

Integracija dinamičnih sistemov

4. domača naloga pri predmetu *Praktikum strojnega učenja v fiziki*

Domen Vaupotič

1 Strojno učenje in dinamični sistemi

Za reševanje diferencialnih enačb obstaja kopica raznovrstnih numeričnih metod. Reševanja navadnih diferencialnih enačb se običajno lotimo z metodami integracije (npr. Runge-Kutta), medtem ko se reševanje parcialnih diferencialnih enačb običajno prevede na sistem reševanja velikega in redkega sistema enačb. Ker so (globoke) nevronske mreže sposobne aproksimirati poljubno funkcijo, si lahko zamislimo, da bi bile sposobne tudi reševanja diferencialnih enačb. Osnovna ideja je torej uporabiti nevronske mreže kot integratorje:

$$(\mathbf{X}(t - k\Delta t), \dots, \mathbf{X}(t)) \longrightarrow \mathbf{X}(t + \Delta t).$$

Nevronsko mrežo naučimo na učni množici, ki jo tvorimo z numeričnim integratorjem, seveda pa je osrednji namen ideje v tem, da nevronske mreže uporabimo na dinamičnih sistemih, za katere ne znamo zapisati točnega sistema enačb (ali jih ne znamo učinkovito rešiti), pri čemer mrežo tako učimo na meritvah.

V nalogi se bomo ukvarjali z dvema dinamičnima sistemoma: v prvem delu se bomo posvetili Lorenzovemu sistemu '63, ki ga znamo reševati z numeričnimi integratorji, v drugem delu pa se bomo ukvarjali z napovedovanjem vremena na Zemlji.

2 Lorenzov sistem '63

Lorenzov sistem '63 je sistem treh sklopljenih diferencialnih enačb z zanimivimi lastnostmi. Je nelinearen, neperiodičen, determinističen, a zelo občutljiv na spremembe v začetnih pogojih. Pravimo, da izkazuje lastnosti kaosa, saj ima pozitivne koeficiente Ljapunova, dve bližnji trajektoriji pa se z razvojem eksponentno oddaljujeta. Sistem sestavljajo tri enačbe:

$$\begin{aligned}\dot{X} &= \sigma(Y - X) \\ \dot{Y} &= X(r - Z) - Y \\ \dot{Z} &= XY - bZ,\end{aligned}$$

v katerih nastopajo konstante σ , b in r . Vrednosti parametrov ne bomo spreminjali in bodo v celotni nalogi zavzeli vrednosti $\sigma = 10$, $b = 8/3$ in $r = 28$. Za začetek izračunajmo nabor stacionarnih točk. Rešimo sistem enačb $\mathbf{X} = 0$ in pridemo stacionarno točko:

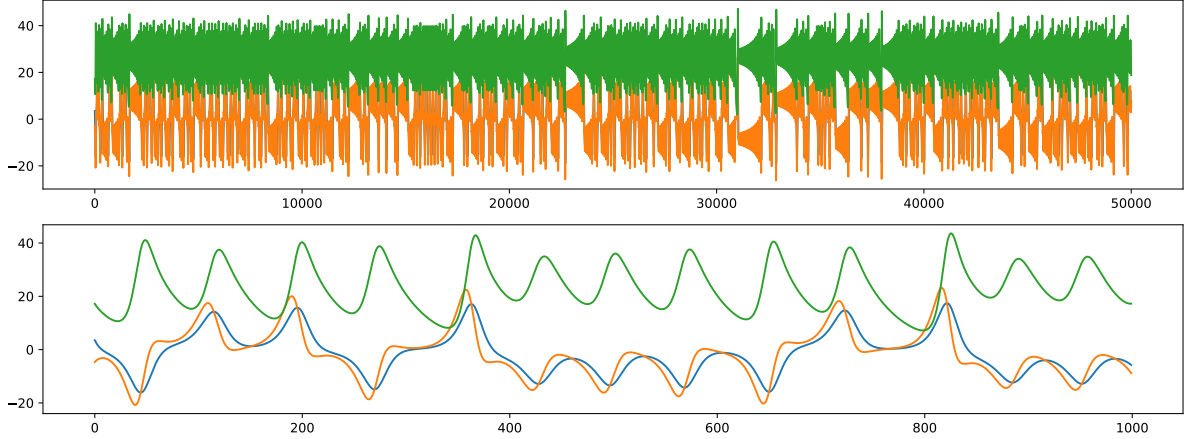
$$\begin{aligned}X^* &= \sqrt{b(r - 1)} \\ Y^* &= \sqrt{b(r - 1)} \\ Z^* &= r - 1\end{aligned}$$

V primeru zgoraj izbranih parametrov torej:

$$(X^*, Y^*, Z^*) = (6\sqrt{2}, 6\sqrt{2}, 27).$$

3 Metodologija

Lorenzov sistem smo reševali z numerično integracijo z metodo RK45 s korakom $\Delta t = 10^{-2}$. Za učenje nevronske mreže smo tvorili trajektorijo, dolgo 500 s (torej 5×10^4 zaporednih točk), ki je prikazana na sliki 1.



Slika 1: Učna trajektorija: celotna (zgoraj) in prvih 10 s (spodaj).

Za integriranje z nevronske mreže smo preizkusili več različnih arhitektur, in sicer:

★ navadna

$$\mathbf{X}(t) \longrightarrow \mathbf{X}(t + \Delta t)$$

★ difference

$$\mathbf{X}(t) \longrightarrow \mathbf{X}(t + \Delta t) - \mathbf{X}(t)$$

★ večji korak

$$\mathbf{X}(t) \longrightarrow \mathbf{X}(t + k\Delta t)$$

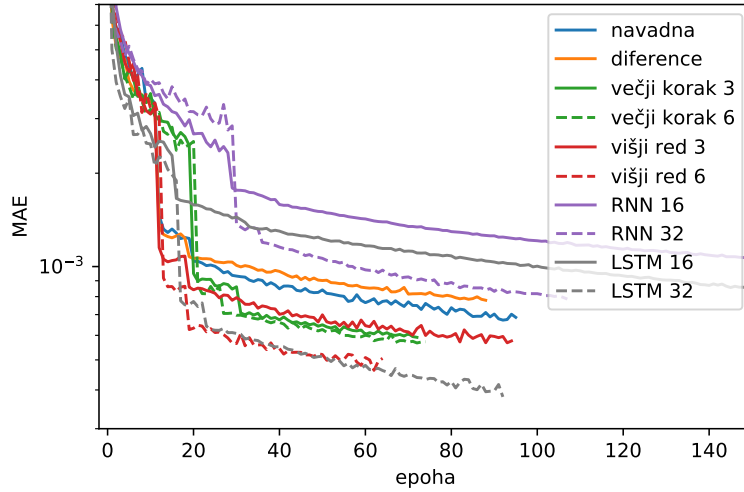
★ višji red

$$(\mathbf{X}(t), \mathbf{X}(t - \Delta t), \dots, \mathbf{X}(t - k\Delta t)) \longrightarrow \mathbf{X}(t + \Delta t)$$

★ rekurentna nevronska mreža (RNN)

★ nevronska mreža z dolgim kratkoročnim spominom (LSTM)

Prve štiri mreže so sestavljene iz gosto povezanih zaporednih plasti, RNN in LSTM pa napovedujeta enako kot mreža višji red. Vse gosto povezane nevronske mreže so bile sestavljene iz štirih zaporednih plasti s 100 nevroni in izhodne plasti s 3 nevroni. Aktivacijska funkcija na plasteh je bila ReLU, na izhodni plasti pa sigmoidna. Nevronske mreže so tako vsebovale približno 31 000 učljivih parametrov. Učenje je potekalo z optimizatorjem Adam, cenovno funkcijo MAE in zmanjševalnikom hitrosti učenja (ReduceLROnPlateau), in sicer največ 200 epoh. Za validacijo med učenjem je bilo uporabljenih 10 % vhodnih podatkov. Podatki so bili tudi predhodno transformirani z metodo MinMaxScaler. Potek učenja je prikazan na sliki 2.



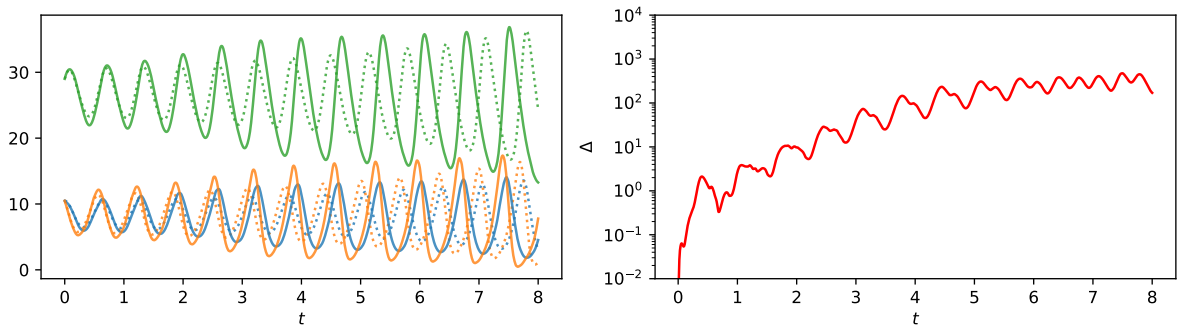
Slika 2: Potek učenja nevronske mreže.

4 Rezultati

Oglejmo si za začetek napovedi dveh izmed mrež, in sicer navadne mreže in mreže LSTM z 32 enotami. Napovedi so prikazane na slikah 3–6, in sicer za začetne pogoje v bližini in v oddaljenosti od stacionarne točke. Izračunana je napaka napovedi kot

$$\Delta(t) = \left| \mathbf{X}(t) - \widetilde{\mathbf{X}}(t) \right|^2,$$

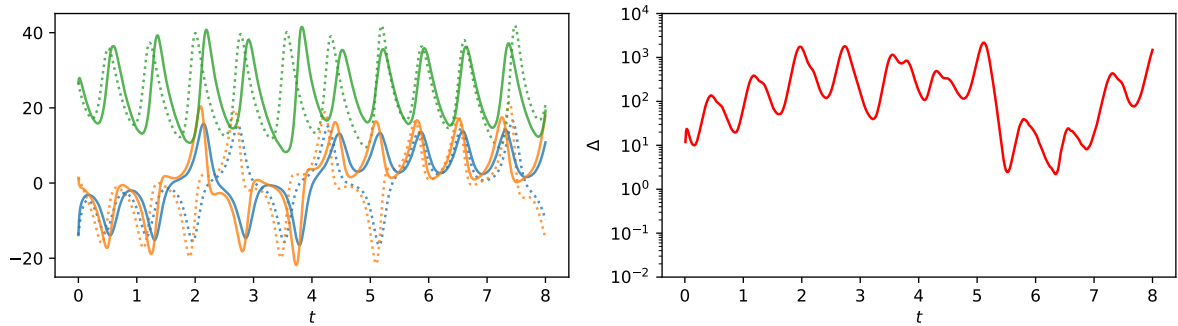
pri čemer $\widetilde{\mathbf{X}}(t)$ označuje trajektorijo, izračunano z numeričnim integratorjem. Opazimo, da mreža LSTM32 deluje bistveno bolje od navadne mreže, prav tako opazimo, da se obe bolje odrežeta v okolici stacionarne točke.



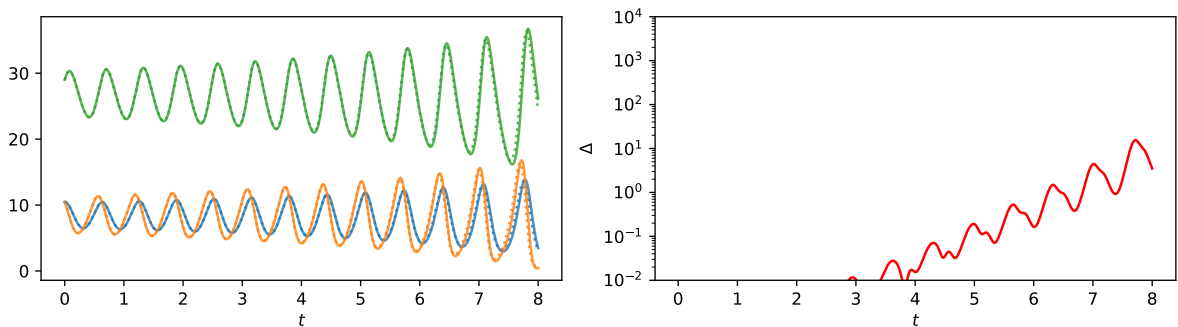
Slika 3: Napoved navadne mreže (polne črte) v bližini stacionarne točke, primerjana z integratorjem (črtkane črte).

Ovrednotimo sedaj vse tipe mrež, in sicer na $N = 1000$ različnih začetnih točkah. Napako napovedi bomo izračunali kot:

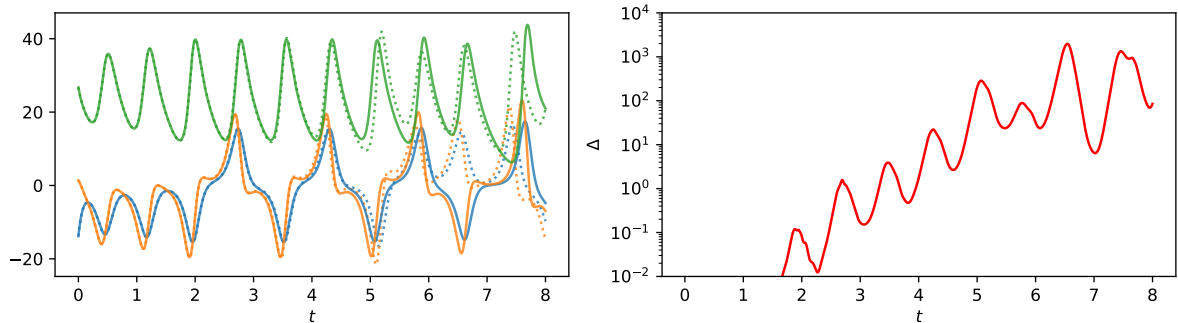
$$E(t) = \sqrt{\frac{1}{N} \sum_{i=1}^N \left| \mathbf{X}(t) - \widetilde{\mathbf{X}}(t) \right|^2}.$$



Slika 4: Napoved navadne mreže v oddaljenosti od stacionarne točke.



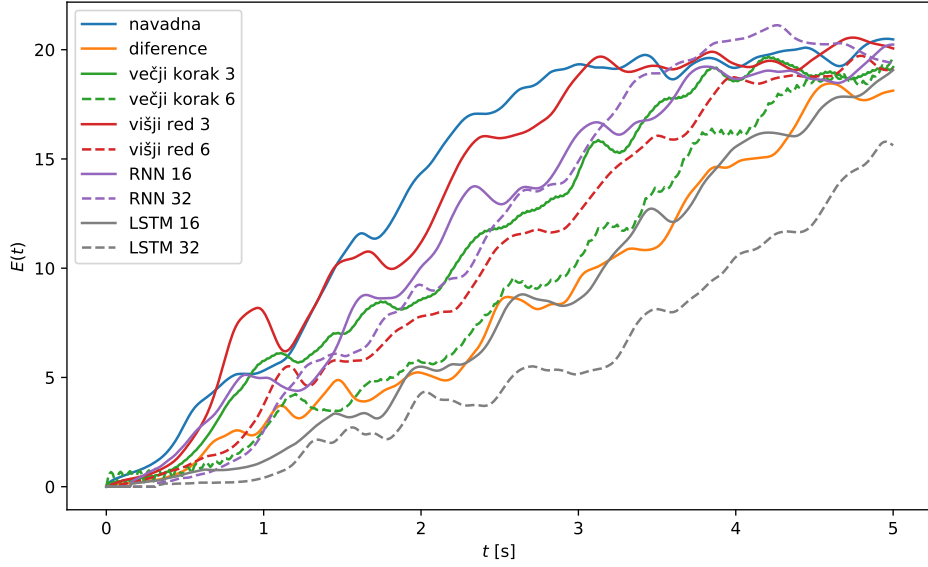
Slika 5: Napoved mreže LSTM32 v bližini stacionarne točke.



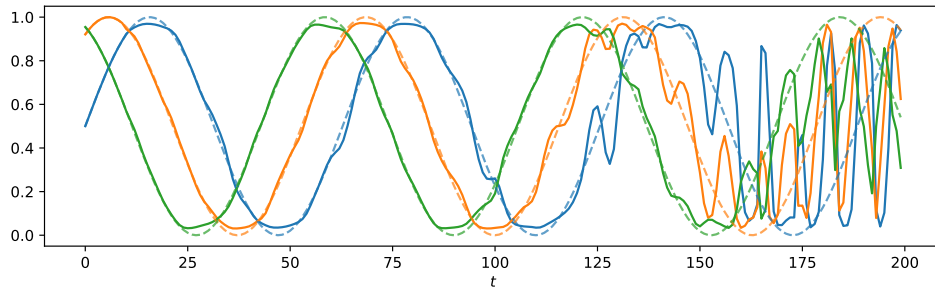
Slika 6: Napoved mreže LSTM32 v oddaljenosti od stacionarne točke.

Rezultati so prikazani na sliki 7. Opazimo, da se je najboljše odrezala mreža LSTM32, najslabše pa navadna mreža. Največja zanimivost je uspešnost mreže, ki napoveduje difference, saj se je izkazala kot druga najboljša oziroma praktično enako dobra kot LSTM16! Glede na ta rezultat bi bilo smiselno razmisliti, ali je bila uporaba metode MinMaxScaler pri predprocesiranju podatkov smiselna, saj je morda povzročila nereprezentativno transformacijo izhodnih podatkov.

Pri testiranju delovanja mreže RNN sem preveril tudi, kako dobro se mreža nauči navadne sinusoide. Rezultati so prikazani na sliki 8. Opazimo, da napoved dobro sledi približno dve periodi, nato pa začne dajati povsem nesmiselne rezultate.



Slika 7: Časovni potek napake napovedi, $E(t)$, za različne arhitekture nevronske mreže.



Slika 8: Napoved mreže RNN (polna črta), ki se je učila na sinusni trajektoriji (črtkana črta).

5 Meteorološka napoved

V drugem delu naloge smo se ukvarjali z napovedovanjem vremena na Zemlji. Čeprav lahko atmosfersko dinamiko zapišemo s sistemom nelinearnih parcialnih diferencialnih enačb, je njihovo reševanje zapleteno, saj izkazujejo močno kaotične lastnosti, posledično pa so potrebni tudi znatna računska moč ter dobre meritve. V nalogi bomo obravnavali meteorološke podatke iz zbirke ERA5, in sicer 41 let podatkov med letoma 1979 in 2019 s šesturno časovno ločljivostjo. Omejili se bomo na geopotencial Φ pri pritiskovi ploskvi 500 hPa. Ta meteorološka spremenljivka je definirana z integralom temperature med gladino morja (pri tlaku p_s) in končnim tlakom p :

$$\Phi(p) \equiv R \int_{p_s}^p T(p') d \ln p' = gz(p).$$

6 Metodologija

Za napovedovanje vremena smo uporabili konvolucijski 'avtoenkoder'. Ta je kot vhodni podatek sprejel matriko geopotenciala velikosti 29×64 ob času t in vrnil matriko

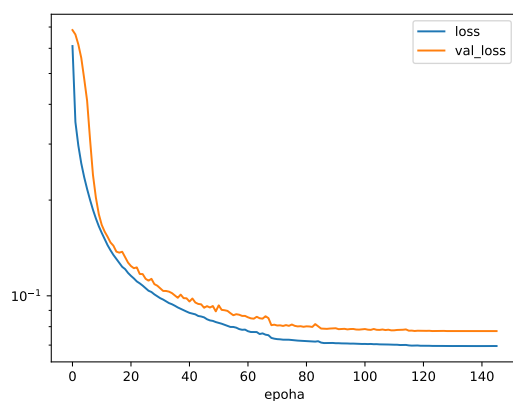
geopotenciala ob času $t + 6$ h. Struktura nevronske mreže je bila:

Layer (type)	Output Shape	Param #
=====		
(ZeroPadding2D)	(None, 30, 64, 1)	0
(Conv2D)	(None, 30, 64, 32)	320
(BatchNormalization)	(None, 30, 64, 32)	128
(MaxPooling2D)	(None, 15, 32, 32)	0
(Conv2D)	(None, 15, 32, 64)	18496
(BatchNormalization)	(None, 15, 32, 64)	256
(MaxPooling2D)	(None, 5, 8, 64)	0
(Conv2D)	(None, 5, 8, 128)	73856
(BatchNormalization)	(None, 5, 8, 128)	512
(UpSampling2D)	(None, 15, 32, 128)	0
(Conv2DTranspose)	(None, 15, 32, 64)	73792
(BatchNormalization)	(None, 15, 32, 64)	256
(Conv2DTranspose)	(None, 15, 32, 32)	18464
(BatchNormalization)	(None, 15, 32, 32)	128
(UpSampling2D)	(None, 30, 64, 32)	0
(Conv2DTranspose)	(None, 30, 64, 1)	289
(Cropping2D)	(None, 29, 64, 1)	0
=====		
Total params: 186,497		
Trainable params: 185,857		
Non-trainable params: 640		

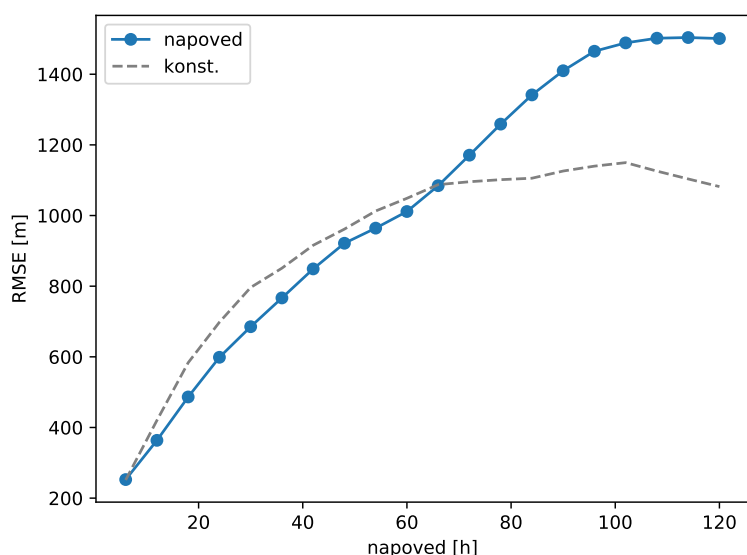
Mreža je torej najprej vhodni matriki dodala prazno vrstico, da je pridelala matriko sode velikosti (30×64) , nato je sledilo zaporedje treh konvolucijskih plasti z normalizacijo in združevanjem po največjem elementu, kar je vodilo v latentni tenzor velikosti $5 \times 8 \times 128$. Sledil je dekoder, sestavljen iz plasti za razširitev slike in 'dekonvolucijskih' plasti. V zadnji plasti se matrika obreže nazaj na velikost 29×64 . V vseh plasteh je bila uporabljena aktivacijska funkcija ReLU, le v zadnji dekonvolucijski plasti linearna funkcija, kot je to običajno pri avtoenkoderjih. Velikost konvolucijskih jeder je bila $(3, 3)$, združevanje pa je potekalo z velikostjo $(2, 2)$. Mrežo smo učili z optimizatorjem Adam z začetno hitrostjo učenja 10^{-3} in cenovno funkcijo MSE. Geopotencial je bil predhodno tudi periodiziran v smeri vzporednikov (ponovljen prvi stolpec na desnem koncu) ter transformiran z metodo StandardScaler. Za učenje mreže so bili uporabljeni meteorološki podatki iz obdobja 1979–2018, za testiranje mreže pa podatki iz leta 2019.

7 Rezultati

Na sliki 9 je prikazan potek učenja nevronske mreže. Slika 10 prikazuje napako meteorološke napovedi, ki je primerjana z napako, kot bi jo dobili, če bi ohranili konstantno meteorološko sliko. Opazimo, da se nevronska mreža v splošnem slabo odreže, saj je le nekoliko boljša od konstantne napovedi, po 60 urah pa postane celo slabša. Na sliki 11 so prikazani še napovedani geopotenciali ter razlike med meritvami in napovedmi. Opazimo, da so absolutne napake najmanjše v bližini ekvatorja, saj so tam tudi spremembe najmanjše, medtem ko je napoved slabša na zgornji in spodnji polobli, saj je meteorologija tam bolj spremenljiva. Za boljšo napoved bi morali vključiti več meteoroloških polj, verjetno pa bi se mreža bolje odrezala, če bi napovedovala zgolj diferenco.



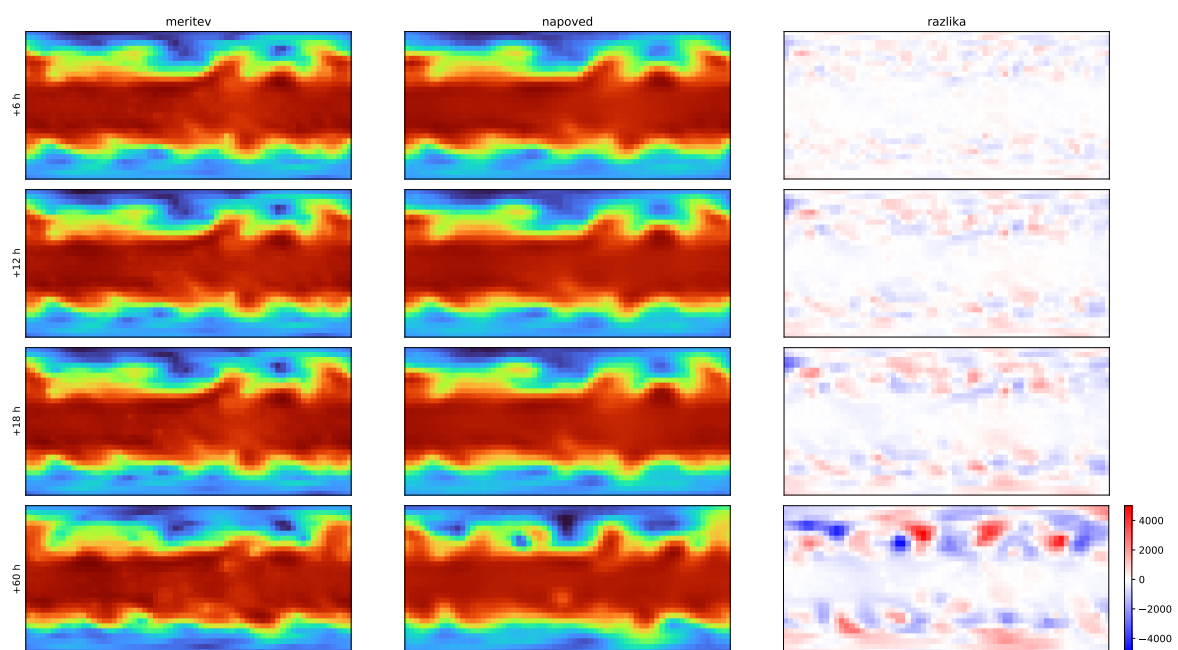
Slika 9: Potek učenja konvolucijske nevronske mreže za meteorološko napoved.



Slika 10: Časovna napaka napovedi geopotenciala za leto 2019, primerjana s konstantno napovedjo.

8 Zaključek

V nalogi smo se ukvarjali s kaotičnimi dinamičnimi sistemi, ki smo jih reševali z metodami strojnega učenja. Ogledali smo si preprostejši primer Lorenzovega sistema '63, ki ga znamo eksplicitno zapisati in rešiti z numeričnimi integratorji, ter zapletenejši primer meteorologije, pri katerem smo nevronske mreže učili na dejanskih meritvah. Pri Lorenzovem sistemu se je najbolje odrezala mreža z dolgim kratkoročnim spominom, meteorološka napoved pa je bila le neznatno boljša od konstantne napovedi.



Slika 11: Meritve in napovedi geopotenciala za leto 2019 ter razlika med obema.