Inhaltsverzeichnis

Abbildungsverzeichnis

List of Equations

Lecture 1: Introduction

Date: 03/05/2020

Lecturer: David Silver *By: Nithish Moudhgalya*

1.1. Overview

•p.4 expert talks are not relevant for the exam

- **scalar**: x, A, α
- column **vector**: $\underline{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = [x_i] \in \mathbb{R}^N$ with element $x_i = [\underline{x}]_i$
- **matrix**: $\mathbf{A} = [a_{ij}]_{1 \le i \le M, 1 \le j \le N} = [a_{ij}] \in \mathbb{R}^{M \times N}$ with element $a_{ij} = [\mathbf{A}]_{ij}$
- 3D, 4D **tensor**: $\mathbf{A} = [a_{ijk}], \mathbf{A} = [a_{ijkl}]$
- **transpose** of vector and matrix: $\underline{x}^T = [x_1, \ldots, x_N], \mathbf{A}^T = [a_{ji}]_{ij}$
- **determinant** of a square matrix $\mathbf{A}$: $|\mathbf{A}|$
- **inverse** of a square matrix $\mathbf{A}$: $\mathbf{A}^{-1}$
- **trace** of a square matrix $\mathbf{A}$: $\mathrm{tr}(\mathbf{A}) = \sum_i a_{ii}$

Abbildung 1.1: Mathematical notations

•Element notation (element of vector): $[x_i = \underline{a} + \underline{b}]_i$

1.2. Chapter 1.1 - What is machine learning

•Signal processing is not a subset of ML or the other way around, they are identical in the task
•Basic difference is in how to design the processing rule
•Regression means to calculate (SP,ML) a continuous-valued output in the real numbers from a signal, can be one-dimensional
•Classification means to calculate (SP,ML) a discrete-valued output from the natural numbers
•p. 1-5 filter in the middle is the view of a pixel and its 8 neighbours

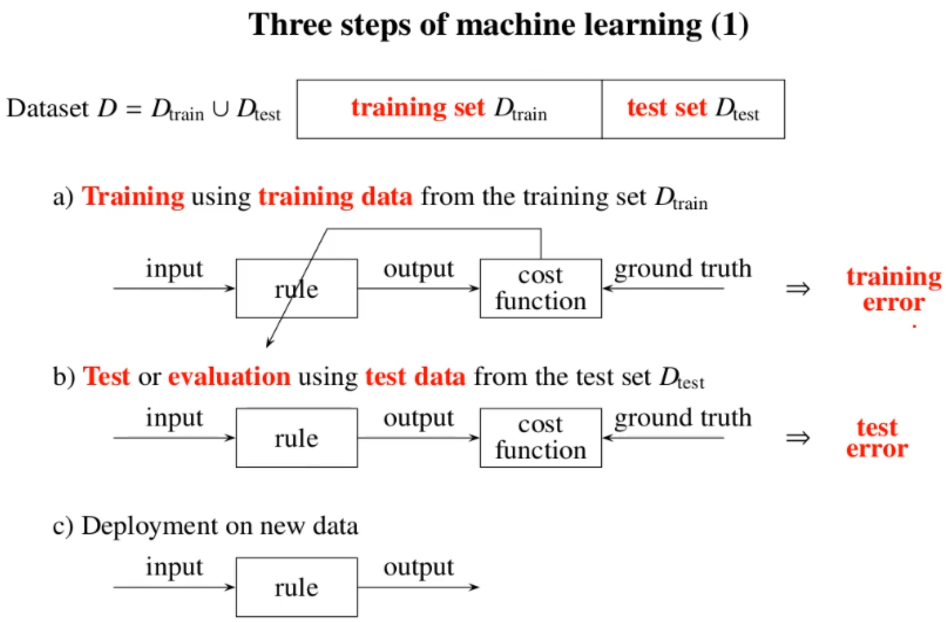


Abbildung 1.2: Three steps of machine learning

•over-fitting just memorizes the training set so test set is needed
•testerrorrate is similar to training error rate $\rightarrow$ no over-fitting
•Test set and training set need to be disjuct concatenated
•TODO: Summary supervised learning and unsupervised learning

1.3. Chapter 1.2 - What is deep learning

•feature extraction: a feature is a clustered subset of information for recognition
•fish example: fish length, color etc.
•Needs human experience, can't be calculated
•DNN solves the problem without feature extraction

1.3.1. What is a neural network (NN)

•Cascaded pipeline of layers

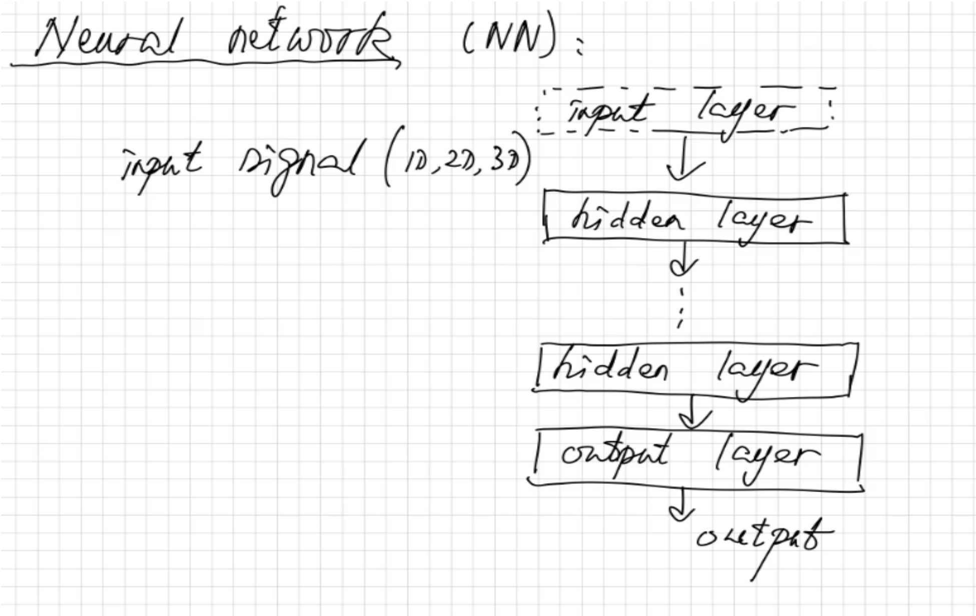


Abbildung 1.3: Neural Network

1.4. Chapter 1.3 - Examples for Deep Learning

•Semantic image segmentation is pixelwise classification

Lecture 2: Tools for Deep Learning

Date: 02/05/2020

Lecturer: David Silver *By: Nithish Moudhgalya*

2.1. Datasets

•Overview for Training Datasets up to ImageNet they are for teaching

---

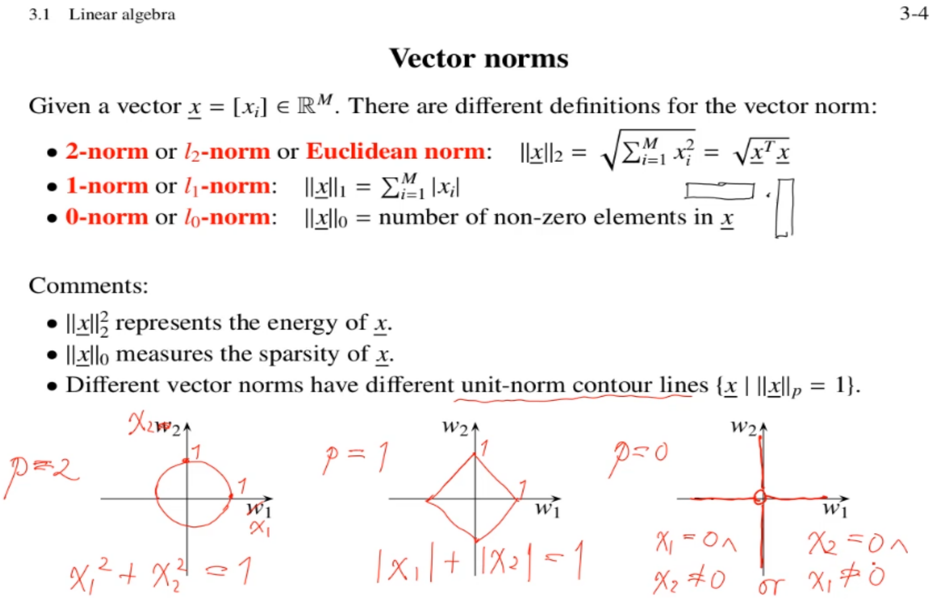
Lecture 3: Machine learning basics

Date: 02/05/2020

Lecturer: David Silver *By: Nithish Moudhgalya*

3.1. Linear Algebra

•TODO get Math nicely into the Context book


3.1 Linear algebra 3-4

Vector norms

Given a vector $\underline{x} = [x_i] \in \mathbb{R}^M$. There are different definitions for the vector norm:

- **2-norm** or l_2-**norm** or **Euclidean norm**: $\|\underline{x}\|_2 = \sqrt{\sum_{i=1}^M x_i^2} = \sqrt{\underline{x}^T\underline{x}}$
- **1-norm** or l_1-**norm**: $\|\underline{x}\|_1 = \sum_{i=1}^M |x_i|$
- **0-norm** or l_0-**norm**: $\|\underline{x}\|_0$ = number of non-zero elements in $\underline{x}$

Comments:

- $\|\underline{x}\|_2^2$ represents the energy of $\underline{x}$.
- $\|\underline{x}\|_0$ measures the sparsity of $\underline{x}$.
- Different vector norms have different unit-norm contour lines $\{\underline{x} \mid \|\underline{x}\|_p = 1\}$.

Abbildung 3.1: Vector Normes

3.2. Chapter 3.2 Random variable and probability distribution

3.2.1. 3.2.1 One random vector

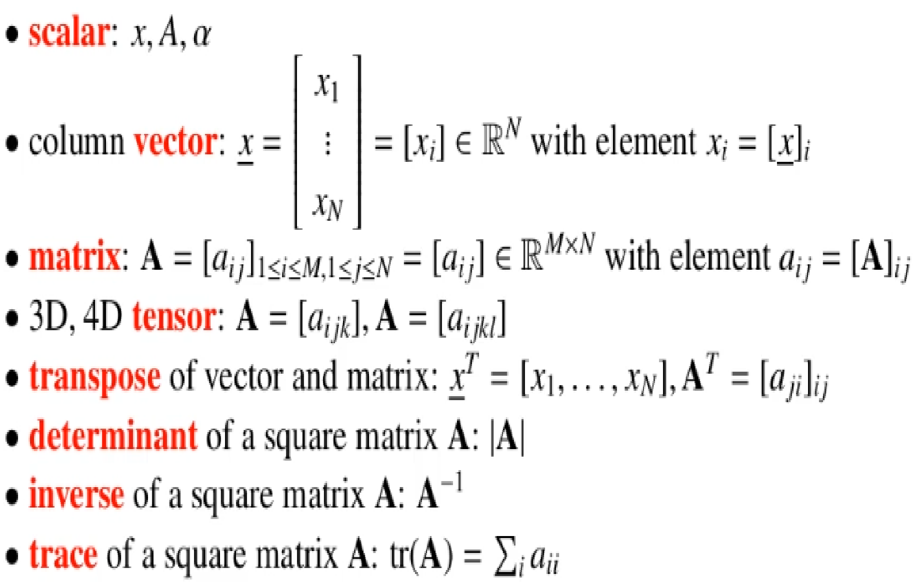


Abbildung 3.2: One random vector

PDF for a discrete-valued RV:
$p(\underline{x}) = \sum_i p_i \delta(\underline{x} - \underline{x}_i,)$ $\delta(\underline{x} :)$ Dirac function
cumulative distribution function CCDF
$F(\underline{x}) = \int_{-\infty}^{\infty} p(\underline{z}) d\underline{z}, \quad p(\underline{x}) = \frac{\partial^d F(\underline{x})}{\partial x_i \ldots \partial x_d}$

- **scalar**: x, A, α
- column **vector**: $\underline{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = [x_i] \in \mathbb{R}^N$ with element $x_i = [\underline{x}]_i$
- **matrix**: $\mathbf{A} = [a_{ij}]_{1 \leq i \leq M, 1 \leq j \leq N} = [a_{ij}] \in \mathbb{R}^{M \times N}$ with element $a_{ij} = [\mathbf{A}]_{ij}$
- 3D, 4D **tensor**: $\mathbf{A} = [a_{ijk}], \mathbf{A} = [a_{ijkl}]$
- **transpose** of vector and matrix: $\underline{x}^T = [x_1, \ldots, x_N], \mathbf{A}^T = [a_{ji}]_{ij}$
- **determinant** of a square matrix $\mathbf{A}$: $|\mathbf{A}|$
- **inverse** of a square matrix $\mathbf{A}$: $\mathbf{A}^{-1}$
- **trace** of a square matrix $\mathbf{A}$: $\mathrm{tr}(\mathbf{A}) = \sum_i a_{ii}$

Abbildung 3.3: Moments of a vector

special case d = 1: $\underline{X} \to X \in \mathbb{R}$
$\underline{\mu} \to \mu \in \mathbb{R}$
$\underline{\underline{C}} \to$ variance of $X = Var(X) = \delta^2 = E[(X-\mu)^2] = ... = E(X^2) - \mu^2$
$\delta = \sqrt{Var(X)}$: standard deviation
For any function $g(\underline{X})$ of $\underline{X}$: $E[g(\underline{x})] = \int g(x) \cdot p(\underline{x}) d\underline{x} = (d.v.) \sum_i g(\underline{x}_i \cdot P(\underline{x}_i))$

Multivariate normal (Gaussian) distribution

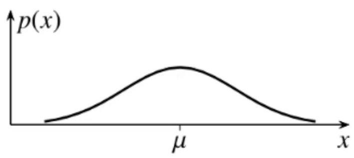
PDF

$$\begin{aligned} \underline{X} \in \mathbb{R}^d &\sim N(\underline{\mu}, \mathbf{C}) \\ p(\underline{x}) &= \frac{1}{(2\pi)^{d/2}|\mathbf{C}|^{1/2}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu})^T \mathbf{C}^{-1}(\underline{x}-\underline{\mu})} \\ \ln(p(\underline{x})) &= -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln(|\mathbf{C}|) - \frac{1}{2}(\underline{x}-\underline{\mu})^T \mathbf{C}^{-1}(\underline{x}-\underline{\mu}) \end{aligned}$$

$N(\underline{0}, \mathbf{I})$ is called the **standard normal distribution**.

$\underline{\underline{I}} = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$ identity matrix

1D-Visualization



Moments

$E(\underline{X}) = \underline{\mu}, \quad \mathrm{Cov}(\underline{X}) = \mathbf{C}$

Abbildung 3.4: Multivariate normal (Gaussian) distribution

one-hot coding : Only one bit is 1 e.g. 0100 one cold coding is the inverse
Coding for class label, random vector y of length c so the identity matrix with dimension c is used for class labels
Reforumulation of the PMF (categorical distribution) by one-hot coding of the classes
$\underline{x} = [x_i] \in \{\underline{e}_1, \underline{e}_2, .., \underline{e}_c\}$, i.e. all $x_i = 0$ except for one single element equal to 1

PMF: $P(\underline{X} = \underline{x}) = P(\underline{x}) = \begin{cases} P_i \text{ if } \underline{x} = \underline{e}_1 \text{ or } x_1 = 1 \\ \ldots \\ P_c \text{ if } \underline{x} = \underline{e}_c \text{ or } x_c = 1 \end{cases} = {p_1}^{x_1} \cdot {p_2}^{x_2} \cdot, \ldots \cdot {p_c}^{x_c} = \prod_{i=1}^{c} p_i^{x_i}$

$$ln(P(\underline{x}) = sum_{i=1}^{c} x_i \cdot ln(p_i) = [x_i, ..., x_c] \cdot \begin{bmatrix} ln(P_i) \\ ... \\ ln(P_c) \end{bmatrix} = \underline{x}^T \cdot ln(\underline{P)}$$

ln function applied element wise