Hybrid Classical-Quantum Transfer Learning for Text Classification

Ebrahim Ardeshir-Larijani

e.a.larijani@ipm.ir

Iran University of Science and Technology

Mehdi Nasiri

Pasargad Institute for Advanced Innovative Solutions

Research Article

Keywords: Quantum Machine Learning, Natural Language Processing, Variational Quantum Computing

Posted Date: June 28th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3094921/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Quantum Machine Intelligence on March 17th, 2024. See the published version at https://doi.org/10.1007/s42484-024-00147-2.

Hybrid Classical-Quantum Transfer Learning for Text Classification

Ebrahim Ardeshir-Larijani[1,2*] and Mehdi Nasiri[2]

[1*]School of Mathematics and Computer Science, Iran University of Science and Technology, Tehran, Iran.
[2], Pasargad Institute for Advanced Innovative Solutions, Tehran, Iran.

*Corresponding author(s). E-mail(s): e.a.larijani@ipm.ir;
Contributing authors: Mehdi.nafa1373@gmail.com;

Abstract

Quantum Machine Learning (QML) is a promising field that combines the power of quantum computing with machine learning. Variational quantum circuits, where parameters of circuits are learned classically, have been widely used in many recent applications of QML. This is an instance of a hybrid quantum-classical framework, where both classical and quantum components are present. However, applying these techniques to applications involving massive data is a challenging task. One way to overcome this, is using the concept of classical-quantum transfer learning with the help of dressed quantum circuit, introduced recently, where the underlying neural architecture is pre-trained classically, but at the final steps (decision layer), a quantum circuit is used, followed by quantum measurements and post-processing to classify images with high precision. In this paper, we applied hybrid classical-quantum transfer learning to another task of massive data processing,i.e. Natural Language Processing (NLP). We show how to (binary) classify short texts (e.g., SMS) with classical-quantum transfer learning, which was originally applied to image processing only. Our quantum network was pre-trained by Bidirectional Encoder Representations from the Transformers (BERT) model, and its variational quantum circuit is fine-tuned for text processing. We evaluated the performance of our hybrid neural architecture using the Receiver Operating Characteristic (ROC) curve, which is typically used in the evaluation of classification problems. The results indicate high precision as well as lower loss function. To our knowledge, our work is the first application of quantum transfer learning to the area of NLP. Finally a comparison with a tool that uses learning but in a different way than transfer learning is presented

Keywords: Quantum Machine Learning, Natural Language Processing, Variational Quantum Computing

1 Introduction

With the current rapid development of quantum technology, a fundamental quest has begun to find out the possible applications of the available devices which are referred to as Noisy Intermediate Scale (NISQ) devices. Among different areas, Quantum Machine Learning (QML) has attracted many researchers due to the necessity of processing high dimensional big data. On one hand, the characteristics of quantum computation facilitate fast computations over superpositioned qubits, and hence, resulting in more effective QML. On the other hand, the main challenge here is to transform massive data into *large* quantum states. One way to overcome this is to use transfer learning with the help of dressed quantum circuit [1], where massive data is partly processed by a classical neural architecture and then will be fed to a relatively small parametrized (variational) quantum circuit. This is the main idea of classical-Quantum Transfer Learning (CQTL) introduced in [2]. Moreover, QTL is applied to the area of image classification [2],[3]. With the recent interest in transfer learning applied to NLP, manifested in technologies such as Generative Pre-Trained Transformer (GPT), one may ask: can QML can be leveraged to process massive textual data such as GPT? In this paper, we answer this question positively by incorporating CQTL applied to a pre-trained network, for the classical part, followed by a variational circuit embedding, called a dressed quantum circuit, for the quantum part, to the problem of (binary) classification of (short) texts. To this end, we fine-tune the original pre-training network BERT, for the specific problem of interest by learning dressed quantum circuits, so that with the help of a quantum simulator, our dataset be classified.

The contribution of this paper towards text classification of the hybrid classical-quantum model is as follows: (1) Applying the concept of QTL to the problem of classification of texts. (2) Fine tuning the Neural architecture in [4], for the problem of (short) text classification. (3) Carrying out experiments, with a hybrid classical-quantum neural architecture for text classification, using Pennylane [5] for simulation. (4) Evaluating the proposed model in terms of learning *precision* and losses.

The rest of the paper is organized as follows: In Section 2 we explain the concept of quantum transfer learning. In Section 3 we show how QTL can be applied to NLP, specifically text classification. Also, we sketch our model structure with relevant implementation details. Section 4 describes our experimental details including measuring the precision of our model. In Section 5 we review two recent works on using quantum variational methods

2 Hybrid Classical-Quantum Transfer Learning

In this section we describe the method of classical-quantum transfer learning as described in [2]. The crux of this method is called dressed quantum circuit which is a variational quantum circuit, embedded at the decision layer of a pre-trained network.

It is then followed by a classical layer for post-processing the outcome of variational quantum circuit measurements. This hybrid approach is very convenient because a quantum computer is applied only to a fairly limited number of abstract features, which is much more feasible compared to embedding much more features.

2.1 Variational Quantum Circuits

Variational quantum circuits (VQCs) are a type of quantum circuit used in quantum machine learning and optimization tasks. In a VQC, a quantum circuit is used to prepare a quantum state that depends on one or more parameters, which are then optimized to minimize or maximize a certain cost function. VQCs have been used in a wide range of applications, including quantum chemistry, machine learning, and finance. One of the earliest papers on VQCs is the 2014 paper by Peruzzo et al. [1], where they showed that a VQC could be used to simulate the ground state of a small molecule. Since then, VQCs have been used in a variety of applications, including solving optimization problems [2], training quantum neural networks [3], and performing quantum approximate optimization algorithms (QAOA) [4]. VQCs have also been used to study many-body physics [5] and to design quantum error correction codes [6]. The basic idea behind VQCs is to use a quantum circuit to prepare a parameterized quantum state. The parameters are then optimized using classical algorithms to minimize or maximize a certain cost function. In quantum chemistry, for example, the goal is to find the ground state energy of a molecule, which can be calculated using the variational principle. By using a VQC to prepare a trial wave function, one can optimize the parameters to find the ground state energy of the molecule. In machine learning, VQCs have been used to train quantum neural networks, which are a type of neural network that uses quantum circuits as their basic building blocks [3]. In this case, the goal is to optimize the parameters of the quantum circuit to minimize a cost function that measures the difference between the predicted and actual output of the neural network. VQCs have also been studied in the context of hybrid quantum-classical algorithms, where a classical computer is used to optimize the parameters of a VQC [7]. These hybrid algorithms have been used to solve classical optimization problems, such as the traveling salesman problem [8].

A variational quantum circuit of depth q is a concatenation (i.e. the output of a layer goes to the input of the next layer) of many quantum layers. A quantum layer is a unitary operation that can be physically realized by a low-depth variational circuit acting on the input state $|x\rangle$ of n_q quantum subsystems and producing the output state $|y\rangle$), corresponding to the product of numerous unitaries parametrized by different weights as defined in Equation 1.

$$\mathfrak{L} : |x\rangle \mapsto |y\rangle = U(\mathbf{w})|x\rangle \tag{1}$$

where $\mathbf{w}$ is an array of classical variational parameters. To inject classical data into a quantum network, it is necessary to encapsulate a real vector x into a quantum state $|x\rangle$. This is also possible with a variational embedding layer based on x and applied

to a reference state (e.g., the vacuum or ground state), shown in Equation

$$\epsilon : x \mapsto |x\rangle = E(x)|0\rangle \tag{2}$$

In order to extract a classical output vector y from the quantum circuit, the expectation values of n_q local observables $y = [y1, y2, ...y_{n_q}]$ are measured. This procedure, which maps a quantum state to a classical vector, is described as a measurement layer.

$$M|x\rangle \mapsto y = \langle x|\hat{y}|x\rangle \tag{3}$$

Finally, the complete quantum network is obtained by concatenation of the above embedding, variational and measurement layer. For more details see [2].

2.2 Transfer Learning

Transfer learning is a type of machine learning technique that involves reusing a pre-trained model on a new task or domain, rather than training a model from scratch. The idea behind transfer learning is to leverage the knowledge learned by a model on a large, general dataset to improve its performance on a smaller, more specific dataset. In transfer learning, the pre-trained model is usually trained on a large dataset for a related task, such as image classification or natural language processing. The model learns to identify patterns and features in the data that are useful for the task it was trained on. When a new task or domain is encountered, the pre-trained model is fine-tuned on a smaller dataset that is specific to the new task or domain.

Fine-tuning a pre-trained model involves training the model on the new dataset while keeping the pre-trained weights fixed for some of the layers (typically the earlier layers), and adjusting the weights of other layers (typically the earlier layers) to adapt the model to the new task or dataset. This allows the model to learn the specific patterns and features that are relevant to the new task while retaining the general knowledge learned from the pre-trained model. There are several benefits of using transfer learning: Speed: Transfer learning can significantly reduce the amount of time and resources required to train a new model from scratch since the pre-trained model has already learned useful features and patterns. Data efficiency: Transfer learning can improve the performance of a model on a new task, even with a smaller dataset, since the pre-trained model has already learned general knowledge that can be applied to the new task. Generalization: Transfer learning can improve the generalization of a model to new data since the pre-trained model has learned features and patterns that are common to many different tasks and domains. There are different types of transfer learning, including feature extraction, fine-tuning, and multi-task learning. Feature extraction involves using the pre-trained model to extract features from the data and using these features to train a new model on the new task. Fine-tuning involves training the pre-trained model on the new dataset while adjusting its weights to better fit the new task. Multi-task learning involves training the pre-trained model on multiple related tasks simultaneously.

Therefore, following the works [6–8], transfer learning structure can be summarized as follows:

1. Take a network A that has been pre-trained on a dataset DA and for a given task TA.
2. Freeze all the layers of the A model and attach a few trainable neural networks B at the end of the pre-trained network A
3. Train the final block B with a new dataset D_B and for a new task of interest TB

2.3 Transfer Learning with Dressed Quantum Circuits

Classical to quantum transfer learning is a technique where a pre-trained classical neural network is used to initialize the parameters of a quantum circuit for a related task. The pre-trained classical neural network has already learned to extract features from classical data, and these features can be used as input to the quantum circuit. The basic idea of classical quantum transfer learning is to leverage the pre-trained knowledge from the classical neural network to improve the training of the quantum circuit for a related task. By initializing the parameters of the quantum circuit using the pre-trained classical network, the quantum circuit can start from a better initial configuration, which can lead to faster convergence and better performance[2].

"Dressed quantum circuits"(DQS) [2] is an example of classical to quantum transfer learning, where a hybrid quantum-classical architecture combines quantum circuits with classical neural networks. In a dressed quantum circuit, a classical neural network is used to control the parameters of a quantum circuit, also known as "dressing" the circuit. This can be used to improve the performance of quantum circuits by optimizing the input parameters and reducing the number of required quantum gates. The goal of using dressed quantum circuits is to achieve a better balance between the classical and quantum components of the architecture, which can lead to better performance on certain tasks. Additionally, using a classical neural network to optimize the quantum circuit can make the circuit more robust to errors, as the neural network can compensate for imperfections in the quantum hardware.

Overall, dressed quantum circuits are a promising approach to hybrid quantum-classical computing, for example, in classical to quantum transfer learning, we need to link classical neural networks to quantum variational circuits. As in general the scale of the classical and quantum networks might be substantially different, it is advantageous to utilize a more flexible model of quantum circuits.

The structure of DQS consists of units for pre-processing and post-processing of the input and output data which are installed as classical layers at the beginning and at the end of the quantum network, getting a quantum architecture that is called "dressed":

$$\tilde{Q} = L_{n_q \mapsto n_{out}} \circ Q \circ L_{n_{in}} \mapsto n_q \tag{4}$$

Where in Equation 4, $L_{n_{in}} \mapsto n_{in}$ is the classical linear layer which its output is fed to the variational circuit Q with n_q qubits. The $L_{n_q} \mapsto n_{out}$ is a classical linear layer that is responsible for post-processing the measurement of circuit Q.

In contrast to a complex hybrid network in which the computing is split between cooperative classical and quantum processors, the major computation in this instance is conducted by the quantum circuit, while the classical layers are primarily responsible for data embedding and reading. In[9] and [10], a similar hybrid model was applied to

a generative quantum Helmholtz machine and a variational circuit, respectively. To summarize, the benefits of using dressed quantum circuit is as follows.

1. Two classical linear layers are trained to optimally execute the embedding of input data and post-processing of measurement outcomes;
2. The number of input and output variables is independent of the number of subsystems, enabling flexible connectivity to other conventional or quantum networks.

More details can be found in [2].

3 Problem Definition and Neural Network Architecture

In this section, we define the architecture and model that is used in our text classification problem. Since the original application of classical-quantum transfer learning was on image processing, for natural language processing we had to change the configuration of the variational quantum circuit. These modifications are applied according to the output of the pre-trained network. In the following first the pre-training model is called BERT (Bidirectional Encoder Representations from Transformers), is explained and then we explain how we applied classical-quantum transfer learning to the problem of text classification.

In the application of deep learning in computer vision, it is preferable to use a pre-trained model as a starting point to address a problem, rather than developing a model from scratch. For example, the dataset ImageNet is widely used in pre-training of transfer learning in computer vision in order to do different tasks. Moreover, transfer learning has been applied to the NLP, for the same reason. A recent and rather controversial example is Open AI chat GPT which is a generative pre-trained transformer model on NLP[4].

The majority of problems in natural language processing, such as text categorization, language modeling, and machine translation, are sequence modeling tasks. Thus, the demand for NLP transfer learning was at an all-time high. Google launched the transformer in 2018[4], which proved to be a landmark achievement in NLP.

Subsequently, several transformer-based models for various NLP tasks began to emerge. In particular, using transformer-based models has the following advantages:

1. These models do not analyze an input sequence token by token; instead, they accept the complete sequence at once, which is a significant advance over RNN-based models since the model can now be accelerated by GPUs.
2. These models do not need labeled data for pre-training. To train a transformer-based model, it is sufficient to give a massive quantity of unlabeled text data. This trained model may be used for other NLP tasks such as text classification, named entity identification, text generation, etc. In NLP, transfer learning operates in this manner.

Now we explain BERT and learn how to utilize a pre-trained BERT model to conduct text categorization using classical to quantum transfer learning.

3.1 Pre-training Language Models

For some language processing tasks such as language inference and paraphrasing (sentence-level), and also question answering and name recognition (token-level), pre-training models have been shown better performance. [4]. Among approaches for implementing a pre-training model is fine-tuning in which a small number of task-specific parameters are involved. In this model, the main task is fine-tuning the pre-trained model. However, using standard language models which are unidirectional is a limitation for fine-tuned pre-trained models that deal with tasks that are sensitive to the both left and right context of given input tokens. The BERT model [4] overcomes this limitation by using a Masked Language Model (MLM) which considers both contexts of tokens and hence offers a bidirectional method. In addition to MLM, BERT is also pre-trained to do next-sentence prediction (NSP). During NSP, the model is trained to predict whether two input sentences are consecutive in a document or not. This helps the model learn to understand the relationships between different sentences in a document. Overall, the BERT architecture is a powerful tool for natural language processing tasks because it allows the model to capture complex relationships between different parts of a sentence or document.

3.2 Problem Definition:

Now we explain the problem of text classification which we address in this paper: We have a collection of Short Message Service (SMS) messages. Each message has a maximum of 175 characters. It contains an alphabet and special characters. Our task is to build a system that would automatically detect whether a message in our dataset which consists of 5572 messages, is spam or not. Hence this is an instance of binary classification, though extending to multi-class cases follows the same principle and very similar network architecture. As a part, the BERT model is fined tuned for text classification such that classical-quantum transfer learning via dressed quantum circuit become possible. The overall network design for our problem is illustrated in Figure 1.

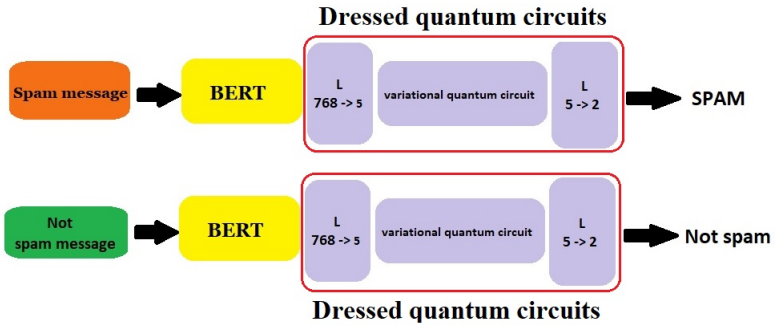


Fig. 1: Problem Definition

3.3 Hybrid Model Structure

In this section, we explain the steps that our model takes in order to classify text, based on a pre-trained network.

1. **Data Loading**: The dataset (spam and non-spam messages) is loaded in a two-dimensional table by Pandas data frame. Moreover, by using the scikit-learn library, the dataset is split into training, validation, and test sets automatically. Next, we load the BERT-base pre-trained model and tokenizer using the transformers library. The tokenizer is used to convert text messages into numerical sequences that can be processed by the BERT model.
2. **Parameter Freezing**: all the parameters in the BERT model will be frozen. This is done to prevent the gradients from flowing back through the BERT model during training.
3. **Classical Pre-processing**: a (classical) pre-processing (linear) layer is used to perform on the BERT output to prepare for the next layer which is the quantum variational circuit, as shown in Figure 2

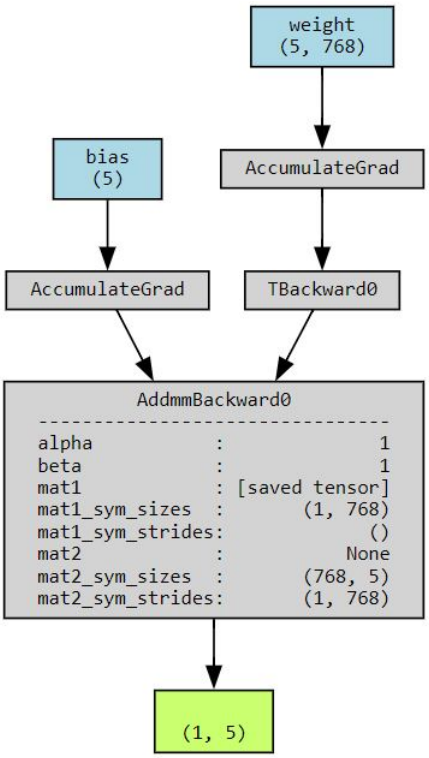


Fig. 2: Pre processing layer. 1- Accumulate Grad: is a flag that indicates whether or not the gradients should be accumulated or overwritten during the backward pass. 2- TBackward0: is a function that is called during the backward pass of the computation graph. 3- AddmmBackward0: is a function that is called during the backward pass of the computation graph. 4- bias: is a parameter that represents the bias of each neuron in the output layer. 5- weight: The weight term is a parameter that represents the weight matrix of size (768, 5) to transform the input into the output.

4. **Variational quantum circuit**: The structure of our variational quantum circuit consists of *embedding*, where all qubits are first initialized in equal superposition using Hadamard gates, then they are rotated according to the input parameters for local embedding. Next, in the *variational* part of the circuits, A sequence of trainable rotation layers and constant entangling layers is applied. Finally, For each qubit, the local expectation value of the Z operator is measured that produces a classical output vector, for further post-processing. For the simulation of this circuit, the Pennylane library is used. These are shown in Figure 3

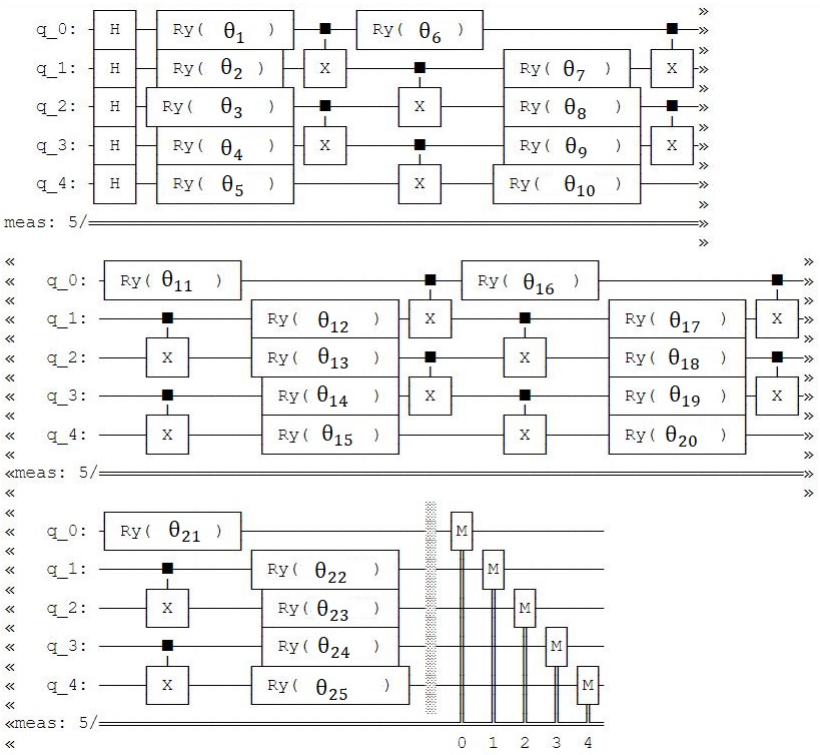


Fig. 3: The variational quantum circuit for transfer learning. The number of qubits is equal to 5 and the depth of the circuit is 4

5. **Post processing**: A classical linear layer to perform post-processing of the measurement results of the quantum variational circuit that finally classifies the text. The architecture of this layer is shown in Figure 4.

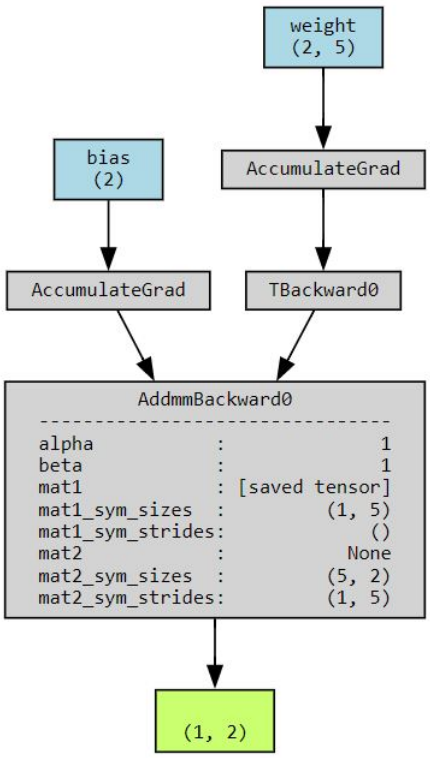


Fig. 4: Post processing layer architecture 1- Accumulate Grad: The Accumulate Grad is a flag that indicates whether or not the gradients should be accumulated or overwritten during the backward pass. 2- TBackward0: is a function that is called during the backward pass of the computation graph. It is responsible for computing the gradients of the parameters with respect to the input. The gradients are stored in the .grad attribute of the parameters. 3- AddmmBackward0: is a function that is called during the backward pass of the computation graph. It is responsible for computing the gradients of the parameters with respect to the input. 4- bias: is a parameter that represents the bias of each neuron in the output layer. 5- weight: is a parameter that represents the weight matrix of size (5, 2) that is used to transform the input into the output.

4 Experiments and Performance Evaluation

In this section first, we report on implementing our hybrid algorithm on the data set we mentioned in Section 3.2. Then we explain the tool LambeQ[11], where we compare our approach with.

4.1 Performance Evaluation

Our model is trained with ten epochs on the dataset described in Section 3.2 "The training and validation loss, as illustrated in Figure 5, exhibit a consistent decrease in loss values over the course of multiple epochs. At the beginning of training, the training loss was 0.7, and the validation loss was 0.6. However, after just 10 epochs, the training loss decreased to 0.17, while the validation loss reached 0.135. These decreasing loss values indicate that the model is effectively learning from the training data and gradually converging towards a lower loss. Furthermore, the close tracking of the validation loss with the training loss suggests that the model generalizes well to unseen data. This alignment between the two loss curves demonstrates that the model's performance on the validation set mirrors its performance on the training data, without significant divergence. Such behavior indicates that the model is robust and can successfully generalize to new, unseen examples. Overall, these findings highlight the efficacy of our model in capturing the underlying patterns within the training data and its ability to generalize well to unseen instances, as evidenced by the decreasing training and validation losses. These results provide confidence in the reliability and effectiveness of our model's performance.

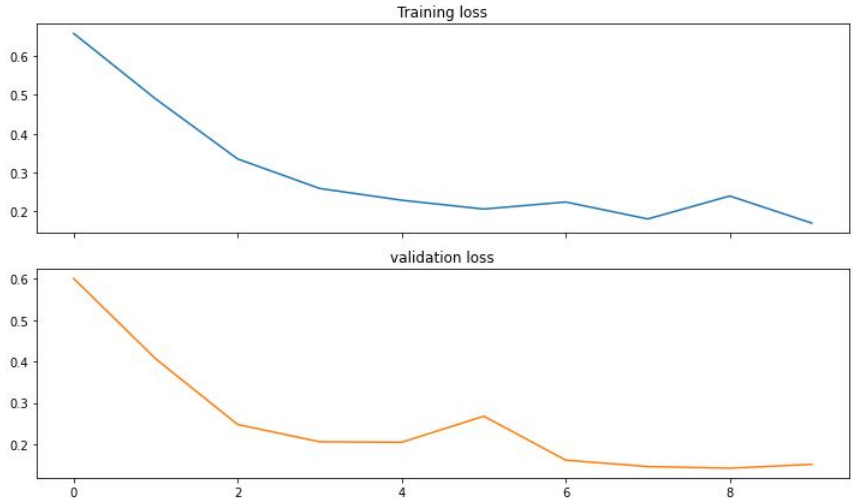


Fig. 5: Training and Validation Loss Curves over Epochs

We select *Evaluation metrics, confusion matrix, and Receiver Operating Characteristic (ROC) curve*to evaluate our model. The evaluation was conducted on a test dataset consisting of 836 samples.

The evaluation metric results are summarized in Figure 6: Precision measures the accuracy of positive predictions, while recall quantifies the proportion of actual positive instances correctly identified by the model. F1-score represents the harmonic mean of precision and recall, providing a balanced evaluation measure. For Class 0,

our model achieved a precision of 0.99, a recall of 0.98, and an F1-score of 0.98. This indicates high accuracy in predicting instances belonging to Class 0. Class 1 exhibited a precision of 0.87, recall of 0.93, and an F1-score of 0.90, suggesting a relatively lower precision but a good ability to capture instances of Class 1. Considering the entire dataset, the micro-average precision, recall, and F1-score were calculated to be 0.97, indicating a high overall performance. The macro-average scores, which consider an equal contribution from each class, resulted in a precision of 0.93, a recall of 0.95, and an F1 score of 0.94. Finally, the weighted average scores, which account for class imbalance, yielded a precision, recall, and F1-score of 0.97.

	precision	recall	f1-score	support
0	0.99	0.98	0.98	724
1	0.87	0.93	0.90	112
micro avg	0.97	0.97	0.97	836
macro avg	0.93	0.95	0.94	836
weighted avg	0.97	0.97	0.97	836

Fig. 6: Evaluation Metric

The confusion matrix, depicted in Figure 7, displays the number of samples correctly and incorrectly classified for each class. in our test dataset, 709 samples belonging to Class 0 were correctly classified as 0, while 15 samples belonging to Class 0 were incorrectly classified as 1. Similarly, 8 samples from Class 1 were misclassified as 0, and 104 samples from Class 1 were correctly classified as 1. The confusion matrix provides insights into the model's performance for each class. It indicates that our model successfully identifies the majority of samples from both Class 0 and Class 1, with a relatively small number of misclassifications.

col_0	0	1
row_0		
0	709	15
1	8	104

Fig. 7: Confusion Matrix

ROC curve metric is a widely used assessing the performance of binary classification models. It plots the true positive rate (TPR) against the false positive rate (FPR) at various classification thresholds. The TPR is defined as the proportion of positive instances that are correctly identified as positive (also called sensitivity or recall). The FPR is defined as the proportion of negative instances that are incorrectly identified as positive (also called the fall-out). To construct the ROC curve, the classification threshold of the model is varied from 0 to 1, and the TPR and FPR are computed at each threshold. The resulting curve is a plot of the TPR vs FPR, and it represents

the trade-off between the sensitivity and specificity of the model at different threshold values. The area under the ROC curve (AUC-ROC) is often used as a summary statistic to evaluate the overall performance of the model.

A perfect classifier would have an AUC-ROC of 1.0, while a random classifier would have an AUC-ROC of 0.5. The closer the AUC-ROC is to 1.0, the better the model's performance is. So, the ROC curve and AUC-ROC are useful tools for evaluating the performance of binary classification models, especially when the data is imbalanced, meaning one class is more prevalent than the other. In our model, the area under the ROC curve is 0.95, which suggests that our model performs well with high precision. This is depicted in Figure 8.

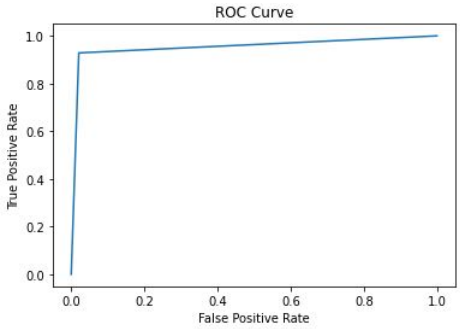


Fig. 8: Performance Evaluation of our Model with ROC Curve

5 Related works

In this section, we review recent works on NLP using hybrid quantum-classical machine learning.

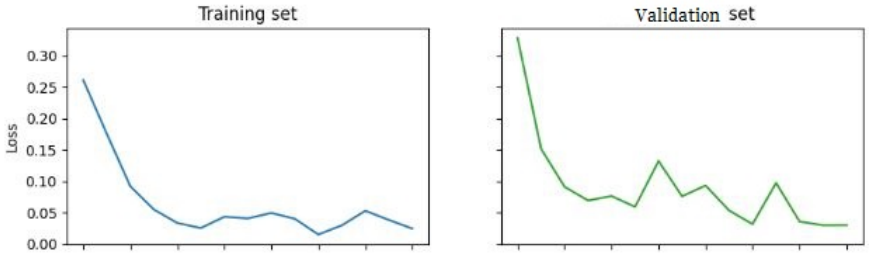
Lambeq [11] is a Python library for natural language processing, based on the compositional model DiscoCat [12]. This model can capture the compositional nature of natural language with the help of category theory. The tool implements a pipeline for converting sentences to *string diagrams*, *tensor networks*, and *and quantum circuits*. The latter is then passed to the Pennylane package which we also use for the simulation of quantum circuits.

A crucial step here is to assign quantum states to the words of a given text. This is called *Ansatz* i.e. A map that determines choices such as the number of qubits that every wire of a string diagram is associated with and the concrete parameterized quantum states that correspond to each word. As a special case, tensor ansatz describes how large tensors are represented as matrix product states. After the ansatz phase, the Pennylane is used for constructing the correspondent variational circuit for the later training phase.

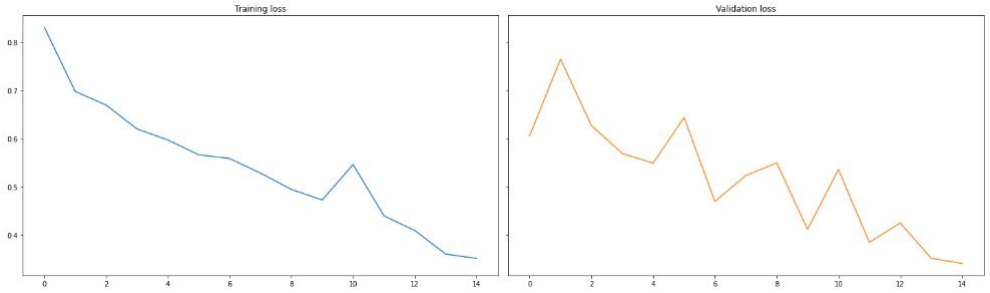
For the sake of comparison, we run our model on the dataset which Lambeq used for text classification [11] which a given sentence is about cooking or computing. The dataset consists of 130 sentences, generated with simple context-free grammar. Both

models were trained for 15 epochs with a batch size of 10, and their performance was evaluated using various metrics. The results demonstrate that the models exhibit almost similar performance on the structure of *small and simple* sentences.

Figure 9 illustrates the training and validation curves of both models. The curves demonstrate the progress of the models' performance over the training epochs. Both models show consistent improvement, indicating effective learning and convergence.



(a) Training and Validation Loss Curves over Epochs of Lambeq



(b) Training and Validation Loss Curves over Epochs of our pre-trained Model

Fig. 9: comparison of training and validation Losses of the LambeQ with our Hybrid model on small text

Figure 10 showcases the evaluation metrics of both models. Precision, recall, and F1-score were calculated to assess the models' classification performance. The results indicate that both models achieved comparable performance, with scores close to each other.

	precision	recall	f1-score	support
0	0.88	1.00	0.94	15
1	1.00	0.87	0.93	15
accuracy			0.93	30
macro avg	0.94	0.93	0.93	30
weighted avg	0.94	0.93	0.93	30

(a) Evaluation metrics of Lambeq

	precision	recall	f1-score	support
0	1.00	0.93	0.97	15
1	0.94	1.00	0.97	15
micro avg	0.97	0.97	0.97	30
macro avg	0.97	0.97	0.97	30
weighted avg	0.97	0.97	0.97	30

(b) Evaluation metrics of our pre-trained Model

Fig. 10: comparison of evaluation metrics of the LambeQ with our Hybrid model on small text

The confusion matrix in Figure 11 presents the performance of both models in terms of correctly and incorrectly classified instances for each category (cooking and computing). The results demonstrate that both models exhibit an almost similar ability to distinguish between the two categories, with a relatively small number of misclassifications.

col_0	0	1
row_0		
0	15	0
1	2	13

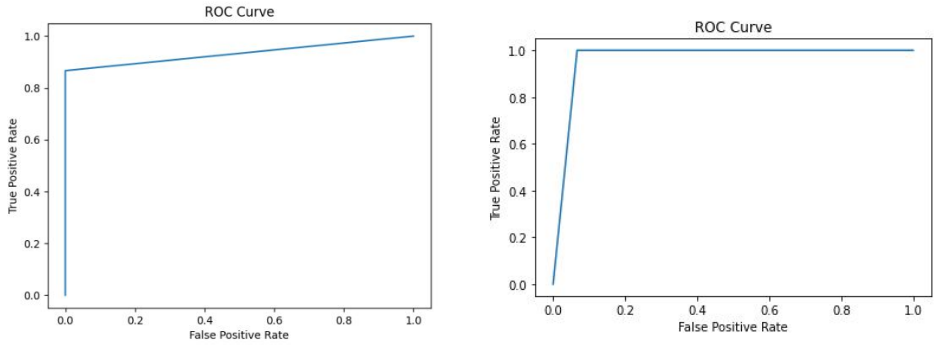
(a) Confusion matrix of Lambeq

col_0	0	1
row_0		
0	14	1
1	0	15

(b) Confusion matrix of our pre-trained Model

Fig. 11: comparison of confusion matrix of the LambeQ with our Hybrid model on small text

Figure 12 showcases the ROC curve, which provides insights into the models' performance across various thresholds. The curve demonstrates the trade-off between a true positive rate (sensitivity) and a false positive rate (1 - specificity). The proximity of the curve to the top-left corner indicates a higher model performance.



(a) ROC curve of Lambeq

(b) ROC curve of our pre-trained Model

Fig. 12: comparison of ROC curve of the LambeQ with our Hybrid model on small text

The emergence of scalability challenges becomes apparent when dealing with extensive and intricate data sets. As an illustration, the utilization of Lambeq specifically requires an immense memory capacity, estimated at a staggering 65 Petabytes, for processing our dataset in Section 3.2. This noteworthy finding implies that our proposed approach, incorporating a pre-trained hybrid network, exhibits commendable performance when confronted with colossal textual data, surpassing the capabilities observed in Lambeq's case studies.

Complex values Pre-trained network [13] is another recent approach to leverage the NLP tasks such as semantics analysis. It is worth noting that the mentioned approach came to our attention when this paper writing was completed. Unfortunately, the work in [13] is not reproducible currently by presented codes for the sake of comparison. Nonetheless, here the idea is to design a a pre-trained network with complex value, described earlier in [14, 15]. Thus [13] introduces a quantum version of BERT. However, at this point, it is not clear how complex valued BERT performs better than original data embedding of quantum transfer learning of [2], though we conjecture this might be the case. Finally, although quantum BERT might have different applications, quantum transfer learning is currently applied to classification problems.

Remark 1. *To answer why we get good result by using hybrid quantum-classical network, we argue that our classification problem which consists of classically mapping features and kernelization using variational circuit, followed by measurement and post-processing, can be viewed in the line of Abramsky and Hardy logical bell inequality [16]. This is because in variational quantum circuit, after data embedding, we end up with highly entangled quantum state. In our architecture after measurement, the linear layer learns that non-classical correlation (produced by measuring entangled quantum state) by updating probabilities (weights) in neural networks differently than classical ones, and because of such correlations the decision on class memberships of the input*

has higher precision, which is reflected in fewer losses (better loss function) in our experiments.

In this work, a text classification method based on the quantum transfer learning method is presented. Experimental studies using quantum simulation suggest that this method is both efficient and precise in the binary classification of input text. This view is further supported by a comparison with the Lambeq which operates differently. The two main factors that we believe help in the performance of our model are (1) an LLM that has a pre-trained network for later transfer learning (2) the contextuality of quantum mechanics, realized by a quantum variational circuit operating on an entangled quantum state. For future work, first, we want to modify our model with other types of neural networks, namely CNN and GAN. Second, we aim to do different tasks such as sentiment analysis or text-to-image generation. Finally, defining a notion of quantum fairness for the hybrid classical-quantum model, based on variational circuits would be very interesting.

Supplementary information. The codes for experimentation reported in this paper is available at the following link:
https://github.com/mehdinasiri1373/Classical-to-quantum-transfer-learning-for-Natural-Language-Processing-NLP-.git

Note. As already we mentioned in Section 5, at the time of completion of this paper the work in [13] came to our attention. However, we have done our work independently at the same time and the codes used in this work is openly available.

Declarations

- **Conflict of interest**
  There is no conflict of interests regarding this work.
- **Authors' contributions**
  Both authors contributed in technical parts and writing of this work. The second author has carried out the experiments.
- Funding
  There is no direct funding for this work. %item Consent for publication
- **Availability of data and materials**
  All materials including codes for experimentation and data sets and relevant explanations are available at:
  https://github.com/mehdinasiri1373/Classical-to-quantum-transfer-learning-for-Natural-Language-Processing-NLP-.git

References

[1] Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering **22**(10), 1345–1359 (2010) https://doi.org/10.1109/TKDE.2009.191

[2] Mari, A., Bromley, T.R., Izaac, J., Schuld, M., Killoran, N.: Transfer learning in hybrid classical-quantum neural networks. Quantum **4**, 340 (2020) https://doi.org/10.22331/q-2020-10-09-340

[3] Azevedo, V., Silva, C., Dutra, I.: Quantum transfer learning for breast cancer detection. Quantum Machine Intelligence **4** (2022) https://doi.org/10.1007/s42484-022-00062-4

[4] Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, ??? (2019)

[5] Bergholm, V., Izaac, J.A., Schuld, M., Gogolin, C., Killoran, N.: Pennylane: Automatic differentiation of hybrid quantum-classical computations. CoRR **abs/1811.04968** (2018) 1811.04968

[6] Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering **22**(10), 1345–1359 (2010) https://doi.org/10.1109/TKDE.2009.191

[7] Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y.: Self-taught learning: Transfer learning from unlabeled data. In: Proceedings of the 24th International Conference on Machine Learning. ICML '07, pp. 759–766. Association for Computing Machinery, New York, NY, USA (2007). https://doi.org/10.1145/1273496.1273592 . https://doi.org/10.1145/1273496.1273592

[8] Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. NIPS'14, pp. 3320–3328. MIT Press, Cambridge, MA, USA (2014)

[9] Adhikary, S., Dangwal, S., Bhowmik, D.: Supervised learning with a quantum classifier using multi-level systems. Quantum Information Processing **19**(3), 89 (2020) https://doi.org/10.1007/s11128-020-2587-9 . Accessed 2023-05-31

[10] Benedetti, M., Realpe-Gómez, J., Perdomo-Ortiz, A.: Quantum-assisted Helmholtz machines: A quantum–classical deep learning framework for industrial datasets in near-term devices. Quantum Science and Technology **3**(3), 034007 (2018) https://doi.org/10.1088/2058-9565/aabd98 . Accessed 2023-05-31

[11] Kartsaklis, D., Fan, I., Yeung, R., Pearson, A., Lorenz, R., Toumi, A., Felice, G., Meichanetzidis, K., Clark, S., Coecke, B.: lambeq: An Efficient High-Level Python Library for Quantum NLP. arXiv preprint arXiv:2110.04236 (2021)

[12] Lorenz, R., Pearson, A., Meichanetzidis, K., Kartsaklis, D., Coecke, B.: QNLP in practice: Running compositional models of meaning on a quantum computer. J. Artif. Intell. Res. **76**, 1305–1342 (2023)

[13] Li, Q., Wang, B., Zhu, Y., Lioma, C., Liu, Q.: Adapting pre-trained language models for quantum natural language processing. CoRR **abs/2302.13812** (2023) https://doi.org/10.48550/arXiv.2302.13812 2302.13812

[14] Georgiou, G.M., Koutsougeras, C.: Complex domain backpropagation. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing **39**(5), 330–334 (1992) https://doi.org/10.1109/82.142037

[15] Scardapane, S., Van Vaerenbergh, S., Hussain, A., Uncini, A.: Complex-valued neural networks with nonparametric activation functions. IEEE Transactions on Emerging Topics in Computational Intelligence **PP** (2018) https://doi.org/10.1109/TETCI.2018.2872600

[16] Abramsky, S., Hardy, L.: Logical bell inequalities. Phys. Rev. A **85**, 062114 (2012) https://doi.org/10.1103/PhysRevA.85.062114