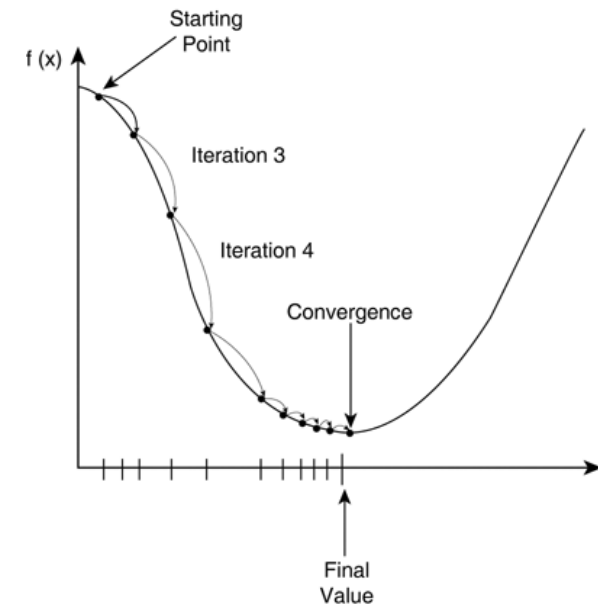
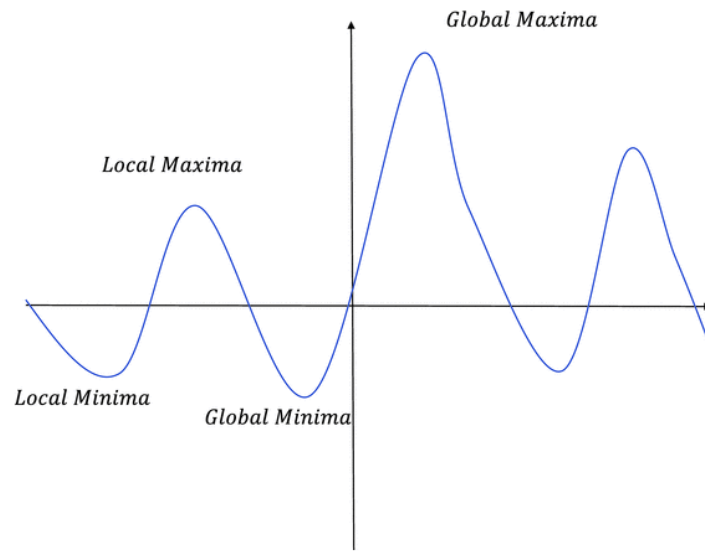
- Distribution of expected information



- Linear Algebra
- Probability
- **Differential Calculus**
- Gradient Descent

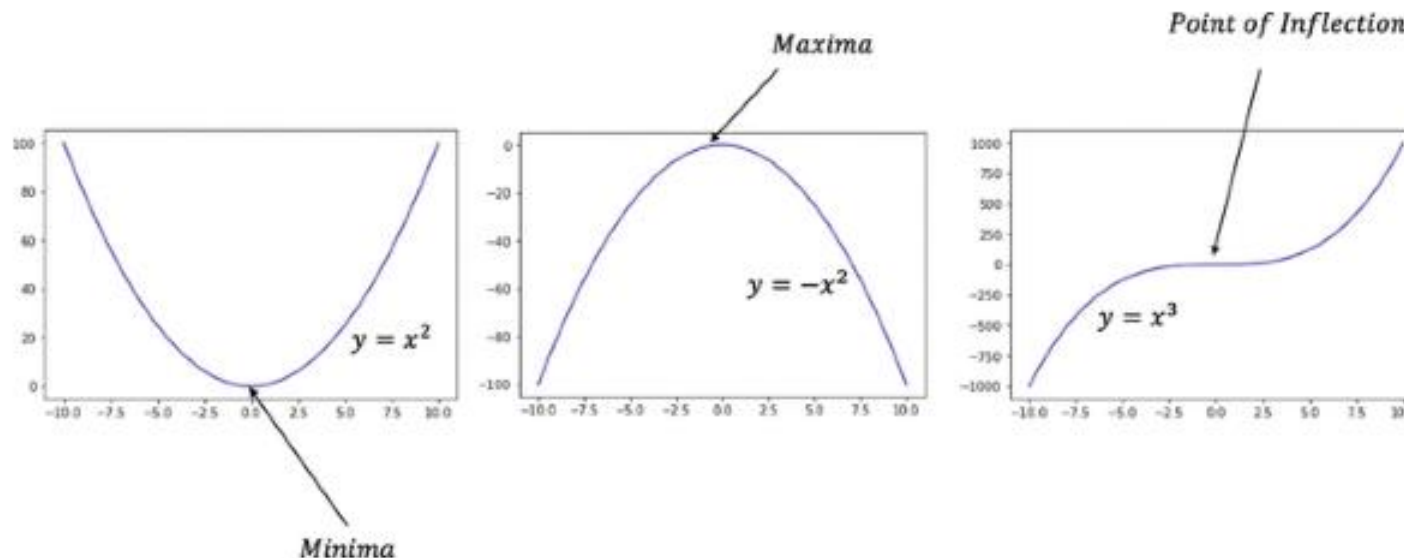
- **A good understanding of calculus is essential for machine learning**
 - Machine learning models are (normally) a function of several variables
 - In building a model we generally need to compute a cost function, we derive the models that best explain the training data by optimising this cost function
 - Optimisation refers to the task of minimising (or maximising) a function $f(\mathbf{x})$

- **Maxima and Minima of Functions**
 - Building machine-learning models relies on iteratively minimising a cost function



- **Rules for locating maxima/minima**

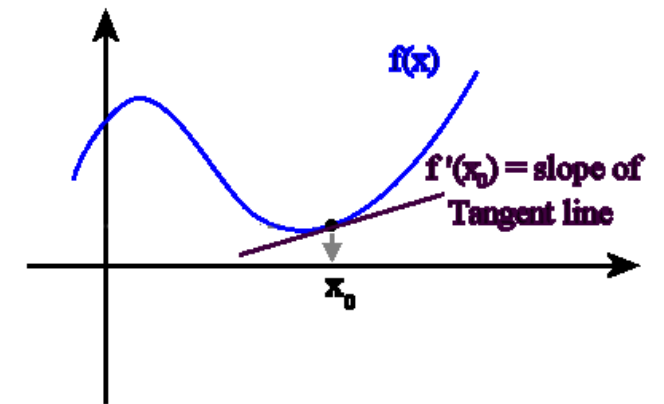
- 1st order derivative is zero
- Maxima: 2nd order derivative is *less* than zero
- Minima: 2nd order derivative is *greater* than zero
- Point of inflection: 2nd order derivative equals zero



- **Derivative of a function $f(t)$**

- Rate of change of a quantity represented by a function with respect to another quantity on which the function is dependent on.

$$\frac{df}{dt} = \lim_{h \rightarrow 0} \frac{f(t+h) - f(t)}{h}$$



- Derivative of a function $f(t)$

$f(x)$	$f'(x)$	$f(x)$	$f'(x)$
x^n	nx^{n-1}	e^x	e^x
$\ln(x)$	$1/x$	$\sin(x)$	$\cos(x)$
$\cos(x)$	$-\sin(x)$	$\tan(x)$	$\sec^2(x)$
$\cot(x)$	$-\operatorname{cosec}^2(x)$	$\sec(x)$	$\sec(x) \tan(x)$
$\operatorname{cosec}(x)$	$-\operatorname{cosec}(x) \cot(x)$	$\tan^{-1}(x)$	$1/(1+x^2)$
$\sin^{-1}(x)$	$1/\sqrt{1-x^2}$ for $ x < 1$	$\cos^{-1}(x)$	$-1/\sqrt{1-x^2}$ for $ x < 1$
$\sinh(x)$	$\cosh(x)$	$\cosh(x)$	$\sinh(x)$
$\tanh(x)$	$\operatorname{sech}^2(x)$	$\coth(x)$	$-\operatorname{cosech}^2(x)$
$\operatorname{sech}(x)$	$-\operatorname{sech}(x) \tanh(x)$	$\operatorname{cosech}(x)$	$-\operatorname{cosech}(x) \coth(x)$
$\sinh^{-1}(x)$	$1/\sqrt{x^2+1}$	$\cosh^{-1}(x)$	$1/\sqrt{x^2-1}$ for $x > 1$
$\tanh^{-1}(x)$	$1/(1-x^2)$ for $ x < 1$	$\coth^{-1}(x)$	$-1/(x^2-1)$ for $ x > 1$

Differential Calculus (Refresher)

- **Product Rule**

- If $f(x)$ and $g(x)$ are differentiable on x then:

$$(f \cdot g)'(x) = f(x)g'(x) + g(x)f'(x)$$

- **Chain Rule**

- If $f(x)$ and $g(x)$ are differentiable on x

$$(f \circ g)'(x) = f'(g(x))g'(x)$$

- If $y = g(u)$ and $u = g(x)$ the derivative of y is

$$\frac{dy}{dx} = \frac{dy}{du} \frac{du}{dx}$$

- **Partial derivatives** $z = f(x, y)$

$$\frac{\partial z}{\partial x} = \lim_{h \rightarrow 0} \frac{f(x + h, y) - f(x, y)}{h}$$

$$\frac{\partial z}{\partial y} = \lim_{h \rightarrow 0} \frac{f(y + h, x) - f(x, y)}{h}$$

- Successive Partial Derivatives

$$\frac{\partial}{\partial y} \left(\frac{\partial z}{\partial x} \right) = \frac{\partial^2 z}{\partial y \partial x}$$

$$\frac{\partial}{\partial x} \left(\frac{\partial z}{\partial y} \right) = \frac{\partial^2 z}{\partial x \partial y}$$

- Note that if the second derivatives are continuous, $\frac{\partial^2 z}{\partial y \partial x} = \frac{\partial^2 z}{\partial x \partial y}$

- **Gradient of a function**

- Vector of first order partial derivatives

$$f(\mathbf{x}), \text{ where } \mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^{n \times 1}$$

Then,

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]^T$$

- The gradient is important in machine-learning algorithms when we try to maximize or minimize cost functions with respect to the model parameters,

- **Hessian Matrix of a function**
 - Matrix of second order partial derivatives
 - Useful in optimisation problems
 - Especially when cost function is non linear

For a function: $f(x, y, z)$:

$$Hf(x, y, z) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2} \end{bmatrix}$$

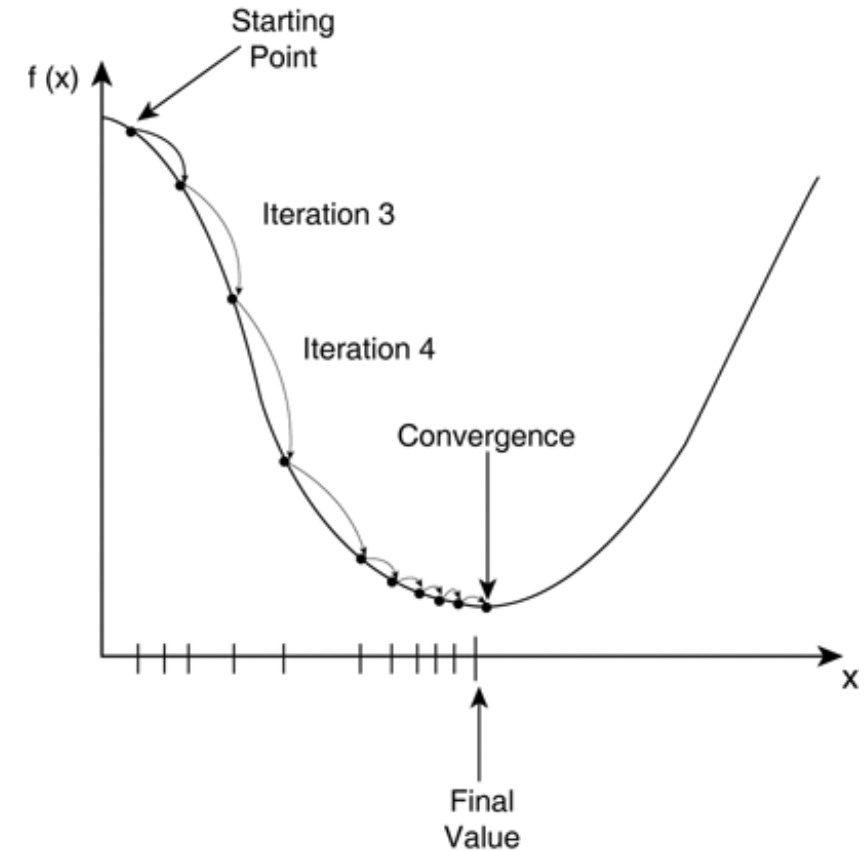


- Linear Algebra
- Probability
- Differential Calculus
- **Gradient Descent**

- **Gradient Descent Algorithms**

- Arguably the most widely used optimisation technique
- Iterative solution
 - Uses the negative gradient of the cost function to determine the direction they parameters need updating

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla C(\theta^{(t)})$$

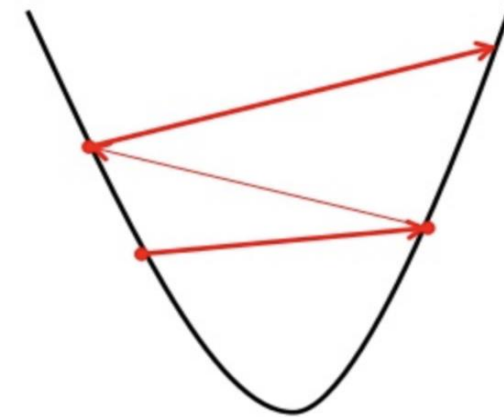


- **Gradient Descent Algorithms**

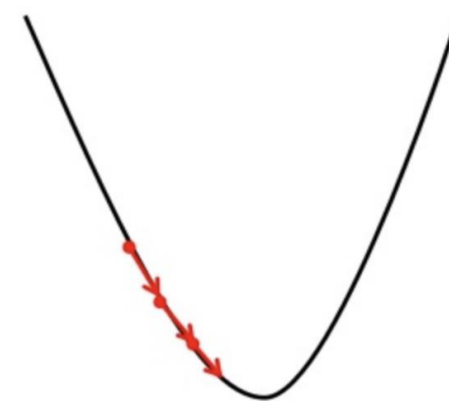
$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla C(\theta^{(t)})$$

- η is the learning rate
- A constant that defines the size of the gradient descent step
- Size is important:
 - Too large and the update function might oscillate over the minima
 - Too small and convergence is slow

Big learning rate



Small learning rate



- **Multivariate Gradient Descent Algorithms**

- Lets consider a cost function $C(\theta)$ where $\theta \in \mathbb{R}^{n \times 1}$
- At every iteration we want to update θ to $\theta + \Delta\theta$ such that $C(\theta + \Delta\theta)$ is less than $C(\theta)$
- Achieved by assuming linearity and using a *Taylor series expansion* we get:

$$C(\theta + \Delta\theta) = C(\theta) + \Delta\theta^T \nabla C(\theta)$$

- Need to choose $\Delta\theta$ such that $C(\theta + \Delta\theta)$ is less than $C(\theta)$

- **Multivariate Gradient Descent Algorithms**

- Need to choose $\Delta\theta$ such that $C(\theta + \Delta\theta)$ is less than $C(\theta)$

$$C(\theta + \Delta\theta) = C(\theta) + \Delta\theta^T \nabla C(\theta)$$

- To get the minimum value of the dot product $\Delta\theta^T \nabla C(\theta)$, the direction of $\Delta\theta$ should be the opposite of $\nabla C(\theta)$

$$\Delta\theta \propto -\nabla C(\theta)$$

Hence

$$\Delta\theta = -\eta \nabla C(\theta)$$

$$\theta + \Delta\theta = \theta - \eta \nabla C(\theta)$$

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla C(\theta^{(t)})$$