A close up of a piece of paper

Description automatically generated

Master’s Studies

Field of studies: Advanced Analytics

Specialisation2: Big Data

Author’s Name and Surname: Oleksandr Romanchenko

Student No.: 83459

Deep Reinforcement Learning: training intelligent agent to play game “Flappy Bird” with evolution strategy algorithm

Master’s thesis

Written in the Department/Institute1

……………………………………………

Under scientific supervision of

……………………...……………………

Warsaw 2019

1. Apply as appropriate
2. Skip the line if there is no major or it has not been declared

Topic: Deep Reinforcement Learning: training intelligent agent to play “Flappy Bird” with evolution strategy algorithm

sources:

<https://towardsdatascience.com/introduction-to-machine-learning-for-beginners-eed6024fdb08>

<https://blog.soshace.com/deep-learning-vs-machine-learning-overview-comparison/>

**Introduction to Machine Learning**

**A close up of a logo

Description automatically generated**

We have seen Machine Learning as a buzzword for the past few years, the reason for this might be the high amount of data production by applications, the increase of computation power in the past few years and the development of better algorithms.

Machine Learning is used anywhere from automating mundane tasks to offering intelligent insights, industries in every sector try to benefit from it. We are already using devices that utilizes it. For example, a wearable fitness tracker like Fitbit, or an intelligent home assistant like Google Home. But there are much more examples of ML in use.

* Prediction — Machine learning can also be used in the prediction systems. Considering the loan example, to compute the probability of a fault, the system will need to classify the available data in groups.
* Image recognition — Machine learning can be used for face detection in an image as well. There is a separate category for each person in a database of several people.
* Speech Recognition — It is the translation of spoken words into the text. It is used in voice searches and more. Voice user interfaces include voice dialing, call routing, and appliance control. It can also be used a simple data entry and the preparation of structured documents.
* Medical diagnoses — ML is trained to recognize cancerous tissues.
* Financial industry and trading — companies use ML in fraud investigations and credit checks.

**A Quick History of Machine Learning**

A screenshot of a cell phone

Description automatically generated

Image: Linked In | Machine Learning vs Deep learning

It was in the 1940s when the first manually operated computer system, ENIAC (Electronic Numerical Integrator and Computer), was invented. At that time the word “computer” was being used as a name for a human with intensive numerical computation capabilities, so, ENIAC was called a numerical computing machine! Well, you may say it has nothing to do with learning?! WRONG, from the beginning the idea was to build a machine able to emulate human thinking and learning.

A group of people standing in front of a building

Description automatically generated

EIMC — Electronic Numerical Integrator and Computer | Image: www.computerhistory.org

In the 1950s, we see the first computer game program claiming to be able to beat the checkers world champion. This program helped checkers players a lot in improving their skills! Around the same time, Frank Rosenblatt invented the Perceptron, which was a very, very simple classifier but when it was combined in large numbers, in a network, it became a powerful monster. Well, the monster is relative to the time and in that time, it was a real breakthrough. Then we see several years of stagnation of the neural network field due to its difficulties in solving certain problems.

Thanks to statistics, machine learning became very famous in the 1990s. The intersection of computer science and statistics gave birth to probabilistic approaches in AI. This shifted the field further toward data-driven approaches. Having large-scale data available, scientists started to build intelligent systems that were able to analyze and learn from large amounts of data. As a highlight, IBM’s Deep Blue system beat the world champion of chess, the grand-master Garry Kasparov. Kasparov accused IBM of cheating, but this is a piece of history now and Deep Blue is resting peacefully in a museum.

**What is Machine Learning?**

A picture containing clipart

Description automatically generated

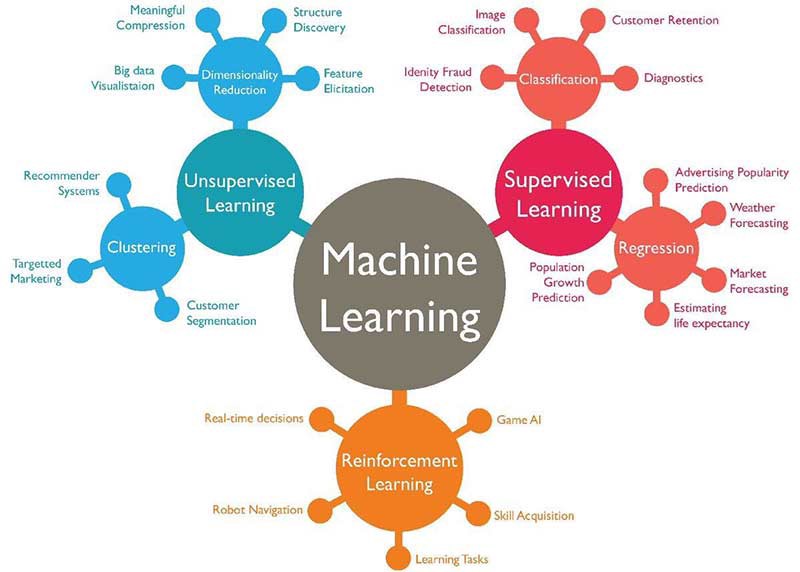
According to Arthur Samuel, Machine Learning algorithms enable the computers to learn from data, and even improve themselves, without being explicitly programmed.

Machine learning (ML) is a category of an algorithm that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available.

**Types of Machine Learning?**

Machine learning can be classified into 3 types of algorithms.

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning



3 Types of Learning

**Overview of Supervised Learning Algorithm**

In Supervised learning, an AI system is presented with data which is labeled, which means that each data tagged with the correct label.

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.

A screenshot of a cell phone

Description automatically generated

Example of Supervised Learning

As shown in the above example, we have initially taken some data and marked them as ‘Spam’ or ‘Not Spam’. This labeled data is used by the training supervised model, this data is used to train the model.

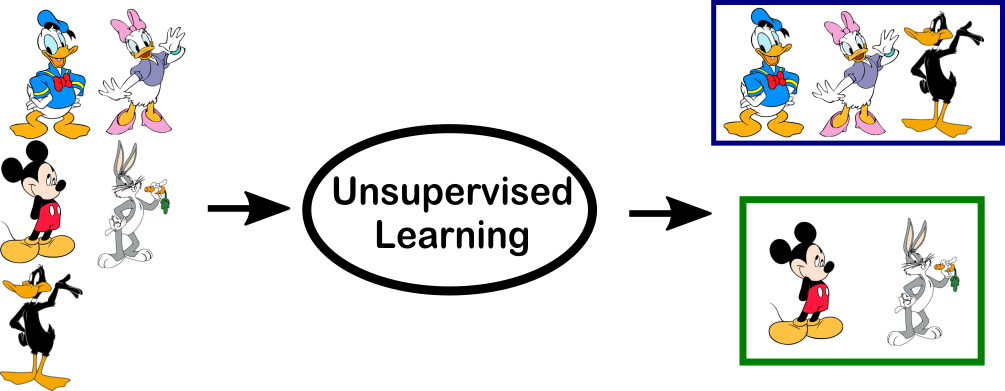
Once it is trained, we can test our model by testing it with some test new mails and checking of the model is able to predict the right output.

**Types of Supervised learning**

* **Classification**: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”.
* **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

**Overview of Unsupervised Learning Algorithm**

In unsupervised learning, an AI system is presented with unlabeled, uncategorized data and the system’s algorithms act on the data without prior training. The output is dependent upon the coded algorithms. Subjecting a system to unsupervised learning is one way of testing AI.



Example of Unsupervised Learning

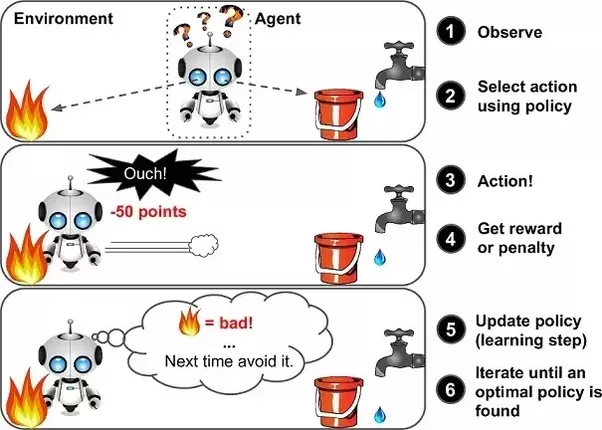
In the above example, we have given some characters to our model which are ‘Ducks’ and ‘Not Ducks’. In our training data, we don’t provide any label to the corresponding data. The unsupervised model is able to separate both the characters by looking at the type of data and models the underlying structure or distribution in the data in order to learn more about it.

**Types of Unsupervised learning**

* **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
* **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

**Overview of Reinforcement Learning**

A reinforcement learning algorithm, or agent, learns by interacting with its environment. The agent receives rewards by performing correctly and penalties for performing incorrectly. The agent learns without intervention from a human by maximizing its reward and minimizing its penalty. It is a type of dynamic programming that trains algorithms using a system of reward and punishment.



Example of Reinforcement Learning

In the above example, we can see that the agent is given 2 options i.e. a path with water or a path with fire. A reinforcement algorithm works on reward a system i.e. if the agent uses the fire path then the rewards are subtracted, and agent tries to learn that it should avoid the fire path. If it had chosen the water path or the safe path then some points would have been added to the reward points, the agent then would try to learn what path is safe and what path isn’t.

It is basically leveraging the rewards obtained, the agent improves its environment knowledge to select the next action.

## What is deep learning?

Deep learning, on the other hand, is a subset of machine learning, which is inspired by the information processing patterns found in the human brain. The brain deciphers the information, labels it, and assigns it into different categories. When confronted with new information, the brain compares it with the existing information and arrives at the conclusion that spurs future action based on this analysis. Deep learning is based on numerous layers of algorithms (artificial neural networks) each providing a different interpretation of the data that’s been fed to them.

### **How does deep learning work?**

Before we tackle the question of “how it works,” let’s briefly define a few other necessary terms.

Supervised learning is using labeled data sets that have inputs and expected outputs. Unsupervised learning is using data sets with no specified structure.

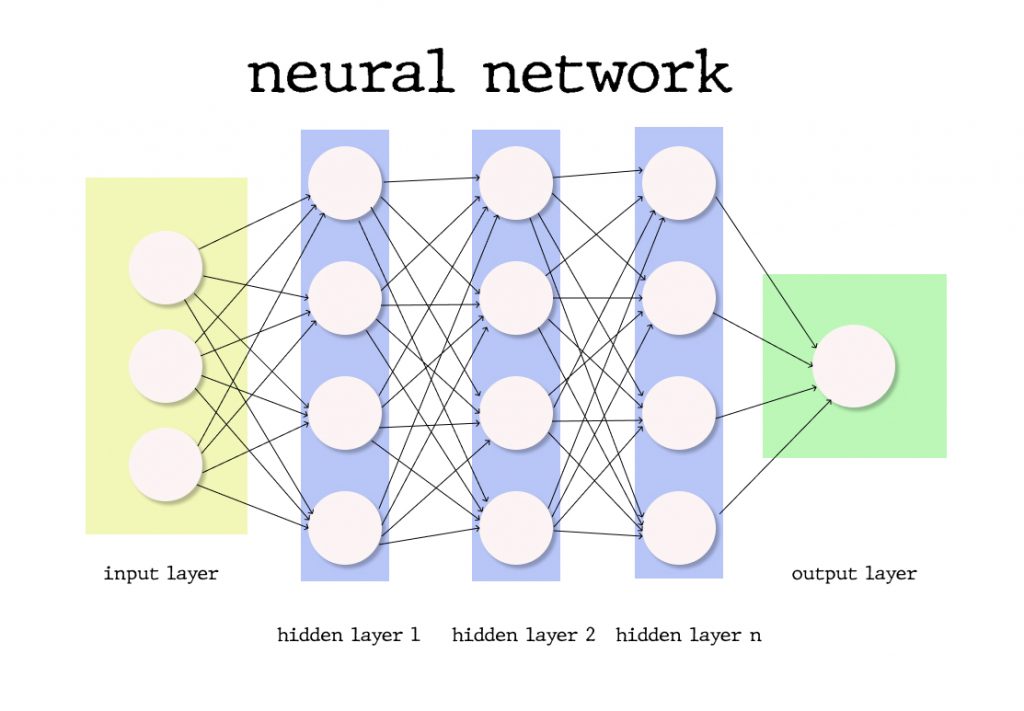
In the case of supervised learning, a user is expected to train the AI to make the right decision: the user gives the machine the input and the expected output, if the output of AI is wrong, it will readjust its calculations; the iterative process goes on until the AI makes no more mistakes. Among the popular supervised algorithms are linear regression, logistic regression, decision trees, support vector machines, and non-parametric models such as k-Nearest Neighbors. In the case of unsupervised learning, the user lets the AI make logical classifications from the data. Here, algorithms such as hierarchical clustering, k-Means, Gaussian mixture models attempt to learn meaningful structures in the data.

Deep Learning operates without strict rules as the ML algorithms should extract the trends and patterns from the vast sets of unstructured data after accomplishing the process of either supervised or unsupervised learning. To proceed further, we’ll need to define neural networks.

### **Deep Learning vs Neural Network**

The Deep Learning underlying algorithm is neural networks — the more layers, the deeper the network. A layer consists of computational nodes, “neurons,” every one of which connects to all of the neurons in the underlying layer. There are three types of layers:

* The input layer of nodes, which receive information and transfers it to the underlying nodes
* Hidden node layers are the ones which take all calculations
* Output node layer is a place for computational results



Neural Network

By adding more hidden layers into the network, the researchers enable more in-depth calculations; however, the more layers — the more computational power is needed to deploy such a network.

Each connection has its weight and importance, the initial values of which are assigned randomly or according to their perceived importance for the ML model training dataset creator. The activation function for every neuron evaluates the way the signal should be taken, and if the data analyzed differs from the expected, the weight values are configured anew, and the iteration begins. The difference between the yielded results and the expected is called the loss function, which we need to be as close to zero as possible. Gradient Descent is a function that describes how changing connection importance affects output accuracy. After each iteration, we adjust the weights of the nodes in small increments and find out the direction to reach the set minimum. After several of said iterations, the trained Deep Learning model is expected to produce relatively accurate results and can be deployed to production, however, some tweaking and adjustments can be necessary if the weight of the factors change over time.

### **Deep learning Learning Overview: summary of how DL works**

Deep Learning is one of the ways of implementing Machine Learning through artificial neural networks, algorithms that mimic the structure of the human brain. Basically, DL algorithms use multiple layers to progressively extract higher-level features from the raw input. In DL, each level learns to transform its input data into more abstract representation, more importantly, a deep learning process can learn which features to optimally place in which level on its own, without human interaction. DL is both applicable for supervised and unsupervised learning tasks, where for supervised tasks DL methods eliminate feature engineering and derive layered structures that remove redundancy in representation; DL structures that can be used in an unsupervised manner are deep belief networks and neural history compressors.

### **Deep Learning Applications**

Now, let’s look at some of the top applications of deep learning, which will help you better understand DL and how it works, besides some of those offer fantastic tutorials and source code detailing how to implement those algorithms.

The most well-known application of deep learning is a recommendation engine that’s supposed to enhance the user experience and provide a better service to its users. There are two types of recommendation engines: content-based and collaborative filtering. Until you have a sizable user-base, it’s best recommended to start with the content-based engine first.

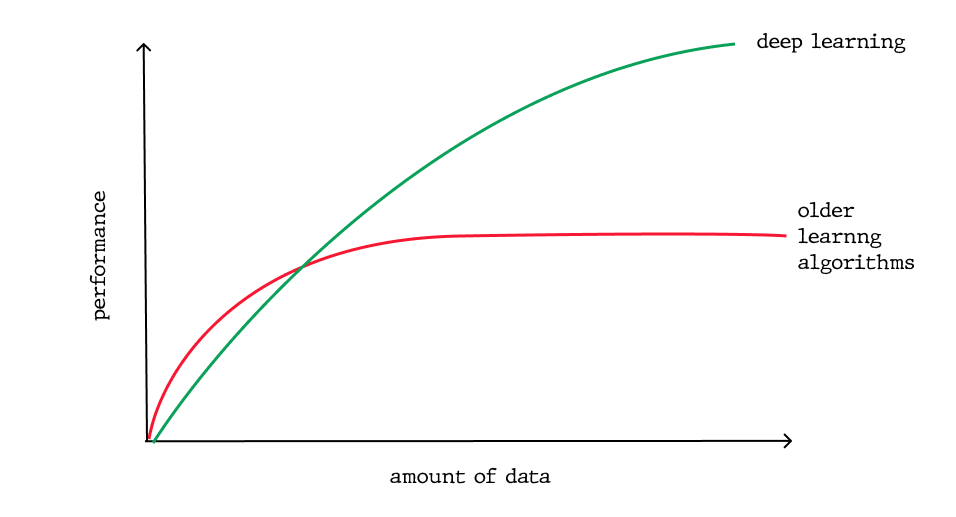
Natural Language Processing and Recurrent Neural Networks are used in the text to extract higher level information, also known as text sentiment analysis.

Another popular application is chatbots that can be trained with samples of dialogs and recurrent neural networks

Another popular application of DL models is image retrieval and classification using recognition models to sort images into different categories or using auto-encoders to retrieve images based on visual similarity.

## Machine Learning vs Deep Learning: comparison

One of the most important differences is in the scalability of deep learning versus older machine learning algorithms: when data is small, deep learning doesn’t perform well, but as the amount of data increases, deep learning skyrockets in understanding and performing on that data; conversely, traditional algorithms don’t depend on the amount of data as much.



Scaling with Amount of Data

Another important distinction which ensues directly from the first difference is the deep learning hardware dependency: Dl algorithms depend on high-end machines and GPUs, because they do a large amount of matrix multiplication operations, whereas older machine learning algorithms can work on low-end machines perfectly well.

In machine learning, most of the applied features need to be identified by a machine learning expert, who then hand-copies them as per domain and data type. The input values (or features) can be anything from pixel values, shapes, textures, etc. The performance of the older ML algorithm will thus depend largely on how well and accurately the features were inputted, identified, extracted. Deep Learning learns high-level features from data, this is a major shift from traditional ML since it reduces the task of developing new feature extractor for every problem, in turn, DL will learn low features in early layers of the neural network and then high-level as it goes deeper into the said network.

Again, because of the large amount of data that needs to be learned from, deep learning algorithms take quite a lot of time to train, sometimes as long as several weeks, comparatively, machine learning takes much less time to train to range from a second to a few hours. However, during the testing time, deep learning takes less time to run than an average machine learning algorithm.

Also, interpretability is a factor for comparison. With deep learning algorithms, sometimes it’s impossible to interpret the results, that’s exactly why some industries have had slow adoptions of DL. Nevertheless, DL models can still achieve high accuracy but at the cost of higher abstraction. To elaborate on this a little further, let’s get back to the weights in a neural network (NN), which essentially indicates a measure of how strong each connection is between each neuron. So by looking at the first layer, you can tell how strong is the connection between the inputs and the first layer’s neurons. But at the second level, you’ll lose the relationship, because the one-to-many relationship has turned into many-to-many relationships, exactly because of the high complexity of the NN nature: a neuron in one layer can be related to some other neurons which are far away from the first layer, deep into the network. Again, weights tell the story about the input, but that information is compressed after the application of the activation functions making it near impossible to decode. On the other hand, machine learning algorithms like decision trees give explicit rules as to why it chose what it chose and thus, they are easier to interpret.

**Theory behind Reinforcement Learning**

The idea that we learn by interacting with our environment is probably the first to occur to us when we think about the nature of learning. When an infant plays, waves its arms, or looks about, it has no explicit teacher, but it does have a direct sensorimotor connection to its environment. Exercising this connection produces a wealth of information about cause and effect, about the consequences of actions, and about what to do in order to achieve goals. Throughout our lives, such interactions are undoubtedly a major source of knowledge about our environment and ourselves. Whether we are learning to drive a car or to hold a conversation, we are acutely aware of how our environment responds to what we do, and we seek to influence what happens through our behaviour. Learning from interaction is a foundational idea underlying nearly all theories of learning and intelligence. Rather than directly theorizing about how people or animals learn, we explore idealized learning situations and evaluate the effectiveness of various learning methods. That is, adopting the perspective of an artificial intelligence researcher or engineer. We explore designs for machines that are effective in solving learning problems of scientific or economic interest, evaluating the designs through mathematical analysis or computational experiments. The approach we explore, called reinforcement learning, is much more focused on goal-directed learning from interaction than are other approaches to machine learning.

Reinforcement learning is like many topics with names ending in -ing, such as machine learning, planning, and mountaineering, in that it is simultaneously a problem, a class of solution methods that work well on the class of problems, and the field that studies these problems and their solution methods. Reinforcement learning problems involve learning what to do - how to map situations to actions so as to maximize a numerical reward signal. In

an essential way they are closed-loop problems because the learning system's actions influence its later inputs. Moreover, the learner is not told which actions to take, as in many forms of machine learning, but instead must discover which actions yield the most reward by trying them out. In the most interesting and challenging cases, actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. These

three characteristics being closed-loop in an essential way, not having direct instructions as to what actions to take, and where the consequences of actions, including reward signals, play out over extended time periods are the three most important distinguishing features of reinforcement learning problems. A full specification of reinforcement learning problems in terms of optimal control described with Markov decision processes which idea is simply to capture the most important aspects of the real problem facing a learning agent interacting with its environment to achieve a goal. Clearly, such an agent must be able to sense the state of the environment to some extent and must be able to take actions that affect the state. The agent also must have a goal or goals relating to the state of the environment. The formulation is intended to include just these three aspects: sensation, action, and goal in their simplest possible forms without trivializing any of them. Any method that is well suited to solving this kind of problem we consider to be a reinforcement learning method. Reinforcement learning is different from supervised learning, the kind of learning studied in most current research in field of machine learning. Supervised learning is learning from a training set of labelled examples provided by a knowledgeable external supervisor. Each example is a description of a situation together with a specification the label of the correct action the system should take to that situation, which is often to identify a category to which the situation belongs. The object of this kind of learning is for the system to extrapolate, or generalize, its responses so that it acts correctly in situations not present in the training set. This is an important kind of learning, but alone it is not adequate for learning from interaction. In interactive problems it is often impractical to obtain examples of desired behaviour that are both correct and representative of all the situations in which the agent has to act. In uncharted territory where one would expect learning to be most beneficial an agent must be able to learn from its own experience. Reinforcement learning is also different from what machine learning researchers call unsupervised learning, which is typically about finding structure hidden in collections of unlabelled data. The terms supervised learning and unsupervised learning appear to exhaustively classify machine learning paradigms, but they do not. Although one might be tempted to think of reinforcement learning as a kind of unsupervised learning because it does not rely on examples of correct behaviour, reinforcement learning is trying to maximize a reward signal instead of trying to find hidden structure. Uncovering structure in an agent's experience can certainly be useful in reinforcement learning, but by itself does not address the reinforcement learning agent's problem of maximizing a reward signal. We therefore consider reinforcement learning to be a third machine learning paradigm, alongside of supervised learning, unsupervised learning, and perhaps other paradigms as well. One of the challenges that arise in reinforcement learning, and not in other kinds of learning, is the trade-off between exploration and exploitation. To obtain a lot of reward, a reinforcement learning agent must prefer actions that it has tried in the past and found to be effective in producing reward. But to discover such actions, it has to try actions that it has not selected before. The agent has to exploit what it already knows in order to obtain reward, but it also has to explore in order to make better action selections in the future. The dilemma is that neither exploration nor exploitation can be pursued exclusively without failing at the task. The agent must try a variety of actions and progressively favor those that appear to be best. On a stochastic task, each action must be tried many times to gain a reliable estimate its expected reward. The exploration-exploitation dilemma has been intensively studied by mathematicians for many decades.

Another key feature of reinforcement learning is that it explicitly considers the whole problem of a goal-directed agent interacting with an uncertain environment. This is in contrast with many approaches that consider subproblems without addressing how they might \_t into a larger picture. For example, we have mentioned that much of machine learning research is concerned with supervised learning without explicitly specifying how such an ability would

finally be useful. Other researchers have developed theories of planning with general goals, but without considering planning's role in real-time decision-making, or the question of where the predictive models necessary for planning would come from. Although these approaches have yielded many useful results, their focus on isolated subproblems is a significant limitation. Reinforcement learning takes the opposite tack, starting with a complete, interactive, goal-seeking agent. All reinforcement learning agents have explicit goals, can sense aspects of their environments, and can choose actions to in influence their environments. Moreover, it is usually assumed from the beginning that the agent has to operate despite significant uncertainty about the environment it faces. When reinforcement learning involves planning, it has to address the interplay between planning and real-time action selection, as well as the question of how environment models are acquired and improved. When reinforcement learning involves supervised learning, it does so for specific reasons that determine which capabilities are critical and which are not. For learning

research to make progress, important subproblems have to be isolated and studied, but they should be subproblems that play clear roles in complete, interactive, goal-seeking agents, even if all the details of the complete agent cannot yet be filled in. One of the most exciting aspects of modern reinforcement learning is its substantive and fruitful interactions with other engineering and scientific disciplines. Reinforcement learning is part of a decades-long trend within artificial intelligence and machine learning toward greater integration with statistics,

optimization, and other mathematical subjects. For example, the ability of some reinforcement learning methods to learn with parameterized approximators addresses the classical \curse of dimensionality" in operations research and control theory. More distinctively, reinforcement learning has also interacted strongly with psychology and neuroscience, with substantial benefits going both ways. Of all the forms of machine learning, reinforcement learning

is the closest to the kind of learning that humans and other animals do, and many of the core algorithms of reinforcement learning were originally inspired by biological learning systems. And reinforcement learning has also given back, both through a psychological model of animal learning that better matches some of the empirical data, and through an influential model of parts of the brain's reward system. Finally, reinforcement learning is also part of a larger trend in artificial intelligence back toward simple general principles. Since the late 1960's, many artificial intelligence researchers presumed that there are no general principles to be discovered, that intelligence is instead due to the possession of vast numbers of special purpose tricks, procedures, and heuristics. It was sometimes said that if we could just get enough relevant facts into a machine, say one million, or one billion, then it would become intelligent. Methods based on general principles, such as search or learning, were characterized as “weak methods”, whereas those based on specific knowledge were called “strong methods." This view is still common today, but much less dominant. Modern AI now includes much research looking for general principles of learning, search, and decision-making, as well as trying to incorporate vast amounts of domain knowledge. It is not clear how far back the pendulum will swing, but reinforcement learning research is certainly part of the swing back toward simpler and fewer general principles of artificial intelligence.

**1.2 Examples**

A good way to understand reinforcement learning is to consider some of the examples and possible applications that have guided its development.

- A master chess player makes a move. The choice is informed both by planning/anticipating possible replies and counterreplies and by immediate, intuitive judgments of the desirability of particular positions and moves.

- An adaptive controller adjusts parameters of a petroleum refinery's operation in real time. The controller optimizes the yield/cost/quality trade-off on the basis of specified marginal costs without sticking strictly to the set points originally suggested by engineers.

- A gazelle calf struggles to its feet minutes after being born. Half an hour

later it is running at 20 miles per hour.

- A mobile robot decides whether it should enter a new room in search of more trash to collect or start trying to find its way back to its battery recharging station. It makes its decision based on the current charge level of its battery and how quickly and easily it has been able to find

the recharger in the past.

- Phil prepares his breakfast. Closely examined, even this apparently mundane activity reveals a complex web of conditional behaviour and interlocking goal-subgoal relationships: walking to the cupboard, opening it, selecting a cereal box, then reaching for, grasping, and retrieving the box. Other complex, tuned, interactive sequences of behaviour are required to obtain a bowl, spoon, and milk jug. Each step involves a series of eye movements to obtain information and to guide reaching and locomotion. Rapid judgments are continually made about how to carry the objects or whether it is better to ferry some of them to the dining table before obtaining others. Each step is guided by goals, such as grasping a spoon or getting to the refrigerator, and is in service of other goals, such as having the spoon to eat with once the cereal is prepared and ultimately obtaining nourishment. Whether he is aware of it or not, Phil is accessing information about the state of his body that determines his nutritional needs, level of hunger, and food preferences. These examples share features that are so basic that they are easy to overlook. All involve interaction between an active decision-making agent and its environment, within which the agent seeks to achieve a goal despite uncertainty about its environment. The agent's actions are permitted to affect the future state of the environment (e.g., the next chess position, the level of reservoirs of the refinery, the robot's next location and the future charge level of its battery), thereby affecting the options and opportunities available to the agent at later times. Correct choice requires taking into account indirect, delayed consequences of actions, and thus may require foresight or planning.

At the same time, in all these examples the effects of actions cannot be fully predicted; thus the agent must monitor its environment frequently and react appropriately. For example, Phil must watch the milk he pours into his cereal bowl to keep it from overflowing. All these examples involve goals that are explicit in the sense that the agent can judge progress toward its goal based on what it can sense directly. The chess player knows whether or not he wins, the refinery controller knows how much petroleum is being produced, the mobile robot knows when its batteries run down, and Phil knows whether or not he is enjoying his breakfast. Neither the agent nor its environment may coincide with what we normally think of as an agent and its environment. An agent is not necessarily an entire robot or organism, and its environment is not necessarily only what is outside of a robot or organism. The example robot's battery is part of the environment of its controlling agent, and Phil's degree of hunger and food preferences are features of the environment of his internal decision-making agent. The state of an agent's environment often includes information about the state of the machine or organism in which the agent resides, and this can include memories and even aspirations.

In all of these examples the agent can use its experience to improve its performance over time. The chess player refines the intuition he uses to evaluate positions, thereby improving his play; the gazelle calf improves the efficiency with which it can run; Phil learns to streamline making his breakfast. The knowledge the agent brings to the task at the start either from previous experience with related tasks or built into it by design or evolution - influences what is useful or easy to learn, but interaction with the environment is essential for adjusting behaviour to exploit specific features of the task.

**1.3 Elements of Reinforcement Learning**

Beyond the agent and the environment, one can identify four main subelements of a reinforcement learning system: a policy, a reward signal, a value function, and, optionally, a model of the environment. A policy defines the learning agent's way of behaving at a given time. Roughly speaking, a policy is a mapping from perceived states of the environment to actions to be taken when in those states. It corresponds to what in psychology would be called a set of stimulus-response rules or associations (provided that stimuli include those that can come from within the animal). In some cases, the policy may be a simple function or lookup table, whereas in others it may involve extensive computation such as a search process. The policy is the core of a reinforcement learning agent in the sense that it alone is sufficient to determine behaviour. In general, policies may be stochastic. A reward signal defines the goal in a reinforcement learning problem. On each time step, the environment sends to the reinforcement learning agent a single number, a reward. The agent's sole objective is to maximize the total reward it receives over the long run. The reward signal thus defines what are the good and bad events for the agent. In a biological system, we might think of rewards as analogous to the experiences of pleasure or pain. They are the immediate and defining features of the problem faced by the agent. The reward sent to the agent at any time depends on the agent's current action and the current state of the agent's environment. The agent cannot alter the process that does this. The only way the agent can influence the reward signal is through its actions, which can have a direct effect on reward, or an indirect effect through changing the environment's state. In our example above of Phil eating breakfast, the reinforcement learning agent directing his behaviour might receive different reward signals when he eats his breakfast depending on how hungry he is, what mood he is in, and other features of his of his body, which is part of his internal reinforcement learning agent's environment. The reward signal is the primary basis for altering the policy. If an action selected by the policy is followed by low reward, then the policy may be changed to select some other action in that situation in the future. In general, reward signals may be stochastic functions of the state of the environment and the actions taken. Whereas the reward signal indicates what is good in an immediate sense, a value function specifies what is good in the long run. Roughly speaking, the value of a state is the total amount of reward an agent can expect to accumulate over the future, starting from that state. Whereas rewards determine the

immediate, intrinsic desirability of environmental states, values indicate the long-term desirability of states after taking into account the states that are likely to follow, and the rewards available in those states. For example, a state might always yield a low immediate reward but still have a high value because it is regularly followed by other states that yield high rewards. Or the reverse could be true. To make a human analogy, rewards are somewhat like pleasure (if high) and pain (if low), whereas values correspond to a more refined and

farsighted judgment of how pleased or displeased we are that our environment is in a particular state. Expressed this way, it is clear that value functions formalize a basic and familiar idea. Rewards are in a sense primary, whereas values, as predictions of rewards,

are secondary. Without rewards there could be no values, and the only purpose of estimating values is to achieve more reward. Nevertheless, it is values with which we are most concerned when making and evaluating decisions. Action choices are made based on value judgments. We seek actions that bring about states of highest value, not highest reward, because these actions obtain the greatest amount of reward for us over the long run. In decision-making and

planning, the derived quantity called value is the one with which we are most concerned. Unfortunately, it is much harder to determine values than it is to determine rewards. Rewards are basically given directly by the environment, but values must be estimated and re-estimated from the sequences of observations an agent makes over its entire lifetime. In fact, the most important component of almost all reinforcement learning algorithms we consider is a

method for efficiently estimating values. The central role of value estimation is arguably the most important thing we have learned about reinforcement learning over the last few decades.

The fourth and final element of some reinforcement learning systems is a model of the environment. This is something that mimics the behaviour of the environment, or more generally, that allows inferences to be made about how the environment will behave. For example, given a state and action, the model might predict the resultant next state and next reward. Models are used for planning, by which we mean any way of deciding on a course of

action by considering possible future situations before they are actually experienced. Methods for solving reinforcement learning problems that use models and planning are called model-based methods, as opposed to simpler model-free methods that are explicitly trial-and-error learners viewed as almost the opposite of planning.

**1.4 Limitations and Scope**

Most of the reinforcement learning methods are structured around estimating value functions, but it is not strictly necessary to do this to solve reinforcement learning problems. For example, methods such as genetic algorithms, genetic programming, simulated annealing, and other optimization methods have been used to approach reinforcement learning problems

without ever appealing to value functions. These methods evaluate the lifetime behaviour of many non-learning agents, each using a different policy for interacting with its environment, and select those that are able to obtain the most reward. We call these evolutionary methods because their operation is analogous to the way biological evolution produces organisms with skilled behavior even when they do not learn during their individual lifetimes. If the

space of policies is su\_ciently small, or can be structured so that good policies are common or easy to \_nd|or if a lot of time is available for the search then evolutionary methods can be e\_ective. In addition, evolutionary methods have advantages on problems in which the learning agent cannot accurately sense the state of its environment. Our focus is on reinforcement learning methods that involve learning while interacting with the environment, which evolutionary methods do not do (unless they evolve learning algorithms, as in some of the approaches that have been studied). It is our belief that methods able to take advantage of the details of individual behavioural interactions can be much more efficient than evolutionary methods in many cases. Evolutionary methods ignore much of the useful structure of the reinforcement learning problem: they do not use the fact that the policy they are searching for is a function from states to actions; they do not notice which states an individual passes through during its lifetime, or which actions it selects. In some cases this information can be misleading (e.g., when states are misperceived), but more often it should

enable more efficient search. However, we do include some methods that, like evolutionary methods, do not appeal to value functions. These methods search in spaces of policies

defined by a collection of numerical parameters. They estimate the directions the parameters should be adjusted in order to most rapidly improve a policy's performance. Unlike evolutionary methods, however, they produce these estimates while the agent is interacting with its environment and so can take advantage of the details of individual behavioral interactions. Methods like this, called policy gradient methods, have proven useful in many problems, and some of the simplest reinforcement learning methods fall into this category. In

fact, some of these methods take advantage of value function estimates to improve their gradient estimates. Overall, the distinction between policy gradient methods and other methods we include as reinforcement learning methods is not sharply defined. Reinforcement learning's connection to optimization methods deserves some additional comment because it is a source of a common misunderstanding. When we say that a reinforcement learning agent's goal is to maximize a numerical reward signal, we of course are not insisting that the agent has to actually achieve the goal of maximum reward. Trying to maximize a quantity

does not mean that that quantity is ever maximized. The point is that a reinforcement

learning agent is always trying to increase the amount of reward it receives. Many factors can prevent it from achieving the maximum, even if one exists. In other words, optimization is not the same as optimality.

Evolution strategies:

* other methods use gradient descent to minimize a loss function, evolutionary methods take a biologically inspired approach instead (but has little to do with how evolution works. Similar to how neural networks have little to do with how brain works

A screenshot of a cell phone

Description automatically generated

A screenshot of text

Description automatically generated

What I’ve done:

Manually implemented neural network to be able to put inside it learning algorithm (evolution strategy)