QUANTUM CIRCUIT LEARNING TO COMPUTE OPTION PRICES AND THEIR SENSITIVITIES

TAKAYUKI SAKUMA

Abstract. Quantum circuit learning is applied to computing option prices and their sensitivities. The advantage of this method is that a suitable choice of quantum circuit architecture makes it possible to compute the sensitivities analytically by applying parameter-shift rules. We expect our numerical result to pave the way for using quantum machine learning for option pricing.

1. Introduction

The classical–quantum hybrid approach, which uses both quantum and classical computers, has received much attention recently from the expectation of is use on near-term quantum devices. For example, one approach called the variational quantum eigensolver, proposed by Peruzzo et al. (2014), has been studied intensively in field of quantum chemistry. Another approach called quantum machine learning is also an active area of research (for example, Rebentrost et al. (2014)). As a trial of using quantum computers in the field of quantitative finance, Sakuma (2020) used a simulator of a deep quantum neural network (DQNN) provided by Beer et al. (2018) to learn implied volatilities and Black-Scholes European option prices. However, a DQNN is assumed to be used in real quantum computers but is too complicated to be implemented in such computers. Therefore, in this paper, quantum circuit learning (QCL), one type of quantum machine learning, is investigated in the context of computing option prices and their sensitivities. Application of classical machine learning to quantitative finance is very active. Differential machine learning developed by Huge and Savine (2020) in classical computing is especially useful because it can compute sensitivities via twin neural networks. The analogous approach in quantum machine learning is the use of parameter-shift rules (Mitarai et al. (2018), Schuld et al. (2019)). Using parameter-shift rules, together with chain rules, makes it possible to compute sensitivities analytically from a learned quantum circuit. We expect our numerical results to pave the way to the application of quantum machine learning to option pricing.

Date: September 13, 2021.

Key words and phrases. option pricing, sensitivities, quantum circuit, quantum machine learning, parameter shift rule.

Faculty of Economics, Soka University.

E-mail: tsakuma@soka.ac.jp.

2. Quantum circuit learning

As mentioned in the previous section, QCL, which was introduced by Mitarai et al. (2018), is the quantum analogue of classical deep learning; the neural networks are replaced by a parameterized quantum circuit. The motivation is that the superposition principle makes it possible to enhance learning of expressing a high-dimensional nonlinear function.

We use QCL in this paper as follows. We use a 2-qubit gate. Suppose we have N training pairs S_i, V_i $(i = 1, ..., N)$, where S_i are stock prices and V_i are Black-Scholes European call prices. First we convert S_i and V_i into numbers between 0 and 1 via

$$x(S_i) = tanh((\frac{S_i}{C_1})^{\beta}), y(V_i) = tanh((\frac{V_i}{C_2})^{\gamma}), \tag{1}$$

where C_1, C_2, β, γ are fixed parameters.

Then we apply the following U_{in} in order to embed real data into quantum states: $|\Psi_{in}(\varphi)\rangle = U_{in}(\varphi)|0\rangle$, where

$$U_{in}(\varphi) = [R_2^Y(\varphi_5) \otimes R_1^Y(\varphi_4)]R^{XX}(\varphi_3)[R_2^Y(\varphi_2) \otimes R_1^Y(\varphi_1)]. \tag{2}$$

$|0\rangle$ represents the initial state and $\varphi_i(x) = arcsin(x)$ for all i. R_i^Y is a rotation gate acting on the ith qubit around the y-axis and R^{XX} is Ising coupling gate.

Finally, applying $U_{in}(\varphi)$ to a parameterized quantum circuit, $U(\theta)$ gives the final state as

$$|\Psi_{out}(\theta, \varphi)\rangle = U(\theta)|\Psi^{in}(\varphi)\rangle. \tag{3}$$

Measuring some observable B gives

$$\langle B(\theta, \varphi)\rangle \equiv \langle \Psi_{out}|B|\Psi_{out}\rangle \tag{4}$$

and we regard this as output $y(V_i, \theta, \varphi)$ of this quantum circuit. We choose the parameter θ by minimizing the following loss function:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{(y(V_i) - y(V_i, \theta, \varphi))^2}{y(V_i)} \tag{5}$$

and the optimization is conducted on classical computers.

3. Parameter-shift rules

One advantage of QCL is that we can compute sensitivities analytically from the learned quantum circuit by applying parameter-shift rules. Suppose we have the following function of θ:

$$f(\theta) = \langle \Psi_{in}|U^{\dagger}(\theta)BU(\theta)|\Psi_{in}\rangle, U(\theta) = e^{-ia\theta G}, \tag{6}$$

where a is a real number. Then if G has two unique eigenvalues e_0 and e_1, the following holds:

$$\frac{\partial f(\theta)}{\partial \theta} = r[f(\theta + \frac{\pi}{4r}) - f(\theta - \frac{\pi}{4r})], r = \frac{a}{2}(e_1 - e_0). \tag{7}$$

For example, suppose U is one of rotation gates R^X, R^Y, R^Z where $a = 1/2$ and G is a Pauli gate. Then $e_0 = -1, e_1 = 1$, so $r = 1/2$. This means that if U_{in} consists of rotation gates R^X, R^Y, R^Z and Ising coupling gates R^{XX}, R^{YY}, R^{ZZ}, then the sensitivities can be calculated analytically via parameter-shift rules:

$$\frac{\partial \langle B(\theta, \varphi)\rangle}{\partial \varphi} = \frac{1}{2}(\langle B(\theta, \varphi + \frac{\pi}{2})\rangle - \langle B(\theta, \varphi - \frac{\pi}{2})\rangle). \tag{8}$$

For example, we can calculate delta as

$$\frac{\partial V}{\partial S} = \frac{\partial \langle B(\theta, \varphi)\rangle}{\partial S}\frac{\partial y}{\partial V}, \tag{9}$$

where

$$\frac{\partial \langle B(\theta, \varphi)\rangle}{\partial S} = \frac{\partial x}{\partial S}\sum_{j=1}^{5}\frac{\partial \varphi_j}{\partial x}\frac{\partial \langle B(\theta, \varphi)\rangle}{\partial \varphi_j}, \tag{10}$$

$$\frac{\partial \varphi_j}{\partial x} = \frac{1}{\sqrt{1-x^2}}, \frac{\partial x}{\partial S} = \frac{\beta}{C_1}(\frac{x}{C_1})^{\beta-1}(1-x^2), \frac{\partial y}{\partial V} = \frac{\gamma}{C_2}(\frac{V}{C_2})^{\gamma-1}(1-y^2). \tag{11}$$

Parameter optimization in classical deep learning is conducted via backpropagation, but we cannot observe gradients when using quantum computers. Instead, quantum circuit $U(\theta)$ employed in this paper can use parameter-shift rules:

$$\frac{\partial \langle B(\theta, \varphi)\rangle}{\partial \theta} = \frac{1}{2}(\langle B(\theta + \frac{\pi}{2}, \varphi)\rangle - \langle B(\theta - \frac{\pi}{2}, \varphi)\rangle). \tag{12}$$

4. Numerical examples

As a numerical example, we consider the data given in table 1. The following parameters are used: $K = 100, T = 0.25, \sigma = 0.15, \beta = 5.0, \gamma = 0.75, C_1 = 100, C_2 = 10$. For implementation of QCL, we use the Pennylane simulator developed by Bergholm et al. (2018). Pennylane is a Python library specialized in quantum machine learning. The template we used as $U(\theta)$ is StronglyEntanglingLayer with the number of layers 3, which consists of rotation gates and CNOT gates and was introduced in Schuld et al. (2020). To compute θs, we use an Adam optimizer with a learning rate of 0.01 for 1000 iterations. The gradients used in the Adam optimizer are computed analytically by parameter-shift rules. As table 2 demonstrates, QCL succeeded in learning the prices with high accuracy. The mean relative error is 0.059%. Table 3 gives a comparison between the delta computed analytically and the delta computed via parameter-shift rules. The mean relative error is 0.228% and some relative errors become large compared to prices. One possible reason

could be the parameter-shift rules themselves: shifting angles by 0.5π in a periodic function could produce inappropriate output from the learned quantum circuit.

One scheme to increase the accuracy of sensitivities is to use the following additional loss function:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{(y(V_i)-y(V_i,\theta,\varphi))^2}{y(V_i)}+\frac{\lambda}{N}\sum_{i=1}^{N}\frac{(\frac{\partial y(V_i)}{\partial x}-\frac{\partial y(V_i,\theta,\varphi)}{\partial x})^2}{\frac{\partial y(V_i)}{\partial x}}, \tag{13}$$

where

$$\frac{\partial y(V,\theta,\varphi)}{\partial x}=\frac{\partial \varphi}{\partial x}\sum_{j=1}^{N}\frac{(\langle B(\theta,\varphi_j(x(S))+\frac{\pi}{2})\rangle-\langle B(\theta,\varphi_j(x(S))-\frac{\pi}{2})\rangle)}{2}, \tag{14}$$

$$\frac{\partial y(V)}{\partial x}=\frac{\frac{\partial V}{\partial S}\frac{\partial y}{\partial V}}{\frac{\partial x}{\partial S}}. \tag{15}$$

If we increase λ, we can expect the accuracy of delta to increase and that of price to decrease.

5. Conclusion

In this paper, a version of quantum machine learning called quantum circuit learning is applied to learning European option prices and their sensitivities and the usefulness of this method is demonstrated by numerical examples. One possible extension is the use of the natural gradient approach proposed by Stokes et al. (2019), which could make the Adam optimization more efficient. In addition, to consider practical use of QCL on real quantum computers, we need to investigate the robustness of our approach under noisy environments. Extensions of the use of QCL to learning Bermudan swaptions are under investigation.

6. Acknowledgments

We appreciate Keisuke Fujii and Naosuke Nakamura for their valuable feedback. We also acknowledge support from the MITOU Target Program of Information-technology Promotion Agency, Japan.

References

[1] Beer, K., Bondarenko, D., Farrelly, T., Osborne, T.J., Salzman, R., Scheiermann, D., Wolf, R. (2020). Training deep quantum neural networks. *Nature Communications*, **11**, 808.

[2] Bergholm, V., Izaac, J., Schuld,M., Gogolin, C., Alam, M.S., Ahmed, S., Arrazola, J.M, Blank, C., Delgado, A., Jahangiri, S., McKiernan, K., Meyer, J.J., Niu, Z., Száva, A., Killoran, N. (2018). Pennylane: Automatic differentiation of hybrid quantum-classical computations. arXiv:1811.04968.

[3] Huge, B., Savine, A. (2020). Differential Machine Learning. arXiv:2005.02347.

[4] Mitarai, K., Negoro, M., Kitagawa, M. and Fujii, K. (2018). Quantum circuit learning. *Physical Review A*, **98**(3), 032309.

[5] Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P.J., Aspuru-Guzik, A., .O ' Brien, J.L. (2014). A variational eigenvalue solver on a photonic quantum processor. *Nature Communication*, **5**, 4213.
[6] Rebentrost, P., Mohseni, M., Lloyd, S. (2014). Quantum support vector machine for big data classification. *Physical Review Letters*, **113**(13), 130503.
[7] Sakuma, T. (2020). Application of deep quantum neural networks to finance. arXiv:2011.07319.
[8] Schuld, M., Bergholm, V., Gogolin, C., Izaac, J., Killoran, N. (2019). Evaluating analytic gradients on quantum hardware. *Physical Review A*, **99**(3), 032331.
[9] Schuld, M., Bocharov, A., Svore, K.M., Wiebe, N. (2020). Circuit-centric quantum classifiers. *Physical Review A*, **101**(3), 032308.
[10] Stokes, J., Izaac, J., Killoran, N., Carleo, G. (2020). Quantum Natural Gradient. *Quantum*, **4**, 269.

TABLE 1. Data used

Spot price	BS call price
93	0.6404
95	1.0728
97	1.6861
100	2.9914
103	4.7689
105	6.1925
107	7.7758

TABLE 2. QCL performance for learning price

Spot	Training price	Output	Relative error
93	0.6404	0.6400	0.057%
95	1.0728	1.0732	0.035%
97	1.6861	1.6877	0.096%
100	2.9914	2.9900	0.047%
103	4.7689	4.7643	0.096%
105	6.1925	6.1927	0.002%
107	7.7758	7.7818	0.078%

TABLE 3. QCL performance for learning delta

Spot	Training delta	Output	Relative error
93	0.1762	0.1757	0.269%
95	0.2590	0.2599	0.328%
97	0.3562	0.3563	0.032%
100	0.5150	0.5131	0.357%
103	0.6670	0.6677	0.104%
105	0.7543	0.7580	0.496%
107	0.8263	0.8264	0.009%