ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Quantum-Inspired Learning Vector Quantization for Classification Learning

T. Villmann[1], J. Ravichandran[1], A. Engelsberger[1], A. Villmann[2], and M. Kaden[1] *

1- University of Applied Sciences Mittweida,
Saxony Institute for Comp. Intelligence and Machine Learning
Mittweida - Germany

2- Berufliches Schulzentrum Döbeln-Mittweida - Germany

Abstract. This paper introduces a variant of the prototype-based generalized learning vector quantization algorithm (GLVQ) for classification learning, which is inspired by quantum computing. Starting from the motivation of kernelized GLVQ, the nonlinear transformation of real data and prototypes into quantum bit vectors allows to formulate a GLVQ variant in a (n-dimensional) quantum bit vector space $\mathcal{H}^n$. A key feature for this approach is that $\mathcal{H}^n$ is an Hilbert space with particular inner product properties, which finally restrict the prototype adaptation to be unitary transformations. The resulting approach is denoted as Qu-GLVQ. We provide the mathematical framework and give exemplary numerical results.

1 Introduction

Learning vector quantizers (LVQ) are sparse models for classification learning, which constitute a promising alternative to deep networks for many applications [18]. Although heuristically introduced, nowadays LVQ variants are well-defined in terms of loss functions and respective gradient based optimization schemes can be applied (Generalized LVQ - GLVQ,[12]). Of particular interest for user is the advantage of easy interpretability of LVQ networks [21], as this feature is being required more and more in machine learning [10]. Further, LVQ networks are proven to be robust against adversarial attacks [11].

Although LVQ networks usually rely on the Euclidean metric to compare prototype vectors with data vectors, other proximity measures can be applied [3]. To this end, kernel metrics play an important role. If differentiable kernels are considered, the kernel trick known from support vector machines (SVM) can be adopted for GLVQ [13, 19]. Doing so, the data as well as the prototypes are implicitly mapped into a high-dimensional Hilbert space where the mapped data form a low-dimensional manifold. This mapping frequently is non-linear depending on the kernel in use. This implicit non-linear mapping into the potentially infinite-dimensional Hilbert space provides an increased flexibility for the prototype adaptation, which frequently leads to improved performance compared to standard (Euclidean) LVQ as it also observed to be a key feature for high performances of SVMs [6]. A disadvantage of this approach, however, is that the (mapped) prototypes move freely in the Hilbert space and, therefore, may leave the low-dimensional manifold. Thus the interpretability of the kernel

*M.K. was supported by grants of the European Social Fund (ESF) for a Young Researcher Group 'MACS' in cooperation with the TU Bergakademie Freiberg (Germany) and for the project titled 'Digitale Produkt- und Prozessinovationen' at the UAS Mittweida. J.R. and A.E. are supported by a PhD-grant of ESF.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

LVQ network is reduced because the (mapped) prototypes may observe the data from 'outside'.

In this contribution we pick up the idea of non-linear data mapping into a particular Hilbert space but restrict the GLVQ-prototypes to live in that space. In particular, we transform the data vectors non-linearily into *quantum state vectors* and require at the same time that the prototypes are always quantum state vectors, too. Thus the prototype adaptation has to keep the quantum state vector property, i.e. prototype adaptations are restricted to be unitary transformations. We show that this approach is mathematically consistent with the standard as well as kernel GLVQ and give exemplary application examples.

We introduce a GLVQ for classification of quantum state vectors and describe the learning dynamic in terms of quantum dynamic transformation. More precisely, we assume that the data as well as the GLVQ-prototypes are given as quantum state vectors. The respective prototype adaptation keeps the attraction and repulsing strategy known to be the essential ingredients of LVQ algorithms but here based on quantum dynamics. In particular, the adaptation follows unitary transformations, which ensure reversibility. The resulting algorithm is denoted as *Quantum GLVQ* (Qu-GLVQ).

2 Quantum-inspired GLVQ approach

2.1 Standard GLVQ

Standard GLVQ as introduced in [12] assumes data vectors $\mathbf{v} \in V \subseteq \mathbb{R}^n$ with class labels $c(\mathbf{v}) \in \mathcal{C} = \{1, \ldots, C\}$ for training. Further, a set $P = \{\mathbf{p}_k\} \subset \mathbb{R}^n$ of prototypes with class labels $c_k = c(\mathbf{p}_k)$ is supposed together with a differentiable dissimilarity measure $d(\mathbf{v}, \mathbf{p}_k)$ frequently chosen as the (squared) Euclidean distance. Classification of an unknown data sample takes place as a nearest prototype decision with the class label of the winning prototype as response. GLVQ considers the cost function $E_{GLVQ}(V, P) = \sum_{\mathbf{v} \in V} L(\mathbf{v}, P)$ for optimization of the prototypes with the local loss

$$L(\mathbf{v}, P) = f(\mu(\mathbf{v})) \tag{1}$$

where f is the activation function frequently chosen as sigmoid and

$$\mu(\mathbf{v}) = \frac{d^+(\mathbf{v}) - d^-(\mathbf{v})}{d^+(\mathbf{v}) + d^-(\mathbf{v})} \tag{2}$$

is the classifier function $\mu(\mathbf{v}) \in [-1, 1]$ with $d^\pm(\mathbf{v}) = d(\mathbf{v}, \mathbf{p}^\pm)$ is the dissimilarity of a given input to the best matching correct/incorrect prototype $\mathbf{p}^\pm$ regarding the class labels. The function $\mu(\mathbf{v})$ delivers negative values for correct classification. Thus, $E_{GLVQ}(V, P)$ approximates the classification error an is optimized by stochastic gradient descent learning regarding the prototypes according to

$$\Delta \mathbf{p}^\pm \propto -\frac{\partial L(\mathbf{v}, P)}{\partial \mathbf{p}^\pm} = -\frac{\partial f \mu(\mathbf{v})}{\partial \mu} \frac{\partial \mu(\mathbf{v})}{\partial d^\pm} \frac{\partial d^\pm(\mathbf{v})}{\partial \mathbf{p}^\pm} \tag{3}$$

as local derivatives. It constitutes a margin classifier and is robust against adversarial attacks [5, 11].

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

For the kernel GLVQ (KGLVQ) discussed in [19], the dissimilarity measure $d(\mathbf{v}, \mathbf{p}_k)$ is set to be the (squared) kernel distance

$$d_\kappa(\mathbf{v}, \mathbf{p}_k) = \kappa(\mathbf{v}, \mathbf{v}) - 2\kappa(\mathbf{v}, \mathbf{p}_k) + \kappa(\mathbf{p}_k, \mathbf{p}_k)$$

with the differentiable kernel κ and the respective (implicit) kernel mapping $\Phi_\kappa : \mathbb{R}^n \to H$ into the reproducing kernel Hilbert space H [17]. As mentioned in the introduction, the implicitly mapped data $\Phi_\kappa(\mathbf{v})$ form a low-dimensional manifold $D_\Phi(V)$ in the feature mapping space H whereas the prototypes $\Phi_\kappa(\mathbf{p}_k)$ are allowed to move freely in H and, hence may leave $D_\Phi(V)$. In this case, the prototypes recognize the data from outside, which could be a disadvantage for particular applications.

2.2 Quantum-inspired GLVQ

2.2.1 Quantum bits, Quantum State Vectors and Transformations

Quantum-inspired machine learning gains more and more attention [4, 9, 16]. Following the usual notations, the data are required to be quantum bits (*qubits*)

$$|x\rangle = \alpha(|x\rangle) \cdot |0\rangle + \beta(|x\rangle) \cdot |1\rangle \tag{4}$$

$$= \begin{bmatrix} \alpha(|x\rangle) \\ \beta(|x\rangle) \end{bmatrix} \tag{5}$$

with the normalization condition

$$|\alpha(|x\rangle)|^2 + |\beta(|x\rangle)|^2 = 1 \tag{6}$$

defining the Bloch-sphere [22] and

$$\langle x|y\rangle = \alpha(|x\rangle)\,\alpha(|y\rangle) + \beta(|x\rangle)\,\beta(|y\rangle) \tag{7}$$

as the dot product calculated by the Euclidean inner product of the amplitude vectors $\begin{bmatrix} \alpha(|x\rangle) \\ \beta(|x\rangle) \end{bmatrix}$ and $\begin{bmatrix} \alpha(|y\rangle) \\ \beta(|y\rangle) \end{bmatrix}$. Hence, we have $\langle x|x\rangle = 1$ such that the qubit distance δ can be calculated as

$$\delta(|x\rangle, |y\rangle) = \sqrt{2} \cdot \sqrt{(1 - \langle x|y\rangle)}\,. \tag{8}$$

Transformations $\mathbf{U} \begin{bmatrix} \alpha(|x\rangle) \\ \beta(|x\rangle) \end{bmatrix} = \mathbf{U}|x\rangle$ of qubits are realized by *unitary matrices* $\mathbf{U} \in \mathbb{C}^{2\times 2}$, i.e. $\mathbf{U} \cdot \mathbf{U}^* = \mathbf{E}$ with $\mathbf{U}^*$ is the Hermitian transpose. If $\mathbf{U} \in \mathbb{R}^{2\times 2}$ then $\mathbf{U}$ has to be orthonormal. Further, the unitary matrices remain the inner product invariant, i.e. $\langle \mathbf{U}x|\mathbf{U}y\rangle = \langle x|y\rangle$.

Now we define *qubit vectors* as $|\mathbf{x}\rangle = (|x_1\rangle, \ldots, |x_n\rangle)^T$ with qubits $|x_k\rangle$. For the inner product we get $\langle \mathbf{x}|\mathbf{w}\rangle_{\mathcal{H}^n} = \sum_{k=1}^n \langle x_k|w_k\rangle$ with $\langle \mathbf{x}|\mathbf{x}\rangle = n$ according to the normalization condition (6). Thus we get

$$\delta(|\mathbf{x}\rangle, |\mathbf{w}\rangle) = 2\,(n - \langle \mathbf{x}|\mathbf{w}\rangle_{\mathcal{H}^n}) \tag{9}$$

as the squared distance between qubit vectors. Unitary transformations of qubit vectors are realized by block-diagonal matrices according to $\mathbf{U}^{(n)}|\mathbf{x}\rangle = \mathrm{diag}(\mathbf{U}_1, \ldots, \mathbf{U}_n) \cdot |\mathbf{x}\rangle$. Obviously, the quantum space $\mathcal{H}^n$ of n-dimensional qubit vectors is an Hilbert space with the inner product $\langle \mathbf{x}|\mathbf{w}\rangle_{\mathcal{H}^n}$ as also recognizeed in [14, 15].

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

2.2.2 The Quantum-inspired GLVQ algorithm

The Quantum-inspired GLVQ approach (Qu-GLVQ) takes qubit vectors $|\mathbf{x}\rangle$ and $|\mathbf{w}\rangle$ as elements of the data and the prototype sets X and W, respectively. Thus, the dissimilarity measure $d^{\pm}(\mathbf{v}) = d(\mathbf{v}, \mathbf{p}^{\pm})$ in (2) has to be replaced by the squared qubit vector distance $\delta(|\mathbf{x}\rangle, |\mathbf{w}\rangle)$ from (9). Starting from usual real data vectors, we take a *non-linear* mapping $\Phi : \mathbb{R}^n \ni \mathbf{v} \to |\mathbf{x}\rangle \in \mathcal{H}^n$ to obtain qubit vectors. In context of quantum machine learning Φ is also denoted as *quantum feature map* [15, 14]. This mapping can be realized taking

$$\alpha(|x_l\rangle) = \cos(\xi_l) \text{ and } \beta(|x_l\rangle) = \sin(\xi_l)$$

keeping in mind the normalization (6) and applying an appropriate squashing function $\varphi : \mathbb{R} \to [0, 2\pi]$ such that $\xi_l = \varphi(v_l)$ is valid.[1] Possible choices are

$$\varphi(v_l) = \frac{2\pi}{1 + \exp(-v_l)} \text{ or } \varphi(v_l) = \pi \cdot (\tanh(v_l) + 1)$$

as suggested in [7]. The components of the prototype qubit vector are respectively considered as

$$|w_l\rangle = \cos(\omega_l) \cdot |0\rangle + \sin(\omega_l) \cdot |1\rangle$$

with the angle $\omega_l \in [0, 2\pi]$ as the determining quantity, which form the angle vector $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)^T$. Thus the prototype update can be realized adapting the angle vectors in complete analogy to (3) according to

$$\Delta\boldsymbol{\omega}^{\pm} \propto -\frac{\partial L(|\mathbf{x}\rangle, W)}{\partial \boldsymbol{\omega}^{\pm}} = -\frac{\partial f\mu(|\mathbf{x}\rangle)}{\partial \mu} \frac{\partial \mu(|\mathbf{x}\rangle)}{\partial d^{\pm}} \frac{\partial \delta^{\pm}(|\mathbf{x}\rangle)}{\partial |\mathbf{w}^{\pm}\rangle} \frac{\partial |\mathbf{w}^{\pm}\rangle}{\partial \boldsymbol{\omega}^{\pm}} \tag{10}$$

where $\delta^{\pm}(|\mathbf{x}\rangle) = \delta(|\mathbf{x}\rangle, |\mathbf{w}^{\pm}\rangle)$ and the componentwise derivatives

$$\frac{\partial |\mathbf{w}^{\pm}\rangle}{\partial \omega_k^{\pm}} = -\cos(\xi_k) \cdot \sin\left(\omega_k^{\pm}\right) + \sin(\xi_k) \cdot \cos\left(\omega_k^{\pm}\right) \tag{11}$$

delivers the gradient vector $\frac{\partial |\mathbf{w}^{\pm}\rangle}{\partial \boldsymbol{\omega}^{\pm}}$. We observe that the unitary transformations

$$\mathbf{U}_k(\Delta\omega_k) \cdot |w_k\rangle = \begin{pmatrix} \cos(\Delta\omega_k) & -\sin(\Delta\omega_k) \\ \sin(\Delta\omega_k) & \cos(\Delta\omega_k) \end{pmatrix} \begin{bmatrix} \alpha(|w_k\rangle) \\ \beta(|w_k\rangle) \end{bmatrix} \tag{12}$$

would realize the update $\Delta|\mathbf{w}^{\pm}\rangle$ directly in the quantum space $\mathcal{H}^n$. Further, we can collect all transformations by $\mathbf{U}_k^{\Sigma} = \Pi_{t=1}^{N} \mathbf{U}_k(\Delta\omega_k(t))$ where $\Delta\omega_k(t)$ is angle change at time step t. Due to the group property of orthogonal transformations the matrix $\mathbf{U}_k^{\Sigma}$ is also orthogonal and allows toc re-calculate the initial state $|w_k\rangle$ from the final.

[1] In this approach we restrict the amplitudes to be $\alpha, \beta \in \mathbb{R}$, which reduces the Bloch-sphere to a circle. With other words, here the phase information is set to $e^{i\phi}$ with $\phi = 0$.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

Dataset	N	d	#C	GLVQ	Qu-GLVQ	KGLVQ	SVM
WHISKY	1188	401	3	92.5 ±1.9	92.6 ±1.4	86.9 ±2.8	99.3 ±0.7 ($\|SV\|$:140)
WBCD	569	30	2	96.8 ±2.7	96.1 ±3.4	94.0 ±3.4	96.5 ±1.5 ($\|SV\|$: 43)
HEART	297	13	2	56.9 ±4.0	81.8 ±5.3	84.3 ±5.4	80.4 ±9.5 ($\|SV\|$: 97)
FLC1	82045	12	10	92.8 ±0.3	92.0 ±0.4	94.8 ±0.4	97.4 ±0.1 ($\|SV\|$: 2206)
PIMA	768	8	2	77.3 ±2.3	75.4 ±2.4	76.4 ±4.6	75.5 ±4.1 ($\|SV\|$: 311)

Table 1: Classification accuracies in % and standard deviations for GLVQ, Qu-GLVQ, KGLVQ, and SVM for the considered datasets (ten-fold cross-validation). For all LVQ-algorithms only one prototype per class was used. The number of support vectors for SVM is given by SV. N - number of data samples, d - data dimensionality, $\#C$ - number of classes.

3 Numerical Experiments

We tested the Qu-GLVQ algorithm for several data sets in comparison to standard GLVQ, KGLVQ and SVM, both SVM and KGLVQ with rbf-kernel. The data sets are a) WHISKY - a spectral data set to classify Scottish whisky described in [2, 1], WBCD - UCI Wisconcin Breast Cancer Data Set, HEART - UCI Heart disease data set, PIMA - UCI diabetes data set, and FLC1 - staellite remote sensing LANDSAT TM data set [8].[2]

We observe an overall good performance of QU-GLVQ compared to the other methods. If kernel methods seem to be beneficial as indicated by KGLVQ and SVM for HEART, Qu-GLVQ delivers similar results. If SVM yields significant better performance than LVQ methods, then we have to take into account that here the SVM complexity (number of support vectors) is much higher than in LVQ-networks, where the number of prototypes was chosen always to be only one per class. Further, for WHISKY the KGLVQ was not able to achieve a classification accuracy comparable to the other approaches, whereas Qu-GLVQ performed well. A reason could be that the prototypes in the feature mapping space do not belong to the data manifold in the kernel feature mapping space, as discussed in sec 2.1.

4 Conclusion

In this contribution we introduced a GLVQ-variant inspired by quantum machine learning strategies (Qu-GLVQ). Usual data and prototype vectors are replaced by respective quantum bit vectors as a result of a non-linear mapping. This approach shows mathematical equivalence to kernel approaches in topological sense. The resulting adaptation dynamic in Qu-GLVQ is consistent with the unitary transformations required for quantum state changes. Further investigations should include modifications of transfer function as proposed in [20] as well as the consideration of entanglements for qubits and complex amplitudes $\beta \in \mathbb{C}$ for qubits.

References

[1] P. Ashok, B. Praveen, and K. Dholakia. Near infrared spectroscopic analysis of single malt Scotch whisky on an optofluidic chip. *Optics Express*, 19(23):1–11, 2011.

[2]The UCI-datasets are available from: https://archive.ics.uci.edu/ml .

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

[2] A. Backhaus, P. Ashok, B. Praveen, K. Dholakia, and U. Seiffert. Classifying scotch whisky from near-infrared Raman spectra with a radial basis function network with relevance learning. In M. Verleysen, ed., *Proc. of the European Symposium on Artificial Neural Networks and Machine Learning (ESANN)*, pages 411–416, Brussels, 2012. i6doc.com.

[3] M. Biehl, B. Hammer, and T. Villmann. Prototype-based models in machine learning. *Wiley Interdisciplinary Reviews: Cognitive Science*, (2):92–111, 2016.

[4] C. Ciliberto, M. Herbster, A. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig. Quantum machine learning: A classical perspective. *Proceedings of the Royal Society A*, 474(2209):1–26, 2018.

[5] K. Crammer, R. Gilad-Bachrach, A. Navot, and A.Tishby. Margin analysis of the LVQ algorithm. In S. Becker, S. Thrun, and K. Obermayer, eds., *Advances in Neural Information Processing (NIPS 2002)*, vol. 15, p. 462–469, Cambridge, MA, 2003. MIT Press.

[6] T. Hoffmann, B. Schölkopf, and A. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171–1220, 2008.

[7] X. Hou. Research of model of quantum learning vector quantization neural network. In *Proceedings of the International Conference on Electronic and Mechanical Engineering and Information Technology*, pages 3893–3896, Los Alamitos, 2011. IEEE Society.

[8] D. Landgrebe. *Signal Theory Methods in Multispectral Remote Sensing*. Wiley, Hoboken, New Jersey, 2003.

[9] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 10th edition, 2016.

[10] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and useinterpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[11] S. Saralajew, L. Holdijk, M. Rees, and T. Villmann. Robustness of generalized learning vector quantization models against adversarial attacks. In A. Vellido, K. Gibert, C. Angulo, and J. Guerrero, eds., *Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization – Proc. of the 13th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization, WSOM+2019, Barcelona*, vol. 976 of *Advances in Intelligent Systems and Computing*, pages 189–199. Springer Berlin-Heidelberg, 2019.

[12] A. Sato and K. Yamada. Generalized learning vector quantization. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, eds., *Advances in Neural Information Processing Systems 8 (NIPS 1995)*, pages 423–9. MIT Press, Cambridge, MA, USA, 1996.

[13] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, 2002.

[14] M. Schuld. Machine learning in quantum spaces. *Nature*, 567:179–181, 2019.

[15] M. Schuld and N. Killoran. Quantum machine learning in feature hilbert spaces. *Physical Review Letters*, 122(040504):1–6, 2019.

[16] M. Schuld, I. Sinayskiy, and F. Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, pages 1–14, 2014.

[17] I. Steinwart and A. Christmann. *Support Vector Machines*. Information Science and Statistics. Springer Verlag, Berlin-Heidelberg, 2008.

[18] T. Villmann, A. Bohnsack, and M. Kaden. Can learning vector quantization be an alternative to SVM and deep learning? *Journal of Artificial Intelligence and Soft Computing Research*, 7(1):65–81, 2017.

[19] T. Villmann, S. Haase, and M. Kaden. Kernelized vector quantization in gradient-descent learning. *Neurocomputing*, 147:83–95, 2015.

[20] T. Villmann, J. Ravichandran, A. Villmann, D. Nebel, and M. Kaden. Investigation of activation functions for Generalized Learning Vector Quantization. In A. Vellido, K. Gibert, C. Angulo, and J. Guerrero, eds., *Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization – Proc. of the 13th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization, WSOM+2019, Barcelona*, vol. 976 of *Advances in Intelligent Systems and Computing*, pages 179–188. Springer Berlin-Heidelberg, 2019.

[21] T. Villmann, S. Saralajew, A. Villmann, and M. Kaden. Learning vector quantization methods for interpretable classification learning and multilayer networks. In C. Sabourin, J. Merelo, A. Barranco, K. Madani, and K. Warwick, eds., *Proc. of the 10th International Joint Conference on Computational Intelligence (IJCCI), Sevilla*, pages 15–21, Lissabon, Portugal, 2018. SCITEPRESS - Science and Technology Publications, Lda.

[22] P. Wittek. *Quantum Machine Learning - What Quantum Computing Means to Data Mining*. Elsevier Insights. Elsevier, 1st edition, 2014.