Tensor Solutions - Research
HPC TTN contraction for machine learning real-time applications
from: The Tensor Solutions team,
November 22, 2022

Master thesis: HPC TTN contraction for machine learning real-time applications

Required work

- Tensor HPC computing and tensor algebra.
- GPU/TPU coding.
- Industrial-standard code development.
- Efficient contraction strategy for Tensor Network Machine Learning (single and multi sample) predictions.
- Prediction for data with missing values.

In a machine learning context, the prediction of a Tree Tensor Network (TTN) learner is performed via a contraction of the full network with a (tensorised) sample of the dataset. This contraction can be carried out in many different ways and allows parallelism and different exploits such as:

- reshaping internal tensors to better fit the hardware;
- CPU/GPU/TPU parallelisation over single or multiple contractions or over multiple samples, based on the size of the problem and of the available hardware;
- ordering of the contraction to perform;

Having an optimal strategy for fast predictions is of the uttermost importance in every ML scenarios but especially in real-time deployment of ML models. Some of the most prominent applications field range from: object-detection in autonomous-driving, on-line data processing in manifacturing, big data finance analysis, etc.

Tensor Solutions - Research
HPC TTN contraction for machine learning real-time applications
from: The Tensor Solutions team,
November 22, 2022

1 Background

Tensor Solutions is a BMWi-funded start-up project that aims to make the field of Artificial Intelligence (AI) more transparent, comprehensible and efficient. Our core technology, the Tensor Networks, were developed over the last 30 years to simulate quantum many-body systems on classical computers. At Tensor Solutions, we have further developed this technology with our expertise in both quantum physics and data science to solve machine learning problems in various sectors of industry.

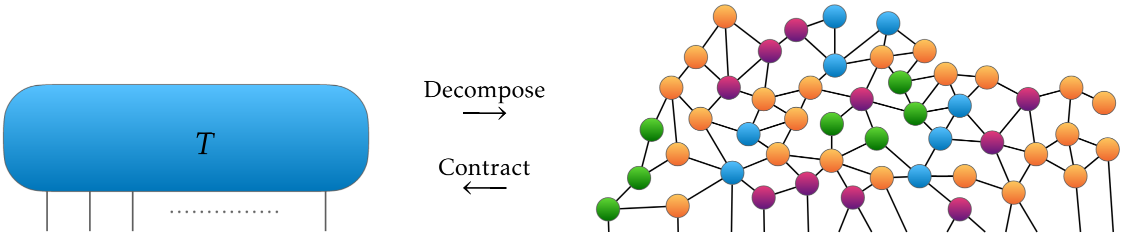

Figure 1: Left: High-order tensor T representing information with exponentially many coefficients. Right: A possible Tensor Network decomposition of the same information where the tensors are illustrated in different colors and connected with each other over *internal bond-links*. The dimension of the bond-links can be control to compress the information represented within the network.

Tensor Networks, as illustrated in Fig. 1, decompose a large tensor into a set of smaller tensors that are connected over some auxiliary indices being summed over, called *bond-links*. In this representation, the amount of information can be controlled by a *bond-dimension* m. Representing the information in such a Tensor Network gives raise to major benefits, such as an exponential reduction in memory, an exponential speedup of computations, a theoretical insight and interpretation, an estimation of missing or corrupted entries and many optimisation algorithms and strategies. Consequently, Tensor Networks are an efficient linear algebra tool for problems in exponentially large, high-dimensional spaces. (for more information, see http://dx.doi.org/10.22028/D291-35211)

A Tree Tensor Network Ψ is a collection of rank-3 tensors $\mathcal{T}$ with a binary-tree pattern of connections (implicitly contracted indices). In ML, we call ℓ the open label index and $p_{i=1,\dots,n}$ the set of open physical links. In this framework the TTN is seen as the group of weights (to be optimised) representing a learner. We can then write the TTN learner as

$$\Psi^{\ell}_{p_1\dots p_n} = \mathcal{T}^{\ell}_{v_1 v_2}\mathcal{T}^{v_1}_{v_3 v_4}\mathcal{T}^{v_2}_{v_5 v_6}\dots\mathcal{T}^{v_{2n-2}}_{p_{n-1} p_n}. \tag{1}$$

Given a sample $\mathbf{x}$ composed of n features $x_1,\dots,x_n$, the feature map

$$\Phi = \bigotimes_i \phi_i \tag{2}$$

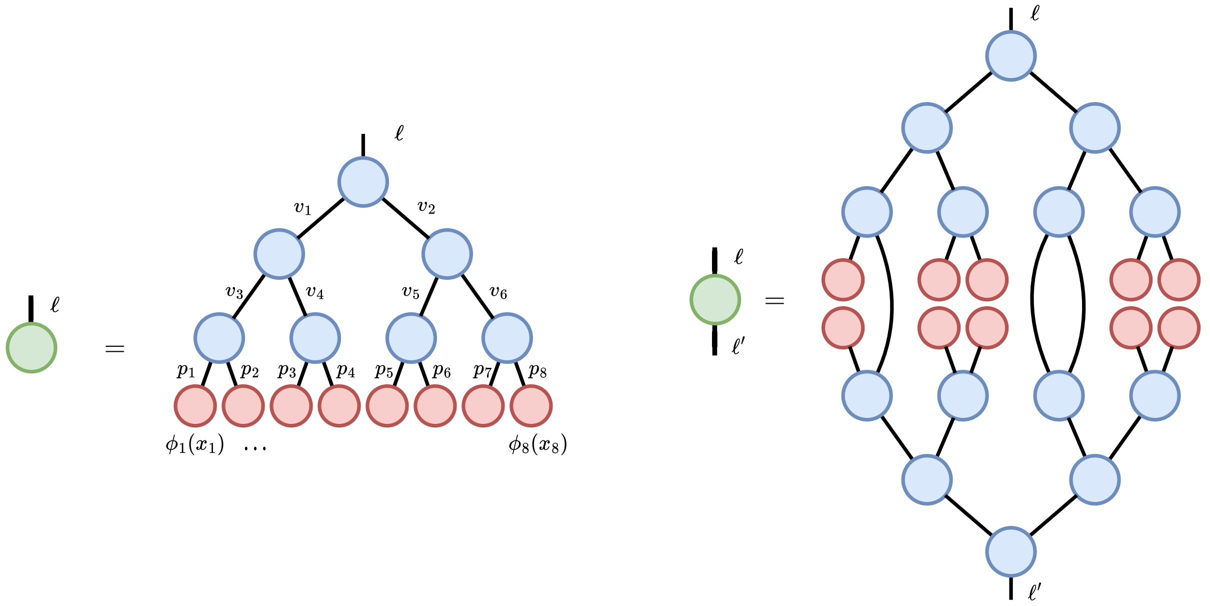

Figure 2: Prediction schemes for TTNs. In blue the Tree Tensor network learner (TTN), in red the tensorised samples (row) of a dataset, in green the predicted confidences. (Left) A full sample is contracted to predict a vector of confidences. (Right) A sample with missing values is predicted as an expected value over the sample operator.

is applied on each feature, as

$$\Phi(\mathbf{x})_{p_1 \dots p_n} = \phi_1(x_1)_{p_1} \otimes \phi_2(x_2)_{p_2} \otimes \dots \otimes \phi_n(x_n)_{p_n}. \tag{3}$$

A prediction is then the contraction of the TTN learner and the mapped data sample

$$y^\ell = \Psi^\ell(\Phi(\mathbf{x})) = \Psi^\ell_{\{p\}_i} \otimes \Phi(\mathbf{x})_{\{p\}_i}. \tag{4}$$

If the data sample contain missing values, the we can write a prediction as an expectation value over the operator $\Phi(\mathbf{x})^T \Phi(\mathbf{x})$ where all the missing values are then implicitly contracted physical links (see right of Fig. 2). That is, the predicted confidences are the element-wise square root of the diagonal of matrix

$$Y^{\ell\ell'} = \Psi^{\ell'}(\Phi(\mathbf{x}))^T \otimes \Psi^\ell(\Phi(\mathbf{x})). \tag{5}$$