THE DAWN OF QUANTUM NATURAL LANGUAGE PROCESSING

Riccardo Di Sipio[1], Jia-Hong Huang[2], Samuel Yen-Chi Chen[3], Stefano Mangini[4], Marcel Worring[2]

[1]Ceridian HCM Inc., [2]University of Amsterdam, [3]Brookhaven National Laboratory, [4]University of Pavia

ABSTRACT

In this paper, we discuss the initial attempts at boosting understanding human language based on deep-learning models with quantum computing. We successfully train a quantum-enhanced Long Short-Term Memory network to perform the parts-of-speech tagging task via numerical simulations. Moreover, a quantum-enhanced Transformer is proposed to perform the sentiment analysis based on the existing dataset.

***Index Terms*—** Natural Language Processing, Quantum Computing, Quantum Machine Learning, Quantum Neural Networks, Transformer, LSTM, Variational Quantum Circuits

1. INTRODUCTION

In recent times, large pre-trained neural network models have been successfully used to achieve state-of-the-art performance in computer vision (CV) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] and language processing (NLP) [14, 15, 16, 17, 18, 19, 20, 21, 22, 23]. In particular, the field of NLP is seeing an exponential expansion in terms of real-life applications such as machine translation [24], document classification [25], interaction with a chatbot [26]) but also network size as Transformer-based models seem to outperform long-standing architectures such as Long Short-Term Memory (LSTM) recurrent neural networks [27]. However, the main issue with this marvellous kind of neural networks is that the appalling size of parameters (in the order of hundreds of billions in the case of OpenAI's GPT-3 [28]) usually requires a training dataset of huge dimension such as the complete Wikipedia corpus in several languages, and a massive computer cluster to carry out the training.

On the other hand, the past years have witnessed the rise of quantum computing both in terms of hardware development [29, 30] and implementation of algorithms on such platforms [31, 32]. In quantum computing [33], computations are carried over the equivalent of bits called qubits, which notoriously can handle information in a non-binary state thanks to a property of quantum systems called superposition [34]. A quantum circuit is a series of operations applied to qubits that change their state, *e.g*. by changing their relative phase or angles in the quantum Hilbert space. Qubits can be represented geometrically with a so called Bloch sphere [34], so an operation onto a qubit corresponds to a rotation of the quantum state vector in this virtual space. One key concept of quantum algorithms is that, because of the intrinsic nature of qubits [34], certain calculations can be carried out with smaller complexity compared to the classical equivalent (this property is known as super-polynomial speedup [33, 35, 36]). This is especially true in the case of chemistry, where quantum algorithms are used to predict electronic structures [37], ground-state energies[38], and the spatial configuration of proteins [39]. In fact, quantum computers are extremely efficient in cases where CPUs may take — literally — forever, such as in the simulation of the Ferredoxin molecule [40].

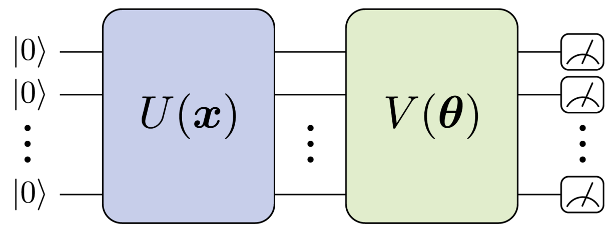

Fig. 1. Generic architecture for variational quantum circuits. $U(\mathbf{x})$ is the quantum routine for encoding the (classical) input data $\mathbf{x}$ into a quantum state, and $V(\boldsymbol{\theta})$ is the variational circuit block with tunable parameters $\boldsymbol{\theta}$. A quantum measurement over some or all of the qubits follows.

Practical applications are more likely to be a hybrid of classical and quantum operations since the currently available quantum computers are only small to intermediate scale. Under this hybrid paradigm, most of the calculations are carried out on classical computers, while the quantum devices are deployed to solve the computational tasks for which certain quantum advantages can be demonstrated. This hybrid approach is not too different from what has been done in the past decade with GPUs. Thus, the main idea behind Quantum Machine Learning (QML) [41, 42] is to replace parts of a neural network (*e.g*. linear layers) with a quantum counterpart. A comparison between different learning models is shown in Tab. 1.

Finally, as classical machine learning does not end with just neural network, also quantum machine learning find its strength in a variety of methods, such as quantum support vector machines [43, 44]. While it is still too early to claim that quantum computing has taken over [45], there are some areas where it can give an advantage as for example in drug discovery [46] or finance [47]. We argue that one field that so far has been poorly explored in QML is NLP.

Our contributions include:

- We are the first who successfully train a quantum-enhanced Long Short-Term Memory model to perform the parts-of-speech tagging task.

- We are the first to propose a quantum-enhanced Transformer model to perform the sentiment analysis task.

2. QUANTUM COMPUTING AND HUMAN LANGUAGE

In early 2020, the British company called Cambridge Quantum Computing announced the introduction of a "meaning-aware" Quantum NLP model, *i.e.* a mathematical theory able to splice the semantic information of words with the syntactic structure of a sentence [48]. The bold claim is based on the observation that syntactic structures such as Noam Chomsky's Context-free grammars or Categorical Compositional Distributional (DisCoCat) can be formulated in the framework of quantum physics [48]. While the concept of a Hilbert space for NLP may seem far-fetched at first, an intuitive explanation goes as follows: NLP-as-we-know-it is rooted in classical statistics (for example, word embeddings are vectors in a $\mathbb{R}^d$ space [49]), but classical statistics itself has its own limits. The field of physics called quantum mechanics is described by the mathematics of quantum statistics, which extends classical statistics by representing objects with matrices of complex numbers. Even if we are not claiming that words are made of particles, it does happen by accident that the kind of statistics needed to describe human language is substantially the same of quantum physics. Coincidentally, we are also entering a time in which rudimentary and still quite noisy devices [50] able to carry out computations based on qubits rather than digital bits of information are becoming more and more accessible to the wider public (including small businesses, start-ups and individuals). Thus, it makes sense in this decade to explore the possibility that quantum computing may give a boost to natural language processing.

3. DOCUMENT CLASSIFICATION WITH VARIATIONAL QUANTUM CIRCUITS

In the first example that we want to discuss, a deep neural network [51] is used to provide a sentence embedding. A final layer ("head") maps these embeddings to a probability vector of dimension $(n_{tokens}, n_{classes})$.

In a classical machine learning setting, this is easily achievable by the means of a linear layer with `softmax` activation. In the QML framework, the calculation is carried by first putting the input qubits into the initial state (*e.g.* a string such as 010010), then they are entangled between each other, rotated by arbitrary angles, and finally observed ("measured"). The goal of the training is to find the rotation angles mentioned above that optimize some cost function such as the mean squared error (MSE) between the output and some pre-determined labels. This kind of quantum circuit is known as Variational Quantum Circuit (VQC) [41, 52]. See Fig. 1 for VQC architecture visualization. However, a VQC cannot change the dimensionality of the input but only rotate the state of the qubits. Hence, the VQC is "dressed", *i.e.* it is sandwiched between two classical linear layers to match the dimensionality. A classical layer "squeezes" the input to match the number of qubits, and another classical layer "bloats" the output to match the dimension of hidden vectors.

Preliminary results indicate that such a hybrid network can be successfully trained to perform classification. However, one has to keep in mind the heavy lifting of this operation is performed by the sentence embedding deep network.

Table 1. Comparison between different learning models.

Classical	Quantum	Hybrid
bits	qubits	bits + qubits
DNN, GBDT	VQC	DNN + VQC
Broad applications	Quantum advantage	Broad applications with Quantum Advantage
Hardware-ready	Limited resources	Quantum hardware only where necessary

4. QUANTUM-ENHANCED LONG SHORT-TERM MEMORY NEURAL NETWORK

As documents are usually presented as sequences of words, historically one of the most successful techniques to manipulate this kind of data has been the Recurrent Neural Network (RNN) architecture, and in particular a variant called Long Short-Term Memory (LSTM) [27]. The "trick" that makes these networks so popular in the analysis of sequential data is a combination of "memory" and "statefulness" that help with identifying which components of the input are relevant to compute the output. While the mathematics is quite thick as we will see later on, it suffices to say for now that LSTMs allowed machines to perform translations, classification and intent detection with state-of-the-art accuracy until the advent of Transformer networks [14]. Still, it is interesting at least from an educational point of view to dig into LSTMs to see what good quantum computing may bring to the field. For a more thorough discussion, refer to "Quantum LSTM" [53] and "Recurrent Quantum Neural Network" [54].

To begin with, let's review the inner workings of a LSTM. We assume that the input is composed of a sequence of t timesteps (*e.g.* words), each represented by a N-dimensional feature vector (in practical applications N can be large, *e.g.* 512). Also, the network stores a hidden array of vectors h and a state vector c that are updated for each element of the input sequence. For example, if we are only interested in a summary of the sequence (*e.g.* in a sentiment analysis), the last element of the array h will be returned. Instead, if we are interested in a representation of each element (*e.g.* to assign to each word a part-of-speech tag such as noun, verb, etc), we want to have access to every element of h. The calculation can be summarized in the formulas below:

$$f_t = \sigma(W_f \cdot v_t + b_f) \quad (1)$$
$$i_t = \sigma(W_i \cdot v_t + b_i) \quad (2)$$
$$\tilde{C} = \tanh(W_C \cdot v_t + b_C) \quad (3)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{C} \quad (4)$$
$$o_t = \sigma(W_o \cdot v_t + b_o) \quad (5)$$
$$h_t = o_t \odot \tanh(c_t), \quad (6)$$

where v_t is a concatenation of the input element at step t and the hidden state at step $t-1$, *i.e.* $v_t = [h_{t-1}, x_t]$. In the machine learning lingo, borrowed from electric circuit analysis, f_t is called forget gate, i_t is the input gate, c_t is the update gate and o_t is the output gate. The matrices W_f, W_i, W_c and W_o and the bias vectors b_f, b_i, b_c and b_o are the parameters that have be learned during the supervised training and implement the part of the calculation called linear dense layer that we want to replace with the quantum equivalent. As is often the case, a non-linearity is introduced by the application of sigmoid (σ) and hyperbolic tangent ($\tanh$) functions to the output of these four dense layers, which effect is to determine whether a part of the input has to be considered (values close to 1) or ignored (values close to 0). Fig. 2 shows a graphical representation of the information flow and operations carried out inside a LSTM.

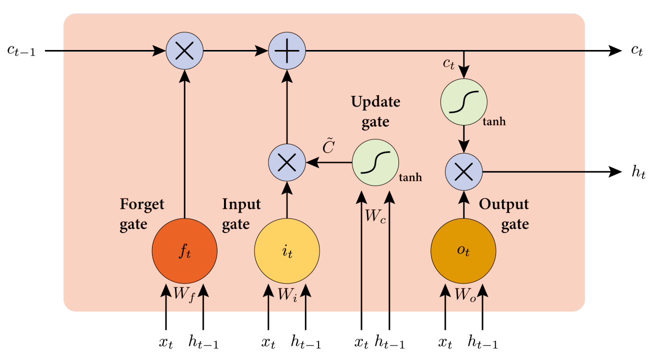

Fig. 2. Graphical representation of the information flow and operations carried out inside a LSTM neural network layer. In the quantum-enhanced LSTM, each of the classical operation W_f, W_i, W_C and W_o is replaced by a hybrid quantum-classical component which includes a VQC sandwiched between classical layers.

Notably, the linear layers of LSTM change the dimensionality of the input tensor, something that is not possible with qubit rotations. As described in Sec. 3, the VQC layers have to be "dressed", *i.e.* sandwiched between two classical linear layers to match the dimensionality. This is applied to f_t, i_t, c_t and o_t separately. Thus, overall there is still a sizeable number of classical parameters to be learned during the training.

To test the setting described above, an intuitive but not trivial example is provided: a *part-of-speech tagging* task. In an example, two sentences ("The dog ate the apple" and "Everybody read that book") have been annotated with POS tags. For example, the first sentence is ["DET", "NN", "V", "DET", "NN"]. The LSTM will output the hidden array of vectors $[h_0, h_1, h_2, h_3, h_4]$, one for each word. A dense layer "head" with `softmax` activation is attached to the LSTM's outputs to calculate the probability that each word may be a determinant, noun or verb. We trained the two networks (classical and quantum LSTM) for 300 epochs each. The parameters used to define the quantum network are described in Tab. 2. As shown in Fig. 3, the cross-entropy loss decreases as a function of the training epoch, and after 150 epochs both networks are able to tag correctly the two sentences. Due to the complexity of the simulation of the quantum circuit, it took approximately 10 minutes to finish the training, to be compared to a mere 10 seconds for the classical case. Also, it seems that the quantum LSTM needed to be trained for longer epochs to achieve 100% accuracy. At the end of the 300 epochs, the loss of the classical network is 0.053 while that of the quantum network is 0.056. It is worth noticing that the quantum LSTM uses less than half the parameters of the purely classical one (199 vs. 477), while retaining the same overall performances. The open-source code is available at: `https://github.com/rdisipio/qlstm`.

Table 2. Parameters used to define the quantum-enhanced LSTM network.

Parameter	Value
epochs	300
vocab size	5
number of tags	3
embedding dim	8
hidden dim	6
no. of quantum layers	1
no. of qubits in VQC	4
Total number of weights	199

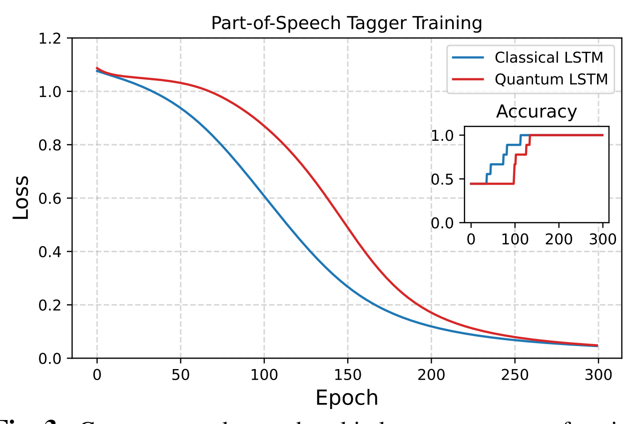

Fig. 3. Cross-entropy loss and multi-class accuracy as a function of the training epoch for the classical and quantum LSTM Part-of-Speech taggers.

5. TOWARD A QUANTUM TRANSFORMER

As mentioned in the introduction, the Transformer architecture [14] revolutionized the analysis of sequential data, and in particular that of human-written documents. Transformers are a neural network architecture that is optimized to analyze sequential data on highly-parallel devices such as GPUs and TPUs. Differently from recurrent networks, Transformers do not have "memory" but are still able to perform the trick by a combination of position-dependent embeddings (*i.e*. words embeddings are supplemented by another set of vectors that depend on the position the word has in the sentence, and on the index of the embedding dimension) and attention (*i.e*. figuring out which parts of the inputs are relevant to calculate the output). At the core of any Transformer sits the so-called Multi-Headed Attention [14]. The idea is to apply three different linear transformations W_Q, W_K and W_V to each element of the input sequence to transform each word embedding into some other internal representation states called Query (Q), Key (K), and Value (V) [14]. These states are then passed to the function that calculates the attention weights, which is simply defined as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (7)$$

To promote the Transformer from the classical to quantum real, one can replace the linear transformations W_Q, W_K and W_V with VQCs [42]. A preliminary experiment for sentiment analysis made use of the IMDB dataset [55] for binary classification. We used PennyLane [56] version 0.13.0 and its plugin to perform simulations of quantum processes with the Qulacs library [57], which is a VQC simulator that runs on a GPU. The parameters of the hybrid network are described in Tab. 3. It took about 100 hours to train the classifier for a single epoch. It is clear that unless a more sophisticated training method is devised, model development and parameter tuning is prohibitive with the hardware currently available to the general public. The open-source code is available at `https://github.com/rdisipio/qtransformer`.

Table 3. Parameters used to define the proposed quantum-enhanced transformer network.

Parameter	Value
batch size	32
epochs	1
vocab size	50,000
embedding dim	8
max seq length	64
feed-forward net dim	8
drop-out rate	0.1
no. of transformer blocks	1
no. of transformer heads	2
no. of quantum layers	1
no. of qubits in transformer blocks	2
no. of qubits in feed-forward net dim	2
Total number of weights	$\mathcal{O}$(150,000)

6. CONCLUSIONS

It is a legitimate question whether it would be possible to devise a fully-quantum Transformer, *i.e*. a quantum circuit acting upon qubits in a fashion similar to a combination self-attention and positional embedding. While we cannot offer a definite answer here, we argue that most of the building blocks of the basic Transformer have a corresponding quantum operation, as arithmetic operations and exponentiation can be implemented as quantum circuits [58], or that they can be cast in form of variational quantum circuits. It is the sheer size of the computation that is likely beyond the limits of current quantum computers, although we foresee that this situation may change rapidly over the course of this decade. The end of this paper is a good place to circle back to the original question of what a Transformer actually is, and why it is so successful at interpreting human language. While a number of interpretations have been given in terms of computations on graphs [59, 60], in due course quantum computing may offer alternative ways to apply attention to sequential data, or provide some completely different conceptual framework altogether [61]. Only time will tell.

7. REFERENCES

[1] C-H Huck Yang, Fangyu Liu, Jia-Hong Huang, Meng Tian, I-Hung Lin, Yi Chieh Liu, Hiromasa Morikawa, Hao-Hsiang Yang, and Jesper Tegner, "Auto-classification of retinal diseases in the limit of sparse data using a two-streams machine learning model," in ACCV. Springer, 2018, pp. 323–338.

[2] Yi-Chieh Liu, Hao-Hsiang Yang, C-H Huck Yang, Jia-Hong Huang, Meng Tian, Hiromasa Morikawa, Yi-Chang James Tsai, and Jesper Tegner, "Synthesizing new retinal symptom images by multiple generative models," in ACCV. Springer, 2018, pp. 235–250.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," NIPS, vol. 25, pp. 1097–1105, 2012.

[4] Jia-Hong Huang, "Robustness analysis of visual question answering models by basic questions," King Abdullah University of Science and Technology, Master Thesis, 2017.

[5] C-H Huck Yang, Jia-Hong Huang, Fangyu Liu, Fang-Yi Chiu, Mengya Gao, Weifeng Lyu, Jesper Tegner, et al., "A novel hybrid machine learning model for auto-classification of retinal diseases," Workshop on Computational Biology, ICML, 2018.

[6] Jia-Hong Huang, Cuong Duc Dao, Modar Alfadly, and Bernard Ghanem, "A novel framework for robustness analysis of visual qa models," in AAAI, 2019, vol. 33, pp. 8449–8456.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in CVPR, 2016, pp. 770–778.

[8] Jia-Hong Huang and Marcel Worring, "Query-controllable video summarization," in ICMR, 2020, pp. 242–250.

[9] Tao Hu, Pascal Mettes, Jia-Hong Huang, and Cees GM Snoek, "Silco: Show a few images, localize the common object," in ICCV, 2019, pp. 5067–5076.

[10] Jia-Hong Huang, Luka Murn, Marta Mrak, and Marcel Worring, "Gpt2mvs: Generative pre-trained transformer-2 for multi-modal video summarization," in ICMR, 2021, pp. 580–589.

[11] Jia-Hong Huang, Modar Alfadly, and Bernard Ghanem, "Vqabq: Visual question answering by basic questions," VQA Challenge Workshop, CVPR, 2017.

[12] Jia-Hong Huang, Modar Alfadly, and Bernard Ghanem, "Robustness analysis of visual qa models by basic questions," VQA Challenge and Visual Dialog Workshop, CVPR, 2018.

[13] Jia-Hong Huang, Modar Alfadly, Bernard Ghanem, and Marcel Worring, "Assessing the robustness of visual question answering," arXiv preprint arXiv:1912.01452, 2019.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in NIPS, 2017, pp. 5998–6008.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[16] Jia-Hong Huang, Ting-Wei Wu, C-H Huck Yang, Zenglin Shi, I Lin, Jesper Tegner, Marcel Worring, et al., "Non-local attention improves description generation for retinal images," in WACV, 2022, pp. 1606–1615.

[17] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al., "Language models are unsupervised multitask learners," OpenAI blog, vol. 1, no. 8, pp. 9, 2019.

[18] Jia-Hong Huang, Ting-Wei Wu, and Marcel Worring, "Contextualized keyword representations for multi-modal retinal image captioning," in ICMR, 2021, pp. 645–652.

[19] Chao-Han Huck Yang, Yun-Yun Tsai, and Pin-Yu Chen, "Voice2series: Reprogramming acoustic models for time series classification," 2021.

[20] Jia-Hong Huang, Ting-Wei Wu, Chao-Han Huck Yang, and Marcel Worring, "Deep context-encoding network for retinal image captioning," in ICIP. IEEE, 2021, pp. 3762–3766.

[21] Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Xiaoli Ma, and Chin-Hui Lee, "Characterizing speech adversarial examples using self-attention u-net enhancement," 2020.

[22] Chao-Han Huck Yang, Jun Qi, Samuel Yen-Chi Chen, Pin-Yu Chen, Sabato Marco Siniscalchi, Xiaoli Ma, and Chin-Hui Lee, "Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition," in ICASSP, 2021, pp. 6523–6527.

[23] Jia-Hong Huang, C-H Huck Yang, Fangyu Liu, Meng Tian, Yi-Chieh Liu, Ting-Wei Wu, I Lin, Kang Wang, Hiromasa Morikawa, Hernghua Chang, et al., "Deepopht: medical report generation for retinal images via deep models and visual explanation," in WACV, 2021, pp. 2442–2452.

[24] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.

[25] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy, "Hierarchical attention networks for document classification," in NAACL, 2016, pp. 1480–1489.

[26] Jack Cahn, "Chatbot: Architecture, design, & development," UPenn Department of Computer and Information Science, 2017.

[27] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] Tom B. Brown, Benjamin Mann, Ilya Sutskever, and Dario Amodei, "Language models are few-shot learners," 2020.

[29] Göran Wendin, "Quantum information processing with superconducting circuits: a review," Reports on Progress in Physics, vol. 80, no. 10, pp. 106001, 2017.

[30] Hiroyuki Mizuno and Masanao Yamaoka, "Research and development of annealing and gate-based quantum computers," Hitachi Review, vol. 70, pp. 486–490, 2021.

[31] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran, "Evaluating analytic gradients on quantum hardware," Physical Review A, vol. 99, no. 3, pp. 032331, 2019.

[32] Marcello Benedetti, Mattia Fiorentini, and Michael Lubasch, "Hardware-efficient variational quantum algorithms for time evolution," Physical Review Research, vol. 3, no. 3, pp. 033083, 2021.

[33] Michael A. Nielsen and Isaac L. Chuang, Quantum computation and quantum information, Cambridge University Press, Cambridge, UK, 10th anniversary ed. (2010) edition, 2010.

[34] Basem R Kazem and Mezher B Saleh, "The effect of pauli gates on the superposition for four-qubit in bloch sphere," journal of kerbala university, vol. 18, no. 1, 2, 3, 4, 2020.

[35] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd, "Quantum machine learning," Nature, vol. 549, no. 7671, pp. 195–202, Sept. 2017.

[36] Shalev Ben-David, Andrew M Childs, András Gilyén, William Kretschmer, Supartha Podder, and Daochen Wang, "Symmetries, graph properties, and quantum speedups," in FOCS. IEEE, 2020, p. 649.

[37] James D. Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik, "Simulation of electronic structure hamiltonians using quantum computers," Molecular Physics, vol. 109, no. 5, pp. 735–750, Mar 2011.

[38] Cornelius Hempel, Christine Maier, Jonathan Romero, Jarrod McClean, Thomas Monz, Heng Shen, Petar Jurcevic, Ben P. Lanyon, Peter Love, Ryan Babbush, Alán Aspuru-Guzik, Rainer Blatt, and Christian F. Roos, "Quantum chemistry calculations on a trapped-ion quantum simulator," Phys. Rev. X, vol. 8, pp. 031022, Jul 2018.

[39] Mark Fingerhuth, Tomáš Babej, and Christopher Ing, "A quantum alternating operator ansatz with hard and soft constraints for lattice protein folding," 2018.

[40] Dave Wecker, Bela Bauer, Matthew B. Hastings, and Matthias Troyer, "Gate-count estimates for performing quantum chemistry on small quantum computers," Physical Review A, vol. 90, no. 2, Aug 2014.

[41] Stefano Mangini, Francesco Tacchino, Dario Gerace, Daniele Bajoni, and Chiara Macchiavello, "Quantum computing models for artificial neural networks," EPL, vol. 134, no. 1, pp. 10002, 2021.

[42] Samuel Yen-Chi Chen, Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Xiaoli Ma, and Hsi-Sheng Goan, "Variational quantum circuits for deep reinforcement learning," 2020.

[43] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," Nature, vol. 567, no. 7747, pp. 209–212, Mar 2019.

[44] Jamie Heredge, Charles Hill, Lloyd Hollenberg, and Martin Sevior, "Quantum support vector machines for continuum suppression in b meson decays," arXiv preprint arXiv:2103.12257, 2021.

[45] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R. McClean, "Power of data in quantum machine learning," Nature Communications, vol. 12, no. 1, pp. 2631, May 2021.

[46] Maximillian Zinner, Florian Dahlhausen, Philip Boehme, Jan Ehlers, Linn Bieske, and Leonard Fehring, "Quantum computing's potential for drug discovery: early stage industry dynamics," Drug Discovery Today, 2021.

[47] Daniel J Egger, Claudio Gambella, Jakub Marecek, Rudy Raymond, Andrea Simonetto, Stefan Woerner, and Elena Yndurain, "Quantum computing for finance: state of the art and future prospects," IEEE Transactions on Quantum Engineering, 2020.

[48] Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis, and Alexis Toumi, "How to make qubits speak," 2021.

[49] Jeffrey Pennington, Richard Socher, and Christopher D Manning, "Glove: Global vectors for word representation," in EMNLP, 2014, pp. 1532–1543.

[50] John Preskill, "Quantum Computing in the NISQ era and beyond," Quantum, vol. 2, pp. 79, Aug. 2018.

[51] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil, "Universal sentence encoder," 2018.

[52] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles, "Variational quantum algorithms," Nature Reviews Physics, vol. 3, no. 9, pp. 625–644, Sept. 2021.

[53] Samuel Yen-Chi Chen, Shinjae Yoo, and Yao-Lung L. Fang, "Quantum long short-term memory," 2020.

[54] Johannes Bausch, "Recurrent quantum neural networks," 2020.

[55] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, and Christopher Potts, "Learning word vectors for sentiment analysis," in ACL, Portland, Oregon, USA, June 2011, pp. 142–150, ACL.

[56] Ville Bergholm, Josh, Maria, Antal, and Killoran, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," 2020.

[57] Yasunari Suzuki, Yoshiaki Kawase, and Keisuke Fujii, "Qulacs: a fast and versatile quantum circuit simulator for research purpose," 2021.

[58] Vlatko Vedral, Barenco, and Artur Ekert, "Quantum networks for elementary arithmetic operations," PRA, vol. 54, pp. 147–153, Jul 1996.

[59] Seongjun Yun, Jeong, and J Kim, "Graph transformer networks," in NIPS, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. 2019, vol. 32, Curran Inc.

[60] Deng Cai and Wai Lam, "Graph transformer for graph-to-sequence learning," in AAAI, 2020, vol. 34, pp. 7464–7471.

[61] Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, and Bob Coecke, "Qnlp in practice: Running compositional models of meaning on a quantum computer," 2021.