

# Modeliranje dinamike fizikalnih sustava naprednim metodama dubokog učenja

---

Filip Mirković

Mentor: izv. prof. dr. sc. Davor Horvatić

Prirodoslovno matematički fakultet, Fizički odjek  
Bijenička cesta 28, 10000 Zagreb



# Dinamički sustavi i opis problema

---

## Definicija

Dinamički sustav je uređena trojka prostora stanja  $\mathcal{S}$ , vremenskim skupom  $\mathcal{T}$  i funkcije

$$\mathcal{F} : \mathcal{S} \times \mathcal{T} \longrightarrow \mathcal{S} \quad (1)$$

- Dinamički sustav je kontinuran ako vrijedi:  $\mathcal{T} \subseteq \mathbb{R}$
- U tom slučaju  $\mathcal{F}$  je obična ili parcijalna diferencijalna jednačba

Evolucija stanja  $\phi \in \mathcal{S}$  dana je

$$\partial_t \phi = \mathcal{F}_\eta(\mathbf{x}, \phi, \nabla_x \phi, \nabla_x^2 \phi, \phi \nabla_x \phi, \dots) \quad (2)$$

- $\eta$  je vektor parametara jednačbe. Npr.  $\omega$  u harmoničkom oscilatoru.

# Opis problema

## Želimo:

Za svaki  $\phi_0 \in \mathcal{S}$  poznavati buduće  $\phi_t$  za proizvoljno velike  $t$ .

## Priječi nas:

1.  $\mathcal{F}$  je nelinearna
2. Dinamika je kaotična
3.  $\mathcal{S}$  je visokodimenzionalan
4.  $\mathbf{x}$  je visokodimenzionalan
5.  $\mathcal{F}$  je djelomično ili potpuno nepoznat
6. Sve od navedenog

## Sretni smo ako:

Za veliki broj  $\phi_0$  uspijevamo precizno procijeniti  $\phi_t$  do nekog konačnog  $T \in \mathcal{T}$ .

Iznos  $T$  određen je pojedinim problemom.

## Konstruckija minimalnog modela

- Rješenje nije skupo
- Temelji se na teoriji, iskustvu i podacima
- Model opisuje samo općenite značajke dinamike
- Numerika je i dalje potrebna

## Numeričko rješavanje diferencijalnih jednažbi

- Dinamika treba biti potpuno poznata
- Proces je skup
- Visoka dimenzionalnost prostora postaje velika prepeka

## Modeliranje iz podataka

Nije potrebno poznavati dinamiku

Često je potrebno puno podataka

# Transformerski model i mehanizam pozornosti

---

## Uvod

- Originalno nastali 2017. u području obrade prirodnog jezika (NLP) [14]
- Pokazali se izuzetno uspješnim, npr. GPT3 chatbot je transformer
- Brzo se šire na ostala područja strojnog učenja, npr. računalni vid

## Od prevođenja teksta do modeliranja dinamike

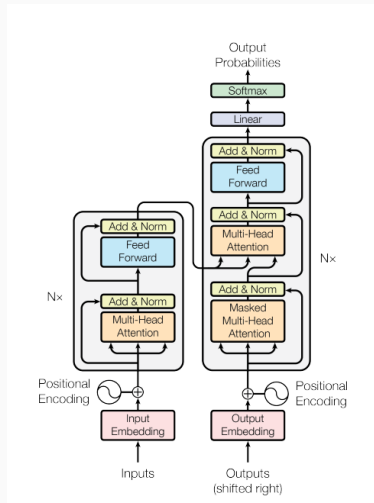
- U obradi prirodnog jezika riječi se često reprezentiraju vektorima
- Uz vektorske reprezentacije prevođenje teksta postaje nastavljanje niza vektora
- U računalima dinamika fizikalnih sustava je diskretizirana i u formi niza vektora

# Transformerski model - arhitektura

- Koder-Dekoder struktura
- Koder lijevo, dekodeo desno
- U modeliranju dinamike koristimo samo dekodeo

## Pozornost

Ključna komponenta transformera je višestruka pozornost (eng. Multi-Head Attention)



Slika 1: Arhitektura originalnog transformera. Preuzeto is [14].



## Motivacija

Neka je  $\{\xi_1, \dots, \xi_n\}$   $n$  opaženi niz vektora stanja koji predstavlja evoluciju sustava kroz vrijeme.

- Ako je dinamika poznata (i možemo baratati jednadžbama), jedina informacija koju trebamo za nastavak vremenskog niza je  $\xi_n$
- Ako ne znamo dinamiku korisno je modelu dati informaciju o  $c$  prethodnih stanja
- Time dobivamo kontekstni interval  $\mathcal{C} = \{\xi_{n-c}, \dots, \xi_n\}$
- Iz kontekstnog intervala konstruiramo kontekstni vektor  $\mathbf{c}$  kojeg pružamo modelu kao dodatnu informaciju o protekloj evoluciji

**Ideja:** Kontekstni vektor  $\mathbf{c}$  sadrži informaciju o dalekosežnim vremeskim ovisnostima.

# Mehanizam pozornosti [2] - Pojednostavljena pozornost

Pojednostavljena pozornost - recept:

1.) Računamo sličnost između  $\xi_n$  i svih vektora  $\xi_l \in \mathcal{C} \subset \mathbb{R}^d$ . Mjera sličnosti je *normirani* skalarni produkt

$$s_l = \frac{\xi_n \cdot \xi_l}{\sqrt{d}} \quad (3)$$

2.) Iz sličnosti dobivamo koeficijente pozornosti

$$\alpha_l = \text{softmax}(\mathbf{s})_l = \frac{e^{s_l}}{\sum_{i=1}^c e^{s_i}} \quad (4)$$

3.) Kontekstni vektor je

$$\mathbf{c} = \sum_{l=1}^c \alpha_l \xi_l \quad (5)$$

Ostatku modela se pružaju informacije o  $\xi_n$  i  $\mathbf{c}$  za stvaranje predikcije o  $\xi_{n+1}$ .

# Mehanizam pozornosti [3] - Potpuna pozornost

Unapređenja pojednostavljene pozornosti:

Uvodimo tri neuronske mreže  $\mathcal{F}_v$ ,  $\mathcal{F}_q$ ,  $\mathcal{F}_k$  takve da

$$\mathcal{F}_v(\xi_j) = \mathbf{v}_j \in \mathbb{R}^d \quad (6)$$

$$\mathcal{F}_q(\xi_i) = \mathbf{q}_i \in \mathbb{R}^{d_k} \quad (7)$$

$$\mathcal{F}_k(\xi_i) = \mathbf{k}_i \in \mathbb{R}^{d_k} \quad (8)$$

Veličine  $\mathbf{v}$ ,  $\mathbf{k}$ ,  $\mathbf{q}$  zovemo vrijednost, *ključ* i *upit*. Računamo ih za svaki  $\xi \in \mathcal{C}$ .

Ostatku modela pružaju se informacije o  $\mathbf{v}_n$  i  $\mathbf{c}$  za stvaranje predikcije o  $\xi_{n+1}$ .

Potpuna pozornost - recept:

1.) Sličnosti računamo između svih ključeva i zadnjeg upita  $\mathbf{q}_n$ .

$$s_l = \frac{\mathbf{q}_n \cdot \mathbf{k}_l}{\sqrt{d_k}} \quad (9)$$

2.) Koefficijente pozornosti računamo na isti način

$$\alpha_l = \text{softmax}(\mathbf{s})_l = \frac{e^{s_l}}{\sum_{i=1}^c e^{s_i}} \quad (10)$$

3.) Kontekstni vektor računamo u prostoru *vrijednosti*

$$\mathbf{c} = \sum_{i=1}^c \alpha_i \mathbf{v}_i \quad (11)$$

Zašto uvodimo  $\mathcal{F}_v$ ,  $\mathcal{F}_k$  i  $\mathcal{F}_q$ ?

- Dodavanjem mreža povećavamo ekspresivnost modela.
- Želja nam je da mreže nauče bitne značajke stanja  $\xi$ .
- Značajke su reprezentirane vektorima  $\mathbf{q}$ ,  $\mathbf{k}$  i  $\mathbf{v}$ .
- Bez korištenja  $\mathcal{F}_v$ ,  $\mathcal{F}_k$  i  $\mathcal{F}_q$ , sličnost pridodaje svim komponentama  $\xi$  jednaku važnost.

### Primjer:

Promotrimo gibanje točkaste čestice u potencijalu 2-D harmoničkog oscilatora

$$V(x, y) = ax^2 + \epsilon y^2$$

Neka je  $\epsilon \ll a$  te vektor stanja  $\xi = (x, y, \dot{x}, \dot{y})$ .

Kada bismo računali mjere sličnosti između dvaju vektora, običan skalarni produkt pridodaje jednaku važnost koordinatama  $x$  i  $y$ .

Zbog manje krutosti potencijala u  $y$  smijeru  $y$  koordinata je većeg raspona, međutim dinamika je osjetljivija na perturbacije u  $x$  i  $\dot{x}$ .

Primjerice, uvođenjem linearnih operatora  $\mathcal{F}_v = \mathbb{1}$ ,  $\mathcal{F}_q = \mathcal{F}_k = [w_{ij}]$ , gdje su  $w_{xx} > w_{yy}$  te  $w_{i,j} = 0$  za  $i \neq j$  zaobilazimo ovaj problem.

Najskuplja komponenta izrade dubokog modela je **treniranje**

### Paralelizacija

Tijekom treniranja računamo  $\mathbf{v}$ ,  $\mathbf{q}$  i  $\mathbf{k}$  za svaki  $\xi \in \mathcal{C}$  te iz njih radimo matrice  $V$ ,  $Q$  i  $K$  koje imaju  $c$  redova, jedan za svaki vektor iz  $\mathcal{C}$ .

Sličnosti tada možemo izraziti matričnim množenjem koje se efikasno izvršava na grafičkim karticama.

$$S = \frac{QK^T}{\sqrt{d_R}} \quad (12)$$

Analogno dobivamo matricu pozornosti:

$$A = \text{softmax}(S) \quad (13)$$

Normalizacija u softmax se provodi po stupcima matrice  $S$ .

## Problem:

- Za neki vektor  $\xi_i \in \mathcal{C}$  izračunat ćemo koeficijente pozornosti za sve preostale  $\xi_j$  uključujući i buduće  $\xi_j, j > i$ .
- Ako računamo stanje  $\xi_n$  zadnji vektor na kojeg model smije obratiti pozornost je  $\xi_{n-1}$ .

## Rješenje: Maskiranje

- Sve elemente iznad glavne dijagonale matrice  $S$  postavimo na  $-\infty$  (u praksi koristimo jako neki jako negativan broj)
- Time je koeficijent pozornosti 0 za te elemente.

## Problem

- Tijekom treniranja model obrađuje sve vektore iz  $\mathcal{C}$  istovremeno
- Postojeći vremenski uređaj ovime je narušen

## Kodiranje vremenskog uređaja

- Neka je  $\xi_{pos_i}$   $i$ -ta komponenta vektora na poziciji  $pos$  u kontekstnom intervalu.
- Toj komponenti dodajemo vrijednost  $PE_{pos_i}$

$$PE_{pos_i} = \begin{cases} \sin(pos/10000^{\frac{1}{d}}) & i \text{ je paran} \\ \cos(pos/10000^{\frac{1}{d}}) & i \text{ je neparan} \end{cases} \quad (14)$$



# Funkcija gubitka transformera

Neka je  $\{(\mathcal{C}_i, \tilde{\mathcal{C}}_i) \mid i = 1 \dots D\} = \mathcal{D}$  skup podataka. S time da je  $\mathcal{C}_i = \{\xi_j \mid j = n \dots n + c\}$  ulazni kontekstni interval, a  $\tilde{\mathcal{C}}_i = \{\xi_j \mid j = n + 1 \dots n + c + 1\}$  kontekstni interval kojeg model treba predvidjeti iz  $\mathcal{C}_i$ .

Funkcija gubitka je

$$\mathcal{L} = \sum_{i=1}^D \sum_{j=1}^c \|\xi_j^i - \hat{\xi}_j^i\|_2^2 \quad \xi_j^i \in \tilde{\mathcal{C}}_i \quad (15)$$

Simbol  $\hat{\xi}$  označava sanja koja predviđa model.

# Vektorske reprezentacije

---

# Učenje vektorskih reprezentacija sustava [1]

## Koji sustavi nisu opisani vektorom?

- Sve što je dosada objašnjeno funkcionira ako je stanje sustava  $\phi$  reprezentirano vektorom  $\xi$ .
- Bitna klasa dinamički sustava koji nisu vektori su polja, npr. brzina strujanja fluida, elektromagnetsko polje, valne funkcije...

## Naivan pristup

Recimo da je  $\phi^t(x, y)$  dvodimenzionalno skalarno polje, npr. iznos brzine toka fluida.

U računalu polje je spremljeno u obliku matrice  $\phi_{ij}^t$ , gdje  $i = 1 \dots N$ ,  $j = 1 \dots M$ .

Naivno možemo reprezentirati  $\phi_{ij}^t$  vektorom tako da matricu *razvučemo* po jednoj dimenziji, recmo po stupcima.

Dobivamo vektor  $\vec{\phi}_n^t$ ,  $n = 1 \dots N \cdot M$ , međutim komponente originalne matrice  $\phi_{ij}^t$  i  $\phi_{ij+1}^t$  su u vektoru  $\vec{\phi}^t$  udaljene za  $N$  redova.

# Učenje vektorskih reprezentacija [2]

## Narušena lokalnost

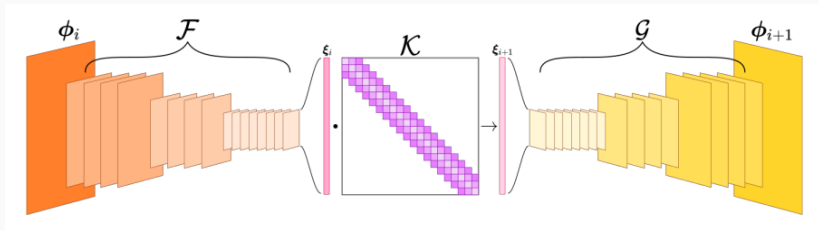
Budći da je evolucija sustava opisana parcijalnom diferencijalnom jednačžbom, na evoluciju  $\phi_{ij}^t$  utječu stanja poput  $\phi^{t,ij+1}$  i  $\phi^{t,i+1,j}$ .

U vektoru  $\vec{\phi}^t$  ta su stanja jako *daleko* jedna od drugog.

## Dinamika na mnogostrukosti

- Osim narušene lokalnosti problem s naivnom vektorskom reprezentacijom jest što je vektorski prostor koji sadrži  $\vec{\phi}$  prevelik.
- Skup svih matrica tipa  $N \times M$  sadrži puno više od rješenja Navier-Stokesovih jednačžbi koje opisuju tok fluida, npr. Matrica kojoj su komponente slučajni brojevi isto je sadržana u  $\mathbb{R}^{N \times M}$ .
- Dinamika danog sustava odvija se na nekoj mnogostrukosti, a ne čitavom  $\mathbb{R}^{N \times M}$ .

$\implies$  Uvodimo neuronske mreže kojima učimo prikladnu vektorsku reprezentaciju sustava iz podataka.



Slika 2: Koder-Koopman-Dekoder struktura. Preuzeto iz [5].

## Koder-Koopman-Dekoder struktura (DKD)

1. Koder  $\mathcal{F}$  preslikava stanje sustava  $\phi \in \mathcal{S}$  u prikladnu vektorsku reprezentaciju  $\xi \in \Xi$
2. Dekoder  $\mathcal{G}$  rekonstruira stanje sustava  $\phi$  iz reprezentacije  $\xi$ . On je inverz kodera.
3. Koopmanov  $\mathcal{K}$  operator uređuje prostor reprezentacija.

## Koder

Za generiranje vektorskih reprezentacija koristimo neuronsku mrežu  $\mathcal{F} : \mathcal{S} \rightarrow \Xi$ .

Čest izbor slojeva koderu su konvolucijski slojevi. Konvolucijski slojevi uče lokalne značajke stanja  $\phi$  [2] [7].

## Dekoder

Nakon što obavimo predikcije nad vektorima iz  $\Xi$  želimo od njih ponovno rekonstruirati stanja sustava.

Za to koristimo dekodeer  $\mathcal{G} : \Xi \rightarrow \mathcal{S}$ .

Kada je to moguće, arhitektura dekodeera je zrcalna arhitekturi koderu.

## Problem

- Ništa ne prijeći sustav kodera i dekodera da nasumično ne razbacaju  $\mathcal{F}(\phi)$  po  $\Xi$
- Želimo da originalna dinaimka bude na neki način prisutna u  $\Xi$
- Problem zaobilazimo tako da stanja  $\phi^t$  i  $\phi^{t+\Delta t}$  preslikamo u bliska stanja  $\xi^t$  i  $\xi^{t+\Delta t}$ , s obzirom na euklidsku normu.

## Koopmanov operator

Stanja  $\phi$  možemo preslikati u beskonačno dimenzionalni vektorski prostor u kojem sukcesivna stanja generiramo grupnim translacijama [10].

$$\xi(t) = e^{Ht}\xi(t_0) \quad (16)$$

Operator  $\mathcal{K}$  zocemo Koopmanovim operatorom. Detalji teorije Koopmanovog operatora su u [3], [8], [1].

Operator translacija u vremenu razvijamo do linearnog člana

$$\xi(t_0 + \Delta t) = (1 + H\Delta t)\xi(t_0) = \mathcal{K}\xi(t_0) \quad (17)$$

Operator  $\mathcal{K}$  zovemo Koopmanov operator. Pretpotavljamo ga u obliku jedinične matrice koja ima korekcije reda  $\mathcal{O}(\Delta t)$ .

Služi za strukturiranje latentnog prostora  $\Xi$ .



## Funkcija gubitka

Funkciju gubitka  $\mathcal{L}_{DKD}$  dijelimo na tri dijela:  $\mathcal{L}_{DKD} = \mathcal{L}_R + \mathcal{L}_D + \lambda\mathcal{L}_{reg}$

### 1. Rekonstrukcijski gubitak

$$\mathcal{L}_R = \sum_{i=1}^D \sum_{j=0}^T \text{MSE}(\phi_j^i, \mathcal{G} \circ \mathcal{F}(\phi_j^i)) \quad (18)$$

Osigurava da su  $\mathcal{F}$  i  $\mathcal{G}$  aproksimativno inverzni.

### 2. Dinamički gubitak

$$\mathcal{L}_D = \sum_{i=1}^D \sum_{j=1}^T \text{MSE}(\phi_j^i, \mathcal{G} \circ \mathcal{K}^j \mathcal{F}(\phi_0^i)) \quad (19)$$

Osigurava da Koopmanov operator generira sukcesivna stanja, odnosno uređuje prostor  $\Xi$ .

### Regularizacijski gubitak

$$\mathcal{L}_{reg} = \|\mathcal{K}\|_2^2 \quad (20)$$

Regularizira oblik Koopmanovog operatora [9].

# Gissingerov sustav

---

## Gissingerov sustav

- Uveden kao minimalna aproksimacija magneto-hidrodinamičkih jednadžbi [6].
- Stanje sustava je opisano trima modovima: dipolnim  $D$ , kvadrupolnim  $Q$  i brzinskim  $V$ .
- Namjenjen da opiše obrtaje polova Zemljinog magnetskog polja - prejednostavan.

## Jednadžbe gibanja

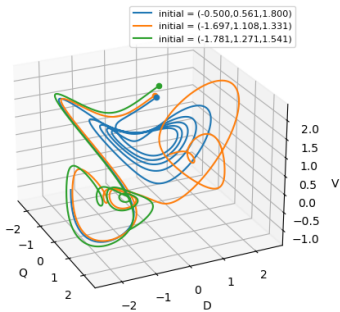
$$\dot{Q} = \mu Q - VD \quad (21)$$

$$\dot{D} = -\nu D + VQ \quad (22)$$

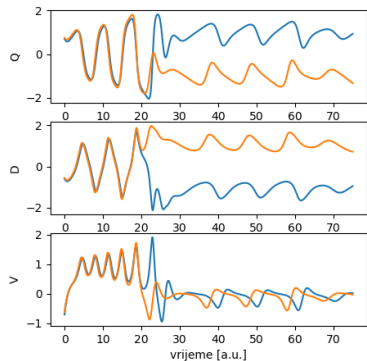
$$\dot{V} = \Gamma - V + DQ \quad (23)$$

Posjeduje bogatu dinamiku koja postaje kaotična za određene parametre  $\mu, \nu, \Gamma$ . Sustav je analiziran u kaotičnom režimu za  $\mu = 0.119, \nu = 0.1, \Gamma = 0.9$

# Gissingerov sustav [2] - Primjer kaosa



**Slika 3:** Bogata dinamika Gissingerovog sustava. Narandžasta krivulja doživljava jedan kaotičan obrtaj dipolnog moda  $D$ .



**Slika 4:** Dvije krivulje počinju iz jako bliskih početnih uvijeta te naglo divergiraju jedna od druge.

# Gissingerov sustav [3] - Lyapunovljev eksponent

## Lyapunovljev eksponent

- Daje mjeru brzine divergencije bliskih trajektorija
- Za kaotične atraktore brzina divergencije je eksponencijalna
- Koristimo ga za ocjenu kvalitete modela

## Kratak izvod

Neka je stanje sustava definirano varijablama  $x_\mu$ , a njihova evolucija

$$\dot{x}_\mu = f_\mu(\mathbf{x})$$

Tajektoriji  $\mathbf{x}$  dodajemo infinitezimalu smetnju  $\delta\mathbf{x}$ .

$$\dot{x}_\mu + \delta\dot{x}_\mu = f_\mu(\mathbf{x} + \delta\mathbf{x})$$

Razvojem u red dobivamo jednadžbu za evoluciju smetnje  $\delta\mathbf{x}$

$$\delta\dot{x}_\mu = \sum_{\nu} \partial_{\nu} f_{\mu}(\mathbf{x}) \delta x_{\nu} \quad (24)$$

Jakobijan  $[\partial_{\nu} f_{\mu}]$  izračunat je duž nesmetane trajektorije.

## Gissingerov sustav [4] - Lyapunovljevo vrijeme

Integriramo li jednadžbu (24) divergenciju trajektorija ocjenjujemo veličinom

$$d(t) = \sqrt{\sum_{\nu} \delta x_{\nu}(t)^2}$$

Lyapunovljev eksponent definiramo

$$\lambda = \lim_{t \rightarrow \infty} \lim_{d(0) \rightarrow 0} \frac{1}{t} \log \left( \frac{d(t)}{d(0)} \right)$$

Lyapunovljevo vrijeme definirano je kao inverz Lyapunovljeva eksponenta i daje vremensku skalu na kojoj sustav postaje kaotičan [13] [12]. Za Gissingerov sustav smo Lyapunovljevo vrijeme procijenili numerički postupkom definiranim u [11].

$$\tau_L = 13.4$$

## Rezultati

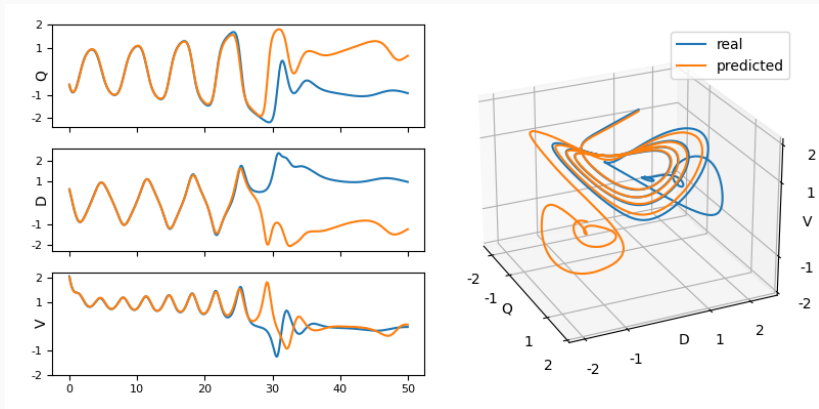
---

## Rezultati

- Transformer uspjeva naučiti općenite karakteristike faznog prostora čime sam postaje kaotičan [4].
- Model smatramo uspješnim, ako daje precizne predikcije barem dva Lyapunovljeva vremena.
- Predikcije se za većinu početnih uvjeta poklapaju sa stvarnim rješenjima na vremenskoj skali dva Lyapunovljeva vremena.
- Postoje regije u na atraktoru gdje je model izrazito uspješan ili izrazito neuspješan.
- Na kraju kaos uvijek prevlada.

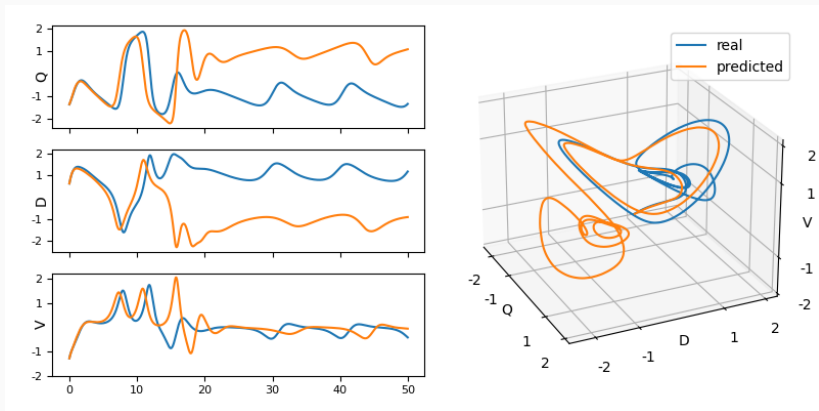


## Rezultati [2] - Tipična uspješnost



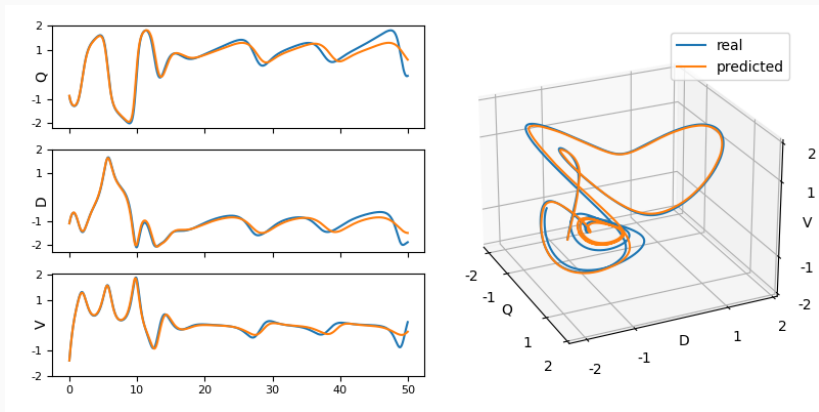
**Slika 5:** Predikcija modela i stvarna dinamika sustava prate se na skali dva Lyapunovljeva vremena. Model zadržava ovakvu preciznost za većinu početnih uvjeta na atraktoru.

## Rezultati [3] - Niska uspješnost



**Slika 6:** Postoje i nazigled nasumične točke u kojima model gubi preciznost iznimno brzo  $\sim 0.5\tau_L$ .

## Rezultati [4] - Velika uspješnost



Slika 7: Iznimno dugotrajna preciznost modela  $\sim 3\tau_L$ .

## Zaključak i diskusija

---

## Prednosti modela

- Arhitekturna transformacija pokazuje se uspješnom u modeliranju dinamike fizikalnih sustava.
- Može se koristiti u situacijama kada sve dinamičke varijable nisu poznate ili osmotrive.

## Mane modela





- Potrebne su velike količine podataka o fenomenu.
- Kaos uvijek prevlada i predikcije postanu neprecizne.
- Loša interpretabilnost

## Unaprjeđenja modela

- Treniranje DKD sustava uz transformer (eng. finetuning)
- Bolje kodiranje vremenskog uređaja
- Ugrađivanje fizikalnih spoznaja o problemu

Hvala na pažnji!  
Pitanja?

---

-  Petar Bevanda, Stefan Sosnowski, and Sandra Hirche.  
**Koopman operator dynamical models: Learning, analysis and control.**  
*Annual Reviews in Control*, 52:197–212, 2021.
-  Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković.  
**Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021.**
-  Akshunna S. Dogra and William T Redman.  
**Optimizing neural networks via koopman operator theory.**  
2020.
-  Jaime Lopez Garcia and Angel Rivero Jimenez.  
**Phase space learning with neural networks, 2020.**





Nicholas Geneva and Nicholas Zabarar.  
**Transformers for modeling physical systems.**  
*Neural Networks*, 146:272–289, feb 2022.



C. Gissinger.  
**A new deterministic model for chaotic reversals.**  
*The European Physical Journal B*, 85(4), apr 2012.



Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville.  
***Deep Learning.***  
MIT Press, Cambridge, MA, USA, 2016.  
<http://www.deeplearningbook.org>.



Di Luo, Jiayu Shen, Rumen Dangovski, and Marin Soljačić.  
**Koopman operator learning for accelerating quantum optimization and machine learning, 2022.**

-  Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G.R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab.  
**A high-bias, low-variance introduction to machine learning for physicists.**  
*Physics Reports*, 810:1–124, may 2019.
-  Igor Mezic.  
**Koopman operator, geometry, and learning, 2020.**
-  Thomas S. Parker and Leon O. Chua.  
***Practical Numerical Algorithms for Chaotic Systems.***  
Springer-Verlag, Berlin, Heidelberg, 1989.
-  Steven H. Strogatz.  
***Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering.***  
Westview Press, 2000.



Michael Tabor.

***Chaos and Integrability in Nonlinear Dynamics- An Introduction.***

Wiley, 1989.



Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin.

**Attention is all you need.**

In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.