

Cohort Project update

Cohort 4

Quantum Engineering CDT
University of Bristol

May 22, 2018

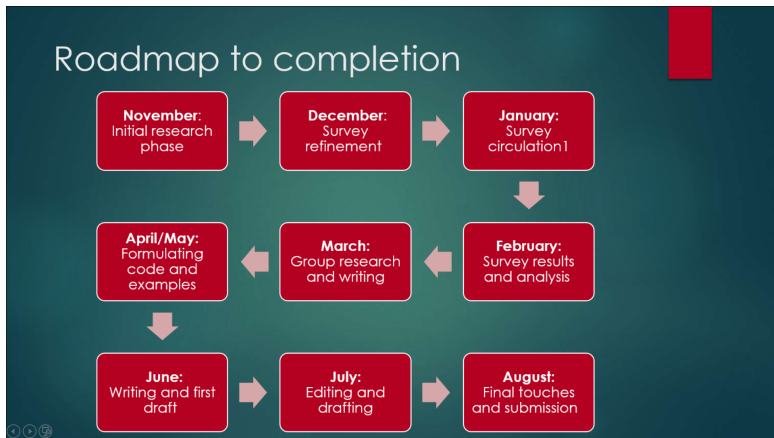


Figure: Roadmap

¹Away-day presentation

Survey results

Cohort
Project
update

Cohort 4

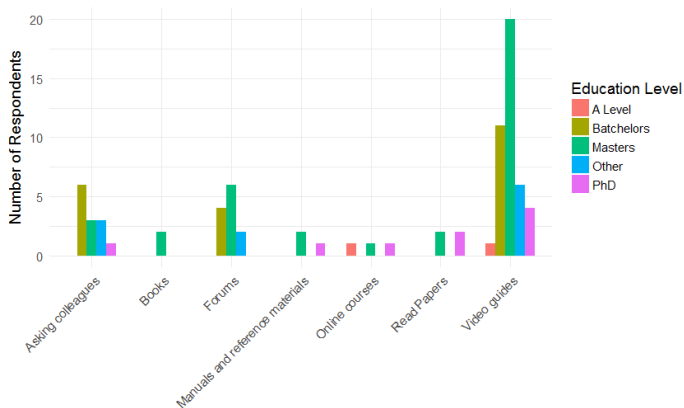


Figure: Results from onedrive ¹

Survey results

Cohort
Project
update

Cohort 4

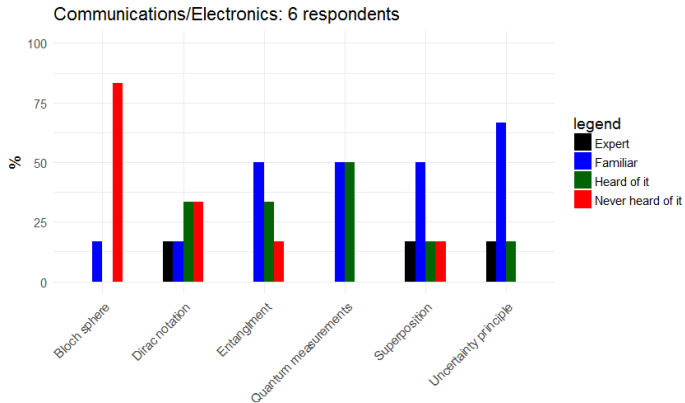


Figure: Results from onedrive ²

Sections we plan to include

Cohort
Project
update

Cohort 4

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 Preface	1
2 Introduction	4
2.1 Why you should be interested in quantum computers	6
2.2 What are quantum computers	6
2.3 Traditional computers shortcomings & quantum supremacy	6
3 Weird Vector things	6
3.1 Quantum Circuits	7
3.1.1 Digital logic	7
4 Short term quantum computing	8
4.1 Adiabatic quantum computing & quantum annealers	8
4.2 Rigetti Forest	8
4.2.1 Example Codes	9
4.3 IBM-Project Q	9
4.3.1 Example Codes	10
5 Algorithms and applications	11
5.1 Quantum transforms	11
5.1.1 Quantum Fourier transform	11
5.1.2 Scalar transform	11
5.2 Number theory algorithms	11
5.2.1 Shor's algorithm	11
5.2.2 Discrete Logarithm problem	11

5.3 Oracle algorithms	11
5.3.1 Grover's algorithm	12
5.3.2 Construction of gate D	13
5.3.3 Construction of gate U_f	13
5.3.4 The hidden subgroup problem	13
5.4 Approximations & Simulating quantum systems	13
5.4.1 Approximating Matrix powers	13
5.4.2 Approximating Partition functions	13
6 Programming a future universal quantum computer	13
6.1 Implementing Shor's algorithm	13
6.1.1 Language 1	14
6.1.2 Language 2	14
6.2 Implementing Grover's algorithm	14
6.2.1 Language 1	14
6.2.2 Language 2	14
6.3 Implementing the matrix power approximation	14
6.3.1 Language 1	14
6.3.2 Language 2	14
6.4 Language features	14
6.4.1 Language 1	14
6.5 The universal quantum computer	14
7 Implementations	15
7.1 Computer architecture and programming languages	16
7.1.1 Classical computer architecture	16
7.1.2 Low level classical languages	16
7.1.3 Compilers and abstraction	16
7.2 Quantum computer architectures	16
7.2.1 Overview	16
7.2.2 What are the options?	16
7.2.3 What are the operations?	16
7.2.4 Putting it all together	16
7.3 Comparison of classical and quantum architectures	16
7.4 Low level quantum programming languages	16
7.5 Quantum compilers and high level languages	16
7.6 The future	16
8 Advanced topics	16
8.1 Quantum mechanics: The basics	16
8.1.1 Quantum States	16
8.1.2 Superposition	17
8.1.3 Entanglement	19
8.1.4 Errors & Decoherence	16
8.2 Error correcting codes	19

Figure: Document at ³

Current work

Cohort
Project
update

Cohort 4

25 Apr 2018

Quantum machine learning for data scientists

David Kopecký
Queen's University, 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 notation, instead, the procedure is put on examples and step by step explanation of different topics. This contribution gives an overview of selected quantum machine learning algorithms, however there is also a method of feature 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 architectures 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.

Contents

I. Introduction

2

10 Apr 2018

Quantum Algorithm Implementations for Beginners

Patrick J. Côté, Stephan Eidenberg,[†] Scott Pakin, Adetokunbo Adedoyin, John Ambrose, Petr Anisimov, William Cragg, Gopaltha Chennupati, Carsten Coffrin, Hiroo Dojima, David Gao, Saeed Kaya, Nathan Leseau, Shuang Lin, Andrey Lokhov, Alexander Malyukov, David Mauerer, Susan Matsuoka, Bala Nalliah, Dan O'Malley, Dima Ouy, Lakshman Prasad, Randy Roberts, Phil Romero, Nandakishore Sankar, Nikolai Shikhov, Pieter Swart, Man Valling, Jan Wenzelberger, Idriss Youn, Richard Zou, 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 concise 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 implementation.

Strawberry Fields: A Software Platform for Photonic Quantum Computing

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

Quesada, 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