

SAMOSTATNÁ PRÁCE Č. 5



Kurz: MPC-PDA, Advanced Data Structures and Algorithms

Lektor: Radim Burget, Angelika Mezina

Autor: Tadeáš Zachoval, Adam Turek, Tomáš Závada, Jakub Čurilla

Samostatná práce 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.



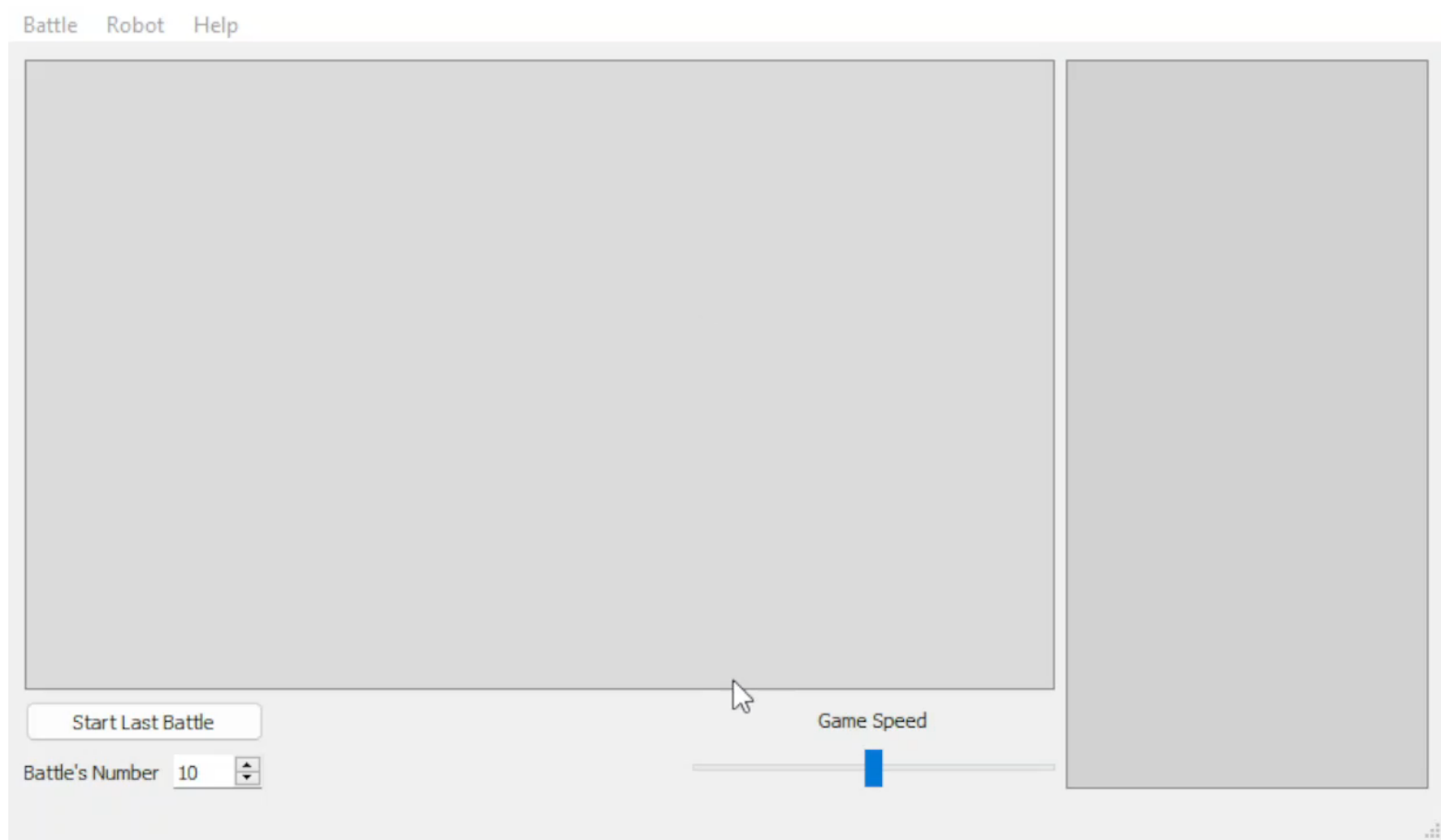
Motivace

- Zpřesnění střelby robota
- Porozumění principů strojového učení
- Hlubší seznámení se s dostupnými knihovnami
- Zápočet z MPC-PDA

Řešení

- Sběr a normalizace dat
- Sestavení neuronové sítě
- Naučení neuronové sítě
- Využití naučené sítě

Ukázka 1



Sběr a normalizace dat

- CSV soubor
- Po dopadu střely
- Distance, velocity, gun heading, position
- Normalizace hodnot

```
=====
Dataset
-----
      distX    distY  myGunHeading  velocityX  velocityY  hit
0      0.032000 -0.188817      0.416667      0.00000      0.00000  0.0
1      0.351613 -0.059433      0.777778      0.00000      0.00000  0.0
2     -0.112338 -0.277200      0.513889      0.00000      0.00000  0.0
3      0.166725  0.057683      0.861111     -0.59875     -0.18125  0.0
4     -0.200000 -0.008333      0.013889      0.00000      0.00000  0.0
...      ...      ...      ...      ...      ...      ...
78589 -0.200000 -0.266667      0.000000      0.00000      0.00000  0.0
78590  0.261125  0.176400      0.805556     -0.56375     -0.26875  1.0
78591  0.285950  0.100367      0.750000      0.00000      0.00000  0.0
78592 -0.084887  0.088733      0.361111      0.47875      0.40125  0.0
78593  0.024362  0.692300      0.041667      0.00000      0.00000  1.0

[78594 rows x 6 columns]
=====
```

```
# Scale values between 0 and 1
if scale_values:
    df["distX"] /= 800
    df["distY"] /= 600
    df["myGunHeading"] /= 360
    df["velocityX"] /= 8
    df["velocityY"] /= 8
```

Sestavení neuronové sítě

- Knihovna tensorflow.keras
- Vrstvy Dense

```
m = keras.Sequential([
    keras.layers.Input(shape=(number_of_fields-1)),
    keras.layers.Dense(32, activation='relu'),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(256, activation='relu'),
    keras.layers.Dense(1, activation='sigmoid'),
])

print(m.summary())

# optimizer for correcting weights in machine learning, loss function for calculating mistakes
m.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Naučení neuronové sítě

- Knihovna tensorflow.keras
- Vrstvy Dense
- Optimizer adam
- Úspěšnost přibližně 90%

```
Epoch 1/100
1216/1216 [=====] - 3s 1ms/step - loss: 1.0280 - accuracy: 0.6629 - val_loss: 0.4771 - val_accuracy: 0.7748
Epoch 2/100
1216/1216 [=====] - 2s 1ms/step - loss: 0.7936 - accuracy: 0.8017 - val_loss: 0.4501 - val_accuracy: 0.8193
Epoch 3/100
1216/1216 [=====] - 2s 1ms/step - loss: 0.7124 - accuracy: 0.8473 - val_loss: 0.4251 - val_accuracy: 0.8219
Epoch 4/100
1216/1216 [=====] - 2s 1ms/step - loss: 0.6754 - accuracy: 0.8667 - val_loss: 0.3893 - val_accuracy: 0.8537
Epoch 5/100
1216/1216 [=====] - 2s 1ms/step - loss: 0.6530 - accuracy: 0.8786 - val_loss: 0.3453 - val_accuracy: 0.8715
Epoch 6/100
1216/1216 [=====] - 2s 1ms/step - loss: 0.6393 - accuracy: 0.8852 - val_loss: 0.3273 - val_accuracy: 0.8969
Epoch 7/100
1216/1216 [=====] - 2s 2ms/step - loss: 0.6293 - accuracy: 0.8899 - val_loss: 0.4925 - val_accuracy: 0.8422
Epoch 8/100
1216/1216 [=====] - 2s 2ms/step - loss: 0.6224 - accuracy: 0.8929 - val_loss: 0.3426 - val_accuracy: 0.8791
Epoch 9/100
1216/1216 [=====] - 2s 1ms/step - loss: 0.6128 - accuracy: 0.8952 - val_loss: 0.3354 - val_accuracy: 0.8944
Epoch 10/100
1216/1216 [=====] - 2s 1ms/step - loss: 0.6132 - accuracy: 0.8955 - val_loss: 0.3787 - val_accuracy: 0.8728
```

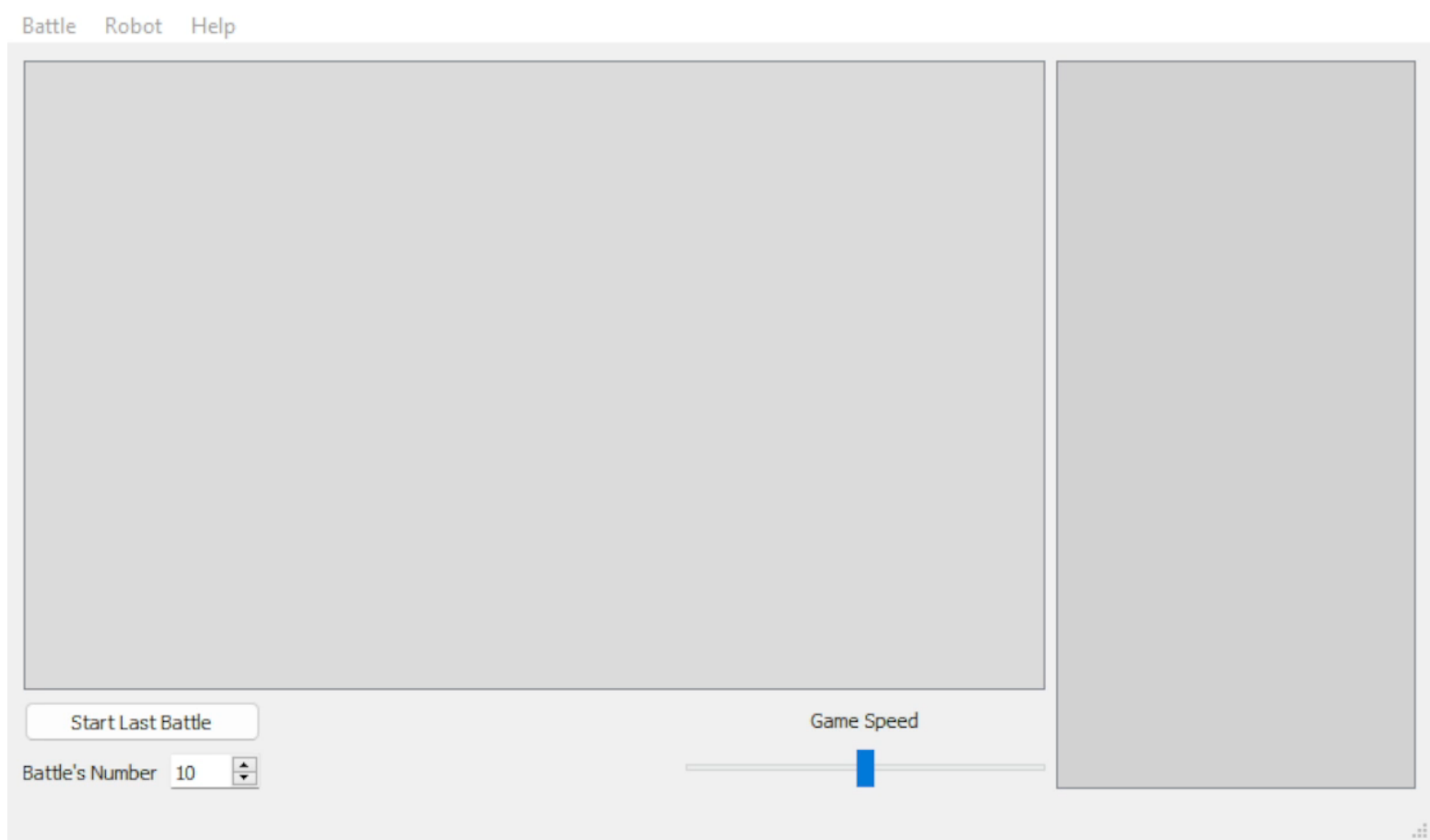

Výsledky

- Hodnotu confidence rozhodne neuronová síť
- Střelba podle velikosti hodnoty confidence

```
1/1 [=====] - 0s 18ms/step  
Received data: -55.57;-172.16;165.00;0.00;0.00 -> confidence 0.9837665557861328 decision: shoot
```

```
1/1 [=====] - 0s 19ms/step  
Received data: -226.34;-356.43;170.00;0.00;0.00 -> confidence 0.1126442551612854 decision: do not shoot
```

Ukázka 2



Zhodnocení

- Sběr a normalizace dat
- Sestavení neuronové sítě
- Naučení neuronové sítě
- Využití naučené sítě

Děkuji za pozornost!