A Deep Learning / Neuroevolution Hybrid for Visual Control

Andreas Precht Poulsen, Mark Thorhauge, Mikkel Hvilshj Funch, Sebastian Risi
Center for Computer Games Research
IT University of Copenhagen, Denmark
{appo, marth, mhvf, sebr}@itu.dk

ABSTRACT

This paper presents a deep learning / neuroevolution hybrid approach called *DLNE*, which allows FPS bots to learn to aim & shoot based only on high-dimensional raw pixel input. The deep learning component is responsible for visual recognition and translating raw pixels to compact feature representations, while the evolving network takes those features as inputs to infer actions. The results suggest that combining deep learning and neuroevolution in a hybrid approach is a promising research direction that could make complex visual domains directly accessible to networks trained through evolution.

KEYWORDS

Neuroevolution, Deep Learning, Visual Control, NEAT

ACM Reference format:
Andreas Precht Poulsen, Mark Thorhauge, Mikkel Hvilshj Funch, Sebastian Risi. 2017. A Deep Learning / Neuroevolution Hybrid for Visual Control. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3076016

1 INTRODUCTION

The main idea of the DLNE approach is to separate the visual recognition and action inferring component for visual control (Figure 1). A deep convolutional network (DCNN) is trained in a *supervised* fashion through gradient descent to determine the position of an enemy bot based on high-dimensional raw pixel input. This positional information is then used as input to another network that is evolved to aim and shoot at a given target. Importantly, the evolving network is trained in a *non-supervised* way, i.e. it only relies on a fitness function and not on a large number of labeled examples. The hybrid combination of these two techniques is an unexplored area in visual control. In fact, this combination could combine the advantages of both methods. Neuroevolution has often difficulties scaling to problems with a large number of inputs [5], such as 3D shooting games [4], which could be solved with a deep learning-based visual recognition component. On the other hand, evolutionary-based approaches do not rely on differentiable architectures, and work well in domains with sparse rewards, a challenge for most deep reinforcement learning approaches [1, 3].

Approaches based on neuroevolution alone [2] have the advantage of being more general (i.e. no identification of useful features for supervised training is required), however they take considerably longer to train. The approach presented here could offer a good compromise between training time and generality.

GECCO '17 Companion, Berlin, Germany
 978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3076016

2 EXPERIMENTS

The goal of the agent in this paper is to aim and shoot a stationary enemy. The position of the agent is fixed but it can turn vertically, shoot and reload. The game's arena is quadratic, with the agent spawning on one side and the targets spawning on the other. Both the agent and the target spawn in a random x and y, while the z coordinate is fixed.

2.1 Feature Representations

We compare two different visual representations: the angular representation and the visual partitioning representation. This representations determine what the DCNN outputs and therefore in turn what the evolving network receives as input. **Angular representation (AR):** The angular feature representation defines the position of the target on the screen with two angles (horizontal, vertical), a distance and a binary output indicating whether the target is within sight. **Visual Partitioning Representation (VPR):** The visual partitioning representation defines the position of the target as a classification task, where each point on the screen belongs to a class bounded by a square (Figure 2). The partitioning is finer in the center of the agent's view field, allowing for more precise aim adjustments the closer the target is to the line-of-fire.

2.2 Neuroevolutionary Training

The training of the agent was performed in the Unity 5 game engine with the UnityNEAT framework[1], which is a port of the C# NEAT framework SharpNEAT. Instead of evolving the network with the DCNN as input, which is computationally expensive, here the evolving network is trained with the ground truth information from the game engine itself (e.g. exact position of the enemy). The agent is awarded for hitting the target and aiming close to the target.

2.3 DCNN Architecture and Training

A regression network outputs the AR, scaled to $[-1, 1]$, while a classification network outputs 26 probabilities for the VPR setup. The networks take as input an RGB image with a shape of 256×256×3. We compare a deep and a more shallow network architecture. Of the deep network's 12 layers, the first 8 alternate between convolutional and pooling layers, while the next 3 are fully connected layers followed by an output layer. The shallow convolutional network has a similar architecture but only 6 layers, the first 4 being alternating conv and pool layers, followed by a fully connected layer and an output layer.

[1] https://github.com/lordjesus/UnityNEAT

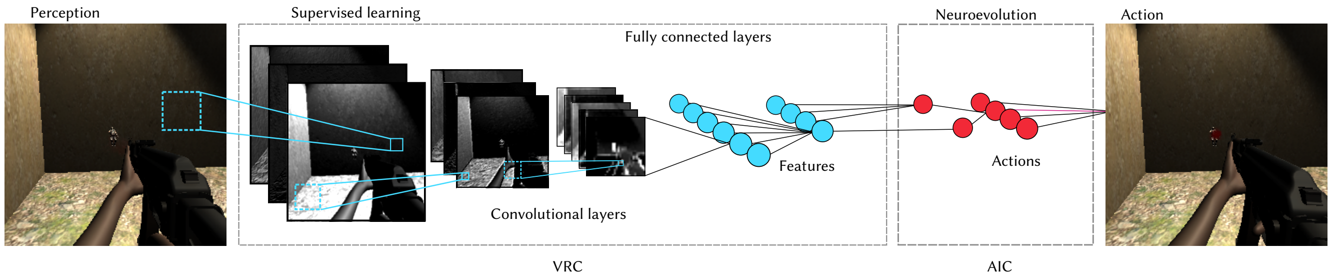

Figure 1: DLNE Approach. The combination of supervised learning and neuroevolution translates the visual state to actions.

Visual Settings. We tested the networks without (V1) and with player and weapon overlay (V2), which can partially or fully cover the target.

3 RESULTS

Deep Network Accuracy. We first tested how good the DCNN is in estimated the correct position of the target. The accuracy of VPR is measured as the percentage of correct predictions. The topologies of the networks do not seem to have a significant impact on the accuracy, with the best shallow network reaching an accuracy of 96.05 in V1, and 86.10 in V2, and the best deep network an accuracy of 95.94 in V1 and 86.60 in V2. The performance measure of the

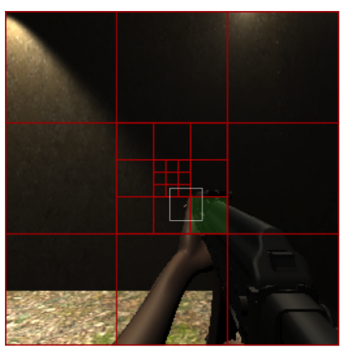

Figure 2: The VPR is more fine grained in the direction the gun is pointing. Shown is an example of the DCNN correctly predicting the class (green square) with a confidence of 55.7%, with only four pixels of the target being visible.

AR representation is the absolute error on the predicted target distances. The distance error of the deep network is 0.0893 in V1 and 0.1614 in V2. The shallow network reaches a distance error of 0.153 in V1 and 0.1658 in V2. The distance error is almost twice as high with the more complex V2 setup and increasing network depth seems to have a greater effect on the AR representation than on the VPR representation.

Evolutionary Training. A total of ten independent evolutionary runs were performed for each of the two neuroevolution setups: networks with VPR and networks with AR as input representation trained using ground truth enemy locations. The average final fitness for the VPR setup is 318,4 (sd = 58,36), while it is 436,9 (sd = 30,12) for AR. This difference is significant ($p < 0.01$; Mann Whitney U-test). Each run took approximately 10 hours to complete.

Hybrid Approach. Here we report results on taking the best network found during evolution on the ground truth enemy locations combined with the best 12-layer DCNN we found during supervised training (for both VPR and AR representations). Table 1 shows the performance of the three different treatments averaged over 200 trials. The evolved network with AR representation as input reaches a significantly higher score using the ground truth than any other method ($p < 0.05$; all tests are two-tailed Mann Whitney U-tests) but the distances and angles returned by the DCNN component are not accurate enough for the network to perform well. In the case of the VPR, performance decreases slightly (though not significantly) from ground truth to visual setting V1, but significantly when compared to the more complicated V2 setting ($p < 0.05$) that includes the weapons overlay.

Table 1: Performance based on input representation.

	Ground truth	DCCN (V1)	DCNN (V2)
VPR	310.71	281.97	193.71
AR	396.72	74.73	106.42

4 CONCLUSION

While the combination of the two techniques is not perfect (see the example video of the VPR hybrid approach at: https://youtu.be/daFvJa90f8Y), the hybrid approach performs reasonably well and is in fact able to aim&shoot based on the raw pixel representation of 256×256×3 images. With 22.87/31.31 hits averaged over the 200 trials, the best hybrid approach has a shooting accuracy of 42%.

REFERENCES

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT Press.

[2] Jan Koutník, Juergen Schmidhuber, and Faustino Gomez. 2014. Evolving Deep Unsupervised Convolutional Networks for Vision-based Reinforcement Learning. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation (GECCO '14)*. ACM, New York, NY, USA, 541–548. DOI: http://dx.doi.org/10.1145/2576768.2598358

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, and others. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[4] Matt Parker and Bobby D Bryant. 2008. Neuro-visual control in the Quake II game engine. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. IEEE, 3828–3833.

[5] S. Risi and J. Togelius. 2017. Neuroevolution in Games: State of the Art and Open Challenges. *IEEE Transactions on Computational Intelligence and AI in Games* 9, 1 (March 2017), 25–41. DOI: http://dx.doi.org/10.1109/TCIAIG.2015.2494596