Quantum machine learning with small-scale devices

Maria Schuld, Mark Fingerhuth, F. Petruccione

Quantum Techniques in Machine Learning
Verona, 14 September 2017

Implementing a distance-based classifier with a quantum interference circuit

Maria Schuld, Mark Fingerhuth, F. Petruccione

www.proteinqure.com

Quantum Techniques in Machine Learning
Verona, 14 September 2017

Classical machine learning

pattern recognition

supervised learning

unsupervised learning

reinforcement learning

data clustering

intelligent agents/games

### Overview

<table>
<thead>
<tr>
<th>Machine Learning Method</th>
<th>Quantum Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-nearest neighbour</td>
<td>Efficient calculation of classical distances on a quantum computer</td>
</tr>
<tr>
<td>Support Vector Machines</td>
<td>First explorations of quantum models</td>
</tr>
<tr>
<td>k-means clustering</td>
<td>Reformulation in the language of open quantum systems</td>
</tr>
<tr>
<td>Neural Networks</td>
<td></td>
</tr>
<tr>
<td>Decision Trees</td>
<td></td>
</tr>
<tr>
<td>Bayesian Theory</td>
<td></td>
</tr>
<tr>
<td>Hidden Markov Models</td>
<td></td>
</tr>
</tbody>
</table>

Combining QC and ML

Aimeur, Brassard, Gambs (2006)
Data encoding and read out

ML algorithm

Dataset $\mathcal{D}$, new instance $\tilde{x}$

Prediction algorithm

Prediction $\tilde{y}$

QML algorithm

Dataset $\mathcal{D}$, new instance $\tilde{x}$

Encoding

Quantum prediction algorithm

Read out

Prediction $\tilde{y}$

Quantum system

state preparation

unitary evolution

measurement

Questions

Can a quantum computer enhance methods known from machine learning?

What classifier can be realized by a (minimal) quantum circuit?
A simple quantum circuit can be used as a (simple) model of a classifier.

Implementing a distance-based classifier with a quantum interference circuit

- Theory
Supervised binary pattern classification

Given a training data set

\[ D = \{(x^1, y^1), \ldots, (x^M, y^M)\} \]

Inputs: \( x^m \in \mathbb{R}^N \) \hspace{1cm} \text{Target labels} \hspace{1cm} y^m \in \{-1, 1\}

\( m = 1, \ldots, M \)

find the label \( \tilde{y} \in \{-1, 1\} \) that corresponds to the new input \( \tilde{x} \in \mathbb{R}^N \)
Outlook

Classifier implemented by the QML circuit

\[ \tilde{y} = \text{sgn} \left( \sum_{m=1}^{M} y^m \left[ 1 - \frac{1}{4M} |\tilde{x} - x^m|^2 \right] \right) \]

\[ \kappa(x, x') = 1 - \frac{1}{4M} |x - x'|^2 \]

(distance measure)

kernelised binary classifier

“distance-based classifier”
Amplitude encoding

Idea: Encode input features into the amplitudes of a quantum system and manipulate through quantum gates

Given: \( \mathbf{x} \in \mathbb{R}^N \), where \( N = 2^n \), with \( \mathbf{x}^T \mathbf{x} = 1 \)

\[ \mathbf{x} = (x_1, \ldots, x_N)^T \quad \Rightarrow \quad |\psi_{\mathbf{x}}\rangle = \sum_{i=0}^{N-1} x_i |i\rangle \]

index register
State preparation

Training data set

\[ \mathcal{D} = \{(x^1, y^1), \ldots, (x^M, y^M)\} \]

Inputs: \( x^m \in \mathbb{R}^N \) \quad \text{and} \quad \text{Targets: } y^m \in \{-1, 1\}

State preparation

\[ |\mathcal{D}\rangle = \frac{1}{\sqrt{2MC}} \sum_{m=1}^{M} |m\rangle \left( |0\rangle |\psi_{\tilde{x}}\rangle + |1\rangle |\psi_{x^m}\rangle \right) |y^m\rangle \]

Index register

(new input training state)

|0\rangle, if \( y^m = -1 \)

|1\rangle, if \( y^m = 1 \)

ancilla

amplitude vector contains training inputs and M copies of new input
Classification (1)

Step A \[ |D\rangle \rightarrow \frac{1}{2\sqrt{M}} \sum_{m=1}^{M} |m\rangle \left( |0\rangle \psi_{\tilde{x}+x^m} + |1\rangle \psi_{\tilde{x}-x^m} \right) |y^m\rangle \]

where \[ \psi_{\tilde{x} \pm x^m} = \psi_{\tilde{x}} \pm \psi_{x^m} \]

interference of copies of new input and training sets
Step B  Conditional measurement (selects ancilla in $|0\rangle$) 

Post-selection succeeds with probability

$$p_{\text{acc}} = \frac{1}{4M} \sum_{m} |\tilde{x} + x^m|^2$$

usually around 0.5
Step C  After the successful conditional measurement

\[
\frac{1}{2\sqrt{M_{\text{acc}}}} \sum_{m=1}^{M} \sum_{i=1}^{N} |m\rangle \left( \tilde{x}_i + x_i^m \right) |i\rangle |y^m\rangle
\]

Probability of measuring class qubit \( |y^m\rangle \) in 0

\[
p(\tilde{y} = 0) = \frac{1}{4M_{\text{acc}}} \sum_{m|y^m=0} |\tilde{x} + x^m|^2 = 1 - \frac{1}{4M_{\text{acc}}} \sum_{m} |\tilde{x} - x^m|^2
\]

(normalisation)
Classification (4)

Class with higher probability implements classifier:

\[
\tilde{y} = \text{sgn}\left( \sum_{m=1}^{M} y^m \left[ 1 - \frac{1}{4M} |\tilde{x} - x^m|^2 \right] \right)
\]

\[
\kappa(x, x') = 1 - \frac{1}{4M} |x - x'|^2
\]

(distance measure)

kernelised binary classifier

Number of measurements needed to estimate \( p(\tilde{y} = 0) \) to error \( \epsilon \) with a reasonably high confidence grows with \( \mathcal{O}(\epsilon^{-2}) \)
Implementing a distance-based classifier with a quantum interference circuit

- Experiment on IBMQE
IBM Quantum Experience

- 5 superconducting quits
- 80 gates

http://www.research.ibm.com/quantum
Iris flower data set

R. A. Fisher (1936)

Iris setosa  Iris versicolor  Iris virginica

(Sepal length & width, petal length & width)

Data pre-processing

First 2 classes of Iris dataset

**Step 1**
- zero mean
- unit variance

**Step 2**
- normalise feature vector
Consider pre-processed training dataset:

\[ D_1 = \{ (x^0, y^0), (x^1, y^1) \} \]

with training vectors

\[ x^0 = (0, 1), \quad y^0 = -1 \quad \text{(Iris sample 33)} \]

\[ x^1 = (0.789, 0.615), \quad y^1 = 1 \quad \text{(Iris sample 85)} \]

Classification of 2 new input vectors (class -1):

\[ \tilde{x}' = (-0.549, 0.836) \quad \text{(Iris sample 28)} \]

\[ \tilde{x}'' = (0.053, 0.999) \quad \text{(Iris sample 36)} \]
Experimental implementation

Input vectors:  \[ |\psi_\mathbf{x}'\rangle = -0.549 |0\rangle + 0.836 |1\rangle, \]
\[ |\psi_\mathbf{x}''\rangle = 0.053 |0\rangle + 0.999 |1\rangle, \]

Training vectors:  \[ |\psi_\mathbf{x}^0\rangle = |1\rangle, \]
\[ |\psi_\mathbf{x}^1\rangle = 0.789 |0\rangle + 0.615 |1\rangle. \]
Quantum circuit

Training vectors: $|\psi_{x_0}\rangle = |1\rangle,$
$|\psi_{x_1}\rangle = 0.789 |0\rangle + 0.615 |1\rangle.$

Input vector: $|\psi_z\rangle = -0.549 |0\rangle + 0.836 |1\rangle$
The Iris dataset was randomly divided into a training and test set (ratio 80:20). Table II shows the error, or deviation from the theoretical predictions, while the experimental results show yields simulation results that closely resemble the theoretical predictions.

In order to analyse the performance of the classifier yielding classes 2 and 3 only yield a 93% average success rate. The variance of the error is very low in all cases, and the proportion of misclassified test instances to all test instances is 0.5 due to the standardisation of the data. The results show that running the quantum classifier circuit on around 0.913* 0.547* 0.453* acceptance probability of selecting the correct branch is very high.

Implementing a polynomial feature map [1] requires to prepare copies of the quantum states which represent the training vectors and the ancilla is measured with the training vectors and the ancilla is measured with the excited state of the ancilla and the index qubit (step C) followed by a measurement of the class qubit (due to prior swapping now at the position of the index qubit) when the ancilla was again leads to perfect classification.

The rapid decoherence of the class qubit is especially troublesome since it makes classification of instances, for 1000 repetitions of the random division. The Iris dataset was randomly divided into a training set (marked with asterisks). The variance of the error is very low in all cases, and the proportion of misclassified test instances to all test instances is 0.5 due to the standardisation of the data. The results show that running the quantum classifier circuit on around 0.913* 0.547* 0.453* acceptance probability of selecting the correct branch is very high.
Quantum circuit

Step B: Input vector $\tilde{x}'$ entangled with ground state of ancilla
Quantum circuit

**Step C:** Training vector $x^0$ entangled with excited state of ancilla and ground state of index qubit
Quantum circuit

The quantum circuit is designed to perform a distance-based classifier using two training vectors. The steps involved are:

- **Initial state preparation**:
  - Apply Hadamard gate to the ancilla state $|a_0\rangle = |0\rangle$.
  - Apply Hadamard gate to the index state $|m_0\rangle = |0\rangle$.
  - Apply Hadamard gate to the training vector state $|i_0\rangle = |0\rangle$.
  - Apply Hadamard gate to the class state $|y_0\rangle = |0\rangle$.

- **Training vector $\tilde{x}$**: Apply rotation operator $R_y(\theta)$ to $\tilde{x}$, with $\theta = 4.304$.

- **Entangling training vector**: Apply a controlled-$X$ gate with $y$ and $i$ as controls.

- **Classification**: Apply Hadamard gate to the ancilla and index qubits, followed by a measurement of the class qubit.

**Step D**: Training vector $\tilde{x}$ entangled with excited state of ancilla and index qubit.

The table below shows the classification results for the two-dimensional input vectors.

<table>
<thead>
<tr>
<th>Input $x$</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x^1$</td>
<td>0.911</td>
<td>0.731</td>
<td>0.494</td>
</tr>
<tr>
<td>$x^2$</td>
<td>0.729</td>
<td>0.629</td>
<td>0.371</td>
</tr>
<tr>
<td>$x^3$</td>
<td>0.913</td>
<td>0.547</td>
<td>0.453</td>
</tr>
</tbody>
</table>

The table reflects the classification accuracy for the Iris flower dataset.
Quantum circuit

\[ |a_0\rangle = |0\rangle \]
\[ |m_0\rangle = |0\rangle \]
\[ |i_0\rangle = |0\rangle \]
\[ |y_0\rangle = |0\rangle \]

Step E: Data and class qubits are swapped and class qubit flipped conditioned on index qubit being \(|1\rangle\)
Quantum circuit

Step F: H interferes copies of $\tilde{x}'$ with training vectors and ancilla is measured followed by measurement of class qubit when the ancilla is found to be in $|0\rangle$
That was the theory, in practice...

\[ |a_0\rangle = |0\rangle \quad \begin{array}{c}
H \\
\end{array} \]
\[ |m_0\rangle = |0\rangle \quad \begin{array}{c}
H \\
\end{array} \]
\[ |i_0\rangle = |0\rangle \quad \begin{array}{c}
\oplus \; R_y(-2.152) \; \oplus \; R_y(2.152) \\
\end{array} \]
\[ |c_0\rangle = |0\rangle \]

\[ |a_1\rangle \]
\[ |m_1\rangle \quad \begin{array}{c}
H \\
\end{array} \]
\[ |i_1\rangle \quad \begin{array}{c}
\oplus \; R_y(-0.331) \; \oplus \; R_y(-0.331) \; \oplus \; R_y(0.331) \\
\end{array} \]
\[ |c_1\rangle \]

with

\[ \begin{array}{c}
H \\
\end{array} = \begin{array}{c}
H \\
\end{array} \]
\[ \begin{array}{c}
H \\
\end{array} = \begin{array}{c}
H \\
\end{array} \]
Results

| Input vector | $p_{acc}$ | $p(|c\rangle = |0\rangle)$ | $p(|c\rangle = |1\rangle)$ | Exp | Sim | Theo |
|--------------|-----------|---------------------------|---------------------------|-----|-----|------|
| $\tilde{x}'$ | 0.455     | 0.516                     | 0.484                     |     |     |      |
| $\tilde{x}'$ | 0.731$^\triangleright$ | 0.629$^\triangleright$  | 0.371$^\triangleright$    |     | sim |     |
|              | 0.729$^*$ | 0.629$^*$                 | 0.371$^*$                 |     |     | theo |
| $\tilde{x}''$  | 0.494     | 0.589                     | 0.411                     |     |     |      |
| $\tilde{x}''$  | 0.911$^\triangleright$ | 0.548$^\triangleright$  | 0.452$^\triangleright$    |     | sim |     |
|              | 0.913$^*$ | 0.547$^*$                 | 0.453$^*$                 |     |     | theo |

Correctly classified

short coherence times of ancilla and class qubits
Thank you!

petruccione@ukzn.ac.za

http://quantum.ukzn.ac.za