CHAPTER

4

Quantum-Inspired Computing: Classical Approaches to Machine Learning

***Sathish Krishna Anumula[1], Niranjan Talluru[1], Rajesh Gangavarapu[1], Santosh Gupta[2] and Kalyan Tatavarthy[1]**

[1] Sr. Architect, IBM Corporation, USA
[2] Principal Architect, IBM Corporation, USA

This chapter provides a formal, self-contained exposition of quantum-inspired machine learning (QiML), a subfield focusing on applying ideas of quantum mechanics to the development of powerful classical algorithms. Our starting point is setting up a defensible taxonomy to dissociate QiML from actual quantum machine learning, and to describe its historical antecedents in simulated annealing and more recent dequantization of quantum algorithms. The highlights of the chapter pursue four central streams of inspiration systematically, namely: (1) quantum many-body physics is used to turn a model and its data into simple tensor networks; (2) quantum feature maps lead to randomized kernel methods, quantum inpainting, and quantum groups; (3) quantum numerical linear atomization algorithm (HHL) provides the template to develop efficient linear solvers and randomized numerical linear algebra; and (4) quantum approximate optimization algorithm (QAOA) is used to analyze non-convex optimization heuristics. In every region, we sketch the supporting theory, provide concrete algorithms along with quasi-pseudocode and complexity examples, and introduce theoretical guarantees. A large experimental section compares these approaches to classical baselines on standard data, as well as providing detailed case studies in resource-constrained ML, large-scale linear models, and combinatorial optimization. The chapter is rounded off by concluding with some practical hints and tips, a critical reflection on the ethical issues, and a discussion of open problems in the field, providing relevant literature to graduate students and practitioners in a well-brought-up and detailed piece of work.

*Corresponding author:

4.1 Introduction

The fusion of quantum computing and machine learning has been a source of enormous excitement, with the prospect of resolving problems that are beyond the scope of even the most powerful classical supercomputers (Biamonte et al., 2017). The working implications of this promise, however, are moderated by the considerable hardware difficulties of the current Noisy Intermediate-Scale Quantum (NISQ) age. Large-scale, fault-tolerant quantum computers are not yet on the horizon, so many quantum machine learning (QML) algorithms cannot yet be applied to real-world data. Such a technological fact has spurred the development of a parallel and heavily practical discipline: Quantum-inspired machine learning (QiML).

QiML is a different game: it seeks to translate the powerful mathematical principles and algorithmic structures of quantum mechanics to new algorithms that run on classical computers. Its guiding premise is that conceptual models from quantum computing, such as high-dimensional linear algebra, superposition, and entanglement, can inspire novel classical data processing methods that deliver improvements in computational performance, memory bandwidth, and model capabilities all without involving a single qubit. This chapter contains a graduate-level orientation of theory, methods, and applications of this emerging discipline.

4.1.1 Why Quantum-Inspired?

The main motivation for the development of QiML is the disconnect between expectations of the theoretical outcome of QML and the real constraints of available quantum hardware. Prominent quantum algorithms, including the Harrow-Hassidim-Lloyd (HHL) algorithm on solving linear systems of equations, assume such access to large, fault-tolerant quantum computers, which we do not currently possess. Some studies have found that, even under idealized simulations, the pursuit of quantum advantage in digital health, for example, still lacks decisively substantiated conclusions, and that most experiments lack the technical sophistication to account for noise and overhead costs (Harrow et al., 2009).

An attractive alternative is represented by quantum-inspired algorithms which exploit the intellectual fertility of quantum theory in the short term, in the classical domain. The present approach is based on two observations:

1. **Transferability of Algorithmic Knowledge:** Quantum mechanics first offers a radically new notation to compute in, based on the linear algebra of complex vector spaces. Such a view may demonstrate new structures and efficiency in classical problems. As an example, the structure of the Quantum Approximate Optimization Algorithm (QAOA) can inspire new forms of alternating optimization schemes, while quantum many-body mathematics provides tensor network techniques to compress high dimensional databases (Farhi et al., 2014). These represent mathematical and algorithmic discoveries that can be adapted for classical computing.

2. **The "Dequantization" Phenomenon:** A major development in QiML was the dequantization of several well-known QML algorithms. Landmark results by Ewin Tang and others demonstrated that the purported exponential speedups of certain quantum algorithms, such as those for recommendation systems, principal component analysis, and linear system solving, were not inherently dependent on uniquely quantum features like entanglement (Tang, 2019). Instead, these advantages arose from strong assumptions about the input data, namely that it could be efficiently stored in a low-rank representation and accessed through a specialized data structure known as Quantum Random Access Memory (QRAM). It was shown that classical algorithms employing randomized numerical linear algebra (RLA) techniques, such as ℓ_2-norm sampling, could achieve comparable performance under similar assumptions, thereby dequantizing the supposed quantum speedup (Drineas and Mahoney, 2016). This insight shifted attention away from claims of unconditional quantum advantage toward examining the structure of data that enables computational speedups, whether on classical or quantum computers (Martinsson and Tropp, 2020).

4.1.2 What Counts as "Quantum-Inspired" vs Truly Quantum?

The rather broad term quantum-inspired has been used to describe a wide range of classical algorithms. To navigate this landscape, it is essential to establish a clear taxonomy. A quantum-inspired classical algorithm is defined as one that draws on quantum mechanical principles, quantum algorithms, or mathematical formalisms of quantum physics in its design. These algorithms are implemented purely on classical hardware. This chapter addresses three major types of inspiration:

1. **Emulating Quantum Heuristics:** Classical algorithms that mirror the dynamic nature of quantum optimization procedures fall into this category. Han and Kim (2002, 2005) introduced the earliest quantum-inspired evolutionary algorithms (QIEAs), which employ concepts such as qubits and superposition to enhance population diversity in genetic algorithms. A more recent and striking example is the use of classical hardware to solve Quadratic Unconstrained Binary Optimization (QUBO) or Ising model problems, the native format of quantum annealers. Digital Annealers such as those developed by Fujitsu, represent a direct hardware embodiment of this approach.
2. **Transfer of Mathematical Formalisms:** This entails laying hands on mathematical tools that were created to deal with the complexity of quantum systems and utilizing them on classical machine learning. The most dominant one is the application of tensor networks (TNs), including Matrix Product States (MPS) that were initially developed in order to represent quantum many-body wavefunctions with low entanglement. In ML, the structures can be employed to compress the large weight tensors of a neural network, or can be used to approximate high-dimensional feature spaces, using the

assumption that the data underlying the models have a low-rank correlation structure, similar to low entanglement.

3. **Algorithmic Dequantization:** This is the most explicit way of inspiration in that a quantum algorithm is routinely optimized into a classical one. This often consists of determining the computational primitive at the heart of the quantum algorithm (e.g. quantum phase estimation in HHL) and attempting to identify a classical, usually randomized, subroutine equivalent that achieves a comparable result under particular data conditions (e.g. fast sampling techniques of RLA).

These should be separated, though, with true quantum machine learning that requires quantum hardware in order to run algorithms that utilize such quantum phenomena as superposition and entanglement in order to manipulate information. Although simulating a quantum circuit on a classical computer can be a useful tool in algorithm verification, it does not count as a quantum-inspired algorithm in the sense of using a scalable classical algorithm; the amount of work per operation scales exponentially in the number of qubits, exactly the bottleneck QiML wants to avoid.

4.1.3 Historical Arcs: From Simulated Annealing & Adiabatic Ideas to Tensor Networks and Sublinear Sampling

The history of intellectual origins of QiML does not represent anything that can be called a flow, but an aggregation of multiple strands of research. The initial thread is the physics-led optimization. Simulated annealing is a classical heuristic which is analogous to metallurgical annealing. It has a quantum mechanical analog though, quantum annealing, which substitutes thermal fluctuations with quantum tunnelling to traverse non-trivial energy landscapes, originally suggested in the late 1980s and developed formally in the 1990s (Kadowaki and Nishimori, 1998). This spawned the development of physical quantum annealers, and their classical analogs which can simulate this process.

Quantum Approximate Optimization Algorithm (QAOA) may be viewed as a gate-based and digitized variant of quantum adiabatic evolution, which in turn is a close cousin to quantum annealing. The main principle alternation between a problem Hamiltonian and a mixer Hamiltonian has spawned an entire family of classical algorithms alternating optimization.

A second type of transfer involves importing mathematical structures from quantum many-body physics. The density matrix renormalization group (DMRG) algorithm was recast in Matrix Product States (MPS) language in the 1990s and 2000s as a kind of tensor network, thus giving access to a robust method of simulating one-dimensional quantum systems. The discovery that these networks are particularly suited to approximate particular correlation structures (low entanglement) prompted their use in machine learning in the 2010s to tasks such as neural network compression and supervised classification, where the weight tensors were taken to also have low entanglement.

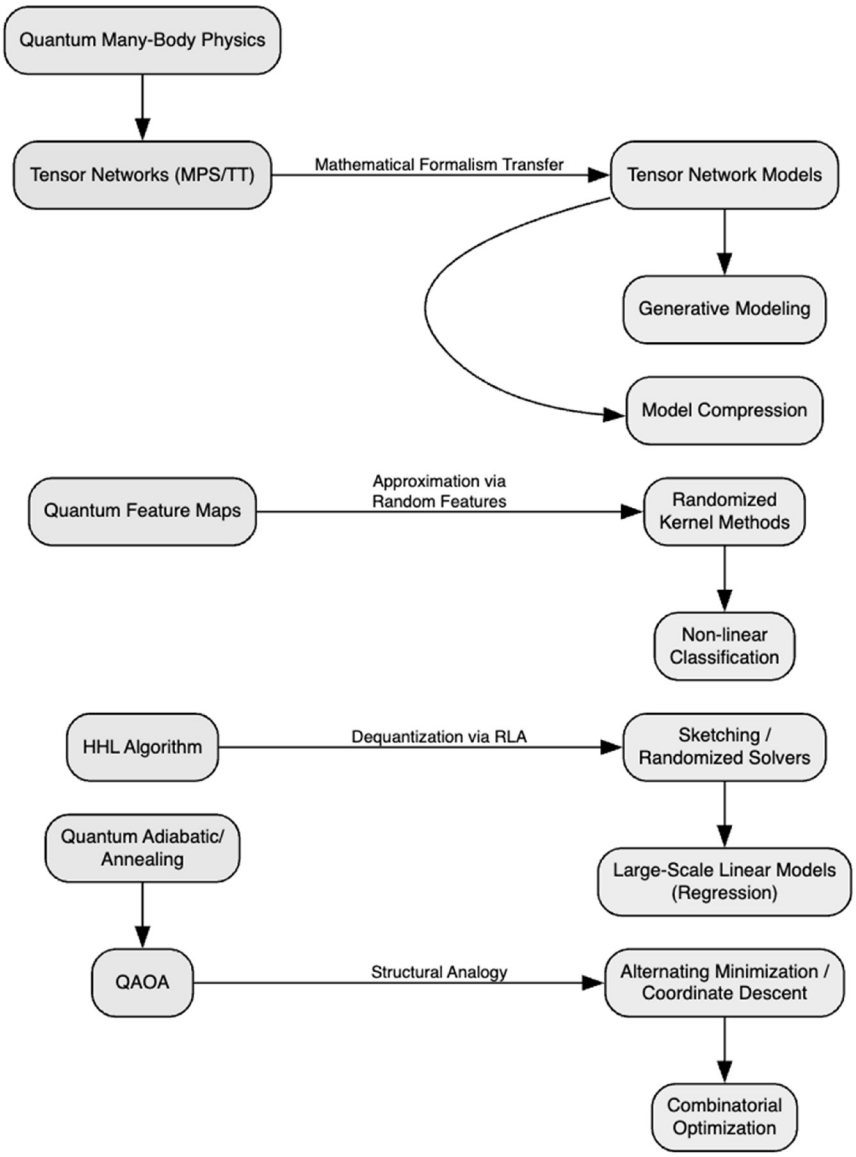

Figure 4.1: A taxonomy of quantum-inspired methods with indication of how quantum machine learning algorithms and formalisms can be converted to classical equivalents in machine learning tasks by intermediary areas such as Randomized Numerical Linear Algebra and Tensor Networks.

The third and most active arc is dequantization which really started in earnest around 2018. This thread was directly tied to the emergence of quantum algorithms designed for machine learning tasks, such as recommendation systems and the Harrow–Hassidim–Lloyd (HHL) algorithm for solving linear equations. It was a watershed moment when it was discovered that these quantum algorithms could not only be matched, but in some cases, surpassed by classical algorithms employing advanced techniques from Randomized Numerical Linear Algebra (RLA), including importance sampling. It showed that the nature of the speedup source was not strictly quantum but associated with data characteristics that can be used also by classical randomized algorithms, therefore making certain that RLA is a hallmark of present-day QiML. The three arcs of optimization heuristics, mathematical formalisms, and algorithmic translation have become the main pillars of the field (as shown in Figure 4.1).

4.2 Core Quantum-Inspired Ideas for ML

In this section, we go into the key conceptual foundations of quantum-inspired machine learning. The three pillars include the different routes through which quantum principles have been brought into classical algorithmic strategies. The focus is on developing intuition for these methods and relating them to the algorithms discussed in Section 3. One of the unifying threads is the adaptation of quantum concepts to find effective methods of data representation, function approximation, and optimization concepts such as quantum states, dynamics, and measurement.

4.2.1 Tensor-Network Representations for Models and Data

The original seed of the first core idea lies in quantum many-body physics, where tensor networks were developed to effectively describe complex quantum states that inhabit an exponentially large Hilbert space yet occupy a physically small, though not negligible region of it. The key insight is that the states of interest, such as ground states of local Hamiltonians, are constrained to possess only a finite amount of entanglement, and can therefore be compressed into a network of smaller, entangled tensors.

Quantum States to ML Models: In machine learning, this corresponds to using a tensor network to represent the weight parameters of a model or features of a dataset. As an example, the weight vector of a one-dimensional linear model with d features can be restructured into an order-d tensor W. In the case it can be approximated using a low-rank representation via Tensor Train (TT) or Matrix Product State (MPS) decomposition, the parameter count goes from exponential in d to polynomial (Oseledets, 2011).

Consider a full tensor representation as shown in Eq. 4.1:

$$W \epsilon, R^{n_1 \times n_1 \times \ldots \times n_d} \tag{4.1}$$

which requires storage on the order as given in Eq. 4.2:

$$O\left(\prod_{k=1}^{d} n_k\right) \tag{4.2}$$

By contrast, if W is represented in the **Tensor Train (TT) format** with cores $\{G_k\}_{k=1}^{d}$, *t*. the storage requirement reduces to

$$O\left(\sum_{k=1}^{d} n_k r_{k-1} r_k\right) \tag{4.3}$$

where r_k are the TT-ranks with $r_0 = r_d = 1$

This compression is particularly effective for data with an inherent sequential or local structure, such as time series, text, or pixels in an image, where the correlations between distant features are weak, a direct analog to the "area law" of entanglement in physics.

Expressivity vs. Contraction Cost: The Expressivity of a tensor network model is determined by the bond dimensions (TT-ranks). Increased ranks enable the network to reflect more complex functions and longer-distance correlations; however, this comes at an increasingly high computational cost. The operations involved, including making a prediction (contracting the network with a feature tensor), scale polynomially with bond dimensions. This introduces a fundamental trade-off: rank must be large enough to capture the relevant structure in the data, yet small enough to keep the computation from becoming overly demanding.

Use in Classifiers and Generative Models:

- **Classifiers:** A linear classifier can also be trained by updating the small core tensors to learn the TT-format representation of the weight tensor. This approach can substantially reduce the number of individually learned parameters, and, in principle, improve generalization, since adopting a simple form such as the TT-decomposition serves as a form of structural regularization.
- **Generative Models:** Tensor networks can also be used to represent probability distributions in high-dimensional spaces. A TT-based model or an autoregressive matrix product state (AMPS) can efficiently compute normalized probability distributions, and enable direct, unbiased sampling, giving them potential as generative models that may compete with current state-of-the-art neural network approaches.

4.2.2 Quantum-Inspired Kernels and Random Features

Kernel methods, such as Support Vector Machines (SVMs), are highly effective non-linear learning procedures that employ a kernel function $k(x, y)$ to compute inner products in a high-dimensional feature space, without explicitly constructing the feature vectors (Schuld, 2021). Quantum kernel techniques extend this concept by estimating the kernel on a quantum computer, where the kernel is defined as the inner product of quantum states representing the classical data:

$$k(x,y) = |\langle \phi(x) \; |\phi(y)\rangle|^2$$

where $|\phi(x)\rangle$ is the quantum state corresponding to data point x. The hope is that quantum feature maps can access feature spaces intractable for classical methods, thereby providing a quantum advantage.

Quantum Feature Maps to Classical Random Features: The quantum-inspired suggestion arises from classical random feature techniques, which provide a means of explicitly approximating kernel features. By Bochner's theorem, the envelope of any shift-invariant kernel $k(x, y) = k(x\ y)$ can be expressed as the Fourier transform of a probability distribution density. Rahimi and Recht (2007) proposed an approach related to the random features method, under which the kernel is approximated by sampling frequencies ω (equivalently, by sampling $p(\omega)$) and building an explicit, low-dimensional feature map $z(x)$ as shown in Eq. 4.4:

$$k(x, y) \approx z(x)^T z(y) \text{ where } z(x) = \sqrt{\frac{2}{D}}^T \tag{4.4}$$

This enables one to realize fast linear models in the feature space $z(x)$ to approximate the much slower kernel machine.

The relationship with quantum models is that, in many proposed quantum feature maps, the resulting kernel corresponds to a classical random feature that can be readily approximated using a Fourier or orthogonal basis. This process dequantizes the quantum kernel, yielding a classically computable algorithm with comparable expressive power. While the quantum formalism naturally inspires new types of random feature constructions, the resulting algorithm remains classically computable.

Approximation Guarantees and Memory Footprint: Random feature methods approximate kernels using a feature dimension D. The greater the number of features, the better the approximation, with error decreasing approximately as $1/\sqrt{D}$. To reach accuracy ϵ, one needs about $1/\epsilon^2$ features. The memory cost is $n \times D$, which is far smaller than storing the full $n \times n$ kernel matrix when $D << n$.

4.2.3 HHL-Inspired Linear Solvers

Quantum HHL algorithm also alluded to exponential speedup in solving the linear systems of equations, but only in constrained circumstances. Its classical version was a result of randomized linear algebra (RLA) (Woodruff, 2014). In this case, a small data sketch is used to reduce the system, thus becoming cheaper to solve. The sketch gives a reasonably good solution that is easy to understand, given its careful design. This trade-off reflects the dependence of accuracy on the size of the sketch, while computation is accelerated by several orders of magnitude. Moreover, sketch preconditioning may also speedup traditional solvers such as the conjugate gradient method.

4.2.4 QAOA/Adiabatic-Inspired Optimization

QAOA adopts two processes in turn, one which optimizes cost, and one which searches the solution space. This resembles the alternating minimization method of classical optimization, where the blocks of variables and the minimizations alternate (Karimi et al., 2016). Several types of ML problems – such as feature selection or graph partitioning – can be posed as Ising or QUBO problems. Quantum algorithms are aimed at such formulations, but usually classical heuristics, including simulated annealing, tabu search or message passing, are also effective. The motivation is the main point to regard ML tasks as energy-minimization and devise nesting, alternating heuristics.

4.2.5 Quantum Walks-Inspired Sampling and Diffusion

Quantum walks are used to propagate over graphs at a faster pace compared to conventional random walks, and it has given rise to new methods of faster sampling. In machine learning, this corresponds to improved MCMC designs, with proposal moves that mix states more rapidly and reduce sample-to-sample correlation. The aim is to replicate the effect of quantum interference, e.g., via symmetries or reflection-like steps, such that the sampler walks over complex distributions more rapidly than ordinary random walks. Figure 4.2 maps design patterns for selecting a quantum-inspired approach.

4.3 Algorithms

In this section, we outline simplified versions of algorithms inspired by the quantum principles described earlier. The focus is on practical sketches that convey the mechanics without excessive mathematical detail.

4.3.1 Tensor Train Classifier

Idea: Compress the weight tensor of a linear model into a Tensor Train (TT) format to reduce parameters and improve efficiency.

Algorithm 1

1. Input: Training data $X \in R\hat{}(n \times d)$, labels y
2. Initialize TT-cores $\{G_1, G_2, \ldots, G_d\}$ with small ranks
3. For each epoch
4. For each sample (x, y)
5. Construct a feature tensor from x
6. Predict $\hat{y}$ = Contract (feature tensor, TT-cores)
7. Compute loss $L(\hat{y}, y)$
8. Update TT-cores via gradient step
9. Output: TT-cores representing classifier weights

Note: Training scales polynomially in TT-rank; compactness acts as structural regularization.

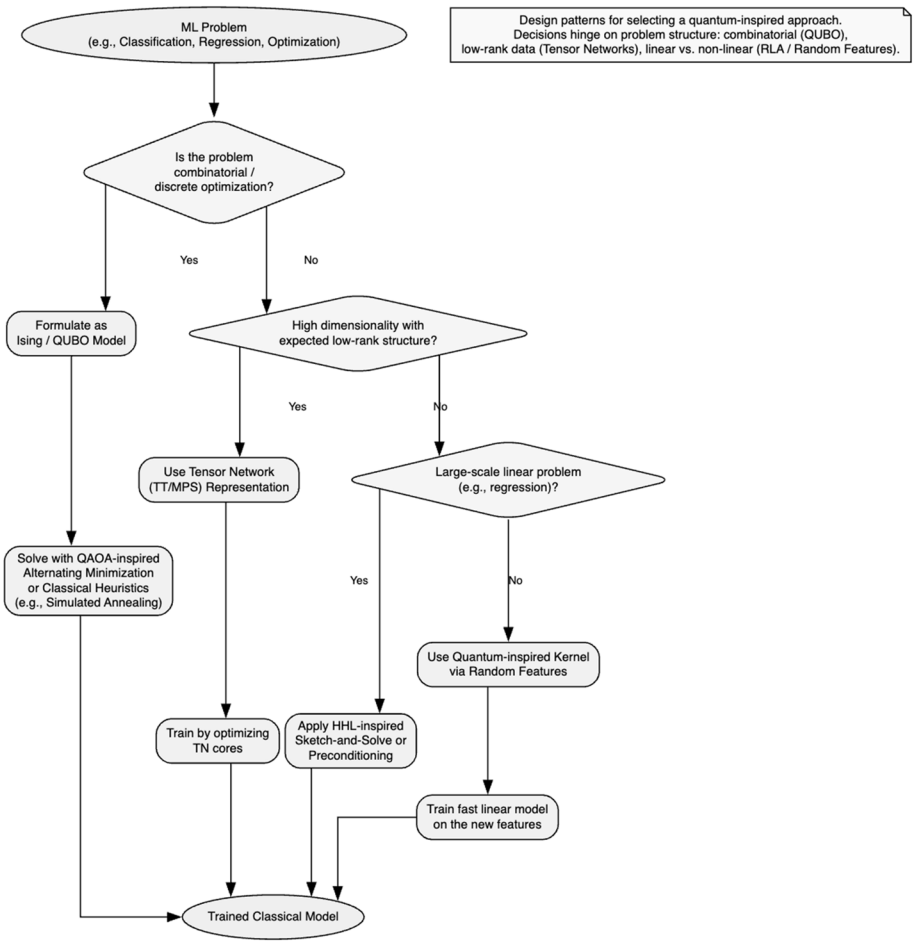

Figure 4.2: Patterns to choose a quantum-inspired approach. This flow chart offers a framework for decision-making regarding the solution of mapping a given machine learning task to a suitable strategy inspired by quantum ideas relative to the structure of such a problem, including its combination, dimension, and linearity.

4.3.2 Random Features Kernel Approximation

Idea: Approximate a kernel using observation with explicit random features, which allows for fast linear training.

Algorithm 2

1. Input: Data points $\{x_1, \ldots, x_n\}$, kernel $k(\cdot,\cdot)$, feature dimension D
2. Sample D random frequencies $\{\omega_1, \ldots, \omega_D\}$ from distribution p
3. For each x_i
4. Construct feature map $z(x_i) = [\cos(\omega_1 \hat{} T x_i), \ldots, \cos(\omega_D \hat{} T x_i)]$

5. Train linear model w on $\{z(x_i), y_i\}$
6. Output: Trained linear model in feature space

Note: Requires $O(n_D)$ memory, much less than $O(n^2)$ for kernel matrix when $D << n$.

4.3.3 Sketch-and-Solve Linear Solver

Idea: Solve faster, a big linear system $Ax = b$, by reducing it to a smaller sketched system.

Algorithm 3

1. Input: Matrix $A \in R\hat{}(n \times d)$, vector b $\in R\hat{}n$, sketch size m
2. Generate sketching matrix $S \in R\hat{}(m \times n)$
3. Compute sketched system: $A' = S\,A$, $b' = S\,b$
4. Solve smaller system: $(A'\hat{}T\,A')\,w \approx A'\hat{}T\,b'$
5. Return approximate solution w

Note: Accuracy improves with sketch size m; it works well for well-conditioned systems.

4.3.4 QAOA-Inspired Alternating Minimization

Idea: Take turns between taking cost improvement and going through exploration/mixing.

Algorithm 4

1. Input: Initial solution x_0, cost function C, mixing operator M
2. Initialize $x \leftarrow x_0$
3. For $t = 1 \ldots T$
4. $x \leftarrow$ ImproveCost (x, C)
5. $x \leftarrow$ MixSolution (x, M)
6. Return the best solution found

4.3.5 Quantum Walks-Inspired Sampler

Idea: Modify Markov Chain Monte Carlo (MCMC) with reflection/mixing steps inspired by quantum walks.

Algorithm 5

1. Input: Target distribution π, initial state x_0
2. For $t = 1 \ldots T$
3. Propose a new state x' using the reflection/mixing rule
4. Accept x' with probability $\min(1, \pi(x')/\pi(x))$
5. Update chain state
6. Return sampled states $\{x_1, \ldots, x_T\}$

4.4 Applications and Case Studies of Quantum-Inspired Machine Learning

Although quantum-inspired algorithms originate from abstract mathematical analogies, their most convincing practical applications have emerged in machine learning,where they have been tested in real world settings. In this section it is therefore important to highlight the tangible areas in which QiML techniques have already demonstrated benefits over standard baselines.

4.4.1 Recommendation Systems

Among the most compelling success stories is the quantum-inspired recommendation algorithm by Tang (2019), which showed that classical randomized linear-algebra techniques can match the performance of a quantum algorithm previously believed to offer exponential speedups. QiML methods can scale recommendation models to millions of users and items, enhance financial and computational efficiency through low-rank structure in user-item matrices, and leverage efficient 2-norm sampling.

4.4.2 Resource-Constrained Machine Learning

The value of a tensor-network-based QiML approach lies in memory- or compute-constrained environments i.e., embedded systems or the edge. As an example, MPS or TT representations with parameter reductions of tens to hundreds of thousands are possible with neural networks having correspondingly fewer weight matrices, with accurate models retaining the same expressiveness. Low-rank approximation has been used in image classification and natural language processing, in both cases to minimize storage space since the approximation does not cost predictive power.

4.4.3 Large Linear Models

Large-scale regression, classification, and scientific computing problems are particularly applicable in HHL-inspired linear solvers and sketch-and-solve approaches. These solvers save massive training time using sketches that can be made in a low dimension and can be just as accurate. Practically, randomized sketching has been used in climate modeling, genomic prediction, and financial forecasting, where millions of features would make the equivalent methods computationally infeasible (Halko et al., 2011).

4.4.4 Optimization Combinatorial

Many machine learning tasks may be reduced to a combinatorial optimization framework like Quadratic Unconstrained Binary Optimization (QUBO). Classical alternatives are given by quantum-inspired heuristics extending QAOA or annealing to these settings, which prove to be effective. The case studies

presented are: feature selection on high-dimensional data, graph partitioning (community detection), and logistic network scheduling.

4.4.5 Generative Models Sampling

Sampling efficiency-enhancing quantum-walk-inspired techniques have been derived in Markov Chain Monte Carlo (MCMC), which enhance Bayesian inference and generative modeling. As an example, autoregressive matrix product states (AMPS) have demonstrated the ability to draw high-dimensional probability distributions with unbiased sampling, as an alternative to deep generative networks.

In each of these areas, recommendation systems, resource-constrained ML, large-scale linear models, combinatorial optimization, and generative modeling, quantum-inspired approaches have demonstrated meaningful gains in scalability, efficiency, or interpretability relative to fully classical baselines. These studies illustrate that QiML is not merely a theoretical curiosity, but an emerging, practical toolbox already capable of delivering value in real-world machine learning applications.

4.5 Conclusion

This chapter has traversed the terrain of quantum-inspired machine learning, an area created at the confluence of the profound algorithmic lessons of quantum theory and the scale requirements of classical, large-scale computing. As we know already, QiML is not a monoculture but rather a gathering of these intellectual traditions: the translation of mathematical tools of quantum computation, such as tensor networks, the quantization of particular quantum algorithms, such as HHL in the language of randomized numerical linear algebra, and importation of structural analogies in quantum heuristics, such as QAOA.

The main point is that the inspiration from quantum mechanics is not just a source of new heuristics but rather it yields a principled methodology for the construction of classical algorithms that prove especially productive in exploring certain data structures (Aggarwal, 2020). The strengths of tensor networks include data sets where information or parameters in the model have low-rank connections, like those of the area law entanglement in physical systems. The HHL-inspired solvers prefer large over completed linear systems, where the statistical characterization is very well described by a sketch. Quantum-inspired kernel methods harness the power of random features to scale the classification problem to non-convex, non-convexity to classify using kernel methods. QAOA-inspired optimizers serialize efficiently, parameterizing a difficult, non-convex combinatorial problem with a group structure.

Not every method is, however, advantageous. Their utility depends on whether the algorithm's implicit assumptions of the algorithm align with the structural peculiarities of the problem at hand. A tensor-network classifier, for example,

will fail when the data exhibits an incompressible correlation structure, while a sketch-and-solve technique offers only marginal benefits on small, dense matrices. Practitioners are therefore responsible for diagnosing the structural characteristics of their data, its effective rank, sparsity, connectivity patterns, and selecting the appropriate tool from the QiML toolbox. The route between a quantum idea and a feasible classical algorithm is one of translation and approximation, and it brings an obligation of intellectual integrity. We as a field need more discipline in our claims, properly separating the demonstrated vineyard of efficiency advantages of these classical strategies from the unfulfilled promise of fault-tolerant quantum computing. What is important about QiML is not necessarily to ape quantum mechanics per se, but to apply the notation and machinery of the quantum setting to bring back new, efficient, and provably effective classical algorithms that will open new horizons of machine learning today. The principles analyzed in this chapter will thus remain core despite further development of classical and quantum hardware and offer a lasting connection between the two exciting fields of computation.

References

Aggarwal, C. C. (2020). Linear Algebra and Optimization for Machine Learning: A Textbook. Springer Nature. DOI: 10.1007/978-3-030-40344-7.

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N. & Lloyd, S. (2017). Quantum machine learning. *Nature*, 549(7671), 195–202. DOI: 10.1038/nature23474.

Drineas, P. & Mahoney, M. W. (2016). RandNLA: Randomized numerical linear algebra. *Communications of the ACM*, 59(6), 82–90. DOI: 10.1145/2897955.

Farhi, E., Goldstone, J. & Gutmann, S. (2014). A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028.

Halko, N., Martinsson, P. G. & Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2), 217–288. DOI: 10.1137/090771806.

Han, K. H. & Kim, J. H. (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, 6(6), 580–593. DOI: 10.1109/TEVC.2002.804320.

Harrow, A., Hassidim, A. & Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15), 150502. DOI: 10.1103/PhysRevLett.103.150502.

Kadowaki, T. & Nishimori, H. (1998). Quantum annealing in the transverse Ising model. *Physical Review E*, 58(5), 5355. DOI: 10.1103/PhysRevE.58.5355.

Karimi, H., Nutini, J. & Schmidt, M. (2016). Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 795–811). DOI: 10.1007/978-3-319-46227-1_50.

Martinsson, P.-G. & Tropp, J. A. (2020). Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29, 403–572. DOI: 10.1017/S096249292000004X.

Oseledets, I. V. (2011). Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5), 2295–2317. DOI: 10.1137/090752286.

Rahimi, A. & Recht, B. (2007). Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20, 1177–1184.

Schuld, M. (2021). Supervised quantum machine learning models are kernel methods. arXiv preprint arXiv:2101.11020.

Stoudenmire, E. M. & Schwab, D. J. (2016). Supervised learning with tensor networks. *Advances in Neural Information Processing Systems*, 29, 4799–4807.

Tang, E. (2019). A quantum-inspired classical algorithm for recommendation systems. *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (*STOC '19*) (pp. 217–228). Doi: 10.1145/3313276.3316310.

Woodruff, D. P. (2014). Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2), 1–157. DOI: 10.1561/0400000060.