CASE STUDY

Quantum Computing: Applications for the Economy

Project: AutoQML Developer-Suite for automated Machine Learning with Quantum Computers

About AutoQML: The primary goal is to create a software package using the Python programming language. This package will facilitate the efficient transpiling of customer-provided circuits onto quantum computers yet to be specified.

The core objective of the collaboration between HQS and AutoQML was to develop a resilient software library capable of accomplishing two critical functions. Firstly, the library was designed to automatically transpile quantum machine learning circuits. Secondly, it aimed to allocate the qubits of each circuit to a suitable quantum backend. The crucial aspect of this process is its ability to dynamically respond to various factors.

These factors include the circuit‘s topology, the configuration of the backends, and the ever-changing noise models associated with each backend, potentially affected by calibration processes.

Challenge

The challenges in this project were twofold to ensure the quality of the quantum machines used, and also to consider the financial and time aspects.Initially, IonQ devices were chosen due to their high-quality qubits and all-to-all connections, which simplified the challenge of topology mapping. However, cost considerations led to a pivot towards IBM backends, which come with their unique topology of heavy-hex rings, but also faster execution times.

Solution

HQS adopted a multifaceted approach to solve these issues. Using the existing Qiskit library Mapomatic as a foundation, HQS engineered an additional heuristic algorithm, HQAA, developed in-house. Two operational modes were introduced.

Quality Mode: This mode scrutinizes a range of available backends to identify the most optimal one in terms of machine quality and configuration. While at the same time transpiring circuits for different backend families to find the best-suited topology.

Speed Mode: This mode is designed for scenarios where time is of the essence. It targets the backend with the shortest queue, allowing for quicker execution. Moreover, HQS incorporated specific adjustments for AutoQML’s unique toroidal circuit types. A specialized detection system was developed to identify backend topologies compatible with these circuit architectures.

Product and Services: The AutoSelectionBackend library is a versatile tool that abstracts complexities, offers a user-friendly approach, and provides flexibility to experts. It represents the culmination of a collaboration effort to make quantum computing more accessible and efficient, ultimately benefiting both newcomers and experts in the field.

Benefits and Partnerships: The integration of the AutoSelectionBackend library into AutoQML‘s sQUlearn framework benefits users by providing a more comprehensive and user-friendly quantum machine learning solution. It also strengthens partnerships and collaboration within the quantum machine learning community, potentially leading to advancements in the field.

Community: The AutoSelectionBackend library is not just a technical achievement; it‘s a community asset that bridges the gap between newcomers and experts in the field of quantum computing. It simplifies the process for those new to the technology and offers advanced users the tools to customize backend selection to meet their specific requirements.