StormMind
AI based Web-Service for Storm Damage Prediction

Bachelor Thesis

ZHAW School of Engineering
Institute of Computer Science

Submitted on:
06.06.2025

Authors:
Gämperli Nils
Ueltschi Damian

Supervisor:
Andreas Meier

Study Program:
Computer Science

Declaration of Authorship

We, Damian UELTSCHI, Nils GÄMPERLI, declare that this thesis titled, “StormMind - AI based Web-Service for Storm Damage Prediction” and the work presented in it are our own. We confirm that:

- All external help, aids and the adoption of texts, illustrations and programs from other sources are properly referenced in the work.
- This is also a binding declaration that the present work does not contain any plagiarism, i.e. no parts that have been taken over in part or in full from another’s text or work under pretence of one’s own authorship or without reference to the source.
- In the event of misconduct of any kind, Sections 39 and 40 (Dishonesty and Procedure in the Event of Dishonesty) of the ZHAW Examination Regulations and the provisions of the Disciplinary Measures of the University Regulations shall come into force.

I have no limitations.
— Thomas Shelby

To our parents...

Abstract

Extreme weather events cause considerable damage to infrastructure, the economy and the environment worldwide. Predicting such damage can help to optimize preventive measures and reduce costs. In this project, a neural network is being developed that analyzes historical weather data and damage reports to predict potential storm damage in Switzerland based on new weather forecasts. Using modern machine learning techniques, relevant patterns are recognized in order to train a predictive model. The model is tested for accuracy and optimized to enable reliable predictions.

Key words: Machine Learning, Neural Network, Storm Damages

Acknowledgements

We would like to thank Andreas Meier for his valuable support and guidance throughout this project. Additionally, we would like to thank to Dr. Katharina Liechti which has provided us with valuable insights about the sciences of weather and storm damages. For providing and collecting data to make this project possible, we also like to thank to the Swiss Federal Institute for Forest, Snow and Landscape Research WSL.

Preface

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

Key words:

Contents

1 Introduction

1.1 Motivation

Why do we do this?

1.2 Work Outline

What do we want to predict etc.

2 Theoretical Background

2.1 Weather Research

2.1.1 Reason for Flooding

- Drought Followed by Heavy Reain
- Snowmelt in Spring

2.1.2 Reason for Landslide

- Slope, Water, Soil Condition

2.2 Machine Learning

Machine Learning, especially with Artificial Neural Networks became increasiningly popular in the last years, due to further achievments in this field. In this chapter Neural Networks will be mainly discussed, as their where our main modelling approach in this project.

2.2.1 Neural Networks

Neural Networks are designed to simulate the techniques which are also used in biological neurons.

Structure

An Artificial Neural Network trained with backpropagation, which is discussed in Section 2.2.2, can be illustrated as a directed acyclic Graph with inter-connections. It contains a set of neurons distributed in different layers.

- Each neuron has a activation function.

- The first layer, shown on the left side in Figure 2.1, is called the input layer and has no predacessors in the inter-connection graph. Additionally, is their input value the same as their output value.
- The last layer, shown on the right side in Figure 2.1, is called the output layer and have no successors in the inter-connection graph. Their value represents the output of the Network
- All other Neurons are grouped in the so called hidden layers. In Figure 2.1 this is represented by the layer in the middle. A neural network can have an arbitrary amount of hidden layers.
- The edges in the inter-connection graph, are so called weights, which represent an arbitrary number in $\mathbb{R}$. These weights are updated during the trianing process.

[1], [2]

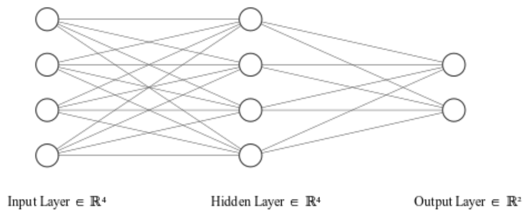

Figure 2.1 – Illustration of a Neural Network with 3 layers. Illustrated with [3]

Computation of the Output

The callculation of the output of the Neural Network is also called a forward pass. To do so, the value of each neuron needs to be calculated. This is done by summing all the inputs and then put this value into a given activation function. Mathematically, this process can be represented as: $y = f\left(\sum_i^n \xi_i\right)$ wher f represents the activation function of the neuron and ξ_i the input or also called the potential of a neuron. To compute a full forward pass, this is done for each neuron from the input layer towards the output layer. When a value is passed throughtouh a weight to a successor neuron, the value is multiplied by the value of the weight. This process can then be summarized to:

$$
\begin{aligned}
& y_1 = f\left(\sum_i^n w_{ij} x_i\right) \text{ [input to hidden layer]} \\
& y_{j+1} = f\left(\sum_i^n w_{ij} y_j\right) \forall j \in \{1...k-1\} \text{[hidden to hidden layer]} \\
& o = f\left(w_{ij+1} y_j\right) \text{ [hidden to output layer]}
\end{aligned}
\tag{2.1}
$$

where j denotes the layer, ascending from input layer to output layer, f activation function, w_{ij} weight at index i and layer j, x as input at index i. The state of the neuron in the output layer o can then by denoted as the output vector. [1]

2.2.2 The Backpropagation Training Algorithm

Objective: To identify a set of weights that guarantees that for every input vector, the output vector generated by the network is identical to (or sufficiently close to) the desired output vector.

Note: The actual or desired output values of the hidden neurons are not explicitly specified by the task.

For a fixed and finite training set: The objective function represents the total error between the desired and actual outputs of all the output neurons for all the training patterns.

Error Function

$$E = \frac{1}{2} \sum_{p}^{P} \sum_{i}^{N} \left(y_{ip} - d_{ip}\right)^2 \tag{2.2}$$

[2]

where P is the number of training patterns, N the number of output neurons, d_{ip} is the desired output for pattern p, y_{ip} the actual output of the neuron i and output neuron i.

Procedure

1. Compute the actual output for the presented pattern
2. Compare the actual output with the desired output
3. Adjustment of weights and thresholds against the gradient of the error function (Equation (2.2)) for each layer from the output layer towards the input layer

Figure 2.2 – Training Procedure of the backpropagation algorithm

Adjustment Rules

$$\begin{aligned} w_{ij}(t+1) &= w_{ij} + \Delta_E w_{ij}(t) \\ \Delta_E w_{ij} &= -\frac{\partial E}{\partial w_{ij}} = -\frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial \xi_j} \frac{\partial \xi_j}{\partial w_{ij}} \end{aligned} \tag{2.3}$$

where $\Delta_E w_{ij}$ denotes the change of the Error Function with respect to w_{ij}, E the Error Function, y_j the output of the output neuron j, ξ_j the potential of the neuron j and w_{ij} the weight with index i at layer j.

2.2.3 LSTM Neural Networks

2.3 Software Engineering

3 Methodology

3.1 AI Engineering

3.1.1 Data

3.1.2 Data Cleaning

3.1.3 Deep Learning Model

3.2 Software Engineering

3.2.1 Backend

3.2.1.1 Technologies

3.2.1.2 Architecture

3.2.2 Fronttend

3.2.2.1 Technologies

3.2.3 Test Concept

4 Results

4.1 Prediction Results

4.2 Software Results

5 Discussion and Outlook

Bibliography

[1] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Cham: Springer International Publishing, 2023. doi: 10.1007/978-3-031-29642-0°.

[2] I. Mrázová, "Multi-Layered Neural Networks."

[3] "NN SVG." Accessed: May 03, 2025. [Online]. Available: http://alexlenail.me/NN-SVG/°

List of Figures

List of Tables

A Appendix

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.