

University of Stuttgart

Institute for Modelling Hydraulic and Environmental Systems

Department of Stochastic Simulation and Safety Research for Hydrosystems

Matthias Gueltig

Learning a contaminant transport solver for PFAS with Finite Volume Neural Networks



Bachelor's Thesis

Learning a contaminant transport solver for PFAS with Finite Volume Neural Networks

Submitted by

Matthias Gueltig

Matriculation Number: 3469020

Examiners: Prof. Dr.-Ing. Wolfgang Nowak

Supervisors: M.Sc. Nils Wildt

Institute for Modeling Hydraulic and Environmental Systems
Department of Stochastic Simulation and Safety Research for Hydrosystems
Stuttgart, 27 December 2022

Author declaration

I declare that I have developed and written the enclosed thesis completely by myself and that I have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked.

The thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere.

The enclosed electronic version is identical to the printed versions.

Date

Signature

Contents

0 Propaedeuticum: PyTorch - A powerful tool for machine learning	1
0.1 Introduction	1
0.2 General principles of Neuerl Networks	1
0.2.1 Probabilistic framework of supervised learning	2
0.2.2 Kulback - Leibler divergence and cruss entropy	2
1 First Chapter	3
List of Figures	5
List of Tables	7
Listings	9

0 Propaedeuticum: PyTorch - A powerful tool for machine learning

0.1 Introduction

Forschung beschreibt mit wissenschaftlichen Methoden systematisch Neues zu suchen. Supercomputer und riesige Rechencluster können zur Forschung in mehrerlei Hinsicht Unterstützung leisten: Mithilfe moderner Simulationssoftware und entsprechenden Libraries können bspw. Differentialgleichungen, die komplexe Strömungs- und Transportprozesse, wirtschaftliche Wachstumsprozesse oder chemische Reaktionen beschreiben akkurat gelöst werden und somit Daten generieren die der Forschung zur Verfügung stehen. Auf der anderen Seite können Rechenmaschinen seit den letzten Jahrzehnten auch selbst Forschung betreiben, indem vom Menschen implementierte, insbesondere Optimierungsalgorithmen, Daten systematisch miteinander in Beziehung bringen und verknüpfen, um nach Neuem zu suchen. Man spricht hierbei vom Deep Learning, einem Teilbereich des Machine Learnings. Mithilfe Neuronaler Netze können bspw. basierend auf großen Datensätzen Maschinen dazu trainiert werden selbst Muster und Texte zu verstehen, oder Entscheidungen zu treffen. Physikalisch informierte neuronale Netze (PINNs) können physikalische Gesetze in ihren Lernprozess für einen gegebenen Datensatz einbetten.

In diesem einführenden Kapitel sollen Grundzüge des Deep Learnings mithilfe einer Python - basierten Library mit dem Namen PyTorch vorgestellt werden und anhand eines binary classification problems illustriert werden.

PyTorch is an open source machine learning framework, welches in Python implementiert wurde. Aufgrund eines Autograd-Systems, GPU - Beschleunigung und hoher Flexibilität, eignet es sich besonders gut um Neuronale Netze zu erstellen.

0.2 General principles of Neueral Networks

Um die grundlegende Funktionsweise eines Neuronalen Netzes zu erklären beschränkt sich der folgende Abschnitt auf Dense Neural Networks die supervised lernern. Training datasets thus contain input and corresponding outputs where the computer learns the relation between in- and output data e. g in regression or classification. As illustration a binary classification problem is used where the computer learns to map wine according to its ingredients to one of two wine cultivators.

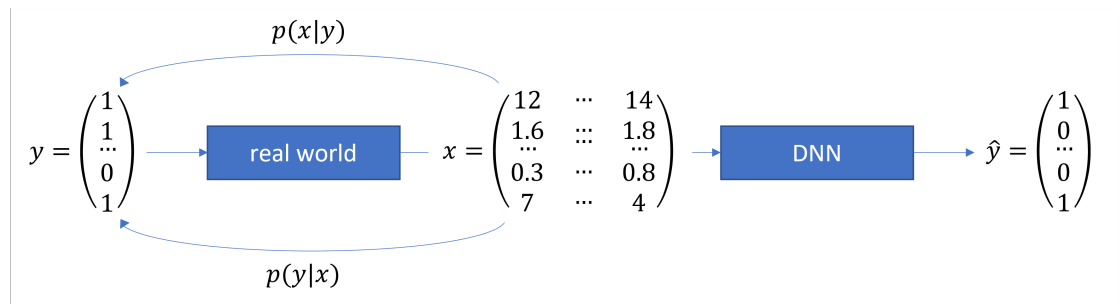


Figure 1: Probabilistic framework of supervised learn

0.2.1 Probabilistic framework of supervised learning

In ?? y is the latent variable that contains the values 0 or 1 according to wine cultivator labeld with 0 or 1. x describes the observed variable, thus the properties of wine of the dataset. E. g. the alcohol percentage or colour intesity. *yistheoutputoftheDNNandtheestimatef*

In our case y and x can be modeled as random variables. They are described by the joint generationg distribution:

$$p(x,y) = p(x|y)p(y) = p(y|x)p(x)(0.1)$$

In general a DNN has to roles:

1. approximate true posteriori $p(y|x)$ by $q(y|x; \theta)$
2. learn θ from D_{train}

In this case

0.2.2 Kulback - Leibler divergence and cruss entropy

Bayes rule

$$p(x|y) = p(x|y) * p(y)_{\overline{p(x)}} \quad (0.2)$$

1 First Chapter

List of Figures

1	Probabilistic framework of supervised learn	2
---	---	---

List of Tables

Listings