Position: Tensor Networks are a Valuable Asset for Green AI

Memmel, Eva; Menzen, Clara; Schuurmans, Jetze; Wesel, Frederiek; Batselier, Kim

Publication date
2024

Document Version
Final published version

Published in
Proceedings of Machine Learning Research

Citation (APA)
Memmel, E., Menzen, C., Schuurmans, J., Wesel, F., & Batselier, K. (2024). Position: Tensor Networks are a Valuable Asset for Green AI. *Proceedings of Machine Learning Research*, *235*, 35340-35353.

Position: Tensor Networks are a Valuable Asset for Green AI

Eva Memmel [1] **Clara Menzen** [1] **Jetze Schuurmans** [2] **Frederiek Wesel** [1] **Kim Batselier** [1]

Abstract

For the first time, this position paper introduces a fundamental link between tensor networks (TNs) and Green AI, highlighting their synergistic potential to enhance both the inclusivity and sustainability of AI research. We argue that TNs are valuable for Green AI due to their strong mathematical backbone and inherent logarithmic compression potential. We undertake a comprehensive review of the ongoing discussions on Green AI, emphasizing the importance of sustainability and inclusivity in AI research to demonstrate the significance of establishing the link between Green AI and TNs. To support our position, we first provide a comprehensive overview of efficiency metrics proposed in Green AI literature and then evaluate examples of TNs in the fields of kernel machines and deep learning using the proposed efficiency metrics. This position paper aims to incentivize meaningful, constructive discussions by bridging fundamental principles of Green AI and TNs. We advocate for researchers to seriously evaluate the integration of TNs into their research projects, and in alignment with the link established in this paper, we support prior calls encouraging researchers to treat Green AI principles as a research priority.

1. Introduction

More than ever, we have access to data sets throughout almost all science and engineering disciplines. Fueling our economies and shaping our society, data is therefore considered the oil of the 21st century. At the same time, AI algorithms become increasingly powerful to transform large amounts of raw data into valuable information. Consequently, AI development and data availability are deeply intertwined, and they have a common characteristic: both are growing exponentially (Wu et al., 2021). While this wealth of information is opening the doors for extraordinary opportunities, the downside of this development cannot be ignored: AI research on a large scale has adverse side effects on economic, social, and environmental sustainability. Let us consider the example from Strubell et al. that analyzes the energy required for training popular off-the-shelf natural language processing models (2019). The training causes CO_2 emissions up to $280\,000\,\mathrm{kg}$ and cloud computing costs up to 3 million dollars. As a comparison, a person could fly more than 300 times between Amsterdam and New York to emit the same amount of CO_2 (myclimate, 2024).

On another note, Schwartz et al. argued that one of the reasons for the unsustainable development in AI research is unfortunately anchored in what the AI community commonly defines as state-of-the-art results (2020), namely focusing on accuracy or similar measures. They claim that to obtain more accurate results, the number of model parameters and hyperparameters is increased, as well as the size of the training data. The result is an exponentially growing demand for compute used to train AI models (Schwartz et al., 2020). This development is not only unsustainable from an environmental and economic point of view but also from a social one: this exponential growth has a large carbon footprint, is expensive, and excludes researchers with fewer resources (Schwartz et al., 2020; Ahmed et al., 2023).

The problems associated with the unsustainable development of AI have led to a growing awareness in the AI community. Several researchers call for redirecting the focus of AI research by implementing efficiency as an additional benchmark alongside accuracy to assess algorithmic progress (Schwartz et al., 2020; Strubell et al., 2020; Tamburrini, 2022). In fact, efficiency has always been the primary criterion to measure algorithmic progress in computer science (Knuth, 1976; 1973; Cormen et al., 2022). Inspired by this, a similar approach can be adapted to AI algorithms. For this purpose, different metrics have been proposed in the literature, e.g. estimating the carbon footprint, reporting the energy consumption, or stating the number of floating-point operations (FLOPS) (Lacoste et al., 2019; Henderson et al., 2020; Lannelongue et al., 2021). In this context, a new vocabulary has been suggested in (Schwartz et al., 2020). It

[1]Delft Center of Systems and Control, Delft University of Technology, Delft, The Netherlands [2]Xebia Data, Amsterdam, The Netherlands. Correspondence to: Eva Memmel <e.m.memmel@tudelft.nl>, Kim Batselier <k.batselier@tudelft.nl>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

distinguishes between AI focusing solely on accuracy versus AI considering efficiency and accuracy as equal criteria: Red AI versus Green AI.

With this paper, we want to contribute to the emerging research field into which practices and methods are suitable for Green AI (Verdecchia et al., 2023; Yarally et al., 2023). Green AI methods can roughly be organized into three categories: AI model development, computing infrastructure and system level (Wu et al., 2021; Kaack et al., 2022). AI model development focuses on reducing energy consumption during data processing, experimentation, training and inference. Computing infrastructure methods aim to, e.g., reduce the environmental impact of computing hardware or data centres. Methods at the system level are, e.g., policy considerations or technology management (Wu et al., 2021; Kaack et al., 2022). A holistic approach with a portfolio of different methods is critical to achieve a broad effect for Green AI. An in-depth review can be found in several surveys (Wu et al., 2021; Kaack et al., 2022) .

For the first time, we discuss an established tool from multilinear algebra from a Green AI perspective: Tensor Networks (TNs), also called tensor decompositions (Kolda & Bader, 2009). TNs fall under the category of AI model development. There, they stand alongside numerous strategies in the category AI model development, such as regularization, automated hyperparameter search (Yang & Shami, 2020; Bischl et al., 2023), pruning (Reed, 1993; Frankle & Carbin, 2018), quantization (Gholami et al., 2022), physics informed learning (Cuomo et al., 2022) or drop-out methods (Li et al., 2022). Most of these methods leverage different properties and do consequently not compete with each other. Instead, they can be combined to provide amplified benefits. To implement a holistic portfolio, we suggest prioritizing understanding the strengths and limitations of individual methods for Green AI. One strength of TNs is that they approximate data in a compressed format while preserving the essence of the information. In this way, they often achieve logarithmic compression while having the potential to achieve competitive accuracy. This makes TNs precisely the type of tool to apply to large-scale and/or high-dimensional problems: they allow for an efficient and, thus, sustainable way of representing and handling big data (Cichocki, 2014). Various promising results regarding accuracy and efficiency, as, e.g. (He et al., 2018; Izmailov et al., 2018; Richter et al., 2021; Wesel & Batselier, 2021), show their potential.

This potential is well-recognized in the TN community, as evidenced, e.g., by numerous comprehensive surveys (Cichocki et al., 2016; 2017; Panagakis et al., 2021; Sengupta et al., 2022; Wang et al., 2023). Although the emphasis and motivation in the TN community so far have not been on promoting the sustainability of AI, the existing technical

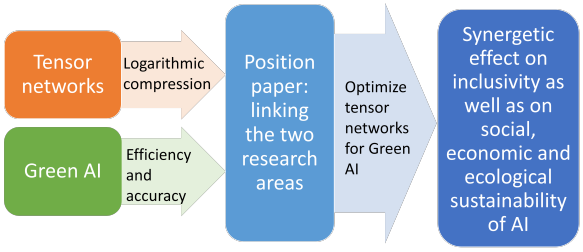

Figure 1. This position paper underscores the importance of connecting the research fields of TNs and Green AI. By establishing this link for the first time, we aim to achieve two primary goals. Firstly, to encourage the TN community to embrace Green AI practices and consciously tailor their TN models for enhanced sustainability. Secondly, to motivate the AI community to adopt Green AI practices and to explore the integration of TNs in their research. We believe that the combined application of Green AI practices and TNs will not only foster the social, economic, and ecological sustainability of AI models but also enhance the inclusivity and diversity of AI research. Ultimately, this synergy is expected to amplify the positive impact of AI research as a whole.

contributions are a solid foundation for the arguments presented in this position paper. **This position paper argues that TNs are a valuable asset for Green AI.** To make a substantial impact on Green AI, prioritizing and actively optimizing for sustainability is essential (Schwartz et al., 2020). As shown in Figure 1, by establishing the link between research on Green AI and TNs for the first time, this position paper aims to encourage interested AI researchers to adopt TNs and Green AI practices in their future work. To that end, we provide an economic, social, and environmental analysis of the possibilities and challenges of Green AI as well as the potential of TNs for Green AI. Understanding the impact of TNs on Green AI is not trivial. To the best of the authors' knowledge, this is the first paper that highlights TNs from a Green AI perspective. To support our position, the paper is organized as follows: In Section 2, we underscore the importance of Green AI research by exploring the detrimental effects of the exponentially growing computational demand reported in the literature. We also lay the groundwork for analyzing selected TN examples by reviewing a range of efficiency metrics suggested in Green AI literature. In Section 3, we introduce commonly used TNs and their logarithmic compression potential. In Section 4, we provide evidence from the literature on how applying TNs in kernel machines and deep learning contexts can lead to efficiency gains. Finally, in Section 5, we summarise our findings, demonstrate the benefits of linking TN and Green AI research, critically evaluate our stance and point towards promising future research directions.

2. Green AI in Related Literature

In their pioneering work, Schwartz et al. point out that AI models are commonly considered state-of-the-art when they achieve greater accuracy (or similar measures) than previously reported (2020). To incentivize competition and thus accelerate AI innovation, results are made public on leaderboards based on accuracy metrics. Schwartz et al. assert that beating the state-of-the-art algorithm is often achieved by at least one of the following three aspects: more extensive data sets, a more complex model, which is highly correlated with the number of parameters, or more extensive hyperparameter experiments (2020). This results in an exponentially growing demand for compute to train AI models (Amodei & Hernandez), requiring an unsustainable amount of hardware, energy, and computational time.

We review the related literature in two parts. First, we summarise how existing literature views the negative impact of increasing computation on sustainability in general and AI progress in particular. Second, we highlight metrics proposed in the literature for measuring efficiency or estimating the environmental impact of AI models.

2.1. The Negative Impact of Growing Compute on Sustainability and AI Progress

Sustainability is based on three fundamental, intersecting dimensions: economic, social, and environmental (Purvis et al., 2019). Research indicates that the rapidly increasing computational demands in AI have the potential to challenge these three dimensions. The implications of this are explored in the subsequent discussion.

As large computations can have a hefty price tag (Strubell et al., 2019; 2020), they negatively impact both the economic and social dimensions of sustainability. Concerning the **economic** dimension, linear gains in accuracy contrast with an exponentially growing amount of compute (Schwartz et al., 2020). Thus, diminishing returns contrast with increasing costs associated with e.g. cloud computing and hardware (such as CPU, GPU, and TPU). In the **social** dimension, high costs can create barriers for researchers with fewer resources to participate in computationally expensive AI development. The barriers for researchers with fewer resources (academia typically has fewer resources than industry) are reflected in the drop of academic contributions to large-scale AI results from 89% in 2010 to 4% in 2021, with the rest claimed by industry (Ganguli et al., 2022; Ahmed et al., 2023). Beyond the financial aspect, being dependent on external cloud compute providers can be problematic, too (Ahmed et al., 2023). For example, computations may need to be performed on-site in applications where privacy or security-relevant data is handled. Therefore, state-of-the-art computations with sensitive data may only be possible if enough financial resources are available to provide for large amounts of expensive hardware. Concerning the **environmental** aspect, several studies show that AI research causes a substantial carbon footprint (Strubell et al., 2020; Brown et al., 2020; Dodge et al., 2022). The emissions are attributable to operational emissions, associated, e.g., with cloud computing, and embodied emissions, associated, e.g., with hardware. In case operational emissions are decreased by relying on electricity with a low carbon intensity, embodied emissions make up for the largest share (Gupta et al., 2021). In fact, a recent study shows that more than 50% of Meta's emissions are attributed to embodied costs (Wu et al., 2021).

Beyond sustainability, an exponentially growing computational need can cause an additional problem. When computational needs cannot be met anymore, they will negatively impact progress and innovation in AI. Until now, progress and innovation in AI strongly rely on the increase of available computational resources (Thompson et al., 2020). According to forecasts, the number of transistors on a microchip doubling every two years (Moore's Law), will reach its limits (Leiserson et al., 2020). Furthermore, its two-year doubling period has already been overtaken by the exponentially growing need for compute, which is doubling every 3.4 months (Amodei & Hernandez) with a total increase of $300,000$ times between 2012 and 2018 (Amodei & Hernandez). Consequently, the growing need for compute will ultimately limit AI progress and innovation, especially in computationally expensive fields (Strubell et al., 2019; Thompson et al., 2020). Another aspect is that exponentially growing demand for compute can compromise reproducibility. The AI community already has a growing awareness of its importance; see, e.g., the ML Reproducibility Challenge (Papers with Code, 2024).

To summarize, reducing compute can tackle many of the problems mentioned above. One suggested direction is for AI researchers to redirect their focus toward Green AI and redefine what a state-of-the-art model entails. First, it is suggested that accuracy and efficiency be considered equally important when measuring progress in AI. Second, an analysis of the trade-off between performance and computational resources used should be included in AI research. It can be concluded that AI algorithms with higher efficiency will, therefore, positively impact both sustainability and AI progress (Strubell et al., 2019; Schwartz et al., 2020). In addition, reporting on efficiency metrics has other benefits: it will raise awareness and incentivize progress in efficiency (Henderson et al., 2020; Tamburrini, 2022).

2.2. Efficiency Metrics Suggested in Literature

So far, there is a tendency at major AI venues to report accuracy rather than efficiency or both metrics (Schwartz et al., 2020). Reporting efficiency for AI models can present

challenges, and currently, there is no standardized approach for this task in the AI community (Hernandez & Brown, 2020). A thorough understanding of their benefits and limitations is essential to employ efficiency metrics effectively. Consequently, we will discuss various evaluation criteria for efficiency that are considered in the literature, namely CO_2 equivalent (CO_2e) emissions, electricity usage, floating point operations (FLOPS), the big $\mathcal{O}$ notation, elapsed runtime and the number of parameters.

An appealing criterion to measure efficiency is CO_2e emissions since they are directly related to climate change. Disadvantages include that they are challenging to measure and are influenced by factors that do not account for algorithmic optimization, e.g. hardware, the carbon intensity of the used electricity, and time as well as the location of the compute (Schwartz et al., 2020; Strubell et al., 2020). An alternative criterion independent of time and location is to state the electricity usage in kWh. It can be quantified because GPUs commonly report the amount of consumed electricity. However, it is still hardware-dependent. There are several websites and packages available to compute the CO_2e and electricity usage of algorithms (Lacoste et al., 2019; Lottick et al., 2019; Anthony et al., 2020; Henderson et al., 2020; Lannelongue et al., 2021; Schmidt et al., 2021).

A hardware-independent criterion is the number of FLOPS required to train the model. On the one hand, they are computed analytically and facilitate a fair comparison between algorithms (Hernandez & Brown, 2020; Schwartz et al., 2020). On the other hand, they do not account, e.g., for optimized memory access or memory used by the model (Henderson et al., 2020; Schwartz et al., 2020). Next to giving the absolute number of FLOPS, it is possible to additionally report e.g. a FLOPS-based learning curve (Hernandez & Brown, 2020). Several packages are available to compute FLOPS, such as flops-counter, torchstat, pytorch-0pCounter or GFlops (Swall0w; sovrasov; GFlops; Lyken17).

A commonly used criterion in computer science is the big $\mathcal{O}$ notation (Knuth, 1976). It is used to report an algorithm's storage complexity and computational cost. While the big $\mathcal{O}$ notation may be impractical for AI practitioners due to the strong influence of application-specific termination criteria on run time, it offers significant theoretical benefits. It allows for easy comparison of algorithm sections and required storage complexities, making it a valuable tool in theoretical analysis.

The elapsed runtime is easy to measure but highly dependent on the used hardware and other jobs running in parallel (Schwartz et al., 2020). The number of parameters, whether they are learnable or total, is a widely used measure of efficiency. It is reassuringly agnostic to hardware and takes into account the memory needed by the model. However, it is essential to note that different model architectures can result in varying workloads for the same number of parameters.

3. Essentials of Tensor Networks for Green AI

As mentioned earlier, increasing the model parameters, hyperparameters, and training data size can boost an AI model's performance. TNs can handle the resulting large-scale or high-dimensional data objects. This paper builds upon the evidence for the compression potential of TNs provided in prior work. By establishing the link between TNs and Green AI explicitly for the first time, we aim to inspire future research that utilizes TNs beyond efficiency improvement but optimizes TNs as a part of a holistic portfolio for Green AI.

Alongside several reviews on TNs (Kolda & Bader, 2009; Cichocki et al., 2016), several surveys and technical contributions discuss the broad applicability and successful implementation of TNs in AI. The work referenced below is a small selection of existing work for TNs in AI; a complete overview is well beyond the scope of this paper. The surveys (Cichocki et al., 2016; 2017; Ji et al., 2019) give a broad overview of the applications of TNs, covering data preprocessing, supervised and unsupervised learning, regression and classification tasks, Gaussian Processes (GPs), kernel machines or deep learning, among others. A detailed overview of the various applications of TNs in neural networks (NNs) is provided by the survey (Wang et al., 2023), covering, for example, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and transformers. Collaborative filtering, mixture and topic modelling with TNs are discussed in (Sidiropoulos et al., 2017). Finally, in (Signoretto et al., 2011), a framework for kernel machines with TNs is discussed.

While the referenced sources are excellent for detailed technical knowledge, our paper primarily focuses on the potential of TNs for Green AI. Before applying TNs to AI is discussed in Section 4, this section explains the foundational concepts of TNs.

3.1. From Vectors and Matrices to Tensors and then Tensor Networks

Multidimensional arrays, widely known as tensors, are a generalization of vectors and matrices to higher orders[1]. The primary data structures in various fields are large vectors or matrices rather than tensors, which could imply that TNs might not be suitable for many standard applications. However, there is a solution to the problem described above: vectors and matrices can be rearranged into tensors by a procedure called tensorization (Oseledets, 2010; Khoromskij, 2011). The row and column size are factorized into multiple factors and then reshaped into a tensor. A small tensorization example is illustrated in Figure 2, where a matrix of size

[1] In the tensor community, order commonly refers to the number of indices: For example, a third-order tensor has three indices and can be visualized in a cube (Kolda & Bader, 2009).

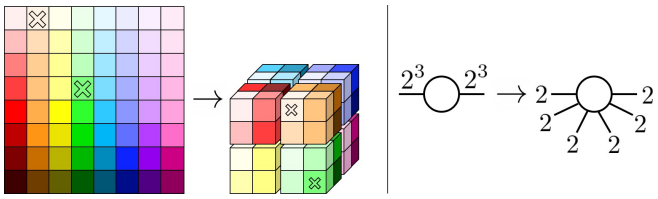

Figure 2. Left: tensorization of a matrix of size $2^3 \times 2^3$ into a sixth-order tensor of size $2 \times 2 \times 2 \times 2 \times 2 \times 2$. The two crosses illustrate the organization of the entries. Based on Fig. 2 of (Cichocki et al., 2015). Right: tensorization in diagram notation. The 2 edges of the node representing the 8-by-8 matrix become 6 edges sticking out the node representing the sixth-order tensor.

$2^3 \times 2^3$ is transformed into a sixth-order tensor, visually represented as a cube of smaller cubes. Tensorization enables the application of TNs to vectors and matrices through systematic index bookkeeping. The two 'X' shapes on the left side of Figure 2 illustrate this method of organizing entries. The essential requirement to preserve information during tensorization is meticulous indices tracking. For instance, the element labelled with the light orange 'X' in Figure 2, which is represented as $(1,\ 2)$ in matrix format, transforms into $(1,\ 1,\ 1,\ 1,\ 2,\ 1)$ in the tensorized form. Due to this bookkeeping approach, the impact of the newly imposed structure is expected to be minimal. In practice, the tensorization is simply performed as a `reshape()` operation, which does not introduce relevant computational costs. To simplify the depiction of higher-order objects, it is possible to use the diagram notation shown on the right side of Figure 2. In this notation, a matrix or a tensor is depicted as nodes with as many edges sticking out as its number of orders. In the following subsection, we introduce commonly used TNs.

3.2. Commonly Used Tensor Networks

A tensor can be expressed as a function of simpler tensors that form a TN, also called tensor decomposition. The idea of a TN originates in the generalization of the Singular Value Decomposition (SVD) to higher orders. In many applications, TNs can represent data in a compressed format with a marginal loss of information because of correlations present in the data (Cichocki et al., 2016; 2017).

In literature, the most commonly used TNs include the Canonical Polyadic (CP) (Carroll & Chang, 1970; Harshman, 1970), the Tucker (Tucker, 1966), and the Tensor Train (TT) decomposition (Oseledets, 2011). Without loss of generality, in this subsection, we will treat a third-order tensor $\boldsymbol{\mathcal{Y}}$ to introduce the TNs mentioned above. The **CP decomposition** (Carroll & Chang, 1970; Harshman, 1970) of $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ consists of a set of factor matrices $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^{I_j \times R}$, $j = 1, 2, 3$ and a weight vector

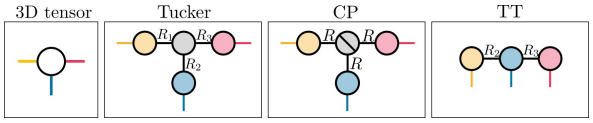

Figure 3. Graphical depiction of commonly used TNs for a third-order tensor. Connected edges are indices that are being summed over. The CP decomposition is a special case of Tucker, where the core tensor is diagonal. This is shown by the diagonal in the node.

$\boldsymbol{\lambda} \in \mathbb{R}^{R \times 1}$. Elementwise, $\boldsymbol{\mathcal{Y}}$ can be computed from

$$y_{i_1 i_2 i_3} = \sum_{r=1}^{R} \lambda_r \, a_{i_1 r} \, b_{i_2 r} \, c_{i_3 r}. \tag{1}$$

The scalars $a_{i_1 r}, b_{i_2 r}, c_{i_3 r}$ are the entries of the three factor matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, λ_r is the r-th entry of $\boldsymbol{\lambda}$, R denotes the rank of the decomposition. The **Tucker decomposition** (Tucker, 1966) of $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ consists of a 3-way tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$, called the core tensor, and a set of matrices $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^{I_j \times R_j}$, $j = 1, 2, 3$. Elementwise, $\boldsymbol{\mathcal{Y}}$ can be computed from

$$y_{i_1 i_2 i_3} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} g_{r_1 r_2 r_3} \, a_{i_1 r} \, b_{i_2 r} \, c_{i_3 r}. \tag{2}$$

The scalars $a_{i_1 r}, b_{i_2 r}, c_{i_3 r}$ are the entries of the three factor matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, $g_{r_1 r_2 r_3}$ is the $(r_1 r_2 r_3)$-th entry of $\boldsymbol{\mathcal{G}}$ and R_1, R_2, R_3 denote the ranks of the decomposition. The **TT decomposition** (Oseledets, 2011) of $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ consists of a set of three-way tensors $\boldsymbol{\mathcal{Y}}^{(j)} \in \mathbb{R}^{R_j \times I_j \times R_{j+1}}$, $j = 1, 2, 3$ called TT-cores. Elementwise, $\boldsymbol{\mathcal{Y}}$ can be computed from

$$y_{i_1 i_2 i_3} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} \sum_{r_4=1}^{R_4} y^{(1)}_{r_1 i_1 r_2} \, y^{(2)}_{r_2 i_2 r_3} \, y^{(3)}_{r_3 i_3 r_4}, \tag{3}$$

where R_1, R_2, R_3, R_4 denote the ranks of the TT-cores and by definition $R_1 = R_4 = 1$. Figure 3 shows a diagram depicting the CP, Tucker, and TT decomposition for the case of a third-order tensor. Connected edges are indices being summed over, whereas the number of free edges corresponds to the order of the tensor.

When a TN approximates a tensor, the chosen ranks are pivotal. These ranks act as hyperparameters and are crucial for determining the approximation's accuracy, necessitating careful tuning. The usual goal is to balance compression and accuracy effectively. It is essential to select ranks that are low enough to ensure effective compression but still high enough to maintain the desired level of accuracy of the approximation.

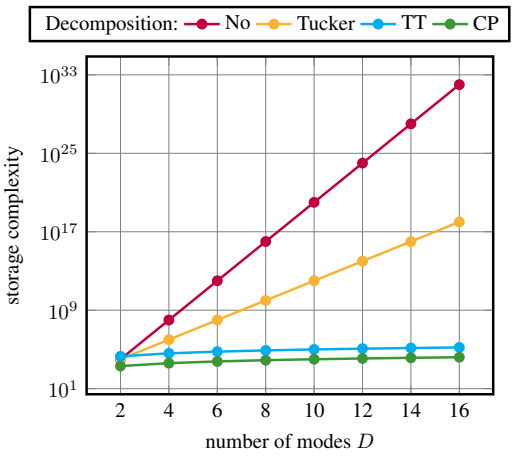

Figure 4. Demonstrating the impact of Tucker, TT, and CP decompositions on a I^D tensor with $I = 100$ and $R = 10$. Without decomposition, the tensor's storage complexity increases exponentially with D. The Tucker decomposition yields a slower exponential growth, while CP and TT decompositions grow linear in D.

3.3. Logarithmic Compression Potential of Low-Rank Tensor Networks as an Enabler for Green AI

The low-rank approximation of TNs is powerful for two key reasons. First, computations can be performed in the compressed low-rank format, typically executed at the level of individual TN components such as CP factor matrices or TT cores. Due to these mathematical properties, TNs can efficiently compress both data and model parameters (Cichocki et al., 2016; 2017). Second, low-rank TNs can transform an exponential complexity into a linear complexity with minimal loss of information (Cichocki et al., 2016; 2017). We call this type of transformation logarithmic compression. The inherent logarithmic compression capability of TNs is a key reason we present them as a powerful tool for Green AI in this position paper.

Considering a CP decomposition for a given tensor with I^D elements, the number of elements in its rank-R decomposition are $\mathcal{O}(RID)$, thus linear in D. Assuming a uniform rank R in a TT decomposition yields a storage complexity of $\mathcal{O}(R^2ID)$ elements. Consequently, both decompositions achieve logarithmic compression. The Tucker decomposition, on the other hand, still scales exponentially with D: a given tensor with I^D elements requires $\mathcal{O}(R^D)$ in the Tucker format. Tucker decompositions can still achieve significant compression if $R \ll I$. Figure 4 illustrates the effects of different decompositions for a numerical example.

4. Applications of Tensor Networks to AI

This position paper promotes the strategic application of TNs to develop more sustainable AI algorithms. So far, we have established that TNs have broad applicability for various AI model architectures, learning paradigms, and tasks. When optimally leveraged, they have the potential to meet the growing computational needs in AI due to their logarithmic compression capability while maintaining accuracy. Thanks to efficient bookkeeping, TNs are versatile. They are adept at processing not only tensor-formatted data but also vectors and matrices, making them suitable for a wide range of AI applications.

This section aims to enrich existing technical surveys on TNs by injecting a distinct Green AI perspective supported by carefully curated examples from kernel machines and deep learning. We delve into a typical supervised learning scenario to illustrate the practical integration of TNs into AI algorithms. Given a set of independent and identically distributed input/output pairs $\mathbf{x}_n$ and $\mathbf{y}_n$, a supervised learning problem can be described as minimizing the measure of loss L and a regularization term R

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} L(\mathbf{y}_n, f(\mathbf{x}_n \mid \mathbf{w})) + R(\mathbf{w}), \tag{4}$$

where $f(\mathbf{x} \mid \mathbf{w})$ is a nonlinear function parameterized by the weights $\mathbf{w}$. To incorporate TNs, $f(\mathbf{x} \mid \mathbf{w})$ is parametrized with low-rank TNs. We will explore two specific implementations of this parametrization: a kernel machine and a NN. In discussing these models, we will highlight efficiency improvements and the associated changes in accuracy as documented in the literature, demonstrating how TNs enhance the efficiency of AI algorithms. The insights gained from these examples showcase the practical efficacy of TNs in advancing sustainable AI and underscore their potential for broader application across various AI domains.

4.1. Tensor Networks in Kernel Machines

Kernel machines, such as Gaussian processes (Rasmussen & Williams, 2006) and support vector machines (Cortes & Vapnik, 1995), can be universal function approximators (Hammer & Gersmann, 2003). While they have shown equivalent or superior performance compared to NNs (Lee et al., 2017; Garriga-Alonso et al., 2018; Novak et al., 2018), they scale poorly for high-dimensional or large-scale problems. A single-output kernel machine is given by

$$f(\mathbf{x} \mid \mathbf{w}) = \langle \boldsymbol{\varphi}(\mathbf{x}), \mathbf{w} \rangle, \tag{5}$$

where $\boldsymbol{\varphi}(\cdot)$ is a feature map, $\mathbf{w}$ is a weight vector and $\langle \cdot, \cdot \rangle$ denotes the inner product. A common choice is to represent $\boldsymbol{\varphi}(\cdot)$ as a Kronecker product of D regressors, computed from a chosen number I basis functions. The adoption of a Kronecker product structure can lead to significant storage and computational complexity, resulting from the exponential increase of the number of basis functions $\boldsymbol{\varphi}(\cdot) \in \mathbb{R}^{I^D}$

and parameters in $\mathbf{w} \in \mathbb{R}^{I^D}$. The issue of this exponential growth can be completely alleviated by imposing a TN structure on both $\boldsymbol{\varphi}(\mathbf{x})$ and $\mathbf{w}$, leveraging the logarithmic compression potential of TNs. In other words, the basis functions $\varphi(\mathbf{x})$ and weight vector $\mathbf{w}$ are never explicitly calculated. Instead, their TN representations are used throughout training and inference. For kernel machines with TNs, the following basis functions have been explored in the literature: polynomial basis function are treated in (Rendle, 2010; Blondel et al., 2016; Batselier & Wong, 2017), pure-power-1 polynomials in (Novikov et al., 2017), lagged timeseries in (Batselier et al., 2017), pure-power-I polynomials in (Chen et al., 2017) and B-splines in (Karagoz & Batselier, 2020). The use of trigonometrical basis functions is described in (Stoudenmire & Schwab, 2016), and Fourier features in (Wahls et al., 2014; Kargas & Sidiropoulos, 2021; Wesel & Batselier, 2021). Wesel & Batselier, for example, achieved superior accuracy results with TNs on a laptop compared to those previously obtained by an alternative method on a cluster (2021).

Employing TNs can significantly reduce a model's memory requirements. For example, applying CP decomposition reduces the storage complexity from $\mathcal{O}(I^D)$ down to $\mathcal{O}(DIR)$, and the total computational runtime for training from $\mathcal{O}(NI^{2D} + I^{3D})$ down to $\mathcal{O}(DN(IR)^2 + D(IR)^3)$ (Wesel & Batselier, 2021). Significant computational savings are attained by performing both training and inference computations at the level of a single factor matrix with dimensions $I \times R$. The key insight is the transformation of the dependency on D from an exponential to a linear scale, a change that substantially boosts efficiency without compromising accuracy.

4.2. Tensor Networks in Deep Learning

Deep learning has achieved state-of-the-art performance in many fields, such as computer vision (Krizhevsky et al., 2012; He et al., 2016) and Natural Language Processing (NLP) (Devlin et al., 2019; Brown et al., 2020). The success, however, comes at a cost: models in deep learning are large and require a lot of compute (Strubell et al., 2019; 2020). NNs have been made more efficient with TNs for a variety of application fields, including computer vision (Jaderberg et al., 2014; Lebedev et al., 2015; Kim et al., 2016) and NLP (Ma et al., 2019; Hrinchuk et al., 2020; Abronin et al., 2024).

We address the learning problem as described in (4), where the function $f(\mathbf{x} \mid \mathbf{w})$ is parameterized by a NN. TNs can be integrated into NNs by compressing a pretrained NN or directly training a compressed NN. A NN that incorporates TNs, either wholly or partially, is referred to as a factorized NN. The first method involves taking a readily available pretrained network (Abadi et al., 2015; Paszke et al., 2019). It then decomposes layers with a TN of choice to minimize the approximation error on the pretrained weights. Subsequently, it is a standard practice to fine-tune the factorized network, aiming to restore any performance that may have been compromised during the decomposition process (Denton et al., 2014; Lebedev et al., 2015; Kim et al., 2016). The second approach entails starting with a randomly initialized factorized network and proceeding to train it while it remains compressed. This method is recognized for improving training efficiency through the decrease in the number of parameters (Ye et al., 2018). In both methodologies, whether during fine-tuning of the first or training of the second, back-propagation is conducted based on equation (4), with the weights represented in the form of TNs.

As an example that works for both methodologies, we show how to compress the weights for a fully connected layer with a TT decomposition following the foundational methodology of Novikov et al. (2015). In a fully connected layer, the primary operation is the matrix-vector multiplication $\mathbf{W}\mathbf{x}$, involving the weight matrix $\mathbf{W} \in \mathbb{R}^{I^D \times J^D}$ and input vector $\mathbf{x} \in \mathbb{R}^{J^D}$. To simplify the notation, we omit the bias term. To prepare for applying a TN, both the weight matrix $\mathbf{W}$ and the input vector $\mathbf{x}$ must first be tensorized into higher-order tensors, resulting in $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{I \times \ldots \times I \times J \times \ldots \times J}$ and $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{J \times \ldots \times J}$, respectively. The tensorization is usually performed as shown in Section 3.1. Without loss of generality, we simplify the notation by setting $I_i = I$ and $J_j = J$ for $i, j \in 1, ..., D$. Attaining the desired efficiency is accomplished by substituting $\boldsymbol{\mathcal{W}}$ with a TN. Assuming a TT decomposition with uniform rank R, the entries of the matrix-vector product are computed as

$$\sum_{r_2 \ldots r_D} \sum_{j_1 \ldots j_D} w^{(1)}_{i_1 j_1 r_2} w^{(2)}_{r_2 i_2 j_2 r_3} \cdots w^{(d)}_{r_D i_D j_D} x_{j_1 \ldots j_D}. \quad (6)$$

Introducing TTs in fully connected layers reduces the computational complexity of the forward pass from $\mathcal{O}(I^D J^D)$ for the matrix representation to $\mathcal{O}(DR^2 I \max\{I^D, J^D\})$. The learning complexity of one backward pass is reduced from $\mathcal{O}(I^D J^D)$ to $\mathcal{O}(D^2 R^4 I \max\{I^D, J^D\})$. The memory complexity is reduced from $\mathcal{O}(I^D J^D)$ to $\mathcal{O}(R \max\{I^D, J^D\})$ for the forward pass and to $\mathcal{O}(R^3 \max\{I^D, J^D\})$ for the backward pass. Choosing a sufficiently small rank R increases the efficiency compared to the full rank model. A theoretical guarantee for the optimal low-rank approximation, similar to the Eckhardt-Schmidt-Young theorem for matrices, does not exist for TNs (Vannieuwenhoven et al., 2014). In Section 4.3, we provide empirical evidence to showcase the effect of low-rank TNs on the accuracy and efficiency of NNs.

Besides TTs (Novikov et al., 2015; Garipov et al., 2016; Tjandra et al., 2017; Wu et al., 2020; Yang et al., 2017; Efthymiou et al., 2019; Yu et al., 2019; Cheng et al., 2021), other decompositions, e.g. Tucker (Kim et al., 2016; Calvi et al., 2020; Chu & Lee, 2021) and CP (Mamalet & Garcia,

2012; Rigamonti et al., 2013; Denton et al., 2014; Jaderberg et al., 2014; Lebedev et al., 2015; Astrid & Lee, 2017; Chen et al., 2020; Kossaifi et al., 2020) have been proposed for fully connected, convolutional, recurrent, and attention layers.

4.3. Green AI Analysis of Selected Tensor Network Models with Regard to Efficiency Metrics

In this section, we assess how the application of TNs has led to improvements in the efficiency metrics presented in Section 2.2. The prevalent focus in the AI community is on reporting solely accuracy rather than both accuracy and efficiency. This fact, combined with the absence of a standardized method for reporting efficiency, results in only a limited amount of TN-related papers that can be used in our analysis. Table 1 summarizes the results discussed below.

For kernel machines, several examples the examples effectively showcase how TN can contribute towards Green AI (Novikov et al., 2017; Kargas & Sidiropoulos, 2021; Wesel & Batselier, 2021). Kargas & Sidiropoulos show that the CP decomposition model for supervised learning can match or even outperform NNs. Their research highlights comparable execution times and marginally better average accuracy for the TN methods (2021). Novikov et al. report that their TN model not only offers competitive training and inference times but also yields a significant 58% increase in accuracy compared to a NN (2017). Compared to the best competing method (Mikhail Trofimov, 2016), they achieved a $21\times$ speed up with a drop in accuracy of $11.46\,\%$ (Novikov et al., 2017). Wesel & Batselier achieved superior accuracy 2.5 times faster on a laptop than the best competing method (Hensman et al., 2013) on a cluster ($7141\,\mathrm{s}$ vs. $18360\,\mathrm{s}$, respectively) (2021). Compared to the uncompressed problem, using a TN reduced the number of parameters by up to 99.9% (Wesel & Batselier, 2021).

We highlight the contribution of TN in deep learning towards Green AI by showcasing efficiency improvements with four examples (Kim et al., 2016; Ye et al., 2018; Yin et al., 2021; Abronin et al., 2024). Kim et al. and Ye et al. have demonstrated that TNs, when applied to convolutional and recurrent layers, respectively, enhance training and inference efficiency compared to standard NNs without significantly compromising accuracy (2016; 2018). Across a range of datasets, Kim et al. managed significant efficiency improvements by applying TNs, achieving up to an 86.4% reduction in memory, 79.7% in FLOPS, 72.8% in runtime, and 76.6% in electricity usage while maintaining robust top-5 accuracy, reduced by no more than 2.2% (2016). With the application of TNs, Ye et al. managed to drastically cut down the number of parameters for the uncompressed model (by over 80,000 times), speed up convergence by 14 times, and even increase accuracy by 15.6% (2018). Yin et al. proposed a framework to compress both CNNs and RNNs, which they have tested for various image classification and video recognition tasks (2021). For CNN models, they were able to slightly increase the accuracy compared to the uncompressed model for multiple data sets, with a compression ratio between $2.4\times$ and $8.3\times$ (Yin et al., 2021). Abronin et al. compressed GPT-2_{small}, utilizing only a fraction of the training data set to recover the performance loss. Their method outperforms competing compression approaches, achieving a $1.5\times$ compression ratio with a relative loss of 30.83% in performance (2024).

Employing the efficiency metrics advocated in Green AI literature clearly demonstrates the considerable potential of TNs to uphold Green AI principles, namely enhancing efficiency while maintaining accuracy. In some scenarios, TNs enhance both accuracy and efficiency, which is an ideal outcome. However, in others, a trade-off may occur between a minor decrease in accuracy and significant improvements in energy efficiency and memory usage. The decision to engage in this trade-off is highly dependent on the specific context and application, making it impractical to offer a one-size-fits-all recommendation. Nevertheless, a fundamental guideline remains: a thorough analysis of accuracy and efficiency metrics is indispensable to arrive at a well-informed decision regarding the trade-off.

5. Critical Discussion and Conclusion

This proposition paper is the first paper to propose TNs as an essential tool for realizing Green AI. First, we addressed the exponential rise in computational demands in AI, followed by a review of the Green AI literature and various efficiency metrics suggested by researchers. A brief technical overview of TNs was provided, followed by a practical introduction to the practical implementation of TNs in deep learning and kernel machines. We demonstrated through specific examples how TNs can boost efficiency while maintaining accuracy.

To summarize, the strength of TNs for Green AI lies in their solid mathematical foundation, logarithmic compression potential and broad applicability across different data formats and model architectures. They are a versatile tool that can significantly increase the efficiency of data preprocessing, training, and inference.

Regarding the three categories for Green AI, the impact of TNs can have limitations. A limitation of TNs on AI model development, namely to enhance model efficiency, is anticipated for weakly correlated data since TNs are based on extending the truncated SVD to higher orders. In this case, other methods can be used, or TNs can be combined with other methods. So far, TNs have, for example, been effectively paired with methods such as regularization (Sofuoglu & Aviyente, 2020; Wesel & Batselier, 2021), au-

Table 1. Overview over TN models discussed in Section 4.3. Here, accuracy is to be understood as a umbrella term (Schwartz et al., 2020). The following terms are abbreviated: classification (class.), regression (reg.), action recognition (act. reg.), Natural Language Processing (NLP), Kernel Machine (KM), Long Short-Term Memory (LSTM).

TN-MODEL	TASK	MODEL	DATA SET	SPEED UP COMP. RATIO	RELATIVE CHANGE IN ACCURACY
(NOVIKOV ET AL., 2017)	CLASS.	KM	SYNTHETIC	21.3× SPEED-UP	−11.46 %
(KARGAS & SIDIROPOULOS, 2021)	REG.	KM	ABALONE	1.6× SPEED-UP	−4.39 %
(WESEL & BATSELIER, 2021)	REG.	KM	AIRLINE	2.5× SPEED-UP	+1.13 %
(KIM ET AL., 2016)	CLASS.	ALEXNET	IMAGENET	1.8× SPEED-UP	+2.17 %
(YE ET AL., 2018)	ACT. REC.	LSTM	UCF11	14.0× SPEED-UP	+1.13 %
(YIN ET AL., 2021)	CLASS.	RESNET	CIFAR-100	2.4× COMP. RATIO	+3.25 %
(ABRONIN ET AL., 2024)	NLP	GPT-2 SMALL	WIKITEXT-2	1.5× COMP. RATIO	−30.83 %

tomated hyperparameter search (Deng & Xiao, 2022) or knowledge distillation (Abronin et al., 2024).

The effect of TNs on computing infrastructure and system-level impacts is clearly limited. TNs have no effect on the computing infrastructure beyond their reduced hardware requirements. Alternative low-carbon technologies are required to provide sufficient renewable energy sources, to reduce the energy demand of data centres and the emissions associated with manufacturing, transporting and recycling hardware. Potential system-level impacts of TNs (and of most methods that increase the efficiency of AI models) include the acceleration of carbon-intensive technologies and lifestyle choices or the rebound effect. This effect occurs when the expected gains are diminished due to increased technology usage. However, the rebound effect is not exclusively detrimental. It can enhance inclusivity by lowering barriers to entry for previously excluded researchers and, therefore, increase usage. This broader inclusion has the potential to bring together a more diverse group of researchers, which could enrich the progress of AI. Lastly, the role of AI in addressing climate change issues, as discussed in works like Huntingford et al., must be considered (2019). Mitigating undesired system-level impacts requires, e.g., policy considerations and technology management. Ensuring that AI solutions foster sustainability requires a careful appraisal of the costs and benefits, a consideration that, while important, is beyond the scope of this paper.

Regarding the three pillars of sustainability, TNs offer notable potential benefits. First, by reducing the need for hardware and computational resources, TNs can cut costs, making them an economically more sustainable solution and a viable choice for industrial applications. Second, the reduced hardware and computational demands of TNs decrease reliance on expensive, potentially external computing resources. As discussed above, reduced hardware requirements can lower entry barriers to AI research, potentially fostering inclusivity and social sustainability. Additionally, the capacity to process sensitive data on-site can enhance data privacy. Third, the ability of TNs to minimize hardware use and shorten computational time can significantly decrease the embodied and operational emissions associated with AI. Consequently, the widespread adoption of TNs in AI models could profoundly affect the environmental sustainability of AI. Looking forward, as we approach the constraints imposed by Moore's Law, the quest for algorithmic innovation is anticipated to increasingly focus on enhancing efficiency. In this scenario, TNs, renowned for their logarithmic compression potential, can play a crucial role in facilitating continued advancement of AI.

A potential direction for future work could be to provide a comprehensive overview and assessment of particularly beneficial combinations of additional Green AI methods with TNs, considering application-specific requirements. Regarding TNs specifically, there are open challenges that can dampen their potential benefits on Green AI. For example, implementing favourable design choices for TNs often requires experience. Practitioners' guidelines or theoretical frameworks to support decision-making are yet to be developed. A first step in that direction could be quantifying TNs' effect on the whole model development phase and the savings in embodied emissions.
We advocate for optimizing TNs to Green AI objectives. That includes intensifying research on data- or hardware-efficient TN models, training TN models from scratch, or gradually compressing TN models while training. Open questions evolve around tensor rank approximation. Finding the true tensor rank is an NP-hard problem. Furthermore, for tensors, there exists no theorem akin to the Schmidt-Eckart-Young theorem to provide a theoretical guarantee for the optimal low-rank approximation. Moreover, future research could further investigate proper weight initialization or efficient hyperparameter training, among others.

We envision that TNs will have a broad application in diverse sectors of AI and contribute to more sustainable AI. We encourage researchers to integrate TNs in their research, address the open questions, and advance AI progress.

Acknowledgements

We would like to thank the anonymous reviewers for their numerous suggestions and improvements which have greatly improved the quality of this paper. Eva Memmel and Frederiek Wesel, and thereby this work, are supported by the Delft University of Technology AI Labs program. The authors declare no competing interests.

Impact Statement

This paper links tensor networks and Green AI research fields to enhance AI model sustainability and research inclusivity. The potential societal impact is discussed throughout the paper, especially in Section 5.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/.

Abronin, V., Naumov, A., Mazur, D., Bystrov, D., Tsarova, K., Melnikov, A., Oseledets, I., Dolgov, S., Brasher, R., and Perelshtein, M. Tqcompressor: improving tensor decomposition methods in neural networks via permutations. *arXiv preprint arXiv:2401.16367*, 2024.

Ahmed, N., Wahed, M., and Thompson, N. C. The growing influence of industry in AI research. *Science*, 379(6635): 884–886, 2023.

Amodei, D. and Hernandez, D. AI and Compute. https://openai.com/blog/ai-and-compute/. Accessed: 2024-01-29.

Anthony, L. F. W., Kanding, B., and Selvan, R. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems, 2020. arXiv:2007.03051.

Astrid, M. and Lee, S.-I. CP-decomposition with tensor power method for convolutional neural networks compression. In *IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 115–118, 2017. doi: 10.1109/BIGCOMP.2017.7881725. ISSN: 2375-9356.

Batselier, K. and Wong, N. A constructive arbitrary-degree Kronecker product decomposition of tensors. *Numerical Linear Algebra with Applications*, 24(5):e2097, 2017.

Batselier, K., Chen, Z., and Wong, N. Tensor network alternating linear scheme for MIMO Volterra system identification. *Automatica*, 84:26–35, 2017.

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.-L., et al. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(2):e1484, 2023.

Blondel, M., Fujino, A., Ueda, N., and Ishihata, M. Higher-order factorization machines. *Advances in Neural Information Processing Systems*, 29, 2016.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Calvi, G. G., Moniri, A., Mahfouz, M., Zhao, Q., and Mandic, D. P. Compression and interpretability of deep neural networks via tucker tensor layer: From first principles to tensor valued back-propagation. *arXiv:1903.06133 [cs, eess]*, 2020.

Carroll, J. D. and Chang, J. J. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

Chen, W., Zhu, X., Sun, R., He, J., Li, R., Shen, X., and Yu, B. Tensor low-rank reconstruction for semantic segmentation. *arXiv:2008.00490 [cs]*, 2020.

Chen, Z., Batselier, K., Suykens, J. A., and Wong, N. Parallelized tensor train learning of polynomial classifiers. *IEEE transactions on neural networks and learning systems*, 29(10):4621–4632, 2017.

Cheng, Y., Yang, Y., Chen, H.-B., Wong, N., and Yu, H. S3-net: A fast and lightweight video scene understanding network by single-shot segmentation. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3329–3337, 2021.

Chu, B.-S. and Lee, C.-R. Low-rank tensor decomposition for compression of convolutional neural networks using funnel regularization. *arXiv:2112.03690 [cs]*, 2021.

Cichocki, A. Era of big data processing: A new approach via tensor networks and tensor decompositions. *arXiv preprint arXiv:1403.2048*, 2014.

Cichocki, A., Mandic, D., De Lathauwer, L., Zhou, G., Zhao, Q., Caiafa, C., and Phan, H. A. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE signal processing magazine*, 32(2):145–163, 2015.

Cichocki, A., Lee, N., Oseledets, I. V., Phan, A.-H., Zhao, Q., and Mandic, D. P. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016.

Cichocki, A., Phan, A.-H., Zhao, Q., Lee, N., Oseledets, I. V., Sugiyama, M., and Mandic, D. P. Tensor networks for dimensionality reduction and large-scale optimizations: Part 2 applications and future perspectives. *Learning*, 9(6):431–673, 2017.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to algorithms*. MIT press, 2022.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 20:273–297, 1995.

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., and Piccialli, F. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022.

Deng, L. and Xiao, M. A new automatic hyperparameter recommendation approach under low-rank tensor completion e framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4038–4050, 2022.

Denton, E. L., Zaremba, W., Bruna, J., LeCun, Y., and Fergus, R. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*, pp. 9, 2014.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805 [cs]*, 2019. version: 2.

Dodge, J., Prewitt, T., Tachet des Combes, R., Odmark, E., Schwartz, R., Strubell, E., Luccioni, A. S., Smith, N. A., DeCario, N., and Buchanan, W. Measuring the carbon intensity of AI in cloud instances. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1877–1894, 2022.

Efthymiou, S., Hidary, J., and Leichenauer, S. TensorNetwork for machine learning. *arXiv:1906.06329 [cond-mat, physics:physics, stat]*, 2019.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018.

Ganguli, D., Hernandez, D., Lovitt, L., Askell, A., Bai, Y., Chen, A., Conerly, T., Dassarma, N., Drain, D., Elhage, N., et al. Predictability and surprise in large generative models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1747–1764, 2022.

Garipov, T., Podoprikhin, D., Novikov, A., and Vetrov, D. Ultimate tensorization: Compressing convolutional and FC layers alike. *arXiv:1611.03214 [cs]*, 2016.

Garriga-Alonso, A., Rasmussen, C. E., and Aitchison, L. Deep convolutional networks as shallow Gaussian processes. In *International Conference on Learning Representations*, 2018.

GFlops. Gflops.jl. https://github.com/triscale-innov/GFlops.jl. Accessed: 2024-01-29.

Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., and Keutzer, K. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pp. 291–326. Chapman and Hall/CRC, 2022.

Gupta, U., Kim, Y. G., Lee, S., Tse, J., Lee, H.-H. S., Wei, G.-Y., Brooks, D., and Wu, C.-J. Chasing carbon: The elusive environmental footprint of computing. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 854–867. IEEE, 2021.

Hammer, B. and Gersmann, K. A note on the universal approximation capability of support vector machines. *neural processing letters*, 17(1):43–53, 2003.

Harshman, R. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16 (10):1– 84, 1970.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. IEEE, 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.90.

He, L., Chen, K., Xu, W., Zhou, J., and Wang, F. Boosted sparse and low-rank tensor regression. *Advances in Neural Information Processing Systems*, 31, 2018.

Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., and Pineau, J. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43, 2020.

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

Hernandez, D. and Brown, T. B. Measuring the algorithmic efficiency of neural networks. *arXiv preprint arXiv:2005.04305*, 2020.

Hrinchuk, O., Khrulkov, V., Mirvakhabova, L., Orlova, E., and Oseledets, I. V. Tensorized embedding layers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4847–4860. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.436.

Huntingford, C., Jeffers, E. S., Bonsall, M. B., Christensen, H. M., Lees, T., and Yang, H. Machine learning and artificial intelligence to aid climate change research and preparedness. *Environmental Research Letters*, 14(12): 124007, 2019.

Izmailov, P., Novikov, A., and Kropotov, D. Scalable Gaussian processes with billions of inducing inputs via tensor train decomposition. In *International Conference on Artificial Intelligence and Statistics*, pp. 726–735. PMLR, 2018.

Jaderberg, M., Vedaldi, A., and Zisserman, A. Speeding up convolutional neural networks with low rank expansions. *arXiv:1405.3866*, 2014.

Ji, Y., Wang, Q., Li, X., and Liu, J. A survey on tensor techniques and applications in machine learning. *IEEE Access*, 7:162950–162990, 2019. doi: 10.1109/ACCESS.2019.2949814.

Kaack, L. H., Donti, P. L., Strubell, E., Kamiya, G., Creutzig, F., and Rolnick, D. Aligning artificial intelligence with climate change mitigation. *Nature Climate Change*, 12(6):518–527, 2022.

Karagoz, R. and Batselier, K. Nonlinear system identification with regularized tensor network b-splines. *Automatica*, 122:109300, 2020.

Kargas, N. and Sidiropoulos, N. D. Supervised learning and canonical decomposition of multivariate functions. *IEEE Transactions on Signal Processing*, 69:1097–1107, 2021.

Khoromskij, B. N. O (dlog n)-quantics approximation of nd tensors in high-dimensional numerical modeling. *Constructive Approximation*, 34(2):257–280, 2011.

Kim, Y.-D., Park, E., Yoo, S., Choi, T., Yang, L., and Shin, D. Compression of deep convolutional neural networks for fast and low power mobile applications. In *ICLR*, 2016.

Knuth, D. E. Fundamental algorithms. 1973. URL http://papers.cumincad.org/cgi-bin/works/paper/ef80.

Knuth, D. E. Big omicron and big omega and big theta. *ACM Sigact News*, 8(2):18–24, 1976.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Kossaifi, J., Toisoul, A., Bulat, A., Panagakis, Y., Hospedales, T. M., and Pantic, M. Factorized higher-order CNNs with an application to spatio-temporal emotion estimation. In *Computer Vision and Pattern Recognition*, pp. 6060–6069, 2020.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, 2012. doi: 10.1145/3065386.

Lacoste, A., Luccioni, A., Schmidt, V., and Dandres, T. Quantifying the carbon emissions of machine learning. *Workshop on Tackling Climate Change with Machine Learning at NeurIPS 2019*, 2019.

Lannelongue, L., Grealey, J., and Inouye, M. Green algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, 2021.

Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I. V., and Lempitsky, V. Speeding-up convolutional neural networks using fine-tuned CP-decomposition. In *International Conference Learning Representations*, 2015.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. Deep neural networks as Gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

Leiserson, C. E., Thompson, N. C., Emer, J. S., Kuszmaul, B. C., Lampson, B. W., Sanchez, D., and Schardl, T. B. There's plenty of room at the top: What will drive computer performance after Moore's law? *Science*, 368 (6495), 2020.

Li, Y., Ma, W., Chen, C., Zhang, M., Liu, Y., Ma, S., and Yang, Y. A survey on dropout methods and experimental verification in recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Lottick, K., Susai, S., Friedler, S. A., and Wilson, J. P. Energy usage reports: Environmental awareness as part of algorithmic accountability. *Workshop on Tackling Climate Change with Machine Learning at NeurIPS 2019*, 2019. arXiv:1911.08354.

Lyken17. pytorch-opcounter. hhttps://github.com/Lyken17/pytorch-OpCounter. Accessed: 2024-01-29.

Ma, X., Zhang, P., Zhang, S., Duan, N., Hou, Y., Zhou, M., and Song, D. A tensorized transformer for language modeling. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Mamalet, F. and Garcia, C. Simplifying ConvNets for fast learning. In *Artificial Neural Networks and Machine Learning – ICANN 2012*, pp. 58–65. Springer, 2012. ISBN 978-3-642-33266-1. doi: 10.1007/978-3-642-33266-1_8.

Mikhail Trofimov, A. N. tffm: Tensorflow implementation of an arbitrary order factorization machine. https://github.com/geffy/tffm, 2016.

myclimate, F. Calculate your flight emissions! https://co2.myclimate.org/en/flight_calculators/new, 2024. Accessed: 2024-01-29.

Novak, R., Xiao, L., Bahri, Y., Lee, J., Yang, G., Hron, J., Abolafia, D. A., Pennington, J., and Sohl-dickstein, J. Bayesian deep convolutional networks with many channels are Gaussian processes. In *International Conference on Learning Representations*, 2018.

Novikov, A., Podoprikhin, D., Osokin, A., and Vetrov, D. P. Tensorizing neural networks. In *Neural Information Processing Systems*, 2015.

Novikov, A., Trofimov, M., and Oseledets, I. V. Exponential machines. In *Conference on Learning Representations ICLR 2017, Workshop Track Proceedings*, 2017.

Oseledets, I. V. Approximation of 2d x 2d Matrices using Tensor Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2130–2145, 2010.

Oseledets, I. V. Tensor-train decomposition. *SIAM J. Sci. Comput.*, 33(5):2295–2317, 2011. ISSN 1064-8275, 1095-7197. doi: 10.1137/090752286.

Panagakis, Y., Kossaifi, J., Chrysos, G. G., Oldfield, J., Nicolaou, M. A., Anandkumar, A., and Zafeiriou, S. Tensor methods in computer vision and deep learning. *Proceedings of the IEEE*, 109(5):863–890, 2021.

Papers with Code. ML Reproducibility Challenge 2021. https://paperswithcode.com/rc2021, 2024. Accessed: 2024-01-29.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in neural information processing systems 32*. Curran Associates, Inc., 2019.

Purvis, B., Mao, Y., and Robinson, D. Three pillars of sustainability: In search of conceptual origins. *Sustainability science*, 14(3):681–695, 2019.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

Reed, R. Pruning algorithms-a survey. *IEEE transactions on Neural Networks*, 4(5):740–747, 1993.

Rendle, S. Factorization machines. In *2010 IEEE International conference on data mining*, pp. 995–1000. IEEE, 2010.

Richter, L., Sallandt, L., and Nüsken, N. Solving high-dimensional parabolic PDEs using the tensor train format. In *International Conference on Machine Learning*, pp. 8998–9009, 2021.

Rigamonti, R., Sironi, A., Lepetit, V., and Fua, P. Learning separable filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2754–2761, 2013.

Schmidt, V., Goyal, K., Joshi, A., Feld, B., Conell, L., Laskaris, N., Blank, D., Wilson, J., Friedler, S., and Luccioni, S. CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing. https://github.com/mlco2/codecarbon, 2021.

Schwartz, R., Dodge, J., Smith, N. A., and Etzioni, O. Green AI. *Communications of the ACM*, 63(12):54–63, 2020.

Sengupta, R., Adhikary, S., Oseledets, I., and Biamonte, J. Tensor networks in machine learning. *European Mathematical Society Magazine*, (126):4–12, 2022.

Sidiropoulos, N. D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E. E., and Faloutsos, C. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017. doi: 10.1109/TSP.2017.2690524.

Signoretto, M., De Lathauwer, L., and Suykens, J. A. A kernel-based framework to tensorial data analysis. In *International Conference on Artificial Neural Networks*, volume 24, pp. 861–874, 2011. doi: https://doi.org/10.1016/j.neunet.2011.05.011.

Sofuoglu, S. E. and Aviyente, S. Graph regularized tensor train decomposition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3912–3916. IEEE, 2020.

sovrasov. flops-counter.pytorch. https://github.com/sovrasov/flops-counter.pytorch. Accessed: 2024-01-29.

Stoudenmire, E. M. and Schwab, D. J. Supervised learning with quantum-inspired tensor networks. *arXiv:1605.05775*, 2016.

Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for deep learning in NLP. *arXiv preprint arXiv:1906.02243*, 2019.

Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13693–13696, 2020.

Swall0w. torchstat. https://github.com/Swall0w/torchstat. Accessed: 2024-01-29.

Tamburrini, G. The AI carbon footprint and responsibilities of AI scientists. *Philosophies*, 7(1):4, 2022.

Thompson, N. C., Greenewald, K., Lee, K., and Manso, G. F. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 2020.

Tjandra, A., Sakti, S., and Nakamura, S. Compressing recurrent neural network with tensor train. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 4451–4458, 2017. doi: 10.1109/IJCNN.2017.7966420. ISSN: 2161-4407.

Tucker, L. R. Some Mathematical Notes on Three-Mode Factor Analysis. *Psychometrika*, 31(3):279–311, 1966. ISSN 00443409.

Vannieuwenhoven, N., Nicaise, J., Vandebril, R., and Meerbergen, K. On generic nonexistence of the schmidt–eckart–young decomposition for complex tensors. *SIAM Journal on Matrix Analysis and Applications*, 35(3):886–903, 2014.

Verdecchia, R., Sallou, J., and Cruz, L. A systematic review of green AI. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, pp. e1507, 2023.

Wahls, S., Koivunen, V., Poor, H. V., and Verhaegen, M. Learning multidimensional fourier series with tensor trains. In *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 394–398. IEEE, 2014.

Wang, M., Pan, Y., Yang, X., Li, G., and Xu, Z. Tensor networks meet neural networks: A survey. *arXiv preprint arXiv:2302.09019*, 2023.

Wesel, F. and Batselier, K. Large-scale learning with fourier features and tensor decompositions. *Advances in Neural Information Processing Systems*, 34, 2021.

Wu, B., Wang, D., Zhao, G., Deng, L., and Li, G. Hybrid tensor decomposition in neural network compression. *Neural Networks*, 132:309–320, 2020. ISSN 08936080. doi: 10.1016/j.neunet.2020.09.006.

Wu, C.-J., Raghavendra, R., Gupta, U., Acun, B., Ardalani, N., Maeng, K., Chang, G., Behram, F. A., Huang, J., Bai, C., et al. Sustainable AI: Environmental implications, challenges and opportunities. *arXiv:2111.00364*, 2021.

Yang, L. and Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 2020.

Yang, Y., Krompass, D., and Tresp, V. Tensor-train recurrent neural networks for video classification. *arXiv:1707.01786 [cs]*, 2017.

Yarally, T., Cruz, L., Feitosa, D., Sallou, J., and Van Deursen, A. Uncovering energy-efficient practices in deep learning training: Preliminary steps towards green ai. In *2023 IEEE/ACM 2nd International Conference on AI Engineering–Software Engineering for AI (CAIN)*, pp. 25–36. IEEE, 2023.

Ye, J., Wang, L., Li, G., Chen, D., Zhe, S., Chu, X., and Xu, Z. Learning compact recurrent neural networks with block-term tensor decomposition. In *Computer Vision and Pattern Recognition*, pp. 10, 2018.

Yin, M., Sui, Y., Liao, S., and Yuan, B. Towards efficient tensor decomposition-based dnn model compression with optimization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10674–10683, 2021.

Yu, R., Zheng, S., Anandkumar, A., and Yue, Y. Long-term forecasting using higher order tensor RNNs. *JMLR*, 2019.