Data re-uploading for a Universal quantum classifier
Outlook

- From classical to quantum NN
- Single-qubit classifier
- Universality
- Multi-qubit classifier
- Benchmarks
- Conclusions and remarks
From classical to quantum NN


The minimal QNN

What is the most simple (but universal) NN?
Single hidden layer NN

What is the most simple (but universal) QNN?
Single-qubit QNN

---

What are the power and minimal needs of a quantum circuit to carry out a general supervised classification task?

- Qubits
- Operations
- Parameters

1 qubit is enough if data is re-uploaded along the circuit and if assisted with a classical optimization subroutine.
Encoding the data

A product of free single-qubit unitaries can be written with another single-qubit unitary

\[ U(\phi_1) \ldots U(\phi_N) \equiv U(\phi) \]

If we add some fixed parameter dependency (the data), the operation becomes flexible and data-dependent.

Data re-uploading

\[ U(\phi, \vec{x}) \equiv U(\phi_N)U(\vec{x}) \ldots U(\phi_1)U(\vec{x}) \]

The paths depend on the data \( x \).
“data re-uploading” in other works

The idea of encoding data in more than one circuit operation is not new. However, the motivation and methodology varies from one proposal to another.

Circuit-centric classification:
construct a classically hard feature map

Kernel methods:
Construct the Kernel to measure it.

Data re-uploading layers

The total unitary is divided into layers. Each layer encodes the data.

\[ U(\tilde{\phi}, \vec{x}) = L(N) \ldots L(1) \]

\[ L(i) \equiv U(\tilde{\phi}_i)U(\vec{x}) \]

\[ L(i) = U(\tilde{\theta}_i + \tilde{w}_i \circ \vec{x}) \]

(a) Original scheme

(b) Compressed scheme

Why this particular encoding?

Target states

Convenient: choose the most orthogonal states to define each target state.

Single-qubit  Divide the Bloch sphere into $N_{\text{class}}$ sections


Extension for multi-qubits:

Measurement and cost function

Target state: one for each label/class.

Compute the fidelity (overlap) between the quantum circuit state and the target state:

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

Weighted fidelity (for multiclassification): compute the overlap w.r.t. target state – distance w.r.t. other class target state:

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

\[ L(i) = U \left( \bar{\theta}_i + \bar{w}_i \circ \bar{x} \right) \]

Why this particular encoding?

The choose of this encoding allows us to connect the classifier with the Universality proof.
Universal Approximation Theorem

Any continuous function $f(x)$ can be approximated with $\epsilon$ accuracy by the function

$$h(x) = \sum_{i=1}^{N} \alpha_i \varphi(w_i \cdot x + b_i)$$

where $\varphi$ is a nonconstant, bounded and continuous function.

A single-layer neural network can approximate any continuous function (providing enough neurons in the hidden layer)
Universal Quantum Circuit approximation

Single-qubit quantum gate = SU(2) operator:

\[ U(\phi) = e^{i\phi(\sigma_x)} \]

Linear encoding

\[ \phi(x) = (\phi_1(x), \phi_2(x), \phi_3(x)) = \theta + \bar{w} \circ x. \]

Multiple products of SU(2) operators are also a SU(2) operator

\[ U(x) = U_N(x) U_{N-1}(x) \cdots U_1(x) = \prod_{i=1}^{N} e^{i\phi_i(x)} \]

Circuit layers

Continous, bounded, nonconstant

\[ \omega_1(\phi) = d \mathcal{N} \sin ((\phi_2 - \phi_3)/2) \sin (\phi_1/2) \]

\[ \cos d = \cos ((\phi_2 + \phi_3)/2) \cos (\phi_1/2) \]

Applying the BCH formula:

\[ U(x) = \exp \left[ i \sum_{i=1}^{N} \phi_i(x) \cdot \sigma + O_{\text{corr}} \right] = e^{i\tilde{f}(x) \cdot \sigma + i\tilde{g}(x) \cdot \sigma} \]

Continous functions

\[ \left( \omega_1(\tilde{\theta}_i + \tilde{\omega}_i \circ \tilde{x}), \omega_2(\tilde{\theta}_i + \tilde{\omega}_i \circ \tilde{x}), \omega_3(\tilde{\theta}_i + \tilde{\omega}_i \circ \tilde{x}) \right) = (f_1(x), f_2(x), f_3(x)) \]

A single-qubit quantum classifier can be simulated classically.

We need to introduce entanglement (therefore, more qubits) to eventually prove any quantum advantage.
Multi-qubit quantum classifier

No mathematical proofs: heuristic experiment.

Is entanglement playing any role or just the fact that we are considering more qubits?

Which entanglement ansatz should we use? We tried alternating entanglement ansatz.

(a) Ansatz with no entanglement

(b) Ansatz with entanglement
## Benchmarks

<table>
<thead>
<tr>
<th>Problem</th>
<th># Classes</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circle</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3 circles</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Hypersphere</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Annulus</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Non-convex</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Binary annulus</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Sphere</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Squares</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Wavy lines</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>
2D circle

Training/test points = 200/4000
Random accuracy = 50%

<table>
<thead>
<tr>
<th>Qubits</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layers</td>
<td></td>
<td>$\chi_f^2$</td>
<td>$\chi_{\text{Ent}}^2$</td>
<td>$\chi_{\text{Ent}}^2$</td>
<td>$\chi_{\text{Ent}}^2$</td>
</tr>
<tr>
<td>1</td>
<td>0.50</td>
<td>0.75</td>
<td>–</td>
<td>0.50</td>
<td>–</td>
</tr>
<tr>
<td>2</td>
<td>0.85</td>
<td>0.80</td>
<td>0.73</td>
<td>0.94</td>
<td>0.96</td>
</tr>
<tr>
<td>3</td>
<td>0.85</td>
<td>0.81</td>
<td>0.93</td>
<td>0.94</td>
<td>0.97</td>
</tr>
<tr>
<td>4</td>
<td>0.90</td>
<td>0.87</td>
<td>0.87</td>
<td>0.94</td>
<td>0.97</td>
</tr>
<tr>
<td>5</td>
<td>0.89</td>
<td>0.90</td>
<td>0.93</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>6</td>
<td>0.92</td>
<td>0.92</td>
<td>0.90</td>
<td>0.95</td>
<td>0.96</td>
</tr>
<tr>
<td>8</td>
<td>0.93</td>
<td>0.93</td>
<td>0.96</td>
<td>0.97</td>
<td>0.95</td>
</tr>
<tr>
<td>10</td>
<td>0.95</td>
<td>0.94</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
</tr>
</tbody>
</table>

2D: 3 circles

Training/test points = 200/4000
Random accuracy = 25%
2D: Annulus

Training/test points = 200/4000
Random accuracy = 33%
Other 2D problems

(a) $\chi_{w,f}^2$, 1 qubit, 6 layers

(b) $\chi_{w,f}^2$, 2 qubits without entanglement, 4 layers

(c) $\chi_{f}^2$, 2 qubits without entanglement, 6 layers

(d) $\chi_{w,f}^2$, 2 qubits with entanglement, 6 layers

The aim of this classical benchmarking is not to make an extended review of what classical machine learning is capable to perform. The aim is to compare our simple quantum classifier to simple models such as shallow neural networks and simple support vector machines.

The result of the single-qubit classifier is comparable with classical models.
Conclusions and remarks

- A single-qubit is capable of performing a multiclassification task when:
  1. Assisted with a classical optimization subroutine (VQA).
  2. Data is re-uploaded along the circuit.

- Its performance is comparable with other classical methods such as NN and SVM.

- Its extensión to multiple qubits and the entanglement role should be studied in more detail.

- Is it affected by the barren plateau problem?
  There exist a correlation between the different layers: the data points encoded.

- Are other encoding strategies better than the linear encoding?

- Use this model beyond classification: meta-VQE algorithm uses data re-uploading strategy.

Aknowledgements

Adrián Pérez-Salinas
Elies Gil-Fuster
José Ignacio Latorre