Editors' Suggestion

Circuit-centric quantum classifiers

Maria Schuld,[1,*] Alex Bocharov,[2,†] Krysta M. Svore,[2] and Nathan Wiebe[2,3,4]

[1]*University of KwaZulu-Natal, Durban 4001, South Africa*

[2]*Quantum Architectures and Computations Group, Microsoft Research and Microsoft Azure, Redmond, Washington 98052, USA*

[3]*Pacific Northwest National Laboratory, Richland, Washington 98382, USA*

[4]*Department of Physics, University of Washington, Seattle, Washington 98195, USA*

(Received 15 October 2019; accepted 4 February 2020; published 6 March 2020)

Variational quantum circuits are becoming tools of choice in quantum optimization and machine learning. In this paper we investigate a class of variational circuits for the purposes of supervised machine learning. We propose a circuit architecture suitable for predicting class labels of quantumly encoded data via measurements of certain observables. We observe that the required depth of a trainable classification circuit is related to the number of representative principal components of the data distribution. Quantum circuit architectures used in our design are validated by numerical simulation, which shows significant model size reduction compared to classical predictive models. Circuit-based models demonstrate good resilience to noise, which makes then robust and error tolerant.

DOI: 10.1103/PhysRevA.101.032308

I. INTRODUCTION

In the last five years, quantum computing [1] has been seeing a transition from a largely academic discipline to an industrial technology. So-called "intermediate-scale" quantum devices are being developed on a variety of hardware platforms and offer for the first time a test-bed for quantum algorithms [2].

One increasingly popular candidate application for quantum computing is machine learning [3–5]. To make quantum machine learning practical one needs to develop concise resource-sensitive quantum predictive models that are robust against errors [6–8]. A particularly interesting framework uses hybrid quantum-classical algorithms called variational or parameterized quantum circuits [6,9–12], quantum circuits that can be trained with the help of a classical coprocessor. This paper proposes to use a hybrid quantum algorithm as a model for binary classification (see Fig. 1). Given an input x and a set of parameters $\bar{\theta}$, this circuit-centric quantum classifier first embeds x into the physical state of a quantum system and subsequently uses a generic learnable quantum circuit $U(\bar{\theta})$ to compute a predictive model in this "quantum feature space." The predicted class label $y = f(x, \bar{\theta})$ is retrieved by measuring a designated qubit in the state $U(\bar{\theta})|x\rangle$.

In order to study the classification power of a few parametrized gates, we focus on the conceptually most simple strategy of *amplitude encoding*, which in principle allows one to encode feature vectors of dimension up to 2^n when using n qubits. In our circuit designs we draw some inspiration from recent theoretical analyses of classical convolutional neural nets and Boltzmann machines (e.g., [13,14]), in which observe that the capacity of a neural net for processing multirange intradata correlations is related to the amount of entanglement in a quantum many-body wave function. In line with this, our quantum classifiers use entanglement as a key resource.

Beyond proposing design decisions for the quantum classifier, this paper makes four contributions: First, we develop a specific hybrid quantum-classical training algorithm where the gradients of the circuit can be retrieved from running slight variations of the classification algorithm. Since the preprint version of this paper [15] the theory of gradient-based hybrid optimization [11,16] has been further consolidated and calls the type of rule presented here a "parameter shift rule." Second, we note that a quantum classifier is similar to a support vector machine that classifies data using learnable linear transformations in an exponentially large feature space. We observe that in the worst case scenarios the depth of optimal classifier circuit might scale with the number of features. Third, we present numerical evidence suggesting that in practice such worst cases are unlikely and that circuits with a polylogarithmic parameter ansatz perform reasonably well on classical benchmarks. Lastly, we note that due to the unitary nature of the classification circuits, they do not amplify data noise or label noise. Our empirical results support the theoretically motivated conjecture that uncorrelated parameter noise has a moderate effect on the classification error (in practice even smaller than theoretical bounds suggest).

The preprint version of this paper [15] appeared close in time with the much related work in Ref. [17]. We differentiate our classification algorithms by proposing concrete and specific quantum circuit designs, putting a focus on amplitude encoding and keeping the size of our models (measured both in qubits and learnable parameters) polylogarithmic in the dimension of the input and feature space.

II. PRELIMINARIES AND CIRCUIT DESIGN

A. Input embedding

For the classification circuits to work, the data samples from the subject data domain $\mathcal{D}$ must be converted into

*schuld@ukzn.ac.za

†alexeib@microsoft.com

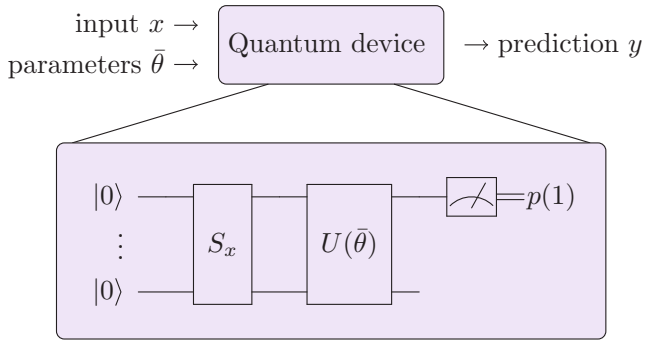

FIG. 1. Design of circuit-centric quantum classifier. Inference with the model $f(x, \bar{\theta}) = y$ uses a quantum device which executes a *state preparation circuit* S_x encoding the input x into the state of a quantum system, a trainable *model circuit* $U(\bar{\theta})$, and a measurement. Repeated measurements resolve a binary probability distribution from which the binary prediction can be computed. The model circuit parameters $\bar{\theta}$ can be learned by optimizing the goal function (3).

quantum states by application of a suitable "quantum" feature map $\varphi : \mathcal{D} \rightarrow H$, where H is the Hilbert space of quantum states.

To allow a focus on the variational classification circuit only, we leave out a discussion of possible data preprocessing strategies that can be integrated into φ [18,19] and instead limit ourselves to minimal preprocessing needed for the quantization of the data. Note that this means that the power of the circuit-centric quantum classifier used in this investigation is severely limited for the sake of conceptual clarity of the study.

Assuming N features, we first consider *simple normalizing state preparation*,

$$(x_1, \ldots, x_N)^T \mapsto \bar{x} = \frac{1}{\chi}(x_1, \ldots, x_N, c_1, \ldots, c_P), \quad (1)$$

where $c_1, \ldots, c_P$ are some padding values that are constant across the data domain and $\chi = \sqrt{\sum_j x_j^2 + \sum_k c_k^2}$. The padding dimension P is selected so that $N + P$ is some power of 2: $N + P = 2^n$.

The L_2-normalized vector $\bar{x}$ can be converted (by standard means, e.g., [20]) into an amplitude-encoded n-qubit state which we denote by $\varphi(x)$. It is notable that the subsequent classification circuit is going to be almost[1] linear in the amplitudes of $\varphi(x)$, but, due to normalization and padding, the $\varphi(x)$ itself is somewhat nonlinear in terms of the original feature values.

We further define *state preparation with fanout*. For this purpose we reserve $d\,n$ qubits for some *register count* d and consider encoding x into a tensor product state of the $d\,n$ qubits $\varphi_1(x) \otimes \cdots \otimes \varphi_d(x)$, where all the φ_ℓ are structured as per Eq. (1) but may each have a different set of padding constants. We observe that the amplitudes of such a d-register state now make a d-linear form in terms of the amplitudes of the φ_ℓ factors, and they are quasipolynomials of degree d in the original feature values (the normalization coefficient $\frac{1}{\prod_\ell \chi_\ell}$ prevents them from being true polynomials). This construct is a quantum analog of polynomial kernel maps and has first been introduced in Ref. [21]. In the following, we limit our study to input embedding without fanout.

B. Likelihood and utility

For simplicity we define here a binary classifier that processes a data set $\mathcal{D} = \{x^1, \ldots, x^M\}$ of inputs $x^m \in \mathbb{R}^N$ that is given together with an effective labeling function $\ell : \mathcal{D} \rightarrow \{\lambda_1, \lambda_2\}$. For brevity we assume that there are exactly two classes with two corresponding class labels $\lambda_1 > \lambda_2$ and that these labels are eigenvalues of a Hermitian operator A, which describes the measurement observable that defines the class inference.

The circuit-based classification model is defined by a parameterized quantum circuit $U(\bar{\theta})$, as shown in Fig. 1. Here $\bar{\theta}$ is the learnable vector of parameters. The learning goal is to maximize the mean likelihood of inferring the correct label for a data sample, which can be written as

$$\frac{1}{M}\left(\sum_{x:\ell(x)=\lambda_1} p[\lambda_1||U(\bar{\theta})\varphi(x)\rangle] + \sum_{x:\ell(x)=\lambda_2} p[\lambda_2||U(\bar{\theta})\varphi(x)\rangle]\right), \quad (2)$$

where $p(\lambda||z\rangle)$ is the probability of measuring λ in the quantum state $|z\rangle$.

Introducing $\mathcal{H}(\bar{\theta}) = U(\bar{\theta})\,A\,U(\bar{\theta})^\dagger$, the maximum likelihood goal can be rewritten as a maximization of

$$\frac{1}{M}\left(\sum_{x:\ell(x)=\lambda_1} \langle\varphi(x)|\mathcal{H}(\bar{\theta})|\varphi(x)\rangle - \sum_{x:\ell(x)=\lambda_2} \langle\varphi(x)|\mathcal{H}(\bar{\theta})|\varphi(x)\rangle\right). \quad (3)$$

We derive the latter equation in the Appendix A.

Here we take $U(\bar{\theta})$ from a pool of rapidly entangling quantum circuits of specific geometry, and we learn an estimate for $\bar{\theta}$ by stochastic gradient descent (SGD).[2]

C. Circuit geometry

In this work we use a specific quantum circuit geometry ansatz, which is motivated by the apparent propensity of strongly entangling networks for capturing multirange correlations in data (cf. [22]). To this end we define the *classification circuit* as a composition of some number of *code blocks*, also known as variational layers:

$$U(\bar{\theta}) = G_{\text{final}}(\bar{\theta}_{\text{final}}) \cdot B_L(\bar{\theta}_L) \cdots B_\ell(\bar{\theta}_\ell) \cdots B_1(\bar{\theta}_1). \quad (4)$$

Here the overall parameter vector $\bar{\theta}$ is the union of parameter subvectors, and each $B_\ell(\bar{\theta}_\ell)$ is a block consisting of $d\,n$ local single-qubit gates, one for each qubit, and $d\,n$ controlled single-qubit gates forming the so-called *cyclic code*. The key hyperparameter of a cyclic code is a control proximity range $0 < r < d\,n$. For any qubit index $j \in [0, d\,n - 1]$, the code block must have one controlled single-qubit gate with the

[1]The measurement introduces a weak nonlinearity.

[2]By nature of SGD and due to the nonconvex optimization landscape, we often end up in a local maximum of the goal function.

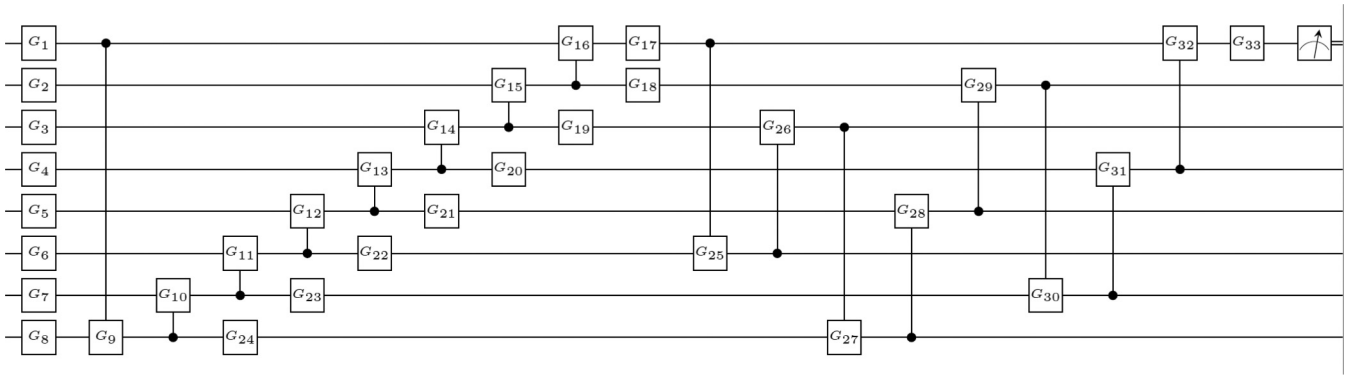

FIG. 2. Rapidly entangling eight-qubit circuit of depth 19, size 33. Horizontal wires correspond to qubits. All G_j, $j = 1, \ldots, 33$ are (parameterized) single-qubit quantum gates with trainable parameters. The $G_1, \ldots, G_8, G_{17}, \ldots, G_{24}$ and G_{33} are referred to as "local gates." The remaining gates form two cycles of *controlled* gates that are known to affect the entanglement entropy of the quantum states they act upon. With the geometry as shown, we learn to infer the class label λ from the probability of obtaining λ as the result of measuring the top qubit.

jth qubit as the target and the qubit with the index $k = (j + r)\mathrm{mod}(\mathrm{d\,n})$ as the control qubit.

Figure 2 depicts an example of this circuit geometry with $d\,n = 8$ and two code blocks: the leftmost with the $r = 1$ and the rightmost with the $r = 3$. In this example we have the single-qubit gate G_{33} as the "final" gate. (It has been discovered empirically that having one trainable single-qubit adjustment before measurement improves the quality of the trained circuit, often significantly).[3]

It can be shown by standard quantum algebra that applying a cyclic code block to a quantum state, under appropriate parameter assignment, can cause a near-maximum increase or decrease of the bipartite entanglement entropy. Therefore, by concatenating a sufficient number of such blocks, we can in principle learn how much entanglement and between what groups of inputs must optimally exist in the classification state before measurement.

D. Universality and circuit size

Since its early days [23] quantum circuit synthesis relied on the fact that single-qubit and two-qubit gates generate the entire n-qubit unitary group $U(2^n)$. As a matter of principle, any unitary weight matrix can be written as a composition of the one- and two-qubit factors. However, more than $O(4^n)$ such factors may be required for exact representation of a unitary target $G \in U(2^n)$ in general position.

Fortunately, for classification purposes we are never required to deal with the exact representations. We observe here that the quality of a classifier circuit $U(\bar{\theta})$ hinges on its ability to bring a certain unitary basis within a good cosine distance from a reference unitary basis.

Using notations laid out in Sec. II B, consider the following *discriminant Hamiltonian*:

$$\rho = \frac{1}{M}\left(\sum_{x:\ell(x)=\lambda_1} |\phi(x)\rangle\langle\phi(x)| - \sum_{x:\ell(x)=\lambda_2} |\phi(x)\rangle\langle\phi(x)|\right),$$

where $\phi(x)$ is a feature encoding of the data sample x. Let V_+ be the subspace of the feature space, spanned by the eigenvectors of ρ with positive eigenvalues.

It can be shown that the maximum likelihood estimate of probability $p(\mathrm{label(z)} = \lambda_1|\mathcal{D})$ is the square of cosine similarity between $|z\rangle$ and V_+. In particular, any $v \in V_+$, when considered as a (synthetic) data vector, is assigned class label λ_1 with probability 1.

Our classification scheme is based on probabilities of the outcome λ_j, $j = 1, 2$ when measuring a chosen observable A. Let us denote by W_j, $j = 1, 2$ the span of the eigenvectors of A with the corresponding eigenvalues. In this context an ideal classification circuit U should bring the cosine similarity between $U(V_+)$ and W_1 arbitrarily close to 1. In the worst case this would require $O[\dim(\mathrm{V}_+)]$ quantum gates.

We note that the existence of a classification circuit U of size $k \ll \dim(\mathrm{V}_+)$ is implicitly equivalent to *classification by dimensionality reduction* in the following sense:

Definition 1. [(k, δ) reduction] Given data set $\mathcal{D}$ allows classification by (k, δ) *reduction* for integer k and $\delta > 0$ if there exists a k-dimensional subspace $V^{(k)}$ of the feature space such that the class label of any sample $x \in \mathcal{D}$ can be inferred from its projection into $V^{(k)}$ with probability at least $1 - \delta$.

Observation 1. If a data set $\mathcal{D}$ allows classification by (k, δ) *reduction*, then there exists a quantum circuit U with $O(k\,n)$ single- and dual-qubit gates such that for any $x \in \mathcal{D}$ the correct class label of x can be inferred from measurement of the observable A in the state $U|x\rangle$ with probability at least $1 - O(\delta)$.

Both Definition 1 and Observation 1 are nonconstructive, and their direct effective verification for a given data set is likely to incur costs that scale polynomially with the size of the data set and the number of features. In our classification circuit ansatz we operate with circuits that scale polylogarithmically in these variables, that is, we implicitly assume that the data set allows for (k, δ) reductions, where k is a polynomial in n. Clearly this limits the applicability of our ansatz; however, data sets with this reduction property are fairly common, and our numerical simulations using established classical benchmarks suggest that the benchmark data sets are indeed of this nature.

[3]It is easy to see that having "final" single-qubit gates on any of the nonmeasured qubits does not affect the measurement results or probabilities whatsoever.

III. TRAINING AND TESTING

We use a standard stochastic gradient descent for training. However, when quantum hardware is concerned, an additional protocol is needed since there is no "classical" access to its derivative. We will show how to use the quantum model itself to extract estimates of the analytical gradients (see Fig. 3). Similar approaches, but for different gate representations, have been proposed in Refs. [11,17,24]. A generalized theory of quantum gradients [16], partially based on this work, has recently given rise to the PENNYLANE software framework for hybrid optimization [25].

A. Circuit parameterization

The quantum state prepared by the classification circuit is subsequently measured out, and by the Born rule the probabilities of alternative measurement outcomes are fully defined by absolute values of the amplitudes of the final state. Therefore the two states that differ only by some set of phase shifts $\mathrm{diag}(e^{i\alpha_0},\ldots,e^{i\alpha_{dn-1}})$, $\alpha_j \in \mathbb{R}$, $j = 0,\ldots,(dn-1)$ would yield identical classification distributions. Due to this equivalence and within our circuit ansatz, any of our circuits can be equivalently viewed as a collection of three-parameter gates.

Namely, it is sufficient to consider single-qubit gates (both "local" and in controlled position) of the form

$$G(\alpha,\beta,\gamma) = \begin{pmatrix} e^{i\beta}\cos(\alpha) & e^{i\gamma}\sin(\alpha) \\ -e^{-i\gamma}\sin(\alpha) & e^{-i\beta}\cos(\alpha) \end{pmatrix}, \qquad (5)$$

where $\alpha,\beta,\gamma \in [0,\pi]$.

Following our choice of circuit geometry (Sec. II C), the parameterized classification circuit is represented as a structured product of the single-qubit gates of the form $G(\alpha_j,\beta_j,\gamma_j)$ and controlled gates of the form $C_{c_k}[G(\alpha_k,\beta_k,\gamma_k)]$, where c_k is an appropriately chosen control qubit index. We recall, for completeness, that for some single-qubit quantum gate G the $C_c(G)$ applies gate G to the target qubit q_t if and only iff the control qubit q_c is in the state $|1\rangle$. In the two-qubit register $[q_c,q_t]$ this can be alternatively written in bra-ket notation as $|0\rangle\langle 0|\otimes I + |1\rangle\langle 1|\otimes G$. Assuming L code blocks the circuit of the width dn will have $2dnL+1$ three-parameter gates and hence $6dnL+3$ trainable parameters.

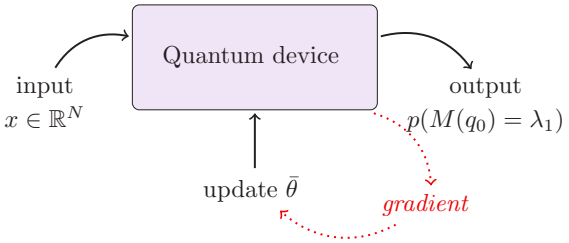

FIG. 3. Idea of the hybrid training method. The quantum device is used to compute outputs and gradients of the model in order to update the parameters for each step of the gradient descent training.

B. Reading out the prediction

After executing the quantum classification circuit with the proposed parameter vector $\bar{\theta}$, the measurement of the first qubit results in λ_1 with probability $p = p(M(q_0)=\lambda_1|x;\bar{\theta})$. To estimate p we have to run the entire circuit S times and measure the first qubit. The number of runs grows with the precision of the estimation ϵ roughly as $1/\epsilon^2$ as in any other Bernoulli parameter estimation problem. The classical postprocessing consists of adding a learnable bias term b to produce the continuous output of the model, $\pi(x;\bar{\theta};b) = p(M(q_0)=\lambda_1|x;\bar{\theta})+b$. Our simulations confirmed the importance of the bias parameter. Thresholding the value finally yields the binary output that is the overall prediction of the model.

C. Hybrid gradient descent scheme

We learn the optimal parametrization of the classification circuit (4) by maximizing the goal function given by Eq. (3).

Note that it is possible to add a regularization term to this goal function. Instead, we propose a quantum version of *dropout regularization* that randomly selects and measures one of the qubits and sets it aside for a certain number N_{dropout} of optimization steps. After that, the qubit is readded to the circuit and another qubit is randomly dropped.

A partial derivative of the objective (3) with respect to a certain circuit parameter $\mu \in \bar{\theta}$ is a sum of terms of the form $\pm\partial_\mu\langle x|\mathcal{H}(\bar{\theta})|x\rangle$, where the derivative of each individual term can be rewritten as

$$\pm 2\,\Re\langle\partial_\mu U(\bar{\theta})x|A|U(\bar{\theta})x\rangle.$$

As per our ansatz (Sec. II C), $U(\bar{\theta}) = \prod_{k=1}^{K} G_k(\bar{\theta}_k)$, where each G_k is an elementary gate such as (3) or its controlled version. Therefore, using the shorthand $G_j(\bar{\theta}_j) = G_j$,

$$\partial_\mu U(\bar{\theta}) = \sum_{j=1}^{K}\left[\left(\prod_{k=1}^{j-1} G_k\right)\partial_\mu(G_j\left(\prod_{k=j+1}^{K} G_k\right)\right]. \qquad (6)$$

The partial derivatives of elementary gate $G(\alpha,\beta,\gamma)$ [Eq. (5)] with respect to a $\mu \in [\alpha,\beta,\gamma]$ are computed analytically, yielding a gate of the same type. Thus for $G = G(\alpha,\beta,\gamma)$: $\partial_\alpha G = G(\alpha+\pi/2,\beta,\gamma)$; $\partial_\beta G = \cos(\alpha)G(0,\beta+\pi/2,0)$; $\partial_\gamma G = \sin(\alpha)G(\pi/2,0,\gamma+\pi/2)$.

Finally, a partial derivative of a controlled single-qubit gate is no longer a unitary operator, but it can be represented as a linear combination of two unitary operators in the following manner:

$$\begin{aligned}\partial_\mu(|0\rangle\langle 0|\otimes I + |1\rangle\langle 1|\otimes G) &= |1\rangle\langle 1|\otimes(\partial_\mu G)\\ &= \tfrac{1}{2}[I\otimes(\partial_\mu G) - \sigma_z\otimes(\partial_\mu G)].\end{aligned}$$

Applying these properties to Eq. (6), we conclude that the $\partial_\mu U(\bar{\theta})$ can be represented as a certain linear combination of $O(K)$ unitary circuits of the same architecture. Note for clarity that our gradient estimator does not attempt to represent the derivative in Eq. (6) by a single unitary circuit. Instead, we run a circuit for each of the terms in (6) separately and collect the sum of the corresponding overlap terms by classical postprocessing.

Obtaining classical estimates for the partial derivatives is thus reduced to the core task of estimating the *overlap terms* of the form $\Re\langle V(\bar{\theta})x|A|U(\bar{\theta})x\rangle$, where $V(\bar{\theta})$ is a minor variation of the circuit $U(\bar{\theta})$. This latter core task can be achieved using a known interference circuit construction that requires only one additional ancillary qubit. (A version of the Hadamard test is described, for example, in Ref. [4], Sec. 6.1.1.2). The complexity of the gradient estimation is roughly $O(K)$ times higher than the complexity of computing a class label, and the complexity overhead factor scales with the size of the circuit.

IV. SIMULATION RESULTS

Here we present some evidence that one can analyze hypothesis classes using circuits with depths that are significantly smaller than the dimension of the feature space. Quantum machine learning models are notoriously hard to simulate and benchmark; therefore the simulations are limited to relatively small data sets. The main purpose of this initial set of results is further numerical experiments with the models of the proposed type. All simulations were done on a classical computer using a Microsoft Language-integrated quantum simulator [26].

A. Data sets

We selected four standard benchmarking data sets from the UCI repository (see Table I): CANCER, SONAR, WINE, SEMEION, and also the MNIST handwritten digits data set from [27]. While CANCER and SONAR are binary classification exercises, the other data sets call for multiclass classification. Although the variational quantum classifier could be operated as a multiclass classifier, we limit ourselves to the case of the binary classification discussed above and cast the multilabel tasks as a set of "one-versus-all" binary discrimination subtasks.

For all experiments, the data has been preprocessed for the needs of the quantum algorithm. The MNIST data vectors were course grained into real-valued data vectors of dimension 256.

B. Benchmark models

For the circuit-centric quantum classifier (QC) we use the data-agnostic entangling circuit of n qubits, which has been explained in Sec. II C. We use one, two, or three entangling blocks in our experiments. For each data set we selected the circuit architecture with the lowest training error while reducing overfitting.[4]

TABLE I. Benchmark data sets and preprocessing. N is the input dimension and M the number of samples. MNIST has been coarse grained and deskewed.

ID	N	CLASSES	M	PREPROCESSING
CANCER	32	2	569	none
SONAR	60	2	208	pad
WINE	13	3	178	pad
SEMEION	256	10	1593	none
MNIST256	256	10	2766	simplify

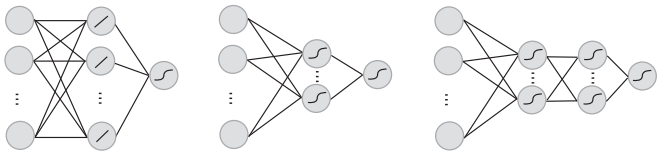

FIG. 4. The three architectures of the benchmark artificial neural network models. To the left is the MLPl model, which has a hidden linear layer of the size of the input N and a logistic output layer. The MLPs model in the middle has a hidden layer of size $\lceil \log_2 N \rceil$ with nonlinear activations and a logistic output layer. The MLPd model model on the right adds a second nonlinear hidden layer.

Without the use of more advanced feature maps, our classifier has only limited flexibility, and benchmarking against state-of-the-art models such as convolutional neural networks therefore will not provide much insight. Instead, we choose to compare our model to six classical benchmark models that were selected for their mathematical structure, which is related to the quantum classifier. Among the classical benchmarks are three different neural network architectures (MLPl, MLPs, MLPd shown in Fig. 4), a support vector machine with polynomial kernel of two different degrees (SVMPoly1 with degree $d = 1$ and SVMPoly2 with degree $d = 2$), as well as a perceptron model (PERC). Since one of our goals is to build a model with a small parameter space, we compare the number of trainable parameters for each model in Table II.

C. Results

For every benchmarking test (except for the QC model on SEMEION and MNIST256) we use fivefold cross-validation with ten repetitions. Due to the significant cost of quantum circuit simulations, for the QC experiments on the SEMEION and MNIST256 data sets only one repetition of the fivefold cross-validation was carried out. Results are summarized in Table III.

As follows from the training error of the PERC model, none of the benchmarks had linearly separable data classes. One finds that the QC model performs significantly better than the SVMpoly1 and SVMpoly2 models across all data sets. In further simulations we verified that for support vector machines with polynomial kernel, degrees of $d = 6$ to $d = 8$ return the best results on the data sets, which are also better than those of the MLP models. Although showing slightly worse test errors than the MLPs and MLPd (and with mixed success compared to the MLPl), the QC performs comparably with the MLP models that train a lot more parameters. For the

[4]Model selection included a heuristic search over range parameters r, as well as trimming the number of gates. The block counts and r parameters used in the benchmark experiments were as follows: WINE – one full block with $r = 1$; SONAR – two full blocks with $r = 1$ and $r = 5$ and one trimmed block; SEMEION – two full blocks with $r = 1$ and $r = 7$; CANCER – three full blocks with $r = 1, 2, 4$ and one trimmed block; MNIST – three full blocks with $r = 1, 3, 7$ and one trimmed block.

TABLE II. Number of parameters each model has to train for the different benchmark data sets. The quantum classifier (QC) has a logarithmic growth in the number of parameters with the input dimension N and data set size M, while all other models show a linear or polynomial parameter growth in either N or M. For the MLP and PERC models, the number of parameters is the amount of weights, while the SVM optimizes one parameter per data sample. The parameter count of the QC model depends on the number of full and trimmed blocks used.

ID	QC	PERC	MLPL	MLPs	MLPD	SVM
CANCER	79	32	1056	165	190	208
SONAR	60	60	1952	305	330	569
WINE	28	13	272	51	60	178
SEMEION	100	256	65792	2056	2120	1593
MNIST256	124	256	65792	2056	2120	800

SONAR and WINE data set we find that the QC model overfits the training data. The observation is interesting, since the QC model is "slimmer" than the MLP and SVM models in terms of its parameter count. The dropout qubit regularization does not eliminate the overfitting. Thus finding a better quantum regularization requires further work.

D. Resilience to noise

An important feature of the circuit-centric classifier is its robustness to noise in the inputs and parameters. Suppose $\delta > 0$ is some small value and we allow parameter perturbations (resp. input perturbations) such that for each constituent gate G the permuted gate G' is δ close to $G : ||G - G'|| < \delta$. To introduce noise, a gate parameter φ is replaced by a perturbed input φ' that is δ close to φ. We allow certain imprecisions in some or all parameter values and that such imprecisions are bounded below some constant δ. Since all the constituent operations are unitary, the impact of the parameter imprecisions is never amplified across the circuit at the defect imposed by the imperfect circuit on the final state before the measurement is bounded by $3L\delta$ in the worst theoretical case, where L is the number of elementary parameterized gates which have at most three parameters.

In practice the propagated error should be much smaller that this bound. The same analysis applies to imperfections in the quantum gates execution (other than parameter drift). There is no amplification of defects across the circuit, and the imperfection of the final state is bounded by the sum of imperfections of individual gates. Finally, the amplitude encoding of the input data does not have to be perfect either. A possible imperfection or approximation during the state preparation will not be amplified by the classification circuit, and the drift of the classifier state will be never be greater than the drift of the initial state. Another widely advertised advantage of variational quantum algorithms is that they can learn to counterbalance systematic errors in the device architecture—for example, when one gate always over-rotates the state by the same value. In our simulation experiments we have systematically evaluated the effects on the quality of the classification of 0.1%, 1%, and 10% random perturbations in (a) the circuit parameters and (b) the input data vectors. As expected due to the unitary properties of quantum mechanics, the effect of input noise is not amplified by the classifier circuit and thus had proportionate impact on the percentage of misclassifications. Somewhat more surprisingly, random perturbations of the trained circuit parameters almost never had the worst case estimated impact on the classification error. The 0.1% uncorrelated parameter noise in the majority of cases had no impact on the classification results. Here we limit the noise impact discussion to the context of the SEMEION and MNIST256 data sets of the benchmark sets displayed in Table III. Our observations are easier to calibrate in this context, since both data sets are encoded with eight qubits and the same model circuit architecture with 33 gates at depth 19 (as shown in Fig. 2) is used in the classifier. The 0.1% parameter noise had no impact on classification in about 60% of our test runs. The maximum relative drift of the test error has been 3.5% (in one the of remaining runs), and the mean drift has been 1% with the standard deviation of approximately 1.47%. The 1% parameter noise had a more pronounced, albeit fairly robust impact, which was nontrivial in about 90% of our test runs. The maximum relative change in the test error rate has been 17%, and the mean relative change has been 7% with the standard deviation of approximately 6.67%. These statistics are summarized in Table IV. Finally, the 10% parameter noise leads to significant loss of classification robustness (although still smaller than the worst case analysis suggests). The maximum relative change in the test error rate has been 192.3%, and the mean

TABLE III. Results of the benchmarking experiments. The cells are of the format "training error/validation error." The variance between the 50 repetitions for each experiment was of the order of 0.01–0.001 for the training and test error. The value of the best classifier for each dataset is printed in bold.

	CANCER	SONAR	WINE[a]	SEMEION[a]	MNIST256[a]
QC	0.022/**0.058**	0.000/0.195	0.000/**0.028**	0.031/0.031	0.031/0.033
PERC	0.128/0.137	0.283/0.315	0.067/0.134	0.022/0.038	0.065/0.066
MLPl	0.060/0.075	0.117/0.263	0.001/0.039	0.001/0.025	0.038/0.041
MLPs	0.064/0.077	0.010/**0.174**	0.029/0.063	0.002/**0.024**	0.011/**0.018**
MLPd	0.056/0.076	0.001/**0.174**	0.010/0.063	0.001/0.026	0.014/0.021
SVMpoly1	0.373/0.367	0.452/0.477	0.430/0.466	0.101/0.100	0.092/0.092
SVMpoly2	0.169/0.169	0.334/0.383	0.090/0.099	0.100/0.101	0.091/0.092
Average	0.125/0.136	0.171/0.283	0.090/0.137	0.037/0.049	0.040/0.043

[a]For multilabel classification problems with d labels, the average of all d one-versus-all problems train and test errors were taken.

TABLE IV. Relative impact (RI) of uncorrelated parameter noise on the classification test error over SEMEION and MNIST256 data using the generic eight-qubit model circuit displayed in Fig. 2.

NOISE LEVEL	RI MEAN	RI ST.DEV
0.1%	1%	1.47%
1%	7%	6.67%
10%	60.2%	55.8%

has been 60.2% with the standard deviation of 55.8%. This suggests that 10% perturbation of parameters has no stable amplification pattern, and the model should best be retrained after such perturbation. The practical takeaway from these observations is that the circuit-centric classifiers may work on small quantum computers even in the absence of strong quantum error correction.

V. CONCLUSION AND FUTURE WORK

We have developed a machine learning design which is both quantum ready and implementable on near-term intermediate-scale quantum devices. The key building block of this design is a unitary model circuit with relatively few trainable parameters that systematically use the entangling properties of quantum circuits as a resource for capturing correlations in the data. After state preparation, the prediction of the model is computed by applying only a small number of one- and two-qubit quantum gates. At the same time, simulating these gates on a classical computer requires computational resources that scale with the number of features. We understand this investigation as a starting point to research that considers quantum-hardware-inspired machine learning models. We also expect our designs and algorithms to be directly applicable to classification of quantum states produced by actual quantum devices without intermediate classical sampling.

ACKNOWLEDGMENT

The authors are grateful to Jeongwan Haah and Martin Roetteler for useful discussions and insightful comments.

APPENDIX: DERIVATION OF THE UTILITY FUNCTION FOR A CIRCUIT-CENTRIC CLASSIFIER

To recall, the mean likelihood of inferring the correct label (by measurement) of given data and given a candidate classifier circuit is

$$\frac{1}{M}\left(\sum_{x:\ell(x)=\lambda_1} p[\lambda_1||U(\bar{\theta})\varphi(x)\rangle] + \sum_{x:\ell(x)=\lambda_2} p[\lambda_2||U(\bar{\theta})\varphi(x)\rangle]\right),$$

where $p(\lambda||z\rangle)$ is the probability of measuring λ in the quantum state $|z\rangle$.

Consider the spectral decomposition of the observable A being measured:

$$A = \lambda_1\Pi_{\lambda_1} + \lambda_2\Pi_{\lambda_2},$$

where Π_{λ_j}, $j = 1, 2$ is the projector onto the corresponding eigenspace.

In this notation the mean likelihood above is literally rewritten as

$$\frac{1}{M}\left(\sum_{x:\ell(x)=\lambda_1} \langle U(\bar{\theta})\varphi(x)|\Pi_{\lambda_1}|U(\bar{\theta})\varphi(x)\rangle + \sum_{x:\ell(x)=\lambda_2} \langle U(\bar{\theta})\varphi(x)|\Pi_{\lambda_2}|U(\bar{\theta})\varphi(x)\rangle\right). \quad \text{(A1)}$$

Since $\Pi_{\lambda_1} + \Pi_{\lambda_2} = I$ we derive, however, that

$$\Pi_{\lambda_1} = \frac{1}{\lambda_1 - \lambda_2}(A - \lambda_2 I), \quad \Pi_{\lambda_2} = \frac{1}{\lambda_2 - \lambda_1}(A - \lambda_1 I).$$

Substituting into the equation above we obtain

$$\frac{1}{M(\lambda_1 - \lambda_2)}\left(\sum_{x:\ell(x)=\lambda_1} \langle U(\bar{\theta})\varphi(x)|A|U(\bar{\theta})\varphi(x)\rangle - \sum_{x:\ell(x)=\lambda_2} \langle U(\bar{\theta})\varphi(x)|A|U(\bar{\theta})\varphi(x)\rangle\right) + \text{constant}.$$

Recalling that we have chosen $\lambda_1 > \lambda_2$, we conclude that the maximization of the mean likelihood is equivalent to maximization of the utility (3).

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2010).

[2] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[3] A. Perdomo-Ortiz, M. Benedetti, J. Realpe-Gómez, and R. Biswas, Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers, Quantum Sci. Technol. **3**, 030502 (2018).

[4] M. Schuld and F. Petruccione, *Supervised Learning with Quantum Computers* (Springer, New York, 2018), Vol. 17.

[5] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[6] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

[7] M. J. Bremner, R. Jozsa, and D. J. Shepherd, Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy, Proc. R. Soc. London A **467**, 459 (2010).

[8] J. Huh, G. G. Guerreschi, B. Peropadre, J. R. McClean, and A. Aspuru-Guzik, Boson sampling for molecular vibronic spectra, Nat. Photonics **9**, 615 (2015).

[9] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, New J. Phys. **18**, 023023 (2016).

[10] J. Romero, J. P. Olson, and A. Aspuru-Guzik, Quantum autoencoders for efficient compression of quantum data, Quantum Sci. Technol. **2**, 045001 (2017).

[11] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).
[12] G. Verdon, M. Broughton, and J. Biamonte, A quantum algorithm to train neural networks using low-depth circuits, arXiv:1712.05304.
[13] Y. Levine, D. Yakira, N. Cohen, and A. Shashua, Deep learning and quantum entanglement: Fundamental connections with implications to network design, arXiv:1704.01552.
[14] D.-L. Deng, X. Li, and S. D. Sarma, Quantum Entanglement in Neural Network States, Phys. Rev. X **7**, 021021 (2017).
[15] M. Schuld, A. Bocharov, K. Svore, and N. Wiebe, Circuit-centric quantum classifiers, arXiv:1804.00633.
[16] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A **99**, 032331 (2019).
[17] E. Farhi and H. Neven, Classification with quantum neural networks on near term processors, arXiv:1802.06002.
[18] M. Schuld and N. Killoran, Quantum Machine Learning in Feature Hilbert Spaces, Phys. Rev. Lett. **122**, 040504 (2019).
[19] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature (London) **567**, 209 (2019).
[20] M. Plesch and Časlav Brukner, Quantum-state preparation with universal gate decompositions, Phys. Rev. A **83**, 032302 (2011).
[21] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum Support Vector Machine for Big Data Classification, Phys. Rev. Lett. **113**, 130503 (2014).
[22] E. Stoudenmire and D. J. Schwab, Supervised learning with tensor networks, in *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, MA, 2016), pp. 4799–4807.
[23] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, Phys. Rev. A **52**, 3457 (1995).
[24] G. G. Guerreschi and M. Smelyanskiy, Practical optimization for hybrid quantum-classical algorithms, arXiv:1701.01450.
[25] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, and N. Killoran, Pennylane: Automatic differentiation of hybrid quantum-classical computations, arXiv:1811.04968.
[26] D. Wecker and K. M. Svore, Liqui| >: A software design architecture and domain-specific language for quantum computing, arXiv:1402.4467.
[27] Y. LeCun, The MNIST database of handwritten digits, http://yann.lecun.com/exdb/mnist/ (1998).