Detecting “Wolves in Sheepskin” in an A-life environment through Neuroevolution of Autoencoders

Everton Schumacker Soares and **Abhishek Nan** and **Nitakshi Sood**
Department of Computing Science
University of Alberta
Edmonton, Alberta, T6G 2E8, Canada
{schumack|anan1|nitakshi}@ualberta.ca

Abstract

A-life simulations are an interesting tool to study life in a digital petri dish. In most simulations, the intended outcome/behaviour can take a long time to emerge, and in many cases accompanying unexpected behaviours can be observed too. But their emergence is down to human interpretation of the past and current data. Graphical representations can help, but in some cases, such phenomenon might be difficult to interpret even from graphical perspectives. In this work, we propose an approach to automatically detect emergence of such phenomenon. Such behaviour can be considered as anomalous when compared to “normal” behaviour exhibited by A-life agents in the history of the simulation. As such, we propose to use a neural network with autoencoders to detect these “anomalies”. We evaluate the approach on a predator-prey model, where the prey learn to recognize predators from their behaviour, irrespective of their appearance.

1 Introduction

Artificial Life (A-life) simulations have several applications such as the study of social phenomena, animal behaviour analysis and optimization. They can lead to the emergence of different types of agent behaviours, which can be interesting or not. The automatic detection and interpretation of such behaviours still pose a challenging problem for designers of these artificial systems. Even within the simulations, this problem have some relevance, since agents need to know how to interpret the other artificial organisms and the environment in order to act and survive. A common design choice for this type of environments is to give semantic labels for elements in the environment; therefore, agents gain certain level of omniscience by accessing such information. A more biased free design is to let such agents act in the environment by given then as little information as possible, such as raw images of the environment. In this case, they have to overcome the challenges of first recognizing the elements of the environment and make inferences over it to choose the proper actions to be taken.

Our work proposes the study of behaviour recognition within an A-life system; more specifically, the inter-agent detection of anomalous behaviours. In this context, we create an predator/prey environment in which the predator looks exactly like the prey, so the second have to learn how to identify the first type of agent by interpreting its behaviour.

2 Problem Formulation

The problem explored in this work is inter-agent detection of anomalous behaviours in a population of evolving A-life agents in a predator/prey environment. This can be understood as the, here called, “wolf in sheepskin” problem: in which the predator looks like one of the preys, so the prey cannot recognize the danger by only relying on appearance. Therefore, in order to survive, it is needed to learn how to detect anomalous behaviours present in the environment: here we define anomalous as behaviours that deviate from the ones present within the agent’s species. Hence, the prey needs to learn which kind of behaviours differs (i.e, anomalous) from members of its own species, and, therefore, may present a treat.

2.1 A-life environment

For this work, we propose a predator/prey Artificial Life environment build in Matlab 2018a(cite) composed of wolves, rabbits, and grass. In this grid world, only rabbits are allowed to evolve; while wolves have their behaviour manufactured. Grass grows throughout the environment, serving as food for rabbits, which eat a portion of the grass automatically every time they reach a cell containing it. In order to make sure that all evolution runs reach an eventual end without setting a maximum number of time ticks, the world suffers a cool down: the amount of grass growth is multiplied by a scaler $0 < \beta < 1$ on every tick, so the rabbits population starve as the resources in the world become scarce. Walls may also be present in the environment, in such a way that agents cannot pass through it, nor grass can grow in it.

All the evolving preys have a convolutional brain and a sight radius r. This means that the rabbits can only see a limited amount of cells surrounding it in a square of sides $2r$ centered on itself. Each rabbit has a convolutional network with topology, weights and bias encoded into their genes, which works as the prey's brain. This network receives the RGB raw image delimited by the sight radius, outputting a cell with the best utility, towards which the agent should move.

The predators behave in a predefined way, so they are not affect by evolution. Similar to the rabbits, the wolves automatically eat every time they arrive in a cell containing rabbits. The predators also have a sight radius, moving always toward the cells with best utility (i.e, the one with more rabbits, and fewer wolves). The difference with this type of agent, is that the utility function computed for each cell is not genetically inherent, it is an well defined function of the form:

$$U_{\text{cell}} = w_r \times \#_{\text{rabbits}} + w_w \times \#_{\text{wolves}} + w_g \times \#_{\text{grass}} \quad (2.1)$$

Where the utility U_{cell} of a cell is computed as a linear combination of the number of rabbits ($\#_{\text{rabbits}}$), the number of wolves ($\#_{\text{wolves}}$), and the amount of grass ($\#_{\text{grass}}$) within the it. w_r, w_w, w_g are weights related to the rabbits, wolves and grass present in the cell, respectively. In the proposed problem, the wolves have the same appearance of a rabbit, both represented by small squares of same colour; hence, the wolves is wearing the sheepskin (or in this case, the rabbit skin). This way, in order to survive in this environment, the prey need to learn to interpret behaviours, and not only rely in the appearance of the elements.

The agents also have energy levels in their body, which are partially restored every time they eat, according to a scaler defining the food energy gain. They spend energy with cognition, reproduction, motion, or even just for existing (which is removed on every tick). Once the agent reaches certain age and minimum reproduction energy level, it can reproduce assexualy or sexually, giving half of its energy to the offspring. The agent have a cognitive cost, which is proportional to the number of cells being observed in their sight radius and to the size of the brain. Therefore, bigger brains have more capacity to emulate complex utility functions, but they are more energetically expensive.

This genetic evolution does not have any explicitly function, so the fitness of an agent is given by its ability to survive in the environment without starving or getting eaten. The wolves are injected into the simulation accordingly to a certain frequency to be determined, and being restricted to a maximum population size. There is also a burn out period (first T_{burn} ticks), in which no predator is present in the system, so the rabbits can freely evolve to move within the environment, as much as evolving to interact with other rabbits.

3 Related Work

Review related research and discuss its shortfalls with respect to the problem in the previous section.

4 Proposed Approach

We proposed the use of autoencoders as an anomaly detection system genetically evolved within the A-life agents, in such a way that the prey can detect and avoid predators by hiding every time an anomalous behaviour is observed.

4.1 Anomaly detection through Autoencoders

Autoencoders (AE) is a special type of unsupervised Deep Neural Networks responsible for reconstructing the input X. This network is composed of an encoder and decoder. The encoder is a function that maps the input X to a lower-dimensional feature vector $\phi(x)$ (i.e, the code); while the decoder is a function $X' = \rho(\phi(X))$ that maps the code into an reconstructed version of the input, X'. Therefore, the model can be trained by minimizing the reconstruction loss $L(X,X')$ given by some distance measure between input and reconstruction, the most common being the element-wise mean squared error.

In this approach, there is the assumption that, in order to learn how to properly reconstruct the input, the AE will have to learn how to represent recurrent patterns in the data. This way, after training the network in a dataset composed of normal well known data, it will be able to represent the underlying distribution of normal patterns. After trained, however, the network should not be able to reconstruct anomalous input completely, since it should include patterns never seen and, therefore, unknown within the representation learned by the AE. The anomaly detection can then be performed by finding a proper value for the threshold θ such that:

$$\begin{cases} L(X,X') <= \theta, \text{ if } X \text{ is normal} \\ L(X,X') > \theta, \text{ if } X \text{ is anomalous} \end{cases} \quad (4.1)$$

The search for a proper value of θ is a problem for itself, having different approaches to solve it. One can use the mean and the standard deviation of the reconstruction error of the train data as a threshold as done by (cite). Another possible way is to use a validation set composed of synthetic anomalies derived from the train set (e.g, shuffling pixels of an input image) varying θ, choosing the one that leads to best validation accuracy, which here is given by $\frac{TP+TN}{N}$. Where TP is the number of true positives (correctly detected anomalies), TN is the number of true negatives (normal samples correctly classified as non-anomalous), and N is the size of the validation set.

4.2 Anomaly detection within A-life

Similarly to the method mentioned above, autoencoders can be used within an A-life context to detect anomalous behaviours. In this case, each agent can have its own anomaly detection system, so it can recognize deviant behaviours in another agents. In our rabbit/wolves environment, preys can have an autoencoder attached to their brains so it can recognize the predators by their anomalous behaviour.

Each rabbit receives RGB input representing a subsection of the grid world, which is used by its convolutional brain to compute utilities for the cells. This brain is given by a convolutional network with its topology and weights encoded into the rabbits genome. This input is delimited by the agent's sight radius. Similarly, the autoencoder's representation can be encoded to the agent's genome so it can get the same RGB input image I, which should be used as input to the agent's own AE to generate the reconstructed image I'. The detection van then be done by comparing the reconstruction error with its own threshold θ_{ind}.

This approach prevents the environment's designer from having to manually create the AE architecture or even training it, since the optimal model should emerge via genetic evolution. Each agent has its own topology, weights, and bias encoded into its genome, so the one with the best model (i.e, capable of recognizing wolves as anomalous) should survive and pass their model to its offspring. However, two problems still need to be addressed: the choice of θ and the use of temporal information. For this approach, we represent the threshold θ in the agent's genome; this way, each prey has a gene responsible for given a threshold θ_{ind} for anomalous behaviour. Another problem is that the recognition of behaviours requires temporal information, so it is hard to recognize anomalies by simply using a single evolution frame. Therefore, the agent can have a memory component that remembers the last M frames (which is also genetically encoded). This way the agent's autoencoder have to recover the stacked $M+1$ evolution frames including the current one. In order to prevent M from growing without bounds, a proportional energy cost can be added to it, such that, on every evolution tick, αM is deducted from the agent's energy level just for remembering events.

Finally, in order to provided evolutional pressure for the neuroevoltion of autoencoders, the anomaly signal given by the agent's anomaly detection system have to be used to prevent predators. Hence, this system will be hard-wired to the prey's behaviour: every time that a detection occurs, the rabbit should hide. In addition, to prevent rabbits from always remaining hidden, we add a penalty to the system: once hidden, the rabbit remains in this state for a short period of time T, during which it is unable to move and eat. This way, the AEs that can correctly distinguish friends from foes will survive, eating whenever they are safe and hiding once danger is spotted, passing their genes to their offspring. The mapping from genome to autoencoder can be done in different ways as described bellow.

4.3 Fixed topology

In this approach, a predefined AE topology is shared among all agents, such that only the weights and bias are encoded into the genome. This can be done by concatenating the weight matrix W, the bias matrix B, the threshold θ, and the number of remembered frames M into the agent's genome.

Two types of autoencoder can be used for comparison reasons: one composed only of fully connected layers and another composed of a convolutional autoencoder (cite). In this approach, since the topology is fixed, the evolution has has only to tune the weights, biases, threshold and remembered frames. This is equivalent to training the model, but not the same as designing it.

4.4 Neuroevolution

In this approach, the topology of the autoencoder is allowed to change, so each agent has its own model. First, AE with only fully connected layers can be evolved with the use of the NEAT (cite) framework. This method remembers each gene origin though historical markings, applying crossover only between allele genes. In addition, the framework also protects innovative changes in the model's topology by dividing the population into species. These factors allow the topology to evolve without having its performance compromised by mutations and cross-overs. Therefore, the anomaly detection system can be encoded by adding to the agent's genome the graph representation of the autoencoders as defined by NEAT.

Furthermore, in order to evolve convolutional autoencoders the ConvDeepNEAT framework can be used. One advantage of using this method in the A-life context is that no explicit fitness function is required; hence, the network do not need to be trained on each evolution tick to compute the fitness. In our simulation, the agent with the best convolutional autoencoders have an evolutionary advantage over the others, having increased chances of survival, passing the topology to their offspring.

5 Theoretical Analysis

The evolution of autoencoders towards a model that correctly detect the wolves' behaviour as anomalies makes some assumptions about the role of environmental pressure in the network's optimization. First, we are assuming that the survival of the prey depends on the anomaly detection. This assumption is fair under the imposed conditions that wolves and rabbits have the exactly same shape and collour; therefore, the only way of distinguishing both is by analyzing its behaviour (i.e, detecting anomalies). If an agent is unable to make such distinction, it will fail to recognize real treats, being more vulnerable. This way, the pressure for survival also influences the network to be efficient in detecting anomalies.

Secondly, the autoencoders can simply minimize the reconstruction error by learning a trivial identity function: for example, for each hidden layer h of the autoencoder, the outuput activation is equal to the input, $h(x) = x$. This is usually prevented by first making sure that the code has a lower dimensionality when compared to the input's size; therefore, some compression is required to represent the data. Furthermore, some regularization can be applied to the model to prevent this problem, such as adding noise to the input image and requiring the network to recover the denoised version (cite denoise AE); or the addition of sparsity to the autoencoder, meaning that most neurons activation should be closer to zero, so a compressed representation is imposed to the model. This is usually accomplished through the addition of a regularization term in the form: $\alpha \sum_{w_i \in W} |w_i|^n$, for some real n. However, since our simulation does not provide an explicit fitness function, no regularization can be applied to ensure a non-trivial (i.e, compressed) representation of the data distribution. This property can still be achieved via evolution, since a model that only copy the input image to the output would have the same amount of error for normal and anomalous input, thus failing to recognize anomalies and treats, being vulnerable to predators.

6 Empirical Evaluation

Our empirical evaluational aims the investigation of two hypotheses. First, the autoencoders should evolve to detect anomalous behaviours, helping rabbits to distinguish prey and predator based solely on behavioural patterns, leading to a long extinction time. Second, the longest living rabbit should have the best anomaly detection system.

In order to accomplish these goals, four measures can be used to evaluate the experiments: extinction time, detection accuracy, area under the ROC curve (cite) and average detection ratio over time.

Average Extinction time: is the average total number of evolution ticks that a simulation takes to finish due to the extinction of the entire rabbit's population. To ensure that the simulation eventually ends — extinction time can tend to infinity if the rabbits learn to overcome wolves —, the grass growth is gradually decreased over time until a point where all preys starve to death.

Detection accuracy: the autoencoder of the longest living rabbit should be investigated to validate (or not) the second hypotheses. This way a test dataset can be build, containing unseen samples of normal behaviours (rabbits) and anomalies (wolves). The accuracy can then be compute through the rate:

$$\frac{TP + TN}{N}$$

Where again TN is the number of true negatives (normal data with reconstruction error bellow the rabbits own θ_{ind}), TP number of true positives (anomalies with reconstruction error above θ_{ind}), and N is the test set size.

Area under the ROC curve: Another measure is the area under the ROC curve built by measuring the detection accuracy of the longest living rabbit when varying the value of θ.

Average detection ratio over time: Although the detection accuracy gives a good measure for the performance of a single autoencoder, and it can be used to evaluate any anomaly detection system of a saved agent; another measure can help to illustrate how the autoencoders are affecting the population as a whole: the population average detection ratio. For each agent, we can compute the number of times an anomaly signal is raised when a wolf is nearby as the individual true positive ratio (TPR_{ind}). By recording the average true positive ratio of the population (TPR_{avrg}) during each tick, we can plot how the ratio evolve over time. The same can be done for the individual and average false negative ratio — (FNR_{ind}) and (FNR_{avrg}), respectively — by compute the times that no signal was raised, but wolves were present.

For each approach — fixed topology using fully connected layers only AE, fixed topology using convolutional autoencoders, NEAT autoencoders, ConvDeepNEAT autoencoders — several evolutions runs should be performed, and the average extinction time will be reported. Furthermore, for each evolution run, the detection accuracy and area under the ROC curve will be computed for each one of the longest living rabbits, and the average accuracy and area will be reported over the run. For comparison reasons, a fifth approach will be used as base line in which no autoencoders are present.

Since the evolution itself has its own parameters such as mutation rate, cross-over rate, agents speed, reproduction age, grass growth, and other elements pertaining to the physics of the world have to be fine tunned, a meta-evolution will be performed in order to maximize extinction time of the rabbits. This can be seem as repeating the above experiments for different combinations of hyperparamenters using classical genetic evolution of worlds, in each the fitness function for each physics is the extinction time of the rabbits population. The results reported will be the ones corresponding to the optimal set of hyperparameters.

7 Discussion

It is expected that the baseline containing no autoencoders will have the worst performance, since the predators and prey look alike and no behaviour detection is used. In this case, the rabbits should consider all agents as foe, or all of them as friends, or just acting completely random. Therefore, the chances of survival in this simulation can be only increased due to the dilution effect (cite). This base line can thus give a good measure of the dilution effect impact in the other simulations.

The approaches with fixed topology should perform worst than NEAT and ConvDeepNEAT, but better than the baseline, since the capacity of the model is predefined and, therefore, limited. It is also expected that the convoluational autoencoders will perform better than the fully connected ones, since convolution helps to recognize visual patterns important to the recognition of behaviours. The longest average extinction time should be given by the ConvDeepNEAT autoencoders, followed by the NEAT autoencoders.

It is also expected that the longest living rabbits will have an anomally detection system with higher accuracy when compared to majority class and other rabbits. Otherwise, the detection would not provide strong advantages when compared to other environmental pressures or the dilution effect.

Finally, it is expected that TPR_{avrg} increases and FNR_{ind} decreases over time. The population should adapt to learn to recognize anomalous behaviours, since the distinction of prey and predator is crucial for survival. Therefore, an increasing number of agents with this ability should emerge in the simulation, thus the ratio of positive detections should increase and the negative detection should decrease.

8 Future Work

After the work here proposed is completed, it can be expanded to explore other properties of the system. Our approach proposes the evolution of prey agents (rabbits) only, so we can have a better control of the evaluation and the neuroevolution. For future work, the simulation can be extended so the predators co-evolve with the prey. This way, no predator injection is required, nor is needed to manufacture behaviours for wolves, which removes human bias from the system.

In this work, our target is the predator's behaviour. However, this approach can be extended to detect any kind of anomalous behaviours, even within the rabbits population. This environment can be used as an approach for generating of automatic detection systems capable of identifying truly surprising/novel behaviours in an A-life population.

This work can also be extended for the study of plan recognition. The problem explored in this work is only the recognition of predator's anomalous behaviour, and the reaction to such detection is hardwired in the rabbits brain. This can be expanded to a work in which the agents have to recognize the other behaviours and identify the other agents' plan in order to decide its own plan. This could be done, for example, by using the anomaly score as part of the state in an reinforcement learning algorithm.

9 Conclusions

Acknowledgments

Colter MacDonald and Robert MacGillivray helped to develop and maintain the A-life simulation implemented in Matlab used as base for this project.

References

Bulitko, V., and Brown, M. 2012. Flow maximization as a guide to optimizing performance: A computational model. *Advances in Cognitive Systems* 2:239–256.

Bulitko, V.; Sturtevant, N.; and Kazakevich, M. 2005. Speeding up learning in real-time search via automatic state abstraction. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1349 – 1354.