Senior Design Capstone 2021-2022

Department of Computer Science
UNIVERSITY OF COLORADO BOULDER

NORTHROP GRUMMAN

QUANTUM AI
AT THE EDGE

TEAM MEMBERS

Adam Hoerger
Alan Yu
Cade Gorman
Giovanni Visco
Jarek Reynolds
John Ortiz
Jorge Ortiz
Noah Svensson

Project Description

OBJECTIVE

>>> Apply Quantum AI to train a simple classifier network and then use that network for inference on an emulated embedded platform. If quantum techniques do not show improvement over traditional techniques, then use a traditionally trained model for inference on the emulator.

WHAT IS QUANTUM COMPUTING?

>>> Quantum computing harnesses the properties of quantum states, such as superposition, interference, and entanglement, to perform calculations. The most widely used model is the quantum circuit, based on the quantum bit, or "qubit."

WHAT IS A QUBIT?

>>> Qubits or quantum bits are the basic units of quantum information - the quantum version of the classic binary bit physically realized with two-state devices. A qubit can be in a 1 or 0 quantum state, or in a superposition of the 1 and 0 states.

QUANTUM COMPUTING POTENTIAL?

>>> Quantum computing has the potential to revolutionize computation by making certain types of classically intractable problems solvable.

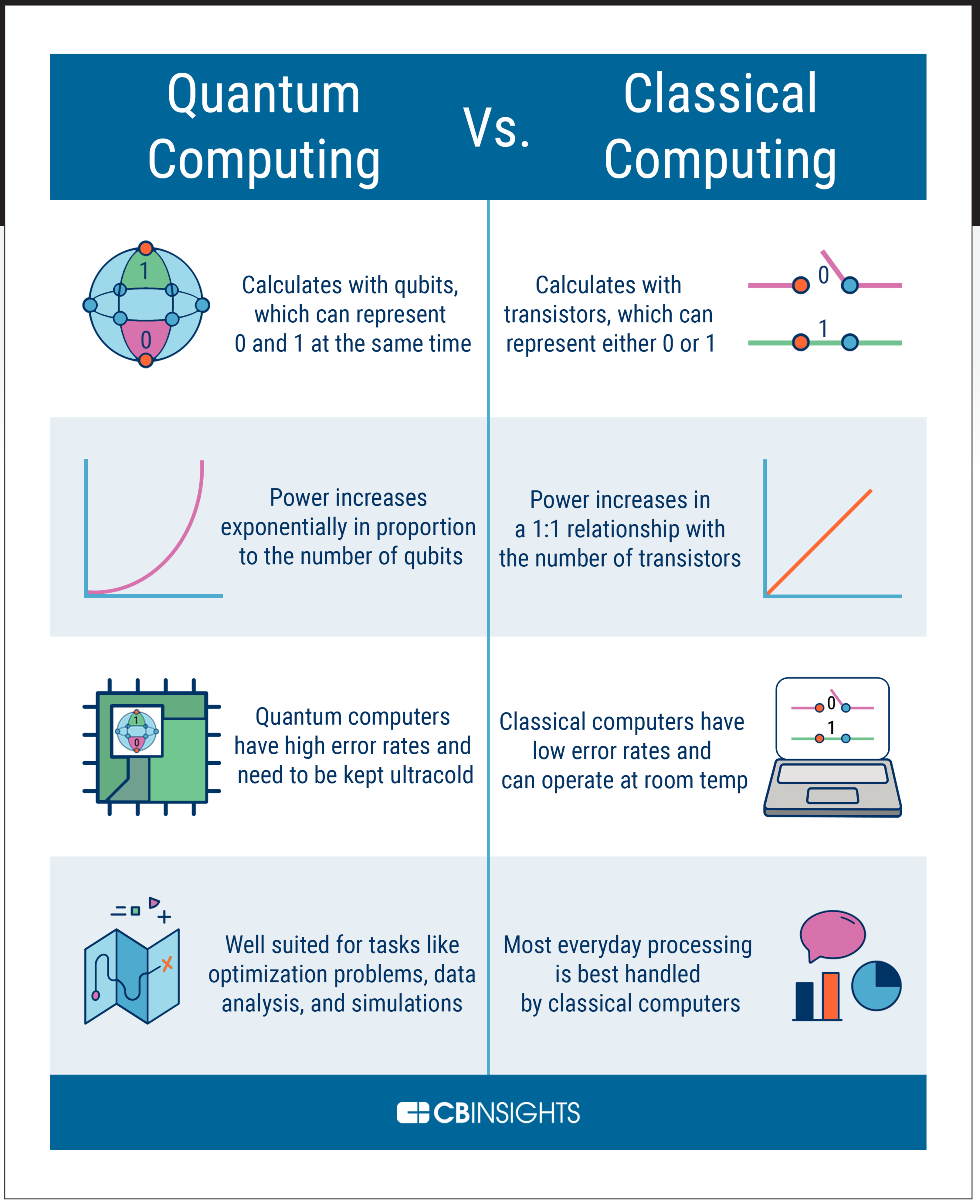
>>> While no quantum computer is yet sophisticated enough to carry out calculations that a classical computer can't, great progress is underway.

WHAT ABOUT QUANTUM AI?

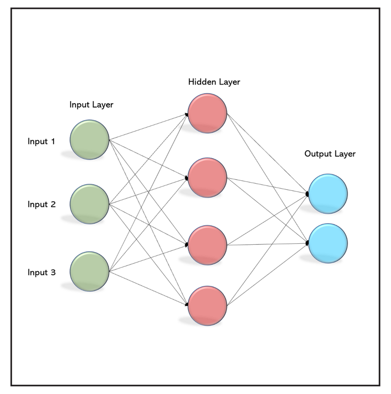
>>> Thanks to the computational advantages of quantum computing, quantum AI can help achieve results that are not possible to achieve with classical computers.

>>> Simple classifiers are starting to show promise on quantum processors, and these classifiers are deployable on edge processors for inference.

>>> However, Quantum AI is an immature technology, there are improvements in quantum computing that increase the potential of quantum AI. However, Quantum AI needs critical milestones to become a more mature technology such as:
>> Less error-prone, more powerful quantum computing systems.
>> Widely adopted open-source modeling and training frameworks.
>> Substantial and skilled developer ecosystem.
>> Compelling quantum AI applications that outperform their classical counterparts.

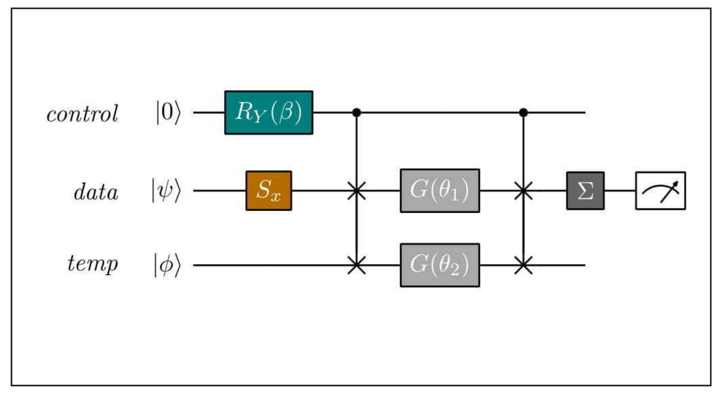


Source: CBINSIGHTS



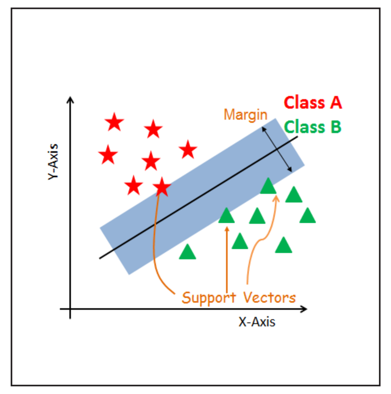
MLP

Multilayer Perceptrons (MLP) are fully connected classes of feed-forward artificial neural networks (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called back-propagation for training.



QUANTUM AI MODELS

The corresponding quantum versions of MLPs and SVMs have the potential to achieve at least quadratic speedup or even exponential speedup over the classical algorithms.



SVM

Support-Vector Machines (SVM) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. SVMs are one of the most robust prediction methods, being based on statistical learning frameworks. SVM training algorithms build models assigning new examples to one category or the other, making it a non-probabilistic binary linear classifier.

Findings

QUANTUM DATASET FINDINGS

>>> The Quantum SVMs were trained and tested in two to three seconds on the quantum datasets.

>>> The Classical SVMs were trained and tested in less than a second on the quantum datasets.

>>> The quantum SVM achieved 100% accuracy on the two quantum datasets, while the classical SVM reached only 62.5% and 90%. It should be noted that these datasets were artificially generated and are fully separable.

>>> Classical MLPs achieved low accuracies unless trained for a large number of epochs (at which point there was likely overfitting).

CLASSICAL DATASET FINDINGS

>>> The Classical MLP was trained in less than a minute while the classical SVM was trained in less than a second.

>>> The Quantum SVM achieved 90% accuracy on MNIST dataset, while classical SVM and MLP both achieved 99%

>>> The Quantum SVM required upwards of one hour to train and test MNIST dataset. However, the time also accounts for the time required to emulate quantum circuits.

>>> For the MNIST dataset, the quantum SVM was unable to classify between all ten digits accurately. It achieved an accuracy of 92% when classifying between two digits. The classical SVM was able to classify between all ten digits at >99% accuracy.

GENERAL FINDINGS

>>> Current physical limitations of quantum computing in terms of the number of qubits being capped, hold a hard constraint on speed up computation time

>>> Quantum methods demonstrated little to no benefit in accuracy, computation time, or convenience unless the data was known to be quantum in nature.

>>> Much of what we found for MLPs in particular with regards to quantum computing implementations is theoretical and in general a hybrid quantum model is used more often than a full quantum model. These hybrid models use both classical and quantum methods to get the best benefits of each.

>>> We suspect if we want to see a significant jump in time-complexity, more research milestones involved with utilizing different quantum gates/entanglements must be reached as quantum technology advances.

>>> Our stretch goal is testing our models using IBM backends located on IBMs physical boards. We suspect models running on physical boards will achieve higher variances and longer run times as it takes time compiling the circuit, executing gates, and resetting qubits.

Technical Challenges

>>> Qiskit is not well documented, existing documentation for certain workflows is scarce.

>>> Vitis hardware emulation is not well documented.

>>> Vitis AI does not currently support Scikit Learn workflows.

>>> Vitis AI does not currently support deep learning processor unit (DPU) hardware emulation, which prevented us from reaching our final deliverable.

>>> Transitioning from classical models to quantum models, while trying best to maintain similarity between layers proved challenging.

Tech Stack

learn | TensorFlow | python | NumPy | pandas | XILINX VITIS | Qiskit | jupyter