High-frequency Forex data

**AUDUSD:CUR**
**AUD-USD X-RATE**

| 0.7060<sub>USD</sub> | +0.0018 | +0.26% |

**Environment (Market)**

**Agent**

**Features Reward**

**Action [-1, 1]**

- 1: Long
- 0: Short
- -1: Short

**Quota**

**Bid**

- LP3
- LP1
- LP4

**Ask**

- LP2
- LP3
- LP4
High Frequency Forex Data (1/2)

<table>
<thead>
<tr>
<th>Time</th>
<th>Bid price</th>
<th>Bid LP</th>
<th>Bid Quota</th>
<th>Ask price</th>
<th>Ask LP</th>
<th>Ask Quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>20190101 00:00:00</td>
<td>0.72714</td>
<td>LP-1</td>
<td>1,000,000</td>
<td>0.72718</td>
<td>LP-2</td>
<td>1,000,000</td>
</tr>
</tbody>
</table>

Center around 0

**AUDUSD log returns**

Fat tail

**AUDUSD log return qq plot**

Low autocorrelation

**Autocorrelation of lag returns**

**No prior distribution assumption over returns**
High Frequency Forex Data (2/2)

**Correlation b/w currency pairs**

**Correlation b/w log return and bid-ask spreads**

*Additional features from other currency pairs and spreads*
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 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 $\pi$
  
  $$J(\theta) = \mathbb{E}_{\pi_\theta} \left[ \sum_{t=0}^{\tau} r_t \right]$$

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

---

**Action**
- Position $[-1,1]$

**Baseline**
- **Tanh**
- **Linear layer (256, 1)**

**Features**
- 8 currency pairs * 16 recent bid and ask prices
Baseline model (2/3)

In detail

\[ a_t = \text{Tanh}(w x_{t-1} + b) \]
\[ r_t = f(a_t, a_{t-1}) \]
\[ R = r_1 + \ldots + r_T \]

Profits are calculated in two ways

Mid-price approximation

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

Incorporating bid-ask spreads

<table>
<thead>
<tr>
<th>( a_{t-1}/a_t )</th>
<th>-1</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0</td>
<td>-\text{Ask}[t]</td>
<td>-2*\text{Ask}[t]</td>
</tr>
<tr>
<td>0</td>
<td>\text{Bid}[t]</td>
<td>0</td>
<td>-\text{Ask}[t]</td>
</tr>
<tr>
<td>1</td>
<td>2*\text{Bid}[t]</td>
<td>\text{Bid}[t]</td>
<td>0</td>
</tr>
</tbody>
</table>
After 5-6 CPU hours’ training, RL agent manages to yield **0.4% per hour** on the validation data.

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

