

Parameter estimation from realistic experiment scenario using artificial neural networks

1st Melánia Puskás

*Physiological Controls Research Center
Research and Innovation Center of Óbuda University
Óbuda University
Budapest, Hungary
puskas.melania@uni-obuda.hu*

2nd Borbála Gergics

*Physiological Controls Research Center
Research and Innovation Center of Óbuda University
Óbuda University
Budapest, Hungary
gergics.borbala@uni-obuda.hu*

3rd Alexander Ládi

*Antal Bejczy Center for Intelligent Robotics
Research and Innovation Center of Óbuda University
Óbuda University
Budapest, Hungary
alexander.ladi@irob.uni-obuda.hu*

4th Dániel András Drexler

*Physiological Controls Research Center
Research and Innovation Center of Óbuda University
Óbuda University
Budapest, Hungary
drexler.daniel@uni-obuda.hu*

Abstract—One of the promising directions in future medicine is the optimization of therapies based on mathematical and engineering methods, with which the treatment can be personalized. In personalization, the key issue is the identification of the model parameters. We carry out the identification using artificial neural networks, which require training. We generate training data that is as realistic as possible, by generating sparse training datasets. We aim to simulate realistic experimental setups that also take the presence of holidays into account, when the drug injection and tumor volume measurement are not possible. We generate several smaller training datasets and their different versions, to determine how many times the parameters can change every week for the proper functioning of neural networks and to examine how the networks can give a proper estimate of the time-varying parameters from sparse measurements. In this research two different types of training datasets will be introduced to train artificial neural network. The training data are generated on known parameter intervals, taking into consideration the experimental setup we use to validate our results. The estimated parameters can be used to track the change of the parameters and personalize the model for optimization algorithms.

Index Terms—tumor model, personalizing therapy, neural networks, identification, sparse dataset

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 679681). Project no. 2019-1.3.1-KK-2019-00007. has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the 2019-1.3.1-KK funding scheme. The present work has also been supported by the Hungarian National Research, Development and Innovation Office no. 2.1.11-T ÉT-SI-2018-00007. The research was also supported by the Eötvös Loránd Research Network Secretariat under grant agreement no. ELKH KÖ-40/2020 ('Development of cyber-medical systems based on AI and hybrid cloud methods. M. Puskás was supported by the ÚNKP-21-2 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

I. INTRODUCTION

Chemotherapy and radiotherapy, as the most commonly used methods for treating cancer besides surgical intervention, do not take into account the unique characteristics of the patients. Mathematical model-based therapy optimization can provide a solution to the harmful effects of therapies developed for average population. The goal of therapy optimization is to create a personalized therapy that minimizes drug dosage, leading to less side effects, preventing drug resistance in theory, and making therapy more cost-effective [1]–[4].

The combination of engineering and medical knowledge can be promising for the optimization and personalization of drug delivery, in the case of tumor chemotherapy. A number of approaches are available to optimize therapy, via, optimal control theory [5]–[9], in conjunction with impulsive control [4], [10]–[12]. In our work, we assume instantaneous administration, thus we use impulsive differential equations to model the tumor [13], [14].

In the literature there are several consideration of mathematical modeling of tumor growth, see for example, the reviews by Jarett et al. [15] or Altrock et al. [16]. Some models describe the tumor growth and supportive vasculature growth [17], others describe the growth of tumor cells, healthy cells and immune cells [18], and there are tumor models which model tumor cell dynamics involving drug resistance [1]. The model used in this paper describes the tumor proliferation, necrosis, dead cell washout, pharmacodynamics and pharmacokinetics of the drug [2], [19], described in Subsection II-A.

A therapy is said to be personalized if the parameters of the mathematical model are determined by the individual parameters of the patient. Identification must be carried out for this purpose. However, there are some parameters with

low sensitivity [20], making their estimation impossible. A solution to this problem could be to generate virtual patient population with realistic parameter values and offline therapies based on the fitted model, and to train estimators with the measurement data, using artificial intelligence [21]–[23]. The realistic parameters are acquired using mixed-effect models [2], [24], [25] based on measurements from in vivo and in vitro experiments [26]. This approach uses local search algorithms, requiring initial estimations for the parameters, and can not handle inpatient variability.

An initial estimate of the model parameters can be obtained using pretrained expert systems. In our previous work we used and compared two different estimators, a neural network (NN), and an adaptive neuro-fuzzy inference system (ANFIS) [27]. We trained the two different estimators with given experimental setups with a large dataset acquired from in silico experiments. The results showed, that with the exception of a few cases, the systems are able to provide a reliable initial estimation of the parameter values that can be fine-tuned later by local search algorithms. The NN performed better on the training data with performance over 90% for almost all cases. The performance of the ANFIS was worse on the training data, however, its generalization power was better. The analysis showed that both experts can be used to estimate the parameters.

In this paper, we are focusing on estimating the model parameters from realistic experiment setups with neural networks. The inputs of the NN have changed compared to previous experimental setups [28], [29]. The experimental setups are detailed in Section III. Our goal is to generate training data that is as realistic as possible. We achieved this by using time-window neural networks which are not learning from daily measurements but estimating parameters from sparse measurement data.

This approach describes more realistic experimental design, as it also takes into account the presence of holidays when the drug injection and measuring tumor volume are not possible. Furthermore, in order to realize virtual patients, we do not generate constant parameters, but create a time-varying set of parameters, and then simulate the measurements with these virtual patients. As in reality, the speed of tumor volume increase or decrease can change during the experiment, moreover the tumor may become more resistant or more sensitive to the given dose. The task of the trained networks is to give an estimate to the value of the time-varying parameters of the given patient, based on the generated data. The algorithm we have developed is able to generate therapy with any input (Section IV). Networks provide adequate performance for training and test datasets also, as well as for sparse measurement data.

II. PRELIMINARIES

A. The tumor growth model

We use a fourth-order differential equation to describe tumor dynamics, the pharmacokinetics, and the pharmacodynamics

of the drug [19]. The system of differential equations of the tumor growths is

$$\dot{x}_1 = (a - n)x_1 - b \frac{x_1 x_3}{ED_{50} + x_3} \quad (1)$$

$$\dot{x}_2 = nx_1 + b \frac{x_1 x_3}{ED_{50} + x_3} - wx_2 \quad (2)$$

$$\dot{x}_3 = -(c + k_1)x_3 + k_2x_4 \quad (3)$$

$$\dot{x}_4 = k_1x_3 - k_2x_4. \quad (4)$$

In these equations, x_1 is the time function of the living tumor volume [mm³], x_2 is the time function of the dead tumor volume [mm³], x_3 is the time function of the drug level in the central compartment (i.e., the blood) [mg · kg⁻¹], and x_4 is the time function of the drug level in the peripheral compartment (i.e., the tissues) [mg · kg⁻¹].

The inputs are the drug injections given on specified days which lead to rapid change in the drug level. We assign a dose to each injection time instant. Let t_k , $k = 0, 1, 2, \dots$ be the times of the injections, while u_k , $k = 0, 1, 2, \dots$ are the injected drug doses in mg · kg⁻¹. Then x_3 has a discontinuity at time t_k , modeled as

$$x_3(t_k^+) = x_3(t_k^-) + \kappa u_k. \quad (5)$$

The model parameters were identified from mice experiments using a mixed-effect model [2], [19], assuming constant parameters. We suppose that these parameters are realistic values, and use them for virtual patient generation in Section III. The assumption that the parameters are constant limit the modeling power and application of the mathematical model. In the following Sections, we will show that using artificial intelligence can solve this problem, since we use these parameter values only as initial values. During the simulation, the parameters of our virtual patients change over time.

B. Structure of neural networks

We use neural networks to estimate the model parameters. The neural networks have a four-layer feed-forward structure with one input layer, two hidden layers and one output layer. The output layer contains linear neurons, therefore the outputs are the weighted sums of the outputs of the neurons in the previous layer and a bias [29]. The output of the r th neuron in the l th layer is calculated as

$$y_r^{(l)}(\mathbf{x}) = \varphi \left(\sum_{i=1}^{N_i^{(l)}} w_{r,i}^{(l)} x_i(p) + \underbrace{w_{r,N_i^{(l)}+1}^{(l)}}_{\text{bias}} \right), \quad (6)$$

where $r = 1, 2, \dots, R^{(l)}$ and $R^{(l)}$ is the number of neurons in the l th hidden layer, the superscript (l) refers to layer l , $\mathbf{x}$ is the input of the neural network, x_i is the i th input of the neural network, where $i = 1, 2, \dots, N_i$ and N_i is the number of inputs, $w_{r,i}^{(l)}$ is the weight of the i th input of the r th neuron in layer l , and $w_{r,N_i+1}^{(1)}$ is the weight of the bias of the r th

neuron. The activation function of the neurons in the hidden layer is a hyperbolic tangent-sigmoid function

$$\varphi(\epsilon) = \frac{2}{1 + e^{-2\epsilon}} - 1 \quad (7)$$

because this activation function provided the best results during tests. We tried besides this function the linear and the log-sigmoid activation functions as well.

The m th output of the neural network is

$$y_m^{(3)}(\mathbf{x}) = \sum_{r=1}^{R^{(2)}} w_{m,r}^{(3)} y_r^{(2)}(\mathbf{x}) + \underbrace{w_{m,R^{(2)}+1}^{(3)}}_{\text{offset}}, \quad (8)$$

where the superscript (3) refers to layer 3, $w_{m,r}^{(3)}$ is the weight of the m th neuron and r th input in the third layer, and $w_{m,R^{(2)}+1}^{(3)}$ is the weight of the bias for the m th neuron in the output layer.

The neural network is trained with supervised learning utilizing error backpropagation using a Levenberg-Marquardt algorithm [30]. The training datasets are generated with in silico experiments from a large parameter set. We use 50 and 20 neurons in the first and second hidden layers respectively. In the application in the upcoming Sections, we use neural networks with seven inputs and eight outputs. The inputs are the tumor volume measurements from Monday to Friday (with measurement value of zero if there was no measurement on that day), and the doses of injections on Tuesday and Friday (see Fig. 1). The outputs of the network are the estimated parameters.

III. EXPERIMENTAL SETUP AND TRAINING DATA

A. Experimental setups

Figure 1 shows the currently used sparse measurement protocols (experimental setup #1-#3). For comparison, the experimental setup #0 shows the protocol when measurements are taken daily (such as the training data of the neural networks presented in our previous works [27], [29]). In Figure 1, the days marked in the blue background indicate measurement days, and the green arrows indicate the day of injection. In experimental setup #0, tumor volume is measured every day of the week (except on weekends) and drug is injected into the patient on Tuesdays and Fridays. In a sparse measurement, the protocol is that the tumor volume is measured three times a week, preferably equidistantly distributed on working days, i.e., on Monday, Wednesday, and Friday.

The experimental setup (denoted by ES) #1 shows the currently used basic protocol, which contains a measurement on Monday and Wednesday, an injection on Tuesday, and both on Friday. There are cases when ES #1 is not possible. For example when, there is a holiday on Monday, thus there is no measurement taken on Monday, instead, we measure the tumor volume on Tuesday in addition to injection (i.e., we use the second protocol). Then, continuing the basic protocol (ES #1), we take a measurement on Wednesday and Friday, and also give an injection on Friday. The third case illustrates a

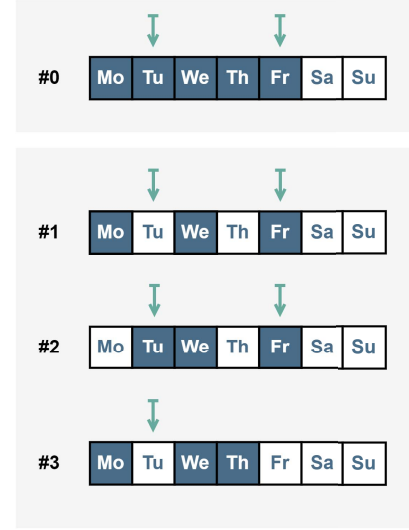


Fig. 1. The different experimental setups, where the days with the blue background indicate measurement days and the green arrows indicate the days with injection.

long weekend, in which case the beginning of the week is the same as the ES #1, but on Friday the injection is missed and not replaced, and the Friday measurement takes place on an earlier day, Thursday.

B. Training data generation

We create two types of neural networks based on the two different training datasets. First, the network is learning based on ES #0 (Fig. 1.), in this case the measurements of the tumor volumes are taken every day (except weekends). Let A denote this type of data set. The other network is trained by sparse measurement dataset. In this case the measurement protocols are randomly generated with different probabilities between ES #1, #2 and #3 (Fig. 1.). The probability of generating ES #1 is 90% and that of the ES #2 and #3 is 5-5%. So the ES #1 will be generated most often, which is the basic protocol in the current experimental procedure. This type of dataset is denoted by B.

A further aim is to examine changes in the parameters, so we evaluate how sensitive the networks are to parameter changes of different magnitudes. For this purpose, we created several smaller training datasets and their different versions.

The generation of virtual patients and the simulation of the measurements and injections that build up the training dataset were performed to mimic the real animal experiments as much as possible. The experiment lasts for 105 days (15 weeks) and can be halted in three cases. The first is when the experiment time window is exceeded, the second is when the tumor volume reaches a maximum value which is not allowed to exceeded according to the protocol, and the third is when the tumor volume reaches a minimum value where it is almost impossible to measure.

We generated 16-16 different versions of type A and B datasets. These differed in that we always changed the param-

eters of the virtual patients, and we made these changes more and more frequently. We changed the parameters at every 4, 3, 2 or 1 weeks. These different change scheduling have four further versions. In order to examine the effect of the maximum change rate of the parameters, we added a perturbation on the parameters in the interval $\pm 5\%$, $\pm 10\%$, $\pm 15\%$, and $\pm 20\%$. The Tables III. and IV. contain the change rate and scheduling of the parameters. These Tables are detailed in IV-A. So we created 32 different training datasets, therefore a total of 32 neural networks were trained.

Each dataset consisted of 80 variations for 100 different patients, 8000 patients in total. The training dataset consisted of approximately 120000 weeks of measurement data and their associated time-varying parameter values. The approximate value is not accurate because it is not possible to predict whether a virtual patient will survive 15 weeks or not.

The parameters of the virtual patients are based on the results of real animal experiments. We randomized these parameters in the ranges shown in Table I. This table shows the ranges for the initial values of the parameters. When the parameters of the virtual patients change during the generation of the training dataset, the values of the parameters may deviate from these ranges.

Parameter	Minimum value	Maximum value
a [day ⁻¹]	0.5811	0.6871
b [day ⁻¹]	1.0234	1.0261
n [day ⁻¹]	6.5696e-06	1.1555e-05
w [day ⁻¹]	0.1742	0.3229
ED_{50} [mg · kg ⁻¹]	0.1000	1.0000
c [day ⁻¹]	0.3425	0.9772
k_1 [day ⁻¹]	6.7697	7.7738
k_2 [day ⁻¹]	48.5147	374.6637

TABLE I

THE RANGES OF THE MODEL PARAMETERS FROM REAL MEASUREMENTS

The trained neural networks will estimate the parameters using measurements from the last week, thus they can be used to track the changes of the parameters, handle inpatient variability and parameter changes. We will create NNs based on the dataset A and B and their different versions, train them with the same versions of the A and B type, and compare their performance. The purpose of this is to examine the extent to which neural network performance is affected by changes in parameters or the absence of a few measurements during the week. The inputs of the NNs are measurements of the total tumor volume. For all training datasets, five tumor volume measurements per week are the neural networks inputs, but it is important that the missing data (in sparse datasets) is replaced by a value of zero. The initial conditions are different for all the patients. The living tumor volume at the time of the first treatment ($x_1(0)$) is a random number and the dead tumor volume at the same time is ($x_2(0)$) also a uniformly distributed random number from the ranges shown in Table II. The initial values for x_3 and x_4 are considered to be zero.

Parameter	$x_1(0)$ [mm ³]	$x_2(0)$ [mm ³]	$dose$ [mg · kg ⁻¹]
Minimum value	0	0	0.5
Maximum value	200	50	8

TABLE II

THE RANGES OF THE THE INITIAL CONDITIONS FOR THE LIVING ($x_1(0)$) AND DEAD ($x_2(0)$) TUMOR VOLUMES, AND THE POSSIBLE VALUES OF THE DRUG DOSE.

IV. TRAINING RESULTS

A. Comparison of neural networks learned from sparse and normal datasets

We created 32 neural networks (the training took less than 3807 epoch for all networks), 16 were trained from the current sparse protocols (versions of the dataset B), and a further 16 from daily measurements (versions of the dataset A). The second one is only for comparison purposes and will not be presented in details in this paper. Tables III and IV summarize the performance of neural networks. Table III contains performance data for neural networks learned from sparse datasets, and Table IV contains performance data for networks learned from normal datasets.

For both tables, the first column contains the ID of the network. Networks with an even ID were trained on a variation of dataset A, and those with an odd ID learned on a variation of dataset B. The second column shows the interval within which the parameters changed relative to themselves when generating the training data. The third column indicates the schedule for changing parameters in weeks. For example, if a value of four is shown in this part of the table, the parameter values changed at the end of every four weeks. So if a mouse survives 15 weeks, its parameters will go through a total of three changes. The fourth column shows the regression values of the networks, from which the performance of the networks can be deduced.

In our case, if the regression value of the network reaches one, the network is able to estimate the parameter values without error. In one of our previous papers, the best regression value with the networks trained on the normal dataset was 0.98, but it is important to point out that those time-window neural networks were trained with constant parameters for each variant of patients. The parameters already varied from individual to individual, but the variants differed only in treatment (injections) [29].

It is important for comparison that the rows of the two tables are related. For example, network #1 (trained on dataset A) and network #2 (trained on dataset B) are well comparable, because the parameter change rate and scheduling of change are always the same for networks in a corresponding row. Based on this, it can be concluded that sparse measurements only slightly deteriorated the performance of the networks, since if we examine the regression values in the rows, we see similar values in most cases. We can also observe the same when increasing the rate of change, although not in all cases. In most cases increasing amplitude of parameter change degrades network performance. This can also be observed in reducing the scheduling of changes, so if parameter values change more

and more frequently, it can also result in a small decrease in performance.

Based on these, we can conclude that increasing the rate and frequency of parameter changes during the simulation as much as possible can result in the lowest regression value, and thus the weakest performance. However, the lowest regression value in the table was reached by network #23, where the parameters changed weekly (this is the most frequent change), but the rate of change is a maximum of $\pm 15\%$. In comparison, network #31 produces better results with the same weekly schedule, but with parameter changes of up to $\pm 20\%$. In evidence from these results, the nonlinearities in the model can sometimes yield unexpected results and it is essential for the research to examine the parameters and to know their changes.

B. Estimation results for sparse dataset

During the training, 70% of the approximately 120000 training data was used for actual training, 15% for validation and 15% for test. Training datasets for all 32 neural networks were used with these settings.

We selected two neural networks to present their parameter estimation results in detail. Both networks were trained on the sparse dataset. The relative estimation errors are shown in Figs. 2-5. In these figures, the red horizontal curves show the median of the data, the black rectangles show the range where the 25%-75% of the data are (thus, $\pm 25\%$ range around the median), the whiskers show data limit that are not considered as outliers, while the crosses show the outliers.

Figure 2 shows the relative percentage errors in the estimation of the neural network #13. This neural network was trained from a training dataset, in which the parameters of virtual patients changed by $\pm 10\%$ compared to themselves at the end of every two week. The figure shows all the data, so also the ones on which the network learned and the remaining 15%, which are the test data, so they were not used during the training. A total of approximately 120000 data distributions per parameter can be viewed, depending on how many of the 8000 virtual patient survived 15 weeks. It can be seen that the median of the errors for each parameter is close to

ID	Change rate	Change scheduling	R
#1	$\pm 5\%$	4 weeks	0.7479
#3	$\pm 5\%$	3 weeks	0.7170
#5	$\pm 5\%$	2 weeks	0.7105
#7	$\pm 5\%$	1 week	0.6600
#9	$\pm 10\%$	4 weeks	0.7163
#11	$\pm 10\%$	3 weeks	0.6934
#13	$\pm 10\%$	2 weeks	0.7423
#15	$\pm 10\%$	1 week	0.6570
#17	$\pm 15\%$	4 weeks	0.7533
#19	$\pm 15\%$	3 weeks	0.6394
#21	$\pm 15\%$	2 weeks	0.6350
#23	$\pm 15\%$	1 week	0.4585
#25	$\pm 20\%$	4 weeks	0.6953
#27	$\pm 20\%$	3 weeks	0.7204
#29	$\pm 20\%$	2 weeks	0.6009
#31	$\pm 20\%$	1 week	0.6641

Mean: 0,6757

TABLE III

PERFORMANCE OF NEURAL NETWORKS TRAINED ON SPARSE DATASETS.

ID	Change rate	Change scheduling	R
#2	$\pm 5\%$	4 weeks	0.7802
#4	$\pm 5\%$	3 weeks	0.7792
#6	$\pm 5\%$	2 weeks	0.7326
#8	$\pm 5\%$	1 week	0.7572
#10	$\pm 10\%$	4 weeks	0.7325
#12	$\pm 10\%$	3 weeks	0.7721
#14	$\pm 10\%$	2 weeks	0.6782
#16	$\pm 10\%$	1 week	0.6987
#18	$\pm 15\%$	4 weeks	0.7025
#20	$\pm 15\%$	3 weeks	0.7274
#22	$\pm 15\%$	2 weeks	0.7117
#24	$\pm 15\%$	1 week	0.6419
#26	$\pm 20\%$	4 weeks	0.7289
#28	$\pm 20\%$	3 weeks	0.7275
#30	$\pm 20\%$	2 weeks	0.6762
#32	$\pm 20\%$	1 week	0.7589

Mean: 0,7253

TABLE IV

PERFORMANCE OF NEURAL NETWORKS TRAINED ON NORMAL DATASETS.

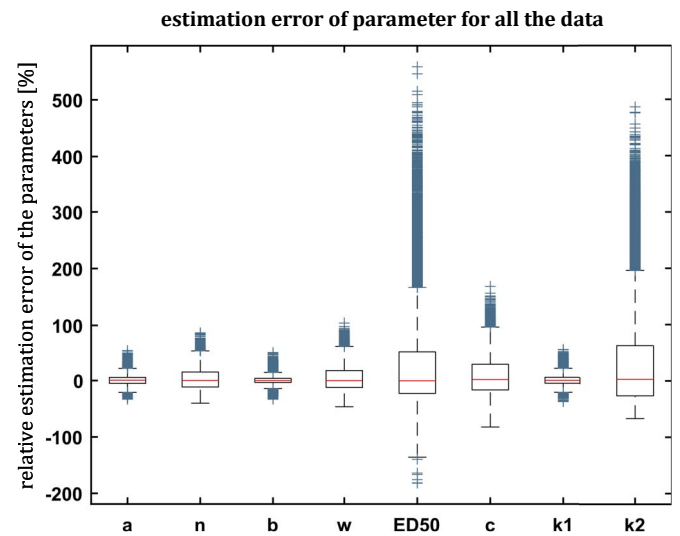


Fig. 2. The relative estimation error of the NN #13 for the parameters on all the data.

zero percent. However, there are two parameters that quite extend the scale, these are the ED_{50} and k_2 parameter. An examination of an earlier version of the system model also revealed that the output of the model is less sensitive to changes in the ED_{50} parameter [20]. This may be one of the reasons why this parameter is difficult to estimate.

For ease of reference, we have created another diagram to examine errors with an accuracy of $\pm 50\%$, the result of this is shown in Figure 3. The percentages over the parameters on the top of the figure show how many of the total data from the dataset falls into this range. The higher percentage rates indicate better estimation. The results show that the parameters can be estimated within the 50% range for most of the data. The percentages are almost 100% for most parameters (except for parameter c , where the result is almost 90%). The two parameters (ED_{50} and k_2) have lower percentages but still managed to give a good estimate of their initial value with NN on the training data.

The other neural network we selected is the neural network

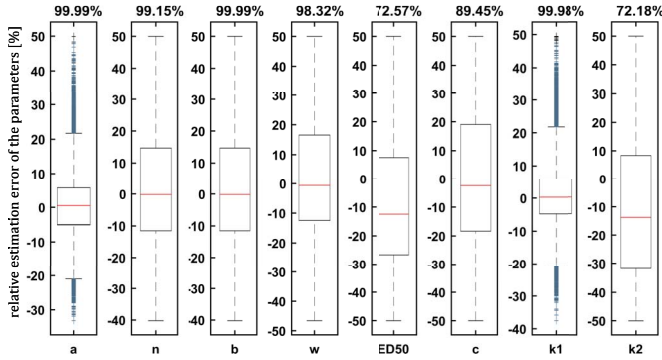


Fig. 3. The relative estimation error of the NN #13 for the parameters in the error range of $\pm 50\%$, indicating the percentage of the samples in this range (on top of the figure for each parameter).

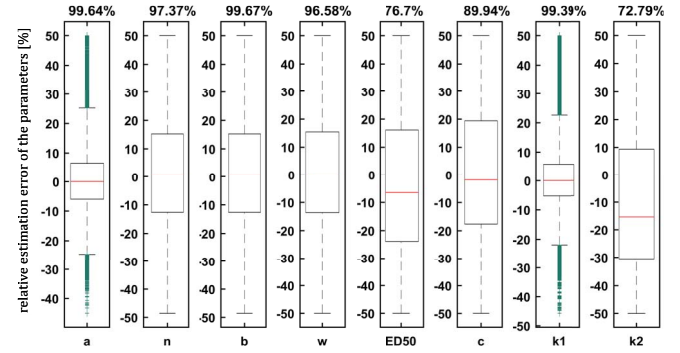


Fig. 5. The relative estimation error of the NN #27 in the error range of $\pm 50\%$, indicating the percentage of the samples in this range (on top of the figure for each parameter).

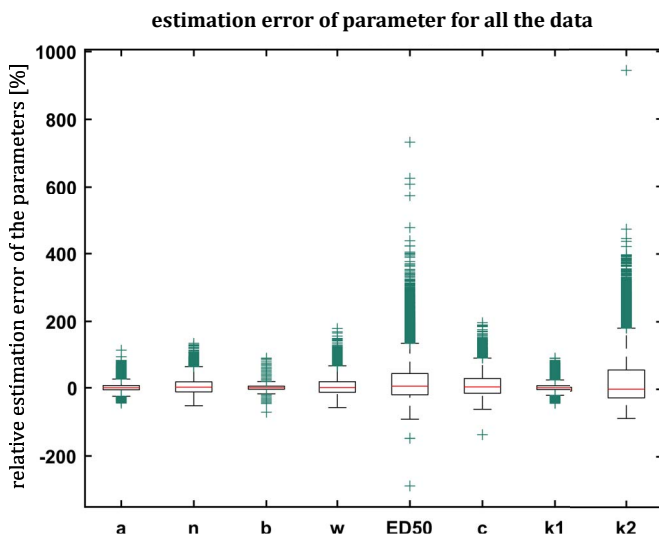


Fig. 4. The relative estimation error of the NN #27 for parameters on all of the data

#27. Figure 4 shows the performance of this network on the training dataset. This neural network we trained from a training dataset, in which the parameters of virtual patients changed by $\pm 20\%$ compared to themselves at the end of every three weeks. Similar to the results of neural network #13, ED_{50} and k_2 have the most protruding outliers. If we compare the two networks, we can see that in the case of network #27 the errors move on a larger scale. This is due to the fact that this network learned on a training dataset with a larger change in parameters.

We also examined the performance of this network in the 50% error band. The percentage values show that the network could also learn the parameters efficiently. From the statistical figure (Figure 5.) we can see that the medians are close to zero except for the parameters ED_{50} and k_2 . The percentage of these parameters is still the lowest. The large parameter change is also seen in the percentages, as which value fell closer to 100% for the NN #13, here to 99%, but this decrease is negligible.

V. CONCLUSION

With neural networks of current complexity, we found that sparsity of measurements had only a small effect on the performance of the networks. Examining the networks trained on sparse and normal datasets, the regression values differ only slightly. However, it is clear that due to time-varying parameters, the complexity of the task has increased compared to the past, as the performance of each of the current networks (based on regression values) has deteriorated compared to time-window networks trained with constant parameter values used in previous works. The problem was further complicated by the fact that these neural networks had already learned from another set of parameters. The range of parameters has changed compared to the previous ones based on the new results and fits. Simulating a weekly or multi-weekly change always involves a change in the characteristic curve that networks must always handle in addition to sparse data. In future works, we will examine how the increased complexity of networks affects their performance.

The system can be used to determine how often and to what extent each parameter changes their value, in order for networks to be able to provide good performance. This may be the basis for tracking changes in parameters in the future for real patients, thus providing an opportunity to generate individual therapy.

REFERENCES

- [1] J. M. Greene, J. L. Gevertz, and E. D. Sontag, "Mathematical approach to differentiate spontaneous and induced evolution to drug resistance during cancer treatment," *JCO Clinical Cancer Informatics*, no. 3, pp. 1–20, 2019.
- [2] D. A. Drexler, T. Ferenci, A. Lovrics, and L. Kovács, "Tumor dynamics modeling based on formal reaction kinetics," *Acta Polytechnica Hungarica*, vol. 16, pp. 31–44, 2019.
- [3] A. R. Akhmetzhanov, J. W. Kim, R. Sullivan, R. A. Beckman, P. Tamayo, and C.-H. Yeang, "Modelling bistable tumour population dynamics to design effective treatment strategies," *Journal of theoretical biology*, vol. 474, pp. 88–102, 2019.
- [4] F. Cacace, V. Cusimano, and P. Palumbo, "Optimal impulsive control with application to antiangiogenic tumor therapy," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 1, pp. 106–117, Jan 2020.

- [5] J. Klamka, H. Maurer, and A. Swierniak, "Local controllability and optimal control for a model of combined anticancer therapy with control delays," *Mathematical Biosciences and Engineering*, vol. 14, no. 1, pp. 195–216, 2017.
- [6] F. S. Lobato, V. S. Machado, and V. Steffen, "Determination of an optimal control strategy for drug administration in tumor treatment using multi-objective optimization differential evolution," *Computer Methods and Programs in Biomedicine*, vol. 131, pp. 51 – 61, 2016.
- [7] D. A. Drexler, J. Sápi, and L. Kovács, " H_∞ control of nonlinear systems with positive input with application to antiangiogenic therapy," in *Proceedings of the 9th IFAC Symposium on Robust Control Design ROCOND 2018*, 2018, pp. 146 – 151.
- [8] J. Sápi, D. A. Drexler, and L. Kovács, "Potential benefits of discrete-time controller-based treatments over protocol-based cancer therapies," *Acta Polytechnica Hungarica*, vol. 14, no. 1, pp. 11–23, 2017.
- [9] L. Kovács and G. Eigner, "Tensor product model transformation based parallel distributed control of tumor growth," *Acta Polytechnica Hungarica*, vol. 15, no. 3, pp. 101–123, 2018.
- [10] H.-P. Ren, Y. Yang, M. S. Baptista, and C. Grebogi, "Tumour chemotherapy strategy based on impulse control theory," *Philosophical Transactions Mathematical Physical & Engineering Sciences*, vol. 375, no. 2088, 2017.
- [11] D. A. Drexler and L. Kovács, "Optimization of impulsive discrete-time tumor chemotherapy," in *Proceedings of the 2019 IEEE 1st International Conference on Societal Automation.*, 2019.
- [12] B. Péceli, D. A. Drexler, and L. Kovács, "Optimal scheduling of low-dose metronomic chemotherapy: an in-silico analysis," in *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*, 2020, pp. 411–416.
- [13] N. A. Perestyuk, V. A. Plotnikov, A. M. Samoilenko, and N. V. Skripnik, *Differential Equations with Impulse Effects*. Berlin, Germany: De Gruyter, 2011.
- [14] T. Yang, *Impulsive Control Theory*. Berlin, Germany: Springer, 2001.
- [15] A. M. Jarrett, E. A. B. F. Lima, D. A. Hormuth, M. T. McKenna, X. Feng, D. A. Ekrut, A. C. M. Resende, A. Brock, and T. E. Yankeelov, "Mathematical models of tumor cell proliferation: A review of the literature," *Expert Review of Anticancer Therapy*, vol. 18, no. 12, pp. 1271–1286, 2018.
- [16] P. M. Altrock, L. L. Liu, and F. Michor, "The mathematics of cancer: integrating quantitative models," *Nature Reviews. Cancer*, vol. 15, no. 12, pp. 730–745, 2015.
- [17] P. Hahnfeldt, D. Panigrahy, J. Folkman, and L. Hlatky, "Tumor development under angiogenic signaling: A dynamical theory of tumor growth, treatment response, and postvascular dormancy," *Cancer Research*, vol. 59, pp. 4770–4775, 1999.
- [18] P. A. Valle, L. N. Coria, and Y. Salazar, "Tumor Clearance Analysis on a Cancer Chemo-Immunotherapy Mathematical Model," *Bulletin of Mathematical Biology*, vol. 81, no. 10, pp. 4144–4173, Oct. 2019, wOS:000487900700018.
- [19] D. A. Drexler, T. Ferenci, A. Füredi, G. Szakács, and L. Kovács, "Experimental data-driven tumor modeling for chemotherapy," in *Proceedings of the 21st IFAC World Congress*, 2020, pp. 16 466–16 471.
- [20] M. Siket, G. Eigner, and L. Kovács, "Sensitivity and identifiability analysis of a third-order tumor growth model," in *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*, 2020, pp. 417–422.
- [21] N. Lutsiv, T. Maksymyuk, M. Beshley, O. Lavriv, V. Andrushchak, A. Sachenko, L. Vokorokos, and J. Gazda, "Deep semisupervised learning-based network anomaly detection in heterogeneous information systems," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 413–431, 2022.
- [22] M. Dobeš, R. Andoga, and L. Főző, "Sensory integration in deep neural networks," *Acta Polytechnica Hungarica*, vol. 18, no. 3, pp. 245–254, 2021.
- [23] U. Farooq, J. Gu, V. E. Balas, G. Abbas, M. U. Asad, and M. M. Balas, "A hybrid time series forecasting model for disturbance storm time index using a competitive brain emotional neural network and neo-fuzzy neurons," *Acta Polytechnica Hungarica*, vol. 16, no. 4, pp. 213–229, 2019.
- [24] M. Fidler, Y. Xiong, R. Schoemaker, J. Wilkins, M. Trame, T. Post, and W. Wang, *nlmixr: Nonlinear Mixed Effects Models in Population Pharmacokinetics and Pharmacodynamics*, 2018, r package version 1.0.0-7. [Online]. Available: <https://CRAN.R-project.org/package=nlmixr>
- [25] J. Owen and J. Fiedler-Kelly, *Introduction to Population Pharmacokinetic / Pharmacodynamic Analysis with Nonlinear Mixed Effects Models*. Wiley, 2014.
- [26] A. Füredi, K. Szebényi, S. Tóth, M. Cserepes, L. Hámori, V. Nagy, E. Karai, P. Vajdovich, T. Imre, P. Szabó, D. Szüts, J. Tóvári, and G. Szakács, "Pegylated liposomal formulation of doxorubicin overcomes drug resistance in a genetically engineered mouse model of breast cancer," *Journal of Controlled Release*, vol. 261, pp. 287–296, 2017.
- [27] E. Nagy, M. Puskás, and D. A. Drexler, "Comparison of artificial neural network and anfis for parameter estimation of a tumor model," in *IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics SAMI (2022)*, 2022, pp. 133–139.
- [28] M. Puskás and D. A. Drexler, "Parameter identification of a tumor model using artificial neural networks," in *Proceedings of the 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics*, 2021, pp. 443–448.
- [29] —, "Tumor model parameter estimation for therapyoptimization using artificial neural networks," in *IEEE International Conference on Systems, Man, and Cybernetics - 2021*, 2021.
- [30] H. Yu and B. Wilamowski, *Levenberg–Marquardt Training*, 02 2011.

