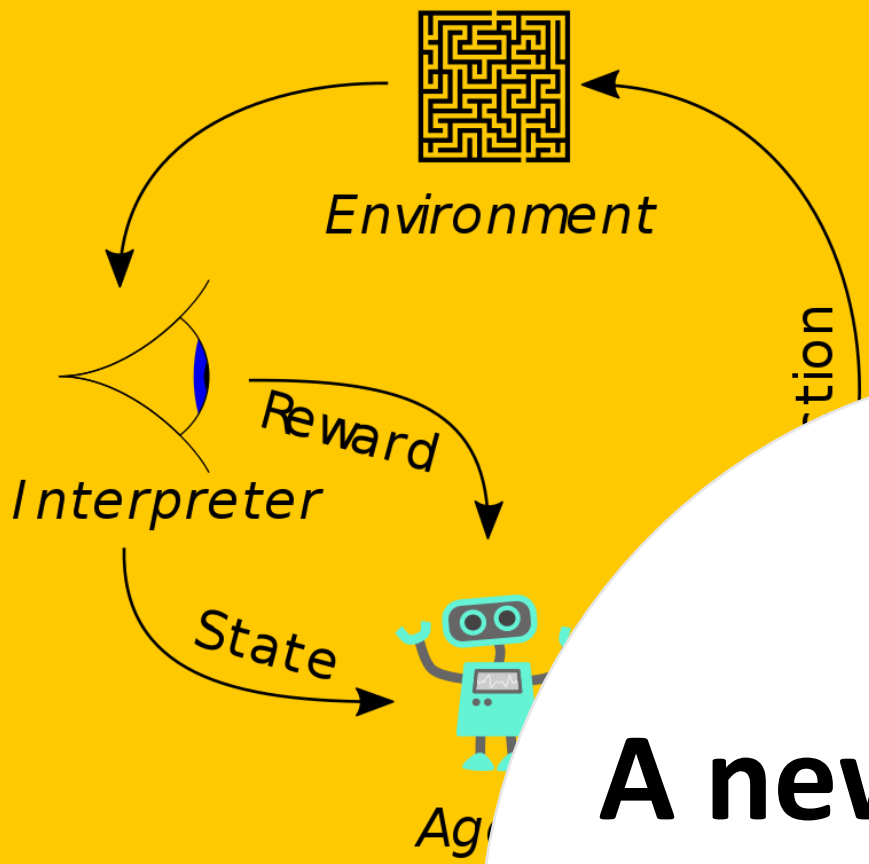
BeeBryte

Energy Intelligence & Automation

7 min



A new way to go! Reinforcement learning explained

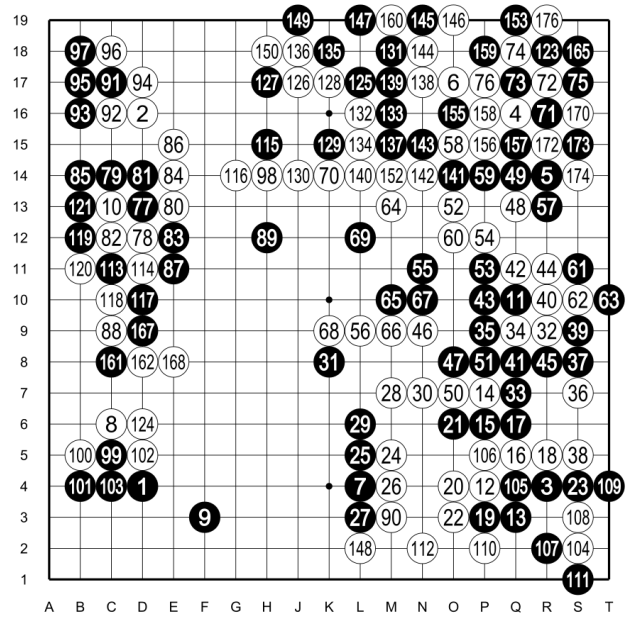
Simon CLOÎTRE, Yu Jia CHEONG, Patrick LEGUILLETTE

July 24th, 2019

Introduction to Reinforcement Learning

In 1997, one year after a first duel won by man, Deep Blue, the supercomputer developed by IBM, defeated the chess world champion Garry Kasparov. A first significant victory for artificial intelligence in such challenges, this event marked an important step in the development of the field.

Researchers have since tackled problems of growing complexity, including in particular another board game renowned for its almost infinite combinations of gameplay: Go. Indeed, the game counts 10^{170} possible legal configurations as compared to a chess game's paltry 10^{40}.



Lee Sedol (B) vs AlphaGo (W) - Game 3

122 at 113 151 at 73 154 at 72 163 at 145 164 at 73 166 at 160 169 at 145 171 at 160 175 at 71

In March 2016, Google DeepMind's engineers achieved a milestone comparable to that of their predecessors from IBM 20 years earlier, by beating Lee Sedol, one of the best players in the world, with their program AlphaGo. One year later, in October 2017, AlphaGo Zero was released, an evolved version of the first victor. That which is remarkable about AlphaGo Zero: the only information given it during its training process by its creators are the rules of Go!

After three days of training by itself and understanding which strategies are the most advantageous according to the final score, AlphaGo Zero faced its predecessor AlphaGo. Irrevocable victory: 100 games to 0 for the newcomer. Such exciting developments have solicited much talk about the potential of the burgeoning domain of machine learning used to train the AlphaGo variants: Reinforcement Learning.

Through this article, we will detail the different basic components of a Reinforcement Learning algorithm, as well as the main considerations of the domain, by applying it to one of BeeBryte's energy optimisation solutions.

REINFORCEMENT LEARNING DISSECTED

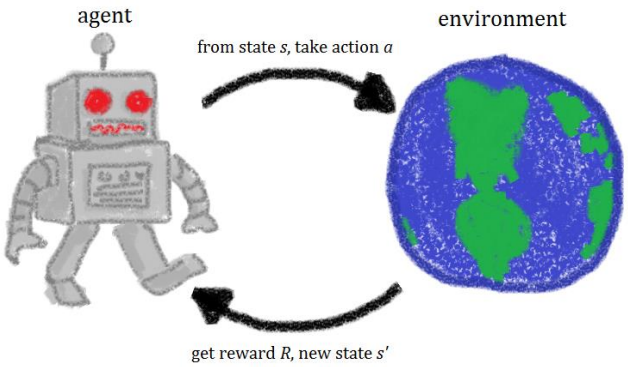
We will here consider a use case that BeeBryte encounters: choosing when to charge and discharge a battery with respect to a price signal in order to optimize the resulting electricity bill.

Two elements constitute the basis of every Reinforcement Learning algorithm. First, we have the "agent": in our example, a smart entity who will seek to determine the best moment to charge and discharge the battery. The agent acts in and changes an "environment" (here the battery and the electricity market) which constitutes the second component of the problem to be solved. At each step, the agent takes a new decision called an action (here the choice of charging, discharging or doing nothing), according to the observations it makes of its environment. We call "state" the information that the agent perceives

of the environment and that will help it choose its next action (here, the state would be the battery state of charge as well as the current and future electricity prices).

During the learning phase, the agent learns the characteristics of the environment by trying different actions and memorizing the relationship between states, actions and their consequences. Its end goal is to learn to optimize its future actions even when faced with previously unseen situations.

Concretely, the learning phase consists of a reward / penalty system that quantifies the quality of the actions taken, allowing the agent to adapt its behaviour for optimal performance. In our use case, it could for instance receive a reward proportional to the electricity price if it discharges the battery (thus consuming less from the grid), and a penalty proportional to the same price if it chooses instead to charge it (thus consuming more from the grid).



Mathematically, each (state, action) couple receives a score, a function of the rewards or penalties that ensue. However, for increasingly complex problems, a mere (state, action) matrix tracking the score associated to each couple is no longer sufficient. Here, the use of neural networks could enable the agent to generalize its understanding of new situations not encountered during the learning phase, on top of being better equipped to deal with high dimensional data.

Reinforcement Learning has the following advantage: once trained, the calculation required at each time step as new information is made available would consist of basic matrix multiplications, which is significantly less costly than standard optimisation regimes. The flip side of the coin is that a long training process is often necessary, and Reinforcement Learning solutions remain in general only approximations of truly optimal behaviour.

LONG-TERM VS SHORT-TERM

One of the problems that could limit a step-by-step learning is that a globally optimal solution is not necessarily the one that gives the best immediate reward. We have therefore to explore decisions that could on the short-term be “bad” in order to achieve potentially better rewards in the long-term that would offset all previously accumulated penalties.

Indeed, in our example, charging the battery is never the best short-term option as it always costs us more than discharging the battery or even doing nothing. Intuitively, we see however that choosing to charge when the price is low and discharge when the price is high would be more profitable than doing nothing.

To get around this issue, we include in the calculation of our score for each (state, action) couple not only the immediate reward that the agent would receive, but also all potential future benefits, whether they be good or bad. A main difficulty here would then be to firstly define the relative importance of future rewards as compared to immediate ones, and secondly to design a corresponding score function directing the agent towards the best global solution.

EXPLORATION VS EXPLOITATION

The fact that the best solution is not always the one that gives the best immediate reward implies that we should not only concentrate on good strategies once they are found, but explore other strategies beyond the first few time steps. We would all the same prefer to avoid the difficult and even impossible process of exploring all combinatorial possibilities, and to limit the time lost on actions with limited future benefit. This is what we call the exploration/exploitation dilemma.

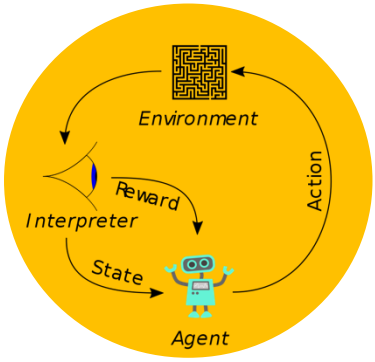
Exploitation refers to using only the results that we have already obtained. The agent will always choose the action with the highest score based on its current level of knowledge of its environment. Conversely, exploration allows the agent to expand its knowledge of its environment, at the expense of taking a few lower value actions. It is necessary to find a good compromise between these two opposing dynamics, as a strategy that purely exploits would cause stagnation in sub-optimal behaviour, while one that purely explores would never use the knowledge collected to become globally optimal.

The role of the data scientist is then to find the best protocol for the agent's action choice (called a policy) depending on the desired performance targets (learning speed vs benefits).

CONCLUSION

In this article, we provided quick overview of Reinforcement Learning, an up-and-coming domain of machine learning. In mentioning common pitfalls to avoid when using this approach, we should also keep in mind that there is no universal solution in AI. Each problem would require a customised application and adaptation of existing algorithms.

Inspired by human learning, this field is intuitive to understand and could offer an amazing efficiency, but is still expanding and proves often tricky to execute. Its potential has already been amply demonstrated (cf. AlphaGo Zero): we await its promise in enhancing the performance of our solutions.



A FEW WORDS ABOUT BEEBRYTE

Founded in 2015 by serial entrepreneurs Frédéric Crampé and Patrick Leguillette, BeeBryte is based in France & Singapore with a team of 25 professionals.

Our strategic investor is the largest French renewable energy producer CNR (Compagnie Nationale du Rhône). BeeBryte is also supported by French-based Bpifrance and ADEME. We were initially accelerated by INTEL, TechFounders, Greentech Verte and Novacité.

BeeBryte is leveraging artificial intelligence to get commercial buildings, factories, EV charging stations or entire eco-suburbs to consume electricity in a smarter, more efficient and cheaper way while reducing their carbon footprint!

Our software-as-a-service + IoT Gateway is minimizing utility bills with automatic control of heating-cooling equipment (e.g. HVAC), pumps, EV charging points and/or batteries. We even take into account any solar energy to maximize self-consumption.

Based on weather forecast, occupancy/usage and energy price signals, BeeBryte maintains processes & temperature within an operating range set by the customer and generates up to 40% savings. We charge a monthly fee (% of savings).

The technology combines a patented real-time optimization technique, proprietary trading algorithms, cloud computing and predictive analytics to offer dynamic energy services fully integrated to the Internet of Things.

contact@beebryte.com

www.twitter.com/BeeBryteGroup
www.linkedin.com/company/beebryte
www.beebryte.com