SMLab Talk

Data Re-uploading For A Universal Quantum Classifier

Adri ´an P ´erez-Salinas | Alba Cervera-Lierta | Elies Gil-Fuster | Jos ´e I. Latorre

OUTLINES

- Introduction
- Structure of a Single-Qubit Classifier
- Structure of a Multi-Qubit Classifier
- Experiments and Results
- Results
- Conclusion

INTRODUCTION

- **QML and Classification**

  Pattern recognition in ML

  Can Quantum computing improve it?

- **Key Question**

  Do we need multiple qubits for classification?

  Can a single qubit be enough?

- **Main Idea**

  Data Re-Uploading : Encode data multiple times.

  A single qubit with multiple layers can classify patterns.

- **Why important?**

  Saves quantum resources.

  Works on NISQ devices.

Structure of a Single-Qubit Classifier

- Single qubit classifiers face limitations in dimensions and processing power.
- Solution: re-upload same classical data multiple times during computation.
- Each upload rotates the qubit, followed by trainable processing units.
- Similar to neural networks where same data feeds multiple neurons.
- Performance improves with number of re-uploads.
- Enables complex pattern recognition with minimal quantum resources.

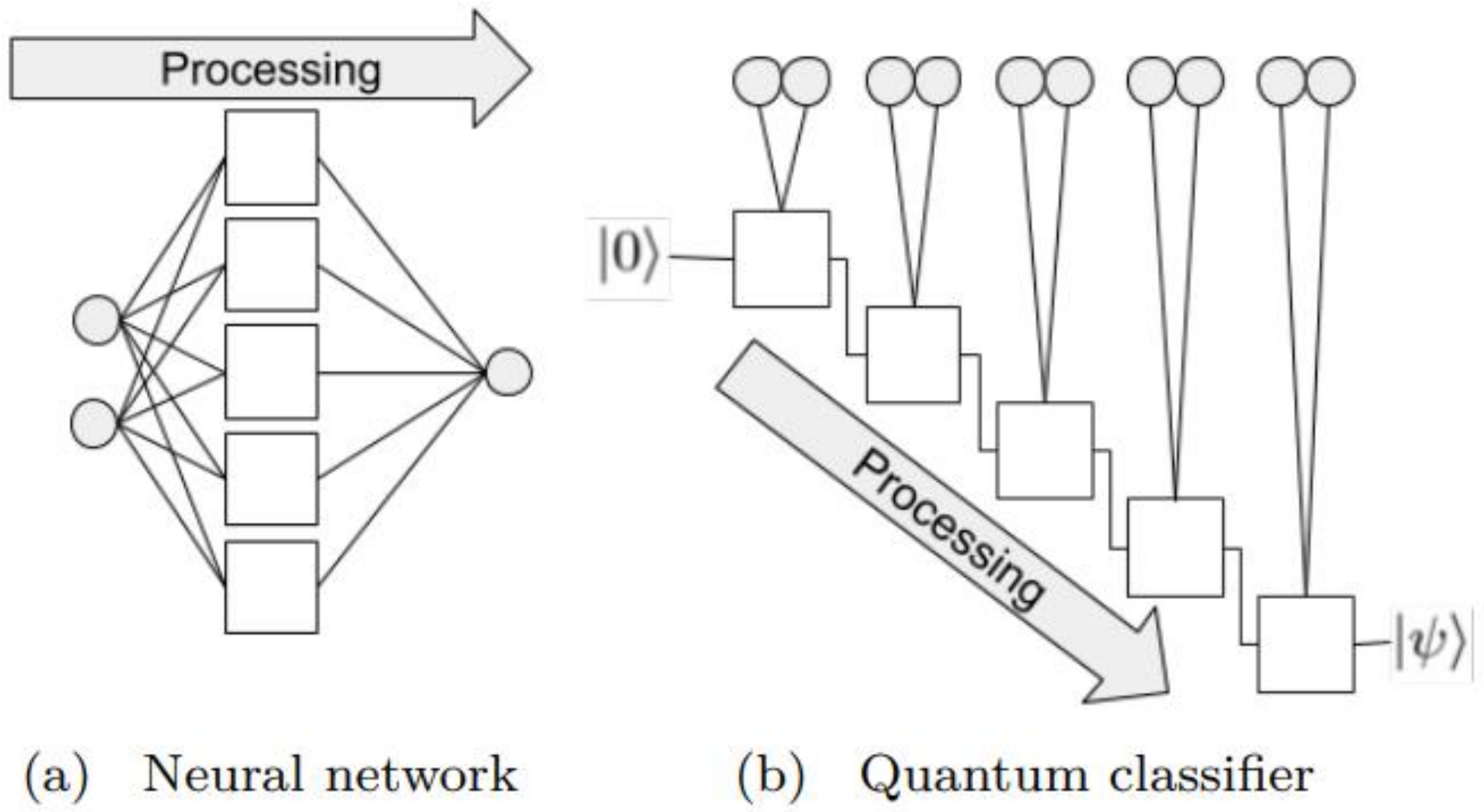


Figure 1: Comparison of a neural network and a single-qubit quantum classifier with data re-uploading. The neural network processes inputs layer by layer, while the quantum classifier reintroduces data at each processing step, encoding multiple repetitions within a quantum state.

Universal Quantum Classifier with a Single Qubit

- General Form of the Classifier:

$$\mathcal{U}(\vec{\phi},\vec{x}) \equiv U(\vec{\phi}_N)U(\vec{x})\cdots U(\vec{\phi}_1)U(\vec{x})$$

- State Preparation:

$$|\psi\rangle = \mathcal{U}(\vec{\phi},\vec{x})|0\rangle$$

Processing Layer Concept

- Layer Representation:

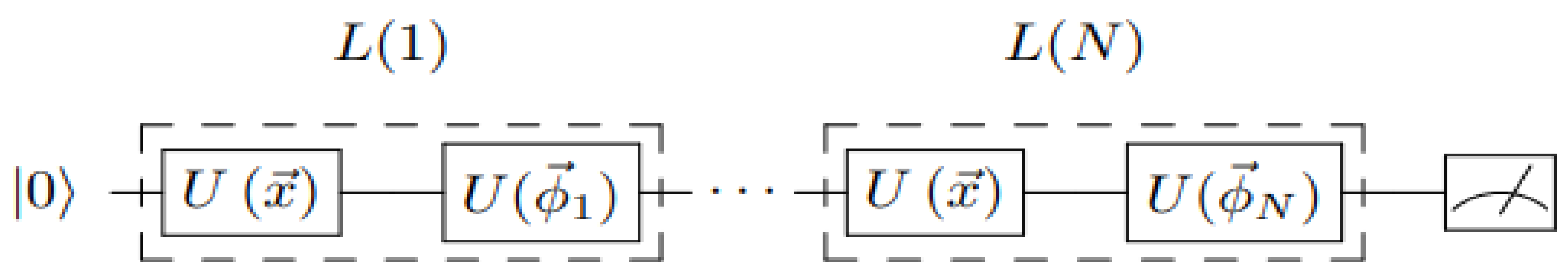
$$L(i) \equiv U(\vec{\phi}_i)U(\vec{x})$$

- Classifier as a Sequence of Layers:

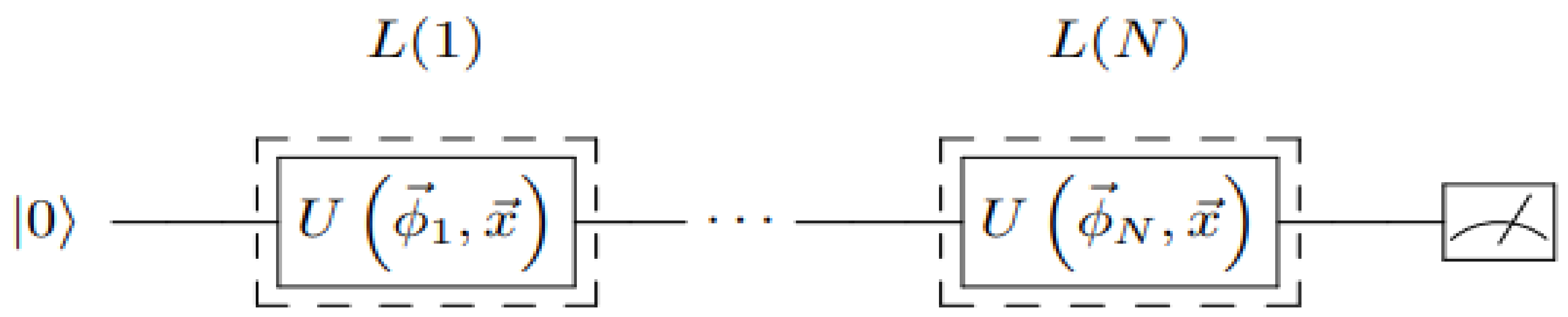
$$\mathcal{U}(\vec{\phi},\vec{x}) = L(N)\cdots L(1)$$

- Circuit Depth:

$$2N$$



(a) Original scheme



(b) Compressed scheme

Figure 2: Single-qubit classifier with data re-uploading. The quantum circuit is divided into layer gates $L(i)$, which constitutes the classifier building blocks. In the upper circuit, each of these layers is composed of a $U(\vec{x})$ gate, which uploads the data, and a parametrized unitary gate $U(\vec{\phi})$. We apply this building block N times and finally compute a cost function that is related to the fidelity of the final state of the circuit with the corresponding target state of its class. This cost function may be minimized by tunning the $\vec{\phi}_i$ parameters. Eventually, data and tunable parameters can be introduced with a single unitary gate, as illustrated in the bottom circuit.

Compact Quantum Circuit

- **Compactified Layer Formulation:**
  Data and parameters are incorporated in a single rotation per layer:

$$L(i) = U\left(\vec{\theta}_i + \vec{w}_i \circ \vec{x}\right)$$

- **Hadamard Product (Element-wise Multiplication):**

$$\vec{w}_i \circ \vec{x} = (w_i^1 x^1, w_i^2 x^2, w_i^3 x^3)$$

- **Key Properties:**
  - Depth of the circuit is **reduced by half**.
  - Each data point can be uploaded with a weight w_i, similar to weights in neural networks.
  - If $\vec{x}$ has fewer than 3 components, remaining values are set to zero.
- **Trade-off:**
  - Further layer reduction can be achieved, but excessive compression reduces non-linearity, degrading performance.

Enlarging Input Space in Quantum Circuits

- **Extended Layer Definition:**
  The input space can be divided into k vectors of dimension 3:

$$L(i) = U\left(\vec{\theta}_i^{(k)} + \vec{w}_i^{(k)} \circ \vec{x}^{(k)}\right) \cdots U\left(\vec{\theta}_i^{(1)} + \vec{w}_i^{(1)} \circ \vec{x}^{(1)}\right)$$

- **Key Points:**
  - Each unitary U handles as many variables as an **SU(2)** unitary allows.
  - Variables are applied sequentially across k iterations.
  - After k iterations, all variables are processed.
- **Complexity:**
  - The circuit complexity grows **linearly** with the input size.

Measurement and Classification

- The classifier produces a final state $|\psi\rangle$ based on processing angles $\vec{\theta}$ and weights $\vec{w}$.
- Measurement probabilities $P(0)$ and $P(1)$ are used for binary classification:

$$\text{Class A if } P(0) > \lambda, \text{ Class B otherwise.}$$

λ is a tunable threshold.

- For multi-class classification (e.g., 4 or 6 classes):
  - Compare probabilities with multiple thresholds $\lambda_1, \lambda_2, \ldots$
  - Alternatively, use state overlaps for robust classification.

Fidelity Cost Function

- Measures how close the output state $|\psi(\vec{\theta}, \vec{w}, \vec{x})\rangle$ is to the target label state $|\tilde{\psi}_s\rangle$.
- The fidelity cost function to minimize is:

$$\chi_f^2(\vec{\theta}, \vec{w}) = \sum_{\mu=1}^{M} \left(1 - \left|\langle\tilde{\psi}_s|\psi(\vec{\theta}, \vec{w}, \vec{x}_\mu)\rangle\right|^2\right)$$

  - M = Total training data
  - $|\tilde{\psi}_s\rangle$ = Correct label state

Weighted Fidelity Cost Function

- The weighted fidelity cost function measures how well the predicted state $|\psi(\vec{\theta}, \vec{w}, \vec{x})\rangle$ matches the target label states $|\psi_c\rangle$ using class-specific weights.
- It is defined as:

$$\chi^2_{wf}(\vec{\alpha}, \vec{\theta}, \vec{w}) = \frac{1}{2} \sum_{\mu=1}^{M} \left(\sum_{c=1}^{C} \left(\alpha_c F_c(\vec{\theta}, \vec{w}, \vec{x}_\mu) - Y_c(\vec{x}_\mu) \right)^2 \right)$$

Where:

- χ^2_{wf} = Weighted fidelity cost function
- M = Number of training points
- C = Number of classes
- α_c = Class weights
- $F_c(\vec{\theta}, \vec{w}, \vec{x}) = \left| \langle \psi_c | \psi(\vec{\theta}, \vec{w}, \vec{x}) \rangle \right|^2$ = Fidelity between predicted and label states
- $Y_c(\vec{x})$ = Target class-wise fidelity

From Single- to Multi-Qubit Quantum Classifier

Concept Overview

- A **single-qubit classifier** has limited capabilities and cannot outperform classical neural networks.
- To achieve **quantum advantage**, multi-qubit classifiers with entanglement are introduced.
- **Entanglement** improves learning capacity, reduces layers, and enhances classification performance.

Measurement Strategies

Two Approaches for Multi-Qubit Classifiers:

1. **State Comparison Strategy**:
    - Compare the final state of the circuit with computational basis states.
    - Similar to the single-qubit approach but computationally expensive with many qubits.
2. **Focused Qubit Strategy**:
    - Measure one qubit and assign a class using thresholds.
    - Allows multi-class classification with reduced measurement complexity.

Cost Function (State Comparison):

$$\chi_f^2(\vec{\theta}, \vec{w}) = \sum_{\mu=1}^{M} \left(1 - \left|\langle \psi_c | \psi(\vec{\theta}, \vec{w}, \vec{x}_\mu)\rangle\right|^2\right)$$

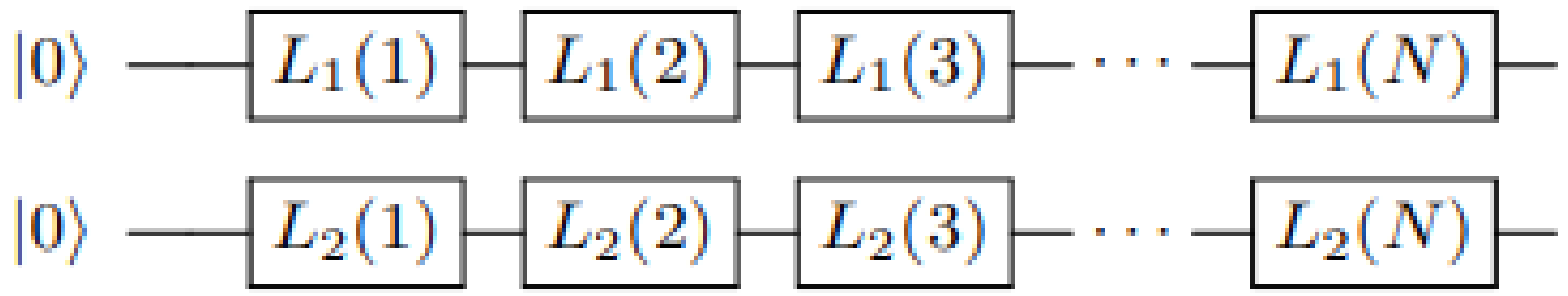
Cost Function (Focused Qubit):

$$\chi_{\mathrm{wf}}^2(\vec{\alpha}, \vec{\theta}, \vec{w}) = \frac{1}{2} \sum_{\mu=1}^{M} \sum_{c=1}^{C} \left[\sum_{q=1}^{Q} \left(\alpha_{c,q} F_{c,q}(\vec{\theta}, \vec{w}, \vec{x}_\mu) - Y_c(\vec{x}_\mu)\right)^2\right]$$

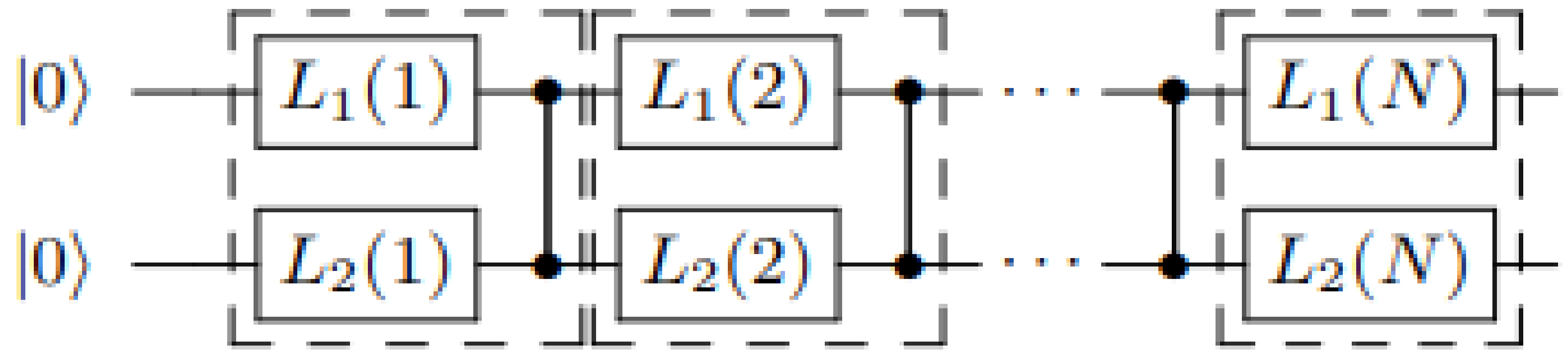
Where:

- $\alpha_{c,q}$ = Class and qubit-specific weight
- $F_{c,q}$ = Fidelity of qubit state with class label
- Y_c = Expected classification result
- M = Number of training points

Structure of a Multi-Qubit Classifier

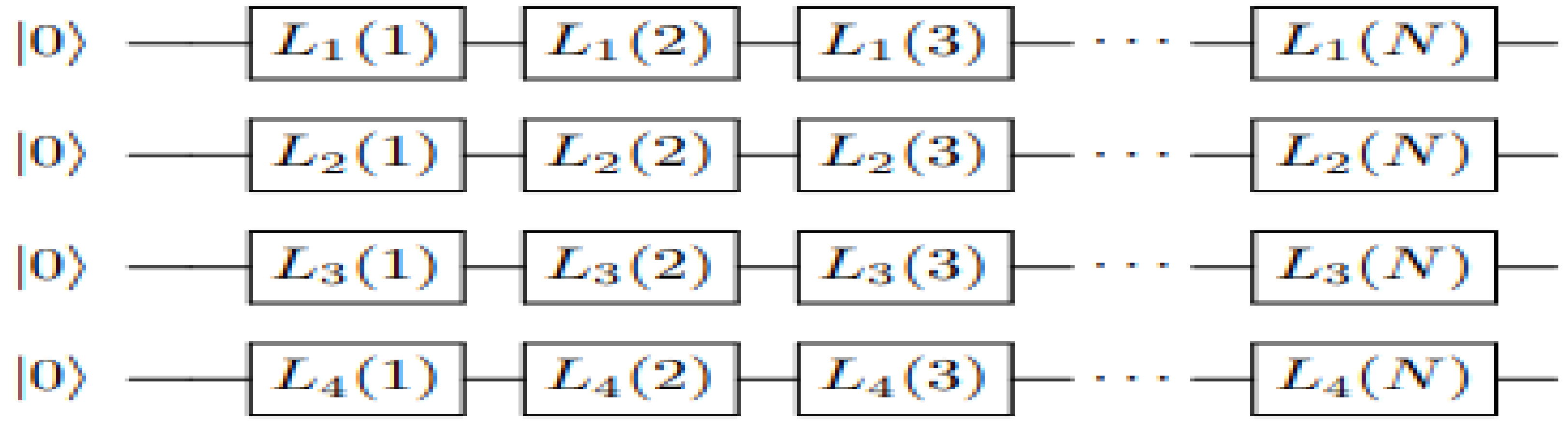


(a) Ansatz with no entanglement

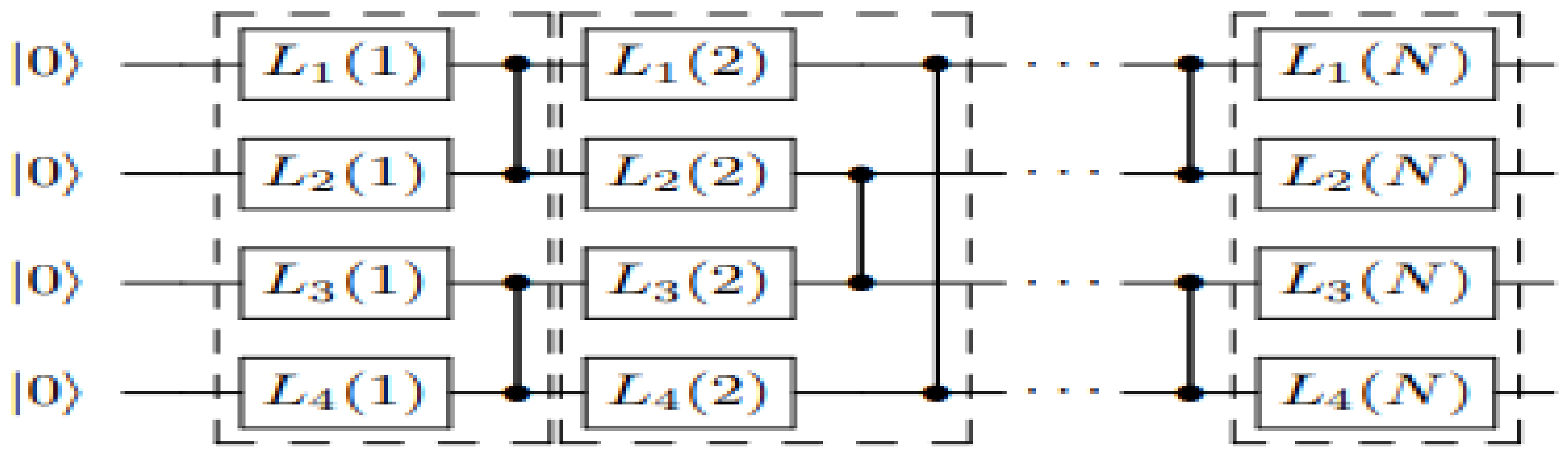


(b) Ansatz with entanglement

Figure 4: Two-qubit quantum classifier circuit without entanglement (top circuit) and with entanglement (bottom circuit). Here, each layer includes a rotation with data re-uploading in both qubits plus a CZ gate if there is entanglement. The exception is the last layer, which does not have any CZ gate associated to it. For a fixed number of layers, the number of parameters to be optimized doubles the one needed for a single-qubit classifier.



(a) Ansatz with no entanglement



(b) Ansatz with entanglement

Figure 5: Four-qubit quantum classifier circuits. Without entanglement (top circuit), each layer is composed by four parallel rotations. With entanglement (bottom circuit) each layer includes a parallel rotation and two parallel CZ gates. The order of CZ gates alternates in each layer between (1)-(2) and (3)-(4) qubits and (2)-(3) and (1)-(4) qubits. The exception is in the last layer, which does not contain any CZ gate. For a fixed number of layers, the number of parameters to be optimized quadruples the ones needed for a single-qubit classifier.

Challenges in Minimization

- **Local Minima:** Due to the trigonometric nature of quantum gates, the cost landscape is highly non-convex.
- **Gradient Trapping:** Classical optimizers can get stuck in poor local minima.
- **Small Training Sets:** Limited data makes the problem even harder.

Minimization Techniques

1. Stochastic Gradient Descent (SGD)

- Commonly used in machine learning.
- Efficient for large datasets.
- May struggle with local minima in small data scenarios.

2. L-BFGS-B (Limited-memory Broyden-Fletcher-Goldfarb-Shanno with Boundaries)

- Suitable for small datasets.
- Less sensitive to local minima.
- Faster convergence compared to SGD.
- Used in this study due to computational limitations.

Experiments and Results

	χ_f^2			χ_{wf}^2				
Qubits	1	2		1	2		4	
Layers		No Ent.	Ent.		No Ent.	Ent.	No Ent.	Ent.
1	0.50	0.75	–	0.50	0.76	–	0.76	–
2	0.85	0.80	0.73	0.94	0.96	0.96	0.96	0.96
3	0.85	0.81	0.93	0.94	0.97	0.95	0.97	0.96
4	0.90	0.87	0.87	0.94	0.97	0.96	0.97	0.96
5	0.89	0.90	0.93	0.96	0.96	0.96	0.96	0.96
6	0.92	0.92	0.90	0.95	0.96	0.96	0.96	0.96
8	0.93	0.93	0.96	0.97	0.95	0.97	0.95	0.96
10	0.95	0.94	0.96	0.96	0.96	0.96	0.96	0.97

Table 1: Results of the single- and multi-qubit classifiers with data re-uploading for the circle problem. Numbers indicate the success rate, i.e. number of data points classified correctly over total number of points. Words "Ent." and "No Ent." refer to considering entanglement between qubits or not, respectively. We have used the L-BFGS-B minimization method with the weighted fidelity and fidelity cost functions. For this problem, both cost functions lead to high success rates. The multi-qubit classifier increases this success rate but the introduction of entanglement does not affect it significantly.

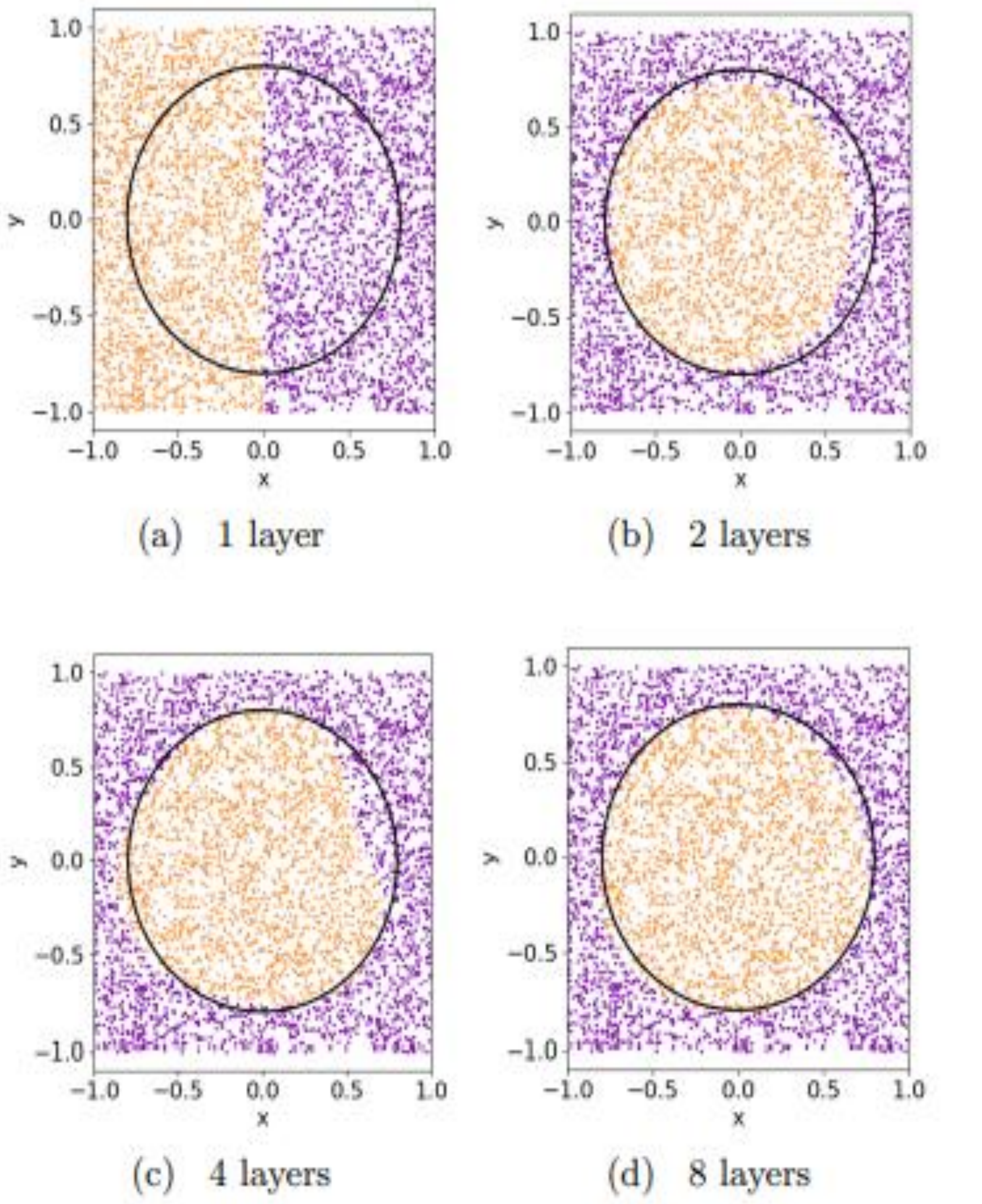


Figure 6: Results of the circle classification obtained with a single-qubit classifier with different number of layers using the L-BFGS-B minimizer and the weighted fidelity cost function. With one layer, the best that the classifier can do is to divide the plane in half. With two layers, it catches the circular shape which is readjusted as we consider more layers.

Qubits	χ^2_f 1	χ^2_f 2		χ^2_{wf} 1	χ^2_{wf} 2		χ^2_{wf} 4	
Layers		No Ent.	Ent.		No Ent.	Ent.	No Ent.	Ent.
1	0.73	0.56	–	0.75	0.81	–	0.88	–
2	0.79	0.77	0.78	0.76	0.90	0.83	0.90	0.89
3	0.79	0.76	0.75	0.78	0.88	0.89	0.90	0.89
4	0.84	0.80	0.80	0.86	0.84	0.91	0.90	0.90
5	0.87	0.84	0.81	0.88	0.87	0.89	0.88	0.92
6	0.90	0.88	0.86	0.85	0.88	0.89	0.89	0.90
8	0.89	0.85	0.89	0.89	0.91	0.90	0.88	0.91
10	0.91	0.86	0.90	0.92	0.90	0.91	0.87	0.91

Table 2: Results of the single- and multi-qubit classifiers with data re-uploading for the 3-circles problem. Numbers indicate the success rate, i.e. number of data points classified correctly over total number of points. Words "Ent." and "No Ent." refer to considering entanglement between qubits or not, respectively. We have used the L-BFGS-B minimization method with the weighted fidelity and fidelity cost functions. Weighted fidelity cost function presents better results than the fidelity cost function. The multi-qubit classifier reaches 0.90 success rate with a lower number of layers than the single-qubit classifier. The introduction of entanglement slightly increases the success rate respect the non-entangled circuit.

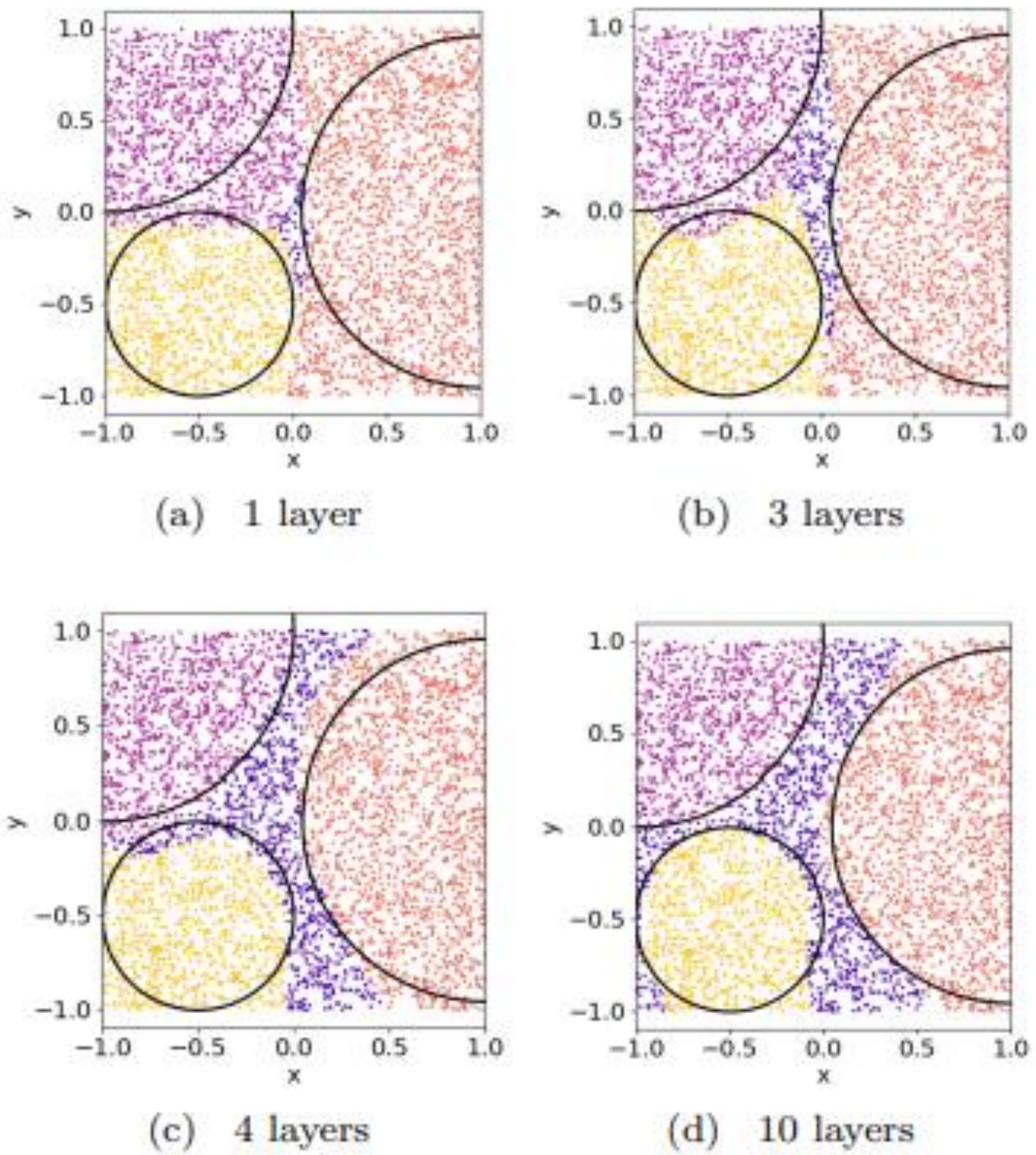


Figure 7: Results for the 3-circles problem using a single-qubit classifier trained with the L-BFGS-B minimizer and the weighted fidelity cost function. With one layer, the classifier intuits the four regions although the central one is difficult to tackle. With more layers, this region is clearer for the classifier and it tries to adjust the circular regions.

Qubits	1 (χ_f^2)	2 (χ_f^2)		1 (χ_{wf}^2)	2 (χ_{wf}^2)		4 (χ_{wf}^2)	
Layers		No Ent.	Ent.		No Ent.	Ent.	No Ent.	Ent.
1	0.87	0.87	–	0.87	0.87	–	0.90	–
2	0.87	0.87	0.87	0.87	0.92	0.91	0.90	0.98
3	0.87	0.87	0.87	0.89	0.89	0.97	–	–
4	0.89	0.87	0.87	0.90	0.93	0.97	–	–
5	0.89	0.87	0.87	0.90	0.93	0.98	–	–
6	0.90	0.87	0.87	0.95	0.93	0.97	–	–
8	0.91	0.87	0.87	0.97	0.94	0.97	–	–
10	0.90	0.87	0.87	0.96	0.96	0.97	–	–

Table 3: Results of the single- and multi-qubit classifiers with data re-uploading for the four-dimensional hypersphere problem. Numbers indicate the success rate, i.e. the number of data points classified correctly over the total number of points. Words "Ent." and "No Ent." refer to considering entanglement between qubits or not, respectively. We have used the L-BFGS-B minimization method with the weighted fidelity and fidelity cost functions. The fidelity cost function gets stuck in some local minima for the multi-qubit classifiers. The results obtained with the weighted fidelity cost function are much better, reaching the 0.98 with only two layers for the four-qubit classifier. Here, the introduction of entanglement improves significantly the performance of the multi-qubit classifier.

Qubits	1 (χ_f^2)	2 (χ_f^2)		1 (χ_{wf}^2)	2 (χ_{wf}^2)		4 (χ_{wf}^2)	
Layers		No Ent.	Ent.		No Ent.	Ent.	No Ent.	Ent.
1	0.34	0.51	–	0.43	0.77	–	0.81	–
2	0.57	0.63	0.59	0.76	0.79	0.82	0.87	0.96
3	0.80	0.68	0.65	0.68	0.94	0.95	0.92	0.94
4	0.84	0.78	0.89	0.79	0.93	0.96	0.93	0.96
5	0.92	0.86	0.82	0.88	0.96	0.96	0.96	0.95
6	0.93	0.91	0.93	0.91	0.93	0.96	0.97	0.96
8	0.90	0.89	0.90	0.92	0.94	0.95	0.95	0.94
10	0.90	0.91	0.92	0.93	0.95	0.96	0.95	0.95

Table 4: Results of the single- and multi-qubit classifiers with data re-uploading for the three-class annulus problem. Numbers indicate the success rate, i.e. the number of data points classified correctly over the total number of points. Words "Ent." and "No Ent." refer to considering entanglement between qubits or not, respectively. We have used the L-BFGS-B minimization method with the weighted fidelity and fidelity cost functions. The weighted fidelity cost function presents better success rates than the fidelity cost function. The multi-qubit classifiers improve the results obtained with the single-qubit classifier but the using of entanglement does not introduce significant changes.

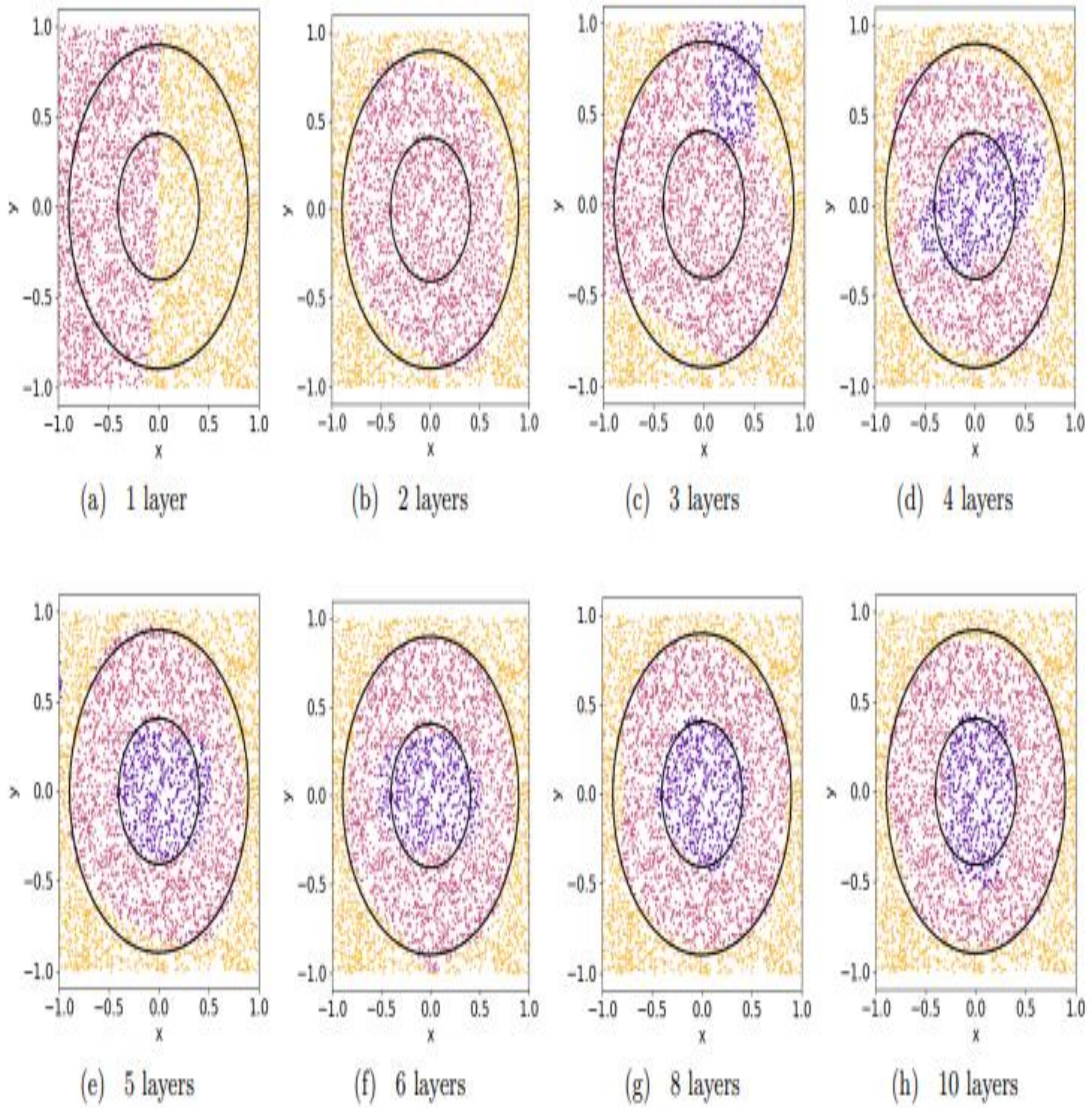


Figure 8: Results obtained with the single-qubit classifier for the annulus problem, using the weighted fidelity cost function during the training. The better results are obtained with a 10 layers classifier (93% of success rate). As we consider more qubits and entanglement, we can increase the success rate up to 96%, as shows Table 4.

RESULTS

Problem	Classical classifiers NN	SVC	Quantum classifier χ_f^2	χ_{wf}^2
Circle	0.96	0.97	0.96	0.97
3 circles	0.88	0.66	0.91	0.91
Hypersphere	0.98	0.95	0.91	0.98
Annulus	0.96	0.77	0.93	0.97
Non-Convex	0.99	0.77	0.96	0.98
Binary annulus	0.94	0.79	0.95	0.97
Sphere	0.97	0.95	0.93	0.96
Squares	0.98	0.96	0.99	0.95
Wavy Lines	0.95	0.82	0.93	0.94

Table 5: Comparison between single-qubit quantum classifier and two well-known classical classification techniques: a neural network (NN) with a single hidden layer composed of 100 neurons and a support vector classifier (SVC), both with the default parameters as defined in `scikit-learn` python package. We analyze nine problems: the first four are presented in Section 6 and the remaining five in Appendix B. Results of the single-qubit quantum classifier are obtained with the fidelity and weighted fidelity cost functions, χ_f^2 and χ_{wf}^2 defined in Eq. (7) and Eq. (9) respectively. This table shows the best success rate, being 1 the perfect classification, obtained after running ten times the NN and SVC algorithms and the best results obtained with single-qubit classifiers up to 10 layers.

Conclusion

- Single-qubit classifiers achieve effective classification using data re-uploading.
- Universal Approximation Theorem ensures accurate function approximation.
- Multi-qubit classifiers with entanglement enhance performance.
- Weighted fidelity cost function supports multi-class classification.
- L-BFGS-B optimization efficiently minimizes quantum cost functions.

References

https://quantum-journal.org/papers/q-2020-02-06-226/pdf/

THANK YOU

A. Preliminaries of Quantum Circuit

In quantum computing, information is often carried by qubits over Hilbert space. A pure quantum state consists of one or more qubits and is usually represented by Dirac's notation, which denotes a unit vector $\mathbf{v}$ as a ket $|v\rangle$ and its conjugate transpose $\mathbf{v}^\dagger$ as a bra $\langle v|$. The inner product between $|v\rangle$ and $|u\rangle$ is denoted as $\langle u|v\rangle$, and the outer product is $|u\rangle\langle v|$. The evolution of a quantum state $|v\rangle$ is accomplished by sequentially applying quantum gates on it, i.e. $|v'\rangle = U_K \ldots U_2 U_1 |v\rangle$, where U_k is the unitary matrix representing the quantum gate and $|v'\rangle$ is the quantum state after evolution. Common single-qubit gates are as follows:

$$
\begin{aligned}
H &:= \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, I := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\
X &:= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, R_X(\theta) := \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, \\
Y &:= \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, R_Y(\theta) := \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, \\
Z &:= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, R_Z(\theta) := \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix},
\end{aligned} \tag{A1}
$$

where H denotes the Hadamard gate, X, Y, Z denote the Pauli gates, $R_X(\theta), R_Y(\theta), R_Z(\theta)$ denote the rotation gates. A common multi-qubit gate is $CNOT$ gate:

$$
CNOT := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{A2}
$$

In a quantum circuit, the initial quantum state is generally $|0\rangle^{\otimes N}$, and after applying a sequence of quantum gates, the measurement will be used to convert quantum information into classical information. For instance, we can design quantum measurements to obtain the expectation $\langle v|O|v\rangle$ of the quantum state $|v\rangle$ about an observable O.