Deep CCA Auto-encoders
(DCCA, disDCCA, DCCAE, disDCCAE)

Muskan Hoondlani(0801CS171047)
gs0801cs171047@sgsitsindore.in

Alefiya Kothari (0801CS171009)
gs0801cs171009@sgsitsindore.in

Anshul Rathore(0801CS171012)
gs0801cs171012@sgsitsindore.in

December 13, 2020

Contents

Introduction

As a leap forward in our journey to learn, understand, implement and explore the various facades of Data Analysis, we have in the following study, explored the wide and vivid avenue of Multi view learning. In pursuit of understanding multi-view learning, we have delved into many of its intricacies and concepts of significance, which is also the highlight of this study— Deep Canonical correlation Analysis, Deep CCA is a novel concept and is still being studied by researchers to disclose its full potential.

The crux of data analysis lies in the fact that most real world data is not singular . i.e: it cannot be considered from one view point alone, even if it could be, it would not impart information that is not mutually exclusive. So, to begin with, data is always imparting one information in reference to another, data viewed singularly would make the data universe immeasurably vast with very scalar, incoherent and unrelated bits of information. Here comes the concept of viewing data from various standpoints and weighing the amount of information it can offer in another subview. This is the base and foundation of correlations that emerge between data bits and collectively generate information. We can say that multi-view learning (MVL) is a strategy for fusing data from different sources or subsets.
Let's understand Multi-view learning in all of it's tones.

Multi-View Learning -

Multi-view learning is also known as data fusion or data integration from multiple feature sets. It is an emerging concept in machine learning which considers learning with multiple views to improve the generalization performance. We can say that multi-view learning (MVL) is a strategy for fusing data from different sources or subsets. Many real-world datasets can be described from multiple "viewpoints".

Examples:

- Pictures taken from different angles of the same object
- Different language expressions of the same semantic, texts and images on the same web page, etc.
- A person can be identified by face, fingerprint, signature or iris with information obtained from multiple sources

The representations from different perspectives can be treated as different views. The essence of multi-view learning (MVL) is to exploit the consensual and complementary information between different views to achieve better learning performance.

MVL approaches can be divided into three major categories:

1. **Co-training** - which exchanges discriminative information between two views by training the two models alternately.
2. **Multi-kernel learning** - which maps data to different feature spaces with different kernels, and then combines those projected features from all spaces.
3. **Subspace learning** - which assumes all views are generated from a latent common space where shared information of all views can be exploited.

Though there is significant variance in the approaches to integrating multiple views to improve learning performance, they mainly exploit either the consensus principle or the complementary principle to ensure the success of multi-view learning.

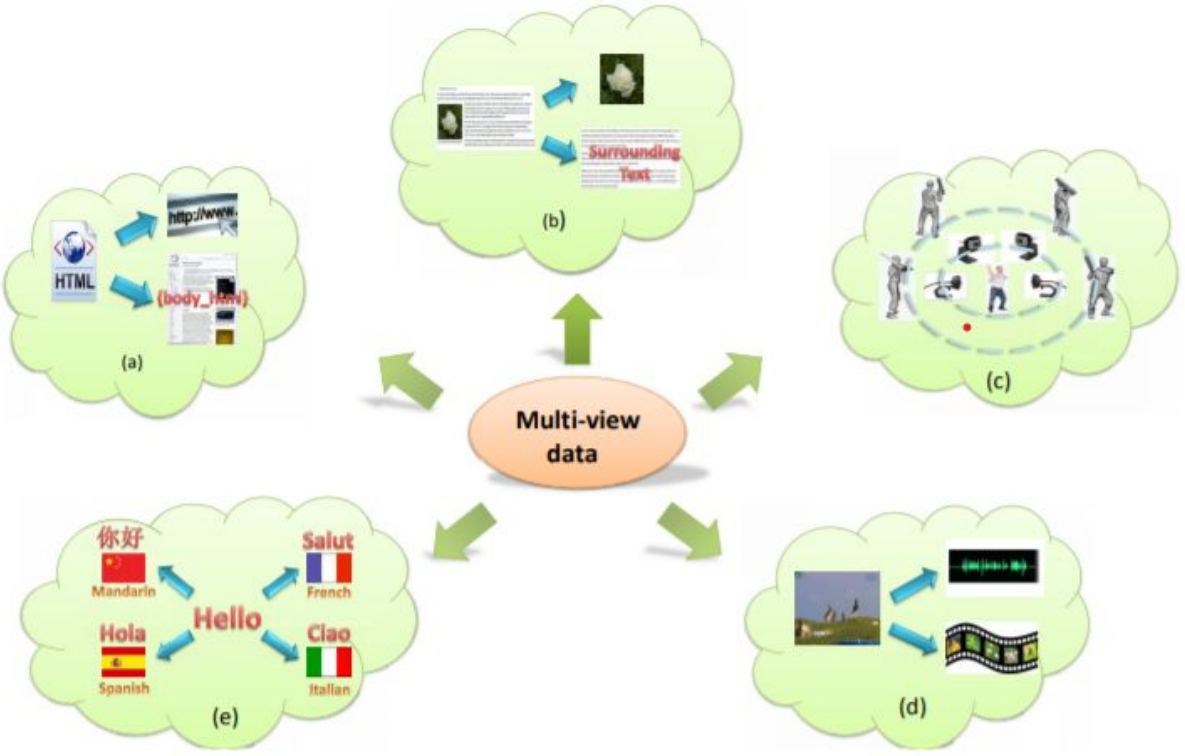


Figure 1: Multi-view data: a) a web document can be represented by its url and words on the page, b) a web image can be depicted by its surrounding text separate to the visual information, c) images of a 3D object taken from different viewpoints, d) video clips are combinations of audio signals and visual frames, e) multilingual documents have one view in each language.

Why was single-view learning insufficient?

Conventional machine learning algorithms, such as support vector machines, discriminant analysis, kernel machines, and spectral clustering, concatenate all multiple views into one single view to adapt to the learning setting. However, this concatenation causes overfitting in the case of a small size training sample and is not physically meaningful because each view has a specific statistical property. In contrast to single view learning, multi-view learning as a new paradigm introduces one function to model a particular view and jointly optimizes all the functions to exploit the redundant views of the same input data and improve the learning performance.

Several methods have been designed to tackle this problem by constructing a latent subspace shared by multiple views, in which distinct views are connected with one another in this subspace, integrating the complementary information underlying different views.
The demand for redundant views of the same input data is a major difference between multi-view and single-view learning algorithms. Thanks to these multiple views, the learning task can be conducted with abundant information.

However if the learning method is unable to cope appropriately with multiple views, these views may even degrade the performance of multi-view learning. Through fully considering the relationships between multiple views, several successful multi-view learning techniques have been proposed. We analyze these various algorithms and observe that there are two significant principles ensuring their success: consensus and complementary principles.

Thus moving forward, we will take a deeper look into one of the Multi-view learning techniques - Subspace learning.

What is subspace Learning?

Subspace learning-based approaches aim to obtain a latent subspace shared by multiple views by assuming that the input views are generated from this latent subspace. The dimensionality of the latent subspace is lower than that of any input view, so subspace learning is effective in reducing the “curse of dimensionality”. Given this subspace, it is straightforward to conduct the subsequent tasks, such as classification and clustering.

Correlation between views is an important consideration in subspace-based approaches for multi-view learning. Hotelling (1936) introduced canonical correlation analysis (CCA) to describe the linear relation between two views which aims to compute a low-dimensional Multi-view Learning of both views of variables such that the correlations among the variables

between the two views is maximized in the embedded space. Since the new subspace is simply a linear system of the original space, CCA can only be used to describe linear relation.

Canonical Correlation Analysis -

Canonical correlation analysis is a method for exploring the relationships between two multivariate sets of variables (vectors), all measured on the same individual.

Consider the following **examples** to understand CCA better:

1. Variables related to exercise and health. On one hand, you have variables associated with exercise, observations such as the climbing rate on a stair stepper, how fast you can run a certain distance, the amount of weight lifted on bench press, the number of push-ups per minute, etc. On the other hand, you have variables that attempt to measure overall health, such as blood pressure, cholesterol levels, glucose levels, body mass index, etc. Two types of variables are measured and the relationships between the exercise variables and the health variables are of interest.
2. Consider a group of sales representatives, on whom we have recorded several sales performance variables along with several measures of intellectual and creative aptitude. We may wish to explore the relationships between the sales performance variables and the aptitude variables.

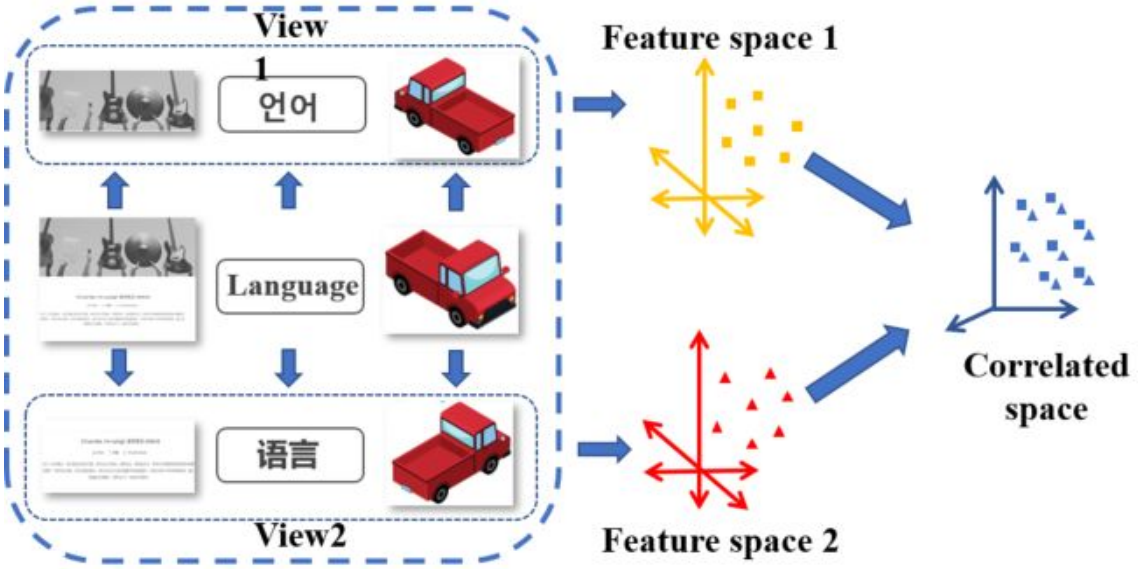
One approach to studying relationships between the two sets of variables is to use canonical correlation analysis which describes the relationship between the first set of variables and the second set of variables. We do not necessarily think of one set of variables as independent and the other as dependent, though that may potentially be another approach.

There are two typical **purposes** of CCA:

1. **Data reduction**: explain covariation between two sets of variables using small number of linear combinations
2. **Data interpretation**: find features (i.e., canonical variates) that are important for explaining covariation between sets of variables.

Finding correlated representations can be used to:

1. Provide insight into the data
2. Detect asynchrony in test data
3. Remove noise that is uncorrelated across views
4. Induce features that capture some of the information of the other view, if it is unavailable at test time.



But there are some **limitations of CCA** also -

1. It cannot handle more than two views.
2. It can only calculate the linear correlation between two views, whereas in many real-world applications the true relationship between the views may be nonlinear.
3. In supervised classification, labels are available; however, CCA, as an unsupervised algorithm, completely ignores the labels, and hence wastes information.

CCA is a linear dimensionality reduction method. However, there are many nonlinear relationships between features in practice. There will be under-fitting phenomenon when learning with CCA under a nonlinear circumstance. To solve the problem, several approaches have been proposed, e.g. kernel based methods.

Mathematical Formulation:

CCA is a classical technique to find linear relationships:

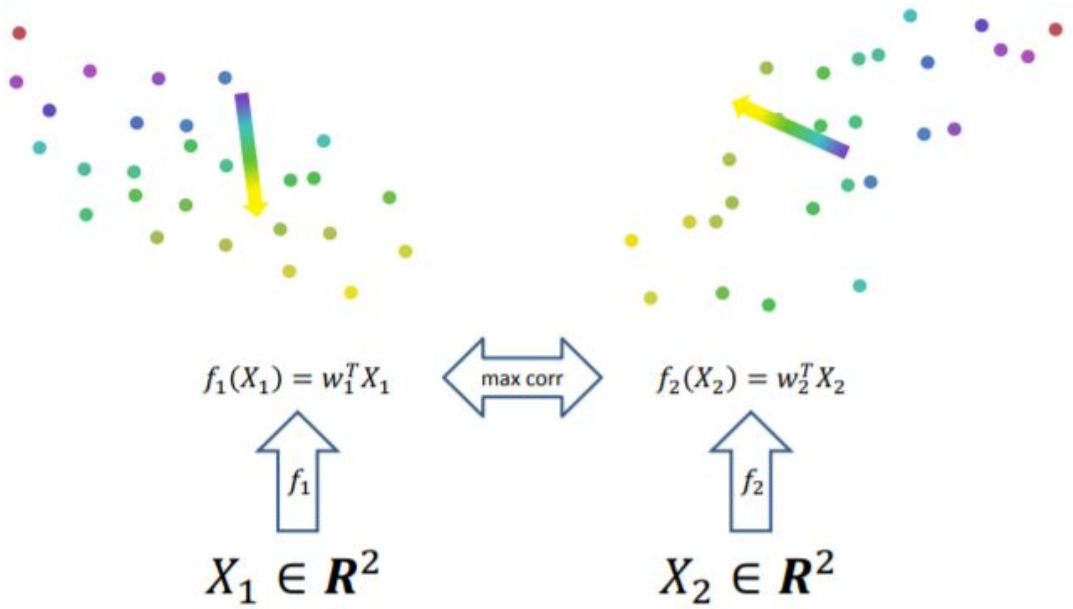
$$f_1(x_i) = W_1' x_1 \text{ for } W_1 \in \mathbb{R}^{n_1 \times k} \text{ (and } f_2\text{)}$$

The first columns (w_1^1, w_2^1) of the matrices W_1 and W_2 are found to maximize the correlation of the projection

$$(w_1^1, w_2^1) = \underset{w_1, w_2}{\operatorname{argmax}} \operatorname{corr}(w_1' X_1, w_2' X_2)$$

Subsequent pairs (w_1^i, w_2^i) are constrained to be uncorrelated with previous components: For j < i,

$$\operatorname{corr}((w_1^i)' X_1, (w_1^j)' X_1)) = \operatorname{corr}((w_2^i)' X_2, (w_2^j)' X_2) = 0.$$



Two views of each instance have the same colour

Canonical correlation is appropriate in the same situations where multiple regression would be, but where there are multiple intercorrelated outcome variables.

CCA is the analysis of multiple-X multiple-Y correlation. The Canonical Correlation Coefficient measures the strength of association between two Canonical Variates. A Canonical Variate is the weighted sum of the variables in the analysis. The canonical variate is denoted CV.

In normal regression or factor analysis the factors are calculated to maximize between-group variance while minimizing in-group variance. They are factors because they group the underlying variables.
Canonical Variants are not factors because only the first pair of canonical variants groups the variables in such a way that the correlation between them is maximized. The second pair is constructed out of the residuals of the first pair in order to maximize correlation between them. Therefore the canonical variants cannot be interpreted in the same way as factors in factor analysis. Also the calculated canonical variates are automatically orthogonal, i.e., they are independent from each other.

Formulation -

CCA, Kernel CCA, Deep CCA all learn functions $f_1(x_1)$ and $f_2(x_2)$

To maximize,

$$\text{corr}(f_1(x_1), f_2(x_2)) = \frac{\text{cov}(f_1(x_1), f_2(x_2))}{\sqrt{\text{var}(f_1(x_1)) \cdot \text{var}(f_2(x_2))}}$$

Algorithm:

CCA Algorithm -

Let's begin with the notation,
We have two sets of variables X = $\begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{pmatrix}$

And Y = $\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_q \end{pmatrix}$

We select X and Y based on the number of variables that exist in each set so that $p \leq q$ and we look at linear combinations of the data, similar to principal components analysis.

We define a set of linear combinations named U and V. U corresponds to the linear combinations from the first set of variables, X, and V corresponds to the second set of variables, Y. Each member of U is paired with a member of V. For example, U_1 below is a linear combination of the p X variables and V_1 is the corresponding linear combination of the q Y variables. Similarly, U_1 is a linear combination of the p X variables, and V_2 is the corresponding linear combination of the q Y variables. And, so on....

$$
\begin{aligned}
U_1 &= a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p \\
U_2 &= a_{21}X_1 + a_{22}X_2 + \cdots + a_{2p}X_p \\
&\vdots \\
U_p &= a_{p1}X_1 + a_{p2}X_2 + \cdots + a_{pp}X_p
\end{aligned}
$$

$$
\begin{aligned}
V_1 &= b_{11}Y_1 + b_{12}Y_2 + \cdots + b_{1q}Y_q \\
V_2 &= b_{21}Y_1 + b_{22}Y_2 + \cdots + b_{2q}Y_q \\
&\vdots \\
V_p &= b_{p1}Y_1 + b_{p2}Y_2 + \cdots + b_{pq}Y_q
\end{aligned}
$$

Thus define, (U_1,V_1) as the i^{th} canonical variate pair. (U_1, V_1) is the first canonical variate pair, similarly (U_2, V_2) would be the second canonical variate pair and so on. With $p \leq q$ there are p canonical covariate pairs.

We hope to find linear combinations that maximize the correlations between the members of each canonical variate pair.

We compute the variance of U_i variables with the following expression:

$$\mathbf{var}(U_i) = \sum_{k=1}^{p}\sum_{l=1}^{p} a_{ik}a_{il}cov(X_k, X_l)$$

The co-efficients a^{i1} through a^{ip} that appear in the double sum are the same coefficients that appear in the definition of . The covariances between the K^{th} and l^{th} X-variables are multiplied by the corresponding coefficients a^{ik} and a^{il} for the variate U_i .

Similar calculations can be made for the variance of V_j as shown below:

$$\text{var}(V_j) = \sum_{k=1}^{p} \sum_{l=1}^{q} b_{jk} b_{jl} \text{cov}(Y_k, Y_l)$$

The covariance between U_i and V_j is:

$$\text{cov}(U_i, V_j) = \sum_{k=1}^{p} \sum_{l=1}^{q} a_{ik} b_{jl} \text{cov}(X_k, Y_l)$$

The correlation between U_i and V_j is calculated using the usual formula. We take the covariance between the two variables and divide it by the square root of the product of the variances:

$$\frac{\text{cov}(U_i, V_j)}{\sqrt{\text{var}(U_i)\text{var}(V_j)}}$$

The *canonical correlation* is a specific type of correlation. The canonical correlation for the i^{th} canonical variate pair is simply the correlation between U_i and V_i:

$$\rho_i^* = \frac{\text{cov}(U_i, V_i)}{\sqrt{\text{var}(U_i)\text{var}(V_i)}}$$

This is the quantity to maximize. We want to find linear combinations of the *X*'s and linear combinations of the *Y*'s that maximize the above correlation.

Canonical Variates -

Let us look at each of the p canonical variates pair one by one.

First canonical variate pair: (U_1, V_1)**:**

The coefficients $a_{11}, a_{12}, \ldots, a_{1p}$ and $b_{11}, b_{12}, \ldots, b_{1q}$ are selected to maximize the canonical correlation ρ_1^* of the first canonical variate pair. This is subject to the constraint that variances of the two canonical variates in that pair are equal to one.

$$\text{var}(U_1) = \text{var}(V_1) = 1$$

This is required to obtain unique values for the coefficients.

Second canonical variate pair: (U_2, V_2)

Similarly we want to find the coefficients $a_{21}, a_{22}, \ldots, a_{2p}$ and $b_{21}, b_{22}, \ldots, b_{2q}$ that maximize the canonical correlation ρ_2^* of the second canonical variate pair, (U_2, V_2). Again, we will maximize this canonical correlation subject to the constraints that the variances of the individual canonical variates are both equal to one. Furthermore, we require the additional constraints that (U_1, U_2), and (V_1, V_2) are uncorrelated. In addition, the combinations (U_1, V_2) and (U_2, V_1) must be uncorrelated. In summary, our constraints are:

$$\begin{aligned}
\text{var}(U_2) &= \text{var}(V_2) = 1, \\
\text{cov}(U_1, U_2) &= \text{cov}(V_1, V_2) = 0, \\
\text{cov}(U_1, V_2) &= \text{cov}(U_2, V_1) = 0.
\end{aligned}$$

Basically, we require that all of the remaining correlations equal zero.

This procedure is repeated for each pair of canonical variates. In general, ...

i^{th} canonical variate pair: (U_i, V_i)

We want to find the coefficients $a_{i1}, a_{i2}, \ldots, a_{ip}$ and $b_{i1}, b_{i2}, \ldots, b_{iq}$ that maximize the canonical correlation ρ_i^* subject to the constraints that

$$\begin{aligned}
\text{var}(U_i) &= \text{var}(V_i) = 1, \\
\text{cov}(U_1, U_i) &= \text{cov}(V_1, V_i) = 0, \\
\text{cov}(U_2, U_i) &= \text{cov}(V_2, V_i) = 0, \\
&\vdots \\
\text{cov}(U_{i-1}, U_i) &= \text{cov}(V_{i-1}, V_i) = 0, \\
\text{cov}(U_1, V_i) &= \text{cov}(U_i, V_1) = 0, \\
\text{cov}(U_2, V_i) &= \text{cov}(U_i, V_2) = 0, \\
&\vdots \\
\text{cov}(U_{i-1}, V_i) &= \text{cov}(U_i, V_{i-1}) = 0.
\end{aligned}$$

Again, requiring all of the remaining correlations to be equal to zero.

Let's take a broader look at other variants of CCA:

Kernel CCA -

There may be nonlinear functions f1, f2 that produce more highly correlated representations than linear maps. Kernel CCA is the principal method to detect such functions.

Advantages of KCCA over linear CCA -

- More complex function space can yield dramatically higher correlation with sufficient training data.
- Can be used to produce features that improve performance of a classifier when second view is unavailable at test time

Disadvantages -

- Slower to train
- Training set must be stored and referenced at test time
- Model is more difficult to interpret

On this note,
We introduce **Deep Canonical Correlation Analysis (DCCA)**, a method to learn complex nonlinear transformations of two views of data such that the resulting representations are highly linearly correlated.

Parameters of both transformations are jointly learned to maximize the (regularized) total correlation. It can be viewed as a nonlinear extension of the linear method canonical correlation analysis (CCA). It is an alternative to the nonparametric method kernel canonical correlation analysis (KCCA) for learning correlated nonlinear transformations. Unlike KCCA, DCCA does not require an inner product.

Deep Canonical Correlation Analysis (DCCA)

As we have seen already both CCA and KCCA are techniques for learning representations of two data views, such that each view's representation is simultaneously the most predictive of, and the most predictable by, the other. CCA and KCCA have been used for unsupervised data analysis when multiple views are available, learning features for a single view when another view is available for representation learning but not at prediction time and reducing sample complexity of prediction problems using unlabeled data.

The applications range broadly across a number of fields like :

- Medicine
- Meteorology
- Biology and neurology etc
- Natural Language Processing
- Computer Vision
- Multimedia and signal processing

Why do we need Deep CCA?

While kernel CCA allows learning of nonlinear representations, it has the drawback that the representation is limited by the fixed kernel. Also, as it is a nonparametric method, the time required to train KCCA or compute the representations of new data points scales poorly with the size of the training set.
So we venture on to learning flexible nonlinear representations via deep networks— Deep CCA

To understand Deep Canonical correlation analysis, it's crucial to understand what the word Deep implies:-
"Deep" networks, having more than two layers, are capable of representing nonlinear functions involving multiple nested high-level abstractions of the kind that may be necessary to accurately model complex real world data.

While deep networks are more used to learn nonlinear transformations of two datasets to a space in which the data is highly correlated, just as KCCA does. The same properties that account for its success are - high model complexity the ability to concisely represent a hierarchy of features for modeling real-world data distributions

These properties are useful in settings where the output space is significantly more complex than a single label.

Deep networks parametrize complex functions with many layers of transformation. In a typical architecture (MLP)

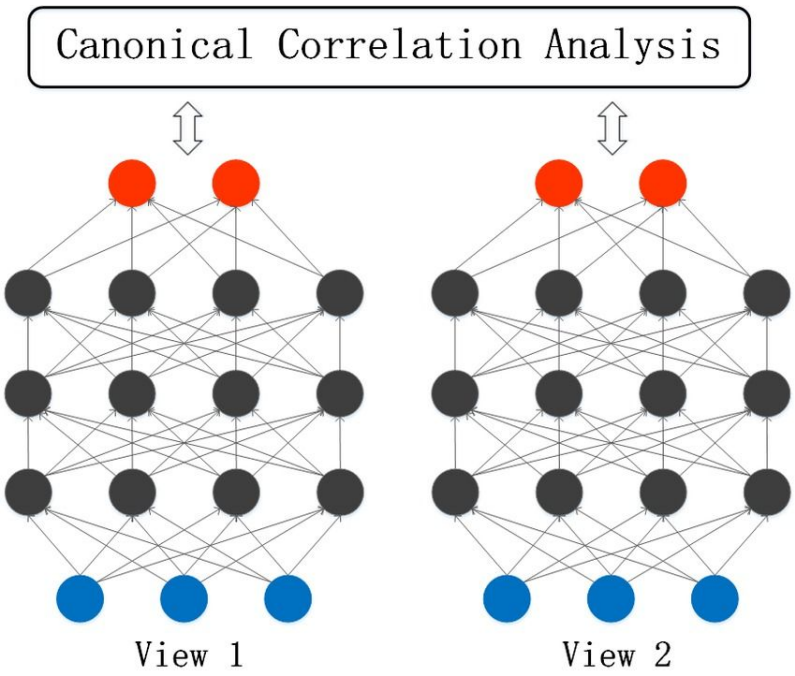
$$h_1 = \sigma(W_1' x + b_1),$$
$$h_2 = \sigma(W_2' h_1 + b_2)$$

etc. σ is a non-linear function (e.g., logistic sigmoid) applied componentwise Each layer detects higher-level features—well suited for tasks like vision, speech processing.
Each layer detects higher-level features—well suited for tasks like vision, speech processing.

How does Deep CCA work?

Deep CCA computes representations of the two views by passing them through multiple stacked layers of nonlinear transformation. The architecture of (Deep − CCA) encourages the network to seek an effective representation of data.

Deep canonical correlation analysis (DCCA), first extracts nonlinear features through deep neural networks (DNNs), and then uses linear CCA to calculate the canonical matrices.



A schematic of deep CCA, consisting of two deep networks learned so that the output layers (topmost layer of each network) are maximally correlated. Blue nodes correspond to input features (n1 = n2 = 3), grey nodes are hidden units (c1 = c2 = 4), and the output layer is red (o = 2). Both networks have d = 4 layers.

Algorithm for DCCA -

To fine-tune all parameters via backpropagation, we need to compute the gradient

$$\partial \text{corr}(H_1, H_2)/\partial H_1.$$

Let $\Sigma_{11}, \Sigma_{22}, \Sigma_{12}$, and $T = \Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1/2} = UDV'$. Then,

$$\frac{\partial \text{corr}(H_1, H_2)}{\partial H_1} = \frac{1}{m-1}\left(\nabla_{12}(H_2 - \bar{H}_2) - \nabla_{11}(H_1 - \bar{H}_1)\right)$$

where

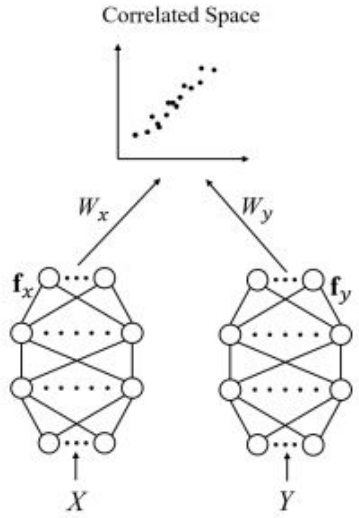
$$\nabla_{12} = \Sigma_{11}^{-1/2} UV' \Sigma_{22}^{-1/2}$$

and

$$\nabla_{11} = \Sigma_{11}^{-1/2} UDU' \Sigma_{11}^{-1/2}.$$

We compare the total correlation of the top k components of each model, for all k ≤ o (DCCA output size).

Deep canonical correlation analysis (DCCA), first extracts nonlinear features through deep neural networks (DNNs), and then uses linear CCA to calculate the canonical matrices.



. Two deep neural networks fx and fy first extract nonlinear features from X and Y , respectively, and then a linear CCA is applied for correlation analysis.

Let fx and fy be two DNNs, and Hx = fx(X) and Hy = fy(Y) be their outputs, respectively. The total correlation of the K canonical variables equals the sum of the first K singular values of matrix T = $\hat{\Sigma}_{xx}^{-1/2}\Sigma_{xy}\hat{\Sigma}_{yy}^{-1/2}$.

$$\hat{T} = \left(\frac{1}{N}H_x H_x^T + r_x I\right)^{-\frac{1}{2}} \left(\frac{1}{N}H_x H_y^T\right)\left(\frac{1}{N}H_y H_y^T + r_y I\right)^{-\frac{1}{2}}.$$

The objective function of DCCA is - (where σk(Tˆ) denotes the k-th largest singular value of Tˆ)

$$\max_{\mathbf{f}_x, \mathbf{f}_y, \mathbf{w}_x, \mathbf{w}_y} \sum_{k=1}^{K} \sigma_k(\hat{T})$$
$$s.t.\ \mathbf{w}_x\left(\frac{1}{N}H_x H_x^T + r_x I\right)\mathbf{w}_x^T = 1,$$
$$\mathbf{w}_y\left(\frac{1}{N}H_y H_y^T + r_y I\right)\mathbf{w}_y^T = 1,$$

Now to increase performance with 10 or more layers various methods are used, some of the most prominent ones include Contrastive divergence and Autoencoders.
Taking a look at one such method of enhancing the output volume in deep learning is **Deep CCA Autoencoders.**

Deep CCA has many **advantages over CCA and KCCA -**

1. May be better suited for natural, real-world data such as vision or audio, compared to standard kernels.
2. The training set can be discarded once parameters have been learned.
3. Computation of test representations is fast.
4. Does not require computing inner products.

Discriminative deep canonical correlation analysis (disDCCA) -

Similar to DCCA, discriminative deep canonical correlation analysis (DisDCCA) is a DNN-based extension of discriminative CCA. Discriminative deep canonical correlation analysis (DisDCCA) simultaneously learns two deep mapping networks of the two sets to maximize the within-class correlation and minimize the inter-class correlation.

DisDCCA Algorithm -

There are two major differences in its implementation compared to DCCA.

In DCCA, we saw that when we let fx and fy be two DNNs, and Hx = fx(X) and Hy = fy(Y) be their outputs, respectively. The total correlation of the K canonical variables equals the sum of the first K singular values of matrix T = $\hat{\Sigma}_{xx}^{-1/2}\Sigma_{xy}\hat{\Sigma}_{yy}^{-1/2}$.

$$\hat{T} = \left(\frac{1}{N}H_x H_x^T + r_x I\right)^{-\frac{1}{2}} \left(\frac{1}{N}H_x H_y^T\right)\left(\frac{1}{N}H_y H_y^T + r_y I\right)^{-\frac{1}{2}}.$$

Similarly in DisDCCA, first when calculating the loss of each batch, the instances are first rearranged according to their classes, and then $\frac{1}{N}H_x H_y^T$ is replaced by $\frac{1}{N}H_x A H_y^T$, in this formula and then, after the model is trained, DisCCA is used to obtain the canonical matrices Wx and Wy.

Deep CCA Autoencoders and Discriminative DCCAE(disDCCAE) -

Deep canonically correlated auto-encoders (DCCAE) was introduced by Wang et al., which as a non-linear model was used to optimize the combination of nonlinear correlations between reconstruction error of general auto-encoders and the learned representations of neural networks.

Deep canonically correlated auto-encoders (DCCAE) improves DCCA and similarly Discriminative DCCAE improves DisDCCA, by using auto-encoders to make sure the information captured by fx and fy can also accurately reconstruct the original X and Y . This figure below shows the DCCAE model structure, where gx and gy represent the decoder networks for reconstructing X and Y , respectively.

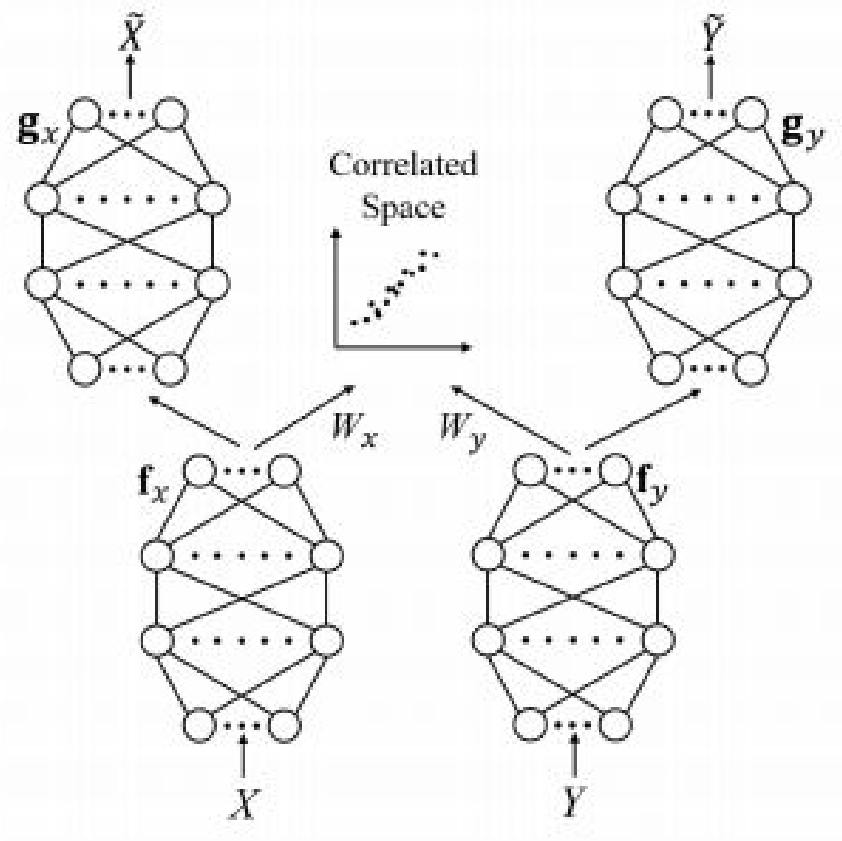


Fig. 3. Schematic diagram of DCCAE. Two deep neural networks $\mathbf{f}_x$ and $\mathbf{f}_y$ first extract nonlinear features from X and Y, respectively, and then a linear CCA is applied for correlation analysis. $\mathbf{g}_x$ and $\mathbf{g}_y$ are two decoders for reconstructing View X and View Y, respectively.

DCCAE Algorithm -

Let fx and fy be two DNNs, and Hx = fx(X) and Hy = fy(Y) be their outputs, respectively. DCCAE and DisDCCAE make sure that the information captured by fx and fy can also accurately reconstruct the original X and Y, where gx and gy represent the decoder networks for reconstructing X and Y , respectively.

Now, two reconstruction errors are incorporated into the objective function of DCCA i.e.

$$\begin{aligned}\max_{\mathbf{f}_x,\mathbf{f}_y,\mathbf{w}_x,\mathbf{w}_y} & \sum_{k=1}^{K}\sigma_k(\hat{T}) \\ s.t.\ \mathbf{w}_x & \left(\frac{1}{N}H_xH_x^T+r_xI\right)\mathbf{w}_x^T=1, \\ \mathbf{w}_y & \left(\frac{1}{N}H_yH_y^T+r_yI\right)\mathbf{w}_y^T=1,\end{aligned}$$

(where σk(Tˆ) denotes the k-th largest singular value of Tˆ)

And we get -

$$\begin{aligned}\max_{\mathbf{f}_x,\mathbf{f}_y,\mathbf{g}_x,\mathbf{g}_y,\mathbf{w}_x,\mathbf{w}_y} & \sum_{k=1}^{K}\sigma_k\left(\hat{T}\right)-\frac{\lambda}{N}\left(\left\|X-\mathbf{g}_x\left(\mathbf{f}_x\left(X\right)\right)\right\|_F^2\right. \\ & \left.+\left\|Y-\mathbf{g}_y\left(\mathbf{f}_y\left(Y\right)\right)\right\|_F^2\right) \\ s.t.\ \mathbf{w}_x & (\frac{1}{N}H_xH_x^T+r_xI)\mathbf{w}_x^T=1, \\ \mathbf{w}_y & (\frac{1}{N}H_yH_y^T+r_yI)\mathbf{w}_y^T=1.\end{aligned}$$

DisDCCAE Algorithm -

Let fx and fy be two DNNs, and Hx = fx(X) and Hy = fy(Y) be their outputs, respectively. From DCCA, we know that the total correlation of the K canonical variables equals to the sum of the first K singular values of matrix T = $\hat{\Sigma}_{xx}^{-1/2}\Sigma_{xy}\hat{\Sigma}_{yy}^{-1/2}$.

$$\hat{T} = \left(\frac{1}{N}H_x H_x^T + r_x I\right)^{-\frac{1}{2}} \left(\frac{1}{N}H_x H_y^T\right)\left(\frac{1}{N}H_y H_y^T + r_y I\right)^{-\frac{1}{2}}.$$

Here for implementing DisDCCA, we replace $\frac{1}{N}H_x H_y^T$ by $\frac{1}{N}H_x A H_y^T$, and similarly we will implement DisDCCAE by replacing these terms to reconstruct the original X and Y in the formula -

$$\max_{\mathbf{f}_x,\mathbf{f}_y,\mathbf{g}_x,\mathbf{g}_y,\mathbf{w}_x,\mathbf{w}_y} \sum_{k=1}^{K}\sigma_k\left(\hat{T}\right) - \frac{\lambda}{N}\left(\|X-\mathbf{g}_x(\mathbf{f}_x(X))\|_F^2 + \|Y-\mathbf{g}_y(\mathbf{f}_y(Y))\|_F^2\right)$$
$$s.t.\ \mathbf{w}_x\left(\frac{1}{N}H_x H_x^T + r_x I\right)\mathbf{w}_x^T = 1,$$
$$\mathbf{w}_y\left(\frac{1}{N}H_y H_y^T + r_y I\right)\mathbf{w}_y^T = 1.$$

Advantage of DCCAE and disDCCAE-

1. These auto-encoders can alleviate the overfitting of the model.
2. It analyzes limitations of traditional CCA which cannot obtain complex nonlinear relations.

Learning outcomes -

1. Data can be viewed from the standpoint of more than one view.

2. Understanding of mathematical concepts related to Multi-view Learning.

3. We looked at the Mathematical Components of Canonical Correlation Analysis to understand Deep and Discriminative Analysis concepts.

4. We learned how Concepts of DCCA, DisDCCAE, DisDCCA are used in better transformation, conceptualization and visualization of complex multiview data.

5. We learned that Deep Canonical Correlation Analysis (DCCA) is a method to learn complex nonlinear transformations of two views of data such that the resulting representations are highly linearly correlated.

6. We learned that Discriminative deep canonical correlation analysis (DisDCCA) simultaneously learns two deep mapping networks of the two sets to maximize the intra-class correlation and minimize the inter-class correlation.

References -

[1] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, “Deep canonical correlation analysis,” in Proc. 30th Int’l Conf. on Machine Learning, Atlanta, GA, Jun. 2013, pp. 1247–1255.

[2] W. Wang, R. Arora, K. Livescu, and J. Bilmes, “On deep multi-view representation learning,” in Proc. 32th Int’l Conf. on Machine Learning, Lille, France, Jul. 2015, pp. 1083–1092.

[3] Chenfeng Guo and Dongrui Wu, “Canonical Correlation Analysis (CCA) Based Multi-View Learning: An Overview,” in arXiv:1907.0169vl [cs.LG] 3 Jul 2019.

[4] C. Xu, D. Tao, and C. Xu, “A survey on multi-view learning,” CoRR, vol. abs/1304.5634, 2013. [Online]. Available: http://arxiv.org/abs/1304.5634