# Probability

- Probability – a mathematical framework for representing uncertainty
- Multiple sources of uncertainty in Engineering and Science
  - Internal Randomness in system
    - Example: Quantum Mechanics
  - Incomplete data/observability
    - Macroscopic Description
  - Incomplete modeling
    - Weather models

# Dual use of Probability Ideas in Machine Learning

- Constructing Learning System
  - Incorporate Probabilistic algorithms by trying to mimic human reasoning about uncertainty
  - Probabilistic models
- Analyzing Learning System
  - Even deterministic learning systems are only correct part of the time. Their output can therefore be analyzed probabilitically
  - Probabilistic analysis of deterministic/probabilistic models

# Frequentist vs Bayesian

- Statement – 60% chance of rain tomorrow
- Two interpretation of probability
- Frequentist
  - Depends on proportion of event in infinite sample space
  - Objective measure
- Bayesian
  - Measure degree of belief
  - Subjective measure
- Mathematics of resulting probabilities works the same way
- P(Disease 1) = 0.1, P(Disease 2) =0.2, P(D1 & D2) = 0.02 if they are independent

# Definitions

- Random experiment: Experiment that results in different outcomes despite being seeming similar conditions
  - Example: tossing a coin, throwing a dice, rainfall amount
- Sample Space: Set of possible outcomes of a random experiment
  - Example: Tossing a coin, S = {H,T}
  - The sample space we choose depends on the purpose of analysis
  - Example: Diameter of a manufactured pipe, S should be
  - $S = \mathbb{R}^+ = \{x \,|x > 0\}\; OR$
  - $S = \{low, medium, high\}\; OR$
  - $S = \{satisfactory, unsatisfactory\}$

# Random Variables

- Useful to denote outcomes of random experiments by number
- Can be done by categorical outcomes
- The variables that associates a number with an outcome of a random experiment is called a random variable
- **Notation:** The random variable is denoted by a capital letter (e.g X) and the value is denoted by a small letter (e.g. x).
  - Example: the rainfall on a particular day is a random variable R.
  - We can ask “What is the probability that the rainfall is greater than 10 mm?” by a mathematical notation P(R>10) = ?

# Probability Distribution

- A probability distribution tells us how likely a random variable is to take each of its possible states
- Discrete Random Variable (DRV)
  - Has finite (or countably infinite) range
  - Example – No. of typographical errors, no. of diagnostic errors, etc
  - Probability measured by Probability Mass Function (PMF)
- Continuous Random Variable (CRV)
  - Has real interval for its range
  - Example: Temperature, Pressure, Voltage, Height, Current, etc
  - Probability measured by Probability Density Function (PDF)

# Probability Mass Function

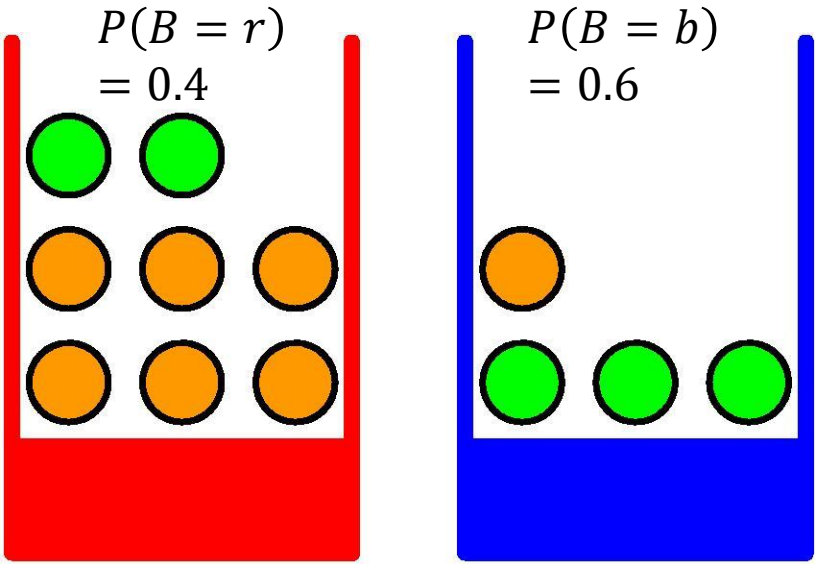
- Discrete Variable -> Probability Mass Function (PMF)
  - PMF – list of possible values along with their probabilities
  - Example
  - X: Number that comes up on throw of a biased dice
  - P(X=1)=0.1, P(X=2) =0.1, P(X=3) =0.2
  - P(X=4)=0.2, P(X=5) =0.2, P(X=6) =0.2

- To be a PMF for a random variable X, a function P satisfies
  - Domain of P is the set of all possible states of X
  - $0 \leq P(\boldsymbol{X} = \boldsymbol{x}) \leq 1$
  - $\sum_{x \epsilon X} P(X = x) = 1$
- Uniform Random variable: $\mathrm{P}(\mathrm{X} = x_i) = \frac{1}{k}$
- Analogous to point load

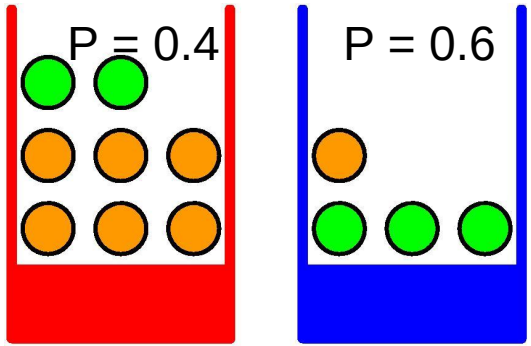
# Probability Density Function

- Continuous Variable -> Probability Density Function (PDF)
  - PDF-Probability density. In 1D, p(x) is probability “per unit length”
  - Like a distributed load
  - Example
  - R: amount of rainfall
  - $\mathrm{P}(10 < \mathrm{R} < 20) \equiv \mathrm{P}(10 \leq R \leq 20) = \int_{10}^{20} p(x)dx$
- To be a PDF for a continuous variable X, a function P satisfies
  - Domain of P is the set of all possible states of X
  - $\forall x \in X, 0 \leq p(\boldsymbol{x})$. Note that it is not necessary for $p(x) \leq 1$
  - $\int_X p(x)dx = 1$
- Normalized histogram approximates a probability density function

# A simple orienting example



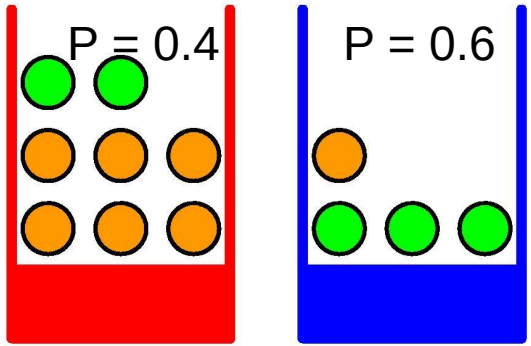
# A simple orienting example




Q1: What is the probability of picking an orange?

Q2: What is the probability of picking a red basket given that the fruit picked is an orange?

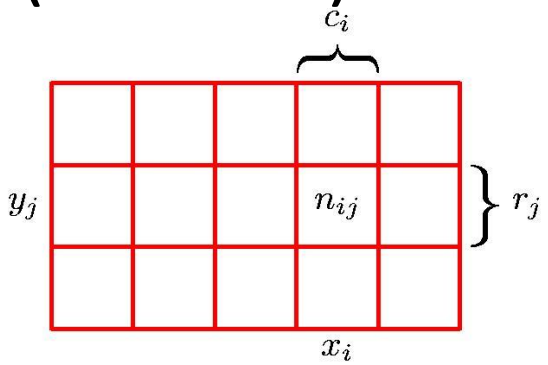
# A simple orienting example




| | F = o | F= a | Total |
|---|---|---|---|
| **B = r** | 30 | 10 | 40 |
| **B = b** | 15 | 45 | 60 |
| | 45 | 55 | |

# Joint Probability (Discrete)

| | F = o | F= a | Total |
|---|---|---|---|
| B = r | 30 | 10 | 40 |
| B = b | 15 | 45 | 60 |
| | 45 | 55 | |




Joint probability

The probability that X will take the value $x_i$ and Y will take the value $y_j$

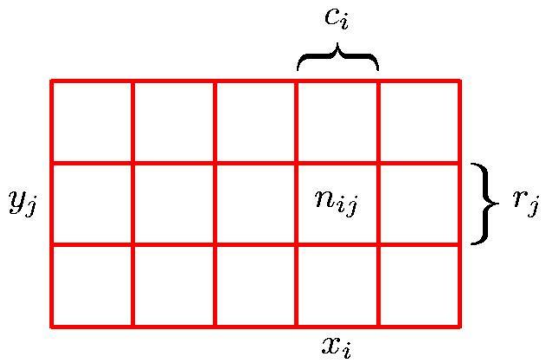
$$P(X = x_i, Y = y_j)$$

Let the number of trials that $X = x_i \; and \; Y = y_j$ be $n_{ij}$

Then, $P(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$

# Sum Rule

| | F = o | F= a | Total |
|---|---|---|---|
| B = r | 30 | 10 | 40 |
| B = b | 15 | 45 | 60 |
| | 45 | 55 | |




Let number of trials that $X = x_i$ be $c_i$

Then, $P(X = x_i) = \frac{c_i}{N}$

Marginal probability

However, $c_i = \sum_j n_{ij}$

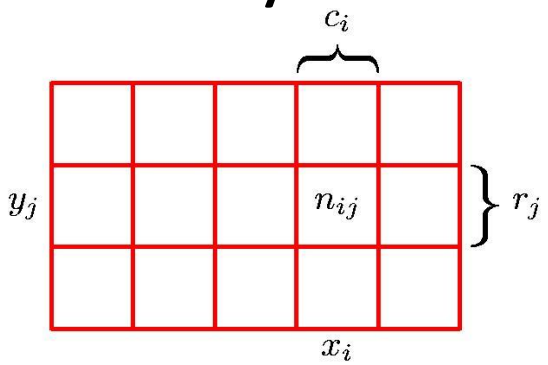
$$\Rightarrow P(X = x_i) = \sum_j \frac{n_{ij}}{N}$$

Sum rule of probability

$$\Rightarrow P(X = x_i) = \sum_j P(X = x_i, Y = y_j)$$

# Conditional Probability

| | F = o | F= a | Total |
|---|---|---|---|
| B = r | 30 | 10 | 40 |
| B = b | 15 | 45 | 60 |
| | 45 | 55 | |




Conditional probability

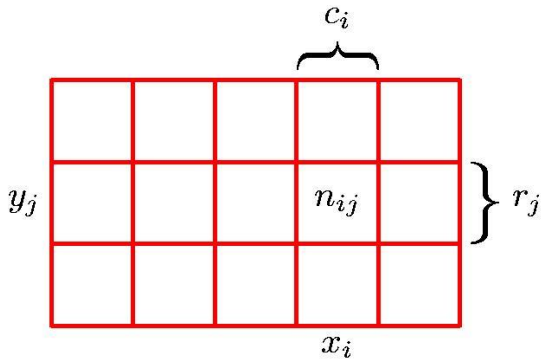
The probability that Y will take the value $y_j$ **given that** X will take the value $x_i$

$$P(Y = y_j | X = x_i)$$

$$P(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

# Product Rule

| | F = o | F= a | Total |
|---|---|---|---|
| B = r | 30 | 10 | 40 |
| B = b | 15 | 45 | 60 |
| | 45 | 55 | |




$$P(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

$$P(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

$$\Rightarrow P(X = x_i, Y = y_j) = P(Y = y_j | X = x_i) P(X = x_i)$$

Product rule of probability

# Rules of Probability (Simplified notation)

$$P(X = x_i) = \sum_j P\left(X = x_i, Y = y_j\right)$$

Sum rule of probability

$$P\left(X = x_i, Y = y_j\right) = P\left(Y = y_j \middle| X = x_i\right) P(X = x_i)$$

Product rule of probability

**Simplified Notation**

| | |
|---|---|
| **Sum Rule** | $P(X) = \sum_Y P(X, Y)$ |
| **Product Rule** | $P(X, Y) = P(Y \mid X) P(X)$ |

# Bayes' Theorem

**Product Rule** $P(X,Y) = P(Y|X)P(X)$

Similarly, $P(Y,X) = P(X|Y)P(Y)$

Since $P(X,Y) = P(Y,X)$ we obtain that

$$P(Y|X)P(X) = P(X|Y)P(Y)$$

So, $P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$

**Bayes' Theorem**

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

# Bayes Theorem -- Likelihood, Prior, Posterior

Bayes' Theorem

$$P(Y|X) = P(X|Y)\frac{P(Y)}{P(X)}$$

$$\Rightarrow P(Y|X)\ \alpha\ P(X|Y)P(Y)$$

$\Rightarrow$ Posterior $\alpha$ Likelihood × Prior

# Example – Cancer diagnosis

Suppose a person goes to a cancer diagnosis centre for a test and the test is positive (i.e. says the person has cancer).

1. What questions must the person ask to determine the accuracy of the diagnosis?

2. What are the chances the person actually has cancer?

# Cancer diagnosis

1. *What questions must the person ask to determine the accuracy of the diagnosis?*

- What percentage of the people with cancer test positive?
  - Ans : 99%
- What percentage of people without cancer test negative?
  - Ans : 99%
- What percentage of the population has cancer?
  - Ans : 0.5%

- Think : What are random variables in this problem?

# Cancer diagnosis

*2. What are the chances the person testing positive actually has cancer?*

Random variables

State of disease D : {C, NC}

Result of test T : {+, -}

Given

$$P(+ \mid \mathrm{C}) = 0.99$$

$$P(-|NC) = 0.99$$

$$P(C) = 0.005$$

Question : What is $P(C \mid +)$ ?

# Cancer diagnosis – Bayes' Theorem

$$P(C|+) = \frac{P(+|C)P(C)}{P(+)}$$

$$\Rightarrow P(C|+) = \frac{P(+|C)P(C)}{P(+|C)P(C) + P(+|NC)P(NC)}$$

$$\Rightarrow P(C|+) = \frac{0.99 \times 0.005}{0.99 \times 0.005 + (1 - P(-|NC) \times 0.995}$$

$$= \frac{0.00495}{.00495 + 0.00995} = 0.33$$

# Cancer diagnosis – With numbers

$$P(C|+) = \frac{\text{No. of people with cancer testing positive}}{\text{No of people testing positive}}$$

Consider a population of 10,000 people who go to the test

People with cancer is 0.5%, that is 0.005*10000 = 50

Out of these, $0.99 * 50 \approx 50$ test positive

People without cancer is 9,950

Out of these, $0.01 * 9950 \approx 100$ test positive.

So, $P(C|+) = \frac{50}{50+100} \approx 0.33$

# Independence

- **Independent random variables** – Two random variables ***X*** and ***Y*** are said to be *statistically independent* if and only if

$$p(x, y) = p(x)p(y)$$

- More precisely, X and *Y* are independent iff

$$p(X = x, Y = y) = p(X = x)p(Y = y) \quad \forall x \in X, y \in Y$$

  - Examples
    - Independent – X: Throw of a dice, Y: Toss of a coin
    - Not independent – X: Height, Y: Weight
- Independence is equivalent to saying

$$p(y|x) = p(y) \text{ OR } p(x|y) = p(x)$$

- Can be seen from product rule $p(x, y) = p(y|x)p(x) = p(x)p(y)$

$$\Rightarrow p(y|x) = p(y)$$

# Conditional Independence

- Two random variables ***X*** and ***Y*** are said to be *independent given Z* if and only if

$$p(x, y \mid z) = p(x|z)p(y|z)$$

- More precisely, *X* and *Y* are independent given Z iff

$$p(X = x, Y = y \mid Z = z) = p(X = x \mid Z = z)p(Y = y|Z = z) \quad \forall x \in X, y \in Y, z \in Z$$

  - Examples
    - *Ind & Cond Ind* – **X**: Throw of a dice, **Y**: Toss of a coin, **Z** : Card from deck
    - *Not Ind* BUT *Cond Ind* – **X**: Height, **Y**: Vocabulary, **Z** : Age
    - *Ind* BUT *Cond Not Ind* – **X**: Dice Throw 1, **Y**: Dice Throw 2, **Z** : Sum of dice

- Denoted by $(x \perp y) \mid z$

# Conditional Independence – Chain rule

- Recall -- $p(x, y) = p(x|y)p(y)$

Consider $p(x, y, z) = p(x, a)$ where $a$ is the event $(y, z)$

$$\Rightarrow p(x, y, z) = p(x, a) = p(x|a)p(a)$$
$$= p(x|a)p(y, z)$$
$$= p(x|a)p(y|z)p(z)$$
$$= p(x|y, z)p(y|z)p(z)$$
$$= p(z)p(y|z)p(x|y, z)$$

- In general,
- $P\left(x^{(1)}, x^{(2)}, \dots, x^{(n)}\right) = P\left(x^{(1)}\right)P\left(x^{(2)}\middle|x^{(1)}\right) \dots P\left(x^{(n)}\middle|x^{(1)}, \dots, x^{(n-1)}\right)$

i.e.

$$P\left(x^{(1)}, x^{(2)}, \dots, x^{(n)}\right) = P\left(x^{(1)}\right)\Pi_{i=2}^{n}P\left(x^{(i)}\middle|x^{(1)}, \dots, x^{(i-1)}\right)$$

Chain rule of conditional probability

# One context for conditional probabilities

- Images may be thought of as a collection of pixels $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$
- The probability of a particular image may be thought of as joint probability

$$P\left(x^{(1)}, x^{(2)}, \ldots, x^{(n)}\right)$$

- Chain rule along with conditional independence can be used to estimate probabilities of the occurrence of images

# Optimization and Machine Learning

- The basic idea behind much of Machine learning is to build models that map input data to output data.
- We derive parameters of these models based on “training” on some given data.
- We can see the whole of machine learning as a process that finds the best/optimal Model for the given data.
- Most machine learning problems are, therefore, ultimately optimization problems
- We will now review and introduce some optimization techniques

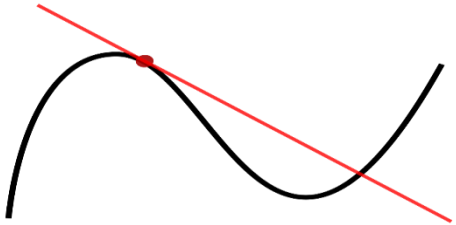
# Optimization

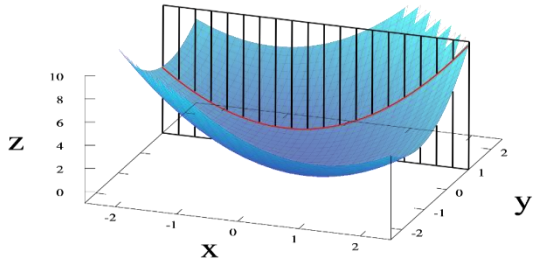
- The general optimization task is to maximize or minimize a function $f(\mathbf{x})$ by varying $\mathbf{x}$.
  - The function $f(\mathbf{x})$ is called the objective function or cost function or loss function
  - The function $f(\mathbf{x})$ maybe a scalar (single objective) or a vector (multi-objective)
  - In this course (and most of Machine Learning) we deal only with a single objective. That is, $f(\mathbf{x})$ is a scalar.
  - However, $\mathbf{x}$ is, in general, a vector.
  - Therefore, $f: \mathbb{R}^n \rightarrow \mathbb{R}$
  - For example, $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2$. Here, $f: \mathbb{R}^3 \rightarrow \mathbb{R}$

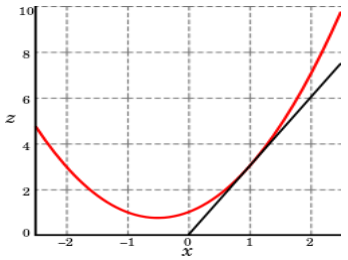
- It is possible to reduce all optimization problems to minimization problems.
  - That is, all problems can be written as find $\mathbf{x}$ that minimizes $f(\mathbf{x})$
  - Any maximization problem can be written as minimization of -$f(\mathbf{x})$

- We denote the solution to the problem as $\mathbf{x}^* = \arg\min f(\mathbf{x})$

# Derivatives





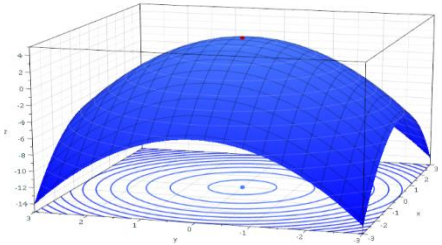

- Derivatives measure how one much quantity changes when there is a small change in another
- Geometrically, in one dimension, this can be given as the slope of the tangent

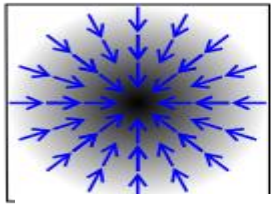
$$f'(a) = \frac{df}{dx}(x = a) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

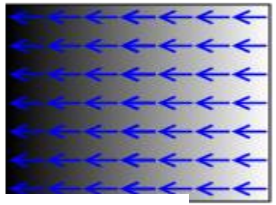
- In higher dimensions (functions of many variables/vectors), we have the idea of partial derivatives

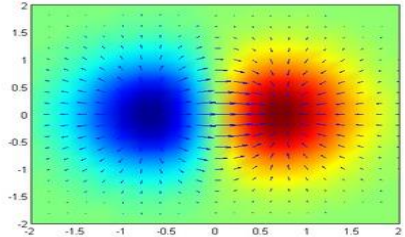
$$\frac{\partial f}{\partial x_i}(a_1, \dots, a_n) = \lim_{h \to 0} \frac{f(a_1, \dots, a_i + h, \dots, a_n) - f(a_1, \dots, a_i, \dots, a_n)}{h}$$

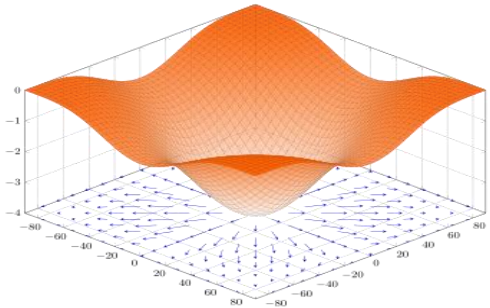
# Gradient











- The gradient is the multivariable generalization of the derivative
- It is a vector the components of which denote the partial derivatives in each direction

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

- It can be used to calculate the directional partial derivative of $f$ along the direction $\hat{\boldsymbol{v}}$

$$D_v f(\mathbf{x}) = \lim_{\alpha \to 0} \frac{\partial f(\mathbf{x} + \alpha \hat{\boldsymbol{v}})}{\partial \alpha} = \nabla_{\mathbf{x}} f(\mathbf{x}) \cdot \hat{\boldsymbol{v}}$$

# Hessian

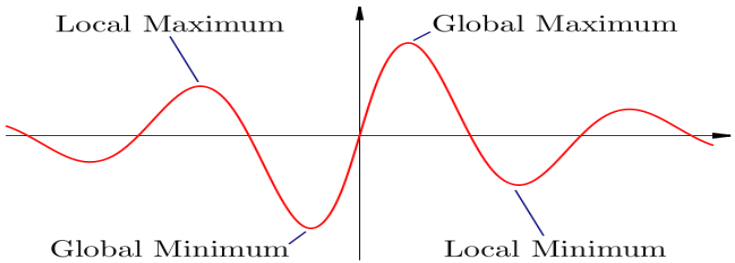
- The Hessian is the gradient of the gradient.
  - It is the equivalent of the second derivative in scalar calculus and has similar uses

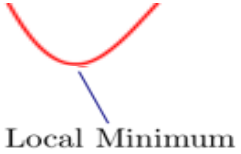
- For $f: \mathbb{R}^n \to \mathbb{R}$, we have $H_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$ is the Hessian which is a $n \times n$ matrix

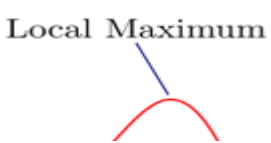
$$H_{i,j} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

- Note that the Hessian is a symmetric matrix

# Optimization – Scalar x



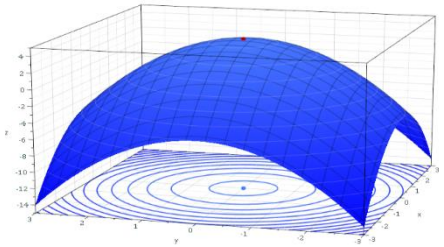



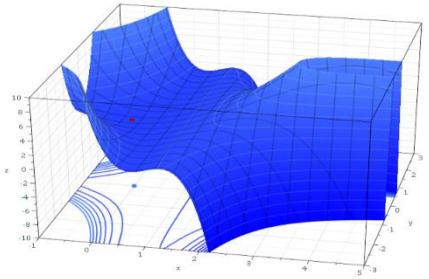


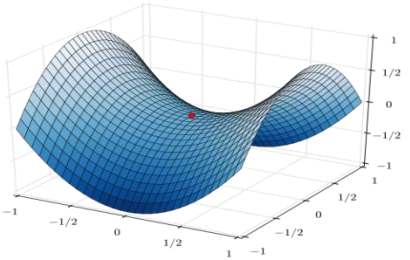
- We will look at the **unconstrained problem**. That is, find $x$ that minimizes $f(x)$ with $x \in \mathbb{R}$. That is, no constraints on $x$.
- It can be shown that any local extremum will have the property $f'(x) = 0$
  - Such points are called **stationary points** or **critical points**.
  - The stationary point may be a (local) minimum, maximum or saddle point

- If $f''(x) > 0$ , it is a local minimum
- If $f''(x) < 0$, it is a local maximum
- If $f''(x) = 0$, it could be a saddle point
- The absolute lowest/highest level of $f(x)$ is called the global maximum/minimum

# Optimization – Multivariate x







- In this case the **unconstrained optimization problem** is to find $\mathbf{x}$ that minimizes $f(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^n$. That is, there are no constraints on $x$.
- Since $\mathbf{x}$ is now a vector quantity, we need to evaluate the gradient $\frac{\partial f}{\partial \mathbf{x}} \equiv \nabla_x f$
  - It can be shown that any local extremum will have the property $\nabla_{\mathbf{x}} f = \mathbf{0}$
  - Such points are called **stationary points** or **critical points**.
  - The stationary point may be a (local) minimum, maximum or saddle point
- The type of critical point is decided by the nature of the Hessian $H_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$
- If $H_{i,j}$ is positive definite it is a local minimum
- If $H_{i,j}$ is negative definite it is a local maximum
- If $H_{i,j}$ is indefinite (i.e. neither p.d or n.d) then it is a saddle point

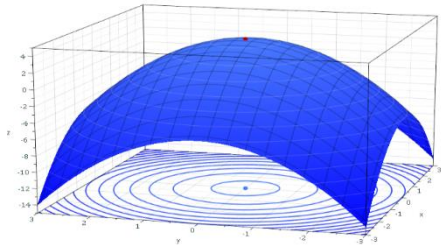
# Constrained Optimization

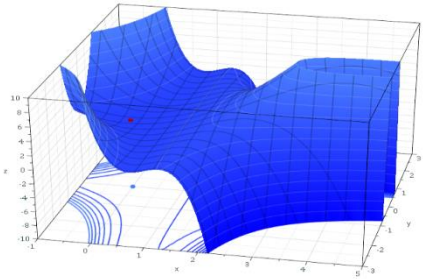
- The general constrained optimization task is to maximize or minimize a function f(x) by varying x given certain constraints on x
  - For example, find minimum of $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2$. where $\|x\|_2 \geq 1$
- Very common to encounter this in engineering practice
  - For example, designing the fastest vehicle with a constraint on fuel efficiency
- All constraints can be converted to two types of constraints
  - Equality constraints – e.g. Minimize $f(x_1, x_2, x_3)$ subject to $x_1 + x_2 + x_3 = 1$.
  - Inequality constraints - e.g Minimize $f(x_1, x_2, x_3)$ subject to $x_1 + x_2 + x_3 < 1$.
- Canonical Form – All optimization problems can be written as
  - Minimize f(x) subject to the constraint that $x \in \mathbb{S}$
  - $\mathbb{S} = \{x \mid \forall i, g^{(i)}(x) = 0 \; and \; \forall j, h^{(j)}(x) \leq 0\}$

# Generalized Lagrange Function

- The constrained optimization problem requires us to minimize the function while ensuring that the point discovered belongs to the feasible set
- There are several techniques that achieve this but it is, in general a difficult problem
- A very common approach is to define a new function called the generalized Lagrangian
  - $L(x, l, a) \quad = f(x) + \sum_i l_i g^{(i)}(x) + \sum_j a_j h^{(j)}(x)$
- Then the constrained minimum is given by
  - Min $_{x \in \mathbb{S}}$ f(x) = min$_x$ max$_l$ max$_{a,a>=0}$ L(x,l,a)

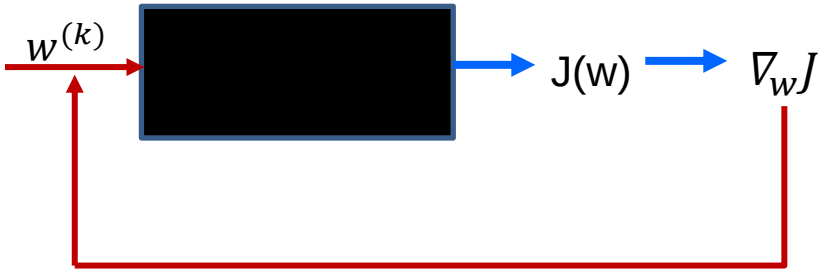
# Need for Numerical Optimization






- Optimization we saw so far was analytical.
- This requires explicit expressions for the objective function in terms of the features (variables).
  - Example : $J(\mathbf{w}) = w_1^2 + w_2^2 + w_3^2 + 4$
- However, usually we only know the function as a “black” box.
  - In machine learning this “black box” is our Machine Learning Model (e.g. Neural network)
- So, we have to develop numerical (rather than analytical techniques)

# Iterative optimization -- Fundamental idea

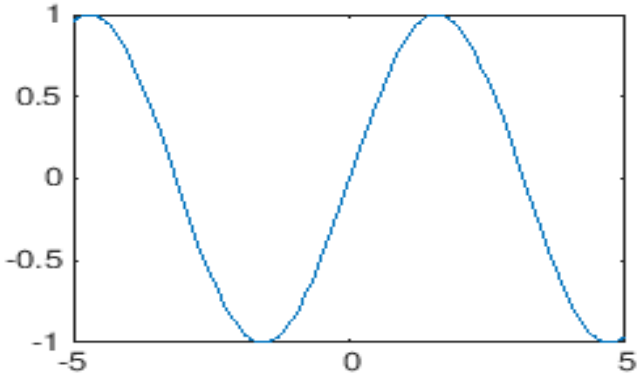



- We want to drive $\nabla_w J$ to $0$ but we do not have an analytical expression.

Iterative Process

- Guess for w
- Run through the black box and find value of J(w)
  - This value may be obtained through a program instead of an expression
- Find $\nabla_w J$
  - We will have to use methods for determining $\nabla_w J$ numerically
- If $\nabla_w J = 0$, we stop, else we need to take a new guess
  - More precisely, improve our guess
- A very common method for improving guess is called Gradient Descent

# Gradient Descent (Scalar case)

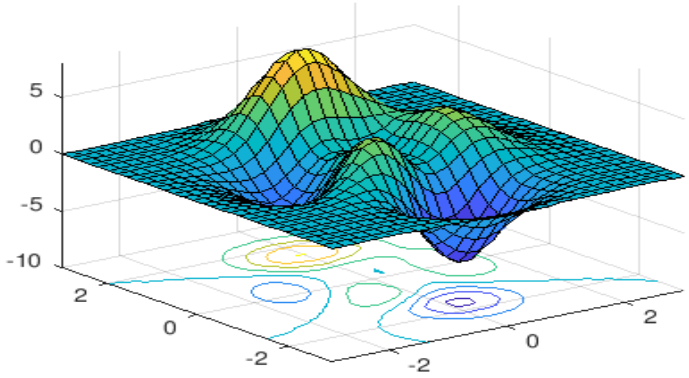


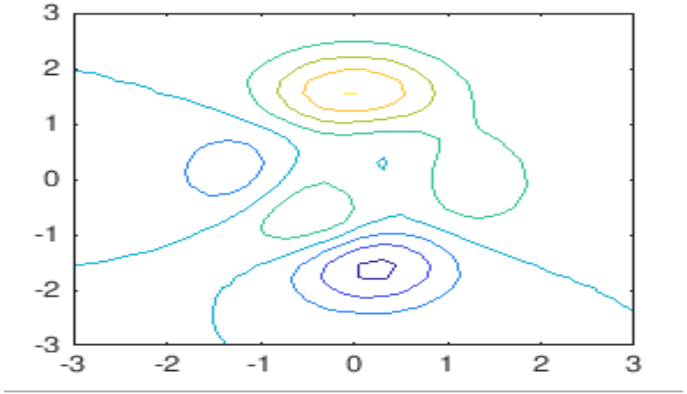

- Our task is to improve our guess for $w$ such that we move from a region of higher gradient to a region of lower gradient

- For scalar (i.e. one component) $w$, this is easy

Updating the function -> $w^{new} = w^{old} - \alpha \left( \frac{dJ}{dw} \right)$

# Gradient Descent (vector case)




- For the vector case, we rely on a theorem that says

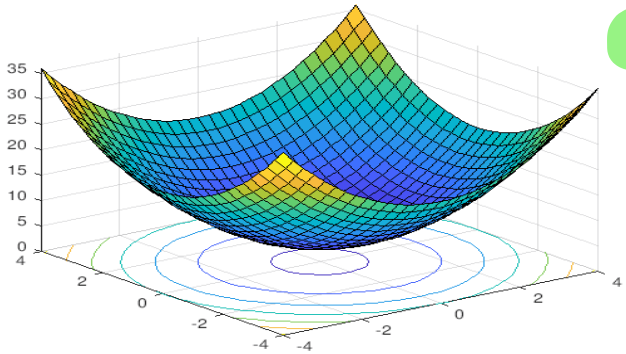
At any given point the gradient gives the direction of steepest descent

- The general gradient descent algorithm is

$$w^{new} = w^{old} - \alpha \nabla_w J$$

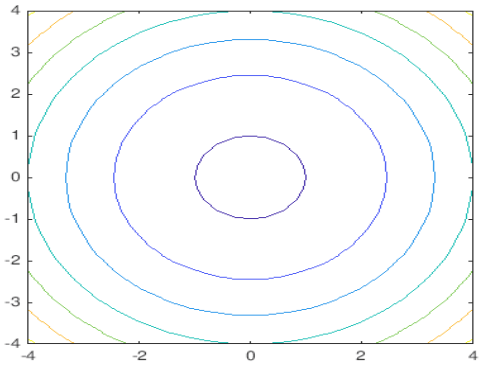
- $\alpha$ is called the learning rate is chosen by the user

# Gradient Descent example




$$J(w) = w_1^2 + w_2^2 + 4$$

$$\nabla_w J(w) = \begin{bmatrix} 2w_1 \\ 2w_2 \end{bmatrix}$$




Gradient Descent gives the iterative formula

$$w_1^{k+1} = w_1^k - \alpha \left(2w_1^k\right)$$
$$w_2^{k+1} = w_2^k - \alpha \left(2w_2^k\right)$$

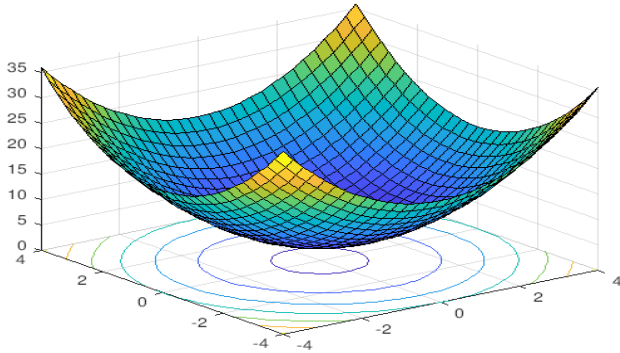
We know that the actual minimum is at $\mathbf{w} = [0 \quad 0]^T$

Let us start with an initial guess of $\mathbf{w}^0 = [3 \quad 4]^T$

Let us see different cases for various choices of α

$$\alpha = 2, 1\ , 0.1, 0.5$$

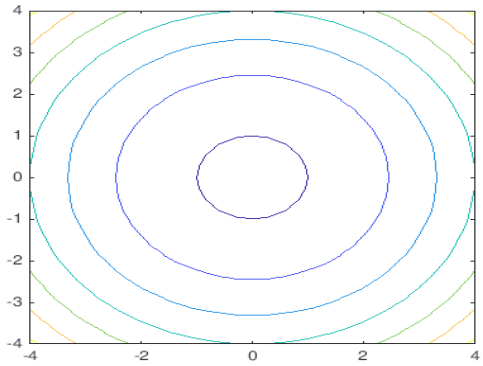
# Gradient Descent example



$$J(w) = w_1^2 + w_2^2 + 4$$

$$\nabla_w J(w) = \begin{bmatrix} 2w_1 \\ 2w_2 \end{bmatrix}$$

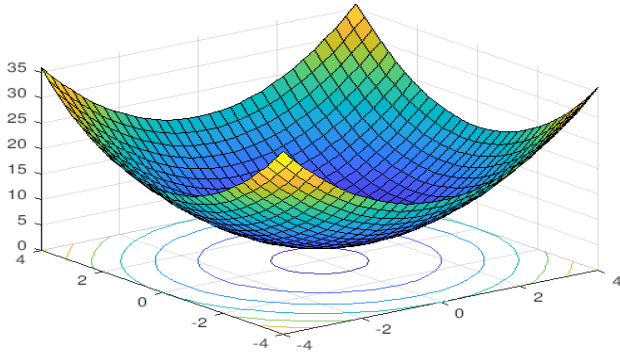
$$w_1^{k+1} = w_1^k - \alpha\left(2w_1^k\right)$$
$$w_2^{k+1} = w_2^k - \alpha\left(2w_2^k\right)$$



$\mathbf{w}^0 = [3 \quad 4]^T \qquad \alpha = 2$

| Iteration (k) | $w^k$ | $\nabla_w J = 2[\mathrm{w}_1 \ \mathrm{w}_2]$ | $J$ | $w^{k+1} = w^k - \alpha\nabla_{\mathrm{w}}\mathrm{J}$ |
|---|---|---|---|---|
| 0 | $[3 \ 4]$ | $[6 \ 8]$ | 29 | $[3 \ 4] - 2 * [6 \ 8] = [-9 \ -12]$ |
| 1 | $-[9 \ 12]$ | $-[18 \ 24]$ | 229 | $[27 \ 36]$ |
| 2 | $[27 \ 36]$ | $[54 \ 72]$ | 2029 | $-[81 \ 108]$ |

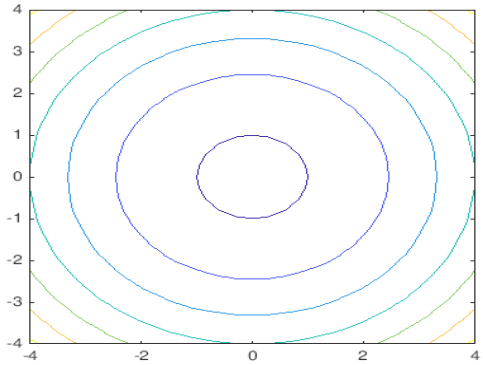
# Gradient Descent example



$$J(w) = w_1^2 + w_2^2 + 4$$

$$\nabla_w J(w) = \begin{bmatrix} 2w_1 \\ 2w_2 \end{bmatrix}$$

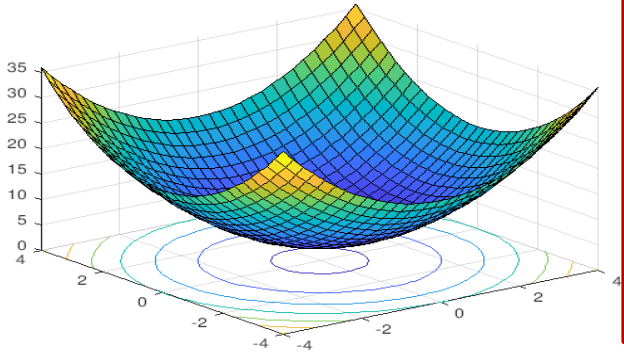
$$w_1^{k+1} = w_1^k - \alpha\left(2w_1^k\right)$$
$$w_2^{k+1} = w_2^k - \alpha\left(2w_2^k\right)$$



$\mathbf{w}^0 = [3 \quad 4]^T \qquad \alpha = 1$

| Iteration (k) | $w^k$ | $\nabla_w J = 2[w_1 \; w_2]$ | $J$ | $w^{k+1} = w^k - \alpha \nabla_w J$ |
|---|---|---|---|---|
| 0 | $[3 \;\; 4]$ | $[6 \;\; 8]$ | 29 | $[3 \;\; 4] - 1 * [6 \;\; 8] = [-3 \;\; -4]$ |
| 1 | $-[3 \;\; 4]$ | $-[6 \;\; 8]$ | 29 | $[3 \;\; 4]$ |
| 2 | $[3 \;\; 4]$ | $[6 \;\; 8]$ | 29 | $[-3 \;\; -4]$ |

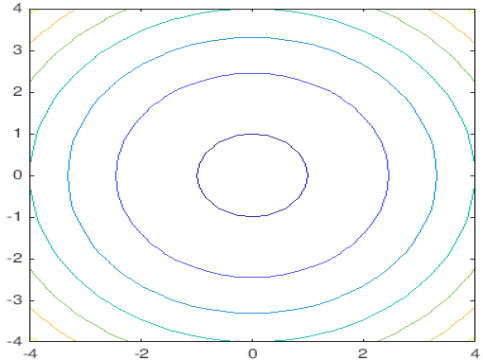
# Gradient Descent example



$$J(w) = w_1^2 + w_2^2 + 4$$

$$\nabla_w J(w) = \begin{bmatrix} 2w_1 \\ 2w_2 \end{bmatrix}$$

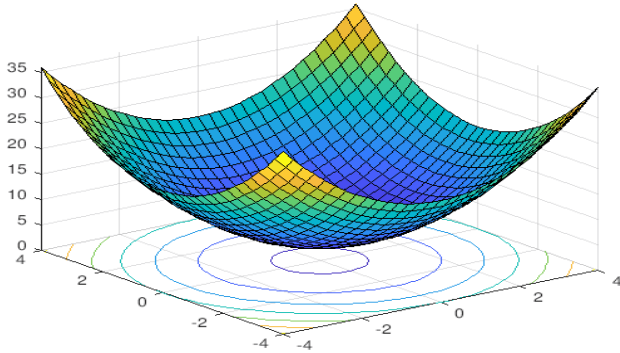
$$w_1^{k+1} = w_1^k - \alpha\left(2w_1^k\right)$$
$$w_2^{k+1} = w_2^k - \alpha\left(2w_2^k\right)$$



$\mathbf{w}^0 = [3 \quad 4]^T \qquad \alpha = 0.1$

| Iteration (k) | $w^k$ | $\nabla_w J = 2[w_1 \; w_2]$ | $J$ | $w^{k+1} = w^k - \alpha \nabla_w J$ |
|---|---|---|---|---|
| 0 | [3 4] | [6 8] | 29 | $[3 \; 4] - 0.1 * [6 \; 8] = [2.4 \; 3.2]$ |
| 1 | [2.4 3.2] | [4.8 6.4] | 20 | [1.92 2.56] |
| 2 | [1.92 2.56] | [3.84 5.12] | 14.24 | [1.536 2.048] |
| 30 | [0.0037 0.005] | … | 4.0000 | … |

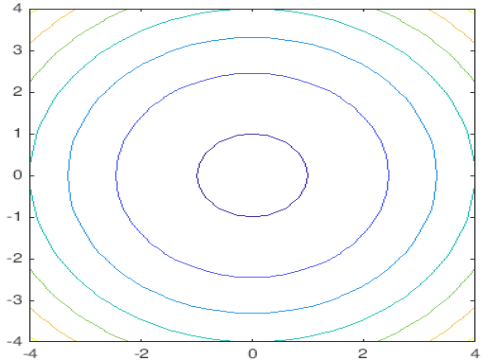
# Gradient Descent example



$$J(w) = w_1^2 + w_2^2 + 4$$

$$\nabla_w J(w) = \begin{bmatrix} 2w_1 \\ 2w_2 \end{bmatrix}$$

$$w_1^{k+1} = w_1^k - \alpha\left(2w_1^k\right)$$
$$w_2^{k+1} = w_2^k - \alpha\left(2w_2^k\right)$$



$\mathbf{w}^0 = [3 \quad 4]^T \qquad \alpha = 0.5$

| Iteration (k) | $w^k$ | $\nabla_w J = 2[\mathrm{w}_1 \ \mathrm{w}_2]$ | $J$ | $w^{k+1} = w^k - \alpha\nabla_{\mathrm{w}}\mathrm{J}$ |
|---|---|---|---|---|
| 0 | $[3 \ 4]$ | $[6 \ 8]$ | 29 | $[3 \ 4] - 0.5 * [6 \ 8] = [0 \ 0]$ |
| 1 | $[0 \ 0]$ | $[0 \ 0]$ | 4 | $[0 \ 0]$ |
| 2 | $[0 \ 0]$ | $[0 \ 0]$ | 4 | $[0 \ 0]$ |

Learning rate is the hyperparameter for gradient descent algorithm.

# Some lessons from the example

- It is possible for the gradient descent algorithm to
  - Diverge
  - Oscillate without diverging or converging
  - Converge slowly
  - Converge rapidly

- All these behaviors can manifest for the sample example depending on the learning rate $\alpha$

- The choice of $\alpha$ is part of algorithm design

- $\alpha$ is a *hyperparameter* – a parameter that must be set before learning begins