

RNF projekt: Detekce chyby

Artyom Voronin, Martin Havelka, Tomáš Kopčil, Jan Bolcek, Jan Hružek

December 14, 2020

Contents

1	Instalace	2
2	Model a generování dat	2
3	Generování korektních a chybných dat	2
4	Statistické zpracování dat	2
5	Dataset	3
6	Neuronová síť	3
7	Výsledky	4
8	Závěr	6
9	Zdroje	6

1 Instalace

- Instalace potřebných balíčků:

```
pip install -r requirements.txt
```

- Instalace PyTorch

2 Model a generování dat

Jako model byl použit jednoduchý mechanický oscilátor skládající se z tělesa, pružiny a tlumiče. Parametry tohoto systému jsou tedy hmotnost m , tuhost pružiny k a tlumení b (podle přiloženého obrázku 1).

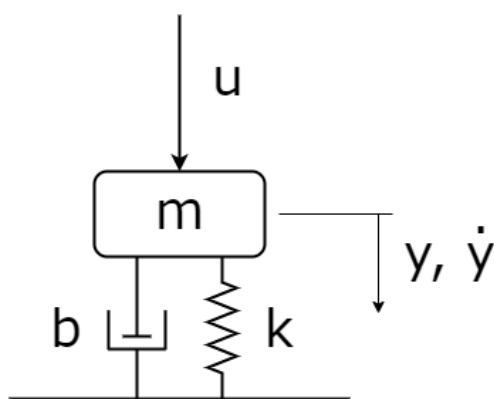


Figure 1: Harmonic oscillator model

Model je popsán diferenciální rovnicí:

$$m\ddot{y} + b\dot{y} + ky = u(t)$$

Oscilátor byl buzen signálem $u(t)$ se skokovým průběhem a měřena byla výchylka a rychlost tělesa.

3 Generování korektních a chybných dat

Pro zmíněný model bylo vygenerováno sto různých kombinací parametrů a pro ně naměřena odezva. Tato data byla označena jako korektní. Jako chybná byla uvažována situace, kdy se v praxi "utrhne" pružina nebo tlumič, tedy $k = 0$ nebo $b = 0$. Dále také, pokud je poměrný útlum soustavy větší, nebo roven jedné, tedy soustava je přetlumená a nedochází ke kmitání.

4 Statistické zpracování dat

Detekce chyby probíhala na základě signálu, jeho statistickém zpracování (signal based fault detection).

Pro každý balík naměřených dat byla zpracována statistická analýza. Určeny byly následující statistické parametry:

- minimum
- maximum
- aritmetický průměr
- medián
- standardní odchylka
- rozptyl
- RMS
- Fourierova transformace pomocí FFT algoritmu a následně vybrány 3 nejvíce dominantní frekvence.

Tyto parametry byly zabaleny společně s označením (label), zda se jedná o chybná, nebo korektní data, a následně použita jako dataset pro neuronovou síť. Pro práci s daty byla využita struktura knihovny *numpy*, které je následně Pytorch schopen zkonvertovat do svého formátu.

5 Dataset

Ze statisticky zpracovaných dat byl vytvořen dataset, který odpovídá vstupům neuronové sítě. Jedná se o tensor, který obsahuje hodnoty features (statistické parametry) a labels (označení správných a chybných dat, 1/0). Následně byl dataset rozdělen na trénovací a validační data v poměru 80% ku 20%. Takto rozdělený dataset byl dále použit v neuronové síti.

6 Neuronová síť

Pro vytvoření neuronové sítě byl použit nástroj PyTorch. Byl vytvořen model s jednou vstupní, skrytou a výstupní vrstvou.

Velikost vrstev:

Vstupní vrstva: 20
 Skrytá vrstva: 16
 Výstupní vrstva: 1

Parametry sítě:

Aktivační funkce: sigmoid
 Optimizátor: Adam
 Loss function: MSELoss
 Learning rate: 0,05
 Počet epoch: 500
 Velikost batch: 5

Trénování neuronové sítě tedy probíhalo v závislosti na velikosti batch a na počtu epoch. Výsledky trénování jsou zobrazeny v následující části.

7 Výsledky

Následující grafy představují chování neuronové sítě při známých vstupních a neznámých vstupních datech (na kterých nebyla naučena). Znázorňují postupné ustálení odchylky s narůstajícím počtem epoch 2, 3.

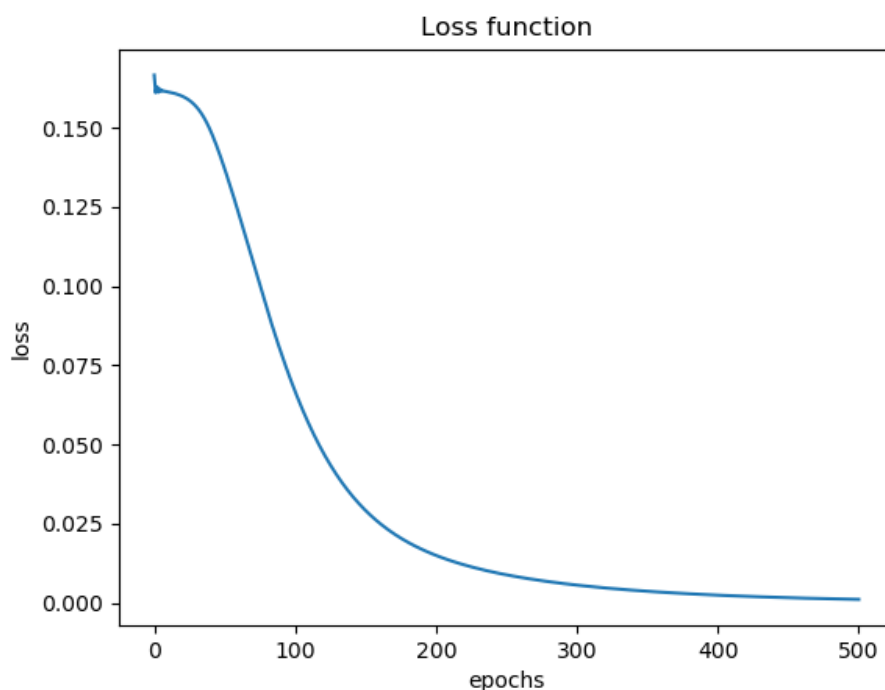


Figure 2: Loss function

Při testovacím běhu neuronové sítě byly dosaženy následující výsledky:

[TEST] CASE 1 "Health data" : label=1.0 net_output=0.9970

[TEST] CASE 2 "Fault data " : label=0.0 net_output=0.0000

Na modelu byla vygenerována korektní i chybná data pro verifikaci funkčnosti neuronové sítě. Tato data jsou zobrazena na následujících obrázcích. První obrázek 4 zobrazuje výchylku a rychlost tělesa v čase pro korektní, plně funkční, model. Jak jde vidět, rychlost a výchylka jsou vůči sobě fázově posunuty a poměrně rychle se ustálí – jedná se o odezvu na skok, na který tlumená soustava reaguje postupným ustálením.

Druhý obrázek 5 zobrazuje situaci pro chybný model, konkrétně případ, kdy se "utrhnul" tlumič a tlumení soustavy je tedy nulové. V tomto případě proto nedochází k ustálení a soustava kmitá dále se stejnou amplitudou.

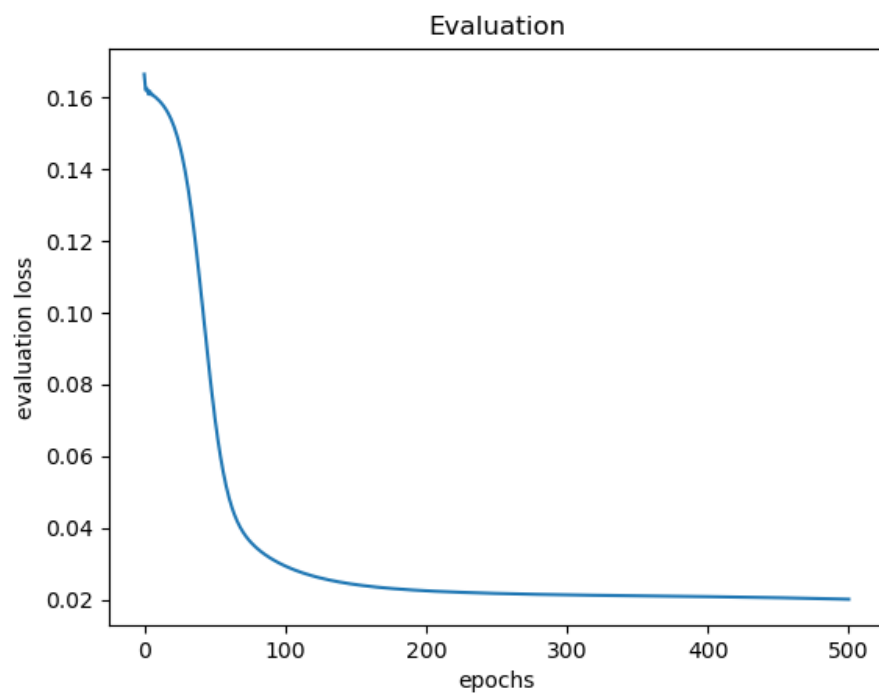


Figure 3: Evaluation

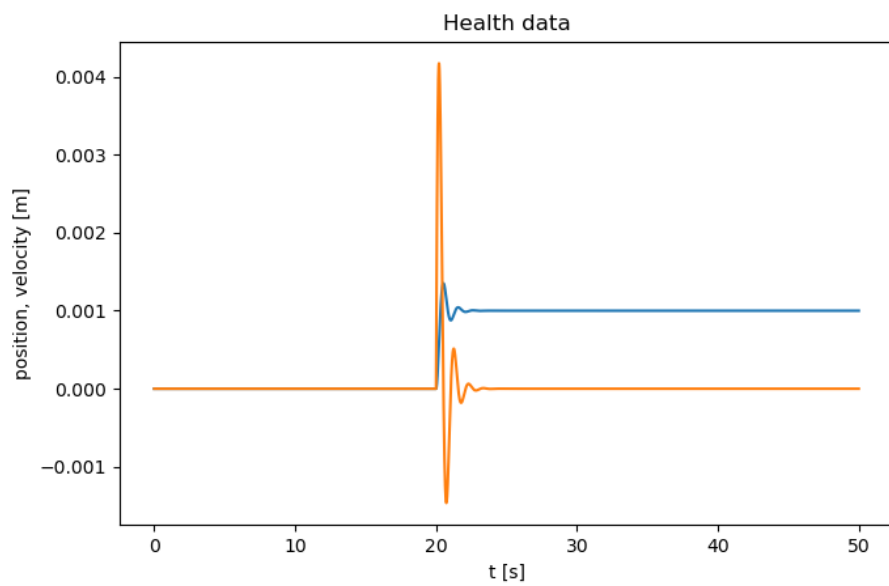


Figure 4: Loss function

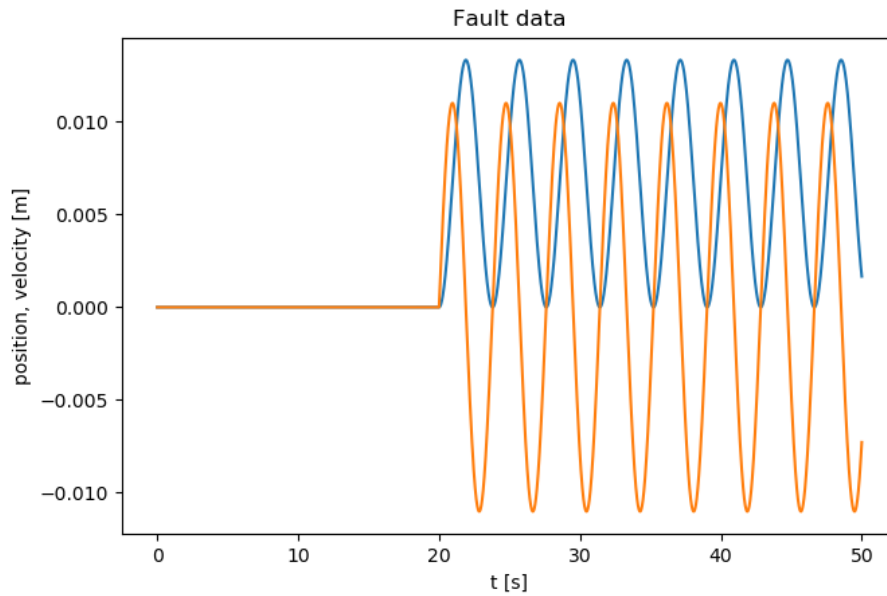


Figure 5: Evaluation

8 Závěr

Pro detekci chyb na základě signálu (signal based fault detection) pomocí neuronové sítě byl vytvořen model, který byl použit pro generování dat. Tato vygenerovaná data ve formě signálů byla následně označena jako korektní a chybná a byla statisticky zpracována pro neuronovou síť. Poté proběhlo učení neuronové sítě na základě těchto statistických dat a labelů.

Neuronová síť, která byla vytvořena pomocí nástroje PyTorch, obsahuje 20 neuronů ve vstupní vrstvě, 16 neuronů ve skryté vrstvě a 1 neuron ve výstupní vrstvě a v rámci její aplikace byla použita sigmoidní aktivační funkce. Nakonec byl proveden testovací běh neuronové sítě a byla vygenerována verifikační data pro ověření funkčnosti neuronové sítě. Ukázalo se, že síť poměrně rychle konverguje ke správnému určení, zda se jedná o korektní, nebo chybný signál.

9 Zdroje

- wiki: Fault detection and isolation
- PyTorch: nn
- A machine learning approach to fault detection in district heating substations
- kaggle: VSB Power Line Fault Detection
- Fault detection methods: A literature survey