Extending Fluctuation Dissipation Relations to Policy Gradient Methods in Reinforcement Learning

Anna Brandenberger
anna.brandenberger@mail.mcgill.ca

Gavin McCracken
gavin.mccracken@mail.mcgill.ca

School of Computer Science, McGill University
For Prof. Prakash Panangaden

1 Introduction

This paper begins by introducing fluctuation dissipation relations in section 2, where we explain both the original physical motivations as well as the derivation. This serves as background for section 3, in which we summarize and critically review Sho Yaida's paper, "Fluctuation Dissipation Relations (FDR) for Stochastic Gradient Descent". In this paper, Yaida introduces a novel physics-inspired idea of characterizing the stationary states of stochastic gradient descent using fluctuation-dissipation relations. He then designs an algorithm which can be used for supervised learning with stochastic gradient descent methods to automatically schedule the reduction of the step-size hyperparameter. In section 4, we extend the idea of detecting stationarity using FDR to the setting of policy gradient methods for reinforcement learning. We do this by deriving a fluctuation dissipation relation for policy gradients and constructing upon it to create the first policy gradient algorithm that automatically schedules both exploration and step-size. It is worth noting that the algorithm also automatically detects when further adjustments to step-size or exploration would be harmful, which makes it a perfect fit for use in arbitrary reinforcement learning environments. We report preliminary empirical results in section 5, to support the previously derived mathematical results.

2 Fluctuation-Dissipation Relations in Physics

Fluctuation-dissipation relations first came up in statistical mechanics in the study of Brownian motion, which is the random motion of a small particle in water [Feynman, 1998, Reif, 2009, Eastman, 2015]. They describe the equilibrium state of many systems which are macroscopically stationary but microscopically rapidly fluctuating. In this background section, we will present the case of Brownian motion.

The FDR equations quantify the relationship between friction, a macroscopic effect, and equilibrium fluctuations on the microscopic scale. Friction is the tendency of a system to move towards equilibrium as it converts kinetic energy to thermal energy. Thus when the system is not in equilibrium, it experiences a force pushing it towards equilibrium. On the other hand, equilibrium fluctuations occur on the microscopic scale as the particle's velocity is changed by collisions with water particles in the medium.

Both of these effects stem from interaction of the particle with the heat bath, thus it is conjectured that they should obey some relationships: this is exactly the fluctuation-dissipation relation. There are many ways of deriving this relation from basic statistical mechanics building blocks – we will present the main steps of Eastman's relatively straightforward analysis.

To analyse Brownian motion, physicists began by assuming a linear response: that the force acting on the particle is inversely proportional to its velocity. This assumption is satisfied in many important systems in physics, such as Brownian motion (the particle moving in water), and Johnson noise (thermal noise in the voltage through electrical resistors) [Marconi et al., 2008]. Taking this linear assumption as a statement about the *average* force, while the actual force at each time instant includes some random fluctuations, we can write the motion of the particle as follows, known as the Langevin equation:

$$m\ddot{x} = -\gamma\dot{x} + R, \tag{1}$$

where γ in first term is the friction coefficient, and R is a "random" force describing the interactions with surrounding environment particles. To describe the randomness of R, we make some reasonable assumptions: it has zero mean $\langle R\rangle = 0$ and is independent of the position x of the particle. It is also assumed that R is uncorrelated except on very short time scales (smaller than some τ) and this correlation is independent of time: $\langle R(t)R(t+dt)\rangle = 0\ \forall dt > \tau$.

The Langevin equation (1) is a simple ODE and can be solved for a time-dependent solution

$$\dot{x}(t) = \dot{x}(0)\cdot e^{-\frac{\gamma}{m}t} + \frac{1}{m}\int_0^t e^{-\frac{\gamma}{m}(t-t')}R(t')dt'. \tag{2}$$

Note that the first term is the solution to (1) without the random force R, and the second term shows the exponentially decaying effect of R.

To study the behaviour at equilibrium, we let $t\to\infty$. We also square $\dot{x}$ and take an ensemble average in order to bring in the average kinetic energy $\frac{1}{2}m\langle\dot{x}^2\rangle$

$$\lim_{t\to\infty}\langle\dot{x}^2\rangle = \frac{1}{m^2}\int_0^\infty\int_0^\infty e^{-\frac{\gamma}{m}(2t-t'-t'')}\langle R(t')R(t'')\rangle dt'dt'' \tag{3}$$

Now recall that we assumed the correlation $\langle R(t')R(t'')\rangle$ to be independent of time t, so we can translate this term in time to $\langle R(0)R(t''-t')\rangle$, then perform a change of variables in the integration to evaluate one of the two integrals. We then obtain

$$\lim_{t\to\infty}\langle\dot{x}^2\rangle = \frac{1}{2\gamma m}\int_{-\infty}^\infty e^{-\frac{\gamma}{m}s}\langle R(0)R(s)\rangle ds \tag{4}$$

Finally, we recall the equipartition theorem of classical statistical mechanics, which states that for a system at equilibrium with a reservoir of temperature T, the average kinetic energy is $kT/2$ for each degree of freedom. We can therefore replace $\lim_{t\to\infty}\langle\dot{x}^2\rangle$ by $(2/m)(kT/2)$. Assuming our Brownian motion here is in one dimension, we obtain our final result, the fluctuation-dissipation equation:

$$\gamma = \frac{1}{2kT}\int_{-\infty}^\infty e^{-\frac{\gamma}{m}s}\langle R(0)R(s)\rangle ds. \tag{P-FDR}$$

This equation quantifiably relates the friction force to the correlations in the random fluctuating force. Some generalizations of this equation to general Hamiltonian systems, out-of-equilibrium systems, etc. are presented in the review by Marconi et al. [2008]. We will not discuss them in this paper, as we are presenting the physics mostly as motivation and background for [Yaida, 2018].

Indeed, inspired by these systems in physics where one can link the macroscopic behaviour of the system to its microscopic fluctuations, Yaida examined stochastic gradient descent as such a system. The fluctuation relations he obtained for SGD relate the global optimization behaviour to the noise induced by the random sampling used to compute the gradient at each step.

3 Fluctuation-dissipation Relations for SGD

We begin by presenting and discussing Yaida [2018]'s two fluctuation relations for stochastic gradient descent (SGD), which were derived from the master equation.

Recall that gradient descent is a method of minimizing an objective function $f(\boldsymbol{\theta})$ with respect to its parameters $\boldsymbol{\theta} = \{\theta_i\}_{i=1}^{P}$, given N_S training examples. This loss function $f(\boldsymbol{\theta})$ can be written as a function of the examples as $f(\boldsymbol{\theta}) = \frac{1}{N_S}\sum_{i=1}^{N_S} f_i(\boldsymbol{\theta})$, where $f_i(\boldsymbol{\theta})$ evaluates the function on example i. At each step of the optimization, the gradient is exactly computed and the parameters are updated via the following rule:

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \alpha\nabla f(\boldsymbol{\theta}(t)) \coloneqq \boldsymbol{\theta}(t) - \alpha\sum_{i=1}^{N_S}\frac{1}{N_S}\nabla_i f_i(\boldsymbol{\theta}(t)),$$

where α is the step-size. This is known as batch gradient descent.

In stochastic gradient descent [Robbins and Monro, 1951], instead of computing the exact gradient, which requires evaluating the objective function for all the N_S training examples, the gradient is estimated from a randomly chosen "mini-batch" $\mathcal{B} \subset \{1, 2, \ldots, N_S\}$. So now we have $\nabla f^{\mathcal{B}}(\boldsymbol{\theta}) \coloneqq \frac{1}{|\mathcal{B}|}\sum_{\alpha\in\mathcal{B}} f_\alpha(\boldsymbol{\theta})$ and update rule

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \alpha\nabla f^{\mathcal{B}}(\boldsymbol{\theta}(t)). \tag{5}$$

Given this update rule for each time step, we can determine how the probability distribution of the model evolves in time. Furthermore, the stochasticity induced by the random mini-batch selection is reminiscent of the equilibrium fluctuations of a particle in a heat bath. Thus inspired by the physics literature discussed above in section 2, where a relationship was found between the random fluctuations of the particle and the dissipative force acting on it, Yaida [2018] finds fluctuation-dissipation relations for SGD which quantitatively link the noise in mini-batches to the evolution of the model performance.

Yaida first derives a master equation, then finds the two FDR relations by considering the master equation applied to different observables. Yaida argues that the first relation allows one to identify whether equilibriation has occurred, and the second gives information about the loss function landscape, specifically whether the harmonic approximation (quadratic truncation of the loss-function) is applicable.

We will discuss the assumptions made during the derivation of the master equation, as well as Yaida's conclusions regarding the first and second fluctuation-dissipation relations.

3.1 Derivation of the Master Equation

First note that, looking at the update equation (5), the $\nabla f^{\mathcal{B}}(\boldsymbol{\theta})$ part has mean (over possible mini-batches) the true gradient, denoted as following:

$$[\![\nabla f^{\mathcal{B}}(\boldsymbol{\theta})]\!] = \nabla f(\boldsymbol{\theta}), \tag{6}$$

where the notation $[\![\cdot]\!]$ denotes mean over mini-batches.

Let's define some more notation before getting to the fluctuation-dissipation relations.

Definition 3.1 (Noise Tensor $\tilde{C}$)**.** The k-point noise tensor is defined as having elements

$$\tilde{C}_{i_1,i_2,\ldots,i_k}(\boldsymbol{\theta}) \coloneqq [\![\partial_{i_1} f^{\mathcal{B}}(\boldsymbol{\theta}) \cdot \partial_{i_2} f^{\mathcal{B}}(\boldsymbol{\theta}) \cdots \partial_{i_k} f^{\mathcal{B}}(\boldsymbol{\theta})]\!].$$

This tensor characterizes the higher order fluctuations in the mini-batch gradient.

For example, the two point noise matrix would be

$$\tilde{C}_{i,j} = [\![\partial_i f^{\mathcal{B}}(\boldsymbol{\theta}) \cdot \partial_j f^{\mathcal{B}}(\boldsymbol{\theta})]\!] = \left[\!\!\left[\begin{matrix} \left(\partial_i f^{\mathcal{B}}(\boldsymbol{\theta})\right)^2 & \partial_i f^{\mathcal{B}}(\boldsymbol{\theta}) \cdot \partial_j f^{\mathcal{B}}(\boldsymbol{\theta}) \\ \partial_j f^{\mathcal{B}}(\boldsymbol{\theta}) \cdot \partial_i f^{\mathcal{B}}(\boldsymbol{\theta}) & \left(\partial_i f^{\mathcal{B}}(\boldsymbol{\theta})\right)^2 \end{matrix}\right]\!\!\right].$$

Now comes the most significant assumption of the paper. Yaida supposes that once learning is completed, SGD causes the state distributions $p(\boldsymbol{\theta}, t)$ to converge to a stationary distribution $p_{ss}(\boldsymbol{\theta})$ which dictates the SGD sampling at long time:

$$\exists\, T \text{ such that } \forall t > T,\ p(\boldsymbol{\theta}, t) = p_{ss}(\boldsymbol{\theta})$$

Definition 3.2 (Stationary-State Distribution)**.** This $p_{ss}(\boldsymbol{\theta})$ is a stationary distribution over parameters such that when acted on by SGD, the distribution $p_{ss}(\boldsymbol{\theta})$ remains constant. It is not unique: a given optimization problem may have several possible stationary distributions. For some intuition, an example of stationary distributions in the exact gradient descent limit ($|\mathcal{B}| = N_S$) are delta-distributions centered at all local minima.

Definition 3.3 (Stationary-State Average)**.** For any observable (measurable) quantity $\mathcal{O}(\boldsymbol{\theta})$ of the system (for example, θ or $f(\boldsymbol{\theta})$), its stationary-state average is defined as the mean over parameters when the system has converged to the stationary distribution $p_{ss}(\boldsymbol{\theta})$:

$$\langle \mathcal{O}(\boldsymbol{\theta}) \rangle \coloneqq \int d\boldsymbol{\theta}\, p_{ss}(\boldsymbol{\theta}) \mathcal{O}(\boldsymbol{\theta}).$$

To derive the master equation for SGD, we first consider the time evolution of the probability distribution of the model. This distribution evolves in time as an average over mini-batches of the probability of any previous state $\boldsymbol{\theta}'$ at time t (with distribution $p(\boldsymbol{\theta}', t)$) transitioning to state θ at time $t + 1$ via a SGD step:

$$p(\boldsymbol{\theta}, t + 1) = \left[\!\!\left[\int d\boldsymbol{\theta}' p(\boldsymbol{\theta}', t) \delta\left[\boldsymbol{\theta} - (\boldsymbol{\theta}' - \alpha \nabla f^{\mathcal{B}}(\boldsymbol{\theta}'))\right] \right]\!\!\right]. \tag{7}$$

Then, the master equation is found by considering the stationary distribution $p_{ss}(\boldsymbol{\theta})$ satisfying $p_{ss}(\boldsymbol{\theta}, t) = p_{ss}(\boldsymbol{\theta}, t + 1)$ and the stationary-state average of some observable $\mathcal{O}(\boldsymbol{\theta})$:

$$\langle \mathcal{O}(\boldsymbol{\theta}) \rangle = \langle [\![\mathcal{O}(\boldsymbol{\theta} - \alpha \nabla f^{\mathcal{B}}(\boldsymbol{\theta})]\!]_{m.b.} \rangle \tag{M}$$

Derivation.

$$\int d\boldsymbol{\theta}\, p_{ss}(\boldsymbol{\theta}, t) \mathcal{O}(\boldsymbol{\theta}) = \int d\boldsymbol{\theta}\, p_{ss}(\boldsymbol{\theta}, t + 1) \mathcal{O}(\boldsymbol{\theta}) \tag{8}$$

$$\text{Now plugging in (7)} \quad = \left[\!\!\left[\int d\boldsymbol{\theta} \int d\boldsymbol{\theta}' p_{ss}(\boldsymbol{\theta}', t) \delta\left[\theta - (\boldsymbol{\theta}' - \alpha \nabla f^{\mathcal{B}}(\boldsymbol{\theta}'))\right] \mathcal{O}(\boldsymbol{\theta}) \right]\!\!\right]_{m.b.}$$

$$= \int d\boldsymbol{\theta}' p_{ss}(\boldsymbol{\theta}') \left[\!\left[\mathcal{O}(\boldsymbol{\theta}' - \alpha \nabla f^{\mathcal{B}}(\boldsymbol{\theta}'))\right]\!\right]_{m.b.} \qquad \square$$

This master equation can be applied to any observables of the system, i.e. anything that can be measured, such as $\boldsymbol{\theta}$ or $f(\boldsymbol{\theta})$. We will consider $\theta_i\theta_j$ and $f(\boldsymbol{\theta})$ to derive the two fluctuation-dissipation relations FDR1 and FDR2. But first we can confirm our intuition that the gradient should have average zero, since it is randomly wandering around the local minimum. Indeed, applying the master equation (M) to the observable $\langle\boldsymbol{\theta}\rangle$, we get

$$\langle\boldsymbol{\theta}\rangle = \langle[\![\boldsymbol{\theta} - \alpha\nabla f^{\mathcal{B}}(\boldsymbol{\theta})]\!]_{m.b.}\rangle = \langle\boldsymbol{\theta}\rangle - \alpha\langle\nabla f(\boldsymbol{\theta})\rangle \implies \langle\nabla f\rangle = 0.$$

3.2 First Relation

$$\langle\boldsymbol{\theta}\cdot\nabla f\rangle = \frac{1}{2}\alpha\langle\operatorname{Tr}\tilde{C}\rangle \tag{FDR1}$$

For the more general case of SGD with momentum μ and dampening ν with update equation:

$$\begin{cases}(\boldsymbol{v}(t+1) = \mu\boldsymbol{v}(t) - (1-\nu)\nabla f^{\mathcal{B}}(\boldsymbol{\theta}(t))\\ \boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + \alpha\boldsymbol{v}(t+1)\end{cases} \tag{9}$$

the first fluctuation-dissipation relation is

$$\langle\boldsymbol{\theta}\cdot\nabla f\rangle = \frac{1+\mu}{2(1-\nu)}\alpha\left\langle\boldsymbol{v}^2\right\rangle. \tag{FDR1'}$$

Derivation. (FDR1) comes from considering the quadratic observable $\langle\theta_i\theta_j\rangle$ in (M):

$$\langle\theta_i\theta_j\rangle = \langle[\![(\theta_i - \alpha\partial_i f^{\mathcal{B}})(\theta_j - \alpha\partial_j f^{\mathcal{B}})]\!]\rangle$$

Expanding gives $\langle\theta_i\cdot\partial_j f\rangle + \langle\partial_i f\cdot\theta_j\rangle = \alpha\langle\tilde{C}_{i,j}\rangle$ and then taking the trace gives (FDR1).

(FDR1') can be derived similarly. Linear observables $\langle\boldsymbol{v}\rangle$ and $\langle\nabla f\rangle$ give $\langle\boldsymbol{v}\rangle = 0 = \langle\nabla f\rangle$, as expected; and the quadratic observables $\langle v_i v_j\rangle$, $\langle v_i\theta_j\rangle$ and $\langle\theta_i\theta_j\rangle$ expanded using (M) and combined yield the required equation, after some rearrangement. □

This first relation (FDR1) holds for any stationary state and both sides are easy to measure on the fly. Yaida [2018] therefore uses it to check if the model has reached equilibrium, and designs an adaptive training procedure to schedule hyperparameter changes once the model has reached an appropriate degree of stationarity.

(FDR1) is a nice theoretical result; however there are a few things to consider, regarding the practical utility of Yaida's learning-rate scheduling algorithm. First, we know that (FDR1) being satisfied does not guarantee that the model has converged, in the same way that $\langle\nabla f\rangle = 0$ does not guarantee it. Using this relation to test for convergence thus relies on the assumption that the chance of (FDR1) holding outside of stationary states is relatively low. This could be a reason for using (FDR1) in the algorithm rather than the simpler $\langle\nabla f\rangle = 0$ which only includes information about convergence on the first order.

3.3 Second Relation

$$\left\langle(\nabla f)^2\right\rangle = \frac{\alpha}{2}\langle\operatorname{Tr}\!\left(H\tilde{C}\right)\rangle - \alpha^2\left[\sum_{k=3}^{\infty}\frac{(-\alpha)^{k-3}}{k!}\left\langle\sum_{i_1=1}^{P}\cdots\sum_{i_k=1}^{P}F_{i_1,i_2,\ldots,i_k}\tilde{C}_{i_1,i_2,\ldots,i_k}\right\rangle\right] \tag{FDR2}$$

where $F_{i_1,i_2,\ldots,i_k} \coloneqq \partial_{i_1}\partial_{i_2}\cdots\partial_{i_k}f(\boldsymbol{\theta})$.

This relation applies to standard SGD. In a similar fashion as in the previous section, a more complicated relation can also be found for SGD with momentum and dampening.

Derivation of (FDR2). We consider the observable $\langle f(\boldsymbol{\theta})\rangle$ and apply the master equation (M). We then Taylor expand in the step-size α:[1]

$$\begin{aligned}\langle f(\boldsymbol{\theta})\rangle &= \left\langle \left[\!\left[f(\boldsymbol{\theta} - \alpha \nabla f^{\mathcal{B}}(\boldsymbol{\theta}))\right]\!\right]_{m.b.}\right\rangle \\ &= \left\langle f + \sum_{k=1}^{\infty} \frac{(-\alpha)^k}{k!} \sum_{i_1=1}^{P} \cdots \sum_{i_k=1}^{P} (\partial_{i_1}\cdots\partial_{i_k} f) \left[\!\left[\partial_{i_1} f^{\mathcal{B}} \cdot \partial_{i_2} f^{\mathcal{B}} \cdots \partial_{i_k} f^{\mathcal{B}}\right]\!\right] \right\rangle\end{aligned}$$

We then get (FDR2) by introducing $F_{i_1,i_2,\ldots,i_k}$ as defined above, recognizing the last term as $\tilde{C}$, and pulling the term with $k=1$ out of the sum to get $-\alpha \left\langle (\nabla f)^2 \right\rangle$. □

This second relation is potentially useful because the Hessian H, which gives information about the loss landscape, is computationally more expensive to compute than $(\nabla f)^2$. In the small step-size regime, the second term in (FDR2) is negligible, so $\mathrm{Tr}(H\tilde{C})$ can be approximated by $2\left\langle (\nabla f)^2 \right\rangle/\alpha$ at a local minimum. For larger step-sizes, $\left\langle (\nabla f)^2 \right\rangle$ can no longer be approximated to only have a linear dependence on α.

Yaida then suggests the use of (FDR2) for determining the validity of the harmonic approximation. This is the approximation in which we consider that the Hessian is constant and that all the higher order derivatives of f vanish. Yaida claims that $\langle(\boldsymbol{\nabla} f)^2\rangle$ having a nonlinear dependence on α in (FDR2) would indicate the breakdown of the harmonic approximation for the optimization problem. This is all in the small step-size regime, so we are looking at

$$\langle (\nabla f)^2 \rangle \approx \frac{1}{2}\left\langle \mathrm{Tr}\Big(H\tilde{C}\Big)\right\rangle \alpha$$

We can note that this linear dependence relies on $\langle \mathrm{Tr}(H\tilde{C})\rangle$ being constant. However, when the harmonic approximation applies, we know that H is constant, but $\tilde{C}$ may not be (it only involves first order derivatives of f). Therefore there could be situations where even though the harmonic approximation applies and H is constant, non-linearity still arises due to the noise from $\tilde{C}$. So Yaida's conclusion that (FDR2) is a test for the harmonic approximation is not entirely accurate.

Having presented this overview of [Yaida, 2018], we notice that in reinforcement learning, policy gradient methods rely on optimizing the parameters of a policy based on received rewards and gradient ascent. We thus apply the ideas of this paper to the policy gradient framework in order to detect stationarity in reinforcement learning.

4 Reinforcement Learning

We begin by introducing the reinforcement learning (RL) setting.

Definition 4.1 (Markov Decision Process)**.** A Markov decision process (MDP) is a 4-tuple, $\mathcal{M} = (\mathcal{S},\ \mathcal{A},\ \mathcal{P},\ r)$, where $\mathcal{S}$ is the set of the states, $\mathcal{A}$ is the set of all possible actions, $\mathcal{P}^a_{s,s'}$ is the

[1] Recall that a Taylor expansion of a function of of a vector valued function ($\mathbf{x} = (x_1, \ldots, x_P)^T$) is

$$f(\mathbf{x}+\boldsymbol{\alpha}) = f(\mathbf{x}) + \sum_{j=1}^{P} \frac{\partial f(\mathbf{x})}{\partial x_j}\alpha_j + \frac{1}{2}\sum_{i=1}^{P}\sum_{j=1}^{P} \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}\alpha_i\alpha_j + \ldots$$

transition probability from state s to state s' when performing action $a \in \mathcal{A}$ in state s, and $r(s \mid a)$ is the reward received when performing action a in state s.

An RL algorithm's goal is to interact with an MDP and discover an optimal mapping from states to the probabilities of selecting actions [Sutton and Barto, 2018]. This map is called a policy and is defined as a distribution, $\pi_{\boldsymbol{\theta}}(a \mid s) = \Pr\{\mathcal{A}_t = a \mid \mathcal{S}_t = s, \boldsymbol{\theta}_t = \boldsymbol{\theta}\}$. In essence, RL algorithms update a policy by navigating through an MDP's states, while constructing a history of the rewards that are received by taking actions in particular states. This history contains the rewards distributed according to $\mathcal{R}(a_t, s_t)$ and the state transitions distributed according to the transition matrix $\mathcal{P}^{a_t}_{s,s'}$. We thus get an optimization problem, where the objective function is generally the maximization of the return, defined as the cumulative reward received from following a policy. For some specific applications, other performance criteria may be used – e.g. in autonomous driving, the probability of catastrophic failure is used [Uesato et al., 2018].

4.1 Why Reinforcement Learning is Difficult

One of the more difficult problems in reinforcement learning relates to the exploration exploitation trade-off, that is, deciding when to explore a new action versus when to exploit an action currently expected to give a large reward. This phenomena was well studied in the multi-armed bandit problem, which is essentially an MDP with one state and a transition matrix where all actions cause deterministic transitions back to the only state, yet yield different rewards. In this situation, the goal is simply to discover the best action. However, in an MDP where we have multiple states and a stochastic transition matrix, the goal is now to discover an optimal sequence of actions (trajectory) through the MDP.

Exploration in this situation is substantially harder. Indeed, most multi-armed bandit algorithms store information between time steps by maintaining an array with running estimates of the expected value of each arm. Letting $\mathcal{K}$ be the set of arms, the space complexity for such an array is $O(|\mathcal{K}|)$. In contrast, in reinforcement learning, such a dynamic programming solution involves enumerating the $(\mathcal{S}, \mathcal{A}, \mathcal{S}')$ tuple into a three dimensional array with a space complexity of $O(|\mathcal{S}| \times |\mathcal{A}| \times |\mathcal{S}'|)$. This simple analysis shows that even for small MDPs, the space complexity is prohibitive. Consider the complete graph on 1000 vertices K_{1000}, with 1000 actions in each state. This MDP has $|\mathcal{S}| = 1000$, $|\mathcal{A}| = 1000$, $|S'| = 1000$. Thus, assuming 64-bit floats are used, a dynamic programming solution to this simple MDP with a one dimensional state-space already requires $1000^3 \times 64\text{bit} = 8\text{Gb}$ of space and it is clear that the worst case complexity of RL is many orders of magnitude more difficult than the bandit problem. In fact, RL in partially observable MDPs (where the agent is restricted from knowledge of its state in the MDP) has been determined to be PSPACE-complete [Papadimitriou and Tsitsiklis, 1987].

This prohibitive space complexity implies that RL algorithms must be much more complex with respect to how they explore than in state of the art bandit algorithms, where "remembering" past exploration is possible. In bandit problems, exploration could be done by simply choosing, with ε probability, any action uniformly at random. Of course, at long time such an algorithm will always select sub-optimal (random) actions with constant probability. To solve this problem, state of the art probably approximately correct (PAC) algorithms use concentration inequalities to decrease ε over time, by calculating the number of samples needed to guarantee that the expected reward estimates for each action are correct with high probability. We will not elaborate further on these algorithms in this paper.

In RL, exploration is often done with the exact same idea of choosing an action uniformly at random with ε probability. To emphasize the difficulty of exploration, consider how the diameter of an MDP comes into play for an MDP in which all transitions are deterministic. We define this diameter as the maximum distance between two nodes $D = \max_{x,y \in \mathcal{S}} d(x, y)$, where the distance $d(x, y)$ between states $x, y \in \mathcal{S}$ is the minimum number of transitions to get from x to y. In order to sample this trajectory with ε exploration, an event with probability $\leq (\varepsilon/|\mathcal{A}|)^D$ must occur. Compared to being able to choose which bandit to sample in $O(1)$ sample complexity, RL can easily be seen to yield an exponential number of samples with respect to diameter for exploration.

4.2 Preliminaries

We will investigate policies that are naturally stochastic by using the Boltzmann distribution to sample actions. Using a Boltzmann policy requires computing action preferences $h(s, a, \boldsymbol{\theta})$ for each action, which represent the policy's preference towards that action. The intuition behind this is that preferred actions will be explored orders of magnitude more often than actions thought to be close to the worst. Both the policy and the action preferences are parameterized via $\boldsymbol{\theta}$. Thus, the Boltzmann exploration policy is defined as

$$\pi_{\boldsymbol{\theta}}(a \mid s) = \frac{e^{h(s,a,\boldsymbol{\theta})/\tau}}{\sum_{b \in \mathcal{A}} e^{h(s,b,\boldsymbol{\theta})/\tau}}. \tag{10}$$

The parameters $\boldsymbol{\theta}$ are updated via gradient ascent according to the Policy Gradient Theorem [Sutton et al., 2000], which quantifies how changing the parameters of a policy changes the reward received by following the policy. The reward function of an MDP under a policy $\pi_{\boldsymbol{\theta}}$ is

$$J(\boldsymbol{\theta}) = \sum_{s \in \mathcal{S}} \mu^{\pi}(s) V^{\pi}(s) = \sum_{s \in \mathcal{S}} \mu^{\pi}(s) \sum_{a \in \mathcal{A}} \pi_{\boldsymbol{\theta}}(a|s) Q^{\pi}(s, a),$$

where $\mu^{\pi}(s)$ is the stationary state distribution under policy $\pi_{\boldsymbol{\theta}}$, $V^{\pi}(s)$ is the expected return of a state if this policy is followed, and $Q^{\pi}(s, a)$ is the Q value, which is defined to be the expected cumulative reward received by taking the action a in state s and following the policy afterwards. An intuitive description of the reward function for parameter $\boldsymbol{\theta}$ is the sum over all possible states $s \in \mathcal{S}$ of the visitation frequency of a state multiplied by the expected return yielded from following the policy $\pi_{\boldsymbol{\theta}}$. If we update our parameters using gradient ascent, the update rule is as follows:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \hat{\nabla}_{\boldsymbol{\theta}} J \text{ where } \boldsymbol{\theta} \text{ parametrizes the policy } \pi_{\boldsymbol{\theta}}(a|s).$$

Finally, the policy gradient theorem [Sutton et al., 2000] gives us an expansion of this gradient:

$$\begin{aligned}\hat{\nabla}_{\boldsymbol{\theta}} J &= \mathbb{E}_{\pi}\left[G_t \frac{\nabla \pi\left(A_t | S_t, \boldsymbol{\theta}\right)}{\pi\left(A_t | S_t, \boldsymbol{\theta}\right)}\right] \\ &= \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1}^{T} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_{i,t}|s_{i,t})\right) \left(\sum_{t=1}^{T} r(s_{i,t}|a_{i,t})\right)\end{aligned}$$

4.3 Bringing (FDR1) into policy gradient algorithms

We can notice that this policy gradient update rule is a gradient ascent rule, exactly like the updates used in SGD. Thus, we inspire ourselves from Yaida's analysis of stationarity of SGD in order to now investigate stationarity of parameters $\boldsymbol{\theta}$ in policy gradient RL.

Applying the procedure that Yaida used to derive the master equation (M) and the first fluctuation dissipation relation (FDR1) for SGD to our setting of policy gradient RL, we get

$$\langle \boldsymbol{\theta} \cdot \nabla_{\boldsymbol{\theta}} J \rangle = -\frac{\alpha}{2} \langle \operatorname{Tr} C_{ij} \rangle \tag{11}$$

where $C_{ij} = \partial_i J \partial_j J$ is the two-point noise matrix and $\langle \mathcal{O}(\boldsymbol{\theta}) \rangle = \int p_{ss}(\boldsymbol{\theta}) \mathcal{O}(\boldsymbol{\theta})$ is the stationary state average, as defined previously. This tells us that we can use this fluctuation dissipation relation to check if our policy gradient algorithm has converged into a maximum. The problem however, is that the above derivation does not include an exploration parameter. Rederiving the relation with a Boltzmann policy by allowing temperature to be a parameter, i.e. letting $\boldsymbol{\theta}' = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n, \tau]$, yields for $\frac{\partial J}{\partial \tau}$ (see Appendix A.1):

$$\frac{\partial J}{\partial \tau} = \frac{1}{\tau^2} \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1}^{T} \frac{\sum_{b \in \mathcal{A}} h(s, b, \boldsymbol{\theta}) e^{h(s,b,\boldsymbol{\theta})/\tau}}{\sum_{c \in \mathcal{A}} e^{h(s,c,\boldsymbol{\theta})/\tau}} - h(s, a_{i,t}, \boldsymbol{\theta}) \right) \left(\sum_{t=1}^{T} r(s_{i,t} | a_{i,t}) \right)$$

and thus the left hand side of (11) becomes

$$\left\langle \boldsymbol{\theta}' \cdot \nabla_{\boldsymbol{\theta}'} J \right\rangle = \langle \boldsymbol{\theta} \cdot \nabla_{\boldsymbol{\theta}} J \rangle + \left\langle \tau \frac{\partial J}{\partial \tau} \right\rangle . \tag{FDR1RL}$$

This is an exciting result because it tells us two things. Firstly, we can hold the temperature constant without affecting the assumptions of (FDR1) because the derivative with respect to the temperature term becomes 0. Secondly, we can perform gradient updates with a policy that can set its own exploration parameter τ. Indeed, for example, if a maximum for a particular value function were to require a degree of stochasticity in the policy, given that stochasticity is controlled by the temperature τ, the algorithm could increase or decrease τ appropriately until this parameter reaches stationarity.

Using (FDR1RL), we devise Algorithm 1, which automatically schedules the reduction of the step-size parameter α and the exploration parameter, which can be either temperature τ or ε.

Algorithm 1: Automatic Step-Size and Exploration Scheduling

```
Result: Returns the values of $\alpha$ and $\tau$ to use in the next gradient step
Initialization: $\delta$: threshold at which (FDR1RL) should be considered approximately true
                X: how much to reduce $\alpha$ each time (FDR1RL) $< \delta$
                Y: how much to reduce $\tau$ each time (FDR1RL) $< \delta$
Arguments   : $\alpha$: step-size
                $\tau$: temperature, or $\varepsilon$ if doing $\varepsilon$-greedy exploration
if FDR1 $< \delta$ then
    $\alpha \leftarrow \alpha \cdot X$;
    $\tau \leftarrow \tau - \frac{\partial J}{\partial \tau} \cdot Y$;
    return ($\alpha$, $\tau$);
else
    return ($\alpha$, $\tau$);
end
```

Now, in the following section, we report the results of using (FDR1RL) and Algorithm 1 on a modified gambler's ruin problem (see the classical literature on random walks [Feller, 1968] for the Markov chain that inspired the MDP).

5 Empirical Results

Recalling the Boltzmann policy definition (10), we note that a policy can only be deterministic if one of its parameters reaches $+\infty$. For MDPs in which this situation occurs, (FDR1RL) may never become true. This is because the left hand side will approach 0 due to the gradient approaching 0 as the policy gets closer and closer to determinism. Meanwhile however, the right hand side of (FDR1RL) is always strictly greater than 0. The only time at which the two sides become equal would be at $t = \infty$, which is of course not testable empirically.[2] This fits exactly with the initial assumptions of Yaida's paper, where we required convergence to a stationary distribution after some finite time T.

This situation explains why we empirically test (FDR1RL) on an environment in which all maxima in the value function correspond to stochastic policies – in this situation, (FDR1RL) will be satisfied in finite time, and we can qualitatively observe its behaviour.

5.1 Gambler's Ruin

First we present the gambler's ruin MDP [Sutton and Barto, 2018], see Figure 1. This environment is intuitively described as a non-symmetric random walk on a closed and bounded integer number line with two absorbing states. It is formally defined as follows:

- The state space is $\mathcal{S} = \{0, \ldots, L\}$. There is a fixed starting state, $s_0 \in \mathcal{S}$. The leftmost and rightmost states, $s = 0$ and $s = L$, are absorbing states: once they are reached, the random walk ends and the agent receives its total reward. Finally, the agent is completely restricted from information related to its state.
- The action space is $\mathcal{A} = \{+1, -1\}$, which respectively increment the current state value by $+1$ or -1.
- The reward of every action is -1, except for the action leading to the rightmost absorbing state s_L, which gives a reward of $+\lambda$.

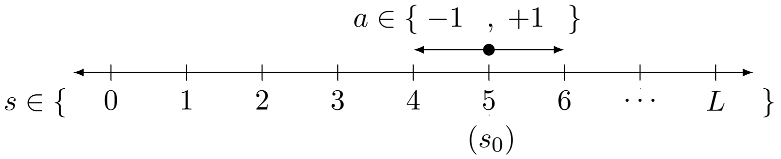

Figure 1: Depiction of the gambler's ruin environment. The far-left ($s = 0$) and far-right ($s = L$) states are absorbing (terminating) states, the starting state in this figure is $s_0 = 5$. The reward for reaching $s = L$ is λ, and every other action gives a reward of -1.

We slightly modify the gambler's ruin MDP by only changing the dynamics of two states: we make the states immediately adjacent to the left and right absorbing states have flipped actions. In other words, if the agent tries to go right on the flipped state, it will actually go left, and vice versa. This modification forces any policy to be stochastic. If the policy were deterministic (for example, always go right or always go left) then the agent will get stuck in an infinite loop and never hit an absorbing state. For example: if the policy goes right with probability p, and p is not

[2] Perhaps surprisingly, this situation doesn't actually break Algorithm 1, because in such a situation, we wouldn't want to reduce the step size. Doing so would just make it take longer to reach determinism.

1, then in expectation it will visit the second rightmost state $1/(1-p)$ times before it makes a left action (and actually goes right due to the flipped actions at this state).

5.2 Results

Using a linearly parameterized Boltzmann policy, we observe that the derivative with respect to temperature becomes zero once the parameters have converged to a maximum. This is great news, because it empirically affirms the intuition that (FDR1RL) can be used for the automatic scheduling of exploration. Essentially, if the derivative with respect to temperature is 0, we know that for the current values of the parameters, there is no benefit to increasing or decreasing exploration. This affirms our mathematically grounded intuitions that using fluctuation dissipation relations to detect stationarity holds merit for not only adjustments of step-size, but also adjustments of exploration.

Figure 2 elaborates on this. It shows that Algorithm 1 is successful on the flipped state gambler's ruin, where all maxima correspond to stochastic policies. It converges to the policy corresponding to "go right with probability $p = 0.76$" and does this without the need for a problem dependent "minimum value of stochasticity" hyperparameter that many state of the art algorithms use. This kind of parameter would essentially prevent a policy from converging to determinism by forcing it to select an action uniformly at random some percentage of the time. Naturally, in any MDP, this hyperparameter would have to be learned via hyperparameter search before an algorithm relying on it can even begin interacting with the problem.

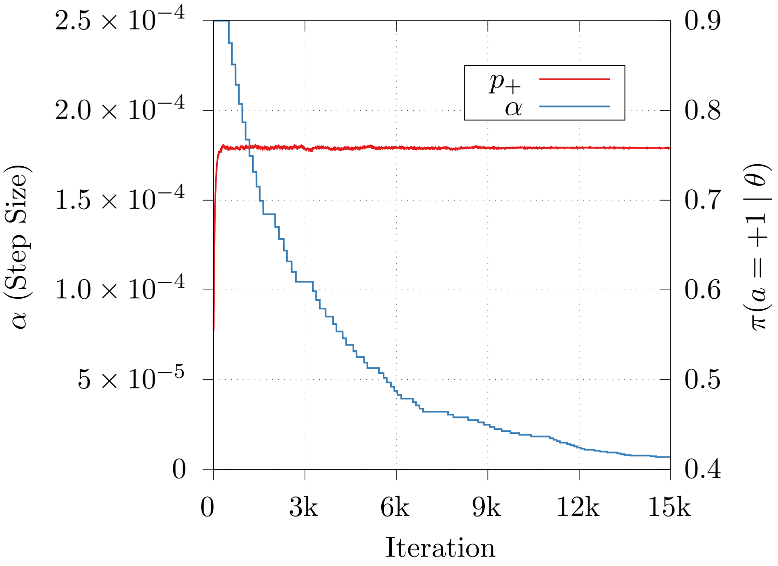

Figure 2: This figure shows the convergence to the global maximum on the flipped gambler's ruin MDP with $|\mathcal{S}| = 10$. The red line is the probability of taking action $+1$ (a step to the right), and the blue line is the value of the step size. This plot emphasizes two things. Firstly, our FDR relation can remove the "minimum value of stochasticity" parameter. This is great because it justifies the use of (FDR1RL) in practice. Secondly, we did not need to create problem dependent schedules for either the reduction of step-size or the reduction of exploration. Algorithm 1 took care of this automatically.

It is worth noting that although this experiment was performed with just linear function approximation of the action preferences, it could be repeated where action preferences are approximated using any arbitrary differentiable function. This follows from our derivation of (FDR1RL), which isolated the temperature parameter by separating it from the other parameters.

6 Conclusion and Future Work

We introduced fluctuation dissipation relations as they were originally conceived in the physics literature, then summarized and critiqued Yaida's paper "Fluctuation Dissipation Relations for Stochastic Gradient Descent". Following this, we used [Yaida, 2018] as a guide to derive a fluctuation dissipation relation for use in policy gradient methods in reinforcement learning, (FDR1RL). We then created an algorithm based on this relation, Algorithm 1, that replaces the need for problem dependent schedules for the adjustment of exploration and step-size over time. We verified the correctness of our algorithm on a simple environment in which we knew that all maxima in the value function resulted from stochastic policies.

Future empirical work could be done to compare the performance of (FDR1RL) versus predetermined schedules on some of the mainstream reinforcement learning environments. A strong argument for (FDR1RL) could be made if we obtain a small enough the difference in performance between our algorithm and the current state of the art method: performing a hyperparameter search for the minimum stochasticity parameters value, and subsequently searching via trial and error for a predetermined problem dependent schedule.

As far as a theoretical work goes, we could perform a comparison of how various optimizers for SGD change the convergence probabilities to different maxima in the gradient. Such research would require building on unpublished work currently under review in ICML by (McCracken, Daniels, Panangaden, Precup) which establishes theoretically understood non-observable environments. Extending this work to either partially observable or fully observable environments would allow for probabilistic analysis of different optimizers. It is likely that this extension can be made through using some group theoretic counting tools[3] to devise environments with appropriate symmetries, and thus derive probability density functions for the returns of these environments. These PDFs would allow us to determine the desired convergence probabilities of various optimizers at long time.

Finally, it has been empirically verified that in state of the art deep actor critic RL agents, the critic neural network must be close to stationarity before the actor neural network is updated. If this condition is not met, instability are caused by the gradient updates to the actor. Current solutions are purely empirical and are essentially just hacks. They all involve simple tricks, such as updating the critic 10 times for each single update to the actor. This is a place where our (FDR1RL) could shine, and yield a strong theoretical result where one is desperately needed.

References

Peter Eastman. *Introduction to Statistical Mechanics*. 2015. URL `https://web.stanford.edu/~peastman/statmech/`.

[3] e.g. Burnside's lemma and the Pólya enumeration theorem

William Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, S.l., 3rd edition, 1968. ISBN 978-0-471-25708-0.

Richard Feynman. Statistical mechanics: a set of lectures (advanced book classics). 1998.

Umberto Marconi, Andrea Puglisi, Lamberto Rondoni, and Angelo Vulpiani. Fluctuation-dissipation: Response theory in statistical physics. *Physics Reports*, 461, 04 2008. doi: 10.1016/j.physrep.2008.02.002.

Christos H Papadimitriou and John N Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

Frederick Reif. *Fundamentals of statistical and thermal physics*. Waveland Press, 2009.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

Jonathan Uesato, Ananya Kumar, Csaba Szepesvari, Tom Erez, Avraham Ruderman, Keith Anderson, Krishmamurthy, Dvijotham, Nicolas Heess, and Pushmeet Kohli. Rigorous Agent Evaluation: An Adversarial Approach to Uncover Catastrophic Failures. *arXiv:1812.01647 [cs, stat]*, December 2018.

Sho Yaida. Fluctuation-dissipation relations for stochastic gradient descent. *arXiv preprint arXiv:1810.00004*, 2018.

A Derivations

A.1 FDR1RL

Allowing temperature to be a parameter, let $\boldsymbol{\theta'} = [\theta_1, \ldots, \theta_n, \tau]$, and applying the steps outlined in Sho Yaida's paper to RL we derive the LHS of FDR1RL:

$$\left\langle \boldsymbol{\theta'} \cdot \nabla_{\boldsymbol{\theta'}} J \right\rangle = \langle \boldsymbol{\theta} \cdot \nabla_{\boldsymbol{\theta}} J \rangle + \left\langle \tau \frac{\partial J}{\partial \tau} \right\rangle$$

In order to calculate $\frac{\partial J}{\partial \tau}$ we need to calculate how the policy is changing with respect to temperature:

$$\begin{aligned}
\frac{\partial}{\partial \tau} \log(\pi_\theta(a \mid s)) &= \frac{\partial}{\partial \tau} \log\left(\frac{e^{h(s,a,\theta)/\tau}}{Z}\right) \\
&= \frac{\partial}{\partial \tau} \log\left(e^{h(s,a,\theta)/\tau}\right) - \frac{\partial}{\partial \tau} \log(Z) \\
&= -\frac{h(s,a,\theta)}{\tau^2} - \frac{\partial \beta}{\partial \tau}\frac{\partial}{\partial \beta} \log(Z) \qquad \text{where } \beta \equiv -\frac{1}{\tau} \\
&= -\frac{h(s,a,\theta)}{\tau^2} + \langle h \rangle_s \frac{\partial \beta}{\partial \tau} \\
&= -\frac{h(s,a,\theta)}{\tau^2} + \langle h \rangle_s \frac{1}{\tau^2} \\
&= \frac{1}{\tau^2} \left(\langle h \rangle_s - h(s,a,\theta)\right)
\end{aligned}$$

Now, we can substitute $\frac{\partial}{\partial \tau} \log(\pi_\theta(a \mid s))$ into the policy gradient theorem and pull out the temperature term as follows:

$$\begin{aligned}
\frac{\partial J}{\partial \tau} &= \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\frac{1}{\tau^2}\left(\langle h \rangle_s - h(s,a,\theta)\right)\right)\left(\sum_{t=1}^{T} r(s_{i,t}|a_{i,t})\right) \\
&= \frac{1}{\tau^2}\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\langle h \rangle_s - h(s,a,\theta)\right)\left(\sum_{t=1}^{T} r(s_{i,t}|a_{i,t})\right)
\end{aligned}$$

Thus, the left hand side of the fluctuation dissipation relation for policy gradient with a Boltzmann policy becomes:

$$\begin{aligned}
\left\langle \boldsymbol{\theta}' \cdot \nabla_{\boldsymbol{\theta}'} J \right\rangle &= \left\langle \boldsymbol{\theta} \cdot \nabla_{\boldsymbol{\theta}} J \right\rangle + \left\langle \tau \frac{\partial J}{\partial \tau} \right\rangle \\
&= \left\langle \boldsymbol{\theta} \cdot \nabla_{\boldsymbol{\theta}} J \right\rangle + \frac{1}{\tau}\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\langle h \rangle - h(s,a,\theta)\right)\left(\sum_{t=1}^{T} r(s_{i,t}|a_{i,t})\right)
\end{aligned}$$

B Implementation Details

B.1 Gambler's Ruin

Derivative wrt temperature for a parameterized Boltzmann policy in gamblers ruin:

$$\begin{aligned}
\frac{\partial}{\partial \tau} \log(\pi_\theta(a \mid s)) &= \frac{\partial}{\partial \tau}\{\log\left(1 - [0.5 \cdot (\tanh\frac{x}{\tau} + 1.0)\right],\ \log\left(0.5 \cdot (\tanh\frac{x}{\tau} + 1.0\right)\} \\
&= \left\{-\frac{x \operatorname{sech}^2\left(\frac{x}{t}\right)}{t^2\left(\tanh\left(\frac{x}{t}\right) - 1\right)}, -\frac{x \operatorname{sech}^2\left(\frac{x}{t}\right)}{t^2\left(\tanh\left(\frac{x}{t}\right) + 1\right)}\right\}
\end{aligned}$$