



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

AUTOENKÓDER FEJLESZTÉSE TUMORMODELL PARAMÉTEREINEK PREDIKCIÓJÁHOZ

Szerző(k):

Kisbenedek Lilla

mérnökinformatikus MSc. szak, II. évf.

Konzulens(ek):

Dr. Drexler Dániel András

egyetemi docens

Puskás Melánia

doktorandusz hallgató

Budapest, 2024. április 18.

Nomenklatúra

a	: Tumor növekedési ráta
b	: Inhibíciós ráta
c	: Módosított klírensz
$d_{i,j}$	: Az i -edik virtuális pácienshez tartozó dózis a j -edik napon
ED_{50}	: Medián effektív dózis
h_t	: A NODE architektúra esetén a t -edik lépésben a hálózat állapota
k_1	: A gyógyszer áramlási sebességi együtthatója a centrális kompartmentből a perifériás kompartmentbe
k_2	: A gyógyszer áramlási sebességi együtthatója a perifériás kompartmentből a centrális kompartmentbe
n	: Nekrotikus ráta
N	: A tanító adatok száma, a dolgozatban $N = 20000$
p	: A tumormodell összes paraméterét tartalmazó halmaz
$p^{\max}$	: A paraméterek felső határa
$p^{\min}$	: A paraméterek alsó határa
p'	: A paraméterek értéke normalizálást követően
t_k	: A k -edik injekciózás időpontja
u	: Beadott kemoterápiás dózisokat tartalmazó vektor
w	: Kimosódási ráta
W	: Az enkóder neurális hálózatához tartozó súlyokat tartalmazó mátrix
x_1	: Az élő tumortérfogat időfüggvénye [mm^3]
x_2	: A halott tumortérfogat időfüggvénye [mm^3]

- x_3 : A gyógyszer szint időfüggvénye a vérben [mg/kg]
 x_4 : A gyógyszer szint időfüggvénye a szövetekben [mg/kg]
 X_i : Az i -edik bemeneti érték az autoenkóderben
 $\hat{X}_i$: Az X_i bemenetre adott kimeneti predikció (rekonstrukció) az autoenkóderben
 y : A teljes tumortérfogat [mm³]
 Y : Az L hosszúságú teljes tumortérfogatokat tartalmazó vektor
 $\hat{Y}$: Az L hosszúságú becsült teljes tumortérfogatokat tartalmazó
 z_i : Az autoenkóder enkóder részének i -edik
 bemenetre adott kimenete (tömörített reprezentáció)
 η : A hálózat tanulási sebessége (learning rate)
 θ : A neurális hálózat összes paramétereit tartalmazó halmaz
 σ : A szigmoid függvény
 ω : A PINN költségfüggvényében alkalmazott súly

Tartalomjegyzék

Nomenklatura	2
1. Bevezetés	5
2. Irodalmi háttér	8
2.1. A tumor dinamikáját leíró matematikai modell	8
2.2. Paraméterbecslési probléma és megoldási lehetőségei	11
2.3. Neurális hálózatok és autoenkóderek	15
3. Megvalósítás	18
3.1. Paraméterbecslő rendszer felépítése	18
3.2. Tanító adatok generálása	19
3.3. A hálózat előtanítása	21
3.4. Az autoenkóder tanítása	24
4. Eredmények	29
4.1. A rendszer tanításának kiértékelése	29
5. Konklúzió	37
Irodalomjegyzék	39
Ábrák jegyzéke	44

1. fejezet

Bevezetés

A daganatos megbetegedések a vezető halálokok közé tartoznak, továbbá az egyik leginkább meghatározó tényező, amely lassítja a várható élettartam növekedését [1]. Bár egyes fejlett országokban a rákos megbetegedések halálozási rátája az elmúlt évtizedekben csökkent [2], a betegek kezelése és ellátása a közegészségügyben hatalmas gazdasági erőforrásokat emészt fel és a következő 40 évre pedig növekvő tendenciát jósolnak az esetszámokat tekintve [3].

A rákos megbetegedéseket a sejtek szabályozatlan növekedése és a sejtek terjedése a származási helyükről más testrészekre jellemzi. A rák különböző betegségek összefoglaló neve, amelyek különböző tünetekkel társulnak és eltérő kezelési módszereket igényelnek. Évtizedeken át csak néhány lehetőség állt rendelkezésre a rák kezelésére: a műtét, a sugárterápia és a kemoterápia, amelyeket önállóan vagy kombinációban alkalmaztak [4]. A sugárterápia során a cél az, hogy az ionizáló sugárzásnak kitett daganatsejtek DNS-ét károsítsák, így megakadályozzák azok szaporodását vagy elpusztítsák azokat. A hagyományos kemoterápia célja szintén a gyorsan növekvő és osztódó rákos sejtek elpusztítása, különböző kemoterápiás gyógyszerek alkalmazásával. Utóbbi hátránya, hogy az alkalmazott gyógyszerek nem szelektívek, és az egészséges, normál szöveteket is károsíthatják, súlyos, nemkívánatos mellékhatásokat, például étvágytalanságot, hányingert és hajhullást okozva [5].

Az olyan tulajdonságok, mint a rák típusa, helye és súlyossága befolyásolják a kezelési mód kiválasztását. A sebészeti eltávolítás sugárterápiával kombinálva az elsődleges módszer az áttét nélküli elsődleges daganatok esetében. Mindazonáltal, számos erősen agresszív daganat esetén ezeknek a módszereknek a korlátai nyilvánvalóak a betegeknél. A leggyakrabban ajánlott terápia a

műtéti beavatkozást követő sugár- és kemoterápia. Bár a két módszer súlyos mellékhatásokat okoz, meglehetősen rontva az életminőséget, alkalmazásuk nagy mértékben csökkenti a morbiditást és a mortalitást [6].

A rákos megbetegedések kezelése során a gyógyszer-rezisztencia és a gyógyszerhordozó rendszerek kifejezetlensége jelentik az egyik legnagyobb problémát a rákgyógyításban [6]. Az utóbbi évtizedekben nagy mértékben fejlődtek a rák kezelésére szolgáló különböző modalitások. A hagyományos módszerek mellett nagy hangsúlyt kaptak az alternatív kezelési módszerek is, mint az őssejtterápia, a célzott gyógyszerek, a hormonterápia és az immunterápia. Összességében a rákelenes gyógyszer kutatások a sokkal pontosabb és kevesebb gyógyszer alkalmazása irányába fordultak.

A gyógyszerkutatáson felül a gyógyszer adagolásának szabályozása is megélénkült az utóbbi évtizedekben. Az új gyógyszeradagoló rendszerek fejlesztésének fő célja, hogy lehetővé tegyék a tartós és ellenőrzött gyógyszerleadást, a hatékony gyógyszer szintet és ezzel egyidejűleg csökkentsék a káros hatásokat [7]. A kemoterápia során beadott gyógyszerek meghatározása jelenleg egy meghatározott protokoll alapján történik, mely a betegek testsúlyát veszi figyelembe. Az adagolás során a maximálisan tolerálható dózist alkalmazzák (MTD - Maximum Tolerated Dosage), ahol pár hetes időközönként nagy dózisú kemoterápiás szert adnak be a betegeknek, melynek hátránya a hosszabb kezelésmentes időszakok alatt a rezisztencia-mechanizmusok megjelenése, továbbá a mellékhatások súlyosbodása.

A gyógyszerhordozó rendszerek létrehozásának alapja egy robusztus algoritmus, mely képes a páciens mért tulajdonságai alapján meghatározni a szükséges gyógyszer mennyiségét. A dózisok megállapítása egy matematikai modellen alapuló algoritmus alapján történik, melynek a paramétereinek a meghatározása elengedhetetlen a folyamat során. A munkám során egy korábban megalkotott tumormodellt használok fel, mely egy négy állapotváltozót tartalmazó közönséges differenciálegyenlet-rendszer. A rendszer bemenete a beadott dózis, míg a kimenete a teljes tumortérfogat. Utóbbi az általunk mérhető változó, mely leírja a tumor reakcióját a beadott gyógyszer mennyiségekre. Az egyenletek nyolc paramétert tartalmaznak, melyeket a szimulált kezelések teljes időtartama alatt konstansnak tekintettem. Ezeknek a paramétereknek az identifikált értéke alapján történik a személyreszabott, optimális terápia megalkotása. A paraméter identifikáció gyakran hosszas és bizonytalanságokat tartalmazó folyamat, melynek a meggyorsítására egy ne-

urális hálózat alkalmas lehet. Mindazonáltal, figyelembe kell venni, hogy egy neurális hálózatok betanításához nagy mennyiségű adat szükséges mind a modellezett, mind pedig a mért tumortérfor-gatokból. Munkám során beépítettem a tumormodellt a hálózat architektúrájába, ezáltal a tanítási folyamat felgyorsítható, a pontosság számszerűsíthető és növelhető lett.

A dolgozatom elején ismertetem a felhasznált tumordinamikai modellt (2.1. alfejezet). Mivel a modell kimenete alapján történik az egyenletekben szereplő paraméterek meghatározása, a 2.2. alfejezetben annak identifikációjának lehetőségeit részletezem. Ezt követően a 3. fejezetben rész-letezem a a paraméterek becslésére létrehozott neurális hálózat felépítésének és implementációjá-nak részleteit. Végül, pedig a 4. fejezetben a tanítás során kapott eredményeket ismertetem több különböző hosszúságú előre jelzett mérési intervallumra. A létrehozott rendszer alkalmas lehet a paraméterek gyors meghatározására, illetve kezdeti értékük becslésére.

A kezelések személyre szabása során lehetséges akár egy olyan gyógyszerhordozó rendszer megalkotása, mely a betegek egyedi jellemzői alapján állapítja meg a kemoterápiás szerek beadott mennyiségét és annak időpontját, azáltal hogy információt kap a daganat adott pillanatbeli állapo-táról. Munkámmal az algoritmus paraméterbecslő szakaszának a létrehozásához járultam hozzá.

2. fejezet

Irodalmi háttér

2.1. A tumor dinamikáját leíró matematikai modell

Egy kísérletekkel validált modell segítségével előrejelezhető, hogy egy adott kezelési protokoll milyen hatással lesz a tumor dinamikájára. Ezáltal a kezelések hatékonysága növelhető, a mellékhatások csökkenthetők és a kezelési költségek optimalizálhatók. A szakirodalomban számos modell található a tumor viselkedésének leírására [8, 9, 10]. Ezek a megalkotott modellek vagy csak az időben, vagy időben és térben próbálják meg leírni közönséges és parciális differenciálegyenletek által a tumor dinamikáját.

A dolgozat során egy korábban létrehozott tumormodellt használók fel a tumortérfogatok szimulációjára [11]. A tumormodell egyenletei reakciókinetikai analógia alapján lettek létrehozva, ahol X_1 , X_2 , X_3 , és X_4 fiktív tagok. Az X_1 az élő tumor térfogata, az X_2 az elhalt tumor térfogata, az X_3 a gyógyszer szint a centrális kompartmentben, míg az X_4 a gyógyszer szint a perifériás kompartmentben. A (2.1) egyenlet a tumorsejtek osztódását, míg (2.2) az elhalását írja le. A (2.3) egyenlet a tumorsejtek kimosódását modellezi. A (2.4) egyenlet a gyógyszer hatását, (2.5) a gyógyszer kiürülését, míg (2.6) a két kompartment közötti reverzibilis gyógyszeráramlást írja le.



A (2.1)-(2.6) fiktív reakciókból felírható egy differenciálegyenlet-rendszer:

$$\dot{x}_1 = (a - n)x_1 - b \frac{x_1 x_3}{ED_{50} + x_3}, \quad (2.7)$$

$$\dot{x}_2 = nx_1 + b \frac{x_1 x_3}{ED_{50} + x_3} - wx_2, \quad (2.8)$$

$$\dot{x}_3 = -(c + k_1)x_3 + k_2 x_4 + u, \quad (2.9)$$

$$\dot{x}_4 = k_1 x_3 - k_2 x_4, \quad (2.10)$$

ahol x_1 az élő tumortérfogat időfüggvénye [mm^3], x_2 az elhalt tumortérfogat időfüggvénye [mm^3], x_3 és x_4 a gyógyszer szint időfüggvénye a centrális kompartmentben [mg/kg] és a perifériás kompartmentben [mg/kg]. A modellt daganatos egereken végzett mérésekkel igazolták [12]. A létrehozott modell célja, hogy leírja a pegilált liposzómális doxorubicin (PLD) kemoterápiás szer tumorellenes hatását és a tumor dinamikáját. A (2.7)-(2.8) egyenletek a tumor dinamikáját és a gyógyszer hatását (farmakodinamikát), míg (2.9)-(2.10) a gyógyszer áramlását (farmakokinetikát) írják le.

Az egyenletekben szereplő paraméterek a reakciókinetikai együtthatóknak tekinthetők a formális felírás alapján. Az együtthatók közül a , n , w esetünkben a proliferációs ráta, a nekrotikus ráta valamint a kimosódási ráta, míg b , ED_{50} a gyógyszert jellemző együtthatók (a gyógyszer maximális hatását leíró paraméter és a medián effektív dózis). A c , k_1 , k_2 paraméterek a gyógyszer kimosódását, valamint a centrális és a perifériás kompartment közötti áramlását írják le. A 3. fejezetben a felhasznált paramétereknek a korábbi állatkísérletek során meghatározott határait vettem figye-

lembe. A 2.1. táblázatban foglaltam össze a paraméterek jelölését, mértékegységét és a korábban meghatározott értékeinek a felső és alsó határát.

Korábbi munkák során megállapításra került, hogy a kemoterápiás szer csak akkor van hatással a tumor térfogatára, ha az

$$0 > a - b - n \quad (2.11)$$

egyenlőtlenség teljesül [13]. Ha az összefüggés nem teljesül, az a daganat gyógyszerre való rezisztenciáját jelenti.

A rendszer bemente az u [mg/kg/nap], ami az időegység alatt bevitt gyógyszer mennyisége, melyre igaz, hogy $u \geq 0$, tehát nem vehet fel negatív értéket, mely adódik abból, hogy gyógyszert nem vonhatunk el a szervezetből. Ez a felírás folytonos bemenetet feltételez. A gyakorlatban azonban a bemenetek injekciók, azaz az egyes beadott gyógyszer dózisokat impulzív hatásnak tekintjük a vérben (x_3). Tehát, $t_0, \dots, t_{K-1}$ időpillanatban beadott K számú injekció esetén, mivel $t_k \geq 0$ és $k = 0, 1, 2, \dots, K - 1$, valamint $t_0 < t_1 < \dots < t_{K-1}$, így x_3 -ban folytonossági szakadás van a t_k időpontban, mely az alábbi összefüggéssel írható fel:

$$x_3(t_k^+) = x_3(t_k^-) + u_k, \quad (2.12)$$

ahol u_k a t_k időpontban beadott gyógyszer dózisa mg/kg-ban. A rendszer kimenete – az általunk mérhető változó – az élő és a halott tumor térfogatok összege:

$$y = x_1 + x_2. \quad (2.13)$$

Az x_1 és a x_2 tumortérfogat nem határozható meg külön, a teljes tumortérfogat mérése történik az egereken. A teljes tumortérfogat (y) mérése tolmérővel történik két ponton, a tumor szélességét és hosszúságát mérik, melyből egy korábban meghatározott közelítő képlet alapján történik a térfogatának becslése [14]. A mérések tartalmazhatnak mérési zajt, melynek modellezése egy különálló munka a kutatás részeként [15].

Paraméter	Mértékegység	Minimum érték	Maximum érték
a	nap ⁻¹	0,5811	0,6871
b	nap ⁻¹	1,0234	1,0261
n	nap ⁻¹	$6,5696 \cdot 10^{-6}$	$1,1555 \cdot 10^{-5}$
w	nap ⁻¹	0,1742	0,3229
ED ₅₀	mg · kg ⁻¹	0,1	1,0
c	nap ⁻¹	0,3425	0,9772
k ₁	nap ⁻¹	6,7697	7,778
k ₂	nap ⁻¹	48,51	474,66

2.1. táblázat. A korábban meghatározott paraméterek minimum és maximum értékei. Az a a tumor proliferációs rátája, a b a gyógyszer maximális hatása, az n a tumor nekrotikus rátája, a w a tumor kimosódási rátája, ED₅₀ a medián effektív dózis, c a gyógyszer kimosódási rátája, míg k₁ az áramlási együttható a vérből a szövetekbe, míg k₂ a szövetekből a vérbe.

2.2. Paraméterbecslési probléma és megoldási lehetőségei

A differenciálegyenletek esetében gyakran nem áll rendelkezésre az analitikus megoldás, így azok megoldása és a paramétereik meghatározása hosszadalmas folyamatot igényelhet. Egy adott differenciálegyenlet vagy rendszer megoldásának keresését gyakran direkt problémának nevezzük, míg az inverz probléma alatt olyan megfelelő modell meghatározását értjük, amelynek megoldása jól egyezik néhány kísérleti vagy valós megfigyeléssel. Tehát a paraméterbecslés egy inverz problémára keres megoldást, célja a megfigyeléseket tartalmazó adathalmazhoz legjobban illeszkedő modell ismeretlen paramétereinek meghatározása [16]. Míg a direkt problémának egyedi megoldása van, az inverz problémának több is lehet. Ebből adódóan az inverz problémában minden rendelkezésre álló információt figyelembe kell venni a meghatározandó paraméterekről. A dolgozatban a cél a 2.1. alfejezetben részletezett közönséges differenciálegyenleteket tartalmazó tumormodell paramétereinek meghatározása.

A paraméterbecslés folyamatában egy költségfüggvény optimalizálására törekszünk, amelynek keretében a leggyakrabban alkalmazott módszerek a Bayes-i becslés, a maximum likelihood módszer és a legkisebb négyzetek módszere [17]. A Bayes-i becslés a Bayes-tételt használja a valószínűségi modellek paramétereinek becslésére, ezáltal kifejezi a bizonytalanságot is a paraméterbecslés

körül. A maximum likelihood (ML) módszer a paramétereknek azt a legvalószínűbb értékét adja meg, ami a legnagyobb valószínűséggel következik be. Az ML módszer nem ad közvetlen becslést a paraméterek bizonytalanságáról és eloszlásáról. A legkisebb négyzetek módszere pedig a mért adatpontok és a modell predikciói között minimalizálja a négyzetes eltérést. A felsorolt paraméterbecslési módszerek által általában olyan nemlineáris optimalizálási problémát kapunk, amelyet zárt formában nem lehet megoldani, így iteratív numerikus optimalizálókat alkalmazunk a megoldás keresésére [18]. A felsorolt algoritmusok gyakran kiindulási érték megadását igénylik, melyből az iteráció elindítható. Korábbi munkákban a bemutatott tumor modell paramétereinek meghatározására többek között SAEM algoritmust alkalmaztak maximum-likelihood becslésre [19].

Az optimalizálásra alkalmazott algoritmusokat megkülönböztethetjük aszerint, hogy lokális vagy globális minimumok megkeresésére alkalmasak. Ez a megkülönböztetés csak a nemlineáris optimalizálás kontextusában szükséges, mivel a lineáris problémáknak mindig egyedi optimuma van. A lokális optimalizálók egy nagy csoportját alkotják a gradiens alapú módszerek, mint a Newton-módszerek, a kvázi-Newton módszerek és a konjugált gradiens módszer, melyek a paraméterek elsőrendű deriváltakon alapuló frissítésével keresik az optimális értékeket. Léteznek algoritmusok melyen a másodrendű deriváltak értékét használják fel, mint a Broyden-Fletcher-Goldfarb-Shanno (BFGS) módszer, Gauss-Newton módszer és a Levenberg–Marquardt algoritmus. A globális optimalizálók között a legelterjedtebbek a populáció alapú algoritmusok, mint az evolúción alapulóak (például genetikusan algoritmus) vagy részecske-raj optimalizálók.

A paraméterek meghatározása során fontos terület az identifikálhatósági vizsgálata a paramétereknek. Az azonosíthatósági elemzése felméri, hogy elméletileg meghatározhatók-e a modell paraméterei az adatokból, megvizsgálva a méréseket, a modell szerkezetét, valamint az előforduló hibák tulajdonságait. Egy modell strukturálisan azonosítható, ha létezik egyedi paraméterezés bármely adott modell kimenetéhez. Ebből adódik, hogy egy paraméter szerkezetileg nem azonosítható, ha a paraméter megváltoztatása nem feltétlenül változtatja meg a modell kimenetét, mert a változások teljes mértékben kompenzálhatók más paraméterek megváltoztatásával. A biológiai rendszerek modellezésekor különösen fontos az identifikálhatóság vizsgálata, mert a kísérleti adatok korlátozott mennyisége és minősége csak részben megfigyelt rendszerekben gyakran rossz modellekhez és paraméterekhez vezet a modellezési folyamat során [20]. Korábbi munkákban identifikálhatósági és érzékenységi vizsgálatot végeztek az általam is alkalmazott tumormodellen, mely során megál-

lapították, hogy az azonosítandó paraméterek és a többi névleges populációs érték kiválasztásával a rendszer azonosítható. Továbbá megállapították, hogy a paraméterek érzékenységi sorrendje az alábbi: a , b , n és w , ahol magasan a tumor proliferációs rátája a legérzékenyebb a tumormodell kimenetére [21].

Az irodalomban az utóbbi években számos megközelítés jelent meg a különböző differenciálegyenletek és neurális hálózatok integrációjával kapcsolatban. Ezek a hálózatok a fizikai, biológiai vagy kémiai törvényeket – jellemzően differenciálegyenletek formájában – közvetlenül a mély tanulási modellek architektúrájába illesztik be, ezáltal irányítják a tanulási folyamatot az egyenletek betartására [22, 23, 24]. A legnépszerűbb megközelítés az utóbbi években az ún. fizikával informált neurális hálózatok (PINN - Physics Informed Neural Network). A PINN hálózatokat először kezdeti érték problémák pontosabb megoldásának keresésére használták a neurális hálózatok által nyújtott függvény approximáció segítségével. Mindazonáltal inverz problémákra is alkalmazható, azáltal, hogy a költségfüggvénybe bele vesszük nem csak a differenciálegyenletek reziduumaikat tartalmazó tagot (a differenciálegyenletek betartásáért felelős tag), hanem az illeszkedésért felelős tagot is (az előre jelzett kimenetek és az aktuális megfigyelt adatok közötti eltérést) [23, 25, 26]. Az általános felírása egy PINN hálózat költségfüggvényében szereplő tagoknak az alábbi egyenlettel szemléltethető:

$$\theta^* = \arg \min_{\theta} (\omega_F L_F(\theta) + \omega_B L_B(\theta) + \omega_D L_D(\theta)). \quad (2.14)$$

A neurális hálózat megtanulja közelíteni a differenciálegyenleteket a hálózat θ paramétereinek meghatározásával, egy olyan költségfüggvény minimalizálásával, amely függ a differenciálegyenlettől (L_F), a határfeltételektől (L_B), és esetlegesen néhány ismert mérési adatponttól (L_D). Az $\omega_F, \omega_B, \omega_D$ a tagok megfelelő súlyozásait jelentik. A tagok az egyenletben az alábbi jelentéssel bírnak. L_D az illeszkedésért felelős komponens:

$$L_D = \frac{1}{N} \sum_{i=1}^N \|(y_i - \hat{y}_i)\|^2, \quad (2.15)$$

ahol $i = 1, 2, \dots, N$ és N a rendelkezésre álló adatpontok száma, y_i a megfigyelt adatpont, míg $\hat{y}_i$ a hálózat kimenete. L_F a hálózat kimenetét arra kényszeríti, hogy eleget tegyen a megadott biológiai,

fizikai törvényeknek, melyek differenciálegyenletek formájában vannak definiálva:

$$L_F = \frac{1}{M} \sum_{j=1}^M \|\mathcal{D}(\hat{y}(t_j); \theta) - f(t_j)\|^2, \quad (2.16)$$

ahol $\mathcal{D}$ differenciál-operátor, y a kimeneti változó, $j = 0, 1, \dots, M$, ahol M azoknak a pontoknak a száma ahol a differenciálegyenletnek érvényesnek kell lennie. A derivált értékét automatikus differenciálással kapjuk meg, míg az $f(t_j)$ a differenciálegyenlet jobb oldali kifejezése t_j időpontban. Az L_F hiba minimalizálásával biztosíthatjuk, hogy a hálózat kimenete közelítse a differenciálegyenletek megoldását.

A peremfeltételekre is felítható egy általános hibafüggvény:

$$L_B = \frac{1}{P} \sum_{k=1}^P \|\hat{y}_k - g_k\|^2, \quad (2.17)$$

ahol P a peremértékek száma, valamint g_k egy adott peremfeltétel függvény kimenete [27]. Mindazonáltal esetünkben a 2.1. fejezetben bemutatott szakadás következtében – a dózisok beadása során – a teljes mérési időintervallumra alapvetően nem modellezhető a problémánk PINN-nel.

Egy másik megközelítése a hálózatok és a differenciálegyenletek integrációjának a neurális köznséges differenciálegyenletek (NODE - Neural Ordinary Differential Equations) [28]. Az Euler módszer az egyik legegyszerűbb eljárás a differenciálegyenletek kezdeti érték problémáinak megoldására, ahol a következő érték az előző pontban meghatározott meredekség és a lépésköz szorzata alapján számítható ki. A reziduális hálózatok hasonló koncepción alapszanak, hasonlóan a mély neurális hálózatokhoz annyi különbséggel, hogy egy réteg bemenetéhez hozzáadjuk az előző előtti réteg kimenetét is. Így a hálózat következő állapotára az alábbi összefüggés írható fel:

$$h_{t+1} = h_t + f(h_t, \theta_t), \quad (2.18)$$

ahol t a lépésszámot, h_t pedig a t -edik lépésben a hálózat állapotát jelöli. Ha a lépésközöket egyre kisebbre vesszük, akkor ez a folyamat egy differenciálegyenlet folytonos formájához konvergál:

$$\frac{dh(t)}{dt} = f(h(t), t, \theta), \quad (2.19)$$

ahol $h(0)$ a kiindulási állapot, $h(t)$ pedig a t időpillanatban a kimenet, ami megfelel az ODE kezdeti érték probléma megoldásának. A NODE felhasználásával kezdeti érték problémákat tudunk megoldani állandó memóriafelhasználással, hiszen nem szükséges a köztes lépések tárolása. A módszert többek között paraméter identifikációra is alkalmazták [29], [30], [31].

Munkám korábbi szakaszában már kidolgoztam egy neurális hálózat alapú eljárást a paraméterek előrejelzésére [32]. A hálózat két neurális háló összeépítéséből tevődött össze, az első hálózat a paraméterbecslő hálózatként funkcionált és annak a költségfüggvényébe volt beépítve egy második – előtanított – hálózat, mely kezdeti érték problémák megoldását szolgálta. A költségfüggvényben szereplő második neurális hálózat előtanításához minden egyedhez – melyeknek a paraméterét szerettem volna meghatározni – tanító adatokat generáltam, majd generált adatokkal egyedi neurális hálózatot rendeltem hozzá. A második, kezdeti érték problémát megoldó hálózat létrehozását az indukálta, hogy az általam akkor alkalmazott keretrendszer (*TensorFlow*) nem rendelkezett beépített differenciálható ODE solverrel, mely párhuzamosan több kezdeti érték problémát is meg tudott volna oldani. Ennek következtében az első hálózat nem tudta a hibát visszaterjeszteni a súlyaira az ODE megoldón keresztül. A létrehozott algoritmus alkalmas volt tumortérfogat és dózis párokból adott egyedek paramétereinek megállapítására, viszont csak nagy számú iterációt követően, melyet egy előtanítási szakasz előzött meg minden egér esetében. Továbbá a lehetséges megoldásokat az előtanított ODE megoldó hálózat pontossága korlátozta a költségfüggvényben.

Ezt követően megvizsgáltam a paramétereket és klaszterező algoritmussal csoportosítottam a hasonló tumordinamikával rendelkező egyedeket [33], majd az általunk mért tumortérfogatok közötti kiugró értékek detektálására és eltávolítására hoztam létre, és alkalmaztam két algoritmust [34]. A kiugró értékek detektálása elengedhetetlen a paraméterbecslést megelőzően, mivel a konvergenciát, ezáltal a meghatározott paramétereket is nagymértékben befolyásolhatják.

2.3. Neurális hálózatok és autoenkóderek

A mesterséges neurális hálózatok (ANN - Artificial Neural Network) a gépi tanulás egy fontos részterületét alkotják. Ezek a rendszerek az emberi agy neuronhálózatainak működésének analógiájára lettek megalkotva. Az agyban lévő neuronoknak több ezer szinaptikus kapcsolata van más neuronokkal. Az agykéregben létrejövő összekapcsolt neuronok hálózata felelős a vizuális, hang-

és érzékszervi adatok feldolgozásáért [35]. Általában egy mesterséges neurális hálózat több rétegből áll: a bemeneti, egy vagy több rejtett és a kimeneti réteg. A több rejtett réteget tartalmazó hálózatot mély neurális hálózatnak nevezzük. Minden egyes neuron kapcsolódik a következő réteg neuronjaihoz, ahol az összeköttetések súlyozottak és a neuron aktiválódását egy eltolás, vagy más néven torzítás szabályozza. A rétegek különböző számú neuront tartalmazhatnak, és minden réteg kimenete a következő réteg bemenete. Minden rétegben lineáris transzformáció történik az alábbiak szerint:

$$z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]}, \quad (2.20)$$

ahol $W^{[l]}$ a súlyokat tartalmazó mátrix, $b^{[l]}$ az eltolásokat tartalmazó vektor, $a^{[l-1]}$ az előző réteg kimenete, l pedig a réteg indexe.

Ezt követően a lineáris transzformáció kimenete egy nemlineáris transzformációs függvényen megy keresztül, mint például a szigmoid függvény. Ezeket nevezzük aktivációs függvényeknek. A szigmoid függvény az alábbiak szerint írható fel:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}, \quad (2.21)$$

ahol x a bemenete a függvénynek. A szigmoid függvény 0 és 1 közé képezi le a bemenetet. Tehát az l -edik réteg kimenete

$$a^{[l]} = \sigma(z^{[l]}). \quad (2.22)$$

A betanításhoz szükséges egy költségfüggvény ($\mathcal{L}$), amely méri a különbséget az előre jelzett kimenetek és a valódi címkék között, melynek értéke alapján történik a hiba visszatérjesztése a hálózat súlyaira láncszabály alkalmazásával, azaz

$$\frac{\partial \mathcal{L}}{\partial W^{[l]}} = \frac{\partial \mathcal{L}}{\partial z^{[l]}} \cdot \frac{\partial z^{[l]}}{\partial W^{[l]}}. \quad (2.23)$$

Ezt követően a súlyok frissítése szükséges, egy választott optimalizáló által:

$$W_{ij}^{[l]} = W^{[l]} - \eta \frac{\partial L}{\partial W^{[l]}}, \quad (2.24)$$

ahol η a tanulási sebesség.

A neurális hálózatok kialakítása során a választott architektúra általában az adott megoldandó problémától függ, mint például osztályozási feladatok vagy dimenziócsökkentés. A dimenziócsökkentésre alkalmazott architektúrák gyakori példája az autoenkóder [36]. Az autoenkóder egy típusa a felügyelet nélküli neurális hálózatoknak, amelyet elsősorban főbb jellemzők kinyerésére használnak [37]. Kezdetben úgy hivatkoztak rá, hogy "hibavisszaterjesztés tanár nélkül", hiszen a bemeneti adatok által tanul a hálózat [38], ezáltal nincs szükség címkézett adathalmazra. Ezt a tulajdonságát használom fel a 3. fejezetben részletezett módon.

Az autoenkóder működése formálisan is felírható. Például, ha az enkódert $h(\cdot)$ függvényként jelöljük, akkor:

$$z_i = h(X_i, \theta), \quad (2.25)$$

ahol X_i a bemenet, $i = 0, 1, \dots, N$ és N a bemenetek száma, illetve θ tartalmazza az enkóder paramétereit. A dekóder rész szintén felírható a

$$\hat{X}_i = g(z_i, \omega), \quad (2.26)$$

függvénnyel, ahol ω tartalmazza a dekóder paramétereit. A két szakasz közös költségfüggvénnyel rendelkezik, és céljuk a θ és ω paraméterkészletük optimalizálása a bemenet és kimenet közötti különbség minimalizálása során. Az optimalizáció az alábbi általános költségfüggvényt minimalizálja:

$$SSE = \sum_{i=1}^N (\hat{X}_i - X_i)^2, \quad (2.27)$$

ahol X_i az i -edik bemenet, $\hat{X}_i$ pedig annak a rekonstrukciója.

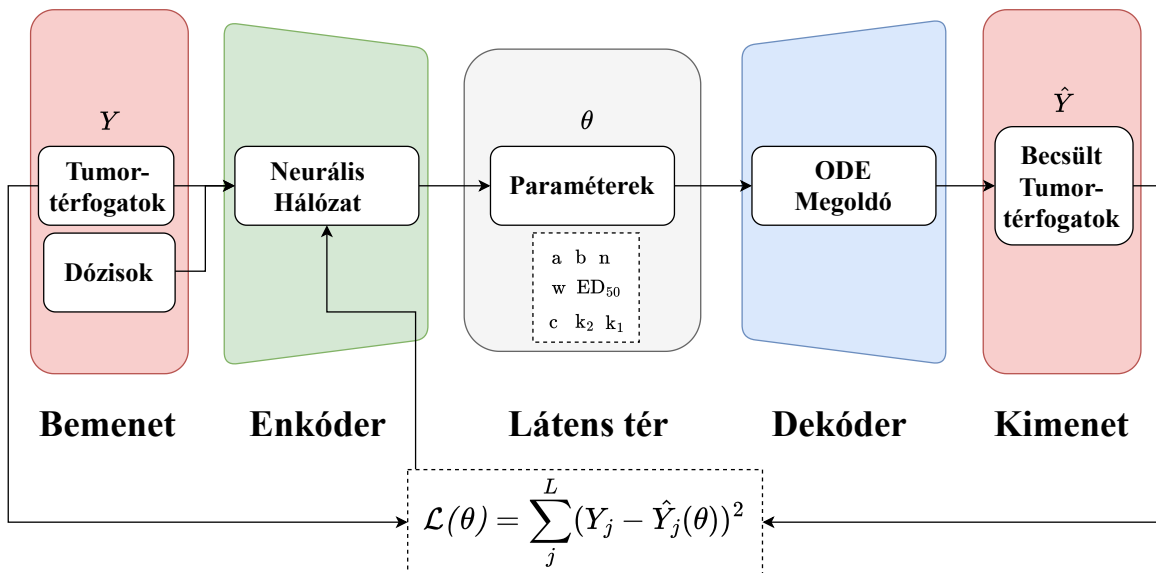
3. fejezet

Megvalósítás

3.1. Paraméterbecslő rendszer felépítése

A 3.1. ábrán látható a létrehozott teljes rendszer felépítése, melynek a struktúrája megfelel egy módosított autoenkódernek. Az autoenkóderek két fő komponensből épülnek fel: egy enkóderből és egy dekóderből. Az enkóder feladata, hogy a bemeneti adatokból állítson elő egy tömörített reprezentációt, míg a dekóder feladata, hogy a tömörített reprezentációt visszaalakítsa az eredeti bemeneti adatokhoz lehető legközelebb álló formájára. Annak mértékét, hogy mennyire sikerült jól megközelíteni az eredeti adatokat, a rekonstrukciós hibával írhatjuk le, mely a bemenet és a kimenet közötti négyzetes különbség.

Esetünkben a bemenetek a rendelkezésünkre álló tumortérfogatok és a beadott kemoterápiás szer dózisainak értékeiből alkotott adott hosszúságú idősorok. A hálózat tanításához nem szükségesek a modell alapján hozzátartozó paraméterértékek, azokat a tanítás folyamata alatt a rendszer magától határozza meg, azáltal, hogy a hálózat súlyait úgy hangolja, hogy a bemeneti tumortérfogat és az aktuális paraméterek által megszabott kimeneti tumortérfogat között a különbség minimális legyen. Az autoenkóder tehát esetünkben abban tér el az irodalomban elterjedt alapkoncepciótól, hogy csak az enkóder egy neurális hálózat, a dekóder nem az, nem tartalmaz tanítható súlyokat. A dekóder egy ODE megoldó, ami a paraméterekből és a kezdeti feltételekből előállítja a tumortérfogatokat. Az autoenkóder tanítása során először tanító adatokat generáltam a tumormodell felhasználásával. Ezt követően a generált adatok első felével előtanítottam a hálózatot, majd pedig a második felével betanítottam az autoenkódert.



3.1. ábra. A paraméterbecslő autoenkóder felépítése. Az autoenkóder a bemeneti tumortérfogatok és a becsült tumortérfogatok közötti különbség minimalizálásával határozza meg a bemeneti tumortérfogatokhoz tartozó paraméterek értékeit.

3.2. Tanító adatok generálása

Mivel valós, egereken végzett mérésekből nem áll rendelkezésünkre elegendő adat egy neurális hálózat betanításához, így a tumormodell felhasználásával létrehoztam 20000 szimulált bemeneti adatot tartalmazó adathalmazt. A tanító adatgenerálás során pseudo-véletlenszám számgenerátorral állapítottam meg dózisokat és paramétereket adott intervallumon belül, majd pedig a kezdeti értékek és paraméterek ismeretében szimuláltam a 2.1. alfejezetben ismertetett tumormodell kimenetét egy numerikus integrátor segítségével. Ezáltal dózisokat és tumortérfogatok tartalmazó idősorokat kaptam. Ezeket két mátrixba gyűjtöttem, ahol minden sor egy idősort tartalmazott. A generált paramétereket szintén egy mátrixba gyűjtöttem, ahol a sorok szintén egy virtuális pácienshez tartoztak. Egy virtuális páciens alatt egy paraméterhalmazt értek, ahol a paraméterek a szimulációkhoz használt differenciálegyenletek paraméterei. Ezekkel a szimulációkkal *in silico* méréseket tudunk végezni.

Egy tumortérfogatok tartalmazó idősor szimulációja során először egyenletes eloszlásból generáltam a 2.1. táblázatban feltüntetett tartományból egy-egy értéket, mind az a , b , n , w , ED_{50} , c ,

k_1 és k_2 paraméterek esetében. Ezt követően virtuális kezelést generáltam, azáltal, hogy figyelembe vettem a 3.2. ábrán feltüntetett kezelési rendet. Az ábrán az látható, hogy a normál protokoll szerint minden hétköznap tumortérfogat mérés van, kedden és pénteken pedig kemoterápiás szer beinjekciózása történik. Ezt a szimuláció során úgy vettem figyelembe, hogy a véletlenszám-generátorral létrehoztam minden keddi és pénteki napra nulla és nyolc mg/kg nagyságú dózisokat, egyenletes eloszlással. A felső határa a beadott dózisoknak az állatkísérletek során alkalmazott felső határ volt.

Dózis	☐	☒	☐	☐	☒	☐	☐
Mérés	☒	☒	☒	☒	☒	☐	☐
	Hétfő	Kedd	Szerda	Csütörtök	Péntek	Szombat	Vasárnap

3.2. ábra. A tanító adatok generálása során alkalmazott mérési elrendezés. Az alkalmazott protokoll alapján hetente kétszer, kedden és pénteken történik injekciózás, míg hétköznaponta történik tumortérfogat mérés.

Mivel a differenciálegyenletek megoldása függvények sokasága, így az adott feltételekhez tartozó megoldást keressük. Tehát a numerikus közelítő módszerek esetében is szükséges a kezdeti értékek megadása, amivel leszűkíthető a keresési tartomány. A tumormodell négy állapotváltozót tartalmaz, ezért mindegyik változónak szükséges megállapítani az értékét a kezdeti időpillanatban. A szimuláció során létrehoztam egy kezdeti értékeket tartalmazó mátrixot is, ahol minden sor egy virtuális pácienshez tartozó kezdeti értékeket tartalmazó vektor, mely négy hosszúságú és elemei az x_1 , x_2 , x_3 és x_4 a $t = 0$ időpillanatban. Az $x_1(0)$ értékét – mely az élő tumor térfogata – a teljes tumor értékével közelítettem, azzal a feltételezéssel élve, hogy a mérés elején még nincs elhalt tumorrész és véletlenszám-generátorral határoztam meg, 0 és 200 mm³ között egyenletes eloszlással. Ennek következtében az elhalt tumor mennyiségét ($x_2(0)$) viszont nullának vettem. A gyógyszer koncentrációjának értékét a vérben ($x_3(0)$) és a szövetben ($x_4(0)$) szintén nullának vettem a kezdeti időpillanatban.

Tehát a négy – dózisokat, tumortérfogatokat, paramétereket, kezdeti értékeket tartalmazó – mátrix sorainak a sorszáma tekinthető az elsődleges kulcsnak vagy azonosítónak, ami összeköti a három tanításra létrehozott táblát. Minden táblázat adott sora egy adott virtuálisan generált páciens

mentett értékeit tartalmazza. Minden virtuális páciens esetében szimulációkat egyszerre egy napra futtattam le *torchode* [39] közönséges differenciálegyenlet numerikus megoldójával, a kapott tumortérfogatokat kimentettem a hétköznapiakon, kedden és pénteken pedig hozzáadtam az x_3 változóhoz a generált adott napra eső dózist. Összesen 20000 virtuális páciens generáltam, tehát az idősorokat tartalmazó mátrixok 20000×105 méretűek voltak, a kezdeti értékeket tartalmazó mátrix 20000×4 -es, a paramétereket tartalmazó pedig 20000×8 -as méretű volt. A 20000 méretű adatbázist elfeleztem, első felét előtanításra a második felét pedig az autoenkóder tanítására használtam fel.

3.3. A hálózat előtanítása

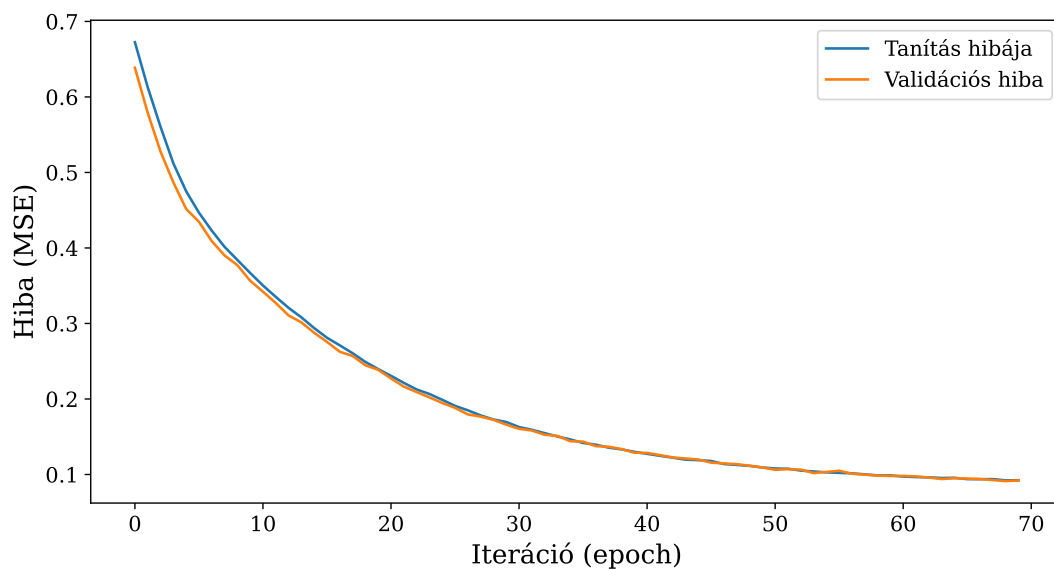
Ahhoz, hogy az autoenkóder a tanítás során biológiailag értelmezhető nagyságrendű paramétereket állítson elő, először előtanítottam a hálózatot ismert paraméterekkel. A paraméterek többsége nap^{-1} mértékegységű, és egy adott skálán mozognak, melynek van egy biológiailag limitált alsó és felső határa. Az előtanításra továbbá azért is szükség volt, mivel az enkóder kimenete a tumormodell paraméterei, ami egyben a bemenete az ODE megoldónak. Mivel a numerikus integrátor csak adott intervallumon belüli értékekkel tudja szimulálni a tumormodell kimenetét, a bemeneti paraméterek nem vehetnek fel tetszőlegesen nagy értéket, különben numerikusan instabil kimenetet eredményezhetnek. Ha nem tanítottam volna elő a hálózatot, akkor olyan paraméterkombinációkat is előre jelzett volna, amelyeknél az ODE megoldó a hirtelen nagy változások esetében nem tudná kezelni az adaptív lépésköz választást.

Az előtanítás során a hálózat bemenete két idősor volt: a tumortérfogatokat és a dózisokat tartalmazó mátrixok sorai, míg a kimenete a paraméterek voltak. A bemeneteket és a kimeneteket normalizáltam mielőtt betanítottam a hálózatot. Normalizálás során minden változót 0 és 1 közé skáláztam át, min-max normalizációval az alábbiak szerint:

$$p'_j = \frac{p_j - p_j^{\min}}{p_j^{\max} - p_j^{\min}}, \quad (3.1)$$

ahol p'_j a j -edik paraméter normalizált értéke, míg $p_j^{\min}$ és $p_j^{\max}$ a j -edik paraméter határai a 2.1. táblázatban.

A tanítás során az adathalmazt felosztottam 0,70:0,15:0,15 arányokban tanító, validáló és teszt adathalmazra. A tanító adathalmazt használtam a hálózat előtanításához, míg a validáló adatokat arra, hogy a hálózat hiperparamétereit az előtanítás során hangoljam és a túlillesztést elkerüljem. Minden iteráció végén a validációs adathalmazon értékeltem ki a hálózat teljesítményét, viszont az ebből számított hibát nem terjesztettem vissza a hálózat súlyaira. A 4. fejezet 4.2., 4.3. és 4.4. ábráit is a validációs adatok alapján hoztam létre, melyekkel a hálózat tanulását szemléltettem az iterációk növekedésével. A teszt adathalmazt a már betanított hálózat kiértékelésére használtam.

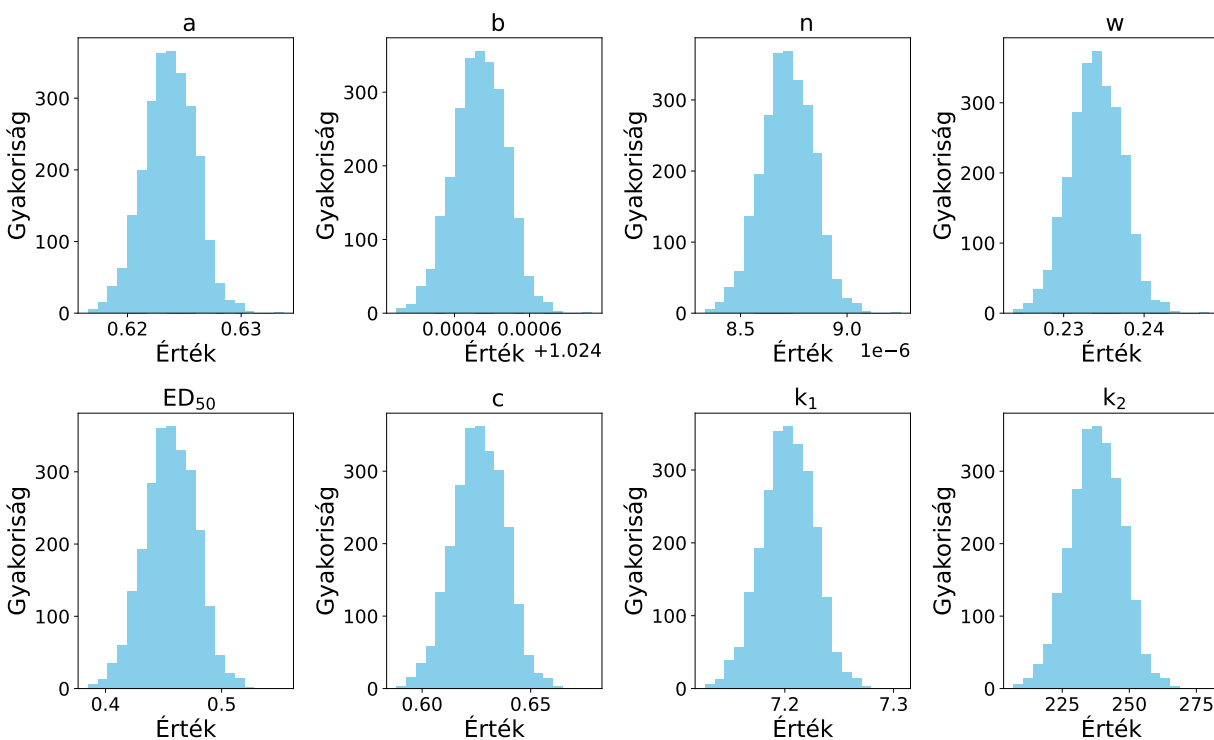


3.3. ábra. A neurális hálózat előtanítása során kapott hibák a tanító adathalmaz esetén, ami a teljes adathalmaz 70%-a (kék) és a validációs adathalmaz esetén, ami a teljes adathalmaz 15%-a (narancssárga). A hibák a becsült paraméterek és az eredeti paraméterek normalizált értékei közötti átlagos négyzetes eltérést mutatják.

Az előtanítás folyamata során 10000 adatot használtam fel. Az adathalmazt 1000 nagyságú kötegekre osztottam. Egy teljes iteráció abból állt, hogy minden kötegre megtörtént a hiba kiértékelése és annak a visszaterjesztése a hálózat súlyaira, tehát összesen egy iteráción belül 10 alkalommal terjesztettem vissza a hibát. A folyamat során ún. korai leállási feltételt alkalmaztam, mely szerint a tanítás leállt, ha 5 iteráción keresztül nem változott a hiba értéke a validációs adathalmazon. A hiba számítása során átlagos négyzetes hibát alkalmaztam. A 3.3. ábrán látható az előtanítás során kapott átlagos négyzetes eltérés iterációnkénti értéke. Csak a görbéket kiértékelve megállapítható,

hogy a hálózat be tudott tanulni a paraméterek értékeire. A validáció és a tanítás görbéje hasonló ívet követ, mely alapján sem túl, sem pedig alulillesztés nem figyelhető meg. Ha a validációs hiba egy idő után növekedni kezdett volna, miközben a tanítás alatt számított hiba nem, az azt jelentette volna, hogy a hálózat túlilleszkedett (overfitting) a tanító adathalmazra. Mindazonáltal a korai leállási feltétel ezt az esetet hivatott megakadályozni.

A teszt adathalmazon kiértékeltem az előtanított hálózat predikciós képességét. Az erre a célra elkülönített adathalmazon előre jeleztem a paraméterek értékeit majd pedig egy hisztogramon ábrázoltam azokat. A paramétereket ábrázolás előtt denormalizáltam az eredeti intervallumukra. A 3.4. ábráról megállapítható, hogy egy szűkebb intervallumra tanult rá a hálózat, mint az eredeti intervallum. Ezáltal feltételezhető, hogy kevésbé jó általánosító képességgel rendelkezik. Mindazonáltal a cél az előtanítás során az volt, hogy a hálózat megtanuljon egy olyan intervallumból értékeket előre jelezni, amelyek az ODE megoldó bemeneteként szolgálhatnak stabil kimenetet produkálva.

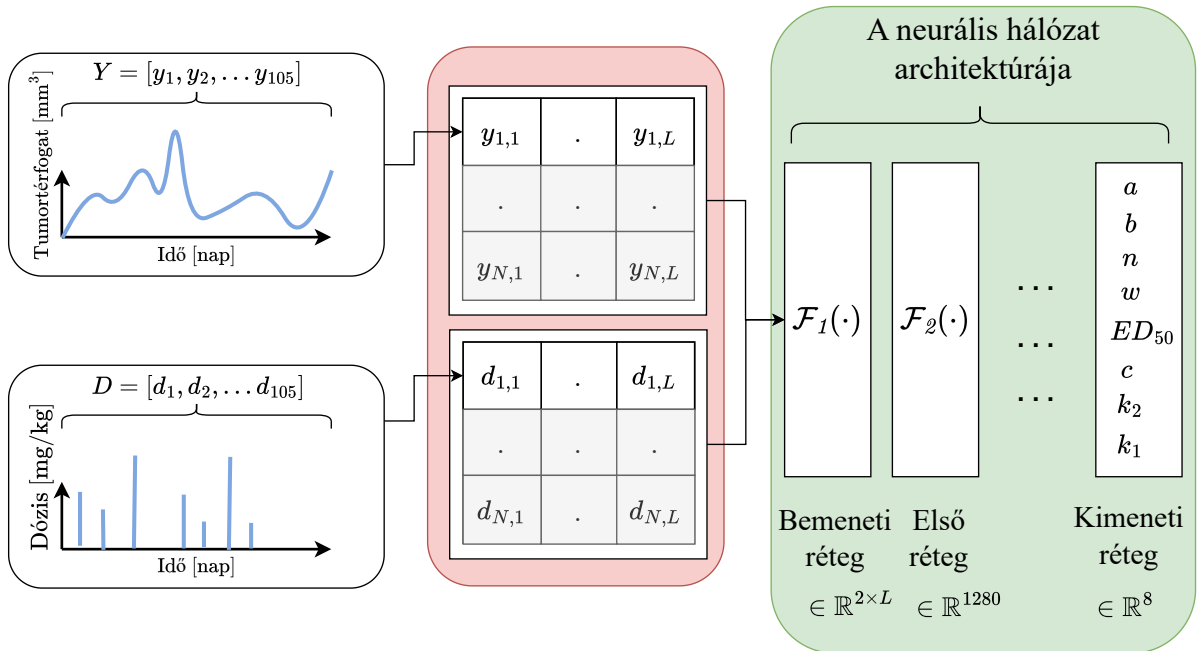


3.4. ábra. A neurális hálózat előtanítását követően kapott paraméterek (a, b, n, w, ED₅₀, c, k₁, k₂) értékeinek gyakoriságai a tesztelésre elkülönített adatokon.

3.4. Az autoenkóder tanítása

A 3.5. ábrán látható a 3.1. ábra bal oldala részletezve. Az autoenkóder bemenetei a tumortérfo-
gatókat és a dózisokat tartalmazó két $N \times L$ nagyságú mátrix, ahol N a bemeneti adatok száma
(esetünkben 10000), míg L a mérés hossza. A valós egérkísérletek során is a mért tumortérfogatok
és a beadott dózisok alapján szeretnénk az adott egér paramétereit meghatározni, majd a paraméte-
rei alapján terápiát generálni.

A kiértékelés során három különböző hosszúságú tumortérfogat mérési intervallumra tanítot-
tam be a hálózatot és vizsgáltam meg a predikciós képességét. A mátrixok minden sora egy egyedi
virtuális pácienset tartalmazott, ahogy az 3.5. ábra mutatja. A két mátrixot normalizáltam, a maxi-
mum és minimum értékeik alapján, hogy egyforma súllyal vegye figyelembe a hálózat mind a két
változót. A normalizálás az (3.1) egyenlethez hasonlóan történt.



3.5. ábra. Az autoenkóder első felének, a bemenetének és a neurális hálózatának a részletezett ábrája. A hálózat bemenete a tumortérfogatókat és a dózisokat tartalmazó idősorok.

A neurális hálózat architektúrája és hiperparaméterei

A hálózat hét rétegből épül fel. A bemenete a tumortérfogatokat és dózisokat tartalmazó idősorok, összesen $2 \times L \times 1280$ nagyságú, míg a kimenete 64×8 méretű. A hálózat a bemeneti adatokat egyetlen vektorra alakítja, ami aztán a hálózat bemenetét képezi. Az első réteg, amely 1280 dimenziós kimenetet hoz létre, szigmoid aktivációs függvényeket használ, amelyeket dropout követ. Az ún. dropout segít megakadályozni a túltanulást azáltal, hogy véletlenszerűen nullázza (kijeti) az egyes bemeneteket a tanulás során. A dropout arányt 0,3-nak állítottam be.

A másodiktól az ötödik rétegig minden réteg szintén 1280 egységet tartalmaz, és mindegyiket szigmoid aktiváció és dropout követ. A hatodik réteg a kimenetét 64 dimenziósra csökkenti, ami előkészíti az adatokat a kimeneti réteg számára. Az utolsó (kimeneti) réteg, amely lineáris kimenetet használ, az előre meghatározott kimeneti méretnek (paraméterek számának) megfelelően adja vissza az eredményt.

A hálózat tanítása során fontosnak bizonyult az aktivációs függvény megválasztása. A kiválasztása során figyelni kellett arra, hogy mind az előtanítás, mind pedig az autoenkóder tanítása során jól teljesítsen a hálózat. Bár a ReLU aktivációs függvény az előtanítás során jobb teljesítményt nyújtott, az autoenkóderben rosszabbul teljesített.

A hálózat költségfüggvényében átlagos négyzetes hibával vizsgáltam az eltérést, mely nagyobb jelentőséget tulajdonít nagyobb hibáknak a négyzetes tulajdonságából adódóan. Ezáltal a hálózatot arra szerettem volna kényszeríteni, hogy azokra az idősorokra is rátanuljon, amelyek kevesebbszer fordultak elő, de nagyobb hibákat tartalmaztak. A tanuláshoz 10^{-3} tanulási sebességet választottam.

A tanításhoz NAdam (Nesterov-accelerated Adaptive Moment Estimation) optimalizálót választottam, amely az Adam optimalizáló algoritmust kombinálja a Nesterov momentummal. A különböző paraméterek adaptív tanulási sebességének meghatározásával megbecsüli a gradiens első és második momentumát, ezáltal a Nesterov momentum beépítésével a konvergencia gyorsabbá válik [40].

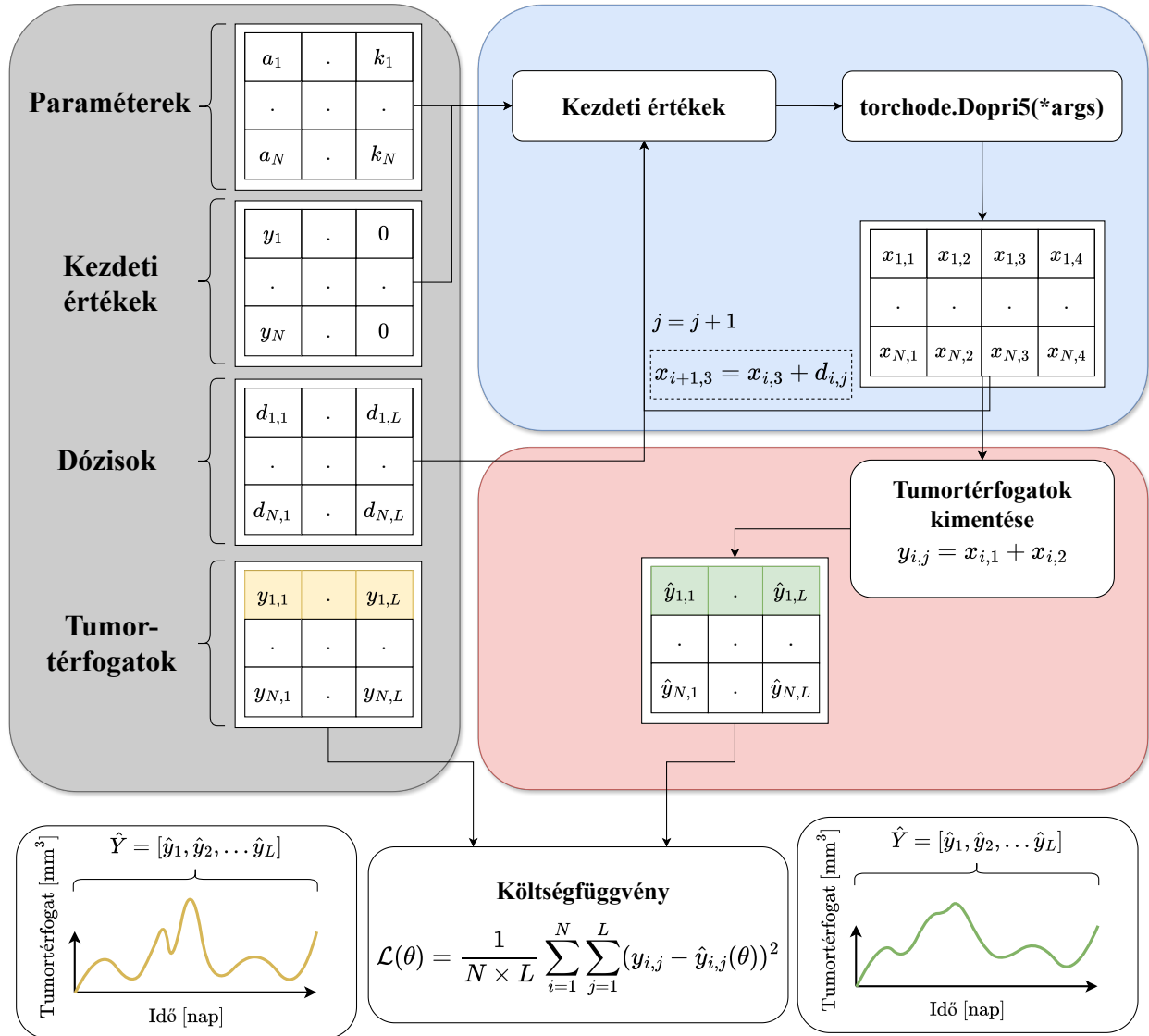
A matematikai modell szimulációja

Az enkóder, azaz a neurális hálózat kimenete a paraméterek. A becsült paramétereket tartalmazó mátrixokból, a kezdeti feltételekből és a beadott dózisok ismeretében egy ODE megoldó segítsé-

gével a rendszer kimenete előállítható. Mivel a differenciálegyenleteket nem tudjuk analitikusan megoldani, ezért numerikus módszerrel szimuláljuk a megoldását. A numerikus integráció a legelterjedtebb módszer a közelítő megoldások előállítására. Különböző integrációs módszerek léteznek, eltérő pontossággal és konvergencia sebességgel, mely közül a legelterjedtebb az Euler-módszer [41]. Mindazonáltal az Euler-módszer sok esetben nem kellően pontos, emiatt számos más módszer is elterjedt. A munkám során ötödrendű Dormand-Prince [42] módszert használtam az egyenletek közelítésére. A Dormand-Prince módszer egy explicit módszer, továbbá a lépésköz nagyságát automatikusan határozza meg. Emiatt fontos beállítani egy toleranciát, amivel szabályozható, hogy mekkora hibát engedhet meg a szimuláció során. Az abszolút megengedett hiba toleranciáját 10^{-5} -nek, míg a relatív hiba toleranciáját 10^{-3} -nak állítottam be.

A neurális hálózatok tanítása során fontos, hogy a lépések differenciálhatóak legyenek és az adatok mátrixok formájában legyenek felírva. Előbbi miatt elengedhetetlen, mert a hálózat autódifferenciálás segítségével terjeszti vissza a műveleteken keresztül a hibát a hálózat paramétereire a tanulás során. Ha olyan műveletet talál, mely nem differenciálható, akkor a hibavisszaterjesztési folyamat megszakad. A mátrixokba történő felírása az adatoknak a párhuzamosítás és a kötegenkénti adatfeldolgozás következtében számottevő. A teljes bemeneti adathalmazt a tanítás során általában kötegenként adjuk át a hálózatnak és egy-egy köteg egészére történik a költségfüggvény kiszámítása, majd pedig a hibavisszaterjesztés. Ez jóval gyorsabb tanítást eredményez, mintha egyesével vezetnénk be a bemeneteket a hálózatba, valamint kiegyenlítettebb tanulási folyamatot biztosít. Ahhoz, hogy kötegenkénti feldolgozást tudjunk végrehajtani, olyan differenciálható ODE numerikus integrátort használtunk a tanítás során, ami képes párhuzamosan több kezdeti érték problémát megoldani a futási idő jelentős megnövekedése nélkül [43].

A 3.6. ábrán látható a dekóder folyamatábrája. A dekóder bemenetei az enkóder által előre jelzett paraméterek. A paramétereket az algoritmus denormalizálja, mielőtt átadja az ODE megoldónak. A kezdeti értékeket, dózisokat és tumortérfogatokat tartalmazó mátrixokkal együtt, a paraméterek ismeretében az algoritmus megoldja párhuzamosan annyi kezdeti érték problémára a tumormodellt, amekkora kötegméretet került beállításra. Minden esetben egy napra számítja ki az egyenletek megoldását, majd az aznapi dózis hozzáadódik az x_3 állapotváltozóhoz ($x_{i,3} = x_{i,3} + d_{i,j}$). Az i az adott kezdeti érték problémát jelöli (az adott sort a mátrixokban), és N -ig tart, míg $j = 1, 2, 3, \dots, L$, ahol L a mérés hossza. A szimulált állapotváltozókból eltárolódik az x_1 (elő



3.6. ábra. Az autoenkóder dekóder része, mely a paraméterekből, kezdeti értékekből és dózisokból előállítja a szimulált tumortérfogatot. A szimulált tumortérfogat és a valós tumortérfogatok közötti átlagos négyzetes eltérés értéke alapján tanítjuk az enkóder neurális hálózatát.

tumortérfogat) és x_2 (halott tumortérfogat) állapotváltozók összege egy új változóban, ami a teljes tumortérfogatokat tartalmazza. Azt követően, hogy minden napra leszimulálta az ODE megoldó a tumortérfogatokat, az algoritmus az autoenkóder bemenete és kimenete közötti különbséget meghatározza. Minden tumortérfogatokat tartalmazó idősor és az előre jelzett paraméterek által meghatározott tumortérfogatok közötti eltérés kiszámításra kerül. Ezt a *torch.backward()* függvény terjeszti vissza a neurális hálózat súlyaira. A függvény kiszámítja egy skalár (általában a költségfüggvény értéke) gradiensét az előre meghatározott tenzorokra vonatkozóan. Miután a gradiensek kiszámításra kerültek, a NAdam optimalizáló felhasználja a gradienseket a hálózat súlyainak frissítésére, azzal a céllal, hogy minimalizálja a hibát. Ez a folyamat a *torch.step()* függvény meghívásával történik.

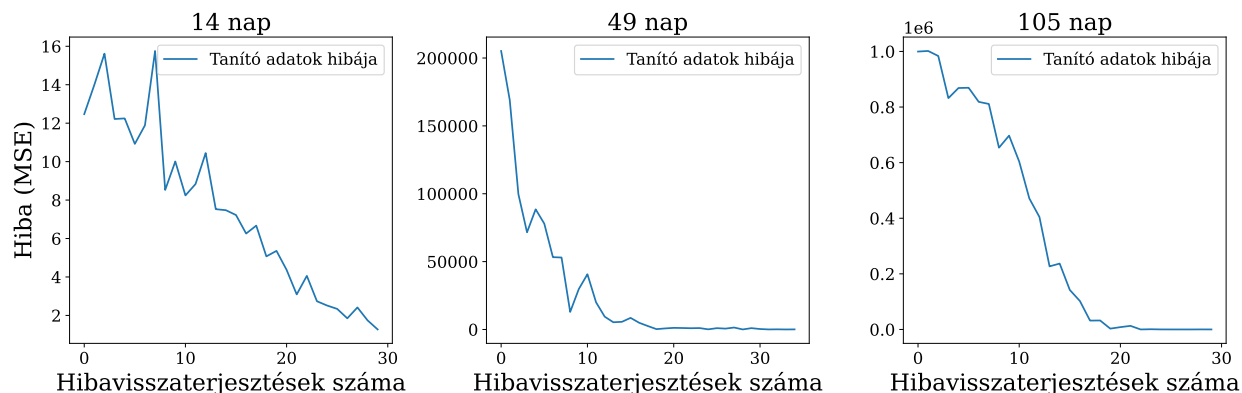
4. fejezet

Eredmények

4.1. A rendszer tanításának kiértékelése

A tanítás során három eltérő hosszúságú – 14 napos, 49 napos és 105 napos – mérési szakaszra tanítottam be a hálózatot a kiértékelések során. A cél az volt, hogy meghatározzam, hogy gyorsabban és pontosabban tanul-e rá a rövidebb hosszúságú tumortérfogatokat tartalmazó idősorokra. A legrövidebb a 2 hetes szakasz volt, majd 7 hétre, végül pedig 15 hétre tanítottam be a hálózatot. A tanítás során a validációs adathalmaz hibájától függő megállási feltételt alkalmaztam (early stopping). Ha a hiba nem változott 3 iteráción keresztül, akkor leállt a tanítás és a legjobban teljesítő hálózat került mentésre. A teljes adathalmazt kötegekbe (részhalmazokba) rendeztem, ahogy azt a 3. fejezetben részleteztem. Minden köteg után hibavisszaterjesztés történt a hálózat súlyaira. Mivel az alkalmazott hálózaton nem hagyományos, pusztán adatokon alapuló tanítást alkalmaztam, hanem a tumormodell beillesztésével plusz információt szolgáltatam a tanuláshoz, így már egy iteráción belül is gyakran nagy hibacsökkenést tapasztaltam. Ebből adódóan nem az iterációk során értékeltem ki a validációs adatokat, hanem minden hibavisszaterjesztést követően, ezáltal jobb képet kapva a tanulás folyamatáról.

A 4.1. ábrán látható a három intervallumra kapott ún. loss görbe, mely a hibák változását mutatja az iterációk alatt. Látható, hogy mind a három esetben körülbelül 30 hibavisszaterjesztés volt szükséges ahhoz, hogy a tanító adathalmazra rátanuljon a hálózat. Továbbá megfigyelhető, hogy minél rövidebb az előrejelzés időintervalluma, annál kisebb hibáról indul a tanítás. Ez abból adódik, hogy minél hosszabb az időintervallum annál nagyobb értékeket vehet fel a tumortérfogat az



4.1. ábra. Az autoenkóder három tanulási görbéje a három intervallumra. Mivel a hálózat nagy mértékben tanult minden kötegen történő hibavisszaterjesztés után, így a hagyományos iterációs (epoch) helyett a hibavisszaterjesztések számát ábrázoltam a vízszintes tengelyen.

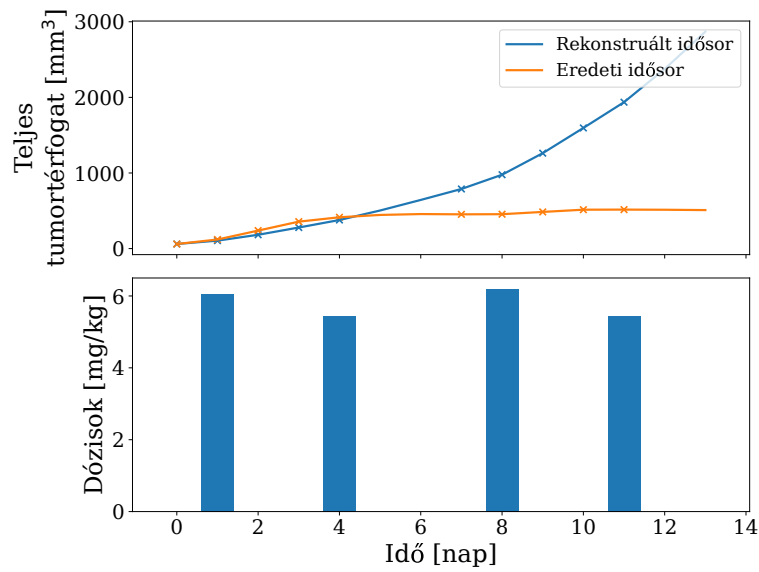
exponenciális növekedésének következtében (ahogy a 4.4. ábra is igazolja). Továbbá, az ábrák alapján megállapítható, hogy az előtanított hálózat képes volt folytatni a tanulást a differenciálegyenletek hálózatba való építését követően. Tehát a létrehozott architektúra felügyelet nélküli tanulással képes volt jobb eredményeket adni az előtanított hálózathoz képest.

Tanítás 14 napos intervallumon

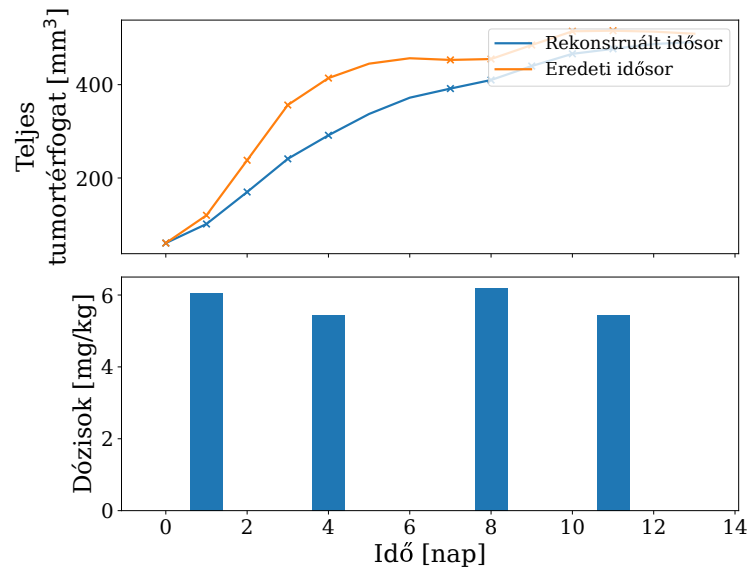
A 4.2. ábrán látható két példa arra, hogy az autoenkóder teljesítménye hogyan fejlődött az iterációk alatt. Mindkét ábrán az előre jelzett tumortérfogatot ábrázoltam késsel, míg narancssárgával az eredeti idősorokat. A normál protokoll szerinti tumortérfogat mérési időpontokat (hétfőtől péntekig) kereszttel jelöltem. A véletlenszerűen nulla és nyolc mg/kg közötti, keddi és pénteki napon generált dózisokat oszlopdia grammal ábrázoltam a tumortérfogat görbék alatt. A felső ábrán az első iterációt követően, míg az alsó ábrán már az autoenkóder betanítását követően történt a validációs adatokon a predikció. Megállapítható, hogy a betanítást követően előre jelzett tumortérfogatot tartalmazó idősor jobban illeszkedik az eredetihez.

Tanítás 49 napos intervallumon

A hét hetes tanítás során szintén kimentettem a validáció során kapott rekonstrukciókat. A 4.3. ábrán látható egy példa a 49 napos előrejelzésre. Ebben az esetben kezdetben túl agresszív tu-



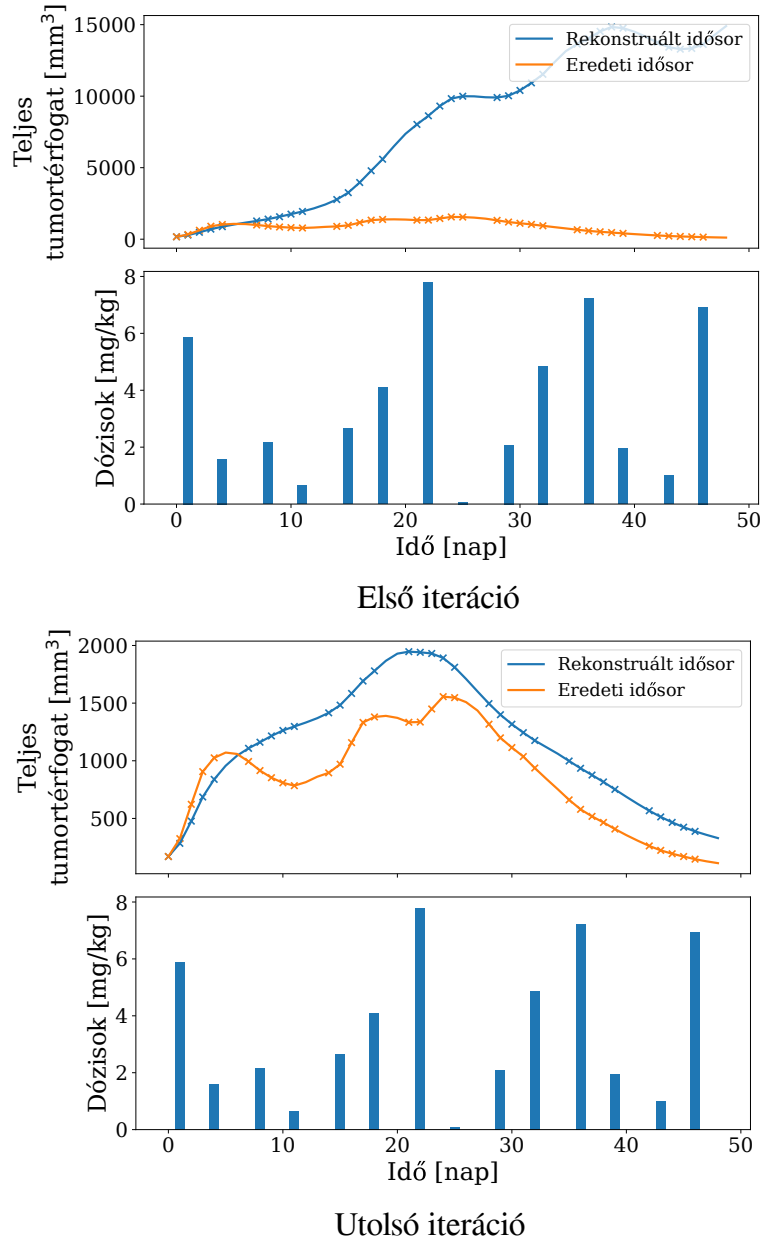
Első iteráció



Utolsó iteráció

4.2. ábra. Példa a validációs adathalmazból a hálózat tanulása alatti teljesítmény változásra két hetes időintervallumon. A tanítás folyamata során kezdetben nagyobb hibával, majd ahogy tanult a hálózat egyre kisebb hibával becsülte meg a tumortérfogatokat.

mordinamikát becsült az autoenkóder. Később azonban, ahogy a tanító adathalmazon betanult – hasonlóan a két hetes verzióhoz – a validációs adatokon is jobban teljesített, amint az alsó görbék összehasonlításából is látszik.



4.3. ábra. Példa a validációs adathalmazból a hálózat tanulása alatti teljesítményváltozásra hét hetes időintervallumon. A tanítás folyamata során kezdetben nagyobb hibával, majd, ahogy tanult a hálózat egyre kisebb hibával becsülte meg a tumortérfogatokat.

Tanítás 105 napos intervallumon

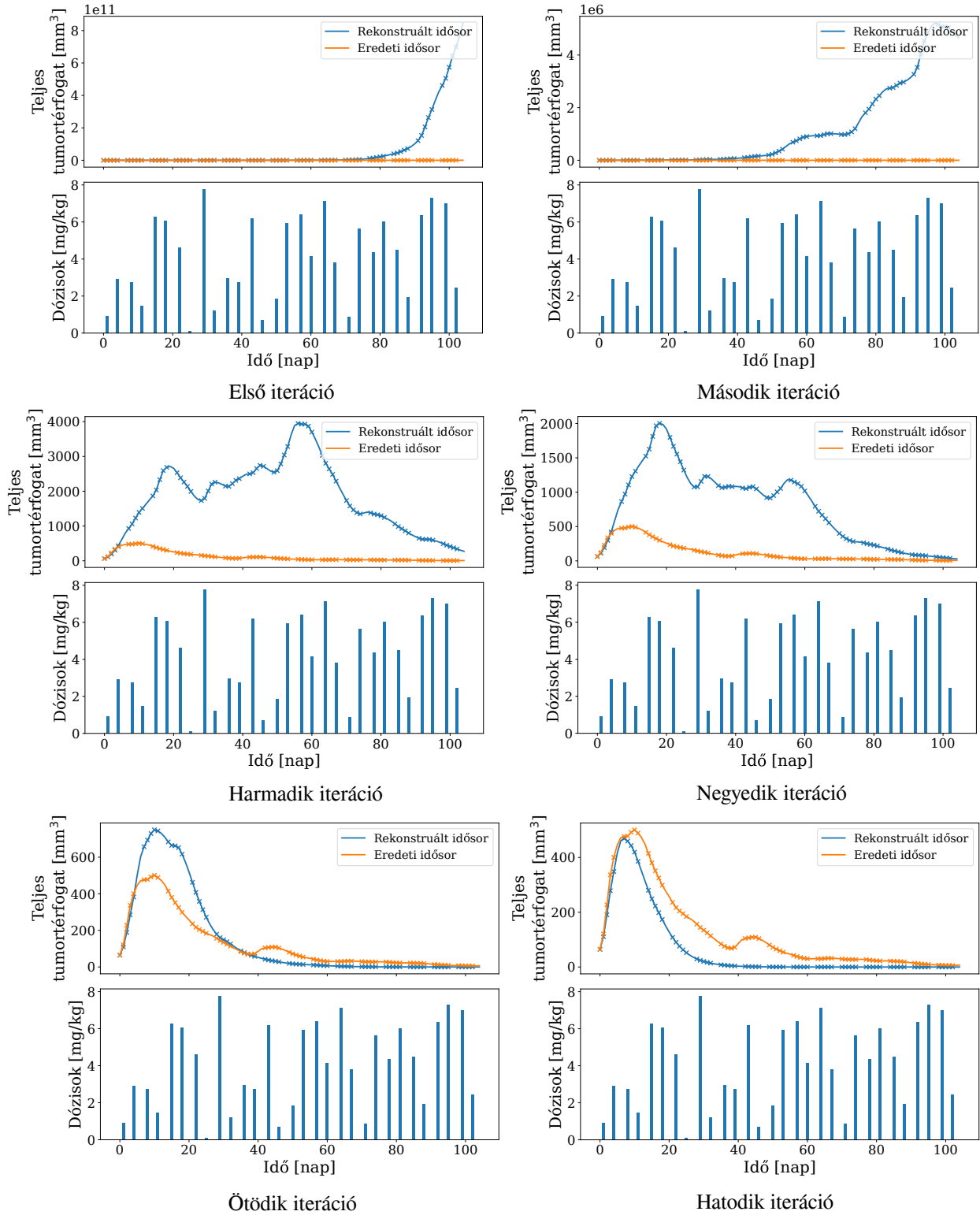
A 105 napos tanítás során – ahogy a 4.4. ábrán is látszik – minden iteráció utáni kimenetet lementtem, hogy vizualizálni tudjam a hálózat tanításának folyamatát. Az ábrán a bal felső sarokban az első iterációt követően látható az előre jelzett paraméterek által meghatározott tumortérfogat érték. Megfigyelhető, hogy az előre jelzett tumortérfogat értéke exponenciálisan elszáll. Mindazonáltal, az iterációk számának növekedésével a két görbe egyre közeledik egymáshoz, míg a tanítás végén egészen hasonló tumordinamikájú görbéket kapunk. A tapasztalatok alapján elmondható, hogy az egyszerűbb, exponenciálisan lecsengő görbéket (kezelésre jól reagáló virtuális páciensek esetét) az esetek többségében jól megtanulta a hálózat. Mindazonáltal, a bonyolultabb tumordinamikát (mely a tanító adathalmaz kisebb részét képezi) az esetek csak kis százalékában tudta helyesen előre jelezni. A jövőben ebből adódóan érdemes a hálózat általánosító képességét növelni, valamint a tanító adathalmaz létrehozása során a különböző tumordinamikájú idősorokat figyelembe venni.

A betanítást követően a paraméterek százalékos eltéréseit is meghatároztam az eredeti értékektől, mely a nyolc paraméterre a 4.5. ábrán látható. Az eltéréseket dobozdiagramon ábrázoltam. A diagram alsó és felső részén a vonal a hibák minimumát jelzi (kivéve szélső értékeket, azaz kiugrókat). A kiugró értékeket körökkel ábrázoltam. Az alsó kvartilis, azaz a doboz alja (Q_1) a hibák alsó 25%-ának felsőhatárát mutatja, míg a felső kvartilis, azaz a doboz teteje (Q_3) a felső 25%-ának az alsó határát jelzi. A mediánt a dobozt keresztbe szelő egyenes jelzi. Az ábráról leolvasható, hogy a relatív százalékos hiba mindenhol 10% körüli volt, egyik paraméter hibája sem különbözött a másiktól szignifikánsan.

Mérés hossza [nap]	14	49	105
Medián	260.5120	303.233	378.7606
Átlag	630.6193	724.867	1044.3889

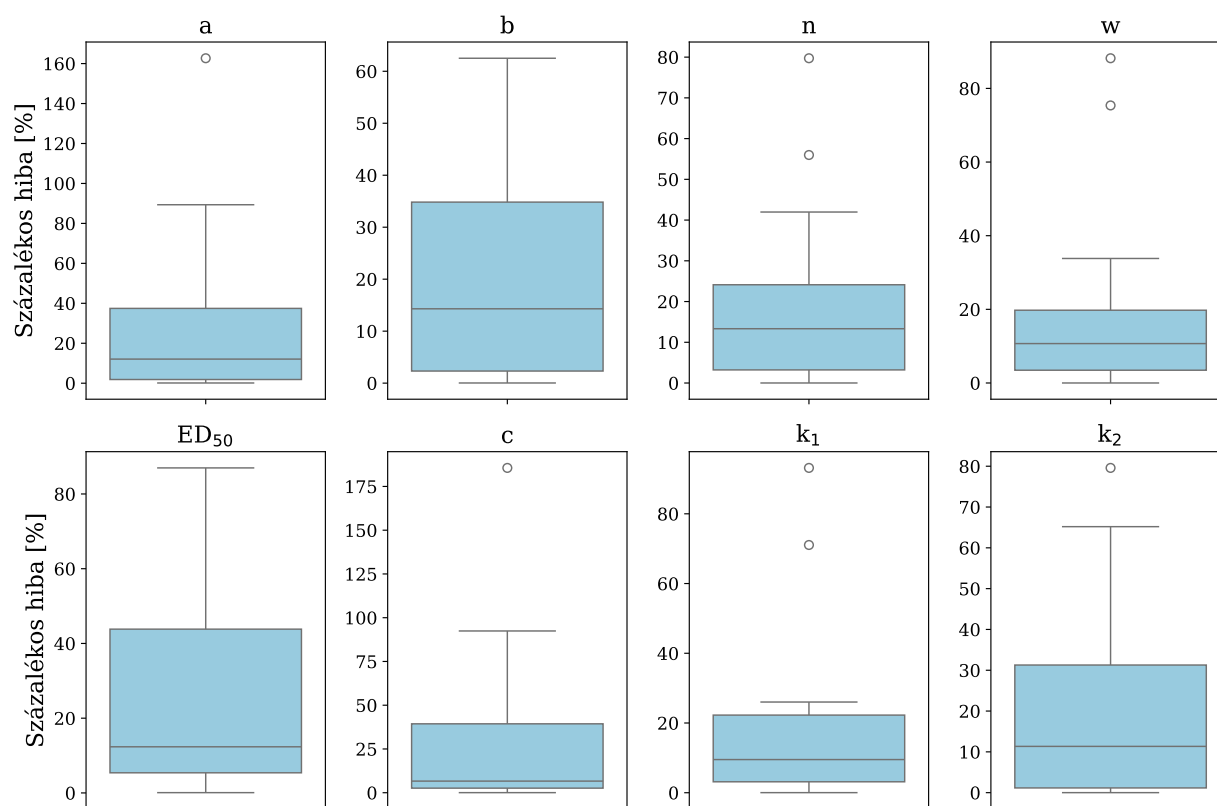
4.1. táblázat. Az idősorok közötti abszolút eltérés a teszt adathalmazon.

Ahhoz, hogy teszteljem a hálózatot betanítást követően is, elkülönítettem a teljes adathalmaz 15%-át, ahogy a 3.3. alfejezetben részleteztem. A 4.1. táblázatban összegyűjtöttem a teszt adatokon végzett kiértékelés hibáit az idősorokra vonatkozóan. A medián és az átlag közötti eltérés azt a feltételezést igazolja, hogy egyes tumordinamikára jól rátanult a hálózat, viszont voltak kiugróan



4.4. ábra. Példa a validációs adathalmazon történő előrejelzésre a hálózat tanulása során a 105 napos intervallumon.

rossz eredmények, mely a hiba átlagát eltolták. Továbbá megfigyelhető az a tendencia is, hogy az idősor hosszának növekedésével az átlagos hiba is nő.



4.5. ábra. A tesztadatokon a paraméterek eltérései az eredeti értékektől mind a három intervallum esetén.

Összességében tehát elmondható, hogy az autoenkóder a rendelkezésünkre álló differenciálegyenlet rendszer beépítésével képes volt megtanulni a modell paramétereit. Három különböző időintervallumra megvizsgáltam a rendszert, melyből megállapítottam, hogy a hálózat azokon az adatokon teljesített jól, amelyek az adathalmaz nagy részét képezték. A futási idő az időintervallum növelésével nem nőtt szignifikánsan, ezáltal a jövőben akár hosszabb intervallumokat is meg lehet vizsgálni.

A jövőben cél a modell módosítása úgy, hogy képes legyen nem csak az egyedek paramétereinek meghatározására, hanem hasonló paraméterű, adott eloszlással rendelkező egér populáció paramétereinek a várható értékének és a szórásának a meghatározására is. Továbbá a hálózat architektúrájának fő jelentősége, hogy nem csak *in silico*, hanem *in vivo* adatokon is képes tanulni,

hiszen nem szükségesek hozzá az adott tumordinamikához tartozó paraméterek értékei, így egy fontos lépés a jövőben a hálózat valós és a zajos mérési adatokon történő kiértékelése és tanítása.

5. fejezet

Konklúzió

A betegségek matematikai modellezése révén azok lefolyása és reakciója különböző kezelésekre leírható és előre jelezhető. A modellek önmagukban azonban nem elegendőek, a modell paramétereinek pontos meghatározása nélkülözhetetlen az optimális terápia megalkotásához. A dolgozatban egy tumordinamikai modell paramétereinek meghatározására hoztam létre gépi tanulási algoritmust, ezáltal a kutatás köztes lépéséhez járultam hozzá. A paraméteridentifikációra alkalmazott iteratív módszerek jelentős időt igényelhetnek az optimális megoldás eléréséhez, továbbá nem garantált, hogy a globális optimumhoz konvergálnak, különösen, ha a kiindulási paraméter értékek távol esnek azok optimális értékeiktől. A rosszul meghatározott értékek által a betegség rossz csoportosítása a kemoterápiás gyógyszer mennyiségének rossz meghatározását eredményezheti. A paraméterek gyors és pontos ismerete által terápia generálható különböző algoritmusokkal egyénre, vagy akár hasonló tumordinamikával rendelkező populációra.

Korábbi munkák során már kifejlesztettem egy neurális hálózatot, ahol az ODE megoldó helyén egy másik neurális hálózatot alkalmaztam a közönséges differenciálegyenlet megoldására [32]. Ez azonban egy másik hálózat betanítását igényelte előre egy adott egyedre. Ebben a munkában olyan speciális autoenkóder architektúrát fejlesztettem ki, mellyel meg lehet gyorsítani a paraméterek neurális hálózattal történő meghatározását daganatos betegek számára. Továbbá, az algoritmus alkalmas lehet nem csak *in silico* (szimulált adatokon alapuló), de akár *in vivo* (valós méréseken alapuló) adatokból történő betanításra, mivel nem igényli a paraméterek előzetes ismeretét, mert architektúrájából adódóan felügyelet nélküli tanulással tanul. Ezáltal a valósághoz közelebbi eredményt kaphatunk, ha mind szimulált, mind pedig valós adathalmazon tanítjuk a hálózatot.

Az autoenkóderek két komponensből épülnek fel: egy enkóderből, melyek a bemenetet tömörítik egy alacsonyabb dimenziójú térbe, illetve egy dekóderből, mely az enkóder tömörített kimenetéből előállítja a bemenet rekonstrukcióját. A kutatás során egy korábban megalkotott tumordinamikai modellt vettem alapul, melyet a 2. fejezetben ismertettem. A 3. fejezetben kifejtettem a megvalósítás részleteit. Először előtanítottam az autoenkóder hálózatát, hogy élettanilag megfelelő tartományba becsüljön paramétereket. Ehhez az alkalmazott tumormodell által tanító adatokat generáltam numerikus ODE megoldóval. A tanító adatok generálás során virtuális pácienseket hoztam létre, melyeket lementettem. Egy virtuális páciens azt jelentette, hogy véletlenszerűen generáltam a normál protokoll szerint dózisokat és paramétereket, majd kimentettem a szimulált tumortérfoogatokat tartalmazó idősorokat. A létrehozott szimulált adatokkal tanítottam elő a hálózatot, mely az autoenkóder első szakaszát alkotta. Ezt követően az autoenkódert betanítottam a generált adathalmaz második felén is nem felügyelt módon, a paraméterek ismerete nélkül is. A 4. fejezetben három különböző idősor hossza tanítottam be az autoenkódert, majd pedig értékeltem ki. Először két hétre vizsgáltam meg az előrejelzés pontosságát, majd pedig hét hétre, végül pedig 15 hétre. Az eredmények azt mutatták, hogy a hálózat képes volt megtalálni azokat a paramétereket, amelyek leginkább visszaadják a tumortérfogató idősorokat. A hálózat teljesítőképességét előre elkülönített teszt adatokon határoztam meg. Megvizsgáltam a paraméterek eltérését az eredeti értékekhez képest valamint a rekonstruált idősorok eltérését a bemeneti idősorokhoz viszonyítva. Összeségében elmondható, hogy a létrehozott architektúra alkalmas a differenciálegyenletek – esetünkben tumormodell – paramétereinek meghatározására és megtanulására.

A kutatás végső célja egy olyan gyógyszerhordozó rendszer kifejlesztése, mellyel a kemoterápiás gyógyszerek adagolásával a tumor mérete szabályozható, ezáltal a beteg életminősége és élethossza megnövelhető. Egy ilyen eszköz alapja egy olyan algoritmus, mely alkalmas a paraméterek identifikálására. Munkám során egy gépi tanulási algoritmust fejlesztettem ki a paraméterek meghatározásához.

A "Tumormodell paramétereinek meghatározása MI alkalmazásával" projekt nevében köszönetet mondok a HUN-REN Cloud (lásd: Héder et al. 2022; <https://science-cloud.hu/>) használatáért, ami hozzájárult a publikált eredmények eléréséhez [44]. Kutatásom az Innovációs és Technológiai Minisztérium ÚNKP-23-2 kódszámú Új Nemzeti Kiválóság Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs alapról finanszírozott szakmai támogatásával készült.

Irodalomjegyzék

- [1] Freddie Bray, Mathieu Laversanne, Elisabete Weiderpass, and Isabelle Soerjomataram. The ever-increasing importance of cancer as a leading cause of premature death worldwide. *Cancer*, 127(16):3029–3030, 2021.
- [2] Rebecca L Siegel, Kimberly D Miller, Nikita Sandeep Wagle, Ahmedin Jemal, et al. Cancer statistics, 2023. *Ca Cancer J Clin*, 73(1):17–48, 2023.
- [3] Camilla Mattiuzzi and Giuseppe Lippi. Current cancer epidemiology. *Journal of Epidemiology and Global Health*, 9(4):217, 2019.
- [4] Zaigham Abbas and Sakina Rehman. An overview of cancer treatment modalities. In *Neoplasm*. InTech, sep 2018.
- [5] Sudipta Senapati, Arun Kumar Mahanta, Sunil Kumar, and Pralay Maiti. Controlled drug delivery vehicles for cancer treatment and their performance. *Signal Transduction and Targeted Therapy*, 3(1), March 2018.
- [6] Dejene Tolossa Debela, Seke GY Muzazu, Kidist Digamo Heraro, Maureen Tayamika Ndalama, Betelhiem Woldemedhin Mesele, Dagimawi Chilot Haile, Sophia Khalayi Kitui, and Tsegahun Manyazewal. New approaches and procedures for cancer treatment: Current perspectives. *SAGE Open Medicine*, 9:205031212110343, January 2021.
- [7] Flavia Laffleur and Valerie Keckeis. Advances in drug delivery systems: Work in progress still needed? *International journal of pharmaceutics*, 590:119912, 2020.
- [8] Xiaoqiang Sun and Bin Hu. Mathematical modeling and computational prediction of cancer drug resistance. *Briefings in Bioinformatics*, 19(6):1382–1399, June 2017.

- [9] Angela M. Jarrett, Ernesto A.B.F. Lima, David A. Hormuth, Matthew T. McKenna, Xinzeng Feng, David A. Ekrut, Anna Claudia M. Resende, Amy Brock, and Thomas E. Yankeeelov. Mathematical models of tumor cell proliferation: A review of the literature. *Expert Review of Anticancer Therapy*, 18(12):1271–1286, October 2018.
- [10] Hsiu-Chuan Wei. Mathematical modeling of tumor growth and treatment: Triple negative breast cancer. *Mathematics and Computers in Simulation*, 204:645–659, 2023.
- [11] Dániel András Drexler, Tamás Ferenci, Anna Lovrics, and Levente Kovács. Tumor Dynamics Modeling based on Formal Reaction Kinetics. *Acta Polytechnica Hungarica*, 16:31–44, 2019.
- [12] Dániel András Drexler, Tamás Ferenci, András Füredi, Gergely Szakács, and Levente Kovács. Experimental data-driven tumor modeling for chemotherapy. In *Proceedings of the 21st IFAC World Congress*, pages 16466–16471, 2020.
- [13] Dániel András Drexler, Johanna Sápi, and Levente Kovács. Modeling of tumor growth incorporating the effects of necrosis and the effect of bevacizumab. *Complexity*, 2017:1–10, 2017.
- [14] Johanna Sápi, Levente Kovács, Dániel András Drexler, Pál Kocsis, Dávid Gajári, and Zoltán Sápi. Tumor volume estimation and quasi-continuous administration for most effective bevacizumab therapy. *PloS one*, 10(11):e0142190, 2015.
- [15] Melánia Puskás, Borbála Gergics, Balázs Gombos, András Füredi, Gergely Szakács, Levente Kovács, and Dániel András Drexler. Noise modeling of tumor size measurements from animal experiments for virtual patient generation. In *2023 IEEE 27th International Conference on Intelligent Engineering Systems (INES)*. IEEE, July 2023.
- [16] Xin-She Yang. *Mathematical modeling with multidisciplinary applications*. John Wiley & Sons, Nashville, TN, December 2012.
- [17] Lennart Ljung. *System Identification*, pages 163–173. Birkhäuser Boston, Boston, MA, 1998.
- [18] Adriaan van den Bos. *Parameter estimation for scientists and engineers*. Wiley-Blackwell, Hoboken, NJ, June 2007.

- [19] Levente Kovács, Tamás Ferenci, Balázs Gombos, András Füredi, Imre Rudas, Gergely Szakács, and Dániel András Drexler. Positive impulsive control of tumor therapy—a cyber-medical approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–12, 2023.
- [20] Franz-Georg Wieland, Adrian L. Hauber, Marcus Rosenblatt, Christian Tönsing, and Jens Timmer. On structural and practical identifiability. *Current Opinion in Systems Biology*, 25:60–69, 2021.
- [21] Mate Siket, Gyorgy Eigner, and Levente Kovacs. Sensitivity and identifiability analysis of a third-order tumor growth model. In *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*. IEEE, June 2020.
- [22] Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.
- [23] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [24] Nazanin Ahmadi Daryakenari, Mario De Florio, Khemraj Shukla, and George Em Karniadakis. Ai-aristotle: A physics-informed framework for systems biology gray-box identification. *PLOS Computational Biology*, 20(3):e1011916, March 2024.
- [25] Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, 2022.
- [26] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, May 2021.
- [27] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics-informed neural networks: Where we are and what’s next. *Journal of Scientific Computing*, 92(3), July 2022.

- [28] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- [29] William Bradley and Fani Boukouvala. Two-stage approach to parameter estimation of differential equations using neural odes. *Industrial & Engineering Chemistry Research*, 60(45):16330–16344, 2021.
- [30] Subiksha Selvarajan, Aike Aline Tappe, Caroline Heiduk, Stephan Scholl, and René Schenkendorf. Parameter identification concept for process models combining systems theory and deep learning. In *ECP 2022*. MDPI, June 2022.
- [31] Qiang Yin, Juntong Cai, Xue Gong, and Qian Ding. Local parameter identification with neural ordinary differential equations. *Applied Mathematics and Mechanics*, 43(12):1887–1900, December 2022.
- [32] Lilla Kisbenedek, Melánia Puskás, and Dániel A. Drexler. Indirect supervised fine-tuning of a tumor model parameter estimator neural network. In *IEEE 17th International Symposium on Applied Computational Intelligence and Informatics SACI 2023*, pages 109–116, 2023.
- [33] Lilla Kisbenedek, Melánia Puskás, Levente Kovács, and Dániel A. Drexler. Clustering-based parameter estimation of a tumor model. In *SISY 2023 IEEE 21st International Symposium on Intelligent Systems and Informatics*, pages 43–48, 2023.
- [34] Lilla Kisbenedek, Melánia Puskás, Levente Kovács, and Dániel A. Drexler. Anomaly detection of time series containing tumor volumes. In *ICCC 2024 IEEE 11th International Conference on Computational Cybernetics and Cyber-Medical Systems*, pages 67–72, 2024.
- [35] Bernhard Mehlig. *Machine learning with neural networks*. Cambridge University Press, Cambridge, England, October 2021.
- [36] Charu C Aggarwal. *Neural networks and deep learning*. Springer International Publishing, Cham, Switzerland, 1 edition, September 2018.

- [37] Dor Bank, Noam Koenigstein, and Raja Giryes. *Autoencoders*, pages 353–374. Springer International Publishing, Cham, 2023.
- [38] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 71:599–607, 1986.
- [39] Marten Lienen and Stephan Günnemann. torchode: A parallel ode solver for pytorch. *arXiv preprint arXiv:2210.12375*, 2022.
- [40] Timothy Dozat. Incorporating Nesterov Momentum into Adam. In *Proceedings of the 4th International Conference on Learning Representations*, pages 1–4.
- [41] J C Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, Nashville, TN, 3 edition, August 2016.
- [42] John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.
- [43] Ricky T. Q. Chen. torchdiffeq, June 2021.
- [44] M. Héder, E. Rigó, D. Medgyesi, R. Lovas, Sz. Tenczer, F. Török, A. Farkas, M. Emődi, J. Kadlecsek, Gy. Mező, Á. Pintér, and P. Kacsuk. The past, present and future of the ELKH cloud. *Információs Társadalom*, 22(2):128, aug 2022.

Ábrák jegyzéke

3.1. A paraméterbecslő autoenkóder felépítése. Az autoenkóder a bemeneti tumortérfo- gatok és a becsült tumortérfogatok közötti különbség minimalizálásával határozza meg a bemeneti tumortérfogatokhoz tartozó paraméterek értékeit.	19
3.2. A tanító adatok generálása során alkalmazott mérési elrendezés. Az alkalmazott protokoll alapján hetente kétszer, kedden és pénteken történik injekciózás, míg hét- köznapi ponton történik tumortérfogatok mérés.	20
3.3. A neurális hálózat előtanítása során kapott hibák a tanító adathalmaz esetén, ami a teljes adathalmaz 70%-a (kék) és a validációs adathalmaz esetén, ami a teljes adathalmaz 15%-a (narancssárga). A hibák a becsült paraméterek és az eredeti pa- raméterek normalizált értékei közötti átlagos négyzetes eltérést mutatják.	22
3.4. A neurális hálózat előtanítását követően kapott paraméterek (a , b , n , w , ED_{50} , c , k_1 , k_2) értékeinek gyakoriságai a tesztelésre elkülönített adatokon.	23
3.5. Az autoenkóder első felének, a bemenetének és a neurális hálózatának a részletezett ábrája. A hálózat bemenete a tumortérfogatok és a dózisokat tartalmazó idősorok.	24
3.6. Az autoenkóder dekóder része, mely a paraméterekből, kezdeti értékekből és dózi- sokból előállítja a szimulált tumortérfogatot. A szimulált tumortérfogatok és a valós tumortérfogatok közötti átlagos négyzetes eltérés értéke alapján tanítjuk az enkóder neurális hálózatát.	27
4.1. Az autoenkóder három tanulási görbéje a három intervallumra. Mivel a hálózat nagy mértékben tanult minden kötegen történő hibavisszaterjesztés után, így a ha- gyományos iterációk (epoch) helyett a hibavisszaterjesztések számát ábrázoltam a vízszintes tengelyen.	30

4.2.	Példa a validációs adathalmazból a hálózat tanulása alatti teljesítmény változásra két hetes időintervallumon. A tanítás folyamata során kezdetben nagyobb hibával, majd ahogy tanult a hálózat egyre kisebb hibával becsülte meg a tumortérfogatokat.	31
4.3.	Példa a validációs adathalmazból a hálózat tanulása alatti teljesítményváltozásra hét hetes időintervallumon. A tanítás folyamata során kezdetben nagyobb hibával, majd, ahogy tanult a hálózat egyre kisebb hibával becsülte meg a tumortérfogatokat.	32
4.4.	Példa a validációs adathalmazon történő előrejelzésre a hálózat tanulása során a 105 napos intervallumon.	34
4.5.	A tesztadatokon a paraméterek eltérései az eredeti értékektől mind a három intervallum esetén.	35