Your Original and Relevant Course Project Title

Social Network Analysis for Computer Scientists — Course paper

Student name 1
student2@umail.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

Student name 2
student2@umail.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

ABSTRACT

KEYWORDS

keyword1, keyword2, keyword3, social network analysis, network science

ACM Reference Format:
Student name 1 and Student name 2. 2022. Your Original and Relevant Course Project Title: Social Network Analysis for Computer Scientists — Course paper. In *Proceedings of Social Network Analysis for Computer Scientists Course 2022 (SNACS '22).* ACM, New York, NY, USA, 2 pages.

1 INTRODUCTION

2 RELATED WORK

3 APPROACH

3.1 Variational Quantum Classifier

Variational Quantum Classifier(VQC) is a kind of parameterized quantum circuit used in classification tasks [1]. Conventional machine learning methods, such as SVM and neural networks, often take advantage of a parameterized model as classifiers. During training process, these methods use a target function or loss function to measure the difference between prediction results and ground true labels and update the parameters in the model to improve its performance. After several training epochs on adequate data, these machine learning methods can achieve great performance on prediction on unseen data as well. VQC works in a similar way, whereas applying a quantum circuit as its parameterized model. The abstract model of VQC is shown in Figure 1, including data encoding (or feature mapping) $U_{enc}(x)$, variational model $U_{train}(\theta)$, measurement, and parameter updating(usually via gradient decent). In the following subsection 3.1.1 to 3.1.4, we will introduce these processing steps in detail. And then, in subsection 3.1.5, the overall working principle of our VQC circuit will be discussed.

3.1.1 Data Encoding.

As what we have discussed above, VQC attempts to apply parameterized quantum circuit as its learning model. However, nowadays most of data is stored in classical computers which quantum circuit cannot directly access to. Hence, it's necessary to encode classical data into quantum state that can be preocessed by quantum circuit at first.

SNACS '22, Master CS, Fall 2022, Leiden, the Netherlands

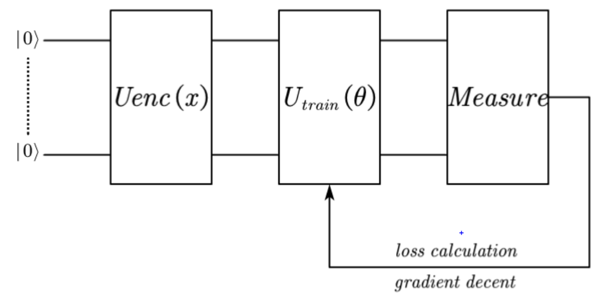

Figure 1: Abstract model of VQC: A Variational Quantum Classifier contains these processing steps: data encoding, varational model, measurement and parameter updating.

There are many basic methods to achieve this encoding preocess, such as basis encoding, amplitude encoding and angle encoding. Besides, several higher order encoding methods which combines different basic encoding methods are studied as well. In this paper, our proposal data encoding method is xxxx, one of the most popular higher order encoding methods. The specific quantum circuit $U_{enc}(x)$ is shown in Figure 2 (2-qubit case). As we can see, in this 2-qubit example, the encoding circuit is simplely composed of two Hamdamard gates, three phase gates with the components of the input data x as their parameter and two $CNOT$ gates as the entangling gate. Due to difference of the input data components of each class, the states produced by this circuit will be different as well. In addition, because of the use of entangling gate, the $CNOT$, the state on each qubit will have interaction. Besides, the initial quantum state of each qubit is $|0\rangle$ on account of the requirement of real quantum computers.

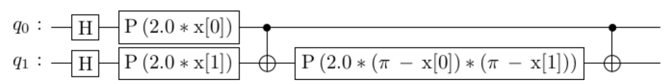

Figure 2: Data encoding circuit: Specific quantum circuit of data encoding in 2-qubit case. This circuit is composed of Hamdamard gates, phase gates, and entangling gate($CNOT$).

3.1.2 Variational Model.

Variational model is a quantum circuit with trainalbe parameters. During the training process, these parameters will be updated to improve the model's performance. There are various kinds of variational model can be applied in VQC. In this paper, the circuit of our variational model $U_{train}(\theta)$is shown in Figure 3 (2-qubit case).

This 2-qubit example is composed by four y-axis rotation gates parameterized by θ which is the parameter set to be trained, and one controlled Z gate as the entangling gate. After training, these parameterized rotation gates along with entangling gate can have the ability to convert input states of diverse classes to different output states which can be measured to make a final prediction.

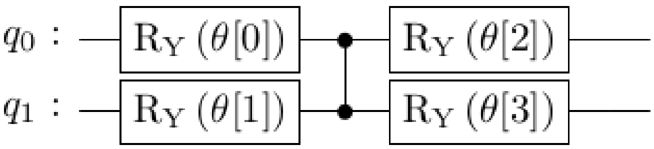

Figure 3: Variational model circuit: Specific quantum circuit of variational model in 2-qubit case. This circuit is composed of parameterized y-axis rotation gates and controlled Z gate.

3.1.3 *Measurement.*

When training most of machine learning models, we need to compare the prediction results and the labels to update parameters, which works the same way in VQC. Nevertheless, after data encoding and variational model, the results are still in quantum states. Hence, we need to extract prediction results from these quantum states. This process is accomplished by measurement. We simply measure all qubits in the circuit and run the circuit for many times. Then we can obtain a distribution of these possible results.

In binary classification tasks, a common way to extract prediction results is to converge the result states distribution into a binary distribution of odd parity states and even parity states. In other words, we consider the parity distribution as the prediction probability of two classes. For example, we run a 2-qubit circuit for 100 times and obtain the measurement result which is [00:21, 01:33, 10:28, 11:18], so the prediction probability for each class shall be [0.39, 0.61]. Therefore, the second class will be predicted as the final result.

3.1.4 *Parameter Updating.*

Parameter updating process is the most significant part of machine learning, which includes loss calculation and gradient decent. Loss calculation is the same as the conventional machine learning. When we obtain the predcition probability distribution of each class, we compare the difference between it and ground true label through a loss function. Despite popularity of cross entropy loss in classification tasks, we still choose MAE loss in our model.

In the conventional machine learning, such as neural network, gradient decent based method is famous and efficient to update parameters in the training process. However, it's hard to calculate gradient as in the conventional machine learning due to the fact that our model is based on the quantum circuit. Hence, [2] proposed a method using difference to subsitute real gradients. In detail, for each parameter θ, we run the circuit and calculate the loss value based on the prediction probability distribution results for parameter $\theta+s$ and $\theta-s$, namely $L(\theta+s)$ and $L(\theta-s)$. Then using the following formula to calculate the gradient:

$$gradient(\theta) = \frac{L(\theta+s) - L(\theta-s)}{2s}$$

where s is usually chosen as a small real positive number. Then this proximate gradient is used to update model's parameters just as the same way in the conventional machine learning.

3.1.5 *Overall Analysis.*

In this subsection, we make a overal analysis on our VQC and the circuit theoretically to illustrate why this process makes a good classifier.

According to the analysis in the previous subsections, our ansatz can be written in the mathematical form:

$$U(x,\theta) = U_{train}(\theta)U_{enc}(x)$$

Before the measurement, the following state is prepared:

$$|\psi(x,\theta)\rangle = U(x,\theta)\left|0^n\right\rangle$$

And then we measure the observable $Z^{\otimes n}$ on the above state to implement the function:

$$f_\theta(x) = \langle\psi(x,\theta)|\, Z^{\otimes n}\, |\psi(x,\theta)\rangle$$

This function is actually the expectation of the observing result distribution on the computational basis of even parity states and odd parity states, when giving even parity states value 1 (represent class 0) and odd parity states value -1 (represent class 1). To be more specific, let us use a 2-qubit example to illustrate it.

Suppose we have a general 2-qubit state $|\phi\rangle = a\,|00\rangle + b\,|01\rangle + c\,|10\rangle + d\,|11\rangle$ as $|\psi(x,\theta)\rangle$. Feed this state into the above function, and the result is actually computed as:

$$\begin{aligned} f_\theta(x) &= \langle\phi|\, Z^{\otimes n}\, |\phi\rangle \\ &= \begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \\ &= a^2 - b^2 - c^2 + d^2 \end{aligned}$$

When we observe $|\phi\rangle$ on the computational basis, the final probability to obtain odd and even parity state shall be:

$$p(even) = p(|00\rangle) + p(|11\rangle) = a^2 + d^2$$

$$p(odd) = p(|01\rangle) + p(|10\rangle) = b^2 + c^2$$

Remember we have given even parity states value 1 and odd parity states value -1 to make it a classifier, so the expectation value of this probability distribution will be:

$$E = 1 * p(even) + (-1) * p(odd) = a^2 - b^2 - c^2 + d^2$$

It's actually the same result compared to $f_\theta(x)$. However, in real cases, we couldn't

4 DATA

5 EXPERIMENTS

6 CONCLUSION

ACKNOWLEDGMENTS

REFERENCES

[1] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. 2019. Supervised learning with quantum-enhanced feature spaces. *Nature* 567, 7747 (2019), 209–212.

[2] Maria Schuld, Alex Bocharov, Krysta M Svore, and Nathan Wiebe. 2020. Circuit-centric quantum classifiers. *Physical Review A* 101, 3 (2020), 032308.