Predicting enzyme kinetical data with neural ordinary differential equations

Bachelor's Project Thesis

Vincent de Wit, s3038858, v.j.de.wit@student.rug.nl,
Supervisor: dr. K. Bunte
Corrector: prof. dr. L.C. Verbrugge

July 28, 2019

Abstract: This research looks into a new evolution of neural networks: neural ode's. A neural ode is a neural network which hidden state space is modelled by a transformational neural network as a differential equation. This means that the hidden state is a continuous vector space instead of consisting of discrete layers. In this research, a network is trained and tested on a small data set of in vivo glucocorticoid activation patterns, by means of leave-one-out cross-validation. The data set contains twelve data points taken over four hours for twelve patients with differing age and sex; each data point consists of a measurement of cortisone, cortisol, prednisone and prednisolone concentration. The performance of the system is measured by means of comparing predicted trajectories of samples from pharmacokinecticaly predicted trajectories to trajectories fitted to patient data. We conclude that the neural ode in some cases is able to predict the enzyme dynamics. Whether the network is able to do this depends strongly on the chosen initial condition, which should be further researched.

Keywords: Neural ode, glucocorticoid activation, leave-one-out cross-validation, differential equation

Contents

1 Introduction

When patients are administered a medicine, insights in the interaction between the patients body and the drug are of the utmost importance. When the physician knows more about how the patient will respond to a drug, the physician will be able to better understand and predict how much of a certain medicine the patient needs. Prednisone is an example of a medicine where the just mentioned issues play an important role. Prednisone is used by physicians to reduce the body's immune system response as to counter inflamation. Furthermore, it is a corticosteroid and it is converted into its active counter part: prednisolone. The conversion is handled by enzymes that are created in the liver. Besides converting prednisone, these enzymes also convert cortisone and cortisol. Cortisone and cortisol are endogenous hormones of the body (Bunte, Smith, Chappell, Hassan-Smith, Tomlinson, Arlt, and Tino (2018)). Since there is a limited amount of enzymes, administering prednisone also influences the concentrations of the endogenous hormones cortisone and cortisol. Besides these effects, prednisone, as any other medicine, also has other side-effects that can influence the patient's health. It is easily understood that accurate pre-emptive patient-specific knowledge over the amount of prednisone needed, is valuable information for a physician when administering prednisone. Given the value of such insight, in this paper the enzyme kinetics which are involved in the conversion of prednisone to prednisolone are investigated. In particular it is assessed how well a neural ordinary differential equation network (Chen, Rubanova, Bettencourt, and Duvenaud (2018)) is able to capture the dynamics involved in these conversion patterns.

Previous attempted research in time series prediction has often focused at weather prediction by means of machine learning; (Maathuis, Boulogne, Wiering, and Sterk (2017); Aras and Kocako (2016)). Other research in the field of neural networks has been done in neural network optimization (Livieris and Pintelas (2019); Rojas Delgado, Trujillo-Rasa, and Bello-Prez (2019)). Further research in machine learning focusses on multiple paradigms as can be found in Oneto, Bunte, and Schleif (2019). As introduced earlier Bunte et al. (2018) uses machine learning and perturbation theory to predict glucocorticoid activations patterns. A new state of the art neural network was introduced in Chen et al. (2018) this new type of neural network will be tested on those glucocorticoid activation patterns (enzyme kinetics).

To investigate these enzyme kinetics, a new type of neural network is used (Chen et al. (2018)). These new types of neural networks –also called "neural ordinary differential equation networks" or abbreviated "neural ode network"– have a continuously defined hidden state space instead of the usual discretely defined hidden state space. To create a continuous hidden state space, the hidden layers are defined as an ordinary differential equation. These functions, that are defined by their derivative, determine the amount of change that occurs in each point. Furthermore, by determining the gradients of a vector field and saving these gradients, a neural ode is able to learn the description of a dynamical system. In doing so, the system is able to propegate a starting point through time in the dynamical system and output the predicted state at specified evaluation points. Therefore, the hidden state space of the system can be used to model a specific dynamical system or, conversely, solely for computation. In this paper, the neural ode is used and tested at describing enzyme kinetical data. The goal of this paper is to examine whether the network is able to generalize the enzyme kinetical dynamics and subsequently able to make accurate distinguishable predictions based on an initial state linked to each patient.

The enzyme kinetics are described by a system of functions, which is derived using perturbation theory using observations of glucocorticoid conversion patterns from patients (Bunte et al. (2018)). The neural ode network is trained on these enzyme kinetics data and is tested to see if it was able to learn the dynamics of the enzyme kinetics. During training the network might extrapolate the dynamics from the changes in concentrations given by the measurements. If the dynamics are succesfully learned, the training results in a system which is able to interpolate the trajectories of the concentrations based on a given initial state. Therefore this research questions the network's predictive capabilities. How accurate are the predictions the system can make given the data it trains on? How big is the error in circumstances that differ in the number of data points and amount of noise? But also, is

the network able to make predictions that show the general dynamics of the enzyme kinetical system?

The paper addressess these questions in the following form. First section 2 describes the methods which are used. After which section 3 describes the results that are derived by means of the methods. Lastly section 4 elaborates on the results and method by contrasting it with eachother. Section 4 concludes with a summary of what has been discussed in the paper.

2 Methods

2.1 Neural ode

Continuous/Discrete

Neural networks come in many shapes and sizes, but all neural networks make use of nodes and connections. Due to nodes and connections, the input is parsed discretely (step-wise) into output. The hidden space of such neural networks can be described by the following formula:

$$h_{t+1} = h_t + f(h_t, \theta_t),$$

which shows that the function of a neural network can be viewed as a chain of a finite number of sub-functions. In this function, h_t denotes hidden layer t, where $t = 0$ is the first hidden layer. The output of f depends on the current hidden layer h_t and the parameters θ_t associated to it. The next hidden layer, h_{t+1}, is therefore defined as the sum of the current hidden layer and the output of f. When considering such a definition for the hidden space, one can also redefine this hidden space to be continuous by using an ordinary differential equation:

$$\frac{dh(t)}{dt} = f(h(t), t, \theta).$$

In this function, the hidden state $h(t)$ is defined by it's change; it is therefore a differential equation. When the hidden space of a neural network is defined in this way, the hidden space can be called a dynamcial system and is continuous (Chen et al. (2018)). The main difference between a discrete and continuous hidden state space, is exemplified in figure 2.1. This figure shows that the continuous hidden state space changes more smoothly compared to the discrete hidden state space. This leads to more reasonable predictions for evaluation points in between data points in the training set. Since, the data points are brought into a smooth relation with each other. Furthermore is the entire hidden state space changed into the dynamical system. Which is usefull in relation to the option of choosing any evaluation point in the system. This neural ode is the network which will be used for the experiments in this paper.

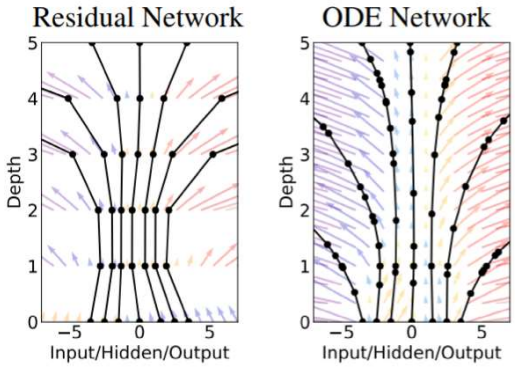


Figure 2.1: Left: A Residual network defines a discrete sequence of finite transformations. Right: An ODE network defines a vector field, which continuously transforms the state. Both: Circles represent evaluation locations. (From: Chen et al. (2018))

Software

To use a neural ode the hidden state space needs to be trained to describe a dynamical system. The training process of the neural ode network is executed in Python, using the pytorch library and the torchdiffeq package (Chen et al. (2018)). The torchdiffeq package implements multiple black box differential equation solvers and uses an implementation of the ajdoint sensitivity method of Pontryagin, Mishchenko, Boltyanskii, and Gamkrelidze (1962) for backpropegation. With this, the code base is able to solve initial value problems. By giving one of the solvers the neural network –which has learned to describe the state space– and an initial state together with evaluation points. Furthermore, Matlab is used together with the fminsearchbnd function package (D'Errico (2012)) to derive optimal parameters for each pa-

tient to describe their respective enzyme kinetics. To find these optimal parameters fminsearchbnd is executed 50 times and the parameters with the lowest error are used. The error of the parameters is relative to the mismatch between the curve and the data points observed in the patients blood over time. In addition to Python and Matlab also excel was used to store the patient data and to aggregate the RMSE (root mean squared error) of the predictions. The RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2},$$

where $x_n - y_n$ is the difference between the curve fitted to the data and the prediction, with N denoting the number of data points.

Backpropagation

Backpropagation is different for a neural ode compared to most neural networks. This is due to the fact that instead of weigths and thresholds, the system now consists of a dynamical system. Therefore, in backpropagation gradients are changed such that the result of the loss function is minimized. Collecting the gradients that describe the dynamical system, through the forward pass of the dynamical system, costs a lot of memory. The writers of the torchdiffeq package therefore implemented an alternative method of backpropagation. This method is called the adjoint sensitivity method and is mostly attributed to the mathematician Pontryagin (Pontryagin et al. (1962)). In this method of backpropagation, first the end point of the trajectory is calculated by solving an inital condition problem for the dynamical system. After this, the system is solved backwards from the end time point to the first time point. To solve the dynamical system backwards, the calculated end point is used as the intial condition and the evaluation points are reversed in order. During the evaluation of each intermittant time point, an adjustment is made to the gradient towards what the gradient at that time point should be. This way the transformational neural network is changed and learns to describe the dynamics of the enzyme kinetic system.

Usage neural ode

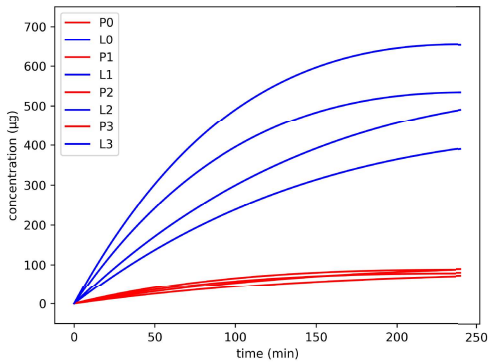
As explained earlier the neural ode network is different from a neural network regarding the transformation space of the network. This transformation space is a continuous vector space, the description of this transformation space is stored in a neural network. The neural network takes the role of a transformation matrix and can therefore be called a transformational neural network (TNN). The writers of the torchdiffeq package chose this TNN to be a neural network, so that the description of the dynamical system can easily be changed by means of backpropagation (Chen et al. (2018)). The procedure of backpropagation on the vectorspace uses the adjoint method (Pontryagin et al. (1962)) which allows the system to be able to use only little memory. To create a prediction, a function call is made to an ode solver. The solver solves the dynamical system with respect to an initial condition and a time series. Then the trajectory is calculated by solving the dynamical system with respect to the initial condition. This is done by means of an evaluation of the transformational neural network for each given time point.

2.2 Data set

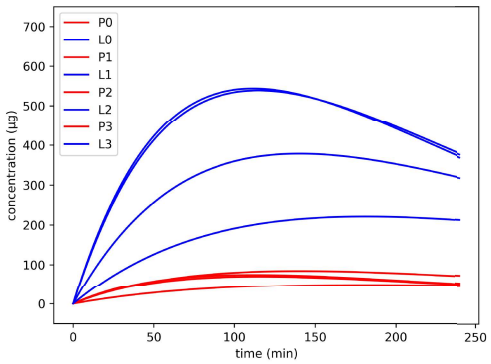
In this paper a dataset on glucocorticoid activation patterns is used from the Queen Elizabeth Hospital Birmingham, where 12 patients were given prednisone (Bunte et al. (2018)). Each of the patients was given 10 mg of prednisone. For each patient, the effect of prednisone has been measured in 20 minute intervals, totalling to four hours. We thus have 12 observations for each of the 12 patients. At each measurement, the level of prednisone, prednisolone, cortisone and cortisol have been examined. Remember that prednisone first goes to the stomach, from where it is absorbed into the blood stream. Once the prednisone is in the blood stream, it is converted into prenisolone and from prednisolone back to prednisone. This conversion is effectuated by two isozymes of 11β-HSD (Bunte et al. (2018)). The data set has been checked for outliers, adding up together with missing data-points to 16 non available values.

Dynamical system

Prednisone and prednisolone form together with, the two enzymes needed for the conversion, a linear system of differential equations. However, since the enzymes are endogenous to the human body, they



(a) The group of patients whose prednisolone (L) concentration does not decrease over the plotted domain (group 1).



(b) The group of patients whose prednisolone (L) concentration starts declining within the plotted domain (group 2).

Figure 2.2: Two groups of four patients, for each patient the concentration of – prednisone (P) is plotted in red and prednisolone (L) in blue –.

also take part in other processes. One of the processes they are involved in is the conversion between cortisone and cortisol. These two substances also are endogenous to the human body and are released by the andrenal gland. They play an important role in, amongst others: memory, stress, immune-response and metabolism. When prednisone enters the blood it is making, due to a high binding strength, a demand on the enzymes for conversion. The enzymes, which thus slightly prefer to bind to prednisone and prednisolone, start converting the exogenous substances, the endogenous substances start building up due to the enzymes being busy converting the exogenous substance. These interrelations result in more complex dynamics that are captured in a non-linear system. However, when excluding cortisone and cortisol, the system becomes a linear system of differential equations. This system has a parameter-space of six dimensions, which consists of parameters that describe: the conversion rate from prednisone to prednisolone and backwards, the excretion of prednisone and prednisolone and the amount of prednisone applied together with it's absorbtion rate (Bunte et al. (2018)). It is this part of the system that the network will be trained and tested on.

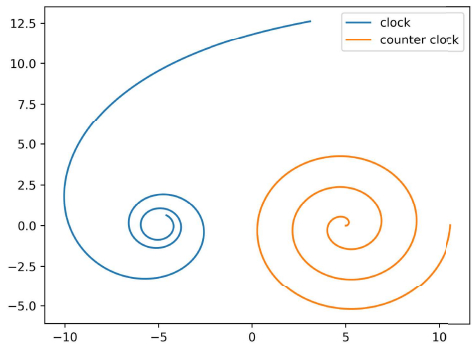
For the experiment, curves are derived from the data points, by defining the introduced parameters. This is done by means of optimizing the parameters on the following formulas for prednisone and prednisolone:

$$S(t) \approx -k_{abs} S_0,$$

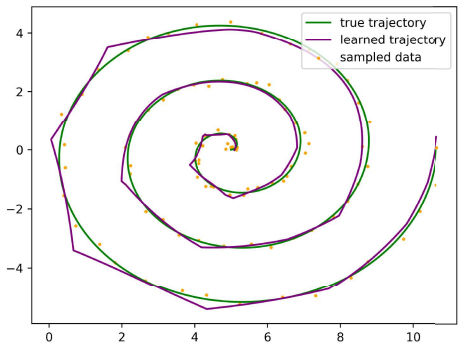
$$P(t) \approx \frac{S_0 k_{abs} k_{LP}}{k_{PL} k_{Lex} + K_{LP} k_{Pex} - k_{abs}(k_{PL} + k_{LP})} \cdot (\exp(-k_{abs} t) - \exp(-\frac{k_{LP} k_{Pex} + k_{PL} k_{Lex}}{k_{PL} + k_{LP}} t)),$$

$$L(t) \approx \frac{S_0 k_{abs} k_{PL}}{k_{PL} k_{Lex} + K_{LP} k_{Pex} - k_{abs}(k_{PL} + k_{LP})} \cdot (\exp(-k_{abs} t) - \exp(-\frac{k_{LP} k_{Pex} + k_{PL} k_{Lex}}{k_{PL} + k_{LP}} t)).$$

The first function describes the concentration of prednisone present in the stomach ($S(t)$) as a function of time. The function is dependent on the absorbtion rate of the substance (k_{abs}) and the content of the tablet (S_0). The concentration of prednisone in the blood at time t, is described by the function $P(t)$ and the concentration of prednisolone is described by $L(t)$. Both functions of time (t) in minutes. These two functions introduce four more parameters: k_{LP}, k_{PL}, k_{Lex} and k_{Pex}, which stand respectively for the conversion rate from L to P, the conversion rate from P to L, the excretion of L and the excretion of P.



(a) Ground truth which shows a clockwise and counter-clockwise turning spiral.



(b) The neural ode predicting a spiral on basis of drawn samples from the spiral. There are 100 samples each with .3 noise on it's x and y coordinate.

Figure 2.3: Ground truth and prediction of the synthetic experiment.

2.3 Experiment

For the experiment, a transformational neural network is used that has an input and output layer of two nodes and a hidden layer of 50 nodes together with a tanh activation function. Furthermore, the learning rate of the neural network is 0.01 and the number of iterations for two training sessions is 200 respectively 500 iterations. To train the model, sampled data is derived from the curves fitted to the observations of the patient-blood data points. In section 2.2 the dynamics of the system were defined and it was explained how parameters for a patient were found. From these parameters and functions curves for each patient are created. The curves approximate the changes of patients' prednisone and prednisolone concentrations.

The curves are grouped by the curves' characteristics: one group consists of curves that only rise over the domain and the other group consists of curves that reach their respective maximum within the domain (Figure 2.2). Besides these two groups, there are also curves that can be considered outliers*; these are not included in the experiment. After the groupings have been made, samples are created with noise from the curves. Multiple experiment settings are created to test the system. The neural ode is tested in three experiment settings: easy, normal and hard. Each setting has a different number of data points and deals with a different level of noise. The easy setting has 25 samples and 10 noise; the medium setting has 13 samples with 20 noise; and the hard setting has 7 samples with 40 noise. The degree of noise influences the concentrations of the samples. The concentrations are indicated by (nmol/L): the tablets consumed by the patients hold 10mg which is 10000μg. Compared to the medium setting, the easy setting has halve the noise and double the data points and the hard setting has halve the data points and double the noise.

The training procedure can thus be described for a setting – low, medium and hard – and group combination – strictly rising curves and not strictly rising curves. As can be seen in Figure 2.2, each group has four members, which is not very many. Therefore, leave-1-out cross-validation is used to maximize the utility of the data. There are four folds per condition in which each member will be left out of training once, to test the trained system on this left out member. Then the procedure is repeated five times to ensure robustness of results. Hereafter, the training over the data set is performed, as earlier noted for 200 iterations and another time for 500 iterations. The training results in 24 graphs each containing five pred-

*A curve is characterized as outlier if the curve shows extra-ordinary dynamics. A total of four curves are excluded as outlier.

nisone predictions, five prednisolone predictions and the samples the neural ode is predicting. Also the RMSE, of the prediction with respect to the curve from which the samples are drawn, is computed and averaged over the five repetitions. It is expected that this average RMSE is smaller for the "easy" setting and highest for the "hard" setting. This would mean that if there are more data points with less noise the neural ode is able to make predictions that are closer to the curve.

Spiral

The first interactions with the torchdiffeq package were mainly focused at learning how to use the neural ode. In the package were a few examples and one of these 'latent_ode.py' (Chen et al. (2018)) is a program which learns to predict spiral trajectories. This is done by means of samples from, and initial conditions of, the true trajectory. This initial condition is derived by the program using statistical analysis (Chérief-Abdellatif (2018)), by using the output of a recurrent neural network which is given all samples of the spiral. After making adjustments by spacing the samples out over the entire trajectory, managing the noise and changing the initial condition to the starting point of the spiral, the synthetic experiment could be run. A total of 1000 spirals were made with different shapes, i.e. ranged from round to oval like, clockwise and counter-clockwise. For the spiral data the neural ode was presented with samples from a range of different spiral curves. This resulted in the following prediction of the model as shown in Figure 2.3. The figure clearly shows that the fit is not perfect, as there are parts of the prediction in which the gradient stays more or less the same while the gradient should change with a constant value in order to make the spiral smooth. The resulting vector space depicts a spiral dynamic, though at some points in the state space the dynamics are smoother than at other points.

3 Results

Enzyme kinetics

To estimate enzyme kinetics with the neural ode, adaptations are made to the code(Chen et al. (2018)) used in the synthetical experiment involving spirals. The adjusted code takes care of the enzyme dynamics experiment, encompassing code that creates the enzyme kinetical curves and code which makes samples according to the needed setting. The experiment with the enzyme kinetics consists of multiple parts taken from the code for the spiral dynamics, but with additional code. Furthermore, grouping the curves of patients into two groups as in Figure 2.2, the leave-one-out cross-validation and repetitions are also included. Therefore, to run the experiment only the number of iterations for one training context need to be specified. Training is done for all the circumstances[†] described in the method section. After training, a prediction is made on the item that was left out of the training set and the prediction is saved. After all training is done and the predictions have been made, graphs are created (Figure 3.1). Each graph contains five prednisone predictions and five prednisolone predictions, also the samples that the neural ode attempts to predict are depicted. The average RMSE over the predictions is added and a label which differentiates the graph from the other graphs based on it's setting, group and fold. Therefore the main means of results are shown in the form of the graphs and the average value of the RMSE with standard deviation (Table 3.1 & Table 3.2).

Group \ Setting	Hard	Medium	Easy
1	120(17)	67(45)	76(46)
2	110(26)	74(50)	64(37)

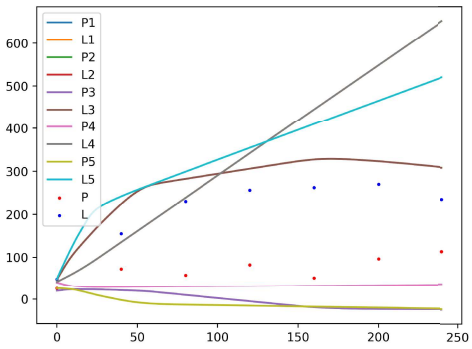
Table 3.1: Average RMSE and standard deviations for 200 training iterations.

Group \ Setting	Hard	Medium	Easy
1	122(32)	91(66)	75(59)
2	133(27)	80(59)	77(38)

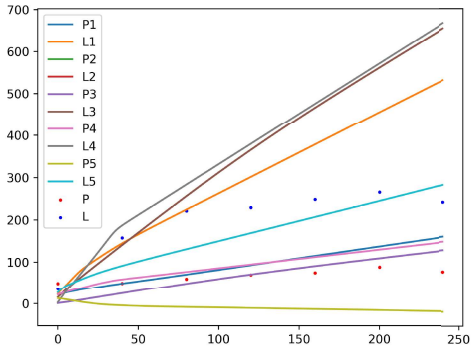
Table 3.2: Average RMSE and standard deviations for 500 training iterations.

Tables 3.1 and 3.2 show that in general the RMSE declines when the number of samples increases and the amount of noise decreases. This was expected, since this means that the condition spec-

[†]four folds, three settings and two groups; this calculates to: 4*3*2=24, 24 circumstances.



(a) Graph of group 1 with setting "hard" and fold four, after 200 iterations of training. The average RMSE of the ten curves is 118.45.



(b) Graph of group 1 with setting "hard" and fold four, after 500 iterations of training. The average RMSE of the ten curves is 133.42.

Figure 3.1: Neural ode predictions. Two graphs which both show five predictions, each prediction consists of a prednisone and prednisolone component (10 curves).

ified as "easy" has lower RMSE than the condition specified as "hard". Besides this result, the average RMSE is in general higher when training consisted of 500 iterations compared to 200 iterations. The most sensible explanation for this is overfitting though this could also be due to the distribution of samples. This shows that there is an ideal range of training iterations that results in the lowest RMSE. It is moreover interesting to note that in the case of 200 iterations (Table 3.1), only the rising curves (ie. group 1) show a lower RMSE in the "medium" case than in the "easy" case. This is most likely due to the training set of samples being closer to the test trajectory in the "medium" case compared to the "easy" case. Lastly these values of RMSE are quite high. If the system is to be used to accurately distinguish between patients, the RMSE should be lower.

Besides the RMSE, the dynamics that the system has learned can be evaluated by evaluating the graphs. There is a total of 24 circumstances, each of which has a graph with ten curves: five of prednisone and five of prednisolone. Two of such graphs from two experiments, one with 200 iterations the other with 500 iterations, are depicted in figure 3.1.a and figure 3.1.b. The majority of graphs show that the system has been able to learn a general trend, namely that there is an increasing component in the dynamical system. A minority of graphs shows that the dynamic was learned, even though awkwardly and barely. The graphs, however, also indicate that the neural ode was in general not able to learn the decreasing component of/in the dynamical system. This means that most predictions only rise over the evaluated domain, whereas the patient-data either stops increasing or starts decreasing. Moreover, the graphs show that the neural ode has trouble predicting the maxims of the curves. Furthermore, some figures show predictions involving negative concentrations, such concentrations are meaningless.

4 Discussion & Conclusion

The results show that the RMSE decreases as the difficulty decreases. This is a result is in line with what was expected, though this is not shown by one case. In the 200 iteration training session, group 1 shows an average RMSE which is lower in the medium condition compared to the easy condition. This could be the result of an unlucky sampling of the ground truth curves. Also the performance of the system decreases after a certain number of training iterations. The most likely cause for this is that the model is overfitted to the training data. Besides the RMSE, a resulting

graph was shown in which multiple predictions have been visualized. The predictions show that the general trend of how the curves behave is learned but the model does not seem to learn to generalize the dynamics of the glucocorticoid system. This is shown by curves that do increase as the fitted curve does, but does not diminish the derivative as it approaches the maximum. Another example are curves that bend awkwardly at a few points, or graphs that show that the prediction skips the maximum and instead only increase towards the concentration level at the last time point.

At the moment of training, the intital condition was defined as the value of the concentration of prednisone and prednisolone at time point zero. But at time point zero, the prednisone is being consumed by the patients. This means that at time point zero, the patients all should have a concentration of 0 for both prednisone and prednisolone. Though due to the way the noise was defined the initial concentration could differ from this value by a value +/- a proportion of the amount of noise. This resulted in different initial conditions for each of the trajectories which also could be negative. This way of defining the noise and giving time point zero as a starting point for the prediction is not completely realistic, because this means that the initial condition can be negative. To circumvent this problem, the noise could be defined as scaling with respect to the concentration value. This would result in a concentration of zero when the concentration is zero. This leaves the problems surrounding the initial condition, since if every initial condition is zero and therefore the same, the model will predict the same each and every time. In other words, due to the dynamical system becoming static after training, it reacts the same to input which is the same. So in order for the predictions to differ, the initial conditions have to be different from eachother. In order to make predictions, the initial condition has to be such that the model is able to predict the trajectory. The parameter space of the functions that describe the curves has six dimensions, ie. three absorbtion rates, two excretion rates and the intial amount of prednisone. This means that there are many curves possible and the initial condition is the only thing which distinguishes one prediction from the other. Furthermore, there are constraints on what is known beforehand. The circumstances in which the network is used give constraints to what the intial condition can be. Therefore the information availabe to the physician about the patient is very important. Because the information the physician has is the basis of the initial condition. Therefore the time point 0 can only be used if besides prednisone and prednisolone concentrations, which are both 0 at that time, another charactarizing component can be presented. Another different charactarizer of a concentration curve is the first data point at time point 20 minutes. Using this data point will very likely result in predictions with low RMSE and should be persued in new experiments. Though this initial condition might not be very usefull to a physician since this requires to first measure the concentrations after 20 minutes. Which implies that in order to predict the concentration dynamics, the concentration dynamics should first be measured, this would nullify the purpose of predicting.

The neural network to which the description of the vector space is saved also plays a large role in the performance of the model. The neural network that describes the vector space can differ with regards to the activation function but also with regards to the number of layers, nodes and connections. These last parameters mainly regard the amount of information the neural network can hold, whereas the activation function mainly influences the way the description of the dynamical system is saved. The transformation neural network used in the experiment had one hidden layer of 50 nodes and both input and output layers had 2 nodes. All nodes of the input and output layers were connected to each hidden node. As activation function it used the arctan function, alternatives are the sigmoid and the relu activation function. Besides testing the difference between these activation functions, it would have been interesting to see how the RMSE of the predictions made by the model would be influenced by changing the number of –nodes and layers– of the neural network to which the dynamical system is saved. Since the model had difficulties learning the dynamics of the system from the training data and the neural network holds the description of the vector space, changing the structure of the transformational neural network could make the difference. It could very well be that the size of the neural network has a direct effect on the

ability of the model to generalize the dynamics. For instance, due to lack of storage capacity the model might be forced to generalize the dynamics, whereas if there is an excess of storage capacity the model might be inclined to specifically learn an item from the training set.

Not only is the RMSE measured for the results as indication of the perfomance of the system, it is also used after the adjoint sensitivity method as means of loss function. The loss is defined as the RMSE between the prediction and the sampling and given as a model selection signal to the neural network which stores the description of the dynamical system. This function could also be changed, for example the mean absolute error can be used or the elbo model selection method (Chérief-Abdellatif (2018)). Just as with the initial condition the choice of heuristic that defines the loss is **important**. By defining the loss and the initial condition using a heuristic function it is possible that the network is able to save and process latent relations among the data and use those to make more accurate predictions.

Conclusion

An experiment was performed to test the performance of the new type of neural network: the neural ode. The experiment made use of a dataset on glucocorticoid activation patterns (Bunte et al. (2018)). For the experiment, curves were fitted to approximate the data and from these curves data points where sampled. From the resulting set of curves, outlier curves were removed and the curves were grouped into two groups of four items. One group consists of curves that were strictly rising over the domain. The other group was not strictly rising over the considered domain. To implement different learning conditions for the model, different difficulties where implemented. The difficulty settings differ in the number of datapoints and the amount of noise. Also due to the size of the training set, leave-one-out cross validation is used to maximize utility of the data. Lastly all circumstances are trained on by a new model a total of five times to establish a robust result. All predictions made in the same circumstances are plotted in a graph together with the average RMSE of each prediction compared to the corresponding pre-sampled curve. The RMSE showed to be lowest in circumstances with many data points and little noise, and showed to be highest in circumstances with little data points and high noise. For backpropagation the system makes use of the RMSE as loss function and the adjoint sensitivity method (Pontryagin et al. (1962)) which has been implemented in the torchdiffeq package (Chen et al. (2018)). The RMSE is too high for accurate predictions. Also the system has trouble figuring out the dynamics of the enzyme kinetical system. This is shown by the predictions which mostly show straight lines with bending points and not as smooth curves. The model does not necessarilly produce better predictions with more learning. Changing the initial condition into an indicative heuristic, instead of the first time-point will increase the predictive power of the model. This way the model could learn latent relations between the more robust initial conditions and their respective curves.

References

Serkan Aras and Ipek Deveci Kocako. A new model selection strategy in time series forecasting with artificial neural networks: Ihts. *Neurocomputing*, 174:974 – 987, 2016. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2015.10.036.

Kerstin Bunte, David J. Smith, Michael J. Chappell, Zaki K. Hassan-Smith, Jeremy W. Tomlinson, Wiebke Arlt, and Peter Tino. Learning pharmacokinetic models for in vivo glucocorticoid activation. *Journal of Theoretical Biology*, 455:222 – 231, 2018.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. *CoRR*, abs/1806.07366, 2018. URL `https://github.com/rtqichen/torchdiffeq`.

Badr-Eddine Chérief-Abdellatif. Consistency of ELBO maximization for model selection. *arXiv e-prints*, art. arXiv:1810.11859, Oct 2018.

John D'Errico. fminsearchbnd, 2012. URL `https://www.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd-fminsearchcon`.

Ioannis E. Livieris and Panagiotis Pintelas. An adaptive nonmonotone active set weight constrained neural network training algorithm. *Neurocomputing*, 2019. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2019.06.033.

Henry Maathuis, Luuk Boulogne, Marco Wiering, and Alef Sterk. Predicting chaotic time series using machine learning techniques. pages 326–340, 2017.

Luca Oneto, Kerstin Bunte, and Frank-Michael Schleif. Advances in artificial neural networks, machine learning and computational intelligence: Selected papers from the 26th european symposium on artificial neural networks, computational intelligence and machine learning (esann 2018). *Neurocomputing*, 342:1 – 5, 2019. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2019.01.081. Advances in artificial neural networks, machine learning and computational intelligence.

Lev Semenovich Pontryagin, EF Mishchenko, VG Boltyanskii, and RV Gamkrelidze. The mathematical theory of optimal processes. 1962.

Jairo Rojas Delgado, Rafael Trujillo-Rasa, and Rafael Bello-Prez. A continuation approach for training artificial neural networks with meta-heuristics. *Pattern Recognition Letters*, 125, 05 2019. doi: 10.1016/j.patrec.2019.05.017.