**Reinforcement Learning is Dead. Long Live the Transformer!**

Large Language Models are more powerful than you imagine

[[Austin Starks](https://medium.com/@austin-starks?source=post_page-----228835689841--------------------------------)](https://medium.com/@austin-starks?source=post_page-----228835689841--------------------------------)

[Austin Starks](https://medium.com/@austin-starks?source=post_page-----228835689841--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2Fc4a298b66f16&operation=register&redirect=https%3A%2F%2Fmedium.com%2F%40austin-starks%2Freinforcement-learning-is-dead-long-live-the-transformer-228835689841&user=Austin+Starks&userId=c4a298b66f16&source=post_page-c4a298b66f16----228835689841---------------------post_header-----------)

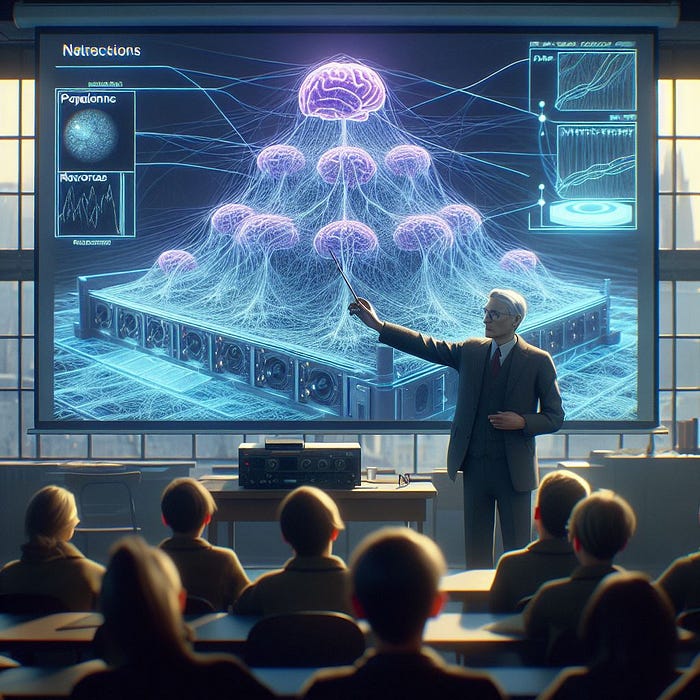
7 min read

·

Jan 14

520

13



“Artificial Intelligence” — by DALL-E

To preface this article, I want to say that I’m not biased. If you search online critically, you’ll find countless other [articles](https://towardsdatascience.com/why-you-shouldnt-use-reinforcement-learning-163bae193da8) and [experiences](https://www.reddit.com/r/reinforcementlearning/comments/sq0n72/im_gonna_have_to_say_it_but_my_experience_with_rl/) explaining how Reinforcement Learning simply does not work for real-world use-cases. The only people that are saying otherwise are course creators and academics within the field.

And I WANTED reinforcement learning to work. When I first heard of it 5 years ago, I was promised that it would revolutionize the world. An algorithm that can optimize ANYTHING with just a clever reward seemed like it could be applied ubiquitously, from designing medicines to advanced robotics. In 2016, when AlphaGo defeated Lee Sedol in Go, a famously complex game, this was supposed to be the turning point where RL would start dominating.

Yet, here we are, 8 years later and none of this materialized. Reinforcement Learning has accomplished nothing in the real-world. It dominates with toy problems and video games, but that’s it. The only notable advances of RL in the past 8 years is Reinforcement Learning with Human Feedback (RLHF), which is used to train Large Language Models like ChatGPT. And in my opinion, we won’t be using it for very long. Other algorithms simply do it better.

**[Mathematically Improve Your Trading Strategy: An In-Depth Guide](https://nexustrade.io/blog/mathematically-improve-your-trading-strategy-an-in-depth-guide-20240103?source=post_page-----228835689841--------------------------------" \t "_blank)**

[The Most Important Guide for All Traders in 2024. The Category 3 Trader, generated by DALL-E I used to be a Category 1…](https://nexustrade.io/blog/mathematically-improve-your-trading-strategy-an-in-depth-guide-20240103?source=post_page-----228835689841--------------------------------" \t "_blank)

[nexustrade.io](https://nexustrade.io/blog/mathematically-improve-your-trading-strategy-an-in-depth-guide-20240103?source=post_page-----228835689841--------------------------------" \t "_blank)

**What is Reinforcement Learning?**

Reinforcement Learning is a subfield of Machine Learning. With traditional supervised learning, we have a bunch of input examples and labels. We train the model to apply the correct label with the input example. We do this with 8 supercomputers and millions of training examples and we eventually get a model that can recognize images, generate text, and understand spoken language.

Reinforcement Learning, on the other hand, learns by a different approach. Typically with reinforcement learning, we don’t have labeled examples. But, we’re able to craft a “reward function” that tells us whether or not the model is doing what we want it to do. A reward function essentially punishes the model when it’s not doing what we want it to do, and rewards the model when it is.

This formulation seems amazing. Getting millions of labeled examples is extremely time-consuming and impractical for most problems. Now, thanks to RL, all we need to do is craft a reward function, and we can generate solutions to complex problems. But the reality is, it doesn’t work.

Interested in how AI applies to trading and investing? Check out my no-code automated investing platform [NexusTrade](https://nexustrade.io/)! It’s free and insanely powerful!

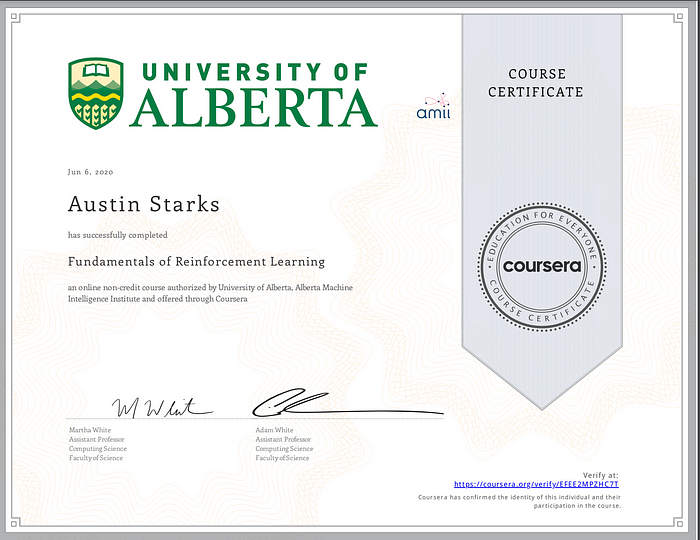
**[Mathematically Improve Your Trading Strategy: An In-Depth Guide](https://nexustrade.io/blog/mathematically-improve-your-trading-strategy-an-in-depth-guide-20240103?source=post_page-----228835689841--------------------------------" \t "_blank)**

[The Most Important Guide for All Traders in 2024. The Category 3 Trader, generated by DALL-E I used to be a Category 1…](https://nexustrade.io/blog/mathematically-improve-your-trading-strategy-an-in-depth-guide-20240103?source=post_page-----228835689841--------------------------------" \t "_blank)

[nexustrade.io](https://nexustrade.io/blog/mathematically-improve-your-trading-strategy-an-in-depth-guide-20240103?source=post_page-----228835689841--------------------------------" \t "_blank)

**My (Terrible) Experience with Reinforcement Learning**

I fell into the hype of reinforcement learning. It started with a course in Introduction to Artificial Intelligence at Cornell University. We talked briefly about RL and the types of problems it could solve, and I became very intrigued. I decided to take an online course from the University of Alberta to dive deeper into reinforcement learning. [I received a certificate, which demonstrates my expertise and knowledge in the field.](https://www.coursera.org/account/accomplishments/certificate/EFEE2MPZHC7T)



Course Certificate from the University of Alberta

After graduating from Cornell, I went to Carnegie Mellon to get my Masters in software engineering. I decided to take a notoriously difficult course, Intro to Deep Learning, because I was interested in the subject. It was here where I was able to apply reinforcement learning to a real-world problem.

In the course, we were given a final project. We had the freedom to implement any Deep Learning algorithm and write a paper on our experiences. I decided to apply my passion for finance with my interest in reinforcement learning and implement a Deep Reinforcement Learning algorithm for stock market prediction.

The project failed spectacularly. It was extremely hard to setup, and once I did, there was always something wrong. It was painful to debug problems because once everything compiled and ran, you don’t know what part of the system wasn’t working properly. It could be the actor network that learns the mapping of states to actions, the critic network that learns the “value” of these state-action pairs, the hyperparameters of the network, or just about anything else.

I’m not mad at RL because the project failed. If a group of graduate students could make a widely profitable stock-trading bot in a semester, that would upend the stock market. No, I’m mad at RL because it sucks. I’ll explain more in the next section.

For the source code for this project, check out the [following repository.](https://github.com/austin-starks/Deep-RL-Stocks) You can also read more technical details in the [paper here.](https://drive.google.com/file/d/1x67IaLpErVw9SwSBjWAdDtNEOcQSgje_/view) For more interesting insights on AI, [subscribe to Aurora’s Insights.](https://nexustrade.io/blog)

**[GitHub - austin-starks/Deep-RL-Stocks](https://github.com/austin-starks/Deep-RL-Stocks?source=post_page-----228835689841--------------------------------" \t "_blank)**

[Contribute to austin-starks/Deep-RL-Stocks development by creating an account on GitHub.](https://github.com/austin-starks/Deep-RL-Stocks?source=post_page-----228835689841--------------------------------" \t "_blank)

[github.com](https://github.com/austin-starks/Deep-RL-Stocks?source=post_page-----228835689841--------------------------------" \t "_blank)

**Why does Reinforcement Learning suck?**

Reinforcement Learning has a plethora of problems with it that makes it unusable for real-world situations. To start, it is **EXTREMELY** complicated. While traditional reinforcement learning makes a little bit of sense, deep reinforcement learning makes absolutely none.

As a reminder, I went to an Ivy League school. Most of my friends and acquaintances would say I’m smart. And deep reinforcement learning makes me feel stupid. There’s just so much terminology involved, that unless you’re getting your PhD in it, you can’t possibly understand everything. There’s “actor networks”, “critic networks”, “policies”, “Q-values”, “clipped surrogate objective functions”, and other non-sensical terminology that requires a dictionary whenever you’re trying to do anything practical.

It’s complexity extends beyond difficult-to-understand terminology. Whenever you’re trying to setup RL for any problem more complicated than CartPool, it doesn’t work, and you have no idea why.

For example, when I did my project on using RL to predict the stock market, I tried several different architectures. I won’t go into the technical details [(check out this article if you want to hear that)](https://nexustrade.io/blog/even-a-failure-is-a-success--failing-to-create-a-reinforcement-learning-stock-trading-agent-20231211), but nothing I tried worked. In the literature, you can see that RL suffers from many problems, including being computationally expensive, having stability and convergence issues, and being sample inefficient, which is crazy considering it’s using deep learning, something that is well-known to handle high-dimensional large-scale problems. For my trading project specifically, the thing that affected the final results the most was the initialization seed for the neural network. That’s pathetic.

**[Even a Failure is a Success — (Failing to) Create a Reinforcement Learning Stock Trading Agent](https://nexustrade.io/blog/even-a-failure-is-a-success--failing-to-create-a-reinforcement-learning-stock-trading-agent-20231211?source=post_page-----228835689841--------------------------------" \t "_blank)**

[From Aurora’s Insights: The AI-Focused Finance Blog. Pssst, You! The original story is on Medium! Please read the…](https://nexustrade.io/blog/even-a-failure-is-a-success--failing-to-create-a-reinforcement-learning-stock-trading-agent-20231211?source=post_page-----228835689841--------------------------------" \t "_blank)

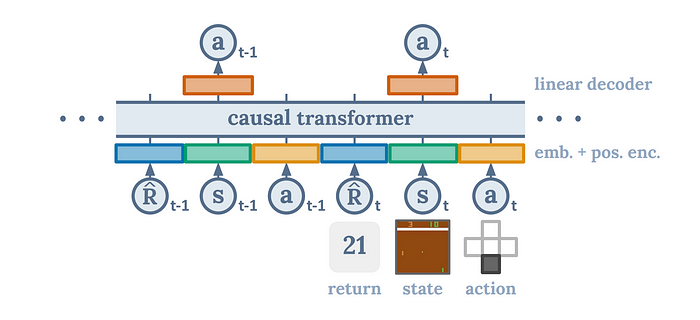
[nexustrade.io](https://nexustrade.io/blog/even-a-failure-is-a-success--failing-to-create-a-reinforcement-learning-stock-trading-agent-20231211?source=post_page-----228835689841--------------------------------" \t "_blank)

**Why are transformers going to replace RL algorithms?**

Transformers solve all of the problems with traditional RL algorithms. To start, it’s probably the easiest, most useful, AI algorithm. After you understand the [Attention Mechanism](https://www.youtube.com/watch?v=iDulhoQ2pro), you can start implementing transformers in Google Colab. And the best part is, it actually works.

We all know that transformers are useful when implementing models similar to ChatGPT. What most people don’t realize is that it can also be used as a replacement for traditional deep RL algorithms.

One of my favorite papers to ever come out was the [Decision Transformer](https://arxiv.org/pdf/2106.01345.pdf). I loved this paper so much that I emailed the authors of it. The Decision Transformer is a new way to think about reinforcement learning. Instead of doing complicated optimizations using multiple neural networks, sensitive hyper-parameters, and heuristics with no theoretical founding, we instead use an architecture that’s proven to work for many problems — the transformer.



The Architecture of the [“Decision Transformer”](https://arxiv.org/pdf/2106.01345.pdf)

We basically want to reframe reinforcement learning as a sequence-modeling problem. We still have states, actions, and rewards like in traditional RL; we just formulate the problem differently. We take our states, our actions, and our rewards, and lay it out in an auto-regressive manner. This leads to a very natural and efficient framework where the transformer’s ability to understand and predict sequences is leveraged to find optimal actions. By framing the problem in this way, the Decision Transformer can efficiently parse through the sequence of states, actions, and rewards, and intuitively anticipate the best course of action. This results in an algorithm that seamlessly uncovers the most effective strategies, elegantly bypassing the complexities and instabilities often encountered in traditional reinforcement learning methods.

For transparency, this is an offline algorithm, which means it can’t work in real-time. However, there is additional work being done [to enable Decision Transformers to be used in an online-manner](https://arxiv.org/pdf/2202.05607.pdf). Even the offline version of the algorithm is far better than traditional reinforcement learning. This paper shows that this architecture is more robust, especially in situations with sparse or distracting rewards. Moreover, this architecture is extremely simple, only requires one network, and matches or surpasses the state-of-the-art reinforcement learning baselines.

For more details on the Decision Transformer, either [check out the original paper](https://arxiv.org/pdf/2106.01345.pdf)or [this extremely helpful video by Yannic Kilcher.](https://www.youtube.com/watch?v=-buULmf7dec)

**[Prompt Engineering: The Definitive Step-By-Step How to Guide](https://nexustrade.io/blog/prompt-engineering-the-definitive-step-by-step-how-to-guide-20240107?source=post_page-----228835689841--------------------------------" \t "_blank)**

[Your No-Nonsense Manual for Mastering Prompt Engineering A Prompt Engineer, generated by DALL-E I used to not…](https://nexustrade.io/blog/prompt-engineering-the-definitive-step-by-step-how-to-guide-20240107?source=post_page-----228835689841--------------------------------" \t "_blank)

[nexustrade.io](https://nexustrade.io/blog/prompt-engineering-the-definitive-step-by-step-how-to-guide-20240107?source=post_page-----228835689841--------------------------------" \t "_blank)

**Conclusion**

Traditional reinforcement learning sucks. Unless the industry comes out with a new, stable, sample-efficient algorithm that doesn’t require a PhD to understand, then I will never change my mind. The decision transformer IS the new RL; it’s just not popularized yet. I’m looking forward to the day where researchers pick it up and use it for a variety of tasks, including Reinforcement Learning with Human Feedback.

Thank you for reading!