ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

A WNN model based on Probabilistic Quantum Memories

Priscila G. M. dos Santos, Rodrigo S. Sousa and Adenilton J. da Silva *

Universidade Federal Rural de Pernambuco - Departamento de Computação
Rua Dom Manoel de Medeiros, S/N. Campus Dois Irmãos,
52171-900, Recife, Pernambuco - Brazil

Abstract. In this work, we evaluate a Weightless Neural Network model based on a Probabilistic Quantum Memory. In order to evaluate the classification capabilities of this quantum model, we conducted classical experiments using an equivalent classical description of the Probabilist Quantum Memory algorithm. We present the first evaluation of a quantum weightless neural networks on public benchmark datasets.

1 Introduction

In this work, we evaluate a weightless network model [1] based on an associative memory model [2] that makes use of quantum computing to store and retrieve information. Quantum computing is a field that has been gathering increasingly attention due to its current advances [3]. It touches upon ideas of quantum mechanics and information theory. A quantum computer is the concept for a computational device capable of representing information by making use of microscopic quantum level effects to perform computational tasks [4]. In quantum computing, the quantum bit (qubit) represents the basic unit of information in a quantum system. Analogously to the behavior of a subatomic particle, the qubit can be in more than one state at a given time. Equation (1) describes one qubit in superposition, where α and β are the probabilistic amplitudes associated with the states $|0\rangle$ and $|1\rangle$, respectively.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

The probabilistic amplitudes are represented by complex numbers, obeying the normalization rule described in equation (2). The probability of a qubit being found in any of the possible states is given by the modulus squared of its amplitudes after a measurement is made.

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2}$$

An important quantum characteristic is the necessity to measure in order to extract information from a quantum state. After a measurement, the system collapses to one of its possible states in the superposition. For instance, given the quantum state described in equation (3), the probability of finding $|i\rangle$ after a measurement is $p_i = |\alpha_i|^2$.

*This work was supported by the Serrapilheira Institute (grant number Serra-1709-22626), CNPq (grant number 420319/2016-6) and FACEPE.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

$$|\psi\rangle = \sum_i \alpha_i |i\rangle \tag{3}$$

Due to the capacity of dealing with states in superposition and other incorporated quantum effects, it has been demonstrated that quantum algorithms are capable of solving problems for which there are no known efficient solutions through classical computation [5]. Lately, much attention has been given to quantum algorithms for machine learning [6]. A quantum generalisation of a neural network is proposed in [7]. In this work, we perform the first experimental evaluation of a quantum model of weightless neural network. The model is able to perform classification tasks without the need to perform any previous training.

2 Probabilistic Quantum Memories

Here, we present the quantum memory model that is used in the nodes of the weightless network model. A Probabilistic Quantum Memory (PQM) [2] is a content-addressable quantum memory. It outputs the probability of a given input pattern being stored on the memory by calculating the Hamming distance between the input pattern and all the other patterns stored on the memory. It is a probabilistic model designed to recognize even incomplete or noisy information. Despite being an associative model, the PQM possess a highly scalable storage capability, being able to store all the possible 2^n binary patterns of n bits. The storage and retrieval PQM procedures are explained in the following subsections.

2.1 Storage Procedure

The PQM stores information in a uniform quantum superposition. The quantum resulting state after the storage mechanism execution is described in Eq. (4), where p is the number of patterns in the dataset and p^i are the stored patterns.

$$|M\rangle = \frac{1}{\sqrt{p}} \sum_{i=1}^{p} |p^i\rangle \tag{4}$$

2.2 Retrieval Procedure

The retrieval procedure computes the Hamming distance between an input and all the patterns superposed on the memory quantum state. It probabilistically indicates the chance of a given input pattern being on the memory based on the results of its distance distribution to the stored patterns in superposition. If the input pattern is very distant from the patterns stored in the memory, one will obtain 1 as a result with a large probability. Otherwise, 0 would be obtained. Since the memory state is prepared in a superposition, the retrieval mechanism can calculate the distances from the input to all the patterns with the computational cost to compute the distance from the input to one pattern.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

3 Quantum Weightless Classifier

The model evaluated here is a weightless network quantum model composed of Probabilistic Quantum Memories acting as the network neurons. The classifier is devised by using an array of PQM instances capable of distance based classification. Each PQM instance, by itself, works as a single class classifier, being responsible for the classification of one class in the dataset. The model does not demand any training procedure in a sense that the neurons do not have to be iteratively adjusted to learn from the training patterns. The model classification procedure and the necessary set-up are detailed bellow.

3.1 Set-up Procedure

Despite not demanding a training procedure, the Quantum Weightless Classifier requires an initial set-up procedure in order to be able to perform classification tasks. For a given dataset with n classes, the model will be composed of n PQMs acting as neurons. The training samples must be divided and grouped by class. For each group, a new PQM is created and used to store all the samples belonging to that group, making in total n PQM instances, one for each class.

The set-up procedure consists in storing the training samples on their respective class PQM. The n PQMs together define a single classifier. The described setup process can be seen in Alg. 1. Once all the training samples are correctly stored, the model can perform the classification task by calling the PQM retrieval algorithm. Hence, the Quantum Weightless Classifier does not require any previous training in order to classify for all the classes present in the dataset.

Algorithm 1: Probabilistic Quantum Memory Classifier Setup

```
Initialize a PQM Classifier
for each class in dataset do
    Create a new PQM and assign the class label to it
    Store the class training samples on the PQM
    Add the PQM to the PQM Classifier
end
Return the PQM Classifier
```

3.2 Classification Procedure

Once the training samples are stored the model is ready to classify new patterns. The classification procedure can be seen in Alg. 2. In order to classify a new sample, the Quantum Weightless Classifier must present it to all the PQM neurons which constitute the classifier network. Each PQM neuron performs its retrieval algorithm using the presented sample as input. Since each PQMs hold the patterns of a specific class, each output will be the probability of the sample having similar features with the patterns of that specific class. Therefore, the

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

PQM neuron which outputs the smallest expected value, $E(X)$, is assumed to be the one that correctly classifies the sample.

Algorithm 2: Probabilistic Quantum Memory Classifier Classification

```
1 for each PQM in PQM Classifier do
2     Run the PQM retrieval algorithm with input testPattern
3     Calculate the expected value E(X) from the retrieval algorithm
        output
4 end
5 Return the label from the PQM Classifier with the smallest E(X)
```

4 Model evaluation

The weightless neural node is a quantum model and as such, would require a large scale quantum computer in order to be tested in its full scope. Quantum computing is currently on the rise and quantum devices with increasingly numbers of qubits are being developed to supply this demand. However, the current publicly accessible quantum devices we have at disposal can only perform experiments with a small amount of qubits. High scale experiments cannot be conducted on such small devices. Considering this, it is still possible to simulate quantum algorithms on a classical computer. The experiments presented in this section were conducted with a classical reduced version of the algorithm without loss of generality.

4.1 Classical evaluation set-up

To evaluate the Quantum Weightless Classifier, we conducted experiments in a classical computer. First, it is required to simulate the Probabilistic Quantum Memory classically. To do so, we simply followed the description of its recover algorithm as presented in [2]. As for the storage mechanism, it is not needed outside the quantum context and could be greatly simplified by just storing the patterns directly on the memory. Having the PQM classical representation, the QWC can be evaluated by following the set-up and the classification procedures described in the previous sections.

To perform the experiments we used categorical and numerical datasets from the UCI Machine Learning Repository [8]. Details of the selected datasets can be seen in Table 1. All the datasets were preprocessed in order to binarize feature values and deal with any missing values. The binarization process is required in order to simplify the PQM usage. It was done by transforming each possible value a feature could assume in its own separate feature in the sample vector. Datasets containing real numerical values were not considered in order to further simplify the process. Sample vectors containing missing feature values were not removed from the datasets. All the missing feature values were replaced by

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

the value with highest occurrence for the corresponding feature in all the other samples of the considered dataset.

Dataset	Classes	Instances	Attributes	Missing Values
Balance scale	3	625	4	No
Breast cancer	2	286	9	Yes
Lymphography	4	148	18	No
Mushroom	2	8124	22	Yes
Tic-tac-toe	2	958	9	No
Voting records	2	435	16	Yes
Zoo	7	101	17	No

Table 1: Datasets characteristics

Following the QWC set-up algorithm, we stored the training samples in specific PQMs according to the class they belong to. Then, we followed the QWC classification algorithm. The model classification accuracy was evaluated by passing the patterns in the test set as input to each of the PQMs and the class of the PQM which outputed the lowest expected value was set as the evaluated pattern class. This procedure was done for each of the evaluated datasets.

4.2 Results

The results obtained with the experimental set-up described above can be seen in Table 2, where the accuracy of the QWC model can be compared against the results obtained using the k-nearest neighbors algorithm (KNN). The accuracy values shown are the average obtained from a 10-fold cross-validation. The respective values for the standard deviation are included between parentheses. We choose KNN as a baseline comparison because, as well as our evaluated model, it is a non-generalizing learning model and does not require training. The tested KNN was set to use uniform weights for all its points and the k nearest neighbors value was fixed to 5 in all datasets.

To perform an appropriate comparison of the models, a nonparametric statistical test was employed. We used the Wilcoxon paired signed-rank test [9] with $\alpha = 0.05$ to verify whether there exist significant differences between the compared classifiers performances over the chosen datasets. We verified that KNN and QWC are statistically equivalent in Balance scale, Breast cancer, Lymphography, Tic-tac-toe, and Zoo datasets. KNN has better accuracy on Mushroom and Voting records datasets. The significant results are highlighted in the table.

The quantum classifier has a performance equivalent to KNN in five out of the seven tested datasets. The main advantage of the QWC in relation to a WiSARD classifier is the QWC memory requirements. While a RAM node memory grows exponentially with the size of the input, the QWC node memory grows linearly with the size of the input. This memory advantage can allow the implementation of new weightless neural networks architectures.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

Dataset	QWC	KNN
Balance scale	0.8111 (0.0666)	0.8483 (0.0821)
Breast cancer	0.7309 (0.2639)	0.6595 (0.3278)
Lymphography	0.7829 (0.0815)	0.7629 (0.0853)
Mushroom	0.886 (0.0919)	**0.9995** (0.0015)
Tic-tac-toe	0.4542 (0.1199)	0.5939 (0.3754)
Voting records	0.892 (0.0575)	**0.9242** (0.0356)
Zoo	0.92 (0.0872)	0.89 (0.1375)

Table 2: 10-fold cross-validation average accuracy per dataset

5 Conclusion

In this work we evaluated a Quantum Weightless Classifier. The model is based on probabilistic quantum memories capable of distance based classification. In order to test its performance, we conducted experiments in a classical computer through the direct simulation of the quantum memory retrieval algorithm. The quantum model has shown an average accuracy similar to the results obtained using the KNN algorithm in seven datasets.

QWC has shown satisfactory classification performance and the mentioned difficulties can be potentially overcome by modifying the PQM distance function. This work performs the first empirical evaluation of a quantum weightless neural network. There are a lot of possible future works. For instance, we want to verify the model with different distance functions and different neural networks architectures.

References

[1] Adenilton Silva, Wilson de Oliveira, and Teresa Ludermir. A weightless neural node based on a probabilistic quantum memory. In *2010 Eleventh Brazilian Symposium on Neural Networks*, pages 259–264. IEEE, 2010.

[2] Carlo A Trugenberger. Probabilistic quantum memories. *Physical Review Letters*, 87(6):067901, 2001.

[3] John Preskill. Quantum computing in the nisq era and beyond. *arXiv preprint arXiv:1801.00862*, 2018.

[4] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[5] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[6] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195, 2017.

[7] Kwok Ho Wan, Oscar Dahlsten, Hlér Kristjánsson, Robert Gardner, and MS Kim. Quantum generalisation of feedforward neural networks. *npj Quantum Information*, 3(1):36, 2017.

[8] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.

[9] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.