Exploring the Capabilities of Quantum Support Vector Machines for Image Classification on the MNIST Benchmark

Mateusz Slysz[1,2][0000−0003−3124−9899], Krzysztof Kurowski[1][0000−0002−4478−6119], Grzegorz Waligóra[2][0000−0003−2108−1113], and Jan Węglarz[2][0000−0002−2237−3479]

[1] Poznań Supercomputing and Networking Center, IBCH PAS
{mslysz,krzysztof.kurowski}@man.poznan.pl
[2] Poznań University of Technology
Institute of Computing Science
Poznań, Poland
{grzegorz.waligora,jan.weglarz}@cs.put.poznan.pl

Abstract. Quantum computing is a rapidly growing field of science with many potential applications. One such field is machine learning applied in many areas of science and industry. Machine learning approaches can be enhanced using quantum algorithms and work effectively, as demonstrated in this paper. We present our experimental attempts to explore Quantum Support Vector Machine (QSVM) capabilities and test their performance on the collected well-known images of handwritten digits for image classification called the MNIST benchmark. A variational quantum circuit was adopted to build the quantum kernel matrix and successfully applied to the classical SVM algorithm. The proposed model obtained relatively high accuracy, up to 99%, tested on noiseless quantum simulators. Finally, we performed computational experiments on real and recently setup IBM Quantum systems and achieved promising results of around 80% accuracy, demonstrating and discussing the QSVM applicability and possible future improvements.

Keywords: Quantum Support Vector Machine · Quantum Kernel Alignment · Image Classification · MNIST Benchmark

1 Introduction

Quantum computing is a growing field of science that uses quantum phenomena and brings much hope for more efficiency than classical computing. It has been proven theoretically that some classically intractable problems can be solved more effectively using quantum approaches. Examples of well-known quantum algorithms include Shor's algorithm for factorizing large numbers [19], or Grover's algorithm for quickly searching through unsorted data sets [8]. However, we are currently in the Noisy Intermediate Scale Quantum (NISQ) era [16], which, due to hardware implementation difficulties, is characterized by quantum devices with limited capacity and accuracy. While attempts to create fault-tolerant

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_15

quantum computers are underway, researchers are looking for different areas in which they could exploit the capabilities of existing quantum devices for computation. Examples of such fields include discrete optimization [6], simulation of quantum systems [11], and machine learning [5][13][18], also specifically for image classification purposes [1][15][20].

The aim of this paper is to demonstrate the potential advantages of existing NISQ devices used in a machine learning image classification problem. Using a gate-based quantum device, we run experiments with the Support Vector Machine (SVM) algorithm [3] with quantum kernels to classify a benchmark dataset.

2 Theoretical background

2.1 Introduction to quantum computing

In a nutshell, quantum computers use qubits instead of bits to encode information. Qubits are two-level quantum variables and are subject to follow principles of quantum mechanics, such as superposition and entanglement. Superposition is a feature of quantum systems that allows particles to be in many states simultaneously (in this case pseudo binary states $|0\rangle$ and $|1\rangle$ are used), and entanglement is a property that binds quantum variables together in a way that allows to transfer more information than would be classically possible. These quantum effects can be exploited by cleverly designed algorithms to achieve faster and better results than it would be possible with any classical machines [14].

2.2 Support Vector Machine

A classical Support Vector Machine is a machine learning model used for data classification and regression. SVM parameters w and b are trained to find an optimal separating hyperplane f to classify data points x into classes (nominally $y = \pm 1$):

$$f(x) = \operatorname{sign}(w \cdot x + b) \tag{1}$$

This problem of maximizing the margin between decision classes resolves to an optimization problem:

$$\begin{aligned} &\min_{w} \frac{1}{2}\|w\|^2 \\ \text{subject to} \quad &y_i (w \cdot x_i + b) \geq 1 \end{aligned} \tag{2}$$

However, classes in most datasets are not linearly separable, so SVMs must be able to tackle this problem. It is done by using kernel functions to map data to some high-dimensional feature space, in which it is possible to separate them with a hyperplane. A kernel function K is, in fact, a dot product between feature maps ϕ. For a pair of points, x_i, x_j it is given by

$$K(x_i, x_j) = \langle \phi(x_i) | \cdot | \phi(x_j) \rangle \tag{3}$$

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_15

and, for the sake of our optimization problem, should keep the similarity measure between data points as close as possible to the original feature space.

After finding the optimal parameter values, a new example $\tilde{x}$ is classified by the model following

$$f(\tilde{x}) = \text{sign}\left(\sum_i y_i \alpha_i K(x_i, \tilde{x}) + b\right) \tag{4}$$

This so-called kernel trick is a very useful tool and allows achieving good results on complicated datasets, without losing the ability of generalization. However, selecting a kernel is a difficult task and different kernel functions (e.g. linear, polynomial, Radial Based Functions, etc.) and have different strengths and weaknesses, making it worthwhile to look for more and more mapping functions that are better suited to specific applications.

2.3 Quantum Kernel

For this reason, we experimented with a new type of kernel which is computed using a quantum computer. Based on [9] and [7] we encode the kernel function as a readout of a parametrised quantum circuit. The data x is mapped to an n-qubit quantum feature state through a unitary operator $U(x)$:

$$\phi(x) = U(x) \left|0^n\right\rangle \left\langle 0^n\right| U^\dagger(x) \tag{5}$$

From equations (3) and (5) it follows that the kernel function of two variables x_i and x_j is of the form of

$$K(x_i, x_j) = \| \left\langle 0^n | U^\dagger(x_i) U(x_j) | 0^n \right\rangle \|^2 \tag{6}$$

which can be estimated as the readout from a quantum circuit implementing $U^\dagger(x_i)U(x_j)$. Since the characteristics of the quantum circuit representing the kernel function must depend not only on the input data x, but also on the parameters describing the feature map, we must take into consideration the more general definition of the feature map, which can be described as:

$$\phi(x) = D_x \left|\psi\right\rangle \left\langle\psi\right| D_x^\dagger \tag{7}$$

where $|\psi\rangle$ is some fiducial quantum state, corresponding to the feature map, and D_x is some data-dependent unitary operator. We can impose a condition in which the $|\psi\rangle$ state value is the effect of applying an operator V_λ on the initial state $|0^n\rangle$:

$$\left|\psi\right\rangle = V_\lambda \left|0^n\right\rangle \tag{8}$$

The unitary operator $U(x)$ is now a combination of V_λ and D_x, so the kernel function can be described as

$$K(x_i, x_j) = \| \left\langle 0^n | V_\lambda^\dagger D_{x_i}^\dagger D_{x_j} V_\lambda | 0^n \right\rangle \|^2 \tag{9}$$

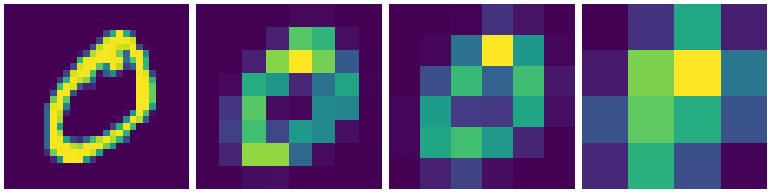
ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_15

This can be represented in the form of a quantum variational circuit, which was designed based on [7]. For the data encoding D block, $R_X(\theta)$ and $R_Z(\theta)$ gates were chosen to represent data points, with angles θ corresponding to the numerical values of variables. Due to the fact that R_X and R_Z are orthogonal state rotations, it is possible to encode two data features on a single qubit – one for each axis. In the variational V block $R_Y(\lambda)$ gates were used and λ parameters are subject to training, alongside with CZ gates to create entanglement between qubits in the quantum circuit.

3 Computational experiments

To test and compare the behaviour of the QSVM algorithm with different parameters, we decided to apply it for image classification. We tested the algorithm on images from the well-known MNIST dataset, containing images of handwritten digits at 28×28 resolution [12]. The data was filtered only to have two classes – '0's and '1's, as the basic version of the SVM algorithm only supports binary classification.

Since the limitations of currently available quantum computers and simulators did not allow the algorithm to fit a full-sized data set with a feature space of 784, reducing the number of features was necessary. Since the data is image-based, it was possible to use image scaling algorithms such as *resize* from Python *scikit-image* library [22]. We obtained images with smaller resolutions of 4×4, 6×6 and 8×8. After scaling, we got data points of dimensionality 16, 36 and 64, respectively, which correspond to the size of a quantum circuit equal to 8, 18 and 32 qubits.



(a) 28×28 (b) 8×8 (c) 6×6 (d) 4×4

Fig. 1: A grid of original size and resized images of a digit '0' in different resolutions.

The quantum circuit was designed so that CZ gates that create an entangled state connect each qubit with its two neighbours. This method allowed to create a maximum number of connections between neighbouring variables, as the original data represents pixels, thus having a spatial interpretation. An example of the circuit with randomly chosen starting parameters λ and two data points from the training set x_1 and x_2 encoded is shown in Figure 2.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_15

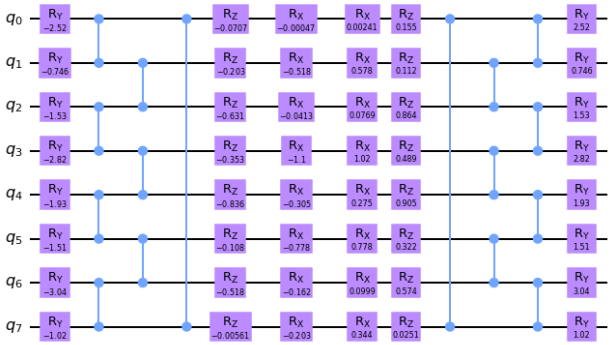


Fig. 2: An example quantum circuit for 2 digits from the training set.

Using this representation, we could run our experiments using Qiskit [17] and IBM Quantum services for all the resolutions on IBM *qasm_simulator*, which supports the simulation of quantum circuits up to 32 qubits. However, due to the connectivity limitations of actual quantum devices, it has been much more difficult to fit such circuits on quantum processors. We were limited to running experiments using 4×4 resolution images on available 27-qubit quantum computers (e.g. *ibmq_montreal* with 128 Quantum Volume) and on 127-qubit *ibm_washington*[10].

The training process was performed using the Simultaneous Perturbation Stochastic Approximation (SPSA) [21] algorithm to find the optimal set of λ parameters and solve the kernel alignment problem [4]. Based on averaged readouts from 1024 samples of the circuit executions, it was possible to construct an approximated kernel matrix that best described the similarity between data points in the dataset. The matrix dimensions were the same as the training set size and was equal to 20. Figure 3 shows an example kernel matrix.

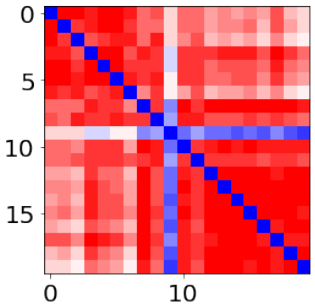


Fig. 3: Example kernel matrix of size 20×20 computed on a NISQ device.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_15

This matrix was later used as a pre-computed kernel for the SVM algorithm with a soft margin [3] factor $C = 1$ and the maximum number of the SPSA iterations = 10 and used for classification on the test set of size 10. The *balanced accuracy* measure [2] was used as a quality indicator of the algorithm. Each configuration was run 10 times. The averaged results obtained from a set of readouts comparing different image resolutions and different devices were presented in Table 1.

To compare the quantum algorithm with a benchmark model, we conducted experiments using the classical SVM algorithm with a linear kernel. To maintain a fair comparison, data preprocessing was done identically, and the instance sizes were also the same. The results of the balanced accuracy for the classical SVM for 3 resolutions - 4×4, 6×6 and 8×8 are also shown in the Table 1.

Computer	Number of qubits	Instance size	Balanced Accuracy [%]
classical	-	4×4	89.36
		6×6	97.71
		8×8	99.38
qasm_simulator	32	4×4	79.83
		6×6	95.79
		8×8	99.29
ibmq_montreal	27	4×4	77.68
ibm_washington	127	4×4	81.03

Table 1: Accuracy of QSVM algorithm on the MNIST dataset, comparing different instance sizes and computers.

The results of the quantum algorithm are slightly worse than those of the classical SVM classifier. However, the differences, especially between experiments on larger instances - 6×6 and 8×8 - are relatively small, which puts the conclusion that the quantum algorithm worked effectively and performed the classification task well. Slightly larger differences can be seen on an 4×4 instance size, but in this case still the vast majority of instances in the test set were classified accurately.

Another interesting comparison is the performance of the quantum computer simulator with the actual quantum machines. For the same configuration, the results are very close to each other, which shows that the simulation process is consistent with the performance of the actual quantum computers. The fact that it was possible to successfully run the machine learning algorithm fully on a quantum processor can already be considered a promising result of the study.

In addition, a trend analogous to classical machine learning algorithms can be seen on quantum machines, where increasing the size of the instance significantly affects the quality of the results. Thus, it can be concluded that in the near future, with the growth of quantum processors, the performance of this and other quantum machine learning algorithms will increase.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_15

4 Conclusions

Quantum Support Vector Machine is a quantum algorithm incorporating quantum computing into the machine learning landscape. So far, the proposed model has been discussed theoretically and tested on relatively simple tasks, which only contain a few variables and is relatively simple to solve by state-of-the-art machine learning techniques. We have demonstrated experimentally in this paper that the algorithm works well on the more complex image classification problem on the MNIST dataset.

Admittedly, it was necessary to perform appropriate preprocessing to reduce the resolution of the original images to fit them on still limited capabilities offered by a quantum simulator and IBM quantum devices. Nevertheless, the experimental results obtained show that relatively high classification accuracy is possible. It was also possible to run instances on real quantum NISQ devices from IBM, utilising IBM flagship quantum computers – the 127-qubit *ibm_washington* and *ibmq_montreal*, receiving consistent results.

Additionally, we noticed an impact on the overall QSVM performance, as the selected quantum devices provide a specific topology and connectivity among superconducting qubits. Although recent research suggests adapting a model to the interconnection network topology of a given quantum processor [9], we decided to stay with a more dense structure of entanglement connections due to the spatial nature of the data set to be processed. Still, it naturally requires further inventions, especially in the context of new NISQ devices with denser connectivity among qubits.

Further experiments with QSVM and other quantum machine learning algorithms can be conducted to study the impact that different entanglement strategies may have on the qualitative results of the considered algorithm. Also, other ideas can be used to further improve the quality of the results, for example using different classical preprocessing algorithms to shrink dataset size, such as PCA and its variations, as well as the usage of error correction and mitigation techniques. As quantum technologies quickly evolve, quantum computers will have more and better quality qubits, which will be more densely connected soon. Consequently, we can run experiments for much larger input datasets by applying the QSVM-based approach discussed in this paper.

References

1. Al-Ogbi, S., Ashour, A., Felemban, M.: Quantum image classification on nisq devices. In: 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN). pp. 1–7 (2022). https://doi.org/10.1109/CICN56167.2022.10008259
2. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The balanced accuracy and its posterior distribution. In: 2010 20th international conference on pattern recognition. pp. 3121–3124. IEEE (2010)
3. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**, 273–297 (1995)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_15

4. Cristianini, N., Shawe-Taylor, J., Elisseeff, A., Kandola, J.: On kernel-target alignment. Advances in neural information processing systems **14** (2001)
5. Dawid, A., Arnold, J., Requena, B., Gresch, A., Płodzień, M., Donatella, K., Nicoli, K.A., Stornati, P., Koch, R., Büttner, M., et al.: Modern applications of machine learning in quantum sciences. arXiv preprint arXiv:2204.04198 (2022)
6. Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028 (2014)
7. Glick, J.R., Gujarati, T.P., Corcoles, A.D., Kim, Y., Kandala, A., Gambetta, J.M., Temme, K.: Covariant quantum kernels for data with group structure. arXiv preprint arXiv:2105.03406 (2021)
8. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 212–219 (1996)
9. Havlíček, V., Córcoles, A.D., Temme, K., Harrow, A.W., Kandala, A., Chow, J.M., Gambetta, J.M.: Supervised learning with quantum-enhanced feature spaces. Nature **567**(7747), 209–212 (2019)
10. IBM: IBM Quantum — quantum-computing.ibm.com. https://quantum-computing.ibm.com/, [Accessed 03-Feb-2023]
11. Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J.M., Gambetta, J.M.: Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. nature **549**(7671), 242–246 (2017)
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
13. Lloyd, S., Schuld, M., Ijaz, A., Izaac, J., Killoran, N.: Quantum embeddings for machine learning. arXiv preprint arXiv:2001.03622 (2020)
14. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information (2002)
15. Park, S., Park, D.K., Rhee, J.K.K.: Variational quantum approximate support vector machine with inference transfer. Scientific Reports **13**(1), 3288 (2023)
16. Preskill, J.: Quantum computing in the nisq era and beyond. Quantum **2**, 79 (2018)
17. Qiskit contributors: Qiskit: An open-source framework for quantum computing (2023). https://doi.org/10.5281/zenodo.2573505
18. Schuld, M., Killoran, N.: Quantum machine learning in feature hilbert spaces. Physical review letters **122**(4), 040504 (2019)
19. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review **41**(2), 303–332 (1999)
20. Singh, G., Kaur, M., Singh, M., Kumar, Y., et al.: Implementation of quantum support vector machine algorithm using a benchmarking dataset. Indian Journal of Pure & Applied Physics (IJPAP) **60**(5), 407–414 (2022)
21. Spall, J.C.: An overview of the simultaneous perturbation method for efficient optimization. Johns Hopkins apl technical digest **19**(4), 482–492 (1998)
22. Van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T.: Rescale, resize, and downscale — skimage v0.19.2 docs — scikit-image.org. https://scikit-image.org/docs/stable/auto_examples/transform/plot_rescale.html (2014), [Accessed 03-Feb-2023]

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_15