Deep canonical correlation analysis

Yaushev Farukh
Optimization Class Project. MIPT

Introduction

In this paper, we solve the problem of predicting the target variable with the presence of dependencies (linear and nonlinear). The problem is that the source data has a high dimension and there are hidden dependencies in the spaces of the target and independent variables. Excessive dimensionality of spaces and multiple correlations lead to instability of the model. To solve this problem, we propose to build a model that takes into account both of these dependencies. The model translates data into low-dimensional spaces and data alignment occurs in the resulting hidden space. Alignment means capturing the relationship between the target variable and the independent variable through feature alignment. This strategy aims to use the additional knowledge contained in different spaces to get a more informative representation of the data.

Two experiments are being conducted. The first experiment on the example of comparing linear CCA and its nonlinear modification Deep CCA aims to show that sometimes it is not enough to take into account only linear dependencies in the source data spaces. The second experiment aims to test the hypothesis about the importance of data alignment. Several models with data conversion are compared. PLS is used as the basic algorithm.

Problem statement

Let the selection be given $(\mathbf{X}, \mathbf{Y})$, $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in R^{n\times m}$ — matrix of independent variables, $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n] \in R^{n\times k}$ — matrix of target variables.

It is assumed that there is a dependency between $\mathbf{X}$ and $\mathbf{Y}$:

$$\mathbf{Y} = f(\mathbf{X}) + \varepsilon,$$

where f — the function of regression, ε — the matrix of regression errors.

The error function is a quadratic loss function:

$$\mathcal{L}(f|\mathbf{X}, \mathbf{Y}) = \|\mathbf{X} - f(\mathbf{Y})\|_2^2 \to \min_f .$$

General scheme of a model with alignment

$$\begin{tikzcd} \underset{n\times m}{\mathbf{X}} \arrow[r, "f"] \arrow[d, shift right, "\varphi_1"'] & \underset{n\times k}{\mathbf{Y}} \arrow[d, shift left, "\psi_1"] \\ \underset{n\times p}{\mathbf{T}} \arrow[r, "g"] \arrow[u, shift right, "\varphi_2"'] & \underset{n\times p}{\mathbf{U}} \arrow[u, shift left, "\psi_2"] \end{tikzcd}$$

g – alignment function.
φ_1, ψ_1 – encoding functions
φ_2, ψ_2 – decoding functions

The optimal parameters $\theta^*_{\varphi_1}, \theta^*_{\psi_1}$ for functions φ_1 and ψ_1 are found as follows:

$$(\theta^*_{\varphi_1}, \theta^*_{\psi_1}) = \underset{(\theta_{\varphi_1}, \theta_{\psi_1})}{\operatorname{argmax}}[g(\varphi_1(\mathbf{X}; \theta_{\varphi_1}), \psi_1(\mathbf{Y}; \theta_{\psi_1}))].$$

Dependency between $\mathbf{T}$ and $\mathbf{U}$ after switching to the latent space:

$$\mathbf{U} = h(\mathbf{T}) + \boldsymbol{\eta},$$

where h — regression dependency function, $\boldsymbol{\eta}$ — regression error matrix.

Optimal h Selected by minimizing the error function. Use the quadratic loss error function $\mathcal{L}$ on $\mathbf{T}$ and $\mathbf{U}$:

$$\mathcal{L}(h|\mathbf{T}, \mathbf{U}) = \Big\| \underset{n\times p}{\mathbf{U}} - h(\underset{m\times p}{\mathbf{T}}) \Big\|_2^2 \to \min_h .$$

The final model looks like:

$$f = \psi_2 \circ h \circ \varphi_1.$$

CCA, PLS

- CCA finds two sets of basis vectors $\{\mathbf{w_x}_i\}_{i=1}^p$, $\mathbf{w_x} \in R^m$and$\{\mathbf{w_y}_i\}_{i=1}^p$, $\mathbf{w_y} \in R^k$, one for $\mathbf{X}$ and another for $\mathbf{Y}$, as follows

$$(\mathbf{w_x}, \mathbf{w_x}) = \underset{\mathbf{w_x}, \mathbf{w_x}}{\operatorname{argmax}}\, g(\mathbf{Xw_x}, \mathbf{Yw_y}) = \underset{\mathbf{w_x}, \mathbf{w_x}}{\operatorname{argmax}}\, \mathbf{corr}(\mathbf{Xw_x}, \mathbf{Yw_y})$$

- PLS also finds the basis vectors by maximizing the covariance:

$$(\mathbf{w_x}, \mathbf{w_x}) = \underset{\mathbf{w_x}, \mathbf{w_x}}{\operatorname{argmax}}\, \mathbf{cov}(\mathbf{Xw_x}, \mathbf{Yw_y})$$

So the encoding functions for CCA and PLS

$$\varphi_1(\mathbf{X}) = \mathbf{XW_x}, \quad \psi_1(\mathbf{Y}) = \mathbf{YW_y}$$

As you can see both of these algorithms are linear.

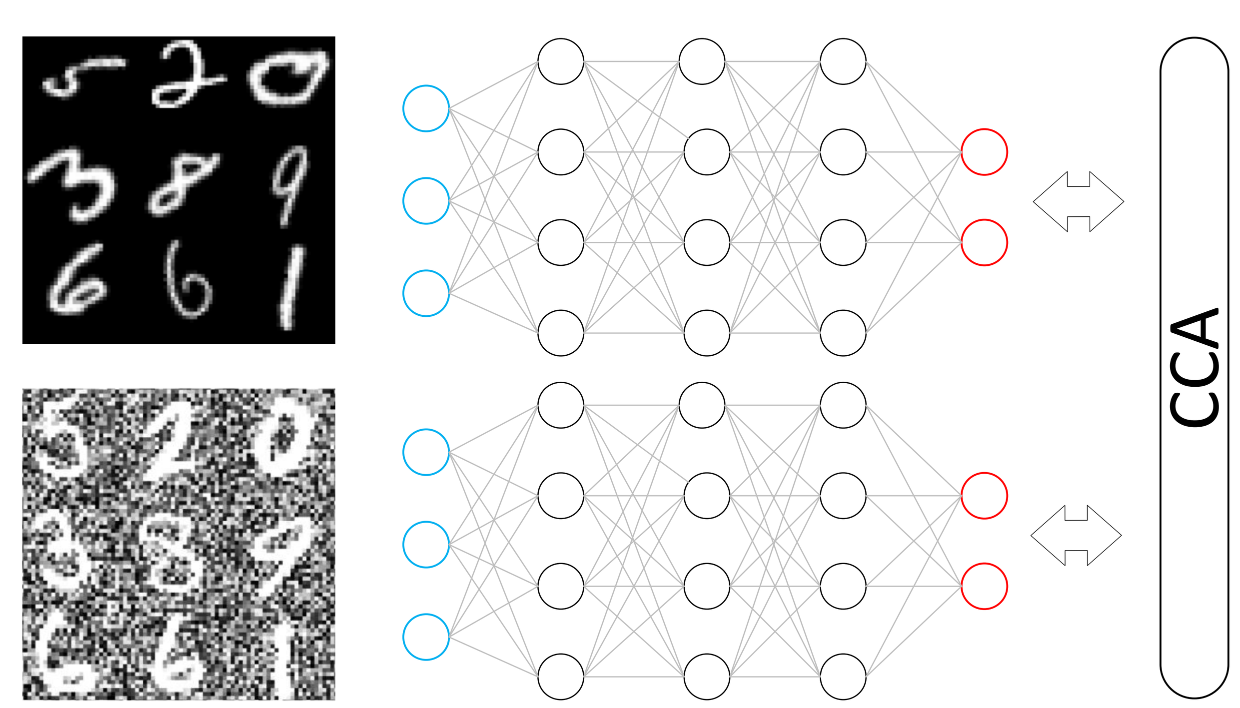
- Deep CCA — nonlinear modification of CCA based on neural networks

Experiment 1

In order to demonstrate that ignoring non-linear dependencies can lead to unsatisfactory results, we compare CCA and Deep CCA for the task of classifying noisy images from the MNIST dataset.

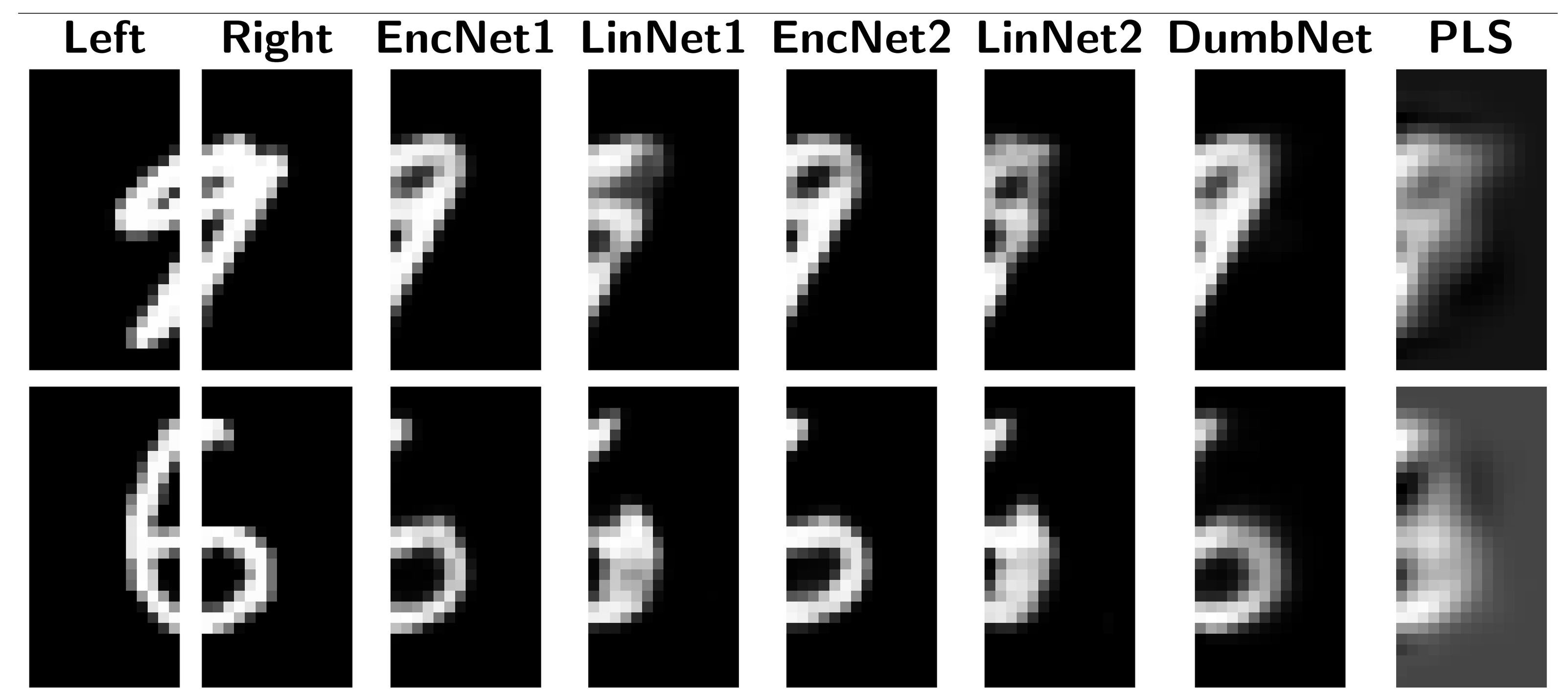
By applying Deep CCA and CCA to two images datasets, we get a new low-dimensional feature space that ignores noise in the source data. The resulting encoding functions φ_1. On the new data view (on the first set of images after applying the function φ_1), we use linear SVM for classification (1).

illustration of Deep CCA

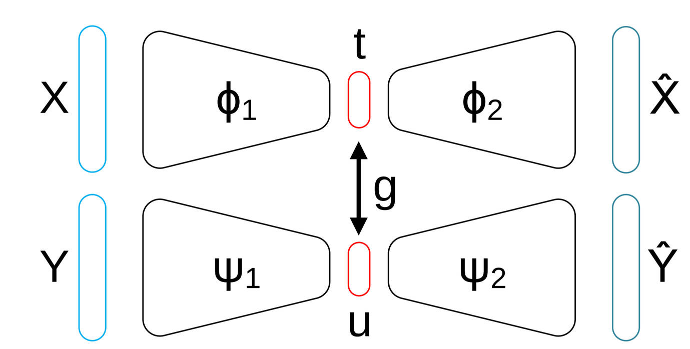


Experiment 2

The MNIST dataset is used, where each image is divided into two parts. The task of restoring the right part of the image by left (2).



Autoencoder with joint loss function



EncNet1 and LinNet1 train on transformed auto-encoder with joint data loss function.

EncNet2 and LinNet2 use separate autoencoders.

DumbNet is trained on source data and has the same structure as Enc Net with an autoencoder.

Results

Table 1: Getting a new feature space of dimension 15 using DCCA and CCA. An indicator of effectiveness will be the accuracy of the classification of linear SVM (ACC).

	DeepCCA(L=3)	CCA	PCA
Validation data	92.74%	76.21%	72.84%
Test data	92.14%	76.07%	72.84%

Table 2: Restore the right side of the image by left using different models. To measure the quality of the models, the standard deviation from the original image is considered.

	EncNet1	LinNet1	EncNet2
Number of weight coefficients	283k	239k	283k
MSE loss	0.151 ± 0.009	0.23 ± 0.01	0.151 ± 0.009
	LinNet2	**DumbNet**	**PLS**
Number of weight coefficients	239k	283k	154k
MSE loss	0.23 ± 0.01	0.146 ± 0.006	0.188 ± 0.001

Source code

Conclusion

The paper considers the task of decoding objects of complex structure.

A predictive model was Proposed with the alignment of the independent and target variables in a low-dimensional hidden space.

omputational experiments were conducted.