Characterizing Quantum Classifier Utility in Natural Language Processing Workflows

Kathleen Hamilton*, Mayanka Chandra Shekar, John Gounley
Computational Science and Engineering Division,
Oak Ridge National Laboratory
Oak Ridge, TN 37830 USA
Email: *hamiltonke@ornl.gov

Dhanvi Bharadwaj
Department of Physics
University of Wisconsin-Madison
Madison, WI 53706 USA

Prasanna Date
Computational Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37830 USA

Eduardo Antonio Coello Pérez, In-Saeng Suh, Georgia Tourassi
National Center for Computational Sciences
Oak Ridge National Laboratory
Oak Ridge, TN 37830 USA

***Abstract*—Quantum Natural Language Processing (QNLP) develops natural language processing (NLP) models for deployment on quantum computers. We explore feature and data prototype selection techniques to address challenges posed by encoding high dimensional features. Our study builds quantum circuit classifiers that includes classical feature pre-processing, quantum embedding and quantum model training. The quantum models are built on 4 or 6 qubits and the quantum neural network (QNN) uses the established bricklayer design. We compare the dependence of model performance (in terms of accuracy and F1 scores) on feature length, embedding gates and parameterized unitary design. We compare the performance of quantum machine learning models to classical convolution neural network model (CNN) on binary and multi-class classification tasks using two datasets of synthetic features and labels. The first is the ECP-CANDLE P3B3 dataset a corpus of synthetically generated cancer pathology reports. The second dataset is extracted from well-known benchmark dataset (MADELON) — features are generated with a combination of informative, repeated and uninformative features. Both datasets are used for binary classification and multi-class classification with 3 classes. We observe robust, accurate performance from all models on the binary classification tasks, but multiclass classification is a challenge for the quantum models–there is a notable decrease in accuracy when using 3 classes. Overall the performance is comparable in terms of recall and accuracy between QNNs and CNNs, even with large datasets. These results provide a point of comparison between quantum and classical models on real-world datasets.**

***Index Terms*—quantum natural language processing, quantum neural networks, quantum machine learning**

Introduction QNLP aims to develop models that can tackle complex linguistic problems on quantum computers [1]. Quantum models can explore exponentially larger solution spaces, which can potentially lead to more accurate results in a shorter amount of time [2], [3]. High-dimensional features require models that can use the increased number of qubits that can be utilized, and the complexity of a quantum circuit to efficiently encode and classify data. In this work we incorporate data-reuploading, dense angle encodings into the bricklayer circuit ansatz.

Datasets In this work we use binary and multi-class classification of synthetic datasets to evaluate the utility of quantum classifiers. If a dataset contains unbalanced classes, then individual training samples x_i can have an associated weight c_i that gives the relative importance of correctly labeling the sample – this addresses the effects of class imbalance by making the minority class have a larger impact on the overall loss function using relative weights determined using heuristic weighting in `scikit-learn` [4].

The first dataset utilized for this project is the MADELON dataset, which serves as a valuable benchmark for evaluating the efficacy of quantum machine learning algorithms in handling multi-dimensional and highly non-linear datasets. It consists of data points clustered on the vertices of an *n*-dimensional hypercube. Each data point has 20 total features of which *n* are informative, 2 are linear combinations of the informative features, and the remaining are uninformative. The data does not have attribute information to avoid biasing feature selection and 10% of the samples are randomly labeled.[1]

The second dataset employed for this study consists of a corpus of synthetically generated cancer pathology reports (CPRs) [2]. These clinical text documents describe the analysis of a tumor biopsy and are labeled for four cancer phenotyping tasks. Automating CPR classification with deep learning is important for achieving near-real-time cancer surveillance [7]. The P3B3 dataset includes four information extraction tasks namely site, laterality, histology and grade and we use "site" as the task for this study.

Both datasets use classical mutual information to reduce the number of features passed to a QNN. The MADELON data used Mutual Information (MI) feature selection to extract the top 2 and 4 features from the original 20-feature dataset. This is a non-parametric method based on entropy estimation from k-nearest neighbors (k-NN) distances. QNNs were trained using these extracted features and compared with the full 20-feature dataset to understand the impact of non-informative

[1]Dataset generated using functionality available in `scikit-learn` adapted from methods in [5].

[2]Data set details are provided in [6] and is available at https://ftp.mcs.anl.gov/pub/candle/public/benchmarks/Pilot3.

features on QNN performance. The P3B3 dataset includes an enumerated and padded token list of sequence length 1500 for every CPRs. The dataset is pre-processed to reduce the sequence length using normalized pointwise MI calculated between a token and a *site* label. This is used to rank the tokens based on the probability of occurrence of the given token by the total number of training documents [8], [9], and is used to filter the important tokens in each pathology report based on any given document length, reducing the documents to one-dimensional vectors.

Quantum Neural Networks QNNs are parameterized quantum circuits that can be trained for tasks such as pattern recognition, optimization, and classification. Our QNNs are constructed using parameterized two-qubit unitaries which incorporates the data encoding through data re-uploading [10]. The label prediction is obtained by projecting m qubits of the final quantum state onto a fixed basis. With $m = 1$ qubits we can assign binary labels from the probability of observing the `0` or `1` bitstring. We predict multi-class labels using $m > 1$ qubits and one-hot encoding – from the 2^m unique bitstrings, we down-select on the weight-1 bitstrings and renormalize the extracted amplitudes.

The choice of gates used for data re-uploading is derived from three-gate decompositions: $\mathcal{D}_1 = R_X(\theta_1)R_Y(\theta_2)R_X(\theta_3)$, $\mathcal{D}_2 = R_X(\theta_1)R_Z(\theta_2)R_X(\theta_3)$ and $\mathcal{D}_3 = R_Y(\theta_1)R_Z(\theta_2)R_Y(\theta_3)$. These sequences use Pauli rotation gates and the angles θ_1, θ_2 embed features x_i re-scaled to $[0, 2\pi]$, and θ_3 is trainable. With a n qubit circuit we can embed $4(n-1)$ unique features.

The general QNN is built using p layers of 2-qubit unitaries which combine the decompositions $\mathcal{D}_i$, either parameterized ZZ coupling gates or using a decomposition of SU(4) operations. The combination of $\mathcal{D}_1, \mathcal{D}_2$ and parameterized ZZ couplings contains $3(n-1)p$ trainable parameters. The combination of $\mathcal{D}_3$ and SU(4) decomposition has $11(n-1)p$ trainable parameters. Each QNN is constructed and trained in `Pennylane` [11] using batch gradient descent with categorical cross entropy loss using batch size 32.

The QNN performance is compared to a CNN that replicates a setup used in a previous study [6], but modified to take the angle embedding prepared for a QNN as the input. The CNN is built and evaluated using `Pytorch` [12]. The binary classifier had around 9,900 trainable parameters and the multiclass classifer has around 10,800 trainable parameters. In comparison the largest QNN trained on P3B3 had 108 parameters, the largest QNN trained on MADELON had 550 parameters.

Results When both CNN and QNN models are given the same P3B3 data we observe that the performance of a classical CNN is comparable to the performance of the QNNs, with the caveat that the data pre-processing is optimized for the QNNs and there are no guarantees that this feature formatting is ideal or optimal for the classical CNN. We observe that QNNs have highly accurate performance for binary classification (MADELON: 80% and P3B3: 88%). For balanced multiclass classification QNNs have high precision (MADELON: 50%/52%/68%) but unbalanced data remains a challenge (P3B3: 71%/74%/16%). Future work will investigate the converse – use the matrix expansion of `state-of-the-art` embeddings which are optimized for classical CNNs, and convert those into quantum circuit parameters, or in general explore the influence of longer feature vector lengths.

Acknowledgment

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The publisher acknowledges the US government license to provide public access under the DOE Public Access Plan (https://energy.gov/doe-public-access-plan).

This research used resources of the Oak Ridge Leadership Computing Facility (OLCF) and the Compute and Data Environment for Science (CADES) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. The research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

REFERENCES

[1] B. Coecke, G. de Felice, K. Meichanetzidis, and A. Toumi. Foundations for near-term quantum natural language processing. *arXiv:2012.03755*, 2020.

[2] Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, and Bob Coecke. Qnlp in practice: Running compositional models of meaning on a quantum computer. *Journal of Artificial Intelligence Research*, 76:1305–1342, 2023.

[3] R. Guarasci, G. De Pietro, and M. Esposito. Quantum natural language processing: Challenges and opportunities. *Appl. Sci.*, 12:5651, 2022.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[5] Isabelle Guyon. Madelon. UCI Machine Learning Repository, 2008. DOI: https://doi.org/10.24432/C5602H.

[6] Hong-Jun Yoon, John Gounley, M Todd Young, and Georgia Tourassi. Information extraction from cancer pathology reports with graph convolution networks for natural language texts. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 4561–4564. IEEE, 2019.

[7] Tanmoy Bhattacharya, Thomas Brettin, James H Doroshow, Yvonne A Evrard, Emily J Greenspan, Amy L Gryshuk, Thuc T Hoang, Carolyn B Vea Lauzon, Dwight Nissley, Lynne Penberthy, et al. AI meets exascale computing: Advancing cancer research with large-scale high performance computing. *Frontiers in Oncology*, 9:984, 2019.

[8] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, 30:31–40, 2009.

[9] Andrew E Blanchard, Shang Gao, Hong-Jun Yoon, J Blair Christian, Eric B Durbin, Xiao-Cheng Wu, Antoinette Stroup, Jennifer Doherty, Stephen M Schwartz, Charles Wiggins, et al. A keyword-enhanced approach to handle class imbalance in clinical text classification. *IEEE journal of biomedical and health informatics*, 26(6):2796–2803, 2022.

[10] Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020.

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

[12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.