



Skolkovo Institute of Science and Technology

MASTER'S THESIS

Selection of Tensor Representations For Multiview Forecasting

Master's Educational Program: Data Science

Student: _____ Nadezhda Alsahanova
signature

Research Advisor: _____ Maxim Panov
signature
PhD, Professor

Co-Advisor _____ Vadim Strijov
signature
Doctor of Physical and
Mathematical Sciences,
Professor

Moscow 2022

Copyright 2022 Author. All rights reserved.

The author hereby grants to Skoltech permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole and in part in any medium now known or hereafter created.



Skolkovo Institute of Science and Technology

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

**Выбор тензорных представлений для прогнозирования по
мультимодальным измерениям.**

Магистерская образовательная программа: Науки о данных

Студент: _____ Надежда Алсаханова
подпись

Научный руководитель: _____ Панов Максим
подпись Евгеньевич
к.ф.-м.н., профессор

Со-руководитель _____ Стрижов Вадим
подпись Викторович
д.ф.-м.н., профессор

Москва 2022

Авторское право 2022. Все права защищены.

Автор настоящим дает Сколковскому институту науки и технологий разрешение на воспроизводство и свободное распространение бумажных и электронных копий настоящей диссертации в целом или частично на любом ныне существующем или созданном в будущем носителе.

Selection of Tensor Representations For Multiview Forecasting

Nadezhda Alsahanova

Submitted to the Skolkovo Institute of Science and Technology on December 16, 2022

ABSTRACT

Brain-computer interfaces (BCI) are a rapidly developing research area that can help patients to restore their movement ability. The measured brain signals are highly correlated and redundant. Therefore, dimensionality reduction methods are used for stable and accurate prediction. To improve data compression during dimensionality reduction, hankelization for time series is used. However, this method has not been used in the forecasting problem in the area of BCI. In this work, we found optimal representations of feature and target tensors in latent space by combining hankelization and dimensionality reduction methods. We tested hankelization along the temporal, along the spatial, and along both dimensions. In addition, we used tree of the most common and modern dimensionality reduction models to check whether hankelization works for all models. The results on three BCI datasets showed that hankelization along both dimensions improves prediction quality the most and simplifies the high-order partial least squares model. Additionally, hankelization along spatial dimension works the best way for MPCA in many cases. These results show that hankelization can improve forecasting quality in the BCI area.

Keywords: signal decoding, ECoG, partial least squares, autoencoder, tensor regression

Research advisor:

Name: Maxim Panov

Degree, title: PhD, Professor

Co-advisor:

Name: Vadim Strijov

Degree, title: Doctor of Physical and Mathematical Sciences, Professor

Contents

1	Introduction	5
2	Problem statement	7
2.1	Notation	7
2.2	Problem statement	8
3	Methodology	10
3.1	Hankelization	10
3.2	MPCA	11
3.3	Standard Partial Least Squares	12
3.4	High-Order PLS	12
3.5	Autoencoder ReducedNet	13
3.6	Multimodal regression	15
3.7	Tensor regression	16
4	Numerical experiments	17
4.1	Datasets	17
4.1.1	Neurotycho dataset	17
4.1.2	Multimodal EEG dataset	18
4.2	Metrics	18
4.3	Results	19
4.3.1	MPCA	19
4.3.2	HOPLS	19
4.3.3	TensorReducedNet	20
5	Discussion and conclusion	23

Chapter 1

Introduction

An initial data in the brain-computer interface's (BCI) task typically has attributes of high dimensionality, strong input correlation, and low signal-to-noise ratio. Dimensionality reduction methods are used to reduce redundancy of this data and so increase the speed of algorithms and their quality. It has previously been observed that tensorization, the generation of higher-order structured tensors from the lower-order data formats, helps to find low-rank approximation with a high level of compression and to reveal hidden correlations. There are many methods of tensorization that can be used in BCI. One of them is hankelization. Several attempts have been made to implement hankelization before the prediction of time series in different tasks. However, no previous study has investigated the influence of hankelization for the BCI area.

The thesis goal is to find optimal representations of feature and target tensors in latent space by combining a hankelization and dimensionality reduction methods. These tensors representations should be optimal in terms of forecasting quality of the target variables and complexity of methods.

Objectives:

- determine whether hankelization along temporal mode improves quality of forecasting;
- determine whether hankelization along spatial mode improves quality of forecasting;
- determine whether hankelization along both modes improve quality of forecasting;

Literature review. The BCI datasets usually contain simple target variable as classes of some movements. However, there are several datasets in which the target variable is a set of time series. These time series can be correlated and redundant as well as independent variable. One of the most popular datasets for forecasting task is food-tracking datasets by project Neurotycho [3], [20]. These datasets consist of epidural and subdural electrocorticography (ECoG) signals from monkeys. In addition, more and more human datasets for BCI forecasting are becoming available. For example, a human electroencephalography (EEG) dataset [8] has 7-channel electromyography signals from arms. The other dataset is a gesture dataset from the Stanford Digital library [13] that contains ECoG signals from humans and signals from fingers.

In the datasets for BCI, the brain signals have been taken from a large number (> 30) of closely spaced electrodes. It leads to redundant measurements and instability of models. Consequently, a great deal of previous research into BCI has focused on dimensionality reduction techniques [11]. One of the most commonly used dimensionality reduction methods is principal component analyses (PCA) and its modifications (Multilinear PCA [12]). PCA and its modifications capture the greatest variance in the input data. However, it also includes the variance in noise signals.

All the above mentioned datasets have not only correlated and noisy brain signals but also correlated target variables. Thus, dimensionality reduction of target variable also should be done. Usually, it is done by alignment between independent and target variables. Therefore, for BCI data, the widely used dimensionality reduction algorithm is partial least squares (PLS) and its modifications [5] - [26].

The algorithm projects the features and the targets onto the joint latent space and maximizes the covariances between projected vectors. It allows to save information about initial input and

target matrices and find their relations. The dimensionality of latent space is much less than the size of initial data description. It leads to a stable linear model built on the small number of features.

For the data in tensor format, there are several modifications of this algorithm, such as High-Order PLS [25], Multi-linear PLS [26] and Multi-way PLS [5]. Another similar approach to PLS is a canonical-correlation analysis (CCA). It maximizes the correlations between projected vectors of the features and the targets.

In recent years, there has been an increasing amount of literature on deep learning (DL) methods for BCI task. The most recent DL models for BCI task are Deep Tensor Canonical Correlation Analysis (DTCCA) [23] and Hybrid Autoencoder [16]. Shallow autoencoders are essentially equivalent to PCA transformations, especially when they are trained, using weight decay regularization, but autoencoders with nonlinear encoder functions and nonlinear decoder functions can learn a more powerful nonlinear generalization of PCA. Furthermore, if a condition module is added to an autoencoder for alignment with the target data, autoencoder's reduced features are connected with target data. Collectively, these studies outline a critical role for dimensionality reduction and that DL models become more prevalent in BCI task.

Despite the redundancy of data in the BCI problem, tensorization is sometimes used, which can help to catch hidden correlations and create additional informative features. Tensorization by appending an additional dimension as degrees of polynomial fittings helped to improve classification metrics [15]. Time delay embedding [7] is used to improve classification quality for BCI task. This approach was inspired by the common spectral patterns method [10] and allowed to identify bases in which the data differ substantially between classes. Moreover, for the task of artefact removal, generation of high-order tensor data for EEG was used [27]. This tensorization was made by applying wavelet transform. But none of these tensorizations has been applied in the task of predicting target time series.

One of the common tensorization techniques for time series is a hankelization. It is a natural data augmentation technique for time series to incorporate the intrinsic temporal correlation. For example, hankelization was used for anomaly detection in traffic data [22]. It has been shown that with the hankelization operation, the model can simultaneously capture the global and local spatio-temporal correlations and exhibit more robust performance. In addition, hankelization was used for forecasting for different time series data. For instance, forecasting of short term wind speed data was conducted via selective hankelization [9].

To apply hankelization for tensor data, multi-way delay embedding transform (MDT) was introduced [24]. It helped to capture some shift-invariant structure in the task of recovery of missing data. Moreover, MDT has been used to forecast short time series [19] and to forecast power consumption [18]. In both works, its ability to capture hidden correlations increased the quality of forecasting.

In this work, the influence of hankelization on the quality of prediction was studied. We considered hankelization with such dimensionality reduction models as MPCA [12], HOPLS [25] and state-of-the-art deep learning model (autoencoder) [16]. The prediction was made for four samples of data: two from the Neurotycho dataset [3], one from the gesture dataset [13] and one from the human EEG dataset [8]. The results showed...

Chapter 2

Problem statement

In this chapter, the main notations are given and the task of decoding the time series is described. An overview of standard methods of time series analysis is provided. The task of constructing an optimal linear regression model of decoding is set.

2.1 Notation

A *tensor* is a multidimensional array $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$. The *order* of a tensor is the number of dimensions, also known as ways or modes. *Vectors* (tensors of order one) are denoted by $\mathbf{x} \in \mathbb{R}^{I_1}$. *Matrices* (tensors of order two) are denoted by $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$. Higher-order *tensors* (order three or higher) are denoted by $\underline{\mathbf{X}}$.

Fibers are the higher-order analogue of matrix rows and columns. A fiber is defined by fixing every index but one. A matrix column is a mode-1 fiber and a matrix row is a mode-2 fiber. Third-order tensors have column, row, and tube fibers, denoted by $\mathbf{x}_{:jk}$, $\mathbf{x}_{i:k}$, $\mathbf{x}_{ij:}$, respectively.

Slices are two-dimensional sections of a tensor, defined by fixing all but two indices, e.g. horizontal, lateral, and frontal slides of a third-order tensor $\underline{\mathbf{X}}$, denoted by $\mathbf{X}_{i::}$, $\mathbf{X}_{:j:}$, $\mathbf{X}_{::k}$. Alternatively, the k -th frontal slice of a third-order tensor, $\mathbf{X}_{::k}$, may be denoted more compactly as \mathbf{X}_k .

The *norm* of a tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is the square root of the sum of the squares of all its elements, i.e.,

$$\|\underline{\mathbf{X}}\| = \sqrt{\sum_{i_1=1}^{I_1} \dots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N}^2}.$$

The *inner product* of two same-sized tensors $\underline{\mathbf{X}}, \underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is the sum of the products of their entries, i.e.,

$$\langle \underline{\mathbf{X}}, \underline{\mathbf{Y}} \rangle = \sum_{i_1=1}^{I_1} \dots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N} y_{i_1 i_2 \dots i_N}.$$

It follows immediately that $\langle \underline{\mathbf{X}}, \underline{\mathbf{X}} \rangle = \|\underline{\mathbf{X}}\|^2$.

Matricization. *Matricization*, also known as *unfolding* or *flattening*, is the process of reordering the elements of an N -way array into a matrix. The mode- n matricization of a tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is denoted by $\mathbf{X}_{(n)}$ and arranges the mode- n fibers to be the columns of the resulting matrix. Tensor element (i_1, i_2, \dots, i_N) maps to matrix element (i_n, j) , where

$$j = 1 + \sum_{k=1, k \neq n} (i_k - 1) J_k, \quad J_k = \prod_{m=1, m \neq n}^{k-1} I_m.$$

Last, we note that it is also possible to vectorize a tensor. The vectorized version is

$$\text{vec}(\underline{\mathbf{X}}) \in \mathbb{R}^{I_1 \cdot I_2 \cdot \dots \cdot I_N}.$$

Tensor Multiplication. The n -mode (matrix) product of a tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted by $\underline{\mathbf{X}} \times_n \mathbf{U}$ and is of size $I_1 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N$. Elementwise, we have

$$(\underline{\mathbf{X}} \times_n \mathbf{U})_{i_1 \dots i_{n-1} j i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \dots i_N} u_{j i_n}.$$

Each mode- n fiber is multiplied by the matrix \mathbf{U} . The idea can also be expressed in terms of unfolded tensors:

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_n \mathbf{U} \iff \mathbf{Y}_{(n)} = \mathbf{U} \mathbf{X}_{(n)}.$$

The n -mode product of a tensor with a matrix is related to a change of basis in the case when a tensor defines a multilinear operator.

Multilinear product of a tensor $\underline{\mathbf{G}}$ and matrices $\mathbf{B}^{(n)}, n = 1, \dots, N$

$$\underline{\mathbf{X}} = \underline{\mathbf{G}} \times_1 \mathbf{B}^{(1)} \dots \times_N \mathbf{B}^{(N)}$$

can be written as follows

$$\underline{\mathbf{X}} = \llbracket \underline{\mathbf{G}}; \mathbf{B}^{(1)}, \dots, \mathbf{B}^{(N)} \rrbracket.$$

Tucker decomposition. It provides a factorization of a N th-order tensor into a relatively small size core tensor and factor matrices. It can be written as follows

$$\underline{\mathbf{X}} \approx \llbracket \underline{\mathbf{G}}; \mathbf{B}^{(1)}, \dots, \mathbf{B}^{(N)} \rrbracket \quad (2.1)$$

where $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is the given tensor, $\underline{\mathbf{G}} \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_N}$ is the core tensor ($R_n < I_n, n = 1, \dots, N$) and $\mathbf{B}^{(n)} \in \mathbb{R}^{I_n \times R_n}, n = 1, \dots, N$ are the factor matrices. In the standart Tucker model, factor matrices are orthogonal.

2.2 Problem statement

Definition 2.1 A time series is a function of a discrete argument $\mathbf{s}(t)$ that matches time reports $t_i \in \mathcal{T}$ with a vector of the values of the measured variables $\mathbf{s}(t_i) = \mathbf{s}_i \in \mathbb{R}^M$.

Definition 2.2 Let $\{\mathbf{s}_n(t)\}_{n=1}^{N_s}$ be a set of time series and $\{\mathbf{y}_n(t)\}_{n=1}^{N_y}$ a set of target time series. The task of finding the values of $\{\mathbf{y}_n(t)\}_{n=1}^{N_y}$ from the previous values of $\{\mathbf{s}_n(t)\}_{n=1}^{N_s}$ is called the task of decoding time series $\{\mathbf{y}_n(t)\}_{n=1}^{N_y}$.

A sample made from the time series $\mathbf{s}(t), \mathbf{y}(t)$: $(\underline{\mathbf{X}}, \underline{\mathbf{Y}})$, where

$$\underline{\mathbf{X}} \in \mathbb{R}^{M \times I_1 \times \dots \times I_{D_x}}, \quad \underline{\mathbf{Y}} \in \mathbb{R}^{M \times J_1 \times \dots \times J_{D_y}}, \quad (2.2)$$

where $\mathbf{y}_m = \mathbf{y}(t_m)$, a $\underline{\mathbf{X}}_m \in \mathbb{R}^{I_1 \times \dots \times I_{D_x}}$ is a tensor.

The goal is to forecast a dependent variable \mathbf{y}_m from an independent input object $\underline{\mathbf{X}}_m, m = 1, \dots, M$.

In this research, the aim is to find a composition of models such as:

$$\Phi = \Psi_x \circ h_x \circ g \circ h_y^{-1} \circ \Psi_y^{-1} : \mathbb{R}^{I_1 \times \dots \times I_{D_x}} \rightarrow \mathbb{R}^{J_1 \times \dots \times J_{D_y}}, \quad (2.3)$$

where Ψ_x, h_x, g, h_y and Ψ_y work within a process:

$$\begin{array}{ccc}
 \underline{\mathbf{X}} & \xrightarrow{\Phi} & \underline{\mathbf{Y}} \\
 \Psi_x \downarrow & & \Psi_y \downarrow \\
 \hat{\underline{\mathbf{X}}} & \xrightarrow{\hat{\Phi}} & \hat{\underline{\mathbf{Y}}} \\
 h_x \downarrow & & h_y \downarrow \\
 \underline{\mathbf{T}} & \xrightarrow{g} & \underline{\mathbf{U}}
 \end{array}$$

Definition 2.3 *Tensorization model $\Psi : \mathbb{R}^{M \times I_1 \times \dots \times I_D} \rightarrow \mathbb{R}^{\tilde{M} \times \tilde{I}_1 \times \dots \times \tilde{I}_G}$, where $D < G$, is a model that increase order of tensor from D to G .*

In this research, hankelization (Sec. 3.1) is used as the tensorization model. In case without tensorization, Ψ is the identity transformation. There are four types of the tensorization model used:

- Without tensorization;
- Hankelization along time dimension;
- Hankelization along space dimension;
- Hankelization along time and space dimensions.

The type of tensorization is a hyperparameter of the model Φ .

Definition 2.4 $h : \mathbb{R}^{\tilde{I}_1 \times \dots \times \tilde{I}_G} \rightarrow \mathbb{R}^{L_1 \times \dots \times L_G}$, where $L_i < \tilde{I}_i$ is called a dimensionality reduction model, which reduces the dimensionality from $\mathbb{R}^{\tilde{I}_1 \times \dots \times \tilde{I}_G}$ to $\mathbb{R}^{L_1 \times \dots \times L_G}$, namely $\underline{\mathbf{T}}_m = h_x(\underline{\mathbf{X}}_m, \theta_y)$ where θ are parameters of the model.

Definition 2.5 $g : \mathbb{R}^{L_1 \times \dots \times L_{G_x}} \rightarrow \mathbb{R}^{K_1 \times \dots \times K_{G_y}}$ is called an alignment model, which aligns each $\underline{\mathbf{T}}_m \in \mathbb{R}^{L_1 \times \dots \times L_{G_x}}$ to $\underline{\mathbf{U}}_m \in \mathbb{R}^{K_1 \times \dots \times K_{G_y}}$, namely $\underline{\mathbf{U}}_m = g(\underline{\mathbf{T}}_m, \eta)$, where η are parameters of the model.

Model Φ is called optimal, if it minimizes error functional \mathcal{L} :

$$\Phi^* = \arg \min_{\{\theta, \eta\}} \mathcal{L}(\Phi(\underline{\mathbf{X}}, \theta, \eta), \underline{\mathbf{Y}}) \quad (2.4)$$

In this work, we investigate how tensorization type in combination with different dimensionality models influence on the quality of the prediction.

Method for tensorization, hankelization, is described in the section 3.1. Possible models of dimensionality reduction are discussed in the sections 3.2-3.5. Tensor regression that can be used as an alignment model is described in the section 3.7.

Chapter 3

Methodology

There are a number of instruments available for dimensionality reduction such as principal component analysis (PCA), partial least squares (PLS), and their tensor versions (High-Order PLS, MPCA). These methods were used directly for the initial data or for data tensorized by its structure. In this research, hankelization was used as a tensorization method before dimensionality reduction techniques. Hankelization is one of the most common tensorization methods for time series. The benefit of this approach is that it reveals hidden correlations in initial data and allows finding low-rank approximations with high levels of compression. In this chapter, hankelization method, dimensionality reduction and alignment models are described.

3.1 Hankelization

To begin, we will provide a standard definition of a delay embedding transform for a vector that can be understood as a time-series signal. Let consider $\mathbf{x} = (x_1, \dots, x_I)^\top \in \mathbb{R}^I$, a standard delay embedding transform of \mathbf{x} with τ is given by

$$\mathcal{H}_\tau(\mathbf{x}) = \begin{pmatrix} x_1 & x_2 & \dots & x_{I-\tau+1} \\ x_2 & x_3 & \dots & x_{I-\tau+2} \\ \vdots & \vdots & \ddots & \vdots \\ x_\tau & x_{\tau+1} & \dots & x_I \end{pmatrix} \in \mathbb{R}^{\tau \times (I-\tau+1)},$$

where $\mathcal{H}_\tau(\mathbf{x})$ is a Hankel matrix of a vector \mathbf{x} .

Let consider a 1D convolution for vector $x \in \mathbb{R}^I$ and kernel $h \in \mathbb{R}^k$, then 1D convolution is represented by

$$x * h = \begin{pmatrix} x_1 & x_2 & \dots & x_k \\ x_2 & x_3 & \dots & x_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{I-k+1} & x_{I-k+2} & \dots & x_I \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \end{pmatrix} = \mathcal{H}_{I-k+1}(x)h$$

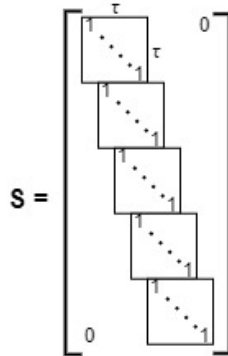


Figure 3.1: The structure of a duplication matrix \mathbf{S} .

Time-series have variables that are temporally nearby are highly correlated. In the BCI task, signals from different channels are also correlated. So, the data in BCI are spatially and temporally correlated. Local correlations are the reasons for the well-known advantages of extracting and combining local features before making a prediction. Convolution extracts local features by restricting the receptive elds of hidden units to be local [1]. The above mentioned connection of hankelization with convolution makes hankelization the helpful tool for revealing the hidden correlations.

Hankelization is an effective way to transform lower-order data to higher-order tensors. MDT is a multi-way extension of hankelization [24]. It combines multi-linear duplication and multi-way folding operations. By denoting $\hat{\mathbf{X}} \in \mathbb{R}^{I_1 \times \dots \times I_G}$ as the block Hankel tensor of $\mathbf{X} \in \mathbb{R}^{I_1 \times \dots \times I_D}$, $D < G$, the MDT for \mathbf{X} is defined by

$$\hat{\mathbf{X}} = \mathcal{H}_\tau(\mathbf{X}) = \text{Fold}_{(\mathbf{I}, \tau)}(\mathbf{X} \times_1 \mathbf{S}_1 \dots \times_D \mathbf{S}_D),$$

where $\mathbf{S}_d \in \mathbb{R}^{\tau_d(I_d - \tau_d + 1) \times I_d}$, $d = 1, \dots, D$ is a duplication matrix (see Fig. 3.1) and

$$\text{Fold}_{(\mathbf{I}, \tau)} : \mathbb{R}^{\tau_1(I_1 - \tau_1 + 1) \times \dots \times \tau_D(I_D - \tau_D + 1)} \rightarrow \mathbb{R}^{\tau_1 \times (I_1 - \tau_1 + 1) \times \dots \times \tau_D \times (I_D - \tau_D + 1)}$$

constructs a higher order block Hankel tensor $\hat{\mathbf{X}}$ from the input tensor \mathbf{X} . The inverse MDT for $\hat{\mathbf{X}}$ is given by

$$\mathbf{X} = \mathcal{H}_\tau^{-1}(\hat{\mathbf{X}}) = \text{Unfold}_{(\mathbf{I}, \tau)}(\hat{\mathbf{X}}) \times_1 \mathbf{S}_1^\dagger \dots \times_N \mathbf{S}_N^\dagger,$$

where \dagger is the Moore-Penrose pseudo-inverse.

3.2 MPCA

Multilinear Principal Component Analysis (MPCA) performs feature extraction by determining a multilinear projection that captures most of the original tensorial input variation. MPCA objective is to define a multilinear transformation $\{\tilde{\mathbf{U}}^{(d)} \in \mathbb{R}^{I_d \times L_d}, d = 1, \dots, D\}$ that maps the original tensor space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \dots \otimes \mathbb{R}^{I_D}$ into a tensor subspace $\mathbb{R}^{L_1} \otimes \mathbb{R}^{L_2} \otimes \dots \otimes \mathbb{R}^{L_D}$ (with $L_d < I_d$, $d = 1, \dots, D$):

$$\hat{\mathbf{X}}_m \approx \mathbf{X}_m \times_1 \tilde{\mathbf{U}}^{(1)\top} \times_2 \tilde{\mathbf{U}}^{(2)\top} \dots \times_D \tilde{\mathbf{U}}^{(D)\top}, \quad m = 1, \dots, M,$$

such that $\{\hat{\mathbf{X}}_m \in \mathbb{R}^{L_1} \otimes \mathbb{R}^{L_2} \otimes \dots \otimes \mathbb{R}^{L_D}, m = 1, \dots, M\}$ captures most of the variations observed in the original tensor objects, assuming that these variations are measured by the total tensor scatter $\Upsilon_{\mathbf{X}}$ which can be calculated as follows:

$$\Upsilon_{\mathbf{X}} = \sum_{m=1}^M \|\mathbf{X}_m - \bar{\mathbf{X}}\|_F^2$$

where $\bar{\mathbf{X}} = \frac{1}{M} \sum_{m=1}^M \mathbf{X}_m$

The MPCA objective is the determination of the D projection matrices $\{\tilde{\mathbf{U}}^{(d)} \in \mathbb{R}^{I_d \times L_d}, d = 1, \dots, D\}$ that maximize the total tensor scatter $\Upsilon_{\mathbf{X}}$

$$\{\tilde{\mathbf{U}}^{(d)} \in \mathbb{R}^{I_d \times L_d}, d = 1, \dots, D\} = \arg \max_{\tilde{\mathbf{U}}^{(1)}, \dots, \tilde{\mathbf{U}}^{(D)}} \Upsilon_{\mathbf{X}}.$$

Here, the dimensionality L_d for each mode is assumed to be known or predetermined.

3.3 Standard Partial Least Squares

The partial least squares (PLS) algorithm projects the design matrix \mathbf{X} and the target matrix \mathbf{Y} to the latent space with low dimensionality ($l < n$). The PLS algorithm finds the latent space matrices $\mathbf{T}, \mathbf{U} \in \mathbb{R}^{m \times l}$ that best describe the original matrices \mathbf{X} and \mathbf{Y} .

The design matrix \mathbf{X} and the target matrix \mathbf{Y} are projected into the latent space in the following way:

$$\underset{m \times n}{\mathbf{X}} = \underset{m \times l}{\mathbf{T}} \cdot \underset{l \times n}{\mathbf{P}^T} + \underset{m \times n}{\mathbf{F}} = \sum_{k=1}^l \underset{m \times 1}{\mathbf{t}_k} \cdot \underset{1 \times n}{\mathbf{p}_k^T} + \underset{m \times n}{\mathbf{F}}, \quad (3.1)$$

$$\underset{m \times r}{\mathbf{Y}} = \underset{m \times l}{\mathbf{U}} \cdot \underset{l \times r}{\mathbf{Q}^T} + \underset{m \times r}{\mathbf{E}} = \sum_{k=1}^l \underset{m \times 1}{\mathbf{u}_k} \cdot \underset{1 \times r}{\mathbf{q}_k^T} + \underset{m \times r}{\mathbf{E}}, \quad (3.2)$$

where \mathbf{T} and \mathbf{U} are the scores matrices in the latent space, \mathbf{P} and \mathbf{Q} are the loading matrices, \mathbf{E} and \mathbf{F} are residual matrices. The PLS maximizes the linear relation between the columns of matrices \mathbf{T} and \mathbf{U} as

$$\mathbf{U} \approx \mathbf{T}\mathbf{B}, \quad \mathbf{B} = \text{diag}(\beta_k), \quad \beta_k = \mathbf{u}_k^T \mathbf{t}_k / (\mathbf{t}_k^T \mathbf{t}_k).$$

We use the PLS algorithm as the dimensionality reduction algorithm in this research.

To obtain the model prediction and find the model parameters, we multiply both sides of (3.1) by \mathbf{W} . Since the residual matrix \mathbf{E} rows are orthogonal to the columns of \mathbf{W} , we have

$$\mathbf{X}\mathbf{W} = \mathbf{T}\mathbf{P}^T\mathbf{W}.$$

The linear transformation between objects in the input and latent spaces is the following

$$\mathbf{T} = \mathbf{X}\mathbf{W}^*, \quad \text{where } \mathbf{W}^* = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}. \quad (3.3)$$

The matrix of the model parameters Θ could be found from the equations (3.2) and (3.3) as

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \mathbf{E} \approx \mathbf{T}\mathbf{B}\mathbf{Q}^T + \mathbf{E} = \mathbf{X}\mathbf{W}^*\mathbf{B}\mathbf{Q}^T + \mathbf{E} = \mathbf{X}\Theta + \mathbf{E}. \quad (3.4)$$

Thus, the model parameters (3.7) are equal to

$$\Theta = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{B}\mathbf{Q}^T.$$

The final model (3.4) is a linear model that is low-dimensional in the latent space. It reduces the data redundancy and increases the model stability.

3.4 High-Order PLS

HOPLS performs simultaneously constrained Tucker decompositions for an $(D+1)$ th-order independent tensor, $\underline{\mathbf{X}} \in \mathbb{R}^{M \times I_1 \times \dots \times I_D}$, and an $(D+1)$ th-order dependent tensor, $\underline{\mathbf{Y}} \in \mathbb{R}^{M \times J_1 \times \dots \times J_D}$, which have the same size in the first mode. Such a model allows us to find the optimal subspace approximation of $\underline{\mathbf{X}}$, in which the independent and dependent variables share a common set of latent vectors in one specific mode (i.e., samples mode).

$$\underline{\mathbf{X}} = \sum_{r=1}^R \underline{\mathbf{G}}_{xr} \times_1 \mathbf{t}_r \times_2 \mathbf{P}_r^{(1)} \dots \times_{D+1} \mathbf{P}_r^{(D)} + \underline{\mathbf{E}}_R,$$

$$\underline{\mathbf{Y}} = \sum_{r=1}^R \underline{\mathbf{G}}_{yr} \times_1 \mathbf{t}_r \times_2 \mathbf{Q}_r^{(1)} \dots \times_{D+1} \mathbf{Q}_r^{(D)} + \underline{\mathbf{F}}_R,$$

where R is the number of latent vectors, $\mathbf{t}_r \in \mathbb{R}^M$ is the r -th latent vector, $\{\mathbf{P}_r^{(d)}\}_{d=1}^D \in \mathbb{R}^{I_d \times L_d}$ and $\{\mathbf{Q}_r^{(d)}\}_{d=1}^D \in \mathbb{R}^{J_d \times K_d}$ are the loading matrices in mode- d , and $\underline{\mathbf{G}}_{xr} \in \mathbb{R}^{1 \times L_1 \times \dots \times L_D}$ and $\underline{\mathbf{G}}_{yr} \in \mathbb{R}^{1 \times K_1 \times \dots \times K_D}$ are core tensors.

By defining a latent matrix $\mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_R]$, mode- d loading matrix $\bar{\mathbf{P}}^{(d)} = [\mathbf{P}_1^{(d)}, \dots, \mathbf{P}_R^{(d)}]$, mode- d loading matrix $\bar{\mathbf{Q}}^{(d)} = [\mathbf{Q}_1^{(d)}, \dots, \mathbf{Q}_R^{(d)}]$ and core tensors

$$\begin{aligned} \underline{\mathbf{G}}_x &= \text{blockdiag}(\underline{\mathbf{G}}_{x1}, \dots, \underline{\mathbf{G}}_{xR}) \in \mathbb{R}^{R \times RL_1 \times \dots \times RL_D}, \\ \underline{\mathbf{G}}_y &= \text{blockdiag}(\underline{\mathbf{G}}_{y1}, \dots, \underline{\mathbf{G}}_{yR}) \in \mathbb{R}^{R \times RK_1 \times \dots \times RK_D}, \end{aligned}$$

the HOPLS model can be rewritten as

$$\begin{aligned} \underline{\mathbf{X}} &= \underline{\mathbf{G}}_x \times_1 \mathbf{T} \times_2 \bar{\mathbf{P}}^{(1)} \dots \times_{D+1} \bar{\mathbf{P}}^{(D)} + \underline{\mathbf{E}}_R, \\ \underline{\mathbf{Y}} &= \underline{\mathbf{G}}_y \times_1 \mathbf{T} \times_2 \bar{\mathbf{Q}}^{(1)} \dots \times_{D+1} \bar{\mathbf{Q}}^{(D)} + \underline{\mathbf{F}}_R, \end{aligned}$$

where $\underline{\mathbf{E}}_R$ and $\underline{\mathbf{F}}_R$ are the residuals obtained after extracting R latent components. The core tensors, $\underline{\mathbf{G}}_x$ and $\underline{\mathbf{G}}_y$, have a special block-diagonal structure (Fig. 3.2) and their elements indicate the level of local interactions between the corresponding latent vectors and loading matrices.

The cross-covariance tensor is defined by

$$\underline{\mathbf{C}} = \text{COV}_{\{1,1\}}(\underline{\mathbf{X}}, \underline{\mathbf{Y}}) \in \mathbb{R}^{I_1 \times \dots \times I_D \times J_1 \times \dots \times J_D}$$

The optimization problem can be formulated as

$$\begin{aligned} \left\| \llbracket \underline{\mathbf{C}}; \mathbf{P}^{(1)\top}, \dots, \mathbf{P}^{(D)\top}, \mathbf{Q}^{(1)\top}, \dots, \mathbf{Q}^{(D)\top} \rrbracket \right\|_F^2 \rightarrow & \max_{\substack{\{\mathbf{P}^{(d)}, \mathbf{Q}^{(d)}\}, \\ \text{s.t. } \mathbf{P}^{(d)\top} \mathbf{P}^{(d)} = \mathbf{I}_{L_d}, \\ \mathbf{Q}^{(d)\top} \mathbf{Q}^{(d)} = \mathbf{I}_{K_d}}}, \end{aligned}$$

where $\llbracket \dots \rrbracket$ denotes the multilinear products between a tensor and a set of matrices, $\mathbf{P}^{(d)}$ and $\mathbf{Q}^{(d)}$, $d = 1, \dots, D$, comprise the unknown parameters.

3.5 Autoencoder ReducedNet

Autoencoders are often used for the problem of dimensionality reduction. For example, for the task of reducing the dimension for electrocorticogram data, the autoencoder ReducedNet [17] was used. Predicting the coordinates of hand movements is found to be more accurate after dimensionality reduction using ReducedNet than after methods such as PLS or KernelPCA. Therefore, it was decided to apply this model in this work as state-of-art model, but with minor modifications.

The model proposed in [17] consists of two blocks: a dimensionality reduction module consisting of an encoder h and a decoder ψ ; and a condition module ω . The dimensionality reduction module is composed of convolutional blocks (Conv1D) and LSTM blocks. Each convolutional block contains of a one-dimensional convolutional layer, one Batch Normalization layer, GELU (gaussian error linear unit) activation function layer and Dropout layer. An LSTM block is a single LSTM layer. The condition module was a single linear layer.

The ReducedNet model has only one tunable parameter, N_{CH} , that was the number of features after encoder.

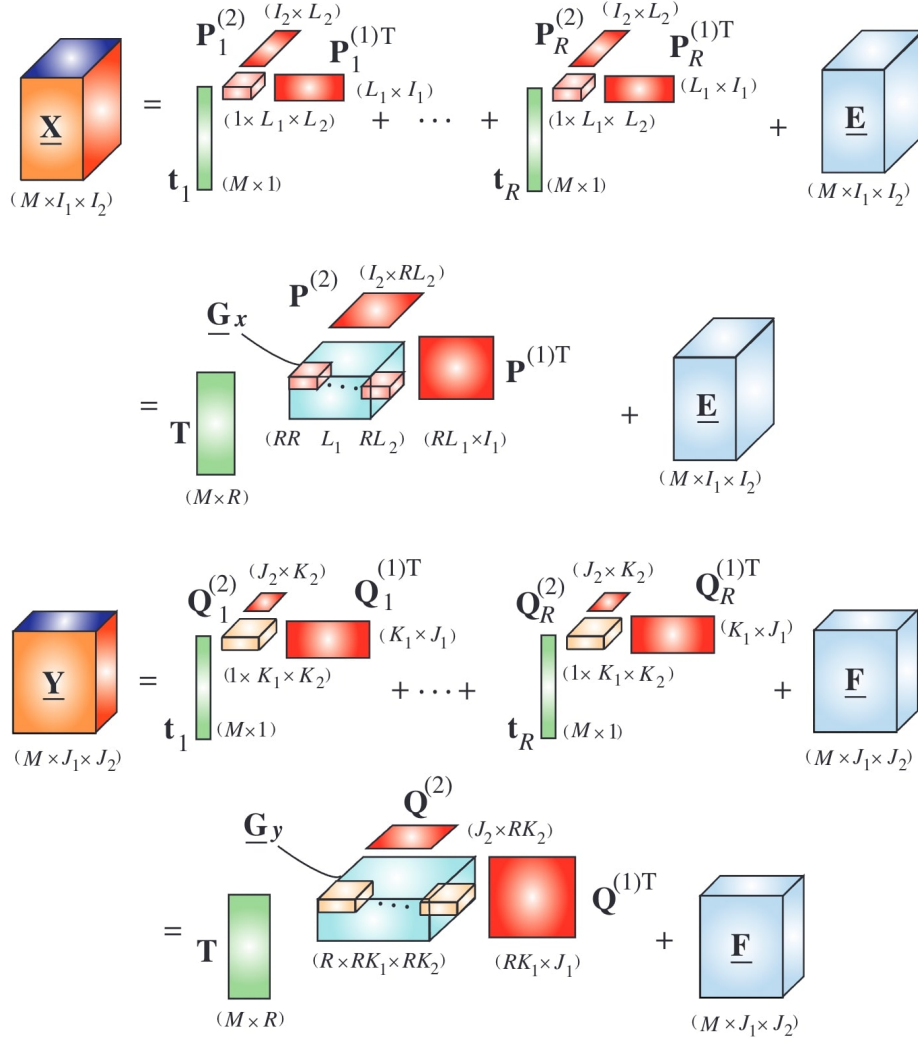


Figure 3.2: The HOPLS model which approximates the independent variables, $\underline{\mathbf{X}}$, as a sum of rank- $(1, L_1, L_2)$ tensors. The approximation for the dependent variables, $\underline{\mathbf{Y}}$, follows a similar principle, whereby the common latent components, $\underline{\mathbf{T}}$, are shared between $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$. This picture is taken from [4].

The loss, proposed in article [17], takes into account outputs from both modules:

$$\mathcal{L}_1 = \mathcal{L}_{rec} + \alpha \cdot \mathcal{L}_{dec}, \quad (3.5)$$

where:

$$\mathcal{L}_{rec} = \frac{1}{M} \sum_{m=1}^M \|\underline{\mathbf{X}}_m - h \circ \psi(\underline{\mathbf{X}}_m)\|^2$$

$$\mathcal{L}_{dec} = \frac{1}{M} \sum_{m=1}^M \|\underline{\mathbf{Y}}_m - \omega \circ h(\underline{\mathbf{X}}_m)\|^2$$

Modification of ReducedNet. The main problem of ReducedNet is that encoder's output is a matrix. It means that the input tensor loses its initial structure. Besides, it can work only with 3D data. Thus, in this work, TensorReducedNet is proposed. The 3D example of TensorReducedNet can be seen in Fig. 3.3. The output from the encoder of 3D TensorReducedNet is in $\mathbb{R}^{T \times N_F \times N_{CH}}$.

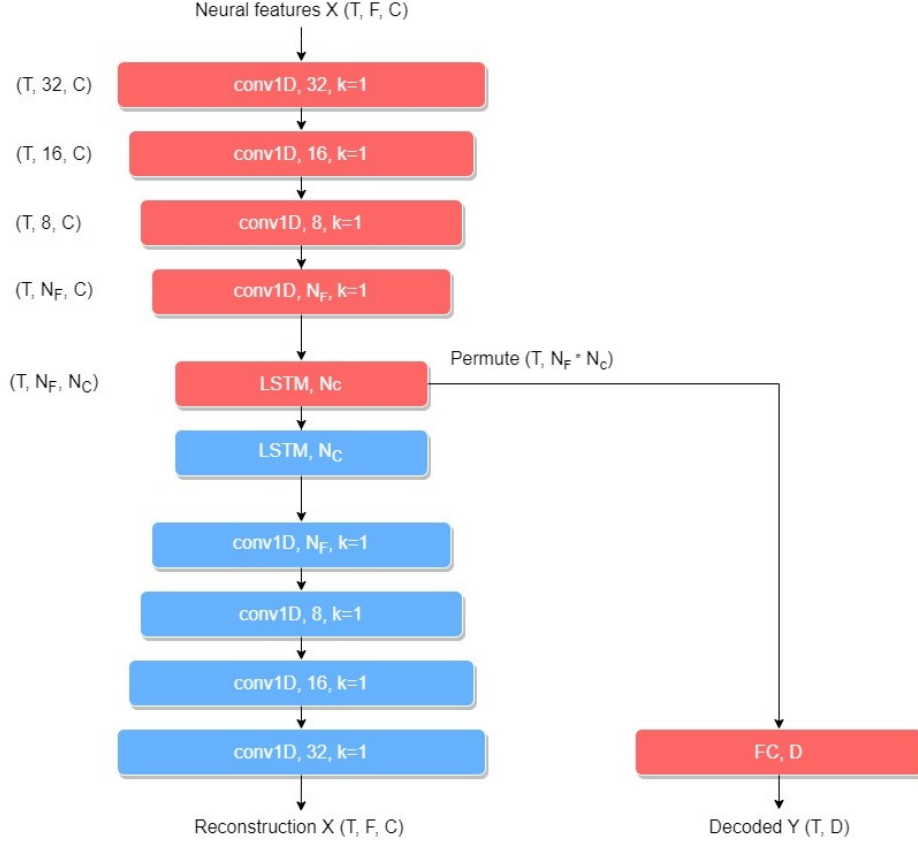


Figure 3.3: TensorReducedNet.

For tensors with order higher than 3, the reduction of dimensionality of all modes except one is achieved by convolutional blocks, the reduction of dimensionality of the last mode (temporal) is made by LSTM. For autoencoder with output in $\mathbb{R}^{T \times L_1 \times L_2 \times \dots \times I_G}$:

$$\underline{\mathbf{X}} \in \mathbb{R}^{M \times I_1 \times \dots \times I_G} \xrightarrow{\text{conv1D blocks}} \tilde{\underline{\mathbf{X}}} \in \mathbb{R}^{M \times I_1 \times L_2 \times \dots \times L_G} \xrightarrow{\text{LSTM block}} \hat{\underline{\mathbf{X}}} \in \mathbb{R}^{M \times L_1 \times L_2 \times \dots \times L_G}.$$

The FC layer for TensorReducedNet is applied to the unfolded tensor $\hat{\underline{\mathbf{X}}}$ along first mode.

3.6 Multimodal regression

The feature tensor $\underline{\mathbf{X}}$ has $D + 1$ dimension. To restore the target time series, the feature tensor $\underline{\mathbf{X}}$ can be unfolded by the first dimension:

$$\underline{\mathbf{X}}_{(1)} = \left[\text{vec}(\underline{\mathbf{X}}_1)^\top, \dots, \text{vec}(\underline{\mathbf{X}}_M)^\top \right]^\top \in \mathbb{R}^{M \times (I_1 \dots I_D)} \quad (3.6)$$

Thus, the problem becomes a multimodal regression problem, where $\mathbf{X} \in \mathbb{R}^{M \times (I_1 \dots I_D)}$ is the input matrix obtained by matricization of the input tensor, $\mathbf{Y} \in \mathbb{R}^{M \times J}$ is the target matrix.

The goal is to forecast a dependent variable $\mathbf{y}_m \in \mathbb{R}^J$ with J targets from an independent input object $\mathbf{x}_m \in \mathbb{R}^I$ with $I = I_1 \dots I_D$ features. We assume that there is a linear dependence between the object \mathbf{x}_m and the target variable \mathbf{y}_m as

$$\mathbf{y}_m = \Theta \mathbf{x}_m + \varepsilon, \quad (3.7)$$

where $\Theta \in \mathbb{R}^{J \times I}$ is the matrix of the model parameters, and $\varepsilon \in \mathbb{R}^J$ is a residual vector. One has to find the matrix of the model parameters Θ by a given dataset (\mathbf{X}, \mathbf{Y}) , where $\mathbf{X} \in \mathbb{R}^{M \times I}$ is a design matrix and $\mathbf{Y} \in \mathbb{R}^{M \times J}$ is a target matrix:

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^\top = [\chi_1, \dots, \chi_n]; \quad \mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_m]^\top = [\nu_1, \dots, \nu_r].$$

The columns χ_j of \mathbf{X} correspond to the object features, and the columns ν_j of \mathbf{Y} correspond to the targets.

The optimal parameters are determined by the minimization of an error function. We define the quadratic loss function as follows:

$$\mathcal{L}(\Theta|\mathbf{X}, \mathbf{Y}) = \left\| \begin{matrix} \mathbf{Y} \\ M \times J \end{matrix} - \begin{matrix} \mathbf{X} \\ M \times I \end{matrix} \cdot \begin{matrix} \Theta \\ J \times I \end{matrix}^\top \right\|_2^2 \rightarrow \min_{\Theta}. \quad (3.8)$$

The solution of (3.8) is given by

$$\Theta = \mathbf{Y}^\top \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1}.$$

The linear dependent columns of \mathbf{X} lead to an instable solution for the optimization problem (3.8). If there is a vector $\alpha \neq \mathbf{0}_I$ such that $\mathbf{X}\alpha = \mathbf{0}_m$, then adding α to any column of Θ does not change the value of the loss function $\mathcal{L}(\Theta|\mathbf{X}, \mathbf{Y})$. In this case, the matrix $\mathbf{X}^\top \mathbf{X}$ is close to singular and is not invertible. One of the possible solutions is not to make a matricization of the input tensor but to use tensor regression. The main advantage of the tensor regression over the multimodal regression is that it stores information about the data structure.

3.7 Tensor regression

The tensor regression can be defined in a similar way as (3.7):

$$\mathbf{y}_m = \langle \underline{\mathbf{X}}_m | \underline{\mathbf{W}} \rangle + \varepsilon, \quad (3.9)$$

where $\langle \underline{\mathbf{X}}_m | \underline{\mathbf{W}} \rangle$ is a tensor contraction along the first D dimensions of the D -dimensional initial tensor $\underline{\mathbf{X}}_m \in \mathbb{R}^{I_1 \times \dots \times I_D}$ and $(D+1)$ -dimensional tensor of model parameters $\underline{\mathbf{W}} \in \mathbb{R}^{I_1 \times \dots \times I_D \times J}$, $\varepsilon \in \mathbb{R}^J$ is an error, $\mathbf{y}_m \in \mathbb{R}^J$ is the target vector.

The j -th element of the vector obtained after tensor contraction of $\underline{\mathbf{X}}_m$ and $\underline{\mathbf{W}}$ is calculated as follows:

$$\langle \underline{\mathbf{X}}_m | \underline{\mathbf{W}} \rangle_j = \sum_{i_1=1}^{I_1} \dots \sum_{i_D=1}^{I_D} x_{i_1, \dots, i_D} w_{i_1, \dots, i_D, j} \quad (3.10)$$

The optimal parameter tensor $\underline{\mathbf{W}}^*$ is found by minimizing the quadratic error function:

$$\underline{\mathbf{W}}^* = \arg \min_{\underline{\mathbf{W}}} \sum_{m=1}^M \|\mathbf{y}_m - \langle \underline{\mathbf{X}}_m | \underline{\mathbf{W}} \rangle\|_2^2 \quad (3.11)$$

In practice, the parameter tensor $\underline{\mathbf{W}}$ is represented in the Tucker decomposition:

$$\underline{\mathbf{W}} \approx \underline{\mathbf{G}} \times_1 \mathbf{U}^{(1)} \dots \times_D \mathbf{U}^{(D)} \times_{D+1} \mathbf{U}^{(D+1)}, \quad (3.12)$$

where $\underline{\mathbf{G}}$ is a smaller core tensor, $\mathbf{U}^{(i)}$ are unitary matrices.

For higher-order $\underline{\mathbf{Y}} \in \mathbb{R}^{M \times J_1 \times \dots \times J_D}$, unfolded tensor $\underline{\mathbf{Y}}$ along the first mode is used as $\mathbf{Y} \in \mathbb{R}^{M \times J}$ where $J = J_1 \cdot \dots \cdot J_D$.

Chapter 4

Numerical experiments

In this chapter, datasets used in this work are described, metrics for performance evaluation are defined, and results are performed and discussed.

4.1 Datasets

In this section the datasets used in this work will be described.

4.1.1 Neurotycho dataset

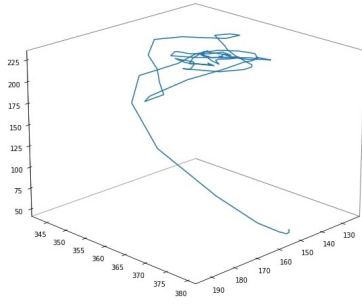


Figure 4.1: Hand movement trajectory

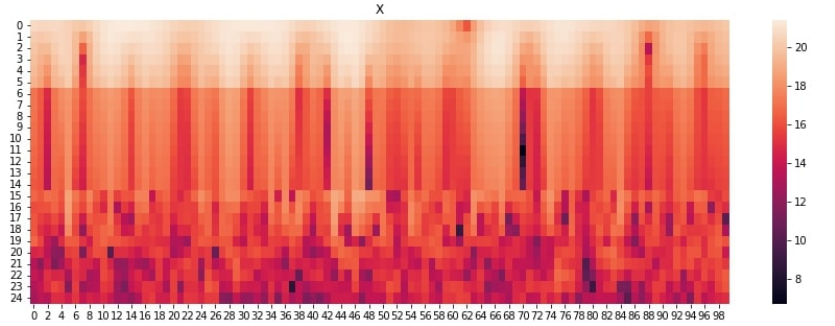


Figure 4.2: One channel data in time-frequency domain

One of the datasets for the computational experiment is electrocorticogram (ECoG) data from the Neurotycho [21] dataset. The data consist of 32-channel voltage signals taken from the monkey's brain. ECoG data are multidimensional and measurements correlate in both temporal and spatial domains. The target variable is the coordinates of the position of the hand in space (Fig. 4.1).

The original voltage signals were converted into a time-frequency representation using a wavelet transform with a parent Morlet wavelet, since this type of transformation is often used in problems with ECoG data [6], [2]. Frequencies was chosen as in [14]. The description of the source signal at each time had the dimension 32 (channels) $\times 27$ (frequencies) $= 864$. Each signal represented a local time interval with a duration of $\Delta t = 1s$. The time step between the signals $\delta t = 0.05s$. The data had dimensions $\underline{\mathbf{X}} \in \mathbb{R}^{T \times 32 \times 27}$ (Fig. 4.2) and $\mathbf{Y} \in \mathbb{R}^{T \times 3}$. The data were divided into training and test samples in the ratio of $\frac{1}{3}$.

We applied hankelization to the initial data with $\tau_0 = 10$ by temporal dimension and with $\tau_1 = 2$ by spatial dimension. So, the hankelized data along temporal dimension is $\underline{\mathbf{X}} \in \mathbb{R}^{(T-9) \times 10 \times 32 \times 27}$ and $\mathbf{Y} \in \mathbb{R}^{(T-9) \times 3}$, along space dimensions $\underline{\mathbf{X}} \in \mathbb{R}^{T \times 10 \times 31 \times 2 \times 27}$ and $\mathbf{Y} \in \mathbb{R}^{T \times 3}$ and along both dimensions is $\underline{\mathbf{X}} \in \mathbb{R}^{(T-9) \times 10 \times 31 \times 2 \times 27}$ and $\mathbf{Y} \in \mathbb{R}^{(T-9) \times 10 \times 3}$.

Also, we used data from Neurotycho [21] dataset with 64 channels, to investigate how hankelization influence quality of prediction for more complex data.

4.1.2 Multimodal EEG dataset

The dataset includes 60-channel electroencephalography (EEG), see Fig. 4.5-4.6, and 7-channel electromyography (EMG), see Fig. 4.3-4.4. Subjects performed different upper-extremity movement tasks. The EEG data were filtered by authors between 8 and 30 Hz (μ and β bands, respectively), known to be within the motor-related frequency range. Besides, artifact rejection was performed. The original EEG signals are converted into a space time representation using a wavelet transform with a Morlet wavelet. The frequencies were chosen between 8 and 30 Hz with step equals 1 Hz. So, $\mathbf{X} \in \mathbb{R}^{T \times 64 \times 24}$, where 24 is a number of frequencies. $\mathbf{Y} \in \mathbb{R}^{T \times 6}$ is the EMG data.

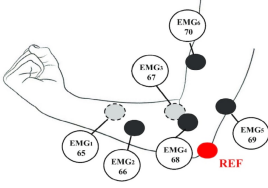


Figure 4.3: EMG electrodes positions

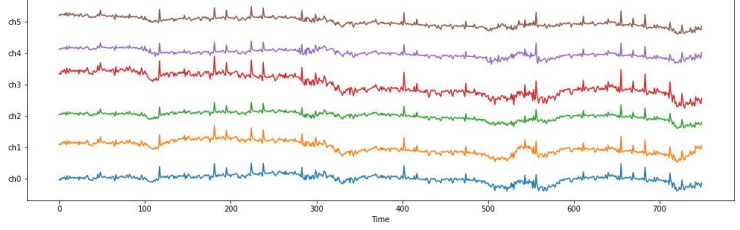


Figure 4.4: EMG data

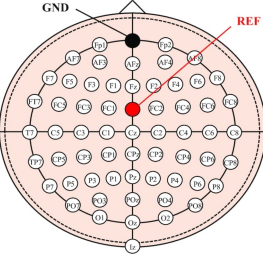


Figure 4.5: EEG electrodes positions

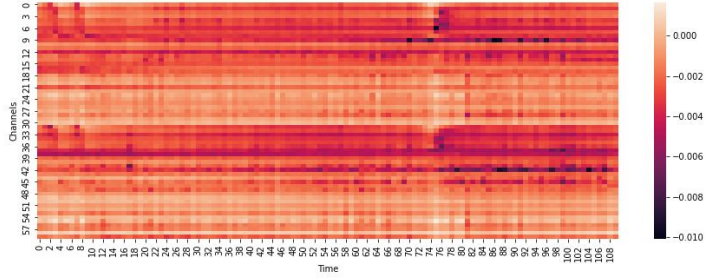


Figure 4.6: EEG data

4.2 Metrics

The scaled Root Mean Squared Error (sRMSE) shows the quality of the model prediction. We estimate sRMSE on train and test data.

$$\text{sRMSE}(\mathbf{Y}, \hat{\mathbf{Y}}_a) = \sqrt{\frac{\text{MSE}(\mathbf{Y}, \hat{\mathbf{Y}}_a)}{\text{MSE}(\mathbf{Y}, \bar{\mathbf{Y}})}} = \frac{\|\mathbf{Y} - \hat{\mathbf{Y}}_a\|_2}{\|\mathbf{Y} - \bar{\mathbf{Y}}\|_2}. \quad (4.1)$$

Here $\hat{\mathbf{Y}}_a = \mathbf{X}_a \Theta_a^T$ is a model prediction and $\bar{\mathbf{Y}}$ is a constant prediction obtained by averaging the targets across all objects. The error on the test set should be as minimal as possible.

To evaluate the complexity of models, we looked at shape of the latent representations of the initial data. Moreover, we estimated number of the parameters of deep learning model, such as TensorReducedNet.

4.3 Results

4.3.1 MPCA

We used MPCA as the most simple tensor dimensionality reduction model. We used MPCA that saves 97% of information. Shapes of latent variables for \mathbf{X} and \mathbf{Y} can be seen in the Table 4.1. Then, we trained Tucker tensor regression with ranks for each tensor mode as hyperparameters. Optimal hyperparameters were found by grid search. For this model, the optimal type of tensorization mostly is hankelization along space dimension. It can be seen on Figures 4.8-4.9. However, no tensorization gives fewer number of values in latent representation of \mathbf{X} . It can be noticed in Fig. 4.7-4.9.

Table 4.1: Results with MPCA

Dataset	Hankelization	Shape of the latent variable of \mathbf{X}	Shape of the latent variable of \mathbf{Y}	sRMSE
Neurotycho A	No	1×8	3	1.764
	Along spatial dimension	$10 \times 1 \times 8$	8×3	1.982
	Along temporal dimension	$1 \times 7 \times 2$	3	1.767
	Along both dimensions	$10 \times 1 \times 7 \times 2$	8×3	1.944
Neurotycho K	No	1×27	3	3.770
	Along spatial dimension	$10 \times 1 \times 26$	4×3	3.722
	Along temporal dimension	$1 \times 22 \times 2$	3	3.578
	Along both dimensions	$10 \times 1 \times 21 \times 2$	4×3	3.625
EEG	No	1×10	3	1.773
	Along spatial dimension	$9 \times 1 \times 11$	9×4	2.487
	Along temporal dimension	$1 \times 14 \times 2$	3	1.563
	Along both dimensions	$9 \times 1 \times 14 \times 2$	9×4	2.335

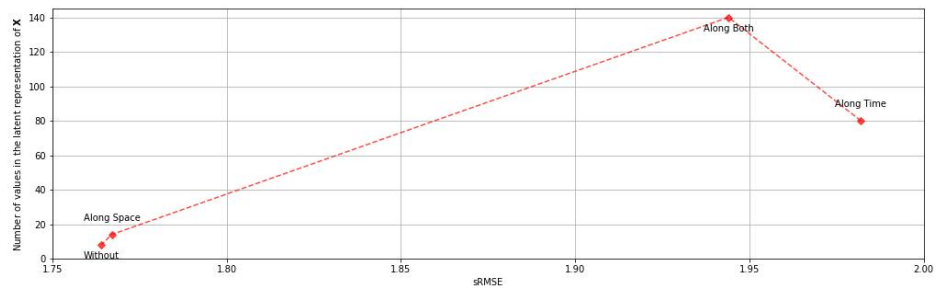


Figure 4.7: Comparison of models for Neurotycho dataset (ch=32) and MPCA model

4.3.2 HOPLS

HOPLS is more complex model than MPCA as it makes an alignment between features and targets. Fitting of the model tunes the number of latent tensors by maximizing the metric. We chose a metric connected with sRMSE 4.1 which equals $1 - sRMSE$. The hyperparameters of the HOPLS model

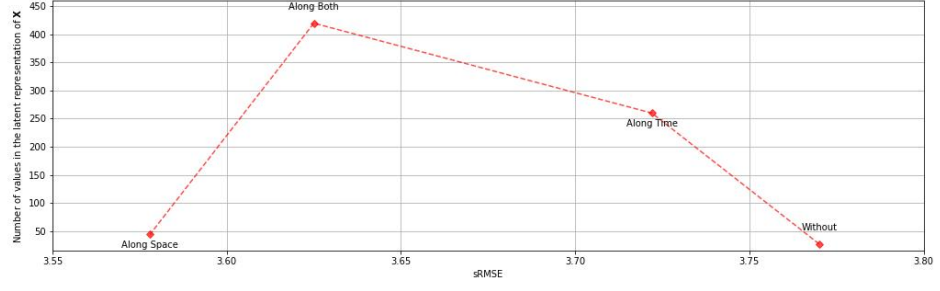


Figure 4.8: Comparison of models for Neurotycho dataset (ch=64) and MPCA model

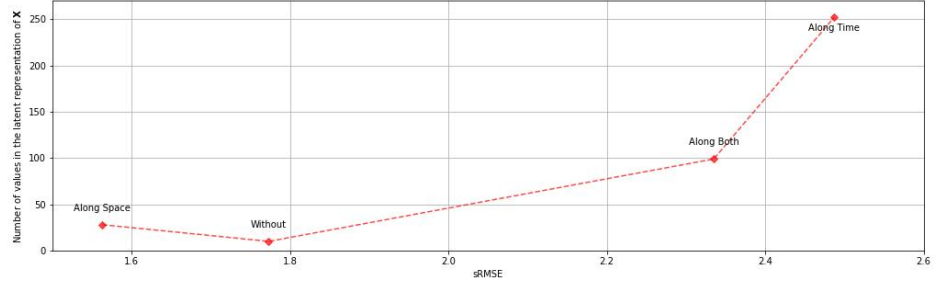


Figure 4.9: Comparison of models for Human EEG dataset (ch=60) and MPCA model

are the shapes of latent representations of \mathbf{X} and \mathbf{Y} . The optimal hyperparameters were found by grid search. The optimal shapes of core tensor of HOPLS are in Table 4.2. For two datasets, the optimal type of tensorization is hankelization along both dimensions. For the Human EEG dataset, the smallest metric was observed without hankelization. However, the fewest number of the values of latent representations was obtained by hankelization along space dimension. It can be seen on Fig. 4.10-4.12.

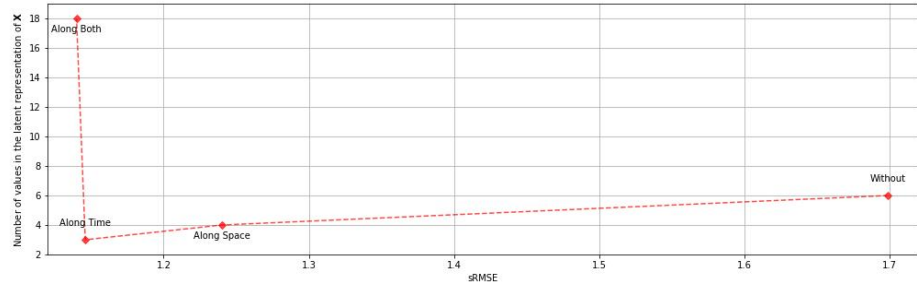


Figure 4.10: Comparison of models for Neurotycho dataset (ch=32) and HOPLS model

4.3.3 TensorReducedNet

We decreased dimensionality only of \mathbf{X} by TensorReducedNet. So, to evaluate complexity of this deep learning model we additionally estimated the number of parameters of the neural network. The hyperparameter of this model is the shape of the latent representation of \mathbf{X} . It was tuned by grid search.

Table 4.2: Results with HOPLS

Dataset	Hankelization	Shape of the latent variable of $\underline{\mathbf{X}}$	Shape of the latent variable of $\underline{\mathbf{Y}}$	sRMSE
Neurotycho A	No	2×3	1	1.699
	Along spatial dimension	$3 \times 1 \times 1$	1×1	1.146
	Along temporal dimension	$2 \times 2 \times 1$	1	1.240
	Along both dimensions	$2 \times 3 \times 3 \times 1$	3×1	1.140
Neurotycho K	No	1×1	1	1.880
	Along spatial dimension	$2 \times 3 \times 1$	1×1	1.488
	Along temporal dimension	$2 \times 1 \times 2$	1	2.136
	Along both dimensions	$1 \times 2 \times 2 \times 2$	1×2	1.398
EEG	No	1×8	1	1.301
	Along spatial dimension	$2 \times 3 \times 7$	2×1	1.685
	Along temporal dimension	$3 \times 1 \times 2$	1	1.334
	Along both dimensions	$2 \times 2 \times 5 \times 2$	1×4	1.698

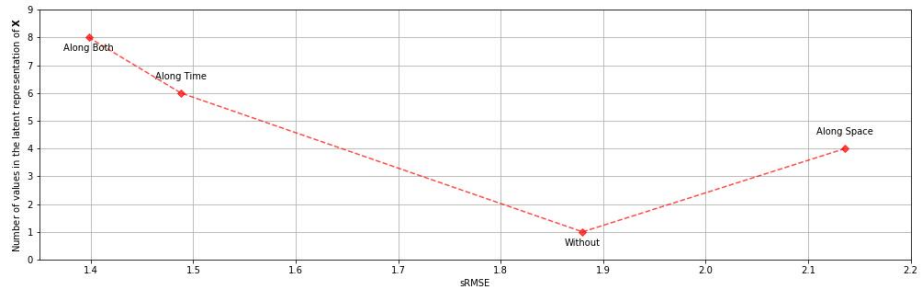


Figure 4.11: Comparison of models for Neurotycho dataset (ch=64) and HOPLS model

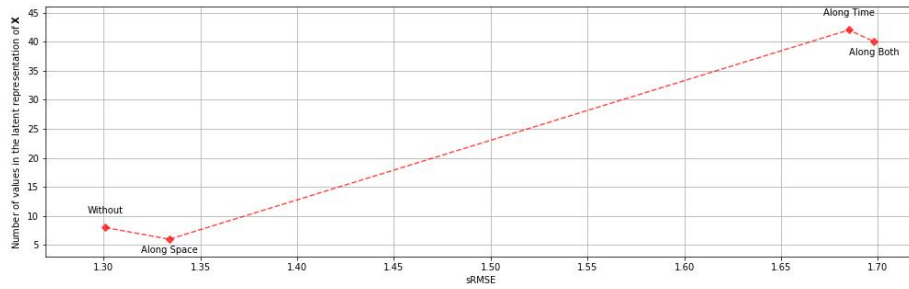


Figure 4.12: Comparison of models for Human EEG dataset (ch=60) and HOPLS model

Table 4.3: Results with TensorReducedNet with loss function \mathcal{L}_1 (3.5)

Dataset	Hankelization	Shape of the latent variable of $\underline{\mathbf{X}}$	sRMSE	Number of parameters of dimensionality reduction model
Neurotycho A	No	-	-	-
	Along spatial dimension	-	-	-
	Along temporal dimension	-	-	-
	Along both dimensions	-	-	-
Neurotycho K	No	-	-	-
	Along spatial dimension	-	-	-
	Along temporal dimension	-	-	-
	Along both dimensions	-	-	-
EEG	No	-	-	-
	Along spatial dimension	-	-	-
	Along temporal dimension	-	-	-
	Along both dimensions	-	-	-

Chapter 5

Discussion and conclusion

In this work, the influence of hankelization on forecasting quality was studied. We tested hankelization on three datasets and before three dimensionality reduction models.

Previously it was shown [19] that the forecasting of time series for other than BCI task is better with hankelization only along temporal dimension. Our results showed that for BCI, hankelization along spatial dimension works the best way for MPCA in many cases. For HOPLS, hankelization along temporal and spatial dimensions works better than only along temporal dimension. It can be because of high correlation between data from different electrodes.

Bibliography

- [1] Bengio, Y., and Lecun, Y. Convolutional networks for images, speech, and time-series.
- [2] Chao, Z. C., Nagasaka, Y., and Fujii, N. Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkey. *Frontiers in neuroengineering* (2010), 3.
- [3] Chao, Z. C., Nagasaka, Y., and Fujii, N. Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkeys. *Frontiers in Neuroengineering* 3 (2010).
- [4] Cichocki, A., Phan, A.-H., Zhao, Q., Lee, N., Oseledets, I., Sugiyama, M., Mandic, D. P., et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives. *Foundations and Trends® in Machine Learning* 9, 6 (2017), 431–673.
- [5] Eliseyev, A., and Aksenova, T. Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording. *PLoS ONE* 11 (2016).
- [6] Eliseyev, A., and Aksenova, T. Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording. *PloS one* (2016), 11(5).
- [7] Eyndhoven, S., Bousse, M., Hunyadi, B., Lathauwer, L., and Huffel, S. *Single-channel EEG classification by multi-channel tensor subspace learning and regression*. IEEE interentional workshop on machine learning for signal processing, 2018.
- [8] Jeong, J. H., Cho, J. H., Shim, K. H., Kwon, B. H., Lee, B. H., Lee, D. Y., Lee, D. H., and Lee, S. W. Multimodal signal dataset for 11 intuitive movement tasks from single upper extremity during multiple recording sessions. *GigaScience* 9 (2020).
- [9] Ji, T., Jiang, Y., Li, M., and Wu, Q. Ultra-short-term wind speed and wind power forecast via selective hankelization and low-rank tensor learning-based predictor. *International Journal of Electrical Power and Energy Systems* 140 (2022).
- [10] Lemm, S., Blankertz, B., Curio, G., and Müller, K. R. Spatio-spectral filters for improving the classification of single trial eeg. *IEEE Transactions on Biomedical Engineering* 52 (2005).
- [11] Li, Y., Yang, M., and Zhang, Z. A survey of multi-view representation learning. *IEEE Transactions on Knowledge and Data Engineering* 31 (2019).
- [12] Lu, H., Plataniotis, K. N., and Venetsanopoulos, A. N. MPCA: Multilinear principal component analysis of tensor objects. *IEEE transactions on Neural Networks* 19, 1 (2008), 18–39.
- [13] Miller, K. J. A library of human electrocorticographic data and analyses. *Nature human behaviour* 3, 11 (2019), 1225–1235.
- [14] Motrenko, A., and Strijov, V. Multi-way feature selection for ecog-based brain-computer interface. *Expert Systems with Applications* 114 (2018), 402–413.
- [15] Onishi, A., Phan, A. H., Matsuoka, K., and Cichocki, A. Tensor classification for p300-based brain computer interface.

- [16] Ran, X., Chen, W., Yvert, B., and Zhang, S. A hybrid autoencoder framework of dimensionality reduction for brain-computer interface decoding. *Computers in Biology and Medicine* 148 (9 2022).
- [17] Ran, X., Chen, W., Yvert, B., and Zhang, S. A hybrid autoencoder framework of dimensionality reduction for brain-computer interface decoding. *Computers in Biology and Medicine* (2022), 148.
- [18] Shen, Z., Liu, B., Zhou, Q., Liu, Z., Xia, B., and Li, Y. Cost-sensitive tensor-based dual-stage attention lstm with feature selection for data center server power forecasting. *ACM Transactions on Intelligent Systems and Technology* (10 2022).
- [19] Shi, Q., Yin, J., Cai, J., Cichocki, A., Yokota, T., Chen, L., Yuan, M., and Zeng, J. Block hankel tensor arima for multiple short time series forecasting.
- [20] Shimoda, K., Nagasaka, Y., Chao, Z. C., and Fujii, N. Decoding continuous three-dimensional hand trajectories from epidural electrocorticographic signals in japanese macaques. *Journal of Neural Engineering* 9 (2012).
- [21] Shimoda, K., Nagasaka, Y., Chao, Z. C., and Fujii, N. Decoding continuous three-dimensional hand trajectories from epidural electrocorticographic signals in japanese macaques. *Journal of neural engineering* (2012), 9(3).
- [22] Wang, X., Miranda-Moreno, L., and Sun, L. Hankel-structured tensor robust pca for multivariate traffic time series anomaly detection.
- [23] Wong, H. S., Wang, L., Chan, R., and Zeng, T. Deep tensor cca for multi-view learning. *IEEE Transactions on Big Data* 8 (2022).
- [24] Yokota, T., Erem, B., Guler, S., Warfield, S. K., and Hontani, H. Missing slice recovery for tensors using a low-rank model in embedded space.
- [25] Zhao, Q., Caiafa, C. F., Mandic, D. P., Chao, Z. C., Nagasaka, Y., Fujii, N., Zhang, L., and Cichocki, A. Higher order partial least squares (hopls): A generalized multilinear regression method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (2013).
- [26] Zhao, Q., Zhang, L., and Cichocki, A. Multilinear and nonlinear generalizations of partial least squares: an overview of recent advances. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 4, 2 (2014), 104–115.
- [27] Zhou, G., Zhao, Q., Zhang, Y., Adali, T., Xie, S., and Cichocki, A. Linked component analysis from matrices to high-order tensors: Applications to biomedical data. *Proceedings of the IEEE* 104 (2 2016), 310–331.