Tensor Networks for Visual and Textual Classification

Sneyder Gantiva, Diego Useche, Fabio A. Gonzalez
Departamento de Ingeniería de Sistemas y Computación
Universidad Nacional de Colombia
Bogotá, Colombia
esgantivar@unal.edu.co
diusecher@unal.edu.co
fagonzalezo@unal.edu.co

***Abstract*—There is an increasing need for more robust, efficient and with less complexity algorithms in Machine Learning. In the following work we present a modern approach to ML based on Tensor Networks, which have been historically applied to physics and more recently to machine learning. The aim of the present article is to show how to do supervised learning of images and texts with the MPS Tensor Network, we apply this network to three data sets of images and texts; the MNIST data set, a data set of Spanish and English words, and the Multimodal IMDb (MM-IMDb) dataset.**

***Index Terms*—machine learning, tensor networks, matrix product states, visual processing, language processing,**

I. Introduction

There has been a growing interest in tensor networks methods for machine learning. Tensor networks have shown important properties for solving problems of many particles systems in quantum mechanics. One of the challenges of many-body problems have been the *curse of dimensionality* [1], i.e the exponential growth of the Hilbert space, which lead to algorithms of exponential complexity. Tensor networks methods have been useful in reducing the complexity of these algorithms by approximating large tensors which arise in quantum mechanics [2].

The tensor network methods have been shown to be successful in describing the ground states of multiparticle systems [3]–[5], in particular tensor networks have been inspired to approximate the one and two dimensional spin systems model known as the Ising Model , problem which has been considered intractable [6]. The most widely used tensor networks are the matrix product states(MPS) [7]–[9] also known as Tensor Trains (TT) [10], which have direct correspondence with the 1 - dimensional spin systems. Other types of tensor networks include the Projected entangled-pair states (PEPS) [11]–[13], and the Multi-scale Entanglement Renormalization Ansatz (MERA), each of them with distinct topology and characteristics.

In addition to the multiple applications of tensor networks in physics, these methods have been applied in machine learning [14], in feature extraction, dimensionality reduction, and deep neural networks [15]–[18]. The MPS tensor networks have found multiple applications in machine learning , such as pattern recognition [2], classification [19], language modeling [20] and unsupervised generative modeling [14].

Another field of recent applications of tensor networks is quantum information and quantum machine learning. Some data sets involve vectors of thousands of components for which quantum computers and quantum algorithms seem promising [21]. Quantum algorithms have been explored for labeling data [22], [23] or generating data [24]–[26].

Given the recent applications of tensor networks on machine learning, the purpose of this article is to give a review of tensor networks and to show how to apply a machine learning framework based on the MPS tensor network developed by (Efthymiou et. al.) [27] and (Stoudenmire et. al.) [2] on images and texts. We will show the results on three data sets, the MNIST data set, an English/Spanish Words data set and the Multimodal IMDb (MM-IMDb) [28] dataset of movie posters and plots.

II. Theory of tensor networks

In the following passages we introduce the fundamental concepts related to tensor networks, we present the idea of a tensor with its operations, and the diagrammatic representations of tensors. Then, we explore the most significant tensor configurations, the MPS, which is the most basic and used tensor network in many-particles quantum mechanics and machine learning.

A. Tensors and their diagrammatic notation

Tensors are multidimensional arrays of complex numbers [29], in other words, they are the multidimensional generalization of scalars, vectors, and matrices. A rank-0 tensor is a scalar, a rank-1 tensor is vector, a rank-2 tensor is a matrix, we can extend to rank-3 or rank-4 tensors as shown in the figure, [17].

The usual way of representing tensors is by Einstein's notation, where each dimension of the tensor is represented by an index which runs over the number of possible values of each index. Hence a scalar is written as x, a vector as v_α, a matrix $M_{\alpha\beta}$ and a 4 rank tensor as $C_{\alpha\beta\gamma\delta}$.

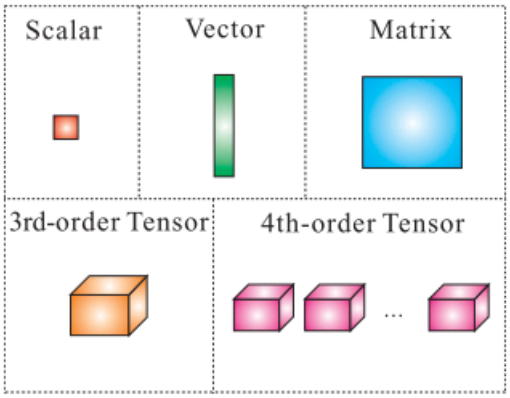

Fig. 1. Graphical representation of tensors. The picture was taken from Cichocki 2016 [17].

It is useful to use the diagrammatic notation to represent tensors. Tensors are represented by shapes and lines, where the shape represent the tensor and each non-connected line represent an index of the tensor, as shown in the figure,

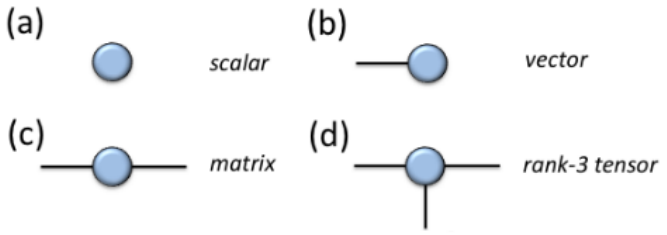

Fig. 2. Diagrammatic representation of tensors. (a) scalar, (b) vector, (c) matrix, (d) rank-3 tensor. The picture was taken from Orus 2014 [29].

The figure shows that a scalar has no lines, and each line represent an index or dimension of the tensor. Therefore a vector has one line, a matrix has two lines, and a rank-3 tensor has three lines.

B. Tensor Contraction

The most important tensor operation that we are going to consider is tensor contraction. The usual way of representing tensor contraction is in the *Einstein summation convention*, in which the index that appears twice is summed over [6]. We illustrate tensor contraction through examples. Consider two vectors $\vec{u}$ and $\vec{v}$ of the same dimension, the dot product is the sum over index $\sum_i u_i v_i$, in *Einstein's notation* that is written as $u_i v_i$, where repeated indices imply summation over the index. In diagrammatic notation, we contract two vectors by connecting the free legs of the two vectors resulting in no free legs(see figure), hence a scalar [30]. In the same way we can extend to matrix-vector multiplication $\vec{b} = A_{i,j} x_j$, matrix-matrix multiplication $C_{i,k} = A_{i,j} B_{j,k}$ contracted in the index j, or higher order tensors $c_{i,j,l,m,p} = a_{i,j,k} b_{k,l,m,p}$, where the line connecting the tensors imply summation over the index as illustrated in the figure [17]. Notice that matrix-vector multiplication results in a diagram with one free leg, hence a vector, and C the result of the matrix-matrix multiplication has two legs, which is a matrix.

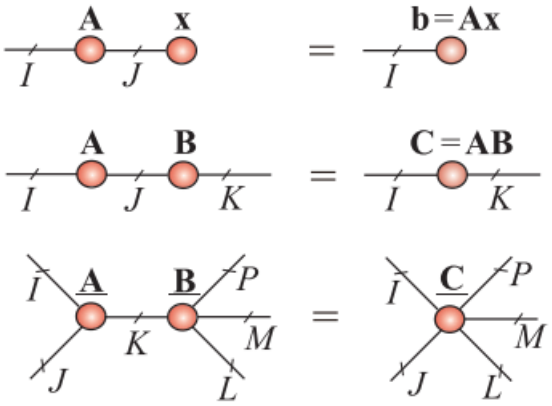

Fig. 3. Diagrammatic representation of tensors contraction, which are the most basic diagrams of tensor networks. In the top we have matrix-vector multiplication, in the middle matrix-matrix multiplication, in the bottom contraction of rank-3 and rank-5 tensor. The picture was taken from Cichocki 2016 [17].

C. The MPS tensor network

We have already seen the graphical and diagrammatic notation of tensors and their contraction. What follows is to explore the most widely used tensor network which is the matrix product states (MPS).

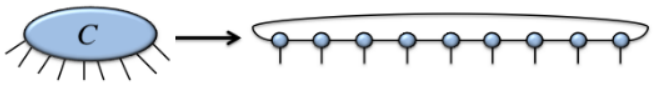

Fig. 4. A tensor of rank N is approximated into a MPS tensor network with periodic boundary conditions. The picture was taken from Orus 2014 [29].

The MPS tensor network [7] was developed to simulate one dimensional quantum systems like the 1 dimensional Ising Model [6], [31]. The MPS tensor network consist on approximating a tensor of rank-N, namely of N dimensions into product of tensors of rank-2 and rank-3 [10], [17]. Consider a data tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, this expression means that $\mathbf{X}$ is a tensor with rank-N and each dimension i runs for I_i values, this vector is represented in the figure. If each of this dimensions can take D values, the complexity to compute this tensor grows in the order of D^N, MPS allow us to reduce this complexity by dividing the tensor into product of tensors of rank-2 and rank-3. The scalar representation $\mathbf{X}$ in MPS representation is,

$$x_{i_1,i_2,\cdots,i_N} = \sum_{r_1,r_2,\cdots,r_{N-1}=1}^{R_1,R_2,\cdots,R_{N-1}} g^{(1)}_{1,i_1,r_1} g^{(1)}_{r_1,i_2,r_2} \cdots g^{(N)}_{r_{N-1},i_N,1}$$

This equation can be more easily understood if we scan through the graphical and diagrammatic representation of the product of the tensors. In the figure , we have a MPS with 4 tensors, the free indices are I_1, I_2, I_3 and I_4 we can observe that $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(4)}$ are matrices, and the $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(4)}$ are tensors of rank 3, the contractions are performed in the R_i's indices, which can be also observed in the equation.

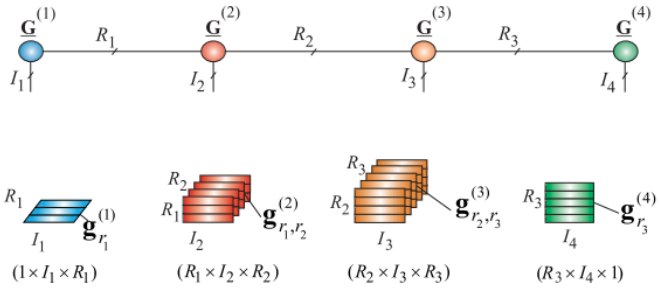

Fig. 5. Graphical and diagrammatical representation of a MPS with 4 sites. The picture was taken from Cichocki 2016 [17].

Other types of tensor networks include the *projected entangled-pair states* (PEPS), which is the generalization of the MPS to higher order dimensions, and the *multi-scale entanglement renormalization ansatz* (MERA), which is a type of tree tensor network (TTN).

D. Tensor networks for machine learning

Now we proceed to analyze the tensor networks' approach to machine learning. Usually a supervised machine learning model is formulated by a set of m training data $x_m \in \mathbb{R}^n$, which live in a n dimensional space, and for each training data there is a label y_m, which is usually a scalar, and the goal is to train the best model which maps the training data with the labels based on a appropriate cost function $J(x_m, y_m, \cdots)$. In neural networks a set of appropriate weights W is looked upon to fit the model which reduces the cost function [17]. For example a binary linear classifier (-1,1) might be defined by its discriminative function as,

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b),$$

In similar fashion we can define the tensorized form,

$$f(\mathbf{X}) = \text{sign}(\langle \mathbf{X}, \mathbf{W} \rangle + b),$$

where $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ represents the input tensor and $\mathbf{W} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ represents the tensor of weights, $\mathbf{y}$ the vector of the scalar outputs and b the bias. The operation is an inner product defined as $\langle \mathbf{X}, \mathbf{W} \rangle = \text{vec}(\mathbf{X})^T \text{vec}(\mathbf{W})$.

There are several possibilities of the cost function to optimize, one simple choice is the *square cost function*, defined as,

$$J(\mathbf{X}, \mathbf{y}|\mathbf{W}, b) = \sum_{m=1}^{M} (y_m - \langle \mathbf{X}, \mathbf{W} \rangle + b)^2$$

The tensor networks approach consists in minimizing the cost function given that the tensor of weights is written in a tensor network form such as MPS or MERA. The optimization problem is reduced to the following,

$$\min J(\mathbf{X}, \mathbf{y}|\mathbf{W}, b), \quad \text{s.t} \quad \mathbf{W} = \langle\langle \mathbf{G}^{(1)}, \cdots, \mathbf{G}^{(N)} \rangle\rangle$$

Hence, we minimize the cost function J given that the weight tensor is presented in tensor train format. Here lies the advantage of the tensor network approach in machine learning, if we assume that $\mathbf{W} \in \mathbb{R}^{D \times D \times \cdots \times D}$ and of rank N, a normal machine learning approach, such as a neural network, would optimize all the weights at each time, which would turn into a complexity of $O(D^N)$, while a tensor network would reduce this complexity as small as $O(NDp^2)$ [29], where p is the dimension of the hidden indices, by means of optimizing each $\mathbf{G}^{(i)}$ tensor at a time with methods such as the Density Matrix Renormalization Group (DMRG) [32].

III. MPS Learning Method

The MPS learning method of this work was based on (Efthymiou et. al.) [27] with a subtle difference, which is that the weights tensor is presented as an MPS with periodic boundary conditions, the weights tensor is,

$$W^l_{i_1, i_2, \cdots, i_N} = \sum_{r_1, r_2, \cdots, r_N} g^{(1)}_{r_N, i_1, r_1} g^{(2)}_{r_1, i_2, r_2} \cdots g^{(N)}_{r_{N-1}, i_N, r_N}$$

Where the i_j are physical indices of the network which run between 0 and 1. The r_j indices are the bond dimensions between neighboring tensors (pixels in the MNIST case), which are fixed along the network in both models, in contrast to (Stoudenmire et. al. model). In our setting the 1st and the N^{th} tensors are contracted, in the usual MPS learning methods they are not. The l index is a free index which accounts for the number of classes in the model.

The data is encoded as a tensor product of qubits, each qubit represents a pixel and a n-gram in the visual and textual classifications respectively. Hence the data is encoded as,

$$\phi_{i_1, i_2, \cdots, i_N}(\mathbf{x}) = \phi_{i_1}(x_1) \otimes \phi_{i_2}(x_2) \cdots \otimes \phi_{i_N}(x_N)$$

the inner product between the weights and the data would take the form,

$$f(\mathbf{x})^l = \sum_{i_1, i_2, \cdots, i_N} W^l_{i_1, i_2, \cdots, i_N} \phi_{i_1}(x_1) \phi_{i_2}(x_2) \cdots \phi_{i_N}(x_N)$$

Up to this point, this method was applied to both multiclass and multilabel classification. The variation resides in the activations and the loss function. In the case of multiclass classification (like MNIST), the discrimination is done by the maximum value of the softmax, and the categorical cross entropy loss function. In the multilabel setting (like the MM-IMBb) the activation is a sigmoid for each label, and a binary cross entropy loss.

In the figure it is shown the contraction of the tensors as done in [27] which is the same contraction type in our model except for the tensors in the extreme, which are also contracted. The Efthymiou et. al. contraction consist on first computing the inner product of the weights and the data tensor, and then contracting by pairs of tensors, taking advantage of the paralelization.

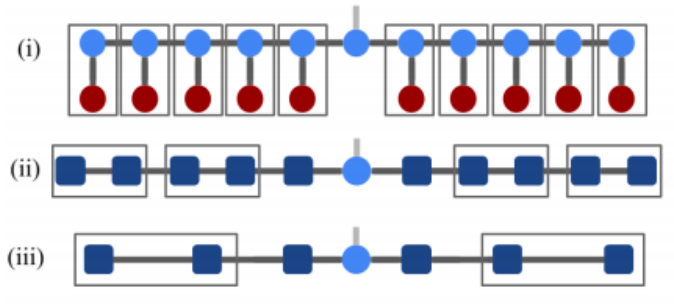

Fig. 6. The contraction of the MPS tensor network by Efthymiou et. al. is done by first contracting the data in red with the weights in light blue, then the resulting tensor are contracted in pairs to take advantage of the paralelization. In our work the tensors in the extremes are also connected and hence contracted. The picture was taken from [27].

IV. Quantum Feature Maps

Given that tensor networks were developed to find the ground states of quantum multiparticle systems, the data in the MPS tensor network should be presented as a quantum state. We explore some quantum feature maps for visual and textual classification.

A. Visual QFM 1: Stoudenmire Quantum Feature Map

A quantum feature map for the MNIST data set was proposed in [2]. It works as follows, the grey scale is normalized from (0, 255) to (0, 1) by dividing each pixel by 255, and then each pixel is mapped to a quantum state given by,

$$|\psi(x)\rangle = \cos\left(\frac{x\pi}{2}\right)|0\rangle + \sin\left(\frac{x\pi}{2}\right)|1\rangle$$

for $x \in (0, 1)$. Then each pixel is concatenated as a tensor product as,

$$|\psi(x_1)\rangle \otimes \cdots \otimes |\psi(x_{l^2})\rangle$$

this quantum feature map was not explored in this work.

B. Visual QFM 2: Binary Pixel Quantum Feature map

The first type of feature map that we explored for the MNIST data set is a simplified version of the Stoudenmire Feature map. In this case, if the gray scale of the pixel is greater or equal 127 then the pixel is mapped to the $|1\rangle$ and if it is less that 127 is mapped to $|0\rangle$, more formally,

$$|\psi(x)\rangle = \begin{cases} |0\rangle & \text{if} \quad \frac{x}{255} < 0.5 \\ |1\rangle & \text{if} \quad \frac{x}{255} > 0.5 \end{cases}$$

And once again the pixels are concatenated as a tensor product as in the Stoudemire Feature Map.

C. Textual QFM 1: N-gram Quantum Feature Map

The N-gram Feature Map was applied to the data set of Spanish and English words, the data was represented as a Bag of n-grams, we used 2-grams, the quantum feature map consists on mapping to the $|0\rangle$ if the n-gram is not present in the word, and to the $|1\rangle$ if the n-gram is present in the word, and then joining the n-grams sites as a tensor product. We illustrate the feature map by an example.

Suppose a word is represented in its n-gram representation as (0,1, 3, 0), then the n-gram feature map would map the data to the quantum state $|0\rangle \otimes |1\rangle \otimes |1\rangle \otimes |0\rangle$, in other terms the MPS tensor network would receive an object of the form ((1,0),(0,1),(0,1),(1,0)). We can summarize the n-gram quantum feature map as follows,

$$|\psi(x)\rangle = \begin{cases} |1\rangle & \text{if} \quad \text{n-gram in text} \\ |0\rangle & \text{otherwise.} \end{cases}$$

D. Textual QFM 1: Tfidf Quantum Feature Map

The TFIDF textual representation stands for term-frequency times inverse document-frequency and it is a variant of the bag of n-gram representation in which the n-grams which are very frequent in the corpus have less weight, compared with n-grams which are not as frequent in the corpus. Hence, the n-grams with high frequency in the text but with low frequency in the corpus are more representative of the text.

A 3-character-gram was trained in [28] for the plots of the movies of the MM-IMDb Data Set. In this document, we propose a quantum feature map for the plots of the MM-IMDb data set based on its TFIDF 3-character-gram. The quantum map works by taking an average value of the non-zero values of the TFIDF vector of a text, and then mapping to the $|0\rangle$ the n-grams which are less than the average and to $|1\rangle$ the n-grams which are greater than the average value of the TFIDF representation. And finally taking a tensor product of the such qubits which would feed the MPS tensor network.

We illustrate the TFIDF quantum feature map with an example: Suppose that a given word has the TFIDF vector (0.12, 0.54, 0, 0,36), then the average of the nonzero values is 0.34, then the vector is mapped to the quantum feature map $|0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle$, and hence the MPS tensor network would receive an object of the form ((1,0),(0,1),(1,0),(0,1)).

V. MPS Experimental Evaluation and Results

We trained the MPS tensor network for the MNIST, the Eng/Spa Data set, and the plots and posters of the MM-IMDb data set using the quantum feature maps previously discussed. We found state-of-the-art results.

A. MNIST Data Set

The MNIST data set is composed of 60000 images for training and 10000 images for validation. We used the binary pixel quantum feature map. Each image has 784 pixels, and each pixel is mapped to a qubit, hence, the MPS tensor network was trained with 784 sites.

We obtained 95.67% of accuracy on the validation data set with a bond dimension $\chi = 10$, 100 epochs and a batch size of 64, in the following two images we show the results of the accuracy and loss in the training process.

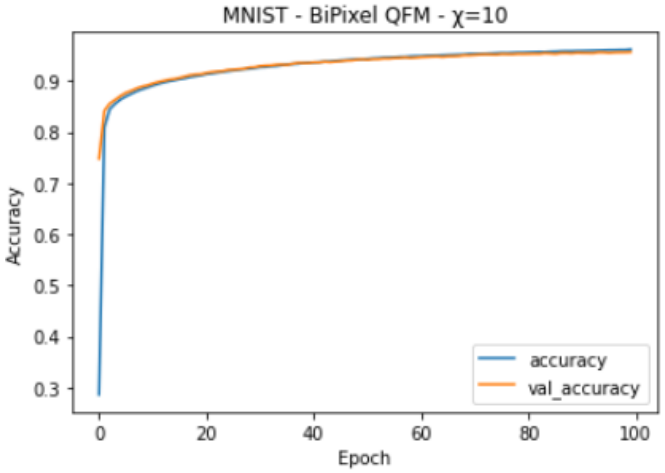

Fig. 7. Accuracy of training and validation data sets on MNIST with a MPS Tensor Network for 100 epochs and bond dimension of 10.

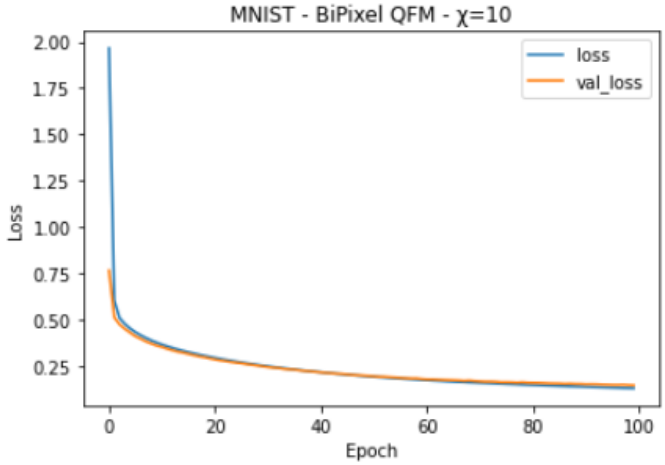

Fig. 8. Loss of training and validation data sets on MNIST with a MPS Tensor Network for 100 epochs and bond dimension of 10.

We also trained the MPS tensor network varying the bond dimension in the network between 10 to 60. We found that the accuracy of training data reaches 100% with bonds of 50 and 60, the best result on the validation data set was 96.83% with a bond dimension of 30, however, this value was not appreciably greater compared with other bond dimensions, indicating that this hyperparameter parameter is not very influential for generalizing in the MNIST data set. In the training process the accuracy of the validation data set was stable and did not decrease significantly, indicating that the MPS tensor network is a form of regularizer as noted in [2].

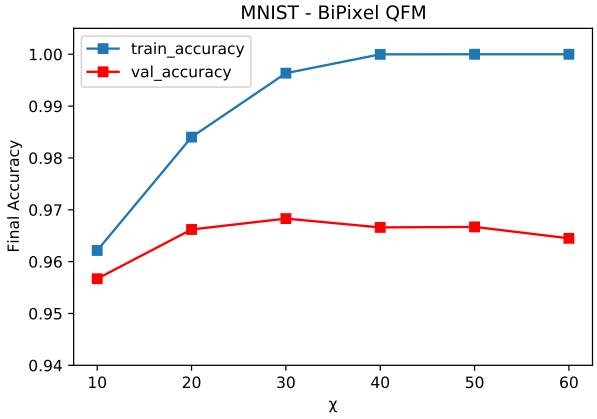

Fig. 9. Final accuracy after 100 epochs on training and validation of MNIST data set for various bond dimensions of the MPS tensor network

B. *Eng/Spa Words Data Set*

The Eng/Spa Words data set is composed of 19001 words with their labeled language, there are 9454 words in English and 9547 in Spanish. To perform the supervised learning we divided the data set in 12730 words for training and 6271 for testing.

To train the MPS tensor network, we used the 2-gram Quantum Feature Map, as described previously. In total there were 464 2-grams and, therefore, the MPS Network was composed of 464 sites.

We could have worked with other types of n-grams like (1,3)grams, however, we decided to explore only 2-grams in order to test whether the order of n-grams and the bond dimension affected the accuracy of the results of 2-grams.

To test if the order of the 2-grams modifies the accuracy, we worked with three types of 2-grams orderings; an alphabetical order of the 2-grams, a randomized order, and a customized order. The customized order consisted on creating a matrix which counts the number of occurrences of consecutive 2-grams, and placing 2 grams together in order, based on their co-occurrence.

In the following two graphs, we show the accuracy after 200 epochs in the test data set for bond dimensions given by the powers of 2, and multiples of 10.

We also trained the MPS tensor network 10 times for each point for the custom and random orderings. We found that the ordering of the 2grams does not appreciably affect the accuracy, but higher bond dimension leads to higher accuracy on test data as shown in the figures.

To compare the results of the MPS with state-of-the-art methods. We trained a Support Vector Machine with the Histogram Intersection Kernel over 2grams, in the same data set. In the following table we show the results of our best model and the SVM. The MPS approach outperformed the SVM.

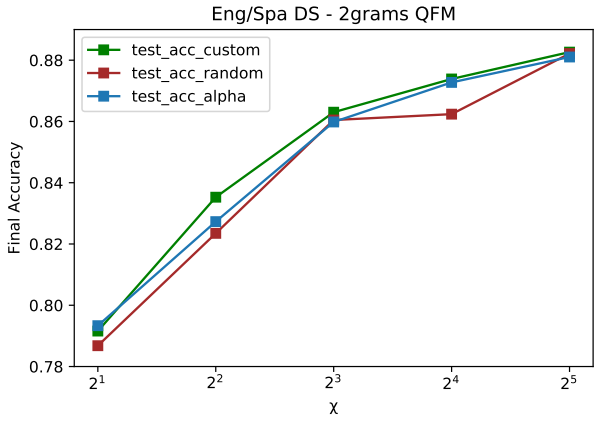

Fig. 10. Final accuracy in test data set after 200 epochs for the three types of orderings of the 2-grams, for bond dimensions in the powers of 2.

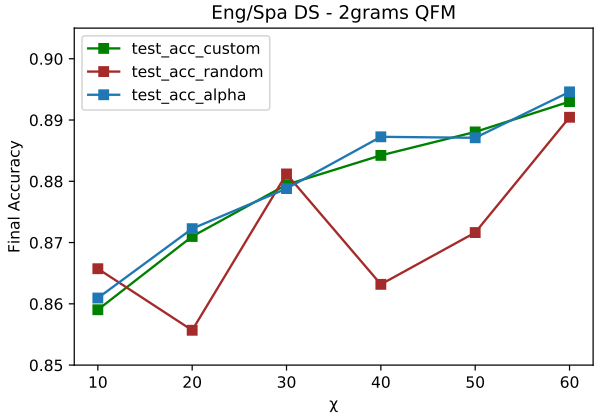

Fig. 11. Final accuracy in test data set after 200 epochs for the three types of orderings of the 2-grams, for bond dimensions in multiples of 10.

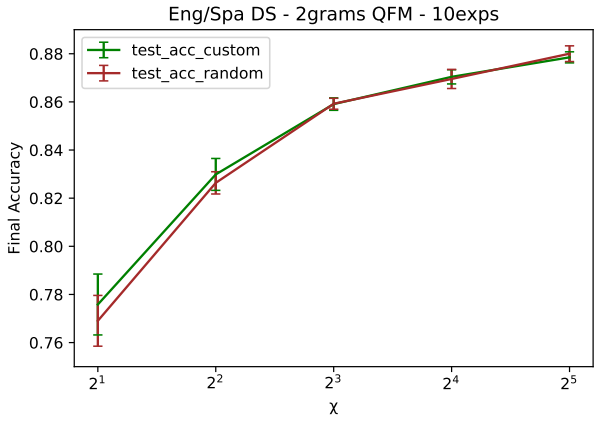

Fig. 12. Final accuracy in test data set after 200 epochs for 10 experiments per point for the custom and random orderings, for bond dimensions in powers of 2.

Accuracy on Test: Eng/Spa Words DS	
Model	**Accuracy**
MPS ($\chi = 60$)	89.45%
SVM HistoInter K	87.82%

C. MM-IMDb Posters Data Set

The MM-IMDb poster data set consists of movie posters, with the usual 3 color channels red, green and blue, the dimension of the posters are variable, so we applied a preprocess function to all images to normalize the data. The data was split (15552,2608,7799) for the training, validation, and test data sets respectively.

To solve the classification task we proposed two approaches, the first one consists on making a fusion between channels creating an image in grey scale using some constants that are commonly used in visual computing. The function applied in this fusion is,

$$X = R * 0.21 + G * 0.72 + B * 0.07$$

With the image in grey scale, we then applied the Binary Pixel Quantum Feature Map that was presented before. The process is depicted in the figure 13

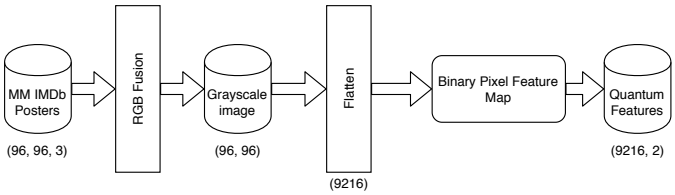

Fig. 13. Process to extract quantum features from rgb fusion grayscale images

In the following table, we show the 4 types of f-scores obtained by varying the bond dimension hyperparameter in the test data set.

F-Score on Test: MM-IMDb Plots - RGB Fusion				
χ	**macro**	**micro**	**weighted**	**samples**
5	0.0285	0.2330	0.1356	0.2288
10	0.0507	0.2144	0.1629	0.1862
15	0.0970	0.2255	0.2188	0.2000
20	0.1387	0.2023	0.2620	0.1641

We can see in general terms that the values of the f-scores progressively improve as the bond hyperparameter increases, however, the values of the loss function in the validation set do not have a descending behavior and grow as the hyperparameter value increases. This behavior can be seen in the figures 14 and 15

The second approach was based on transfer learning, the main idea of transfer learning is to use a pretrained Deep neural network, and then transfer the learned features to a new model, in our case a MPS model. To achieve this purpose, we used VGG16 as a base network to extract the features.

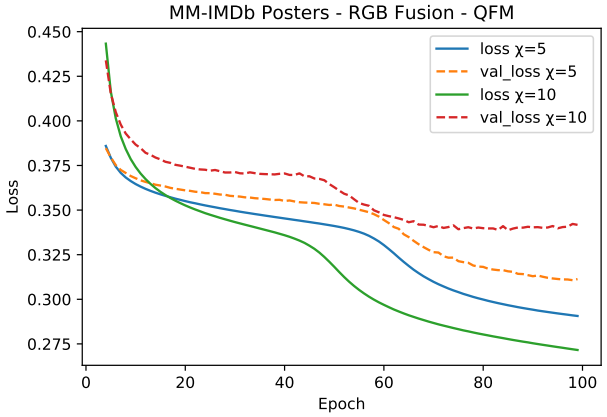

Fig. 14. Loss of training and validation data sets on MM-IMDb posters with RGB Fusion and QFM with a MPS Tensor Network for 100 epochs and bond dimensions 5 and 10.

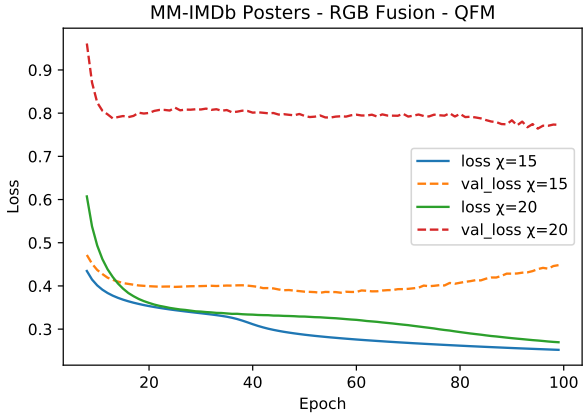

Fig. 15. Loss of training and validation data sets on MM-IMDb posters with RGB Fusion and QFM with a MPS Tensor Network for 100 epochs and bond dimensions 15 and 20.

After extracting the features, we applied a sigmoid activation function to normalize the data and then we applied the Binary Pixel Quantum Feature, as follows,

$$|\psi(x)\rangle = \begin{cases} |0\rangle & \text{if} \quad \sigma(x) < 0.5 \\ |1\rangle & \text{if} \quad \sigma(x) > 0.5 \end{cases}$$

The process of pre processing the raw data is depicted in figure 16

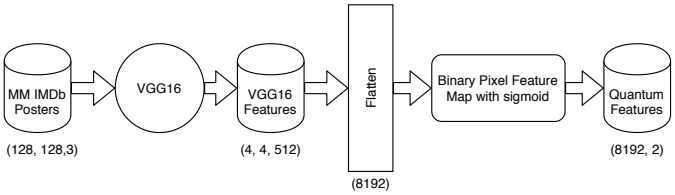

Fig. 16. Process to extract quantum features using a pre trained CNN

In the following table, we show the 4 types of f-scores obtained by varying the hyper-parameter in the test data set.

F-Score on Test: MM-IMDb Posters - FE				
χ	**macro**	**micro**	**weighted**	**samples**
5	0.0311	0.2354	0.1448	0.2248
10	0.0362	0.2323	0.1536	0.2146
15	0.0329	0.2364	0.1495	0.2250
20	0.1106	0.2547	0.2424	0.2275
25	0.1309	0.2465	0.2628	0.2153
30	0.1328	0.2354	0.2612	0.2168

We can see that the f-scores improved as we increased the bond dimension. However, after a bond of 20, the values of the loss function in the validation data set started to increase. The behavior of the training process is shown in figures 17 and 18.

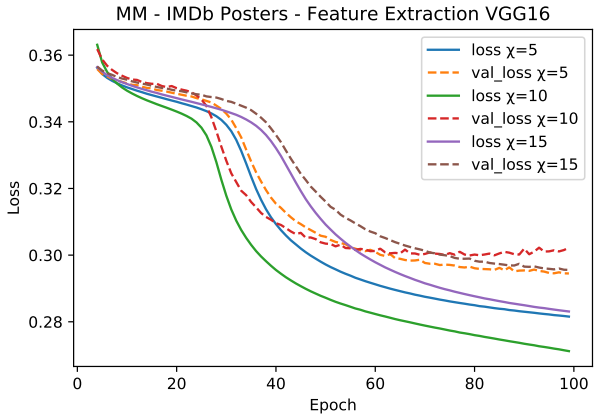

Fig. 17. Loss of training and validation data sets on MM-IMDb posters with feature extraction from VGG16 and QFM with a MPS Tensor Network for 100 epochs and bond dimensions 5, 10 and 15x.

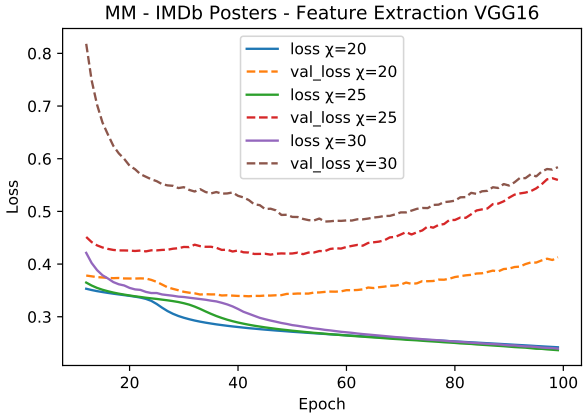

Fig. 18. Loss of training and validation data sets on MM-IMDb posters with feature extraction from VGG16 and QFM with a MPS Tensor Network for 100 epochs and bond dimensions 20, 25 and 30.

The following table, we summarize the f-scores for the best models, and the result of a previous works where we used transfer learning with fine tuning and (100, 50, 23) dense layers for this classification task. In general terms, the usual methods of fine tuning with dense layers performed better than the proposed MPS network probably because we lost some information by doing the quantum feature maps.

F-Score on Test: MM-IMDb Posters				
Model	**macro**	**micro**	**weighted**	**samples**
MobileNetV2 Fine Tuning	0.0847	0.3385	0.2363	0.3453
RGB Fussion	0.0507	0.2144	0.1629	0.1862
Feature Extraction VGG16	0.1106	0.2547	0.2424	0.2275

D. MM-IMDb Plots Data Set

In the MM-IMDb plots classification task, we took the Tfidf-3-gram representation computed by [28], and applied the Tfidf Quantum Feature Map described previously. The Tfidf QFM is a form of discrimination of the most representative 3-grams of each text. Each component of the 3-gram representation vector fed a site in the MPS tensor network, hence, there are as many 3-grams as sites in the network, the bond dimension was set to $\chi = 10$.

This multi-label classification problem had as an output of the MPS 23 neurons, accounting for each movie genre, the activation function was a sigmoid because a movie plot can belong to multiple genres (multi-label task).

The data was split (15552,2608,7799) for the training, validation, and test data sets respectively. In the following image, we show the training and validation loss during training.

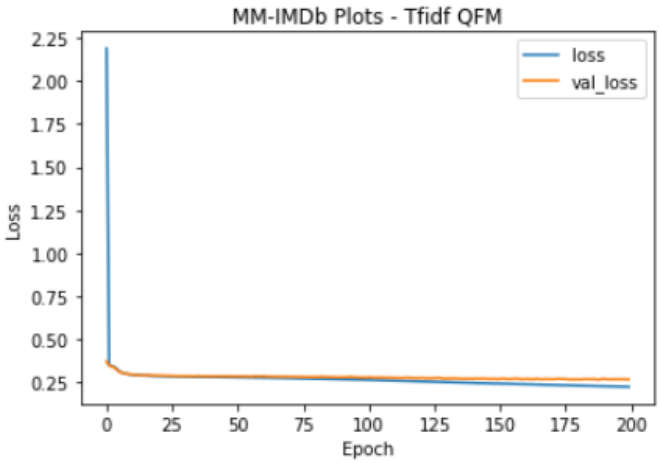

Fig. 19.

In the following table, we show the 4 types of f-scores (micro, macro, weighted, and samples) obtained on the training, validation, and test data sets.

F-Score: MM-IMDb Plots, MPS Tfidf				
Dataset	macro	micro	weighted	samples
Train	0.217	0.472	0.416	0.415
Val	0.169	0.389	0.339	0.351
Test	0.170	0.387	0.338	0.344

A LSTM recurrent neural network with a pretrained Word2Vec embedding [28] was also trained to compare the results of the MPS network. The recurrent network had 60 neurons, and was connected with (100, 50, 23) dense layers. The results on the test data set of both models are shown in the following table.

F-Score on Test: MM-IMDb Plots				
Model	macro	micro	weighted	samples
MPS Tfidf	0.170	0.387	0.338	0.344
LSTM w2v	0.266	0.478	0.434	0.471

In terms of the f-scores, the LSTM achieves higher results compared with the MPS tensor network. However, the MPS tensor network obtains acceptable results taking into account that some information of the representation of the documents was lost by means of the Tfidf QFM.

VI. Conclusions

We have trained the MNIST, the Eng/Spa Words, and the Multimodal IMDb data sets with the MPS tensor network framework developed by (Efthymiou et. al.) [27] which was in turn based on the (Stoudenmire et. al.) [2]. We have applied various types of quantum feature maps on the data over the pixels and the n-grams in the visual and textual classification problems respectively, obtaining comparable results with the state-of-the-art.

With respect to the MNIST data set, we applied the Binary Pixel quantum feature map, the best obtained result was 96.83% on test data with a bond dimension of $\chi = 30$. We found that the accuracy on test data does not improve for bonds dimensions above 30. We also found that the loss does not increase after various epochs, which might indicate that the MPS tensor network is a form of regularization as noted in [2].

The textual classification was based on n-grams, we have not found up-to-date previous works which uses the MPS tensor network on n-grams. In the Eng/Spa data set we applied the 2-gram quantum feature map, and we explored the bond dimension for various orderings of the n-grams. In general terms, we found that the accuracy on test data improves with higher bond dimensions, but the ordering of the 2-grams does not affect this accuracy. The best result on test data was 89.45% with a bond dimension $\chi = 60$, this result outperforms a SVM learning method with the histogram intersection kernel which obtained 87.82% on test data.

For the MM-IMDb posters, we explored two approaches, the first one was based on transforming the images on grayscale, the second one was based on learning from the features extracted from the pretrained VGG16 network. The worst

result obtained was the model that learns from the grayscale images, we believe that this is because the learning process is pixel by pixel, additionally some information was lost in the process of the fussion of the channels. The second approach took the features extracted from pre-trained model, this model obtained acceptable results, however, the f-scores were below state-of-the-art results [28]. We believe that the results of the MPS model were lower because the proposed quantum feature map lost some information on the process of activating or deactivating features with a fixed threshold.

For the MM-IMDb plots, we applied a quantum feature map to the Tfidf 3-grams vectors computed in [28]. We obtained acceptable results on the f-scores, but these f-scores were below a LSTM network with word-2-vec embedding. We believe that the results of the MPS were below the LSTM because some important information of the data was lost when we applied the quantum feature map to the original Tfidf vectors.

Some future efforts lie on reproducing the results with the MPS training framework and feature map developed by Stoudentmire et. al. [2]. Also to adapt the MPS tensor network to learn from sequential data in similar fashion as recurrent neural networks.

Link presentation Youtube

REFERENCES

[1] J. C. Bridgeman and C. T. Chubb, "Hand-waving and interpretive dance: An introductory course on tensor networks," 3 2017.

[2] E. M. Stoudenmire and D. J. Schwab, "Supervised Learning with Quantum-Inspired Tensor Networks," 5 2016.

[3] G. Evenbly and G. Vidal, "Tensor Network States and Geometry," *Journal of Statistical Physics*, vol. 145, pp. 891–918, 11 2011.

[4] F. Pollmann, E. Berg, A. M. Turner, and M. Oshikawa, "Symmetry protection of topological phases in one-dimensional quantum spin systems," *Physical Review B - Condensed Matter and Materials Physics*, vol. 85, 2 2012.

[5] X. Chen, Z. C. Gu, and X. G. Wen, "Classification of gapped symmetric phases in one-dimensional spin systems," *Physical Review B - Condensed Matter and Materials Physics*, vol. 83, 1 2011.

[6] D. Griffith, "Tensor Networks and The Ising Model," tech. rep., 2012.

[7] M. Fannes, B. Nachtergaele, and R. F. Werner, "Finitely correlated states on quantum spin chains," *Communications in Mathematical Physics*, vol. 144, pp. 443–490, 3 1992.

[8] S. Stlund and S. Rommer, "Thermodynamic limit of density matrix renormalization," *Physical Review Letters*, vol. 75, pp. 3537–3540, 3 1995.

[9] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac, "Matrix product state representations," *Quantum Information and Computation*, vol. 7, pp. 401–430, 8 2007.

[10] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, pp. 2295–2317, 1 2011.

[11] F. Verstraete and J. I. Cirac, "Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions," 7 2004.

[12] G. Sierra and M. A. Martin-Delgado, "The Density Matrix Renormalization Group, Quantum Groups and Conformal Field Theory," 11 1998.

[13] F. Verstraete, V. Murg, and J. I. Cirac, "Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems," 3 2008.

[14] Z. Y. Han, J. Wang, H. Fan, L. Wang, and P. Zhang, "Unsupervised Generative Modeling Using Matrix Product States," *Physical Review X*, vol. 8, 7 2018.

[15] J. A. Bengua, H. N. Phien, and H. D. Tuan, "Optimal Feature Extraction and Classification of Tensors via Matrix Product State Decomposition," in *2015 IEEE International Congress on Big Data*, pp. 669–672, 2015.

[16] A. Novikov, A. Rodomanov, A. Osokin, and D. Vetrov, "Putting MRFs on a tensor train," tech. rep., 2014.

[17] A. Cichocki, N. Lee, I. Oseledets, A. H. Phan, Q. Zhao, and D. P. Mandic, "Tensor networks for dimensionality reduction and large-scale optimization part 1 low-rank tensor decompositions," *Foundations and Trends in Machine Learning*, vol. 9, pp. 249–429, 9 2016.

[18] N. Cohen, O. Sharir, and A. Shashua, "On the Expressive Power of Deep Learning: A Tensor Analysis," 9 2015.

[19] A. Novikov, M. Trofimov, and I. Oseledets, "Exponential machines," *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 2, no. 1, pp. 789–797, 2018.

[20] A. J. Gallego and R. Orus, "Language Design as Information Renormalization," 8 2017.

[21] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, "Towards quantum machine learning with tensor networks," *Quantum Science and Technology*, vol. 4, p. 024001, 3 2019.

[22] E. Farhi and H. Neven, "Classification with Quantum Neural Networks on Near Term Processors," 2 2018.

[23] M. Schuld and N. Killoran, "Quantum Machine Learning in Feature Hilbert Spaces," *Physical Review Letters*, vol. 122, 2 2019.

[24] X. Gao, Z. Zhang, and L. Duan, "An efficient quantum algorithm for generative machine learning," 11 2017.

[25] M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz, "A generative modeling approach for benchmarking and training shallow quantum circuits," *npj Quantum Information*, vol. 5, 12 2019.

[26] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Physical Review A*, vol. 98, 9 2018.

[27] S. Efthymiou, J. Hidary, and S. Leichenauer, "TensorNetwork for Machine Learning," 6 2019.

[28] J. Arevalo, Solorio, M. Montes-Y-Gómez, and F. A. González, "Workshop track-ICLR 2017 GATED MULTIMODAL UNITS FOR INFORMATION FU-SION," tech. rep.

[29] R. Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states," 6 2014.

[30] C. Roberts, A. Milsted, M. Ganahl, A. Zalcman, B. Fontaine, Y. Zou, J. Hidary, G. Vidal, and S. Leichenauer, "TensorNetwork: A Library for Physics and Machine Learning," 5 2019.

[31] G. Mussardo, *Statistical field theory : an introduction to exactly solved models in statistical physics*. Oxford University Press, 2010.

[32] S. R. White, "Density matrix formulation for quantum renormalization groups," Tech. Rep. 19, 1992.