Cohort Project update

Cohort 4

Quantum Engineering CDT
University of Bristol

May 22, 2018

Figure: Roadmap

[1] Away-day presentation

Survey results

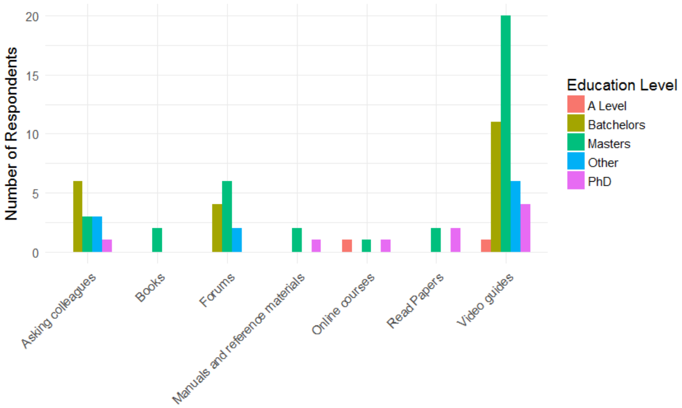


Figure: Results from onedrive [1]

[2] fig

Survey results

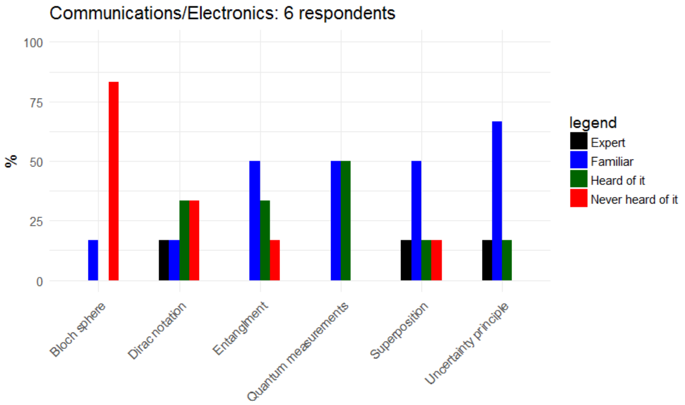
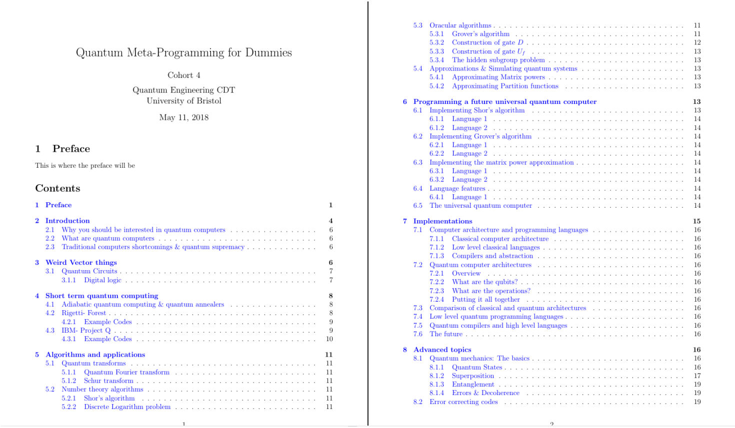


Figure: Results from onedrive [2]

Sections we plan to include



Quantum Meta-Programming for Dummies

Cohort 4

Quantum Engineering CDT
University of Bristol

May 11, 2018

1 Preface

This is where the preface will be

Contents

1

2

Figure: Document at [3]

[3] https://github.com/ot561/qprogramming/blob/master/main.pdf

Current work

Cohort Project update

Cohort 4

Quantum machine learning for data scientists

Dawid Kopczyk
Quantee Limited, Manchester, United Kingdom

This text aims to present and explain quantum machine learning algorithms to a data scientist in an accessible and consistent way. The algorithms and equations presented are not written in rigorous mathematical fashion, instead, the pressure is put on examples and step by step explanation of difficult topics. This contribution gives an overview of selected quantum machine learning algorithms, however there is also a method of scores extraction for quantum PCA algorithm proposed as well as a new cost function in feed-forward quantum neural networks is introduced. The text is divided into four parts: the first part explains the basic quantum theory, then quantum computation and quantum computer architecture are explained in section two. The third part presents quantum algorithms which will be used as subroutines in quantum machine learning algorithms. Finally, the fourth section describes quantum machine learning algorithms with the use of knowledge accumulated in previous parts.

25 Apr 2018

Contents

Quantum Algorithm Implementations for Beginners

Patrick J. Coles, Stephan Eidenbenz,* Scott Pakin, Adetokunbo Adedoyin, John Ambrosiano, Petr Anisimov, William Casper, Gopinath Chennupati, Carleton Coffrin, Hristo Djidjev, David Gunter, Satish Karra, Nathan Lemons, Shizeng Lin, Andrey Lokhov, Alexander Malyzhenkov, David Mascarenas, Susan Mniszewski, Balu Nadiga, Dan O'Malley, Diane Oyen, Lakshman Prasad, Randy Roberts, Phil Romero, Nandakishore Santhi, Nikolai Sinitsyn, Pieter Swart, Marc Vuffray, Jim Wendelberger, Boram Yoon, Richard Zamora, and Wei Zhu
Los Alamos National Laboratory, Los Alamos, New Mexico, USA

As quantum computers have become available to the general public, the need has arisen to train a cohort of quantum programmers, many of whom have been developing classic computer programs for most of their career. While currently available quantum computers have less than 100 qubits, quantum computer hardware is widely expected to grow in terms of qubit counts, quality, and connectivity. Our article aims to explain the principles of quantum programming, which are quite different from classical programming, with straight-forward algebra that makes understanding the underlying quantum mechanics optional (but still fascinating). We give an introduction to quantum computing algorithms and their implementation on real quantum hardware. We survey 20 different quantum algorithms, attempting to describe each in a succinct and self-contained fashion; we show how they are implemented on IBM's quantum computer; and in each case we discuss the results of the implementation with respect to differences of the simulator and the actual hardware runs. This article introduces computer scientists and engineers to quantum algorithms and provides a blueprint for their implementations.

10 Apr 2018

**Strawberry Fields:
A Software Platform for Photonic Quantum Computing**

Nathan Killoran, Josh Izaac, Nicolás Quesada, Ville Bergholm, Matthew Amy, and Christian Weedbrook

Xanadu, 372 Richmond St W, Toronto, M5V 1X6, Canada

We introduce Strawberry Fields, an open-source quantum programming architecture for light-based quantum computers. Built in Python, Strawberry Fields is a full-stack library for design, simulation, optimization, and quantum machine learning of continuous-variable circuits. The platform consists of three main components: (i) an API for quantum programming based on an easy-to-use language named Blackbird; (ii) a suite of three virtual quantum computer backends, built in NumPy and Tensorflow, each targeting specialized uses; and (iii) an engine which can compile Blackbird programs on various backends, including the three built-in simulators, and – in the near future – photonic quantum information processors. The library also contains examples of several paradigmatic algorithms, including teleportation, (Gaussian) boson sampling, instantaneous quantum polynomial, Hamiltonian simulation, and variational quantum circuit optimization.

h] 9 Apr 2018