Marcantonio Soda Jr

CSE431

Playing Atari with Six Neurons Critique

24 April 2023

Introduction and Motivation

Playing Atari with Six Neurons by Giuseppe Cuccu, Julian Togelius, Philippe Cudre-Mauroux is a paper on a novel infrastructure for learning policies and compact state representations separately but simultaneously for policy approximation in reinforcement learning. Their specific use case was implementing a system to teach itself how to play different Atari games, but the system is meant to be extendable to a great breadth of reinforcement learning applications. The goal of the project is not to design a system that outperforms existing implementations in terms of accuracy, but to minimize the network size while maintaining state of the art accuracy. They were successful, producing trained neural networks that are a full two orders of magnitude (only 6-18 neurons) smaller than any existing implementation while producing comparable scores.

The researchers claim that the neural networks produced by existing reinforcement learning techniques are unnecessarily large and overcomplicated because only a small portion of the network represents the actual policy approximation, which is the main function of the network. In theory, decoupling the image processing and the policy approximation into two separate components, rather than both being done in the same network, should make the end product significantly smaller, less complex, and more performant. The system that the researchers crafted was able to do just that. It has three modules that serve to learn policies and compact state representations separately but simultaneously: the compressor, the controller, and the optimizer, all of which are detailed in the next section.

Compressor

The compressor represents the image processing stage. The goal of the compressor is to “compress” the image that would normally be fed directly to a huge neural network. Instead of inputting each and every pixel into the network for each frame, each frame is fed to the compressor which processes the image and extracts relevant features from it. The extracted features end up being much,

much smaller than the 210x180x3 matrix that it takes to represent one frame of an Atari game. The numerical representation of these features are then fed to the controller which does the policy approximations, outputting the controller input which should be fed to the Atari.

The compressor uses a combination of two mechanisms to extract features from the input image: Increasing Dictionary Vector Quantization (IDVQ) and Direct Residuals Sparse Coding (DRSC). IDVQ automatically increases the size of its dictionary over successive training iterations, building new centroids (features) from the positive part of the reconstruction error. Growth in dictionary size is regulated by a predefined threshold which indicates the minimal aggregated residual considered to be a meaningful addition to the set. The centroids that are added to the dictionary are not refined any further, but each centroid is purposely constructed to represent one particular feature, which was found in an observation and was not available in the dictionary before.

DRSC, on the other hand, produces binary encodings by selecting centroids to add to the encoding, based on how much of the residual information can they encode. The algorithm computes the differences between the residual information and each centroid in the dictionary, aggregating each of these differences as sums. The centroid with the smallest difference is the one that's most similar to the residual information and is therefore chosen to be included in the encoding. The corresponding bit in the binary code is flipped to '1', and the residual information is updated by subtracting the new centroid. The algorithm keeps looping and adding centroids until the residual information is lower than a predefined threshold, which corresponds to an arbitrary precision in capturing the information in the original image.

Together, these mechanisms enable the compressor to provide a compact representation of the input image that allows the neural network to entirely focus on decision-making. IDVQ extracts a low-dimensional code from the observation in an unsupervised learning fashion and DRSC produces binary encodings of these features, enabling the compressor to efficiently encode the information from the original image. This compressed representation enables the controller to output the most relevant actions to be fed back to the Atari, improving the overall performance of the system.

Controller

The controller is likely the most familiar step in the pipeline as it is more closely related to a traditional neural network that one would develop when using traditional reinforcement learning

techniques. The difference between the controller and the aforementioned traditional neural network is that it is much smaller, less complex, and more performant because of the nature of the compressor. Because the information fed to the controller by the compressor is representative of observed features within a frame rather than the entire frame itself, the controller is allowed to be significantly simplified. The tricky part is that the number of inputs accepted by the controller dynamically changes with the number of outputs from the compressor. As the dictionary of the compressor grows, so do the inputs to the controller. This nuance is combatted simply by setting the weights of all new connections to zero (at first). The number of neurons in the output layer is always equal to the number of possible actions that the Atari game may take. Interestingly, there are no hidden layers to any of the tested networks, the only two layers are input and output.

Optimizer

The optimizer used in the pipeline is a variation of Exponential Natural Evolution Strategy (XNES) tailored for evolving networks with dynamically varying size. XNES is a family of evolutionary strategy algorithms that maintain a distribution over the parameters space rather than an explicit population of individuals. The parameters maintained by the optimizer are updated based on the natural gradient, constructed by rescaling the vanilla gradient based on the Fischer information matrix. The researchers introduce a novel twist to the algorithm due to the fact that the dimensionality of the distribution (and therefore its parameters) varies during the run. Since the parameters are interpreted as network weights in direct encoding neuroevolution, changes in the network structure need to be reflected by the optimizer in order for future samples to include the new weights

The optimizer plays a crucial role in the pipeline as it is responsible for evolving the network that makes decisions based on the features extracted by the compressor. The compressor extracts the most relevant features from each frame of the Atari game and provides a compact representation of the input image, which is then fed to the optimizer. The optimizer evolves the network that takes in the compressed input and outputs the most relevant actions to be fed back to the Atari. The optimizer dynamically changes the dimensionality of the distribution based on the size of the compressor's dictionary, ensuring that the network is always able to handle the inputs from the compressor. This allows the system to be

trained and make decisions in a more efficient manner, improving the overall performance and simplicity of the system.

Testbed and Results

The system was run on a single machine with a 32-core Intel Xeon E5-2620 clocked at 2.10 GHz with 96 GB of memory. The maximum run length for all tests was capped at 200 interactions which equates to about 1000 frames. Population size was kept around 18-32. Resolution was reduced from 210x180x3 (length, width, RGB) to 70x80. They opted to average the color channels to obtain a grayscale image rather than represent RGB directly. Every individual is evaluated 5 times to reduce fitness variance. Each experiment only allowed for 100 generations.

The system was run on a single machine with a 32-core Intel Xeon E5-2620 clocked at 2.10 GHz with 96 GB of memory. The maximum run length for all tests was capped at 200 interactions which equates to about 1000 frames. Population size was kept around 18-32. Resolution was reduced from 210x180x3 (length, width, RGB) to 70x80. They opted to average the color channels to obtain a grayscale image rather than represent RGB directly. Every individual is evaluated 5 times to reduce fitness variance. Each experiment only allowed for 100 generations.

The researchers only opted to include two figures: one representing game scores and one representing the properties of the network of their approach compared to other industry standard approaches. Keep in mind that this research was not intended to outperform competitors in terms of score, but rather to attain comparable scores with significantly smaller neural nets. They appear to have succeeded. The scores benchmarked across 10 games appear to show that the scores are consistent between the different approaches. Some results are better, some of them are worse. However, the implementation certainly blows the competition out of the water in terms of the simplicity of the trained neural network. They produced networks with 18 neurons, 0 hidden layers, and only 3 connections whereas the competitors produced networks that attained similar scores with 650-3034 neurons, 2-3 hidden layers, and 346,000-906,000 connections.

Critique

The research conducted by Giuseppe Cuccu, Julian Togelius, Philippe Cudre-Mauroux is not only novel and extremely interesting but also demonstrates a high level of ingenuity and innovation in the

reinforcement learning field. The system the authors were able to develop is a testament to their deep understanding of the challenges faced in traditional approaches, and their ability to devise creative solutions to address these issues. One of the many notable aspects of the paper is the elegance and simplicity of the system they proposed. The researchers were able to nicely break down the complex problem of policy approximation into distinct components: the compressor, the controller, and the optimizer, each with its unique function and contribution to the overall efficiency of the system. This modular approach allows for better understanding, analysis, and potential improvements within each component, making the research more accessible and useful to the community. The authors also provided good explanations for the methods employed in each component of the system. Their detailed descriptions of the mechanisms behind the three components demonstrate a high level of expertise and contribute to the overall transparency of the research. This transparency is essential to encourage other researchers to build upon the methods introduced in the research paper. Furthermore, the statistical results presented in the paper serve as great evidence of the effectiveness of the overall system. They were clearly able to achieve similar performance to state-of-the-art methods, using significantly smaller networks, validating their goals and showcasing the practical implications of their research effectively. These results may very well inspire new researchers, subsequently paving the way for more efficient reinforcement learning methodologies in the future, which is the goal behind any solid research.

However, there are some areas that the paper is lacking. Some of their design choices are underexplained and therefore not very compelling. For example, in section 3.3 when the paper explains the connections to the neurons, it claims that each neuron has three connections: the inputs to the network, the output of all neurons from the previous activation, and a constant bias which is alway set to 1. It is unclear why this bias is always set to 1. Intuitively, this seems to be a huge waste of space. They do not explain the function of a constant bias, why they chose to make it 1 and not, say, 52, or the implications that this has on the system as a whole. This is a small gripe to have, but a gripe nonetheless. Furthermore, they opted to set the population sizes between 18 and 42, which is extremely small compared to most neuroevolution applications, and the only explanation they give is that it decreases the runtime of a given test. They set the maximum run length at 200 interactions per test. This is incredibly small. It stands to reason that they would have been able to attain significantly higher scores if they let

their networks learn longer. Again, the only reason that they gave for this was optimizing runtime. It seems that runtime was a major consideration for this particular research. Many of their design decisions were made to increase performance, but, why then did they not benchmark it? Why wouldn't they stress the system and compare different values for population size and the number of interactions, or, at the very least show how they were able to optimize these parameters? The same is true for the learning rate of 0.5 that they set. They don't explain why, they don't explain how they optimized this value, or what its contribution to network size or score was. The paper only included two plots: score and network size. It certainly lacks sufficient benchmarking. Future researchers looking to adopt the proposed methodologies will be less likely to do so due to incomplete information.

Conclusion

The paper "Playing Atari with Six Neurons" byGiuseppe Cuccu, Julian Togelius, Philippe Cudre-Mauroux presents a seemingly groundbreaking approach to reinforcement learning by introducing a system that decouples image processing and policy approximation, enabling the design of neural networks that are orders of magnitude smaller than existing implementations while maintaining state-of-the-art accuracy. The proposed system, composed of a compressor, controller, and optimizer, effectively processes input data and makes decisions in a highly efficient and performant manner. The authors definitely did achieve their primary goal of developing significantly smaller neural networks with comparable performance. However, the paper should have offered some further clarification on a few design choices and provided additional benchmarks to illustrate a more complete picture for future innovators. Nevertheless, this innovative work has the potential to inspire new state-of-the-art implementations in reinforcement learning in the future.