entropy

MDPI

Article

Quantum Computing Approaches for Vector Quantization—Current Perspectives and Developments

Alexander Engelsberger * and Thomas Villmann

Saxon Institute for Computational Intelligence and Machine Learning (SICIM), University of Applied Sciences Mittweida, Technikumplatz 17, 09648 Mittweida, Germany; villmann@hs-mittweida.de
* Correspondence: engelsbe@hs-mittweida.de

Abstract: In the field of machine learning, vector quantization is a category of low-complexity approaches that are nonetheless powerful for data representation and clustering or classification tasks. Vector quantization is based on the idea of representing a data or a class distribution using a small set of prototypes, and hence, it belongs to interpretable models in machine learning. Further, the low complexity of vector quantizers makes them interesting for the application of quantum concepts for their implementation. This is especially true for current and upcoming generations of quantum devices, which only allow the execution of simple and restricted algorithms. Motivated by different adaptation and optimization paradigms for vector quantizers, we provide an overview of respective existing quantum algorithms and routines to realize vector quantization concepts, maybe only partially, on quantum devices. Thus, the reader can infer the current state-of-the-art when considering quantum computing approaches for vector quantization.

Keywords: vector quantization; quantum machine learning; prototype-based learning

Citation: Engelsberger, A.; Villmann, T. Quantum Computing Approaches for Vector Quantization—Current Perspectives and Developments. *Entropy* **2023**, *25*, 540. https://doi.org/10.3390/e25030540

Academic Editors: Rosario Lo Franco and GuiLu Long

Received: 11 February 2023
Revised: 15 March 2023
Accepted: 20 March 2023
Published: 21 March 2023

1. Introduction

Quantum computing is an emerging research field, and the current wave of novelties is driven by advances in building quantum devices. In parallel to this hardware development, new quantum algorithms and extensions of already known methods like Grover search emerged during the last few years, for example, for graph problems [1] or image processing [2]. One field of growing interest is Quantum Machine Learning. On the one hand, we can consider quantum algorithms to accelerate classical machine learning algorithms [3,4]. On the other, machine learning approaches can be used to optimize quantum routines [5].

In this paper, we focus on the first aspect. In particular, we consider the realization of unsupervised and supervised vector quantization approaches by means of quantum routines. This focus is taken because vector quantization is one of the most prominent tasks in machine learning for clustering and classification learning. For example, (fuzzy-) k-means or its more modern variants k-means and neural gas constitute a quasi-standard in an unsupervised grouping of data, which frequently is the starting point for sophisticated data analysis to reduce the complexity of those investigations [6–8]. The biologically inspired self-organizing map is one of the most prominent tools for visualization of high-dimensional data, based on the concept of topology preserving data mapping [9–12]. In the supervised setting, (generalized) learning vector quantization for classification learning is a powerful tool based on intuitive learning rules, which, however, are mathematically well-defined such that the resulting model constitutes an adversarial-robust large margin classifier [13–15]. Combined with the relevance learning principle, this approach provides a precise analysis of the data features weighting for optimal performance, improving classification decision interpretability and, hence, allows causal inferences to interpret the feature influence for the classification decision [12,16,17].

Further, the popularity of vector quantization methods arises from their intuitive problem understanding and the resulting interpretable model behavior [8,10,18,19], which frequently is demanded for acceptance of machine learning methods in technical or biomedical applications [20–22]. Although these methods are of only lightweight complexity compared to deep networks, frequently sufficient performance is achieved.

At the same time, the current capabilities of quantum computers only allow a limited complexity of algorithms. Hence, the implementation of deep networks is currently not realistic apart from any mathematical challenges for realization. Therefore, vector quantization methods became attractive for the investigation of corresponding quantum computing approaches, i.e., respective models are potential candidates to run on the limited resources of a quantum device.

To do so, one can either adopt the mathematics of quantum computing for quantum-inspired learning rules to vector quantization [23], or one gets motivation from existing quantum devices to obtain quantum-hybrid approaches [24,25].

In this work, we are considering vector quantization approaches for clustering and classification in terms of their adaptation paradigms and how they could be realized using quantum devices. In particular, we discuss model adaptation using prototype shifts or median variants for prototype-based vector quantization. Further, unsupervised and supervised vector quantization is studied as a special case of set-cover problems. Finally, we also explain an approach based on Hopfield-like associative memories. Each of these adaptation paradigms comes with advantages and disadvantages depending on the task. For example, median or relational variants come into play if only proximity relations between data are available but with reduced flexibility for the prototypes [26,27]. Vector shift adaptation relates to Minkowski-like data spaces with corresponding metrics, which usually provide an obvious interpretation of feature relevance if combined with a task depending on adaptive feature weighting. Attractor networks like the Hopfield model can be used to learn categories without being explicitly trained on them [28]. The same is true of cognitive memory models [29], which have great potential for general learning tasks [30].

Accordingly, we subsequently examine which quantum routines are currently available to realize these adaptation schemes for vector quantization adaptation completely or partially. We discuss the respective methods and routines in light of the existing hardware as well as the underlying mathematical concepts. Thus, the aim of the paper is to give an overview of quantum realizations of the adaptation paradigms of vector quantization.

2. Vector Quantization

Vector Quantization (VQ) is a general motif in machine learning and data compression. Given a data set $\mathcal{X} \subset \mathbb{R}^n$ with $|\mathcal{X}| = N$ data points $\boldsymbol{x}_i$, the idea of VQ is representing $\mathcal{X}$ using a much smaller set $\mathcal{W} \subset \mathbb{R}^n$ of vectors $\boldsymbol{w}_i$, where $|\mathcal{W}| = M \ll N$. We will call these vectors prototypes; sometimes, they are also referred to as codebook vectors. Depending on the task, the prototypes are used for pure data representation or clustering in unsupervised learning, whereas in the supervised setting, one has to deal with classification or regression learning. A common strategy is the nearest prototype principle for a given data $\boldsymbol{x}$ realized using a winner takes all rule (WTA-rule), i.e.,

$$s(\boldsymbol{x}) = \operatorname{argmin}_{j=1,\ldots,M}\left(d\left(\boldsymbol{x}, \boldsymbol{w}_j\right)\right) \in \{1, \ldots, M\} \tag{1}$$

for a given dissimilarity measure d in $\mathbb{R}^n$ and where $\boldsymbol{w}_s$ is denoted as the winning prototype of the competition. Hence, an appropriate choice of the metric d in use seriously influences the outcome of the VQ approach. Accordingly, the receptive fields of the prototypes are defined as

$$R\left(\boldsymbol{w}_j\right) = \{\boldsymbol{x}_i \in \mathcal{X} | s(\boldsymbol{x}_i) = j\}$$

with $\mathcal{X} = \cup_{j=1}^{M} R\left(\boldsymbol{w}_j\right)$.

2.1. Unsupervised Vector Quantization

Different approaches are known for optimization of the prototype set $\mathcal{W}$ for a given dataset $\mathcal{X}$, which are briefly described in the following. In the unsupervised setting, no further information is given.

2.1.1. Updates Using Vector Shifts

We suppose an energy function

$$E_{VQ}(\mathcal{X}, \mathcal{W}) = \sum_{i=1}^{N} E_{VQ}(\boldsymbol{x}_i, \mathcal{W})$$

with local errors $E_{VQ}(\boldsymbol{x}_i, \mathcal{W})$ to be assumed as differentiable with respect to the prototypes and, hence, the dissimilarity measure d is also supposed to be differentiable. Further, the prototype set $\mathcal{W}$ is randomly initialized. Applying the stochastic gradient descent learning for prototypes, we obtain the prototype update

$$\Delta \boldsymbol{w}_j \propto -\frac{\partial E_{VQ}(\boldsymbol{x}_i, \mathcal{W})}{\partial d(\boldsymbol{x}_i, \boldsymbol{w}_j)} \cdot \frac{\partial d(\boldsymbol{x}_i, \boldsymbol{w}_j)}{\partial \boldsymbol{w}_j}$$

for a randomly selected sample $\boldsymbol{x}_i \in \mathcal{X}$ [31]. If the squared Euclidean distance $d_E(\boldsymbol{x}, \boldsymbol{w}_j) = (\boldsymbol{x} - \boldsymbol{w}_j)^2$ is used as the dissimilarity measure, the update obeys a vector shift

$$\frac{\partial d_E(\boldsymbol{x}_i, \boldsymbol{w}_j)}{\partial \boldsymbol{w}_j} = -2(\boldsymbol{x} - \boldsymbol{w}_j)$$

attracting the prototype $\boldsymbol{w}_j$ towards the presented data $\boldsymbol{x}_i$.

Prominent in those algorithms is the well-known online k-means or its improved variant, the neural gas algorithm, which makes use of prototype neighborhood cooperativeness during training to accelerate the learning process as well as for initialization insensitive training [8,32].

Further, note that similar approaches are known for topologically more sophisticated structures like subspaces [33].

2.1.2. Median Adaptation

In median VQ approaches, the prototypes are restricted to be data points, i.e., for a given $\boldsymbol{w}_j$ exists a data sample $\boldsymbol{x}_i$ such that $\boldsymbol{w}_j = \boldsymbol{x}_i$ is valid. Consequently, $\mathcal{W} \subset \mathcal{X}$ holds. The inclusion of a data point into the prototype set can be represented using a binary index variable; using this representation, a connection to the binary optimization problem becomes apparent.

Optimization of the prototype set $\mathcal{W}$ can be achieved with a restricted expectation maximization scheme (EM) of alternating optimization steps. During the expectation step, the data are assigned to the current prototypes, whereas in the maximization step, the prototypes are re-adjusted with the median determination of the current assignments. The corresponding counterparts of neural gas and k-means are median neural gas and k-medoids, respectively [26,34].

2.1.3. Unsupervised Vector Quantization as a Set-Cover Problem Using ϵ-Balls

Motivated by the notion of receptive fields for VQ, an approach based on set covering was introduced. In this scenario, we search for a set $\mathcal{W}_\epsilon \subset \mathbb{R}^n$ to represent the data $\mathcal{X}$ through prototype-dependent ϵ-balls

$$B_\epsilon(\boldsymbol{w}_j) = \{\boldsymbol{x} \in \mathbb{R}^n | d(\boldsymbol{x}, \boldsymbol{w}_j) < \epsilon\} \tag{2}$$

for prototypes $\boldsymbol{w}_j \in \mathcal{W}_\epsilon$. More precisely, we consider the ϵ-restricted receptive fields of prototypes

$$R_\epsilon(\boldsymbol{w}_j) = \{\boldsymbol{x}_i \in \mathcal{X} | s_\epsilon(\boldsymbol{x}_i) = j\}$$

for a given configuration $\mathcal{W}_\epsilon$, where

$$s_\epsilon(\boldsymbol{x}) = \begin{cases} j & \text{if } s(\boldsymbol{x}) = j \text{ and } d(\boldsymbol{x}, \boldsymbol{w}_j) < \epsilon \\ \varnothing & \text{else} \end{cases}$$

is the ϵ-restricted winner determination, and '$\varnothing$' denotes the no-assignment-statement. Hence, $R_\epsilon(\boldsymbol{w}_j)$ consists of all data $\boldsymbol{x}_i \in \mathcal{X}$ covered by an ϵ-ball such that we have $R_\epsilon(\boldsymbol{w}_j) \subseteq B_\epsilon(\boldsymbol{w}_j)$.

The task is to find a minimal prototype set $\mathcal{W}_\epsilon$ such that the respective cardinality M_ϵ is minimum while the unification $B_\epsilon(\mathcal{W}_\epsilon) = \cup_{j=1}^{M_\epsilon} B_\epsilon(\boldsymbol{w}_j \in \mathcal{W}_\epsilon)$ is covering the data $\mathcal{X}$, i.e., $\mathcal{X} \subseteq B_\epsilon(\mathcal{W}_\epsilon)$ has to be valid. A respective VQ approach based on vector shifts is proposed [35].

The set-covering problem becomes much more difficult if we restrict the prototypes $\boldsymbol{w}_j \in \mathcal{W}_\epsilon$ to be data samples $\boldsymbol{x}_i \in \mathcal{X}$, i.e., $\mathcal{W}_\epsilon \subset \mathcal{X}$. This problem is known to be *NP*-complete [36]. A respective greedy algorithm was proposed [37]. It is based on a kernel approach, taking the kernel

$$\kappa_\epsilon(\boldsymbol{x}_j, \boldsymbol{x}_i) = \begin{cases} 1 & \text{if } d_E(\boldsymbol{x}_j, \boldsymbol{x}_i) < \epsilon \\ 0 & \text{else} \end{cases}$$

as an indicator function. The kernel κ_ϵ corresponds to a mapping

$$\boldsymbol{\phi}_\epsilon(\boldsymbol{x}_i) = (\kappa_\epsilon(\boldsymbol{x}_1, \boldsymbol{x}_i), \ldots, \kappa_\epsilon(\boldsymbol{x}_N, \boldsymbol{x}_i))^T \in \mathbb{R}^N$$

known as kernel feature mapping [38]. Introducing a weight vector $\mathbf{w} \in \mathbb{R}^N$, the objective

$$E_{q,\epsilon}(\mathcal{X}) = \min_{\substack{\mathbf{w} \in \mathbb{R}^N \\ \text{subject to } \langle \mathbf{w}, \boldsymbol{\phi}_\epsilon(\boldsymbol{x}_i) \rangle_E \geq 1 \, \forall i}} \|\mathbf{w}\|_q$$

appears as the solution of a minimum problem depending on the parameter q in the Minkowski-norm $\|\mathbf{w}\|_q$. For the choice $q = 0$, we would obtain the original problem. However, for $q = 1$, good approximations are achieved and can be done efficiently using linear programming [37]. After optimization, the data samples $\boldsymbol{x}_i$ with $w_i \approx 1$ serve as prototypes. The respective approach can be optimized online based on neural computing [39,40].

2.1.4. Vector Quantization by Means of Associative Memory Networks

Associative memory networks have been studied for a long time [9,41]. Among them, Hopfield networks (HNs) [41,42] have gained a lot of attraction [30,43,44]. In particular, the strong connection to physics is appreciated [45]; it is related to other optimization problems as given in Section 3.2.3.

Basically, for $\mathcal{X} \subset \mathbb{R}^n$ with cardinality N, HNs are recurrent networks of n bipolar neurons $s_i \in \{-1, 1\}$ connected to each other by the weights $W_{ij} \in \mathbb{R}$. All neurons are collected in the neuron vector $\mathbf{s} = (s_1, \ldots, s_n)^T \in \{-1, 1\}^n$. The weights are collected in the matrix $\mathbf{W} \in \mathbb{R}^{m \times m}$ such that to each neuron s_i belongs a weight vector $\mathbf{w}_i$. The matrix $\mathbf{W}$ is assumed to be symmetric and hollow, i.e., $W_{ii} = 0$. The dynamic of the network is

$$s_i = \text{sgn}(\langle \mathbf{s}, \mathbf{w}_i \rangle_E - \theta_i) \tag{3}$$

where

$$\text{sgn}(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{else} \end{cases}$$

is the standard signum function of $z \in \mathbb{R}$ and θ_i is the neuron-related bias generating the vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)^T$. According to the dynamic (3), the neurons in an HN are assumed to be perceptrons with the signum function as activation [46,47]. Frequently, the vectorized notation

$$\mathbf{s}' = \operatorname{sgn}[\mathbf{W}\mathbf{s}] - \boldsymbol{\theta} \tag{4}$$

of the dynamic (3) is more convenient, emphasizing the asynchronous dynamic. The network minimizes the energy function

$$E_H(\mathbf{s}) = -\frac{1}{2}\mathbf{s}^T\mathbf{W}\mathbf{s} + \langle \mathbf{s}, \boldsymbol{\theta} \rangle_E \tag{5}$$

in a finite number of steps, with an asynchronous update dynamic [45].

For given bipolar data vectors $\boldsymbol{x}_i \in \mathcal{X}$ with dataset cardinality $N \ll n$, the matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ is obtained with the entries

$$W_{ij} = \frac{1}{N}\sum_{k=1}^{N} [\boldsymbol{x}_k]_i \cdot [\boldsymbol{x}_k]_j = \frac{1}{N}\sum_{k=1}^{N} \boldsymbol{x}_k \cdot \boldsymbol{x}_k^T - \mathbf{I} \tag{6}$$

where $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix. This setting can be interpreted as Hebbian learning [45]. Minimum solutions $\mathbf{s}^* \in \{-1, 1\}^n$ of the dynamic (7) are the data samples $\boldsymbol{x}_i$. Thus, starting with arbitrary vectors $\mathbf{s}$, the network always relaxes to a stored pattern $\boldsymbol{x}_i$ realizing an association scheme if we interpret the starting point as a noisy pattern. The maximum storage capacity of an HN is limited to $c_s = \frac{N}{n}$ patterns with $c_s \leq c_{\max} \sim 0.138$. Dense Hopfield networks (DHNs) are generalizations of HNs with general data patterns $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^n$ having a much greater storage capacity of $c_{\max} = 1$ [48].

For the unsupervised VQ, an HN can be utilized using a kernel approach [49]: Let

$$p(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} \kappa_\phi(\boldsymbol{x}, \boldsymbol{x}_i)$$

be an estimate of the underlying data density $\mathbb{R}^n$ based on the samples $\mathcal{X} \subset \mathbb{R}^n$ with $|\mathcal{X}| = N$. Analogously,

$$\hat{q}(\boldsymbol{x}) = \frac{1}{M}\sum_{j=1}^{M} \kappa_\phi(\boldsymbol{x}, \boldsymbol{w}_j) \approx \frac{1}{N}\sum_{i=1}^{N} \kappa_\phi(\boldsymbol{x}, \boldsymbol{x}_i) \cdot a_i$$

is an estimate of the data density $\mathbb{R}^n$ based on the M prototypes $\mathcal{W} \subset \mathbb{R}^n$. The density $\hat{q}(\boldsymbol{x})$ can be approximated with

$$q(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} \kappa_\phi(\boldsymbol{x}, \boldsymbol{x}_i) \cdot a_i$$

for assignment variables $a_i \in \{0, 1\}$ collected in the vector $\mathbf{a} = (a_1, \ldots, a_N)^T$ with the constraint $\sum_{i=1}^{N} a_i = M$. According to the theory of kernels, the kernel κ_ϕ relates to a map $\phi : \mathbb{R}^n \to \mathbb{H}$, where $\mathbb{H}$ is a reproducing kernel Hilbert space (RKHS) endowed with an inner product $\langle \cdot | \cdot \rangle_{\mathbb{H}}$ such that

$$\langle \phi(\boldsymbol{x}) | \phi(\boldsymbol{x}') \rangle_{\mathbb{H}} = \kappa_\phi(\boldsymbol{x}, \boldsymbol{x}')$$

holds [38].

For a good representation of $\mathcal{X}$ with the prototype $\mathcal{W}$, it is possible to minimize the quantity

$$\hat{D}(\mathcal{X}, \mathcal{W}) = \| E_{\mathcal{X}}[\phi] - E_{\mathcal{W}}[\phi] \|_{\mathbb{H}},$$

where $E_{\mathcal{X}}[\phi]$ and $E_{\mathcal{W}}[\phi]$ are the expectations of ϕ based on the sets $\mathcal{X}$ and $\mathcal{W}$, respectively, using the densities $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ [49]. We obtain

$$\hat{D}(\mathcal{X},\mathcal{W}) = \frac{1}{N^2}\mathbf{1}^T\mathbf{\Phi}\mathbf{1} + \frac{1}{M^2}\mathbf{a}^T\mathbf{\Phi}\mathbf{a} - \frac{2}{N\cdot M}\mathbf{1}^T\mathbf{\Phi}\mathbf{a}$$

with $\mathbf{1} = (1,\ldots,1)^T \in \mathbb{R}^N$, $\mathbf{\Phi} \in \mathbb{R}^{N\times N}$ and $\Phi_{ij} = \kappa_\phi(\boldsymbol{x}_i,\boldsymbol{x}_j)$. Because the first term $\mathbf{1}^T\mathbf{\Phi}\mathbf{1}$ does not depend on the assignment, minimization of $D(\mathcal{X},\mathcal{W})$ with respect to the assignment vector $\mathbf{a}$ is equivalent to a minimization of

$$\mathrm{D}(\mathcal{X},\mathcal{W}) = \frac{1}{M^2}\mathbf{a}^T\mathbf{\Phi}\mathbf{a} - \frac{2}{N\cdot M}\mathbf{1}^T\mathbf{\Phi}\mathbf{a}$$

subject to the constraint $\langle \mathbf{1}^T,\mathbf{a}\rangle_E = M$ or, equivalently, $\left(\mathbf{1}^T\cdot\mathbf{a} - M\right)^2 = 0$ such that it constitutes a Lagrangian optimization with the multiplier λ_L. Transforming the binary vector $\mathbf{a}$ using $\mathbf{s} = 2\cdot\mathbf{a} - \mathbf{1}$ into a bipolar vector, the constraint minimization problem is reformulated as

$$\mathbf{s}^* = \mathrm{argmin}_{\mathbf{s}\in\{-1,1\}^N}\left(\mathbf{s}^T\mathbf{Q}\mathbf{s} + \langle \mathbf{s},\mathbf{q}\rangle_E\right) \tag{7}$$

with

$$\mathbf{Q} = \frac{1}{4}\left(\frac{1}{M^2}\mathbf{\Phi} - \lambda_L\mathbf{1}\cdot\mathbf{1}^T\right)$$

and

$$\mathbf{q} = \frac{1}{2}\left(\frac{1}{M^2}\mathbf{\Phi} - \lambda_L\mathbf{1}\cdot\mathbf{1}^T\right)\cdot\mathbf{1} - \left(\frac{2}{M\cdot N}\mathbf{\Phi}^T\cdot\mathbf{1} + 2\cdot\lambda_L\cdot M\cdot\mathbf{1}\right),$$

both depending on the Lagrangian multiplier λ_L. Thus, the problem (7) can be translated into the HN energy $E(\mathbf{s})$ with $m = M$, $\boldsymbol{\theta} = \mathbf{q}$,

$$\mathbf{W} = -2\cdot\mathbf{Q} - \frac{\lambda_L}{2\cdot M^2}\cdot\mathbf{I},$$

where $\mathbf{I} \in \mathbb{R}^{N\times N}$ is the unity matrix and $\mathbf{s}^*$ obtained using the HN dynamic (5).

Complex-valued Hopfield networks (CHN) are extending the HN concept to complex numbers [50]. For this purpose, the symmetry assumption for the weights W_{ij} is transferred to the Hermitian symmetry $W_{ij} = \bar{W}_{ij}$ of the conjugates. As in the real case, the complex dynamic is structurally given as in (3) but replacing the real inner product using the complex-valued Euclidean inner product and, as the consequence of that, replacing the signum function $\mathrm{sgn}(z)$, too. Instead of this, the modified 'signum' function

$$\mathrm{csgn}(z) = \begin{cases} e^{0\cdot i} = 1 & \text{if} \quad 0 \le \arg(z) < \varpi_R \\ e^{1\cdot i\cdot\varpi_R} & \text{if} \quad \varpi_R \le \arg(z) < 2\varpi_R \\ \vdots & \vdots \\ e^{(R-1)\cdot i\varpi_R} & (R-1)\cdot\varpi_R \le \arg(z) \le R\cdot\varpi_R \end{cases}$$

for complex-valued z is used, with R being the resolution factor for the phase range delimitation [51]. Thus, $\arg(z)$ is the phase angle of z and $\varpi_R = \frac{2\pi}{R}$ determines the partition of the phase space. The Hebbian learning rule (6) changes to

$$W_{ij} = \frac{1}{N}\sum_{k=1}^{N}[\boldsymbol{x}_k]_i\cdot\overline{[\boldsymbol{x}_k]}_j$$

and the energy of the CHN is obtained as

$$E_H(\mathbf{s}) = -\frac{1}{2}\mathbf{s}^T\mathbf{W}\mathbf{s}$$

for zero bias, which delivers

$$\mathbf{s}' = \mathrm{csgn}[\mathbf{W}\mathbf{s}]$$

as the corresponding dynamic in complete analogy to (4). Note, for the resolution $R = 2$, the usual HN is obtained.

2.2. Supervised Vector Quantization for Classification Learning

For classification learning VQ, we assume that the training data $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^n$ are endowed with a class label $\boldsymbol{y}_i = c(\boldsymbol{x}_i) \in \mathcal{C} = \{1, \ldots, C\}$. Besides the widespread deep networks, which are powerful methods in classification learning but do not belong to VQ algorithms, support vector machines (SVMs) are promising robust classifiers optimizing the separation margin [52]. However, the support vectors, which determine the class borders of the problem, sometimes are interpreted as prototypes such that SVM could be taken as a supervised prototype classifier, too [53]. However, we do not focus on SVM here.

2.2.1. Updates Using Vector Shifts

Prototype-based classification learning based on vector shifts is dominated by the family of learning vector quantizers (LVQ), which was heuristically motivated and already introduced in 1988 [54]. These models assume that for each prototype $\boldsymbol{w}_j \in \mathcal{W}$, we have an additional class label $c(\boldsymbol{w}_j) \in \mathcal{C}$, such that at least one prototype is dedicated to each class. For a given training data pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, let $\boldsymbol{w}^+$ denote the best matching prototype $\boldsymbol{w}_s$ determined with the WTA-rule (1) with additional constraint that $\boldsymbol{y}_i = c(\boldsymbol{w}_s)$ and $d^+(\boldsymbol{x}_i) = d(\boldsymbol{x}_i, \boldsymbol{w}^+)$ denotes the respective dissimilarity. Analogously, $\boldsymbol{w}^-$ is the best matching prototype $\boldsymbol{w}_{s'}$ with the additional constraint that $\boldsymbol{y}_i \neq c(\boldsymbol{w}_{s'})$ and $d^-(\boldsymbol{x}_i) = d(\boldsymbol{x}_i, \boldsymbol{w}^-)$. The basic principle in all LVQ models is that if $d = d_E$ is the squared Euclidean distance, the prototype $\boldsymbol{w}^+$ is attracted by the presented training data sample $\boldsymbol{x}_i$ whereas $\boldsymbol{w}^-$ is repelled. Particularly, we have

$$\Delta\boldsymbol{w}^+ \propto -2 \cdot (\boldsymbol{x}_i - \boldsymbol{w}^+) \quad \text{and} \quad \Delta\boldsymbol{w}^- \propto -2 \cdot (\boldsymbol{w}^- - \boldsymbol{x}_i),$$

which is known as the *attraction-repulsing-scheme* (ARS) of LVQ.

The heuristic LVQ approach can be replaced by an approach grounded on a cost function [55], which is based on the minimization of the approximated classification error

$$E_{GLVQ}(\mathcal{X}, \mathcal{W}) = \sum_{i=1}^{N} E_{GLVQ}(\boldsymbol{x}_i, \mathcal{W}) \tag{8}$$

with local errors

$$E_{GLVQ}(\boldsymbol{x}_i, \mathcal{W}) = f_\theta(\mu(\boldsymbol{x}_i))$$

evaluating the possible classification mismatch for a given data sample $\boldsymbol{x}_i$. Thereby,

$$\mu(\boldsymbol{x}_i) = \frac{d^+(\boldsymbol{x}_i) - d^-(\boldsymbol{x}_i)}{d^+(\boldsymbol{x}_i) + d^-(\boldsymbol{x}_i)} \in [-1, +1]$$

is the so-called classifier function resulting in non-positive values when the sample $\boldsymbol{x}_i$ would be incorrectly classified. The function

$$f_\theta(z) = \frac{1}{1 + \exp(-z \cdot \theta)}$$

is the sigmoid, approximating the Heaviside function

$$H(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{else} \end{cases}$$

but keeping the differentiability. Following this definition, the updates for $\boldsymbol{w}^+$ and $\boldsymbol{w}^-$ in (8) are obtained as

$$\Delta \boldsymbol{w}^{\pm} \propto -2 \cdot \frac{f'_\theta(\mu(\boldsymbol{x}_i)) \cdot d^{\mp}(\boldsymbol{x}_i)}{\left(d^+(\boldsymbol{x}_i) + d^-(\boldsymbol{x}_i)\right)^2} \cdot \left(\boldsymbol{x}_i - \boldsymbol{w}^{\pm}\right),$$

realizing an ARS [55].

This variant of LVQ is known as Generalized LVQ and is proven to be robust against adversarials [14]. For variants including metric learning, we refer to [12]. Complex-valued GLVQ using the Wirtinger calculus for gradient calculations are considered [56].

Learning on topological structures like manifolds and subspaces follows the same framework, considering attraction and repulsing more general in the respective vector spaces [57,58]. An interesting variant, where the prototypes are spherically adapted according to an ARS to keep them on a hypersphere, was proposed—denoted as Angle-LVQ [59].

2.2.2. Median Adaptation

Median LVQ-like adaptation of prototypes for classification learning is possible [27]. This variant is based on an alternating optimization scheme similar to that of medoid k-means and median neural gas but adapted to the classification-restricted setting.

2.2.3. Supervised Vector Quantization as a Set-Cover Problem Using ϵ-Balls

Another classification scheme can be based on prototype selection out of the training samples and ϵ-balls [60]. In analogy to ϵ-balls for prototypes defined in (2), Data-dependent counterparts are defined as

$$B_\epsilon(\boldsymbol{x}_i) = \left\{\boldsymbol{x}_j | d(\boldsymbol{x}_i, \boldsymbol{x}_j) < \epsilon\right\}$$

the union of which trivially covers $\mathcal{X}$. The classification problem is then decomposed into separate cover problems per class, as discussed in Section 2.1.3. For this purpose, each ϵ-ball gets a local cost based on the number of covered points, punishing false classified points using a penalty

$$\frac{1}{|\mathcal{X}|} + |B_\epsilon(\boldsymbol{x}_i) \cap (\mathcal{X} \setminus \mathcal{X}_c)|$$

where $\mathcal{X}_c$ is the set of all data points with the same class as $\boldsymbol{x}_i$. Combined with a unit cost for not covering a point, a prize-collecting set-cover problem is defined that can be transformed into a general set-cover problem. Hence, as an objective, the number of covered and correctly classified data points has to be maximized while keeping the overall number of prototypes low. We refer to [60,61] for detailed mathematical analysis. In particular, a respective approach is presented [61], being similar to the optimization scheme from support vector machines [52].

2.2.4. Supervised Vector Quantization by Means of Associative Memory Networks

Classification by means of associative memory networks is considered classification using Hopfield-like networks [30]. An approach based on spiking neurons instead of perceptron-like neurons in HNs as depicted in (3) was presented using a classical spike-timing-dependent-plasticity (STDP) rule for learning to adapt HNs for classification learning [62].

In contrast, a modified HN for classification can be used [63]. We suppose a dataset $\mathcal{X} \subset \mathbb{R}^n$ consisting of N samples distributed to C classes. A template vector $\boldsymbol{\xi}^c \in \mathbb{R}^N$ is introduced for each class $c \in \mathcal{C}$ with $\xi_i^c = 1$ if $c = \boldsymbol{y}_i$ and $\xi_i^c = -1$, otherwise. The states of neurons s_k are extended to be $s_k \in \{-1, 1, 0\}$ for $k = 1, \ldots, N$ constituting the vector $\mathbf{s}$. We

consider a diluted version of the Hopfield model, where the weight matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$ is considered to be

$$W_{ij} = \begin{cases} -\frac{C}{N} & \text{if } \boldsymbol{y}_i = \boldsymbol{y}_j \\ \frac{C}{2\cdot N}\sum_{c=1}^{C} \xi_i^c \cdot \xi_j^c + 2 - C & \text{else} \end{cases}$$

realizing a slightly modified Hebb-rule compared to (6). The dynamic is still (3) as in the usual Hopfield model. However, if a switch from $s_k = 1$ to $s_k = -1$ is observed as the result of the dynamic, $s_k = 0$ is set to switch of the respective neuron [63].

3. Quantum Computing—General Remarks

In the following, we use the terms quantum and classical computer to describe whether a machine exploits the rules of quantum mechanics to do its calculations or not.

3.1. Levels of Quantum Computing

Quantum Algorithms can be classified into at least three levels: quantum-inspired, quantum-hybrid, and quantum(-native), with increasing dependence on the capabilities of quantum computers.

Working with the mathematical foundation of quantum computing may reveal new insides into classical computing. In this view, classical algorithms appear in a new form, which is not dependent on the execution on real quantum computers but incorporates the mathematical framework of quantum systems to obtain specific variants of the original algorithm. This category of algorithms is called *quantum-inspired algorithms*. For example, in supervised VQ, an approach inspired by quantum mechanics has been developed, based on standard GLVQ, but now adapted to problems where both the data and the prototypes are restricted to the unit sphere [23]. Thus, this algorithm shows similarities to the already mentioned classical Angle LVQ. However, in contrast to this, here, the sphere is interpreted as a Bloch sphere, and the prototype adaptation follows unitary transformations.

While quantum-inspired algorithms only lend the mathematical background of quantum computing, *quantum-hybrid algorithms* use a quantum device as a coprocessor to accelerate the computations. The quantum chip is also referred to as Quantum Processing Unit (QPU) [64]. The QPU is used to solve expensive computational tasks like searching or high-dimensional distance calculations, whereas all other program logic, like data loading or branching, is done using a classical machine.

The quantum-hybrid algorithm can also be defined in more rigorous terms. That is, a quantum-hybrid algorithm requires, for example, "non-trivial amounts of both quantum and classical computational resources" [64]. Following this definition, classical control elements, like repetition until a valid state is found, are not considered hybrid systems.

Finally, as *quantum-native algorithms*, we would like to denote those algorithms that run entirely on a quantum machine after the data is loaded into it. Because of the limitations of the present hardware generation, their physical implementation is not feasible so far, and therefore, ongoing research is often focused on quantum-hybrid strategies under the prevailing circumstances.

3.2. Paradigms of Quantum Computing

Quantum Physics can be harnessed for computing using different kinds of computing paradigms. Currently, there are two major paradigms intensively investigated and discussed for applications: *Gate-based* and *adiabatic* quantum computing. It can be shown that both paradigms are computationally equivalent [65]. Nevertheless, it is interesting to consider these two approaches separately, as they lead to different problems and solutions that are better suited for their underlying hardware. There are several other paradigms, such as measurement-based and topological quantum computing. We will not focus on them in this paper but concentrate on gate-based and adiabatic methods as the most important.

3.2.1. Gate Based Quantum Computing and Data Encoding

Classical computers store information as bits that are either 0 or 1. The smallest unit of a quantum computer is called a *qubit* [66]. It can represent the classical states as $|0\rangle$ and $|1\rangle$. Besides these basis states, every linear combination of the form

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle \qquad \text{with} \qquad a, b \in \mathbb{C}: \quad |a|^2 + |b|^2 = 1.$$

is a valid state of a qubit. If $ab \neq 0$, the qubit is in a so-called superposition state. Alternatively, the qubit can also be written as a wave function

$$\psi = \begin{pmatrix} a \\ b \end{pmatrix}$$

with the normalization constraint for a and b remains to be valid.

When measured, the qubit turns into one of the two classical states according to the probabilities $|a|^2$ and $|b|^2$, respectively. In other words, during measurement, the state changes into the observed one; this effect is called the collapse of the wave function. To get the probabilistic information about a and b, it is, in general, necessary to measure a state multiple times. Because of the collapsing wave function and the so-called no-cloning theorem, this can only be achieved by preparing a qubit multiple times in the same known manner [67].

A collection of qubits is called a quantum register. To represent the state of a quantum register, we write $|i\rangle$ if the quantum register is the binary representation of the non-negative integer i. The wave function for a register containing N qubits is represented by a normalized complex vector of length 2^N:

$$\psi = \sum_{i=0}^{2^N-1} \psi_i\,|i\rangle =: |\psi\rangle \qquad \text{with} \qquad \sum_{i=0}^{2^N-1} |\psi_i|^2 = 1$$

with the complex amplitudes $\psi_i \in \mathbb{C}$. For independent qubits, the state of the register is the tensor product of its qubits, and otherwise, we say that the qubits are entangled. For a deeper introduction to the mathematics of qubits and quantum processes, we recommend [66,68] to the reader.

Basis Encoding

In classical computing, information is represented by a string of bits. Obviously, it is possible to use coding schemes such as floating-point numbers to represent more complex data structures, too. These methods can also be used on a quantum computer without the application of superposition or entanglement effects. However, taking these quantum effects into account enables quantum-specific coding methods.

Besides storing a single bit-sequence, a superposition of multiple sequences of the same length can be stored in a single quantum register as

$$\sum_{i=0}^{2^N-1} w_i\,|\boldsymbol{x}_i\rangle\,,$$

where w_i is the weight of the sequence $\boldsymbol{x}_i$. Thus, the measurement probability $p_i = |w_i|^2$ is valid. Algorithms that run on basis encoding often amplify valid solution sequences of a problem by using interference patterns of the complex phases of various w_i.

A state in this basis encoding scheme can be initialized using the Quantum Associative Memory Algorithm [69].

Amplitude Encoding

In the amplitude encoding scheme, for a given complex vector $\boldsymbol{x}$, its entries are encoded inside the amplitudes ψ_i of a quantum register. For this purpose, first, the vector

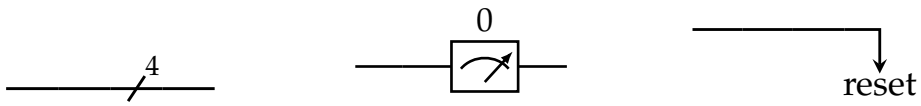
has to be normalized, choosing a normalization that limits the impact on a given task with data distortion. If the vector size is not a power of two, zero padding is applied. We can now, in the second step, initialize a quantum state with $\psi_i = \hat{x}_i$ for the normalized and padded vector $\hat{x}$. A state in this amplitude encoding can be generated using a universal initialization strategy [70].

A highly anticipated, but still not realized, hardware concept is the QRAM [71]. It is key for the speedup of many quantum algorithms, but its viability remains open. Still, its future existence is commonly assumed.

Gate-Based Quantum Paradigm

A common concept for quantum computing is the gate notation, originally introduced by Feynman [72]. In this notation, the time evolution of a qubit is represented by a horizontal line. Evolution is realized by quantum gates that are defined by a unitary matrix applied to a number of qubits. Unitary matrices are vector norm preserving and, therefore, they also preserve the property of being a wave function [68]. Combined with measurement parts, we get a quantum circuit description. A quantum circuit can be seen as the quantum counterpart to a logical circuit.

We will utilize the bundle notation given in Figure 1a to combine multiple qubits into quantum registers. In some quantum routines, the concept of branching is used, where the computation is only continued if measuring a qubit achieves a certain result. In Figure 1b, the output of the circuit is only considered if the qubit is measured as zero. Finally, we use the arrow notation in Figure 1c to represent garbage states. They do not contain usable information anymore, but are still entangled qubits related to the system. We use the term reset over garbage, or simply garbage problem, to emphasize the necessity of appropriately handling this situation. Generally, since garbage states are usually entangled, they cannot be reused, and hence, one resets them using un-computation, i.e., setting them to zero. Of course, the details of the garbage problem are dependent on the circuit in use.



(**a**) (**b**) (**c**)

Figure 1. Example for notations. (**a**) Quantum Register. (**b**) Branching Measurement. (**c**) Garbage State.

3.2.2. Adiabatic Quantum Computing and Problem Hamiltonians

Adiabatic Quantum Computing (AQC) is a computing thought emerging from the adiabatic theorem [73]. It is based on Hamiltonians, which describe the time evolution of the system inside the Schrödinger Equation [74]. A Hamiltonian is realized as a Hermitian matrix $\mathbf{H}$. For adiabatic computing, the corresponding eigenequation is considered. Due to the Hermitian property, all eigenvalues are real, and hence, they can be ordered. They are known as energy levels, with the smallest one being called the ground state.

In this view, if a problem solution can be transformed into the ground state of a known problem Hamiltonian $\mathbf{H}_P$, the adiabatic concept defines a quantum routine that finds this ground state [75]. It starts from an initial Hamiltonian $\mathbf{H}_B$, with a known and simple ground state preparation. On this initial state, usually the equal superposition of all possible outcomes, a time-dependent Hamiltonian

$$\mathbf{H}(t) = \frac{1-t}{T}\mathbf{H}_B + \frac{t}{T}\mathbf{H}_P,$$

that slowly shifts from $\mathbf{H}_B$ to $\mathbf{H}_P$, is applied over a time period T. The adiabatic theorem ensures that if the period T is sufficiently large, the system tends to stay in the ground state of the gradually changing Hamiltonian. After application, the system is in the ground state of $\mathbf{H}_P$ with a very high probability. For a given problem, the final ground state is the single solution or a superposition of all valid solutions. One solution is then revealed by

measuring the qubits. If AQC is run on hardware, manufacturers use the term *quantum annealing* instead to underline the noisy execution environment. The capabilities of a quantum annealer are restricted to optimization problems by their design; it is not possible to use the current generation for general quantum computing that is equivalent to the gate-based paradigm.

The dynamic AQC can be approximated using discrete steps on a gate-based quantum computer [76].

3.2.3. QUBO, Ising Model, and Hopfield Network

Depending on the theoretical background an author is coming from, three main kinds of optimization problems are often encountered in the literature that share similar structures and can be transformed into each other. First, the Quadratic Unconstrained Binary Optimization problem (QUBO) is the optimization of a binary vector $\boldsymbol{x} \in \{0,1\}^n$ for a cost function

$$y = \boldsymbol{x}^T \mathbf{A} \boldsymbol{x} = \sum_{i \leq j} A_{ij} x_i x_j$$

with a real valued upper triangle matrix $\mathbf{A}$. Second, the Ising model is motivated by statistical physics and based on spin variables, which can be in state -1 and 1 [67]. The objective of the Ising model is finding a spin vector $\boldsymbol{x} \in \{-1,1\}^n$, which optimizes

$$y = \sum_i h_i x_i + \sum_{i<j} J_{ij} x_i x_j$$

with pairwise interactions J_{ij} and an external field h_i. A Quantum Annealer is a physical implementation of the Ising Model with limited pairwise interactions. Binary variables b can be transformed into spin variables s and vice versa by the relation

$$b = \frac{1+s}{2},$$

making the Ising model and QUBO mathematically equivalent. Third, the Hopfield energy function (5) was introduced as an associative memory scheme based on Hebbian learning [42,45]. Its discrete form is equivalent to the Ising model if the neurons in this associative memory model are interpreted as bipolar. All models are NP-hard and can, therefore, in theory, be transformed into all NP problems. For a broad list of these transformations, we recommend [77].

3.3. State-of-the-Art of Practical Quantum Experiments

In the last few years, the size of commercial gate-based general-purpose quantum computers did grow from 27 (2019 IBM Falcon) to 433 qubits (2022 IBM Osprey). Thus, the hardware has grown from simple physical demonstrators to machines called Noisy Intermediate-Scale Quantum Computer (NISQ) [78]. However, this hardware generation is still severely restricted by its size and a high error rate.

The latter problem could be solved using quantum error correction or quantum error mitigation schemes. Quantum error mitigation is a maturing field of research, with frameworks like Mitiq [79] being published. Common to most of these mitigation techniques is that a higher number of physical qubits is required to obtain a single logical qubit with a lower noise level, making the size problem the major one.

Different physical realizations of quantum computer hardware exist; we can only give some examples. Realizations based on superconducting qubits for gate-based (IBM Q System One) and for adiabatic (D-Wave's Advantage QPU) are available. Further, quantum devices that are based on photons (Xanadu's Borealis) or trapped ions (Honeywell System Model H1) exist.

For small toy application problems, it is possible to simulate the behavior of a quantum computer by means of a classical computing machine. Particularly, single steps of the

gate-based concept can be simulated using respective linear algebra packages. Otherwise, circuits can be built in quantum computing frameworks, like IBM's Qiskit [80] or Xanadu's Pennylane [81]. It is also possible to simulate AQC behavior for evolving quantum systems [82]. Quantum machines that are available through online access allow observing the influence of noise on quantum algorithms based on tiny examples.

4. Quantum Approaches for Vector Quantization

The field of quantum algorithms for VQ is currently a collection of quantum routines that can solve particular sub-tasks than complete algorithms available for practical applications. Combinations of those routines with machine learning approaches beside traditional VQ-learning have been proposed for different fields, for example, in connection to support vector machines [83] or generative adversarial networks [84].

In this section, we present two strategies to combine classical prototype-based vector quantization principles for VQ with appropriate quantum algorithms. Thereby, we roughly follow the structure for unsupervised/supervised vector quantization learning, as explained in the Sections 2.1 and 2.2.

By doing so, we can replace, on the one hand, single routines in the (L)VQ learning schemes using quantum counterparts. On the other, if we can find a VQ formalism that is based on a combinatorial problem, preferably a QUBO, several quantum solvers have already been proposed and, hence, could be used to tackle the problem.

4.1. Dissimilarities

As previously mentioned at the beginning of Section 2, the choice of the dissimilarity measure in vector quantization is crucial and influences the outcome of the learning. This statement remains true also for quantum vector quantization approaches. However, in the quantum algorithm context, the dissimilarity concepts are closely related to the coding scheme as already discussed in Section 3.2. Here it should be explicitly mentioned that the coding can be interpreted as *quantum feature mapping* of the data into a Hilbert space, which is the Bloch-sphere [4,23]. Hence, the dissimilarity calculation represents distance calculations in the Bloch sphere. However, due to this quantum feature mapping, the interpretation of the vector quantization algorithm with respect to the original data space may be limited, whereas, within the Bloch sphere (Hilbert space), the prototype principle and interpretation paradigms remain true. Thereby, the mapping here is analogous to the kernel feature mapping in support vector machines [38] as pointed out frequently [85–87].

Two quantum routines are promising for dissimilarity calculation: the SWAP test [88] and the Hadamard test, used in quantum classification tasks [89,90]. Both routines generate a measurement that is related to the inner product of two normalized vectors in the Bloch sphere. These input vectors are encoded using amplitude encoding. The strategies differ in their requirements for state preparation.

The *SWAP test* circuit is shown in Figure 2. This circuit is sampled multiple times. From these samples, the probability distribution of the ancilla bit is approximated, which is connected to the Euclidean inner product by

$$p_a(|0\rangle) = \frac{1}{2}\left(1 + |\langle \boldsymbol{x}|\boldsymbol{w}_k\rangle|^2\right).$$

Thus, we can calculate the inner product from the estimated probability and, hence, from that, the Euclidean distance.

Another but similar approach [89,90], which is based on the Hadamard gate, sometimes denoted as a (modified) *Hadamard test*, is shown in Figure 3. For this circuit, the probability of measuring the ancilla in zero state is

$$p_a(|0\rangle) = \frac{1}{2}(1 + \mathrm{Re}\{\langle \boldsymbol{x}|\boldsymbol{w}_k\rangle\}).$$

Due to the superposition principle, it is possible to run these tests in parallel on different inputs. This method was demonstrated to work [91] and has been further adapted and improved [25] in this way that the test is applicable on different vectors by means of appropriately determined index registers. It is not possible to read out all values at the end, but it is proposed as a possible replacement of QRAM in some cases [91]. Whether this parallel application can replace QRAM in the VQ application is an open question.

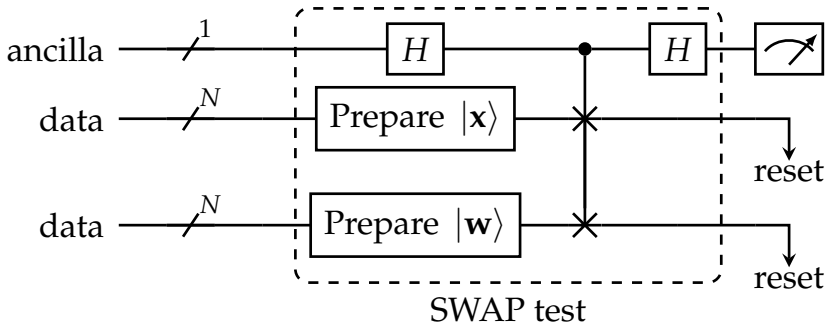


Figure 2. The SWAP-test to measure the dissimilarity between the states $|\mathbf{x}\rangle$ and $|\mathbf{w}\rangle$ using the ancilla qubit.

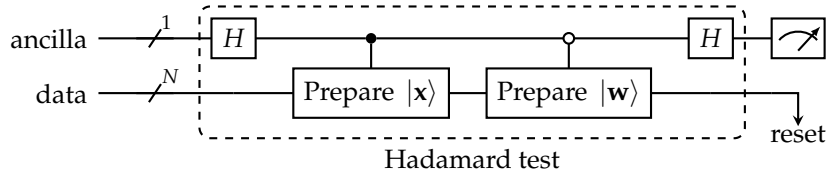


Figure 3. The Hadamard test to measure the dissimilarity between the states $|\mathbf{x}\rangle$ and $|\mathbf{w}\rangle$ using the ancilla qubit.

4.2. Winner Determination

Winner determination in prototype-based unsupervised and supervised vector quantization is one of the key ingredients for vector-shift-based adaptation for learning as well as median variants, which both inherently follow the winner-takes-all (WTA) principle (1). Obviously, the winner determination is not independent of the dissimilarity determination and, in quantum computing, is realized as a minimum search according to the list of all available dissimilarity values for a current system state.

An algorithm to find a minimum is the algorithm provided by Dürr and Høyer [92,93], which is, in fact, an extension of the often referenced Grover search [94]. Another sophisticated variant for minimum search based on a modified swap test, a so-called *quantum phase estimation* and the Grover search has been proposed [95]. Connections to the similar k-nearest neighbor approach were shown [96].

4.3. Updates Using Vector Shift

The normalization of quantum states places them on a hypersphere; this allows the transfer of the spherical linear interpolation (SLERP) to a quantum Computer [25]. This method is known as qSLERP, and the respective circuit is depicted in Figure 4. The qSLERP-circuit takes the two vectors $|\mathbf{x}\rangle$ and $|\mathbf{w}\rangle$ as input as well as the angle θ between them, which can be derived from the inner product and the interpolation position. The ancilla bit is measured, and the result in the data register is only kept if the ancilla is in the zero state. To store the result, the probability of the state of the data register has to be determined using repeated execution of the circuit.

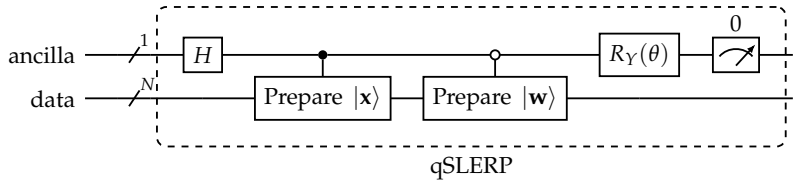


Figure 4. qSLERP circuit.

From a mathematical point of view, the qSLERP approach is similar to the update used in Angle-LVQ [59] for non-quantum systems.

4.4. Median Adaptation

A selection task based on distances in median approaches is the Max–Sum Diversification problem; it can be mathematically transformed into an equivalent Ising model [97]. Other median approaches in VQ depend on the EM algorithm, like median k-means (k-medoids). A quantum counterpart of expectation maximization [98] was introduced as an extension of the q-means [99], a quantum variant of k-means. The authors showed the application of a fitting Gaussian Mixture Model. A possible generalization to other methods based on EM needs to be verified.

4.5. Vector Quantization as Set-Cover Problem

Above, in Section 2.1.3, we introduced the set-cover problem for unsupervised vector quantization. The QUBO model is NP-hard. Hence, at least in theory, the NP-complete set-cover problem can be transformed into it. A transformation from a (paired) set cover to the Ising model and, therefore, to QUBO can be solved with AQC [100]. Taking the view of vector quantization, the following transformation of an unsupervised ϵ-ball set-cover problem to a corresponding QUBO formulation can be done [77]:

Let $\{B_\epsilon(\boldsymbol{x}_i)\}$ with $i \in \{1, \ldots, N\}$ be the set of ϵ-balls surrounding each data point $\boldsymbol{x}_i \in \mathcal{X}$. We introduce binary indicator variables z_i, which are zero if $B_\epsilon(\boldsymbol{x}_i)$ does not belong to the current covering, and it is one elsewhere. Further, let c_k be the number of sets $B_\epsilon(\boldsymbol{x}_i)$ with $z_i = 1$ and $\boldsymbol{x}_k \in B_\epsilon(\boldsymbol{x}_i)$, i.e., c_k counts the number of covering ϵ-balls in the current covering. In the next step, we code the integer variables c_k using binary coding according to let $c_{k,m} = 1$ iff $c_k = m$ and zero otherwise. We impose the following constraint

$$\sum_{m=1}^{N} c_{k,m} = 1 : \forall k \,,$$

reflecting that the binary counting variables are consistent, and exactly one is selected. The second constraint establishes logical connections between the selected sets in the considered current covering and the counting variables by requiring that

$$\sum_{i|\boldsymbol{x}_k \in B_\epsilon(\boldsymbol{x}_i)} z_i = \sum_{m=1}^{N} m \cdot c_{k,m} \quad : \forall k \,,$$

where $m \geq 1$ ensures that every point is covered. These constraints can be transformed into penalty terms using the squared differences between the left and the right side for each. Then the clustering task is to minimize the sum of all indicator variables z_i, taking the penalty terms into account. Using the explained construction scheme, this resulting cost function only contains pairwise interactions between binary variables without explicit constraints. Therefore, the set-cover problem is transformed into a QUBO problem.

Analog considerations are valid for the supervised classification task.

4.6. Vector Quantization by Means of Associative Memory

One of the first quantum associative memories based on a Hopfield network (HN) approach was proposed in 2000 [69]. Recently, a physical realization based on an actual

quantum processor was provided [101]. As shown before, the HN energy function is identical to the QUBO problem, which can be solved by applying the quantum strategies in Section 4.7. Further, AQC for VQ was proposed, using HNs as an intermediate model [49].

A connection between gate-based quantum computing and HNs can be shown [102]. There, a solver based on Hebbian learning and mixed quantum states is introduced. The connection to complex-valued HN, as discussed in Section 2.1, is straightforward.

4.7. Solving QUBO with Quantum Devices

While we transformed most problems into QUBO in the previous subsections, we now connect them to quantum computing. Different strategies based on quantum computing hardware are available to solve QUBO problems. Heuristic approaches exist for many commercially available hardware types, from quantum annealers and gate-based computers to quantum devices based on photons.

Solve QUBO with AQC

A commercial approach in quantum annealing to solve QUBO or Ising models is described in the white paper [103] using the Company D-Wave. The solving of QUBO problems is the major optimization problem that is proposed to run on the limited hardware of a quantum annealer. According to this, the binary variables are physically implemented as quantum states. Values of the model interactions are implemented using couplers between pairs of qubits. Restrictions of the hardware make it necessary to order and map the qubits accordingly. The major open question about AQC is whether the length of the period grows slowly enough to be feasible.

Solve QUBO with Gate-Based Computing

For gate-based quantum computers, a heuristic called QAOA can approximately solve QUBO problems [104]. It contains two steps, first, optimizing a variational quantum circuit and second, sampling from this circuit. The ansatz of this circuit is a parametrized alternating application of the problem Hamiltonian and a mixing Hamiltonian. The expected value of the state gets then minimized using a classical computer, and different strategies have been proposed. With the found (local) minima, the quantum circuit gets executed, and the output gets sampled. Heuristically, low-energy states have a high chance of being sampled. It should be emphasized that it remains to be proven that QAOA has a computational advantage for any type of problem.

Solve QUBO with Photonic Devices

Gaussian Boson Sampling is a tool realized using quantum photonic computers, a kind of quantum hardware that has potential physical benefits that could lead to fast adoption. Quantum photonic devices introduce new types of quantum states into the field of quantum computing, like Fock states or photon counts. Gaussian Boson Sampling is seen as a near-term approach to utilizing quantum photonic computers. A solving strategy for QUBO by means of an Ising model taking a hybrid approach using Boson-sampling has been presented [105].

4.8. Further Aspects—Practical Limitations

Impact of Coding

We can replace all steps in the vector shift variant of VQ with quantum routines, but it is not possible to build up a complete algorithm so far. The main difficulty is that these atomic parts do not share the same encoding.

One example of this fact is the SWAP-test: Here, the result is stored as the probability of a qubit being in state $|0\rangle$. However, we have to get rid of the phase information to obtain a consistent result. Otherwise, this could lead to unwanted interference. A possible solution could be the exploration of routines based on mixed quantum states. However, the use of a Grover search is inconvenient for this task because it is based on basis encoded values, while the dissimilarity measures are stored as probabilities.

Impact of Theoretical Approximation Boundaries and Constraints

Some algorithms use probability or state estimation with sampling because it is impossible to directly observe a quantum state. For example, the output of the SWAP test has to be estimated using repeated measurements. The problem with an estimation of a measurement probe is well-known [25,90]. The field of finding the best measurement strategy for state estimation is called quantum tomography.

Another theoretical boundary is the loading of classical data to a real quantum device. Initializing an arbitrary state efficiently would be possible within the framework and regarding the implementation of the QRAM concept. However, the efficiency of those approaches is demanded because of the repeating nature of most algorithms and from the perspective of the non-cloning theorem.

Impact of Noisy Circuit Execution

The noisy nature of the current quantum hardware defeats most, if not all, of the theoretical benefits of quantum algorithms. A combination of improved hardware and quantum error correction will potentially solve this issue, allowing large-scale quantum computers.

5. Conclusions

The abstract motif of vector quantization learning has several adaptation realizations based on distinct underlying mathematical optimization problems. Vector shifts in prototype-based vector quantizers frequently are obtained as gradients of respective cost functions, whereas set-cover problem-related optimization belongs to binary optimization. Associative memory recalls rely on attractor dynamics. For these diverse paradigms, we highlighted (partially) matching quantum routines and algorithms. Most of them are, unfortunately, only heuristics. Further, their advantages over classical approaches have not been proven in general. However, the wide range of quantum paradigms, quantum algorithms, and quantum devices capable of assisting vector quantization translates into a broad potential of vector quantization for quantum machine learning. It is not possible to predict which quantum paradigm will succeed in the long term. Therefore, there is no outstanding vector quantization approach for quantum computing at the moment. But because many of the presented approaches can be transformed into QUBO problems, improved quantum solvers of each paradigm would have a strong impact. Especially, discrete strategies like median vector quantization, which are heavily restricted by classical computers, could become feasible. In other words, if a quantum advantage can be demonstrated in the future, vector quantization will likely benefit, but the direction will be set with improvements in the construction of quantum devices.

Finally, we want to emphasize that the overview in the paper is not exhaustive. For example, a possible connection that was not introduced above is the use of the probabilistic nature of quantum computing in combination with the probabilistic variants of Learning Vector Quantization [106].

However, we also should mention that the question of possible quantum supremacy, or even quantum advantages, is currently still considered an open problem in the literature. It has been discussed to be merely a weak goal for quantum machine learning [107]. Due to the lack of the existence of sufficient hardware today, it is also not possible to compare real runtimes adequately.

Nevertheless, the theoretical understanding of the respective mathematical concepts and their physical realization is important for progress in quantum computing and, hence, also in quantum-related vector quantization.

Author Contributions: Conceptualization A.E.; writing—original draft A.E. and T.V.; project administration T.V.; funding acquisition T.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the project "PAL" funded by the German Federal Ministry of Education and Research (BMBF) and the project "AI4OD" funded by the German Federal Ministry of Economics (BMWi).

Data Availability Statement: Not applicable.

Acknowledgments: T.V. acknowledges fruitful discussions with M. Schuld during NeurIPS'2020 about quantum-inspired machine learning and related kernel interpretations giving the inspiration to deal with this topic in the context of vector quantization.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AQC	Adiabatic Quantum Computing
HN	Hopfield Network
VQ	Vector Quantization
LVQ	Learning Vector Quantization
QUBO	Quadratic Unconstrained Binary Optimization

References

1. Arrazola, J.M.; Bromley, T.R. Using Gaussian Boson Sampling to Find Dense Subgraphs. *Phys. Rev. Lett.* **2018**, *121*, 030503. [CrossRef]
2. Le, P.Q.; Dong, F.; Hirota, K. A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Inf. Process.* **2010**, *10*, 63–84. [CrossRef]
3. Martín-Guerrero, J.D.; Lamata, L. Quantum Machine Learning: A tutorial. *Neurocomputing* **2022**, *470*, 457–461. [CrossRef]
4. Schuld, M.; Petruccione, F. *Machine Learning with Quantum Computers*; Springer: Cham, Switzerland, 2021. [CrossRef]
5. Ostaszewski, M.; Trenkwalder, L.M.; Masarczyk, W.; Scerri, E.; Dunjko, V. Reinforcement learning for optimization of variational quantum circuit architectures. In Proceedings of the Advances in Neural Information Processing Systems 34, NeurIPS 2021, Online, 6–14 December 2021; Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W., Eds.; Curran Associates, Inc.: Nice, France, 2021; Volume 34.
6. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Plenum: New York, NY, USA, 1981.
7. Arthur, D.; Vassilvitskii, S. K-Means++: The Advantages of Careful Seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2007; pp. 1027–1035.
8. Martinetz, T.M.; Berkovich, S.G.; Schulten, K.J. 'Neural-Gas' Network for Vector Quantization and Its Application to Time-Series Prediction. *IEEE Trans. Neural Netw.* **1993**, *4*, 558–569. [CrossRef] [PubMed]
9. Kohonen, T. *Self-Organization and Associative Memory*; Springer: Berlin/Heidelberg, Germany, 1989. [CrossRef]
10. Kohonen, T. *Self-Organizing Maps*; Springer: Berlin/Heidelberg, Germany, 1995. [CrossRef]
11. Villmann, T.; Der, R.; Herrmann, M.; Martinetz, T. Topology Preservation in Self-Organizing Feature Maps: Exact Definition and Measurement. *IEEE Trans. Neural Netw.* **1997**, *8*, 256–266. [CrossRef]
12. Biehl, M.; Hammer, B.; Villmann, T. Prototype-based models in machine learning. *Wiley Interdiscip. Rev. Cogn. Sci.* **2016**, *7*, 92–111. [CrossRef]
13. Crammer, K.; Gilad-Bachrach, R.; Navot, A.; Tishby, N. Margin analysis of the LVQ algorithm. In Proceedings of the Advances in Neural Information Processing (Proc. NIPS 2002), Vancouver, BC, Canada, 9–14 December 2002; Becker, S., Thrun, S., Obermayer, K., Eds.; MIT Press: Cambridge, MA, USA, 2003; Volume 15, pp. 462–469.
14. Saralajew, S.; Holdijk, L.; Villmann, T. Fast Adversarial Robustness Certification of Nearest Prototype Classifiers for Arbitrary Seminorms. In Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Online, 6–12 December 2020; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Nice, France, 2020; Volume 33, pp. 13635–13650.
15. Voráček, V.; Hein, M. Provably adversarially robust nearest prototype classifiers. In Proceedings of the 39th International Conference on Machine Learning (ICML), Baltimore, MD, USA, 17–23 July 2022; Volume 162, pp. 22361–22383. [CrossRef]
16. Hammer, B.; Villmann, T. Generalized Relevance Learning Vector Quantization. *Neural Netw.* **2002**, *15*, 1059–1068. [CrossRef]
17. Schneider, P.; Hammer, B.; Biehl, M. Adaptive Relevance Matrices in Learning Vector Quantization. *Neural Comput.* **2009**, *21*, 3532–3561. [CrossRef]
18. Lisboa, P.; Saralajew, S.; Vellido, A.; Villmann, T. The coming of age of interpretable and explainable machine learning models. In Proceedings of the 29th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2021), Bruges, Belgium, 6–8 October 2021; Verleysen, M., Ed.; i6doc.com: Louvain-La-Neuve, Belgium, 2021; pp. 547–556. [CrossRef]

19. Lisboa, P.; Saralajew, S.; Vellido, A.; Fernández-Domenech, R.; Villmann, T. The Coming of Age of Interpretable and Explainable Machine Learning Models. *Neurocomputing* **2023**, *535*, 25–39. [CrossRef]
20. Li, O.; Liu, H.; Chen, C.; Rudin, C. Deep Learning for Case-Based Reasoning through Prototypes: A Neural Network that Explains Its Predictions. In Proceedings of the Thirty-Second AAAI Conferenceon Artificial Intelligence (AAAI-18), New Orleans, LA, USA, 2–7 February 2018; pp. 3530–3537. [CrossRef]
21. Rudin, C.; Chen, C.; Chen, Z.; Huang, H.; Semenova, L.; Zhong, C. Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges. *Stat. Surv.* **2022**, *16*, 1–85. [CrossRef]
22. Vellido, A. The importance of interpretability and visualization in machine learning for applications in medicine and health care. *Neural Netw. Appl.* **2020**, *32*, 18069–18083. [CrossRef]
23. Villmann, T.; Engelsberger, A.; Ravichandran, J.; Villmann, A.; Kaden, M. Quantum-Inspired Learning Vector Quantizers for Prototype-Based Classification. *Neural Comput. Appl.* **2022**, *34*, 79–88. [CrossRef]
24. Aïmeur, E.; Brassard, G.; Gambs, S. Quantum clustering algorithms. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; ACM: New York, NY, USA, 2007. [CrossRef]
25. Engelsberger, A.; Schubert, R.; Villmann, T. Steps Forward to Quantum Learning Vector Quantization for Classification Learning on a Theoretical Quantum Computer. In Proceedings of the Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization, Prague, Czech Republic, 6–7 July 2022; Faigl, J., Olteanu, M., Drchal, J., Eds.; Springer: Cham, Switzerland, 2022; pp. 63–73. [CrossRef]
26. Cottrell, M.; Hammer, B.; Hasenfuß, A.; Villmann, T. Batch and median neural gas. *Neural Netw.* **2006**, *19*, 762–771. [CrossRef] [PubMed]
27. Nebel, D.; Hammer, B.; Frohberg, K.; Villmann, T. Median variants of learning vector quantization for learning of dissimilarity data. *Neurocomputing* **2015**, *169*, 295–305. [CrossRef]
28. Gorman, C.; Robins, A.; Knott, A. Hopfield networks as a model of prototype-based category learning: A method to distinguish trained, spurious, and prototypical attractors. *Neural Netw.* **2017**, *91*, 76–84. [CrossRef] [PubMed]
29. Pulvermüller, F.; Tomasello, R.; Henningsen-Schomers, M.R.; Wennekers, T. Biological constraints on neural network models of cognitive function. *Nat. Rev. Neurosci.* **2021**, *22*, 488–502. [CrossRef]
30. Ramsauer, H.; Schafl, B.; Lehner, J.; Seidl, P.; Widrich, M.; Gruber, L.; Holzleitner, M.; Pavlovi'c, M.; Sandve, G.K.; Greiff, V.; et al. Hopfield Networks is All You Need. *arXiv* **2021**, arXiv:2008.02217.
31. Graf, S.; Luschgy, H. *Foundations of Quantization for Probability Distributions*; Springer: Berlin/Heidelberg, Germany, 2000. [CrossRef]
32. Linde, Y.; Buzo, A.; Gray, R. An Algorithm for Vector Quantizer Design. *IEEE Trans. Commun.* **1980**, *28*, 84–95. [CrossRef]
33. Kirby, M.; Peterson, C. Visualizing Data Sets on the Grassmannian Using Self-Organizing Maps. In Proceedings of the 12th Workshop on Self-Organizing Maps and Learning Vector Quantization, Nancy, France, 28–30 June 2017; IEEE: New York, NY, USA, 2017; pp. 32–37. [CrossRef]
34. Kaufman, L.; Rousseeuv, P.J. Partitioning Around Medoids (Program PAM). In *Finding Groups in Data: An Introduction to Cluster Analysis*; John Wiley & Sons, Inc.: New York, NY, USA, 1990; pp. 68–125. [CrossRef]
35. Staps, D.; Schubert, R.; Kaden, M.; Lampe, A.; Hermann, W.; Villmann, T. Prototype-Based One-Class-Classification Learning Using Local Representations. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; IEEE: New York, NY, USA, 2022. [CrossRef]
36. Attali, D.; Nguyen, T.B.; Sivignon, I. ϵ-covering is NP-complete. In Proceedings of the European Workshop on Computational Geometry (EuroCG), Lugano, Switzerland, 30 March–1 April 2016.
37. Tipping, M.E.; Schölkopf, B. A Kernel Approach for Vector Quantization with Guaranteed Distortion Bounds. *Proc. Mach. Learn. Res.* **2001**, *R3*, 298–303.
38. Steinwart, I.; Christmann, A. *Support Vector Machines*; Information Science and Statistics; Springer: New York, NY, USA, 2008. [CrossRef]
39. Bauckhage, C. A neural network implementation of Frank-Wolfe optimization. In *Artificial Neural Networks and Machine Learning, Proceedings of the International Conference on Artificial Neural Networks (ICANN), Alghero, Italy, 11–14 September 2017*; Lecture Notes in Computer Science; Lintas, A., Rovetta, S., Verschure, P., Villa, A., Eds.; Springer: Cham, Switzerland, 2017; Volume 10613. [CrossRef]
40. Bauckhage, C.; Sifa, R.; Dong, T. Prototypes Within Minimum Enclosing Balls. In *Artificial Neural Networks and Machine Learning, Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2019: Workshop and Special Sessions, Munich, Germany, 17–19 September 2019*; Lecture Notes in Computer Science; Tetko, I., Kurkova, V., Karpov, P., Theis, F., Eds.; Springer: Cham, Switzerland, 2019; Volume 11731, pp. 365–376. [CrossRef]
41. Amari, S.I. Neural Theory of Association and Concept-Formation. *Biol. Cybern.* **1977**, *26*, 175–185. [CrossRef]
42. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558. [CrossRef] [PubMed]
43. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
44. Haykin, S. *Neural Networks—A Comprehensive Foundation*; Macmillan: New York, NY, USA, 1994.
45. Hertz, J.A.; Krogh, A.; Palmer, R.G. *Introduction to the Theory of Neural Computation*; Avalon Publishing: London, UK, 1991.

46. Minsky, M.L.; Papert, S. *Perceptrons—An Introduction to Computational Geometry*; MIT Press: Cambridge, MA, USA, 1969.
47. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **1958**, *65*, 386–408. [CrossRef] [PubMed]
48. Krotov, D.; Hopfield, J.J. Dense Associative Memory for Pattern Recognition. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; Lee, D.D., von Luxburg, U., Garnett, R., Sugiyama, M., Guyon, I., Eds.; Curran Associates, Inc.: Nice, France, 2016; pp. 1180–1188.
49. Bauckhage, C.; Ramamurthy, R.; Sifa, R. Hopfield Networks for Vector Quantization. In Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2020, Bratislava, Slovakia, 15–18 September 2020; Farkaš, I., Masulli, P., Wermter, S., Eds.; Springer: Cham, Switzerland, 2020; Volume 12397, pp. 192–203. [CrossRef]
50. Jankowski, S.; Lozowski, A.; Zurada, J.M. Complex-valued multistate neural associative memory. *IEEE Trans. Neural Netw.* **1996**, *7*, 1491–1496. [CrossRef] [PubMed]
51. Lee, D.L. Improvements of complex-valued Hopfield associative memory by using generalized projection rules. *IEEE Trans. Neural Netw.* **2006**, *17*, 1341–1347. [CrossRef]
52. Schölkopf, B.; Smola, A.J. *Learning with Kernels—Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press: Cambridge, MA, USA, 2002.
53. Villmann, T.; Bohnsack, A.; Kaden, M. Can Learning Vector Quantization Be an Alternative to SVM and Deep Learning?—Recent Trends and Advanced Variants of Learning Vector Quantization for Classification Learning. *J. Artif. Intell. Soft Comput. Res.* **2016**, *7*, 65–81. [CrossRef]
54. Kohonen, T. Learning Vector Quantization. *Neural Netw.* **1988**, *1*, 303. [CrossRef]
55. Sato, A.; Yamada, K. Generalized Learning Vector Quantization. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 27–30 November 1995; MIT Press: Cambridge, MA, USA, 1995; Volume 8.
56. Gay, M.; Kaden, M.; Biehl, M.; Lampe, A.; Villmann, T. Complex Variants of GLVQ Based on Wirtinger's Calculus. In *Advances in Intelligent Systems and Computing, Proceedings of the Advances in Self-Organizing Maps and Learning Vector Quantization, Houston, TX, USA, 6–8 January 2016*; Merényi, E., Mendenhall, M.J., O'Driscoll, P., Eds.; Springer: Cham, Switzerland, 2016; Volume 428, pp. 293–303. [CrossRef]
57. Tang, F.; Fan, M.; Tino, P. Generalized Learning Riemannian Space Quantization: A Case Study on Riemannian Manifold of SPD Matrices. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 281–292. [CrossRef]
58. Mohammadi, M.; Biehl, M.; Villmann, A.; Villmann, T. Sequence Learning in Unsupervised and Supervised Vector Quantization Using Hankel Matrices. In Proceedings of the Artificial Intelligence and Soft Computing, ICAISC 2017, Zakopane, Poland, 11–15 June 2017; Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M., Eds.; Springer: Cham, Switzerland, 2017; pp. 131–142. [CrossRef]
59. Taghribi, A.; Canducci, M.; Mastropietro, M.; Rijcke, S.D.; Bunte, K.; Tiňo, P. ASAP—A sub-sampling approach for preserving topological structures modeled with geodesic topographic mapping. *Neurocomputing* **2022**, *470*, 376–388. [CrossRef]
60. Bien, J.; Tibshirani, R. Prototype Selection for Interpretable Classification. *Ann. Appl. Stat.* **2011**, *5*, 2403–2424. [CrossRef]
61. Paaßen, B.; Villmann, T. Prototype selection based on set covering and large margins. *Mach. Learn. Rep.* **2021**, *14*, 35–42.
62. Hu, X.; Wang, T. Training the Hopfield Neural Network for Classification Using a STDP-Like Rule. In *Neural Information Processing, Proceedings of the Neural Information Processing, Long Beach, CA, USA, 4–9 December 2017*; Lecture Notes in Computer Science; Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; pp. 737–744. [CrossRef]
63. Cantini, L.; Caselle, M. Hope4Genes: A Hopfield-like class prediction algorithm for transcriptomic data. *Sci. Rep.* **2019**, *9*, 337. [CrossRef] [PubMed]
64. Callison, A.; Chancellor, N. Hybrid Quantum-Classical Algorithms in the Noisy Intermediate-Scale Quantum Era and Beyond. *Phys. Rev. A* **2022**, *106*, 010101. [CrossRef]
65. Aharonov, D.; van Dam, W.; Kempe, J.; Landau, Z.; Lloyd, S.; Regev, O. Adiabatic quantum computation is equivalent to standard quantum computation. In Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, Rome, Italy, 17–19 October 2004; pp. 42–51. [CrossRef]
66. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2012. [CrossRef]
67. Lindner, A.; Strauch, D. *A Complete Course on Theoretical Physics*; Springer: Cham, Switzerland, 2018. [CrossRef]
68. Scherer, W. *Mathematics of Quantum Computing—An Introduction*; Springer: Cham, Switzerland, 2019. [CrossRef]
69. Ventura, D.; Martinez, T. Quantum associative memory. *Inf. Sci.* **2000**, *124*, 273–296. [CrossRef]
70. Möttönen, M.; Vartiainen, J.; Bergholm, V.; Salomaa, M. Transformation of quantum states using uniformly controlled rotations. *Quantum Inf. Comput.* **2005**, *5*, 467–473. [CrossRef]
71. Giovannetti, V.; Lloyd, S.; Maccone, L. Quantum Random Access Memory. *Phys. Rev. Lett.* **2008**, *100*, 160501. [CrossRef] [PubMed]
72. Feynman, R.P. Quantum mechanical computers. *Found. Phys.* **1986**, *16*, 507–531. [CrossRef]
73. Kato, T. On the Adiabatic Theorem of Quantum Mechanicsk. *J. Phys. Soc. Jpn.* **1950**, *5*, 435–439. [CrossRef]
74. Schrödinger, E. An Undulatory Theory of the Mechanics of Atoms and Molecules. *Phys. Rev.* **1926**, *28*, 1049–1070. [CrossRef]

75. Farhi, E.; Goldstone, J.; Gutmann, S.; Sipser, M. Quantum Computation by Adiabatic Evolution. *arXiv* **2000**, arXiv:quant-ph/0001106.
76. Barends, R.; Shabani, A.; Lamata, L.; Kelly, J.; Mezzacapo, A.; Heras, U.L.; Babbush, R.; Fowler, A.G.; Campbell, B.; Chen, Y.; et al. Digitized Adiabatic Quantum Computing with a Superconducting Circuit. *Nature* **2016**, *534*, 222–226. [CrossRef]
77. Lucas, A. Ising Formulations of Many NP Problems. *Front. Phys.* **2014**, *2*, 5. [CrossRef]
78. Preskill, J. Quantum Computing in the NISQ Era and Beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
79. LaRose, R.; Mari, A.; Kaiser, S.; Karalekas, P.J.; Alves, A.A.; Czarnik, P.; Mandouh, M.E.; Gordon, M.H.; Hindy, Y.; Robertson, A.; et al. Mitiq: A software package for error mitigation on noisy quantum computers. *Quantum* **2022**, *6*, 774. [CrossRef]
80. Treinish, M.; Gambetta, J.; Thomas, S.; Nation, P.; Qiskit-Bot; Kassebaum, P.; Rodríguez, D.M.; De La Puente González, S.; Lishman, J.; Hu, S.; et al. Qiskit/qiskit: Qiskit 0.41.0. *Zenodo* **2023**. [CrossRef]
81. Bergholm, V.; Izaac, J.; Schuld, M.; Gogolin, C.; Ahmed, S.; Ajith, V.; Alam, M.S.; Alonso-Linaje, G.; AkashNarayanan, B.; Asadi, A.; et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations. *arXiv* **2018**, arXiv:1811.04968.
82. Bauckhage, C.; Sanchez, R.; Sifa, R. Problem Solving with Hopfield Networks and Adiabatic Quantum Computing. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–6. [CrossRef]
83. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machines for big data classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [CrossRef] [PubMed]
84. Dallaire-Demers, P.L.; Killoran, N. Quantum generative adversarial networks. *Phys. Rev. A* **2018**, *98*, 012324. [CrossRef]
85. Schuld, M.; Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* **2019**, *122*, 040504. [CrossRef]
86. Villmann, T.; Ravichandran, J.; Engelsberger, A.; Villmann, A.; Kaden, M. Quantum-Inspired Learning Vector Quantization for Classification Learning. In Proceedings of the 28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2020), Bruges, Belgium, 2–4 October 2020; Verleysen, M., Ed.; i6doc.com: Louvain-La-Neuve, Belgium, 2020; pp. 279–284.
87. Jerbi, S.; Fiderer, K.; Nautrup, H.P.; Kübler, J.; Briegel, H.; Dunkjo, V. Quantum machine learning beyond kernel methods. *Nat. Commun.* **2023**, *14*, 517. [CrossRef]
88. Buhrman, H.; Cleve, R.; Watrous, J.; de Wolf, R. Quantum Fingerprinting. *Phys. Rev. Lett.* **2001**, *87*, 167902. [CrossRef]
89. Aharonov, D.; Jones, V.; Landau, Z. A Polynomial Quantum Algorithm for Approximating the Jones Polynomial. *Algorithmica* **2009**, *55*, 395–421. [CrossRef]
90. Schuld, M.; Fingerhuth, M.; Petruccione, F. Implementing a Distance-Based Classifier with a Quantum Interference Circuit. *EPL (Europhys. Lett.)* **2017**, *119*, 6. [CrossRef]
91. Gitiaux, X.; Morris, I.; Emelianenko, M.; Tian, M. SWAP test for an arbitrary number of quantum states. *Quantum Inf. Process.* **2022**, *21*, 344. [CrossRef]
92. Durr, C.; Hoyer, P. A Quantum Algorithm for Finding the Minimum. *arXiv* **1996**, arXiv:quant-ph/9607014.
93. Boyer, M.; Brassard, G.; Høyer, P.; Tapp, A. Tight Bounds on Quantum Searching. *Fortschritte Physik* **1998**, *46*, 493–505. [CrossRef]
94. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the STOC '96: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; ACM Press: New York, NY, USA, 1996; pp. 212–219. [CrossRef]
95. Wiebe, N.; Kapoor, A.; Svore, K.M. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Inf. Comput.* **2015**, *15*, 316–356. [CrossRef]
96. Schuld, M.; Sinayskiy, I.; Petruccione, F. Quantum Computing for Pattern Classification. In *PRICAI 2014: Trends in Artificial Intelligence*; Pham, D.N., Park, S.B., Eds.; Springer: Cham, Switzerland, 2014; pp. 208–220. [CrossRef]
97. Bauckhage, C.; Sifa, R.; Wrobel, S. Adiabatic Quantum Computing for Max-Sum Diversification. In Proceedings of the 2020 SIAM International Conference on Data Mining, Cincinnati, OH, USA, 7–9 May 2020; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2020; pp. 343–351. [CrossRef]
98. Kerenidis, I.; Luongo, A.; Prakash, A. Quantum Expectation-Maximization for Gaussian Mixture Models. In Proceedings of the 37th International Conference on Machine Learning, PMLR, Virtual Event, 13–18 July 2020; pp. 5187–5197.
99. Kerenidis, I.; Landman, J.; Luongo, A.; Prakash, A. Q-Means: A Quantum Algorithm for Unsupervised Machine Learning. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Curran Associates, Inc.: New York, NY, USA, 2019; Volume 32.
100. Cao, Y.; Jiang, S.; Perouli, D.; Kais, S. Solving Set Cover with Pairs Problem using Quantum Annealing. *Sci. Rep.* **2016**, *6*, 33957. [CrossRef]
101. Miller, N.E.; Mukhopadhyay, S. A quantum Hopfield associative memory implemented on an actual quantum processor. *Sci. Rep.* **2021**, *11*, 23391. [CrossRef]
102. Rebentrost, P.; Bromley, T.R.; Weedbrook, C.; Lloyd, S. Quantum Hopfield neural network. *Phys. Rev. A* **2018**, *98*, 042308. [CrossRef]
103. D-Wave. Practical Quantum Computing: An Update. In *D-Wave Whitepaper Series*; D-Wave Systems Inc.: Burnaby, BC, Canada, 2020.
104. Farhi, E.; Goldstone, J.; Gutmann, S. A Quantum Approximate Optimization Algorithm. *arXiv* **2014**, arXiv:1411.4028.
105. Banchi, L.; Quesada, N.; Arrazola, J.M. Training Gaussian Boson Sampling Distributions. *Phys. Rev. A* **2020**, *102*, 012417. [CrossRef]

106. Villmann, A.; Kaden, M.; Saralajew, S.; Villmann, T. Probabilistic Learning Vector Quantization with Cross-Entropy for Probabilistic Class Assignments in Classification Learning. In *Artificial Intelligence and Soft Computing*; Springer: Cham, Switzerland, 2018; pp. 724–735. [CrossRef]
107. Schuld, M.; Killoran, N. Is Quantum Advantage the Right Goal for Quantum Machine Learning? *PRX Quantum* **2022**, *3*, 030101. [CrossRef]