Image [1]

SYSC 5703 – Integrated Database and Cloud Systems

Project Presentation

Deep Learning-Based Sequential Model for Predicting Victories in Video Games

Group #1:
Ali Mohammadi Esfahani
Azhar Syed
Françoise Blanc
Sandeep Reddy Venna

April 6, 2022

Introduction

- **Challenge:** Predicting victories in the video game *Tactical Troops: Anthracite Shift*
- **Goal:** Implement a new approach that may provide better performance over the top-performing solutions
- **Our solution:** Create a sequential dataset generated from the game logs to train Deep Neural Networks to predict victories

Image [2]

Introduction

- **Main contributions:**
  - A review on the use of Deep Neural Networks using sequential data for predicting victories;
  - A demonstration of a new approach using sequential data to predict victories in the video game *Tactical Troops: Anthracite Shift;*
  - Training and evaluation of 6 different Deep Neural Networks; and
  - Recommendations to improve our proposed approach.

Outline

1. **Related Work**
2. **Methodology**
   - ❖ Game Description
   - ❖ Data Description
   - ❖ Data Preprocessing
   - ❖ Models Description
   - ❖ Models Implementation
3. **Demonstration**
4. **Results**
5. **Discussion**
6. **Conclusion**

Related Work

- **Top-performing solutions to the challenge [3]**
    - gradient boosting decision trees variants (e.g., LightGBM and XGBoost)
    - exhaustive feature engineering and feature selection techniques
    - gaps:
        - limited comparison with other algorithms
        - limited use of the truncated data format

Our proposition: better exploit the truncated log files by creating sequential data over each game

RNN: Recurrent Neural Network
GRU: Gated Recurrent Unit
LSTM: Long Short Term Memory
SVM: Support Vector Machines
CNN: Convolutional Neural Networks
SAE: Stacked Autoencoder

Related Work

- Predicting victories using sequential data is a task that falls under time series classification
- Best Deep Neural Networks for time series classification [8]:
  - Residual Network (ResNet)
  - Fully Convolutional Neural Networks (FCN)
- Deep learning-based sequential model for predicting victories in video games and sports:

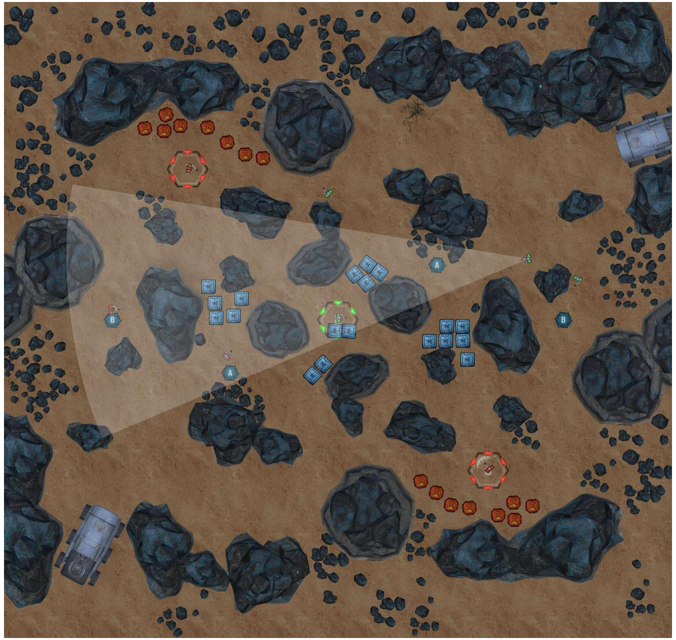
	Study	Video game / sports	Algorithms
VIDEO GAME	Silva et al. [4]	League of Legends	**Simple RNN***, GRU, and LSTM
	Qi et al. [5]	Defense of Ancients 2	GRU, **LSTM***, SVM, CNN, and SAE
SPORTS	Watson et al. [6]	Rugby	CNN, GRU, **LSTM***, BI-RNN, and **CNN-RNN***

*best algorithms

Carleton University

Game Description

- Each game is between 2 players
- Each player has 4 controllable units
- Characteristics of each unit:
  - unit type;
  - health points;
  - action points;
  - weapons, and
  - gadgets.
- 2 game modes:
  - Devastator
  - Domination



Data Description

The data for the challenge is provided in 4 formats:

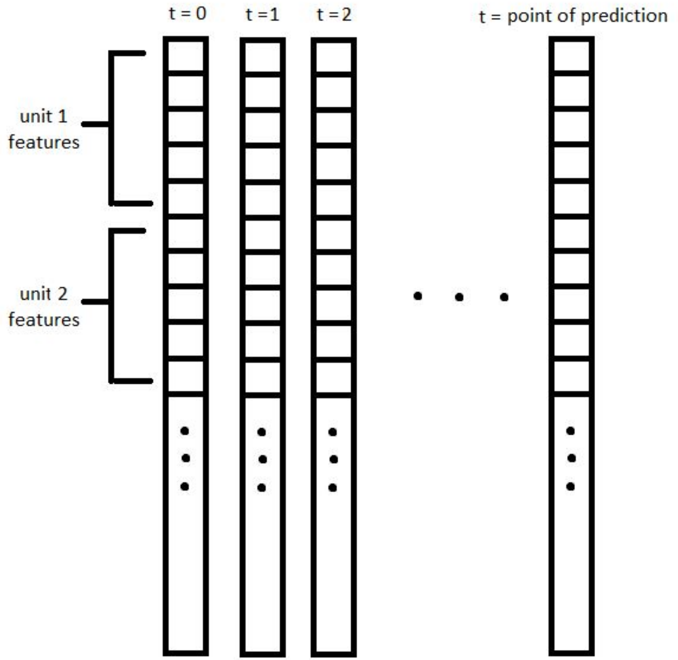
- **Tabular data**:
  - Contains the aggregated features of each game at the point of prediction
  - The binary target label (i.e., winner of the game) is present only in this format
- **Game screenshots**:
  - A screenshot of the complete map at the point of prediction
  - Image-based data
- **Flattened logs**:
  - Final states of all the registered objects at the point of prediction in a game
- **Truncated logs**:
  - State changes throughout a game until the point of prediction

Data Preprocessing

- **Feature selection**:
  - Most informative and generic features
  - Selected features:
    - health points;
    - action points;
    - weapons (categorical); and
    - unit type (categorical).

- **Modelling game logs as time-series data**

- **Handling categorical values**: binary encoding



Data Preprocessing

- **Generator based training**
- **Padding and masking technique**
- **Training and validation split**: 8:2

Models Description

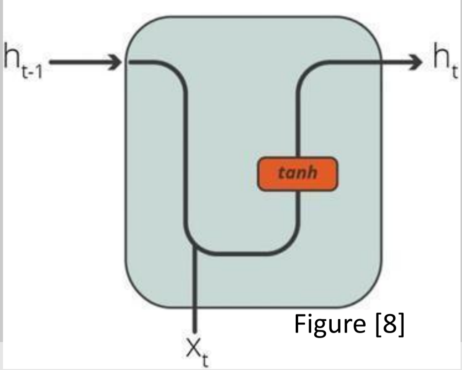


Figure [8]

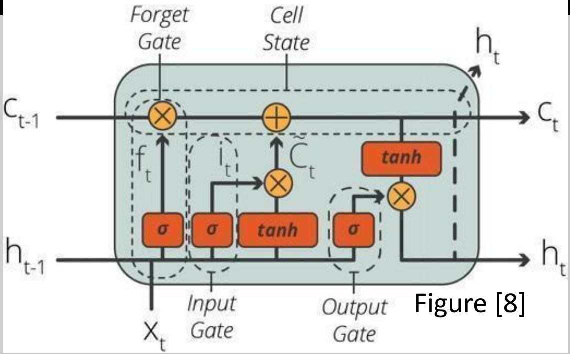


Figure [8]

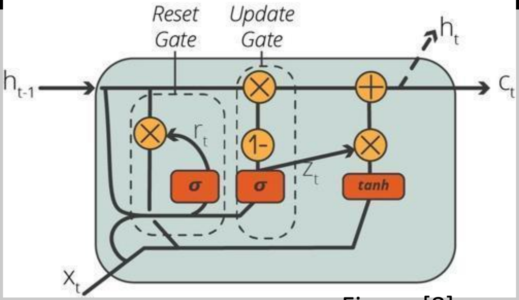


Figure [8]

- Backpropagation through time
- Powerful for modelling time series

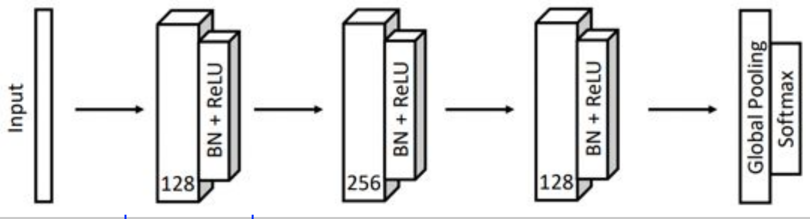
- Learns lengthy-time period dependencies
- More complex than simple RNN

- Improved version of simple RNN
- Faster than LSTM

Carleton University

Models Description

[redacted]



Simplified FCN

Figure [9]

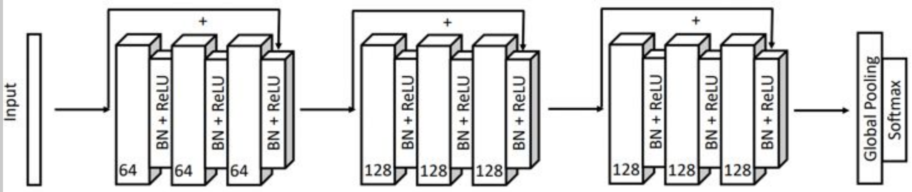


Figure [9]

- Special CNN without fully connected layers
- Can be applied to inputs of any size
- **Simplified FCN**: One block of original FCN

- Handles vanishing/exploding gradient problem in Deep Neural Networks
- 9 convolutional layers and 1 global average pooling layer

Models Implementation

- **Performance metric:**
  - Area Under the Curve (AUC)

- **Hyperparameter tuning:**
  - Hyperparameters: number of neurons, batch size, activation functions, dropout, learning rate, optimizers, etc.
  - Hyperparameter optimization framework:

OPTUNA

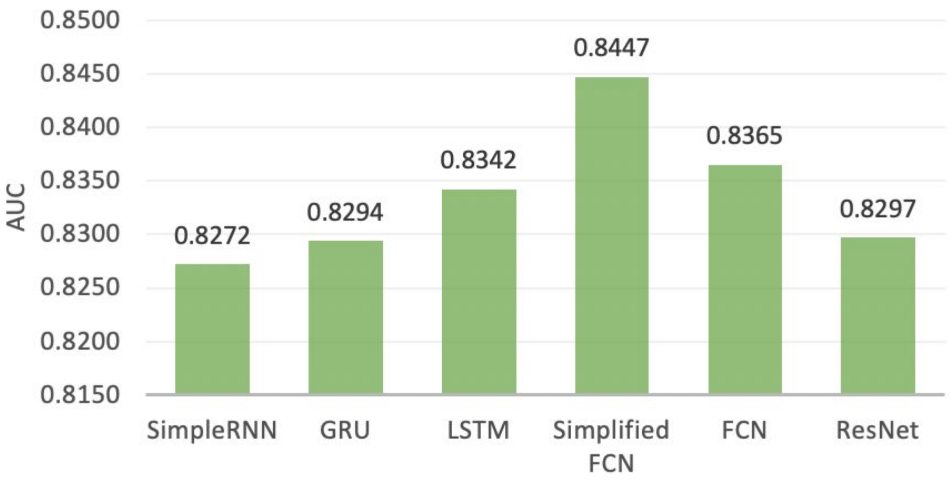
Model	Hyperparameters	Optimized value
SimpleRNN GRU LSTM	number of neurons	43
	dropout	0.1107
	activation function	sigmoid
	weight constraints	1
	initializer	Glorot uniform
	optimizer	Adam
	learning rate	0.0008
	batch size	188
Simplified FCN	number of filters	36
	kernel size	12
	initializer	He normal
	optimizer	SGD
	learning rate	0.0003
	batch size	32

Demonstration

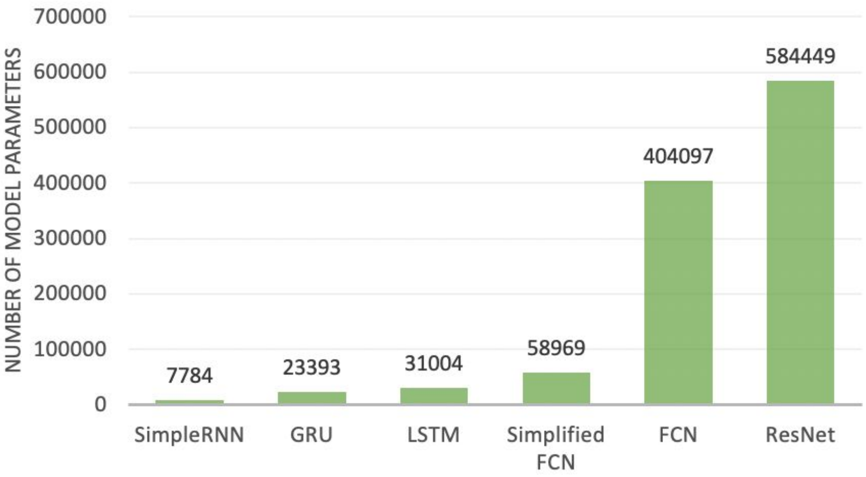
- Project source code available at: https://github.com/azhartalha/PredictingVictories

Results

- Models performance:



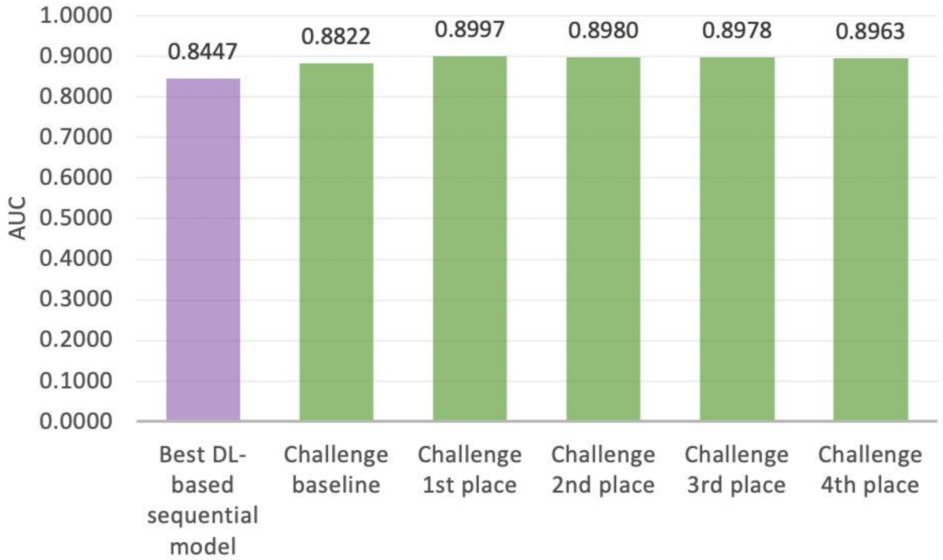
- Models complexity (number of parameters):



Discussion

- Advantages of our approach:
  - does not require extensive feature engineering
  - easily adaptable to other video games
- Hyperparameter optimization is not optimal due to limited computer resources

- Comparison with the top-performing solutions to the challenge:



Carleton University

Conclusion

- **Key points**
  - Trained and evaluated 6 Deep Neural Networks to predict victories using sequential data
  - Simplified FCN achieved the best AUC among all tested models
  - Major advantage: elimination of feature engineering

- **Possible future work**
  - Combination of sequential data and fixed data
  - Train other Deep Neural Networks (e.g., CNN-RNN)
  - Consider ensemble model (e.g., gradient boosting decision tree + simplified FCN)
  - Improve hyperparameter optimization
  - Test our generalized approach on other video games

References

[1] S. Sharma. “Explained: Deep Learning in Tensorflow — Chapter 0.” Towards Data Science. https://towardsdatascience.com/explained-deep-learning-in-tensorflow-chapter-0-acae8112a98 (accessed Mar. 29, 2022).

[2] Tactical Troops. https://tacticaltroops.net/ (accessed Mar. 29, 2022).

[3] M. Matraszek, A. Janusz, M. Świechowski and D. Ślęzak, "Predicting victories in video games - IEEE BigData 2021 Cup report," *2021 IEEE International Conference on Big Data (Big Data),* 2021, pp. 5664-5671, doi: 10.1109/BigData52589.2021.9671650.

[4] A. L. C. Silva, G. L. Pappa, and L. Chaimowicz, "Continuous outcome prediction of League of Legends competitive matches using recurrent neural networks," *Proceedings of SBGames 2018*, 2018, pp. 639-642.

[5] Z. Qi, X. Shu and J. Tang, "DotaNet: two-stream match-recurrent neural networks for predicting social game result," *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM),* 2018, pp. 1-5, doi: 10.1109/BigMM.2018.8499076.

[6] N. Watson, S. Hendricks, T. Stewart, and I. Durbach, "Integrating machine learning and decision support in tactical decision-making in rugby union," *The Journal of the Operational Research Society*, vol. 72, no. 10, Taylor & Francis, 2021, pp. 2274–85, doi: 10.1080/01605682.2020.1779624.

[7] Q. Zhang, X. Zhang, H. Hu, C. Li, Y. Lin, and R. Ma, "Sports match prediction model for training and exercise using attention-based LSTM network." *Digital Communications and Networks*, 2021, doi: 10.1016/j.dcan.2021.08.008.

[8] O. G. Yalçın, "Applied neural networks with TensorFlow 2 : API oriented deep learning with Python," 1st ed. 2021., Apress, 2021, doi: 10.1007/978-1-4842-6513-0.

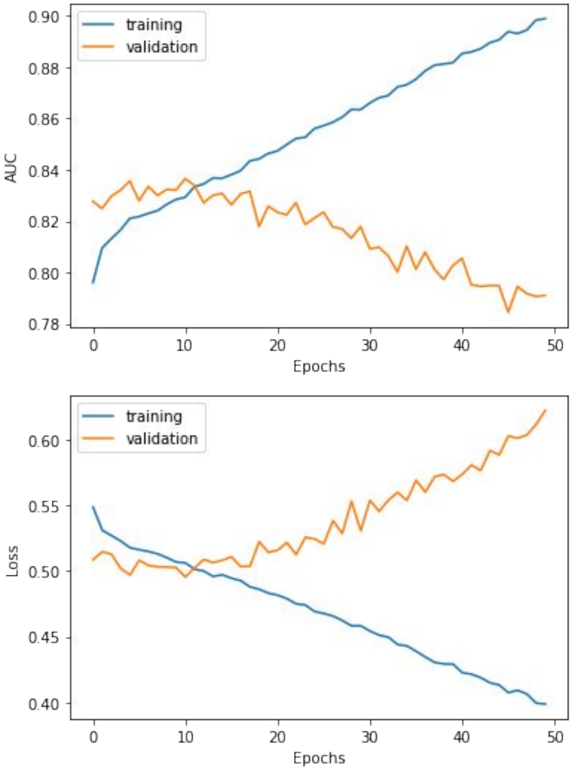
[9] Z. Wang, W. Yan and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1578-1585, doi: 10.1109/IJCNN.2017.7966039.

Thank you!

Any questions or comments?

Extra slide

- **FCN**



- **Simplified FCN**