Gradient-based Interpretability of Quantum-inspired neural networks

Antoine Debouchage

November 13, 2022

Abstract

In the thriving world of physics and quantum computing, researchers have elaborated a multitude of methods to simulate complex quantum systems since the 90s. Among these techniques, tensor networks have an important role in the way they make sense of the compression of data to fit complex quantum systems in the memory of classical computers. The rapid expansion of tensor networks is due to the need to visualize and store physical structures.

In this paper, a tensor train decomposition of the linear layers of a simple Convolutional Neural Network has been implemented and trained on the dataset Cifar10. The observations show that the various attention images inferred on both a neural network and its tensor network equivalent has radically different and the models focus on different parts. Secondly, I proposed some considerations on miscellaneous gradient descent methods that can be used to specifically optimise tensor networks. Tensor networks evolve in a smooth Riemannian manifold, using Riemannian optimisation (RO) techniques to perform gradient descent geometrically. Indeed, the projections implied by the RO allow the traceability of the gradients and thus, easily reverse engineer a backpropagation.

The code implemented can be foud as well as the pdf version of all cited papers and images used in my github : https://github.com/antoine311200/Hackaton-Interpretability. Its usage is quite straight-forward, simply run all cells in both notebooks after having installed the requirements.

Contents

Introduction

Over the past few years, neural networks have reached top performance in a wide range of fields dethroning every previous algorithm known so far, especially in natural language processing and computer vision to name but a few. This leads to models with an ever-growing number of parameters getting as far as billions of trainable parameters. On the other hand, exponentially growing variables have been present for decades in the simulation of complex quantum systems[3]. Diverse algorithms were created to compensate for this issue by approximating tensors of high orders into the product of smaller ones: tensor networks[13 2 11 17]. Exponential growth is then constrained to a polynomial number of values. A vast variety of tensor networks have been developed such as Matrix Product States (MPS), Projected Entangled Pair States (PEPS) and Multiscale Entanglement Renormalization Ansatz (MERA) among the principals. They all use the approximation of huge tensors by discarding uninformative parameters through the study of singular values. It has been proved that tensor networks form a dense subspace of the Hilbert space where they evolved. Thus, one can hope for a huge compression ratio of a given high-order tensor by a corresponding tensor network up to an approximation error. The physical and quantum nature of these objects allow for a wide range of application from physics simulation to constrained optimisation and machine learning with complete performances. In this paper, we use a particular kind of tensor networks that are Matrix Product States (MPS) and their operator counterpart, Matrix Product Operators (MPO), to replace fully-connected layers in neural networks[2 18 12 19]. Indeed, fully-connected layers account for the majority of parameters in deep learning models. Thus, one would like to reduce their size without losing too much precision. Multiple methods already exist for a whole model such as distillation or layer fusion but it is quite hard to focus on a single layer. Here, we propose to compare the behaviour of a neural network with tensorised fully-connected layers (in the sense of their tensor train decomposition[13 11]) with its untensorised (standard neural network) version to highlight major changes in the way the network understands information through gradient descent (in a similar way to[4] but for quantum-inspired neural networks). It will be emphasized by the visualisation of two attention mechanisms for images: GradCam[14] for gradient-based localisations and SmoothGrad[7] for gradient-based saliency maps, showing how tensor networks are good candidates for future implementation in deep learning and especially in physics-informed deep learning where its link to the representation of quantum states and ability to represent symmetry groups is a top asset. Finally, we exposed theoretically several gradient-based approaches[20 16 10 2 ?] to optimize these tensor layers for future works where the interpretability of the neurons would be even more visible by the geometric nature of these algorithms.

1 Tensor train decomposition & Matrix Product States

Given a d-dimensional tensor $A_{i_1\cdots i_d}$, one can decompose it into a product of 3-dimensional tensors called a tensor-train decomposition or a Matrix Product State (MPS)[13]

$$A_{i_1\cdots i_d} \approx G_{1\,\alpha_0,i_1}^{\alpha_1} G_{2\,\alpha_1,i_2}^{\alpha_2} \cdots G_{1\,\alpha_{d-1},i_d}^{\alpha_d}$$

with $G_{k\,\alpha_{k-1},i_k}^{\alpha_k}$ being a $r_{k-1} \times n_k \times r_k$ array ($r_0 = r_d = 0$). This decomposition can be algorithmically done by performing a series of SVDs (or QR decomposition for better complexity) truncating the singular values to keep r_k values at each step. The same process can be apply to get an operator equivalent, that is for a tensor with input indices and output ones.

$$B_{i_1\cdots i_d}^{j_1\cdots j_d} \approx G_{1\,\alpha_0,i_1}^{\alpha_1,j_1} G_{2\,\alpha_1,i_2}^{\alpha_2,j_2} \cdots G_{1\,\alpha_{d-1},i_d}^{\alpha_d,j_d}$$

with $G_{k\,\alpha_{k-1},i_k}^{\alpha_k,j_k}$ being a $r_{k-1} \times n_k \times m_k \times r_k$ array ($r_0 = r_d = 0$).

When subjected to a weight matrix W of size $n \times m$, one can decide to arbitrarily reshape it into a $2d$-order tensor of size $(n_1 \times \cdots \times n_d) \times (m_1 \times \cdots \times m_d)$ with $\prod_i n_i = n$ and $\prod_j m_j = m$. Then, performing a tensor-train decomposition gives the possibilty to approximate the weight matrix by an

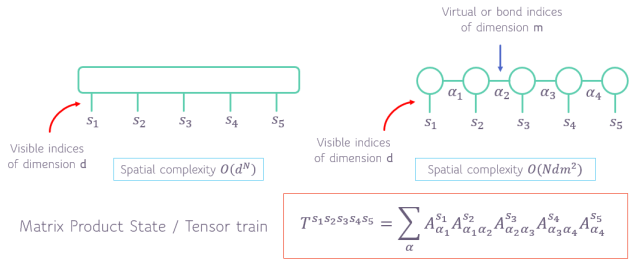

Figure 1: Matrix Product State diagram

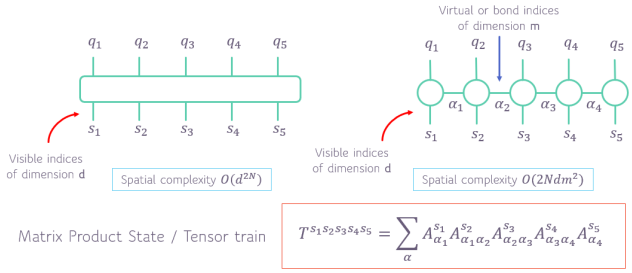

Figure 2: Matrix Product Operator diagram

MPS. Hence, converting a general neural network into a tensorised neural network is not a big deal.

Tensor train decomposition has been widely used by some niche researchers and they provide a way to compress the information and reduce the total number of parameters by a huge amount in the dense layers allowing to convert a 300 millions parameters Transformer to a 100 millions one with almost no losses (https://blog.tensorflow.org/2020/02/speeding-up-neural-networks-using-tensornetwork-in-keras.html).

Now, here conversion is not needed as directly initialising an random MPS instead of a random weight matrix and optimising it is the simplest approach both spatially and temporarily. In this way, from a simple Convolutional Neural Network, all fully-connected layers are replaced by their tensor dense equivalent (an MPO to account for the operator-ness of the weight matrix of dense layers).

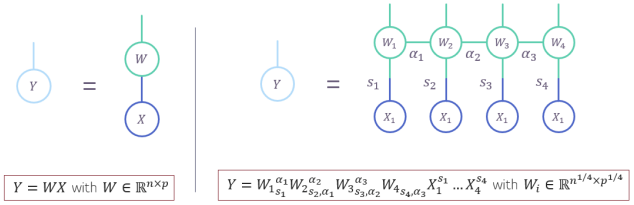

Figure 3: Tensor diagram of a dense layer vs a tensor dense layer

2 Safety & Alignment

Previous works on alignment on fully-connected layers[1] have been realised and proven remarkable results. Alignment is a form of implicit regularization in linear neural networks under gradient descent. There always exists a global minimum corresponding to an aligned solution for the linear layer. Thus, a tensor train decomposition is a decomposition of a linear layer into a more compact form under a dense space of the global Hilbert space considered the result in the paper[1] is straightforward. Nonetheless, it has been shown that a fully connected layer under gradient descent while fulfilling the definition of alignment given by these previous works does not generalize as an invariant under a constrained layer and suggests that better approaches might be necessary to understand exactly the implicit regularization. Hence, our understanding of a tensorised neural network under gradient descent is partially given by the previous proofs. Furthermore, with the result that I was able to get below, future work must extend the notion of alignment on tensorised neural networks as they may be key to understanding the implicit that fully connected layers cannot entirely provide.

As for safety, the global concern in today's world is how can we face the quantum revolution and its threat of decrypting the classical RSA algorithm through the Shor algorithm. Even though there is still plenty of time before anything can threaten our data (modern quantum computing are barely giving proper results and are extremely constrained), people need to focus on the potentiality of quantum attacks that could well reverse engineer easily neural network and endangered the world of AI. Indeed, quantum-inspired algorithms and methods such as tensor networks can be extended to practically any numerical and simulation field as has been the case with tensor trains for machine learning and their ability to simulate quantum circuits efficiently under classical machines may be problematic for neural networks. Thus, ensuring the use of tensorised layers until more complex ones are created that account for more safety is primordial and the cornerstone of the future of quantum-inspired AI. As we will show below, with tensorised layers gradients are more localised and using proper gradient descent such as tangent space gradient optimisation (TSGO)[20] and Riemannian optimisation of

submanifolds of fixed TT-rank[16] could improve the interpretability of these models. It would permit the creation of complex quantum-inspired neural networks given total interpretability, explainability, control and safety of everything that is happening under the hood lifting the black-box approach with white-box models locked by safe mechanisms against any kind of attacks.

3 Gradient-based Visualisation methods

3.1 Interpretability visualisation Methods

Using the Cifar10 dataset with a toy model composed of three blocks of (Conv2D-BatchNorm-Conv2D-BatchNorm-MaxPooling2D-Dropout) with sizes respectively 32, 64 and 128 followed by a feed forward network containing our fully-connected layers for the neural network model and tensor dense layers for the tensorised neural network. The model is trained for 50 epochs with an Adam optimiser with the results of the sparse categorical cross-entropy loss and accuracy in Figure [4] and [5].

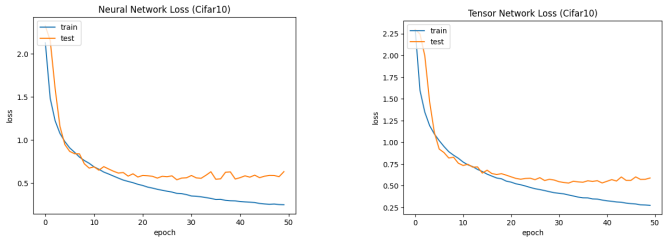

Figure 4: Training and Evaluation loss of both standard and tensorised neural network

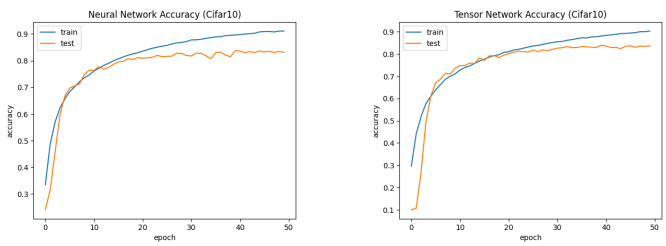

Figure 5: Training and Evaluation accuracy of both standard and tensorised neural network

An interesting result that has not been yet published and not proved by what I observed during my previous work on tensorised neural networks is that the loss is always converging faster for the tensorised network. This is a hint to explore future work on the explainability of tensor network models. Combining this statement with how gradient-based optimisation works for tensor networks might lead to stunning results in the mechanistic interpretability of tensorised neural networks.

The idea is to see how different tensorised neural networks are from their standard sibling. For this purpose, the GradCAM (Gradient-weighted Class Activation Mapping) technique for producing visual explanations for decisions has been used to create a heatmap of the main pixels acknowledging the class predictions of an image for the given model. It uses the gradients flowing into the final convolutional layer to produce an unrefined localization map emphasising important regions in the image for a particular class prediction. Secondly, we used SmoothGrad[7] to compute saliency maps of given images with respect to some target classes. SmoothGrad creates noisy copies of an input image and average gradients highlighting areas of visual importance in the picture by sharpening the resulting saliency map and removing irrelevant noisy regions. We used the tf-keras-vis Python package to visualize efficiently and utilise GradCAM and SmoothGrad.

Let's take a batch of Cifar10 images (Figure 6) and process both the models through GradCAM and SmoothGrad.

Figure 6: A batch of Cifar10 images

3.2 Results - GradCAM

The results are shown for the batch of images in Figures [8] and [10] for the sample batch and in

Figures [13] and [15] for a sample of 16 images of frogs (Figure [11].

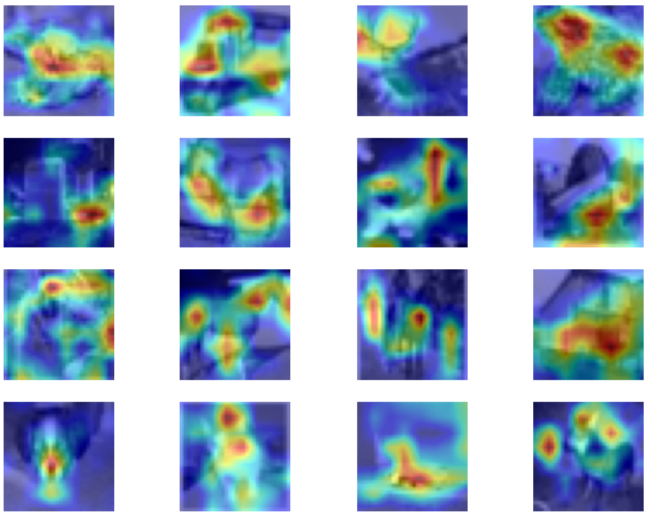

Figure 7: Neural Network GradCAM of the sample batch

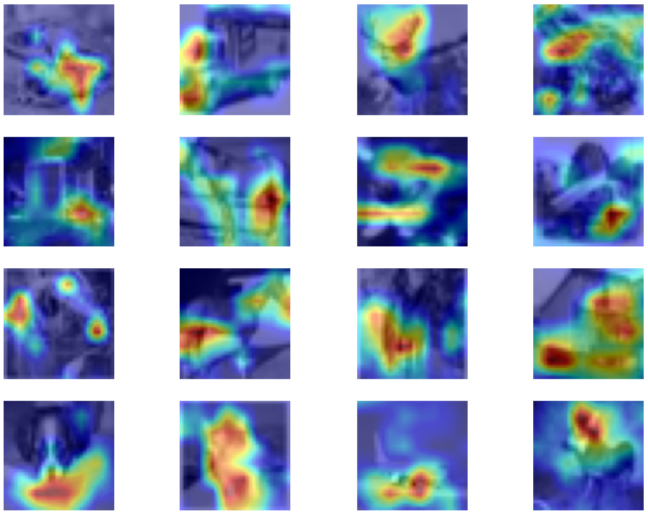

Figure 8: Tensor Network GradCAM of the sample batch

Our observations show that in the GradSCAM method[14], the gradients are less localised for the standard neural network with attention focused on several parts of the images with spreading areas and dissipated gradients. Contrastingly, the tensorisation of the neural network induced a more localised gradient heatmap with high values on specific parts of the image that correspond more to the actual attention mechanism of human vision. Indeed, taking the bottom-right images of the deer, it is clear that the gradients for the neural network are less interpretable than the one of the tensorised version which focus on the head and antlers of the Cervidae.

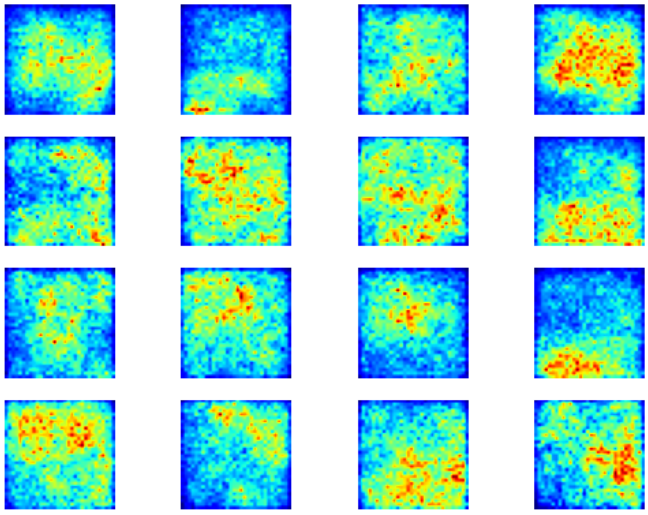

Figure 9: Neural Network SmoothGrad of the sample batch

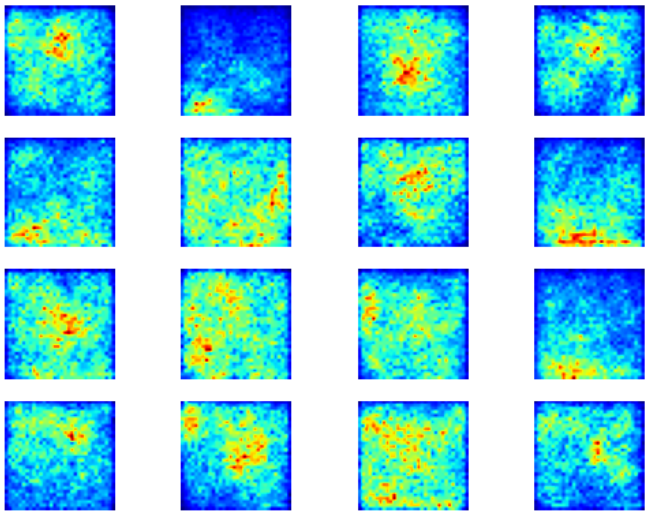

Figure 10: Tesnor Network SmoothGrad of the sample batch

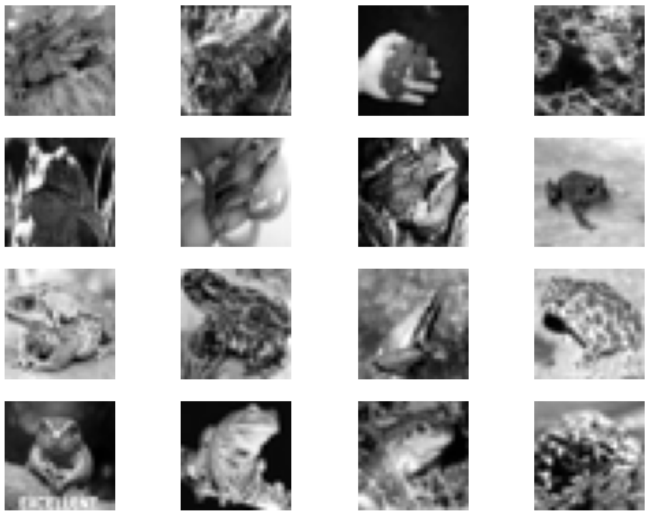

Figure 11: A batch of frog images

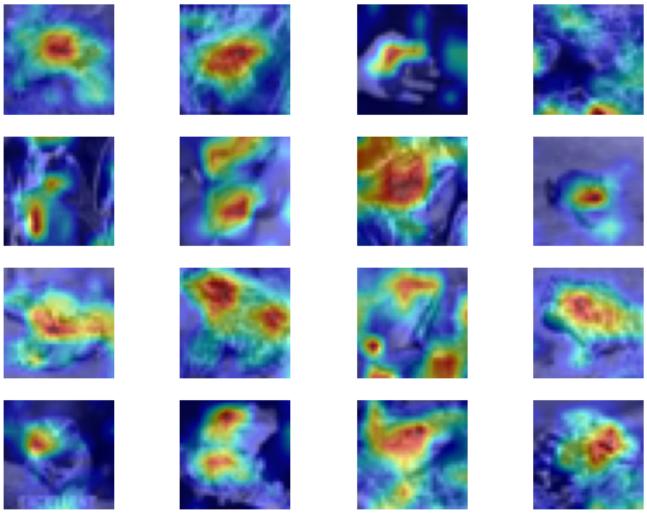

Figure 12: Neural Network GradCAM of frogs

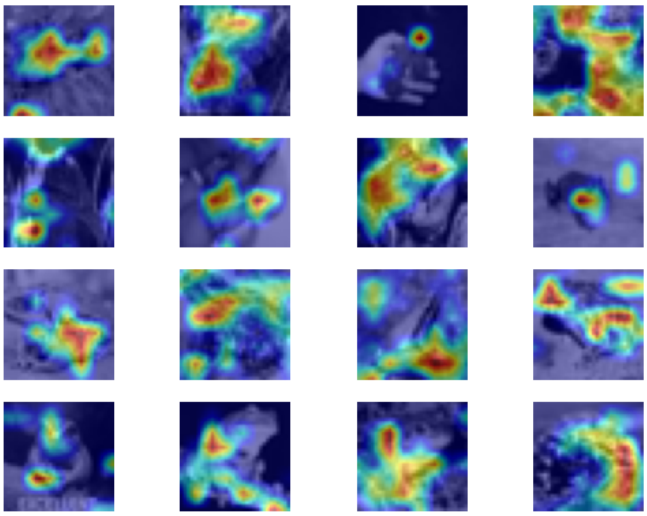

Figure 13: Tensor Network GradCAM of frogs

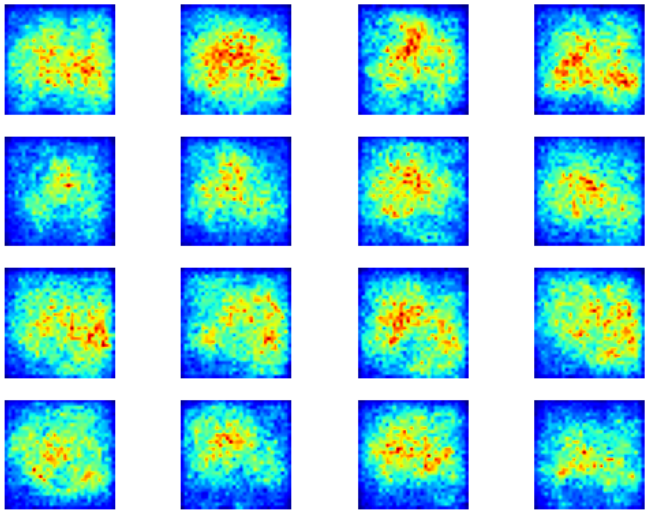

Figure 14: Neural Network SmoothGrad of frogs

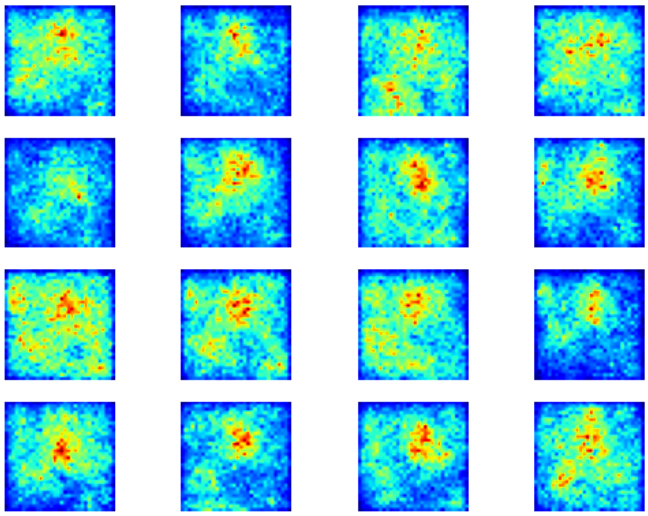

Figure 15: Tensor Network SmoothGrad of frogs

3.3 Results - SmoothGrad

In the SmoothGrad method, we aim at getting a saliency map which is an image that highlights the region on which people's eyes focus first. The same phenomenon as for GradCAM is to be observed with fewer spread gradients on the tensorised neural network. Moreover, we see a high correlation between areas highlighted in GradCAM and SmoothGrad, thus, the flow of the gradients through the tensorised layers is more controlled..

3.4 Results - Conclusion

By the very nature of tensor trains, with several cores encapsulating all the information, we can expect a better distribution of the weights with entanglement (in the quantum physical sense). Thus, the previous results conclude that the distribution of information in the different cores enables the model to better grasp the accurate part of the images that clarify the predictions. The localisation of the gradients on important parts of the images in contrast to classical neural networks which give a widespread with a lot of uninteresting gradients give a better understanding of the model predictions and can provide an efficient way to compress the gradient information to make distillations of models in the future.

4 Future perspectives

In this work, we studied the flowing of gradients in a standard neural network and tensorised neural network with an Adam optimiser to train the model. However, more theoretically accurate gradient optimisation methods exist for tensor networks and would be interesting to attempt. On

top of that, the study of how the Von Neumann entropy of each tensorised layer evolved and is spread for each core and compare them to the entropy of the weight matrices in standard fully-connected layers.

$$S(\rho) = -tr\left(\rho \ln \rho\right)$$

It is simple to compute it given the fact that there is a low entanglement in the underlying system (which is the case as we are not dealing with the superposition of states here and only the one induced by the noise and the optimisation process are present). It suffices to compute each reduced density matrices for each core of the tensor train (see Figure [16]).

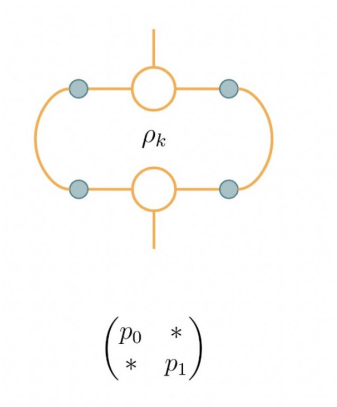

Figure 16: A single-core reduced density matrix

4.1 DMRG-like optimisation

The first optimisation algorithm that comes to mind is DMRG-like optimisation[15,18] where we optimise both tensor cores side by side from the left to the right of the MatrixProduct State in a sweep-like fashion. It allowed us to reduce the global complexity because we had to contract each index to result in a tensor output in each dense tensorised layer in our model. This accounts for much of the training time.

4.2 Tangent Space Gradient optimisation

Tangent spaces are inescapable in the context of tensor networks when dealing with complex optimisation algorithms. It has been shown[20] that the gradient of a quantum state in a Hilbert space can be restricted in the tangent space of a one-dimensional hyper-sphere with a learning rate that is naturally determined by the angle of the gradients $\eta = \tan\theta$.

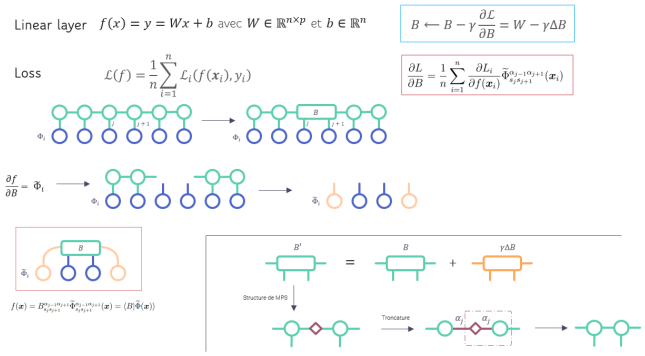

Figure 17: One optimisation step of the DMRG-like gradient update

4.3 Riemanian optimisation

The set of all d-dimensional tensors that can be entirely represented[16] with a fixed tensor train rank of $r = (r_1, \cdots, r_d)$

$$\mathcal{M}_r = \left\{W \in R^{n_1 \times \cdots \times n_d},\ tt - rank = r\right\}$$

forms a Riemannian manifold. This allows us to use Riemannian optimisation[5] to train tensor networks. THe Riemannian optimisation in the context of tensor trains is quite simple to explain (but more arduous to implement)

$$W_{k+1} = ttr\left(W_k - \lambda P_{T_{\mathcal{M}_r}(W)}\left(\frac{\partial \mathcal{L}}{\partial W}\right), r\right)$$

where $P_{T_{\mathcal{M}_r}}$ is the projection operator onto the tangent space in $\mathcal{M}_r$ of W and $ttr(\cdot, r)$ is the function that decreases the tt-rank of a tensor train to r.

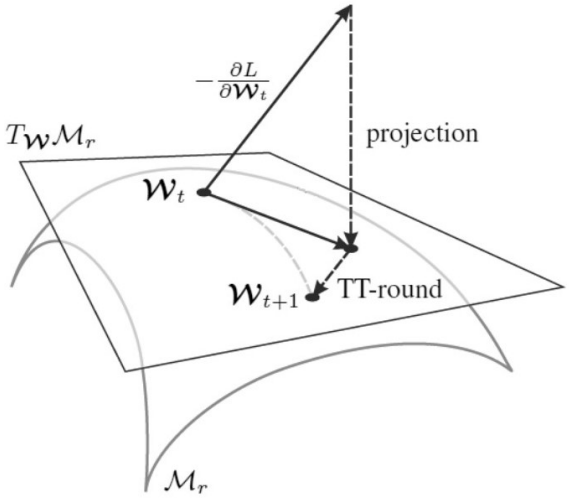

Figure 18: Visual representation of Riemannian gradient descent[18]

Conclusion

We showed how a tensorised neural network and its standard version result in different learning results with the tensorised one having a faster loss convergence for a decrease in trainable parameters $498,282$ vs $551,146$. Using more advanced research on tensorised deep learning, one can hope to replace the CNN layers with a tensorised version and reduce the total number of parameters drastically. On top of that, the interpretability of the attention mechanism of Grad-CAM and SmoothGrad through the flow of gradients in the model indicates that tensorised neural networks are a good candidate for mechanistic interpretability, explainability and model compression of neural networks in the future while being well-performing for physics-based models with its ease to encompass quantum information (because they have been developed for this purpose first) and to restrain tensors to symmetry groups that are utterly important in quantum physics.

Different optimisation algorithms might be interesting to employ to see if their logical nature to boast with tensor networks would acknowledge an even greater interpretation of the networks.

In the field of Natural Language Processing where Transformers are reigning supreme, the tensorisation of Transformer is a hot topic because of the high number of dense layers that could be converted leading to a reduced model with high accuracy.

5 Acknowledgements

This research report has been made in the context of the Interpretability Hackaton for Alignment Jam 2 hosted by Esben Kran, Apart Research and Alignmentjam. I, thereby gratefully thank all of them for the organisation of this Jam that was extremely fun to join and participated in.

References

[1] Daniel Bernstein Caroline Uhler Adityanarayanan Radhakrishnan, Eshaan Nichani. On alignment in deep linear neural networks. https://arxiv.org/abs/2003.06340.

[2] Anton Osokin Dmitry Vetrov Alexander Novikov, Dmitry Podoprikhin. Tensorizing neural network. https://arxiv.org/abs/1509.06569.

[3] Ivan Oseledets Alexander Novikov, Mikhail Trofimov. Exponential machines. 2015. https://arxiv.org/abs/1605.03795.

[4] Simon J. Julier Amy Widdicombe. Gradient-based interpretability methods and binarized neural networks. 2021. https://arxiv.org/abs/2106.12569.

[5] Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. 2011. https://arxiv.org/abs/1111.5280.

[6] Bart Vandereycken Christian Lubich, Ivan Oseledets. Time integration of tensor trains. 2014. https://arxiv.org/abs/1407.2042.

[7] Been Kim Fernanda Viégas Martin Wattenberg Daniel Smilkov, Nikhil Thorat. Smoothgrad: removing noise by adding noise. 2017. https://arxiv.org/abs/1706.03825.

[8] Been Kim Fernanda Viégas Martin Wattenberg Daniel Smilkov, Nikhil Thorat. Smoothgrad: removing noise by adding noise. 2017. https://arxiv.org/abs/1706.03825.

[9] Jimmy Lei Ba Diederik P. Kingma. Adam: A method for stochastic optimisation. 2015. https://arxiv.org/abs/1412.6980.

[10] David J. Schwab E. Miles Stoudenmire. Supervised learning with quantum-inspired tensor networks. 2017. https://arxiv.org/abs/1605.05775.

[11] Patrick Gelß. The tensor-train format and its applications. 2017. https://www.semanticscholar.org/paper/The-Tensor-Train-Format-and-Its-Applications-Gel%C3%9F/0d1eefdcaa5b994b2f2719d5282f861a987c008a.

[12] A. Cichocki J. M. T. Romano M. Nazareth da Costa, R. Attux. Tensor-train networks for learning predictive modeling of multidimensional data. 2021. https://arxiv.org/abs/2101.09184.

[13] Ivan Oseledets. Tensor-train decomposition. 2011. https://www.researchgate.net/publication/220412263_Tensor-Train_Decomposition.

[14] Abhishek Das Ramakrishna Vedantam Devi Parikh Dhruv Batra Ramprasaath R. Selvaraju, Michael Cogswell. Grad-cam: Visual explanations from deep networks via gradient-based localization. 2019. https://arxiv.org/abs/1610.02391.

[15] Ulrich Schollwoeck. The density-matrix renormalization group. 2004. https://arxiv.org/abs/cond-mat/0409292.

[16] Reinhold SchneiderReinhold Schneider Sebastian HoltzThorsten Rohwedder, Thorsten Rohwedder. On manifolds of tensors of fixed tt-rank. https://www.researchgate.net/publication/227007498_On_manifolds_of_tensors_of_fixed_TT-rank.

[17] Andreas Swoboda Salvatore R. Manmana Ulrich Schollwöck Claudius Hubig Sebastian Paeckel, Thomas Köhler. Time-evolution methods for matrix-product states. 2015. https://arxiv.org/abs/1901.05824.

[18] Kenneth Stoop. Machine learning with tensor networks. https://libstore.ugent.be/fulltxt/RUG01/002/782/897/RUG01-002782897_2019_0001_AC.pdf.

[19] Danilo P. Mandic Yao Lei Xu, Giuseppe G. Calvi. Tensor-train recurrent neural networks for interpretable multi-way financial forecasting. 2021. https://arxiv.org/abs/2105.04983.

[20] Gang Su Zheng-zhi Sun, Shi-ju Ran. Tangent-space gradient optimization of tensor network for machine learning. 2020. https://arxiv.org/abs/2001.04029.