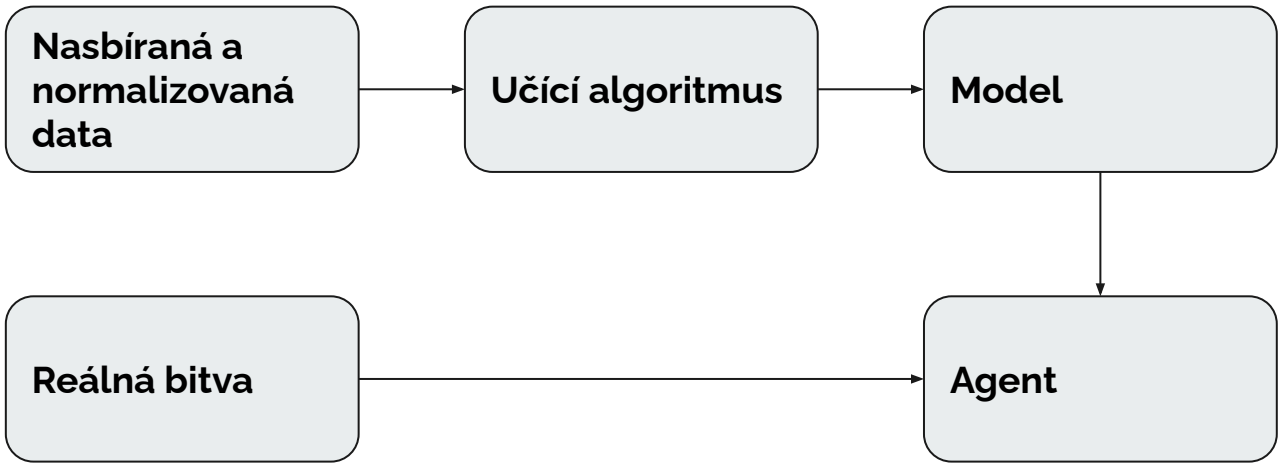
Strojové učení
Projekt 5

Autoři:
- Patrik Dobiáš
- Vladimír Janout
- Jaromír Štefánik
- Libor Sasák
- Jiří Březina
- Patrik Končitý

Zadání č. 5

Seznamte se s problematikou strojového učení. Pro vybranou metodu vytvořte vhodnou trénovací množinu, která bude posuzovat vhodné **načasování střílení** takovým způsobem, aby se maximalizovala pravděpodobnost zásahu. Natrénujte pro vybranou metodu a zhodnoťte dosažené výsledky.

Struktura projektu



Sběr dat

- Náš tank sbírá všechny informace v momentě kdy naskenuje protivníka

```
...
public void onScannedRobot(ScannedRobotEvent e) {
    super.onScannedRobot(e);
    observation = new Observation(getX(), getY(), e.getBearing(), getHeading(),
getGunHeading(), e.getDistance(), e.getHeading(), e.getVelocity());
}
```

- Zajímají nás jen případy, kdy je vystřelena kulka - Pokud ano, uloží se záznam

```
@Override
public void onFired(Bullet bullet) {
    if (bullet != null) {
        observations.add(observation);
        observation.setBullet(bullet);
    }
}
```

Sběr dat

- Na konci kola uložíme data do .csv souboru
- Takto sesbíraná data normalizujeme, abychom je mohli použít pro vstup učícího algoritmu

```
// save observations on end
    @Override
    public void onRoundEnded(RoundEndedEvent
e)          {
        super.onRoundEnded(e);
        save();
    }
```

```
// method for adding info to CSV file
    private void save() {
        // open observations file in append mode
        try (FileWriter writer = new FileWriter("observations.csv",
true)) {
            for (Observation observation: observations)
                writer.write(observation.toString());
            ...
```

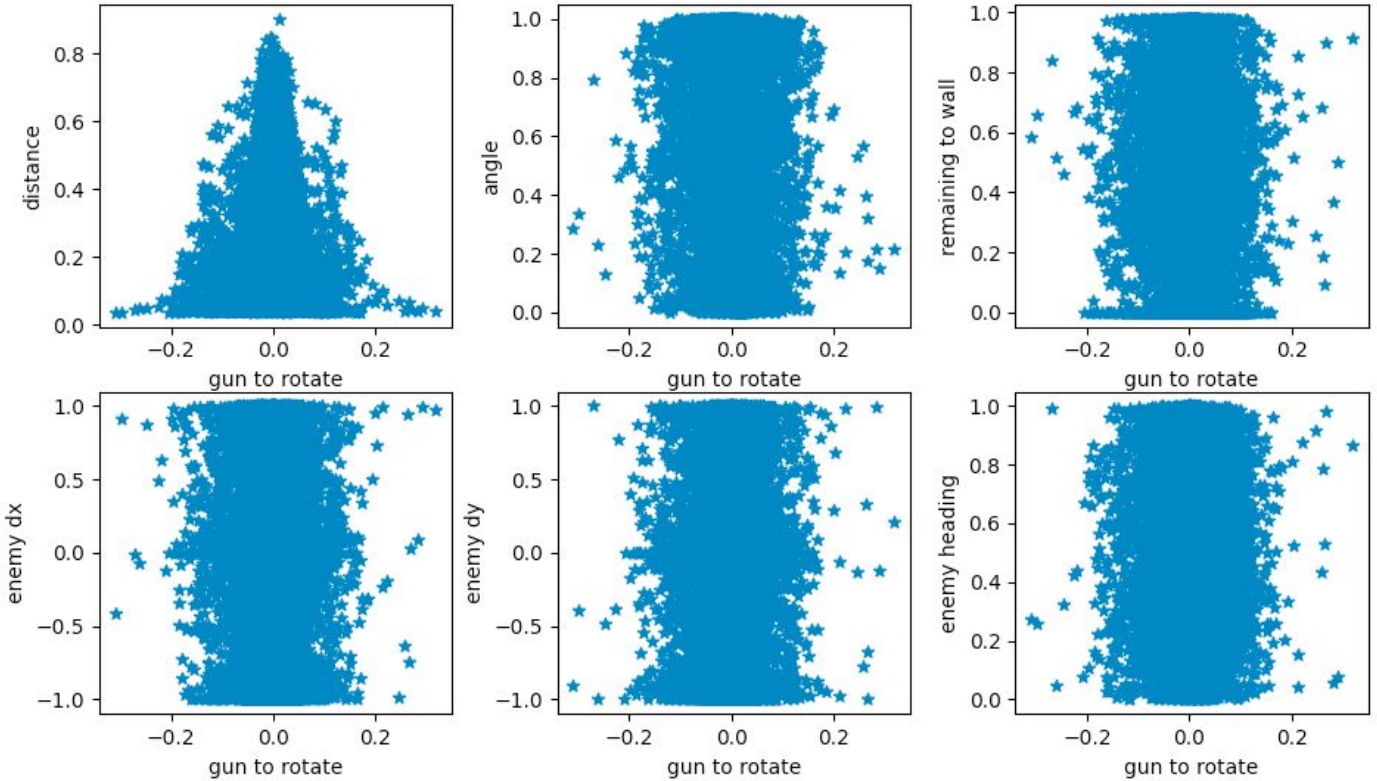
Normalizace dat

- Z dat vytváříme vstupy pro učící algoritmus
- Používáme **relativní hodnoty**
- Nejvíce mají na výsledek vliv hodnoty **distance** a **gunToTurn**

```
private double distance;
private double enemyHeading;
private double enemyDx;
private double enemyDy;
private double angle;
private double gunToTurn;
private double remainingToWall;
```

```
...
double relativeGunHeading = toRelative(gunHeading - heading);
this.gunToTurn = toRelative(bearing - relativeGunHeading);
...
```

Závislost zásahů na sbíraných parametrech



Normalizace dat

- Normalizované hodnoty jsou reálná čísla z intervalu <-1, 1>

```
private void normalize() {
        this.distance /= 1200;
        this.enemyHeading /= 90;
        this.enemyDx /= 8;
        this.enemyDy /= 8;
        this.gunToTurn /= 180;
        this.angle /= 180;
    }
```

Trénovací množina

- 4 trénovací množiny, každá z nich jsou data z her proti jednomu specifickému protivníkovi
- zvolení protivníci: **crazy, sitting duck, veloci robot, walls**
- Pro každého protivníka vypadá dataset trochu jinak (10 - 30k řádků)
- Filtrace duplicitních a podobných záznamů před procesem učení

Algoritmy

kNN - k-nejbližších sousedů

SVM - Metoda podpůrných vektorů

ANN - Umělá neuronová síť

kNN a SVM

- Nevytváříme vlastní model
- Obecná implementace
  - knihovna: **scikit-learn**
- Statistické modely

```
from sklearn import svm, neighbors
```

```
from models import SVMModel, KNNModel
from util import read_data, number_of_inputs

data = read_data('training_data/walls.csv')

# divide loaded data to inputs and outputs
X = data[:, :number_of_inputs]
y = data[:, -1]

model = SVMModel()
model.fit(X, y)
model.save('model/walls_svm.pkl')

model = KNNModel()
model.fit(X, y)
model.save('model/walls_knn.pkl')
```

ANN - Model

- využití knihovny keras
- základní model s plně propojenými vrstvami
- aktivační funkce
  - skryté vrstvy - relu
  - výstupní vrstva - sigmoid

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 28)                224
_________________________________________________________________
dense_1 (Dense)              (None, 14)                406
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 15
=================================================================
Total params: 645
Trainable params: 645
Non-trainable params: 0
_________________________________________________________________
```

ANN - Model

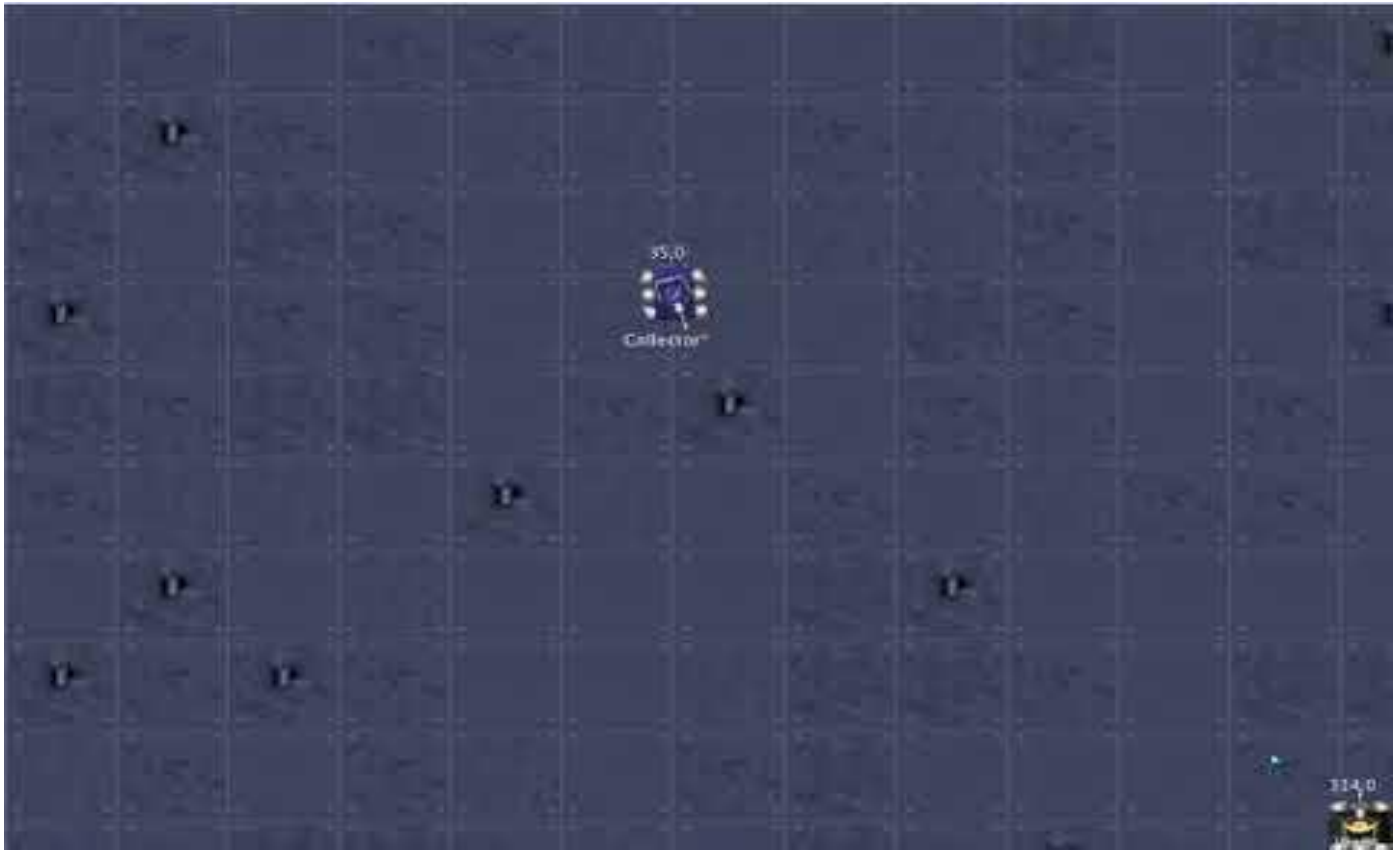
- vstupné dáta
  - X - pozbierané dáta behom hry
  - Y - predikované výsledky

```
m.compile(optimizer = 'adam', loss = 'mse', metrics = ['accuracy'])
```

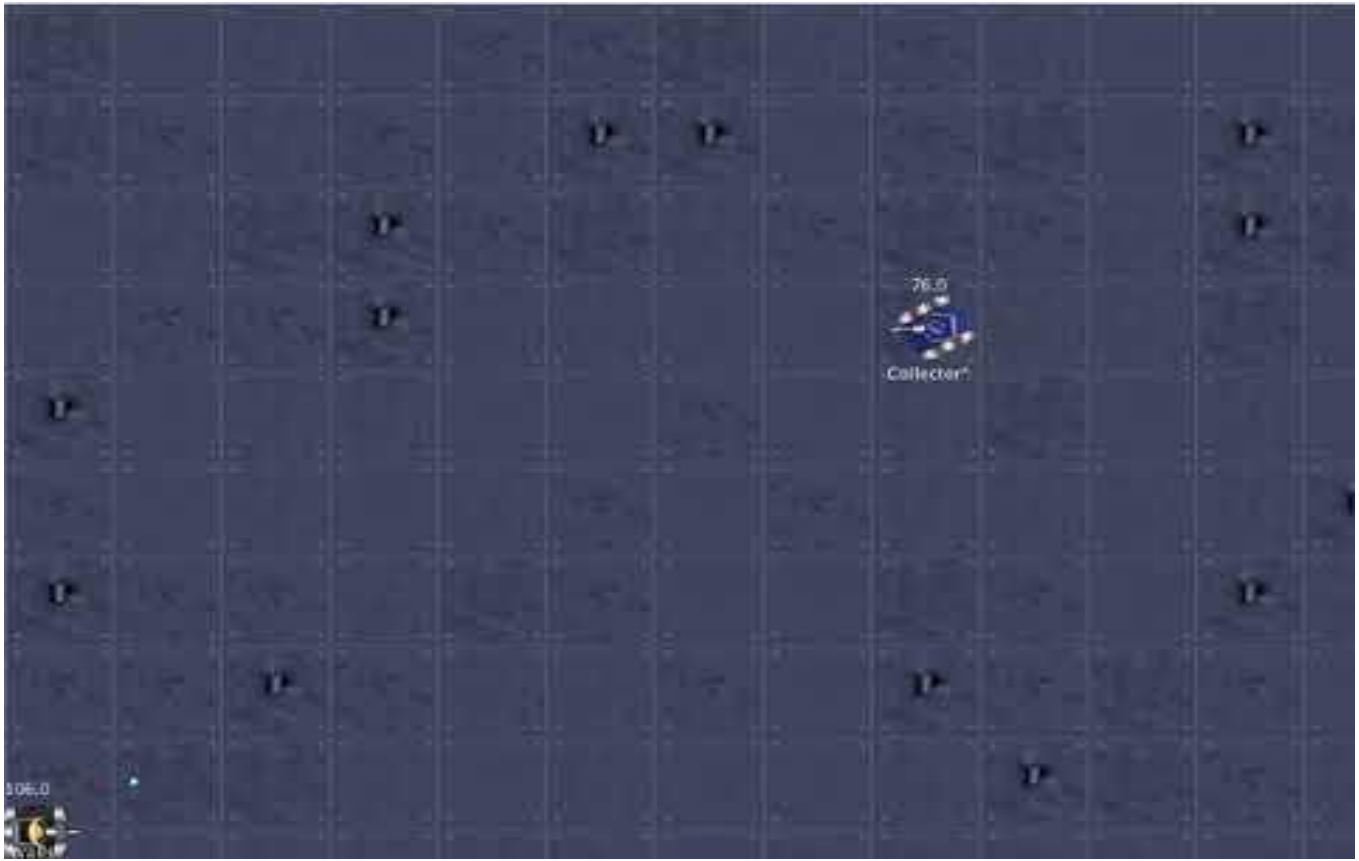
- optimizer
- loss function
- váhy

Naše implementace

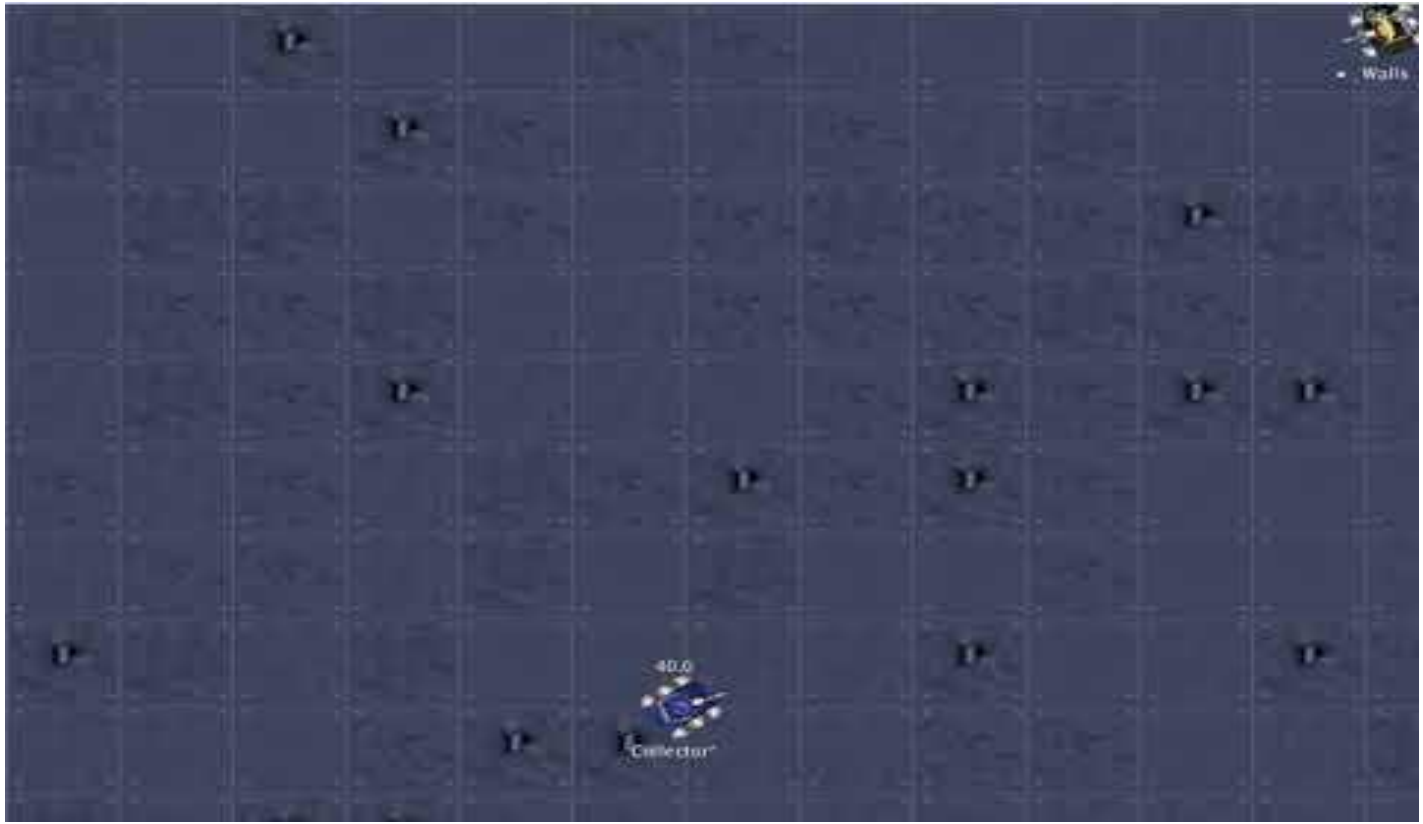
- třídy Robot a Collector v javě
  - komunikace se serverem v pythonu
  - sbírání dat
- třída Agent v pythonu
  - ovládání tanku
- pomocné funkce pro čištění dat a učení
- chování tanku
  - trénování => neustálé otáčení hlavně, výstřel po naskenování protivníka
  - zhodnocení => otáčí podle parametru gunToTurn, výstřel při predikované hodnotě > 0.9
- sbírání dat i s naučeným modelem



SVM vs Walls

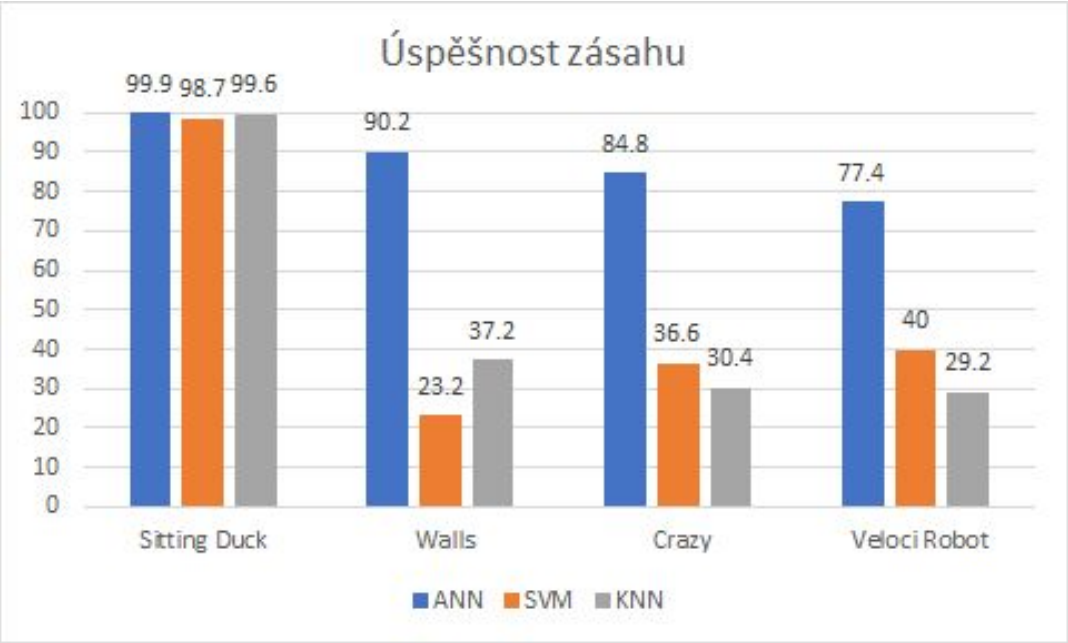


kNN vs Walls



ANN vs Walls

Výsledky použitých algoritmů



Shrnutí

- Načasování střílení s dobrými výsledky 
- Vyzkoušeno více algoritmů učení 
- Sesbírána data od různých protivníků 
- Implementování python agenta pro reálné použití našeho tanku 

Děkujeme za pozornost!

https://github.com/Koncpa/MPC-PDA-STROJOVE-UCENI.git