Investigating the Evolution of the COVID-19 Pandemic in Germany Using Physics-Informed Neural Networks

Bachelor Thesis in Computer Science

submitted by
Phillip Rothenbeck
born February 22, 2002 in Eckernförde

written at
Computer Vision Group
Department of Mathematics and Computer Science
Friedrich-Schiller-Universität Jena

Supervisor:	Prof. Dr.-Ing. Joachim Denzler
Advisor:	Niklas Penzel, Sai Karthikeya Vemuri
Started:	May 1, 2024
Finished:	September 14, 2024

Eigenständigkeitserklärung

1. Hiermit versichere ich, dass ich die vorliegende Arbeit - bei einer Gruppenarbeit die von mir zu verantwortenden und entsprechend gekennzeichneten Teile - selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich trage die Verantwortung für die Qualität des Textes sowie die Auswahl aller Inhalte und habe sichergestellt, dass Informationen und Argumente mit geeigneten wissenschaftlichen Quellen belegt bzw. gestützt werden. Die aus fremden oder auch eigenen, älteren Quellen wörtlich oder sinngemäß übernommenen Textstellen, Gedankengänge, Konzepte, Grafiken etc. in meinen Ausführungen habe ich als solche eindeutig gekennzeichnet und mit vollständigen Verweisen auf die jeweilige Quelle versehen. Alle weiteren Inhalte dieser Arbeit ohne entsprechende Verweise stammen im urheberrechtlichen Sinn von mir.

2. Ich weiß, dass meine Eigenständigkeitserklärung sich auch auf nicht zitierfähige, generierende KI-Anwendungen (nachfolgend „generierende KI“) bezieht. Mir ist bewusst, dass die Verwendung von generierender KI unzulässig ist, sofern nicht deren Nutzung von der prüfenden Person ausdrücklich freigegeben wurde (Freigabeerklärung). Sofern eine Zulassung als Hilfsmittel erfolgt ist, versichere ich, dass ich mich generierender KI lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss deutlich überwiegt. Ich verantworte die Übernahme der von mir verwendeten maschinell generierten Passagen in meiner Arbeit vollumfänglich selbst. Für den Fall der Freigabe der Verwendung von generierender KI für die Erstellung der vorliegenden Arbeit wird eine Verwendung in einem gesonderten Anhang meiner Arbeit kenntlich gemacht. Dieser Anhang enthält eine Angabe oder eine detaillierte Dokumentation über die Verwendung generierender KI gemäß den Vorgaben in der Freigabeerklärung der prüfenden Person. Die Details zum Gebrauch generierender KI bei der Erstellung der vorliegenden Arbeit inklusive Art, Ziel und Umfang der Verwendung sowie die Art der Nachweispflicht habe ich der Freigabeerklärung der prüfenden Person entnommen.

3. Ich versichere des Weiteren, dass die vorliegende Arbeit bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt wurde oder in deutscher oder einer anderen Sprache als Veröffentlichung erschienen ist.

4. Mir ist bekannt, dass ein Verstoß gegen die vorbenannten Punkte prüfungsrechtliche Konsequenzen haben und insbesondere dazu führen kann, dass meine Prüfungsleistung als Täuschung und damit als mit „nicht bestanden“ bewertet werden kann. Bei mehrfachem oder schwerwiegendem Täuschungsversuch kann ich befristet oder sogar dauerhaft von der Erbringung weiterer Prüfungsleistungen in meinem Studiengang ausgeschlossen werden.

5. Die Richtlinien des Lehrstuhls für Examensarbeiten habe ich gelesen und anerkannt. Seitens des Verfassers/der Verfasserin bestehen keine Einwände, die vorliegende Examensarbeit für die öffentliche Benutzung zur Verfügung zu stellen.

Jena, den 14. September 2024 Phillip Rothenbeck

Überblick

German version of the abstract.
Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Abstract

English version of the abstract.
Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Contents

Chapter 1

Introduction 5

1.1 Related work 2

Chapter 2

Theoretical Background 12

This chapter introduces the theoretical foundations for the work presented in this thesis. In Section 2.1 and Section 2.2, we describe differential equations and the underlying theory. In these Sections both the explanations and the approach are based on a book on analysis by Rudin [Rud07] and a book about ordinary differential equations by Tenenbaum and Pollard [TP85]. Subsequently, we employ this knowledge to examine various pandemic models in Section 2.3. Finally, we address the topic of neural networks with a focus on the multilayer perceptron in Section 2.4 and physics informed neural networks in Section 2.5.

2.1 Mathematical Modelling using Functions 1

To model a physical problem mathematically, it is necessary to define a set of fundamental numbers or quantities upon which the subsequent calculations will be based. These sets may represent, for instance, a specific time interval or a distance. The term *domain* describes these fundamental sets of numbers or quantities [Rud07]. A *variable* is a changing entity living in a certain domain. In this thesis, we will focus on domains of real numbers in $\mathbb{R}$.

meeting question 1

The mapping between variables enables the modeling of a physical process and may depict semantics. We use functions in order to facilitate this mapping. Let $A, B \subset \mathbb{R}$ be to subsets of the real numbers, then we define a function as the mapping

$$f : A \to B. \tag{2.1}$$

In other words, the function f maps elements $x \in A$ to values $f(x) \in B$. A is the *domain* of f, while B is the *codomain* of f. Functions are capable of representing the state of a system as a value based on an input value from their domain. One

illustrative example is a function that maps a time step to the distance covered since a starting point. In this case, time serves as the domain, while the distance is the codomain.

2.2 Mathematical Modelling using Differential Equations 1

meeting question 2

Often, the behavior of a variable or a quantity across a domain is more interesting than its current state. Functions are able to give us the latter, but only passively give information about the change of a system. The objective is to determine an effective method for calculating the change of a function across its domain. Let f be a function and $[a, b] \subset \mathbb{R}$ an interval of real numbers. The expression

$$m = \frac{f(b) - f(a)}{a - b} \tag{2.2}$$

look up in Rudin - cite (wordly)

gives the average rate of change. While the average rate of change is useful in many cases, the momentary rate of change is more accurate. To calculate this, we need to narrow down, the interval to an infinitesimal. For each $x \in [a, b]$ we calculate

$$\frac{df}{dx} = \lim_{t \to x} \frac{f(t) - f(x)}{t - x}, \tag{2.3}$$

if it exists. As the Tenenbaum and Pollard [TP85] define, df/dx is the *derivative*, which is "the rate of change of a variable with respect to another". The relation between a variable and its derivative is modeled in a *differential equation*. The derivative of df/dx yields d^2f/dx^2, which is the function that calculates the rate of change of the rate of change and is called the *second order derivative*. Iterating this n times results in d^nf/dx^n, the derivative of the n'th order. A method for obtaining a differential equation is to derive it from the semantics of a problem. For example, in physics a differential equation can be derived from the law of the conservation of energy [Dem21]. Differential equations find application in several areas such as engineering *e.g.*, the Chua's circuit [Mat84], physics with, *e.g.*, the Schrödinger equation [Sch26], economics, *e.g.*, Black-Scholes equation [Oks00], epidemiology, and beyond.

is this good?

In the context of functions, it is possible to have multiple domains, meaning that function has more than one parameter. To illustrate, consider a function operating in two-dimensional space, wherein each parameter represents one axis. Another ex-

ample would be a function, that maps its inputs of a location variable and a time variable on a height. The term *partial differential equations* (*PDE*'s) describes differential equations of such functions, which contain partial derivatives with respect to each individual domain. In contrast, *ordinary differential equations* (*ODE*'s) are the single derivatives for a function having only one domain [TP85]. In this thesis, we restrict ourselves to ODE's.

A *system of differential equations* is the name for a set of differential equations. The derivatives in a system of differential equations each have their own codomain, which is part of the problem, while they all share the same domain.

Tenenbaum and Pollard [TP85] provide many examples for ODE's, including the *Motion of a Particle Along a Straight Line*. Further, Newton's second law states that "the rate of change of the momentum of a body ($momentum = mass \cdot velocity$) is proportional to the resultant external force F acted upon it" [TP85]. Let m be the mass of the body in kilograms, v its velocity in meters per second and t the time in seconds. Then, Newton's second law translates mathematically to

$$F = m\frac{dv}{dt}. \tag{2.4}$$

It is evident that the acceleration, $a = \frac{dv}{dt}$, as the rate of change of the velocity is part of the equation. Additionally, the velocity of a body is the derivative of the distance traveled by that body. Based on these findings, we can rewrite the Equation (2.4) to

$$F = ma = m\frac{d^2s}{dt^2}. \tag{2.5}$$

To conclude, note that this explanation of differential equations focuses on the aspects deemed crucial for this thesis and is not intended to be a complete explanation of the subject. To gain a better understanding of it, we recommend the books mentioned above [Rud07, TP85]. In the following section we describe the application of these principles in epidemiological models.

2.3 Epidemiological Models 4

Better?

Pandemics, like *COVID-19*, which have resulted in a significant number of fatalities. Hence, the question arises: How should we analyze a pandemic effectively? It is essential to study whether the employed countermeasures are efficacious in combating the pandemic. Given the unfavorable public response to measures such as lockdowns, it is imperative to investigate that their efficacy remains commensurate with the costs incurred to those affected. In the event that alternative and novel technologies were in use, such as the mRNA vaccines in the context of COVID-19, it is needful to test the effect and find the optimal variant. In order to shed light on the aforementioned events, we need a method to quantify the pandemic along with its course of progression.

The real world is a highly complex system, which presents a significant challenge attempting to describe it fully in a mathematical model. Therefore, the model must reduce the complexity while retaining the essential information. Furthermore, it must address the issue of limited data availability. For instance, during COVID-19 institutions such as the Robert Koch Institute (RKI)[1] were only able to collect data on infections and mortality cases. Consequently, we require a model that employs an abstraction of the real world to illustrate the events and relations that are pivotal to understanding the problem.

2.3.1 SIR Model 3

In 1927, Kermack and McKendrick [KM27] introduced the *SIR Model*, which subsequently became one of the most influential epidemiological models. This model enables the modeling of infections during epidemiological events such as pandemics. The book *Mathematical Models in Biology* [EK05] reiterates the model and serves as the foundation for the following explanation of SIR models.

The SIR model is capable of illustrating diseases, which are transferred through contact or proximity of an individual carrying the illness and a healthy individual. This is possible due to the distinction between infected individuals who are carriers of the disease and the part of the population, which is susceptible to infection. In the model, the mentioned groups are capable to change, *e.g.*, healthy individuals becoming infected. The model assumes the size N of the population remains con-

[1] `https://www.rki.de/EN/Home/homepage_node.html`

stant throughout the duration of the pandemic. The population N comprises three distinct compartments: the *susceptible* group S, the *infectious* group I and the *removed* group R (hence SIR model). Let $\mathcal{T} = [t_0, t_f] \subseteq \mathbb{R}_{\geq 0}$ be the time span of the pandemic, then,

$$S : \mathcal{T} \to \mathbb{N}, \quad I : \mathcal{T} \to \mathbb{N}, \quad R : \mathcal{T} \to \mathbb{N}, \tag{2.6}$$

give the values of S, I and R at a certain point of time $t \in \mathcal{T}$. For S, I, R and N applies:

$$N = S + I + R. \tag{2.7}$$

The model makes another assumption by stating that recovered people are immune to the illness and infectious individuals can not infect them. The individuals in the R group are either recovered or deceased, and thus unable to transmit or carry the disease. As visualized in the Figure 2.1 the individuals may transition between

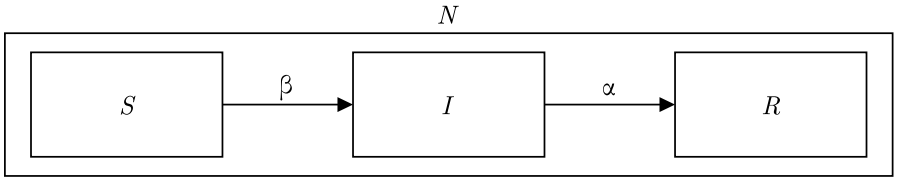


Figure 2.1: A visualization of the SIR model, illustrating N being split in the three groups S, I and R.

groups based on transition rates. The transmission rate β is responsible for individuals becoming infected, while the rate of removal or recovery rate α (also referred to as δ or ν, *e.g.*, [EK05, MPF23]) moves individuals from I to R.

We can describe this problem mathematically using a system of differential equations (see Section 2.2). Thus, Kermack and McKendrick [KM27] propose the following set of differential equations:

$$\begin{aligned} \frac{dS}{dt} &= -\beta SI, \\ \frac{dI}{dt} &= \beta SI - \alpha I, \\ \frac{dR}{dt} &= \alpha I. \end{aligned} \tag{2.8}$$

This set of differential equations, is based on the following assumption: “The rate of transmission of a microparasitic disease is proportional to the rate of encounter

of susceptible and infective individuals modelled by the product (βSI)", according to Edelstein-Keshet [EK05]. The system shows the change in size of the groups per time unit due to infections, recoveries, and deaths.

The term βSI describes the rate of encounters of susceptible and infected individuals. This term is dependent on the size of S and I, thus Anderson and May [And91] propose a modified model:

$$\begin{aligned} \frac{dS}{dt} &= -\beta\frac{SI}{N}, \\ \frac{dI}{dt} &= \beta\frac{SI}{N} - \alpha I, \\ \frac{dR}{dt} &= \alpha I. \end{aligned} \tag{2.9}$$

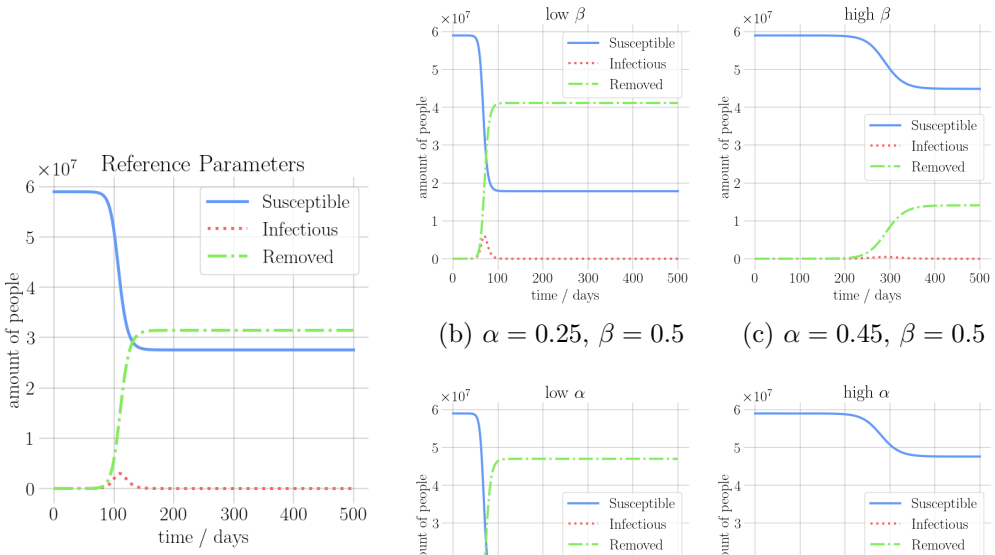
In Equation (2.9) βSI gets normalized by N, which is more correct in a real world aspect [And91].

The initial phase of a pandemic is characterized by the infection of a small number of individuals, while the majority of the population remains susceptible. The infectious group has not yet infected any individuals thus neither recovery nor mortality is possible. Let $I_0 \in \mathbb{N}$ be the number of infected individuals at the beginning of the disease. Then,

$$\begin{aligned} S(0) &= N - I_0, \\ I(0) &= I_0, \\ R(0) &= 0, \end{aligned} \tag{2.10}$$

describes the initial configuration of a system in which a disease has just emerged.

In the SIR model the temporal occurrence and the height of the peak (or peaks) of the infectious group are of paramount importance for understanding the dynamics of a pandemic. A low peak occurring at a late point in time indicates that the disease is unable to keep pace with the rate of recovery, resulting in its demise before it can exert a significant influence on the population. In contrast, an early and high peak means that the disease is rapidly transmitted through the population, with a significant proportion of individuals having been infected. Figure 2.1 illustrates this effect by varying the values of β or α while simulating a pandemic using a model



(a) $\alpha = 0.35$, $\beta = 0.5$

(b) $\alpha = 0.25$, $\beta = 0.5$

(c) $\alpha = 0.45$, $\beta = 0.5$

(d) $\alpha = 0.35$, $\beta = 0.4$

(e) $\alpha = 0.35$, $\beta = 0.6$

Figure 2.2: Synthetic data, using Equation (2.9) and $N = 7.9 \cdot 10^6$, $I_0 = 10$ with different sets of parameters. We visualize the case with the reference parameters in (a). In (b) and (c) we keep α constant, while varying the value of β. In contrast, (d) and (e) have varying values of α.

such as Equation (2.9). It is evident that both the transmission rate β and the recovery rate α influence the height and time of the peak of I. When the number of infections exceeds the number of recoveries, the peak of I will occur early and will be high. On the other hand, if recoveries occur at a faster rate than new infections the peak will occur later and will be low. Thus, it is crucial to know both β and α, as these parameters characterize how the pandemic evolves.

The SIR model makes a number of assumptions that are intended to reduce the model's overall complexity while simultaneously increasing its divergence from actual reality. One such assumption is that the size of the population, N, remains constant, as the daily change is negligible to the total population. This depiction is not an accurate representation of the actual relations observed in the real world, as the size of a population is subject to a number of factors that can contribute to change. The population is increased by the occurrence of births and decreased by the occurrence

other assumptions in a bad light?

of deaths. Other examples are the impossibility for individuals to be susceptible again, after having recovered, or the possibility for the transition rates to change due to new variants or the implementation of new countermeasures. We address this latter option in the next Section 2.3.2.

2.3.2 Reduced SIR Model and the Reproduction Number 1

The Section 2.3.1 presents the classical SIR model. This model contains two scalar parameters β and α, which describe the course of a pandemic over its duration. This is beneficial when examining the overall pandemic; however, in the real world, disease behavior is dynamic, and the values of the parameters β and α change throughout the course of the disease. The reason for this is due to events such as the implementation of countermeasures that reduce the contact between the infectious and susceptible individuals, the emergence of a new variant of the disease that increases its infectivity or deadliness, or the administration of a vaccination that provides previously susceptible individuals with immunity without ever being infected. To address this Millevoi *et al.* [MPF23] introduce a model that simultaneously reduces the size of the system of differential equations and solves the problem of time scaling at hand.

sai correction -¿ is this point not already included?

are there older sources

First, they alter the definition of β and α to be dependent on the time interval $\mathcal{T} = [t_0, t_f] \subseteq \mathbb{R}_{\geq 0}$,

$$\beta : \mathcal{T} \to \mathbb{R}_{\geq 0}, \quad \alpha : \mathcal{T} \to \mathbb{R}_{\geq 0}. \tag{2.11}$$

Another crucial element is $D(t) = \frac{1}{\alpha(t)}$, which represents the initial time span an infected individual requires to recuperate. Subsequently, at the initial time point t_0, the *reproduction number*,

$$\mathcal{R}_0 = \beta(t_0)D(t_0) = \frac{\beta(t_0)}{\alpha(t_0)}, \tag{2.12}$$

represents the number of susceptible individuals, that one infectious individual infects at the onset of the pandemic. In light of the effects of β and α (see Section 2.3.1), $\mathcal{R}_0 > 1$ indicates that the pandemic is emerging. In this scenario α is relatively low due to the limited number of infections resulting from $I(t_0) << S(t_0)$. Further, $\mathcal{R}_0 < 1$ leads to the disease spreading rapidly across the population, with an increase in I occurring at a high rate. Nevertheless, $\mathcal{R}_0$ does not cover the entire time span. For this reason, Millevoi *et al.* [MPF23] introduce $\mathcal{R}_t$ which has the

same interpretation as $\mathcal{R}_0$, with the exception that $\mathcal{R}_t$ is dependent on time. The time-dependent reproduction number is defined as,

$$\mathcal{R}_t = \frac{\beta(t)}{\alpha(t)} \cdot \frac{S(t)}{N}, \tag{2.13}$$

on the time interval $\mathcal{T}$. This definition includes the transition rates for information about the spread of the disease and information of the decrease of the ratio of susceptible individuals in the population. In contrast to β and α, $\mathcal{R}_t$ is not a parameter but a state variable in the model and enabling the following reduction of the SIR model.

Sai comment - earlier?

Equation (2.7) allows for the calculation of the value of the group R using S and I, with the term $R(t) = N - S(t) - I(t)$. Thus,

$$\begin{aligned} \frac{dS}{dt} &= \alpha(\mathcal{R}_t - 1)I(t), \\ \frac{dI}{dt} &= -\alpha\mathcal{R}_t I(t), \end{aligned} \tag{2.14}$$

is the reduction of Equation (2.8) on the time interval $\mathcal{T}$ using this characteristic and the reproduction number $\mathcal{R}_t$ (see Equation (2.13)). Another issue that Millevoi *et al.* [MPF23] seek to address is the extensive range of values that the SIR groups can assume. Accordingly, they initially scale the time interval $\mathcal{T}$ using its borders to calculate the scaled time $t_s = \frac{t-t_0}{t_f-t_0} \in [0, 1]$. Subsequently, they calculate the scaled groups,

$$S_s(t_s) = \frac{S(t)}{C}, \quad I_s(t_s) = \frac{I(t)}{C}, \quad R_s(t_s) = \frac{R(t)}{C}, \tag{2.15}$$

using a large constant scaling factor $C \in \mathbb{N}$. Applying this to the variable I, results in,

$$\frac{dI_s}{dt_s} = \alpha(t_f - t_0)(\mathcal{R}_t - 1)I_s(t_s), \tag{2.16}$$

which is a further reduced version of Equation (2.8). This less complex differential equation results in a less complex solution, as it entails the elimination of a parameter (β) and the two state variables (S and R). The reduced SIR model, is more precise in applications with a worse data situation, due to its fewer input variables.

2.4 Multilayer Perceptron 2

In Section 2.2, we demonstrate the significance of differential equations in systems, illustrating how they can be utilized to elucidate the impact of a specific parameter on the system's behavior. In Section 2.3, we show specific applications of differential equations in an epidemiological context. The final objective is to solve these equations by finding a function that fits. Fitting measured data points to approximate such a function, is one of the multiple methods to achieve this goal. The *Multilayer Perceptron* (MLP) [RHW86] is a data-driven function approximator. In the following section, we provide a brief overview of the structure and training of these *neural networks*. For reference, we use the book *Deep Learning* by Goodfellow *et al.* [GBC16] as a foundation for our explanations.

The objective is to develop an approximation method for any function f^*, which could be a mathematical function or a mapping of an input vector to the desired output. Let $\boldsymbol{x}$ be the input vector and $\boldsymbol{y}$ the label, class, or result. Then, $\boldsymbol{y} = f^*(\boldsymbol{x})$, is the function to approximate. In the year 1958, Rosenblatt [Ros58] proposed the perceptron modeling the concept of a neuron in a neuroscientific sense. The perceptron takes in the input vector $\boldsymbol{x}$ performs an operation and produces a scalar result. This model optimizes its parameters θ to be able to calculate $\boldsymbol{y} = f(\boldsymbol{x}; \theta)$ as accurately as possible. As Minsky and Papert [MP72] demonstrate, the perceptron is only capable of approximating a specific class of functions. Consequently, there is a necessity for an expansion of the perceptron.

As Goodfellow *et al.* [GBC16] proceed, the solution to this issue is to decompose f into a chain structure of the form,

$$f(\boldsymbol{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\boldsymbol{x}))). \tag{2.17}$$

This nested version of a perceptron is a multilayer perceptron. Each sub-function, designated as $f^{(i)}$, is represented in the structure of an MLP as a *layer*, which contains a linear mapping and a nonlinear mapping in form of an *activation function*. A multitude of *Units* (also *neurons*) compose each layer. A neuron performs the same vector-to-scalar calculation as the perceptron does. Subsequently, a nonlinear activation function transforms the scalar output into the activation of the unit. The layers are staggered in the neural network, with each layer being connected to its neighbors, as illustrated in Figure 2.3. The input vector $\boldsymbol{x}$ is provided to each unit of

the first layer $f^{(1)}$, which then gives the results to the units of the second layer $f^{(2)}$, and so forth. The final layer is the *output layer*. The intervening layers, situated between the first and the output layers are the *hidden layers*. The term *forward propagation* describes the process of information flowing through the network from the input layer to the output layer, resulting in a scalar loss. The alternating structure of linear and nonlinear calculation enables MLP's to approximate any function. As Hornik *et al.* [HSW89] proves, MLP's are universal approximators.

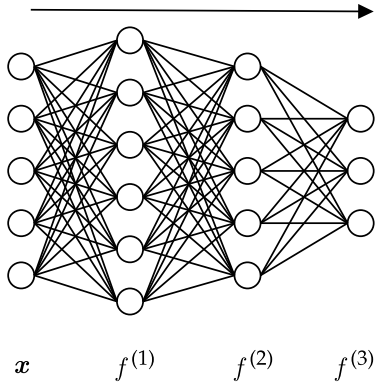


Figure 2.3: A illustration of an MLP with two hidden layers. Each neuron of a layer is connected to every neuron of the neighboring layers. The arrow indicates the direction of the forward propagation.

The term *training* describes the process of optimizing the parameters θ. In order to undertake training, it is necessary to have a set of *training data*, which is a set of pairs (also called training points) of the input data $\boldsymbol{x}$ and its corresponding true solution $\boldsymbol{y}$ of the function f^*. For the training process we must define a *loss function* $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$, using the model prediction $\hat{\boldsymbol{y}}$ and the true value $\boldsymbol{y}$, which will act as a metric for evaluating the extent to which the model deviates from the correct answer. One common loss function is the *mean square error* (MSE) loss function. Let N be the number of points in the set of training data. Then,

$$\mathcal{L}_{MSE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{N} \sum_{i=1}^{N} ||\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)}||^2, \tag{2.18}$$

calculates the squared difference between each model prediction and true value of a training and takes the mean across the whole training data.

Ultimately, the objective is to utilize this information to optimize the parameters, in order to minimize the loss. One of the most fundamental optimization strategy is *gradient descent*. In this process, the derivatives are employed to identify the location of local or global minima within a function, which lie where the gradient is zero. Given that a positive gradient signifies ascent and a negative gradient indicates descent, we must move the variable by a *learning rate* (step size) in the opposite direction to that of the gradient. The calculation of the derivatives in respect to the parameters is a complex task, since our functions is a composition of many functions (one for each layer). We can address this issue taking advantage of Equation (2.17) and employing the chain rule of calculus. Let $\hat{\boldsymbol{y}} = f(\boldsymbol{x}; \theta)$ be the model prediction with the decomposed version $f(\boldsymbol{x}; \theta) = f^{(3)}(w; \theta_3)$ with $w = f^{(2)}(z; \theta_2)$ and $z = f^{(1)}(\boldsymbol{x}; \theta_1)$. $\boldsymbol{x}$ is the input vector and $\theta_3, \theta_2, \theta_1 \subset \theta$. Then,

$$\nabla_{\theta_3} \mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{d\mathcal{L}}{d\hat{\boldsymbol{y}}} \frac{d\hat{\boldsymbol{y}}}{df^{(3)}} \nabla_{\theta_3} f^{(3)}, \tag{2.19}$$

is the gradient of $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ in respect of the parameters θ_3. To obtain $\nabla_{\theta_2} \mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$, we have to derive $\nabla_{\theta_3} \mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ in respect to θ_2. The name of this method in the context of neural networks is *back propagation* [RHW86], as it propagates the error backwards through the neural network.

In practical applications, an optimizer often accomplishes the optimization task by executing back propagation in the background. Furthermore, modifying the learning rate during training can be advantageous. For instance, making larger steps at the beginning and minor adjustments at the end. Therefore, schedulers are implementations algorithms that employ diverse learning rate alteration strategies.

leave whole paragraph out? - Niklas

For a more in-depth discussion of practical considerations and additional details like regularization, we direct the reader to the book *Deep Learning* by Goodfellow *et al.* [GBC16]. The next section will demonstrate the application of neural networks in approximating solutions to differential systems.

2.5 Physics Informed Neural Networks 4

In Section 2.4, we describe the structure and training of MLP's, which are wildely recognized tools for approximating any kind of function. In this section, we apply this capability to create a solver for ODE's and PDE's as Legaris *et al.* [LLF97] describe in their paper. In this approach, the model learns to approximate a function using provided data points while leveraging the available knowledge about the problem in the form of a system of differential equations. The *physics-informed neural network* (PINN) learns the system of differential equations during training, as it optimizes its output to align with the equations.

In contrast to standard MLP's, PINNs are not only data-driven. The loss term of a PINN comprises two components. The first term incorporates the equations of the aforementioned prior knowledge to pertinent the problem. As Raissi *et al.* [RPK17] propose, the residual of each differential equation in the system must be minimized in order for the model to optimize its output in accordance with the theory. We obtain the residual r_i, with $i \in \{1, ..., N_d\}$, by rearranging the differential equation and calculating the difference between the left-hand side and the right-hand side of the equation. N_d is the number of differential equations in a system. As Raissi *et al.* [RPK17] propose the *physics loss* of a PINN,

$$\mathcal{L}_{physics}(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \frac{1}{N_d} \sum_{i=1}^{N_d} ||r_i(\boldsymbol{x}, \hat{\boldsymbol{y}})||^2, \tag{2.20}$$

takes the input data and the model prediction to calculate the mean square error of the residuals. The second term, the *observation loss* $\mathcal{L}_{obs}(\hat{\boldsymbol{y}}, \boldsymbol{y})$, employs the mean square error of the distances between the predicted and the true values for each training point. Additionally, the observation loss may incorporate extra terms of inital and boundary conditions. Let N_t denote the number of training points. Then,

$$\mathcal{L}_{PINN}(\boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N_d} \sum_{i=1}^{N_d} ||r_i(\boldsymbol{x}, \hat{\boldsymbol{y}})||^2 + \frac{1}{N_t} \sum_{i=1}^{N_t} ||\hat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)}||^2, \tag{2.21}$$

represents the comprehensive loss function of a physics-informed neural network.

Given the nature of residuals, calculating the loss term of $\mathcal{L}_{physics}(\boldsymbol{x}, \hat{\boldsymbol{y}})$ requires the calculation of the derivative of the output with respect to the input of the neural

network. As we outline in Section 2.4, during the process of back-propagation we calculate the gradients of the loss term in respect to a layer-specific set of parameters denoted by θ_l, where l represents the index of the respective layer. By employing the chain rule of calculus, the algorithm progresses from the output layer through each hidden layer, ultimately reaching the first layer in order to compute the respective gradients. The term,

$$\nabla_{\boldsymbol{x}} \hat{\boldsymbol{y}} = \frac{d\hat{\boldsymbol{y}}}{df^{(2)}} \frac{df^{(2)}}{df^{(1)}} \nabla_{\boldsymbol{x}} f^{(1)}, \tag{2.22}$$

illustrates that, in contrast to the procedure described in eq. (2.19), this procedure the *automatic differenciation* goes one step further and calculates the gradient of the output with respect to the input $\boldsymbol{x}$. In order to calculate the second derivative $\frac{d\hat{\boldsymbol{y}}}{d\boldsymbol{x}} = \nabla_{\boldsymbol{x}}(\nabla_{\boldsymbol{x}} \hat{\boldsymbol{y}})$, this procedure must be repeated.

Above we present a method for approximating functions through the use of systems of differential equations. As previously stated, we want to find a solver for systems of differential equations. In problems, where we must solve an ODE or PDE, we have to find a set of parameters, that satisfies the system for any input $\boldsymbol{x}$. In terms of the context of PINN's this is the inverse problem, where we have a set of training data from measurements, for example, is available along with the respective differential equations but information about the parameters of the equations is lacking. To address this challenge, we set these parameters as distinct learnable parameters within the neural network. This enables the network to utilize a specific value, that actively influences the physics loss $\mathcal{L}_{physics}(\boldsymbol{x}, \hat{\boldsymbol{y}})$. During the training phase the optimizer aims to minimize the physics loss, which should ultimately yield an approximation of the true value.

One illustrative example of a potential application for PINN's is the *damped harmonic oscillator* [Dem21]. In this problem, we displace a body, which is attached to a spring, from its resting position. The body is subject to three forces: firstly, the inertia exerted by the displacement u, which points in the direction the displacement u; secondly the restoring force of the spring, which attempts to return the body to its original position and thirdly, the friction force, which points in the opposite direction of the movement. In accordance with Newton's second law and the combined influence of these forces, the body exhibits oscillatory motion around its position of rest. The system is influenced by m the mass of the body, μ the coefficient of friction

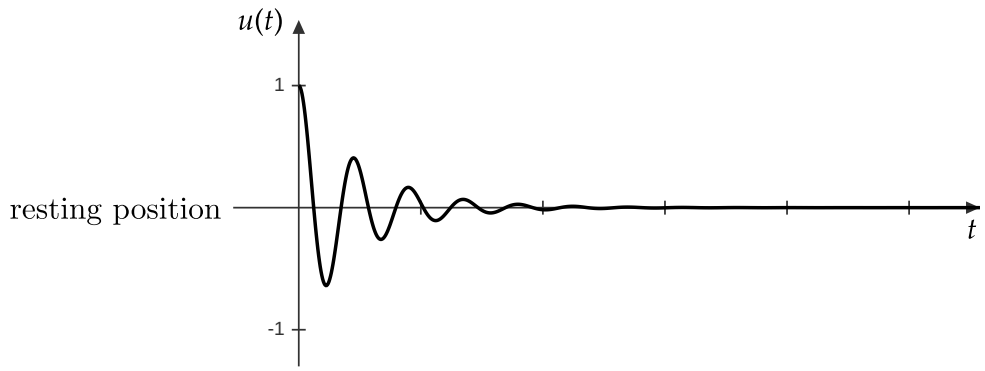


Figure 2.4: Illustration of of the movement of an oscillating body in the underdamped case. With $m = 1kg$, $\mu = 4\frac{Ns}{m}$ and $k = 200\frac{N}{m}$.

and k the spring constant, indicating the stiffness of the spring. The residual of the differential equation,

$$m\frac{d^2u}{dx^2} + \mu\frac{du}{dx} + ku = 0, \tag{2.23}$$

shows relation of these parameters in reference to the problem at hand. As Tenenbaum and Morris provide, there are three potential solutions to this issue. However only the *underdamped case* results in an oscillating movement of the body, as illustrated in Figure 2.4. In order to apply a PINN to this problem, we require a set of training data x. This consists of pairs of time points and corresponding displacement measurements $(t^{(i)}, u^{(i)})$, where $i \in \{1, ..., N_t\}$. In this hypothetical case, we know the mass $m = 1kg$, and the spring constant $k = 200\frac{N}{m}$ and the initial displacement $u^{(1)} = 1$ and $\frac{du(0)}{dt} = 0$. However, we do not know the value of the friction μ. In this case the loss function,

$$\mathcal{L}_{osc}(\boldsymbol{x}, \boldsymbol{u}, \hat{\boldsymbol{u}}) = (u^{(1)} - 1) + \frac{du(0)}{dt} + ||m\frac{d^2u}{dx^2} + \mu\frac{du}{dx} + ku||^2 + \frac{1}{N_t}\sum_{i=1}^{N_t} ||\hat{\boldsymbol{u}}^{(i)} - \boldsymbol{u}^{(i)}||^2, \tag{2.24}$$

includes the border conditions, the residual, in which μ is a learnable parameter and the observation loss.

2.5.1 Disease Informed Neural Networks 1

In this section, we describe the capability of MLP's to solve systems of differential equations. In Section 2.3.1, we describe the SIR model, which models the relations of susceptible, infectious and removed individuals and simulates the progress of a disease in a population with a constant size. A system of differential equations models these relations. Shaier *et al.* [SRS21] propose a method to solve the equations of the SIR model using a PINN, which they call a *disease-informed neural network* (DINN).

To solve Equation (2.8) we need to find the transmission rate β and the recovery rate α. As Shaier *et al.* [SRS21] point out, there are different approaches to solve this set of equations. For instance, building on the assumption, that at the beginning one infected individual infects $-n$ other people, concluding in $\frac{dS(0)}{dt} = -n$. Then,

$$\beta = -\frac{\frac{dS}{dt}}{S_0 I_0} \tag{2.25}$$

would calculate the initial transmission rate using the initial size of the susceptible group S_0 and the infectious group I_0. The recovery rate, then could be defined using the amount of days a person between the point of infection and the start of isolation d, $\alpha = \frac{1}{d}$. The analytical solutions to the SIR models often use heuristic methods and require knowledge like the sizes S_0 and I_0. A data-driven approach such as the one that Shaier *et al.* [SRS21] propose does not have these problems. Since the model learns the parameters β and α while learning the training data consisting of the time points $\boldsymbol{t}$, and the corresponding measured sizes of the groups $\boldsymbol{S}, \boldsymbol{I}, \boldsymbol{R}$. Let $\hat{\boldsymbol{S}}, \hat{\boldsymbol{I}}, \hat{\boldsymbol{R}}$ be the model predictions of the groups and $r_S = \frac{d\hat{\boldsymbol{S}}}{dt} + \beta\hat{\boldsymbol{S}}\hat{\boldsymbol{I}}, r_I = \frac{d\hat{\boldsymbol{I}}}{dt} - \beta\hat{\boldsymbol{S}}\hat{\boldsymbol{I}} + \alpha\hat{\boldsymbol{I}}$ and $r_R = \frac{d\hat{\boldsymbol{R}}}{dt} - \alpha\hat{\boldsymbol{I}}$ the residuals of each differential equation using the model predictions. Then,

$$\begin{aligned}\mathcal{L}_{SIR}() = ||r_S||^2 + ||r_I||^2 + ||r_R||^2 + \frac{1}{N_t}\sum_{i=1}^{N_t} ||\hat{\boldsymbol{S}}^{(i)} - \boldsymbol{S}^{(i)}||^2 + \\ ||\hat{\boldsymbol{I}}^{(i)} - \boldsymbol{I}^{(i)}||^2 + \\ ||\hat{\boldsymbol{R}}^{(i)} - \boldsymbol{R}^{(i)}||^2,\end{aligned} \tag{2.26}$$

is the loss function of a DINN, with α and *beta* being learnable parameters.

Chapter 3

Methods 8

3.1 Data Preprocessing 3

3.1.1 RKI Data 2

3.1.2 Recovery Queue 1

3.2 PINN for the SIR Model 3

3.3 PINN for the reduced SIR Model 2

Chapter 4

Experiments 10

4.1 SIR Model 5

4.1.1 Setup 1

4.1.2 Results 4

4.2 Reduced SIR Model 5

4.2.1 Setup 1

4.2.2 Results 4

Chapter 5

Conclusions 5

5.1 Further Work

Bibliography

[And91] Anderson, Robert M. Roy Malcolm; May M. Roy Malcolm; May: *Infectious diseases of humans : dynamics and control*. Oxford University Press, 1991

[Dem21] Demtröder, Wolfgang: *Lehrbuch*. Bd. 1: *Experimentalphysik 1*. 9. Auflage. Berlin : Springer Spektrum, 2021. – ISBN 978–3–662–62727–3. – Auf dem Umschlag: Mit über 2,5 h Lösungsvideos zu ausgewählten Aufgaben

[EK05] Edelstein-Keshet, Leah: *Mathematical Models in Biology*. Society for Industrial and Applied Mathematics, 2005

[GBC16] Goodfellow, Ian ; Bengio, Yoshua ; Courville, Aaron: *Deep Learning*. MIT Press, 2016. – http://www.deeplearningbook.org

[HSW89] Hornik, Kurt ; Stinchcombe, Maxwell ; White, Halbert: Multilayer feedforward networks are universal approximators. In: *Neural Networks* 2 (1989), Januar, Nr. 5, S. 359–366. http://dx.doi.org/10.1016/0893-6080(89)90020-8. – DOI 10.1016/0893–6080(89)90020–8. – ISSN 0893–6080

[KM27] Kermack, William O. ; McKendrick, A. G.: A contribution to the mathematical theory of epidemics. In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 115 (1927), August, Nr. 772, S. 700–721. http://dx.doi.org/10.1098/rspa.1927.0118. – DOI 10.1098/rspa.1927.0118. – ISSN 2053–9150

[LLF97] Lagaris, I. E. ; Likas, A. ; Fotiadis, D. I.: Artificial Neural Networks for Solving Ordinary and Partial Differential Equations. (1997). http://dx.doi.org/10.48550/ARXIV.PHYSICS/9705023. – DOI 10.48550/ARXIV.PHYSICS/9705023

[Mat84] Matsumoto, T.: A chaotic attractor from Chua's circuit. In: *IEEE Transactions on Circuits and Systems* 31 (1984), Dezember, Nr. 12, S. 1055–1058. http://dx.doi.org/10.1109/tcs.1984.1085459. – DOI 10.1109/tcs.1984.1085459. – ISSN 0098–4094

[MP72] Minsky, Marvin ; Papert, Seymour A.: *Perceptrons*. 2. print. with corr. Cambridge/Mass. [u.a.] : The MIT Press, 1972. – ISBN 9780262630221. – Literaturangaben

[MPF23] Millevoi, Caterina ; Pasetto, Damiano ; Ferronato, Massimiliano: A Physics-Informed Neural Network approach for compartmental epidemiological models. (2023). http://dx.doi.org/10.48550/ARXIV.2311.09944. – DOI 10.48550/ARXIV.2311.09944

[Oks00] Oksendal, Bernt: *Stochastic Differential Equations*. 5th ed. Berlin, Heidelberg : Springer Berlin / Heidelberg, 2000 (Universitext Ser.). – ISBN 3–540–63720–6. – Description based on publisher supplied metadata and other sources.

[RHW86] Rumelhart, David E. ; Hinton, Geoffrey E. ; Williams, Ronald J.: Learning representations by back-propagating errors. In: *Nature* 323 (1986), Oktober, Nr. 6088, S. 533–536. http://dx.doi.org/10.1038/323533a0. – DOI 10.1038/323533a0. – ISSN 1476–4687

[Ros58] Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. In: *Psychological Review* 65 (1958), Nr. 6, S. 386–408. http://dx.doi.org/10.1037/h0042519. – DOI 10.1037/h0042519. – ISSN 0033–295X

[RPK17] Raissi, Maziar ; Perdikaris, Paris ; Karniadakis, George E.: *Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations*

[Rud07] Rudin, Walter: *Analysis*. Oldenbourg Wissenschaftsverlag GmbH, 2007

[Sch26] Schrödinger, E.: An Undulatory Theory of the Mechanics of Atoms and Molecules. In: *Physical Review* 28 (1926), Dezember, Nr. 6, S. 1049–1070. http://dx.doi.org/10.1103/physrev.28.1049. – DOI 10.1103/physrev.28.1049. – ISSN 0031–899X

[SRS21] Shaier, Sagi ; Raissi, Maziar ; Seshaiyer, Padmanabhan: *Data-driven approaches for predicting spread of infectious diseases through DINNs: Disease Informed Neural Networks*

[TP85] Tenenbaum, Morris ; Pollard, Harry: *Ordinary Differential Equations*. Harper and Row, Publishers, Inc., 1985

List of Figures

List of Tables