VARIATIONAL BAYESIAN TENSOR NETWORKS WITH STRUCTURED POSTERIORS

Kriton Konstantinidis[1], *Yao Lei Xu*[1], *Qibin Zhao*[2], *Danilo P. Mandic*[1]

[1]Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ, UK
[2]Tensor Learning Team, RIKEN Center for Advanced Intelligence Project
E-mails: {k.konstantinidis19, yao.xu15, d.mandic}@imperial.ac.uk, qibin.zhao@riken.jp

ABSTRACT

Tensor network (TN) methods have proven their considerable potential in deterministic regression and classification related paradigms, but remain underexplored in probabilistic settings. To this end, we introduce a variational inference framework for supervised learning in the context of TNs, referred to as the Bayesian Tensor Network (BTN). This is achieved by making use of the multi-linear nature of tensor networks which allows us to construct a structured variational model which scales linearly with data dimensionality. The so imposed low rank structure on the tensor mean and Kronecker separability of the local covariances makes it possible to efficiently induce weight dependencies in the posterior distribution. This is shown to enhance model expressiveness at a drastically lower parameter complexity compared to the standard mean-field approach. A comprehensive validation of the proposed framework demonstrates the competitiveness of BTNs against existing structured Bayesian neural network approaches, while exhibiting enhanced interpretability, computational efficiency, and ability to yield credibility intervals.

Index Terms— Tensor Networks, Variational Inference, Regression, Bayesian supervised learning

1. INTRODUCTION

Tensor networks (TNs) have been recently employed in a variety of Machine Learning (ML) paradigms, owing to their: 1) suitability to operate with both dense and sparse data, 2) Big Data compatible properties, such as linear scaling with data dimensionality and constant scaling with the training size (assuming mini-batch gradient descent) [1], 3) enhanced interpretability, stemming from their multi-linear nature. Indeed, the potential of TNs has been demonstrated in areas including multi-modal learning [2], compression of large-dimensional data [3], sequence to sequence learning [4], anomaly detection [5], efficient tensor computation [6], and theoretical analysis of neural networks [7], to name but a few.

A vast majority of current deployments of TNs has been in deterministic settings, yet modern ML has tremendously benefited from probabilistic tools. To fill this void, and in order to make best use of both worlds, we introduce the Bayesian Tensor Network (BTN) model, in the form of a variational inference TN framework for probabilistic supervised learning.

The idea of treating model parameters as a realization from a higher-order distribution in the context of variational Bayesian deep learning was first introduced in [8], where a structured matrix-variate Gaussian was used to model the posterior distribution of a Bayesian Neural Network (BNN). We here move beyond NNs and consider models which are described fully in a TN form, in order to allow for a physically meaningful treatment of model parameters as a realization of a tensor-variate Gaussian posterior. Another related work [9] develops a tensor-variate Gaussian Process Prior Variational Autoencoder, in the context of learning latent representations, different from our focus on regression tasks. The work in [9] considers NNs and imposes a tensor structure on the latent space, while **our approach is quite general as it imposes structure on the TN model itself rather than on the latent space.** Finally, the data used in this work come in the form of standard feature vectors, as in a typical supervised learning context, rather than as multi-dimensional arrays.

The contributions of this work are as follows:

- A basic Mean-Field Variational Inference approach for TN based supervised learning is introduced.
- By leveraging on the multi-linear properties of TNs, we introduce a structured approach, whereby the learnable parameters are not treated independently, but as a realization of a tensor variate Gaussian random variable [10].
- This interpretation is shown to allow for efficient information sharing in the posterior distribution. This stems from a Kronecker structure of the tensor mean and covariances, which makes their forms more flexible while simultaneously allowing for more accurate predictions.
- A significantly reduced parameter complexity compared to the mean-field case.

Overall, to the best of our knowledge, this is the first work to consider variational inference in the context of TN methods for supervised machine learning, as well as the first work to consider structured posterior distributions that fully exploit the multi-linear nature of TNs, thus enabling meaningful model interpretations. The advantages of the proposed framework are demonstrated through the construction of a credibility interval based on the multi-linear properties of TNs. This approach, when coupled with domain-dependent and physically relevant feature maps (basis functions), promises to provide meaningful model interpretations without any sacrifice on model expressiveness.

2. PRELIMINARIES

2.1. Tensors and Tensor Networks

The following conventions for basic linear/multilinear operations are employed throughout the paper. The *outer product* of two vectors $\mathbf{a} \in \mathbb{R}^I$ and $\mathbf{b} \in \mathbb{R}^J$ is given by $\mathbf{c} = \mathbf{a} \circ \mathbf{b} \in \mathbb{R}^{I\times J}$, with $c_{i,j} = a_i b_j$. The *Kronecker product* of two matrices $\mathbf{A} \in \mathbb{R}^{I\times J}$ and $\mathbf{B} \in \mathbb{R}^{K\times L}$ is denoted by $\mathbf{C} = \mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{IK\times JL}$, with $c_{(i-1)K+k,(j-1)L+l} = a_{i,j}b_{k,l}$. The *Hadamard product* of two order-P tensors, $\mathcal{A} \in \mathbb{R}^{I_1\times\cdots\times I_P}$ and $\mathcal{B} \in \mathbb{R}^{I_1\times\cdots\times I_P}$ is denoted by $\mathcal{C} = \mathcal{A} \circledast \mathcal{B} \in \mathbb{R}^{I_1\times\cdots\times I_P}$, with $c_{i_1,\ldots,i_P} = a_{i_1,\ldots,i_P} b_{i_1,\ldots,i_P}$. Finally, the *inner product* of two order-P tensors $\mathcal{A} \in \mathbb{R}^{I_1\times\cdots\times I_P}$ and $\mathcal{B} \in \mathbb{R}^{I_1\times\cdots\times I_P}$ is denoted by $c = \langle \mathcal{A}, \mathcal{B}\rangle \in \mathbb{R}$ with $c = \sum_{i_1,\ldots,i_P} a_{i_1,\ldots,i_P} b_{i_1,\ldots,i_P}$.

The Canonical Polyadic Decomposition (CPD) [11, 12] is a tensor decomposition which expresses a tensor $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_P}$ as a sum of outer products of vectors $\mathbf{a}_r^{(1)}, \ldots, \mathbf{a}_r^{(P)}$ (i.e., rank-1 terms), that is:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \cdots \circ \mathbf{a}_r^{(P)} = \sum_{r=1}^{R} \bigcirc_{p=1}^{P} \mathbf{a}_r^{(p)}, \tag{1}$$

where R is the CP rank of the tensor (equal to the standard matrix rank when $P = 2$). We can arrange the vectors $\mathbf{a}_r^{(p)}$ into P factor matrices $\mathbf{A}^{(p)} = [\mathbf{a}_1^{(p)}, \ldots, \mathbf{a}_R^{(p)}] \in \mathbb{R}^{I_n\times R}$, $(1 \leq p \leq P)$, which allows the CPD to be expressed as a contraction [6] of the identity tensor, $\mathcal{I} \in \mathbb{R}^{I_1\times\cdots\times I_P}$ (with unit super-diagonal values) with the so-formed factor matrices. This view allows for the CPD to be formulated as a TN.

2.2. Tensor Networks for Supervised Learning

Consider a supervised learning task whereby each sample is represented through a set of P features [1]. A *local feature mapping*, ϕ: $\mathbb{R} \to \mathbb{R}^d$, is next applied to every feature, x_p, with d as the *local dimension* of the mapping. The choice of the feature map, ϕ, is flexible and application-dependent. The outer product between the so mapped feature vectors then yields

$$\Phi(\mathbf{x}) = \bigcirc_{p=1}^{P} \phi(x_p) \in \mathbb{R}^{d^P}, \tag{2}$$

where $\mathbb{R}^{d^P} = \mathbb{R}^{\overbrace{d \times \cdots \times d}^{P \text{ times}}}$. For one output (e.g., single-target regression or binary classification), the prediction in (2) is given by the inner product

$$g(\mathbf{x}) = \langle \Phi(\mathbf{x}), \mathcal{W}\rangle, \tag{3}$$

where $\mathcal{W} \in \mathbb{R}^{d^P}$ is the *weight tensor*, which comprises all model coefficients.

Remark 1. *The size of the weight tensor, $\mathcal{W}$, scales exponentially with the number of features and is therefore computationally prohibitive to learn. To this end, $\mathcal{W}$ can be represented as a TN [13], to ensure that the number of parameters scale linearly with the number of features.*

3. VARIATIONAL BAYESIAN TENSOR NETWORKS

Training a deterministic tensor network amounts to learning appropriate values for the entries of the TN-decomposed parameter tensor, $\mathcal{W}$. In order to extend this framework to the Bayesian setting, we shall treat model parameters as random variables, so that the training goal becomes that of calculating the posterior distribution of the weights given the training data, $p(\mathcal{W}|D)$.

Remark 2. *The posterior distribution, $p(\mathcal{W}|D)$, can be used for predictive purposes on unseen data, by averaging over the predictions of each possible configuration of the model parameters, and weighted according to the learnt posterior distribution. As is also the case with NNs, the true posterior distribution in TNs is intractable; we therefore resort to approximation through variational inference.*

In variational learning, the true posterior distribution, $p(\mathcal{W}|D)$, is approximated by a learnable tractable variational posterior, $q_{\boldsymbol{\theta}}(\mathcal{W}|D)$. In the sequel, the conditioning on D is removed from the notation for notational simplicity. Variational inference aims to learn the parameters, $\boldsymbol{\theta}$, that parameterize the weight distribution, $q_{\boldsymbol{\theta}}(\mathcal{W})$, by minimizing the Kullback-Leibler (KL) divergence between the true and variational posterior distributions. Following standard variational learning, it is straightforward to show that the optimal parameters, $\boldsymbol{\theta}^*$, can be found through the expression

$$\begin{aligned}\boldsymbol{\theta}^* &= argmin_{\boldsymbol{\theta}} KL\left[q_{\boldsymbol{\theta}}(\mathcal{W})||p(\mathcal{W}|D)\right] \\ &= argmin_{\boldsymbol{\theta}} KL\left[q_{\boldsymbol{\theta}}(\mathcal{W})||p(\mathcal{W})\right] - \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathcal{W})}[\log p(D|\mathcal{W})]\end{aligned} \tag{4}$$

which admits the use of stochastic gradient descent in its optimization. Denote the argument of the $argmin$ function in equation (4) by $\mathcal{F}(D, \boldsymbol{\theta})$. Then, $\mathcal{F}(D, \boldsymbol{\theta})$ can be approximated through Monte-Carlo sampling as [14]

$$\mathcal{F}(D, \boldsymbol{\theta}) \approx \sum_{i=1}^{N} \log q_{\boldsymbol{\theta}}(\mathcal{W}^{(i)}) - \log p(\mathcal{W}^{(i)}) - \log p(D|\mathcal{W}^{(i)}) \tag{5}$$

where $\mathcal{W}^{(i)}$ is a sample from the variational posterior, $q_{\boldsymbol{\theta}}(\mathcal{W})$.

3.1. The Mean-Field Approach

In the common mean-field approach [14], the weight posterior is approximated by a diagonal multivariate Gaussian distribution, leading to mean-field Bayesian tensor networks (MF-BTN), whereby each weight is treated as coming from a separate univariate distribution while the learning procedure consists of separately learning the mean and variance. For more detail, we refer the reader to [14]. Notice that this approach leads to a higher (double) parameter complexity compared to the deterministic approach, since a separate mean and variance are learnt for every weight.

Remark 3. *Even though the mean-field approximation allows for tractable and fast inference over the weight distributions, the independence assumption over the weights proves to be too restrictive [8]. To this end, a structured approach that embarks upon the structural properties of TNs to enhance flexibility in the posterior is a prerequisite, and is introduced next.*

3.2. Tensor-Variate Structured Posterior

Without loss of generality, the proposed Structured Posterior Bayesian Tensor Networks (SP-BTN) is introduced for the case of CPD, as it is considered in our regression type experiments.

Tensor Gaussian distribution. A random tensor, $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, exhibits a Gaussian distribution, defined by the tensor mean, $\mathcal{M} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and mode-n covariance, $\mathbf{R}^{(n)} \in \mathbb{R}^{I_n \times I_n}$, if and only if its vector representation, $\mathbf{x} = vec(\mathcal{X}) \in \mathbb{R}^K$, where $K = \prod_{i=1}^{N} I_i$, is distributed according to [15]

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{m}, \bigotimes_{i=N}^{1} \mathbf{R}^{(i)}\right) \tag{6}$$

where $\mathbf{m} = vec(\mathcal{M})$.

Reinterpretation of the weight tensor. Note that the learnable weights of a CPD-decomposed weight tensor, $\mathcal{W}$, form a 3rd order tensor. Indeed, we have P factor matrices, each of dimensionality $d \times R$, as defined[1] in Section 2.

Remark 4. *Instead of considering every weight, $w \in \mathcal{W}$, as being drawn from an independent univariate Gaussian distribution, the weight structure of the CPD admits the following reinterpretation. Namely, the parameters of a CPD-decomposed weight tensor, $\mathcal{W}$, can be considered as a single realization of a tensor-variate Gaussian distribution with the tensor mean, $\mathcal{M} \in \mathbb{R}^{d \times R \times P}$, and mode-n covariance, $\mathbf{R}^{(n)} \in \mathbb{R}^{n \times n}$, where $n \in \{d, R, P\}$.*

When adopting the weight tensor reinterpretation of the variational learning framework, it can be assumed that the variational posterior of the weight tensor is also a tensor-variate Gaussian distribution. Note that the parameters of a tensor-variate Gaussian cannot be estimated by simply e.g., maximum likelihood estimation, as this is only possible with tensor-valued observations. In contrast, we here deal with typically lower dimensional feature vectors and labels. The probability density function of the tensor valued variational posterior is thus defined as

$$q_{\boldsymbol{\theta}}(\mathcal{W}) = \frac{\exp\left[-\frac{1}{2}(\mathbf{w}-\mathbf{m})^T(\bigotimes_{i\in\{d,R,P\}}\mathbf{R}^{(i)})^{-1}(\mathbf{w}-\mathbf{m})\right]}{(2\pi)^{\frac{dRP}{2}}\det^{\frac{1}{2}}\left[\bigotimes_{i\in\{d,R,P\}}\mathbf{R}^{(i)}\right]} \tag{7}$$

where $\boldsymbol{\theta} = \{\mathbf{m}, \mathbf{R}^{(i)}\}$ denotes the trainable variational parameters.

Next, instead of considering a separate mean for every weight distribution, we impose a low rank structure on the tensor mean, $\mathcal{M}$, in the form

$$\mathcal{M} = \sum_{m=1}^{M_R} \mathbf{b}_m^{(d)} \circ \mathbf{b}_m^{(R)} \circ \mathbf{b}_m^{(P)} \tag{8}$$

where M_R denotes the mean rank [10].

This approximation further reduces parameter complexity while preserving the expressiveness of the posterior by introducing weight dependencies. More specifically, the proposed approximation amounts to $(d + R + P)M_R$ parameters, in comparison to the dRP parameters in the mean-field case. In practice, it was found that very low ranks were too restrictive, but after a relatively low rank (e.g, $M_R = 10$), the posterior mean was flexible enough for efficient model learning.

Prior distribution. We have chosen the multivariate diagonal Gaussian distribution given by

$$p(\mathcal{W}) = \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_{dRP}) \tag{9}$$

as the prior distribution. In this work, we diverge from the approach in [14], where a scale mixture of two Gaussian densities was suggested in order to a-priori cluster many weights around zero. Also, TNs can become sensitive to initialization, especially for large P, due to the many matrix (for CPD) or tensor (for the tensor train) multiplications, as has been previously pointed out in [1, 5]. In other words, the final result can vanish or explode if every initialized matrix/tensor is too small or too big. We have found that, in general, a normal distribution with standard deviation equal to 0.5 worked well for our experiments.

Tensor posterior sampling. For efficient stochastic variational inference, it is necessary to extend the reparameterization trick [14] for the tensor-variate posterior distribution. In this case, sampling from the posterior amounts to sampling random tensor noise, $\mathcal{E} \in \mathbb{R}^{d \times R \times P}$, from a standard Gaussian distribution. A sample from the posterior is then obtained as $\mathcal{W}^{(i)} = \mathcal{M} + \mathcal{E} \times L_d \times L_R \times L_P$, where L_i denotes the Cholesky factor of the corresponding mode-n covariance matrix $\mathbf{R}^{(i)}, i \in \{d, R, P\}$, and $\times$ denotes the mode-n product.

[1]Note that in the case of CPD, even though $\mathcal{W}$ is originally a P-th order tensor, this is achieved through the contraction of the factor matrices with the superdiagonal tensor, as explained in Section 2. In this case, we can employ P factor matrices, each of size $d \times R$, to yield a 3rd order tensor.

KL-Divergence between tensor-variate Gaussians. It was empirically found that training stability is significantly enhanced when using the closed-form KL divergence. To explicitly obtain the KL-divergence between the prior and the variational posterior, we employ (6) and the standard KL-divergence between two multivariate Gaussians, to yield

$$
\begin{aligned}
&KL\left[q_{\boldsymbol{\theta}}(\mathcal{W})||p(\mathcal{W})\right] = \\
&= \frac{1}{2}[\mathrm{tr}\left(\left(\bigotimes_{i\in\{d,R,P\}}\mathbf{R}_p^{(i)}\right)^{-1}\bigotimes_{i\in\{d,R,P\}}\mathbf{R}_q^{(i)}\right) \\
&+ \left(\mathbf{m}_p-\mathbf{m}_q\right)^T\left(\bigotimes_{i\in\{d,R,P\}}\mathbf{R}_p^{(i)}\right)^{-1}\left(\mathbf{m}_p-\mathbf{m}_q\right) \\
&- dRP + \log\frac{\det\bigotimes_{i\in\{d,R,P\}}\mathbf{R}_p^{(i)}}{\det\bigotimes_{i\in\{d,R,P\}}\mathbf{R}_q^{(i)}}] \qquad (10)
\end{aligned}
$$

The cost in (4) can now be minimized using gradient descent, with the expression in (10) for the closed form KL-divergence and the proposed posterior given in (7).

4. EXPERIMENTS

The experimental setups were conceived to: 1)provide a comparison between the proposed framework and state-of-the-art BNNs, and 2) to illustrate the additional benefits of a probabilistic TN model, such as its ability to provide credibility intervals.

4.1. Benchmark Regression Datasets

Experiments were performed on benchmark UCI regression datasets [16], by employing polynomial feature mapping with unit norm and CPD for the representation of $\mathcal{W}$ [1]. The scores were reported only for SP-BTN, as they were consistently higher than those for MF-BTN at a comparable parameter complexity. The proposed SP-BTN was evaluated against the Bayes By Backprop (BBB) [14], Variational Matrix Gaussian (VMG) [8], and functional BNN (fBNN) [17] models. The BTN models were trained until convergence with the Adam optimizer [18], at a learning rate of 0.001, and a mini-batch size of 128.

For a fair comparison between BTNs and BNNs, the BTN models were kept simple, with $R = 10$ and $d = 10$, while M_R was ranging between 10 and 30; higher values were used in datasets where $M_R = 10$ turned out to be too restrictive (and cause underfit on the training set). These settings led to parameter complexity similar to that of the BNNs (one hidden layer with 50 units, as reported in [17]). Table 1 shows the mean and standard deviations for the models considered. Similarly, for the large scale regression datasets, values for R, d, and M_R were chosen so as to lead to similar parameter complexity as the BNNs (one hidden layer with 100 units, as reported in [17]). The mean and standard deviation for different models are shown Table 2. In both cases, the experimental procedure was identical to that in [17].

Table 1: Averaged test RMSE for small scale regression datasets.

Data	BBB	VMG	fBNN	SP-BTN
Wine	0.643 ± 0.012	$\mathbf{0.63 \pm 0.01}$	0.673 ± 0.014	0.6417 ± 0.011
Concrete	5.678 ± 0.08	$\mathbf{4.89 \pm 0.12}$	4.9355 ± 0.18	5.50 ± 0.23
Energy	0.565 ± 0.018	0.54 ± 0.02	$\mathbf{0.412 \pm 0.017}$	0.549 ± 0.2
Yacht	1.174 ± 0.086	0.71 ± 0.05	0.607 ± 0.068	$\mathbf{0.5061 \pm 0.091}$

Table 2: Averaged test RMSE for large scale regression datasets.

Data	BBB	fBNN	SP-BTN
Protein	4.331 ± 0.033	4.326 ± 0.019	$\mathbf{4.25 \pm 0.012}$
Naval	0.01 ± 0.0	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
GPU	21.886 ± 0.673	19.50 ± 0.171	$\mathbf{19.36 \pm 0.212}$

Overall, it can be observed that the proposed BTN models exhibited a comparable performance to state-of-the-art BNNs, at a similar parameter complexity but with enhanced physical intuition.

4.2. Construction of the Credibility Interval

The proposed SP-BTN models was next tested on fitting a benchmark dataset that was used in [16], whereby 20 inputs were sampled from $\mathcal{U}[-4, 4]$ so as to construct the target variable, $y_n = x_n^3 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 9)$. To illustrate the enhanced interpretability of the proposed TN based Bayesian setting, the coefficient computation for each (transformed) feature interaction were obtained as described in [1].

Remark 5. *The stochastic nature of the proposed BTN framework makes it possible to compute credibility intervals for individual coefficients, unlike with the existing approaches.*

In this example, after model training, the estimated single regression coefficient taking part in the model employed was 0.97 ± 0.05, which includes the ground truth value (equal to 1).

5. CONCLUSION

We have introduced variational inference into the tensor network representation, a novel framework for employing compressed models in the Bayesian setting (BTN). In this way, a structured model has been established that takes advantage of the multi-linear nature of TNs to efficiently induce information sharing in the posterior distribution of the model weights. The so introduced structured BTN has been shown not only to provide competitive results against existing BNN models in benchmark regression datasets, but also to enable enhanced physical interpretability. While the present work has focused on the regression paradigm, the BTN framework can be readily extended to other domains, such as classification and clustering, through appropriate modifications of the tensor network used to decompose the weight tensor (e.g., 2-D tensor networks for images), a subject of future work.

6. REFERENCES

[1] A. Haliassos, K. Konstantinidis, and D. P. Mandic, "Supervised learning for non-sequential data: A Canonical Polyadic Decomposition Approach. *IEEE Transactions on Neural Networks and Learning Systems*," in print, 2021.

[2] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and A. H. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, March 2015.

[3] A. Cichocki, N. Lee, I. Oseledets, A. Phan, Q. Zhao, and D. P. Mandic, "Tensor networks for dimensionality reduction and large-scale optimization. Part 1 Low-rank tensor decompositions," *Foundations and Trends® in Machine Learning*, vol. 9, no. 4-5, pp. 249–429, 2016.

[4] C. Guo, Z. Jie, W. Lu, and D. Poletti, "Matrix product operators for sequence-to-sequence learning," *Physical Review E*, vol. 98, pp. 042114, Oct 2018.

[5] J. Wang, C. Roberts, G. Vidal, and S. Leichenauer, "Anomaly detection with tensor networks," in *Proceedings of the First Workshop on Quantum Tensor Networks in Machine Learning, 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[6] I. Kisil, G. Calvi, K. Konstantinidis, Y. L. Xu, and D. P. Mandic, "Reducing computational complexity of tensor contractions via tensor-train networks", *arXiv* 2109.00626," 2021.

[7] N. Cohen, O. Sharir, and A. Shashua, "On the expressive power of deep learning: A tensor analysis," in *Proceedings of the Conference on Learning Theory*, 2016, pp. 698–728.

[8] C. Louizos and M. Welling, "Structured and efficient variational deep learning with matrix Gaussian posteriors," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 1708–1716.

[9] A. Campbell and P. Liò, "TVGP-VAE: Tensor-variate Gaussian process prior variational autoencoder, *arXiv* 2006.04788," 2020.

[10] B. Scalzo Dees, A. H. Phan, and D. P. Mandic, "A statistically identifiable model for tensor-valued Gaussian random variables", *arXiv* 1911.02915," 2019.

[11] J. D. Carroll and J. J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[12] R. A. Harshman, *Foundations of the PARAFAC Procedure: Models and conditions for an "explanatory" multi-modal factor analysis*, UCLA working papers in phonetics. University of California at Los Angeles, 1970.

[13] E. Stoudenmire and D. Schwab, "Supervised learning with tensor networks," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 4799–4807.

[14] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 1613–1622.

[15] M. Singull, M. Ahmad, and D. von Rosen, "The multilinear normal distribution: Introduction and some basic properties," *Journal of Multivariate Analysis*, vol. 113, pp. 37-47, 2011.

[16] J. M. Hernandez-Lobato and R. Adams, "Probabilistic backpropagation for scalable learning of Bayesian neural networks," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 1861–1869.

[17] S. Sun, G. Zhang, J. Shi, and R. Grosse, "Functional variational Bayesian neural networks," in *Proceedings of the International Conference on Learning Representations*, pp. 1-22, 2019.

[18] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, pp. 1-15, 2015.