university of groningen

faculty of science and engineering

Bachelor Reasearch Project

UTILISING MACHINE LEARNING TO PREDICT THE SOLAR SPECTRUM FROM ALL DIRECTIONS

October 27, 2020

M.J. Blum S3483509

Supervisor: Dr. Bruno Ehrler (AMOLF), Benjamin Daiber (AMOLF)
First Examiner: Prof. Dr. L.J.A. (Jan Anton) Koster (RUG)
Second Examiner: Prof. Dr. Maria Loi (RUG)

Abstract

To make a prediction of the performance of solar panels, a sensor called LAD has been installed that measures the intensities of four colours (blue, green, red and infrared) as incident from sunlight in twelve directions. From the data collected by this sensor, the goal is to predict a continuous spectrum from these four intensities. Ultimately, the goal is to use these intensities from the other directions to be able to predict the performance of bifacial solar cells (irradiated from two sides) with the knowledge on diffuse light (not directly incident from the sun).

For this prediction of a continuous solar spectrum for wavelengths on the interval 280 nm to 1121 nm, an artificial neural network (ANN) has been trained. The network has one layer, connecting in- and output linearly, as this is the system benefiting most from training, significantly improving the prediction with higher volume of data and yielding the best results compared to other architectures. Training this network requires two sets of data, both measurements of the solar spectrum (desired output) as well as the corresponding measurements of the LAD (input).

Data collected in summer features the highest intensities, the most important information for solar cells. Unfortunately, the collected data contains many unphysical measurements, most likely due to reflections of surrounding buildings. These were too many to take out manually in the framework of this project, so the data is cut off at the intensities from where wrong measurements start to occur. The result is a network that can predict the spectrum very well for low intensities (irradiance up to 400 $\mathrm{W\,m^{-2}}$), with less than 2% error in the irradiance on average over more than 30000 measurements.

It was shown that the higher-intensity data shows systematic problems with the predictor model, so that future research needs to address the issue of the outliers first. Then, this result can be used to apply to the rest of the data collected by the LAD to build a forecasting model for diffuseness, broken down for an entire spectrum.

Acknowledgements

I want to thank Benjamin Daibler for always being available for questions, genuinely caring for my research and regularly checking up on me and calling for extended periods of time, always being able to help. Thank you for everything you taught me.

Additional thanks are extended to Dr. Bruno Ehrler both for organising my internship and so wholeheartedly welcoming me to the team, especially during these crazy times. I wish I can have managers like you in the future.

The entire team has been very welcoming and especially during the quarantine time managed to still make me feel like part of the team. This means a lot to me, and my thanks go out to every member of the Hybrid Solar Cells group at AMOLF.

Lastly, I want to thank my examiner, Prof. Dr. L.J.A. (Jan Anton) Koster, for putting me in touch with Dr. Ehrler and as such making my wish of a project outside the university on solar cells come true.

Contents

Chapter 1

Introduction

In the framework of the bachelor's research project in Applied Physics at the University of Groningen, I conducted an internship at research institute AMOLF in Amsterdam from April to July 2020. I was granted a spot among Dr. Bruno Ehrler's research group on *hybrid solar cells*. Due to the measures around the Coronavirus, the project was fully taken on from a distance, with the focus lying on a pure coding project.

To predict the spectra of diffuse light for the use of improving the positioning of a bifacial solar cell (absorbs light from both sides), previous interns at AMOLF created a sensor called LAD measuring the light intensity in twelve different directions[1][2]. From the data collected with this device, the next steps are now to both interpolate the discrete measurements into a continuous spectrum and generalise the findings for every angle, interpolating over the twelve discrete orientations. With the deepened understanding of diffuseness gained, it will be easy to now improve the orientation of solar cells, specifically bifacial cells and increase the energy gained by the cell.

This thesis will thus be concerned with solving the above problem, specifically finding a way to predict a continuous light spectrum from the data collected by the LAD, paving the way for follow-up research to interpolate this prediction over the entire sphere. This prediction will be performed using the program Wolfram Mathematica, Version 12.1[1]. All results and information required to reproduce the results presented in this thesis can be found at this link [2].

Throughout the research presented in this thesis, the method employed is machine learning, specifically aritficial neural networks (ANN). These will be introduced in more detail and their use also motivated in Chapter 2. The four chapters this thesis is divided into are Background, Method, Results and Conclusion.

Chapter 2, Background will introduce fundamental theory required for performing and motivating this research. It will describe a short introduction into photovoltaics, present the data utilised in this research and discuss basics of machine learning and motivates aand discusses its use. Chapter 3, Methodology will introduce the method to building the

[1] Find an entire beginners guide online at https://www.wolfram.com/language/elementary-introduction/2nd-ed/.

[2] https://drive.google.com/drive/folders/1tAS6Jo5NrayJm7grV6Gizy1R0LhLmkQT?usp=sharing

predictor model, showing how the data is appropriately formatted and how to arrive at the final specifics of the model. In the fourth chapter, Results and Discussion, the results from following the procedure described in the previous chapter will be shown and discussed, presenting the final network and discussing its potential future applications. Ultimately, the work will be summarised and shown how to use these insights in follow-up research.

Chapter 2

Background

2.1 Solar Cell Physics

To put the predictor model into perspective, this section will show some basic principles of solar cells. Since the ultimate goal is to help find the ideal orientation for a (bifacial) solar cell, it first needs to be understood what exactly is to be optimised for.

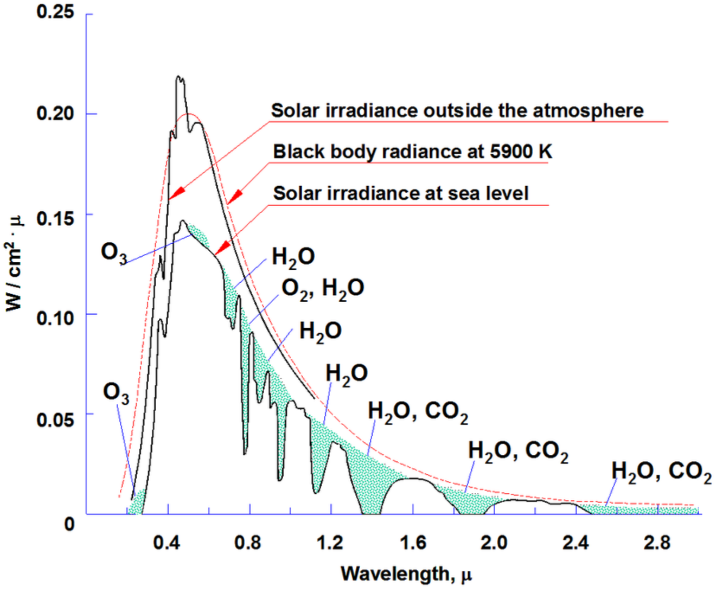


Figure 2.1: AM1.5G tilt spectrum, illustrated with the associated absorption spectrum. The image was taken from [3].

Given that the aim is to predict a continuous spectrum of sunlight from collected data, the focus should first be laid on the AM1.5 spectrum. Essentially, this is a standardised measurement of the solar spectrum actually reaching the earth's surface. The number 1.5 in the name indicates that the distance travelled through the atmosphere is not exactly the thickness of the atmosphere, but its 1.5-fold, AM for air mass. This is because sunlight typically comes in at an angle. Figure 2.1 shows a AM1.5 spectrum. Typically, this spectrum is broken down per wavelength in the unit $\mathrm{W\,m^{-2}\,nm^{-1}}$, so as the irradiance $[\mathrm{W\,m^{-2}}]$ per wavelength $[\mathrm{nm^{-1}}]$.

This is calculated by taking the emission of the sun as a blackbody and subtracting all the absorbed parts. Each absorption peak can be associated with a certain material absorbing, e.g. O_3 or water vapour. The most prominent absorption peaks in the graph are associated with water vapour in the atmosphere near 650 nm, 700 nm [4], 970 nm and 1200 nm [5], see figure 2.1. Further absorption is associated with Ozone, Oxygen and CO_2 [3].

For the predictions as presented in this thesis, not the entire spectrum as depicted in Figure 2.1 will be used. The measurement tool only provides information on the spectrum relevant to Silicon cells, i.e. light promoting electrons across the band gap, with energies higher than 1.1 eV [6], the energy carried by a photon of wavelength $\lambda_{max} \approx 1120\,\mathrm{nm}$. This is the interval of interest due to the fact that silicon solar cells are the commercially most widely available.

Combining the above, the total amount of usable power incident on a unit of area of a silicon cell can be calculated as the integral of the (AM1.5) spectrum over the relevant wavelengths, called irradiance. As a reference value for more calculated irradiance values throughout this paper, here is the irradiance for the "ideal" AM1.5 spectrum:

$$\int_{280nm}^{1120nm} I_{AM1.5}(\lambda) d\lambda \approx 1\,\mathrm{kW} \tag{2.1}$$

There is much more information available on solar cells, but the above information is enough to put the predictor model into perspective.

2.2 Light Ambient Detector

2.2.1 Geometry and functioning principle

As a tool to determine the diffusivity of sunlight, a sensor to measure intensities coming in from different directions was built at AMOLF. The "Light Ambient Detector" (LAD) effectively measures the intensities in 12 different directions, realised as a dodecahedron with sensors on each surface. To protect it from weather, it is encapsuled in a transparent sphere. The device can be seen in Figure 2.2.

2.2.2 Data

Every one of the twelve directions the LAD measures four different intensities and temperature. These measurements are performed every ten seconds. Figure 2.3 shows the output of the device. This output can be broken down into 14 lines per measurement, of which the first two provide the time stamp of each measurement. The latter 12 lines correspond

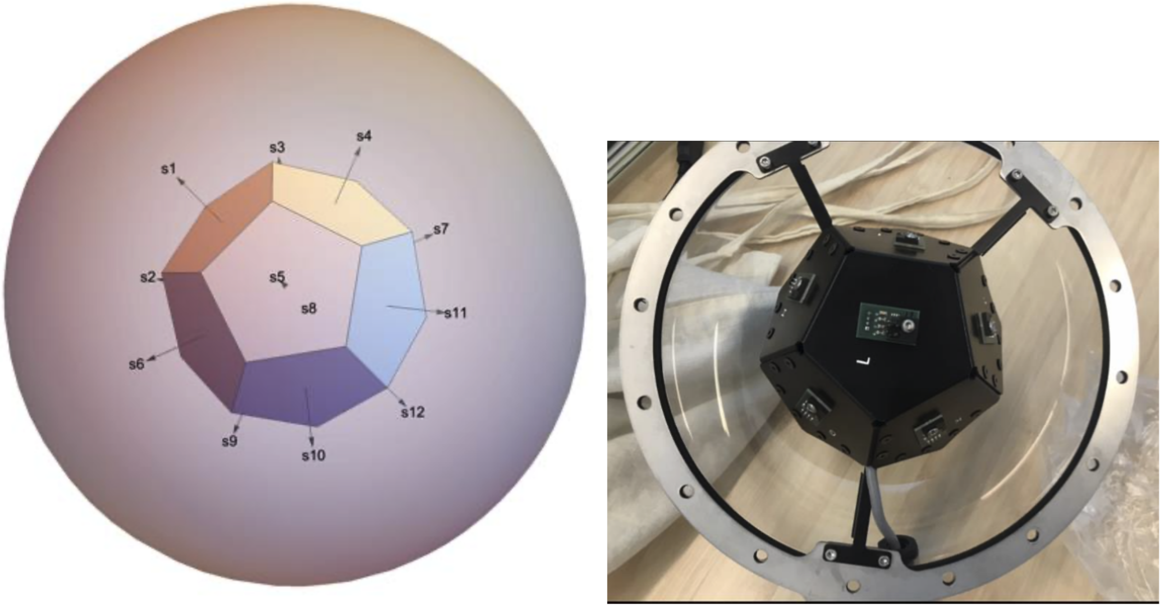


Figure 2.2: Depiction of the Sensor LAD, as a simulated dodecahedron (left) and as an image taken during assembly with only half the protective sphere attached. Image taken from [1]

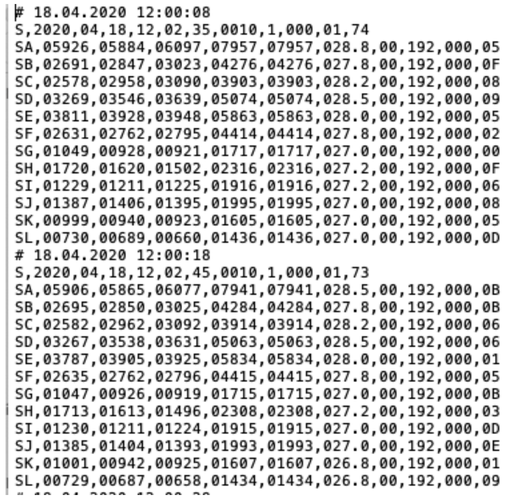
to the measurements as performed by each direction. These lines can then be respectively be broken down as follows:

First, SA, SB etc. indicate which sensor is reporting measurements. SA is the sensor aligned with the solar panels and the spectroradiometer, and will thus be used for the training of the network. The following five-digit numbers are the measured intensities for the red, green, blue and IR neighbourhood in relative units. The exact wavelengths that the respective sensor measures for are given in Table 2.1. The last five-digit number is a repetition of the IR intensity (a fragment of the working principle of the sensors), succeeded by the temperature and some system information that can be looked up in the manual.

The LAD was calibrated in December 2018 and has collected 15 months worth of data since.

2.2.3 Previous Work

Former AMOLF interns Andrea Pallostri [1] and Merlijn Kersten [2] have spent their time working with the LAD. In 2018, Kersten proceeded to simulate the increase in effectivity if the previous cubic sensor were to be extended to having more sides. He found out that given the shading of the transparent sphere surrounding it, the sensor would yield a high quality with 12 sides. In the succeeding year, Pallostri finalised the build and calibration of this sensor and has left it running ever since.



```
# 18.04.2020 12:00:08
S,2020,04,18,12,02,35,0010,1,000,01,74
SA,05926,05884,06097,07957,07957,028.8,00,192,000,05
SB,02691,02847,03023,04276,04276,027.8,00,192,000,0F
SC,02578,02958,03090,03903,03903,028.2,00,192,000,08
SD,03269,03546,03639,05074,05074,028.5,00,192,000,09
SE,03811,03928,03948,05863,05863,028.0,00,192,000,05
SF,02631,02762,02795,04414,04414,027.8,00,192,000,02
SG,01049,00928,00921,01717,01717,027.0,00,192,000,00
SH,01720,01620,01502,02316,02316,027.2,00,192,000,0F
SI,01229,01211,01225,01916,01916,027.2,00,192,000,06
SJ,01387,01406,01395,01995,01995,027.0,00,192,000,08
SK,00999,00940,00923,01605,01605,027.0,00,192,000,05
SL,00730,00689,00660,01436,01436,027.0,00,192,000,0D
# 18.04.2020 12:00:18
S,2020,04,18,12,02,45,0010,1,000,01,73
SA,05906,05865,06077,07941,07941,028.5,00,192,000,0B
SB,02695,02850,03025,04284,04284,027.8,00,192,000,0B
SC,02582,02962,03092,03914,03914,028.2,00,192,000,06
SD,03267,03538,03631,05063,05063,028.5,00,192,000,06
SE,03787,03905,03925,05834,05834,028.0,00,192,000,01
SF,02635,02762,02796,04415,04415,027.8,00,192,000,05
SG,01047,00926,00919,01715,01715,027.0,00,192,000,0B
SH,01713,01613,01496,02308,02308,027.2,00,192,000,03
SI,01230,01211,01224,01915,01915,027.0,00,192,000,0D
SJ,01385,01404,01393,01993,01993,027.0,00,192,000,0E
SK,01001,00942,00925,01607,01607,026.8,00,192,000,01
SL,00729,00687,00658,01434,01434,026.8,00,192,000,09
```

Figure 2.3: The output as given directly from the LAD, as taken from April 18, 2019

Colour	Wavelength	Maximum measured
Red	615nm	40293
Green	525nm	33456
Blue	465nm	34052
IR	850nm	49221
Temp		0.8-64.5°C

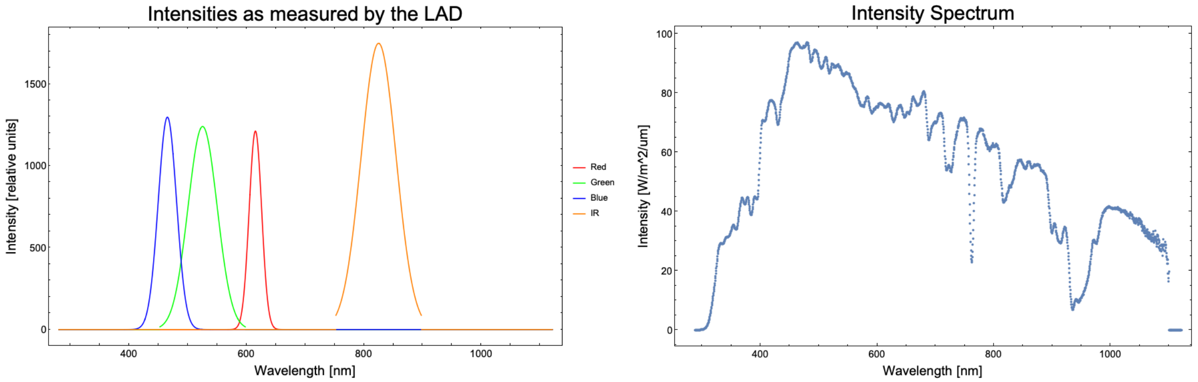
Table 2.1: The colours and wavelengths that the individual sensors measure for. It should be noted that every sensor measures for a range of wavelengths, peaking around the given lengths.

2.2.4 Spectral measurement

The goal of this thesis is to predict a continuous spectrum like the AM1.5 spectrum from solely the four measured intensities of LAD. At this point, it is not certain that this is at all possible, as it would require the solar spectrum to have mere 4 degrees of freedom. To do this, reference data is needed that will give a correlation between this collected set and the continuous spectrum. This reference data is supplied by a spectroradiometer stationed near the LAD. This will provide measurements of the same time, i.e. the exact same weather conditions for both the used and the desired data. Momentary differences in shading between the two sensors, e.g. by leaves of a tree blown up by wind are to be expected.

The data this spectrometer collects is the spectral irradiance in $[\frac{W}{m^2 \mu m}]$ in 2048 intervals for the spectrum concerning the silicon solar cell, i.e. 280nm to 1120nm. All collected data is saved as 12 measurements (one hour of data) per file, stored as a .csv file.

Figure 2.4 displays the data collected by both sensors, giving a graphical representation of the goal of this research with 2.4a giving a idea for the spectral region that the intensities



(a) Measurements from the LAD, used as input for prediction.

(b) Measurements of the spectrum, desired output of the predictor.

Figure 2.4: Graphical representation of the aim of this research. In (a) the intensities as measured by the LAD that are used for predicting the spectrum (b) as measured by a spectroradiometer.

are measured for and 2.4b showing an idea .

2.3 Machine Learning

2.3.1 Introduction to Machine learning

This research aims to predict a continuous spectrum of intensities using measurements that do not exhibit a clearly known relationship to said spectrum. To be able to find a relationship "automatically", machine learning (ML) will be applied. ML can be very simply understood as an algorithm that automatically adapts its parameters by trying to optimise its output with respect to some data that is provided, not unlike fitting. In contrast to fitting, machine learning is able to learn repeatedly from the same data and can handle significantly larger volumes of data. Furthermore, it will reference the quality of the prediction to a separate subset of the provided data to validate the results. The term learning in its name stems from the fact that the algorithm will perform better when gaining more "experience", i.e. processing more examples.

ML is a very popular method to use in different areas of data processing, e.g. image recognition [7] or personalised advertisements [8], because it costs less computational power and is easier to use than many alternatives. The less computational power is due to the fact that the program does not require to solve any analytical solutions but rather forces its method to output a close approximation to the desired result. By its popular demand in solving complicated problems, different programming platforms have extended their coverage of machine learning.

In this work we have used the implementation of Machine Learning in Wolfram Mathematica 12. There is a large online library of resources available online to introduce machine learning. The combination of this with their easy-to-use, computationally cheap program motivated the use of this platform.

2.3.2 Why Machine learning?

On a more abstract level, the central problem of this research is the prediction of 2048 real numbers (irradiances) from four integers (intensities) and one real number (temperature). Hence there are a total of 2053 dimensions to this problem. Assuming there is a linear relationship between in- and output of the system, this leaves (in the best case) still for $5 \times 2048 = 10240$ linear equations $a \times x + b$ to solve for, bringing with it 20480 unknowns. To solve for these unknowns analytically, a gigantic system of equations would have to be set up. This is all presuming that there is a linear relationship, assuming there is a physical link between the data collected by the two sensors.

This last point is crucial. For instance, the IR measurement in the LAD can be drastically lowered by an increase of water in the atmosphere above it. This implies that the ratio of the other intensities to IR could be similar in the case of a very humid and a cloudy day, whilst the spectrum is vastly different for the two. This specific example might not be the case, but is likely that two identical LAD measurements might indicate vastly different spectra. In this case, the analytical solution can not be found, wheras ML will converge to the most likely solution.

This is circumvented by using a machine learning algorithm, that can improve its approximation with every set of data added. Other methods of approximation, such as linear interpolation, always need an appropriate initial guess, a certain assumption from the start of how to connect the two sets of data, whereas ML will just impose a connection upon them and refine it. It is important to note that ML need not yield a model that carries with it any physical significance in its parameters.

Over the course of the measurements of both LAD and spectroradiometer, a volume of multiple gigabytes of data have been collected. A further advantage of ML is that all of this data can be used, even with a simple computer, because it can be applied in small samples, as the algorithm keeps learning.

2.3.3 Methods

So far, the concept of machine learning has been introduced solely for the abstract notion of a self-learning algorithm. There are different types of algorithms dedicated to machine learning. Most prominently, machine learning is often understood as a synonym for artificial neural networks (ANN), that imitates the neurons in brains. Besides these, there are other methods, that are equally easy to implement within Mathematica.

Notably, there are geometric methods that are easy to understand. An example of this is kNN, k-nearest neighbours. It will plot all training data and will classify new data by the values of the k (amount) nearest points. In the present example, this is unfeasible, given the high dimensionality of the problem. The data can be reduced in dimensionality using

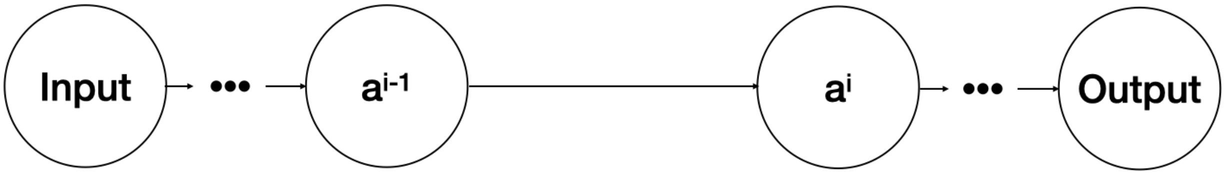


Figure 2.5: Sketch of a neural network with any number of layers, each featuring one single neuron. Depicted are the Input and Output, along with two neighbouring layers, called i and i-1. Since there is only one neuron per layer, each activation can be found interatively by equation 2.2

dimensional reduction techniques as described by Aggarwal and Reddy [9]. The computational effort coming with this is chosen as motivation to not look into geometric methods, although follow-up research is to check the validity of following the method described in this thesis. A similar problem can be encountered when applying decision tree-based algorithms. These will result in such big and complicated trees that it is computationally near impossible to compute on a normal computer.

For the methods available in Mathematica, this leaves only two feasible options: Linear Regression and ANN. ANN will be chosen for two simple reasons. Firstly, the use of ANN are well-documented and secondly can a liner regression be performed by a linear neural network.

2.3.4 Artificial Neural Network

Mathematical description

This method of a self-learning algorithm is the programmed equivalent to how a brain works. Roughly speaking, this means that there are neurons that contain some information, called *activation*, that will be broadcast to a next neuron via a *transfer function*. These neurons are aligned in layers, with one neuron of the one layer connected to all neurons of the next layer.

Mathematically, it is most simple to look at a neural network of one neuron in each layer, as depicted in figure 2.5. The activation of the neuron in the i-th layer will be denoted as a^i. The transfer function f will change depending on the specification of the network, but the connection will always be of the form

$$a^{i+1} = f\big(w^i \times a^i + b^i\big) \tag{2.2}$$

where the transfer function will map the new activation onto the desired range. b^i and w^i are the parameters of the network that will be optimised during training, known as *bias* and *weight* of the neuron.

This simple model is expanded for any network by adding a subscript indicating the neurons position within the layer, following Nielsen's nomenclature [10]: a^i_n. Now, since the activation of one neuron depends on all activations of the previous layer's neurons, it

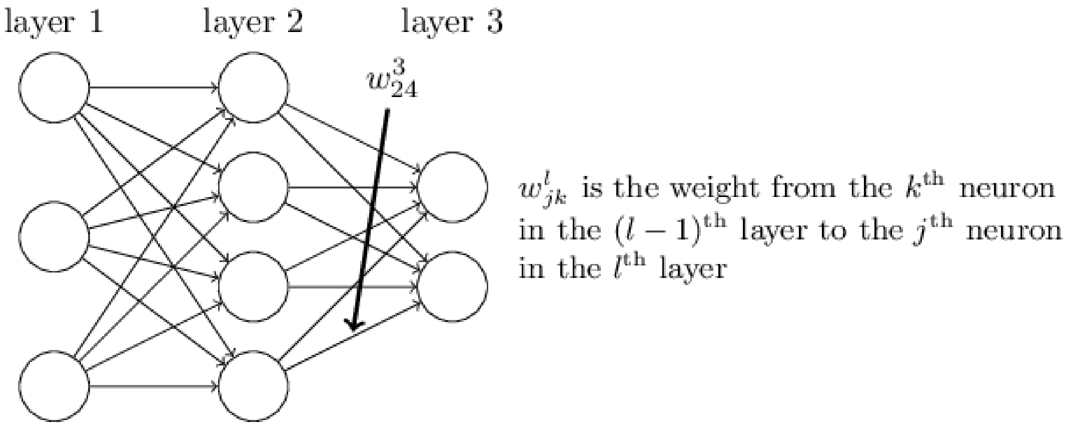

layer 1
layer 2
layer 3
w^3_{24}
w^l_{jk} is the weight from the k^{th} neuron in the $(l-1)^{\text{th}}$ layer to the j^{th} neuron in the l^{th} layer

Figure 2.6: A schematic image showing a three-layered ANN featuring the weight of one specific path from one neuron to a next to clarify the nomenclature. The image was taken from [10].

will be described as:

$$a_n^{i+1} = f\Big(\sum_{m=1}^{N_i} \big(w_{nm}^i \times a_m^i\big) + b_n^i\Big) \tag{2.3}$$

At this point, it is worth spending some time on the nomenclature. In the above equation, N_i indicates the number of neurons in the i-th layer. The weight w_{nm}^i describes the weight *from* the m-th neuron in the $(i-1)$-th layer *to* the n-th neuron in the i-th layer, as shown in Figure 2.6. In the following, an attempt will be made to describe the simplest system that is possible for the present problem. Specifically, the transfer function will be $f(a) = a$. The bias has a more intuitive naming. b_n^i describes the bias of the n-th neuron in the i-th layer [10].

Recall that the problem takes an input of 5 numbers and returns 2048. This can be translated into a neural network with one layer (input does not count as a layer) with five inputs, resulting in 2048 equations with ten degrees of freedom each:

$$a_n^1 = \sum_{m=1}^{5} w_{nm}^1 \times a_m^0 + b_n \tag{2.4}$$

This equation for one neuron in the only layer can be generalised to all neurons by rewriting it as a system of equations in vector form. Here, $\mathbf{a}_1$ and $\mathbf{B}$ are length-2048 vectors and $\mathbf{a}_0$ is a length-5 vector with the input values. The weights are represented in a 2048x5 matrix $\mathbf{W}$.

$$\mathbf{a}^1 = \mathbf{W}\mathbf{a}^0 + \mathbf{B} \tag{2.5}$$

The network can now be designed in a more complex way, by adding layers with different numbers of neurons and by changing the transfer functions. The latter option is of grave importance for applications of classification. If the goal is to build a classifier for e.g.

considering whether or not it was raining during a measurement then the output should be a value between 0 and 1 (1 = rain, 0 = no rain). A normalisation of all activations can be obtained by applying a special function, e.g. a logistic sigmoid or the hyperbolic tangent. Increasing the number of layers can increase the accuracy of the method but will come at the cost of computational speed and will possibly also data.

Training

Given all data and understanding how the networks are built, there is now the question of how to train this network. This section will only describe the way the used program, Wolfram Mathematica, trains the algorithm. To understand training, two concepts will have to be introduced: *Loss* and *Gradient descent*.

Training data consists of both input data and actual correct output data, in the present case LAD and spectrum data. This data will be split into "batches" that are processable in size with the available RAM. Every individual example will be evaluated by the network and the difference between prediction and desired output taken. This difference is called *loss* in Mathematica, or *cost* in different publications[10][11]. Mathematica gives the opportunity to specify the desired loss (or cost) function. In this case, the mean absolute loss function will be chosen, to measure the absolute deviation. In mathematical terms, this means that the computed output $\tilde{\mathbf{y}}$ and the measured output $\mathbf{y}$ combine into the loss l as follows:

$$l = \frac{1}{2048} \sum_{i=1}^{2048} |\tilde{y}_i - y_i| \tag{2.6}$$

Where 2048 is the length of the output in the present predictor model. This average is taken because it is given as a predefined function in Mathematica, since a prefactor as $\frac{1}{2048}$ does not change the outcome. After every batch, the losses of all examples in the batch are averaged and give a quantification of how strongly the parameters are to be altered. To increase or decrease the parameters is determined by computing the gradient of the algorithm with respect to all system parameters (partial derivatives with respect to all b and w). In the simple example above (equation 2.5), this means that the gradient is taken with respect to all W_{ij} and B_i. Then the whole function moves a step into the direction of the negative gradient by a step size corresponding to the size of the loss [11].

This method is called *mini-batch gradient descent* and will approach a local minimum of the error function after enough iterations (batches processed). Computationally, the main drawback of this method is that a global minimum (most optimal system parameters) might never be realised because it will converge to a local minimum by a wrong choice of initially guessed parameters. The advantage of this method is that it will always converge to a minimum at very little requirement in terms of RAM or processing power [12].

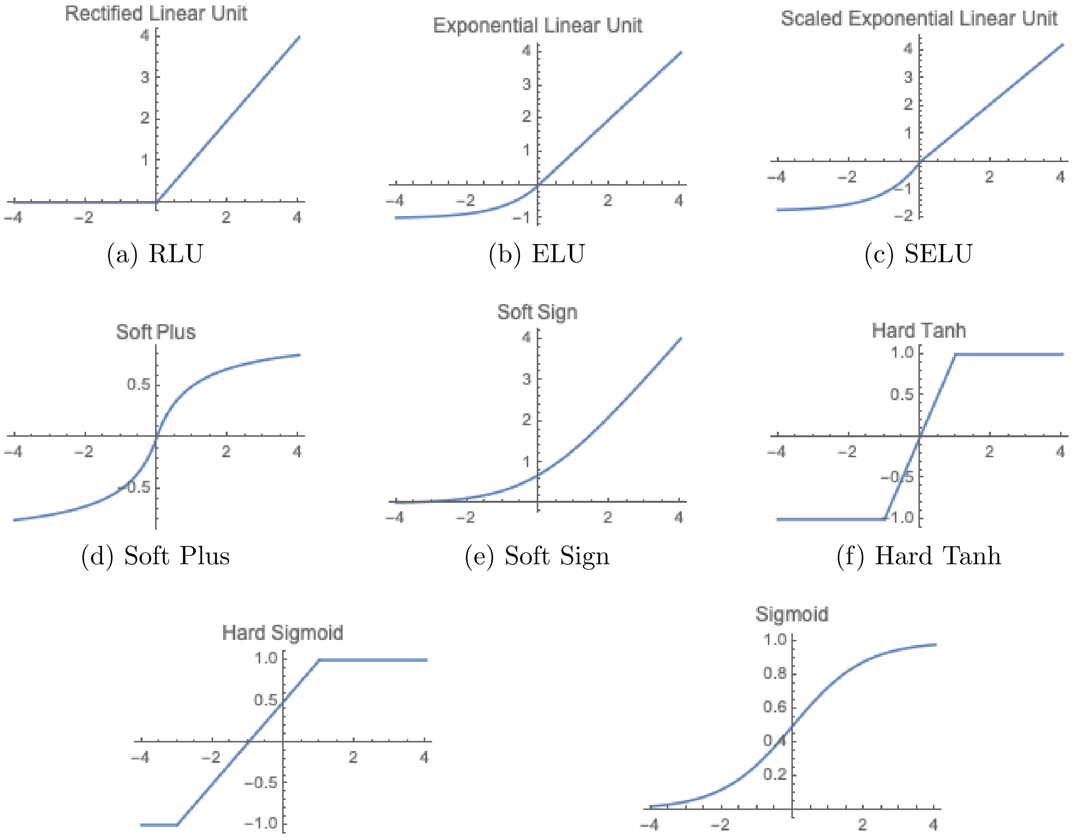
After all data is processed, another set of data will be evaluated. This "test set" or "validation set" has the sole purpose of estimating how well the predictor can predict data it has not been trained on. The loss associated to these predictions are referred to as "validation loss". This is the quantity that will generally be used in the computations run throughout this research to quantify the quality of the work. During the course of this research, the data was split up such that a randomly selected portion of 30% was used as validation set, whereas the rest was used as training data.

Transfer functions

Mathematica supports eight different transfer functions, each functioning as a layer between two (hidden) layers, subjecting the activation of the individual neurons to the function and just mapping them to the next neuron, so that both layers have the same length. Thus equation 2.3 is applied in two stages, first with $a_n^{i+1} = \sum_{m=1}^{N_i} \left(w_{nm}^i \times a_m^i \right) + b_n^i)$ and then the next layer imposing $a_{i+2} = f(a_{i+1})$ [13].

The specific transfer functions that can be used are given in Figure 2.7. It should be noted that functions 2.7d through 2.7h refer to normalised data, with values between -1 and 1. All functions show distinctly different behaviour for positive and negative activations [13]. For the comparison between linear connections and the use of a transfer function, the Scaled Exponential Unit (SELU) will be used, Depicted in Figure 2.7c and expessed as

$$f_{SELU}(x) = \begin{cases} 1.0507 * x \quad if \quad x \geq 0 \\ 1.7581(e^x - 1) \quad if \quad x < 0 \end{cases} \tag{2.7}$$



(a) RLU

(b) ELU

(c) SELU

(d) Soft Plus

(e) Soft Sign

(f) Hard Tanh

(g) Hard Sigmoid

(h) Sigmoid

Figure 2.7: Graphical representation of the transfer functions as provided by Mathematica. The images are taken from [13].

2.3.5 Other work in the field

Solar forecasting has been performed by many research groups in recent years as shown in the review paper by Pazikadin et. al. [14]. Forecasting is generally defined as predicting the solar irradiance for a certain time into the future, called forecast horizon. They discuss different types of data that are used for this forecasting, showcasing that for a prediction as desired in this research ideally the data collected by a pyranometer would be used. There are many examples of forecasting models achieved through the use of machine learning, for instance [15] and [16].

Chapter 3

Methodology

This chapter will be concerned with the methods and research strategy realised during the course of this internship. It will start from the raw data as collected by the respective sensors and motivate each decision made towards the final utilised product.

3.1 Data pretreatment

Any machine learning algorithm can only be as good as the quality of its data allows it to be. Therefore, it is crucial to select the data carefully and correctly. In the following, it will be explained what data is selected and how. The entire process can be followed in the (Wolfram Mathematica) notebook available at this link[1].

When the LAD data in the form of a .txt file as in Figure 2.3 is imported to mathematica, first the question arises which data is relevant for training. The only useful training data is that collected by the sensor aligned parallel to the spectroradiometer, i.e. sensor 1 or SA. Of this sensor, every five minutes of data are to be used.

First, the time stamp rows are taken out and amended to every line so that consequently every 12th line can be taken out, providing the data of the first sensor of the following form: {time stamp, intensities, temperature}. Adding the time stamp is necessary since the LAD measures at every 10s but will switch off every month and a half and not measure for unknown reasons for two minutes. Partly because of this and partly because the measurements are not exactly 10s apart but a few milli- or microseconds less, so that over the course of multiple weeks, the measurements will be "earlier". As an example, if the initial measurement was at 11:00:00am, then the corresponding measurement several weeks later will be taken at 10:59:45am.

Both these defects in the measurement require the data to be monitored closely, as programming an algorithm for this purpose of analysing this string is complicated. Clearly, this means that most of the data will not have been collected at the same moment in time, but seconds apart. This is a systematic error that is just to be accepted, in hope that the large volume of data will counteract this inaccuracy. The data is now chosen such

[1] https://drive.google.com/drive/folders/1tAS6Jo5NrayJm7grV6Gizy1R0LhLmkQT?usp=sharing

that the measurements will be at most five seconds apart. Special care has to be taken at the ”fringe values”, i.e. the values around the full hour, the first and last value of every text file. If the measurements are taken five seconds before and after the full hour, it can mistakenly be identified as two distinct measurements at the five minute mark, resulting in invalid data for all the consecutive data, with the two sensors shifted by one point.

When all LAD measurements in five minute intervals have been taken, there is a long list of six elements per entry. These are still as above, with the time stamp, measured intensities and temperature. The goal is now to match this with the same time interval of spectral data. For that the same time frame in spectral data is imported into Mathematica such that the two sets of data can be easily implemented into one list together. To combine the two sets in a way that the training function can easily understand, the in- and output are stored as a list of associations, i.e. input$-$ >output. In this form, the training will automatically be performed with the correct data associated to one another.

Should the final predictor function not operate consistently, wrong data manipulation (as described above) is most likely the reason for its failure.

Removing nighttime data

After training different networks, it has been explored whether it makes a difference to the ultimate accuracy (final loss) to exclude the data collected at night, with low intensities. To try this, a simple thresholding mechanism is used. All data featuring a red intensity smaller and larger (or equal) than a given threshold will be split up. For the sake of generalisation, these two datasets then train their separate networks and the loss of the two can be combined to an overall estimate how good the prediction is across all measurements, implementing two separate networks.

Beyond the initial thought that at night the measurements will not show a AM1.5-like spectrum, it is also imaginable that the sensors might overheat, be subjected to reflection from surrounding buildings or encounter other problems at very high irradiance. For instance, there are measurements of the radiometer that imply an irradiance one order of magnitude higher than the AM1.5 reference spectrum. Since this is unphysical, it might prove useful to exclude these measurements by such thresholding.

3.2 Artificial Neural Networks

3.2.1 Architectures

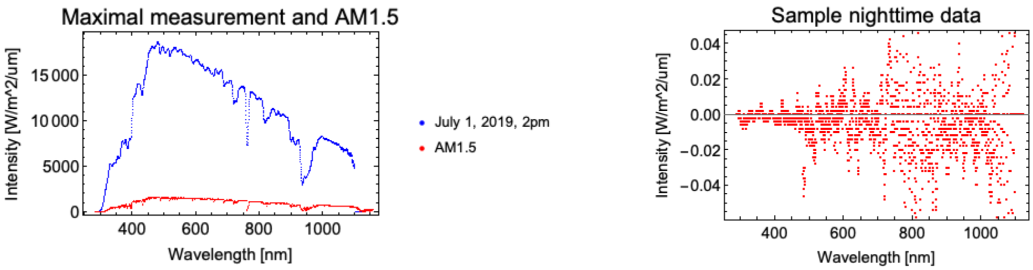
It is difficult to predict which type of network will be most effective for any application. Therefore, different networks will be trained and their performance compared to find the optimal architecture. Different architectures are built using as variables the number of layers, the transfer function between the layers and the number of nodes in the hidden layers. The simplest setup as in equation 2.5 will be used as a reference with clearly very limited computational power and short training time.

More complex architectures will be built and all evaluated with 15 training rounds (training with all examples), choosing the best overall performance. The architectures are tested in three different types: Firstly only linearly connected layers (f(x)=x) with differing number of layers n of equal size, secondly only one layer that is not linear for all predefined transfer functions in Mathematica and lastly the best transfer function from the second part and the linear one with different lengths of the intermediate layers between in- and output. This process is shown in Table 4.1.

The experiments described in this table can be run with simple code over night. Should the third part show certain trends, the trend can be expanded upon and explored further. The experiments will be performed with a small subset of data, namely one month's data, such as to determine the ideal architecture with reasonable computation time.

3.2.2 Data

In this part of the research, it will be found which volume of training data is to be used for reaching the best result for any random data sample taken from throughout the year. From the 15 months of data that were collected, only 12 are usable due to recalibration. To save computational time, it will be checked if all data is, in fact, necessary to achieve a same quality network. Specifically, it stands to reason to assume that data from months within a season will be similar so that maybe two out of three months of data can be used.



(a) reference spectrum and measurement that is orders higher than is allowed.

(b) Example for the data colleced at night

Figure 3.1: These two images showcase two extreme cases of data collection, one "zero measurement" (b) as taken at night and one measurement of intense sunshine reading in at values far higher than is classically allowed (a). Both these sets of data were collected by the spectroradiometer on the Solar Field at AMOLF.

Figure 3.1 depicts two extreme values of the measurements: 3.1a shows a very sunny day and 3.1b data collected at night, both of the same day. Clearly, the night data will not exhibit the kind of pattern that will have to be predicted. Therefore, it stands to reason to eliminate this, for it will merely add redundant information to the predictor, at a volume of nearly half the total training data. This will affect the weights of the system away from the desired ideal. It is to be noted that the quality (loss) is likely to be worse (higher), as the nighttime data will be three orders of magnitude smaller than the normal data, as such providing many very small losses, so that the overall loss averages out to smaller values. In this case, a higher average loss might actually be associated with a more adequate predictor model.

The measurement taken at 2pm on July 1st 2019 is juxtaposed with the AM1.5 spectrum in Figure 3.1a, such as to show the unphysical measurement. It can be seen that the two sets of data are clearly of a different order of magnitude. Naturally, these measurements are then not to be included in the predictor model. Therefore, it has to be found, which values violate the total irradiance of the sun to improve the prediction of the smaller values. As it turns out, nearly 5% of the total data collected in summer falls into this category of unphysical measurements.

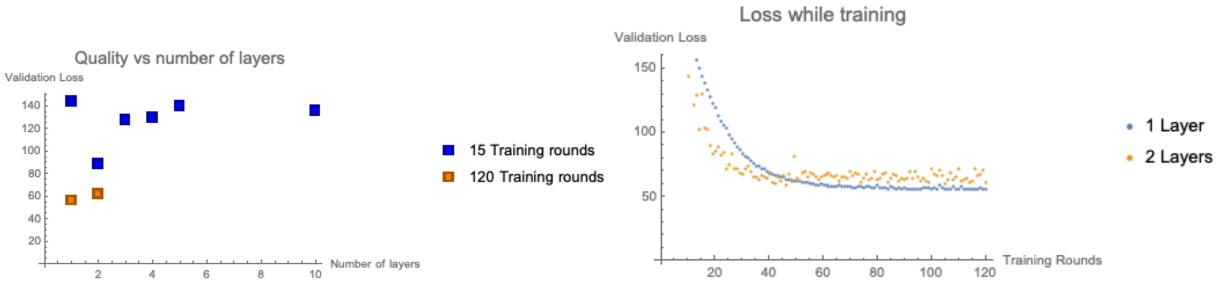
In the course of the research, the data corresponding to February, March and April 2020 have been corrupted, asking for another go at preprocessing it. Since this is very time-intensive, these months was first analysed for their relevance before reproducing them, to know if more data would at all be relevant. The analysed data now thus counts 8 months worth of data, including all summer months and some autumn, spring and winter. Physically, the most interesting data for energy production is clearly in summer, by its increased sun hours and intensity. Therefore, it is expected that the absence of spring and autumn months in the training data will not significantly impact the quality of prediction.

Chapter 4

Results and Discussion

4.1 Architecture

In the following section, the results of the tests proposed in Chapter 3.2 will be portrayed and their implications discussed, starting with the architecture of the network. Firstly, it was to be found what kind of accuracy can be associated with deeper networks, i.e. adding more hidden layers. By using a chain of linearly connected layers, the loss depicted in Figure 4.1 was found using the same number of training rounds, i.e. all networks are trained 15 times with the total volume of training data.



(a) Validation loss vs depth of network

(b) Comparing loss of one- and two-layered network during training

Figure 4.1: Test of the layer numbers on the linear layer.

In Figure 4.1a the flat network (1 layer) clearly shows the worst performance at low training volume, of same order as networks with multiple layers. Notably, the two- and three-layered system show the best performance, with the two layers only having a loss of two thirds of the first. Another valuable bit of information, however, is the computation time, which differs by nearly an order of magnitude (11s versus 95s). Therefore, another plotmarker is noted at the loss that is associated with a flat network being trained for the same amount of time. The one-layered network can outperform the deeper networks.

Next, the other available transfer functions are to be tested in their feasibility on a flat

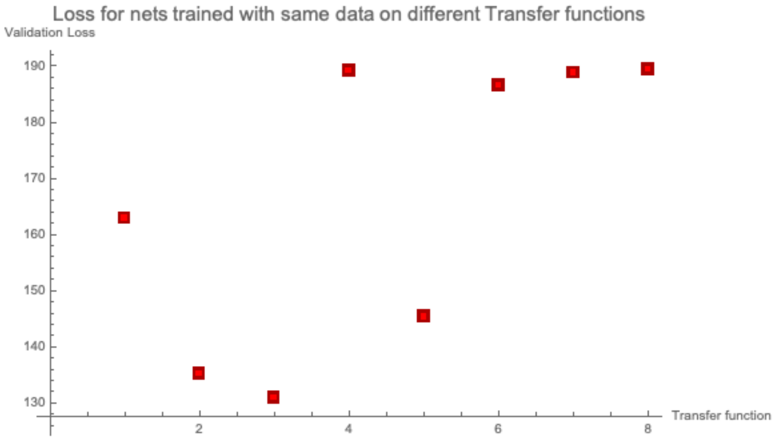


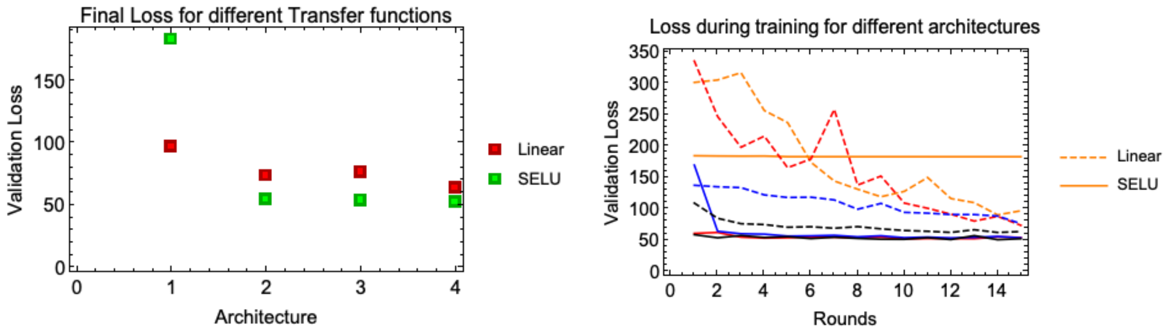
Figure 4.2: Testing for different transfer functions. The number of the transfer function equals the entry in Table 4.1

network, excluding the linear one for comparison in the next part. The results of this test are depicted in Figure 4.2, with the entries following the same chronology as the functions depicted in Figure 2.7. The transfer functions shown in Figures 2.7d, 2.7f-2.7h are of no interest to the application, mostly because all values that are given are positive and large enough that the transfer function becomes effectively $f(x) \approx x$. The exponential transfer functions (shown in Figures 2.7b,2.7c and 2.7e) exhibit best performance, with a preference for the first two. By this computation, the Scaled Exponential Unit will be used for further reference in the latter part of finding the best predictor model, as shown in Figure 2.7c and equation 2.7.

Ultimately, it is the goal to understand if a different transfer function will be able to reach a better performance than the linearly connected layers. Therefore the test with both functions in the same architectures have been performed and the results are depicted in Figure 4.3a. The logical assumption is clearly that the more complex network with the nonlinear transfer function is to be chosen, as it yields the best result. In this case, however, it is important to look at the "learning" of the ML algorithm.

Figure 4.3b shows this development of the Validation loss during training for all the same networks. The corresponding architectures are plotted in the same colour, with dashed lines indicating linearly connected layers. The networks with the added transfer function show no learning, but rather converge to a certain accuracy quickly and do not improve over time, whereas the linear networks keep learning. Because of this, the simple, linear networks are expected to keep improving with a rising volume of training data, beyond the accuracy of the SELU networks. This same development of the validation loss has been observed for all transfer functions in the previous part as well.

Furthermore, this data backs up the previous finding that a higher number of layers hampers the rate at which the network improves. Therefore, the network that will ultimately be used for the output is the simplest of them all, a single linear layer. This leaves out the problem of finding a suitable layer structure, with specific amounts of nodes. An



(a) This plot depicts the ultimate loss of training the networks with architectures according to the third part in Table 4.1

(b) In this graph, the validation loss during training is depicted for all networks in the third part of Table 4.1

Figure 4.3: Comparing the performance of linear and non-linear transfer functions for the same architecture.

added bonus of this system, reminiscent of the equation 2.5, is that a linear mapping such as this will actually contain physical information. It is feasible to expect a correlation between the weights for the different spectral intensities and the part of the spectrum they measure. Additionally, it might be expected that the weights might give rise to an understanding of certain atmospheric absorption.

4.2 Data and Training

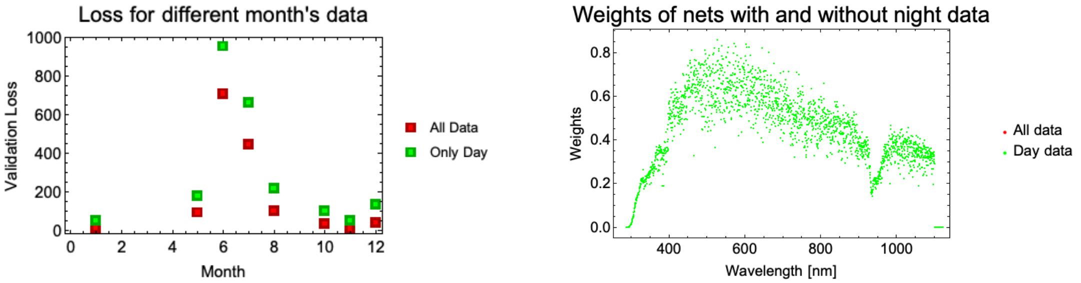
The simple one-layered network has been trained using all the different data sets (months) available. The associated Validation losses are depicted in Figure 4.4a (red). Note that the validation set is chosen as 30% of the total data, chosen at random from all months and times of day. Clearly, the data collected in summer will peak at higher values in intensity and thus make the difference between the different predictions higher resulting in a overall less meaningful predictor model. Therefore, the quality of the network will mainly be determined by the "bottleneck" that is the summer. This does not mean that the prediction is worse, this is a mere artifact of the definition of the measure of quality.

The next step is to look at two different parts of the data: the brightest and the data collected at night. The former has proved to be faulty at times for the simple reason that the intensity is so high that the measurement gives an intensity higher than physically possible for incoming solar radiation. Therefore, the use of these chosen 8 months can be justified for granting enough data, while focusing on the most important prediction of summer.

As expected, taking out the nighttime data does increase the loss as computed during training. The question now is if the prediction does actually improve. For that, the weights of the networks trained on data from July will be compared, see Figure 4.4b. The figure only depicts the weights for the red input but clearly these describe the same network, all

Phase	Transfer function	number of layers n	Length of layers m_i	Loss
1	linear	1	2048	143.8
		2	2048	87.99
		3	2048	127.4
		4	2048	129.6
		5	2048	139.7
		10	2048	135.3
2	Tanh	1	2048	162.8
	Sigmoid	1	2048	135.2
	Hard Sigmoid	1	2048	130.9
	SELU	1	2048	189.1
	ELU	1	2048	145.3
	Ramp	1	2048	186.5
	$\frac{x}{1+\lvert x\rvert}$	1	2048	188.8
	$log(e^x + 1)$	1	2048	189.4
3	linear	2	2^{10},2^{11}	96.36
	$f^*(x)$	2		182.6
	linear	4	3×2^{10},2^{11}	73.05
	$f^*(x)$	4		53.68
	linear	3	2^9,2^{10},2^{11}	75.80
	$f^*(x)$	3		53.05
	linear	4	2^8,2^9,2^{10},2^{11}	63.11
	$f^*(x)$	4		51.69

Table 4.1: Plan to find the ideal network geometry. Please refer to Figure 4.2 for the specified transfer functions. Along the ideas in this table, the most ideal neural network will be designed.



(a)

(b)

Figure 4.4: These two images show the difference in prediction with and without the data collected at night, i.e. Figure 3.1b. (a) features the final validation loss for networks trained with only data from the month as on the x axis, whereas (b) compares the weights of the networks, showing they are identical.

other weights are identical as well. For consequent measurements, the nighttime data will be kept for convenience, as they do not affect the outcome. Keeping this data will make the losses computed comparable with the ones associated to the other tests.

Next, the bright day values are to be corrected for. It is found that these measurements occur at measurements of the LAD with a red intensity higher than 6000, roughly. Since this would cut down on effectively all high-intensity data, it is decided not to completely discard of this data, also since it makes up a rough third of the total data collected in June and July. Instead, two separate systems will be trained, one for all legitimate data and one for the unreliable data, such as to improve the quality of prediction and have a model set for all data that is of value. The predictor will now be set up such that it checks the input for the red intensity. Should this exceeds the limit, the high predictor will be used, is it lower, the main network will be used.

4.3 Final Result

Loss

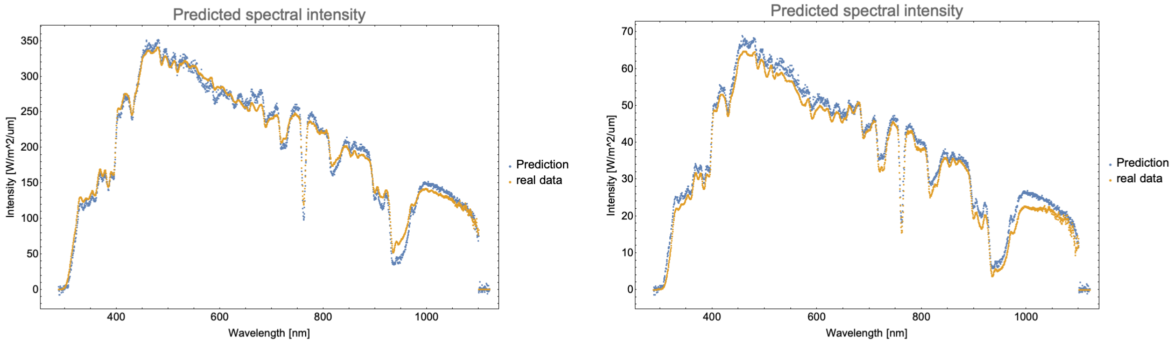
In the following paragraph, the final predictor model will be explained and discussed. Both the product and a notebook with detailed instructions how to create this model are available at the previously shown link 1. Specifically, it will be looked at how well this model predicts, what its weights contribute and what these results mean for the use of this sensor on solar fields.

The loss associated with the now-established network is

$$\text{loss} = \begin{cases} 34.4\,\mathrm{W\,m^{-2}\,\mu m^{-1}} & \text{for} \quad R < 6000 \\ 1880\,\mathrm{W\,m^{-2}\,\mu m^{-1}} & \text{for} \quad R \geq 6000 \end{cases} \tag{4.1}$$

This large discrepancy can be simply attributed to the data associated with nonphysical measurements. Although there might be many valid measurements within this training set, the difference between those and the invalid is so big that the network will not be able to accommodate for both, thus bringing with it a large loss. Nevertheless, more than 80% of the total collected data has been used, still more than 57000 data sets. Obviously, the prediction would be found if all data is correct. It is not unthinkable that the reason for this invalid data lies somewhere in the complicated data pretreatment, so it is advised to revisit and improve this step for future improvements. Additionally, it might prove useful to take less severe measures and pick out only the faulty data. By taking out all data associated with a higher intensity, there will be a large "collateral damage", cutting off data that is perfectly fine.

Beyond the loss, there are two more important ways to evaluate these predictions. Firstly, it will be checked if the typical spectra for rainy and sunny measurements are actually followed, visually. Some examples are shown in Figure 4.5, for lower intensity LAD measurements. Clearly, this network manages to closely resemble the general shape associated to these weather conditions. The second way to judge the goodness of fit is



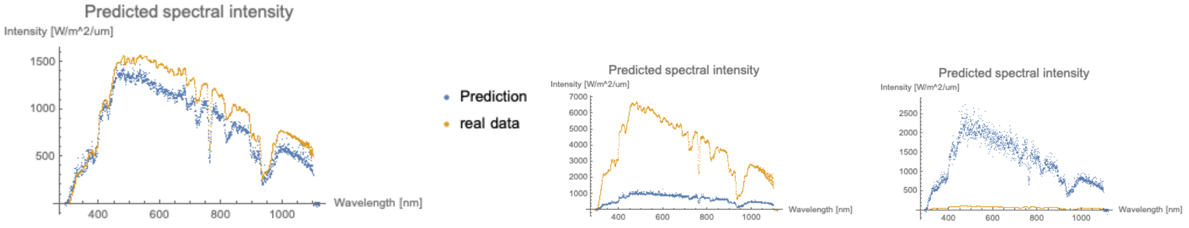
(a) Sunny spectrum

(b) Cloudy spectrum with the flat around 600nm

Figure 4.5: Predictions of two fairly typical spectra, with the actual measurements, predicted with the net trained on low-intensity data.

through the total irradiance [in $\frac{W}{m^2}$]. For all intensities between 40 and 400 $\frac{W}{m^2}$, the network will be able to predict the total irradiance with an relative error of less than 2%, averaged over all data.

Higher intensities in the red are predicted by the other network and actually show very good results, considering they physical impossibilities. For all these instances, the irradiance is predicted on average with a relative error of less than 20%. This goes to show the strong predictive abilities of the neural network. Nevertheless, the loss is very high. This can be understood simply by viewing a handful of typical predictions, shown in Figure 4.6. The error here lies in two cases that can be observed repeatedly.



(a) Prediction closely resembling the data

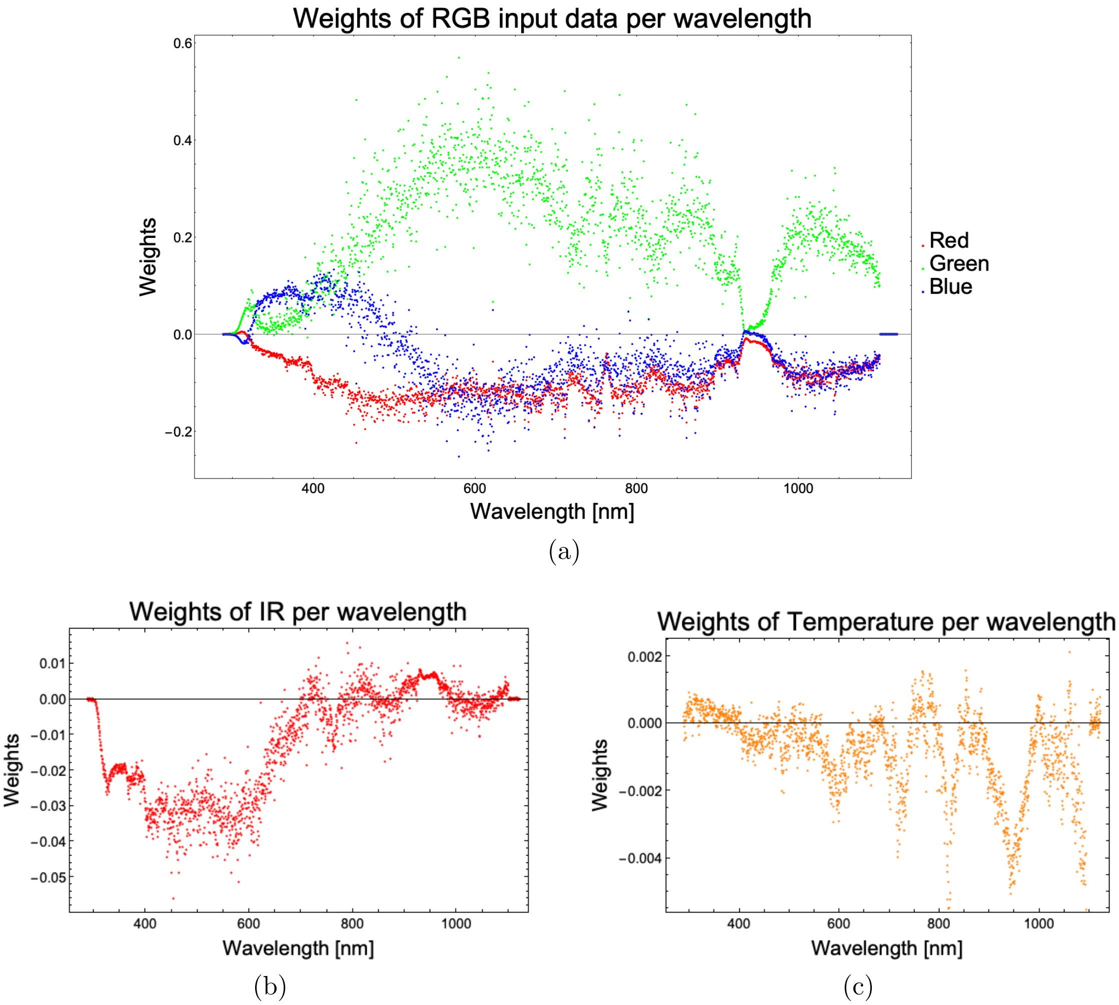
(b) High spectral data with low LAD data

(c) High LAD but small spectral data

Figure 4.6: Predictions of the net trained with high-intensity data.

Firstly, the case of Figure 4.6b, showcases a situation in which the measured spectrum is of such high intensity that something about the data collected by the spectrometer must be corrupted. Should that be the case, then the network will not be systematically trained to predict this error. In this case, the spectrum has a irradiance multiple times higher than that of the AM1.5 spectrum. The second case is an error as can be seen in Figure 4.6c, where the prediction is much higher than the measurement. In this case, the previous measurement had similar LAD intensities, with different spectral measurement. That prediction was very accurate, though. The likely cause for this is shading. Although the two sensors are close by, it is plausible that the spectroradiometer was shaded while the LAD was not, resulting in skewed measurements. Both these cases impose losses of multiple orders of magnitude higher onto the system, significantly raising the overall loss. This is to be addressed in future iterations of this training.

Network



(a)

(b)

(c)

Figure 4.7: Weights of the final system. (a) shows the weights of the intensities in the three colours red, green and blue, whereas (b) and (c) show the IR intensity and temperature

Figure 4.7a shows that the green intensity measurement dominates the green region. The small wavelength-part of the spectrum, the blue region, will be mostly determined in its intensity by the blue measurement. Unfortunately, the same can not be said for the red part of the spectrum. One more notable result of this depiction is that the blue intensity will mostly determine, how steep the angle is coming down from the peak near 500nm, since its weight changes sign and a large blue intensity measurement will thus mean a steep angle.

Naturally, the green and blue make sense to strongly influence their respective region. A simple observation does, however, show that all three effectively follow the general shape of a AM1.5-like spectrum. Although the weights might be strongest for certain parts, the overall shape is determined by all of them together. Most notably, the strong absorption peak near 950nm is present in each of the weight distributions showcasing the same shape as the total spectrum. The neural network does not seek a physical link between the input and output but because it closely resembles a linear interpolation, the weights can be seemingly random, just exhibiting a forceful fit onto the data.

In the lower images, 4.7b and 4.7c, the weights for the temperature and IR measurements are shown. Both these plots are shown on a smaller scale (y-axis) than the RGB weights in Figure 4.7a. Nevertheless, IR is always the biggest value of the measurements, so that in the first half of the plotted region, it will have a notable influence on the prediction, also affecting the aforementioned slope after the maximum. This is a noteable result as this slope is a major indicator of the weather. This observation implies that a higher IR measurement (with respect to the others) will result in a less steep graph, implying sunlight. Physically, this implies that the specific wavelengths measured by the IR sensor are absorbed less by cloudy weather than the other three, giving a correlation between the weather and measurement.

The temperature graph, Figure 4.7c, shows very small weights and by its very small values is effectively negligible in its effect on the outcome. Nevertheless, it shows strong peaks near wavelengths 700nm, 820nm, 950nm and 1100nm. Recalling the absorption spectrum of the atmosphere as in Figure 2.1, it can be seen that these peaks in the weight of the temperature measurement coincide with the water absorption. Clearly, this implies correlation of increased temperature leading to stronger absorption in these peaks. Intuitively, this also makes sense as warmer air will dissolve a higher concentration of water droplets, in turn increasing absorption by water vapour.

4.4 Future Research

For future work with this predictor model, it is advised to first improve the quality of the data and train it again. A simple code that can detect and take out outliers as described in the previous section will make for a significant improvement in predicting quality. Starting from the code as given, it should be easy to take care of the data, so that the total predictor will be able to function as the part predicting the lower part now does. Additionally, using more data is likely to improve the quality of prediction.

Beyond improving, it is also important to cross-reference the quality of prediction with other prediction techniques, both with other machine learning tools and other predictors. This was beyond the scope of the present project. It is suggested to use dimensional reduction and use machine learning algorithms such as k-nearest neighbours. A simple linear interpolation between the data might already be made for comparison as well.

With this higher accuracy, the originally defined goal to **predict a continuous spectrum for a continuum of plane orientations** can be achieved easily. The easiest way to do this would be to combine the measurements of the LAD sensors and use weights to describe the desired orientation with respect to the actually measured direction. Since this algorithm would use the model described in this thesis, it will prove useful to make a code that can automatically use collected LAD and spectroradiometer data to create the neural network, so that the predictor function can be updated and made more accurate automatically.

Should this be realised, the ultimate goal will be to create a *forecasting model*, that can predict the spectrum for each plane orientation given a date, time and weather conditions (rain, cloudy, sunny). The total forecasting model can then be stylised in a compelling way such that the user has to merely input their five required variables, so that this product can be used and sold alongside the LAD for improving the gain of arrays of solar cells.

Chapter 5

Conclusion

Over the course of this research, it has been established what form of machine learning predictor is suitable for the task at hand, namely training a network to predict a continuous spectrum for wavelengths on the interval $280nm < \lambda < 1120nm$. The inputs of the net are four spectral intensities measured for the blue, green, red and infrared. The choice of computational framework fell onto the use of an artificial neural network (ANN). Its advantages are that it is computationally favourable by splitting up the total work volume into small "mini-batches" and that their use is well-documented online.

The ideal network was found to be a network consisting of only one layer, linearly connecting the input and output. It was found through experiments that deeper networks do not improve upon training as the flat network does. Considering different transfer functions, they will converge to the best quality very quickly, making training beyond that point irrelevant. It was not found that any activation function can outperform the linear.

Over the lifetime of the Light Ambient Detector, 12 months of data have been collected. It was found that the data collected in summer will be most prominent for predicting the intensity. Therefore, eight months of data have been used for training, including all summer months. In an attempt to separate the effect of measurement errors in both sensors, most likely due to shading, high-intensity measurements were taken out. The resulting predictor for lower intensities is shown to be highly accurate, imitating the general shape of the spectrum closely and computing a total irradience with an average relative error lower than 2% for more than 30000 examples.

The high-intensity data was also used to train a network such as to indicate what must be improved. The outliers in the unphysical measurements overwhelm the chosen quantification of the difference in prediction and measurement, making its error estimate superfluous. It will be a task for future research on this topic to find a way to take out the outliers that clearly describe physically impossible measurements of high spectal irradiance.

Furthermore, follow-up research is to be done to describe the diffuseness with a continuous spectrum. With improved data and the new algorithm to interpolate over all directions of incoming light, a product can be generated to accompany the device LAD into commercial application for improving the yield of solar parks and especially the orientation of bifacial solar cells, that can also absorb light incoming from the back of the cell.

Bibliography

[1] Andrea L Pollastri. A novel multi-directional light detector for modelling the cost-efficiency benefits of bi-facial solar panels., 2019.

[2] Merlijn Kersten. Outdoor solar cell performance, 2018.

[3] Fred Ortenberg. Ozone: Space vision. *ASRI, Technion, Haifa, Israel*, pages 16–23, 2002.

[4] Robin M Pope and Edward S Fry. Absorption spectrum (380–700 nm) of pure water. ii. integrating cavity measurements. *Applied optics*, 36(33):8710–8723, 1997.

[5] S Jacquemoud and SL Ustin. Application of radiative transfer models to moisture content estimation and burned land mapping. In *4th International Workshop on Remote Sensing and GIS Applications to Forest Fire Management*, volume 312, 2003.

[6] Frank J Blatt. *Modern physics.* McGraw, 1992.

[7] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[8] Claudia Perlich, Brian Dalessandro, Troy Raeder, Ori Stitelman, and Foster Provost. Machine learning for targeted display advertising: Transfer learning in action. *Machine learning*, 95(1):103–127, 2014.

[9] Charu C Aggarwal and Chandan K Reddy. Data clustering. *Algorithms and applications. Chapman&Hall/CRC Data mining and Knowledge Discovery series, Londra*, 2014.

[10] Michael A Nielsen. *Neural networks and deep learning*, volume 2018. Determination press San Francisco, CA, 2015.

[11] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[12] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[13] Wolfram Research, Inc. Mathematica, Version 12.0. Champaign, IL, 2019.

[14] Abdul Rahim Pazikadin, Damhuji Rifai, Kharudin Ali, Muhammad Zeesan Malik, Ahmed N Abdalla, and Moneer A Faraj. Solar irradiance measurement instrumentation and power solar generation forecasting based on artificial neural networks (ann): A review of five years research trend. *Science of The Total Environment*, 715:136848, 2020.

[15] Zhe Wang, Fei Wang, and Shi Su. Solar irradiance short-term prediction model based on bp neural network. *Energy Procedia*, 12:488–494, 2011.

[16] S Amirkhani, Sh Nasirivatan, AB Kasaeian, and A Hajinezhad. Ann and anfis models to predict the performance of solar chimney power plants. *Renewable Energy*, 83:597–607, 2015.