SIT744 Lecture 2

Math Review

Nayyar Zaidi

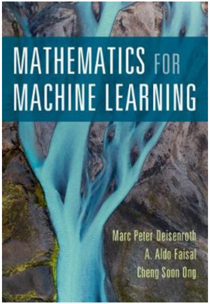
Learning objectives

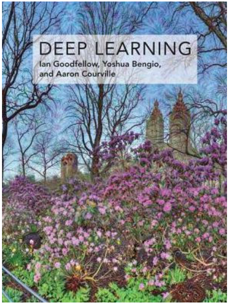
- Be familiar with the math behind deep learning theory

Plan

- Linear Algebra
- Derivatives and gradients

Reading





This lecture is based on chapters in these books (click to view):

- Mathematics for Machine Learning
  - Sections 5.1, 5.2, and 5.6
- Deep Learning Book
  - Chapter 2

Linear algebra

Why linear algebra?

Linear algebra studies vector spaces and matrices

Tensors for efficient computing

- **Data** are represented as vectors, matrices, and tensors
- **Model parameters** are represented as vectors, matrices, and tensors

Scalars, Vectors, Matrices and Tensors

Scalar

A single number

Vector

An array of numbers

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Matrices

A 2-D array

$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}$$

Tensors

An array with more than two axes

Data as vectors

Learning activity

Think about a machine learning problem. How is the data represented in the problem?

Data as vectors

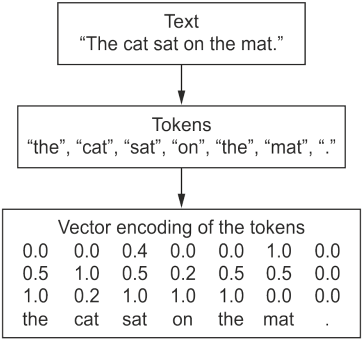


Figure 1: From text to tokens to vectors

Words represented as vectors

Data as vectors

Figure 2: MNIST sample digits

Images are matrices

Dot product

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^\top \boldsymbol{y} = \sum_{i=1}^{n} x_i y_i$$

Euclidean norm

$$\|\boldsymbol{x}\|_2 := \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$$

Other norms

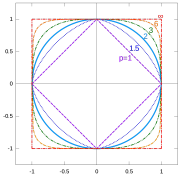


Figure 3: unit circle for different p-norms (src: wikipedia)

- Manhattan norm (L1 norm)

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

- Lp-norm (if $p \geq 1$)

$$\|\boldsymbol{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

Lengths and distance

$\|\boldsymbol{x}\|_2$ is one way to measure the length of $\boldsymbol{x}$.

Distance between vectors

$$d(\boldsymbol{x}, \boldsymbol{y}) := \|\boldsymbol{x} - \boldsymbol{y}\| = \sqrt{\langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y} \rangle}$$

Angle between vectors

$$\cos \omega = \frac{\langle x, y \rangle}{\|x\| \|y\|}$$

It is a way to measure similarity of two vectors.

Transpose of a matrix

The transpose of A, denoted $A^\top$, is defined by

$$\left(A^\top\right)_{i,j} = A_{j,i}$$

example

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,3} \\ A_{3,1} & A_{3,2} \end{bmatrix} \Rightarrow A^\top = \begin{bmatrix} A_{1,1} & A_{2,1} & A_{3,1} \\ A_{1,2} & A_{2,2} & A_{3,2} \end{bmatrix}$$

Sum of two matrices

If A and B have the same shape, $\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{B}$ is defined by

$$C_{i,j} = A_{i,j} + B_{i,j}$$

Scalar multiplication and sum

$\boldsymbol{D} = a \cdot \boldsymbol{B} + c$ is defined by

$$D_{i,j} = a \cdot B_{i,j} + c$$

A confusing notation common in deep learning

$\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{b}$ is defined by

$$C_{i,j} = A_{i,j} + b_j$$

Matrix product

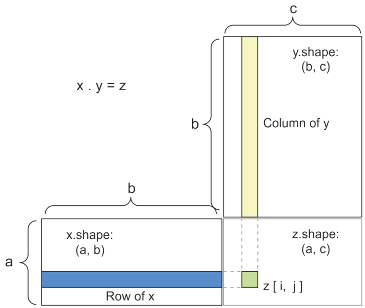


Figure 4: Matrix product

$\boldsymbol{C} = \boldsymbol{AB}$ is defined by

$$C_{i,j} = \sum_k A_{i,k} B_{k,j}$$

Matrix product is not commutative

$$\boldsymbol{AB} \neq \boldsymbol{BA}$$

However,

$$(\boldsymbol{AB})^\top = \boldsymbol{B}^\top \boldsymbol{A}^\top$$

It is easier to think of matrix product as the composition of two functions.

Identity and Inverse Matrices

$\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}_n$

Here $\boldsymbol{A}^{-1}$ is the **matrix inverse** of $\boldsymbol{A}$.

If $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ and $\boldsymbol{A}^{-1}$ exist, then

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

Norms

The L^p norm of a vector $\boldsymbol{x}$ is defined by

$$\|x\|_p = \left(\sum_i |x_i|^p \right)^{\frac{1}{p}}$$

Common L^p norms

- $\|x\|_2$
- $\|x\|_1$
- $\|x\|_\infty = \max_i |x_i|$

Eigendecomposition

Let $\boldsymbol{A}$ be a square matrix. A nonzero vector $\boldsymbol{v}$ is an **eigenvector** of $\boldsymbol{A}$ if

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}.$$

Here λ is a scalar and is called the **eigenvalue** for $\boldsymbol{v}$.

The **eigendecomposition** of $\boldsymbol{A}$ is then given by

$$\boldsymbol{A} = \boldsymbol{V} \operatorname{diag}(\boldsymbol{\lambda}) \boldsymbol{V}^{-1}.$$

Eigendecomposition for real symmetric matrix

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$$

optimize quadratic expressions

$$f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} \text{ subject to } \|\boldsymbol{x}\|_2 = 1$$

- $\max f$ is the maximum eigenvalue
- $\min f$ is the minimum eigenvalue

Machine learning and optimisation

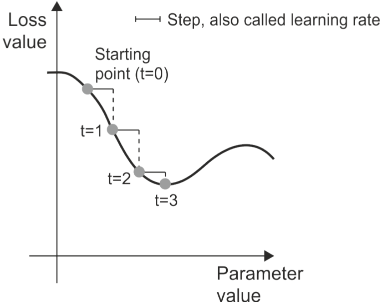


Figure 5: Learning as optimisation

Most machine learning algorithms are optimisation algorithms

- Find an $\boldsymbol{x}$ to **minimise** $f(\boldsymbol{x})$, the loss function.

Other names for the loss function

- Cost function
- Error function

Global minimum

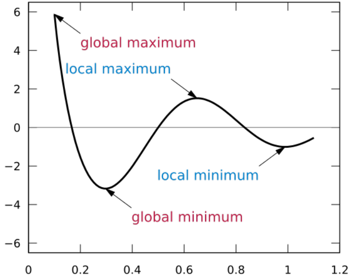


Figure 6: Global minimum (src: wikipedia)

Global minimum is realised at

$$\boldsymbol{x}^* = \arg\min f(\boldsymbol{x})$$

Derivative and gradient

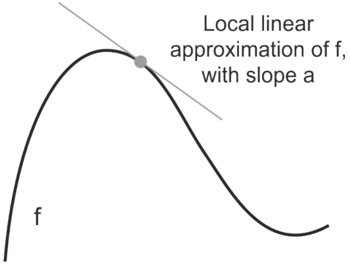
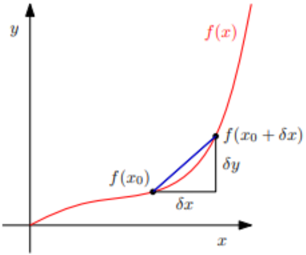


Figure 7: Derivative

Derivative $\frac{df(x)}{dx}$ (or $f'(x)$) measures the slope of f at x.

Derivative



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

Gradient

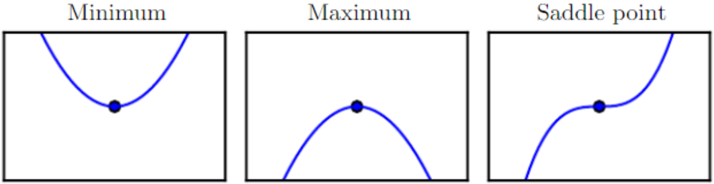
Gradient is for multi-variate functions

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\boldsymbol{x}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\boldsymbol{x}) \end{bmatrix}$$

Partial derivative

$$\frac{\partial f(\boldsymbol{x})}{\partial x_1} = \lim_{h \to 0} \frac{f\left(x_1 + h, x_2, \ldots, x_n\right) - f(\boldsymbol{x})}{h}$$

Stationary points



Consider a single variable. If x^* renders a global minimum, then the derivative $f'(x^*) = 0$.

Gradient-Based optimisation

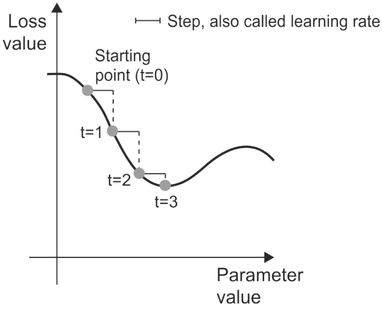


Figure 8: Method of steepest descent

Iteratively update a guess until converging to a stationary point

$$x' = x - \epsilon f'(x)$$

- ϵ is the learning rate.

Gradient descent

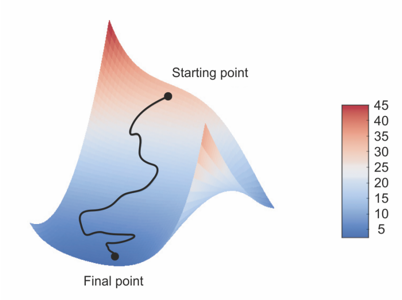


Figure 9: Steepest descent with two variables

$$\boldsymbol{x}' = \boldsymbol{x} - \epsilon \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$$

Differentiation Rules

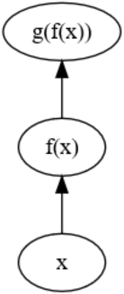


Figure 10: Chain rule

Chain rule

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

Differentiation Rules

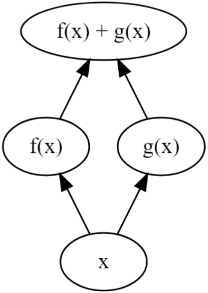


Figure 11: Sum rule

Sum rule

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

Differentiation Rules

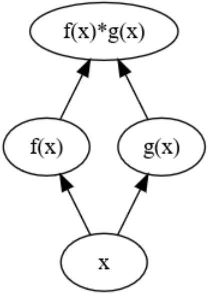


Figure 12: Product rule

Product rule

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

Derivatives of common functions

- $$\frac{d}{dx} 123 = 0$$
- $$\frac{d}{dx} x = 1$$
- $$\frac{\partial}{\partial w} wx = x$$
- $$\frac{\partial}{\partial b} (wx + b) = 1$$
- $$\frac{d}{dx} e^x = e^x$$

Example

What is

$$\frac{d}{dx}x^2$$

- Hint: Product rule

Example

$$\frac{d}{dx}x^2 = \frac{d}{dx}(x \cdot x) = x\frac{d}{dx}x + x\frac{d}{dx}x = 2x$$

Example

How about

$$\frac{d}{dx}x^n$$

Example

$$\frac{d}{dx}x^n = n\frac{d}{dx}x^{n-1}$$

Special case

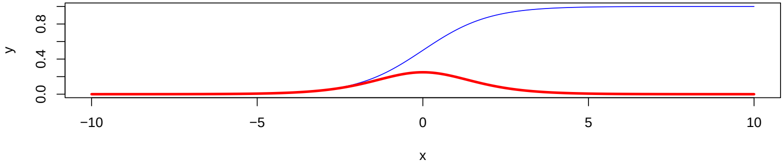
$$\frac{d}{dx}x^{-1} = -\frac{d}{dx}x^{-2}$$

Example

What is

$$\frac{d}{dx}\left(\frac{e^x}{e^x+1}\right)$$

Example



$$\frac{d}{dx}\left(\frac{e^x}{e^x+1}\right) = \frac{e^x}{(e^x+1)^2}$$

Vanishing Gradients Problem

$\frac{d}{dx}\left(\frac{e^x}{e^x+1}\right)$ is getting very small with a moderately sized x.

- $\frac{e^{10}}{(e^{10}+1)^2} = 0.000045$

Chain rule implies that the gradient quickly vanishes with more than one layers with the sigmoid activation function.

Chain rule with computational graph

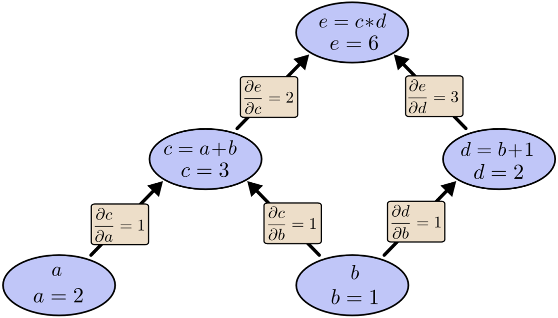


Figure 13: Backpropagation (src:colah.github.io)

Backpropagation is the default algorithm for computing gradients in deep learning.

Backpropagation is simply applying chain-rule on the computational graph.

Summary

- Linear algebra and multivariable differential calculus are fundamental math tools for deep learning
- Linear algebra is useful for representing data and transformations
- Differential calculus is useful for training deep learning models