Skoltech

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
PhD, Professor

Moscow 2022

Skoltech

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 Евгеньевич
подпись
к.ф.-м.н., профессор

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

Москва 2022

Selection of Tensor Representations For Multiview Forecasting

Nadezhda Alsahanova

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

ABSTRACT

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: PhD, Professor

Contents

Chapter 1
Introduction

An initial data in 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 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 prediction for 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 [2], [17]. 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, Ajile12 dataset has human ECoG signals and 2D coordinates of limbs [12]. Another example is a human electroencephalography (EEG) dataset [7] that has 7-channel electromyography signals from arms.

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 [10]. One of the most commonly used dimensionality reduction methods is principal component analyses (PCA) and its modifications. One of the tensor nonlinear modification is Multi-view kernel PCA [6]. PCA and its modifications capture the greatest variance in the input data. However, it also includes the variance in noise signals.

All of 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 [3] - [23].

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 [22], Multi-linear PLS [23] and Multi-way PLS [3]. 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) [20] and Hybrid Autoencoder [13]. 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 was used in [11]. This approach helped to improve classification metrics. Time delay embedding used in [5] to improve classification quality for BCI task. This approach was inspired by the common spectral patterns method presented in [9] 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 in [24]. 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, in [19] hankelization was used for anomaly detection in traffic data. 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, in [8] forecasting of short term wind speed data was conducted via selective hankelization.

To apply hankelization for tensor data, multi-way delay embedding transform (MDT) was introduced in [21]. 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 in [16] and to forecast power consumption in [15]. in both works, its ability to capture hidden correlations increased the quality of forecasting.

Chapter 2
Problem statement

2.1 Notation

A *tensor* is a multidimensional array. The *order* of a tensor is the number of dimensions, also known as ways or modes. *Vectors* (tensors of order one) are denoted by $\mathbf{x}$. *Matrices* (tensors of order two) are denoted by $\mathbf{X}$. 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 $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is the square root of the sum of the squares of all its elements, i.e.,

$$\|\underline{\mathbf{X}}\| = \sqrt{\sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} x^2_{i_1 i_2 \ldots i_N}}.$$

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

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

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

Matricization: Transforming a Tensor into a Matrix. *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 $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is denoted by $\mathbf{X}_{(n)}$ and arranges the mode-n fibers to be the columns of the resulting matrix. Tensor element $(i_1, i_2, \ldots, i_N)$ 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

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

Tensor Multiplication: The n-mode Product. The n-mode (matrix) product of a tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ 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 $I_1 \times \ldots \times I_{n-1} \times J \times$

$I_{n+1} \times \ldots \times I_N$. Elementwise, we have

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

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.

2.2 Problem statement

In this chapter the task of decoding 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. A method of hankelization of tensors is explained. This chapter also provides an overview of methods for reducing the dimensionality.

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* $\mathbf{s}(t_i) = s_i \in \mathbb{R}^M$.

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 $\mathbf{s}(t)$, $\mathbf{y}(t)$: $(\underline{\mathbf{X}}, \mathbf{Y})$, where

$$\underline{\mathbf{X}} \in \mathbb{R}^{M \times n_1 \times \cdots \times n_D}, \qquad \mathbf{Y} \in \mathbb{R}^{M \times K}, \tag{2.1}$$

where $\mathbf{y}_m = \mathbf{y}(t_m)$, a $\underline{\mathbf{X}}_m \in \mathbb{R}^{n_1 \times \cdots \times n_D}$ is a tensor.

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

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

$$\Phi = \Psi_x \circ h_x \circ g \circ h_y^{-1} \circ \Psi_y^{-1} : \quad \mathbb{R}^{n_1 \times \ldots \times n_D} \to \mathbb{R}^K, \tag{2.2}$$

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

$$\begin{tikzcd}
\underline{\mathbf{X}} \arrow[r, "\Phi"] \arrow[d, "\Psi_x"'] & \mathbf{Y} \arrow[d, "\Psi_y"'] \\
\underline{\hat{\mathbf{X}}} \arrow[r, "\underline{\Phi}"] \arrow[d, "h_x"'] & \underline{\mathbf{Y}} \arrow[d, "h_y"'] \\
\mathbf{T} \arrow[r, "g"'] & \mathbf{U}
\end{tikzcd}$$

Definition 2.3 *Tensorization model* $\Psi : \mathbb{R}^{n_1 \times \ldots \times n_D} \to \mathbb{R}^{\ell_1 \times \ldots \times \ell_G}$, *where* $D < G$, *is a model that increase order of tensor from* D *to* G.

Definition 2.4 *$h : \mathbb{R}^{\ell_1 \times \ldots \times \ell_G} \to \mathbb{R}^{m_1 \times \ldots \times m_G}$, where$m_i < \ell_i$ is called a dimensionality reduction model, which reduces the dimensionality from $\mathbb{R}^{\ell_1 \times \ldots \times \ell_G}$ to $\mathbb{R}^{m_1 \times \ldots \times m_G}$, namely $\underline{\mathbf{Q}}_i = h_x\big(\underline{\mathbf{X}}_i, \theta\big)$, where θ are parameters of the model.*

Definition 2.5 *$g : \mathbb{R}^{m_1 \times \ldots \times m_G} \to \mathbb{R}^{k_1 \times \ldots \times k_D}$ is called an alignment model, which aligns each $\underline{\mathbf{Q}}_i \in \mathbb{R}^{m_1 \times \ldots \times m_D}$ to $\mathbf{P}_m \in \mathbb{R}^{k_1 \times \ldots \times k_D}$, namely $\mathbf{P}_i = g\big(\underline{\mathbf{Q}}_i, \eta\big)$, where η are parameters of the mode.*

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

$$\phi^* = \arg\min_{\{\theta,\eta\}} \mathcal{L}\left(\phi(\underline{\mathbf{X}}, \theta, \eta),\ \mathbf{Y}\right) \tag{2.3}$$

Method for tensorazation, hankelization, is described in 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 discribed 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 $\mathbf{x} = (x_1, \ldots, x_L)^\top \in \mathbb{R}^L$, a standard delay embedding transfom of $\mathbf{x}$ with τ is given by

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

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

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

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

Thus, time-series have a strong 1D structure: variables (or pixels) that are spatially or temporally nearby are highly correlated. Local correlations are the reasons for the well-known advantages of extracting and combining local features before recognizing spatial or temporal ob jects. Convolutional networks force the extraction of local features by restricting the receptive elds of hidden units to be local. (useful info)

Hankelization is an effective way to transform lower-order data to higher-order tensors. MDT is a multi-way extension of Hankelization. It combines multi-linear duplication and multi-way folding operations. By denoting $\hat{\underline{\mathbf{X}}} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_M}$ as the block Hankel tensor of $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, $N < M$, the MDT for $\underline{\mathbf{X}}$ is defined by

$$\hat{\underline{\mathbf{X}}} = \mathcal{H}_\tau(\underline{\mathbf{X}}) = \text{Fold}_{(\mathbf{I},\, \tau)} \left(\underline{\mathbf{X}} \times_1 \mathbf{S}_1 \ldots \times_N \mathbf{S}_N \right),$$

where $\mathbf{S}_n \in \mathbb{R}^{\tau_n (I_n - \tau_n + 1) \times I_n}$ is a duplication matrix and

$$\mathrm{Fold}_{(I,\,\tau)}:\mathbb{R}^{\tau_1(I_1-\tau_1+1)\times\ldots\times\tau_N(I_N-\tau_N+1)}\rightarrow\mathbb{R}^{\tau_1\times(I_1-\tau_1+1)\times\ldots\times\tau_N\times(I_N-\tau_N+1)}$$

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

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

where † is the Moore-Penrose pseudo-inverse.

3.2 MPCA

Multilinear Principal Component Analysis (MPCA) performs feature extraction by determing a miltilinear projection that captures most of the original tensorial input variation. Let $\{\underline{\mathbf{A}}_m,\ m=1,\ldots,M\}$ be a set of M tensor samples in $\mathbb{R}^{I_1}\otimes\mathbb{R}^{I_2}\otimes\ldots\otimes\mathbb{R}^{I_N}$. The total scatter of these tensors is defined as

$$\Psi_{\underline{\mathbf{A}}}=\sum_{m=1}^{M}\|\underline{\mathbf{A}}_m-\overline{\underline{\mathbf{A}}}\|_F^2,$$

where $\overline{\underline{\mathbf{A}}}$ is the mean tensor calculated as $\overline{\underline{\mathbf{A}}}=(1/M)\sum_{m=1}^{M}\underline{\mathbf{A}}_m$. The n-mode total scatter matrix of these samples is then defined as

$$\mathbf{S}_{T_{\underline{\mathbf{A}}}}^{(n)}=\sum_{m=1}^{M}(\mathbf{A}_{m(n)}-\overline{\mathbf{A}}_n)(\mathbf{A}_{m(n)}-\overline{\mathbf{A}}_n)^{\top},$$

where $\mathbf{A}_{m(n)}$ is the n-mode unfolded matrix of $\underline{\mathbf{A}}_m$. The MPCA objective is to define a multilinear transformation $\left\{\tilde{\mathbf{U}}^{(n)}\in\mathbb{R}^{I_n\times P_n},\ n=1,\ldots,N\right\}$ that maps the original tensor space $\mathbb{R}^{I_1}\otimes\mathbb{R}^{I_2}\otimes\ldots\otimes\mathbb{R}^{I_N}$ into a tensor subspace $\mathbb{R}^{P_1}\otimes\mathbb{R}^{P_2}\otimes\ldots\otimes\mathbb{R}^{P_N}$ (with $P_n<I_n, n=1,\ldots,N$):

$$\underline{\mathbf{Y}}_m=\underline{\mathbf{X}}_m\times_1\tilde{\mathbf{U}}^{(1)\top}\times_2\tilde{\mathbf{U}}^{(2)\top}\ldots\times_N\tilde{\mathbf{U}}^{(N)\top},\ \ m=1,\ldots,M,$$

such that $\left\{\underline{\mathbf{Y}}_m\in\mathbb{R}^{P_1}\otimes\mathbb{R}^{P_2}\otimes\ldots\otimes\mathbb{R}^{P_N},\ m=1,\ldots,M\right\}$ captures most of the variations observed in the original tensor objects, assuming that these variations are measured by the total tensor scatter.

The MPCA objective is the determination of the N projection matrices $\left\{\tilde{\mathbf{U}}^{(n)}\in\mathbb{R}^{I_n\times P_n},\ n=1,\ldots,N\right\}$ that maximize the total tensor scatter $\Psi_{\underline{\mathbf{Y}}}$

$$\left\{\tilde{\mathbf{U}}^{(n)}\in\mathbb{R}^{I_n\times P_n},\ n=1,\ldots,N\right\}=\arg\max_{\tilde{\mathbf{U}}^{(1)},\tilde{\mathbf{U}}^{(2)},\ldots,\tilde{\mathbf{U}}^{(n)}}\Psi_{\underline{\mathbf{Y}}}.$$

Here, the dimensionality P_n 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:

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

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

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

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

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

$$\mathbf{XW} = \mathbf{TP}^{\mathsf{T}}\mathbf{W}.$$

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

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

The matrix of the model parameters (3.7) could be found from the equations (3.2) and (3.3) as

$$\mathbf{Y} = \mathbf{UQ}^{\mathsf{T}} + \mathbf{E} \approx \mathbf{TBQ}^{\mathsf{T}} + \mathbf{E} = \mathbf{XW}^*\mathbf{BQ}^{\mathsf{T}} + \mathbf{E} = \mathbf{X}\mathbf{\Theta} + \mathbf{E}. \tag{3.4}$$

Thus, the model parameters (3.7) are equal to

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

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 $(N+1)$th-order independent tensor, $\underline{\mathbf{X}} \in \mathbb{R}^{M\times I_1\times\ldots\times I_N}$, and an $(N+1)$th-order dependent tensor, $\underline{\mathbf{Y}} \in \mathbb{R}^{M\times J_1\times\ldots\times J_N}$, 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).

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

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

where R is the number of latent vectors, $\mathbf{t}_r \in \mathbb{R}^M$ is the r-th latent vector, $\{\mathbf{P}_r^{(n)}\}_{n=1}^N \in \mathbb{R}^{I_n\times L_n}$ and $\{\mathbf{Q}_r^{(n)}\}_{n=1}^N \in \mathbb{R}^{J_n\times K_n}$ are the loading matrices in mode-n, and $\underline{\mathbf{G}}_{xr} \in \mathbb{R}^{1\times L_1\times\ldots\times L_N}$

and $\underline{\mathbf{G}}_{yr} \in \mathbb{R}^{1\times K_1\times\ldots\times K_N}$ are core tensors.

By defining a latent matrix $\mathbf{T} = \left[\mathbf{t}_1, \ldots, \mathbf{t}_R\right]$, mode-$n$ loading matrix $\overline{\mathbf{P}}^{(n)} = \left[\mathbf{P}_1^{(n)}, \ldots, \mathbf{P}_R^{(n)}\right]$, mode-$n$ loading matrix $\overline{\mathbf{Q}}^{(n)} = \left[\mathbf{Q}_1^{(n)}, \ldots, \mathbf{Q}_R^{(n)}\right]$ and core tensors

$$\underline{\mathbf{G}}_x = \text{blockdiag}\left(\underline{\mathbf{G}}_{x1}, \ldots, \underline{\mathbf{G}}_{xR}\right) \in \mathbb{R}^{R\times RL_1\times\ldots\times RL_N}, \underline{\mathbf{G}}_y = \text{blockdiag}\left(\underline{\mathbf{G}}_{y1}, \ldots, \underline{\mathbf{G}}_{yR}\right) \in \mathbb{R}^{R\times RK_1\times\ldots\times RK_N},$$

the HOPLS model can be rewritten as

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

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. 3.1) and their elements indicate the level of local interactions between the corresponding latent vectors and loading matrices.

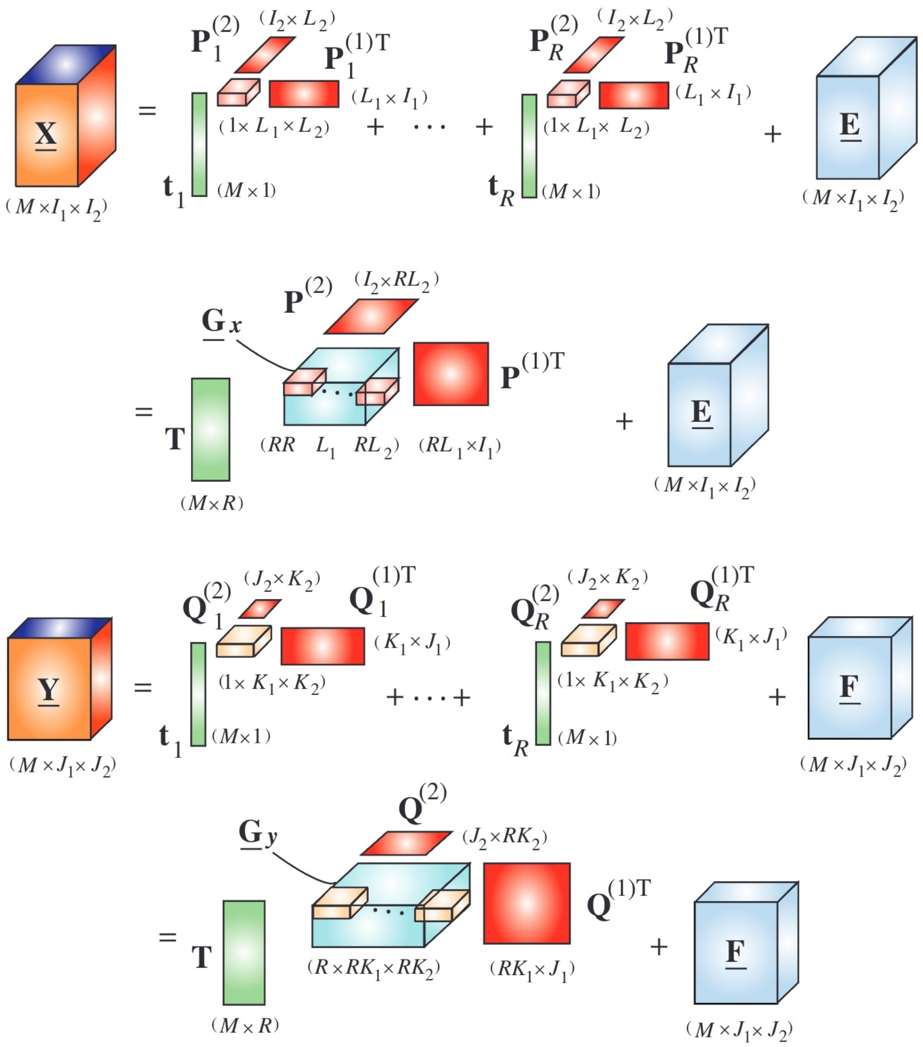

Figure 3.1: The HOPLS model which approximates the independent variables, $\underline{\mathbf{X}}$, as a sum of rank-$(1, L_1, L_2)$ tensors. The approximation for the dependent variables, $\underline{\mathbf{Y}}$, follows a similar principle, whereby the common latent components, $\mathbf{T}$, are shared between $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$.

The optimization problem can be formulated as

$$\left|\left|[\![\underline{\mathbf{C}}; \mathbf{P}^{(1)\top}, \ldots, \mathbf{P}^{(N)\top}, \mathbf{Q}^{(1)\top}, \ldots, \mathbf{Q}^{(N)\top}]\!]\right|\right|_F^2 \to \max_{\left\{\mathbf{P}^{(n)},\, \mathbf{Q}^{(n)}\right\}}, \text{s.t. } \mathbf{P}^{(n)\top}\mathbf{P}^{(n)} = \mathbf{I}_{L_n}, \quad \mathbf{Q}^{(n)\top}\mathbf{Q}^{(n)} = \mathbf{I}_{K_n},$$

where $[\![\ldots]\!]$ denotes the multilinear products between a tensor and a set of matrices, $\mathbf{P}^{(n)}$ and $\mathbf{Q}^{(n)}$, $n = 1, \ldots, N$, 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 [14] 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 [14] 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. The first modification of the ReducedNet is a ModifiedReducedNet (Fig. 3.2). The main difference between the ModifiedReducedNet and the ReducedNet is the presence of a second tunable parameter, such as the number of channels that is obtained at the output of convolutional blocks (N_F). That means, that if the input tensor is $\underline{\mathbf{X}} \in \mathbb{R}^{T \times F \times C}$, the output tensor after the fourth convolutional layer is in $\mathbb{R}^{T \times N_F \times C}$ and the output tensor after whole encoder is in $\mathbb{R}^{T \times N_{CH}}$. Moreover, the model ModifiedReducedNet is more complex than ReducedNet, because our data is more complex than data used in [14].

The loss, proposed in article [14], takes into account outputs from both modules:

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

where:

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

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

Modifications of ReducedNet The main problem of ReducedNet or ModifiedReducedNet is that encoder's output is a matrix. It means, that the input tensor losses its initial structure. Thus, in this work, TensorReducedNet is proposed (Fig. 3.3). The output from the encoder is in $\mathbb{R}^{T \times N_F \times N_{CH}}$.

Moreover, 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.

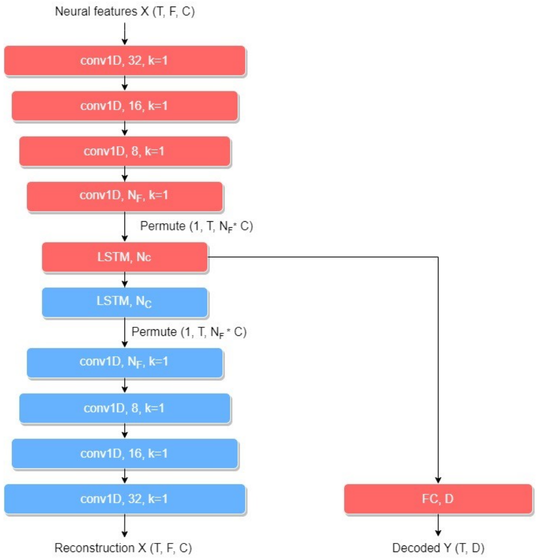

Figure 3.2: ModifiedReducedNet.

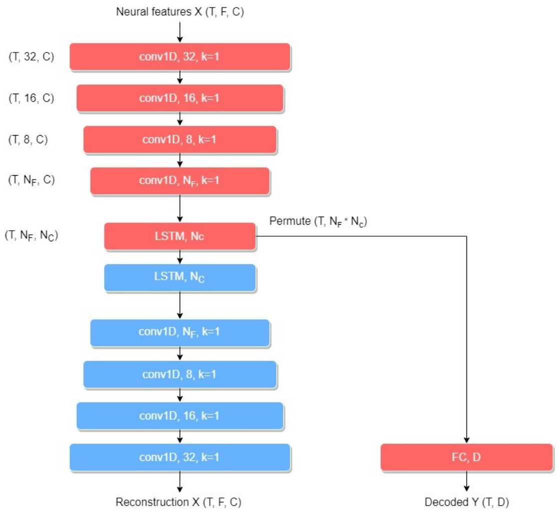

Figure 3.3: TensorReducedNet.

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:

$$\underline{\mathbf{X}}_{(1)} = \left[\operatorname{vec}\left(\underline{\mathbf{X}}_1\right)^\top, \dots, \operatorname{vec}\left(\underline{\mathbf{X}}_M\right)^\top\right]^\top \in \mathbb{R}^{M\times(n_1\cdot\ldots\cdot n_D)} \tag{3.6}$$

Thus, the problem becomes a multimodal regression problem, where $\mathbf{X} \in \mathbb{R}^{M\times(n_1\cdot\ldots\cdot n_D)}$ is the input matrix obtained by matricization of the input tensor, $\mathbf{Y} \in \mathbb{R}^{M\times K}$ is the target matrix.

The goal is to forecast a dependent variable $\mathbf{y}_m \in \mathbb{R}^K$ with K targets from an independent input object $\mathbf{x}_m \in \mathbb{R}^n$ with $n = n_1 \cdot \ldots \cdot n_D$ features. We assume that there is a linear dependence between the object $\mathbf{x}_m$ and the target variable $\mathbf{y}_m$ as

$$\mathbf{y}_m = \boldsymbol{\Theta}\mathbf{x}_m + \boldsymbol{\varepsilon}, \tag{3.7}$$

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

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^\mathsf{T} = [\boldsymbol{\chi}_1, \dots, \boldsymbol{\chi}_n]; \quad \mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_m]^\mathsf{T} = [\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_r].$$

The columns $\boldsymbol{\chi}_j$ of $\mathbf{X}$ correspond to the object features, and the columns $\boldsymbol{\nu}_j$ 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:

$$\mathcal{L}(\boldsymbol{\Theta}|\mathbf{X}, \mathbf{Y}) = \left\| \underset{M\times K}{\mathbf{Y}} - \underset{M\times n}{\mathbf{X}} \cdot \underset{K\times n}{\boldsymbol{\Theta}}^{\mathsf{T}} \right\|_2^2 \to \min_{\boldsymbol{\Theta}}. \tag{3.8}$$

The solution of (3.8) is given by

$$\boldsymbol{\Theta} = \mathbf{Y}^\mathsf{T}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}.$$

The linear dependent columns of $\mathbf{X}$ lead to an instable solution for the optimization problem (3.8). If there is a vector $\boldsymbol{\alpha} \neq \mathbf{0}_n$ such that $\mathbf{X}\boldsymbol{\alpha} = \mathbf{0}_m$, then adding $\boldsymbol{\alpha}$ to any column of $\boldsymbol{\Theta}$ does not change the value of the loss function $\mathcal{L}(\boldsymbol{\Theta}|\mathbf{X}, \mathbf{Y})$. In this case, the matrix $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ 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):

$$\mathbf{y}_m = \langle \underline{\mathbf{X}}_m | \underline{\mathbf{W}} \rangle + \varepsilon \tag{3.9}$$

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 $\underline{\mathbf{X}}_m \in \mathbb{R}^{n_1 \times \cdots \times n_D}$ and $(D+1)$-dimensional tensor of model parameters $\underline{\mathbf{W}} \in \mathbb{R}^{n_1 \times \cdots \times n_D \times K}$, $\varepsilon \in \mathbb{R}^K$ is an error, $\mathbf{y}_m \in \mathbb{R}^K$ is the target vector.

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

$$\langle \underline{\mathbf{X}}_m | \underline{\mathbf{W}} \rangle_k = \sum_{i_1=1}^{n_1} \cdots \sum_{i_D=1}^{n_D} x_{i_1,\dots,i_D} w_{i_1,\dots,i_D,k} \tag{3.10}$$

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

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

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

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

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

Chapter 4
Numerical experiments

4.1 Datasets

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

4.1.1 Neurotyco dataset

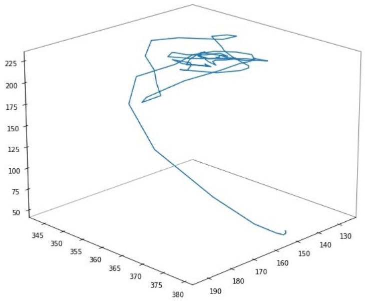

Figure 4.1: Hand movement trajectory

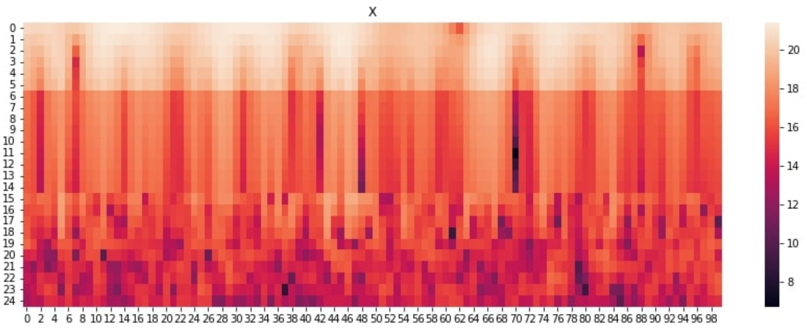

Figure 4.2: One channel data in time-frequency domain

One of the datasets for the computational experiment is electrocorticogram (ECoG) data from the Neurotycho[18] 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 [4], [1]. 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 0.7.

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[18] 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) , 7-channel electromyography (EMG), and 4-channel electrooculography (EOG) of healthy subjects. Subjects performed different upper-extremity movement tasks. The original EEG signals are converted into a space time representation using a wavelet transform with Morlet wavelet. So, $X \in \mathbb{R}^{T\times 64\times 24}$, where 24 - number of frequencies. $Y \in \mathbb{R}^{T\times 3}$ - coordinates of hand position.

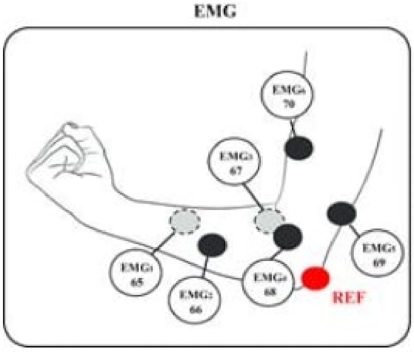

Figure 4.3: EMG electodes positions

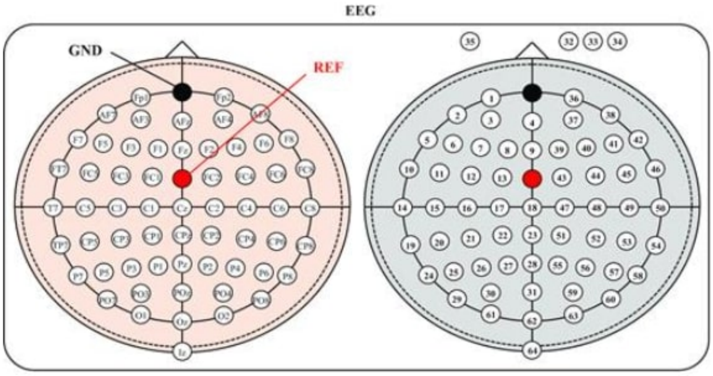

Figure 4.4: Electrode positions for EEG dataset

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.

$$\text{sRMSE}(\mathbf{Y}, \widehat{\mathbf{Y}}_\mathbf{a}) = \sqrt{\frac{\text{MSE}(\mathbf{Y}, \widehat{\mathbf{Y}}_\mathbf{a})}{\text{MSE}(\mathbf{Y}, \overline{\mathbf{Y}})}} = \frac{\|\mathbf{Y} - \widehat{\mathbf{Y}}_\mathbf{a}\|_2}{\|\mathbf{Y} - \overline{\mathbf{Y}}\|_2}. \tag{4.1}$$

Here $\widehat{\mathbf{Y}}_\mathbf{a} = \mathbf{X}_\mathbf{a}\mathbf{\Theta}_\mathbf{a}^\mathsf{T}$ is a model prediction and $\overline{\mathbf{Y}}$ 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

Table 4.1: Results with TensorReducedNet with loss function $\mathcal{L}_1$ (3.5)

Dataset	Hankelization	Shape of the latent variable	sRMSE	Number of parameters of dimensionality reduction model
Neurotyco A	No	-	-	-
	Along spatial dimension	-	-	-
	Along temporal dimension	-	-	-
	Along both dimensions	-	-	-
Neurotyco K	No	-	-	-
	Along spatial dimension	-	-	-
	Along temporal dimension	-	-	-
	Along both dimensions	-	-	-
EEG	No	-	-	-
	Along spatial dimension	-	-	-
	Along temporal dimension	-	-	-
	Along both dimensions	-	-	-

Table 4.2: Results with MPCA

Dataset	Hankelization	Shape of the latent variable	sRMSE	Number of parameters of dimensionality reduction model
Neurotyco A	No	-	-	-
	Along spatial dimension	-	-	-
	Along temporal dimension	-	-	-
	Along both dimensions	-	-	-
Neurotyco 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
No	-	-	-	EEG
	Along spatial dimension	-	-	-
	Along temporal dimension	-	-	-
	Along both dimensions	-	-	-

Table 4.3: Results with HOPLS

Dataset	Hankelization	Shape of the latent variable of **X**	Shape of the latent variable of **Y**	sRMSE
Neurotyco 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
Neurotyco 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	-	-	-
	Along spatial dimension	-	-	-
	Along temporal dimension	-	-	-
	Along both dimensions	-	-	-

Bibliography

[1] 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.

[2] 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).

[3] Eliseyev, A., and Aksenova, T. Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording. *PLoS ONE 11* (2016).

[4] Eliseyev, A., and Aksenova, T. Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording. *PloS one* (2016), 11(5).

[5] Eyndhoven, S., Bousse, M., Hunyadi, B., Lathauwer, L., and Huffel, S. *Single-channel EEG classification by multi-channel tensor subspacelearning and regression*. IEEE interentional workshop on machine learning for signel processing, 2018.

[6] Houthuys, L., and Suykens, J. A. Tensor learning in multi-view kernel pca. vol. 11140 LNCS.

[7] 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).

[8] 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).

[9] 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).

[10] Li, Y., Yang, M., and Zhang, Z. A survey of multi-view representation learning. *IEEE Transactions on Knowledge and Data Engineering 31* (2019).

[11] Onishi, A., Phan, A. H., Matsuoka, K., and Cichocki, A. Tensor classification for p300-based brain computer interface.

[12] Peterson, S. M., Singh, S. H., Dichter, B., Scheid, M., Rao, R. P., and Brunton, B. W. Ajile12: Long-term naturalistic human intracranial neural recordings and pose. *Scientific Data 9* (12 2022).

[13] 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).

[14] 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.

[15] 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).

[16] 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.

[17] 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).

[18] 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).

[19] Wang, X., Miranda-Moreno, L., and Sun, L. Hankel-structured tensor robust pca for multivariate traffic time series anomaly detection.

[20] Wong, H. S., Wang, L., Chan, R., and Zeng, T. Deep tensor cca for multi-view learning. *IEEE Transactions on Big Data 8* (2022).

[21] 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.

[22] 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).

[23] 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.

[24] 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.