CS60010: Deep Learning
Spring 2023

Sudeshna Sarkar

Module 1 Part D

Calculus

12 Jan 2023

Differential Calculus

- For a function $f: \mathbb{R} \rightarrow \mathbb{R}$, the ***derivative*** of *f* is defined as

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

- If $f'(a)$ exists, *f* is said to be differentiable at *a*
- Given $y = f(x)$, where *x* is an independent variable and *y* is a dependent variable, the following expressions are equivalent:

$$f'(x) = f' = \frac{dy}{dx} = \frac{df}{dx} = \frac{d}{dx} f(x) = Df(x) = D_x f(x)$$

Differential Calculus

The following rules are used for computing the derivatives of explicit functions

- **Derivative of constants.** $\frac{d}{dx}c = 0.$
- **Derivative of linear functions.** $\frac{d}{dx}(ax) = a.$
- **Power rule.** $\frac{d}{dx}x^n = nx^{n-1}.$
- **Derivative of exponentials.** $\frac{d}{dx}e^x = e^x.$
- **Derivative of the logarithm.** $\frac{d}{dx}\log(x) = \frac{1}{x}.$

- **Sum rule.** $\frac{d}{dx}(g(x) + h(x)) = \frac{dg}{dx}(x) + \frac{dh}{dx}(x).$

- **Product rule.** $\frac{d}{dx}(g(x) \cdot h(x)) = g(x)\frac{dh}{dx}(x) + \frac{dg}{dx}(x)h(x).$
- **Chain rule.** $\frac{d}{dx}g(h(x)) = \frac{dg}{dh}(h(x)) \cdot \frac{dh}{dx}(x).$

Higher Order Derivatives

- The derivative of the first derivative of a function $f(x)$ is the ***second derivative*** of $f(x)$

$$\frac{d^2 f}{dx^2} = \frac{d}{dx}\left(\frac{df}{dx}\right)$$

- The second derivative quantifies how the rate of change of $f(x)$ is changing
- If we apply the differentiation operation any number of times, we obtain the *n*-th derivative of $f(x)$

$$f^{(n)}(x) = \frac{d^n f}{dx^n} = \left(\frac{d}{dx}\right)^n f(x)$$

Taylor Series

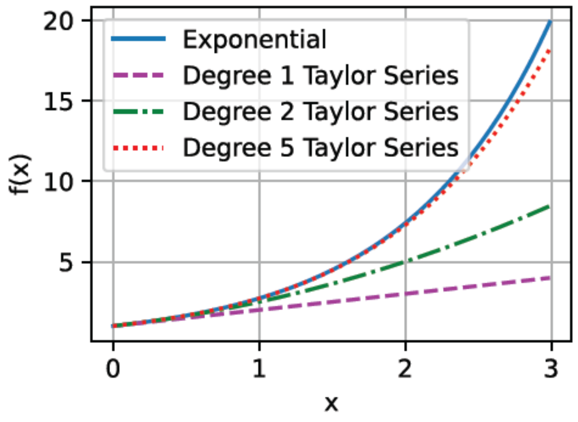
- ***Taylor series*** provides a method to approximate any function $f(x)$ at a point x_0 if we have the first *n* derivatives $\{f(x_0), f^{(1)}(x_0), f^{(2)}(x_0), \dots, f^{(n)}(x_0)\}$
- For instance, for $n = 2$, the second-order approximation of a function $f(x)$ is

$$f(x) \approx \frac{1}{2}\frac{d^2 f}{dx^2}\bigg|_{x_0} (x - x_0)^2 + \frac{df}{dx}\bigg|_{x_0} (x - x_0) + f(x_0)$$

- Similarly, the approximation of $f(x)$ with a Taylor polynomial of *n*-degree is

$$f(x) \approx \sum_{i=0}^{n} \frac{1}{i!}\frac{d^{(i)} f}{dx^i}\bigg|_{x_0} (x - x_0)^i$$

For example, the figure shows the first-order, second-order, and fifth-order polynomial of the exponential function $f(x) = e^x$ at the point $x_0 = 0$



Picture from: http://d2l.ai/chapter_appendix-mathematics-for-deep-learning/single-variable-calculus.html

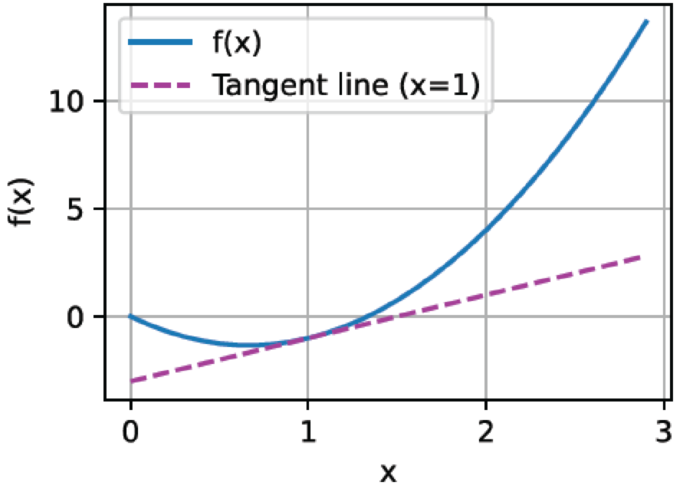
Geometric Interpretation

- To provide a geometric interpretation of the derivatives, let's consider a first-order Taylor series approximation of $f(x)$ at $x = x_0$

$$f(x) \approx f(x_0) + \left.\frac{df}{dx}\right|_{x_0} (x - x_0)$$

- The expression approximates the function $f(x)$ by a line which passes through the point $(x_0, f(x_0))$ and has slope $\left.\frac{df}{dx}\right|_{x_0}$ (i.e., the value of $\frac{df}{dx}$ at the point x_0)

Therefore, the first derivative of a function is also the slope of the tangent line to the curve of the function



Picture from: http://d2l.ai/chapter_appendix-mathematics-for-deep-learning/single-variable-calculus.html

Partial Derivatives

- Let $y = f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$ be a multivariate function with *n* variables
  - The mapping is $f: \mathbb{R}^n \to \mathbb{R}$
- The ***partial derivative*** of *y* with respect to its *i*th parameter x_i is

$$\frac{\partial y}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, x_2, \dots, x_i + h, \dots, x_n) - f(x_1, x_2, \dots, x_i, \dots, x_n)}{h}$$

- To calculate $\frac{\partial y}{\partial x_i}$, we can treat $x_1, x_2, \dots, x_{i-1}, x_{i+1} \dots, x_n$ as constants and calculate the derivative of *y* only with respect to x_i
- For notation of partial derivatives, the following are equivalent:

$$\frac{\partial y}{\partial x_i} = \frac{\partial f}{\partial x_i} = \frac{\partial}{\partial x_i} f(\mathbf{x}) = f_{x_i} = f_i = D_i f = D_{x_i} f$$

Gradient

Gradient vector: The gradient of the multivariate function $f(\mathbf{x})$ with respect to the *n*-dimensional input vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$, is a vector of *n* partial derivatives

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

- In ML, the gradient descent algorithm relies on the opposite direction of the gradient of the loss function $\mathcal{L}$ with respect to the model parameters θ $(\nabla_{\theta}\mathcal{L})$ for minimizing the loss function

Hessian Matrix

- To calculate the second-order partial derivatives of multivariate functions, we need to calculate the derivatives for all combination of input variables.
- For $f(\mathbf{x})$ with an *n*-dimensional input vector $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, there are n^2 second partial derivatives for any choice of *i* and *j*

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_i}\left(\frac{\partial f}{\partial x_j}\right)$$

- The second partial derivatives are assembled in a matrix called the ***Hessian***

$$\mathbf{H}_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

- Computing and storing the Hessian matrix for functions with high-dimensional inputs can be computationally prohibitive

Jacobian Matrix

- The concept of derivatives can be further generalized to vector-valued functions $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$
- For an *n*-dimensional input vector $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$, the vector of functions is given as

$$\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})]^T \in \mathbb{R}^m$$

- The matrix of first-order partial derivates of the vector-valued function $\mathbf{f}(\mathbf{x})$ is an $m \times n$ matrix called a ***Jacobian***

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{bmatrix}$$

Basics of Matrix Calculus

	Scalar	**Vector**	**Matrix**
Scalar	$\frac{dy}{dx}$	$\frac{dy}{d\mathbf{x}}$	$\frac{dy}{d\mathbf{X}}$
Vector	$\frac{d\mathbf{y}}{dx}$	$\frac{d\mathbf{y}}{d\mathbf{x}}$	$\frac{d\mathbf{y}}{d\mathbf{X}}$
Matrix	$\frac{d\mathbf{Y}}{dx}$	$\frac{d\mathbf{Y}}{d\mathbf{x}}$	$\frac{d\mathbf{Y}}{d\mathbf{X}}$

Derivatives of Scalar

1. With respect to a scalar $\frac{dy}{dx}$

2. With respect to a vector $\frac{dy}{d\mathbf{x}} = \begin{bmatrix} \frac{dy}{dx_1} \\ \vdots \\ \frac{dy}{dx_n} \end{bmatrix}$ $\qquad \frac{dy}{d\mathbf{x}^T} = \begin{bmatrix} \frac{dy}{dx_1} & \dots & \frac{dy}{dx_n} \end{bmatrix}$

3. With respect to a matrix $\frac{dy}{d\mathbf{X}} = \begin{bmatrix} \frac{dy}{dX_{11}} & \cdots & \frac{dy}{dX_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{dy}{dX_{m1}} & \cdots & \frac{dy}{dX_{mn}} \end{bmatrix}$

when you take the derivative of a scalar, we end up with the same shape as the variable we took the derivative with respect to.

Derivatives of Vector

1. With respect to a scalar $\frac{d\mathbf{y}}{dx} = \begin{bmatrix} \frac{dy_1}{dx} & \dots & \frac{dy_n}{dx} \end{bmatrix}$
2. With respect to a vector $\mathbf{y} \in \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^p$

$$\frac{d\mathbf{y}}{d\mathbf{x}} = [\nabla y_1(x) \quad \nabla y_2(x) \quad \dots \quad \nabla n(x)] = \begin{bmatrix} \frac{dy_1}{dx_1} & \frac{dy_2}{dx_1} & \dots & \frac{dy_n}{dx_1} \\ \frac{dy_1}{dx_2} & \frac{dy_2}{dx_2} & \dots & \frac{dy_n}{dx_2} \\ \vdots & \ddots & & \vdots \\ \frac{dy_1}{dx_p} & \frac{dy_2}{dx_p} & \dots & \frac{dy_n}{dx_p} \end{bmatrix} \in \mathbb{R}^{p \times n}$$

3. With respect to a matrix $\frac{d\mathbf{y}}{d\mathbf{X}}$:

   In general, this encodes three dimensional information $\frac{dy_i}{dX_{jk}}$

Derivatives of Vector with respect to a vector

With respect to a vector $\mathbf{y} \in \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^p$

$$\frac{d\mathbf{y}}{d\mathbf{x}} = [\nabla y_1(x) \quad \nabla y_2(x) \quad \dots \quad \nabla n(x)] = \begin{bmatrix} \frac{dy_1}{dx_1} & \frac{dy_2}{dx_1} & \cdots & \frac{dy_n}{dx_1} \\ \frac{dy_1}{dx_2} & \frac{dy_2}{dx_2} & \cdots & \frac{dy_n}{dx_2} \\ \vdots & \ddots & & \vdots \\ \frac{dy_1}{dx_p} & \frac{dy_2}{dx_p} & \cdots & \frac{dy_n}{dx_p} \end{bmatrix} \in \mathbb{R}^{p\times n}$$

Consider $\mathbf{y} = \mathbf{A}\mathbf{x}$ for a constant matrix $\mathbf{A} \in \mathbb{R}^{p\times n}$

$$\mathbf{y} = \mathbf{A}\mathbf{x} = \begin{bmatrix} A_{11} & \cdots & A_{1p} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{np} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{p} A_{1k} x_k \\ \vdots \\ \sum_{k=1}^{p} A_{nk} x_k \end{bmatrix}$$

Derivatives of Vector with respect to a vector

$$\mathbf{y} = \mathbf{A}\mathbf{x} = \begin{bmatrix} A_{11} & \cdots & A_{1p} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{np} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{p} A_{1k} x_k \\ \vdots \\ \sum_{k=1}^{p} A_{nk} x_k \end{bmatrix}$$

$y_i = \sum_{k=1}^{p} A_{ik} x_k \therefore \frac{dy_i}{dx_j} = A_{ij}$. Hence, we have

$$\frac{d\mathbf{y}}{d\mathbf{x}} = \begin{bmatrix} \frac{dy_1}{dx_1} & \frac{dy_2}{dx_1} & \cdots & \frac{dy_n}{dx_1} \\ \frac{dy_1}{dx_2} & \frac{dy_2}{dx_2} & \cdots & \frac{dy_n}{dx_2} \\ \vdots & \ddots & & \vdots \\ \frac{dy_1}{dx_p} & \frac{dy_2}{dx_p} & \cdots & \frac{dy_n}{dx_p} \end{bmatrix} = \begin{bmatrix} A_{11} \ A_{21} & \cdots & A_{n1} \\ \vdots & \ddots & \vdots \\ A_{1p} \ A_{2p} & \cdots & A_{np} \end{bmatrix} = \mathbf{A}^T$$

Chain Rule

- For (single-variable functions) $h(x) = f(g(x))$

$$\frac{dh}{dx} = \frac{df}{dg}\frac{dg}{dx} = \frac{dg}{dx}\frac{df}{dg}$$

$$x \xrightarrow{\frac{dg}{dx}} g \xrightarrow{\frac{df}{dg}} f$$

- Multivariable: $h(x) = f(g_1(x), g_2(x))$

$$\frac{dh}{dx} = \frac{\partial f}{\partial g_1}\frac{dg_1}{dx} + \frac{\partial f}{\partial g_2}\frac{dg_2}{dx}$$

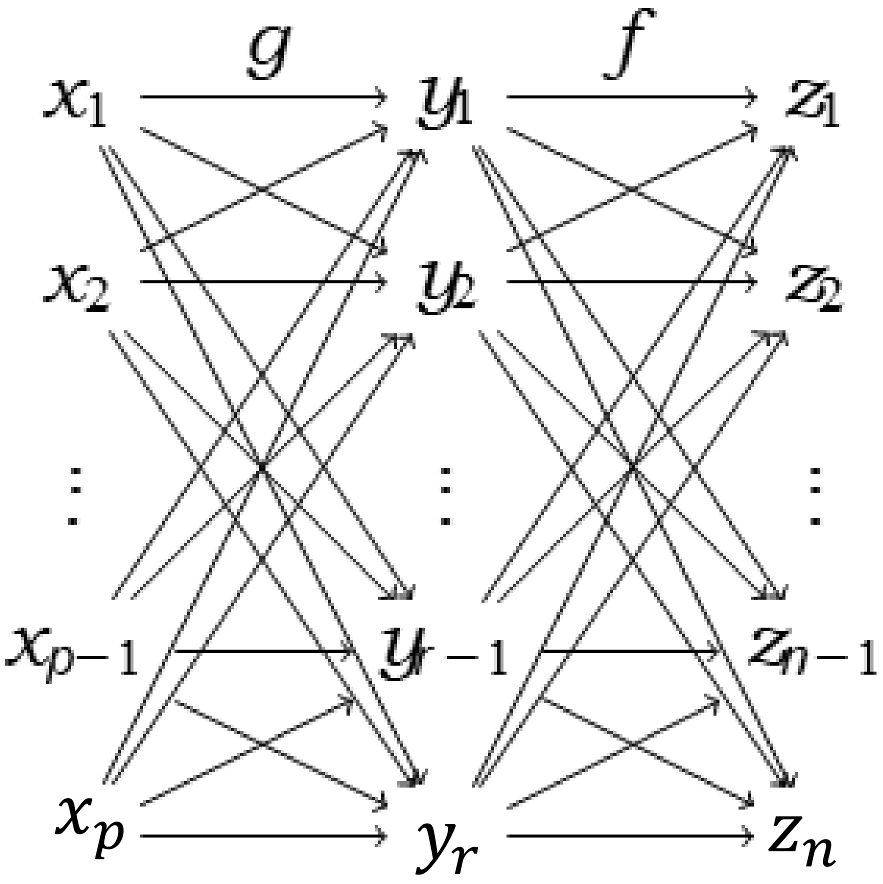
$$= \frac{dg_1}{dx}\frac{\partial f}{\partial g_1} + \frac{dg_2}{dx}\frac{\partial f}{\partial g_2}$$

$$\begin{tikzcd} & g_1 \arrow[dr, "\frac{\partial f}{\partial g_1}"] & \\ x \arrow[ur, "\frac{dg_1}{dx}"] \arrow[dr, "\frac{dg_2}{dx}"'] & & f \\ & g_2 \arrow[ur, "\frac{\partial f}{\partial g_2}"'] & \end{tikzcd}$$

adding all components that contribute to the change of h.

chain rule for vectors in matrix calculus

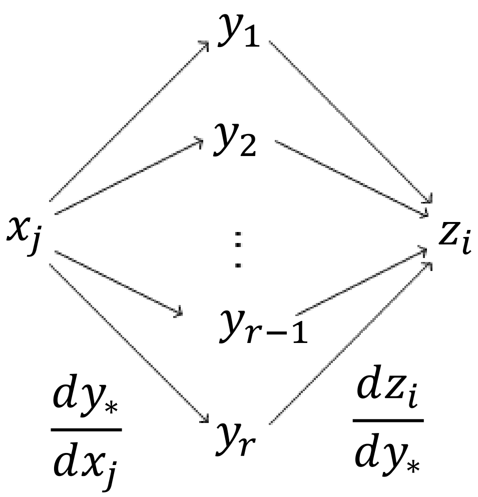
$\mathbf{x} \in \mathbb{R}^p \quad \mathbf{y} \in \mathbb{R}^r, \mathbf{z} \in \mathbb{R}^n \qquad z = f(y), y = g(x), \therefore\ z = f(g(x))$



$$\frac{d\mathbf{z}}{d\mathbf{x}} = \begin{bmatrix} \frac{dz_1}{dx_1} & \frac{dz_2}{dx_1} & \cdots & \frac{dz_n}{dx_1} \\ \frac{dz_1}{dx_2} & \frac{dz_2}{dx_2} & \cdots & \frac{dz_n}{dx_2} \\ \vdots & \ddots & & \vdots \\ \frac{dz_1}{dx_p} & \frac{dz_2}{dx_p} & \cdots & \frac{dz_n}{dx_p} \end{bmatrix}$$

By the chain rule,

$$\frac{dz_i}{dx_j} = \sum_{k=1}^{r} \frac{dz_i}{dy_k}\frac{dy_k}{dx_j} = \sum_{k=1}^{r} \frac{dy_k}{dx_j}\frac{dz_i}{dy_k}$$



Apply the scalar chain rule to each element of $d\mathbf{z}/d\mathbf{x}$. By the definition of matrix mult cation, observe that

$$
\begin{aligned}
\left(\frac{d\mathbf{z}}{d\mathbf{x}}\right)^T &= \begin{bmatrix} dz_1/dx_1 & dz_1/dx_2 & \cdots & dz_1/dx_p \\ dz_2/dx_1 & dz_2/dx_2 & \cdots & dz_2/dx_p \\ \vdots & & \ddots & \vdots \\ dz_n/dx_1 & dz_n/dx_2 & \cdots & dz_n/dx_p \end{bmatrix} \in \mathbb{R}^{n\times p} \\
&= \begin{bmatrix} \sum_{k=1}^r \frac{dz_1}{dy_k}\frac{dy_k}{dx_1} & \sum_{k=1}^r \frac{dz_1}{dy_k}\frac{dy_k}{dx_2} & \cdots & \sum_{k=1}^r \frac{dz_1}{dy_k}\frac{dy_k}{dx_n} \\ \sum_{k=1}^r \frac{dz_2}{dy_k}\frac{dy_k}{dx_1} & \sum_{k=1}^r \frac{dz_2}{dy_k}\frac{dy_k}{dx_2} & \cdots & \sum_{k=1}^r \frac{dz_2}{dy_k}\frac{dy_k}{dx_n} \\ \vdots & & \ddots & \vdots \\ \sum_{k=1}^r \frac{dz_p}{dy_k}\frac{dy_k}{dx_1} & \sum_{k=1}^r \frac{dz_p}{dy_k}\frac{dy_k}{dx_2} & \cdots & \sum_{k=1}^r \frac{dz_p}{dy_k}\frac{dy_k}{dx_n} \end{bmatrix} \\
&= \begin{bmatrix} dz_1/dy_1 & dz_1/dy_2 & \cdots & dz_1/dy_r \\ dz_2/dy_1 & dz_2/dy_2 & \cdots & dz_2/dy_r \\ \vdots & & \ddots & \vdots \\ dz_n/dy_1 & dz_n/dy_2 & \cdots & dz_n/dy_r \end{bmatrix} \begin{bmatrix} dy_1/dx_1 & dy_1/dx_2 & \cdots & dy_1/dx_p \\ dy_2/dx_1 & dy_2/dx_2 & \cdots & dy_2/dx_p \\ \vdots & & \ddots & \vdots \\ dy_r/dx_1 & dy_r/dx_2 & \cdots & dy_r/dx_p \end{bmatrix} \\
&= \left(\frac{d\mathbf{z}}{d\mathbf{y}}\right)^T \left(\frac{d\mathbf{y}}{d\mathbf{x}}\right)^T.
\end{aligned}
$$

Taking the transpose of both sides, we have that the chain rule extends to

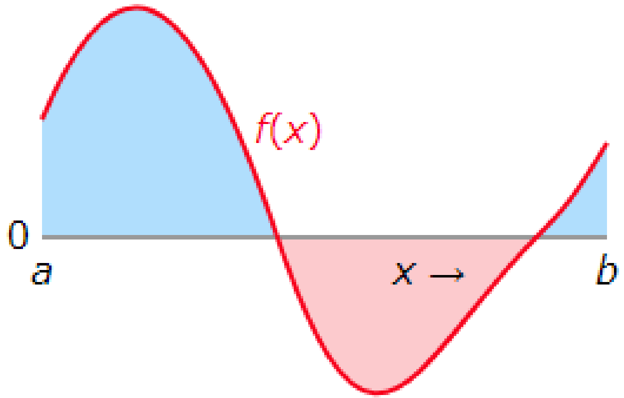
$$
\frac{d\mathbf{z}}{d\mathbf{x}} = \frac{d\mathbf{y}}{d\mathbf{x}}\frac{d\mathbf{z}}{d\mathbf{y}}.
$$

Integral Calculus

- For a function $f(x)$ defined on the domain $[a, b]$, the definite ***integral*** of the function is denoted

$$\int_a^b f(x)dx$$

- Geometric interpretation of the integral is the area between the horizontal axis and the graph of $f(x)$ between the points *a* and *b*
  - In this figure, the integral is the sum of blue areas (where $f(x) > 0$) minus the pink area (where $f(x) < 0$)



Picture from: https://mjo.osborne.economics.utoronto.ca/index.php/tutorial/index/1/clc/t

Optimization

Optimization is concerned with optimizing an objective function — finding the value of an argument that minimizes of maximizes the function

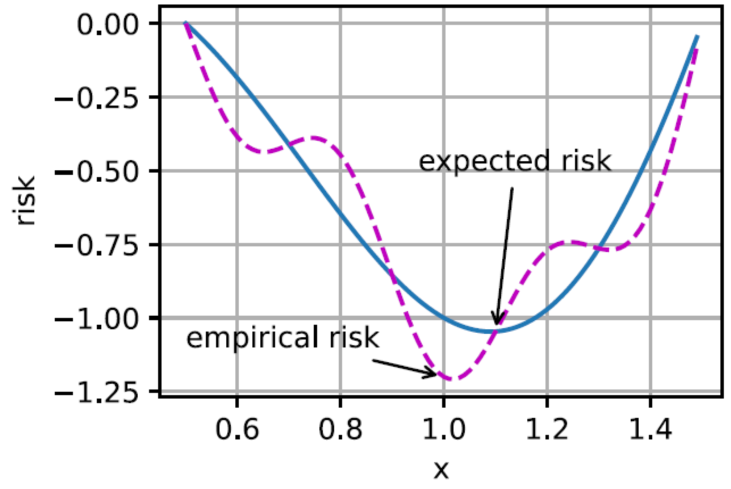
- In minimization problems, the objective function is often referred to as a cost function or loss function or error function

- Optimization is very important for machine learning
- Most optimization problems in machine learning are nonconvex

Optimization

- Optimization and machine learning have related, but somewhat different goals
  - Goal in optimization: minimize an objective function
    - For a set of training examples, reduce the training error
  - Goal in ML: find a suitable model, to predict on data examples
    - For a set of testing examples, reduce the generalization error
- For a given empirical function *g* (dashed purple curve), optimization algorithms attempt to find the point of minimum empirical risk

- The expected function *f* (blue curve) is obtained given a limited amount of training data examples
- ML algorithms attempt to find the point of minimum expected risk, based on minimizing the error on a set of testing examples
  - Which may be at a different location than the minimum of the training examples
  - And which may not be minimal in a formal sense

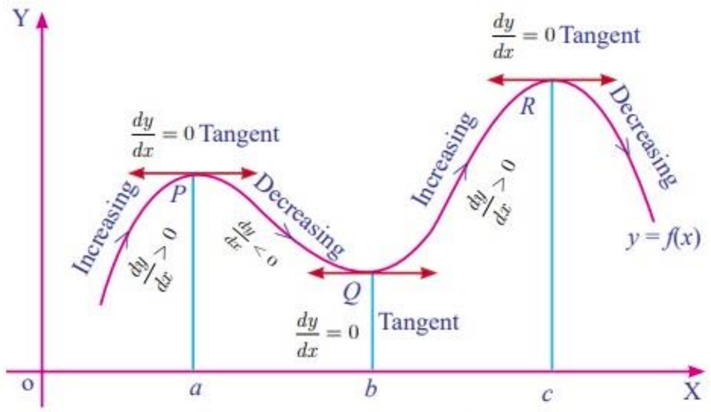


Picture from: http://d2l.ai/chapter_optimization/optimization-intro.html

Stationary Points

- ***Stationary points*** (or critical points) of a differentiable function $f(x)$ of one variable are the points where the derivative of the function is zero, i.e., $f'(x) = 0$
- The stationary points can be:
  - ***Minimum***, a point where the derivative changes from negative to positive
  - ***Maximum***, a point where the derivative changes from positive to negative
  - ***Saddle point***, derivative is either positive or negative on both sides of the point
- The minimum and maximum points are collectively known as extremum points

- The nature of stationary points can be determined based on the second derivative of $f(x)$ at the point
  - If $f''(x) > 0$, the point is a minimum
  - If $f''(x) < 0$, the point is a maximum
  - If $f''(x) = 0$, inconclusive, the point can be a saddle point, but it may not
- The same concept also applies to gradients of multivariate functions



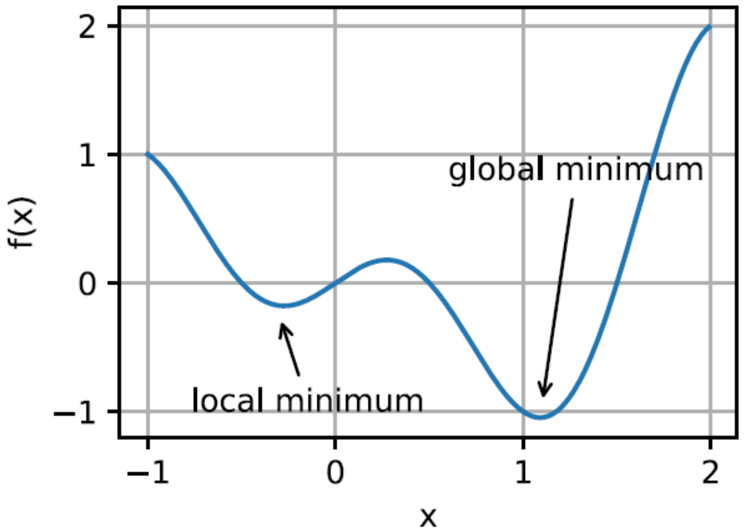
P, Q, R are called Stationary Points

Local Minima

- Among the challenges in optimization of model's parameters in ML involve local minima, saddle points, vanishing gradients
- For an objective function $f(x)$, if the value at a point *x* is the minimum of the objective function over the entire domain of *x*, then it is the ***global minimum***
- If the value of $f(x)$ at *x* is smaller than the values of the objective function at any other points in the vicinity of *x*, then it is the ***local minimum***

The objective functions in ML usually have many local minima

- When the solution of the optimization algorithm is near the local minimum, the gradient of the loss function approaches or becomes zero



Picture from: http://d2l.ai/chapter_optimization/optimization-intro.html