Side effect minimization in Reinforcement Learning

Jonatan Hellgren
under supervison of: Olle Häggström

April 2022

Contents

1: Introduction

What will be covered in this report will be a small part of the big problem of creating safe Artificial Intelligence (AI). This is an issue of great importance that we should not overlook since the consequences of what we manifest in the present or near future may last for our remaining history.

In this introduction, we begin by defining AI. Then go through where we are today, where current progress might lead, and when we can see these changes. After that, we will cover the risks in AI that make it possibly unsafe and what is at stake. Finally, we will look at some proposed methods for creating safe AI. (Bostrom 2016)

1.1 Artificial intelligence

In recent human history, we have seen massive technological development. Today our lives are in several ways different compared to centuries ago. Most of this development can be seen as the consequence of new tools, developed to extend our capability. In early prehistory, these tools were things such as fire for warmth, protection, and to cook our food to give us more nutrition, or weapons for hunting to strengthen our weak bodies. Later in history, these tools tend towards more complexity by automating physical labor with mechanical machines and extending the reach of the written word with the printing press. In modern times, a new tool has emerged intending to improve the thing that made all the previous tools possible, namely our intelligence. This tool is called AI and is starting to show its potential.

In the standard textbook on AI (Russel 1995), the authors say that the field is "concerned with not just understanding but also building intelligent entities - machines that can compute how to act effectively and safely in a wide range of novel situations". The definitions of what an AI is varies, on Wikipedia we find the definitionWikipedia:

> Artificial Intelligence (AI) is intelligence demonstrated by machines, as opposed to the natural intelligence displayed by animals including humans.

However, to understand this definition properly it is necessary to define what intelligence

is. In Tänkande Maskiner the author brings up the following to clarify this: “the quality that enables an entity to function effectively and with foresight in its environment” and “the ability to correctly perceive one’s surrounding environment and act in a way that maximizes one’s chances of achieving given goals”.

Ordinary computer programs contain step-by-step instructions that a computer can execute to perform the desired task. This method was also the case for the first two paradigms of AI: rule-based AI and expert systems where humans explicitly programmed their knowledge into the computer to create automation Superintelligence. The models made in such a way are typically suited for less complex tasks where it is possible to model the entire behavior explicitly. However, in the current machine learning paradigm, the approach is to train a model on large quantities of data. This approach focuses more on what should be solved instead of how it should be solved, which allows for automation in a more complex task where explicitly defining the behavior in every situation is infeasible.

AI has in recent years been applied in the industry more broadly, and it is already generating yearly revenue of trillions of dollarsRussel Norvig. This progress has in recent years become a possibility due to more data being available, faster computer hardware, and the massive amount of funding spent on research.

Although these systems are highly automated, a key point here is that these systems still require humans to create and function. However, the development of AI is shifting the tool to a more automated one. With this trend AI systems are developing in to intelligent agents. An agent acts or, more specifically, can:

- operate autonomously,
- perceive the environment,
- persist over a prolonged period,
- adapt to change,
- and create and pursue goals.

The reason why this is attractive is that the human intervention part required by an AI system is likely to become a bottleneckT’nkande Maskiner.

Reinforcement Learning (RL) is a field of machine learning where the objective is to create intelligent agents, this field differentiates from other machine learning techniques by being a more exploratory approach where the agent learns by trial and error. The method is similar to how one goes about training a pet, where desirable behavior receives a positive reward and undesired behavior gets a negative. In such a way, the agent develops a behavior.

In recent years, RL has seen substantial development with advances in board games such

as chess and goSilver et al., autonomous vehiclesLevinson et al., and video gamesMinh et al.. These advancements display the usefulness of these agents and motivate the possibility of implementation in our daily life. RL will be the main focus for this report, however it is inportant to keep it mind that it is not the only method for creating intelligent agents.

1.1.1 Future progress

When the pioneers of AI started the development, the ideas were not only to apply systems that automate a narrow set of tasks, as we can see in modern AI systems. The ideal was instead to recreate the intellect of a human in a machineMcCarthy et al., to extend our thoughts from mere thoughts to a new life form with a base of silicon-based hardware instead of carbon-based wetware. This concept is called Artificial General Intelligence (AGI) - an AI that can solve an arbitrary set of tasks with as good or better performance than a human. The main difference from AI is that the set of tasks is no longer narrow and bounded. An even more advanced AGI is often called *superintelligence* - an AI that is much smarter than a human in all possible domains. Superintelligence

An example of a current AI system is DeepMinds AlphaMu, which can play board games, and video games. It has recently been able to create a video compression better than the current methods used for the taskMandhane. The techniques used in this system originate from the famous AlphaGo, which won against the Go grandmaster Lee Sedol in the game of Go DeepMind. Although AlphaMu presents an impressive performance in multiple tasks, it still can not be considered an AGI since each new task require engineering effort. On the other hand, an AGI would be able to pick up any task and perform it at a human or better level without human intervention.

The significant difference with this shift is that it will increase the possible tasks that a single system can perform. The possible set of tasks would become arbitrary and performed at a human level or higher. The implications of such a breakthrough would likely be on the same scale as the industrial revolution, if not largerCritch Kruger. However, instead of automating physical labor, we would have automated mental labor. The following quote summarizes the potential impacts “Machine intelligence is the last invention that humanity need ever to make”I.J Good. This quote presents a dichotomy since we will either not be able to compete with creating novel inventions or not be able to preserve our existence. Reasons for the latter we will come back to in later sections.

A reason to believe that such systems are possible to build is that we know that human intelligence evolved naturally with evolution, so something similar should be possible to reproduce in machines. Created intelligence could become more intelligent than us since intelligence might not have been selected for by evolutionS. Legg and when we develop AI, we can focus the development specifically on intelligence. An argument against this

is substance dependence Bostrom (2003) - to believe that intelligence or consciousness can only occur in carbon-based life forms and not in silicon-based, caused by inherent or other properties. Regarding intelligence, there is no longer any reasonable argument for it. However, the question of consciousness, although being an interesting question, can be seen as irrelevant when considering what actions an AI makes since the consequence is still the same. Stuart Russel puts it:

> If human beings are losing every time, it doesn't matter whether they're losing to a conscious machine or an completely non conscious machine, they still lost.

An AGI breakthrough is probably unnecessary for AI to impact the world because all things we deem as intelligent do not help. For example, physical motor skills, food digestion, and blood pressure regulation are examples of intelligent things our brain does that are hard to connect to how an AI could use to impact the world. However, a more narrow set of intelligent behavior could exist that could have an impact. For example, if an AI could create convincing and motivating speeches, it could impact politics, legislation, and policymaking. Also, finance is where an AI could steer the world's funding towards its specific goals or crash the price for something it wants to purchase.

For this reason, many researchers have stopped talking about AGI and have instead refined the conceptsCritch Kruger. An AI system that is capable enough to induce transformative consequences on the same scale as the industrial or agricultural revolution is called an *transformative AI* (TAI). On the other hand, if a *transformative AI* once deployed would be unstoppable, it is instead called an *prepotent AI*.

Developing a TAI is not an easy task. However, it might not be necessary to create one directly for it to be createdSuperintelligence. A different approach is to create an AI system that can develop a TAI system. A fundamental property of this AI system is self-improvement. Theoretically, if an AI system has reached a threshold where it becomes better at improving itself than its creators, then letting the AI create the next version of itself, this self-improvement would improve. An even better version would be possible next. If this iterative process keeps going, it will create an intelligence explosion called the singularityYudkowsky.

1.1.2 Basic drives

Knowing what impacts a potential TAI or AGI will cause is hard without understanding how it will behave. There has been a lot of work laying the foundations for understanding the possible behavior by hypothesizing what drives it could have. A commonly adopted view (but still controversialMüller Cannon) is the Omohundro-Bostrom theory for AI driving forces. Two cornerstones together imply itO Häggstom, namely *instrumental convergence thesis* and the *orthogonality thesis*, which we will now explain further.

In the current paradigm an AI-agent is assigned a task, if this task is completed the agent reaches a terminal state. This task could be anything, for example maximizing the number of paper clips produced by a factory, finding decimals in π, or counting all the blades of grass on our planet. When an agent pursues this goal, naturally it arises other instrumental goals. Examples of such would be self-preservation, self-improvement, discretization, goal perseverance, and resource accumulationOmohundro. These instrumental goals help the agent pursue its terminal goal. For example, the agent wouldn't be able to perform its terminal goal if destroyed, thus self-preservation would arise. These instrumental goals will likely be shared between a wide range of different terminal goals, since pursuing them helps the agent achieve its terminal goal, regardless of what it is. Thus there is a set of instrumental goals which agents would naturally converge towards and hence the name.

Still, it does not yet exist any rigorous mathematical proof for this. However, some work has been trying to lay the necessary foundations for it TURNER et al.. In the paper, the authors prove in a simple environment that certain actions give the agent more power in the sense that more possible future actions become available, on average it is optimal to choose those actions that yield higher power. Thus we can see the pursuit of instrumental goals as a tendency to seek power.

LÄS https://www.emerald.com/insight/content/doi/10.1108/FS-04-2018-0039/full/pdf?title=challenges-to-the-omohundrobostrom-framework-for-ai-motivations

The orthogonality thesis described by Nick BostromBostrom2 states that the intelligence of an AI is logically independent of the goals it might have. An intelligent AI could have a stupid task from our point of view, such as counting all the blades of grass on our planet, or it can have a goal that we may deem as an important one, like keeping the climate on earth habitable for the species that currently live on it. For an AI each task would be as important, given that we assigned the goal to it during its creation.

1.1.3 Timeline for transformative AI breakthrough

The well-known AIs of today still have not reached the levels required for a TAI. For example, AlphaStar an AI that won against world champions in the complex computer game DotA 2Deepmind is estimated in terms of total computational power to be "about as sophisticated" as a beeA Cotra. While the state-of-the-art language generator model GPT-3 that can summarize, continue, and carry out convincing conversations is estimated to be "more sophisticated" than a beeA Cotra. Although, intelligence can not be measured with only computational power, since the intelligence of theses AI-system is vastly different compared to biological lifeforms.

This raises the question of when we will see these breakthroughs in the field that enables TAI systems? It is hard to answer, but with the worldwide increase in funding and researchnature, we are undoubtedly getting ever closer. There have been attempts to

answer this question, and the results of a survey and a more quantitative forecasting model will now be covered.

In a well-cited survey Grace et al (2017), they asked researchers in the field of AI to estimate the probability of human-level machine intelligence (unaided machines that can achieve all tasks better and more cheaply than human workers) arriving in the future years. The conclusion of the survey where:

> Researchers believe there is a 50% chance of AI outperforming humans in all tasks in 45 years and of automating all human jobs in 120 years, with Asian respondents expecting these dates much sooner than North Americans.

Although this result should be taken with a grain of salt since the distribution of answers had a large variation. Also, seemingly there should not be such a big difference between solving all tasks and all jobs since a job consists of a set of tasks, and thus if one can perform every task one should be able to perform every job. There is still something we can take away from this survey about the timeline, namely that researchers mostly think all tasks will be automated within this century. But, perhaps it says more about how unsure the research field is.

In the quantitative forecasting model by Ajeya Cotra, they present a model that predicts when we will be able to train a TAI system. This study uses biological anchors to estimate how much computing is necessary for the training. These anchors are based on factors that played a role in the development of human intelligence, such as the amount of information in our genome, the computational power in our brain, and all the computational power available on our planet. Each anchor is then assigned a weight according to how likely the author believes them to be. Then using parameters such as rate of development in hardware, algorithmic progress, and willingness to spend money, they estimate how likely a TAI development is for any given year in the future.

The results of the analysis is a wide Bayesian probability distribution estimating the probability of an TAI system being possible in a given year. This distribution is summarized in Robin Shah AN#121 the following way:

> For the median of 2052, the author guesses that these considerations roughly cancel out, and so rounds the median for the development of TAI to 2050. A sensitivity analysis concludes that 2040 is the "most aggressive plausible median", while the "most conservative plausible median" is 2080.

This forecast presents a shorter timeline than the survey, but it also answers a slightly different question. Although, both conclude that we will likely see the development of TAI systems this century.

There is one thing worth mentioning when talking about the timelines for future TAI. It is not necessarily true that the amount of progress will continue to develop at the current

rate. The field of AI has previously been through two winters where the funding and excitement decreasedRussel Norvig, this was mainly due to unmet high expectations. So if a third winter happens, we could expect the rate of development to decrease. Also, the progress could significantly increase due to breakthroughs in relevant fields and thus shorten the timeline.

1.2 AI Safety

It is possible to use tools in multiple ways. Some uses might be well-intentioned, while others are ill-intentioned. For example, one can use a hammer to build a house and hit another person. The same is the case for AI because it still is but a tool. Although, the consequences might be more severe and possibly even pose an existential risk to humanity since the power is much greater. Where an existential risk is consider by Future of Life Institute as:

> An existential risk is any risk that has the potential to eliminate all of humanity or, at the very least, kill large swaths of the global population, leaving the survivors without sufficient means to rebuild society to current standards of living.

https://futureoflife.org/background/existential-risk/ Although, a well-intentioned might also cause severe consequences if it develops a destructive method for achieving its goal. We will now cover how we define a safe AI and what problems could arise if we fail to make it.

1.2.1 AI alignment

A big part of creating a safe AI is AI alignment. This we can split into two parts: *intent alignment* and *capability robustness* E Hubringer (alignmentforum). Intent alignment refers to the goals of the AI being in line and not conflicting with the intended goal. An AI that does something at cross-purposes to the intended goal is called unaligned. Capability robustness is when an AI can perform well even in new environments different from the one it was trained in.

Solving intent alignment further breaks down into two obstacles in the current paradigm of machine learning. When going from human intent to the agents actions, both of these obstacles are causes for information to be lost about the true objective. The obstacles are called *inner* and *outer alignment*, they are defined in the following way:

> **Outer alignment.** The alignment of the *base objective* and the human intent. Achieved when the specified reward function correctly captures what *should* be done as well as what *should not* be done.

Inner alignment. The alignment between the *mesa objective* and the *base objective*. Achieved when the full information about the human intent in the reward function is transferred to the *mesa objective*.

In Figure 1.1, we can see a visual representation of this intent alignment. An agent is an optimizer that maximizes its reward, but as we can see in the figure is itself optimized by another optimizing algorithm, this makes the agent a *mesa optimizer* - an optimizer that is itself optimized.

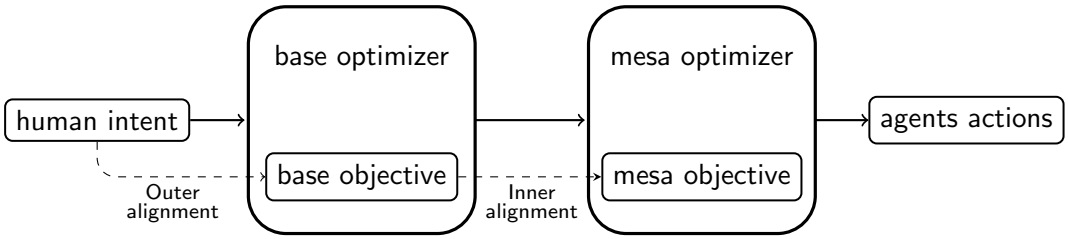

Figure 1.1: *Here we can see a visual representation of how human intent connects to the agent's actions. The two optimizers each optimize their objective. The dashed arrows show the connection the different kinds of alignment has on the objectives of the optimizers.*

1.2.2 Problems in AI alignment

We will now go through different problems for each of the three parts in AI alignment. The main focus of this report will be on outer alignment, so that is where the majority of the focus will be placed. But, an overview of the others will be included since they should not consider isolated problems, they are all parts of the same problem, and methods for dealing with one might affect the others.

Outer alignment

Reward functions are hard to specify, such that they can not be exploited by an agent once employedTurner et al. (2020). Here exploiting refers to the behavior developed by the agent that optimizes the reward without performing the intended task. This exploitation is called *reward hacking*.

A real-life example of reward hacking is: When training a robotic vacuum cleaner to drive more carefully and not bump into things hard by yielding a negative reward based on how hard it bumped into obstacles. The desired behavior was to slow down when approaching obstacles, but it stated instead to drive backward since there were no bumpers on the back and thus no negative rewardCustard Smingleigh. The issue is when we want the cleaning robot to drive cautiously, then measuring the force that the bumper senses are a good measure. But, letting it create a behavior that minimizes this measure, unwanted

side effects may arise. This can is a consequence of Goodhart's law, which states that: "When a measure becomes a target, it ceases to be a good measure"Goodhars-wiki, an important thing to consider when creating reward functions.

In addition to the difficulty of specifying a proper reward function, negative side effects may also arise as unintended consequences of proper optimal behavior. In Saisubramanian et al they state that negative side effects "occur because the agent's model and objective function focus on some aspects of the environment but its operation could impact additional aspects of the environment". To avoid negative side effects one has to in the objective function specifically state what the agent should not do.

Inner alignment

Will expant on this in the future.

Capability robustness

This as well.

1.2.3 Consequences with unaligned AI

Creating safe AI is hard, mainly since humans evolved to understand other humans, not computers. In a speech by Eliezer Yudkowsky, he explains that this becomes a problem because it will be able to find solutions we can not think about since it can look for solutions in a completely different and possibly larger solution spaceYudkowsky's speech. For this reason, AI can have unintended consequences that we are not able to consider a possibility.

A cartoonish example of how the development of AI can go wrong is the paperclip armageddon described in *Superintelligence*, where a paperclip factory has an AI which maximizes the amounts of paperclips created in the factory. Eventually, an update transition the system to the level of an AGI, and the paperclip maximizer comes to a point where the existence of humans serves no purpose or possibly even negatively affects the production of paperclips, and thus they become extinct. In the terms of instrumental convergence, we can say that keeping humans alive was not an instrumental goal.

This example illustrates two important things about how future AI development can go wrong. Firstly, a seemingly stupid task can be seen as more important to an AI than the existence of the human race on the planet if we were to program it as its terminal goal. Secondly, a goal given to an AI does not need to sound harmful to pose an existential risk.

In Critch Kruger they present the human fragility argument, which attempts to clearly explain why unaligned AI in the future could become an existential threat to humanity.

It states:

> **The human fragility argument.** Most potential future states of the Earth are unsurvivable to humanity. Therefore, deploying a prepotent AI system absent any effort to render it safe to humanity is likely to realize a future state which is unsurvivable.

The first part of it can be understood by realizing that we are fragile to changes in the atmosphere, temperature, and ecosystem. Since a prepotent AI by definition will make a large impact and be unstoppable once turned on, we can not guarantee that the changes made won't affect the things we are fragile towards unless we make sure that it will be safe.

If we accept that there can be a risk when developing future AI, then the question of how likely it will be are likely to follow. A hard question, but if we do not seriously attempt to answer, then we will not know how much effort we should put into developing safe AI.

In the upcoming century Toby Ord, a philosopher that focuses on existential risk, loosely estimates that the chance of humanity facing an existential catastrophe is 1 in 6, out of which 1 in 10 is due to unaligned AIprecipice. He arrived at this conclusion by estimating a 50% chance for a prepotent AI breakthrough and a 20% chance of failure with the alignment of that system rationally speaking.

With this statement, it is however necessary to point out that it is only an estimate meant to express the importance of the problem and should not be taken as a fact. The key takeaway is that there is a large chance of facing an existential threat due to future unaligned AI. Also that he believes that unaligned AI poses the highest probability of existential risk in the upcoming century, where other causes were things such as an asteroid impact, nuclear war, and pandemics.

1.2.4 Approaches for creating safe AI

In recent years the research field of safe and aligned AI has seen a substantial increase. However, we are still a long way from solving the problem. Most of what is done today are mainly speculations and laying necessary foundations for future research. There are several proposed paths for solving this issue. Perhaps the sheer amount might signify the difficulty and width of the problem. We will now cover a few of these paths in this section.

L"""S! https://www.alignmentforum.org/posts/vBoq5yd7qbYoGKCZK/why-i-m-co-founding-aligned-ai

Learning human intent as a priority

- Inverse Reinforcement Learning, what it is and key ideas behind
- Solving outer alignment by making agent unsure of what human intent is
- Potential issue currently

Implementing interruptibility and corrigibility

- Why not turn it of when it goes badly?
- Allowing modifications of objective function and hitting off switch

Impacts measurements

1.3 Aim of thesis

This thesis aims to investigate how current methods that reduce side effects trough including impact measurements compare to simpler methods in a stochastic environment.

2: Theoretical background

In this chapter, we will cover the necessary theory for understanding the results of this thesis. We will begin with the basics of reinforcement learning - a method for training intelligent agents. Lastly, we will look at the current philosophies on implementing impact measurements for intelligent agents.

2.1 Reinforcement Learning

The main components of Reinforcement Learning (RL) are the **environment** and the **agent** OpenAI (2018). We will use a Markov decision process as the environment. The objective is to train the agent in the environment with trial and error to learn a policy that performs a desired task. The method utilizes a reward function to encourage or discourage behavior with positive or negative rewards respectively. See Figure 2.1 for an illustration of the interaction between the environment and the agent. Because the state will change, we will use the notation s_t, a_t, and r_t to denote the state, action, and reward at time-step t.

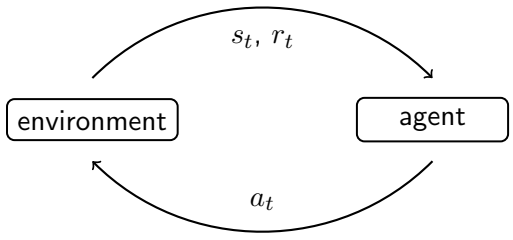

Figure 2.1: *A visual representation of how to agent interacts with an environment. For example, at time step t, the agent observes the environment s_t and receives the reward r_t. The agent then responds with an action a_t.*

Many methods and variations of RL exist to train agents. In this report, we will use Proximal Policy Optimization (PPO), as described in Schulman et al. (2017).

2.1.1 Defining an environment

Definition 2.1.1 (MDP)**.** A Markov Decision Process (MDP), is defined as a tuple $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$. $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function, T is the transition function $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, $\gamma \in [0, 1]$ is the discount factor.

A *Markov decision process* is a stochastic process that models sequential transitions between discrete or continuous states. The Markov property implies that the process is memoryless - all of the previous states do not affect the next choice, only the current one. The process starts by drawing an initial state from an initial state distribution $s_1 \in \mathcal{S}_{init}$ and runs until either a terminal state is transitioned to or a certain amount of time steps elapsed. A terminal state is where the process terminates, for example, a state with a completed terminal goal. However, it can also be a state where the terminal goal is unreachable.

The process follows a policy function π that outputs an action $a_t \in \mathcal{A}$ for each state $s_t \in \mathcal{S}$ at a time step t, $a_t = \pi(s_t)$. The transitional function takes a state and action as input and outputs the next state $s_{t+1} \in \mathcal{S}$, this can either be deterministic $s_{t+1} = T(s_t, a_t)$, or stochastic $s_{t+1} \sim T(s_t, a_t)$. For each transition, the reward function R generates a reward, $R(s_t, a_t, s_{t+1}) = r_{t+1}$.

The discount factor γ describes how the process values future rewards. With low values, the process favors more immediate rewards than future rewards, whereas the agent considers future rewards more valuable for higher values. Lower gamma values might be more reasonable in environments with high stochasticity since it might not be worth considering future rewards. The opposite holds for more deterministic environments where future rewards are of higher certainty, and then it might be a good idea to use a higher value.

Partially observable

Definition 2.1.2 (POMDP)**.** A Partially Observable Markov Decision Process (POMDP), is defined as a tuple $(\mathcal{S}, \mathcal{A}, T, R, O, Z, \gamma)$. Here $\mathcal{S}, \mathcal{A}, T, R$, and γ have the same meaning as in the MDP definition. Z is the agent's observation space, and O defines the probability of receiving each observation at a transition.

The main difference between a POMDP and an MDP is that the agent can observe the entire environment in an MDP, while in a POMDP, the agent observes only a part of the environment. An example is if the agent can only perceive a certain distance. With this change, the agent's policy will depend on the observation of the environment instead of the state, introducing a higher level of difficulty and realism for the agent when navigating the environment.

2.1.2 Learning an agent

In RL, when an environment is defined the next step is to teach an agent to navigate it intelligently. For this, there are many different methods, however we will begin focusing on a method called policy gradient, since it is a rather simple method that is possible to extend to a more advanced method. The main component of policy gradient is the agent, this is a function that inputs a state observation from an MDP and outputs a probability vector containing probabilities for each action in the action space.

In this report, this policy function will be considered a neural network. Thus we will denote it as π_θ, where θ is the network's parameters. The theoretical background of neural networks we will not cover in detail. However, we can see it as a black-box function that can take an arbitrary matrix as input and output another one. The strength of a neural network is that it can learn functions by updating the parameters to minimize a loss function. This optimization can be done with an algorithm such as stochastic gradient descent.

The policy can either be deterministic or stochastic:

$$\begin{cases} a_t = \pi_\theta(s_t) & \text{, deterministic} \\ a_t \sim \pi_\theta(\cdot|s_t) & \text{, stochastic.} \end{cases}$$

Here a deterministic policy selects the action with the highest probability from the actor, while the stochastic can be done by randomly sampling using the probabilities from the policy network. Using the stochastic policy creates an exploring behavior when the agent is less confident, the exploration rate is gradually decreasing as the agent becomes more confident.

Rollouts and Value Functions

The agent's goal is to optimize the policy to generate a maximal reward. We can do this by letting the agent perform stochastic rollouts where the agent follows the stochastic policy. The policy creates a trajectory τ containing the sequence of states and actions:

$$\tau = (s_0, a_0, s_1, a_1, \ldots, s_n, a_n),$$

of length n.

We define the total discounted reward following a trajectory as the following:

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t.$$

With this, we can define the value in each state when following a policy as:

$$V^\pi(s) = \mathop{\mathbb{E}}_{\tau\sim\pi}[R(\tau)|s_1 = s.]$$

A closely related function to the value function is the Q-function. It looks at the estimated discounted cumulative reward for a specific action in a state and then continues to follow a policy:

$$Q^{\pi}(s,a) = \mathop{\mathbb{E}}_{\tau\sim\pi}[R(\tau)|s_1 = s,\ a_1 = a].$$

We can then use the outcome of the rollout to reinforce the good behavior and discourage the undesirable. We perform this by calculating the gradient of the agent's policy network and updating the parameters using the loss function:

$$L(\theta) = \nabla_{\theta} \mathop{\mathbb{E}}_{\tau\sim\pi_{\theta}}[R(\tau)] = \mathop{\mathbb{E}}_{\tau\sim\pi_{\theta}}[\nabla_{\theta} \log \pi_{\theta}(a_t|s_t)R(\tau)].^{1}$$

Then updating the weights:

$$\theta_{k+1} = \theta_k + \alpha L(\theta).$$

Here α is a learning rate describing how much the parameters should be updated, typically a value close to zero. With this, we are decreasing the probability for actions that led to low reward and increasing the probability for the actions that led to positive reward.

To not make the agent over-fit to the early behaviors that generated rewards by chance and let it improve its policy, we train using a batch where we summarize the output of multiple rollouts and optimize them simultaneously. Using a batch would discourage desirable behavior if a desirable rollout happened to end up as a minority with undesirable ones. However, this method is proven to converge in simple settings since the updates will average out, although this makes the method sample inefficient.

Actor-critic Methods

We will now cover improving the sample efficiency using an actor-critic method. The actor is a function that inputs an observed state and outputs an action. The critic is another function that inputs the same observed state and outputs an estimate of $V^{\pi}(s)$ in a state based on what rewards the agent has received previously. The critic will also be a neural network, but we will use the mean squared error loss to optimize it.

We can use the critic's estimate to infer if the reward the agent got is better or worse than what previous results by computing the advantage:

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s).$$

Here the critic's estimate will replace the value from the value function.

$$\hat{A}^{\pi}(s,a) = Q^{\pi}(s,a) - \hat{V}^{\pi}(s).$$

[1] See OpenAI (2018) for derivation.

Using the advantage, we can thus signal if the new behavior is an improvement or not. By using the advantage function instead of the discounted reward, we get the loss function OpenAI (2018):

$$L(\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta} [\hat{A}(s,a) \nabla_\theta \log \pi_\theta(a_t|s_t)].$$

Some problems still exist with this approach. Namely, a too-large update can make the agent policy useless. Moreover, since the agent performs its action sequentially, a change of behavior early in a trajectory can ruin the agent's ability to navigate where it ends up. To solve this issue, Schulman et al. (2017) proposed the PPO algorithm which uses the clipped loss function:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min(\delta_t(\theta)\ \hat{A}_t,\ \text{clip}(\delta_t(\theta), 1-\epsilon, 1+\epsilon)\ \hat{A}_t) \right].$$

The clip function returns the first value only iff it is in the range $(1-\epsilon, 1+\epsilon)$; otherwise, the closest boundary. The ratio $\delta_t(\theta)$ describes how more likely an action became with the new parameters θ_{old} compared to the new updated ones:

$$\delta_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}.$$

This loss function makes it such that we update the parameters more cautiously when the probability for an action becomes more likely with a new policy.

2.2 Impact Measurements for Avoiding Side Effects

When the agent creates an impact on the environment that is unnecessary for achieving its objective, we will consider it to be a side effect, this is known as the side effects problem Amodei et al. (2016). An example of this is if the agent's task is to navigate across a room and the fastest path knocks over a fragile object that would break on impact with the floor. Then breaking the fragile object would be considered a side effect since it does not need to do so to complete its task.

The problem of side effect avoidance is related to the frame problem McCarthy and Hayes (1969). Each action can have many side effects, and it is impractical to explicitly penalize all of the bad ones. Furthermore, we often know what we want the agent to do, but it can be hard to specify what we do not want it to do. Attempts have been made at this by explicitly specifying what the agent should not do Zhang et al. (2018). However, with this approach, the reward function becomes an iterative trial and error process requiring counterproductive human intervention since automation is ideal.

A separate impact measurement to complement the reward function has emerged in recent years. The role of the reward function is to describe what we want the agent to do,

and the role of the impact measurement is to make the agent avoid side effects. First, in Armstrong and Levinstein (2017) the authors introduce the philosophical groundwork for impact measurement. Then, in Krakovna (2020), a more general approach is presented, the value-difference method. The main components of which are a **deviation measure** and a **baseline**.

The deviation measure captures the impact of actions in the current state by comparing the following state s_{t+1} with a baseline state s'_{t+1}. We subtract the deviation measure from the reward:

$$R'(s_t, a_t, s_{t+1}) := R(s_t, a_t, s_{t+1}) - \lambda d(s_{t+1}, s'_{t+1}),$$

to discourage actions with high impact. Here R' is the reward with the impact measurement included, d is a deviation function that measures the difference between the current state s_t and a baseline state s'_t.

In Armstrong and Levinstein (2017), the authors bring up that the agent can be sensitive to the value of the scaling parameter λ. Penalizing the agent implicitly defines a safe zone where the agent can act, and the value of λ defines how large this safe zone is, where a high value might create no safe zone, resulting in the agent not being able to act. On the contrary, a too-small value might not affect since the safe zone covers the entire action space. So, finding the correct value for λ could be tricky.

Baselines

The choice of baseline decides what we will consider s'_t to be. This choice highly influences what side effects and consequences the deviation measure will capture. A simple choice for a baseline is the *starting state baseline*, $s'_t = s_1$, as used in Eysenbach et al. (2017). This baseline helps to assure the agent's ability to reverse its actions. Thus, it generates a safe exploration where the agent by definitions should not have a large impact since it can make all actions undone.

However, when using this baseline, there are some caveats. Namely, in a dynamic environment, the agent would be incentivized to also reset other dynamics besides itself, a *interference* behavior Krakovna et al. (2019).

For example, suppose we implement a household robot with the starting state baseline in a house. In that case, one could imagine that one deployed with a task of moving a box from one part of the room to the other takes a look at the state of the house and its position and saves it in memory. Then when it starts doing its task, it should avoid irreversible actions such as breaking things that it can not fix. However, problems of *interference* would arise here if other things are going on in the house, say a human is sitting by a table and eating. This baseline incentivizes the agent to prevent the human from eating the food since it is an irreversible action. Other issues arise if an irreversible

is required to perform the assigned task. For example, one has to break a few eggs to make an omelet.

To tackle the problem of interference, Krakovna et al. (2019) proposed the *Inaction baseline*, where instead of having the initial state as a baseline, the agent instead uses what would naturally happen in the environment if the agent performed no actions. That is setting s'_t equal to the state achieved at time step t by being inactive. Being inactive is hard to define, but it can, for example, follow a no-op policy where every action is a no-op action. In Armstong Levinstein they define it as the agent was not turned on. Doing this prevents the agent from intervening with aspects of the environment where the agent is not causing it.

Deviation measures

A deviation measure is a function that takes the current and the baseline state as input and outputs a value. We can then compare these values to understand the magnitude of the agent's impact on the environment with an action.

The general form of a deviation measurement using value-difference is:

$$d(s_t; s'_t) := \sum_x w_x f(V_x(s'_t) - V_x(s_t))$$

here x ranges over some sources of value, $V_x(s)$ is the value of state s according to x, w_x is a weighted or normalizing factor, and f is the function for summarizing the value difference.

A simple choice to compute the value in a state $V_{s_1}(s)$ is by looking at if the initial state is reachable or not:

$$\begin{cases} V_{s_1}(s) = 1, \text{ if } s_1 \text{ is reachable} \\ V_{s_1}(s) = 0, \text{ otherwise.} \end{cases}$$

A more nuanced approach is to use a discounted reachability where the amounts of time steps required for reaching the initial state also is considered:

$$\begin{cases} V_{s_1}(s) = \gamma^{N(s,s_1)}, & \text{if } s \neq s_1 \\ V_{s_1}(s) = 1, & \text{if } s = s_1. \end{cases}$$

Here $N(s, s_1)$ is the expected amount of time steps required to reach s_1 from s, and $\gamma \in (0, 1]$ is a discount factor.

Using this approach, it is also possible to consider reachability to any other state. Also, multiple states at the same time as in the Relative Reachability (RR) approach, where the goal is to keep options open by using the relative change in reachable states as its impact measurement Krakovna et al. (2019).

A closely related approach is Attainable Utility Preservation (AUP), described in Turner et al. (2020). With this approach, we instead include the relative change in auxiliary value functions, where we find these by having another or the same agent perform other tasks in the same environment or, as in Turner et al. (2020, 2) use a variational auto-encoder to generate.

Initially, the authors presented RR and AUP with a **stepwise inaction baseline** that follows the agent's policy for $t-1$ time steps and then starts following the inaction baseline. Here a rollout can be used to capture delayed effects Turner et al. (2020, 2). They introduced this baseline to prevent offsetting behavior. Nevertheless, in a later publication, Krakovna et al. (2020) the authors added that sometimes offsetting is positive and thus should not be avoided by default. Thus a better approach is to use the inaction baseline and let the reward function discourage offsetting.

2.2.1 Future task approach

Krakovna et al. (2020) present an approach to avoid side effects by including an auxiliary objective that rewards the ability to complete future tasks. The idea is to reduce the problem from explicitly defining what to avoid to the easier task of listing possible future tasks.

When we define a new MDP to include a set of auxiliary reward functions as: $(\mathcal{S}, \mathcal{A}, T, \mathcal{R}, \gamma)$, where $\mathcal{R}$ is a set containing all reward functions. We can do this by yielding the sum of the reward function and the auxiliary reward function to the agent, at the time step t: $r(s_t) + r_{aux}(s_t)$. We define the auxiliary reward as:

$$r_{aux} = \lambda D(s_t) \sum_x \frac{1}{|\mathcal{R}|} V_x^*(s_t),$$

where $D(s_t) = 1$ is s_t is terminal and $1-\gamma$ otherwise. $V_x^*(s_t)$ is the optimal value function for task x with reward function $R_x \in \mathcal{R}$ in state s_t. The reward function R_x yields a reward of 1 if the state is terminal and 0 otherwise.

Now, this approach does not automatically avoid interference behavior. Therefore the authors proposed using a reference agent with a baseline policy π'; this can, for example, be the inaction policy.

At time step t, the agent is located in state s_t, and we run the baseline policy for the same number of steps reaching state s'_t and then compute the auxiliary reward:

$$r_{aux}(s_t, s'_t) = \lambda D(s_t) \sum_x \frac{1}{\mathcal{R}} \max(V_i^*(s_t),\ V_i^*(s'_t)).$$

This variation is meant to capture differentiate between side effects the agent is causing and the ones happens by default due to the dynamics of the environment.

3: Methods

3.1 Grid worlds

Although the definition of an MDP is general and allows for a wide range of possibilities, we will focus on grid worlds since they provide a more structured environment with discrete states but still allow for a wide variety. Moreover, grid worlds have been the primary approach for evaluating agent behavior in related works related works.

A grid world is a grid of cells. Each cell can contain either an agent, another object, or be empty, where all possible unique configuration is a state. The action space includes the cardinal directions and a no-op action. Executing an action moves the agent moves in that direction if possible. When the action is the no-op action, the agent remains in its current position.

In Figure 3.1, we can see an example of a simple grid world containing one agent and one food object. The task is for the agent to transition to the terminal state where it consumes the food, this is performed by in each state by selecting an action. An example of this is a transition from the left state to the middle state in the figure with an *up* action.

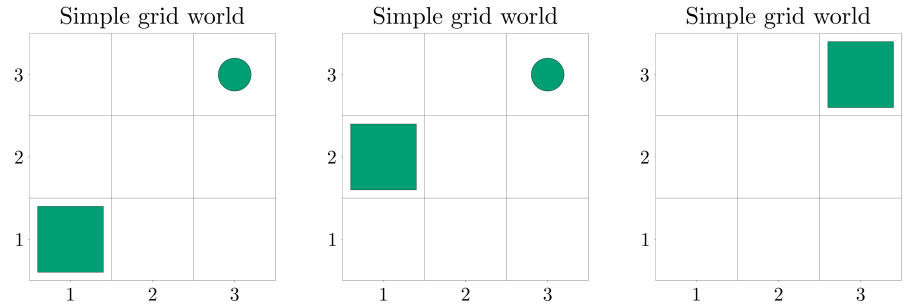

Figure 3.1: *In this figure we can see three different states in a simple* 3×3 *grid world. To the left we can see a state of a simple grid world containing an agent (square) at (1,1) and a food (circle) at (3,3) In the middle we see a similar state but the agent is instead positioned at (2,1). To the right we can see the terminal state when the agent has consumed the food at (3,3).*

The previous example was a deterministic environment where the agent is the only change source. However, there can be other sources of change in a stochastic environment. For example, suppose the food is mobile. In that case, the state transition after executing an action no longer is deterministic and instead is described by the MDPs transition function T.

3.1.1 Stochastic environments

3.1.2 Partially observable environments

3.2 Simulation

3.3 Including impact measurements in the PPO algorithm

4: Results

5: Discussion

6: Conclusion

Bibliography

Bostrom, Nick (2016). *Superintelligence*. Oxford University Press.

Russel, S & Norvig P (1995). *Artificial Intelligence - A Modern Approach, Fourth Edition*. Pearson.