Proceedings of CONF-SEML 2026 Symposium: Computational Analysis and Modeling in Complex Intelligent Systems
DOI: 10.54254/2755-2721/2026.GU32220

Boost High Frequency Trading with Deep Reinforcement Learning and Transformer

Yixuan Li

Springboard International Bilingual School, Beijing, China
nekojishiii@gmail.com

Abstract. High-frequency trading on the short-term base is extremely challenging as it seems to disobey the tenet of long-term value investment lauded by gurus like Warren Buffet. At the same time, it has become a crucial mechanism in modern financial markets for providing liquidity and enhancing market efficiency, which underscores the importance of understanding and developing effective trading algorithms. Motivated by the need to uncover effective approaches in such complex and volatile environments, this paper focuses on analyzing the potential of advanced machine learning techniques in short-term trading. So the paper would explore the most probable weapons that could efficiently help investors glean profits in the volatile and risky markets, the deep reinforcement learning, the transformer, deep residual networks. Readers might find the analysis "far-fetched" because most of the chosen work had no financial root. Rather, they came from fields like gaming or language synthesis. However, this is the magic of algorithms' generalizing ability: after proving its power in its "hometown", it could revolutionize another field. So if the ancient game Go and the modern data center could be treated as close relatives, then readers might find that the explorations are actually very relevant to the future of high-frequency trading. Ultimately, this work aims to provide a conceptual foundation for the application of cross-domain machine learning techniques in financial markets and to highlight promising directions for future research in high-frequency trading strategies.

Keywords: Machine learning, High-Frequency trading, Market noise

1. Introduction

A high frequency trader's main source of profit lies in capturing the bid-ask spread (BAS)in the limit order book (LOB) [1]. More specifically, a LOB lists the real-time interaction between buys and sells so it forms the metric called BAS, which gives the difference between the best bid and better offer. A high BAS usually means that the trading cost is heavy and the market volatility is high. Here paper could treat market as an environment in which the trader as the agent needs to collect reward amid various trade-offs and noise, and such treatment makes Deep Reinforcement learning (DRL) the most natural fit, because in this seminal paper by DeepMind where agent was trained to be a grandmaster of game Breakout, the paper found a lot of parallels, real-time control, reward-collecting, and error tolerance [2].

DOI: 10.54254/2755-2721/2026.GU32220

A trader's all actions could be boiled down to buy, sell and hold, and this could be instantly mirrored to the three actions of the game Breakout: to break all the bricks, player could move left or right and stay still to bounce the ball. The beauty lies in the synergy that reinforcement learning (RL) and deep learning(DL) complement each other's drawbacks: On one hand, the Bellman equation of RL could generate rolls of action-value pairs as the experience, iterating over them to pick the best one according to the current environment is impractical, so DL transforms this cumbersome iteration into a swift function approximation to deliver the best action-value pair. On the other hand, when DL is confined to the predefined rules, RL could explore new territories by epsilon greedy policy [2].

Now with a training mechanism towards the agent at hand, it is needed to focus on environment to find the best way of interpreting the context. This is where the attention mechanism steps in and help us make the most contextual predictions. Even a simple word "Queen" could have very different meaning in different settings: in music it could mean the legendary band while in mathematics it could mean the holy number theory, let alone the volatile market where one misinterpretation could cause catastrophic loss because market is extremely sensitive to the model parameters [3]. In the paper by Google Brain [4], transformer was invented to address this context problem: the attention mechanism would enrich the meaning of every token in the sequence by exploring their dependencies, and it is this dependency that makes the context very specific, thus incorporating every previous token to contribute to the prediction of the very next token and maximize its relevance in this way. Though in the original paper each token represents a word and the task is about language synthesis, this paper finds this core easily transferable to the trading scene as each decision could be regarded as a token, and to make the next decision most pertinent to the context, attention mechanism simply needs to analyze all the dependencies between each token.

In the stage of fine tuning, the paper notice that high frequency trading features extremely high dimensions including timestamps, price quotes, volumes, news sentiment and much more [5]. As neural networks are the integral parts of the model frameworks mentioned above, they are notoriously susceptible to the "curse of high dimension" which could make the gradient (out bread and butter of training) either explode of vanish. Now the residual network is the remedy because its core is about skipping the layers of network by retaining the original signal so that the trading, after many rounds of training, would not get drowned out by the evolution and forget its original flavor [6]. This paper explores whether advanced machine learning approaches—particularly deep reinforcement learning and deep neural architectures such as transformers and residual networks—can be effectively applied to high-frequency trading to model limit order book dynamics, handle extreme market noise, and generate stable short-term trading profits.

2. Technical details

2.1. Deep reinforcement learning

Some researchers have proposed that the unlikely marriage between RL and DL is fulfilled [2]. At the first glance the task seems very playful as shown in the following snapshot: it's necessary to train the agent to master the game Breakout.

As no human intervention is allowed and the agent must evolve on its own, the paper find the training mechanism is very demanding. The following Bellman equation is the heart of the algorithm:

$$q(s,a) = E\left[r + y^{*}maxq\left(s^{'},a^{'}\right)\middle|s,a\right] \tag{1}$$

DOI: 10.54254/2755-2721/2026.GU32220

Here r is the reward and Q holds the value of the action-state pair and it's easily spotted the recursive nature from the gamma factor of discount: the closer the update is, the less discounted the reward gets. And the max shows that people are always picking the optimal action according to the state. Despite of this equation's clarity, its severe bottleneck lies in two aspects. First, for every sequence, the value of each pair has to be evaluated in a separate manner, so this makes generalization impossible. Second, the equation is linear, so it is very hard to the capture the intricacies of the non-linear relationship in the action space. Understanding this bottleneck would make easily grasp why deep learning is the game changer.

Here this paper solves the problem of choosing the best action from a completely different angle: minimizing the loss function by constantly updating the weights of a single neural network via stochastic gradient descent. Here loss function means the gap between the expected and the observed reward, and the samples are directly taken from the emulator, so now the previous concern about the separation problem is gone. Put simply, it emulates, rather than solves, so this makes it computationally very expedient. Second, the non-linear nature of neural network lies in its sigmoid activation function so this makes the model capable of capturing much more nuanced non-linear relationships.

Still, people are facing another fundamental limit of reinforcement learning because each state strictly depends on its previous state, so this a Markov chain and correlation is a big headache. More specifically, as each state could be highly correlated to its neighbors, this could make the training very one-sided [2]:imagine a pianist too obsessed with training his left hand to train his right hand. In the scenario of high frequency trading, such one-sidedness could lead to either very poor local optimum (limited return), or catastrophic divergence (nonsense decision). Imagine again that poor pianist who might either only impress audiences with very limited repertoire, or overwork his left hand to make it break down). So to dilute such correlation, the mechanism of experience replay of all episodes is introduced: an episode records how an action could transform a current state into the next state and how much reward is generated. By drawing this experience as reference in a random manner, the problem of one-sidedness could be greatly alleviated

But this paper did not stop here, it further deploys the epsilon-greedy policy as an art of tradeoff. People know traders breathe with tradeoffs and this policy exactly introduced the possibility of risky adventure out of the comfort zone of experience, so whenever the model chooses an action, there is a probability of epsilon that it deviates from the cumulated experience and start a new venture.

2.2. Transformer

If the deep reinforcement learning enables the model to adapt, then transformer enables it to predict. The prerequisite of an accurate predictions in the cutting-throat scenario of high frequency trading is a precise understanding of the market context, namely its various microstructures [5]. The following section discusses Google brain paper where the transformer is born, here is the key equation [4]:

$$Attention\left(Q,K,V\right)=softmax\left(QK^{T}/\sqrt{dk}\right)V \tag{2}$$

As this paper treat a sentence as a sequence of tokens and the primary task is to predict the next token, so all the meaning of all the previous tokens must be distilled into the current one to give the most pertinent prediction. As context emerges out of a clear grasp of the interdependent relationship between each token, transformer uses the ingenious query matrix (Q) and key matrix (K) to fulfill this task. Transformer models represent each input token as a vector and use a self-attention

DOI: 10.54254/2755-2721/2026.GU32220

mechanism to capture contextual relationships between tokens. Specifically, query (Q), key (K), and value (V) vectors are computed for each token, and the attention score is obtained via the dot product between Q and K, followed by normalization. This mechanism allows the model to assign different importance to different tokens depending on context. In high-frequency trading, the same principle can be applied to sequential market data, where tokens correspond to order book states or market features, and attention weights capture the relative influence of past and present market conditions on trading decisions. However, this paper wants to compute the actual value as the weight, not the "value" bloated by dimensions. Here is a simple analogy: one dollar equals 158 yen, so here 158 is the dimension of yen. Therefore, in a unifying way, by the square root of dk as the denominator, this paper makes the variance of each token around 1.

Next the Softmax function turn each value into the range from 0 to 1, so this is the territory of probability because it's needed to the pick the most probable token, so probability gives us the most appropriate metric to pick the winner. Lastly, as the nature of Q*K product means a change, and it's needed to marry this change to the actual token itself, so here comes the value matrix (V).

After examining every key term of the attention equation, this paper find that its focus on exploring the global dependencies has a downside: it is needed to refine the local features of each token. More specifically, though this paper had already enriched each token's meaning by the query and key, it's needed each individual token's features to be distilled in a more salient way. So this is why the multi perceptron layer (MLP) is introduced right after the attention mechanism.

Now regarding the generalizing power of transformer, a token could not only be a word "queen" with layers of meaning, but also a investment decision influenced by various factors. Once this parallel is drawn, all the rest mechanism could be transferred to the high frequency trading so that the model could accurately grasp the context of the market.

2.3. Deep residual networks

Now this paper comes to the stage of fine tuning. Many facets could shape a great trading decision and behind one such facet (inter-market sweep order), there could be more facets (difference between stock exchanges, latency). Such an exponential growth of influencing factors makes the dimension of a decision ultra high, this means the processing neural network needs to be correspondingly very deep. However, according to Kaiming He, the depth of neural work means degradation [6]. As the essence of updating the weight of each neuron is about chain rule in calculus, so if the chain grows too long between the deeper layers, the exponential could make the less-than-one gradient too small as to vanish and the larger-than-one too large as to explode. This polarization problems have tormented the machine learning community for years because deeper layers mean finer features. However, the price of the vanishing/exploding gradients is becoming a important issue. In the following simple equation, Kaiming gives the answer [6].

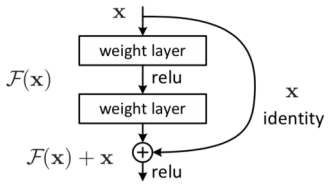

Figure 1. One building block of residual net [6]

DOI: 10.54254/2755-2721/2026.GU32220

As shown in figure 1, here x means the original identity and F(x) means the change, so through each block, the identity is mapped so that the long-pursued stabilization is fulfilled.

3. Application of cutting-edge technology

And now turn to the real-world test of the proposed algorithms. Zheng shows that when different market makers compete to maximize their own performance, the complex underlying Nash Equilibrium is very hard to capture and by reducing this equilibrium to a set of value functions belonging to each market makers, the learning process is much more simplified and the convergence to the equilibrium could be clearly quantized as a function of time increment [1]. To ensure the convergence, the authors carefully fine tune the hyper-parameters of the model and as the two of the most important hyper-parameters, the minimum exploration probability epsilon is tuned as 0.00005 and the learning rate w is tuned as 0.5. In the bigger picture, it is a discrete-time Markov Decision Process and the state is the mid-price and the action is the response from the market makers in the competition. The numerical result shows that as the time increment approaches 0.001 second, the optimal value function is guaranteed to converge. And after merely after 800 steps, the Reinforcement learning could capture the true policy behind the complex Nash equilibrium in the competition of high frequency. For example, the paper reports that in a simulated limit order book environment, the DRL agent was able to consistently outperform baseline strategies such as naive market making and zero-intelligence agents, the error between the learned policy and the true policy is shown to be negligible. And this discrete-time approach is readily applicable to many other decision-making scenarios (optimal execution) of the similar discrete-time nature.

Barez shows that transformer could incorporate a much longer window time as its trading context than its predecessors (long short term memory) does [7]. In one experimental case, the transformer model successfully identified subtle patterns across hundreds of past order book snapshots, enabling more precise timing of trade submissions compared to LSTM-based models. It also highlights the parallel nature of transformer that as each token is processed at the same time, it could make the latency very low and this is a huge advantage for high frequency trading. Another example from Barez demonstrates that the parallel computation allowed the model to generate trading signals within microseconds, a latency reduction that could be decisive in real-world HFT scenarios.

4. Discussion

4.1. Challenges

Though this paper already been shown the power of the proposed models and the industry recognition, there are still downsides to overcome. This paper has examined one of the most advanced reinforce learning paper and saw how it outsmarted its competition by inventing GRPO [8], which hugely cut the training cost by replacing the value function by group evaluation, and it also added KL divergence as a safety check to prevent too risky policies. yet, it still shows 51.7% on the competition-level MATH benchmark [8], and the paper believes there is still much space for progress here.

4.2. Future

Transformers are constantly evolving, and now one of the latest versions is called flash attention [9]. As past transformers have always suffered from the length of context, intense and ongoing efforts are focusing on extending this span. Unlike most research that focused on tuning the model, this

DOI: 10.54254/2755-2721/2026.GU32220

paper focused on the margins hidden in the low-level hardware: how to optimize the reads and writes to different levels of fast and slow memory in GPU [10]. By tracing the memory footprint, the author shows a stunning 20× more memory-efficient improvement than the current attention baselines.

5. Conclusion

In this article the paper proposes the deep reinforcement learning and transformer as the main frame work for modern high frequency trading. Though the former was born from gaming and the latter was invented for language generation, the paper see their strong generalizing power and very solid parallels that make them very expedient to transfer to the high frequency trading field. The paper chose these models because of their fitness: reinforcement learning deals with the reward from the interaction between agent and environment, transformers learn the contextual details to generate the most pertinent token, and to make the article more accessible to the non-academic readers, the paper made the analogies like "queen" in different context to make sense of dimensions and "pianist" to understand the one-sided danger inherent in the training. In each heart of the models lies a key equation and the paper have explained each term both in mathematical details and their meaning in the big picture. Interestingly, though the core mechanism and tasks of these two models are very different, each is greatly empowered by deep learning (multi perceptron layers). For reinforcement learning, the paper see how deep learning could quickly choose the best action-value pair instead of the brute-force iteration, and in transformers, the paper sees that after learning the dependencies between each token, how MLP could further distill the features of each individual token. Perhaps this common ground is a firm testament of the unifying power of gradient-based method, the paper still sees in Kaiming 's work that it also has flaws of vanishing/exploding gradients if layers are getting too deep. How could such a nagging problem get solved but such a simple method of identity mapping (just one addition operation) has become a problem that cannot be ignored. The paper see that it might be very difficult to come up with a very simple solution, maybe this is the mystery of the elegance hidden in the craft of machine learning.

References

[1] Zheng, Y. H. (2024) Reinforcement Learning in High-frequency Market Making. arXiv: 2407.21025.
[2] Volodymyr, M. etc. (2013) Playing Atari with Deep Reinforcement Learning arXiv: 1312.5602.
[3] Sergio, P. etc. (2024) Understanding the worst-kept secret of high-frequency trading. arXiv: 2307.15599.
[4] Ashish, V. etc. (2017) Attention Is All You Need. arXiv: 1706.03762.
[5] Gbenga, I. etc. (2024) Data-Driven Measures of High-Frequency Trading. arXiv: 2405.08101.
[6] He, K. M. (2015) Deep Residual Learning for Image Recognition. arXiv: 1512.03385.
[7] Fazl, B. etc. (2023) Exploring the Advantages of Transformers for High-Frequency Trading. arXiv: 2302.13850.
[8] Shao, Z. H., etc. (2024) DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. ArXiv: 2402.03300.
[9] Tri, D. etc. (2023) FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. arXiv: 2302.13850
[10] Guo, C., Leng, J. W. etc. (2024) GMLake: Efficient and Transparent GPU Memory Defragmentation for Large-scale DNN Training with Virtual Memory Stitching. arXiv: 2401.08156.