

Predikce informační kaskády  
v sítích s využitím fyzikálně  
informované neuronové sítě

# Motivace a podstata problému - informační kaskády na sítích

## Viralizace videoklipů

Jak viralizovat video na YouTube?

- YouTube dominuje trhu s **uživatelsky generovaným videoobsahem**
- **Viralizace**
  - Co přesně je potřeba k tomu, aby se video stalo virálním?
  - Sledování YouTube je příkladem [**šíření informací vytváří závislosti**].



# Virální styl

- Co způsobuje virální šíření videí?
- YouTube zaznamenává chování uživatelů, včetně interakcí s přehrávačem videí.
- Tato data se analyzují, aby se zjistilo, jak lidé videa sledují a která videa se stala virálními.
- YouTube Insight - analytické nástroje
- 1. Video dosahující virálního statusu?
  - **Gangnam Style** od (skupiny) PSY vydaný v červenci 2012: 1. video, které za 5 měsíců překonalo hranici 1 miliardy zhlédnutí; do května 2014 dosáhlo hranice 2 miliard zhlédnutí.



# Virální styl

- Od roku 2013 překonalo hranici jedné miliardy také 12 dalších videí.
- Na druhém místě po Gangnam Style v počtu zhlédnutí se na začátku roku 2016 umístil klip Taylor Swift „Blank Space“ s 1,3 miliardy zhlédnutí.
- Gangnam Style byl stále jediným videoklipem, který dosáhl dvou miliard
- Když v prosinci 2014 překročil hranici **2 147 483 647** zhlédnutí, zdálo se, že YouTube doslova ztratil přehled o počtu zhlédnutí, protože hodnota zobrazená na hlavní stránce se zastavila na **mrtvém bodě**
- Proč?
- 32bitové počítadlo:  $2^{31}-1$

# Přivedení diváků k videím

- Jak se video stane tak populárním?
- K určitému klipu na YouTube mohou diváka přivést především 4 hlavní cesty
  - **vyhledávání** pomocí výrazů, kterými je video označeno, na stránkách jako Google
  - **doporučení** např. z e-mailu, Facebooku nebo reklamy propagující video
  - **přihlášení** k odběru kanálu YouTube, který video zveřejňuje
  - **doporučení** na video uvedené v postranním panelu YouTube.
- **Předplatné** a **doporučení** hrají při rozhodování o popularitě videa větší roli než # lajků/nelajků, které video má.
- Jak YouTube generuje svá **doporučení**?



# Přivedení diváků k videím

- Používá YouTube algoritmus kolaborativního filtrování jako Netflix pro doporučení filmů?
- Používá algoritmus ve stylu PageRank jako Google pro řazení klipů podle „důležitosti“?
- Ani jeden z těchto algoritmů na YouTube dobře nefunguje.
- Proč?
- Na rozdíl od filmů Netflixu mají videa na YouTube obvykle krátkou délku a životní cyklus a jejich chování při sledování je proměnlivé.
  - je obtížné zavést konzistentní systém hodnocení klipů uživateli.
- Pro přístup PageRank je třeba klipy nějak „propojit“, např. hledáním hypertextových odkazů na jiné klipy v popisu videa nebo porovnáváním tagů mezi videi, zda se shodují v klíčových slovech.
  - tagy a popisy mohou být poměrně nespolehlivé, pokud jde o kvalitu
- Doporučování videí na YouTube je jiné a mnohem jednodušší

# Popularita

- **Modely**, které byly vyvinuty pro **šíření informací**, mohou poskytnout náhled na to, proč může dojít k viralizaci.
- **Modely šíření informací**
  - modely analyzovaly šíření „předmětů“ - od fyzických produktů až po nemoci - **v populacích**.
- **Faktory**, které přitahují lidi k danému předmětu v první řadě???
  - **vnitřní hodnota**, kterou předmět člověku přináší - někomu se může líbit bez ohledu na to, co si o něm myslí ostatní.
  - v mnoha případech bude záviset rozhodnutí člověka pořídit si určitý předmět na ostatních.
  - **síťový efekt**



# Sít'ový efekt

**Sít'ový efekt** ze dvou důvodů

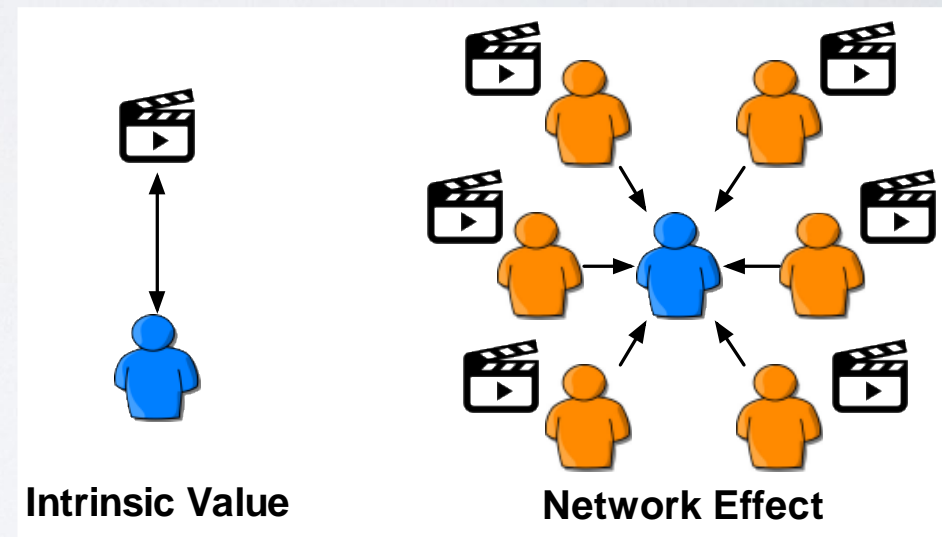
1. **hodnota služby nebo produktu může záviset na počtu lidí, kteří je používají**
  - příklady telefonu a Facebooku
  - **pozitivní sít'ový efekt** - čím více lidí je používá, stávají se cennějšími pro každého jednotlivce
2. **znalost, co si ostatní o předmětu myslí, může ovlivnit naše rozhodnutí**
  - Sledovali jste někdy film, protože vám vaši přátelé řekli, že je dobrý, bez ohledu na to, zda je to žánr, který máte obvykle rádi?
  - názory a rozhodnutí lidí jsou ovlivněny ostatními
  - davy již nejsou „moudré“, protože **předpoklad nezávislosti již neplatí**. Místo toho je výsledkem omyl davů



# Popularita

- Jak **vnitřní hodnota**, tak **sít'ový efekt** se vztahují na to, že se člověk rozhodne sledovat video na YouTube.
- Samotná stránka YouTube **má pozitivní sít'ový efekt**
- Který z nich má větší vliv?

1. **vnitřní hodnota klipu** pro danou osobu (zda odpovídá jejím preferencím).
2. **klam davu** (zda vidí/zná hodně dalších lidí, kteří se na něj dívají)



**sít'ový efekt**, který šíří sledování videa v **populaci**, a má tak větší vliv na virální šíření videa.

Potřeba **kvantifikovat** sít'ový efekt

# Sít'ový efekt (klam davu)

- Kvantifikace sít'ových efektů není snadný úkol
  - závisí na jednotlivci, položce a situaci, která je předmětem zájmu
- Model **informační kaskády**

## Příklad informační kaskády



- Co byste udělali, kdybyste viděli někoho stát na rohu ulice a dívat se na nebe?
- Mysleli byste si, že dotyčnému teče krev z nosu, a pokračovali byste v práci.
- Co kdybyste viděli **deset lidí**, jak stojí pohromadě a dívají se na oblohu?
- Pravděpodobně byste se zastavili a podívali se, protože byste si mysleli, že něco není v pořádku.
- **Tím se dav ještě zvětší**, takže další člověk, který půjde kolem, uvidí 11 lidí, což vás ještě více přesvědčí, abyste se zastavil a podíval se



# Informační kaskáda

- Lidé následují jednání davu a **ignorují** své vlastní vnitřní uvažování.
- **Informační kaskáda** vzniká a zde **předpoklad nezávislosti názorů** (který stojí za moudrostí davu) **neplatí**.
- Místo úplné nezávislosti v rozhodování se rozhodnutí stávají **zcela závislými na tom, co se stalo předtím**



**Pozitivní zpětná vazba v sekvenčním rozhodování**

- Např. bubliny na burze, módní trendy atd.

# Informační kaskáda

- Je pravděpodobnější, že narazíte na video, které je již populární.
- I když video neodpovídá vašemu vkusu, můžete být nuceni se podívat, o co jde.
- Pokud se vám video nelíbí, můžete se rozhodnout, že jeho sledování ukončíte, ale i tak se započítá do počtu zhlédnutí, který se zobrazuje u videa a částečně určuje jeho umístění na stránce s doporučením
- Vyšší počet zhlédnutí zase ovlivní více lidí a tato kumulace se neustále zvyšuje (pozitivní zpětná vazba)



# Postupné přijímání rozhodnutí

- Jaký proces nakonec spustí informační kaskádu?
- Sekvenční rozhodování - každá osoba dostane **soukromý signál** (např. začíná mi téct krev z nosu) a udělá „**veřejnou akci**“ (např. nakloním hlavu k nebi).
- Následní uživatelé mohou pozorovat „**veřejnou akci**“, ale ne soukromý signál
- Když je dostatek „veřejných akcí“ stejného typu (např. deset lidí se dívá na oblohu), pak všichni pozdější uživatelé budou ignorovat své vlastní soukromé signály a budou jednoduše sledovat, co dělají ostatní
- V tomto okamžiku byla spuštěna kaskáda.....
- Klíčová otázka ?
  - Kolik veřejných akcí stačí? Kolik lidí musí udělat stejné?

# Postupné přijímání rozhodnutí

- Kolik „veřejných akcí“ stačí ke spuštění kaskády, závisí na situaci.
- Pravděpodobně je mnohem těžší přimět všechny, aby se podívali na vaše video na YouTube, než přimět lidi, aby se podívali na oblohu.
- Kaskáda se může díky **pozitivní zpětné** vazbě nahromadit do velkých rozměrů



- Více lidí projevujících stejnou veřejnou akci dává dalšímu člověku větší motivaci k následování, čímž se skupina zvětšuje, a tím vzniká další motivace atd.
- Pozitivní zpětná vazba **se živí** vlastním nezmenšeným vlivem, čímž vytváří další vliv, a dále se zvětšuje

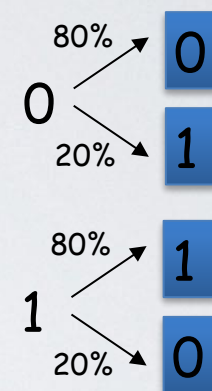


# Postupné přijímání rozhodnutí

- Pozitivní vs. negativní zpětná vazba, která systematicky působí proti účinku, aby bylo dosaženo rovnováhy v síti (např. prostřednictvím distribuované regulace výkonu nebo cen za použití).
- Je „veřejná akce“ „správná“ nebo „špatná“?
  - může být obojí
  - „všichni se dívají nahoru, ale na obloze není nic zajímavého“ je špatně → **příklad klamu davů**
- Kaskáda je **křehká**: i když několik soukromých signálů unikne na veřejnost (jeden člověk vykřikne „dívám se na oblohu, protože mi teče krev z nosu“), kaskáda může rychle zmizet nebo dokonce obrátit směr. Proč?
  - Protože lidé sledují dav, mají malou důvěru v to, co dělají, přestože mnozí dělají totéž.

# Experiment s hádáním čísel

- Skupina lidí se postaví do řady, aby si zahrála hru, ve které **budou hádat** jedno číslo.
- **Moderátor** vybere jako (jediné) **správné** číslo buď 0, nebo 1.
- Každý po jednom přistupuje k tabuli, kam má napsat, co si myslí (hádá), že je to číslo.
- Moderátor má dvě kartičky, na jedné z nich je napsána 0 a na druhé 1
- Když se osoba přihlásí, moderátor jí ukáže kartičku, na které je napsáno 0 nebo 1 - slouží jako **soukromý signál** dané osobě.
- Neexistuje žádná záruka, že číslo, které je osobě ukázáno, bude správné, ale každý je upozorněn, že je **větší šance**, že karta, která je mu ukázána, **je správná než špatná**.
- Pokud je skutečné číslo 0, má moderátor šanci, řekněme 80 %, že ukáže kartu 0, 20 %, že ukáže kartu 1.
- Pokud je skutečné číslo 1, má moderátor šanci, řekněme 80 %, že ukáže kartu 1 a 20 %, že ukáže kartu 0.
- Tip každé osoby napsaný na tabuli je její **veřejná akce**.
- Když osoba hádá, vidí **veřejné akce všech**, kteří hádali **před** ní, ale nevidí soukromé signály, které byly ukázány těm před ní





# Císařovy nové šaty

- Spuštění kaskády je snadné. Jak dlouho bude trvat, když se jednou spustí?
- **Navždy**, pokud nedojde k nějakému **narušení**, např. uvolnění soukromých signálů.
- Kolik narušení by bylo potřeba k přerušení kaskády?
- Často jich stačí několik, bez ohledu na to, jak dlouho kaskáda probíhá
- Navzdory počtu zúčastněných osob všichni vědí, že v podstatě jen hrají hru, ve které následují vůdce, aby **maximalizovali svou šanci na správný odhad**
- **Efekt císařových nových šatů** vystihuje **křehkost** informační kaskády.

# Císařovy nové šaty

- příběh Hanse Christiana Andersona z 19. století, v němž se ješitný císař dozvídá, že jeho nové „šaty“ jsou z té nejjemnější látky, neviditelné pouze pro ty, kteří se na své místo nehodí.
- **Ve skutečnosti žádné šaty nejsou.** Zatímco si všichni hrají na to, že jsou „nevhodní“ (tj. jejich veřejné jednání), a nechtějí vypadat „nevhodně“ (tj. jejich soukromé signály), stačí, aby jedno dítě vykřiklo „**hele, on na sobě nemá vůbec nic!**“, a všichni si jsou jistější, že císař je skutečně nahý.





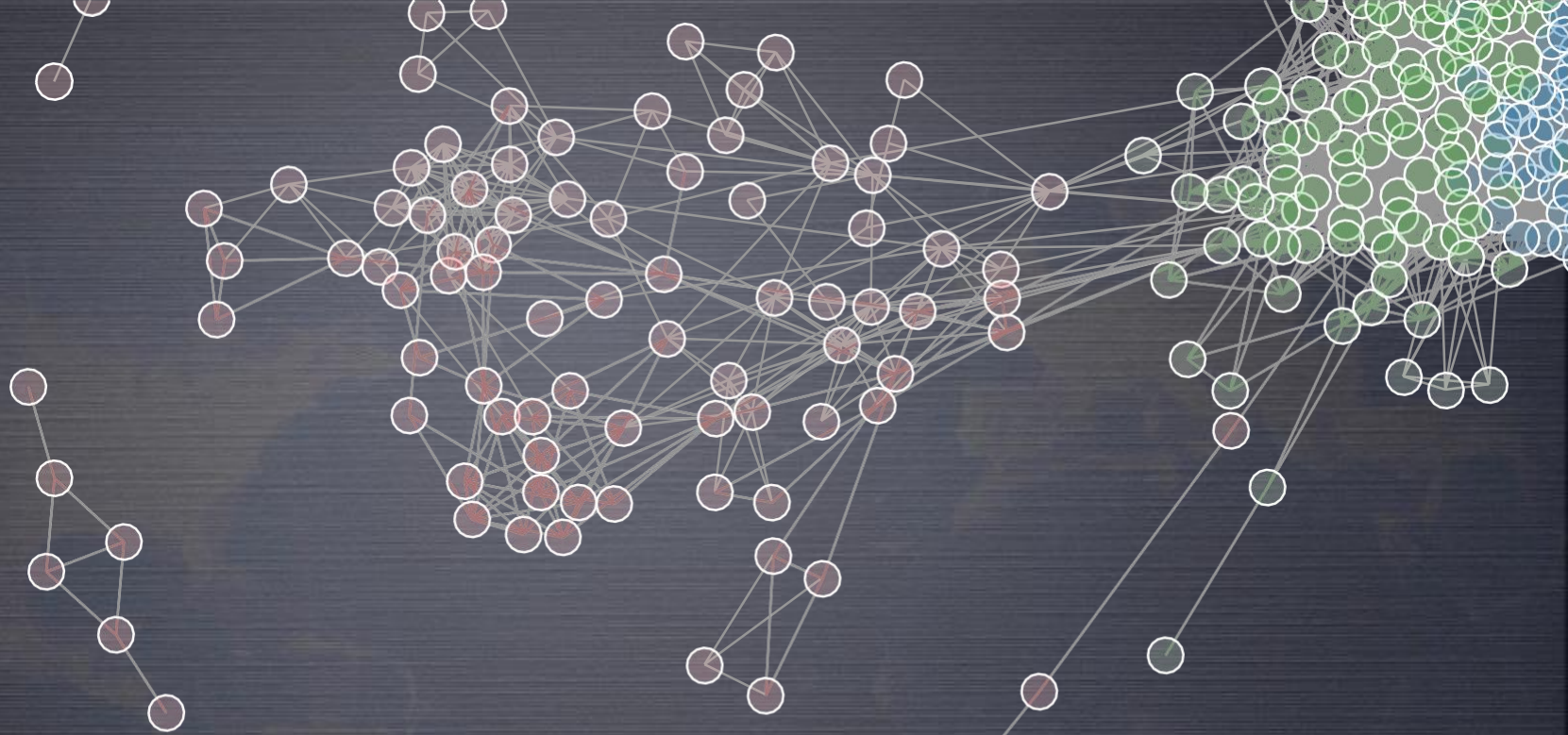
# Předpoklady vzniku kaskády

- V experimentu s hádáním čísel stačí dva lidé, aby spustili kaskádu.
- Obecněji řečeno, velikost davu potřebná k tomu, aby člověk ignoroval svůj instinkt, závisí jak na (i) scénáři, tak na (ii) jednotlivci.
- Také nejobecnější **předpoklad** je, že každý jedná **racionálně** - každý se může rozhodnout a rozhodne, jaký odhad je nejlepší v závislosti na informacích, které má k dispozici.
  - ale většina lidí **neprochází** všechny tyto pravděpodobnostní úvahy na místě ve své hlavě

# Od kaskády k YouTube

- Jak můžeme sekvenční rozhodování převést na viralizaci videa na YouTube?
- Není to snadné, ale hlavní myšlenka je jasná
  - chcete, aby vaše video prošlo informační kaskádou, takže když ho osoba uvidí nebo se o něm dozví (tj. veřejná akce), s největší pravděpodobností si ho pustí bez ohledu na to, zda odpovídá jejímu vnitřnímu zájmu (tj. jejímu soukromému signálu), nebo ne
- Kolik veřejných akcí je potřeba, aby se osoba na vaše video automaticky podívala? Existuje vůbec takové číslo? I kdyby existovalo, bude pro každého jiné, v závislosti na tom, jak je daná osoba tvárná.
- Samé zajímavé otázky bez jasných odpovědí





Predikce kaskády v síti s využitím  
fyzikálně informované neuronové  
sítě

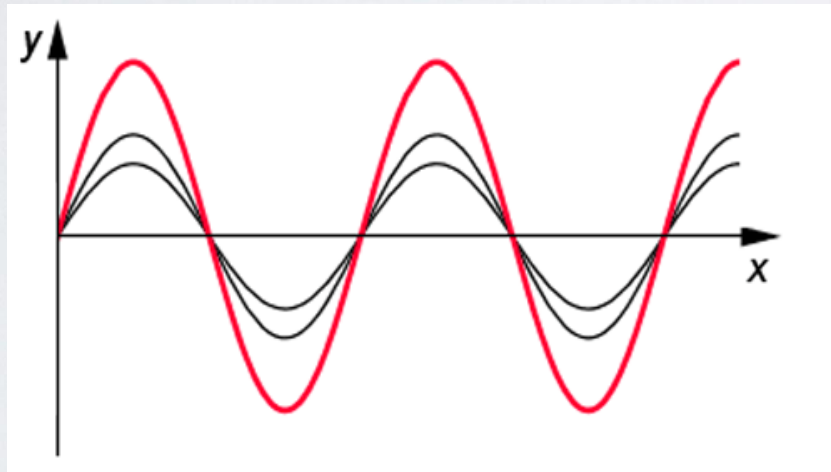
# Fyzikálně informované neuronové sítě

Řešení parciálních diferenciálních rovnic (PDE) je:

Základem pro prakticky všechny vědní obory.

Zásadní pro pochopení chování složitých vědeckých jevů

Vlnová rovnice:



Fyzikálně informované neuronové sítě nabízejí další možnosti řešení složitých problémů založených na PDE

$$\nabla^2 u - \frac{1}{c(x,y)^2} \frac{\partial^2 u}{\partial t^2} = s(x,y,t)$$

$u$  = akustický tlak

$c$  = rychlost

$s$  = funkce

Počáteční podmínky:

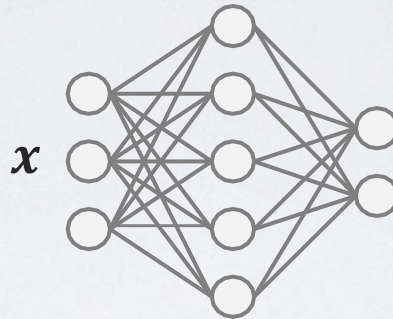
$$u(x,y,t=0) = 0$$

$$u_t(x,y,t=0) = 0$$



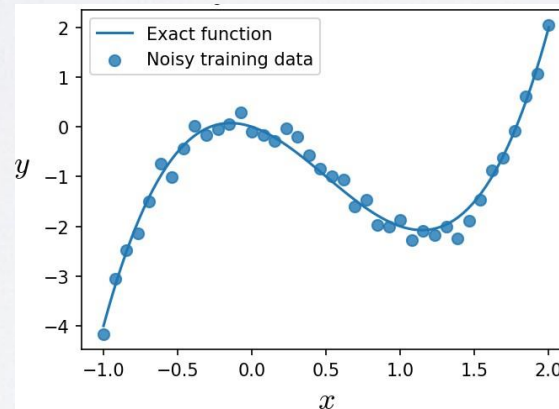
# Co je neuronová síť?

Neuronové sítě jsou jednoduše flexibilní funkce přizpůsobené datům.



$$y = (NN(x, \theta))$$

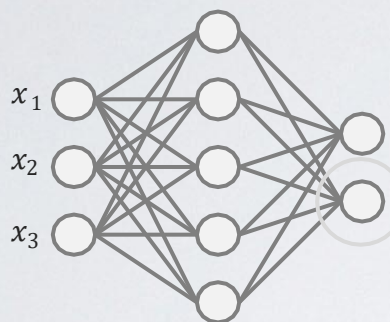
Příklad dat:



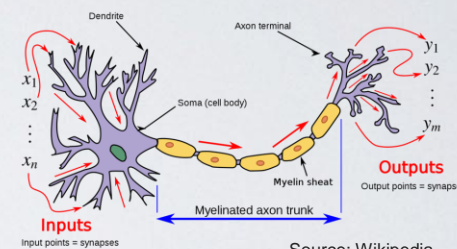
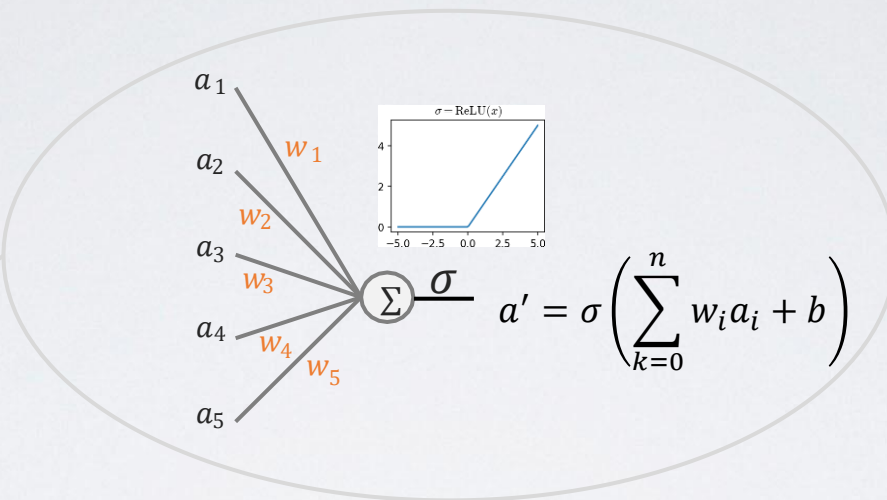
Cíl: vzhledem k trénovacím datům vyladit parametry  $\theta$  tak, aby síť aproximovala skutečnou funkci, tj,

$$NN(x, \theta) \approx y(x)$$

# Co je neuronová síť?



2-vrstvý perceptron

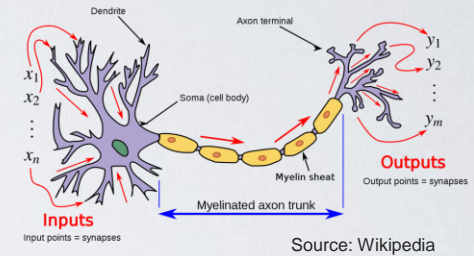
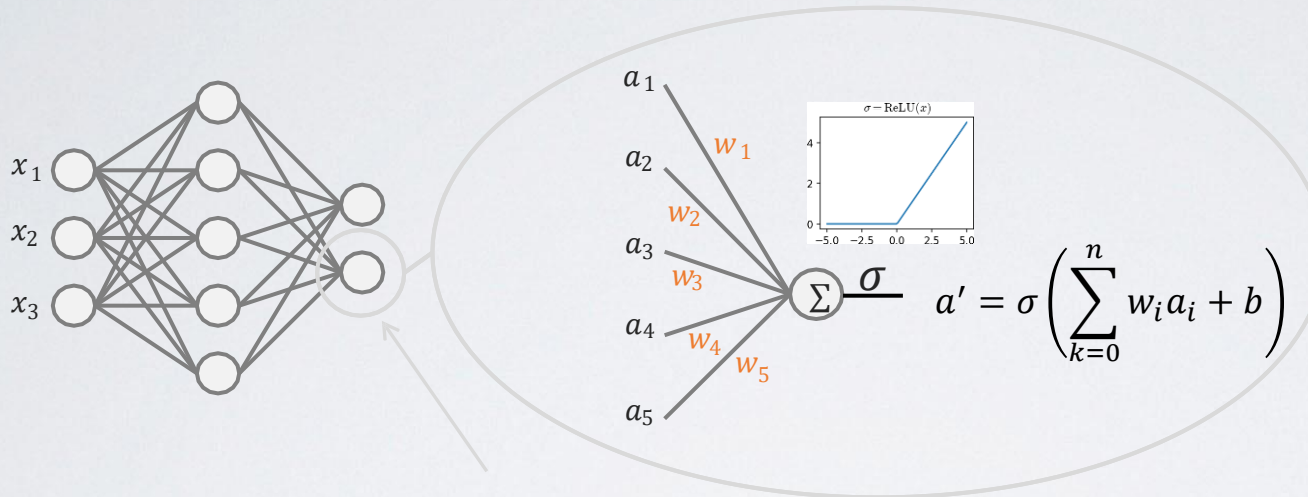


Source: Wikipedia

Biologický neuron



# Co je neuronová síť?



Biologický neuron

Pro poslední vrstvu:

$$\begin{pmatrix} a'_1 \\ a'_2 \end{pmatrix} = \sigma \left( \begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{25} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right)$$

2-vrstvý perceptron

$$NN(x, \theta) = \sigma(W_2 \sigma(W_1 x + b_1) + b_2)$$

Celá síť:

$$NN(x, \Theta) = \sigma(W_2 \sigma(W_1 x + b_1) + b_2)$$

# Trénování neuronové sítě

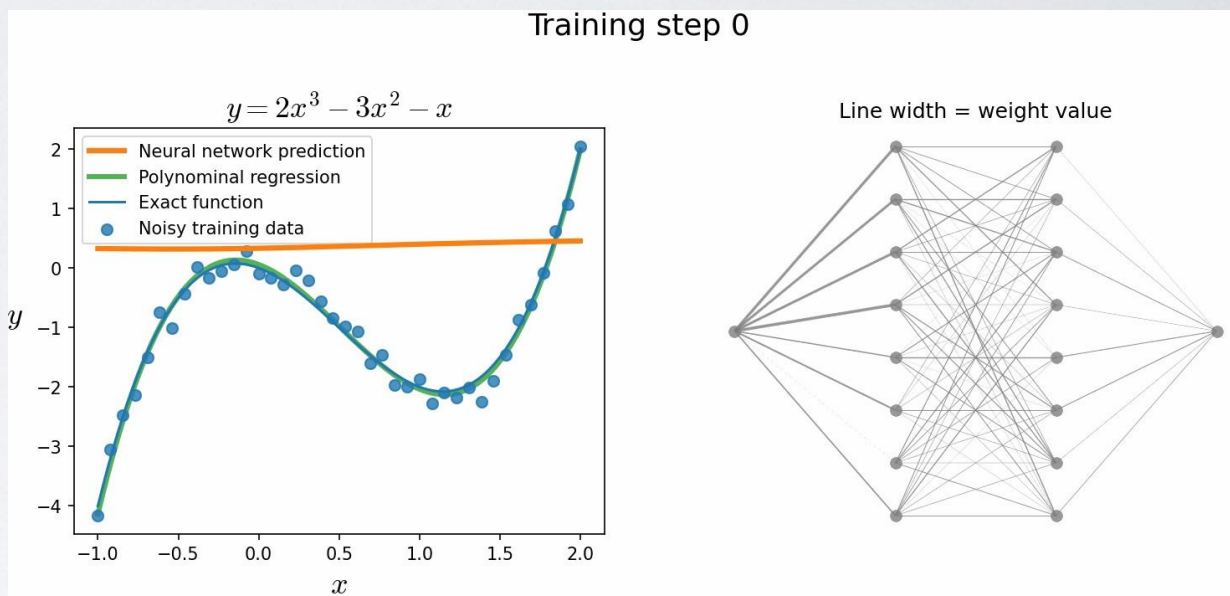
Parametry ladíme tak, aby minimalizovaly nějakou ztrátovou funkci, např.

$$L(\Theta) = \frac{1}{N} \sum_1^N (NN(x_i, \Theta) - y_i)^2$$

Typicky pomocí gradientního sestupu (**gradient descent**):

$$\Theta \leftarrow \Theta - \gamma \frac{\partial L(\Theta)}{\partial \Theta}$$

$\gamma$  = velikost kroku, volíme např. 0,001





# Jak fungují fyzikálně informované neuronové sítě?

Tlumený harmonický oscilátor:

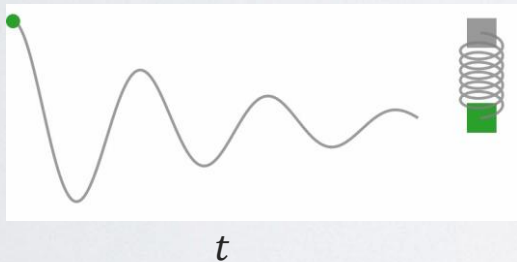
$$m \frac{d^2 u}{dt^2} + \mu \frac{du}{dt} + ku = 0$$

Počáteční podmínky:

$$u(t = 0) = 1$$

$$u_t(t = 0) = 0$$

$u$  = posunutí  
 $m$  = hmotnost oscilátoru  
 $\mu$  = koeficient tření  
 $k$  = konstanta pružiny



# Jak fungují fyzikálně informované neuronové sítě?

Tlumený harmonický oscilátor:



Klíčová myšlenka: použít neuronovou síť k přímé aproximaci řešení

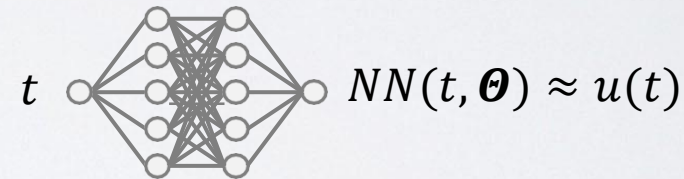
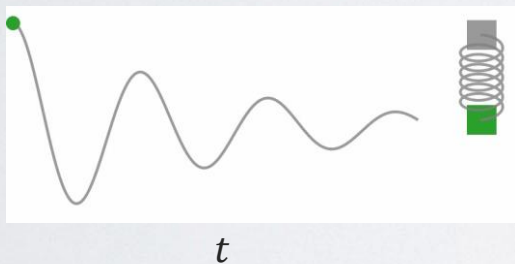
$$m \frac{d^2 u}{dt^2} + \mu \frac{du}{dt} + ku = 0$$

$$NN(t, \theta) \approx u(t)$$

Počáteční podmínky:

$$u(t = 0) = 1$$

$$u_t(t = 0) = 0$$



$u$  = posunutí  
 $m$  = hmotnost oscilátoru  
 $\mu$  = koeficient tření  
 $k$  = konstanta pružiny

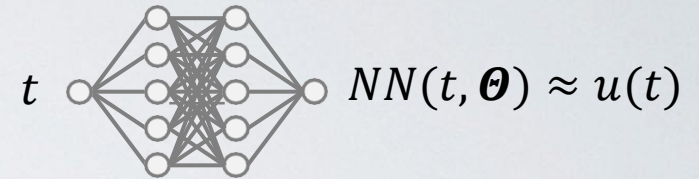


# Jak fungují fyzikálně informované neuronové sítě?

Tlumený harmonický oscilátor:

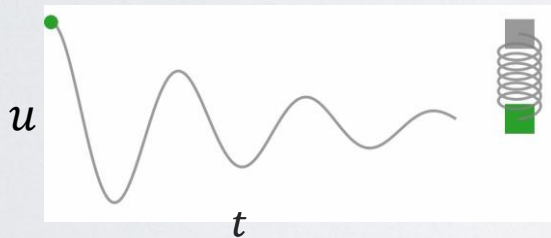
$$m \frac{d^2 u}{dt^2} + \mu \frac{du}{dt} + ku = 0$$

Klíčová myšlenka: použít neuronovou síť k přímé aproximaci řešení



Počáteční podmínky:

$$u(t = 0) = 1$$



$u$  = posunutí  
 $m$  = hmotnost oscilátoru  
 $\mu$  = koeficient tření  
 $k$  = konstanta pružiny

Trénink sítě pomocí ztrátové funkce (dvě části):

$$L_b(\theta) \begin{cases} L(\theta) = \lambda_1 (NN(t=0, \theta) - 1)^2 + \\ + \lambda_2 \left( \frac{dNN}{dt}(t=0, \theta) - 0 \right)^2 + \end{cases}$$

$$L_p(\theta) \begin{cases} + \frac{1}{N_p} \sum_i^{N_p} \left( \left[ m \frac{d^2}{dt^2} + \mu \frac{d}{dt} + k \right] NN(t_i, \theta) \right)^2 \end{cases}$$

(také reziduum PDE)

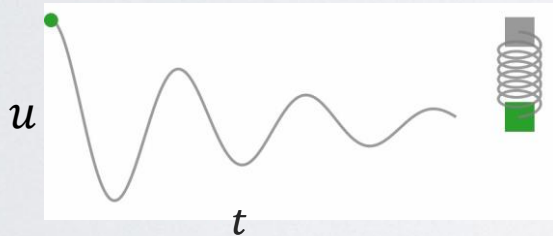
# Jak fungují fyzikálně informované neuronové sítě?

Tlumený harmonický oscilátor:

$$m \frac{d^2 u}{dt^2} + \mu \frac{du}{dt} + ku = 0$$

Počáteční podmínky:

$$u(t = 0) = 1$$



$u$  = posunutí  
 $m$  = hmotnost oscilátoru  
 $\mu$  = koeficient tření  
 $k$  = konstanta pružiny

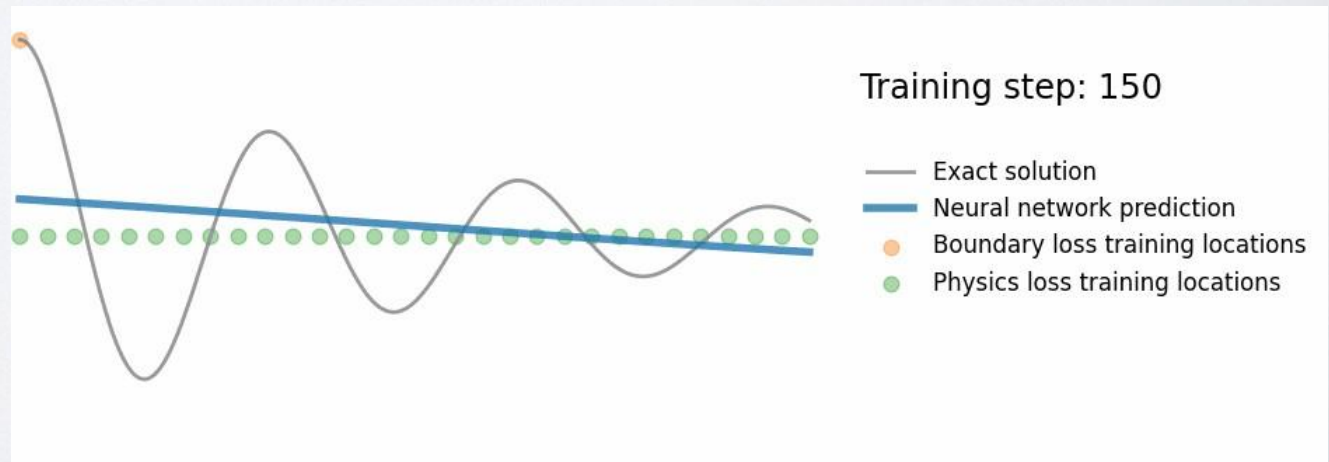
Klíčová myšlenka: použít neuronovou síť k přímé aproximaci řešení



Trénink sítě pomocí ztrátové funkce:

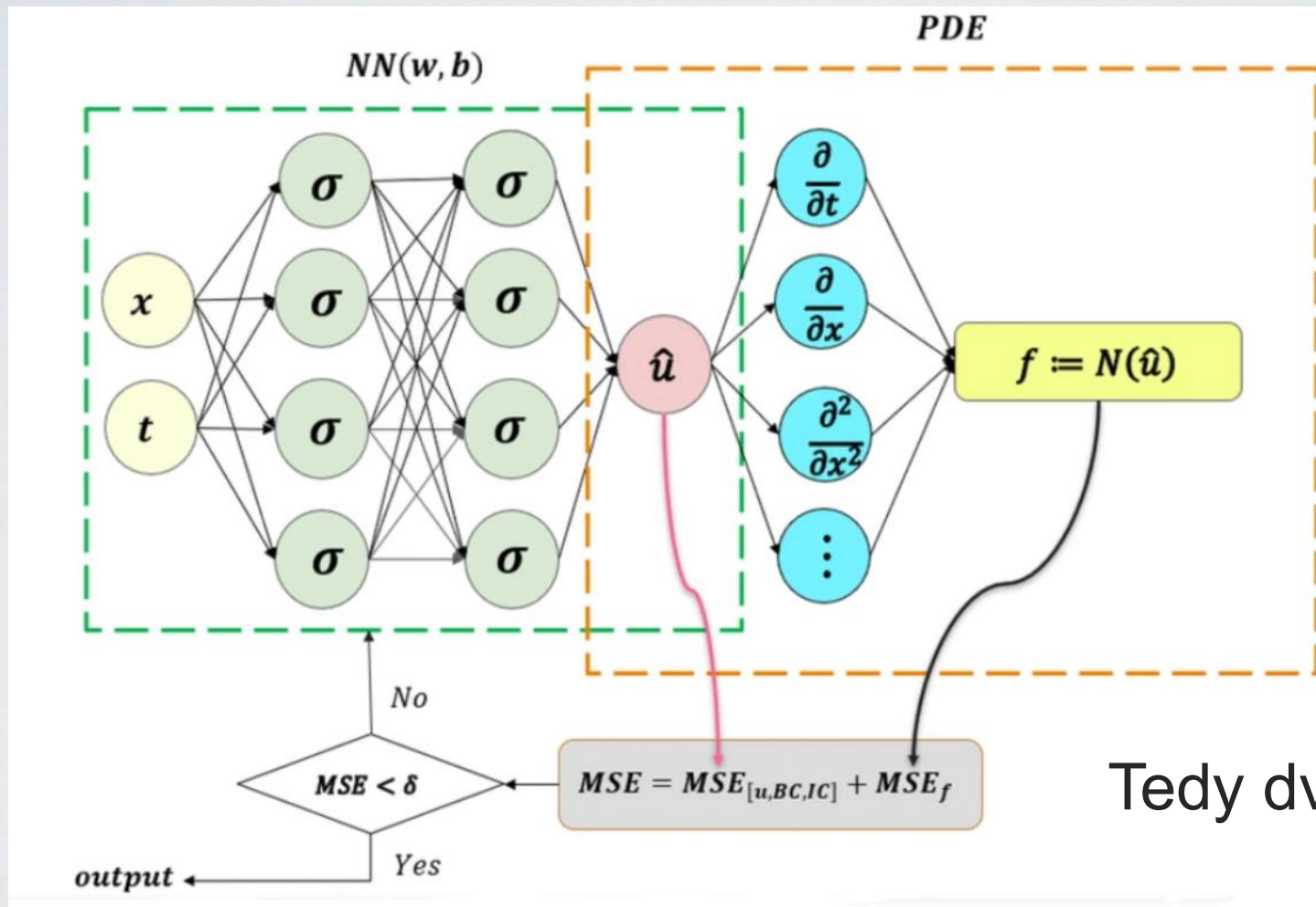
$$L_b(\theta) \begin{cases} L(\theta) = \lambda_1 (NN(t=0, \theta) - 1)^2 \\ + \lambda_2 \left( \frac{dNN}{dt}(t=0, \theta) - 0 \right)^2 \end{cases}$$

$$L_p(\theta) \begin{cases} + \frac{1}{N_p} \sum_i^{N_p} \left( \left[ m \frac{d^2}{dt^2} + \mu \frac{d}{dt} + k \right] NN(t_i, \theta) \right)^2 \end{cases}$$





# Jak fungují fyzikálně informované neuronové sítě?



Tedy dva typy ztráty

# Jak fungují fyzikálně informované neuronové sítě?

Jednorozměrná Burgersova rovnice (vazká Burgersova rovnice)

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}$$

Následně využíváme knihovnu PyTorch



# Potřebné kroky

## 1. Definujeme neuronovou síť

```
import math
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from tqdm import tqdm

import torch
import torch.nn as nn

class NN(nn.Module):
    def __init__(self):
        super(NN, self).__init__()
        self.net = torch.nn.Sequential(
            nn.Linear(2,20),
            nn.Tanh(),
            nn.Linear(20,30),
            nn.Tanh(),
            nn.Linear(30,30),
            nn.Tanh(),
            nn.Linear(30,20),
            nn.Tanh(),
            nn.Linear(20,20),
            nn.Tanh(),
            nn.Linear(20,1)
        )

    def forward(self, x):
        out = self.net(x)
        return out
```

## 2. Definujeme počáteční a okrajové podmínky

```
class Net:
    def __init__(self):
        device = torch.device("cuda") if torch.cuda.is_available()
        else torch.device("cpu")

        self.model = NN().to(device)

        # comp. domain
        self.h = 0.1
        self.k = 0.1
        x = torch.arange(-1,1+self.h, self.h)
        t = torch.arange(0,1+self.k, self.k)

        self.X = torch.stack(torch.meshgrid(x,t)).reshape(2,-1).T

        # train data
        bc1 = torch.stack(torch.meshgrid(x[0],t)).reshape(2,-1).T
        bc2 = torch.stack(torch.meshgrid(x[-1],t)).reshape(2,-1).T
        ic = torch.stack(torch.meshgrid(x,t[0])).reshape(2,-1).T
        self.X_train = torch.cat([bc1, bc2, ic])

        y_bc1 = torch.zeros(len(bc1))
        y_bc2 = torch.zeros(len(bc2))
        y_ic = -torch.sin(math.pi * ic[:,0])
        self.y_train = torch.cat([y_bc1, y_bc2, y_ic])
        self.y_train = self.y_train.unsqueeze(1)

        self.X = self.X.to(device)
        self.y_train = self.y_train.to(device)
        self.X_train = self.X_train.to(device)
        self.X.requires_grad = True
```

### 3. Definujeme optimalizaci

```
# optimizer setting
self.adam
= torch.optim.Adam(self.model.parameters())
#Limited-memory Broyden-Fletcher-
Goldfarb-Shanno (L-BFGS)
self.optimizer = torch.optim.LBFGS(
    self.model.parameters(),
    lr=1.0,
    max_iter = 50000,
    max_eval = 50000,
    history_size = 50,
    tolerance_grad = 1e-7,
    tolerance_change = 1.0*
np.finfo(float).eps,
    line_search_fn = "strong_wolfe"
)

self.criterion = torch.nn.MSELoss()
self.iter = 1
```

### 4. Definujeme ztrátové funkce



```
def loss_func(self):
    self.adam.zero_grad()
    self.optimizer.zero_grad()

    y_pred = self.model(self.X_train)
    loss_data = self.criterion(y_pred,self.y_train)

    u = self.model(self.X)

    du_dX = torch.autograd.grad(u,
        self.X,
        grad_outputs = torch.ones_like(u),
        create_graph = True,
        retain_graph = True
    )[0]
    du_dt = du_dX[:,1]
    du_dx = du_dX[:,0]

    du_dXX = torch.autograd.grad(du_dX,
        self.X,
        grad_outputs = torch.ones_like(du_dX),
        create_graph = True,
        retain_graph = True
    )[0]

    du_dxx = du_dXX[:,0]

    loss_pde = self.criterion(du_dt + u.squeeze()*du_dx , (0.01/math.pi) * du_dxx)

    loss = loss_pde + loss_data
    loss.backward()

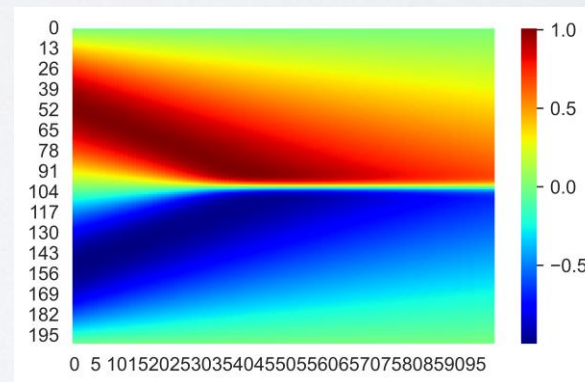
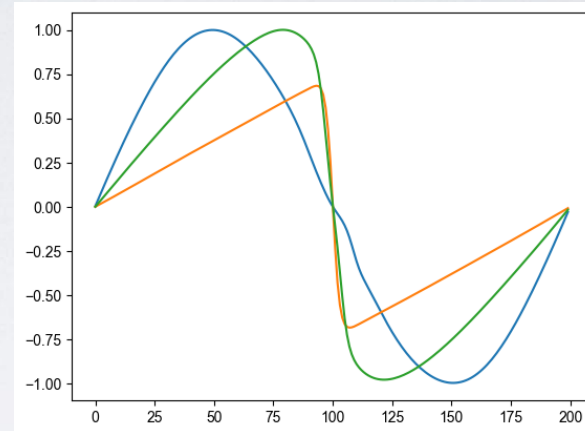
    if self.iter % 100 == 0:
        print(self.iter, loss.item())
    self.iter = self.iter+1
    return loss
```



## 5. Trénování

```
def train(self):  
    self.model.train()  
    for i in range(1000):  
        self.adam.step(self.loss_func)  
        self.optimizer.step(self.loss_func)  
  
def eval_(self):  
    self.model.eval()
```

## 6. (Post Processing) Vykreslení výsledků



# Model epidemie SIR – podklad pro modelování kaskád



Kompartmentový model pro model SIR

$$\frac{dS}{dt} = -\beta S I$$

Aproximováno neuronovou sítí. Přitom celková populace  $N = S + I + R$ .

$$\frac{dI}{dt} = \beta S I - \alpha I$$

$$E_1 \rightarrow \frac{dS}{dt} + \beta S I$$

$$\frac{dR}{dt} = \alpha I$$

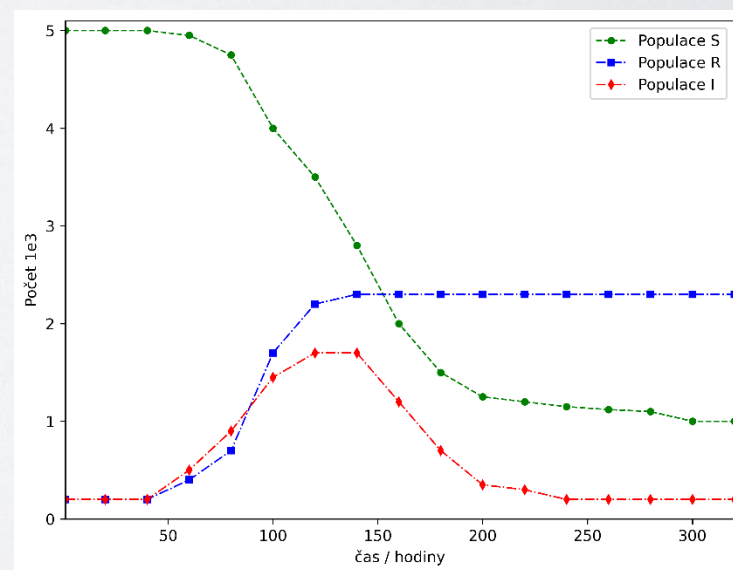
$$E_2 \rightarrow \frac{dI}{dt} - \beta S I + \alpha R$$

$$E_3 \rightarrow \frac{dR}{dt} - \alpha I$$

Model jsme zkoušeli na jednom veřejném souboru dat. Jednalo se o soubor uživatelů na síti Douban<sup>1</sup>

<sup>1</sup> Zhu, Feng and Wang, Yan and Chen, Chaochao and Liu, Guanfeng and Zheng, Xiaolin (2020) A Graphical and Attentional Framework for Dual-Target Cross-Domain Recommendation. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, pp. 3001-3008

Dostupné na: <https://www.kaggle.com/datasets/fengzhujoey/douban-datasetratingreviewside-information>





Physics-informed neural network (PINN) jsou aplikací neuronových sítí na problém řešení diferenciálních a parciálních diferenciálních rovnic. V principu jednoduchá technika, založená na náhodné mřížce a minimalizaci chyby splnění diferenciální rovnice v těchto bodech pro neuronovou síť. Od svého přestavení před několika lety se tato technika vyvinula do bohaté formy metod podporujících práci s neurčitostí, aktivní volbu uzlových bodů, a mnoho dalších rozšíření.

Cílem příspěvku bylo představit řešení problém a ukázat použitou metodu PINN na jednoduchém problému.

**Děkuji za pozornost**