

**reddit** MACHINELEARNING 留言Want to join? [Log in](#) or [sign up](#) in seconds. | [中文](#)

This is an archived post. You won't be able to vote or comment.

0

Leveraging Google DeepMind software and Deep Learning to play the stock market

(self.MachineLearning)

submitted 11 月前 by [chaddjohnson](#)

I have no doubt some of you have seen the video of Google DeepMind learning to play Atari games. If you haven't, have a look:

<https://www.youtube.com/watch?v=V1eYniJ0Rnk>. This

leverages Deep Learning, a **Convolutional neural net**, and **reinforcement learning to learn to play merely from pixel data and the score as input**. Here

are details on running this simulation on your own machine:

<http://superintelligence.ch/deepmind/>. I've successfully run this on my machine for multiple Atari games.

I am modifying this software to treat the stock market like just another Atari game -- and learn to play it. I've analyzed the software and its dependencies, and I think I have a solid implementation strategy in mind. In short, I am creating a custom implementation of the **alewrap** interface and bypassing interaction with **ALE** and the Atari 2600 emulator (Stella).

In more detail, instead of the neural net analyzing frames

search

本文發表於 24 Jul 2015

0 指標 (50% upvoted)shortlink: <https://redd.it/3eeg6v>

使用者名稱

密碼

☐ 記住我 重設密碼

登入

**reddit anywhere**

Now on iPhone and Android

[discuss this ad on reddit](#)[發表新連結](#)[發表新文章](#)

MachineLearning

[訂閱](#) 66,354 讀者

~75 人在這裡

[Ask Simple Questions Here](#)

News, Research Papers, Videos, Lectures, Softwares and Discussions on:

- Machine Learning
- Data Mining
- Information Retrieval
- Predictive Statistics
- Learning Theory
- Search Engines
- Pattern Recognition
- Analytics

Not image posts, [no job posts](#)

AMAs:

[The MalariaSpot Team \(2/6/2016\)](#)[OpenAI Research Team \(1/9/2016\)](#)[Nando de Freitas \(12/26/2015\)](#)[Andrew Ng and Adam Coates \(4/15/2015\)](#)[Jürgen Schmidhuber \(3/4/2015\)](#)[Geoffrey Hinton \(11/10/2014\)](#)

from an Atari game, I am bypassing the Atari emulator and have it analyze stock charts instead, tick by tick. The charts will contain price and volume data, and each **frame** will contain a day chart, a week chart, and a month chart for a given stock symbol. The simulation agent will over time learn that pressing the button on the Atari joystick will trigger a **"buy"** action, and pressing it again (if a position is held) will trigger a **"sell"** action. The agent will then be punished when whenever a buy followed by a sell results in profit, and it will be punished when this resulted in loss. Further, a "game over" will be triggered whenever account balance drops below some threshold.

In addition to price and volume data, the charts will probably include **indicators** (rendered into the charts), as well.

I'm wondering 1) whether you all think this will be a worthwhile endeavor or a waste of time and 2) what flaws you see (besides trying to predict the stock market), and 3) improvements you can suggest.

32 留言 分享

所有 32 則留言

排序依據: 最佳

[-] **Bayes-Ian** 7 指標 11 月前

I've worked both in quantitative trading and at Deepmind, so I have quite a good idea about this. In short, some elements of machine learning are absolutely invaluable to traders (avoiding overfitting, estimating trends from data) but most of the research

[Michael Jordan \(9/10/2014\)](#)

[Yann LeCun \(5/15/2014\)](#)

[Yoshua Bengio \(2/27/2014\)](#)

Beginners:

Please have a look at [our FAQ and Link-Collection](#) [Metacademy](#) is a great resource which compiles lesson plans on popular machine learning topics.

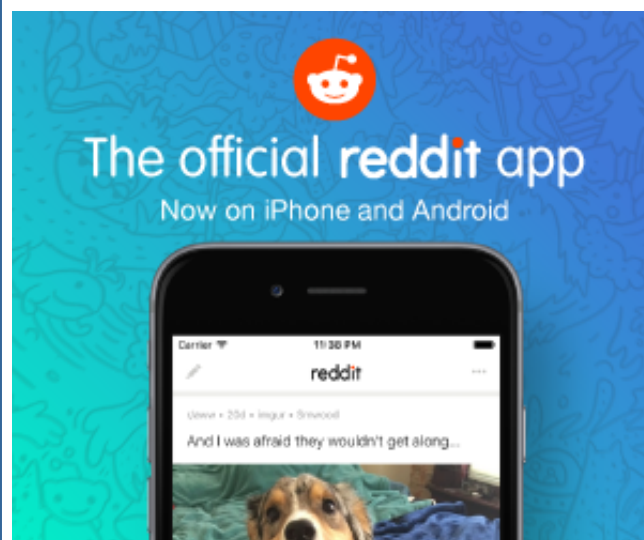
For Beginner questions please try [/r/MLQuestions](#) or <http://stackoverflow.com/>

Related Subreddit :

- [Statistics](#)
- [Computer Vision](#)
- [Compressive Sensing](#)
- [NLP](#)
- [/r/MLjobs](#) and [/r/BigDataJobs](#)
- [/r/datacleaning](#)
- [/r/scientificresearch](#)
- [/r/artificial](#) for general conversation about AI

created by [kunjaan](#)

社群已經成立 6 年



[discuss this ad on reddit](#)

版主

[寄送訊息給版主](#)

[kunjaan](#)
[kanak](#)
[cavedave](#) [naive](#)
[olaf_nij](#)
[BeatLeJuice](#)

[關於管理員團隊 >>](#)

behind the Atari player (DQN) are totally irrelevant to finding effective trading strategies.

Three elements immediately spring to mind of why this Atari example is fundamentally unlike trading in the stock market:

1 - The stock market is time-inhomogenous and stochastic. Atari is fixed and deterministic. 2 - The amount of data used to train the Atari agent would be equivalent to hundreds of thousands of years of stock returns. 3 - In the stock market you can observe the price changes of stocks you don't buy, unlike Atari where you must try an action in order to learn about it. 4 - In Atari (more or less) you control the environment, in the stock market most funds have negligible impact on the market as a whole.

The general problem of using Machine Learning to make good decisions is great. However, for the example of stock market trading you'd be better off using something else. Linear regressions with a whole bunch of cross-validation and regularization for example...

永久連結 **embed**

[\[-\] chaddjohnson](#) [S] 1 指標 11 月前

I appreciate this. This is far more helpful than negative comments. Thank you for taking the time to offer your expertise.

永久連結 **embed** 上層留言

[\[-\] svantana](#) 1 指標 11 月前

Well put! And let's not forget perhaps the most important difference: Atari games were made to be beatable (by humans), whereas there is no evidence that an agent, human or machine, can "beat" the stock market reliably -- unless using non-public information of course. For more on this I recommend the writings of stock trading billionaire Mark Cuban (in short: he made his fortune using knowledge about actual products and markets, and doesn't believe in any other way).

永久連結 **embed** 上層留言

[\[-\] simonhughes22](#) 6 指標 11 月前

IMO it might work, however treating it as a supervised learning algorithm using a deep neural network to predict the price or whether it will go up or down will work much better I strongly suspect. You could use an LSTM and train it on a sequence of price, volume, high and low data for a period of time. That will almost undoubtedly work much better. This isn't really a RL problem, as others have pointed out, and RL will be beat out by supervised learning if you have labelled data and it doesn't need to learn from interacting with the environment (as you have to do when playing a game or controlling

a robot).

I do think it would be really interesting to try for fun to see if it works, but if you are more interested in having it make money, I would suggest the supervised approach.

永久連結 **embed**

[\[-\] cesarsalgado](#) 4 指標 11 月前*

In reinforcement learning you should be able to make actions. How would you get the consequences of a action of your agent? You cannot find a consequences for every possible action in the historical data of the stock market. Will you do a simulation of the stock market? Or will you put the algorithm to play in real time in the real stock market? If you choose the last option your algorithm will take a very long time to learn (centuries or more). In the atari case, you could speedup the playing speed, because the game was a simulation.

What you could do with the stock market historical data is to train an LSTM to predict the next output of a sequence just like char-rnn. Or try to predict many symbols of a sequence, conditioned in the history, as in the sequence to sequence framework. The supervised learning strategy doesn't need your agent to interact with the environment and thus doesn't need a simulation. But the down side is that your agent won't be active.

I also don't agree with vesund answer saying that NN aren't good in this task. If we could make a simulation of the stock market accurate enough and that we could play it at very high speed, then I think a very deep NN would learn to play in the stock market better than humans.

永久連結 **embed**

[\[-\] chaddjohnson](#) [S] 0 指標 11 月前*

You cannot find a consequences for every possible action in the historical data of the stock market.

That's true. It won't be able to see the news, for instance. The network will really only be able to see price, volume, and indicators -- which is all I actually go on with a day trading method I am currently using (which is profitable 70% - 80% of the time).

Will you do a simulation of the stock market? Or will you put the algorithm to play in real time in the real stock market?

I plan to record tick-by-tick data for one stock for a few weeks and play it back for the network. Per some quick calculations, an entire day will be played back in roughly 5 minutes. Basically this will be a backtesting exercise.

And this would actually be an *unsupervised* learning exercise.

If we could make a simulation of the stock market accurate enough and that we could play it at very high speed, then I think a very deep NN would learn to predict better than humans.

Exactly. The playback will be very fast (one day in 5 minutes). Also, if the DeepMind Atari agent is able to learn to play an Atari game in 1 - 2 days at an expert human level consistently, then I figure, why could the same agent not *also* learn to play the stock market at an expert level, as well (as if it's just another Atari game)?

永久連結 **embed** 上層留言

[\[-\] cesarsalgado](#) 3 指標 11 月前*

Now that I thought better about the problem I came to the conclusion that it doesn't make sense to use Reinforcement Learning in this problem unless you are buying enough stocks to affect the prices in the market. This is because if you are not affecting the market with your actions, the NN will be simply learning implicitly to predict the future variables, as in the supervised learning case, but now using weaker reinforcement signals that just come after some time.

If you had an oracle that predicted the future wouldn't be easy to make an non-learning algorithm to be optimal in playing in the market? So why just don't teach the NN to predict the future? There is no need to teach it on when to buy and sell stocks using RL and actually it will be suboptimal because it will have to learn to predict the future by weaker and delayed supervised signals.

A side note: I just remembered this numberphile interview with James Harris Simons. He talks a bit about predicting prices. You might find interesting.

永久連結 [embed](#) 上層留言

[\[-\] chaddjohnson](#) [\[S\]](#) 1 指標 11 月前*

Now that I thought better about the problem I came to the conclusion that it doesn't make sense to use Reinforcement Learning in this problem unless you are buying enough stocks to affect the prices in the market. This is because if you are not affecting the market with your actions, the NN will be simply learning implicitly to predict the future variables, as in the supervised learning case, but now using weaker reinforcement signals that just come after some time.

I thought of the same problem (the agent's actions not affecting stock price).

In addition to pixel data, the agent *will* receive the "score" (profit/loss) as input and thus will be able to observe changes to that immediately following its actions. Would this possibly be enough?

永久連結 [embed](#) 上層留言

[\[-\] GoldmanBallSachs_](#) 1 指標 11 月前

How many securities will be included per frame? If it's less than 10,000, you're gonna have a bad time. Markets are global.

永久連結 [embed](#) 上層留言

[\[-\] chaddjohnson](#) [\[S\]](#) 1 指標 11 月前

Do you mean, how securities (e.g. AAPL, GOOG) will be included in each frame (where there is one chart image per security)? One.

I understand the DeepMind Atari agent receives raw pixel data...probably 30,720 per frame with the 160x192 Atari 2600 resolution. And I believe the Atari 2600 generates ~60 frames per second. So let's say the Agent receives a vector of 30,720 pixel data points with each frame.

With each frame, I'm thinking of providing three images concatenated into one (a day chart, a week chart, and a month chart). Let's say each image has a resolution of 800x150...so $800 * 150 * 3 = 360,000$ pixel data points with each frame.

Let's say a given stock I chose has, on average, around 25,000 ticks per day - that's roughly one tick per second during market hours.

So, if I provide 60 frames per second, then it should take around 7 minutes to process an entire day ($25,000 / 60 = 416.6$ seconds / = 6.94 minutes). Hopefully my math is correct...and hopefully my graphics card can process all this data :D

Does this answer your question? If not, would you mind clarifying?

永久連結 **embed** 上層留言

[–] **dnuffer** 6 指標 11 月前

Don't believe the other commentors saying that it wouldn't work, there are certainly many documented examples of using NNets to predict the stock market which do slightly better than random. There's books, or just peruse some of the many projects focused on trading from Stanfords CS229 Machine Learning course:

<http://cs229.stanford.edu/projects2014.html> (and earlier years as well) I wouldn't recommend using an image of a chart as input, it will be way too much data. Just use log-transformed and normalized prices and volume and the training will be much faster.

永久連結 **embed**

[–] **chaddjohnson** [S] 1 指標 11 月前

I wouldn't recommend using an image of a chart as input, it will be way too much data. Just use log-transformed and normalized prices and volume and the training will be much faster.

I'm new to neural networks, but I'll make a note of this.

The reason I want to use image data is because the DeepMind Atari agent seems to do exactly what I want, and Google has already optimized the software, so I figure re-purposing the software is the fastest route to a proof of concept considering my beginner-level machine learning knowledge.

Also, I am the proud owner of an NVIDIA GeForce Titan Z GPU (5760 CUDA cores!), so "too much data" is no problem :D

永久連結 **embed** 上層留言

[–] **cybrbeast** 3 指標 11 月前

If it is a viable approach, then in all likelihood the [algorithmic traders](#) would have already implemented it on their huge computing clusters. They are very secretive about their algos, so we don't know much about what the state-of-the-art is at the moment. But we know it's a huge market and it leeches some of the [brightest minds in technology and sciences](#), so you can be quite sure they have already beat you to any gains.

Also these traders have an ultra-low latency connection to the stock exchanges, something you will never have as a personal investor, so even if you somehow manage to make tiny gains they [will front-run you](#).

永久連結 **embed**

[–] **chaddjohnson** [S] 2 指標 11 月前

Also these traders have an ultra-low latency connection to the stock exchanges, something you will never have as a personal investor, so even if you somehow manage to make tiny gains they will front-run you.

That's true, though I've made plenty of gains trading manually. I may not be able to make super significant gains like they can; all I really care about is making roughly the same (or better) gains as I'm making now with manual trading (which has been

around 0.125% - 1.25% profit on average each day I trade).

永久連結 [embed](#) 上層留言

[\[-\] cybrbeast](#) 3 指標 11 月前

How long have you been trading? The current bull market which has [held strong for quite a few years now](#) makes it hard to know whether it's your strategy or just chance that makes your gains.

["A blindfolded monkey throwing darts at a newspaper's financial pages could select a portfolio that would do just as well as one carefully selected by experts."](#)

永久連結 [embed](#) 上層留言

[\[-\] chaddjohnson](#) [\[S\]](#) -1 指標 11 月前

Been day trading using this strategy for around 6 months now. It relies heavily on reversals.

永久連結 [embed](#) 上層留言

[\[-\] chaddjohnson](#) [\[S\]](#) 1 指標 11 月前

Why am I being down-voted?

永久連結 [embed](#)

[\[-\] Moseyic](#) 2 指標 11 月前

I brought you back up, seemed harsh

For the record, I don't think it will be very successful with that approach either. I agree that reinforcement learning isn't the way to go because the decisions made by your model won't be affecting the system in way that it can take note and update.

That being said, give it a shot and tell us what you learn!

永久連結 [embed](#) 上層留言

[\[-\] chaddjohnson](#) [\[S\]](#) 1 指標 11 月前

Thanks man. I appreciate it.

I'll give it a shot and let you know what I learn.

永久連結 [embed](#) 上層留言

[\[-\] polongus](#) 2 指標 11 月前

cause you're an optimistic idiot.

永久連結 [embed](#) 上層留言

[\[-\] chaddjohnson](#) [\[S\]](#) -1 指標 11 月前*

Thanks for projecting yourself.

At least I'm not afraid to ask questions and *try*. It's how we learn.

Someone you've probably never heard of once said, *"I have not failed 700 times. I have not failed once. I have succeeded in proving that those 700 ways will not work. When I have eliminated the ways that will not work, I will find the way that will work."*

永久連結 [embed](#) 上層留言

[\[-\] j1395010](#) 1 指標 11 月前

brute force is a stupid search policy. try education.

永久連結 [embed](#) 上層留言

[\[-\] chaddjohnson](#) [\[S\]](#) -2 指標 11 月前*

Look, I'm not taking a brute force approach. I've analyzed this, and as other people have said, it very well could produce some decent results. This is an educational exercise. I and many other people learn by doing, and it exposes us to new things which helps us branch out and learn about other things we need to explore. I've learned by doing for the last 15 years of my career, and it's served me quite well. I read blog articles, Stack Overflow posts, and books as needed, but some level of experimentation is needed.

This is a fun exercise, and it may or may not work -- it's just something fun to do. **Guys, please stop criticizing me and let's just have a nice discussion.**

Deep Learning is new technology, and there are many unknowns. Google surely was not sure whether their Atari agent would successfully learn to play games -- so they gave it a shot and tried.

There is an educational aspect, of course, but one cannot learn and get anywhere by merely schooling -- actually *doing* something is required. And if you have ever tried to entrepreneur something, you'd realize that reaching shooting high and failing is common, and it just takes time.

永久連結 [embed](#) 上層留言

[\[-\] j1395010](#) 0 指標 11 月前

Google surely was not sure whether their Atari agent would successfully learn to play games -- so they gave it a shot and tried. actually they were, because they acquihired the team after that work was done... further evidence of your lack of a clue.

永久連結 [embed](#) 上層留言

[\[-\] chaddjohnson](#) [\[S\]](#) 0 指標 11 月前

Wow dude, you just can't see past your own ego.

永久連結 [embed](#) 上層留言

[\[-\] j1395010](#) -3 指標 11 月前

enjoy losing money, dipshit.

永久連結 [embed](#) 上層留言

[\[-\] \[刪除\]](#) 11 月前*

[deleted]

[\[-\] ZigguratOfUr](#) 3 指標 11 月前

Wrong; neural nets can and have been applied fruitfully to the stock market (and neural nets are still bad at many tasks humans are good at).

A well-implemented neural net algorithm should be able to do pretty well at predicting 'will the price go up in the next second' or similar tasks.

HOWEVER, it should still only be used a signal. A bug in your implementation, or simply an error due to data in a different regime than the training set, could result in repeated and disastrously wrong predictions. If you don't have a risk-control layer, you're asking for trouble. Also, you can fruitfully combine with other signals.

BTW, Using Deepmind's software might be an okay start, but if it seems promising you can probably make it work much more efficiently by getting rid of a lot of Atari-specific stuff.

永久連結 [embed](#)

[\[-\] chaddjohnson](#) [\[S\]](#) 1 指標 11 月前

If you don't have a risk-control layer, you're asking for trouble.

Good call. I've made a note of this.

Using Deepmind's software might be an okay start, but if it seems promising you can probably make it work much more efficiently by getting rid of a lot of Atari-specific stuff.

That's true. I intend to leverage the DeepMind Atari agent in a proof of concept, and then I'll either gut the Atari-specific stuff, or I'll rewrite from scratch (re-using Torch and other libraries).

永久連結 [embed](#) 上層留言

[\[-\] ZigguratOfUr](#) 2 指標 11 月前

I just want to really emphasize that you should not expect ML to automate your strategy, just to augment it. Look up Knight Capital Group (company that went bankrupt over a single bug) and then compare the amount of security and developer talent they had to prevent that compared to yours. Good luck!

永久連結 [embed](#) 上層留言

[\[-\] chaddjohnson](#) [\[S\]](#) 0 指標 11 月前*

I actually have a reliable day trading method that is profitable 70% - 80% of the time. The problem is that it requires a human touch, and trading platforms like thinkorswim and NinjaTrader are too limited. It's is also time consuming and stressful to do it manually all day, so I'd like to automate the strategy.

My hope is that by providing the same charts and data I rely on to the DeepMind agent that the agent will learn to play as well as I am able to.

永久連結 [embed](#)

[\[-\] Sorcerysoft](#) 1 指標 11 月前

How is NinjaTrader limiting to you?

永久連結 [embed](#) 上層留言

[\[-\] master_dude](#) 1 指標 11 月前

look into concept drift which will be what you have to deal with

永久連結 [embed](#)

關於

網誌
[about](#)
原始碼
廣告
工作

說明

網站規定
常見問題
維基
[reddit](#) 站規
透明度

應用程式 & 工具

[Reddit for iPhone](#)
[Reddit for Android](#)
[mobile website](#)

<3

[reddit](#) 金幣
[reddit](#) 禮物

[聯絡我們](#)

[按鈕](#)

使用本網站即代表您接受我們的 [User Agreement {Genitive}](#) 和 [Privacy Policy \(updated\)](#). © 2016 reddit inc. 股份有限
公司 保留所有權利

REDDIT and the ALIEN Logo are registered trademarks of reddit inc.

[Advertise - technology](#)