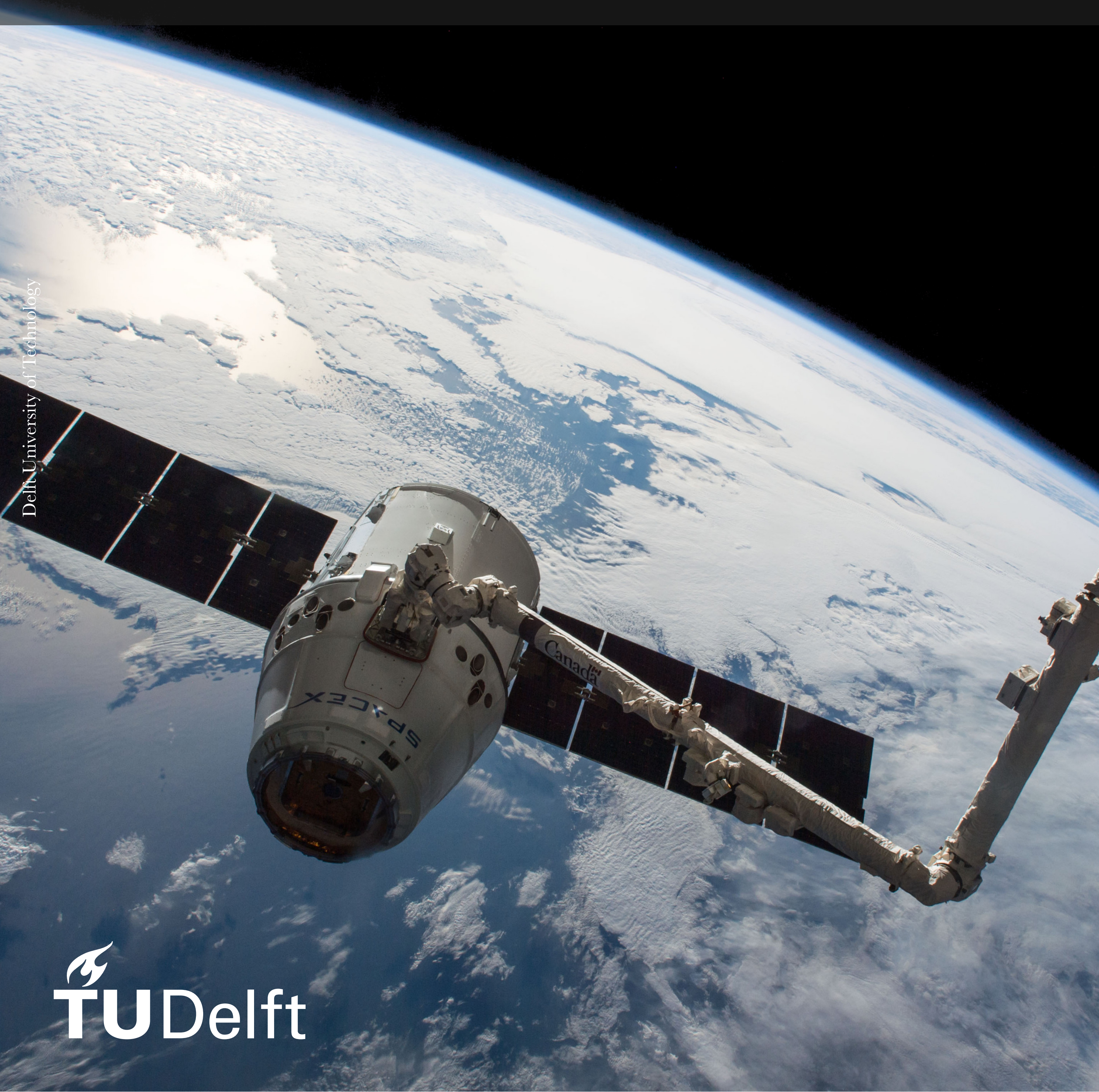
Deep Learning

Come from dive into deep learning
note For reading

我真的不懂忧郁



Delft University of Technology

TU Delft

Deep Learning

Come from dive into deep learning note For reading

by

我真的不懂忧郁

Student Name	Student Number
First Surname	1234567

Instructor:	I. Surname
Teaching Assistant:	I. Surname
Project Duration:	Month, Year - Month, Year
Faculty:	Faculty of Aerospace Engineering, Delft

Cover:	Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA under CC BY-NC 2.0 (Modified)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

Preface

A preface...

我真的不懂忧郁
Delft, August 2024

Summary

A summary...

目录

Nomenclature

If a nomenclature is required, a simple template can be found below for convenience. Feel free to use, adapt or completely remove.

Abbreviations

Abbreviation	Definition
ISA	International Standard Atmosphere
...	

Symbols

Symbol	Definition	Unit
V	Velocity	[m/s]
...		
ρ	Density	[kg/m^3]
...		

Chapter 1

Linear Nerual Network

1.1. Practice 1: 线性回归

Question 1: 假设有一些数据 $x_1, \cdots, x_n \in \mathbb{R}$。找使得 $\sum_i (x_i - b)^2$ 最小化的解析解，这个问题以及其解和正态分布有什么关系？

令 $\mathcal{L}(b) = \sum_i (x_i - b)^2$，则

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial b} &= -\sum_i 2(x_i - b) = 0 \\ \Rightarrow b &= \frac{x_1 + \cdots, x_n}{n}\end{aligned} \tag{1.1}$$

即令解析解最小化的 b 刚好是数据集 $x_1, \cdots, x_n$ 的均值。

Question 2: 推导使用平方误差的线性回归优化问题的解析解。

1. 用向量表示法写出优化问题；
2. 计算损失对 ω 的梯度；
3. 通过将梯度设为 0、求解矩阵方程来找到解析解；
4. 什么时候可能比使用**随机梯度下降**更好？这种方法何时会失效？

首先对于数据矩阵为 $X = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^{m \times n}$，其中 $x_i = (x_{1i}, x_{2i}, \cdots, x_{ni})$ 写成线性回归问题为

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \tag{1.2}$$

因此回归问题写成

$$\hat{y} = \omega^T X \tag{1.3}$$

其中 $\omega = (\omega_1, \cdots, \omega_n)^T$，$\hat{y} \in \mathbb{R}^n$。假设 $y \in \mathbb{R}^n$ 是标签向量，则优化问题写成

$$\omega = \arg\min \|y - \hat{y}\|_2^2 = \arg\min \|y - \omega^T X\|^2 \tag{1.4}$$

因此损失函数为

$$\mathcal{L}(\omega) = \|y - \omega^T X\|^2 \tag{1.5}$$

求损失函数的梯度

$$\nabla_\omega \mathcal{L} = \tag{1.6}$$

Question 3: 假定控制附加噪声 ε 的噪声模型是指数分布：$p(\omega) = \frac{1}{2}exp(-|\varepsilon|)$，

1. 写出模型 $-log\ P(y|X)$ 下数据的负对数似然函数；
2. 试着写解析解；
3. 提出一种 *SGD* 算法来解决这个问题，那里可能出错？（当我们不断更新参数时，在驻点处会发生什么？）

1.2. Practice 2: 线性回归从零实现

Question 4: 如果我们将权重初始化为零，会发生什么？

Question 5: 假设试图为电压和电流关系建立一个模型，自动唯微分可以用来学习模型的参数吗？

Question 6: 能基于普朗克定律使用广谱能量密度来确定物体的温度么？

Question 7: 计算二阶导数时可能会遇到什么问题？

Question 8: 为什么在 *squared_loss* 函数中需要使用 *reshape* 函数

Question 9: 尝试使用不同的学习率，观察损失函数值下降的快慢

Question 10: 如果样本个数不能被批量整除，*data_iter* 函数的行为会有什么变化

References

[1] I. Surname, I. Surname, and I. Surname. “The Title of the Article”. In: *The Title of the Journal* 1.2 (2000), pp. 123–456.

Chapter A

Source Code Example

Adding source code to your report/thesis is supported with the package `listings`. *An example can be found below. Files can be added using* `\lstinputlisting[language=<language>]{<filename>}`.

```python
"""
ISA Calculator: import the function, specify the height and it will return a
list in the following format: [Temperature,Density,Pressure,Speed of Sound].
Note that there is no check to see if the maximum altitude is reached.
"""

import math
g0 = 9.80665
R = 287.0
layer1 = [0, 288.15, 101325.0]
alt = [0,11000,20000,32000,47000,51000,71000,86000]
a = [-.0065,0,.0010,.0028,0,-.0028,-.0020]

def atmosphere(h):
    for i in range(0,len(alt)-1):
        if h >= alt[i]:
            layer0 = layer1[:]
            layer1[0] = min(h,alt[i+1])
            if a[i] != 0:
                layer1[1] = layer0[1] + a[i]*(layer1[0]-layer0[0])
                layer1[2] = layer0[2] * (layer1[1]/layer0[1])**(-g0/(a[i]*R))
            else:
                layer1[2] = layer0[2]*math.exp((-g0/(R*layer1[1]))*(layer1[0]-layer0[0]))
    return [layer1[1],layer1[2]/(R*layer1[1]),layer1[2],math.sqrt(1.4*R*layer1[1])]
```

Chapter B

Task Division Example

If a task division is required, a simple template can be found below for convenience. Feel free to use, adapt or completely remove.

表 **B.1:** Distribution of the workload

	Task	Student Name(s)
	Summary	
Chapter 1	Introduction	
Chapter 2		
Chapter 3		
Chapter *		
Chapter *	Conclusion	
	Editors	
	CAD and Figures	
	Document Design and Layout	

Chapter C

矢量求导

C.1. 多元泰勒展开

一元函数 $f(x)$ 在 x_0 处的泰勒展开表示为

$$f(x_0) = f(x_0) + \frac{f'(x)}{1!}(x - x_0) \tag{C.1}$$