



OPEN Practical application of quantum neural network to materials informatics

Hirotoishi Hirai

Quantum neural network (QNN) models have received increasing attention owing to their strong expressibility and resistance to overfitting. It is particularly useful when the size of the training data is small, making it a good fit for materials informatics (MI) problems. However, there are only a few examples of the application of QNN to multivariate regression models, and little is known about how these models are constructed. This study aims to construct a QNN model to predict the melting points of metal oxides as an example of a multivariate regression task for the MI problem. Different architectures (encoding methods and entangler arrangements) are explored to create an effective QNN model. Shallow-depth ansatzs could achieve sufficient expressibility using sufficiently entangled circuits. The “linear” entangler was adequate for providing the necessary entanglement. The expressibility of the QNN model could be further improved by increasing the circuit width. The generalization performance could also be improved, outperforming the classical NN model. No overfitting was observed in the QNN models with a well-designed encoder. These findings suggest that QNN can be a useful tool for MI.

The application of machine learning (ML) to the development of materials is becoming increasingly important^{1,2}. Materials informatics (MI) is a field of information science used to develop materials^{3–5}. It involves constructing a predictive model of physical properties from a limited amount of data obtained from experiments or simulations and then screening materials with the desired performance from a large group of materials. The challenge with MI is that the data are often limited and prone to noise owing to errors in the experimental data, making it difficult to construct a model with a good generalization performance (prediction performance for unknown materials)^{1,6}.

Recently, a quantum neural network (QNN)⁷, also referred to as quantum circuit learning⁸, has been developed as an ML algorithm for quantum computers⁹. It is a quantum-classical hybrid algorithm based on the variational quantum algorithm¹⁰, which has been developed to work with noisy intermediate-scale quantum (NISQ) devices¹¹. A QNN model is built by minimizing the discrepancy between the output of the quantum circuit and labeled data by adjusting the circuit parameters to their optimal values. The advantage of QNN is that it can use high-dimensional quantum states as trial functions that are hard to generate on a classical computer⁸. Another advantage of a QNN is that the unitarity of quantum circuits serves as regularization to prevent overfitting⁸. In a classical neural network (NN) model, a regularization term is incorporated into the cost function to constrain the norm of the learning parameters and to reduce the model's expressibility to prevent overfitting¹². In contrast, the norm of parameters is automatically limited to one due to unitarity in a QNN model, i.e., the regularization function is inherently provided. QNNs have also been reported to afford predictive models with excellent generalization performance even when only a small amount of training data is available¹³. It has also been reported that the smaller the data size of the problem, the greater the advantage of the generalization performance of QNNs over classical NNs¹⁴.

These characteristics of QNNs may be particularly useful in MI. The atomic configuration can be used to predict the properties of materials because the Hamiltonian can be determined from the atomic configuration and the Schrödinger equation can be solved (in principle) using the Hamiltonian to obtain the properties of the material. ML models can be used instead of solving the Schrödinger equation because solving the many-body Schrödinger equation is extremely difficult¹⁵. Such concepts have been considered in the MI¹⁶ and QSAR (Quantitative Structure-Affinity Relationship)¹⁷ fields. The construction of an ML model that bypasses the Schrödinger equation is expected to be naturally aided by a QNN model with quantum architectures.

In this study, we attempted to construct a successful QNN model to predict the melting points of metal oxides. Calculating thermodynamic properties such as melting points is difficult with first-principles calculations

Toyota Central R&D Labs., Inc., 41-1, Yokomichi, Nagakute, Aichi 480-1192, Japan. email: hirotoishih@mosk.tytlabs.co.jp

because of the high computational cost and lack of accuracy^{18,19}. Therefore, it is important to develop a practical melting point prediction model to identify functional materials^{20–22}. However, because QNNs are an emerging field, there is still a lack of understanding of how to construct effective QNN models. We considered various architectures (ansatz and encoding methods) to create an effective QNN model for the practical task of predicting melting points.

Methods

Data set

This study addresses the issue of predicting the melting points of metal oxides. The melting point data for metal oxides listed in²³ were expanded to 70 metal oxides by adding data from other references^{24–26}. Each material was identified in the Materials Project database²⁷, and the following five explanatory variables were obtained (some variables were calculated from structural data in the database in²⁷).

- formation_energy_per_atom: Formation energy per atom
- band_gap: Band gap energy
- density: Mass density
- cati_anio_ratio: Ratio of the number of cations and anions
- dist_from_o: Minimum distance from the oxygen atom to cation

The constructed dataset is available in the Supplementary Information. These explanatory variables were normalized to have a mean of 0 and a variance of 1 for the training data and further scaled to have a maximum value of 1 and a minimum value of -1. The objective variable (melting point temperature in °C) was divided by 3500 and scaled such that the maximum value was approximately 1 (the highest melting point of metal oxides treated in this study was 3390 °C).

The k-fold cross-validation method²⁸ was used to evaluate the accuracy of the constructed regression models. In this study, the 70 dataset entries were divided into five groups; one group was used as the test data, while the other groups were used as the training data. This procedure was performed for all five combinations, and the average accuracy of the five models was used as the final accuracy. The root mean square error (RMSE) was used as a measure of accuracy. It should be noted that if you want to build a model with uniform predictive performance over any temperature range, it is better to use a metric such as relative error. However, in some cases, you may want to reduce the absolute error of the model, such as when searching for materials with high thermal durability. In this study, assuming such a case, we used RMSE as an indicator.

QNN models

The QNN model is composed of three components: an encoder that transforms explanatory variables into a quantum state, an ansatz which is a quantum circuit with learning parameters, and a decoder that converts the quantum state into an output value. Each component is described in detail in the following sections. In this study, QNN models were implemented using Pytket²⁹, a Python module for quantum computing, and quantum circuit calculations were performed using state vector calculations with the Qulacs³⁰ backend, a quantum computing emulator. The mean squared error (MSE) between the labeled data and model predictions was used as a cost function. The Powell method³¹ was used to optimize the learning parameters.

Encoder

In this study, Ry rotation gates³² were used as encoders. We used two different methods to transform each scaled explanatory variable x into the rotation angle θ : $\theta = \pi x$ and $\theta = \arctan(x) + \pi/2$. The arctangent allows the scaled explanatory variable to be uniquely converted to a rotation angle even if the value is outside the scale range (-1,1) when the scaler is used for the test data. We constructed a 5-qubit QNN model with each explanatory variable encoded in one qubit and a 10-qubit QNN model with each explanatory variable encoded in two qubits, as shown in Fig. 1a,b, respectively.

In the 10-qubit model, two different encoding methods were tested: one with redundant imputation of the explanatory variable x and the other with imputation as x and x^2 , as indicated by the parentheses in Fig. 1b.

Ansatz

In this study, as the ansatz part of the QNN, we examined ansatzs with the quantum circuits shown in Fig. 2 as the depth 1-block.

In these ansatzs, an entangler (a group of 2-qubit operations) was placed after the Ry rotation gate. Although Fig. 2 shows CNOT (CX) gates as 2-qubit gates, and we also examine the case using controlled-Z (CZ) gates. circular2 (c) and circular4 (d) contain 2-qubit operations up to the second and fourth nearest-neighbor qubits, respectively. Each Ry gate has an independent learning parameter θ . Because there are five (10) Ry gates in the depth 1-block of the 5-qubit (10-qubit) model, the number of parameters for the QNN model with depth d is $5d$ ($10d$). In this study, d values of 1 to 7 were considered.

Decoder

The QNN decoder takes the expectation value of an observable quantum state generated by the encoder-ansatz quantum circuit as the output of the regression model. For the 5-qubit QNN models, the expectation value of σ_z^4 (the Z-axis projection of the lower-end qubit) was used as the decoder (note that the number on the label begins with zero). For the 10-qubit QNN models, the expected value of $\sigma_z^4 + \sigma_z^9$ was used.

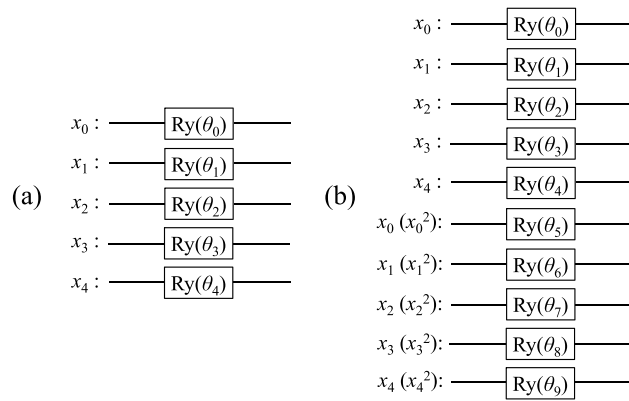


Figure 1. The Ry encoders used in this study: (a) 5-qubit model and (b) 10-qubit model. The Ry gate acts on each qubit initialized to $|0\rangle$. The scaled explanatory variables x_i (or x_i^2) are converted to the rotation angles θ_i according to $\theta = \arctan(x) + \pi/2$ or $\theta = \pi x$.

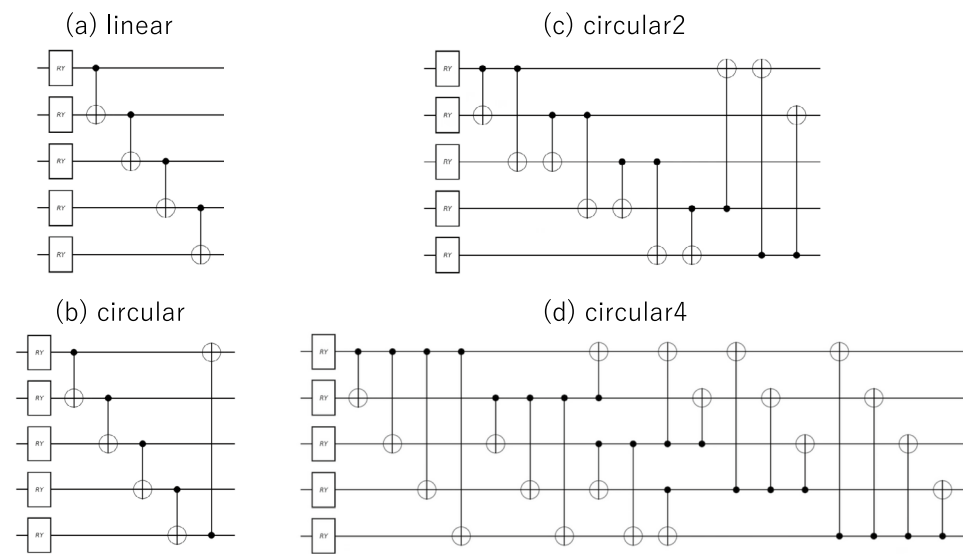


Figure 2. The depth 1-block of each ansatz used in this study. These circuits consist of Ry rotating gates and entanglers (groups of 2-qubit operations).

Circuit analysis

The higher the expressibility of the ansatz, the better the regression accuracy. Therefore, the quantitative evaluation of the expressibility of an ansatz plays an important role in the construction of a QNN model. In this study, Kullback-Leibler (KL) divergence³³ and entanglement entropy³⁴ were used as ansatz evaluation tools. In the KL divergence metric, the KL divergence between the fidelity distribution of quantum states obtained from an ansatz with random parameters and the fidelity distribution for Haar measures is used to quantify expressibility³³,

$$KL(P_{C_{ansatz}}(F)||P_{Haar}(F)) = \int_0^1 P_{C_{ansatz}}(F) \log(P_{C_{ansatz}}(F)/P_{Haar}(F))dF. \tag{1}$$

To obtain the fidelity distribution $P_{C_{ansatz}}(F)$, we sampled a random set of parameters 100,000 times. An analytical solution for $P_{Haar}(F)$ is known, and $P_{Haar}(F) = (N - 1)(1 - F)^{N-2}$ (N is the dimension of the Hilbert space, and for an n -qubit system $N = 2^n$)³³. In the entanglement entropy, we use the following equation as the index to quantify the entanglement strength of the ansatz,

$$\sum_i^n \langle S(\rho_i(C_{ansatz})) \rangle / n. \tag{2}$$

Here, n is the number of qubits, and $S(\rho_i(C_{ansatz})) = -\text{Tr}[\rho_i \log \rho_i]$ is the entanglement entropy calculated using the i th qubit as a subsystem (the entanglement entropy between the i th qubit and other qubits). The above equation means the average of the entanglement entropy of each qubit. $\langle S(\rho_i(C_{ansatz})) \rangle$ is the statistical average

of the entanglement entropy of the i th qubit for a set of random ansatz parameters (we sampled 100,000 sets of parameters).

Classical NN models

A conventional neural network (NN) model was constructed for comparison. To vary the number of learning parameters in the NN regression model, models 5-5-1(36), 5-3-1(22), 5-2-1(15), 5-1(6) were prepared, where the numbers indicate the number of neurons in the fully connected layers, “-” indicates “between layers”, and the numbers in parentheses represent the number of training parameters (weight and bias parameters). A sigmoid function was used as the activation function. PyTorch³⁵ is used to construct and train the NN model. The Adam optimizer³⁶, an extended version of the stochastic gradient descent, was used with a learning rate of 0.02 over 10,000 epochs. L2 regularization was applied to prevent overfitting. The weight parameter for L2 regularization (a hyperparameter set by the user) was used to minimize the RMSE for the test data (average of five groups). We tested the parameters of 10^{-n} with $n = 2, 3, 4$, and 5 , and found that $n = 4$ gave the best performance for all models.

Results and discussions

Encoder

First, we present the results of the analysis of the effects of different methods on transforming the explanatory variable x into the rotation angle θ during $Ry(\theta)$ encoding. The RMSE of the QNN models with $Ry(\pi x)$ and $Ry(\arctan(x)+\pi/2)$ are shown in Fig. 3.

Here, the number of qubits was fixed at five, and the entangler was fixed in a linear arrangement (Fig. 2a). The number of parameters in the model increased with the depth of the ansatz. For comparison, Fig. 3 also shows the results for the classical NNs with and without regularization as “NN reg.” and “NN”, respectively. It can be confirmed that NN models without regularization induce overfitting. That is, the RMSE of the test data increases as the number of parameters increases. When $Ry(\pi x)$ was used as the encoder, QNN models with a small number of parameters (shallow ansatzs) exhibited significantly poorer regression performance. The reasons for this are as follows. Here, the explanatory variable $x(-1,1)$ is converted into a rotation angle $\theta(-\pi,\pi)$, which results in a round trip around the Bloch sphere, and the Z-axis projection after encoding is not unique. In extreme cases, $x = -1$ and $x = 1$ are encoded in the same quantum state. As the number of parameters increases (the ansatz is deepened), the RMSE becomes smaller for the training data. This is thought to be because the data are fully trained by brute force with a large number of parameters. However, for the test data, overfitting was observed for the models with deep ansatzs. However, in the QNN model using $Ry(\arctan(x)+\pi/2)$ as the encoder, the RMSE was small, even for a model with a small number of parameters (shallow ansatzs). It can also be confirmed that overfitting does not occur even in models with a large number of parameters (deep ansatzs). In this case, the RMSE values for the test and training data showed approximately the same dependence on the number of parameters as the classical NN with regularization, confirming that the automatic regularization function of the QNN was effective. In the following discussion, $Ry(\arctan(x)+\pi/2)$ was used as the encoder.

Ansatz

Next, we analyzed the impact of ansatz differences on the regression performance of the QNN. The differences between the CX and CZ gates is shown in Fig. 4, where the number of qubits is fixed to five and the entangler is fixed to the “linear” arrangement.

From the comparison of the ansatzs with the CX and CZ gates, the QNN models with CZ have lower expressibility. Because the observation axis is set to the Z axis (σ_z is used for the decoder), the phase inversion by the

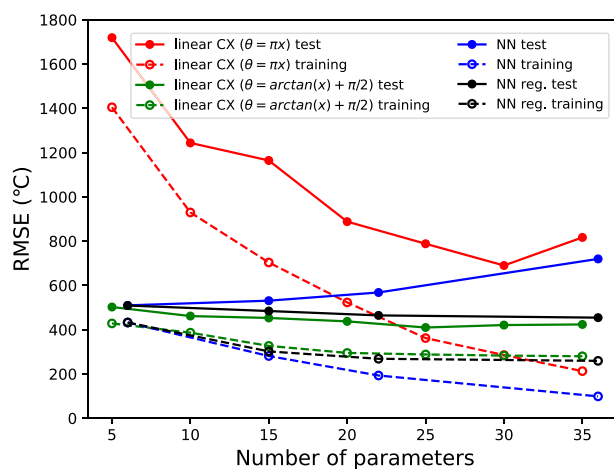


Figure 3. The RMSE for the QNN models with $Ry(\pi x)$ and $Ry(\arctan(x) + \pi/2)$ encoders. The number of qubits is fixed to five and the entangler is fixed to the linear arrangement. The classical NN results with and without regularization are also shown.

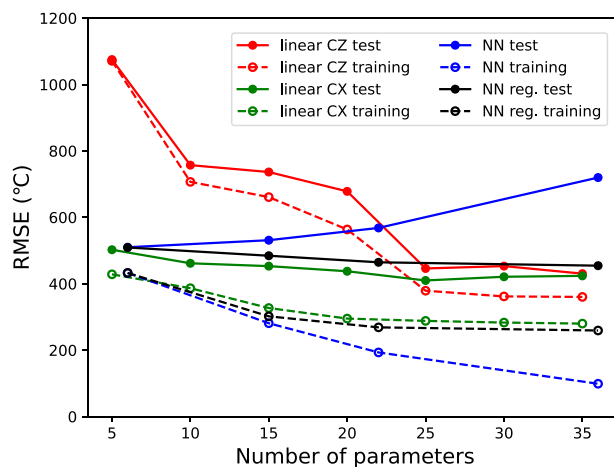


Figure 4. The difference between the CX and CZ gates for QNN regression model performance. The number of qubits is fixed to five and the entangler is fixed to the linear arrangement.

CZ gate does not directly change the projection of the Z axis (the Pauli gate based on the basis axis does not change the state, except for the phase). As a result, QNN models with CZ gates are considered to have lower expressibility, particularly when the number of parameters is small. In the following discussion, only CX gates were used as entanglers.

Figure 5 shows the impact of different entangler structures (Fig. 2) on QNN performance.

The QNN models with ansatz “linear”, “circular”, and “circular2” show similar performances, while the QNN model with ansatz “circular4” performs significantly worse for a small number of parameters (shallower depths). To investigate the factors contributing to these results, the KL divergences and entanglement entropies of these ansatzs were examined, as shown in Fig. 6a,b, respectively.

These figures also show the results for the “full” arrangement shown in Fig. 7a.

These results indicate a correlation between KL divergence and entanglement entropy, with a larger entanglement entropy indicating a smaller KL divergence. Therefore, an ansatz with larger entanglement has greater expressibility. It can be expected that the entanglement becomes stronger as the number of CXs increases, such as “linear”, “circular” and “circular2”, but it is noticeably weaker for the “full” and “circular4” entanglers. This can be understood based on the following facts: It is known that a “full” entangler has a reduced circuit and is equivalent to an inverse “linear” entangler³⁷ (Fig. 7a). This implies that entanglement cannot be enhanced by blindly including a large number of CXs, provided that a simple equivalent circuit (reduced circuit) exists. However, it is difficult to determine whether a circuit has a reduced equivalent circuit. Therefore, we optimized each entangler using the circuit optimization function in tket²⁹ and explored a reduced equivalent circuit. The results are summarized in Fig. 7. There is a significantly reduced equivalent circuit for “circular4”. In the reduced “circular4” entangler, each qubit has only a CX gate with the bottom qubit, so the entanglement is weak, as can also be seen from the entanglement entropy. In contrast, “circular2” is not significantly simplified, and the entanglement is not notably weak.

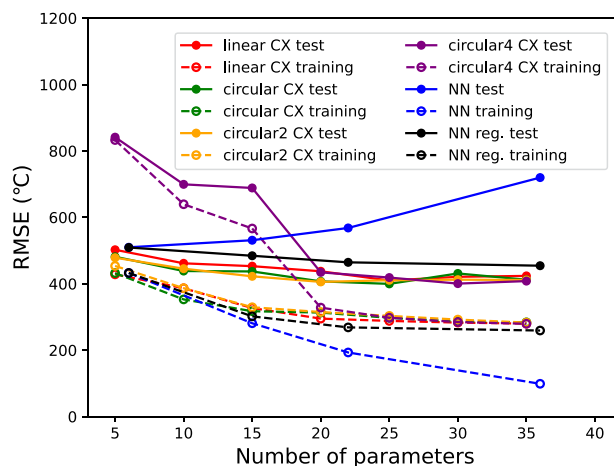


Figure 5. The impact of different entangler structures on QNN performance.

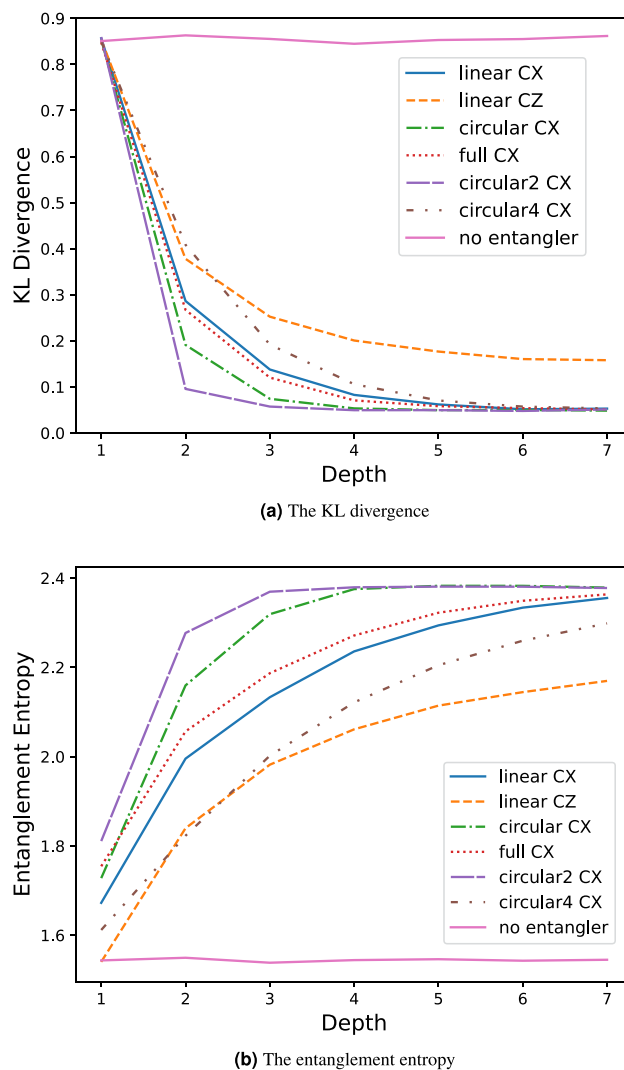


Figure 6. The KL divergence (a) and the entanglement entropy (b) for each ansatz.

Figure 6 show the KL divergence and the entanglement entropy for the “linear CZ” entangler, and indicate that the QNN model with this entangler has less expressibility. These results indicate that KL divergence and entanglement entropy may be able to screen out ansatz with poor expressibility.

In this study, there were no large differences in QNN performance among ansatzs with entanglement greater than the “linear” entangler, and therefore, the “linear” entangler was found to provide sufficient entanglement for the QNN model for this problem. This implies that a model with satisfactory performance can be constructed using only 2-qubit operations between neighboring qubits, suggesting that it may be feasible to operate the QNN model on superconducting quantum computers, which are widely used today, in the near future.

Circuit width

The effect of the number of qubits (circuit width) on the performance of the QNN model is illustrated in Fig. 8.

Here, the entangler is fixed to the “linear” arrangement. When comparing the RMSE for the training data, the model with twice the number of qubits (w2) had a smaller error than the original model, indicating that its expressibility was improved by increasing the basis dimension. The generalization performance (accuracy for the test data) was also improved by increasing the circuit width and outperformed the classical NN model. The generalization gaps (the differences between a model’s performance on training data and its performance on test data) were 195.231 °C for the classical NN model (5-5-1, 36 parameters) with regularization, and 143.755 °C for 5-qubits QNN model with linear CX (depth = 7, 35 parameters), and 179.328 °C, 154.255 °C for 10-qubits QNN models (depth=3, 30 parameters) with explanatory variables ($x-x$) and ($x-x^2$), respectively. Comparing the model with redundant inputs of the explanatory variable ($x-x$) and the model with redundant inputs ($x-x^2$), the latter appears to perform slightly better. This is because it prevents basis duplication and efficiently handles a large number of basis functions.

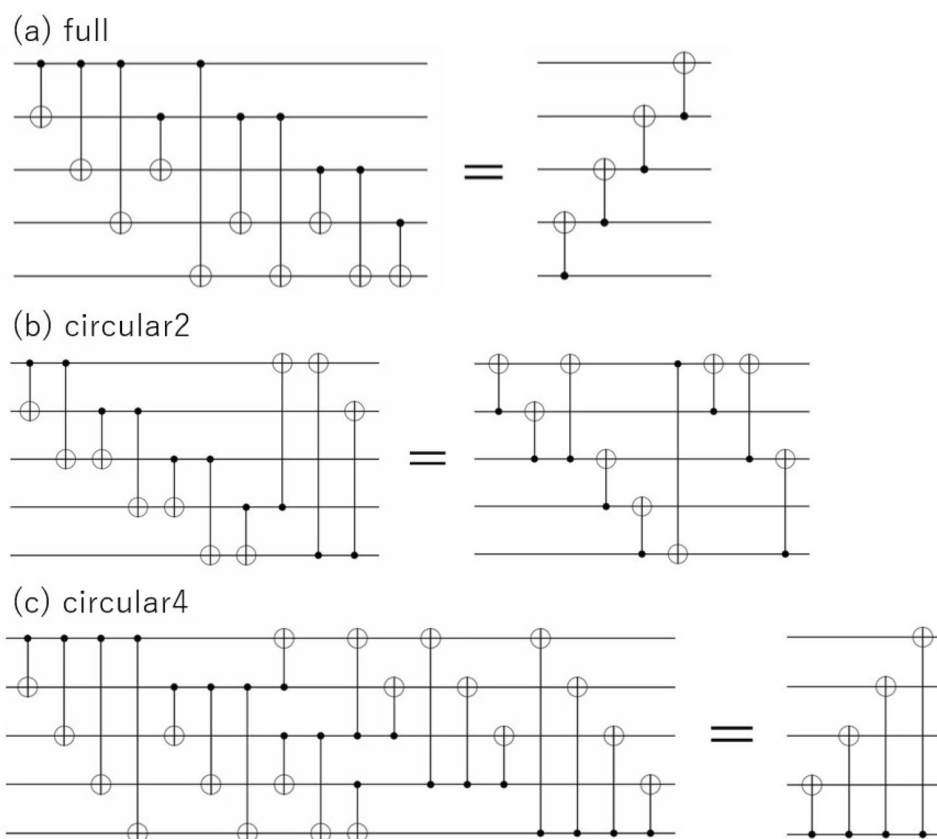


Figure 7. Each entangler and its equivalent reduced circuit.

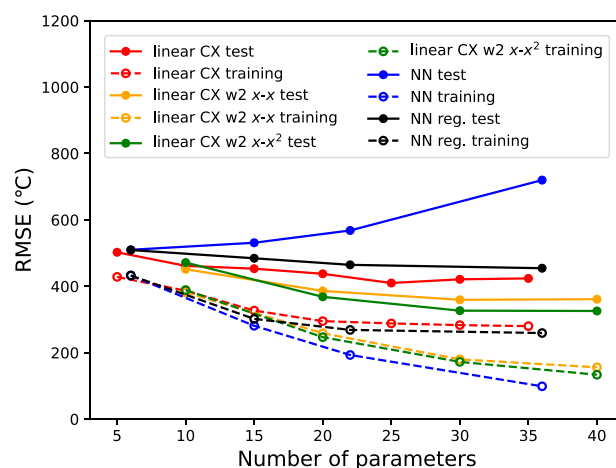


Figure 8. The effect of the number of qubits (circuit width) on the performance of the QNN model.

Conclusion

In this study, we constructed QNN models to predict the melting point of metal oxides by exploring various architectures (encoding methods and entangler arrangements). The explanatory variables should be uniquely converted into rotation angles to obtain good QNN models and avoid overfitting. It was also found that even shallow-depth ansatzs could achieve sufficient expressibility for the present task using sufficiently entangled circuits. It is insufficient to place a large number of CX gates without consideration; it is necessary to set up an entangler that produces entanglements in real terms. In this case, KL divergence and entanglement entropy proved to be good indicators. The “linear” entangler was adequate for providing the necessary entanglement for the QNN model for this particular problem. This result indicates that a model with satisfactory performance can be created using only 2-qubit operations between adjacent qubits. The expressibility of a QNN model can be

improved by increasing the circuit width (number of qubits). This also improved the generalization performance, outperforming the classical NN model. Most importantly, no overfitting was observed in QNN models with well-designed encoders. A QNN can achieve high generalization performance without hyperparameter tuning and is considered an excellent tool for regression tasks.

Data availability

The data that support the findings of this study are available from the corresponding author, H. H., upon reasonable request.

Received: 24 November 2023; Accepted: 9 April 2024

Published online: 13 April 2024

References

- Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559**, 547–555 (2018).
- Schmidt, J., Marques, M. R., Botti, S. & Marques, M. A. Recent advances and applications of machine learning in solid-state materials science. *npj Comput. Mater.* **5**, 83 (2019).
- Ramprasad, R., Batra, R., Pilia, G., Mannodi-Kanakkithodi, A. & Kim, C. Machine learning in materials informatics: Recent applications and prospects. *Npj Comput. Mater.* **3**, 54 (2017).
- Agrawal, A. & Choudhary, A. Perspective: Materials informatics and big data—Realization of the “fourth paradigm” of science in materials science. *Appl. Mater.* **4**, 053208 (2016).
- Rajan, K. Materials informatics. *Mater. Today* **8**, 38–45 (2005).
- Xu, P., Ji, X., Li, M. & Lu, W. Small data machine learning in materials science. *npj Comput. Mater.* **9**, 42 (2023).
- Abbas, A. *et al.* The power of quantum neural networks. *Nat. Comput. Sci.* **1**, 403–409 (2021).
- Mitarai, K., Negoro, M., Kitagawa, M. & Fujii, K. Quantum circuit learning. *Phys. Rev. A* **98**, 032309 (2018).
- Steane, A. Quantum computing. *Rep. Progress Phys.* **61**, 117 (1998).
- Cerezo, M. *et al.* Variational quantum algorithms. *Nat. Rev. Phys.* **3**, 625–644 (2021).
- Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
- Nielsen, M. A. *Neural Networks and Deep Learning* Vol. 25 (Determination Press, 2015).
- Caro, M. C. *et al.* Generalization in quantum machine learning from few training data. *Nat. Commun.* **13**, 4919 (2022).
- Hirai, H. Application of quantum neural network model to a multivariate regression problem. [arXiv:2310.12559](https://arxiv.org/abs/2310.12559) (2023).
- Brockherde, F. *et al.* Bypassing the Kohn–Sham equations with machine learning. *Nat. Commun.* **8**, 872 (2017).
- Faber, F. A. *et al.* Prediction errors of molecular machine learning models lower than hybrid DFT error. *J. Chem. Theory Comput.* **13**, 5255–5264 (2017).
- Tropsha, A. Best practices for QSAR model development, validation, and exploitation. *Mol. Inform.* **29**, 476–488 (2010).
- Sugino, O. & Car, R. Ab initio molecular dynamics study of first-order phase transitions: Melting of silicon. *Phys. Rev. Lett.* **74**, 1823 (1995).
- Puchala, B. & Van der Ven, A. Thermodynamics of the ZR–O system from first-principles calculations. *Phys. Rev. B* **88**, 094108 (2013).
- Karthikeyan, M., Glen, R. C. & Bender, A. General melting point prediction based on a diverse compound data set and artificial neural networks. *J. Chem. Inf. Model.* **45**, 581–590 (2005).
- Ward, L., Agrawal, A., Choudhary, A. & Wolverton, C. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Comput. Mater.* **2**, 1–7 (2016).
- Qu, N. *et al.* Ultra-high temperature ceramics melting temperature prediction via machine learning. *Ceram. Int.* **45**, 18551–18555 (2019).
- Schneider, S. J. *Compilation of Melting Points of the Metal Oxides*. 68 (US Department of Commerce, National Bureau of Standards, 1963).
- Lide, D. R. *CRC Handbook of Chemistry and Physics* Vol. 85 (CRC Press, 2004).
- Coutures, J. & Rand, M. Melting temperatures of refractory oxides—Part II: Lanthanoid sesquioxides. *Pure Appl. Chem.* **61**, 1461–1482 (1989).
- Wang, Y. *et al.* Pubchem: A public information system for analyzing bioactivities of small molecules. *Nucl. Acids Res.* **37**, W623–W633 (2009).
- Jain, A. *et al.* Commentary: The materials project—A materials genome approach to accelerating materials innovation. *APL Mater.* **1**, 011002 (2013).
- Fushiki, T. Estimation of prediction error by using k-fold cross-validation. *Stat. Comput.* **21**, 137–146 (2011).
- Sivarajah, S. *et al.* t ket>: a retargetable compiler for NISQ devices. *Quantum Sci. Technol.* **6**, 014003 (2020).
- Suzuki, Y. *et al.* Qulacs: a fast and versatile quantum circuit simulator for research purpose. *Quantum* **5**, 559 (2021).
- Powell, M. J. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.* **7**, 155–162 (1964).
- Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge University Press, 2010).
- Nakaji, K. & Yamamoto, N. Expressibility of the alternating layered ansatz for quantum computation. *Quantum* **5**, 434 (2021).
- Sim, S., Johnson, P. D. & Aspuru-Guzik, A. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Adv. Quantum Technol.* **2**, 1900070 (2019).
- Imambi, S., Prakash, K. B. & Kanagachidambaresan, G. Pytorch. *Programming with TensorFlow: Solution for Edge Computing Applications* 87–104 (2021).
- Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014).
- Ballarin, M., Mangini, S., Montangero, S., Macchiavello, C. & Mengoni, R. Entanglement entropy production in quantum neural networks. *Quantum* **7**, 1023 (2023).

Author contributions

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

Competing interests

The author declares no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-024-59276-0>.

Correspondence and requests for materials should be addressed to H.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024