THE UNIVERSITY OF MELBOURNE

MAST 90109

Local Intrinsic Dimensionality as an Agument to the Evolutionary Strategies for the Deep Reinforcement Learning Tasks

Youshao Xiao

876548

Master of Data Science

Supervised by

Dr. SARAH MONAZAM ERFANI

School of Computer and Information Systems

Acknowledgments

Firstly, I wish to express my sincere gratitude to my supervisor Dr. Sarah Monazam Erfani in School of Computing and Information Systems at the University of Melbourne.

Also, I would like to thank Dr. Xingjun Ma and Professor Christopher Leckie for providing me the valuable advice in the thesis.

Finally, I would like to thank my parents for the support and encouragement throughout my years of study.

Thanks,

Youshao Xiao

Declaration of Authorship

I certify that this report does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text. The report is 7153 words in length (excluding text in images, tables, bibliographies and appendices).

15 November, 2019

Contents

List of Figures

List of Tables

Abstract

Parallel Evolutionary Strategies have been a scalable alternative to Reinforcement Learning like Q learning or Policy Gradients in the deep reinforcement learning tasks. Instead of using gradient descent, Evolutionary Strategies is applied to optimize the loss function of the neural network and achieve competitive results compared with DQN and other state-of-arts gradient-based algorithm with great stability. However, the deceptive trap (i.e. contain local optima) or the sparse reward is the challenge for the reinforcement learning tasks since the successive states of agents are highly dependent on the current states and actions. To mitigate this problem, the novelty metric like K Nearest Neighbour(KNN) could be combined with reward together to provide the direction for the exploration from the outcome space and archive higher performance. However, KNN has limitations to characterize the local regions in some cases. In the thesis, an expansion-based measure of intrinsic dimensionality, Local Intrinsic Dimensionality(LID), has been introduced to solve the problem. Our experimental results indicate that the LID based quality diversity algorithm could explore novel behaviors that can not be discovered by the KNN based method. It obtains higher scores on several Atari games with a deceptive trap or local optima.

1 Introduction

Due to the rise of deep learning in the Computer Vision [1] and Speech Recognition tasks[2], the researchers have applied the deep learning technologies as a function approximation to estimate the Q values in the reinforcement learning[3][4]. Usually, the gradient descent or its derivatives are used as the optimizer to optimize the loss function to get the optimal policy. A Scalable Evolutionary Strategies framework with about 1440 cores achieve competitive performance on several tasks (humanoid Locomotion and Atari games) in terms of time and scores [5]. However, the Evolutionary Strategies directly searches in the parameter space for the optimal parameters and the diversity of parameters comes from random noise. This is not efficient to get rid of the local optima problem. Different from the supervised machine learning where the training data is usually independent of each other, the local optima is a more serious problem for reinforcement learning due to the dependency between the training data [6][7]. The premature convergence to the local optimum significantly worsens the performance of the objective-oriented policy search from different reinforcement learning tasks, including humanoid locomotion[8], playing Atari games[9], and maze games[10].

The key problem here is how to efficiently alleviate the local optima or deceptive problem in the deep reinforcement learning tasks. The researchers try to alleviate the problem by directed exploration, where the reward is restructured with the reward and some extra information or heuristic to guide the exploration[9][11]. A natural idea is to encourage the agent to visit the state has rarely visited before[7].

A Quality Diversity method, incorporating both KNN-based novelty and reward together, achieves superior performance in terms of scores compared with the Evolutionary Strategies in several deep reinforcement learning tasks[9]. It computes the novelty as a holistic description of the agent's lifetime behavior in a whole episode of game and the agent is encouraged to take different behaviors than what has previously executed. However, the KNN malfunctions in some local cases, which may lose the possibility to get high rewards [12]. This will be further discussed in the section 3.4.3. Thus, a local discriminability measurement, Local Intrinsic Dimension(LID)[13][12][14], is proposed to replace KNN as the novelty to solve

the malfunctions in some local cases and improve the capability to relieve from the deceptive trap.

We test both KNN-based Quality Diversity method and LID-based one in serval atari games with or without the deceptive traps. Our experiments confirm that the LID based method significantly outplays the KNN based method in serval Atari games with local optima where the input is high dimensional. In other words, LID based method could explore more diverse strategies which are not able to be discovered by the KNN and simultaneously produces high rewards. This, in turn, implies that LID has the superior ability to mitigate the local optima problem in the Deep Reinforcement Learning tasks compared with KNN.

Our main contributions are 1. First time to use the LID as the novelty to augment the Evolutionary Strategies in the Deep Reinforcement Learning tasks. 2. Show that LID is possible to find more complex pattern compared with the KNN and this in turn helps the agent escape from the deceptive trap or local optima. 3. The proposed method achieves more promising performance in terms of the score than the original KNN based method in the high dimensional space.

2 Literature Review

By trial and error, the agent in an autonomous system can learn a policy that guides its behaviors to maximize the utility or the total cumulative reward in a task. The utility $U(s, a)$ is defined as the return that the agent will receive from when executing action a when it is in the state s following the policy π. In general, the policy search explores the space of the policy parameters until meeting the satisfactory condition. Policy search is usually classified into the policy with or without the utility model depending on whether the utility function of the model is explicitly given or learned.

Regardless of such utility function, policy search without utility could maximize the cumulative rewards by sampling the policy parameters and moving towards the direction with high rewards. To be extreme, a totally random search will explore the parameters until it accidentally achieves a decent utility[15]. Genetic

algorithm[16], Evolutionary Strategies[17] or NEAT framework[18] are bionic algorithm based on Darwin's natural selection theory. The main idea is to search the optimal parameter by mimicking the evolution of natural creatures.

In the contrast, policy search with utility is generally more effective. The estimator of a utility model could be derived from either policy parameter space, state-action space or the arbitrary outcome space [11].

1. In the policy parameter space, if the utility model is stochastic, it could be estimated by a regression model based on the Bayesian inference, which is known as the Bayesian optimization[19][20]. It initializes the distribution of parameters with a prior distribution and updating the posterior distribution given the evidence when observing a new sample.

2. In the state-action space, the true utility $U(s, a)$ can be approximated with a model $U_\eta(s, a)$ with parameters η, where $U_\eta(s, a)$ is also called critic. The critic will evaluate how good the agent acts a in the state s. This is proved to mitigate the local optima problem to some degree since it provides a long-term insight into the expected rewards[21]. For the actor-critic model, the actor will perform the actions following the policy and the critic will evaluate how good the action it is.

3. Instead of exploring in the policy parameter space or the state-action space, the directed exploration searches in a smaller behavioral space or outcome space to learn an invertible mapping from the policy parameter space to the outcome space. This method is immune to the local optima problem since it entirely ignores the objective[7]. Directed Search could be classified into Novelty Search(NS)[7], Quality Diversity(QD)[22] and goal exploration process(GEP)[23]. The NS and QD come from the evolutionary computing society and they are different depends on whether other information or objective except the novelty has been taken into account.

In the Novelty Search community, the k nearest neighbor is the most popular measurement of novelty due to its simplicity. Intuitively, if the average distance to a given point's nearest neighbors is large then it is in a sparse area. KNN is applied in the novelty search to explore in the maze navigation and biped walking

tasks, which significantly outperforms objective-based search[7]. Also, it is applied to search for diverse morphologies of the soft-bodied robots[24].

In contrast, Local Intrinsic Dimensionality is an extension of the generalized expansion dimension which historically measures the rates of growth in the number of points encountered as the distance from the reference instance increments[13]. It provides a natural measure of the data discriminability that could be widely applied in feature selection or learning similarity. For example, the adversarial regions could be characterized by the LID via assessing the space-filling capability of the area surrounding a reference instance in the adversarial attack[12]. LID also naturally characterizes the inlierness or outlierness of data point with respect to its locality[13]. But currently, there are fewer works focus on the ability of LID in the outlier detection tasks or novelty search tasks.

By combination of the reward-based objective and novelty together, the quality diversity algorithm is an improvement on the original random exploration of the Evolutionary Strategies. Without loss of the parallelism, evolutionary strategies with the KNN based novelty achieves better scores than pure evolutionary strategies in all Atari games or locomotion tasks which contains the local optima in high dimensional space[9]. The combo of novelty and reward is also widely used in different tasks, including evolving a high quality and diverse morphologies of soft-bodied robots[24] or using intrinsic motivation as a novelty to efficiently play the Atari games[25][26]. But all of them evaluate the novelty of the states separately. The KNN-based Diversity Quality paper measures the novelty from the behavior states of the agent in a whole episode game, which generates a high dimensional space to explore[9].

3 Background

3.1 Deep Reinforcement Learning

The reinforcement learning algorithm is a subclass of the machine learning algorithm using weak labels or rewards[6]. A reinforcement learning problem is characterized by the five components: the observation s, agent, the policy π, ac-

tion a and rewards r. At each step, the agent receives the observation from the environment (partially observable or fully observable) and transformed it into the agent state. Based on the given agent state, the agent executes the actions according to the given policy π. After the execution of the action on the environment, the agent receives the immediate reward r and new observation s from the environment. Given the new observation s and reward r, the agent can update its policy π to maximize its expected return.

The recent advances in computer vision[1] and natural language processing[2] by using deep neural network promotes the researchers in reinforcement learning area to combine the deep learning with the reinforcement learning[3][4]. One of the most popular traditional reinforcement learning is the Q learning. It evaluates each state-action pair (s and a) and updates the Q table using Bellman equation[6]. However, in a more complex game where the input state is the pixels of the image like the atari games in the experiments, the dimension of the state dramatically increase and it is a impossible to maintain a large Q table. To solve the problem, a deep neural network is used as a functional approximation to estimate the Q value. Given the state s and action a, the neural network produces the Q-value to evaluate how good the state-action it is. In the current work, the neural network directly outputs the actions given the input state. The action guides the agent's behavior in the game.

However, the deep reinforcement learning tasks suffer from the local optima problem due to the training data are highly correlated with each other. During the training of reinforcement learning, the agent takes a sequence of actions following the policy π which usually motivated from the rewards. If the agent only maximizes the short-term cumulative rewards, it will be stuck in a local optimum where there is no reward gradient to follow in the following timesteps. So actively seeking out novel states and actions that might yield high rewards and result in long-term gains is a core problem to solve. Our research aims to find a better novelty metric that could alleviate the local optima problem and in turn to maximizes the cumulative reward in the game.

3.2 Evolutionary Strategies

Evolutionary Algorithm is a class of black-box optimisation algorithms inspired from Darwin's natural selection theory[27] [17]. The main idea of the evolutionary algorithm is to eliminate individuals with a low score or fitness to produce more competitive offspring. At each generation or iteration, a population of the genomes is mutated and recombined via the crossover to generate the descendants. The mutation is to introduce the diversity of the offspring. The fitness of the resultant offspring is evaluated according to the fitness function like the reward and the natural selection is applied to ensure that the individuals with higher reward tends to produce more descendants. The iterative processes stop until the fitness of the individuals are acceptable. Different variants of the evolutionary algorithm like genetic algorithm or evolutionary strategies varies in how they implement the process of the mutation, crossover, and selection.

Compared with the genetic algorithm which uses the binary coding to encode the genomes of individuals, the evolutionary strategy represents the individual with the real numbers and achieves the mutation by perturbing the individuals with the gaussian noise. In this way, the evolutionary strategy allows successful individuals to dictate the distribution of future generations[17].

To be more specific, we use the natural evolution strategies (NES) in our work. Different from the evolutionary strategies, the NES aims to maximize the average fitness of the population via gradient descent rather than selection[28]. The estimate of the gradient is given by the formula (1).

$$\nabla_\phi E_{\theta\sim\phi}[f(\theta)]) = \frac{1}{n}\sum_{i=1}^{n} f(\theta_t^i)\nabla_\phi \log p_\phi(\theta_t^i) \tag{1}$$

In our case, at each generation, a population of the parameters θ of the convolutional neural network is perturbed and the parameters θ is updated depending on the cumulative rewards. In this work, the fitness $f(\theta)$ is the cumulative stochastic return in a whole episode game provided by the environment and there are n agents in each iteration. The population of parameter θ in NES is depicted by the distribution $p_\phi(\theta)$. In each generation, θ_t is updated using the formula (1)

by an approximate estimate of the expected reward. The procedure stops in a pre-defined number of generations or iterations T.

3.3 Novelty search and Quality Diversity

Sometimes novelty is more attractive than the direct reward objective in the deceptive or sparse reward situation. By guiding the search towards novel behaviors, novelty search could help to overcome environments with deceptive or sparse rewards. Contrary to intuition, the method in some cases outplays the objective-oriented search [7] [29]. The objective-based search does not necessarily reward the intermediate stepping stones which results in the ultimate objective since the stepping stones may not resemble the objective itself. However, the novelty search could find the stepping stones by rewarding the novel behaviors while the objective-based search cannot detect the stepping stone at all in the deceptive or sparse cases.

Also, novelty search is well suited to evolutionary algorithms, like evolutionary strategies or genetic algorithms. The population of the genomes naturally expands to a wide range of different behaviors. By replacing the novelty metric with the fitness metric, we could modified the evolutionary strategies with few changes. The novelty of a new entity is calculated concerning the archive of the past individuals' behavior characterization and the current behavior characterization.

To be more specific, Behaviour Characterisation (BC) is a function of parameter θ. It maps each evaluated entity to a behavior, for example, some vector representation characterizes what the entity it is. It is usually a tensor which characterizes the chronology of the actions taken by the agent and also describes other significant aspects of the agent's behavior. Then the tensor is used to compute the novelty against the archive, which is a collection of past behavior characterizations.

Moreover, the combination of the reward and novelty together is called Quality Diversity since it encourage both the diversity of the behaviors and good performance individuals at the same time. A lot of evolutionary algorithms like MAP-Elites[30] or NS-NS[31] falls in this class. The aim here is to use the novelty to rich the behavior diversity of the agents to escape from the local optima and the reward-based

objective ensures the quality of the behavior which leads to the high rewards.

3.4 K Nearest Neighbour and Local Intrinsic Dimensionality

3.4.1 K-Nearest Neighbour

A simple measurement of the novelty of a data point is the average KNN distance or the mean KNN distance in the Euclidean space. It is the mean distance from the reference point to its k nearest neighbors.

$$Novelty(x) = \frac{1}{k}\sum_{i=0}^{k} dist(x, x_i) = \frac{1}{|S|}\sum_{j \in S} ||b(\pi_\theta) - b(\pi_j)||_2$$

$$S = KNN(b(\pi_\theta), A)$$

In this case, the mean KNN distance is calculated by computing the average distance between behavior characterization $b(\pi_\theta)$ and the k-nearest neighbors in the archive A. A larger KNN value implies a higher novelty of the $b(\pi_\theta)$ or distinctive behaviours in the outcome space.

3.4.2 Local Intrinsic Dimensionality

In the theory of intrinsic dimensionality, (global) Intrinsic Dimensionality is proposed to be an expansion-based measurement of intrinsic dimensionality of the whole data points and Local Intrinsic Dimensionality(LID) quantifies the intrinsic dimensionality within the vicinity of the reference point[13] [32]. The intrinsic dimension could be regarded as the minimum of latent variables needed to characterize the data. For example, the low dimension subsets in high dimension spaces can own pretty low expansion rates, whereas even for one dimension data the expansion rate can be linear in the size of S in terms of the space-filling capability [33]. Intuitively, the volume of an K-dimensional ball grows proportionally to r^K when its size is scaled by a factor of r. From the rate of volume growth with radius,

the dimension K can be deduced from the volume measurements[34] [13].

$$\frac{V_2}{V_1} = (\frac{r_2}{r_1})^K$$

$$K = \frac{ln(\frac{V_2}{V_1})}{ln(\frac{r_2}{r_1})}$$

If the volume is replaced with a cumulative distribution function of the distance in the local area, the expansion model presents a local view of the intrinsic dimensional structure of the data since their estimation is restricted to the vicinity of the reference point. This leads to the formal definition of LID.

Definition (Local Intrinsic Dimensionality)

Given a data sample $x \in X$, let $r > 0$ be a random variable denoting the distance from x to other data samples. If the cumulative distribution function $F(r)$ is positive and continuously differentiable at distance $r > 0$, LID of x at distance, r is given by the following formula whenever the limit exists[13].

$$LID_F(r) = \lim_{\epsilon \to 0} \frac{ln(F((1+\epsilon)r)/F(r))}{ln(1+\epsilon)} = \frac{rF'(r)}{F(r)}$$

$F(r)$ corresponds to the volume V and probability density function $f(r)$ of the distance is the first derivative of CDF function. LID at x is in turn defined as the limit of the radius $r \to 0$:

$$LID_F = \lim_{r \to 0} LID_F(r)$$

LID is interpreted as the rate at which the number of encountered objects grow as the considered range of distances expands from a reference point. LID is extended to outlier detection due to the local assessment of the growth rate of the cumulative distribution function(CDF) when the distance r increases[13][35]. It means that the expansion in the distance results in relatively small increases in the number of observations in the local neighborhood. This can be used to discriminate the distribution of the distance at any given point.

Several estimators of LID were proposed like Method of Moments, Method of Probability-Weighted moments and Maximum Likelihood Estimator[32]. Among these estimators of LID, the Maximum Likelihood Estimator (MLE) is most popular in terms of efficiency and simplicity. One limitation for the MLE estimator is that it suffers from a negative bias for high dimensions data as same as other dimension estimators[36]. Here we use the MLE estimator of the LID:

$$LID(x) = -(\frac{1}{k}\sum_{i=1}^{k} log\frac{r_i(x)}{r_{max}(x)})^{-1} = -(\frac{1}{|S|}\sum_{i=1}^{k} log\frac{r_i(x)}{r_{max}(x)})^{-1}$$

$$Novelty(x) = -(\frac{1}{|S|}\sum_{j\in S} log\frac{dist(x, x_j)}{dist(x, x_{max})})^{-1}$$

$$S = KNN(b(\pi_\theta), A)$$

The $r_i(x)$ is distance between x and its i-th nearest neighbour in the archive, where $i \leq k$. The $r_{max}(x)$ is the maximum of k-nearest neighbour distances. So the novelty tends to have large LID value compared with the common data points.

3.4.3 Failure of KNN in some local cases

Both KNN and LID provide a local view of the data structure of data points. But LID is likely to capture more complex novel patterns than the KNN and this raises the probability that the agent escapes from the deceptive trap. As shown in figure 1, the KNN distance of the red point is small if the blue points are close to the red points. Small KNN value implies that the red point is not novel from the blue points because it is close enough to its k nearest neighbor. However, from the human's view, the red points owns a different pattern from the blue points. So KNN fails in this case. LID can differentiate the red points by evaluating the expansion rate of the red points or its capability to fill the space in its vicinity. Generally speaking, KNN directly measures the novelty based on the average distance from the nearest point in the local neighborhood of the referenced point. In contrast, LID is the derivative of the distance function, which quantifies

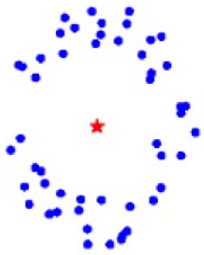

Figure 1: This example shows how KNN measures can fail to characterize the spatial properties of the red point[12].

the growth rate of the distance function in the vicinity of the referenced point. So the derivative of distance is possible to find some complex patterns which is not discovered by the simple distance metric like KNN. Also, it is noticed that both large LID or KNN values do not necessarily imply the novelty since it only provides a local view of the data points.

4 Methods

4.1 Diversity Quality: NSR-ES

The algorithm aims to estimate a parameter θ of the neural network, which can produce a policy which can simultaneously avoid the local optima and obtain higher scores or long-term cumulative rewards. The policy is the sequence of actions that the agent will take. The action is the output of neural network given the input observation in each timestep of the game in our case.

Natural Evolutionary Strategies (NES) searches for the optimal parameter θ in the parameter space Θ of the convolutional neural network. It generates the offspring of candidate parameters by the mutation. That is to say that a random noise from the normal distribution is added to the θ and we get the new descendent parameters: $\theta + \epsilon$. A random noise from a more complex distribution could also be used here. The fitness $f(\theta + \epsilon)$ of the parameter is evaluated by the cumulative reward which the agent has received in a whole episode of the game. θ is updated based on the rewards it has received from the population of the descendant agents

according to the formula 1. Naturally, NES could reduce the possibility of being stuck in the local optima to some degree because it uses a population of the parameters and initializes them randomly. The parameter updated in NES is given by the formula (2), where n is the number of agents, η is the learning rate and σ is the standard deviation of the normal noise.

$$\theta_m^{t+1} = \theta_m^t + \eta \frac{1}{n\sigma} \sum_{i=1}^{n} F(\theta_{i,m}^t)\epsilon_i \tag{2}$$

However, NES is not efficient in avoiding the local optima since the random noise ϵ is undirected and stochastic. So the novelty is proposed to augment the Evolutionary Strategies[9]. Rather than inspire the agent to look for novel state separately, the novelty here encourages the policy which exhibits different behaviors in a whole game compared with those previously seen. Even though search based on novelty only could outperform the reward-based algorithm in some cases with a deceptive trap[7], it generally does not perform well since it does not exploit the reward information. In this case, the quality diversity algorithm: NSR-ES algorithm is proposed to combine the novelty and evolutionary strategies to explore novel behaviors with qualityconti2018improving. The novelty in the original NSR-ES algorithm is measured by the value of KNN distance. By combining with the novelty, we extend the formula (2) into:

$$\theta_m^{t+1} = \theta_m^t + \eta \frac{1}{n\sigma} \sum_{i=1}^{n} \frac{F(\theta_{i,m}^t) + N(\theta_{i,m}^t, A)}{2}\epsilon_i \tag{3}$$

The NSR-ES is generalized by measuring novelty from any metric into the following procedure. The NSR-ES algorithm is shown in Algorithm 1 and a general diagram is presented in figure 2 to give a big picture of how it works[9].

The parallel NSR-ES adopts a Master/slave architecture. Firstly, the master node initializes a meta-population of parameters $\{\theta_1^0, ..., \theta_M^0\}$ for additional diversity. In the first generation, the Convolutional Neural Network plays a whole episode of the game given random parameters. We concatenate a whole sequence of RAM states in the game and add them into the archive A of the behavior characterization.

These RAM states are pre-defined and provided by the openAI gym environment as the behavior characterization to simplify the work[9]. RAM states in the games are integer-valued vectors of length 128 in the range $[0, 255]$ which describes the fully observable states in a game, for example, the position of the agent and enemies. A usual size of $b(\pi_\theta)$ is 512×356. Then a parameter θ_m is selected from the meta-population following the probability distribution of the novelty of the behavior characterization according to formula (4).

$$P(\theta_m^t) = \frac{N(\theta_m^t, A)}{\sum_{i=1}^{M} N(\theta_i^t, A)} \tag{4}$$

After that each slave or agent recieves a copy of the parameter θ_m^t from the master node. A random noise from normal distribution with standard deviation σ is added and get the candidate parameters $\theta_m^t + \epsilon_{i,m}^t$ for $i = 1, ..., n$ workers. The CNN with perturbed parameter $\theta_m^t + \epsilon_{i,m}^t$ plays the same game again. We will calculate the cumulative reward R_i in a game as the fitness F_i and the novelty N_i of the $\theta_m^t + \epsilon_{i,m}^t$ against the archive. The resultant F_i^t and N_i^t are transmitted back to the master node. The master node updates the parameter θ_m^{t+1} according to the formula (3). Finally, the θ_m^{t+1} is added into the meta-population and behavior charaterization of θ_m^{t+1} is addd into the archive A. The algorithm stops until N generations.

Algorithm 1 NSR-ES

Input: standard deviation of normal noise σ, learning rate η, generations N, population of parameters to keep M, the number k of KNN or LID

Output: parameter θ_m^N

Randomly initialize a population of parameters $\{\theta_1^0, ..., \theta_M^0\}$, an archive A of BC, and number of workers n

for $i = 1$ *to* M **do**

  Compute $b(\pi_{\theta_i^0})$ by playing an episode of game and contacting the RAM states

  Add $b(\pi_{\theta_i^0})$ to A

end

while $i \leq N$ **do**

  Sample θ_m^t from $\{\theta_1^0, ..., \theta_M^0\}$ according to $P(\theta_m) = \frac{N(\theta_m, A)}{\sum_{i=1}^{M} N(\theta_i, A)}$

  for $i = 1$ *to* n **do**

    Sample $\epsilon_i \sim N(0, \sigma^2 I)$

    $\theta_{i,m}^t = \theta_m^t + \epsilon_i$

    Compute $b(\pi_{\theta_{i,m}^t})$ by playing an episode of game and contacting the RAM states

    Compute the novelty N_i of $\theta_{i,m}^t$ against the archive A in k nearest neighbourhood

    Compute the fitness F_i as the cumulative rewards in the game

    send F_i and N_i back to the master node

  end

  $\theta_m^{t+1} = \theta_m^t + \eta \frac{1}{n\sigma} \sum_{i=1}^{n} \frac{N_i + F_i}{2} \epsilon_i$

  Compute $b(\pi_{\theta_m^{t+1}})$ by playing an episode of game and contacting the RAM states

  Add $b(\pi_{\theta_{i,m}^t})$ to A

end

4.2 KNN and LID based NSR-ES

KNN and LID based NSR-ES are different in how to calculate the novelty. The novelty is computed as the behavior characterization of the current parameter θ against the archive of the previous behavior characterizations $b(\pi_{\theta_{old}})$.

KNN based NSR-ES calculates the $\theta_{i,m}^t$ against the archive A using the k nearest Neighbour distance shown in the formula (5). S is the set of neareast neighbor of $b(\pi_{\theta_{i,m}^t})$ in the archive A: $b(\pi_{\theta_{j,m}^t})$ where $j = 1, ..., k$.

$$N(\theta_{i,m}^t, A) = \frac{1}{|S|} \sum_{j \in S} ||b(\pi_{\theta_{i,m}^t}) - b(\pi_{\theta_{j,m}^t})||_2 \tag{5}$$

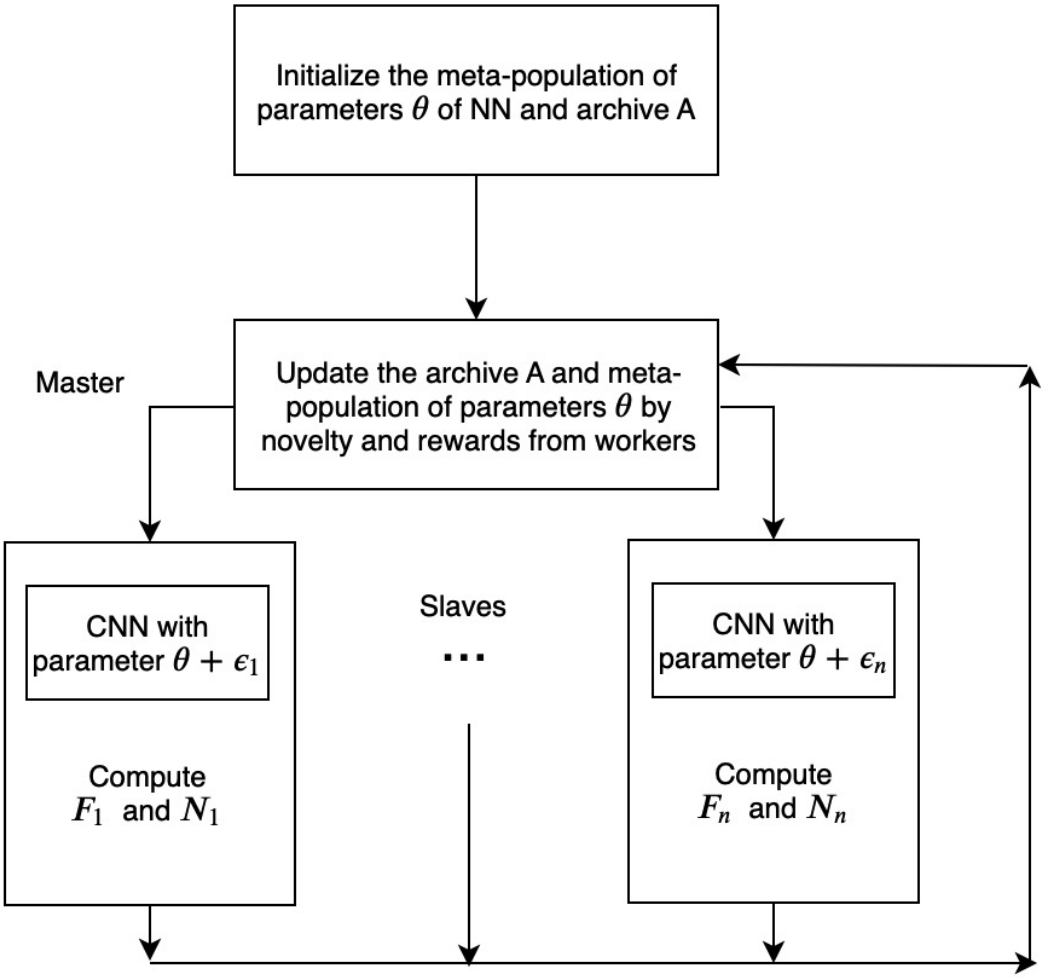

Figure 2: Diagram of the algorithm

LID based NSR-ES calculate the $\theta_{i,m}^t$ against the archive A in the neighborhood of the k nearest Neighbour as shown in the formula (6). The $b(\pi_{\theta_{max,m}^t})$ is the behavior characterisation of parameter has the largest distance against $\theta_{i,m}^t$ in the local neighborhood.

$$N(\theta_{i,m}^t, A) = -(\frac{1}{|S|}\sum_{j\in S}\log\frac{||b(\pi_{\theta_{i,m}^t}) - b(\pi_{\theta_{j,m}^t})||_2}{||b(\pi_{\theta_{i,m}^t}), b(\pi_{\theta_{max,m}^t})||_2})^{-1} \tag{6}$$

4.3 Some Training Tricks

4.3.1 Virtual Batch Normalisation

Instead of using batch normalization, the virtual batch normalization is used to reduce the problem caused by batch normalization that the output of the neural network for an input sample x is largely dependent on other inputs x's in the same mini-batch[37]. In the experiment, the virtual batch normalization will collect a fixed reference batch of size 128 at the very beginning of the training. According to extended, the normal perturbations tend to lead to monotone action policy without using the virtual batch normalization. This single-action policy will result in poor results.

4.3.2 Mirrorred sampling

In the experiment, rather than using a single random vector $\theta + \epsilon$, we always sampling with pairs of perturbations $\theta + \epsilon$ and $\theta - \epsilon$ to increase the robustness of the Evolutionary Strategies, which is known as the mirrored sampling or antithetic sampling [38]. Also, it is shown to lifts the convergence rate of Evolutionary Strategies algorithm with the large sample size.

4.3.3 Rank Normalise

Before averaging the novelty and the reward, we rank-normalize them independently to improve the robustness. That is to say that the novelty or the reward is sorted by the ranking and the new values are based on the ranking[28]. It shows that the rank-normalize will eliminate the influence of outliers in each population

and reduce the trend for Evolutionary Strategies fall into a local optima in the early time.

5 Experiments

5.1 Environment

5.1.1 Physical Environment

In the experiment, we use three Linux servers from the Nectar Research Cloud. Each one is a Ubuntu 16.04 LTS server with 8 CPUs and 32GB physical memory. All of the experiments are executed in the same condition in the sense that no other applications are running in the backend.

5.1.2 Software Environment

For the experiments, we use the gym environment from the OpenAI with Google Deepmind style wrappers[39][3]. In the atari games, the objective is to achieve the highest score. In this case, the state of the agent is a frame of the game or the raw pixels. The agent makes actions e.g left, right, up, down, and fire following the policy π_θ. After each action, the agent gets the immediate reward from the game at each time step reflected by the score.

5.2 Architecture

5.2.1 Architecture of the Program

A good understanding of the architecture contributes to the modification and debugging of the distributed framework. The implementation of the algorithm uses the master/slave architecture. In the figure 3, the master program firstly setups the experiment environments and uploads data like parameters of the model θ into the master Redis. Then it pushes the data into the relay server and the relay server allocates the tasks to different local worker Redis via the pipeline. The workers acquire the tasks and data from the local Redis. After that, each of the workers runs a convolutional neural network based on the parameters from the

local Redis with a normal noise, $\theta + \epsilon$. In the implementation, a huge noise table is set up by the master node in the very beginning and shared by all the workers to reduce the cost and time complexity. Then the convolutional neural network plays a whole episode of the atari games and the cumulative rewards are recorded. After an episode of the game, the workers write the cumulative rewards and the novelty computed from the behavior characterization of parameter $\theta + \epsilon$ against the archive back to the local Redis. The relay server pulls the result from the local Redis and pushes the result back to the master Redis. Finally, the master updates the parameters θ and the archive. The above procedure is a generation of the iteration of the NSR-ES algorithm. The programs end when there is no significant improvement on the rewards or in a limited number of generations.

5.2.2 Architecture of the Convolutional Neural Network

The convolutional neural network outputs the actions in each step given the observations based on the parameters after perturbations $\theta + \epsilon$.

The same framework of the Convolutional neural work is used in the experiment[9] as shown in figure 4. The size of the input layer and output layer have been changed depending on the observation space and the action space of each atari game. Usually, the input is $210 \times 160 \times 3$ tensor and the output is 18 different actions. Except for the input and output layer, the hidden layers are made up of two convolutional layers with 16 filters of size 8x8 with stride 4 and 32 filters of size 4x4 with stride 2. After that, there is a fully-connected layer with 256 hidden units, followed by a linear output layer with one node for each action. ReLu[40] is employed as the activation function between different layers with Adam optimizer[41]. Instead using the batch normalization to speed up the learning, the virtual batch normalization is applied to avoid the dependency problem in the mini-batch dataset[37]. Also, the parameters of the NSR-ES is specified in table 1.

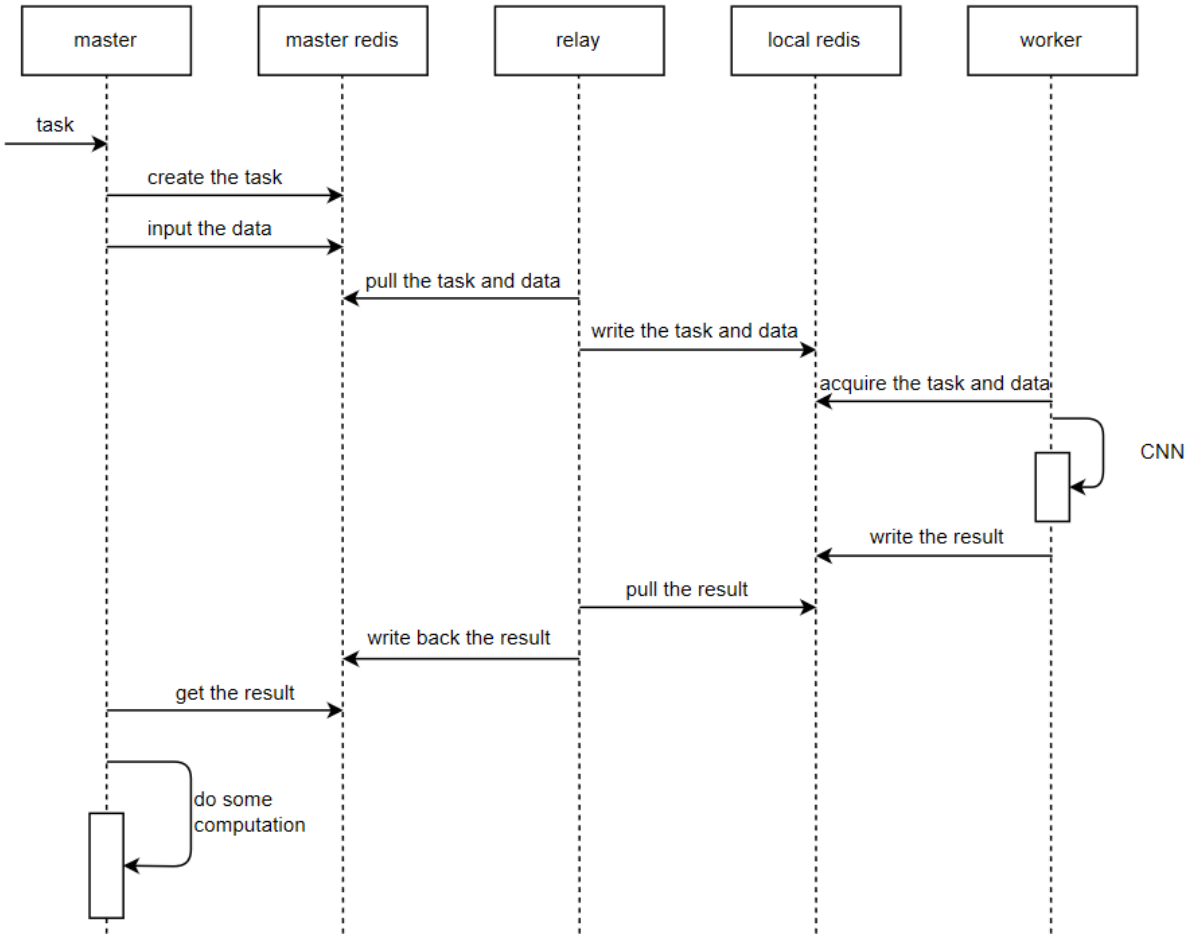

Figure 3: Sequence Diagram

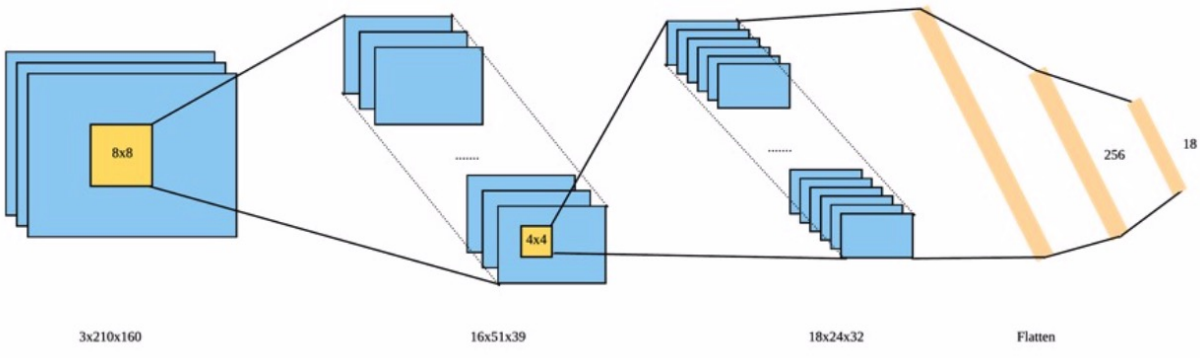

Figure 4: Architecture of Convolutional Neural Network

Table 1: Parameters of NSR-ES algorithm

learning rate	noise standard deviation	size of meta population	generations
0.01	0.02	3	100

5.3 Results

We test both the KNN based NSR-ES algorithm and the LID based NSR-ES algorithm on three Atari games: Alien, Frostbite, and Seaquest respectively. Alien does not have a deceptive trap while the last two have the local optima.

Due to the limited computing resources, each of the experiments usually costs 1 to 3 days to run while 80 machines with 1440 cores are used for the same task in Salimans et.al's experiment[5]. Unlike the original paper, the iterations are limited to 100 generations instead of 150 generations to reduce the computational costs.

5.3.1 Choice of the hyperparameter k

Firstly, k is specified to 10, 20 and 30 respectively in the experiments. The aim is to evaluate the impact of the choice hyperparameter k on the scores of each game. From the experiment result, it is found that the score of KNN-based method does not change with different k while the LID is sensitive to the values of k. With the increment of the k, the average score has increased and the variance is reduced. Moreover, the maximum score is also reduced. It suggests that the LID based-method is more stable with a large k.

Same as KNN, LID only provides a local view of the behavior characterization $b(\pi_\theta)$, so the novelty in the local neighborhood may not be considered to be novel from a global scale. But a larger neighborhood provides a global view of the data, which enables less variance of LID-based method. The reason why KNN based method is more stable will be discussed in the next part.

5.3.2 Average and Maximum Scores

Secondly, we compare the average scores and maximum scores in 100 generations between KNN based NSR-ES and LID based NSR-ES algorithms in three games. k

is specified to 10 since both KNN and LID method have large maximum cumulative rewards in this case. This could help to understand the potential of both methods as the novelty to maximize the expected return. The aim here is to evaluate the performance of KNN and LID as the novelty to augment the Evolutionary Strategies. Each experiment has been executed using the same settings at least five times.

In the game with the local optima like Frostbite and Seaquest, the LID based method outplays the KNN based method by a huge margin. As shown in figure 5, the LID based NSR-ES method nearly doubles the performance of the KNN based method in the game Frostbite. LID based algorithm achieves the maximum scores up to 5000 which nearly doubles the performance of KNN-based result to only 3000. Also, LID-based method scores around 3000 in just 20 generations, while KNN-based takes 90 generations. In other words, LID based NSR-ES is more efficient in the sense that it gets high scores in a shorter time. However, it is noticed that LID based NSR-ES has lower average rewards compared with the KNN based method. This implies that the score of the LID has quite a large variance.

This is also partly true for the game Seaquest. From the figure 6, LID based NSR-ES Algorithm scores up to 1400 in 100 generations while the KNN one could only achieve 1000. But the average score has the same trend for both cases.

In the game without a deceptive trap like Alien, there is no big difference between the maximum score of KNN or LID based NSR-ES as shown in figure 7. However, we could also find that there is a trend that the LID could achieve better scores in the early generations. For example, the LID based method scores up to 1000 score in 10 generations while the KNN one needs 80 generations.

5.3.3 Comparison based on the average values

The values of KNN and LID in each generation could describe how differently they perform in the behavior space. We compare the values of the average KNN and LID against the archive in each generation in one game. The average is computed from different agents since each worker returns a KNN and LID value to the

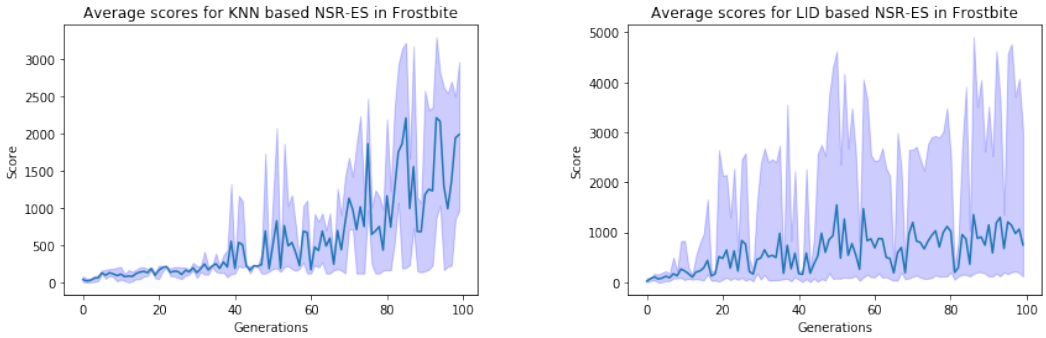

Figure 5: The left graph is the score for the KNN based NSR-ES and right graph is for LID based method in Frostbite. The blue line is the average scores in 5 repeated experiments using the same settings. The peak/valley of the shade is the maximum/minimum score the agent gets in each generation among the experiments.

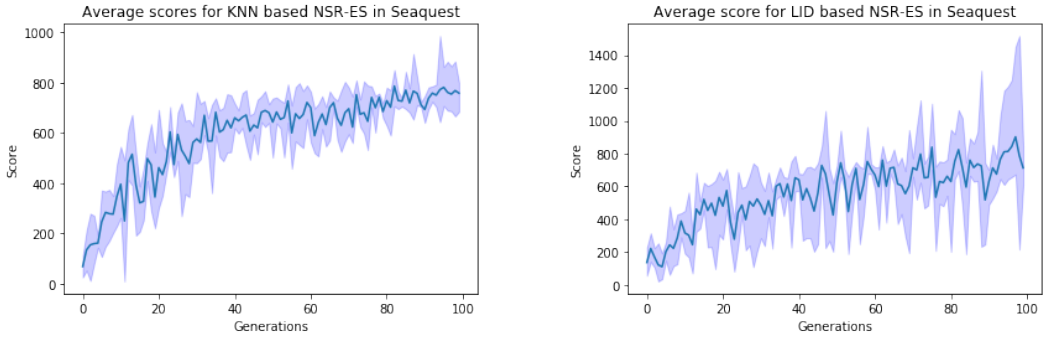

Figure 6: The left graph is the score for the KNN based NSR-ES and right graph is for LID based method in 5 repeated Seaquest games.

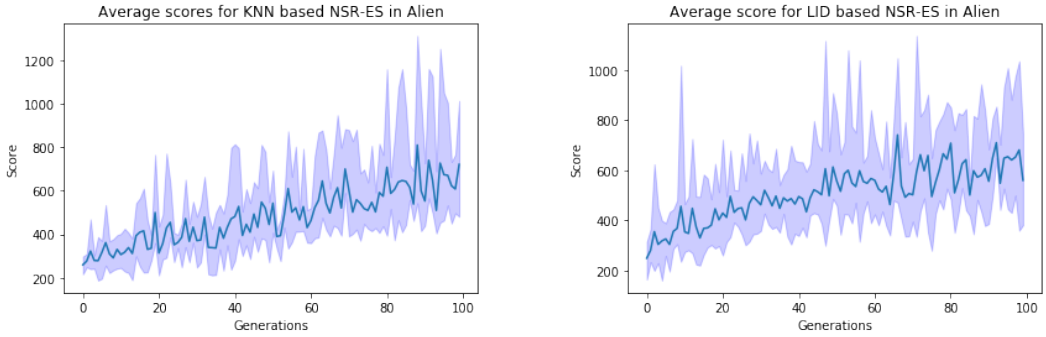

Figure 7: The left graph is the score for the KNN based NSR-ES and right graph is for LID based method in 5 repeated Alien games.

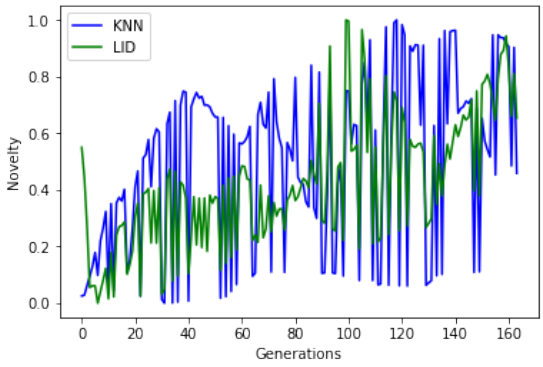

Figure 8: Novelty of the average KNN and LID in the game Frostbite which uses LID based NSR-ES.

master node in every generation. Also, both of KNN and LID values have been normalized by a min-max scaler to have a range in $[0, 1]$ interval. This is because the two metrics have different scales. As shown in figure 8, both of them have the same trend. However, the values of KNN have a rapid fluctuation compared with the LID ones.

6 Discussion

We assume that the agent achieves lower total cumulative reward due to being trapped in the local optima given the objective-oriented search algorithm. From the result of Frostbite and Seaquest, the LID achieves significantly higher performance in terms of the maximum score in the games. This indicates that the LID can discover some complex and novel behaviors which cannot be found by the KNN metric. It should be clear that higher novelty does not necessarily lead to higher rewards. But the novelty is possible to assist the agent to escape from the local optima, which is indirectly reflected by the maximum scores. In this sense, LID based agent escapes the local optima which traps the KNN based agent and obtains a higher score.

Also, the LID is more efficient in the sense that it gets the same score in a shorter time than KNN in both Frostbite and Alien. The reason is that KNN measures novelty based on the distance while LID evaluates the novelty based on the growth

rate of the distance for the data points encountered in the vicinity of $b(\pi_\theta)$. So LID can capture more complex pattern since it is a derivative of distance metric[13]. The first derivative of the distance can capture the change rate of the distance to the data points encountered in the local neighbour. This can be considered to be the expansion rate in the manifold.

However, LID based method also owns a large variance of scores in the Frostbite game. This indicates that the LID based method is unstable in some cases. Section 5.3.2 illustrates this point as well. Even though the LID based agent obtains much higher maximum cumulative rewards, it has lower average scores than the KNN based agent. Also, section 5.3.3 shows that LID has a sluggish fluctuation compared with the KNN as the novelty. This is natural since not all the novel behaviors have a large derivative of the CDF of the distance. But Section 5.3.1 also demonstrates that a large k is possible to reduce the instability by providing an enlarging overview of the data points.

7 Conclusions

In the research, we replace the K-nearest neighbor with Local Intrinsic Dimensionality as a measurement of novelty and hybridise it with the reward objective together to relieve the local optima problem in the deep reinforcement learning tasks. Experiments on several Atari games demonstrate that LID can capture some complex novelty which is not discovered by the KNN distance in the high dimensional space. This, in turn, helps the agent to escape from the deceptive trap and gets higher scores in the deep reinforcement learning tasks. But LID based Quality Diversity algorithm also has the limitation in the stability as shown in the experiments.

In future work, we plan to improve the stability of the LID based quality diversity algorithm by a dynamic weighting between both KNN and LID as the novelty. This may help improve the performance since KNN gives a more stable estimate of the novelty from the experiments and LID is possible to capture complex novelty. We also plan to increase the value of k since the larger k leads to the reduction of the

variance of the scores.

8 References

[1] A. Krizhevsky, I. Sutskever and G. E. Hinton, 'Imagenet classification with deep convolutional neural networks', in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] A. Graves, A.-r. Mohamed and G. Hinton, 'Speech recognition with deep recurrent neural networks', in *2013 IEEE international conference on acoustics, speech and signal processing*, IEEE, 2013, pp. 6645–6649.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, 'Human-level control through deep reinforcement learning', *Nature*, vol. 518, no. 7540, p. 529, 2015.

[4] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver and K. Kavukcuoglu, 'Asynchronous methods for deep reinforcement learning', in *International conference on machine learning*, 2016, pp. 1928–1937.

[5] T. Salimans, J. Ho, X. Chen, S. Sidor and I. Sutskever, 'Evolution strategies as a scalable alternative to reinforcement learning', *arXiv preprint arXiv:1703.03864*, 2017.

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[7] J. Lehman and K. O. Stanley, 'Novelty search and the problem with objectives', in *Genetic programming theory and practice IX*, Springer, 2011, pp. 37–56.

[8] T. Park and S. Levine, 'Inverse optimal control for humanoid locomotion', in *Robotics Science and Systems Workshop on Inverse Optimal Control and Robotic Learning from Demonstration*, 2013.

[9] E. Conti, V. Madhavan, F. P. Such, J. Lehman, K. Stanley and J. Clune, 'Improving exploration in evolution strategies for deep reinforcement learning

via a population of novelty-seeking agents', in *Advances in Neural Information Processing Systems*, 2018, pp. 5027–5038.

[10] S. Levine, Z. Popovic and V. Koltun, 'Nonlinear inverse reinforcement learning with gaussian processes', in *Advances in Neural Information Processing Systems*, 2011, pp. 19–27.

[11] O. Sigaud and F. Stulp, 'Policy search in continuous action domains: An overview', *Neural Networks*, 2019.

[12] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, D. Song, M. E. Houle and J. Bailey, 'Characterizing adversarial subspaces using local intrinsic dimensionality', *arXiv preprint arXiv:1801.02613*, 2018.

[13] M. E. Houle, 'Local intrinsic dimensionality i: An extreme-value-theoretic foundation for similarity applications', in *International Conference on Similarity Search and Applications*, Springer, 2017, pp. 64–79.

[14] S. Romano, O. Chelly, V. Nguyen, J. Bailey and M. E. Houle, 'Measuring dependency via intrinsic dimensionality', in *2016 23rd International Conference on Pattern Recognition (ICPR)*, IEEE, 2016, pp. 1207–1212.

[15] Z. B. Zabinsky, 'Random search algorithms', *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[16] J. Lehman and K. O. Stanley, 'Abandoning objectives: Evolution through the search for novelty alone', *Evolutionary computation*, vol. 19, no. 2, pp. 189–223, 2011.

[17] I. Rechenberg, 'Evolutionsstrategien', in *Simulationsmethoden in der Medizin und Biologie*, Springer, 1978, pp. 83–114.

[18] K. O. Stanley and R. Miikkulainen, 'Efficient evolution of neural network topologies', in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, IEEE, vol. 2, 2002, pp. 1757–1762.

[19] D. J. Lizotte, T. Wang, M. H. Bowling and D. Schuurmans, 'Automatic gait optimization with gaussian process regression.', in *IJCAI*, vol. 7, 2007, pp. 944–949.

[20] J. Hwangbo, C. Gehring, H. Sommer, R. Siegwart and J. Buchli, 'Rock—efficient black-box optimization for policy learning', in *2014 IEEE-RAS International Conference on Humanoid Robots*, IEEE, 2014, pp. 535–540.

[21] V. R. Konda and J. N. Tsitsiklis, 'Actor-critic algorithms', in *Advances in neural information processing systems*, 2000, pp. 1008–1014.

[22] J. K. Pugh, L. B. Soros, P. A. Szerlip and K. O. Stanley, 'Confronting the challenge of quality diversity', in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, ACM, 2015, pp. 967–974.

[23] A. Baranes and P.-Y. Oudeyer, 'Intrinsically motivated goal exploration for active motor learning in robots: A case study', in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 1766–1773.

[24] M. Joachimczak, R. Suzuki and T. Arita, 'Improving evolvability of morphologies and controllers of developmental soft-bodied robots with novelty search', *Frontiers in Robotics and AI*, vol. 2, p. 33, 2015.

[25] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton and R. Munos, 'Unifying count-based exploration and intrinsic motivation', in *Advances in Neural Information Processing Systems*, 2016, pp. 1471–1479.

[26] H. Tang, R. Houthooft, D. Foote, A. Stooke, O. X. Chen, Y. Duan, J. Schulman, F. DeTurck and P. Abbeel, '# exploration: A study of count-based exploration for deep reinforcement learning', in *Advances in neural information processing systems*, 2017, pp. 2753–2762.

[27] E. Zitzler, *Evolutionary algorithms for multiobjective optimization: Methods and applications*. Citeseer, 1999, vol. 63.

[28] D. Wierstra, T. Schaul, J. Peters and J. Schmidhuber, 'Natural evolution strategies', in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, IEEE, 2008, pp. 3381–3387.

[29] J. Lehman and K. O. Stanley, 'Exploiting open-endedness to solve problems through the search for novelty.', in *ALIFE*, 2008, pp. 329–336.

[30] A. Cully, J. Clune, D. Tarapore and J.-B. Mouret, 'Robots that can adapt like animals', *Nature*, vol. 521, no. 7553, p. 503, 2015.

[31] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, 'A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii', in *International conference on parallel problem solving from nature*, Springer, 2000, pp. 849–858.

[32] L. Amsaleg, O. Chelly, T. Furon, S. Girard, M. E. Houle, K.-i. Kawarabayashi and M. Nett, 'Estimating local intrinsic dimensionality', in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2015, pp. 29–38.

[33] D. R. Karger and M. Ruhl, 'Finding nearest neighbors in growth-restricted metrics', in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, ACM, 2002, pp. 741–750.

[34] X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. M. Erfani, S.-T. Xia, S. Wijewickrema and J. Bailey, 'Dimensionality-driven learning with noisy labels', *arXiv preprint arXiv:1806.02612*, 2018.

[35] T. de Vries, S. Chawla and M. E. Houle, 'Density-preserving projections for large-scale local anomaly detection', *Knowledge and information systems*, vol. 32, no. 1, pp. 25–52, 2012.

[36] E. Levina and P. J. Bickel, 'Maximum likelihood estimation of intrinsic dimension', in *Advances in neural information processing systems*, 2005, pp. 777–784.

[37] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, 'Improved techniques for training gans', in *Advances in neural information processing systems*, 2016, pp. 2234–2242.

[38] D. Brockhoff, A. Auger, N. Hansen, D. V. Arnold and T. Hohm, 'Mirrored sampling and sequential selection for evolution strategies', in *International Conference on Parallel Problem Solving from Nature*, Springer, 2010, pp. 11–21.

[39] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang and W. Zaremba, 'Openai gym', *arXiv preprint arXiv:1606.01540*, 2016.

[40] V. Nair and G. E. Hinton, 'Rectified linear units improve restricted boltzmann machines', in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[41] D. P. Kingma and J. Ba, 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980*, 2014.