Multi-Graph Tensor Networks

Imperial College London

Yao Lei Xu, Kriton Konstantinidis, Danilo P. Mandic

Department of Electrical and Electronic Engineering
Imperial College London

Introduction

The multi-modal and irregular nature of modern big data is posing stern challenges to traditional learning systems, owing to the sheer volume, variety, veracity and velocity of modern data sources [1]. To this end, it is necessary to generalize deep learning approaches to handle such irregular and multi-modal data.

Some of the most successful approaches for data analytics on irregular domains resort to graph signal processing techniques, because of their ability to provide insights into the underlying data geometry [2]. When it comes to exceedingly large multi-modal data, tensor-based methods have demonstrated their potential in effectively bypassing the bottlenecks imposed by the curse of dimensionality in various learning tasks [1].

To provide a general framework that fully exploits the advantages of both graphs and tensors in a deep learning setting, we here generalize the RGTN concept in [3] to introduce the novel *Multi-Graph Tensor Network* (MGTN). In this way, the proposed MGTN is capable of handling irregular data residing on multiple graph domains, while simultaneously leveraging the compression properties of tensor networks to enhance modelling power and reduce parameter complexity.

Fast Multi-Graph Tensor Network Model

Consider a multi-graph learning problem where the input is an order-$(M+1)$ tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{J_0 \times I_1 \times I_2 \times \cdots \times I_M}$ with J_0 features indexed along M physical modes $\{I_1, I_2, \dots, I_M\}$, such that a graph $\mathcal{G}^{(m)}$ is associated with each of the I_m modes, $m = 1, \dots, M$. For this problem, we define:

1. $\mathcal{A} = \{\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(M)}\}$, a set of adjacency matrices $\mathbf{A}^{(m)} \in \mathbb{R}^{I_m \times I_m}$ constructed from the corresponding graphs $\mathcal{G}^{(m)}$.
2. $\mathcal{W} = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \dots, \mathbf{W}^{(M)}\}$, a set of weight matrices $\mathbf{W}^{(m)} \in \mathbb{R}^{J_m \times J_{m-1}}$ used for feature transforms, where J_m, for $m = 1, \dots, M$ controls the number of feature maps at every m.
3. $\mathcal{P} = \{\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \dots, \mathbf{P}^{(M)}\}$, a set of propagation matrices $\mathbf{P}^{(m)} \in \mathbb{R}^{J_m \times J_m}$, modelling the propagation of information over the neighbours of the graph $\mathcal{G}^{(m)}$.

Using the above objects and an optional activation function, $\sigma(\cdot)$, we can now define the general Multi-Graph Tensor Network (gMGTN) layer characterized by the following forward pass:

$$\boldsymbol{\mathcal{Y}} = \sigma\left(\boldsymbol{\mathcal{F}}^{(M)} \times_{3,4}^{1,M+1} \mathbf{W}^{(M)} \times_2^1 \cdots \times_2^1 \boldsymbol{\mathcal{F}}^{(2)} \times_{3,4}^{1,3} \mathbf{W}^{(2)} \times_2^1 \boldsymbol{\mathcal{F}}^{(1)} \times_{3,4}^{1,2} \mathbf{W}^{(1)} \times_2^1 \boldsymbol{\mathcal{X}}\right) \quad (1)$$

where $\boldsymbol{\mathcal{F}}^{(m)} = \text{ten}(\mathbf{I} + (\mathbf{A}^{(m)} \otimes \mathbf{P}^{(m)}))$. The so defined forward pass generates a feature map, $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{J_M \times I_1 \times \cdots \times I_M}$, from the input tensor, $\boldsymbol{\mathcal{X}}$, through a series of multi-linear graph filter and weight matrix contractions, which essentially iterates the graph filtering operation across all M graph domains.

We can reduce the parameter complexity of gMGTN by: (i) approximating $\mathbf{P}^{(m)} \approx \mathbf{I}$ for $m = 1, \dots, M$; and (ii) using one single weight matrix, $\mathbf{W}^{(x)} \in \mathbb{R}^{J_1 \times J_0}$, for all of the graph domains, where J_1 controls the number of hidden units (feature maps). This allows us to simplify the graph filter to $\mathbf{F}^{(m)} = (\mathbf{I} + \mathbf{A}^{(m)})$, which leads to the fast Multi-Graph Tensor Network (fMGTN) forward pass:

$$\boldsymbol{\mathcal{Y}} = \sigma\left(\mathbf{F}^{(M)} \times_2^{M+1} \cdots \times_2^4 \mathbf{F}^{(2)} \times_2^3 \mathbf{F}^{(1)} \times_2^2 \mathbf{W}^{(1)} \times_2^1 \boldsymbol{\mathcal{X}}\right) \quad (2)$$

Model Architecture

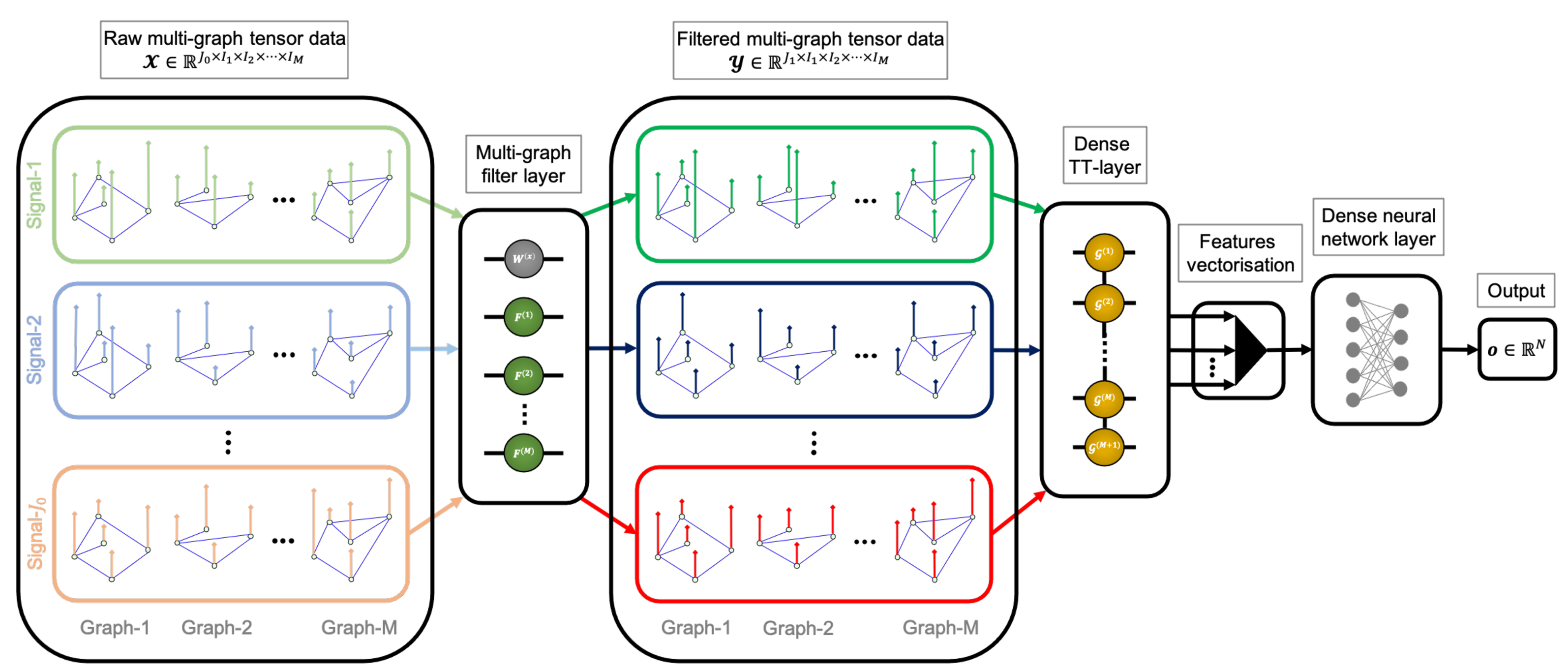

Fig. 1: Data processing diagram of the proposed fMGTN model

Figure 1 illustrates the workings of the proposed fMGTN framework, where the input data tensor, $\boldsymbol{\mathcal{X}}$, is forward-passed from left-to-right to generate the final output vector, $\mathbf{o}$. The given model takes as input a tensor valued sample, where J_0 different signals are indexed along M different physical modes, where each physical mode is associated with a physical graph domain with I_m nodes. The given input tensor is passed through the multi-graph filter layer (represented in tensor network notation), which generates a filtered representation of the signals while maintaining the underlying multi-graph and multi-dimensional structure. This multi-linear graph filtering constitutes a highly localized filtering operation, where locality is defined with respect to the topology of the graph. The filtered multi-graph tensor data is then passed through a dense layer in TTD format (represented in tensor network notation), which combines the localized features maps extracted previously via a global multi-linear map. The resulting global features are then vectorized and passed through a final dense neural network layer to generate the desired output.

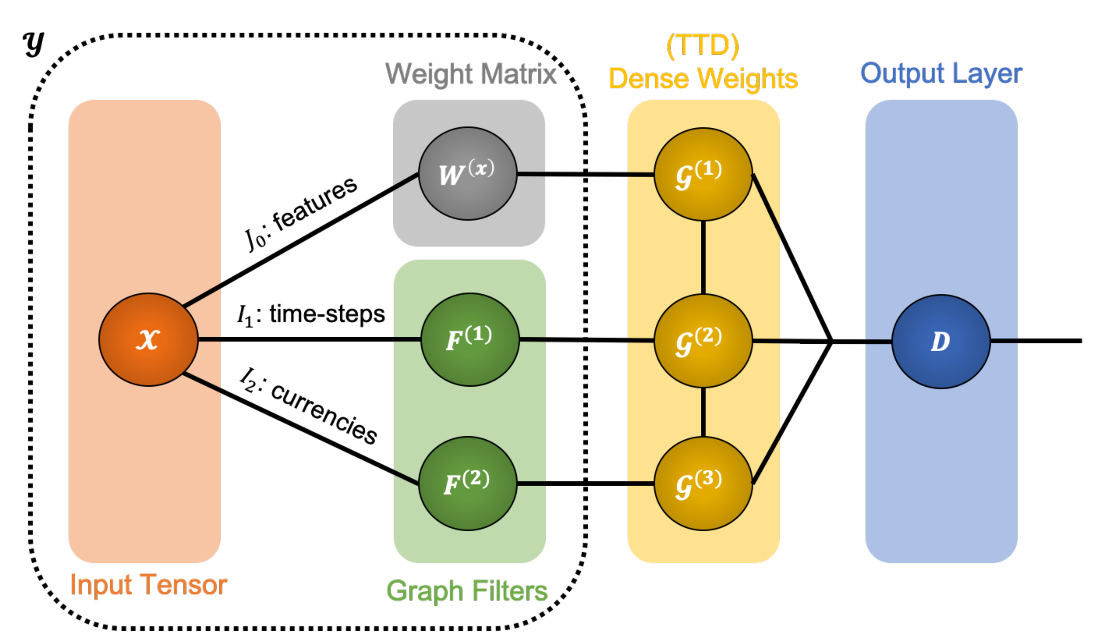

Fig. 2: Tensor network diagram of the proposed fMGTN model

Figure 2 illustrates the proposed model architecture used in our algorithmic trading experiment as a tensor network. The section encircled in dotted line denotes the multi-graph filtering operation for $M = 2$ as per equation (2). The yellow region denotes a tensorized dense layer weight matrix, represented in the Tensor-Train format, which is inherently compatible with the given problem structure. The input data used for our experiment is an order-3 tensor with $J_0 = 4$ pricing features, $I_1 = 30$ past time-steps, and $I_2 = 9$ currencies, as it will be discussed in the experiment section. Note that we define a time-domain graph filter and a currency-domain graph filter for input data modes of dimensions I_1 and I_2, respectively.

Algorithmic Trading Experiment

The proposed fMGTN is implemented and compared against three competing models in the context of algorithmic trading: a notoriously difficult paradigm characterized by high-dimensional, multi-modal, noisy, and irregular data. Specifically, the proposed models are implemented in the double deep-Q learning setting as action-value approximation networks.

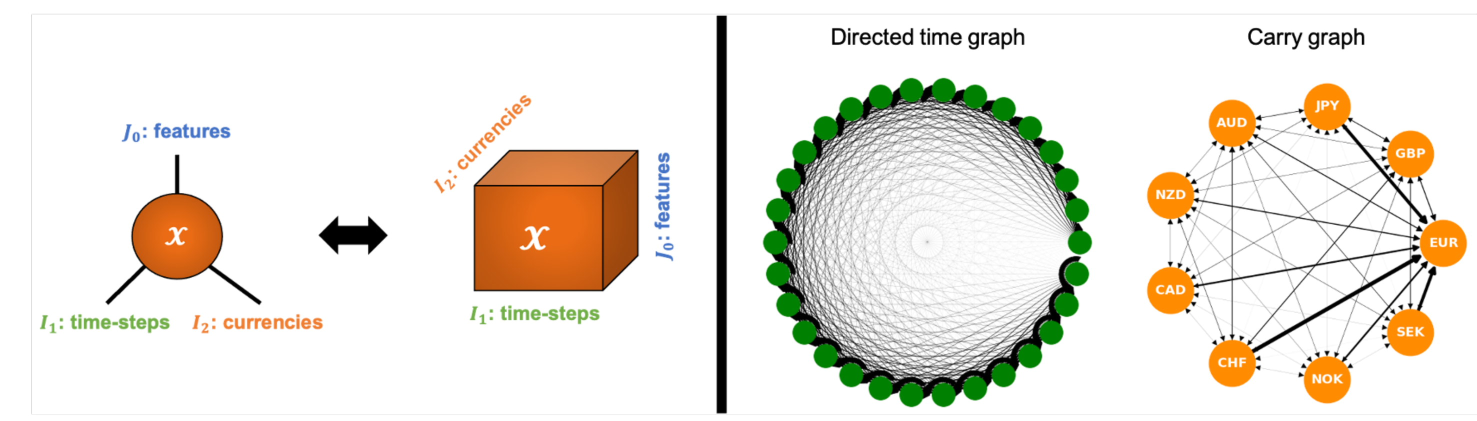

Fig. 3: Input data tensor (left) and the corresponding graph domains (right)

Figure 3 illustrates the data tensor structure for the algorithmic trading experiment, where J_0 features are indexed along both the time-domain and the currency (carry) domain. The time-domain graph is a directed graph where past states can influence future ones but not vice-versa, while the carry graph is a directed graph where the connection between currencies are proportional to the pairwise carry factor. In both graphs, thicker edges indicate stronger connection.

The proposed fMGTN based network is shown to out-perform all other models in consideration across a number of commonly used financial metrics, while using only a fraction of trainable parameters, as shown in the figures below.

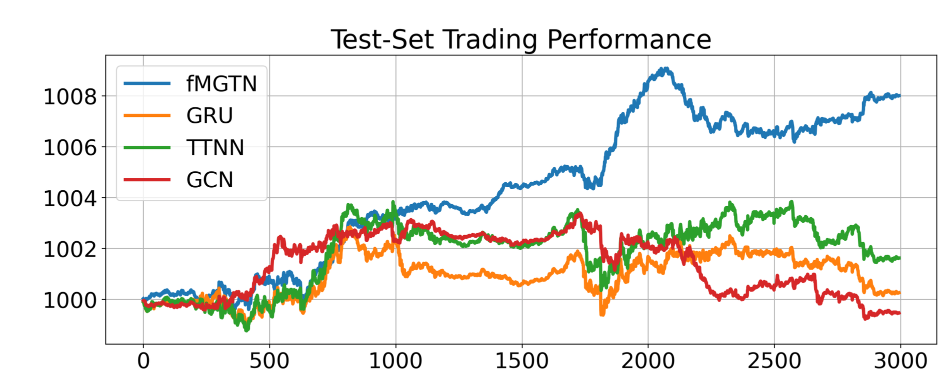

Fig. 4: Backtest trading performance

Agent	TR (%)	SR	MD (%)	HR(%)	NP
fMGTN	**0.8018**	**0.0445**	**0.2893**	**52.8056**	**531**
GRU	0.0260	0.0012	0.3477	50.4008	3107
TTNN	0.1628	0.0064	0.3493	50.6346	451
GCN	-0.0538	-0.0032	0.4180	50.2338	5891

Fig. 5: Performance Metrics

References

[1] A. Cichocki. "Era of Big Data Processing: A New Approach via Tensor Networks and Tensor Decompositions". In: *Proceedings of the International Workshop on Smart Info-Media Systems in Asia*. Mar. 2014.

[2] L. Stankovic et al. "Graph Signal Processing–Part II: Processing and Analyzing Signals on Graphs". In: *arXiv preprint arXiv:1909.10325* (2019).

[3] Y. L. Xu and D. P. Mandic. "Recurrent Graph Tensor Networks". In: *arXiv preprint arXiv:2009.08727* (Sept. 2020). arXiv: 2009.08727 [cs.LG].