Learning Iterative Quantum Feature Maps

Nasa Matsumoto,* Quoc Hoan Tran, Yasuhiro Endo, and Hirotaka Oshima
Quantum Laboratory, Fujitsu Research, Fujitsu Limited, Kawasaki, Kanagawa 211-8588, Japan

Quantum machine learning models that leverage deep quantum circuits as quantum feature maps (QFMs) are theoretically recognized for their superior expressive power in learning tasks. However, the practical application of deep QFMs is significantly hindered by inherent circuit noise and the challenges of implementing them on actual quantum computers. We introduce an architecture employing deep-layer structures connected to QFMs iteratively, facilitating the implementation of shallower quantum circuits. This design bypasses the need to learn the variational parameters of the quantum circuit, focusing instead on optimizing the weights of classical connections. Our approach utilizes contrastive learning and a layer-by-layer training methodology. Numerical experiments conducted with Fashion-MNIST data demonstrate that the test accuracy of a well-designed IQFM can be comparable to that of a similar structure using classical activation functions.

Introduction— In quantum machine learning (QML), the use of quantum feature maps (QFMs) inspired by classical machine learning data mapping strategies has recently become an attractive idea. QFMs employ the dynamics of quantum systems to encode the classical data into quantum states. QFMs emphasize the potential benefits of QML, suggesting a possible exponential speed-up [1] because simulating certain quantum correlations is classically infeasible [2]. It is also possible to design specialized data sets that clearly show the significant differences between quantum and classical models from a learning theory perspective. However, finding similar advantages in practical applications, particularly those feasible for near-term quantum computers, remains challenging.

One of the common strategies is to append variational quantum circuits (VQCs) after the QFM circuit, forming the structure of a quantum neural network. However, learning VQCs may struggle to surpass classical approaches regarding time scaling due to issues like becoming stuck in local minima [3] or encountering plateau regions [4].

We propose a hybrid quantum-classical structure called Iterative Quantum Feature Maps (IQFM) to enhance the expressiveness of QFM. This structure connects to the next feature map after obtaining the measurement output of the QFM and performing classical augmentation on these measurement results [Fig. 1(a)]. The output of the QFM is realized by measuring expectation values of observables with multiple bases or using the measurement records, achieving stable learning in deep layers through normalized classical augmentation. The learning focuses solely on the weights of classical augmentation (located on the classical side) without learning the parameters of the quantum circuit. We employ contrastive learning and a layer-by-layer training mechanism to reduce the quantum resources required for learning [Fig. 1(b)].

Iterative Quantum Feature Map (IQFM)— In the QML framework, a typical QFM includes a feature map circuit $\mathcal{U}_{\Psi(\boldsymbol{h})}$ that maps classical data $\boldsymbol{h}$ into a quantum state in the Hilbert space, coupled with a quantum circuit $\Omega(\boldsymbol{\theta})$, parameterized by $\boldsymbol{\theta}$ for adapting the measurement basis [Fig. 1(a)]. The sequential repetition of the combination of $\mathcal{U}_{\Psi(\boldsymbol{h})}$ and $\Omega(\boldsymbol{\theta})$, with varying parameters $\boldsymbol{\theta}$, is known as the data re-uploading framework. The measurement, termed feature extraction, yields a feature vector $f(\boldsymbol{h}, \boldsymbol{\theta})$ from outcomes such as all qubits measured in the multi-basis.

IQFM employs a deep structure that enhances the connectivity of QFMs by applying classical augmentation to

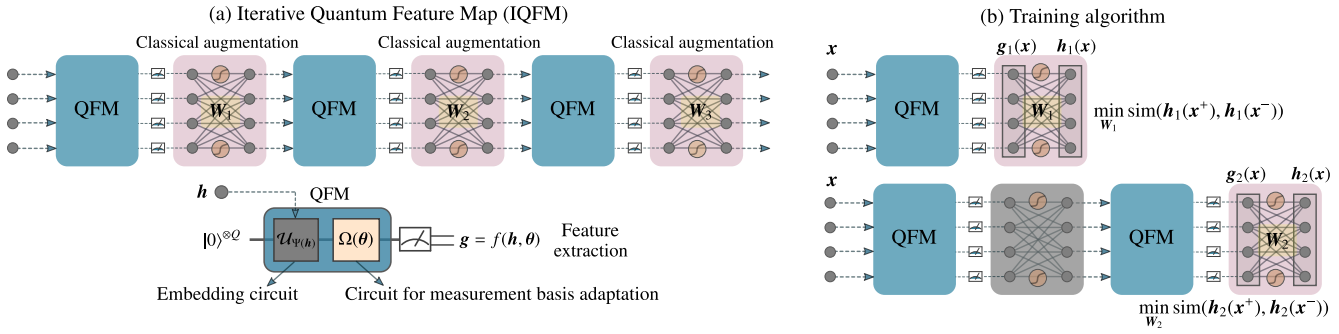

FIG. 1. Overview of Iterative Quantum Feature Map (IQFM) (a) and its training algorithm (b). In this setup, only the weights of classical augmentation layers ($\boldsymbol{W}_1$, $\boldsymbol{W}_2$, ..., $\boldsymbol{W}_L$) are trained. The IQFM training algorithm employs contrastive learning to sequentially train the weights of the classical augmentation layer by layer, leaving quantum circuit parameters untrained.

the measurement outcomes before moving to the subsequent QFM. Given the input sample $\boldsymbol{x}$, IQFM with L layers is constructed as follows:

1. At initial layer $l = 0$, set $\boldsymbol{h}_0(\boldsymbol{x}) = \boldsymbol{x}$.
2. For each $l = 1, \ldots, L$: $\quad \boldsymbol{g}_l(\boldsymbol{x}) = f_l(\boldsymbol{h}_{l-1}(\boldsymbol{x}), \boldsymbol{\theta}_l), \quad \boldsymbol{h}_l(\boldsymbol{x}) = \mathcal{A}_l(\boldsymbol{W}_l, \boldsymbol{g}_l(\boldsymbol{x}))$,

where these equations represent the quantum feature extraction via the lth QFM and the classical augmentation $\mathcal{A}_l$ (such as a classical neural network) at lth layer, respectively. At each layer l, there are two sets of parameters involved: $\boldsymbol{\theta}_l$ for the quantum circuit and $\boldsymbol{W}_l$ for the classical augmentation.

We present the multi-basis measurement approach in the quantum feature extraction. The initial feature vector $\boldsymbol{g}_0$ is derived from the measurement results of all qubits in the Z-basis. To further expand the feature set, measurements are performed in additional bases, denoted by index i, to acquire additional vectors $\boldsymbol{g}_i$. This is achieved by applying a random single-qubit rotation R_i to all qubits after the embedding circuit. The final output feature is a concatenation of the initial feature vector $\boldsymbol{g}_0$ with the additional feature vectors $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_M$, obtained from M different bases. We also intend to develop the feature extraction using the classical shadows to strike a balance between minimizing quantum resource requirements and maintaining the high expressiveness of the framework.

Contrastive Learning Algorithm— In the IQFM framework with L layers, each layer consists of a quantum feature map f_l and a classical augmentation $\mathcal{A}_l$. We assume each augmentation layer has the same number of training parameters, denoted as P; thus, the total parameters for training amount to PL. The output of IQFM can be used to define a loss function, the minimization of which is the objective of training. We use N, S, T, and t_{cir} to represent the number of training data, the number of measurement shots for feature extraction in each layer, the number of training epochs, and the runtime of each QFM circuit per layer, respectively. The gradient-based optimization method requires computing the gradient of the loss function. Since the backpropagation procedure is inapplicable to QFMs, computing the gradient requires a number of IQFM runs proportional to the order of parameters, excluding those in the last augmentation layer. For each run of IQFM with a set of N data, S measurement shots, the required runtime is $N \times L \times S \times t_{\text{cir}}$. Consequently, the total runtime (at the quantum's side) is $O(P \times L^2 \times T \times N \times S \times t_{\text{cir}})$. This presents a challenge to implementing the framework on a large scale, especially when considering a large number of training parameters per layer.

We assume that the computational runtime associated with classical computing can be considered negligible. We present a training algorithm capable of significantly reducing the quantum computational runtime by a factor of P to become $O(L^2 \times T \times N \times S \times t_{\text{cir}})$. Our method trains the classical augmentation component without circuit learning. The training algorithm leverages contrastive learning to sequentially train the weights of the classical augmentation layer by layer, leaving the quantum circuit untrained. For example, in supervised contrastive learning, given an input sample $\boldsymbol{x}$ which includes the original data $\boldsymbol{s}$ and its corresponding label $\boldsymbol{y}$, a positive sample $\boldsymbol{x}^+ = (\boldsymbol{s}, \boldsymbol{y})$ and a negative sample $\boldsymbol{x}^- = (\boldsymbol{s}, \bar{\boldsymbol{y}})$ are generated according to the alignment of $\boldsymbol{s}$ with the true target $\boldsymbol{y}$ and false target $\bar{\boldsymbol{y}}$, respectively. These samples are processed by the QFM at the first layer to yield feature vectors $g_1(\boldsymbol{x}^+)$ and $g_1(\boldsymbol{x}^-)$. Classical augmentation $\mathcal{A}_1$ is then applied to these vectors, and the weight $\boldsymbol{W}_1$ is adjusted to minimize the similarity between augmented vectors $\boldsymbol{h}_1(\boldsymbol{x}^+)$ and $\boldsymbol{h}_1(\boldsymbol{x}^-)$. This procedure continues to subsequent layers, with $\boldsymbol{W}_1$ fixed and $\boldsymbol{W}_2$ targeted for training, and so forth, until the last layer. This process is repeated until a specified number of epochs is reached or termination conditions are met [Fig. 1(b)].

To train the classical augmentation weights $\boldsymbol{W}_l$ for distinguishing between the augmented positive and negative features $\boldsymbol{h}_1(\boldsymbol{x}^+)$ and $\boldsymbol{h}_1(\boldsymbol{x}^-)$, we utilize noise contrastive estimation to address the minimization problem:

$$\boldsymbol{W}_l = \operatorname{argmin} \frac{1}{N} \sum_{\boldsymbol{x}} \mathcal{C}\left(\boldsymbol{h}_l(\boldsymbol{x}^+), \boldsymbol{h}_l(\boldsymbol{x}^-)\right), \tag{1}$$

where the cost function $\mathcal{C}\left(\boldsymbol{h}_l(\boldsymbol{x}^+), \boldsymbol{h}_l(\boldsymbol{x}^-)\right) = \log\left[1 + \exp\left(\frac{\text{cs}(\boldsymbol{h}_l(\boldsymbol{x}^+), \boldsymbol{p}_l) - \text{cs}(\boldsymbol{h}_l(\boldsymbol{x}^-), \boldsymbol{p}_l)}{\tau}\right)\right]$. Here, cs denotes the cosine similarity function, τ is the scale parameter, and $\boldsymbol{p}_l$ represents a random vector fixed for each layer l.

In the testing phase, for a given data $\boldsymbol{s}$, we need to predict the true label for this data. This is achieved by iterating over a set of labels $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K\}$, where each label $\boldsymbol{y}_k$ is used to construct an input sample $\boldsymbol{x}_k = (\boldsymbol{s}, \boldsymbol{y}_k)$. For every $\boldsymbol{x}_k$, we calculate the cumulative similarity score across all layers, represented as $C_k = \sum_l \text{cs}(\boldsymbol{h}_l(\boldsymbol{x}_k), \boldsymbol{p}_l)$. The algorithm then determines the true label k corresponding to the highest cumulative score among $C_1, \ldots, C_K$.

Results— We present benchmark results obtained through simulations using the Fashion-MNIST dataset to demonstrate that IQFM can be learned stably as a deep structure. We consider a random of 5,000 training samples and a

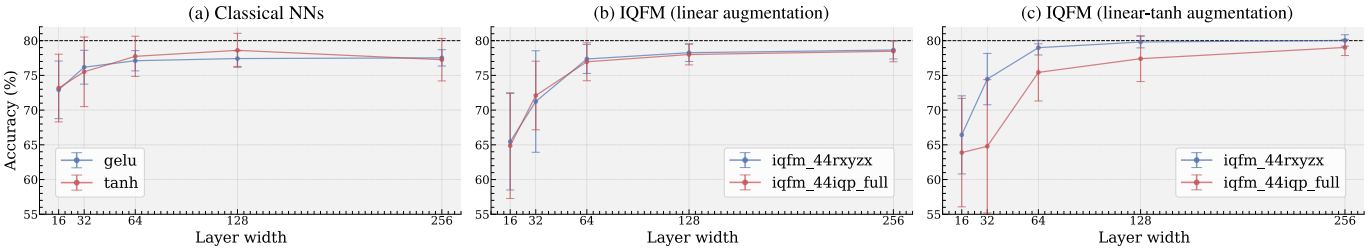

FIG. 2. Test accuracy (over ten experimental runs with ten random parameter initialization) of Classical NNs and IQFMs with different patterns of classical augmentation.

full 10,000 test samples in each experiment run. For classical augmentation, we explore two variants of $\mathcal{A}_l$: a linear augmentation $\mathcal{A}_l(\boldsymbol{W}_l, \boldsymbol{g}_l) = \boldsymbol{g}_l \boldsymbol{W}_l$, and a linear-tanh augmentation $\mathcal{A}_l(\boldsymbol{W}_l, \boldsymbol{g}_l) = \pi \mathbf{tanh}(\boldsymbol{g}_l \boldsymbol{W}_l/2)$, where **tanh** denotes the multi-dimensional hyperbolic tangent function. The latter normalizes the data before its processing through the QFM, potentially improving the model's robustness and performance.

In our QFM design, we construct two variants of an embedding quantum circuit comprising 4 qubits to embed a 16-dimensional vector h: iqfm_44rxyzx and iqfm_44iqp_full. In iqfm_44rxyzx, we apply RX, RY, RZ, and RX gates, which are the rotation operators about the x, y, and z axes, sequentially for each qubit. The rotated angles of these gates correspond to the elements of $\boldsymbol{h}$. In iqfm_44iqp_full, we apply four layers of ZZFeatureMap, which employs ZZ interaction to entangle qubits, capturing the correlations between the features of the input data. In each layer, a Hadamard gate is applied to each of qubit, followed by RZ gates with rotation angles corresponding to every four elements of $\boldsymbol{h}$. Subsequently, ZZ interactions are applied to all pairs of qubits. Measurement of all qubits under the computational basis, namely $\langle Z_1 \rangle, \langle Z_2 \rangle, \langle Z_3 \rangle$, and $\langle Z_4 \rangle$, yield a 4-dimensional feature vector. To provide more informative data for training, we further incorporate measurements of three additional bases. This approach results in a E-dimensional ($E = 16$) feature vector as the embedded output of $\boldsymbol{h}$.

Since each input sample is a gray image with dimension 28×28, we first arrange it into a vector of 784 elements. To ease the numerical simulation, we first apply the input vector with the classical augmentation $\mathcal{A}_0$ to reduce it into a M-dimensional vector, which is then embedded using $\frac{M}{16}$ embedding quantum circuits, requiring a total of $\frac{M}{4}$ qubits. Therefore, after the QFM in the first layer, we obtain the feature vector with the dimension $\frac{M}{16} \times E = M$. Subsequent classical augmentations $\mathcal{A}_l$ and the output of QFMs at layer $l > 0$ do not alter the dimension of the feature vector.

Figure 2 displays the test accuracy of classical NNs and an IQFM with $L = 4$ layers across ten experimental runs with different random parameter initializations. In classical NNs, we do not use quantum circuits to calculate $\boldsymbol{g}_l$, instead we use activation functions such as gelu, tanh. As the layer width (M) increases, there is a tendency for the test accuracy to increase. When the layer width is 256, the test accuracy of IQFM [Fig. 2(b),(c)] is higher than that of classical NNs [Fig. 2(a)]. Also, the highest test accuracy was obtained when using iqfm_44rxyzx with IQMF (linear-tanh augmentation) [Fig. 2(c)]. These results suggest that the choice of the QFM and classical extensions can affect the test accuracy improvements.

Conclusion— We present IQFM architecture that integrates deep-layer structures with QFMs. Utilizing contrastive learning, IQFM eliminates the training of VQCs, instead focusing on the training of classical connection parameters. Numerical experiments conducted with Fashion-MNIST data demonstrate that the test accuracy of a well-designed IQFM can be comparable to that of a similar structure using classical activation functions. Optimizing the design of QFM and classical augmentation will lead to more efficient learning. Furthermore, exploring IQFM's application to diverse tasks and datasets will unlock its full potential.

* matsumoto.nasa@fujitsu.com

[1] Y. Liu, S. Arunachalam, and K. Temme, Nat. Phys. (2021).
[2] S. Bravyi, D. Gosset, and R. König, Science **362**, 308–311 (2018).
[3] E. R. Anschuetz and B. T. Kiani, Nat. Commun. **13**, 7760 (2022).
[4] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Nat. Commun. **9**, 4812 (2018).