Neurocomputing 214 (2016) 618–628

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Deep canonical correlation analysis with progressive and hypergraph learning for cross-modal retrieval

Jie Shao [a,*], Leiquan Wang [a,c], Zhicheng Zhao [a,b], Fei su [a,b], Anni Cai [a]

[a] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China
[b] Beijing Key Laboratory of Network System and Network Culture, Beijing, China
[c] College of computer and communication engineering, China University of Petroleum,Qingdao, China

ARTICLE INFO

Article history:
Received 4 December 2015
Received in revised form
17 June 2016
Accepted 18 June 2016
Communicated by X. Gao
Available online 23 June 2016

Keywords:
Progressive
Semantic
Hypergraph
Search-based

ABSTRACT

This paper deals with the problem of modeling Internet images and associated texts for cross-modal retrieval such as text-to-image retrieval and image-to-text retrieval. We start with deep canonical correlation analysis (DCCA), a deep approach for mapping text and image pairs into a common latent space. We first propose a novel progressive framework and embed DCCA in it. In our progressive framework, a linear projection loss layer is inserted before the nonlinear hidden layers of a deep network. The training of linear projection and the training of nonlinear layers are combined to ensure that the linear projection is well matched with the nonlinear processing stages and good representations of the input raw data are learned at the output of the network. Then we introduce a hypergraph semantic embedding (HSE) method, which extracts latent semantics from texts, into DCCA to regularize the latent space learned by image view and text view. In addition, a search-based similarity measure is proposed to score relevance of image-text pairs. Based on the above ideas, we propose a model, called DCCA-PHS, for cross-modal retrieval. Experiments on three publicly available data sets show that DCCA-PHS is effective and efficient, and achieves state-of-the-art performance for unsupervised scenario.

1. Introduction

Over the last decade there has been a massive explosion of multimedia content on the web. More and more people upload pictures tagged with texts to the Internet. Articles describing news and technologies include a lot of pictures and texts. The presence of massive multimodal data on the Internet brings a growing demand for cross-modal retrieval, such as using a text query to search for images, and using an image query to search for texts. The goal of this work is to model the correlation of images and associated texts.

Deep learning models have achieved great success in representation learning recently. Deep models use a cascade of multiple layers of nonlinear processing units for feature extraction and possess significantly greater representation power than traditional shallow models. However in some applications, such as in cross-modal retrieval, it might be difficult to obtain enough data to train good deep models. To alleviate this problem, we propose a novel progressive framework, in which a linear projection loss layer is added after the input layer of a deep network. We expect that the linear layer can extract a rough (although not optimal) representation since linear transformations that map high dimensional data into a manifold of much lower dimension have long been used in representation learning [1], and then the following non-linear layers can progressively lead to more abstract and more useful representations of the complex input data with relatively less training samples. The difference between our progressive framework and traditional framework is shown in Fig. 1. We will embed DCCA [2], a method for deep canonical correlation analysis, in the proposed framework to construct an effective and efficient model for cross-modal retrieval.

Practical models for cross-modal retrieval tasks should meet two requirements. First, the top one result retrieved should be accurate, which is a big challenge given that image features are noisy and text features are ambiguous. Second, the top n results retrieved should be ranked according to their relevance to the query. In other words, image and text pairs which are coherent in semantics should be close to each other in the latent space for cross-modal retrieval. CCA (Fig. 2(a)) which only maximizes the correlation between image and its corresponding text does not meet the second requirement. We employ hypergraph learning to extract semantic information from text features to regularize our cross-modal retrieval model. So relevant pairs are clustered in the latent space according to the semantic information (Fig.2(b)).

[*] Corresponding author.
E-mail addresses: shaojielyg@163.com (J. Shao), richiewlq@gmail.com (L. Wang), zhaozc@bupt.edu.cn (Z. Zhao), sufei@bupt.edu.cn (F. su), annicai@bupt.edu.cn (A. Cai).

http://dx.doi.org/10.1016/j.neucom.2016.06.047
0925-2312/© 2016 Elsevier B.V. All rights reserved.

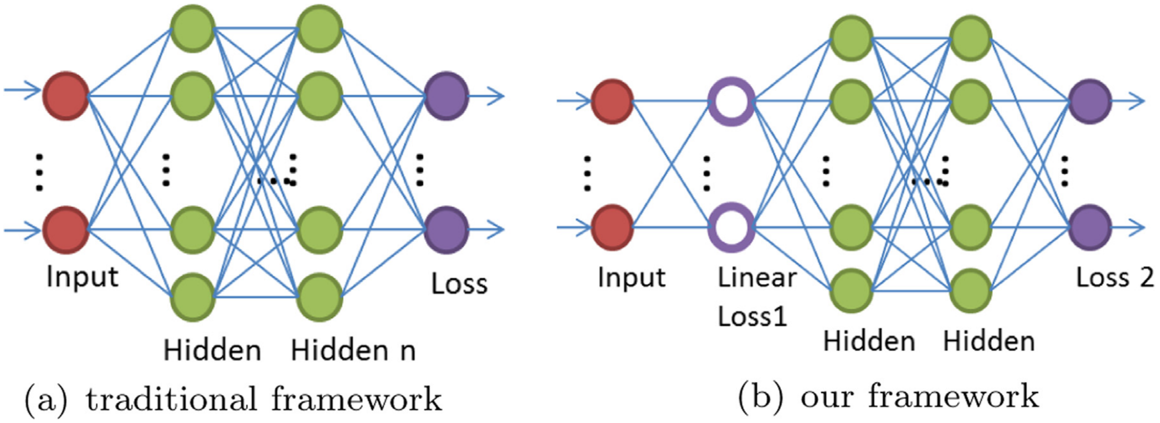


Fig. 1. Structure of traditional framework and our framework. Neurons depicted as empty circles are activated by linear functions. The rest of neurons are activated by non-linear functions.

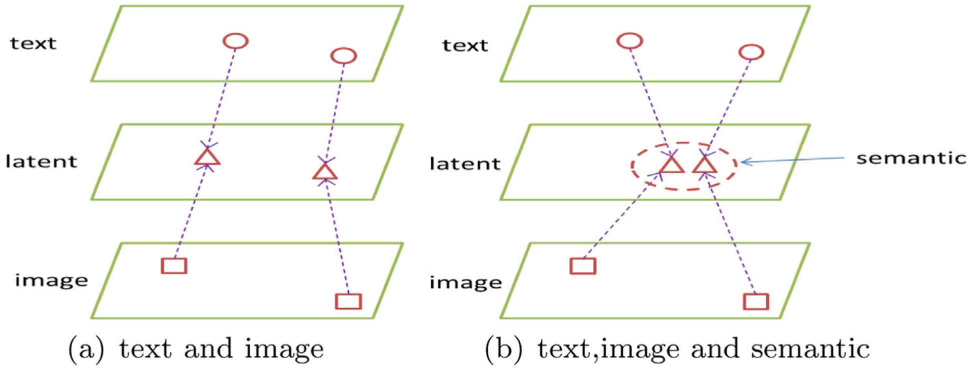


Fig. 2. (a) Paired points in text and image spaces are mapped into a common latent space independently in CCA. (b) Mapped points from relevant pairs are clustered in the latent space according to their semantic information in CCA with hypergraph regularizer.

The main contributions of this work are four-folds:

(1) We propose a novel progressive framework for deep neural networks. Our framework combines the training of linear projection and the training of nonlinear hidden layers to ensure that good features could be learned. Progressive framework could prevent overfitting without combining representation learning as [3–5].

(2) To cluster relevant image-text pairs, we introduce a hypergraph semantic embedding (HSE) into deep models to regularize the latent space learned by image view and text view. Compared with the replicated softmax rbm for semantic extraction in [3,6], HSE could capture high-order relationships between samples. Inspired by RCM-sampling [7], we propose a fast sampling method to generate a large number of hyperedges.

(3) In the latent space, traditional methods score relevance of image-text pairs directly using certain distance metric. Inspired by PageRank [8], we propose a search-based similarity measure to score relevance indirectly. Experiments show that our search-based similarity measure could improve mean Average Precision (mAP) further.

(4) Our method DCCA-PHS (P for progressive, H for hypergraph, and S for Search-based) achieves state-of-the-art performance on three publicly available datasets for unsupervised scenario.

1.1. Related work

Many approaches have been proposed to develop solutions to cross-modal retrieval tasks. We classify these approaches into two categories: shallow models and deep models.

Shallow models: Grangier et al. [9] propose a passive-aggressive model (PAMIR), which is the first attempt to address the problem of ranking images retrieved by text queries. Rasiwasia et al. [10] propose correlation matching to map the features of images and texts into a common latent space using CCA. Complete introduction and recent extensions about CCA can be found in [11,12]. Based on CCA, various variants [13–18] are proposed to model the multi-model correlations. Gong [19] first expands two-view CCA to three-view CCA by incorporating a third view that captures high-level image semantics, represented either by a single category or multiple non-mutually exclusive concepts. Wu et al. [20] formalize the retrieval task as a ranking problem (Bi-CMSRM) similar to PAMIR and try to learn a common latent space for images and texts as CCA. The latent space embedding of Bi-CMSRM is discriminatively learned by the structural large margin learning for optimization with certain ranking criteria (mean average precision) directly. Other algorithms are also proposed to deal with cross-modal problems, such as partial least square (PLS) [21], Bilinear Model (BLM) [22–24], etc.

Deep models: Srivastava et al. [5] propose to learn a generative model of the joint space of image and text inputs using Deep Belief Network (DBN). DBN consists of multiple stacked restricted Boltzmann machine (RBM [25]). Gaussian RBM [26] and replicated softmax RBM [27] are used to model the real-valued feature vectors for image and the discrete sparse word count vectors for text, respectively. Based on DBN, Feng [3] proposes to learn the latent space of image and text inputs by correspondence autoencoder (Corr-AE). Corr-AE defines a novel optimal objective, which minimizes a linear combination of representation learning errors for each modality and correlation learning error between hidden representations of two modalities. We also notice several deep learning methods [28–30,6] for learning a joint embedding space of image and text very recently. Andrew et al. [2] build a deep architecture (DCCA) to learn complex nonlinear transformations of two views of data such that the resulting representations are highly linearly correlated. DCCA can be viewed as a nonlinear extension of traditional linear CCA. Though the representation power improves, DCCA is easy to over-fitting, especially when the datasets are not big enough. In addition, the correlation of relevant pairs in the latent space is considered neither by DCCA nor by CCA. We try to address the shortcomings of DCCA in our DCCA-PHS.

2. Background

2.1. Hypergraph learning

This section introduces hypergraph learning theory. In a simple graph, vertices are used to represent samples, and an edge connects two related vertices (Fig. 3, left). Many graph-based learning methods build an undirected graph based on pairwise distance. However, simple graphs fail to capture high-order relationships between vertices. Unlike edges in a simple graph, a hyperedge in a hypergraph connects more than two vertices (Fig. 3, middle). So hypergraphs could describe high-order relationships between vertices. For convenience, Table 1 lists notations of hypergraph used in this paper.

Let V denote a finite set of samples, and E be a family of subsets e of V such that $\cup_{e\in E}=V$. Each hyperedge e is assigned a positive number $\omega(e)$ as the weight of e. Then we call $G=(V,E,\omega)$ a weighted hypergraph with vertex set V and hyperedge set E. A hyperedge e is said to be incident with a vertex v when $v\in e$. G can be denoted by an incidence matrix H (Fig. 3, right):

$$H(v,e)=\begin{cases}1 & \text{if } v\in e\\ 0 & \text{otherwise}\end{cases} \tag{1}$$

H can also be probabilistic [31] for a continuous case. For vertex $v\in V$, its degree is defined as:

$$d(v)=\sum_{e\in E}\omega(e)H(v,e) \tag{2}$$

and the degree of a hyperedge is defined as:

$$\delta(e)=\sum_{v\in V}H(v,e). \tag{3}$$

We use D_v and $D_e$to denote the diagonal matrices of vertex degrees and hyperedge degrees, respectively. Let W denote the diagonal matrix of the hyperedge weights.

Hypergraphs can be applied to different machine learning tasks [32], e.g., classification, clustering, and embedding. A general framework for conventional hypergraph learning is defined as:

$$\underset{f}{\operatorname{argmin}}\{\Omega(f)+\lambda{*}Remp(f)\} \tag{4}$$

where f is the function to be optimized, $\Omega(f)$ is a regularizer on the hypergraph, *Remp*(*f*) is an empirical loss for specific application, and λ is a trade-off parameter to balance the empirical loss and hypergraph regularizer. The hypergraph regularizer which corresponds to hypergraph normalized cut is defined as [32]:

$$\Omega(f)=\sum_{e\in E}\sum_{\{u,v\}\subset e}\frac{\omega(e)}{\delta(e)}\left(\frac{f(u)}{\sqrt{d(u)}}-\frac{f(v)}{\sqrt{d(v)}}\right)^2=2f^T\Delta f. \tag{5}$$

let $\Theta=D_v^{-\frac{1}{2}}HWD_e^{-1}H^TD_v^{-\frac{1}{2}}$ and $\Delta=I-\Theta$, where I denotes the identity matrix. Δ which is positive semi-definite is called the hypergraph Laplacian. The hypergraph Laplacian can also be built by clique expansion [33], star expansion [33], and Rodriquez Laplacian [34].

Table 1
Notions and their descriptions in hypergraph.

Notion	Description
$G=(V,E)$	A hypergraph G with vertex set V, hyperedge set E
$\omega(e)$	The weight of hyperedge e.p
H	The incidence matrix of hypergraph. H describes the relationship between each vertex and each hyperedge.
$d(v)$	The degree of vertex v
$\delta(e)$	The degree of hyperedge e
W	The diagonal matrix of the hyperedge weights
D_v	The diagonal matrix of vertex degrees
D_e	The diagonal matrix of hyperedge degrees
Δ	The hypergraph Laplacian defined in graph theory is a matrix representation of a hypergraph. It can be used to find many properties of the hypergraph.

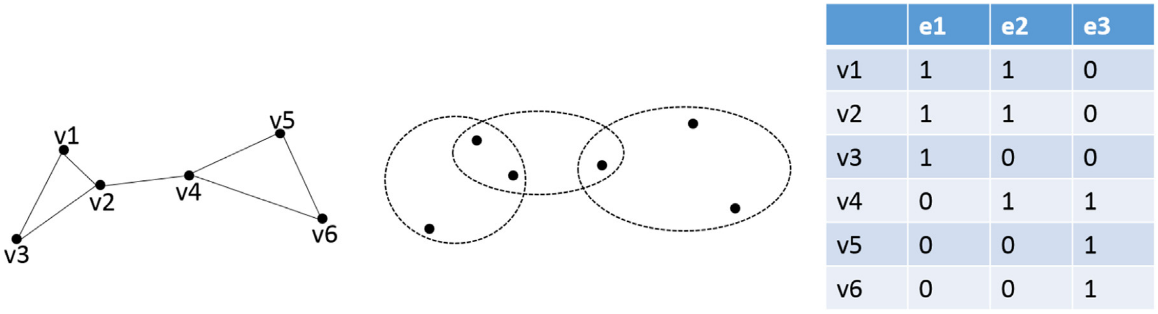
3. Progressive framework

In this section, we first analyze the differences between our progressive framework and traditional framework. Then we describe the advantages of our framework.

Traditional neural networks (Fig. 1(a)) build deep architectures based on end-to-end principle. The first layer corresponds to the input of the network. Each subsequent layer has a connection from the previous layer. The loss in the last layer is minimized to obtain the optimal parameters of the network. The proposed framework differs with the traditional one on three aspects: (1) *Architecture*: A linear projection loss layer is added into the deep architecture after the input layer (Fig. 1(b)). Neurons in the new layer are activated by linear functions. (2) *Loss function*: The optimization objective for network training now becomes a weighted sum of linear projection loss of the first loss layer, the loss of the second loss layer and a regularization penalty, as shown in the following equation:

$$Loss=\lambda_1*L^{loss1}+\lambda_2*L^{loss2}+\lambda_3*\parallel\Theta\parallel_{reg}. \tag{6}$$

(3) *Gradient computation*: Accordingly, during the back-propagate phase, δ^{loss1}, which represents the error term for neurons in the



	e1	e2	e3
v1	1	1	0
v2	1	1	0
v3	1	0	0
v4	0	1	1
v5	0	0	1
v6	0	0	1

Fig. 3. Left: a simple graph of six vertices. Middle: a hypergraph with 3 hyperedges, each hyperedge connects 3 vertices. Right: incidence matrix of the middle hypergraph.

first loss layer, will be the sum of error back-propagated from the last output layer and error produced from its own output:

$$\delta^{loss1} = \frac{\lambda_2 * \partial L^{loss2}}{\partial a^{loss1}} + \frac{\lambda_1 * \partial L^{loss1}}{\partial a^{loss1}} \quad (7)$$

where a^{loss1} is the output of the first loss layer. The former error term can be calculated from error δ^{loss2} which is produced from the output of the second loss layer through chain rule:

$$\delta^{loss2} = \frac{\lambda_2 * \partial L^{loss2}}{\partial a^{loss2}} \quad (8)$$

where a^{loss2} is the output of the second loss layer. The ways to calculate $\frac{\partial L^{loss2}}{\partial a^{loss2}}$ and $\frac{\partial L^{loss1}}{\partial a^{loss1}}$ are the same. Given δ^{loss2}, we can easily get δ^{loss1}.

Therefore, we can see this framework from the following point of view. The linear layer performs a standard procedure of extracting features via linear projection and furnishes the extracted features to the upper nonlinear layers. Very importantly, the training of the linear projection and the training of the nonlinear layers are combined to ensure that the linear projection is well matched with the nonlinear processing stages and good representations of the input raw data are learned at the output of the network.

Our framework has the following advantages: (1) *Available data*: As a weighted combination of linear methods and non-linear methods, our framework suffers less from insufficient training data. (2) *Overfitting*: Deep learning methods are prone to overfitting because of the added layers of abstraction, which allow them to model rare dependencies in the training data. The linear projection loss applied in our framework can be viewed as a regularization term for the errors back-propagated from the output layer and prevents overfitting to some extent. (3) *Convergence rate*: The linear hidden layer in our framework provides a nice input for the high-level non-linear hidden layers, and consequently the convergence can speed up.

We will construct a model with this framework in Section 5 to perform correlation learning for cross-modal retrieval. However, this framework can also be applied to other applications as long as the linear projection is adapted to the target application.

4. Hypergraph semantic embedding (HSE)

In this section, details of our proposed HSE are described.

4.1. Model

In this section, we introduce a novel two-layer model, called HSE, which can be used to extract low-dimensional latent semantic representations from text features.

We view each text feature as a vertex in the graph model. In this section, we assume that hyperedges connecting relevant vertices have been well constructed. The details of our proposed method for hyperedge construction will be discussed in the next subsection. We partition the hypergraph $G = (V, E)$ built upon text features into several parts using normalized cut, with each part corresponding to some kind of latent semantics.

Denote a k-way partitioning by $(V_1, \ldots, V_k)$, where $V_1 \cup V_2 \cup \ldots \cup V_k = V$ and $V_i \cap V_j = \varnothing$ for all $1 \le i, j \le k$. We may obtain a k-way partitioning by minimizing $c(V_1 \ldots V_k) = \sum_{i=1}^{k} \frac{vol\partial V_i}{vol V_i}$ over all k-way partitions. Let r_i be a n-dimensional vector defined by $r_i(v) = 1$ if $v \in V_i$, and 0 otherwise. Then [32]

$$c(V_1 \ldots V_k) = \sum_{i=1}^{k} \frac{r_i^T (D_v - HWD_e^{-1}H^T) r_i}{r_i^T D_v r_i}. \quad (9)$$

Define $s_i = D_v^{-1/2} r_i$, and $f_i = s_i / \| s_i \|$, where $\|\cdot\|$ denotes the usual Euclidean norm. Thus,

$$c(V_1 \ldots V_k) = \sum_{i=1}^{k} f_i^T \Delta f_i = \text{tr } F^T \Delta F \quad (10)$$

where $F = [f_1 \ldots f_k]$. For hypergraph embedding, the elements of r_i are allowed to take arbitrary continuous values rather than Boolean ones only. F is the output of our HSE.

4.2. Hyperedge construction

Based on the concept of random cluster models (RCM), Purkait [7] proposes a guided sampling strategy called RCM-sampling for generating a large number of hyperedges. In RCM-sampling, a connected auxiliary graph with each vertex connected with its k neighbors is built firstly. k is typically set to 3. Then edges in the auxiliary graph are cut off randomly according to certain rules. So the auxiliary graph becomes unconnected, and each connected subgraph corresponds to a subcluster. After the removal of subclusters of size less than the predefined degree of hyperedge, hyperedges are then formed by sampling the vertices of each subgraph randomly. The above process (cutting off edges of the connected auxiliary graph and sampling vertices of the subgraphs) is repeated for several times to get the desired number of hyperedges. However RCM-sampling is time consuming and could not scale well to large date. It takes 24 h to get 2000 20-degree hyperedges in our experiment.

Motivated by RCM-sampling, we propose a fast sampling method to generate a large number of hyperedges. We first perform k-means on the centered training data using cosine distance to obtain the subclusters and subclusters of size less than D (hyperedge degree) are removed. Then a vertex and its D-1 neighbors in one subcluster forms a hyperedge. The incidence matrix H of our hypergraph is probabilistic:

$$H(v_i, e_j) = \begin{cases} \text{cosine}(v_i, v_j) & \text{if } v_i \in e_j \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

5. DCCA-PHS

The architecture of our DCCA-PHS is shown in Fig. 4. DCCA and HSE are well embedded in our progressive framework. DCCA-PHS contains three branches and the right branch learns semantics from texts using HSE. The middle and left ones both contain one input layer, several hidden layers and one linear loss layer (loss 1) and one non-linear loss layer (loss 2), responsible for learning

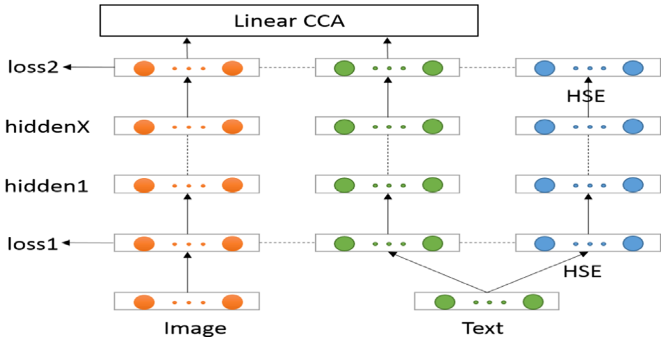


Fig. 4. Architecture of our DCCA-PHS.

Table 2
The numbers of layers and numbers of nodes per layer in DCCA-PHS. For hidden layers, the numbers of nodes in different branches are different (I for image, T for text and S for semantic). Both loss layers have the same number of nodes for all branches.

Dataset	Layers	Input	Hidden(I,T,S)	Loss
Wikipedia	3	128	36, 9, 9	9
Pascal	3	128	72,18, 9	18
Nus-wide-10k	4	256	36, 9, 9	9

latent representations of image and text, respectively. DCCA, as a two-view method, tries to find a transformation to maximize the correlation of text and image. In DCCA-PHS, the semantics extracted from text is viewed as the 3rd view and we try to find a transformation to maximize the correlation of text, image and semantics at the same time. The linear CCA layer attached on the tops of the left and middle deep subnets in Fig.4 is employed for performing canonical correlation analysis of the hidden representations of image and text views, as like in DCCA.

5.1. Loss function and gradient

Let I_j, T_j and S_j, respectively, denote the latent representations for image, text and semantics in the jth loss layer. To learn S_j, the loss function minimized by spectral hypergraph learning is defined as [32]:

$$L_{hypergraph} = \mathrm{tr}(S_j^T \Delta S_j). \tag{12}$$

Traditional two-view CCA tries to project image and text into a k-dimensional latent space such that the correlation between image and text could be maximized. The loss function for two-view CCA is given by:

$$L_{cca}(I_j, T_j) = k - corr(I_j, T_j), \tag{13}$$

k is the upper bound of the correlation between two matrices I_j and T_j. In DCCA-PHS, we try to maximize the correlation among the three views of image, text and semantics. The loss function for a three-view CCA model includes the correlation loss of every two-view:

$$L_{cca}(I_j, T_j, S_j) = L_{cca}(I_j, T_j) + L_{cca}(T_j, S_j) + L_{cca}(S_j, I_j). \tag{14}$$

Thus the whole loss function of DCCA-PHS, which includes the linear projection loss in the first loss layer, the loss in the second loss layer and the regularization penalty to overcome overfitting, is expressed as:

$$\begin{aligned} Loss = {} & \alpha * (L_{cca}(I_1, T_1, S_1) + \beta * L_{hypergraph}(S_1)) + (L_{cca}(I_2, T_2, S_2) \\ & + \beta * L_{hypergraph}(S_2)) + \lambda * \| \Theta \|_F^2 \end{aligned} \tag{15}$$

where α is an inter-layer trade-off between linear projection loss layer and the second loss layer. β is an intra-layer trade-off between correlation learning and hypergraph learning. λ is set to 0.0045.

Table 3
mAP@50 results of CCA-based methods for image-to-text (Img2Txt) and text-to-image (Txt2Img) retrieval.

Methods	Wikipedia		Pascal		NUS-WIDE	
	Img2Txt	Txt2Img	Img2Txt	Txt2Img	Img2Txt	Txt2Img
CCA	0.173	0.179	0.146	0.135	0.286	0.297
PCA-CCA	0.295	0.313	0.217	0.227	0.340	0.341
DCCA	0.235	0.226	0.195	0.192	0.284	0.290
KCCA	0.293	0.262	0.247	0.235	0.333	0.342
DCCA-PHS	**0.341**	**0.379**	**0.292**	**0.284**	**0.395**	**0.408**

Table 4
Architecture difference between DCCA-PHS and others deep learning models.

Methods	Correlation layer	Overcome overfitting	Semantic extraction
Bimodal DBN	Shared layer	None	Replicated softmax rbm
Bimodal AE	Shared layer	Intra-modal Autoencoder	Replicated softmax rbm
Corr-AE	Separate	Inter-modal Autoencoder	Replicated softmax rbm
DCCA	Separate	none	None
DCCA-PHS	Separate	Progressive framework	Hypergraph learning
3view-DCCA	Separate	Intra-modal Autoencoder	Supervised

To optimize Eq. (15), we need to calculate the gradient of $corr(C_i, C_j)$ with respect to C_i and the gradient of $\mathrm{tr}(C_i^T \Delta C_i)$ with respect to C_i. We follow the solution in [2] to get the gradient of $corr(C_i, C_j)$ with respect to C_i (see Appendix A). The gradient for hypergraph embedding is defined as:

$$\frac{\partial\, \mathrm{tr}(C_i^T \Delta C_i)}{\partial C_i} = C_i^T (\Delta + \Delta^T). \tag{16}$$

Compared with conventional eigenvector scheme for hypergraph optimization, our method (Eqs. (12) and (17)) does not need to calculate the matrix inversion which is time consuming. So the training time of our method is much more less than other hypergraph methods.

The whole loss function, Eq. (15), is minimized by using L-BFGS [35] method, which is particularly suitable for optimization of a large number of variables.

6. Search-based similarity

After training, the relevance scores for image-text pairs can be calculated directly using normalized cosine (NC), which achieves the best performance in [10]. However, inspired by PageRank, we propose a search-based similarity measure to indirectly calculate the relevance scores to further improve the performance. No matter whether the query is a text or an image, texts that are relevant to the query are retrieved from the training set. The relevance score in the latent space is then calculated as:

$$sim(Q, I_j/T_j) = \sum_{k=0}^{m} sim(T_k^{train}, I_j/T_j) = \sum_{k=0}^{m} NC(T_k^{train}, I_j/T_j) \tag{17}$$

where $T_0^{train} = Q$ and $T_k{}^{train}$ is the k-th retrieved text in the training set. We only retrieve relevant text features instead of image features because image features are noisy. In our experiment, m is set to 10. This similarity measure will make image-text pairs with high semantic coherence to the query obtain high relevance scores.

7. Experiments and results

7.1. Data sets

Wikipedia: It contains 2866 text-image pairs belonging to 10 semantic categories. We randomly split the data set into three subsets: 2173 pairs for training, 231 pairs for validation and the last 462 pairs for test. Each image is represented by three descriptors, including 1000-D pyramid histogram of dense SIFT, 512-D Gist, and 784-D MPEG-7. And each text is represented by 3000-D

Table 5
mAP@50 results of deep learning methods for image-to-text (Img2Txt) and text-to-image (Txt2Img) retrieval.

Methods	Wikipedia		Pascal		NUS-WIDE	
	Img2Txt	Txt2Img	Img2Txt	Txt2Img	Img2Txt	Txt2Img
Bimodal AE [4]	0.282	0.327	0.250	0.270	0.250	0.297
Bimodal DBN [5]	0.189	0.222	0.219	0.219	0.173	0.203
Corr-AE [3]	0.336	0.368	0.290	0.279	0.331	0.379
DCCA-PHS	**0.341**	**0.379**	**0.292**	**0.284**	**0.395**	**0.408**
3view-DCCA	**0.364**	**0.441**	**0.263**	**0.285**	**0.423**	**0.470**

bag of high-frequency words.

Pascal [36]: It contains 1000 image/text pairs from 20 categories, 50 cases per categories. This data set is a part of Pascal 2008 development kit. Each image is labeled with 5 sentences. We split the data into three subsets, 800 for training, 100 for validation and 100 for testing. The visual features for each image are the same as in Wikipedia. And each text is represented by 1000-D bag of high-frequency words.

Nus-wide-10k: This data set is a subset of NUS-WIDE [37], which contains about 269,648 images with tag annotations from 81 categories. Because some categories are scarce, we only choose 10 most common categories. We have 8000 image-text pairs for training, 1000 for parameter validation, and 1000 for test. For image representation, six types of low-level features are extracted from these images, including 64-D color histogram, 144-D color correlogram, 73-D edge direction histogram, 128-D wavelet texture, 225-D block-wise color moments and 500-D bag of words based on SIFT descriptions. For text representation, we use 1000-D bag of words.

7.2. Evaluation metric

We use mAP as the evaluation criterion. Given one query and top-R retrieved data, the average precision is defined as:

$$\frac{1}{N}\sum_{i=1}^{R} prec(i)*rel(i) \tag{18}$$

where N is the number of the relevant documents in the retrieved set, $prec(i)$ is the percentage of the relevant text documents (images) in the top-i text documents (images). $rel(i)$ is an indicator function, when the i-th result is relevant to the query it equals 1, otherwise 0.

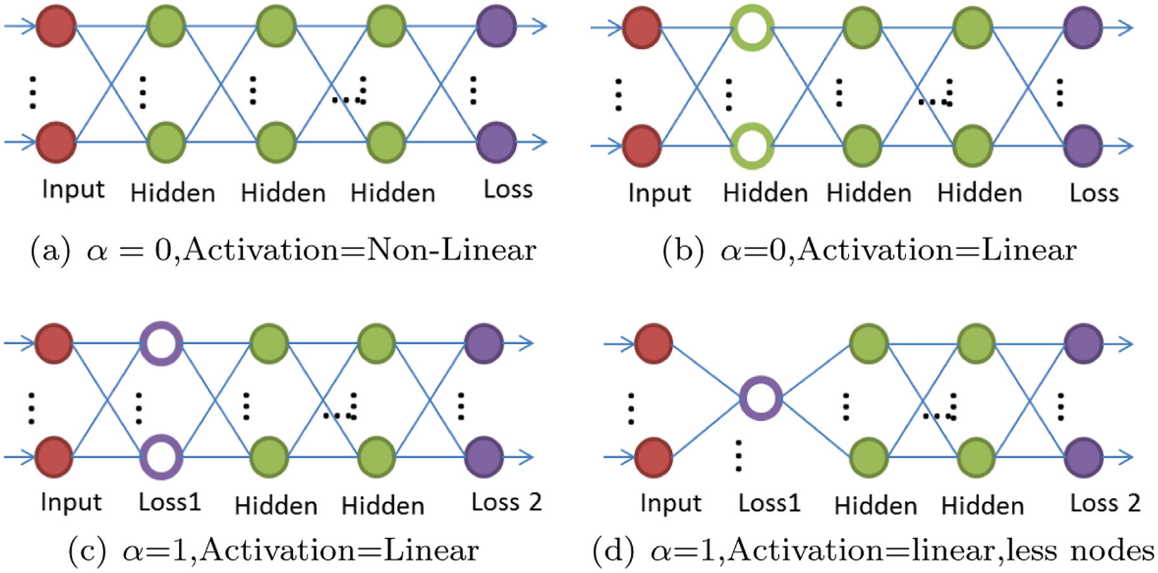
7.3. Results

In Table 2 we show the numbers of layers and numbers of nodes per layer we used in DCCA-PHS. In our experiments, α and β are set to 1 and 3, respectively. The degree of hyperedges is set to 60. For the tolerance of L-BFGS algorithm, we use the default setting of DCCA [2]. The best model evaluated on the validation set is used for testing. Due to the fact that deep learning problems are not convex, we repeat each experiment 5 times and report the averaged result.

7.3.1. On the comparison of CCA-based methods

In this section, we compare our method with traditional CCA-based methods. We implement all the CCA-based methods based on DCCA. CCA, PCA-CCA and KCCA can be viewed as a single-layer DCCA with linear activation function. PCA-CCA conducts dimension reduction on raw features. We choose Gaussian kernel RBF as the kernel function of KCCA. Dimension reduction is also performed in DCCA and DCCA-PHS. To be consistent with the original DCCA, we choose the model resulted from the last iteration, instead of the best model evaluated on validation set, for testing in this experiment.

Table 3 summarizes the mAP@50 scores of CCA-based methods for cross-modal retrieval. PCA-CCA outperforms CCA on all three datasets. Dimension reduction is an effective way to boost the performance of CCA. So we apply dimension reduction on DCCA and DCCA-PHS. The performance of DCCA is worse than PCA-CCA due to overfitting. Compared with KCCA, DCCA can be roughly thought of as learning a kernel for KCCA, but the mapping function is not restricted to live in a reproducing kernel Hilbert space. Our DCCA-PHS performs significantly better than all these CCA-based methods.



(a) $\alpha = 0$,Activation=Non-Linear

(b) α=0,Activation=Linear

(c) α=1,Activation=Linear

(d) α=1,Activation=linear,less nodes

Fig. 5. Structures of our framework with varied α, activation function and number of neurons. Neurons depicted as empty circles are activated by linear functions. The rest of neurons are activated by non-linear functions.

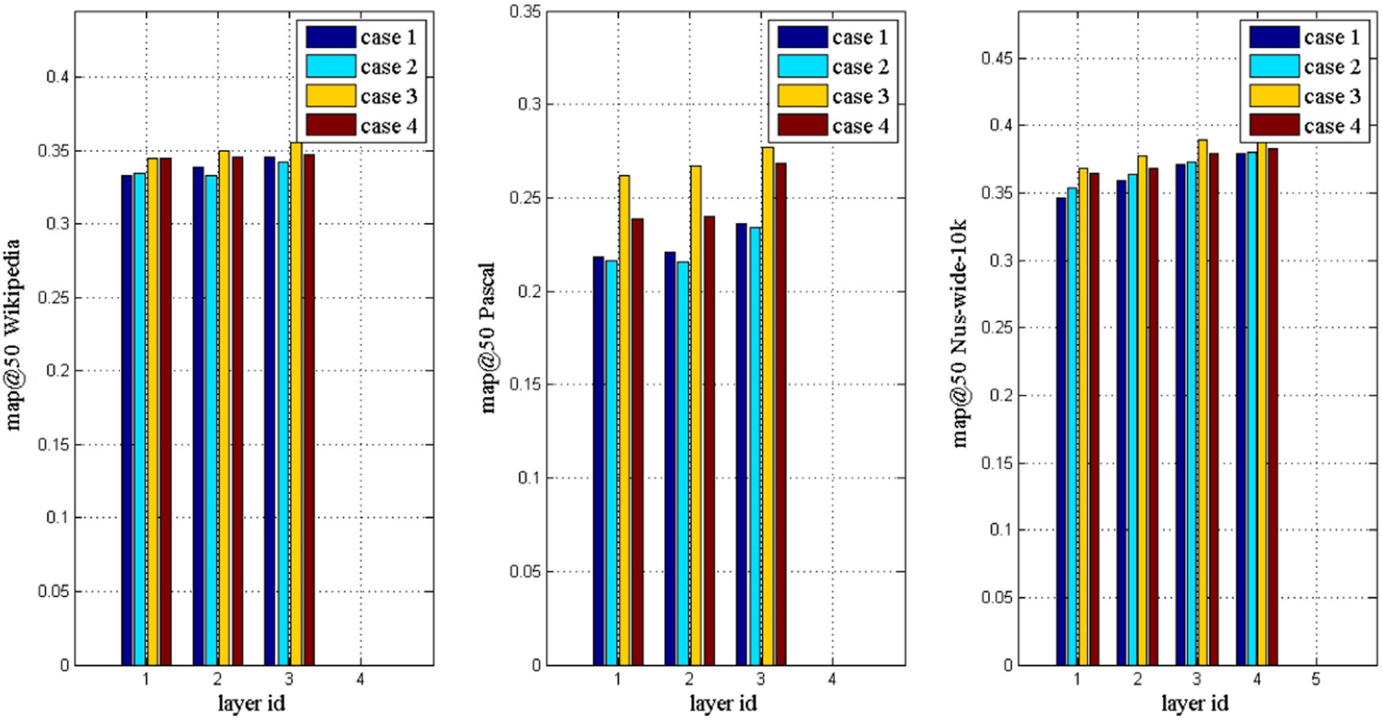


Fig. 6. mAP@50 results for traditional framework and progressive framework.

Table 6
The averaged number of iterations required to reach the tolerance of L-BFGS.

α	Wikipedia	Pascal	Nus-wide-10 k
$\alpha = 0$	224	446.4	459
$\alpha = 1$	195.4	298.8	214.8

7.3.2. On the comparison of deep learning methods

In this section, we compare the performance of our method, DCCA-PHS, with several other deep learning methods for cross-modal retrieval task. Before carrying out the comparison, we first analyze the architecture difference between these methods as follows.

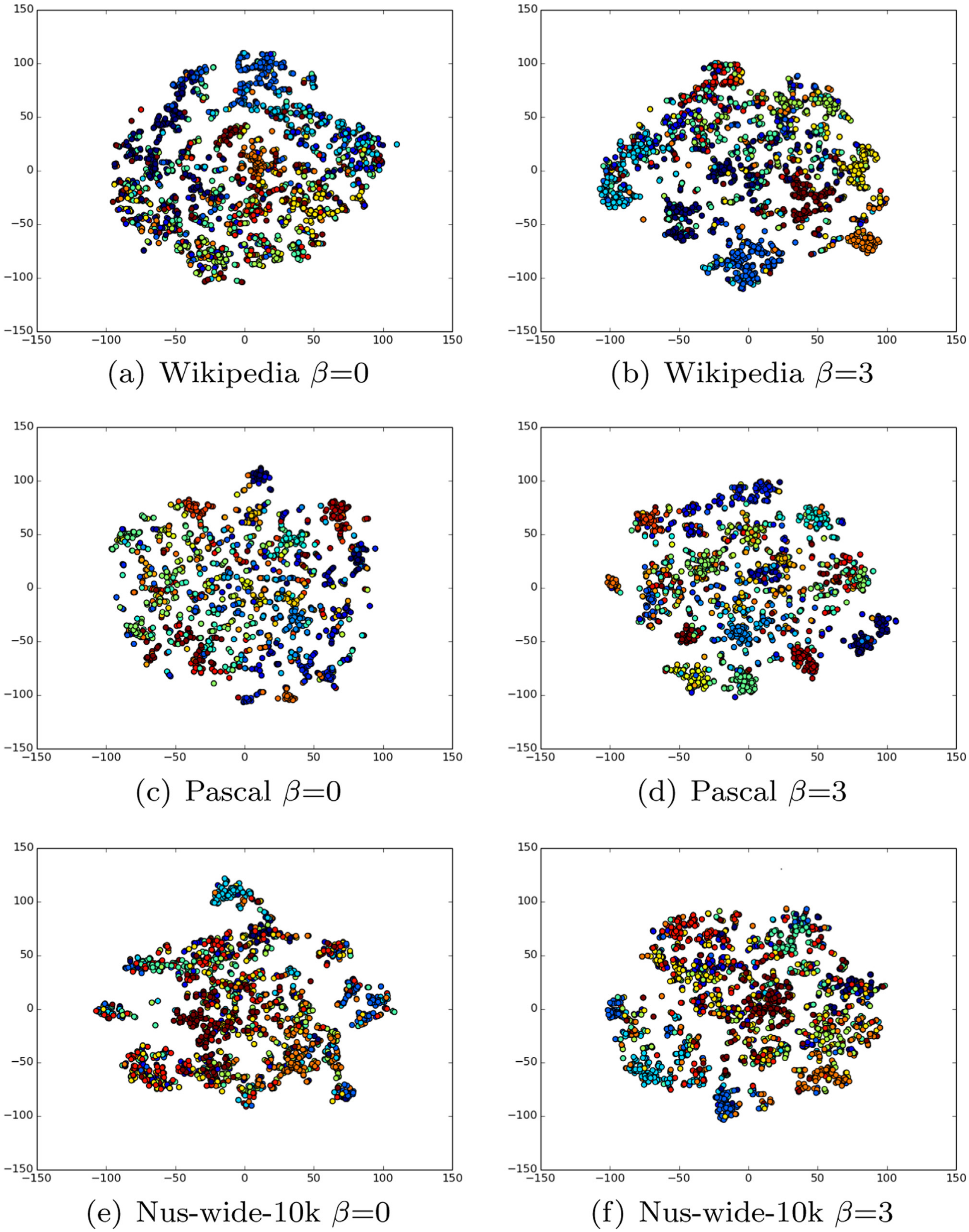
As shown in Table 4, the difference between these methods includes three aspects: (1) *Correlation layer*: Bimodal DBN and Bimodal AE learn a common representation for both modalities in a shared layer. In the absence of one modality, prediction can still be made from another modality using the learned network. Corr-AE and DCCA-based methods build two separate uni-modal subnets and try to learn the correlation between two modalities by correlating hidden representations of two subnets. (2) *Overcome overfitting*: Deep learning methods are prone to overfitting because of the added layers of abstraction, which allow them to model rare dependencies in the training data. Intra-modal and Inter-modal autoencoders are often used to regularize the latent space for both modalities. However, our DCCA-PHS builds a novel progressive framework to cope with overfitting. (3) *Semantic extraction*: All rbm-based methods use replicated softmax rbm to extract latent topics from discrete sparse text features. Our previous method, called 3view-DCCA,[1] directly takes the ground truth labels as semantic information to regularize the common latent space learned by image view and text view. In contrast, DCCA-PHS employs spectral hypergraph embedding for unsupervised semantic extraction.

Table 5 summarizes the mAP@50 scores of several typical deep learning approaches for cross-modal retrieval. We have the following observations: (1) Bimodal AE outperforms Bimodal DBN due to setting up intra-modal autoencoders for representation learning after the shared layer. The way to combine representation learning with correlation learning is helpful for overcoming overfitting. (2) Corr-AE outperforms Bimodal AE mainly by correlating two uni-modal autoencoders instead of using a shared layer. The shared layer acts as a transformation from one coordinate system to another coordinate system, with each coordinate system corresponding to one modality. This is not suitable for correlating noisy image features and ambiguous text features. (3) DCCA-PHS outperforms Corr-AE and achieves state-of-the-art performance on almost all the cases, although both methods have taken overfitting problem and semantic extraction into consideration and learn two separate deep representations, each for one modality. This result demonstrates that the way to cope with overfitting in our progressive framework is more effective compared with autoencoders used in Corr-AE. As for semantic extraction, hypergraph learning in our method can be viewed as an alternative to replicated softmax rbm in real-valued networks. (4) Our previous method, 3view-DCCA, outperforms unsupervised methods in almost all the cases since it is supervised. The performance of DCCA-PHS is the one closest to supervised 3view-DCCA, compared with other unsupervised methods. DCCA-PHS and 3view-DCCA are mutually complementary for cross-modal retrieval.

7.3.3. On the effectiveness of progressive framework

In this subsection we demonstrate the effectiveness of our progressive framework. For convenience and comparison purpose, we do not use the search-based similarity measure to score relevance in this experiment. Two three-layer networks are built, respectively, for Wikipedia and Pascal, and a four-layer network for Nus-wide-10k, since the latter dataset has more training samples. We check the performance of cross-modal retrieval on all layers in four cases: (1) $\alpha=0$ and the first hidden layer is nonlinear (Fig. 5(a)), i.e., a standard traditional network. (2) $\alpha=0$ and

[1] 3view-DCCA has been accepted for oral presentation in VCIP 2015.



(a) Wikipedia β=0

(b) Wikipedia β=3

(c) Pascal β=0

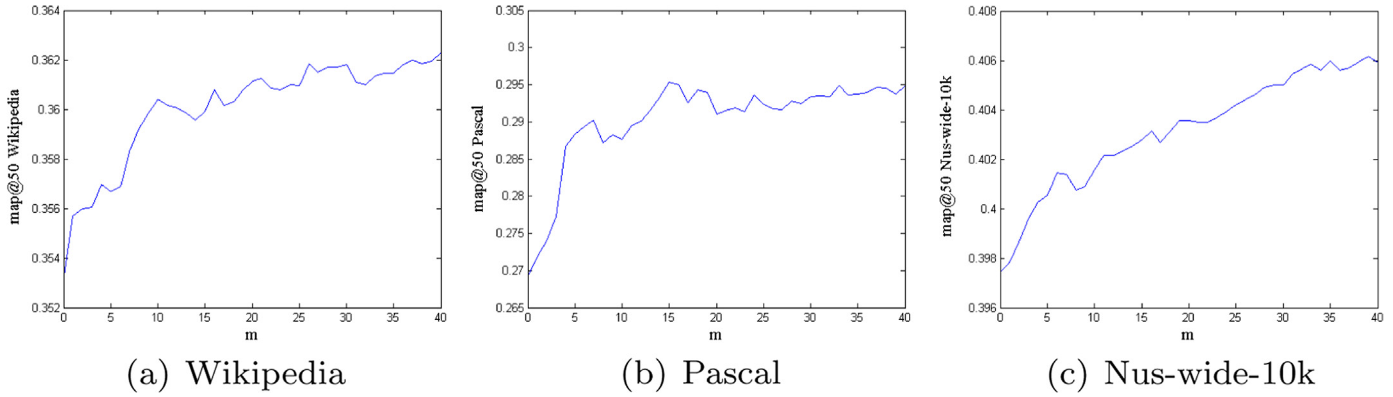
(d) Pascal β=3

(e) Nus-wide-10k β=0

(f) Nus-wide-10k β=3

Fig. 7. Visualizations of the representations in the final layer learned by DCCA-PHS when $\beta=0$ and $\beta=3$. The points in the same color belong to the same category. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

the first hidden layer is linear (Fig. 5(b)). (3) $\alpha=1$ and the first hidden layer is linear (Fig. 5(c)), i.e., the propose method. (4) The setting of α and activation function of the first hidden layer are the same as case 3. We only reduce the number of nodes in the first loss layer (From 9 to 8 for Wikipedia and Nus-wide-10k, and 18 to 16 for Pascal. We do not reduce the number too much because the number of nodes is small). In the above cases, $\alpha=0$ means to ignore the linear loss in network training.



(a) Wikipedia (b) Pascal (c) Nus-wide-10k

Fig. 8. Average mAP of image-to-text retrieval and text-to-image retrieval with varied m on three datasets.

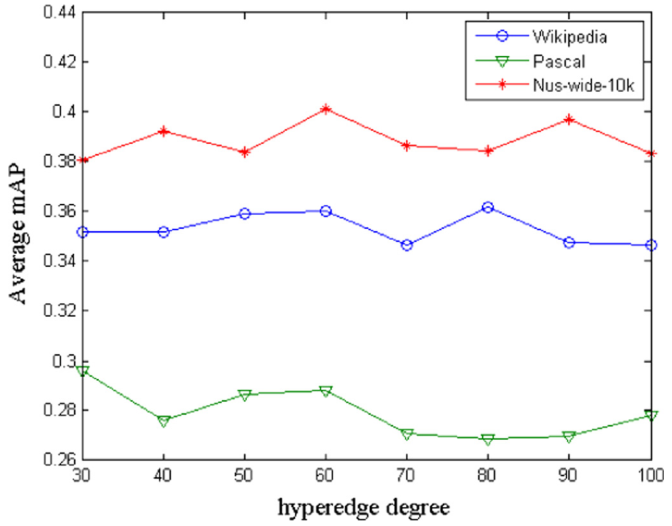


Fig. 9. Average mAP of image-to-text retrieval and text-to-image retrieval with varied hyperedge degree on three datasets.

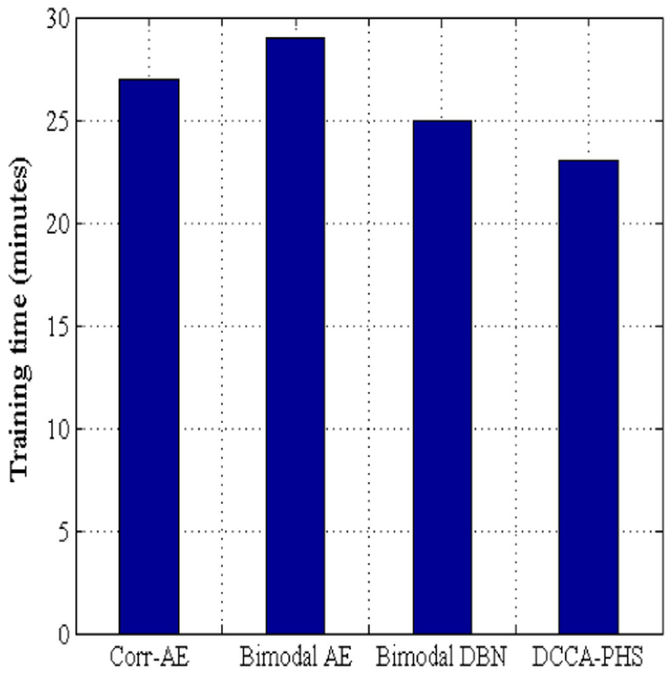


Fig. 10. Training time of different deep learning methods on Nus-wide-10k.

In Fig. 6 we compare the performance of representations learned in all layers of each case. From Fig. 6 we have the following observations: (1) With the same number of network parameters, the performance of our DCCA-PHS (case 3) is significantly better than traditional framework (case 1) at all layers on three datasets. The two loss layers in our framework provide a bi-directional constraint for abstract representations of the middle layers, hence improves the mAP. (2) The performance of the progressive framework degrades to about the same as the traditional one when linear loss is not taken into account in network training (case 2). This fact illustrates the importance of joint optimization of the linear and non-linear loss in our framework. (3) The performance of our framework, even with less parameters (case 4), can still be better than traditional one (cases 1 and 2). (4) When comparing the results on different datasets, we find that the superiority of our framework is shown most obviously on Pascal, which is the one with the least data among the three datasets. This observation proves that our framework is particularly suitable to applications lack of training samples.

As shown in Table 6, the number of iterations required to reach the convergence tolerance of L-BFGS algorithm in training stage is less when α is set to 1. This is because with the help of the first linear loss layer, the upper non-linear hidden layers are fed by linearly projected features, rather than raw input data.

From the results obtained in this subsection, we may expect that our progressive framework could provide a better and faster solution to other problems optimized by deep neural networks if a proper linear transformation is embedded in it.

7.3.4. On the effectiveness of HSE

We use t-SNE [38] to visualize the image and text representations learned by DCCA-PHS in Fig. 7. As discussed above, the similarity metric we use for retrieval is NC which is not suitable for t-SNE. So we apply L2 normalization on features in the latent space. In the normalized space, NC distance is equal to Euclidean distance.

When β is set to 0, the effect of our HSE is ignored. As shown in Fig. 7, when β is set to 3, points with the same color tend to get together. The representation spaces we learn are quite effective for cross-modal retrieval, since a lot of image-text pairs with the same semantic label are clustered. In the latent space, pairs connected by hyperedges are forced to be close to each other under the constraint of hypergraph regularizer. And locally semantic consistent pairs are clustered by partitioning the hypergraph using normalized cut.

HSE is proposed to exploit the high-order relationships among multi-modal pairs to facilitate cross-modal correlation learning. The high-order relationships or semantics can also be constructed by exploiting local manifold structure, context information or semantic labels.

7.3.5. On the effectiveness of search-based similarity

As shown in Fig. 8, we investigate the performance with respect to the number, m, of relevant texts retrieved to score relevance of paired text and image. When m equals 0, Eq. (18) degenerates into NC. With the increase of m, the performance has a rising trend on three datasets. It can be regarded that the search-based similarity measure plays a role similar to transfer learning since the semantic consistent latent space learned from training set is used to evaluate the similarity at the test time. To balance the searching time and mAP scores, m is chosen as 10.

7.3.6. On the number of hyperedge degree

In Fig. 9 we report the mAP scores with varied hyperedge degree. We observe that the degree of hyperedges is relatively insensitive between 30 and 100. Our method DCCA-PHS achieves the best performance around 60.

7.3.7. On the comparison of training time for deep learning methods

Finally in Fig. 10, we report the training time for our model and other deep learning models on Nus-wide-10k data set, since it has the largest number of cases among the three data sets. For Corr-AE, Bimodal AE and Bimodal DBN, experiments are conducted on a computer which has Intel i7 4.0 GHZ 8 processors, 8 GB RAM, 6 GB Nvidia Gefore GTX TITAN GPU, Cuda6.5 and Ubuntu 14.04. Experiments for DCCA-PHS are conducted on a computer which has Intel i7 4.0 GHZ 8 processors, 8 GB RAM, Intel MKL11.2 and Windows 7. We have performed grid search to find the optimal number of hidden nodes and Table 2 shows the results. Trade-off parameters for all methods are fixed during training. Compared with the other deep learning methods, the training of DCCA-PHS includes an extra stage for hypergraph construction. However, DCCA-PHS trains faster than other methods even without the acceleration of GPUs. This is largely due to the help of linear loss layer introduced in our progressive framework and the use of full-batch L-BFGS optimization method.

8. Conclusion

In this work, we propose a progressive framework for deep neural networks. Our framework combines the training of linear projection and the training of nonlinear hidden layers to ensure that good features can be learned. We believe that our framework could provide a better and faster solution to more problems optimized by deep neural networks. We also propose a novel HSE to extract semantics from text features, which can be viewed as an alternative to replicated softmax rbm in real-valued networks. Based on the proposed framework and HSE, we propose a novel deep CCA model for cross-modal retrieval, as well as a search-based similarity measure to further improve its performance. In the future, we would like to develop some semi-supervised methods to make full use of the semantic labels and apply to more datasets.

Acknowledgment

This work was supported by the Chinese National Natural Science Foundation (61471049, 61372169, 61532018), and by the Postgraduate Innovation Fund of SICE, BUPT, 2015.

Appendix A. Derivation of DCCA gradient

Let $\overline{C}_i = C_i - \frac{1}{n}C_i$ be the centered data matrix and define $\hat{\Sigma}_{12} = \frac{1}{n-1}\overline{C}_1\overline{C}_2'$, $\hat{\Sigma}_{11} = \frac{1}{n-1}(\overline{C}_1\overline{C}_1' + r_1 I)$ for regularization constant r_1,and $\hat{\Sigma}_{22} = \frac{1}{n-1}(\overline{C}_2\overline{C}_2' + r_2 I)$ for regularization constant r_2. r_1 and r_2 is set to 41.667 and 59.060, which is the default setting of DCCA. The correlation $corr(C_1, C_2)$ is the sum of the singular values of matrix $T = \hat{\Sigma}_{11}^{-\frac{1}{2}}\hat{\Sigma}_{12}\hat{\Sigma}_{22}^{-\frac{1}{2}}$. If the singular value decomposition of T is $T = UDV'$, then

$$\frac{\partial corr(C_1, C_2)}{\partial C_1} = \frac{1}{n-1}\left(- \hat{\Sigma}_{11}^{-\frac{1}{2}}UDU'\hat{\Sigma}_{11}^{-\frac{1}{2}}\overline{C}_1 + \hat{\Sigma}_{11}^{-\frac{1}{2}}UV'\hat{\Sigma}_{22}^{-\frac{1}{2}}\overline{C}_2\right) \tag{A.1}$$

$\frac{\partial corr(C_1, C_2)}{\partial C_2}$ has a symmetric expression.

References

[1] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.

[2] G. Andrew, R. Arora, J. Bilmes, K. Livescu, Deep canonical correlation analysis, in: Proceedings of the 30th International Conference on Machine Learning, 2013, pp. 1247–1255.

[3] F. Feng, X. Wang, R. Li, Cross-modal retrieval with correspondence autoencoder, in: Proceedings of the ACM International Conference on Multimedia, ACM, 2014, pp. 7–16.

[4] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A.Y. Ng, Multimodal deep learning, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 689–696.

[5] N. Srivastava, R. Salakhutdinov, Learning representations for multimodal data with deep belief nets, in: International Conference on Machine Learning Workshop, 2012.

[6] F. Feng, R. Li, X. Wang, Deep correspondence restricted Boltzmann machine for cross-modal retrieval, Neurocomputing 154 (2015) 50–60.

[7] P. Purkait, T.-J. Chin, H. Ackermann, D. Suter, Clustering with hypergraphs: the case for large hyperedges, in: Computer Vision—ECCV 2014, Springer, 2014, pp. 672–687.

[8] L. Page, S. Brin, R. Motwani, T. Winograd, The Pagerank Citation Ranking: Bringing Order to the Web.

[9] D. Grangier, S. Bengio, A discriminative kernel-based approach to rank images from text queries, IEEE Trans. Pattern Anal. Mach. Intell. 30 (8) (2008) 1371–1384.

[10] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G.R. Lanckriet, R. Levy, N. Vasconcelos, A new approach to cross-modal multimedia retrieval, in: Proceedings of the international conference on Multimedia, ACM, 2010, pp. 251–260.

[11] S. Sun, A survey of multi-view machine learning, Neural Comput. Appl. 23 (7–8) (2013) 2031–2038.

[12] S. Sun, D.R. Hardoon, Active learning with extremely sparse labeled examples, Neurocomputing 73 (16) (2010) 2980–2988.

[13] J. Costa Pereira, E. Coviello, G. Doyle, N. Rasiwasia, G.R. Lanckriet, R. Levy, N. Vasconcelos, On the role of correlation and abstraction in cross-modal multimedia retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 36 (3) (2014) 521–535.

[14] A. Sharma, A. Kumar, H. Daume III, D.W. Jacobs, Generalized multiview analysis: a discriminative latent space, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 2160–2167.

[15] S. Wang, J. Lu, X. Gu, B.A. Weyori, J.Y. Yang, Unsupervised discriminant canonical correlation analysis based on spectral clustering, Neurocomputing 171 (C) (2015) 425–433.

[16] J. Cai, Y. Tang, J. Wang, Kernel canonical correlation analysis via gradient descent, Neurocomputing 182 (2015) 322–331.

[17] C. Zu, D. Zhang, Canonical sparse cross-view correlation analysis, Neurocomputing 191 (2016) 263–272.

[18] S. Wang, F. Zhuang, S. Jiang, Q. Huang, Q. Tian, Cluster-sensitive structured correlation analysis for web cross-modal retrieval, Neurocomputing 168 (2015) 747–760.

[19] Y. Gong, Q. Ke, M. Isard, S. Lazebnik, A multi-view embedding space for modeling internet images, tags, and their semantics, Int. J. Comput. Vis. 106 (2) (2014) 210–233.

[20] F. Wu, X. Lu, Z. Zhang, S. Yan, Y. Rui, Y. Zhuang, Cross-media semantic representation via bi-directional learning to rank, in: Proceedings of the 21st ACM international conference on Multimedia, ACM, 2013, pp. 877–886.

[21] R. Rosipal, N. Krämer, Overview and recent advances in partial least squares, in: Subspace, Latent Structure and Feature Selection, Springer, 2006, pp. 34–51.

[22] J.B. Tenenbaum, W.T. Freeman, Separating style and content with bilinear models, Neural Comput. 12 (6) (2000) 1247–1283.

[23] C. Kang, S. Liao, Y. He, J. Wang, S. Xiang, C. Pan, Cross-modal similarity learning: a low rank bilinear formulation, arXiv preprint arXiv:1411.4738.

[24] T. Yao, X. Kong, H. Fu, Q. Tian, Semantic consistency hashing for cross-modal retrieval, Neurocomputing 193 (C) (2016) 250–259.

[25] P. Smolensky, Information processing in dynamical systems: foundations of harmony theory, in: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, MIT Press, Cambridge, MA, USA, 1986, pp. 194–281.

[26] M. Welling, M. Rosen-Zvi, G.E. Hinton, Exponential family harmoniums with an application to information retrieval, in: Advances in Neural Information Processing Systems, 2004, pp. 1481–1488.
[27] G.E. Hinton, R. Salakhutdinov, Replicated softmax: an undirected topic model, in: Advances in Neural Information Processing Systems, 2009, pp. 1607–1614.
[28] J. Weston, S. Bengio, N. Usunier, Large scale image annotation: learning to rank with joint word-image embeddings, Mach. Learn. 81 (1) (2010) 21–35.
[29] A. Frome, G.S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al., Devise: a deep visual-semantic embedding model, in: Advances in Neural Information Processing Systems, 2013, pp. 2121–2129.
[30] R. Socher, M. Ganjoo, C.D. Manning, A. Ng, Zero-shot learning through cross-modal transfer, in: Advances in Neural Information Processing Systems, 2013, pp. 935–943.
[31] Y. Huang, Q. Liu, S. Zhang, D.N. Metaxas, Image retrieval via probabilistic hypergraph ranking, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 3376–3383.
[32] D. Zhou, J. Huang, B. Schölkopf, Learning with hypergraphs: clustering, classification, and embedding, in: Advances in Neural Information Processing Systems, 2006, pp. 1601–1608.
[33] J.Y. Zien, M.D. Schlag, P.K. Chan, Multilevel spectral hypergraph partitioning with arbitrary vertex sizes, IEEE Trans. Comput.-Aid. Des. Circuits Syst. 18 (9) (1999) 1389–1399.
[34] J. Rodríguez, On the Laplacian spectrum and walk-regular hypergraphs, Linear Multilinear Algebra 51 (3) (2003) 285–297.
[35] J. Nocedal, S.J. Wright, Numerical Optimization, 2nd edition.
[36] A. Farhadi, M. Hejrati, M.A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, D. Forsyth, Every picture tells a story: generating sentences from images, in: Computer Vision—ECCV 2010, Springer, 2010, pp. 15–29.
[37] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y. Zheng, Nus-wide: a real-world web image database from National University of Singapore, in: Proceedings of the ACM International Conference on Image and Video Retrieval, ACM, 2009, p. 48.
[38] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (2579–2605) (2008) 85.

Jie Shao is a Ph.D. candidate in School of Information and Communication Engineering, Beijing University of Posts Telecommunications. His current research interests include cross-modal retrieval and deep learning.

Leiquan Wang is a Ph.D. candidate in School of Information and Communication Engineering, Beijing University of Posts Telecommunications. He is also an Experimenter in College of Computer and Communication Engineering, China University of Petroleum. His current research interests include multimodal fusion, cross modal retrieval and social media analysis.

Zhicheng Zhao now is a Lecturer of BUPT. His research interests are computer vision, image and video semantic understanding and retrieval.

Fei Su is a Professor in the Multimedia Communication and Pattern Recognition lab, School of Information and Telecommunication, Beijing university of Posts and Telecommunications. She received the Ph.D. degree majoring in Communication and Electrical Systems from BUPT in 2000. She was a Visiting Scholar at Electrical Computer Engineering Department, Carnegie Mellon University from 2008 to 2009, Her current interests include pattern recognition, image and video processing and biometrics. She has authored and co-authored more than 70 journal and conference papers and some textbooks.

Anni Cai received the B.S. degree in Radio Engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1965. She received the Ph.D. degree in Electrical and Computer Engineering from University of California, Santa Barbara, CA, USA, in 1988. Since 1965, she has been on the Faculty of School of Telecommunication Engineering, BUPT, where she is presently a Professor. She has co-authored more than 100 papers and four books. Her current interests are in the fields of image and video processing, pattern recognition and multimedia telecommunications.