LIP-STUDENTS-23-27

An Introductory Exploration of Quantum Machine Learning

Gabriel Fernandes[1,a] and Pedro Carvalho[1,b]

[1] *LIP and Universidade do Minho, Braga, Portugal*

Project supervisors: Nuno Filipe Castro, Maria Gabriela Oliveira, Miguel Peixoto *January 4, 2024*

Abstract. In this paper, we explore two ML methods: Support Vector Machine (SVM) and Variational Quantum Circuits (VQC).We compare the performance of both methods using the same dataset filled with data from simulated particle collisions. For the SVM, we calculated an AUC of 0.78, indicating that the model has moderate to strong discriminatory power in distinguishing between classes. In the VQC, we implemented a variational quantum circuit and discussed its training process. Although the results of the VQC were similar to those of the SVM, the AUC was slightly worse than that of the SVM.

Keywords: High Energy Physics, Variational Quantum Classifier, Support Vector Machine, ROC , AUC

1 Introduction

This study was conducted throughout the month of July 2023, under the guidance of Professor Nuno Castro, coordinator of LIP (Laboratório de Instrumentação e Física Experimental de Partículas).

This paper aims to provide a brief introduction to the fundamental concepts on Quantum Machine Learning, where we implemented a VQC. However, recognizing the significance of Support Vector Machine (SVM) as the classical counterpart in classification tasks, we also integrated an SVM model into our research to learn some fundamental concepts in classical Machine Learning.

ML is a subfield of Artificial Intelligence that seeks to develop algorithms capable of learning patterns from data, enabling automated systems to make decisions or predictions based on that information. By analyzing datasets, ML models seek to identify hidden relationships and trends, enabling the execution of complex tasks more effectively and accurately.

An SVM model rely on finding the best separation between different data classes in a multi-dimensional space. On the other hand, a VQC model is an emerging approach that use methods of quantum computing to solve classification problems.

This analysis will be conducted based on the same data set used in the article [1].

This data set consists of a simulation of experimental results from particle collisions at CERN's accelerators [10]. From the experimental values of various physical parameters obtained in this collision, our goal is to employ both ML techniques mentioned earlier to develop an algorithm capable of learning from these values to discern, with new experimental results, the presence of new physics events or their absence. All events in the dataset are categorized with labels "background" or "signal" where a "signal" event indicates a new physical event, while "background" signifies the opposite.

[a] e-mail: a97303@alunos.uminho.pt

[b] e-mail: a100507@alunos.uminho.pt

2 Data Set

The dataset events were described by 47 distinct particle collision features such as masses, transverse momenta, angles, the number of electrons present, and even the energy lost during the collision.

To ensure the results for our models were within a reasonable timeframe, considering the inherent complexity of quantum training in the case of the VQC and the scalability challenges associated with SVM for a large number of events, we have chosen to focus on a dataset consisting of 500 events. This dataset has been evenly split into 250 signal events and 250 background events from the original dataset.

In paper [1] it was concluded that discrete features in the SBS feature selection methodology resulted in erratic performance except in cases where only one continuous feature was used. Therefore, it was found that excluding discrete variables during feature selection led to better performance for VQC circuits in a limited study of 2 features, compared to when discrete variables were included.

This indicates that the choice of input features is crucial for achieving high accuracy in quantum machine learning.

The SBS (Sequential Backward Selection) algorithm begins with the set of all features, including the discrete ones. At each iteration, it generates all possible feature subsets of size n-1 and trains a machine learning model for each of these subsets. Subsequently, the performance is evaluated, and the feature that is absent from the subset with the lowest performance metric is removed. This process is repeated until the feature subset contains k features.

In Table 6 of the referenced paper, you can find the results for the features selected by the SBS algorithm and their corresponding AUC scores on the training dataset with all discrete features removed.

In the following section, we will introduce the concept of AUC. For now, consider that a model with a higher AUC indicates greater discriminative power.

With this in mind, it's important to note that we constructed our models using only the top 2 input features, specifically the two with the highest AUC values from the mentioned table.

It is also worth noting that these events were associated with Monte Carlo "weights," corresponding to the theoretical predictions for each process at the target luminosity of 150 $/fb^{-}1$, which were taken into account in the evaluation of all the considered metrics and loss functions, making events with larger weights more influential for our model's learning process.

3 SVM results and Metrics

We won't delve into the details of the SVM code, but we will use its results to explain the metrics implemented for evaluating a ML model.

The ROC curve is a graphical representation illustrating the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) at different classification thresholds.

Mathematically, TPR, representing the proportion of positive examples correctly classified as positive, can be expressed as:

$$TPR = 1 - \frac{FN}{TP + FN}$$

Where TP (True Positives) is the count of positive examples correctly classified as positive, and FN (False Negatives) is the count of positive examples incorrectly classified as negative.

On the other hand, FPR, representing the proportion of negative examples incorrectly classified as positive, is expressed as:

$$FPR = \frac{FP}{FP + TN}$$

Where FP (False Positives) is the count of negative examples incorrectly classified as positive, and TN (True Negatives) is the count of negative examples correctly classified as negative.

In the ideal scenario, the ROC curve forms a vertical straight line from the origin to the upper-left corner, representing a TPR of 1, signifying that all positive instances are correctly classified as positive (no False Negatives). From the upper-left corner to the upper-right point, the FPR is 0, indicating that there are no False Positives. Visually, this ideal curve resembles a square with a side length of 1.

The AUC (Area Under the Curve) is the area beneath the ROC curve and is another metric that tells us how good the test or model is. In a interval between 0 and 1, a higher the AUC represents a better performance in classification. If the AUC is 0.5, the test is like tossing a coin, meaning it's not very useful. If the AUC is 1 it mean we obtain precisely a Roc curve of an perfect classification situation

We obtained an AUC of 0.78, and the following ROC curve for our SVM model, that is illustrated in Figure 1.

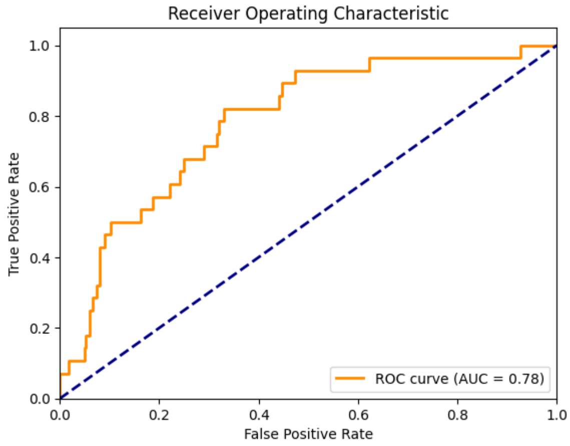

Figure 1: SVM ROC curve and AUC

4 Variational Quantum Circuits

For our initial attempt in quantum ML, we constructed a simple model of a variational quantum circuit, and in general, its training followed the following steps:

- Encoding classical data into a quantum state.
- Applying a parameterized model, measurement, and comparing it with the classification.
- Utilizing optimization techniques for parameter updates.

The quantum circuit used for building the VQC is illustrated in Figure 2. In the following subsections, we will break down the circuit into its respective steps in constructing the model as mentioned earlier.

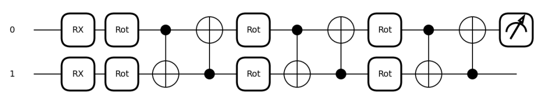

Figure 2: Complete quantum circuit implemented in our model.

Encoding Classical Data into a Quantum State

The encoding of classical data into quantum states (qubits) is a fundamental step in the utilization of quantum algorithms for information processing tasks, given that qubits serve as the fundamental building blocks of quantum circuits. One commonly adopted strategy, of many ones [2], for this encoding is referred to as Angle Embedding.

In this approach, each feature from the classical data is associated with a rotation angle around the X-axis of the Bloch sphere. To ensure uniformity, the feature values are normalized to fall within a range between π and $-\pi$.

As a matter of convention, we begin with each qubit in the $|0\rangle$ state on the Bloch sphere and subsequently apply logical gates that enable rotations along the X or Y-axis. These rotations are performed to position each feature on the Bloch sphere according to the corresponding angle.

For an individual event, since the initial quantum state is defined by a single angle value, and each angle is associated with the value of a specific feature, it can be observed that each qubit effectively represents a single feature.

In our specific case, we opted to select the same two most continuous and relevant features from a pool of 47 features within our dataset, which had been previously chosen for our Support Vector Machine (SVM) model. Consequently, our quantum circuit was designed to accommodate only two qubits.

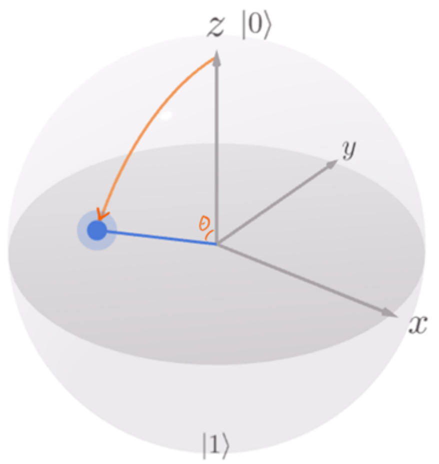

Figure 3: Representation of the Bloch Sphere and demonstration of qubit rotation about the X-axis after passing through the X-rotation logic gate, where the initial state was $|0\rangle$.

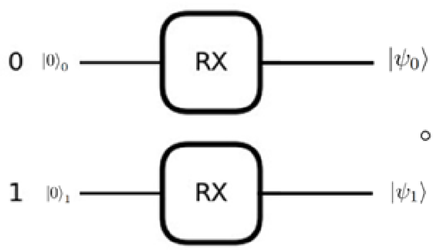

Figure 4: The angle embedding of the quantum circuit used involved two X-rotation logic gates for each qubit, representing each feature.

Apply a parameterized model

After converting classical data into quantum states the qubits are now incorporated into a quantum circuit. This circuit consists of quantum gates associated with adjustable parameters, which will be optimized during training. These parameters have the effect of repositioning the quantum states of the qubits on the Bloch sphere.

The implemented circuit is based on a series of quantum rotation gates and CNOT gates. The parameters in this context are the angles provided to the rotation gates.

The optimal variational circuit for a classifier has not yet been determined. However, we decided to follow the method described in the article [1] , adapting it for a two-qubit circuit represented in Figure 2. We can decompose this circuit into 3 igual layers represented on Figure 6 and a measurement on the 0 wire as you can see on Figure 2.

Each rotation gate represents a rotation that does not necessarily have to be a rotation about the x, y, or z-axis.

The parameters will be described by a tensor , T, of shape

$$(\text{Nlayers}, \text{Nqubits}, 3)$$

From an algebraic perspective, the rotation gate of layer i and qubit j is given by:

$$Rot(\alpha_{ij}, \beta_{ij}, \gamma_{ij}) = R_Z(\gamma_{ij}) R_Y(\beta_{ij}) R_X(\alpha_{ij})$$

As we are working with 2 qubits and 3 layers, the T tensor is algebraically described by a tensor of shape (3,2,3):

$$T = \left[\begin{bmatrix} \alpha_{11} & \beta_{11} & \gamma_{11} \\ \alpha_{12} & \beta_{12} & \gamma_{12} \\ \alpha_{13} & \beta_{13} & \gamma_{13} \end{bmatrix} \begin{bmatrix} \alpha_{21} & \beta_{21} & \gamma_{21} \\ \alpha_{22} & \beta_{22} & \gamma_{22} \\ \alpha_{23} & \beta_{23} & \gamma_{23} \end{bmatrix} \begin{bmatrix} \alpha_{31} & \beta_{31} & \gamma_{31} \\ \alpha_{32} & \beta_{32} & \gamma_{32} \\ \alpha_{33} & \beta_{33} & \gamma_{33} \end{bmatrix} \right]$$

Measuring and Comparing Classification Results

As can be observed in the circuit diagram in Figure 2, a measurement of the average value of qubit on wire 0 is performed. Let the resultant state of this wire be $|\psi\rangle = A|0\rangle + B|1\rangle$, the average value will be:

$$|A|^2 \times 0 + |B|^2 \times 1 = |B|^2 = p(T)$$

Since p is the probability of obtaining the state $|1\rangle$, which we associate with the value 1, the idea is to adjust the angles in such a way that this probability is as close to 1 as possible when the event is classified as 'signal,' and close to zero when the event is labeled as 'background.'

The details of this optimization will be discussed in the next subsection.

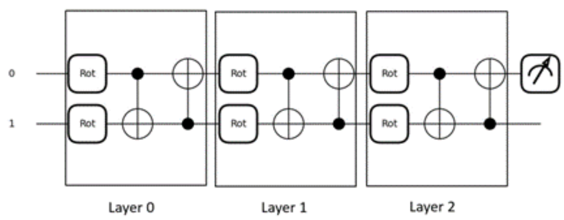

Figure 5: Variational quantum circuit after Angle Embedding consisting of 3 identical layers, as represented in Figure 6, and a measuremen

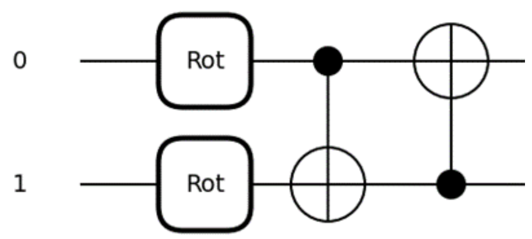

Figure 6: Fundamental layer of the circuit in Figure 5.

Parameter Optimization for T

The training begins with the initialization of a random tensor T. In each training iteration, the circuit evaluates the 500 events and returns, for each event, the average value of the qubit state measurement, which are values between zero and one, as previously mentioned, and are understood as probabilistic values p_i. With the values p_i and the true classification y_i, the Binary Cross-Entropy Loss Function (BCE) is calculated. This is a commonly used metric in binary classification problems in ML and Deep Learning. It is used to measure how well a binary classification model is making predictions in comparison to the true classes. The objective is to minimize the resulting average value $\overline{\text{BCE}}$ across all events during model training so that it makes accurate predictions close to the actual labels.

In our case, minimizing $\overline{\text{BCE}}$ is achieved using the Adam Optimizer [8], which, in each iteration, takes into account the value of $\overline{\text{BCE}}$ and the T parameters from the previous iteration and updates the T parameters in a way that reduces the value of $\overline{\text{BCE}}$. The formula for BCE is as follows:

$$\text{BCE} = -p_i \log(y_i) - (1 - p_i)\log(1 - y_i)$$

whose mean is given by:

$$\overline{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left(p_i \log(y_i) + (1 - p_i)\log(1 - y_i)\right)$$

where N is the number of events in the dataset, and y_i is the actual classification of the event.

However, we had to adjust the previous function to take into account the sampling weights w_i, which determine the impact that each event should have on training, as discussed at the beginning of this paper. The new expression for BCE will be:

$$\text{BCE}^* = w_i\left(-p_i \log(y_i) - (1 - p_i)\log(1 - y_i)\right)$$

Therefore, the impact of each BCE* on the mean, $\overline{\text{BCE}^*}$, is regulated by the value of the sampling weight w_i.

$$\overline{\text{BCE}^*} = -\frac{1}{N}\sum_{i=1}^{N} w_i\left(p_i \log(y_i) + (1 - p_i)\log(1 - y_i)\right)$$

Figure 7 graphically represents the evolution of $\overline{\text{BCE}^*}$ over the iterations.

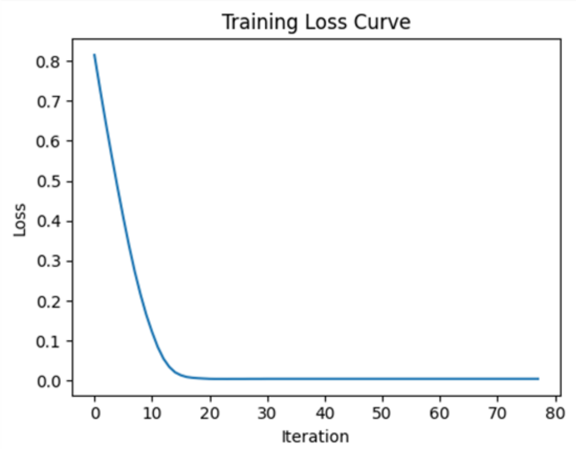

Figure 7: Graphical representation of the mean Binary Cross-Entropy Loss Function in terms of the 500 events, $\overline{\text{BCE}^*}$, over the iterations.

ROC and AUC Results

Next, we present a graphical representation of the ROC curves and the obtained AUCs from training the SVM and the VQC, on Figures 8 and 9.

The fact that the ROC curves are not smooth is due to the use of only 500 events, however, the AUC values are almost identical, with a slight advantage in favor of the SVM.

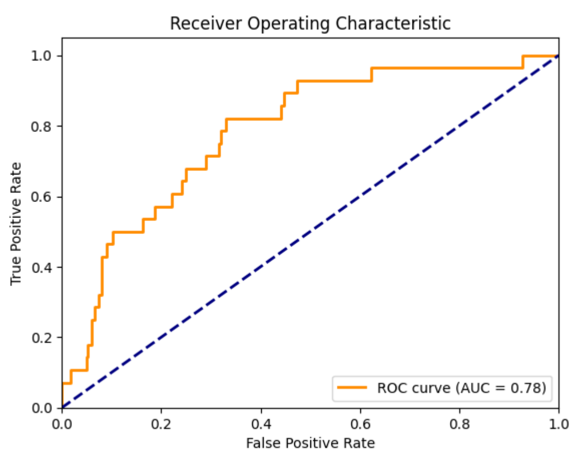

Figure 8: SVM - ROC and AUC

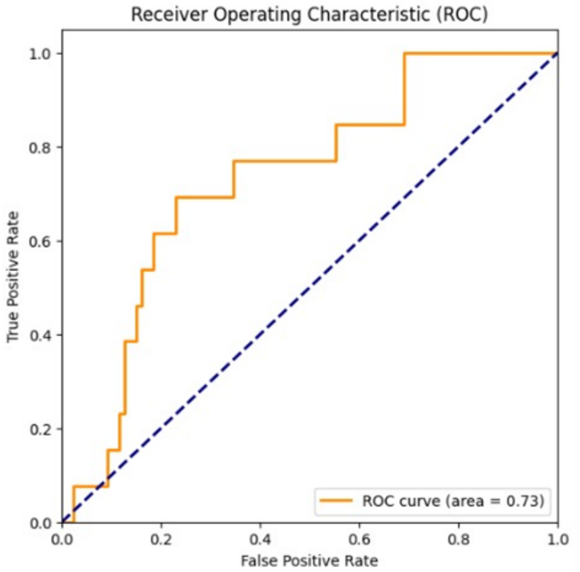

Figure 9: VQC - ROC and AUC

5 Conclusions

The primary goal of this study is to explore machine learning concepts, starting with the introduction of a classical model, SVM, and then moving on to developing a quantum model. We use this model with High-Energy Physics (HEP) datasets to demonstrate its practical application.

In future work, we will focus on investigating new architectures for the Variational Quantum Classifier (VQC) to improve its classification performance, allowing for learning from more than two features in each event. Additionally, we plan to increase the size of the training event dataset to achieve better performance and a more straightforward ROC curve.

We expect that these improvements will enhance the model's performance.

6 Acknowledgements

This internship has been very valuable in allowing us to acquire a solid understanding of the fundamentals of ML and quantum computing and we would like to express our gratitude to the entire team at LIP who guided us throughout this journey.

References

[1] Miguel Caçador Peixoto, Nuno Filipe Castro, Miguel Crispim Romão, Maria Gabriela Jordão Oliveira, e Inês Ochoa, "Fitting a Collider in a Quantum Computer: Tackling the Challenges of Quantum Machine Learning for Big Datasets," *arXiv preprint arXiv:2211.03233*, 2022.

[2] Ryan LaRose e Brian Coyle, "Robust data encodings for quantum classifiers," *Physical Review A*, vol. 102, no. 3, pág. 032420, 2020.

[3] Alessio Gianelle, Patrick Koppenburg, Donatella Lucchesi, Davide Nicotra, Eduardo Rodrigues, Lorenzo Sestini, Jacco de Vries, e Davide Zuliani, "Quantum Machine Learning for b-jet charge identification," *Journal of High Energy Physics*, vol. 2022, no. 8, págs. 1–24, 2022.

[4] Zoë Holmes, Kunal Sharma, M. Cerezo, e Patrick J. Coles, "Connecting Ansatz Expressibility to Gradient Magnitudes and Barren Plateaus," *PRX Quantum*, vol. 3, no. 1, pág. 010313, 2022.

[5] Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, e Patrick J. Coles, "Cost function dependent barren plateaus in shallow parametrized quantum circuits," *Nature Communications*, vol. 12, no. 1, pág. 1791, 2021.

[6] Diederik P. Kingma e Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[7] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M. Sohaib Alam, Guillermo Alonso-Linaje, B. Akash-Narayanan, Ali Asadi, e outros, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," *arXiv preprint arXiv:1811.04968*, 2018.

[8] Diederik P. Kingma and Jimmy Ba. *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980, 2014.

[9] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, e outros, "Scikit-learn: Machine learning in Python," *The Journal of Machine Learning Research*, vol. 12, págs. 2825–2830, 2011.

[10] Miguel Crispim Romao, Nuno F. Castro, e Rute Pedro, "Simulated pp collisions at 13 TeV with 2 leptons + 1 b jet final state and selected benchmark Beyond the Standard Model signals," Zenodo, julho de 2021. doi: 10.5281/zenodo.5126747. [Online]. Disponível em: https://doi.org/10.5281/zenodo.5126747.