Using conditional GANs to develop a realistic human-robot interaction simulator

Guilherme PENEDO, 2022, Embedded Systems, ISAE
Giorgio ANGELOTTI, Caroline CHANEL, Nicolas DROUGARD, ISAE

1 Introduction

To design effective human-robot systems one key approach is to have a dynamic assignment of tasks between the human operator and the automated system. To this end, a policy can be developed to decide when to transfer control from the human to the automated system and the reverse, based on the current state and the operator's mental condition.

Reinforcement Learning can be used to create a generative model to mimic a human's behaviour for a particular mission state, thus allowing for the creation of a human-robot interaction simulator that can be used to further optimize a supervision policy, without the logistic difficulties inherent to observing an actual human responding to the different situations.

Human behavioural data is multimodal in the sense that, for a given context, multiple different actions are possible and perfectly reasonable. As such, common deterministic approaches that attempt to produce the average prediction which minimizes a distance metric from the real value are not adequate. By using a Generative Adversarial Networks (GANs) [1] model, the multimodality of the data can be preserved: for a given moment in the game there are multiple possible predictions, since sampling multiple times from the GAN will generate different outputs, for the same game state.

Conditional Generative Adversarial Networks (CGANs) [2] add additional inputs to both the generator and the discriminator, in order to allow the GAN's generated data to be conditioned on some criteria. In this project, a CGAN will be used to condition the data generated on certain variables of the firefighting robot's game state, so that the generated data will depend on the current state of the game. Later on, Long short-term memory networks (LSTMs) are also introduced, allowing the input to include data from multiple previous time intervals.

Different architectures and training methods involving GANs are considered and quantitative and qualitative methods of validation are proposed. The models developed in this project are able to, given a history of game states and control actions in the "Firefighter Robot Mission" game, generate a possible future control action that a real human could also have taken. These models may, therefore, be used to estimate a supervision policy, thus fulfilling the project's main goal.

2 Dataset: Firefighter Robot Mission (FRM)

An experiment was developed in [3], in which a game, called Firefighter Robot Mission[1], was set up in order to collect behavioral and psycho-physiological features from human operators. The game itself has a graphical user interface (GUI), through which the player is expected to control the movement of a robot (using his arrow and space keys) that has a water deposit, in order to extinguish fires on nearby trees.

The focus of the models developed in this project was on generating a binary output for each key: whether each key (the "front" key, for example) was or was not pressed on a given 1 second time interval. To this end, the input features used were "robot_mode", "robot_x", "robot_y" and "robot_theta" (whether the robot was in automatic or manual control mode and its location).

As part of preprocessing, the dataset was split into samples with a 6 seconds duration: 5 seconds of *observations* and 1 of the real*prediction*: the models generate a prediction from the observations, which is made up of 5 values indicating whether each of the 5 keys was pressed.

[1] http://robot-isae.isae.fr

3 Model Details

The main proposed model uses an observations encoder with LSTM layers to encode the game state and keys pressed information. A diagram of the model's architecture can be seen in Figure 1.

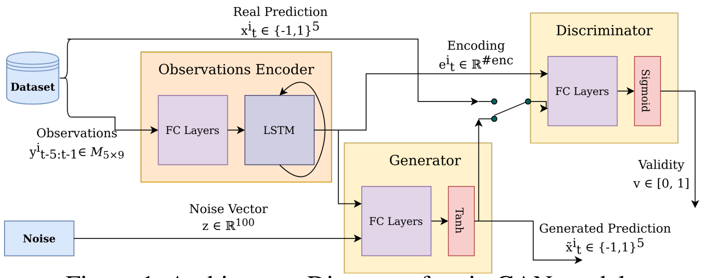

Figure 1. Architecture Diagram of main GAN model

Observations Encoder: the observations, which consist of the 4 state variables and the 5 keys' presses for the 5 preceding time intervals, are encoded by the "Observations Encoder". Its outputs will be the condition of our GAN.

Generator: it has two inputs, the observations encoding, which is the GAN's condition, and the noise/latent vector. Its outputs are 5 values in the $[-1, 1]$ range, one for each key (we consider that a key was pressed if the generated value is positive and not pressed otherwise).

Discriminator: it also receives the observations encoding as an input, but the second input is a prediction, either the actual prediction from the dataset or one obtained from the generator. This network's output is the probability of the prediction given as input being a real prediction for the given encoded observations. It is this value that is then used for adversarial training.

4 Model Validation

As this dataset is unbalanced, where most outputs are likely to correspond to a key not being pressed, a metric such as accuracy would not be useful. Therefore, considering a keypress as the positive class and not pressing a key as the negative class, the F1-Score, which is a metric combining precision and recall into a single number, was used in this project to quantify the models' performance.

Cross validation with 4 folds was implemented, and the results were averaged over these 4 folds to produce the model's final

Table 1. Quantitative evaluation results for all models, out of k validated, with k = 5 samples. F1-score, precision and recall values were averaged over batches and folds; then, the epoch with the highest average F1-score was chosen (Best F1-score for a model). The precision and recall values shown are the ones associated with that epoch. The best values are displayed in **bold**.

Model	GAN-GP	CGAN-GP	RCGAN-V20
Best F1-score	19.95%	48.82%	**62.55%**
Precision/Recall	14.07%/34.27%	68.48%/37.93%	72.89%/54.77%

evaluation score. Following the methodoloy presented in [4], out of k validation was performed.

4.1 Baselines

The model detailed in Section 3 was compared against the following baselines: *GAN-GP*, A Wasserstein GAN trained with gradient penalty without any conditions; and *CGAN-GP*, same as the *GAN-GP*, but the game state of the time interval immediately preceding the current one are used as a condition to generate the next time interval's keypresses.

The results obtained for the different models can be seen in Table 1. Looking at GANGP, it seems clear that not having access to any observational/previous game state data has a great impact in model performance, as this model has an extremely low F1-score. On the other hand, CGANGP has a significantly higher score, not too far from that of the RCGAN models. The difference observed may be due to the fact that RCGAN has access to more observational data, as it takes into account the full 5 preceding seconds and not only the last one.

5 Experiments

Additional experiments and data analysis were performed on the model with the highest performance, "RCGAN-V20". As an example, on Figure 2 were plotted the frequency of generated samples, per game region, in which the right key was pressed. It can be seen that there seems to be a higher proportion of right keypresses on the left side of the board. This seems intuitive: when a player is on the border of the map he will likely try to move towards the center.

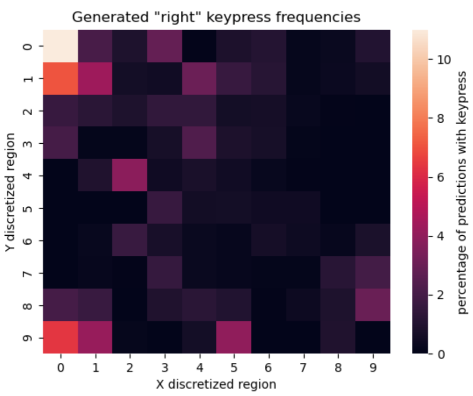

Figure 2. Geographical frequencies for right key

In order to visualize some samples generated by the model, the robot's 5 consecutive positions were plotted on the game map as black triangles facing in the same direction the robot was facing. One sample has been included in Figure 3, where the model seems to have taken into account the leftwards curve the robot made, and has predicted the left key (the model attributed the value of "1.0" to this key) as being pressed.

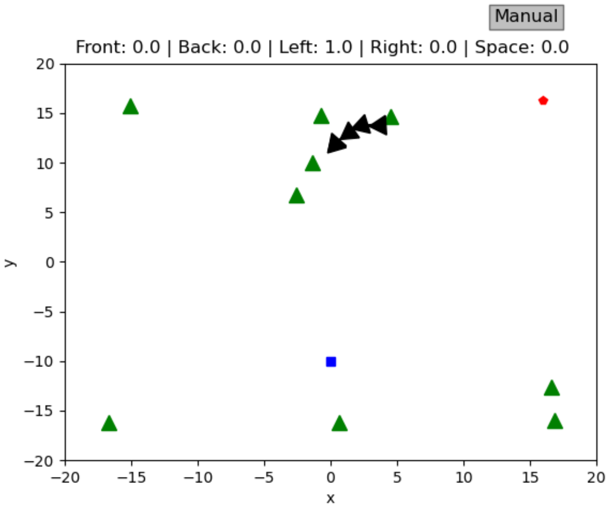

Figure 3. Sample with left keypress

6 Conclusion

Multiple models were introduced over the course of this project in an attempt to properly create a generator of human actions. The CGAN and RCGAN models, with its conditional features, are able to take into account the previous game history, incorporating information such as the robot's position, to more accurately predict reasonable future actions. Additionally, by using Recurrent Neural Networks (RNNs), the RCGAN model may exploit multiple game instants to further improve the predictions.

In terms of the quantitative validation performed, it should be noted that even with the approach that was followed, given the nature of the precision and recall metrics, their trade-off relation and, that effect on the F1-score, there is likely a theoretical limit to the maximum F1-score that may be obtained for this problem by a model. Finding good quantitative metrics for this problem is a difficult task, and it may be that other, more appropriate metrics may be defined in the future.

Further work may be applied to the study and experimentation of different GAN architectures and loss functions, such as, for example, the Info-GAN architecture used in SocialWays (Javad Amirian et al., 2019), as well as to additional validation metrics and strategies. Given the unbalanced nature of the dataset, another avenue for improvement would be to collect additional high quality data.

References

[1] Ian J. Goodfellow et al. Generative adversarial networks, 2014.

[2] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[3] Caroline Chanel et al. Towards mixed-initiative human-robot interaction: Assessment of discriminative physiological and behavioral features for performance prediction. *Sensors, Special issue Human-Machine Interaction and Sensors*, 20(1): 1–20, January 2020. doi:10.3390/s20010296.

[4] Agrim Gupta et al. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2255–2264, 2018.