Lecture 8 Part 1
Math 178

Prof. Weiqing Gu

Today

1. Review Matrix Exponential and Define **Matrix Logarithm**
2. Christoffel Symbols and the Compatibility Equations
3. From Directional Derivative to Covariant Derivative

Recall: last time

- If we exponential a skew symmetric matrix, we get a rotation matrix.
- Where the exponential of a matrix was defined to be:

The exponential of a matrix *A* is defined by

$$e^A \equiv \sum_{n=0}^{\infty} \frac{A^n}{n!}.$$

Matrix Logarithm

Given a matrix *B*, another matrix *A* is said to be a **matrix logarithm** of B if $e^A = B$. Because the exponential function is not one-to-one for complex numbers (e.g. $e^{\pi i} = e^{3\pi i} = -1$), numbers can have multiple complex logarithms, and as a consequence of this, some matrices may have more than one logarithm, as explained below.

If *B* is sufficiently close to the identity matrix, then a logarithm of *B* may be computed by means of the following power series:

$$\log(B) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{(B-I)^k}{k} = (B-I) - \frac{(B-I)^2}{2} + \frac{(B-I)^3}{3} \cdots.$$

Specifically, if $\|B - I\| < 1$, then the preceding series converges and $e^{\log(B)} = B$.[1]

The rotations in the plane give a simple example. A rotation of angle α around the origin is represented by the 2×2-matrix

$$A = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}.$$

For any integer n, the matrix

$$B_n = (\alpha + 2\pi n)\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},$$

is a logarithm of A. Thus, the matrix A has infinitely many logarithms. This corresponds to the fact that the rotation angle is only determined up to multiples of 2π.

In the language of Lie theory, the rotation matrices A are elements of the Lie group SO(2). The corresponding logarithms B are elements of the Lie algebra so(2), which consists of all skew-symmetric matrices. The matrix

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

is a generator of the Lie algebra so(2).

Open Access Article

Deep Reinforcement Learning in Agent Based Financial Market Simulation

by Iwao Maeda [1,*], David deGraw [2], Michiharu Kitano [3], Hiroyasu Matsushima [1], Hiroki Sakaji [1], Kiyoshi Izumi [1] and Atsuo Kato [3]

[1] Department of Systems Innovation, School of Engineering, The University of Tokyo, Tokyo 113-8654, Japan

[2] Daiwa Securities Co. Ltd., Tokyo 100-0005, Japan

[3] Daiwa Institute of Research Ltd., Tokyo 135-8460, Japan

[*] Author to whom correspondence should be addressed.

J. Risk Financial Manag. **2020**, *13*(4), 71; https://doi.org/10.3390/jrfm13040071

Received: 28 February 2020 / Revised: 7 April 2020 / Accepted: 8 April 2020 / Published: 11 April 2020

(This article belongs to the Special Issue AI and Financial Markets)

View Full-Text | Download PDF | Browse Figures | Review Reports | Citation Export

https://www.mdpi.com/1911-8074/13/4/71

Agents registered to the simulator are classified into the following two types:

- Deep Reinforcement Learning (DRL) agent
- Fundamental-Chart-Noise (FCN) agent

The DRL agent is the agent we seek to train, while the FCN agents comprise the environment of agents in the artificial market. Details of the DRL agent are described in **Section 5.1**.

The FCN agent (**Chiarella et al. 2002**) is a commonly used financial agent and predicts the log return *r* of an asset with a weighted average of fundamental, chart, and noise terms.

$$r = \frac{1}{w_F + w_C + w_N}(w_F F + w_C C + w_N N)\,. \tag{3}$$

Each terms is calculated by the following equations below. The fundamental term *F* represents the difference between the reasonable price considered by the agent and the market price at the time, the chart term *C* represents the recent price change, and the noise term *N* is sampled from a normal distribution.

$$F = \frac{1}{T}\log\left(\frac{p_t^*}{p_t}\right) \tag{4}$$

$$C = \frac{1}{T}\log\left(\frac{p_t}{p_{t-\tau}}\right) \tag{5}$$

$$N \sim \mathcal{N}(\mu, \sigma^2). \tag{6}$$

p_t and p_t^* are current market price, fundamental price, and τ is the time window size, respectively. p_t^* changes according to a geometric Brownian motion (GBM). Weight values w_F, w_C, and w_N are independently random sampled from exponential distributions for each agent. The scale parameters σ_F, σ_C, and σ_N are required for each simulation. Parameters of the normal distribution μ and σ are fixed to 0 and 0.0001.

The FCN agents predict future market prices $p_{t+\tau}$ from the predicted log return with the following equation:

$$p_{t+\tau} = p_t \exp\left(r\tau\right). \tag{7}$$

The agent submits a buy limit order with price $p_{t+\tau}\left(1-k\right)$ if $p_{t+\tau} > p_t$, and submits a sell limit order with price $p_{t+\tau}\left(1+k\right)$ if $p_{t+\tau} < p_t$. The parameter *k* is called order margin and represents the amount of profit that the agent expects from the transaction. The submitting volume *v* is sampled from a discrete uniform distribution $u\{1,5\}$. In order to control the number of outstanding orders in the market orderbook, each order submitted by the FCN agents has a time window size, after which the order is automatically canceled.

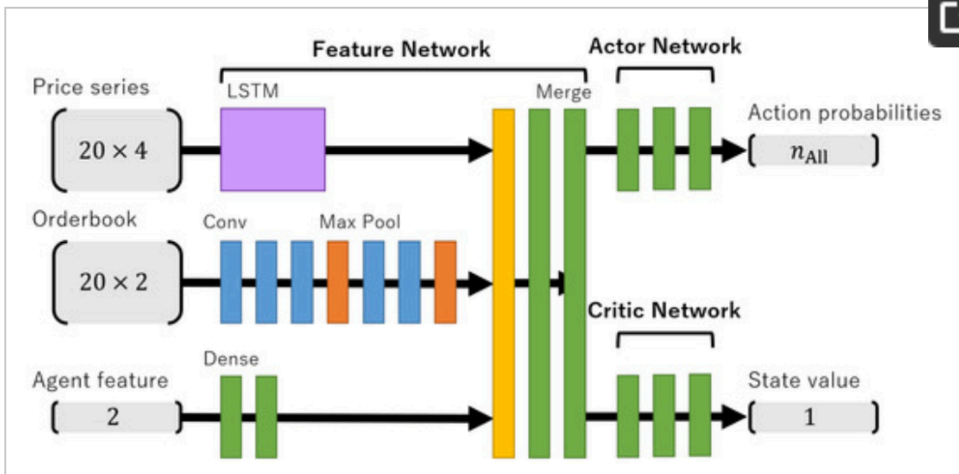

Figure 2. Overview of the A2C network used in this research ($n_{\mathrm{market}} = 1$). The network takes price series, orderbook feature and agent feature as input variables and outputs action probabilities and state values. Purple, blue, orange, green, and yellow boxes in the network represent LSTM, convolutional, max pooling, dense, and merge layers, respectively. LSTM layers have tangent hyperbolic activation. Convolutional and dense layers except the last layers of actor and critic networks have ReLU activation. Action probabilities are calculated by applying softmax activation to the output of actor network.