Learning the N-input Parity Function with a Single Qubit and Single Measurement Sampling

Antonia Tsili *, Georgios Maragkopoulos, Aikaterini Mandilara, Dimitris Syvridis

Posted Date: 29 January 2025

doi: 10.20944/preprints202501.2126.v1

Keywords: Quantum Variational Classifiers; Parity function; Doubly Stochastic Gradient Descent

 doi:10.20944/preprints202501.2126.v1

Article

Learning the N-input Parity Function with a Single Qubit and Single Measurement Sampling

Antonia Tsili [1,2,*], Georgios Maragkopoulos [1,2], Aikaterini Mandilara [1,2] and Dimitris Syvridis [1,2]

[1] Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Panepistimioupolis, Ilisia, 15784, Greece
[2] Eulambia Advanced Technologies, Agiou Ioannou 24, Building Complex C, Ag. Paraskevi, 15342, Greece
* Correspondence: tonyts@di.uoa.gr

Abstract: The parity problem, a generalization of the XOR problem to higher-dimensional inputs, is a challenging benchmark for evaluating learning algorithms, due to its increased complexity as the number of dimensions of the feature space grows. In this work, a single-qubit classifier is developed, which can efficiently learn the parity function from input data. Despite the qubit model's simplicity, the solution landscape created in the context of the parity problem offers an attractive test bed for exploring optimization methods for quantum classifiers. We propose a new optimization method called Ensemble Stochastic Gradient Descent (ESGD), with which density matrices describing batches of quantum states are incorporated into the loss function. We demonstrate that ESGD outperforms both Gradient Descent and Stochastic Gradient Descent given the aforementioned problem. Additionally, we show that applying ESGD with only one measurement per data input does not lead to any performance degradation. Our findings not only highlight the potential of a single qubit model, but also offer valuable insights into the use of density matrices for optimization. Further to this, we complement the outcome with interesting results arising by the employment of a doubly stochastic gradient descent for training quantum variational circuits.

Keywords: quantum variational classifiers; parity function; doubly stochastic gradient descent

1. Introduction: The Parity Problem

Representing the XOR function is a fundamental problem for neural networks (NNs), as it involves solving a nonlinearly separable classification problem. Unlike other problems represented by logical functions with two-bit inputs, the XOR problem requires the use of two layers of neurons [1]. The parity function extends the XOR function from 2 to N inputs, forming a proportionally challenging binary classification problem for 2^N inputs, an illustration of which in the 3-dimensional space ($N = 3$) is depicted in Figure 1. According to the universal approximation theorem, any function, including the parity function, can be represented by a deep NN. However, it is always preferable to employ an NN architecture with minimal cost in terms of the number of synapses, neurons, and layers. Another key consideration is the morphology of the solution landscape, as well as the representational capacity and extrapolation capabilities of the chosen NN model. An optimal NN architecture using perceptrons with one hidden layer of size N is described in [2,3], while a modular NN architecture is presented in [3], and a single product unit approach is discussed in [4].

 doi:10.20944/preprints202501.2126.v1

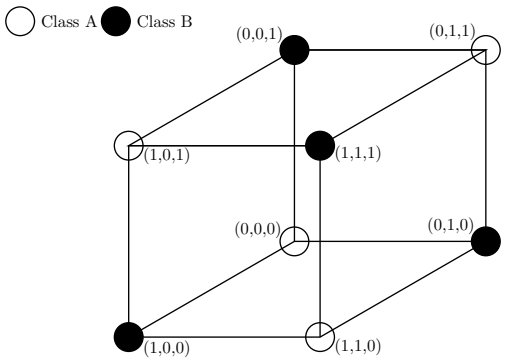

Figure 1. The binary classification problem related to the Parity function with 3-bit input ($N = 3$).

Variational Quantum Circuits (VQCs) [5–7] combine the properties of quantum circuits and classical methods to solve classification tasks in a hybrid way. In particular, quantum circuits perform the data processing, which encompasses data encoding, label extraction, etc., whereas optimization is handled by classical –typically local optimization– algorithms. The data flow follows a sequential combination of the processes, meaning that data collected via measurements on the output quantum state of the VQC are subsequently exploited to perform the optimization task. The purpose of this hybrid formation is to optimally harvest the best of both worlds in an attempt to create new efficient learning methods. However, quantum measurement results are intrinsically stochastic, presenting a challenge in developing suitable optimization algorithms [8].

In this work, our aim is to employ an optimized VQC model in place of an NN and demonstrate that it can be efficiently trained to solve the N-bit parity problem. It is widely known all boolean logical functions can be represented by a quantum circuit[9]. Considering this, one can easily construct a quantum Boolean parity function using N qubits as input and an ancilla qubit to register the output, as shown in Figure 2. While the problem of computing the parity of an oracle function has been explored in literature, it has been shown [10] that at least $N/2$ oracle queries are required, offering no quantum advantage. Finally, another class of parity problems has been used to evaluate the effectiveness of different VQC models in [11], further highlighting the importance of finding an efficient way of solving the parity problem.

The remainder of the manuscript is divided so as to showcase two main concepts; the first concerns the presentation of the quantum variational model for solving the parity problem; the second follows the methodology for training this model. More specifically, in Section 2, we propose a variational Single-Qubit Classifier and investigate the solution landscape for the parity problem. In Section 3, we propose and test a new stochastic gradient method, to the best of our knowledge, for which batches are represented by density matrices. This method can be transformed into a doubly stochastic approach if single stochastic measurement outcomes are considered and averaged over the batches. We present results from numerical tests on both methods, demonstrating the effectiveness of the approach.

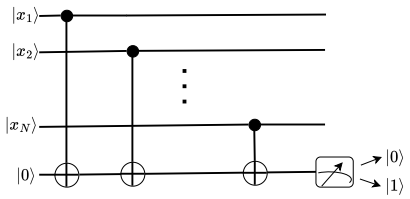

Figure 2. A solution to the N-bit parity problem using a quantum circuit. N qubits encode the Boolean input $x_1, x_2, \ldots, x_N$, and N CNOT gates act on an ancilla qubit, which registers the output.

 doi:10.20944/preprints202501.2126.v1

2. A Single-Qubit Classifier For The Parity Problem

Given a series of N bits, $\vec{x} = x_1, x_2, \ldots, x_N$, where $x_i \in \{0,1\}$, the parity function determines whether the value "1" appears an even or odd number of times in the series. The parity problem can, thus, be considered a classification problem, where the $n = 2^N$ data are divided into two equally populated classes: class A (even parity) and class B (odd parity), as shown in Figure 1. This problem can also be extended to real numbers, i.e., noisy data $\vec{x}$, generated by adding Gaussian noise $\mathcal{N}(\mu, \sigma^2)$ to the integer values of $\vec{x}$.

Following the logic of [12], where a general model of a variational qudit classifier was introduced, we specify the following model for the parity problem:

$$\exp\left[-i\sum_{k=1}^{3} s_k \pi \hat{\sigma}_k\right] \exp\left[-i\left(\sum_{j=1}^{N} w_j x_j\right)\pi\hat{\sigma}_1\right]|0\rangle. \tag{1}$$

We will refer to this model as the Single-Qubit Classifier (SQC). In Equation (1), $\hat{\sigma}_1, \hat{\sigma}_2, \hat{\sigma}_3$ represent the Pauli operators, $|0\rangle$ is the eigenstate of $\hat{\sigma}_3$ with eigenvalue 1, and $\vec{s} = (s_1, s_2, s_3)$, $\vec{w} = (w_1, w_2, .., w_N)$, are the variational parameters, which are optimized during training. The training procedure updates the variational parameters given data points $\vec{x}_i = (x_{i1}, x_{i2}, ..., x_{iN})$, with each data point labeled as $y_j = 1$ for even or $y_j = -1$ for odd parity. The classes are, hence, represented by labels that can be directly matched to the mean measurement outcome of operator $\hat{\sigma}_3$, $\langle\hat{\sigma}_3\rangle_j$, where positive is classified as even parity and negative is classified as odd parity.

2.1. Landscape Of Solution

The SQC in Equation (1) is inspired by the quantum digital circuit shown in Figure 2, the relation between the two can be directly derived. For noiseless data, a periodic (global) solution emerges, allowing the SQC to classify all inputs with perfect accuracy when $\bar{w}_j = 1/2 + l$ and $s_k = 0$ for all j and k, with $l \in \mathbb{Z}$. In this solution, the SQC reproduces the conditional evolution of the ancillary qubit in Figure 2: an X gate is applied whenever $x_j = 1$. An even number of gate applications results in the final state $|0\rangle$, while an odd number of applications leads to the state $|1\rangle$. A measurement of the observable $\hat{\sigma}_3$ at the end of the variational circuit can perfectly distinguish the two cases, resulting in the accurate classification of the input.

Even if the parameters converge to a solution with some deviation δw around the global solution $\bar{w}$, the model classifies noiseless data successfully. Correspondingly, if the parameters are fixed at $\bar{w}$, the classifier can tolerate noise of the same order in the data. We observe that, for $N = 2$, the classifier can tolerate a maximum deviation $\delta w_{max} \sim 1/8$, or equivalently $\sigma_{max} \sim 1/24$. For N inputs, the total noise in the exponent of Equation (1) increases by $\sqrt{N}\sigma$, meaning that the tolerable noise for each individual input scales with $\sigma_{max} \sim 1/24\sqrt{N}$. Finally, the second rotation of the SQC's state – represented by the second term of Equation (1) – is added, so that the model can tolerate recurrent additive errors in the form of $\mathcal{N}(\mu \neq 0, \sigma^2)$ in the input data. In the remainder of this work, we will ignore this term and set $\vec{s} = \mathbf{0}$ in Equation (1).

We initially employ the square loss function $\mathcal{L}(\vec{w}) = \sum_{j=1}^{N}(y_j - \langle\hat{\sigma}_3\rangle_j)^2$ for evaluating the result of the learning procedure of the VQC, however, we observe that no local minima appear in this case. The solution landscape for $N = 2$ can be seen in Figure 3 (a) over two periods. Since illustrations of the landscape's morphology can only be extracted as cross-sections when $N > 2$, we plot an example for $N = 2, 6$, and 10, where the cross-section is applied along the principal hyper-diagonal, namely, where $x_1 = x_2 = \ldots = x_N = z$, shown in Figure 3 (b). We notice that, as N increases, the number of local minima along the diagonal increases moderately. Nonetheless, the existence of local minima persists if $z \to 1 - z$ for some inputs. This suggests that the number of minima within a period increases at least linearly with N, while the global solution remains unique.

On the other hand, our numerical investigation shows that input sets which include a subset of points whose coordinates coincide with that of the global solution and, simultaneously, the rest of the data points have coordinates of local minima, also correspond to local minima of the loss function.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 January 2025 doi:10.20944/preprints202501.2126.v1

Reaching the aforementioned minima will also yield a perfect classification of the input. Based on the analysis of numerical results in the following section, we conclude that both local minima that achieve accuracy 1, meaning that all points are correctly classified, and those that do not, increase in number proportionally as N grows. A final observation on Figure 3 (b), supported by our numerical investigations, is that as N increases, the area in the parametric space covered by barren plateaus also increases. This makes the approach to a local minimum more time-consuming or, in some cases, impossible.

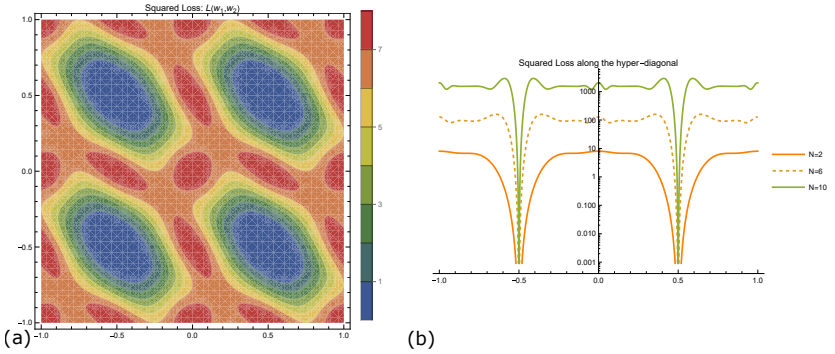

Figure 3. The square loss function $\mathcal{L}(\vec{w})$ for $N = 2$. *(a)* Contour plot versus the parameters w_1 and w_2, Equation (1), over two periods. *(b)* Plot along the hyperdiagonal $x_1 = x_2 = \ldots = x_N$ for $N = 2,\ 6,\ 10$.

3. Learning Methods

The versatile landscape of the parametric space, combined with the fact that there is at least one known solution, renders the parity problem an attractive test-bed for investigating different optimization techniques for the training of SQV. We will, first, provide a description of the common steps involved in training a variational quantum classifier, followed by an introduction to the basic optimization technique for adjusting the weights, namely the Gradient Descent (GD) method. It is essential to gain a deeper understanding of the solution landscape through the application of GD before more advanced optimization techniques can be explored.

Let us restrict the discussion to the case of binary classification tasks, where the data $\vec{x}_j$ are to be classified into two classes, A and B, consisting of N_A and N_B points respectively, where $N_A + N_B = n$. We denote the output circuit state by $|\psi(\vec{w}, \vec{x}_j)\rangle$ and assume that a measurement on a single observable $\hat{G}$ (specifically, $\hat{\sigma}_3$ in our case), which has two eigenvalues ± 1, is performed on the quantum state. For fixed values of the parameters $\vec{w}$, the standard procedure involves running the quantum circuit multiple times to estimate

$$\begin{aligned} g(\vec{w}, \vec{x}_j) = &\quad \langle\psi(\vec{w}, \vec{x}_j)|\, \hat{G}\, |\psi(\vec{w}, \vec{x}_j)\rangle \\ \approx \tilde{g}_M(\vec{w}, \vec{x}_j) = &\quad \tfrac{1}{M} \textstyle\sum_{m=1}^{M} G_m(\vec{w}, \vec{x}_j) \end{aligned} \tag{2}$$

where $G_m(\vec{w}, \vec{x}_j) \in \{-1, +1\}$ represents the stochastic outcome of the m-th (independent) experiment. For $M \gg 1$, we have $\tilde{g}_M(\vec{w}, \vec{x}_j) \to g(\vec{w}, \vec{x}_j)$.

The step of estimating $g(\vec{w}, \vec{x}_j)$ via the quantum circuit and the corresponding measurements is followed by a classical procedure to update the parameters $\vec{w}$ using a classical optimization method. The first step in this procedure is to construct a loss function $\mathcal{L}(\vec{w})$ based on $g(\vec{w}, \vec{x}_j)$ and the information about the classes. Here, we use the negative log-likelihood loss, which is commonly employed to learn the weights in NNs. If the classes A and B – corresponding to even and odd parity in our case study, respectively – are labeled as $y = \pm 1$, the probability that the training data $\vec{x}_i$ belongs to the class y can be evaluated as follows:

$$P(y|\vec{x}_j) = \frac{y g(\vec{w}, \vec{x}_j) + 1}{2}. \tag{3}$$

 doi:10.20944/preprints202501.2126.v1

By defining the class label y_j for each input $\vec{x}_j$, the loss function can be constructed by summing the negative log-likelihood of the probabilities as follows:

$$\mathcal{L}(\vec{w}) = -\frac{1}{n}\sum_{j=1}^{n} log\big(P(y_j|\vec{x}_j)\big) \tag{4}$$

or,

$$\mathcal{L}(\vec{w}) = -\frac{1}{N_A}\sum_{j\in A} log\Big(P\Big(y=1|\vec{x}_j^A\Big)\Big) - \frac{1}{N_B}\sum_{j\in B} log\Big(P\Big(y=-1|\vec{x}_j^B\Big)\Big) \tag{5}$$

Using GD, the weights are updated according to

$$\vec{w}_{t+1} = \vec{w}_t - \alpha\vec{\nabla}\mathcal{L}(\vec{w}_t) \tag{6}$$

where α is the learning rate. The numerical differentiation that produces the gradient $\mathcal{L}(\vec{w})$ of the loss function requires at least $2N$ evaluations of g, defined in Equation (2), to be estimated by the quantum circuit, where N is the dimension of each input data vector. The analytical formulas [13,14] of the partial derivatives of the gradient vector can potentially relate these derivatives to the mean values of other observables in the quantum state, offering an improvement in the precision of the optimization procedure. In our case, where the model is relatively simple, the analytic formula is straightforward and involves the mean value of only one observable:

$$\frac{\partial g(\vec{w},\vec{x}_i)}{\partial w_j} = 2x_j^{(i)}\pi\,\langle\psi(\vec{w},\vec{x}_i)|\,\hat{\sigma}_2\,|\psi(\vec{w},\vec{x})\rangle = 2x_j^{(i)}\pi h(\vec{w},\vec{x}_i) \tag{7}$$

where $x_j^{(i)}$ denotes the jth component of the vector $\vec{x}_i$. Using the chain rule and Equation (7), one can accurately evaluate the elements of the gradient $\frac{\partial\mathcal{L}(\vec{w})}{\partial w_j}$.

For the parity problem with $N = 6,\ldots,10$-bit input, we apply the GD method to optimize the parameters of the SQC, as given by Equation (1). We use a constant learning rate α and a fixed number of epochs, and we employ the analytical formula in Equation (7) for the gradient. We report two important metrics concerning the classification results; the average accuracy $A = (\#\text{inputs correctly classified})/(\#\text{all inputs})$, listed in Table 1, as well as the percentage of times that accuracy reached 1, i.e., where all inputs were correctly classified after training. The metrics are reported over 50 runs where the weight vector is randomly initialized for each run.

Table 1. Application of the GD method to the parity problem using the SQC model. N: the dimension of the input for the parity problem. α: the learning rate, as defined in Equation (7). $\langle\mathbf{A}\rangle$: the average accuracy over 50 runs with random initializations for $\vec{w}$. $\mathbf{\Delta}\langle\mathbf{A}\rangle$: the standard deviation of the accuracy. $\%\mathbf{A}=\mathbf{1}$: the percentage of runs where accuracy reached 1. For all runs, the number of epochs was set to 200.

N	6	7	8	9	10
α	3×10^{-3}	7.8×10^{-4}	2.9×10^{-4}	1.2×10^{-4}	7.3×10^{-5}
$\langle\mathbf{A}\rangle$	0.97	0.97	0.8	0.7	0.6
$\mathbf{\Delta}\langle\mathbf{A}\rangle$	0.10	0.11	0.3	0.2	0.2
$\%\mathbf{A}=\mathbf{1}$	0.96	0.94	0.6	0.34	0.16

The results of the GD optimization, shown in Table 1, as well as the evolution of the observed loss during optimization, lead to the following conclusions. The difficulty of the problem increases rapidly with the input dimension N, as expected, highlighting the need for more fine-tuned optimization methods. For $N = 6,7,8$, the few occurrences of unsuccessful training are mainly due to the optimization method being trapped in the small neighborhood around local minima. However, for $N \geq 9$, the presence of barren plateaus becomes equally problematic and the loss function appears to oscillate randomly over these regions.

 doi:10.20944/preprints202501.2126.v1

3.1. The ESGD Method: Towards An Optimization With Single Measurement Sampling

Inspired by recent studies [8,15,16] on exploiting the intrinsic stochasticity of quantum measurements in gradient optimization methods and, at the same time, motivated by the need to reduce quantum resources, we explore whether the SQC can be trained using just one measurement (one repetition of the quantum circuit) for a very rough estimate of $g(\vec{w},\vec{x})$ and $\frac{\partial g(\vec{w},\vec{x})}{\partial w_j}$. We begin by justifying this approach through the introduction of the ESGD method and then test the use of $\tilde{g}_{M=1}(\vec{w},\vec{x}_j)=\pm 1$ in the loss function. It is clear that some averaging must be involved to induce smoothness in the loss function, and intuition suggests that this average should be taken over different input data $\vec{x}_j$. In the following, we present a logical procedure to justify this averaging, while keeping the negative log-likelihood as the loss function, contrary to previous works.

We will now provide a brief description of the Stochastic Gradient Descent (SGD) optimization method. With SGD, n data points are randomly split into k batches, each batch containing $n_k = n/k$ points. During an epoch, the loss function, in our case given by Equation (8), and its derivatives are estimated k times.

$$\mathcal{L}^{(k)}(\vec{w}) = -\frac{1}{n_k}\sum_{j=(k-1)n_k+1}^{kn_k+1} log\left(P(y_j|\vec{x}_j)\right) \tag{8}$$

Focusing, now, on the specific conditions of the parity problem, where $N_1 = N_2 = n/2$, we introduce a "balanced" cost functional with an equal number of points ($n_k/2$) from each class $\vec{x}_j^A$ and $\vec{x}_j^B$ in each batch:

$$\begin{aligned}\mathcal{L}^{(k)}(\vec{w}) &= -\tfrac{2}{n_k}\textstyle\sum_{j=(k-1)n_k/2+1}^{kn_k/2}\log\left(P\left(y=1|\vec{x}_j^A\right)\right)\\ &\quad -\tfrac{2}{n_k}\textstyle\sum_{j=(k-1)n_k/2+1}^{kn_k/2}\log\left(P\left(y=-1|\vec{x}_j^B\right)\right).\end{aligned} \tag{9}$$

However, quantum mechanics offers an additional perspective, which has been partially explored in a different context in [17]. Let us construct $2k$ density matrices from the batches of states as follows:

$$\hat{\rho}_{+1}^k = \tfrac{2}{n_k}\textstyle\sum_{j=(k-1)n_k/2+1}^{kn_k/2}|\psi(\vec{w},\vec{x}_j^A)\rangle\langle\psi(\vec{w},\vec{x}_j^A)| \tag{10}$$

$$\hat{\rho}_{-1}^k = \tfrac{2}{n_k}\textstyle\sum_{j=(k-1)n_k/2+1}^{kn_k/2}|\psi(\vec{w},\vec{x}_j^B)\rangle\langle\psi(\vec{w},\vec{x}_j^B)|. \tag{11}$$

For the kth batch one can estimate

$$F(\vec{w},\vec{x}_{j\in k}^{A,B}) = Tr\left(\hat{\rho}_{\pm 1}^k\hat{G}\right) = \frac{2}{n_k}\sum_{j=(k-1)n_k/2+1}^{kn_k/2} g(\vec{w},\vec{x}_j^{A,B}), \tag{12}$$

and then define $P_k(\pm 1|\vec{x}_j) = \frac{\pm 1 F(\vec{w},\vec{x}_{j\in k}^{A,B})+1}{2}$ and finally the loss function

$$\tilde{\mathcal{L}}(\vec{w})^{(k)} = -\log\left(P_k\left(y=1|\vec{x}_j^A\right)\right) - \log\left(P_k\left(y=-1|\vec{x}_j^B\right)\right). \tag{13}$$

This method, which we shall refer to as Ensemble Stochastic Gradient Descent (ESGD), has a geometric interpretation. If a Bloch vector is assigned to each of the density matrices from Equations (10)-(11), then this gradient descent method guides the ensemble of Bloch vectors associated with each class to point in opposite directions on the Bloch sphere ($\pm\hat{z}$). The ESGD method is derived on solid foundations, since it is closely related to SGD, as described in the next. The

 doi:10.20944/preprints202501.2126.v1

first requires at least $n_k = 2$ input points to populate each batch, since it includes the evaluation of Equations (10) and (11). Considering, hence, the base case $n_k = 2$, we have that

$$\hat{\rho}^k_{+1} = |\psi(\vec{w}, \vec{x}^A_j)\rangle \langle \psi(\vec{w}, \vec{x}^A_j)| \tag{14}$$

$$\hat{\rho}^k_{-1} = |\psi(\vec{w}, \vec{x}^B_j)\rangle \langle \psi(\vec{w}, \vec{x}^B_j)| \tag{15}$$

and the label assignment becomes

$$F(\vec{w}, \vec{x}^{A,B}_{j\in k}) = Tr\left(\hat{\rho}^k_{\pm 1}\hat{G}\right) = g(\vec{w}, \vec{x}^{A,B}_j), \tag{16}$$

reducing to the SGD base case. Finally, as with the standard GD, the components of the gradient can be calculated analytically using Equations (13) and (7).

Now, we will proceed to describe a modification of ESGD, according to which we consider a single measurement per input $\vec{x}_j$ for calculating either numerically, or analytically using the parametric shift rule [14], the gradient. We will refer to this as Doubly Stochastic Gradient Descent (DSGD), since it incorporates both batch processing and the stochasticity from single measurements. The convergence of DSGD to ESGD becomes clearer with the following formula, as the number of measurements M increases:

$$\begin{aligned} F(\vec{w}, \vec{x}^{A,B}_{j\in k}) &= Tr\left(\hat{\rho}^k_{\pm 1}\hat{G}\right) \approx \tilde{F}_M(\vec{w}, \vec{x}^{A,B}_{j\in k}) \\ &= \frac{2}{Mn_k}\sum_{j=(k-1)n_k/2+1}^{kn_k/2}\sum_{m=1}^{M} G_m(\vec{w}, \vec{x}^{A,B}_{j\in k}). \end{aligned} \tag{17}$$

One may modify ESGD using any finite M to estimate $\tilde{F}_M(\vec{w}, \vec{x}^{A,B}_{j\in k})$, then $\tilde{P}_k(\pm 1|\vec{x}_j) = \left(\pm 1\tilde{F}_M(\vec{w}, \vec{x}^{A,B}_{j\in k}) + 1\right)/2$ and finally the loss function in Equation (13). In this work, however, we exhibit the power of the method for $M = 1$.

We employ the SGD, ESGD, and DSGD optimization methods and conduct classification experiments on instances of the parity problem for an input dimension of $N = 10$. We chose this dimension to illustrate the results, as it marks the point where GD starts to fail and because it allows for varying batch sizes. Our results, based on 25 experiments with random weight vector initialization, are presented in Tables 2-4. From these tables, it is evident that both ESGD and DSGD outperform SGD for all batch sizes. Specifically, the mean accuracy of ESGD and its stochastic counterpart, DSGD, increases as the batch size grows, whereas SGD experiences significant performance degradation, as shown in Figure 4. SGD eventually converges to the performance of GD (see Table 1 for $N = 10$). The close relationship between SGD and ESGD for $n_k = 2$ is also apparent, with both showing poor performance due to the continuous, uncorrelated treatment of the data, which fails to collectively guide the weight vector toward a solution. In contrast, ESGD consistently achieves high accuracy, close to 100% for any $n_k \geq 32$. DSGD, being stochastic, does not easily achieve 100% accuracy, as it tends to oscillate even once average convergence is reached.

Some important technical details about the implementation of the methods are as follows: The random weight vector initialization differs for each point in Figure 4. The performance of the methods is naturally influenced by the choice of hyperparameters, such as the learning rate and the number of epochs. For all three methods and each batch size, we optimized the learning rate to achieve the best outcomes. The number of function evaluations is kept constant at 1024, meaning the number of epochs increases with n_k. Note that the $N = 2,\ 4$ cases are not included in Table 4 due to the fact that the doubly stochastic derivation introduces excessive noise, preventing the model parameters from converging to a solution vector. Finally, in the DSGD method, to achieve smoother evolution, we average over the last four steps when evaluating the gradient vector.

 doi:10.20944/preprints202501.2126.v1

Table 2. Results of SGD optimizations for the parity problem with $N = 10$. The results concern 25 runs, each with random initial $\vec{w}$ and analytical derivatives. Notation is explained as follows. N: the dimension of the input for the parity problem. $\langle \mathbf{A} \rangle$: the average accuracy $\mathbf{\Delta}\langle \mathbf{A} \rangle$: the standard deviation of the accuracy. $\%\mathbf{A} = \mathbf{1}$: the percentage of runs where accuracy reached 1. For all runs, the number of function evaluations was set to 1024.

$\mathbf{n_k}$	2	4	8	16	32	64	128	256	512
# epocs	2	4	8	16	32	64	128	256	512
$\langle \mathbf{A} \rangle$	0.7	0.7	0.6	0.6	0.7	0.58	0.56	0.50	0.55
$\Delta\langle \mathbf{A} \rangle$	0.23	0.23	0.2	0.2	0.2	0.18	0.17	0.02	0.15
$\%\mathbf{A} = \mathbf{1}$	0.24	0.28	0.28	0.24	0.32	0.16	0.12	0.0	0.10

Table 3. Results of ESGD optimizations for the parity problem with $N = 10$. The results concern 25 runs, each with random initial $\vec{w}$ and numerical derivatives. Notation is explained as follows. N: the dimension of the input for the parity problem. $\langle \mathbf{A} \rangle$: the average accuracy $\mathbf{\Delta}\langle \mathbf{A} \rangle$: the standard deviation of the accuracy. $\%\mathbf{A} = \mathbf{1}$: the percentage of runs where accuracy reached 1. For all runs, the number of function evaluations was set to 1024.

$\mathbf{n_k}$	2	4	8	16	32	64	128	256	512
# epocs	2	4	8	16	32	64	128	256	512
$\langle \mathbf{A} \rangle$	0.70	0.86	0.97	0.96	0.997	0.998	0.998	0.97	0.98
$\Delta\langle \mathbf{A} \rangle$	0.25	0.23	0.10	0.14	0.008	0.004	0.006	0.10	0.10
$\%\mathbf{A} = \mathbf{1}$	0.40	0.72	0.84	0.72	0.96	0.96	0.92	0.84	0.96

Table 4. Results of DSGD optimizations, namely using a single-measurement $(M = 1)$, for the parity problem with $N = 10$. The results concern 25 runs, each with random initial $\vec{w}$ and numerical derivatives. Notation is explained as follows. N: the dimension of the input for the parity problem. $\langle \mathbf{A} \rangle$: the average accuracy $\mathbf{\Delta}\langle \mathbf{A} \rangle$: the standard deviation of the accuracy. $\%\mathbf{A} = \mathbf{1}$: the percentage of runs where accuracy reached 1. For all runs, the number of function evaluations was set to 1024.

$\mathbf{n_k}$	8	16	32	64	128	256	512
# epocs	8	16	32	64	128	256	512
$\langle \mathbf{A} \rangle$	0.75	0.91	0.967	0.98	0.98	0.988	0.999
$\Delta\langle \mathbf{A} \rangle$	0.25	0.19	0.10	0.01	0.016	0.016	0.004
$\%\mathbf{A} = \mathbf{1}$	0.44	0.76	0.60	0.32	0.44	0.64	0.96

 doi:10.20944/preprints202501.2126.v1

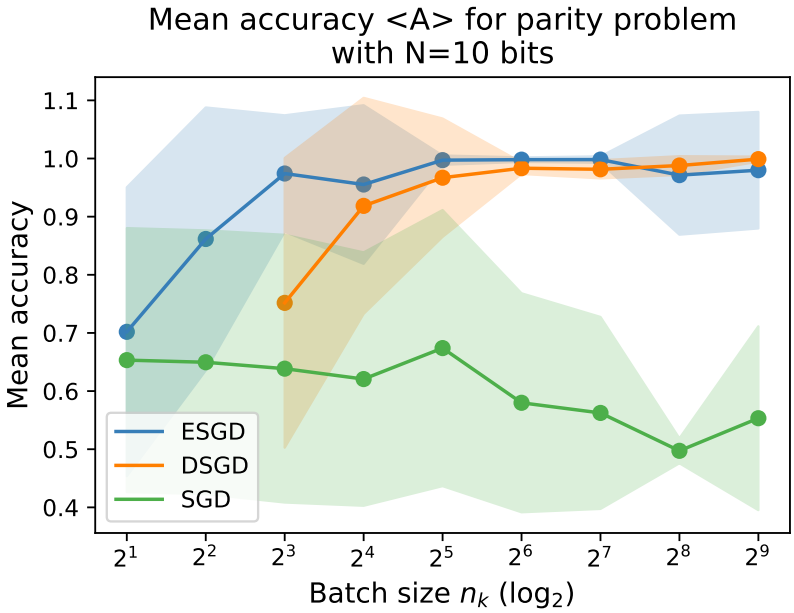

Figure 4. The mean accuracy $\langle \mathbf{A} \rangle$ evolution with respect to the size n_k of the batches for each method, ESGD, DSGD and SGD. The batch size is measured as the number of data points used for each loss and gradient calculation. The x-axis showing the size of the batch is given in the logarithmic scale with base 2. DSGD evaluations start with $n_k = 8$.

4. Conclusions

This work presents two key contributions that advance both the theoretical understanding and practical application of quantum machine learning techniques. First, we demonstrate that the N-bit parity problem can be efficiently solved using a single qubit, highlighting the remarkable potential of qubits as powerful learning units in quantum circuits. This result underscores the fact that any Boolean logical function, including parity, can be implemented within a quantum framework. Our preliminary explorations of other logical functions, such as AND and OR, suggest that this approach may not be universally applicable to all N-input Boolean functions, thus motivating further exploration into the use of qudits. This opens an exciting avenue for future work, where the exploration of quantum systems beyond qubits can be further investigated.

The second significant contribution of this work is the introduction of a novel optimization method, ESGD, which utilizes batches represented by density matrices. Our numerical investigations show that ESGD outperforms traditional SGD in terms of accuracy and efficiency, particularly for the parity problem. Furthermore, ESGD maintains its effectiveness even when reduced to the extreme scenario of using a single measurement per input, a result that highlights its robustness in noisy environments. While our findings are promising, further validation across different problems and for larger quantum systems is required to fully assess the broader applicability and impact of ESGD.

In conclusion, this work makes substantial progress in applying quantum circuits to machine learning tasks, demonstrating the potential of quantum-based optimization methods and offering a new direction for future research in quantum-enhanced learning algorithms.

Author Contributions: Conceptualization, A.M. and A.T.; methodology, A. M. and G.M.; formal analysis, A.T. and A.M.; writing—original draft preparation, A.M., A.T. and G. M. ; writing—review and editing, A.M., A.T., G. M. and D.S.; supervision, D.S.; funding acquisition, D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the project Hellas QCI co-funded by the European Union under the Digital Europe Programme grant agreement No.101091504. A.M. acknowledges partial support from the

 doi:10.20944/preprints202501.2126.v1

European Union's Horizon Europe research and innovation program under grant agreement No.101092766 (ALLEGRO Project).

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: The programs (GD, SGD, ESGD, DSGD methods) are written with Wolfram Mathematica and are available upon request to the corresponding author. No new data were generated for this work.

Acknowledgments: A.M. is grateful to Uwe Jaekel and Babette Dellen for helpful discussions.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ESGD	Ensemble Stochastic Gradient Descent
NN	Neural Network
VQC	Variational Quantum Circuit
SQC	Single-Qubit Classifier
GD	Gradient Descent
SGD	Stochastic Gradient Descent
DSGD	Doubly Stochastic Gradient Descent

References

1. Theodoridis, S.; Koutroumbas, K. *Pattern Recognition, Fourth Edition*; Academic Press, 2009.
2. Fung, H.K.; Li, L.K. Minimal Feedforward Parity Networks Using Threshold Gates. *Neural Computation* **2001**, *13*, 319–326. https://doi.org/10.1162/089976601300014556.
3. Franco, L.; Cannas, S. Generalization properties of modular networks: Implementing the parity function. *IEEE Transactions on Neural Networks* **2001**, *12*, 1306–1313. https://doi.org/10.1109/72.963767.
4. Leerink, L.R.; Giles, C.L.; Horne, B.G.; Jabri, M.A. Learning with product units. In Proceedings of the Proceedings of the 8th International Conference on Neural Information Processing Systems, Cambridge, MA, USA, 1994; NIPS'94, p. 537–544.
5. Havlíček, V.; Córcoles, A.D.; Temme, K.; Harrow, A.W.; Kandala, A.; Chow, J.M.; Gambetta, J.M. Supervised learning with quantum-enhanced feature spaces. *Nature* **2019**, *567*, 209–212. https://doi.org/10.1038/s41586-019-0980-2.
6. Schuld, M.; Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* **2019**, *122*, 040504. https://doi.org/10.1103/PhysRevLett.122.040504.
7. Schuld, M.; Bocharov, A.; Svore, K.M.; Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev. A* **2020**, *101*, 032308. https://doi.org/10.1103/PhysRevA.101.032308.
8. Sweke, R.; Wilde, F.; Meyer, J.; Schuld, M.; Faehrmann, P.K.; Meynard-Piganeau, B.; Eisert, J. Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum* **2020**, *4*, 314. https://doi.org/10.22331/q-2020-08-31-314.
9. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*; Cambridge University Press, 2010.
10. Farhi, E.; Goldstone, J.; Gutmann, S.; Sipser, M. Limit on the Speed of Quantum Computation in Determining Parity. *Phys. Rev. Lett.* **1998**, *81*, 5442–5444. https://doi.org/10.1103/PhysRevLett.81.5442.
11. Jerbi, S.; Fiderer, L.J.; Poulsen Nautrup, H.; Kübler, J.M.; Briegel, H.J.; Dunjko, V. Quantum machine learning beyond kernel methods. *Nature Communications* **2023**, *14*, 517. https://doi.org/10.1038/s41467-023-36159-y.
12. Mandilara, A.; Dellen, B.; Jaekel, U.; Valtinos, T.; Syvridis, D. Classification of data with a qudit, a geometric approach. *Quantum Machine Intelligence* **2024**, *6*, 17. https://doi.org/10.1007/s42484-024-00146-3.
13. Crooks, G.E. Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition. *arXiv: Quantum Physics* **2019**.
14. Schuld, M.; Bergholm, V.; Gogolin, C.; Izaac, J.A.; Killoran, N. Evaluating analytic gradients on quantum hardware. *Physical Review A* **2018**.

 doi:10.20944/preprints202501.2126.v1

15. Harrow, A.W.; Napp, J.C. Low-Depth Gradient Measurements Can Improve Convergence in Variational Hybrid Quantum-Classical Algorithms. *Phys. Rev. Lett.* **2021**, *126*, 140502. https://doi.org/10.1103/PhysRevLett.126.140502.
16. Kübler, J.M.; Arrasmith, A.; Cincio, L.; Coles, P.J. An Adaptive Optimizer for Measurement-Frugal Variational Algorithms. *Quantum* **2020**, *4*, 263. https://doi.org/10.22331/q-2020-05-11-263.
17. Lloyd, S.; Schuld, M.; Ijaz, A.; Izaac, J.; Killoran, N. Quantum embeddings for machine learning, 2020, [arXiv:quant-ph/2001.03622].