

UNIVERSITY OF BERGEN
DEPARTMENT OF INFORMATICS

GNN

Author: Bendik Akselsen Solevåg

Supervisors: Ramin Hasibi and Tom Michoel



UNIVERSITETET I BERGEN
Det matematisk-naturvitenskapelige fakultet

February, 2023

Abstract

Abstract

Acknowledgements

Acknowledgements

Bendik Solevåg
Wednesday 1st February, 2023

Contents

1	Introduction	1
2	Background	2
2.1	Machine Learning	2
2.1.1	What is machine learning?	2
2.1.2	Linear Regression	3
2.1.3	Loss	4
2.1.4	Multi Layer Perceptron	6
2.1.5	Neural Networks	6
2.1.6	Backpropagation	7
2.1.7	Supervised and unsupervised learning	7
2.2	Graphs	7
2.2.1	Nodes and Edges	7
2.2.2	Directed Acyclic Graphs	7
2.3	Graph Neural Networks	7
2.3.1	The Convolution Layer	7
2.3.2	Permutation invariance and equivariance	7
2.3.3	Attention and Message Passing	7
2.4	Gated Recurrent Units	7
2.4.1	Backpropagation Through Time	7
3	Project methodology and setup	8
3.1	The Geauvadis Dataset	8
	Bibliography	9

List of Figures

2.1	Linear regression illustrated	4
2.2	Different loss functions applied to the same prediction function	5

List of Tables

Listings

Chapter 1

Introduction

Chapter 2

Background

This chapter aims to explain the basic concepts that are the most important to the work in this thesis.

2.1 Machine Learning

The following sections will define the term 'Machine Learning'. It will describe some various ways that machine learning can be applied.

2.1.1 What is machine learning?

As a term, 'Machine Learning' is a subcategory of the umbrella term 'Artificial Intelligence'. It is proposed as an alternative to traditional algorithms. Machine Learning takes a set of input (training) data, and attempts to reason about some quality of the input, without the author of the program explicitly telling the program what quality of the input data we are interested in reasoning about. Instead, the author provides the Machine Learning model with their optimal target for the given input, and the model must attempt to generalize over, and design its own algorithm to fit the target.

This approach has become useful in problems where discovering the target based on the model input becomes computationally intractable, or when the target cannot be determined as a direct consequence of the input. An example of such a problem is

sentiment analysis of text input. Given the sentence 'The nice boy made fun of the kind girl', it is difficult to design rules which can capture the sentiment of the sentence. On one hand, the words 'nice', 'fun', and 'kind' indicate this sentence may have a positive sentiment. In reality, our ability to reason tells us that this is not the case, and the sentence is of a negative nature.

Machine Learning is deeply rooted in Bayesian Statistics. (Noen gode setninger om bayes. Gjerne bayes teorem? Ikke mer enn et avsnitt. Dette er en AI-oppgave, ikke en statistikk-oppgave.)

2.1.2 Linear Regression

As a first study in the implementation of machine learning, we look at linear regression. To explain linear regression, we view an example in a two dimensional, euclidean space. We are given a set of input data values X . Each input value $x_i \in X$ has a corresponding output value $y_i \in Y$. We can plot each input value, and its corresponding output value as coordinates.

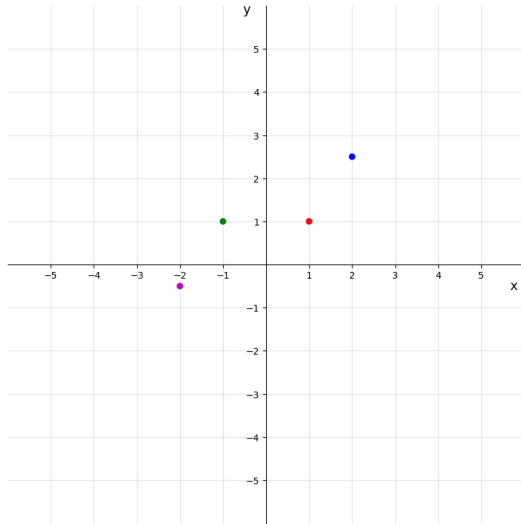
To describe a function that best maps the given inputs with their related outputs, we introduce two variables w_0 and b_0 . We define a function to map our input values to their corresponding output.

$$f(x) = w_0x + b \tag{2.1}$$

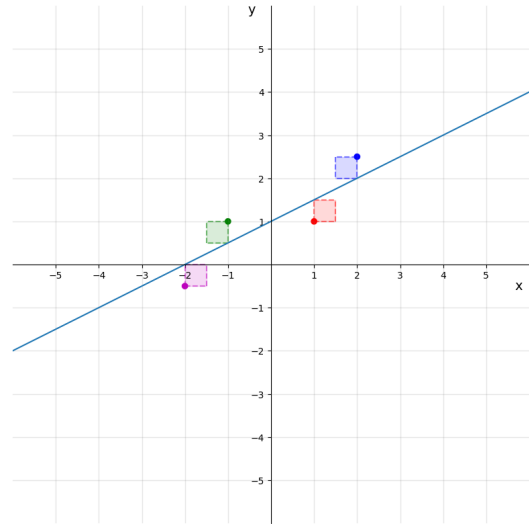
We also introduce some measure of how well our function (2.1) maps our input to our output. As we tune our function variables w_0 and b_0 to better map the input to its corresponding output, the measure will decrease. There exists values for our variables w_0 and b_0 such that the measure defined in (??) reaches its lowest possible value. In this state, the function optimally maps the input data to with its corresponding output, given our function.

$$\mathcal{L} = \frac{1}{|X|} \sum_{i=0}^{|X|} (y_i - f(x_i))^2 \tag{2.2}$$

Of course, the function defined in (2.1) is arbitrary. We can add or remove existing or additional function variables as we wish, if we see that changing the function structure



(a) Input x values plotted against their corresponding y values



(b) Function (2.2) applied to function (2.1) with optimized function variables w_0 and b_0

Figure 2.1: Linear regression illustrated

improves at mapping the input to the output. This example has applied linear regression to single-dimensional inputs and single-dimensional outputs, but the same method can be applied to inputs and outputs of higher dimensions. The method can also be applied when the inputs and outputs are of different dimensions, by applying linear transformations. A general formula for linear regression can therefore be

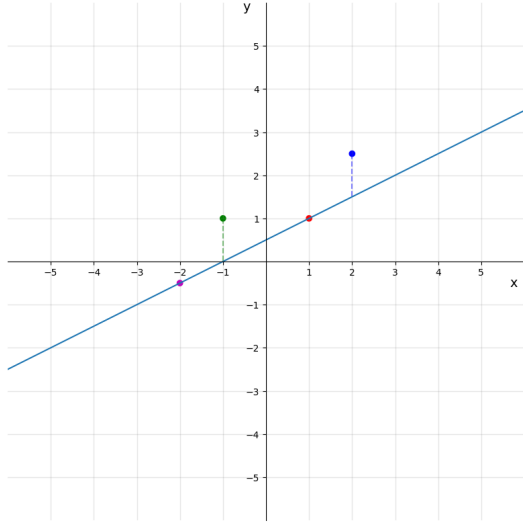
$$X \in \mathbb{R}^m, \{Y, b\} \in \mathbb{R}^n, W \in \mathbb{R}^{m \times n}$$

$$f(X) = WX + b \quad (2.3)$$

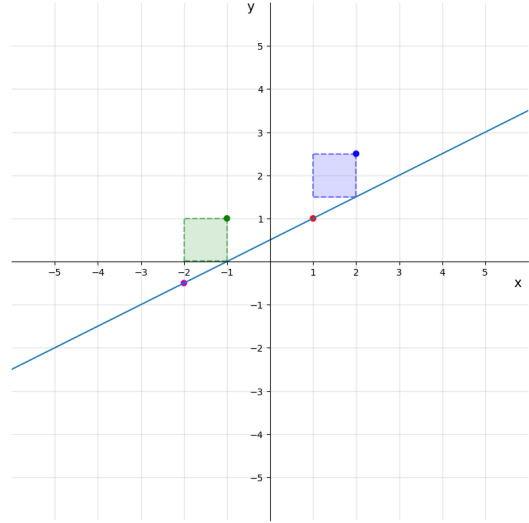
Referanse til taylor series expansion? Finn i Calculus bok

2.1.3 Loss

In the previous section, a measure of how 'wrong' a mapping of input points to their corresponding output points was defined. In the literature, this measure is referred to as the loss function. The loss function defined in (2.2) is named 'Mean Squared Error Loss', due to the fact that it is defined by the mean of the squared difference between our function's predictions and the true values. The 'Mean Squared Error Loss' operator is referred to as 'MSE' for the rest of this thesis.



(a) Loss function (2.4) applied to function (2.1) with optimized function variables w_0 and b_0



(b) Loss function (2.2) applied to function (2.1) with function variables w_0 and b_0 optimised using loss function (2.4)

Figure 2.2: Different loss functions applied to the same prediction function

It is possible to define different loss functions. A loss function must be nonnegative for all input values. Different loss functions will interpret the performance of a functions mapping of the given input data to its corresponding output in different ways. The difference in ... can be illustrated by observing the result of applying the loss function defined in (2.4), typically referred to as 'Mean Absolute Error Loss', shortened to 'MAE' in this thesis, as opposed using MSE.

$$\mathcal{L} = \frac{1}{|X|} \sum_{i=0}^{|X|} \sqrt{(y_i - f(x_i))^2} \quad (2.4)$$

While MSE loss sums over the squares of errors, MAE sums over the absolute errors. Using MAE, each prediction's error term will grow linearly, given the true output value. Using MSE, however, each prediction's error term will grow exponentially, given the true output value. As a consequence, 'MAE' will place equal weight emphasis on each prediction's error. MSE will place more emphasis on predictions that are further from the true output values. Different loss functions can therefore be applied to different problems to find different solutions. Figure (2.2) illustrates an example of different loss functions being applied to the same prediction function. The optimal function parameters were found using the MAE operator. When the MSE operator is applied to the same function parameters, the MSE operator does not produce its lowest possible value.

The loss functions presented so far in this section have all been focused on the regression task in machine learning applications. For prediction tasks, a better suited loss function should be used. A common loss function for this kind of task is the cross entropy loss function, defined in (2.5)

$$\mathcal{L} = - \sum_{i=0}^{|X|} p(y_i) \log(f(x_i)) \quad (2.5)$$

2.1.4 Multi Layer Perceptron

2.1.5 Neural Networks

underfitting and overfitting?

Linear Layers

Activation functions

2.1.6 Backpropagation

2.1.7 Supervised and unsupervised learning

2.2 Graphs

2.2.1 Nodes and Edges

2.2.2 Directed Acyclic Graphs

2.3 Graph Neural Networks

2.3.1 The Convolution Layer

2.3.2 Permutation invariance and equivariance

2.3.3 Attention and Message Passing

2.4 Gated Recurrent Units

2.4.1 Backpropagation Through Time

Chapter 3

Project methodology and setup

3.1 The Geauvadis Dataset

Bibliography