Deep reinforcement learning method for the shepherding problem

Žan Jonke
zj0527@student.uni-lj.si

***Abstract*—In this work we tackle the shepherding problem using a deep reinforcement learning method. We make a brief overview of what the problem entails and proceed to discuss the various solutions and approaches used to solve it and the reinforcement learning methods that have been shown to be successful. Afterwards the used reward policies, for agent training are introduced, followed by the evaluation of their performance, utility and shortcomings. The acquired models are capable of herding sheep only under certain conditions, and a major concern remains the generality of the solution. In the end a few suggestions are provided of how one could tackle the problems that remain unsolved. The code of our project is available at the following link.**

I. Introduction

The herding of sheep is an interesting collective behaviour problem, where a single agent, a dog, is tasked with herding a flock of sheep to a barn or an enclosure. These sheep act as a swarm system, that tries to evade the shepherding dog, while maintaining flock cohesion. As a system guidance problem, a solution to such a task has many other applications beyond the field of agronomy, such as crowd control [1], robotics [2], animal repulsion [3], and even environmental cleanup [4].

Our motivation with this project is to create a deep reinforcement learning method, capable of teaching an agent to guide a swarm system to a desired location. With deep learning methods becoming ever-present, the study of Deep Q-Learning applications on this domain would be a contribution to the field.

In the Related work section, we present the already developed architectures, and how their solutions were applied in the field. In the Methods section, we discuss our experimental framework, and the building block of our proposed architecture, used to train the agent. In the Experiments and Results, we discuss the training and experimental process, and the resulting agent behaviour. In the Conclusion section, the issues that were noticed during implementation are presented, and possible improvements are discussed.

II. Related work

One of the earliest solutions to this problem was introduced by Strömbom *et al.* [5], where the authors proposed a heuristic algorithm that splits the shepherding task into two subtasks - driving and collecting. The authors noticed that when a shepherd is driving a herd towards a goal, the sheep tend to start splitting along the middle. To combat this they added a subtask that collects sheep that stray too far from the center of the flock. Another approach by Licitra *et al.* [6], [7] tackled this problem by dynamically switching between different target agents.

A machine learning approach based on Strömbom's algorithm was developed by Go *et al.* [8], where a model was trained using SARSA, a classical reinforcement learning method. It was successfully proven that a reinforcement learning algorithm is capable of learning a policy that solves the shepherding problem, as described by Strömbom. It achieves this by setting a series of subgoals, that the model solves in order.

Nguyen *et al.* [9] implemented a deep hierarchical reinforcement learning (DHRL) method for the herding of a small number of robotic sheep using a drone. The method employs the same idea Strömbom had, where the algorithm is split into two subtasks, driving and collecting, except that these tasks are learned using DHRL. A similar approach to deep reinforcement learning was discussed by Zhao *et al.* [10], where instead of introducing some knowledge of the domain to the decision making process, a hierarchical architecture of a high-level and a low-level agent is proposed, in which the high-level agent sets abstract goals for the low-level agent to achieve.

Hu *et al.* [11] have shown that the solution to this problem has a real-world application. The authors created a group of robotic drone shepherds and created an algorithm that is capable of herding a flock of real-world sheep. The drones coordinate with each other to complete the task of guiding the sheep towards an enclosure. This approach to shepherding has been shown to increase the safety of farmers, as well as the well-being of animals [12].

III. Methods

For the model to be trained, a framework was created. We implemented an environment that handled sheep behaviour, and created a deep reinforcement learning model, based on Deep Q-Learning. This model also included some custom components that we introduced.

A. Sheep behaviour

To model the movement behaviour of sheep, we used the approach first introduced by Strömbom *et al.* [5]. The movement of individual sheep is dependent on the position of the

shepherd and the positions of the sheep that are in its neighbourhood determined in two different ways. The shepherd acts as a repulsive force on each sheep, giving him the ability to control their movement. The neighbouring sheep acts as both a repulsive and attractive force. To keep each sheep from colliding with its neighbours, all the sheep within a certain radius act as force of repulsion and drive the sheep in the mean opposite direction. However to keep them grouped, each sheep is also attracted to the centre of mass of its n nearest neighbours. In order to make the movement more realistic, inertia is also taken into account. The environments finish when any of the following criteria are met: all the sheep are close enough to the goal, the number of actions that took place is above a certain threshold or the shepherd tries to produce a move which will place him out of frame.

B. State presentation

To present the state of the environment at any given moment an RGB image was used. When constructing the visual representation of the state the following was taken into consideration:

- the state should reflect the number, location and the spatial spread of the sheep
- the state should reflect the sheep which are astray
- the state should reflect the sheep which are located near enough the goal
- the state should reflect the position of the goal along with its radius

Given the above considerations the state presentations shown on figure 1 were experimented with. At any given state a total of 9 moves are available ie. 8 moves for each 8th of circle with 1 additional move which keeps the current position of the shepherd.

C. Reward policy

Reinforcement learning requires a reward policy to be defined, which during training tells the agent how successful an action was in achieving a certain goal. The reward function can be split into two parts:

- Collecting reward : The reward is proportional and conversely dependant on the number of astray sheep, which are determined in the same manner as in [5].
- Driving reward : Similarly as above, the reward is dependant on the number of sheep, however in this case on the ones which are close enough to the goal. Additionally another value is added which inversely dependant on the distance of the centroid of the sheep from the goal.

D. Model

To expand the capabilities of deep reinforcement learning the actor-critic paradigm was used [13], which optimizes both policy and state approximation through tow different networks, that share the same backbone. One is used to approximate the policy function and one is used to approximate the state value. For each network it's respective loss is defined ie. policy loss and for the policy network and MSE loss for the state value approximation. Our approaches used a ResNet18 [14] backbone to generate a state embedding which was then further processed by each network individually.

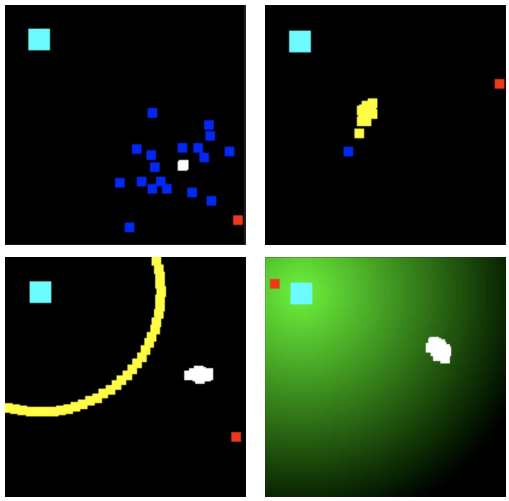

Fig. 1. State presentation examples. Herded sheep are presented with yellow color, astray sheep in blue, sheep close to goal in yellow and the shepherd in red. The goal position is presented with the cyan color. Attempts were made to encode the distance of within the presentation. The first approach was to include a radius circle around the goal position, where the color of the sheep would change should the pass it. The second approach was to include a green color whose intensity is inversely correlated to the distance from the goal.

E. Implementation and training

All implementation was done using Python and openAI gym environemnt, which is a Python package used for development of reinforcement learning models. The model and training pipeline was developed using PyTorch and we used a PyTorch-gym interface called ptan, which handles most of the data gathering technicalities. The training was executed on the Arnes super computing cluster.

IV. Experiments and results

A. Experiments

Our initial goal was to train one model which was able to both gather and drive sheep towards a goal regardless of herd position, the number of sheep, the starting position of the shepherd or the position of the goal. To tackle the inherent complexity of the problem we employed the framework of curriculum learning [15] which is a training strategy that trains a machine learning model from easier data to harder data. In the sheepherding environment defined above this was implemented in the following ways:

- gradual increase of sheep within the environment
- gradual increase of the number of astray sheep
- gradual increase of the goal radius

The above mentioned approaches did not yield a significant increase in performance.

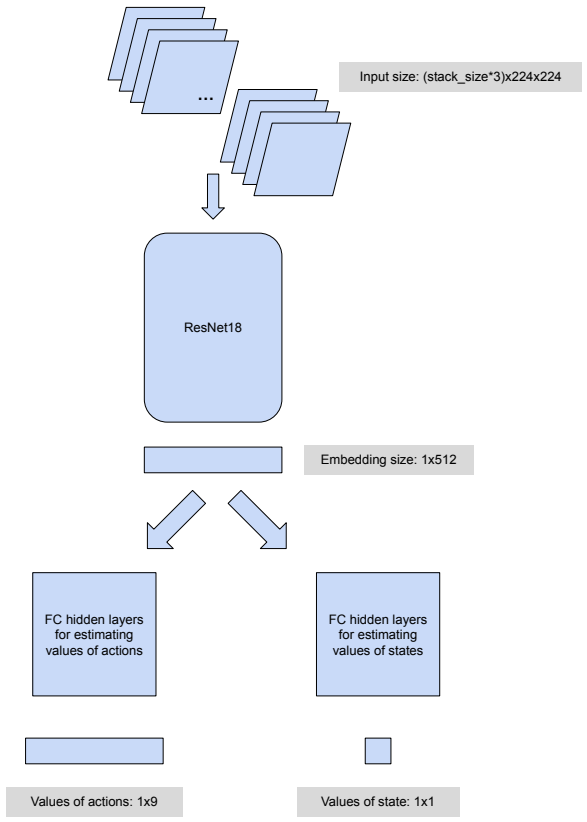

Fig. 2. Model architecture. A ResNet18 backbone was used to generate state embeddings. Two additincal fully connected networks were attached, one for policy optimization and one for state value optimization.

To give the model the ability to learn the motion of the sheep and its effect on the sheep, stacking of the subsequent frames was applied, so that a state was presented not as a single RGB image but rather as a sequence.

We also experimented with different learning hyperparameters ie. learning rate, batch size and number of hidden layers within the policy and value networks. Additionally we tried to optimize the environment hyperparameters as well.

B. Results

Unfortunately none of our experiments yielded a model which was able to generalize well or was able to solve the environment regardless of the above mentioned variables. However in a constrained environment, where the starting position of the dog is always the same, the number of astray sheep is relatively small and the goal radius is large enough, the model was sometimes able to herd the sheep and subsequently lead them towards to goal. However for a valid or more concrete conclusion to be met, further testsing and evaluation of the model would need to have been conducted.

V. Conclusion

In our paper we tackled the sheepherding environment where the agent controls the shepherd whose goal is to herd the sheep and lead them towards a desired location. We adopted the paradigm of deep reinforcement learning with an extension of the actor-critic approach, however our experiments proved this to be a more difficult problem, most likely due to the complicated nature of the reward policy as well as the resistance of the environment to produce good outcome though random action execution, which such method rely heavily on. One of the known problems in the deep reinforcement learning field is training in environments where the variance of the reward function is high, resulting in slow down of convergence or even divergence. In the shepherding environment such behaviour was observed since the reward function is dependant on the proximity of the sheep to the goal. Moving the sheep towards the goal from a given state can be achieved with many actions, which yield a positive reward, but nonetheless only one is optimal. Training on such data, will push the model towards predicting these actions, even though only one is optimal. This makes the learning process more difficult and slower. To try and overcome these problems we employed curriculum learning to no avail. For future work we would recommend to explore the path of reinforcement learning with human feedback, which would tackle the problem explicit definition of the reward function, since in this paradigm a neural network is trained for its approximation through human annotations [16]. Additionally to overcome the complexity of the environment a more complex model could be used ie. ViT [17].

References

[1] R. Hughes, "A continuum theory for the flow of pedestrians," *Transportation Research Part B: Methodological*, vol. 36, pp. 507–535, 07 2002.

[2] A. Turgut, H. Çelikkanat, F. Gökçe, and E. Sahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2, pp. 97–120, 12 2008.

[3] K. L. Genter, "Fly with me : algorithms and methods for influencing a flock," Ph.D. dissertation, University of Texas at Austin, 2017.

[4] M. Fingas, "The basics of oil spill cleanup," 01 2000.

[5] D. Strömbom, R. Mann, A. Wilson, S. Hailes, A. Morton, D. Sumpter, and A. King, "Solving the shepherding problem: Heuristics for herding autonomous, interacting agents," *Journal of The Royal Society Interface*, vol. 11, 11 2014.

[6] R. A. Licitra, Z. D. Hutcheson, E. A. Doucette, and W. E. Dixon, "Single agent herding of n-agents: A switched systems approach," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14 374–14 379, 2017, 20th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896317326587

[7] R. A. Licitra, Z. I. Bell, and W. E. Dixon, "Single-agent indirect herding of multiple targets with uncertain dynamics," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 847–860, 2019.

[8] C. K. Go, B. Lao, J. Yoshimoto, and K. Ikeda, "A reinforcement learning approach to the shepherding task using sarsa," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 3833–3836.

[9] H. T. Nguyen, T. D. Nguyen, M. Garratt, K. Kasmarik, S. Anavatti, M. Barlow, and H. A. Abbass, "A deep hierarchical reinforcement learner for aerial shepherding of ground swarms," in *Neural Information Processing*, T. Gedeon, K. W. Wong, and M. Lee, Eds. Cham: Springer International Publishing, 2019, pp. 658–669.

[10] D. Zhao, L. Zhang, B. Zhang, L. Zheng, Y. Bao, and W. Yan, "Deep hierarchical reinforcement learning based recommendations via multi-goals abstraction," 2019. [Online]. Available: https://arxiv.org/abs/1903.09374

[11] J. Hu, A. E. Turgut, T. Krajník, B. Lennox, and F. Arvin, "Occlusion-based coordination protocol design for autonomous robotic shepherding tasks," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 1, pp. 126–135, 2022.

[12] K. J. Yaxley, K. F. Joiner, and H. Abbass, "Drone approach parameters leading to lower stress sheep flocking and movement: sky shepherding," *Nature*, vol. 11, no. 1, pp. 2045–2322, 2021.

[13] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2019.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[15] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe, "Curriculum learning: A survey," 2022.

[16] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," 2017. [Online]. Available: https://arxiv.org/abs/1706.03741

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.