ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Quantum-ready vector quantization: Prototype learning as a binary optimization problem

Alexander Engelsberger[1,2]*, Thomas Villmann[1]

1 - Mittweida University of Applied Sciences
Saxon Institute for Computational Intelligence and Machine Learning
Mittweida - Germany
2 - Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence
University of Groningen
Groningen - The Netherlands

Abstract. Quantum Computing Research proposed strategies to solve binary optimization problems. Application on current and near-term generation Hardware is possible. Even if computational benefits of the strategies are yet to be shown, we want to explore connections to prototype learning schemes. We examine cost functions for vector quantization based on data point selection and how they can be transformed into a common quadratic unconstrained binary optimization formulation (QUBO). There are different approaches for solving QUBO problems using quantum computer or quantum annealer hardware. We look at their current limits and how they might change.

1 Introduction

Quantum computing enters a phase, where small computers become available and chances are, that larger devices are possible in the near-term future. This development has inspired us to get prototype learning ready for different trajectories of quantum research, and their emerging devices based on different physical phenomena. The research field of quantum computing is broad, ranging from Hardware development to investigations of potential application. To categorize these different directions, the European competence framework for quantum technologies [9] was introduced. This work falls in the subdomain 5.5 Quantum Algorithms of this framework. It is a contribution to Quantum Machine Learning, a growing field of quantum algorithms [18, 14].

We will look at prototype learning and its application on quantum devices. Prototype Learning or vector quantization (VQ) is based on the simple nearest neighbor classification (NN) rule, where a data point gets classified by its closest training data point in the sense of a dissimilarity measure. The simplicity of the classification rule will be beneficial, when it gets adapted to current and future quantum hardware. Storing all training data points for NN classification is storage intensive, among other drawbacks. Therefore, Nearest Prototype Classification (NPC) was introduced, it uses the NN rule, but replaces the training dataset by a set of learned prototypes.

*PerspektiveArbeit Lausitz funded by the Federal Ministry of Education and Research of Germany: 02L19C300 - 02L19C327

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Inspired by the mathematics of quantum physics, the prototype learning problem was applied inside the state Hilbert-space of qubits [19]. Going beyond theoretical influences, if a quantum computer is used inside a classical algorithm as a coprocessor doing a non-trivial subroutine, that algorithm is denoted as quantum-hybrid [3]. Quantum-hybrid approaches for vector quantization like Quantum Hybrid LVQ [6, 16] have been investigated. In the hybrid LVQ [6], the quantum computer was used to calculate distances space-efficiently and update the position of prototypes in vector-shift based prototype learning schemes, similar to LVQ 1 [11].

All hybrid approaches are heavily restricted by the current hardware generation. For some optimization problems, specialized hardware, that utilizes quantum effects, is built by the company D-Wave. These devices are called quantum annealers, and they combine a larger number of qubits, while being much noisier. Quantum annealers still have to prove their potential, that is claimed [12] by their creators. In [7] we introduced a connection between vector quantization and optimization problems, along other connections. More precise, we used a set cover problem for VQ [2], that is compatible with quantum computing, especially quantum annealing.

In this paper, we look at a second binary optimization problem statements for vector quantization, based on directly optimizing accuracy [4]. Show the necessary transformations of the problem definitions to be solvable by quantum heuristics. And conclude with an overview of solving strategies for different device categories, what benefits can be expected.

2 Vector Quantization by binary optimization

Given a dataset $\mathcal{X}$ with data points $\boldsymbol{x}_i$, where $i \in 1, \ldots, N$. The task of (learning) vector quantization is to find a prototype set $\mathcal{W}$ with $|\mathcal{W}| \ll |\mathcal{X}|$ that represents the dataset, in the sense of a nearest neighbor classification rule. Following this rule, a data point is assigned to the class of its closest prototype by given dissimilarity d. A key parameter in vector quantization is the number of prototypes in the output. Two mechanisms can decide this number[1]. Either can it be a hyperparameter and therefore user-defined, or the algorithm solves an optimization algorithm that contains the number of prototypes as a trainable parameter.

A special case is defined by requiring $\mathcal{W} \subset \mathcal{X}$. This can be necessary if distances are only available for a discrete set of objects. The quantization part becomes a selection problem. Because of the combinatorial nature of selection, classically, it can only be approximated.

Quadratic Optimization Problem Given variables a_i to optimize, a quadratic problem has a cost function of the form

$$\sum_{ij} w_{ij} a_i a_j + \sum_k w_k a_k.$$

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

If $a_i \in \{0, 1\}$ the problem is called binary. Additionally, we allow linear constraints that restrict the space of possible solutions, either equalities or inequalities:

$$\sum_k y_k a_k = z, \quad \sum_k y_k a_k < z.$$

An optimization problem with a quadratic cost function, binary variables and no constraints is called Quadratic Unconstrained Binary Optimization (QUBO) problem.

Vector Quantization as Set Cover Problem A strategy that optimizes the number of prototypes is the set cover formulation by [2]. This formulation does not perfectly align with the NPC paradigm. Still, its locality by the epsilon balls could be beneficial for larger problems, while giving practical solutions for the NPC inference. It is an extension of the set cover problem. The possible sets are epsilon balls around each data point, and they get a weight, based on their contribution to the classification task, It generates multiple constrained binary optimization problems, one for each class:

$$\begin{aligned} \text{minimize} \quad & \sum_{i:\boldsymbol{x}_i \in \mathcal{X}} C_c(\boldsymbol{x}_i) x_i + \sum_{j:\boldsymbol{x}_j \in \mathcal{X}_c} z_i \\ \text{subject to} \quad & \sum_{i:j \in \epsilon(\boldsymbol{x}_i)} x_i \geq 1 - z_j \qquad \forall \boldsymbol{x}_j \in \mathcal{X}_c. \end{aligned}$$

The variable $x_i \in \{0, 1\}$ indicates if $\boldsymbol{x}_i$ is in the prototype set and $z_j \in \{0, 1\}$ is 0 if $\boldsymbol{x}_j$ is uncovered. The classification task is encoded in the costs of adding, with $\mathcal{X}_c$ being all data points with class c:

$$C_c(\boldsymbol{x}) = \frac{1}{|\mathcal{X}|} + |\epsilon(\boldsymbol{x}) \cap (\mathcal{X} \setminus \mathcal{X}_c)|.$$

Vector Quantization selection problem The NPC problem can directly be written as an optimization problem called optimal p-Prototype Nearest Neighbor model (p-PNN) [4], for a number p of prototypes. Given a dissimilarity matrix, p-PNN generates the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in I} z_i \\ \text{subject to} \quad & \sum_s x_s = p \\ & \sum_{s \in \mathcal{W}_c} x_s \geq 1 \qquad \forall c \in \mathcal{C} \\ & z_i \leq (1 - x_t) + \sum_{s \in \mathcal{W}_{c(i)} \cap \mathcal{W}_{it}} x_s \qquad \forall i \in I, t \notin \mathcal{W}_{c(i)} \end{aligned}$$

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

In this problem $z_i \in \{0, 1\}$ indicates, whether point $\boldsymbol{x}_i$ is classified correctly. The variables $x_s \in \{0, 1\}$ indicate if $\boldsymbol{x}_s \in \mathcal{W}$. To generate all constraints, the set of prototypes candidates, in our case $\mathcal{X}$, is split into subsets. First, $\mathcal{W}_c$ contains all candidates with class c. Further, $\mathcal{W}_{it}$ is the set of candidates preferred by point $\boldsymbol{x}_i$ over $\boldsymbol{x}_t$. A $\boldsymbol{x}_j$ is preferred, if $d(\boldsymbol{x}_i, \boldsymbol{x}_j) < d(\boldsymbol{x}_i, \boldsymbol{x}_t)$, while in case of ties, a point that misclassifies $\boldsymbol{x}_i$ wins.

3 Optimization Methods

3.1 Classical Reference

To compare the two introduced optimization strategies, we use classical solvers with the iris dataset, which is small enough to solve. For both approaches, the constrained problems are generated using the python library PuLP [13].

To compare the strategies, the generated problems can now be solved in their original form with a classical solver framework, e.g., PuLP has support for different commercial (Gurobi, IBM CPLEX) and open-source solvers (GLPK). The results are shown in Figure 1. For the p-PNN we observe that this optimization chooses the class distribution algorithmically, a distribution could be enforced by changing the constraints. We can also confirm that the epsilon ball radius is an unintuitive parameter, and has to be optimized as a hyperparameter, like the authors of the original paper [2] do.

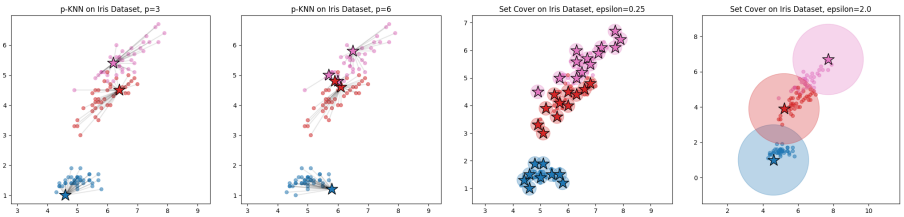

Fig. 1: Comparison of the selection strategy and the set cover strategy for different hyperparameter settings.

3.2 Get Quantum Ready

To solve the combinatorial approaches with quantum devices, they have to be transformed it first. We start from a binary optimization problem with inequality and equality constraints and generate a QUBO problem. To achieve this, we first transform the inequality constraints into equality constraint by adding integer slack variables, which are then encoded by a set of binary variables. All equality constraints can then be transformed into penalty terms. Because all constraints have been linear so far, we are not adding terms of order greater than two, keeping a quadratic cost function.

Inside the python quantum framework Qiskit [15], there is a package for optimization, which implements the necessary steps to transform constrained

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

integer problems into QUBO. With the transformed problem at hand, there are different proposed quantum heuristics (see [7], Section 4.7).

As an example, we transform the smaller set-cover-based approach into a QUBO problem statement. For simplicity, we assume symmetry of the dissimilarity, that is $\boldsymbol{x}_j \in \epsilon(\boldsymbol{x}_i) \iff \boldsymbol{x}_i \in \epsilon(\boldsymbol{x}_j)$ The original statement has inequality constraints of the form:

$$\sum_{i:\boldsymbol{x}_j \in \epsilon(\boldsymbol{x}_i)} p_i^c \geq 1 - z_j.$$

The right side of the equation is 0 or 1, while the left side is between 0 and $|\epsilon(\boldsymbol{x}_j)|$. Each of these constraints can be transformed into equalities by introducing a slack variable $0 \leq s_j \leq |\epsilon(\boldsymbol{x}_j)|$:

$$\sum_{i:\boldsymbol{x}_j \in \epsilon(\boldsymbol{x}_i)} p_i^c + s_j + z_j - 1 = 0.$$

These slack variables can then be replaced by a weighted sum of binary variables $s_{j;n}$ as a binary number, but any binary encoding scheme is possible:

$$s_j = \sum_{n=0}^{\lceil \log_2(|\epsilon(\boldsymbol{x}_j)|) \rceil} s_{j;n} 2^n$$

4 Conclusion

In this work, we presented two vector quantization strategies that can be transformed into optimization problems that can potentially be solved by quantum devices. Both problems are outgrowing current quantum device limits, even for small problems. The largest quantum computer by IBM, a leading company in general purpose quantum computing hardware, has 127 qubits. For quantum annealers, the main competitor just published a paper with a 5000 qubit device [10], but usually qubits get lost by mapping a problem onto a specific hardware, due to low interaction possibilities between the qubits. In addition, the current implementation of quantum devices adds a lot of time overhead, which obscures the potential benefits for small problems. The comparison between classical and quantum solvers would be skewed because classical solvers use some well-tested heuristics to speed up the solution, while quantum optimization is in its early days. Whether quantum devices will ever be capable of solving these problems in application, depends on future hardware developments.

About the capabilities of the devices, there is an ongoing discussion about the concept of quantum annealers, and if they are really beneficial. Some theoretical advances like parity quantum computing [5] could make the problem transformation into QUBO obsolete by introducing new ways to encode integer optimization problems onto quantum hardware. Recently, the question has been raised, whether theoretical quantum advantage is even the right goal for quantum machine learning [17]. Transforming vector quantization into QUBO is not

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

only interesting for quantum devices. As a heuristic for neuromorphic hardware, another emerging paradigm, were presented [8].

If the hardware grows into the size of application problems, the optimization approaches would make vector quantization less ambiguous. The resulting prototypes would only be dependent on the selected dissimilarities, removing variances introduced by the approximation strategies. Until then, the potential of quantum solutions makes approaches, which are classical intractable and therefore less attractive, worth another look.

References

[1] J. C. Bezdek and L. I. Kuncheva. Nearest Prototype Classifier Designs: An Experimental Study. *International Journal of Intelligent Systems*, 16(12):1445–1473, Dec. 2001.

[2] J. Bien and R. Tibshirani. Prototype Selection for Interpretable Classification. *The Annals of Applied Statistics*, 5(4), Dec. 2011.

[3] A. Callison and N. Chancellor. Hybrid Quantum-Classical Algorithms in the Noisy Intermediate-Scale Quantum Era and Beyond. *Physical Review A*, 106(1):010101, July 2022.

[4] E. Carrizosa et al. On the Selection of the Globally Optimal Prototype Subset for Nearest-Neighbor Classification. *INFORMS Journal on Computing*, 19(3):470–479, Aug. 2007.

[5] K. Ender et al. Parity Quantum Optimization: Compiler. *Quantum*, 7:950, Mar. 2023.

[6] A. Engelsberger et al. Steps Forward to Quantum Learning Vector Quantization for Classification Learning on a Theoretical Quantum Computer. In J. Faigl, M. Olteanu, and J. Drchal, editors, *Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization*, volume 533, pages 63–73. Springer International Publishing, Cham, 2022.

[7] A. Engelsberger and T. Villmann. Quantum Computing Approaches for Vector Quantization—Current Perspectives and Developments. *Entropy*, 25(3):540, Mar. 2023.

[8] Y. Fang and A. S. Lele. Solving Quadratic Unconstrained Binary Optimization with Collaborative Spiking Neural Networks. In *2022 IEEE International Conference on Rebooting Computing (ICRC)*, pages 84–88, Dec. 2022.

[9] F. Greinert and R. Müller. European Competence Framework for Quantum Technologies. Apr. 2023.

[10] A. D. King et al. Quantum Critical Dynamics in a 5,000-Qubit Programmable Spin Glass. *Nature*, pages 1–6, Apr. 2023.

[11] T. Kohonen. Learning Vector Quantization. *Neural Networks*, 1:303, Jan. 1988.

[12] C. C. McGeoch and P. Farre. Milestones on the Quantum Utility Highway, May 2023.

[13] S. Mitchell et al. PuLP: A Linear Programming Toolkit for Python.

[14] D. Pastorello. *Concise Guide to Quantum Machine Learning*. Machine Learning: Foundations, Methodologies, and Applications. Springer Nature Singapore, Singapore, 2023.

[15] Qiskit contributors. Qiskit/qiskit: Qiskit 0.42.1. Zenodo, Mar. 2023.

[16] M. Schuld et al. Quantum Computing for Pattern Classification. In D.-N. Pham and S.-B. Park, editors, *PRICAI 2014: Trends in Artificial Intelligence*, volume 8862, pages 208–220. Springer International Publishing, Cham, 2014.

[17] M. Schuld and N. Killoran. Is quantum advantage the right goal for quantum machine learning?, Mar. 2022.

[18] M. Schuld and F. Petruccione. *Machine Learning with Quantum Computers*. Quantum Science and Technology. Springer International Publishing, Cham, 2021.

[19] T. Villmann et al. Quantum-Inspired Learning Vector Quantizers for Prototype-Based Classification. *Neural Computing and Applications*, 34(1):79–88, Jan. 2022.