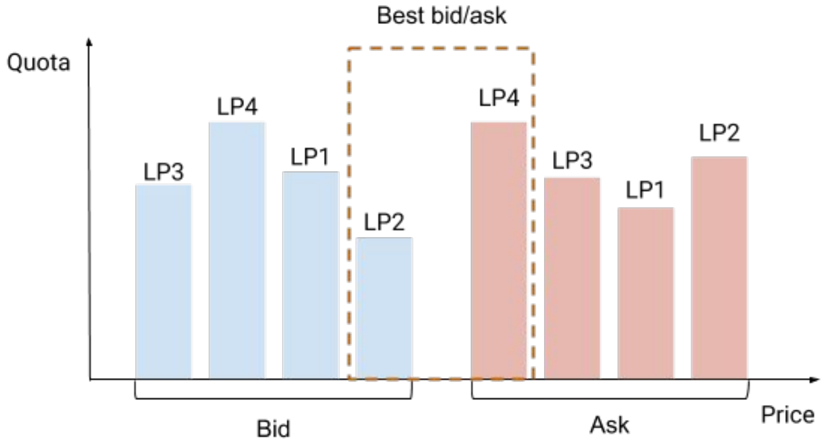
Reinforcement Learning for FX trading

Yuqin Dai, Chris Wang, Iris Wang, Yilun Xu

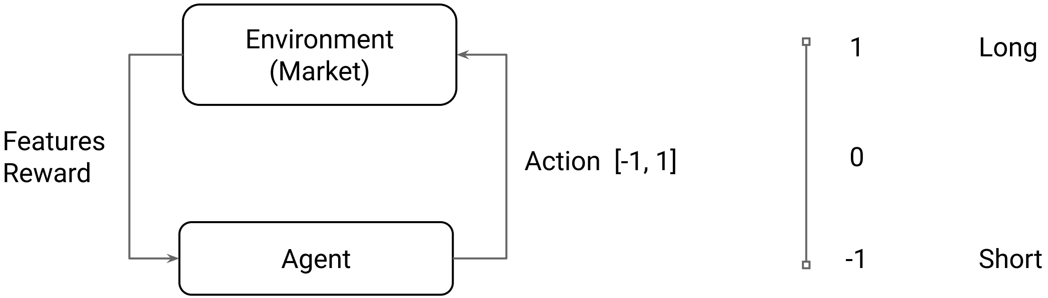
High-frequency Forex data

AUDUSD:CUR
AUD-USD X-RATE

0.7060 USD   +0.0018 +0.26% ▲



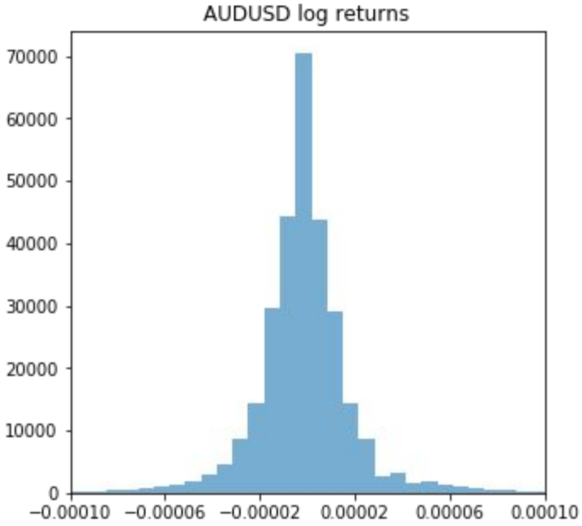
RL Trading Agent



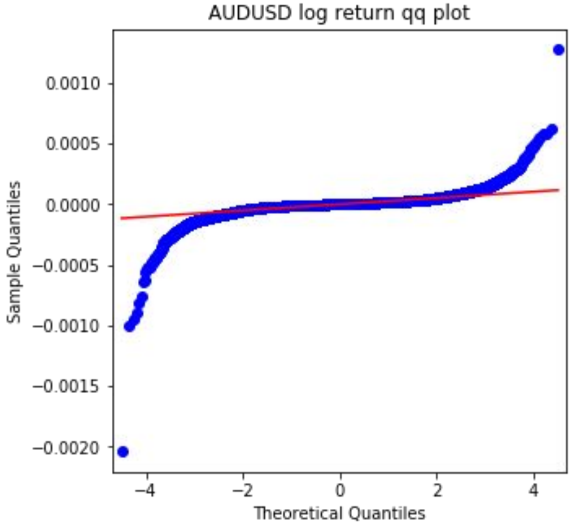
High Frequency Forex Data (1/2)

Time	Bid price	Bid LP	Bid Quota	Ask price	Ask LP	Ask Quota
20190101 00:00:00	0.72714	LP-1	1,000,000	0.72718	LP-2	1,000,000

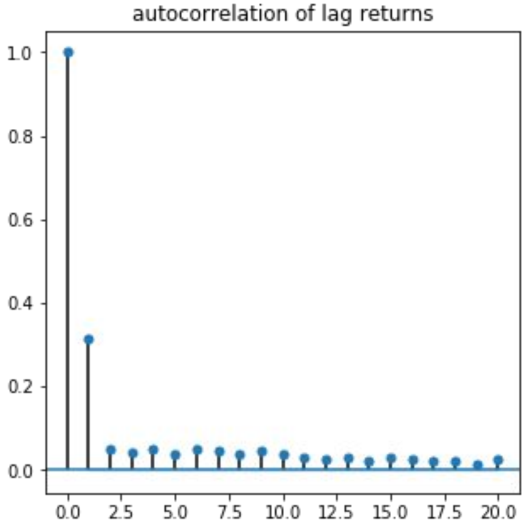
Center around 0



Fat tail



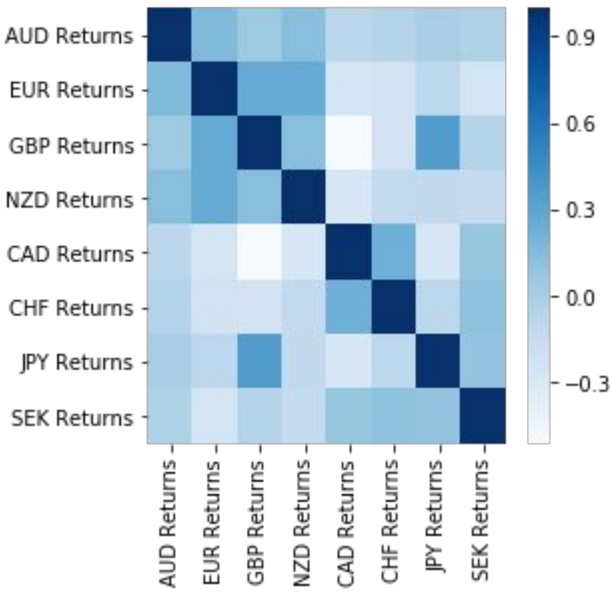
Low autocorrelation



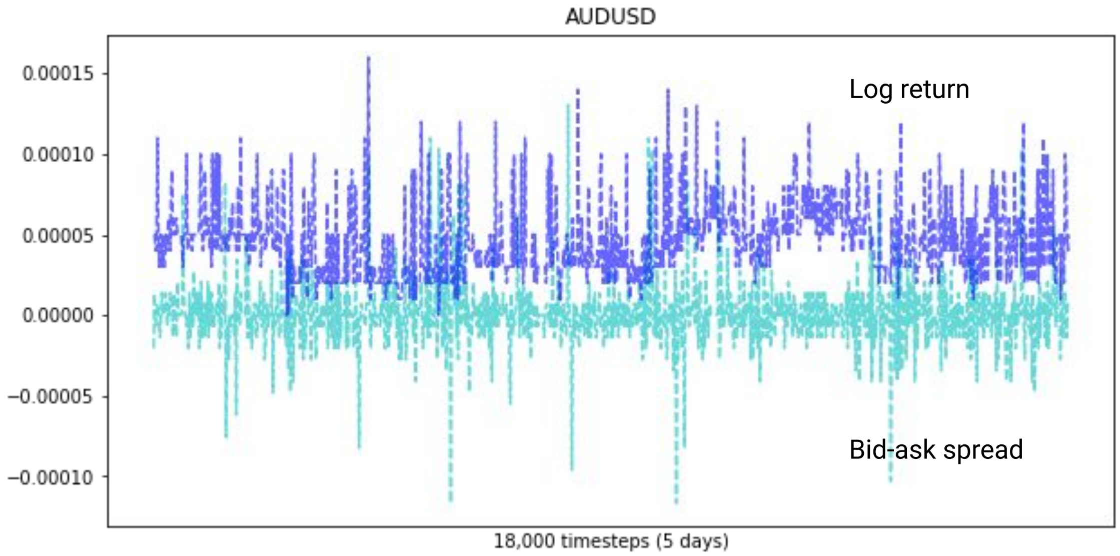
<u>No prior distribution assumption over returns</u>

High Frequency Forex Data (2/2)

Correlation b/w currency pairs



Correlation b/w log return and bid-ask spreads



<u>Additional features from other currency pairs and spreads</u>

Forex Trading Approaches

How is Forex traditionally traded?

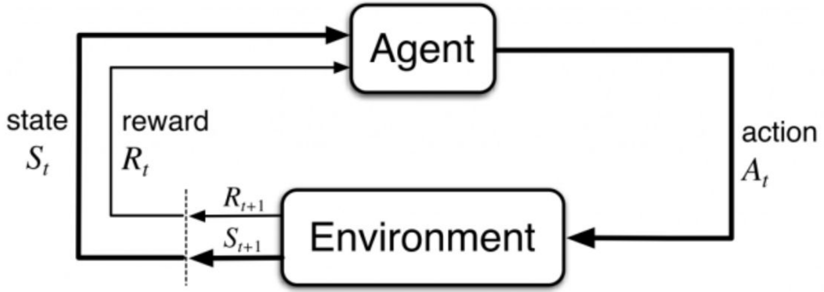
- A few key decisions:
  - Currency pair to trade
  - Position size
  - When to enter/exit
  - Which dealer to use/how to execute the trade
  - Bid-ask spread

- Traditional strategies use Momentum, Mean Reversion, Pivots, Fundamental Strategy, Stop-loss orders
  - Trend-based -> machine learning?
  - Scalping, Day trading, Longer time frames

Reinforcement Learning

Reinforcement learning for forex trading

- Reinforcement Learning (RL) is a type of machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences.
- Trading is an “iterative” process, and past decisions affect future, long-term rewards in indirect ways
    - Compared to supervised learning, we are not making or losing money at a single time step…
- Traditional “up/down” prediction models do not provide an actionable trading strategy
- Incorporate longer time horizon
- Give us more autonomy in trading policy, regularize the model from trading too frequently



Baseline model (1/3)

Goal

Maximize total (undiscounted) return over **1-hour horizon** by making short/long trading decisions for *AUDUSD* per second

Input

Per second bid-ask prices for *AUDUSD* and other available currency pairs; include the recent **16-second returns** as features

Action

Float between -1 (short the currency with all cash) and 1 (long the currency with all cash)

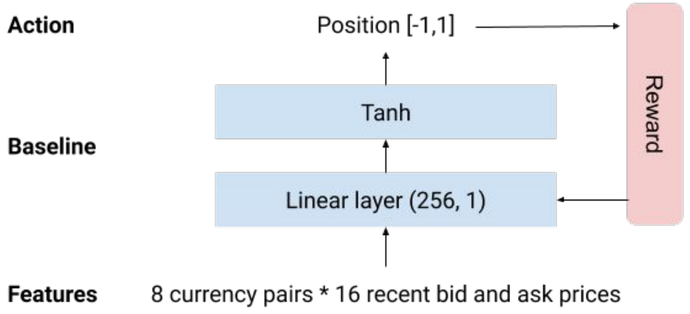
Method

Policy Gradient

- Maximize the “expected” reward when following a policy π

$$J(\theta) = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{\tau} r_t]$$

- Actions are chosen by ‘actor’, i.e. mapping current features to next action
- Gradient descent on π to find the optima



Baseline model (2/3)

In detail

$$a_t = Tanh(< w, x_{t-1} > + b)$$
$$r_t = f(a_t, a_{t-1})$$
$$R = r_1 + \ldots + r_\tau$$

Profits are calculated in two ways

REINFORCE, A Monte-Carlo Policy-Gradient Method (episodic), for estimating $\pi_{\boldsymbol\theta} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol\theta)$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol\theta \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol\theta)$
  Loop for each step of the episode $t = 0, \ldots, T-1$:
    $G \leftarrow$ return from step t (G_t)
    $\boldsymbol\theta \leftarrow \boldsymbol\theta + \alpha\gamma^t G \nabla_{\boldsymbol\theta} \ln \pi(A_t|S_t, \boldsymbol\theta)$

Mid-price approximation

$$\text{action} * \left(\frac{\text{Ask}[t+1] + \text{bid}[t+1]}{2} - \frac{\text{Ask}[t] + \text{bid}[t]}{2}\right)$$

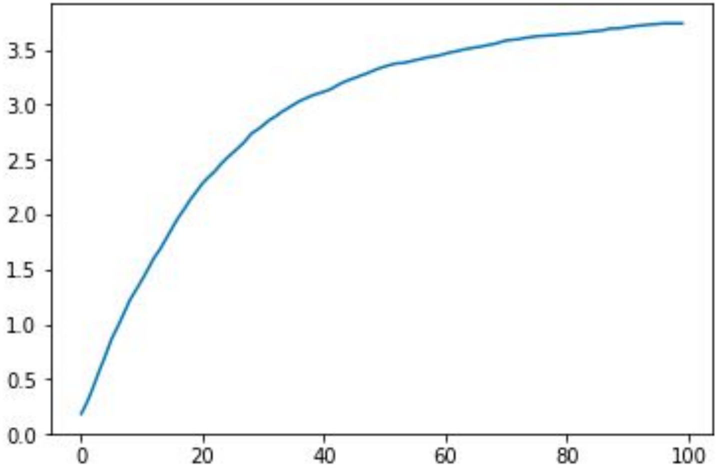
Incorporating bid-ask spreads

a_{t-1}/a_t	-1	0	1
-1	0	-Ask[t]	-2*Ask[t]
0	Bid[t]	0	-Ask[t]
1	2*Bid[t]	Bid[t]	0

Baseline model (3/3)

Total reward using mid-price approximation

Total reward (per $1,000 capital per hour)



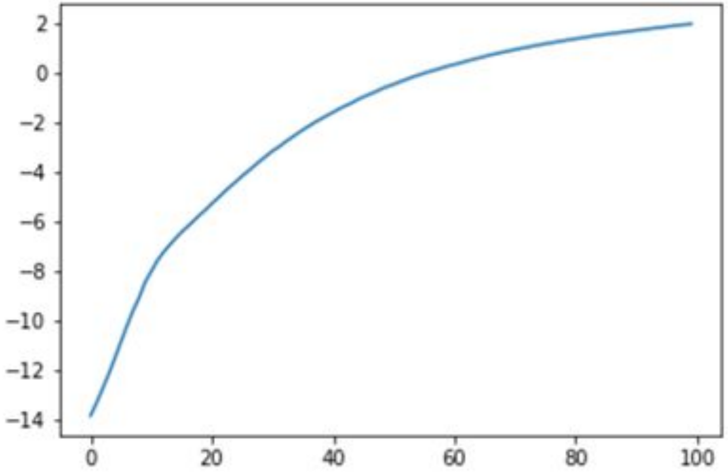
# of epochs

After 5-6 CPU hours' training, RL agent manages to yield **0.4% per hour** on the validation data.

Bid-ask
spread cost

Total reward incorporating bid-ask spread

Total reward (per $1,000 capital per hour)



# of epochs

After 5-6 CPU hours' training, RL agent manages to yield **0.2% per hour** on the validation data.

Next Steps

- **Incorporate better features**
  - Technical features (e.g. chart pattern)

- **Build a better architecture**
  - From linear layers to neural networks

- **Exploration**
  - Explore actions may yield better future rewards

- **Train with more computing power**
  - Cloud computing
  - Parallel computing

Reference

1. Y. Deng, F. Bao, Y. Kong, Z. Ren and Q. Dai, "Deep Direct Reinforcement Learning for Financial Signal Representation and Trading," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 653-664, March 2017.
2. Huang, Chien Yi. "Financial Trading as a Game: A Deep Reinforcement Learning Approach." *arXiv preprint arXiv:1807.02787* (2018).
3. J. Moody and M. Saffell, "Learning to trade via direct reinforcement," in *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 875-889, July 2001.