Task-Driven Common Representation Learning via Bridge Neural Network

Yao Xu,[1,2] **Xueshuang Xiang,**[1,2,*] **Meiyu Huang**[1]
[1]Qian Xuesen Laboratory of Space Technology, China Academy of Space Technology, Beijing, 100190
[2]School of Aerospace Science and Technology, Xidian University, Xian, 710071
{xuyao, xiangxueshuang, huangmeiyu}@qxslab.cn

Abstract

This paper introduces a novel deep learning based method, named bridge neural network (BNN) to dig the potential relationship between two given data sources task by task. The proposed approach employs two convolutional neural networks that project the two data sources into a feature space to learn the desired common representation required by the specific task. The training objective with artificial negative samples is introduced with the ability of mini-batch training and it's asymptotically equivalent to maximizing the total correlation of the two data sources, which is verified by the theoretical analysis. The experiments on the tasks, including pair matching, canonical correlation analysis, transfer learning, and reconstruction demonstrate the state-of-the-art performance of BNN, which may provide new insights into the aspect of common representation learning.

Introduction

In the real world, a potential relationship always exists between two sets of data, which can be either from multi-views of one data source, e.g., two voices of songs, audio and subtitles of movies, or from two different data sources, e.g., faces of couples, objects with the same labels. One idea of mining the potential relationship of given data pairs is learning a common representation of them, which has achieved lots of interests (Ngiam et al. 2011; Eisenschtat and Wolf 2017; Andrew et al. 2013; Chandar et al. 2016). These methods tried to build a universal model motivated by more than one task, including (i) pair matching across views, (ii) canonical correlation analysis (CCA) (Hotelling 1936), (iii) transfer learning and (iv) reconstruction of a missing view. However, different task imposes different complexity levels of common representations, as shown in Figure 1.

To clarify Figure 1, the complexity level of common representations for each task will be described briefly, from simple to complex. (i) For the task of pair matching across views, the common representations are just some potential information which can build a connection between data pairs. This connection can be just a few similarities between pairs, which can be fairly simple. (ii) For the task of CCA, the common representations are the projections of two data sources and it is supposed to maximize the linear correlation between them. CCA methods are always developed as a strategy to build common representations, which makes it an intermediate task rather than a final application objective. (iii) For the task of transfer learning, the common representations need to be features which are useful for a certain learning objective, such as classification. The relationship of those features can be more complex than the linear correlation. (iv) Reconstruction of a missing view requires the common representations to be an encoder, which should contain the information of the views as whole as possible.

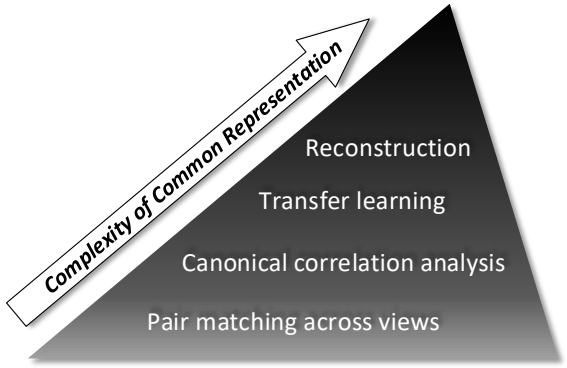

Figure 1: The complexity of common representations varying from different tasks. A task on the top of the triangle is more complex than a bottom one.

Because of the diversity of different tasks, different tasks impose different levels of complexity of the common representations. Motivated by this understanding, we propose a novel deep learning based method, named bridge neural network (BNN) to learn common representations by mining the potential relationship of the specified data sources according to a given task, as shown in Figure 2. Given a task, we first construct the correlated data source, named positive samples and build the negative samples according to the positive samples. Then both positive samples and negative samples are used to train a BNN to handle the given task.

As that shown in Figure 3, BNN employs two convolutional neural networks that project the two data sources into a common feature space and use the Euclidean loss to de-

*Corresponding author.

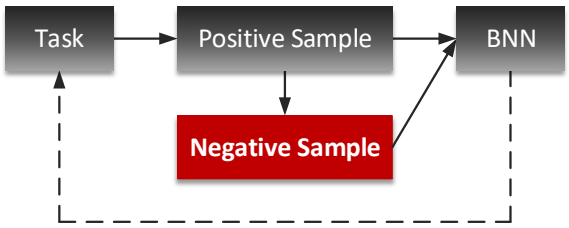

Figure 2: A framework of using Bridge Neural Network (BNN) to handle a task, e.g. as that shown in Figure 1.

termine whether two given data sources have a potential relationship, i.e. positive sample or not, i.e. negative sample. Thus, the problem of mining the potential relationship can be transferred to a binary classification problem by introducing artificial negative samples. The main contributions of the proposed BNN in this paper are:

- First propose a task-driven framework to learn common representations by mining the potential relationship of given data pairs, which is specified task by task;
- A novel optimization problem, i.e. training objective using artificial negative samples is introduced, and it's asymptotically equivalent to maximization of the total correlation;
- BNN with lightweight convolution layers can be trained using Gradient Descent based optimization methods, making it more scalable for dealing with large high dimensional data.

Related work

Canonical Correlation Analysis (CCA) (Hotelling 1936) is often used to build common representations. It is also a general procedure for investigating the relationships between two sets of variables by computing a linear projection for data pairs into a common space which can maximize their linear correlation. It plays a significant role in many fields including biology and neurology (Hardoon et al. 2007), natural language processing (Dhillon, Foster, and Ungar 2011), speech processing (Arora and Livescu 2013) and computer vision tasks, e.g., action recognition (Kim, Wong, and Cipolla 2007), linking text and image (Eisenschtat and Wolf 2017). CCA is also a basic method in multi-view learning, see (Xu, Tao, and Xu 2013; Zhao et al. 2017) for details.

However, traditional CCA method (Hotelling 1936) and its derivatives, such as regularized CCA (Vinod 1976), Nonparametric Canonical Correlation Analysis (NCCA) (Michaeli, Wang, and Livescu 2016), Randomized Canonical Correlation Analysis (RCCA) (Mineiro and Karampatziakis 2014) and Kernel Canonical Correlation Analysis (KCCA) (Akaho 2006; Hardoon, Szedmak, and Shawe-Taylor 2004; Bach and Jordan 2002; Melzer, Reiter, and Bischof 2001), do not scale well with the size of the dataset and the representations. A number of researches were therefore proposed to overcome this drawback. Deep Canonical Correlation Analysis (DCCA) (Andrew et al. 2013) and its improved versions in image and text matching (Yan and Mikolajczyk 2015; Wang, Li, and Lazebnik 2016) are one of those which introduced deep learning method. Based on DCCA, later works such as Correlation Neural Network(CorrNet) (Chandar et al. 2016) and Deep Canonically Correlated Autoencoders (DCCAE) (Wang et al. 2015), brought in Multimodal Autoencoder (MAE) (Ngiam et al. 2011) to extend the task of reconstruction of views. The introducing of MAE also improves the performance of CCA because it aims to capture a more meaningful common representation by adding an optimization objective minimizing the distance between the original input and the decoded output of each view. Along this line, 2-Way Nets (2WayNet) (Eisenschtat and Wolf 2017) gave a different approach by replacing the optimization objective into the Euclidean loss. In summary, these methods achieved the state-of-the-art performance, but still have some problems, such as the scalability, high inference time with fully connected layers and most of these methods were to build a universal model motivated by more than one task.

In order to fix these problems, this paper proposes bridge neural network (BNN) with lightweight convolution layers to learn common representations by mining the potential relationship of the specified data sources according to a given task. The most related work is DCCA (Andrew et al. 2013), 2WayNet (Eisenschtat and Wolf 2017) and Siamese Network (Chopra, Hadsell, and LeCun 2005), which is a similar approach of using two networks to learn the similarity between two inputs. Differences between BNN and each of them will be described in next section.

Bridge Neural Network

This section contains a detailed description of our proposed model. It is termed as Bridge Neural Network (BNN) because it acts as a bridge connecting two sets of variables by projecting them to a common feature subspace. Instead of telling the neural network to use CCA to investigate the relationship between two sets of variables (Andrew et al. 2013), we try to tell it which pairs of views are relevant and which are not, hoping the network to find a rule to analyze the potential relationship of the given data sources. The goal of BNN is to learn a common representation by mining the potential relationship of the specified data sources according to a given computer vision task.

Data sources

As we discussed in Figure 1, different tasks impose different complexity levels of common representations. Given a task, we should first construct the data pairs, which can learn common representations of the corresponding complexity level. For example, suppose we try to build a model to match items across views, we can set the data pairs as the sets of two data sources, denoted by $\{X_1, X_2\} \subset \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$, $X_1 = \{x_1^i\}_{i=1}^N$, $X_2 = \{x_2^i\}_{i=1}^N$. And the i-th component $x_1^i \in X_1$, $x_2^i \in X_2$ match each other. Here we define $S_p = \{x_1^i, x_2^i\}$ as the positive sample set and $S_n = \{x_1^i, x_2^j\}, i \neq j$ as the negative sample set. For the matching task, since the i-th components $x_1^i \in X_1$, $x_2^i \in X_2$ are supposed to match each other, a heuristic idea is the i-th component $x_1^i \in X_1$ should

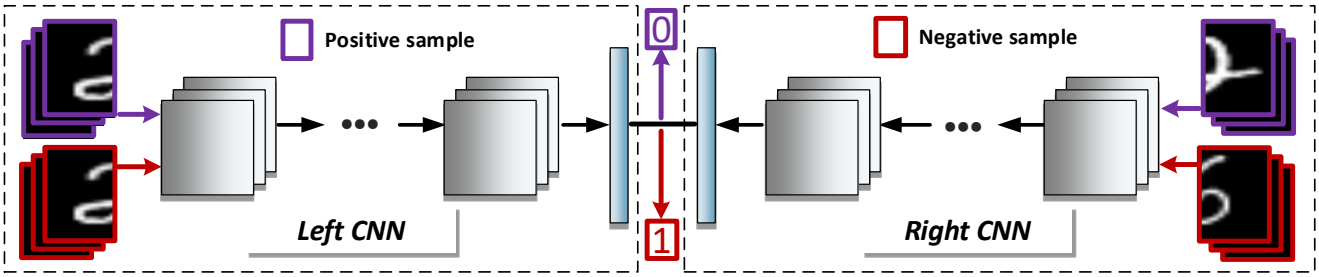

Figure 3: A schematic of bridge neural network, which employs two convolutional neural networks that project two given data sources into a common feature space. The Euclidean loss of the two output layers is supposed to close to 0 or 1 for positive samples or negative samples respectively. For the task of reconstruction of a missing view, transposed convolution networks should be introduced to make the common representation an encoder, see Figure 5.

not match the j-th component $x_2^j \in X_2$, $i \neq j$. Actually, in practice, we may have two components $x_1^i, x_1^j \in X_1$ very close to each other. If x_1^i matches x_2^i and x_1^j matches x_2^j, we may also have x_1^i matches x_2^j. Thus in the training process, the negative samples used to train BNN model are randomly selected from S_n, where we also take account the situation that the size of S_n is $N(N-1)$, that is almost square of the size of positive sample. The details of constructing data pairs according to different computer vision task can be found in the Experiments.

Architecture

Our proposed network architecture is illustrated in Figure 3, which contains two convolutional neural networks: left CNN $f_1(\cdot;\theta_1)$ and right CNN $f_2(\cdot;\theta_2)$ with weights (θ_1, θ_2). For given input data pairs (x_1, x_2), the two outputs of left CNN and right CNN are $f_1(x_1;\theta_1)$ and $f_2(x_2;\theta_2)$ respectively. Both CNNs contain k hidden layers $\{h_1, h_2, ..., h_k\}$ and each layer of the first $k-1$ layers, contains a convolution layer, batch normalization, followed by the activation function ReLU. The k-th layer is a linear projection of the output of h_{k-1}, followed by the sigmoid activation function. The BNN output of given input data pairs (x_1, x_2) is the Euclidean distance of the two outputs of left CNN and right CNN, that is defined as

$$f(x_1, x_2; \theta_1, \theta_2) = \frac{1}{\sqrt{n}} ||(f_1(x_1;\theta_1) - f_2(x_2;\theta_2))||, \quad (1)$$

where n is the dimension of the output of h_k, i.e., the common representation. We use a predefined threshold parameter $\gamma \in (0, 1)$ on $f(x_1, x_2; \theta_1, \theta_2)$ to determine whether input data pairs (x_1, x_2) have a potential relationship or not.

According to the definition of the positive sample set S_p and negative sample set S_n, BNN is supposed to judge that the data pair in S_p is relevant and the data pair in S_n is not relevant. Thus the BNN output $f(x_1, x_2; \theta_1, \theta_2)$ is supposed to be close to 0 if $(x_1, x_2) \in S_p$ and 1 if $(x_1, x_2) \in S_n$. Then define the loss on positive sample set as

$$l_p(S_p; \theta_1, \theta_2) = \frac{1}{|S_p|} \sum_{(x_1,x_2)\in S_p} (f(x_1, x_2; \theta_1, \theta_2) - 0)^2, \quad (2)$$

and the loss on negative sample set as

$$l_n(S_n; \theta_1, \theta_2) = \frac{1}{|S_n|} \sum_{(x_1,x_2)\in S_n} (f(x_1, x_2; \theta_1, \theta_2) - 1)^2, \quad (3)$$

where BNN can be seen as a binary classification problem. Thus the overall loss of BNN on S_p and S_n is

$$L_{bnn}(S_p, S_n; \theta_1, \theta_2) = \frac{l_p(S_p; \theta_1, \theta_2) + \alpha \cdot l_n(S_n; \theta_1, \theta_2)}{1+\alpha}, \quad (4)$$

where α is a parameter to balance the weights of positive samples and negative samples. Then the goal of BNN is, given S_p and S_n, to find weights (θ_1^*, θ_2^*) such that

$$(\theta_1^*, \theta_2^*) = \text{argmin}_{\theta_1,\theta_2} l(S_p, S_n; \theta_1, \theta_2). \quad (5)$$

Relation with DCCA. Here we present a discussion of the relation between BNN and DCCA (Andrew et al. 2013). Besides the slight difference of basic architecture, i.e. DCCA uses fully connected layers and BNN uses convolutional layers, the main difference between DCCA and BNN is the optimization objective function. Since the motivation of DCCA is from the goal of canonical correlation analysis, the objective of DCCA is to find (θ_1^*, θ_2^*) which make the output layers of left and right CNN maximally correlated, that is

$$(\theta_1^*, \theta_2^*) = \text{argmax}_{\theta_1,\theta_2} \text{corr}(f_1(X_1;\theta_1), f_2(X_2;\theta_2)), \quad (6)$$

which just involves the positive samples S_p. As that discussed in (Andrew et al. 2013), since the correlation objective is a function of the entire training set that does not decompose into a sum over data points, the stochastic optimization procedure cannot be directly used. Thus, DCCA proposed a L-BFGS based pre-training strategy for the stochastic method on mini-batches and achieved much better results. However, the objective function of BNN is Euclidean loss involving both positive samples and negative samples, which can be easily used to a stochastic optimization procedure that operates on mini-batch data points one at a time. Actually, the objective function of BNN can be seen as an alternative to the total correlation (6), as that shown in the following theorem.

Theorem 1 *For a group of weights (θ_1^*, θ_2^*) obtained by* (5)*, if the overall loss of BNN on S_p and S_n has $l(S_p, S_n; \theta_1^*, \theta_2^*) \to 0$, then the outputs of BNN has* $\mathrm{corr}(f_1(X_1; \theta_1^*), f_2(X_2; \theta_2^*)) \to n$ *as* $N \to \infty$.

Proof. To make the comparison more clear, we use the similar notations as that in DCCA (Andrew et al. 2013). Let $H_1 \in \mathbb{R}^{n\times N}$, $H_2 \in \mathbb{R}^{n\times N}$ be matrices whose columns are the outputs produced by the left CNN and right CNN on X_1 and X_2 of size N with weights (θ_1^*, θ_2^*). That is $H_1 = f_1(X_1; \theta_1^*), H_2 = f_2(X_2; \theta_2^*)$. Let $\bar{H}_1 = H_1 - \frac{1}{N} H_1 \mathbf{1}$ be the centered data matrix (resp. $\bar{H}_2$), and define $\hat{\Sigma}_{12} = \frac{1}{N-1}\bar{H}_1\bar{H}_2'$, and $\hat{\Sigma}_{11} = \frac{1}{N-1}\bar{H}_1\bar{H}_1' + r_1 I$ for regularization constant r_1 (resp. $\hat{\Sigma}_{22}$). Here $r_1 > 0$ is chosen to make $\hat{\Sigma}_{11}$ positive definite. Then by the discussion of Section 2 in (Andrew et al. 2013), the correlation, i.e., the total correlation of the top n components of H_1 and H_2 is the sum of the top n singular values of matrix $T = \hat{\Sigma}_{11}^{-1/2}\hat{\Sigma}_{12}\hat{\Sigma}_{22}^{-1/2}$, that is exactly the trace of matrix T:

$$\mathrm{corr}(H_1, H_2) = ||T||_{\mathrm{tr}} = \mathrm{tr}(T'T)^{1/2}. \tag{7}$$

Now we come to prove this theorem. For a group of weights (θ_1^*, θ_2^*) obtained by (5), if the overall loss of BNN on S_p and S_n has $l(S_p, S_n; \theta_1^*, \theta_2^*) \to 0$, by the definition in (4), we have

$$l_p(S_p; \theta_1^*, \theta_2^*) \to 0, \quad l_n(S_n; \theta_1^*, \theta_2^*) \to 0,$$

combined with the definition in (2) and (3), we obtain

$$f(x_1, x_2; \theta_1^*, \theta_2^*) \to 0, \quad \forall\, (x_1, x_2) \in S_p,$$
$$f(x_1, x_2; \theta_1^*, \theta_2^*) \to 1, \quad \forall\, (x_1, x_2) \in S_n.$$

By the definition of $f(x_1, x_2; \theta_1^*, \theta_2^*)$ in (1) and $H_1 = f_1(X_1; \theta_1^*)$, $H_2 = f_2(X_2; \theta_2^*)$, the above equations indicate that for the columns of H_1, h_1^i, $i = 1, ..., N$ and the columns of H_2, h_2^i, $i = 1, ..., N$,

$$||h_1^i - h_2^i|| \to 0, \quad ||h_1^i - h_2^j|| \to \sqrt{n}, i \neq j, \tag{8}$$

where the first equation means $H_1 \to H_2$, which yields

$$\bar{H}_1 \to \bar{H}_2. \tag{9}$$

Notice that the output of each CNN is followed with sigmoid activation, then each entry of vector h_1^i or h_2^i is in $(0, 1)$, which yields the l-th entry satisfies $|h_1^i(l) - h_2^i(l)| \in (0, 1)$. Combined with the second equation in (8), we obtain for $l = 1, ..., n$,

$$|h_1^i(l) - h_2^j(l)| \to 1, i \neq j. \tag{10}$$

And since $h_1^i(l), h_2^j(l) \in (0, 1)$, then we have either $h_1^i(l) \to 0$ or $h_2^i(l) \to 0$, which leads to $h_1^i(l)h_2^j(l) = 0, l = 1, ..., n$. This means the inner product of vector h_1^i and h_2^i has $\langle h_1^i, h_2^j \rangle \to 0$, $i \neq j$, leading to $H_2'H_1 \to D$, where $D = \mathrm{diag}(\langle h_1^1, h_2^1 \rangle, \langle h_1^2, h_2^2 \rangle, ..., \langle h_1^N, h_2^N \rangle)$. Since $H_1 \to H_2$, we have $\langle h_1^i, h_2^i \rangle \to ||h_1^i||^2$, which means D is close to a positive diagonal matrix. By the definition of centered data matrix $\bar{H}_1$, $\bar{H}_2$ and a simple calculation, as $N \to \infty$, we have

$$\bar{H}_2'\bar{H}_1 \to D. \tag{11}$$

In order to simplify the discussion, suppose we have $N = Kn$ with $K \in \mathbb{Z}$. Then we can redefine $\bar{H}_i = (\bar{H}_{i,1}, \bar{H}_{i,2}, ..., \bar{H}_{i,K})$ with $\bar{H}_{i,j} \in \mathbb{R}^{n\times n}$, $i = 1, 2$, $j = 1, ..., K$. Then the diagonal matrix D can also be redefined to $D = (D_1, D_2, ...D_K)$ with $D_j \in \mathbb{R}^{n\times n}$, $j = 1, ..., K$. Then the equation (11) means $\bar{H}_{2,j}'\bar{H}_{1,j} \to D_j$. Since D is close to a positive diagonal matrix, combined with lemma that the left inverse of a square matrix is the right inverse, we have $\bar{H}_{1,j}\bar{H}_{2,j}' \to D_j$ and $\bar{H}_{1,j} \to \bar{H}_{2,j}$ according to (9), thus we obtain $\bar{H}_{2,j}\bar{H}_{1,j}' \to D_j$. Then we have $\bar{H}_2\bar{H}_1' \to \hat{D} := \sum_{j=1}^K D_j$, combined with (9), we have

$$\bar{H}_1\bar{H}_1',\ \bar{H}_2\bar{H}_2',\ \bar{H}_2\bar{H}_1' \to \hat{D}, \tag{12}$$

which means that the matrices $\hat{\Sigma}_{11}, \hat{\Sigma}_{12}, \hat{\Sigma}_{22}$ are close to a same positive diagonal matrix, leading to $T \to I$. Combined with the definition of $\mathrm{corr}(H_1, H_2)$ in (7), and $H_1 = f_1(X_1; \theta_1^*)$, $H_2 = f_2(X_2; \theta_2^*)$, as $N \to \infty$ we have

$$\mathrm{corr}(f_1(X_1; \theta_1^*), f_2(X_2; \theta_2^*)) \to n.$$

This ends the proof.

We should remark that the above proof can be just asymptotically understood. Actually, since we always have $n \ll N$, it's impossible to have $H_2'H_1$ be equal to a diagonal matrix. Theorem 1 is proposed to show that the objective of BNN is asymptotically equivalent to maximizing the correlation of the outputs of two CNNs on input data pairs, i.e. two views as that named in DCCA. In conclusion, the benefits of using BNN than DCCA can be summarized as

- Both for 1D signals and 2D signals, DCCA uses fully connected layers, while BNN uses convolutional layers, which makes BNN less computation cost. There are around 10^6 learnable parameters in DCCA model, while only about 10^5 parameters exist in our model.
- DCCA uses L-BFGS based pre-training strategy for a stochastic method on mini-batch training, but with no theoretical analysis to verify the effectiveness of the pre-training strategy and more cost in searching the pre-trained model. BNN proposes a novel optimization objective with asymptotic equivalence to the maximization of the total correlation, and the objective can be easily used to a stochastic optimization procedure that operates on mini-batch data points one at a time.

Actually, there is a potential relation of using L-BFGS based pre-training strategy on the architecture for DCCA. Since DCCA uses fully connected layers, the pre-training of the weights in each layer can be formulated to find a local minimum of the total squared error of a reconstructing problem, where the L-BFGS second-order optimization method can be used. Otherwise, if DCCA uses convolutional layers, it will be complicated to use L-BFGS, since L-BFGS is more suitable to optimize a matrix.

Relation with 2WayNet. The 2WayNet (Eisenschtat and Wolf 2017) is a novel, bi-directional neural network architecture for the task of matching vectors from two data sources. The main contribution of 2WayNet is using Euclidean loss and proposing some techniques to overcome

the common Euclidean regression optimization problems. In conclusion, the benefits of using BNN than 2WayNet can be summarized as

- Both for 1D signals and 2D signals, 2WayNet uses fully connected layers to make the signal able to be reconstructed (but no reconstruction results are found in 2WayNet), while BNN uses convolutional layers, which makes BNN gain performance benefits when dealing with high dimensional inputs, such as datasets of large size images.
- 2WayNet gives the lower bound analysis of the total correlation on the Euclidean loss, but with no upper bound analysis. Instead of just using Euclidean loss, the artificial negative sample is introduced in BNN to make the objective asymptotically equivalent to the total correlation, which can be seen as another technique to handle the Euclidean regression optimization problems.

Relation with Siamese Network. Siamese Network (Chopra, Hadsell, and LeCun 2005) is a classical neural network architecture, which learns the similarity between two inputs. It consists of two identical neural networks, each taking one of the two inputs. The similarity between them is then calculated by feeding the last layers of the two networks into a contrastive loss function. BNN has a similar positive/negative-sample based optimization objective with Siamese Network, however there are three fundamental differences between them.

- The main task of Siamese Network is pair matching problem, while BNN focuses on common representation learning which can be used in variety tasks.
- The loss function of Siamese Network is a summation of product between label and a Euclidean-based pre-designed loss, while the loss function of BNN is a Euclidean-based binary classification loss.
- Siamese Network shares weights for right and left networks, while BNN has two totally independent networks which enable it to solve multimodal problems naturally.

Training process

Once the BNN model, i.e. $\{h_1, h_2, ..., h_k\}$ of left CNN and right CNN, is defined, we can use a stochastic optimization procedure, e.g. SGD with mini-batch to train the weights given the training samples. Here we train the left CNN and the right CNN alternately in a single iteration, i.e. one mini-batch. As that discussed before, the negative samples used are randomly selected from S_n according to a predefined parameter ξ, named NP ratio, which denotes the ratio of the size of the negative samples on that of positive samples. See Algorithm 1 for the details of the training process of BNN.

Actually, using randomly selected negative samples is a trade-off strategy. In the ideal condition, we hope using all negative samples in the training, which is impossible in practice. We compared the performance of the proposed negative sample generating strategy and the ideal one (training with all negative samples) based on a small sub-dataset. The difference between the results of them is negligible.

Algorithm 1 Training process of BNN

Construct positive samples $S_p = \{x_1^i, x_2^i\}$;
Define balance parameter α, learning rate η and NP ratio ξ;
Randomly initialize weights θ_1, θ_2;
for Each epoch **do**
  Randomly construct negative samples: $S_n = \{x_1^i, x_2^j\}, i \neq j$ with size $N\xi$;
  for mini-batch $S_p^i \subset S_p, S_n^i \subset S_n$ **do**
    Update left CNN with $\theta_1 \leftarrow \theta_1 - \eta \frac{\partial l_{bnn}(S_p^i, S_n^i; \theta_1, \theta_2)}{\partial \theta_1}$;
    Update right CNN with $\theta_2 \leftarrow \theta_2 - \eta \frac{\partial l_{bnn}(S_p^i, S_n^i; \theta_1, \theta_2)}{\partial \theta_2}$;
  end for
end for

Experiments

In this section, three experiments are presented, including pair matching test, Canonical Correlation Analysis and transfer learning across views. We perform our experiments based on two datasets commonly used in the recent literature for CCA test: MNIST (LeCun et al. 1998) half matching and X-Ray Microbeam Speech data (XRMB) (Westbury 1994). As for the model we use in our experiment, both the left and right networks have structures of 3 successive convolution layers with 10 filters (3×3 filters for MNIST, 1×3 filters for XRMB) and batch normalization. Each output of the last convolution layers of the two networks connects to a fully connected layer, linearizing into a n-dimensional vector.

Actually, CNN is an optional choice in BNN. However, the experiments demonstrate that using lightweight CNN is a better choice since it can achieve the similar performances as that of using full-connected layers (For MNIST dataset, the corresponding accuracy, precision, recall of pair matching and CCA result is 92.10, 84.61, 93.29, 49.23). Here we ignore the detailed discussion of comparison.

Data Description

MNIST half matching MNIST handwritten digits image dataset (LeCun et al. 1998) is commonly used for training various image processing systems. The database contains 60,000 training images and 10,000 testing images of 28×28 pixels. In our experiments, each image is cut into two 28×14 pixel halves vertically. Two strategies are used to construct positive and negative samples: 1) For matching test and CCA examination, we only consider two halves of the same images as positive samples. 2) For transfer learning across views, all pairs of halves from the same digits (with the same labels) are set as positive samples.

X-Ray Microbeam Speech data (XRMB) In XRMB dataset (Westbury 1994), two sets of data are adopted: simultaneous acoustic and articulatory recordings. The acoustic views are MFCCs with 273-dimensional vector per frame, while the articulatory data is represented as a 112-dimensional vector. In our experiment, 60,000 random sam-

ples are used for training and 10,000 for testing. Simultaneous acoustic and articulatory recording pairs are considered as positive samples.

Pair Matching Across Two Views

Among the existing works mentioned before, this task has not been explicitly implemented. Here we will only show our own results. On the testing data, we generate 10,000 positive pairs and 20,000 negative pairs. The output dimension n of both networks is set to 50 for MNIST dataset, 112 for XRMB dataset, and the judging threshold γ is set to 0.5. To make the results more detailed, we also show the precision, recall, and F1 score in addition to the accuracy measurement.

Table 1: Performance of pair matching test across two views learned by BNN on MNIST dataset (%).

	MNIST	XRMB
Accuracy	95.63	87.77
Precision	88.71	75.64
Recall	99.58	93.39
F1 Score	93.83	83.58

Table 1 suggests that BNN has a outstanding performance on pair matching test, especially the value of recall rate, which proves that BNN is able to learning exact features for accurately predict the relationship of two views as we expected.

(a) Top 100 False Positive Samples (b) Original Images

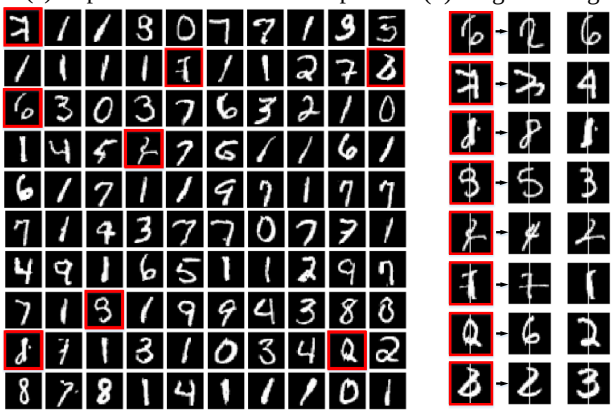

Figure 4: (a) Top 100 false positive samples. Samples outlined in red are obviously negative for humans; (b) The original two images of those negative pairs outlined in (a).

Figure 4 (a) shows top 100 false positive samples on MNIST dataset. Most of those pairs are joined smoothly so that even humans will regard them as positive samples visually. Though some of them are obviously negative samples, their original images are illegible (See Figure 4 (b)). In this case, the results can be considered as reasonable and acceptable.

Correlation between representations of two views

In this section, we will present the CCA result of BNN and compare it with the models we mentioned above. We follow Andrew's Method (Andrew et al. 2013) calculating the total correlation captured in the n=50 (MNIST) or n=112 (XRMB) dimensions of the common representations learned by the BNN. The structure of networks is the same as the one in the previous experiment. However, the testing data only contains positive samples, the same as all the experiments in the other works. The results are reported in Table 2. All CCA results of comparative models are stated in the literature of 2WayNet (Eisenschtat and Wolf 2017).

Table 2: Total correlation of the common representations learned by different models on MNIST and XRMB datasets.

Methods	MNIST	XRMB
CCA	28	16.9
DCCA	39.7	92.9
RCCA	44.5	104.5
DCCAE	25.34	41.47
CorrNet	48.07	95.01
2WayNet	49.15	110.18
BNN	**49.32**	**110.65**
Upper Bound	50	112

The reported values in Table 2 are the sum of the correlations captured in the 50 (MNIST) or 112 (XRMB) dimensions of the learned representations of the two views. The total correlations learned by BNN are closer to the maximal value of 50 (MNIST) and 112 (XRMB), which are clearly better than the other models. Since the CCA results of 2WayNet are already very close to the maximum, any slight improvement can be considered significant.

Transfer Learning Across Views

In this section, all tests are implemented based on MNIST dataset. The experiment shows that the 50-dimension common representation learned in BNN from the half views of the images can be trained to predict the digits of the other half views. Linear SVM classifier provided by Scikit-learn (Pedregosa et al. 2011) is used in our experiment and the common representaion data used to trian the SVM classifier is inferenced by a well-trained and fixed BNN model. For each model list in Figure 3, we report 5-fold cross-validation accuracy on 10,000 images in the MNIST test dataset. Two sets of tests are reported: (i) training on the left views and testing on the right views, (ii) training on the right views and testing on the left views. All transfer learning results of comparative models are stated in CorrNet (Chandar et al. 2016).

In Table 3, single view corresponds to the classifier trained and tested on the same halves of images. The value of single view can represent the effectiveness of the learned common representation. Compared with the singe view results of CorrNet (Chandar et al. 2016), which are only 81.62

Table 3: Transfer learning (Left to Right, Right to Left) accuracy using the representations learned using different models on the MNIST dataset.

Methods	L. to R.	R. to L.
CCA	65.73	65.44
DCCA	68.10	75.71
MAE	64.14	68.88
CorrNet	77.05	78.81
BNN	**90.21**	**91.38**
Single View	**93.17**	**91.52**

and 80.06 for left and right single views, the single view results of BNN show that the learned common representations with only half digits are good enough for classification. In addition, the single view result is also the upper bound of our transfer learning performance. For BNN, the result of transfer learning accuracy is highly close to the accuracy of ideal single view case, indicating that the common representations captured by BNN gains excellent ability of transfer learning. More importantly, it performs significantly better than all the other models as well. The high accuracy of transfer learning and single view on the representation indicates that the common representation found by BNN is much closer to the ideal common feature space for the classification task. Compared with the limited improvement on the task of total correlation captured in Table 2, the huge improvement on transfer learning demonstrates the efficiency of using different positive samples task by task, since the training samples used in comparison methods are consistent across tasks.

Reconstruction Across Views

Firstly, we present the reconstruction architecture using BNN. Although MAE structure is unnecessary to learn common representations in our algorithm, view reconstruction can be achieved by adding two transposed convolution networks following the common representation layers of BNN. The architecture is illustrated in Figure 5 which contains a standard bridge neural network and two transposed convolutional neural networks. For given input data pairs (x_1, x_2), the two outputs of the left CNN and right CNN are $f_1(x_1; \theta_1)$ and $f_2(x_2; \theta_2)$ respectively. For the given common representations (z_1, z_2) of the two views, the reconstruction of them are the outputs of the two transposed convolutional neural networks, which are $f_1'(z_1; \theta_1')$ and $f_2'(z_2; \theta_2')$ respectively.

To minimize the reconstruction error, new loss functions are introduced beside the loss of BNN (4). For self-reconstructions, we have the outputs of the two transposed convolutional neural networks as $i = 1, 2$,

$$x'_{i\_sel} = f'_i(f_i(x_i; \theta_i); \theta'_i). \tag{13}$$

Then we define the loss of self-reconstruction on S_p as

$$l_{self}(S_p; \theta_1, \theta_2, \theta'_1, \theta'_2) = \frac{1}{|S_p|} \sum_{(x_1,x_2)\in S_p} (||x'_{1\_sel} - x_1|| + ||x'_{2\_sel} - x_2||). \tag{14}$$

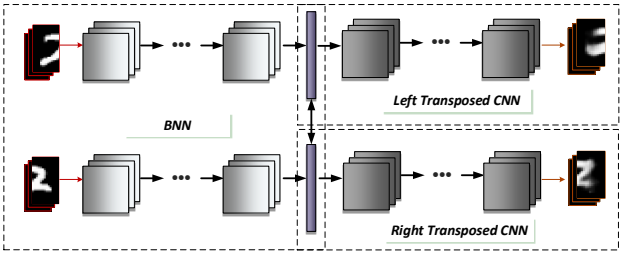

Figure 5: A schematic of bridge neural network with reconstruction structure, which employs a standard BNN (3) and two transposed convolutional neural networks.

For cross-reconstructions, we have two outputs as

$$\begin{aligned} x'_{1\_cro} &= f'_1(f_2(x_2; \theta_2); \theta'_1), \\ x'_{2\_cro} &= f'_2(f_1(x_1; \theta_1); \theta'_2). \end{aligned} \tag{15}$$

Then we define the loss of cross-reconstruction on S_p as

$$l_{cross}(S_p; \theta_1, \theta_2, \theta'_1, \theta'_2) = \frac{1}{|S_p|} \sum_{(x_1,x_2)\in S_p} (||x'_{1\_cro} - x_1|| + ||x'_{2\_cro} - x_2||). \tag{16}$$

Thus the overall loss of BNN with reconstruction is

$$l_{total}(S_p, S_n; \theta_1, \theta_2, \theta'_1, \theta'_2) = l_{bnn} + l_{self} + l_{cross}. \tag{17}$$

In the training process, i.e. Algorithm 1, we replace the loss l_{bnn} with l_{total} and update the weights in transposed convolutional neural networks θ'_i together with θ_i, $i = 1, 2$.

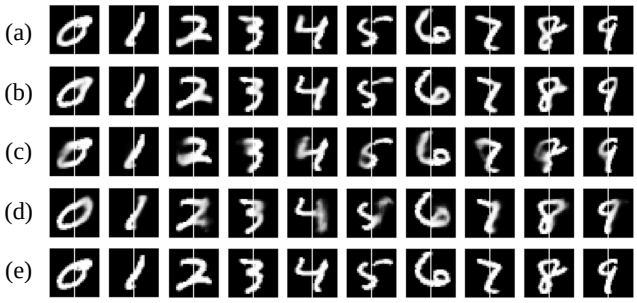

Figure 6: Reconstruction results of BNN for MNIST. (a) Left view self-reconstruction; (b) Right view self-reconstruction; (c) Right to left cross-view reconstruction; (d) Left to right cross-view reconstruction; (e) Original images of two views.

In this experiment, view reconstruction is implemented on the MNIST dataset. The structure of BNN is the same as previous. As for the two transposed convolutional networks, in each of them, the 50-dimension common representation layer connects to a 50×64 fully connected layer and then reshaped into a $4 \times 2 \times 8$ matrix. It is followed by 3 successive transposed convolution layers with $8/16/32$ filters respectively (3×3 kernel), outputting a $28 \times 14 \times 32$ matrix. At last, one 3×3 filter convolutional layer will project the matrix into a $28 \times 14 \times 1$ output image.

Figure 6 shows the output images of self-reconstruction and cross-reconstruction both on left and right views of a few samples. The results are more clear and sharp both in self-reconstruction and cross-reconstruction compared with the visual results of CorrNet (Chandar et al. 2016). The visually satisfying, which indicates that reconstruction can also be an optional function for Bridge Neural Network.

Conclusion

This paper proposes bridge neural network (BNN) with lightweight convolution layers to dig the potential relationship of two given data sources, that can be specified according to a computer vision task. The training objective with the artificial negative samples is introduced and it's asymptotically equivalent to maximizing the total correlation of the two data sources in theory. The experiments on the tasks, including pair matching, canonical correlation analysis, transfer learning and reconstruction demonstrate the state-of-the-art performance of BNN.

Acknowledgements

This work was supported by the Innovation Foundation of Qian Xuesen Laboratory of Space Technology and the National Natural Science Foundation of China (61702520).

References

Akaho, S. 2006. A kernel method for canonical correlation analysis. *arXiv preprint cs/0609071*.

Andrew, G.; Arora, R.; Bilmes, J.; and Livescu, K. 2013. Deep canonical correlation analysis. In *International Conference on Machine Learning*, 1247–1255.

Arora, R., and Livescu, K. 2013. Multi-view cca-based acoustic features for phonetic recognition across speakers and domains. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 7135–7139. IEEE.

Bach, F. R., and Jordan, M. I. 2002. Kernel independent component analysis. *Journal of machine learning research* 3(Jul):1–48.

Chandar, S.; Khapra, M. M.; Larochelle, H.; and Ravindran, B. 2016. Correlational neural networks. *Neural computation* 28(2):257–285.

Chopra, S.; Hadsell, R.; and LeCun, Y. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, 539–546. IEEE.

Dhillon, P.; Foster, D. P.; and Ungar, L. H. 2011. Multi-view learning of word embeddings via cca. In *Advances in neural information processing systems*, 199–207.

Eisenschtat, A., and Wolf, L. 2017. Linking image and text with 2-way nets. *arXiv preprint*.

Hardoon, D. R.; Mourao-Miranda, J.; Brammer, M.; and Shawe-Taylor, J. 2007. Unsupervised analysis of fmri data using kernel canonical correlation. *NeuroImage* 37(4):1250–1259.

Hardoon, D. R.; Szedmak, S.; and Shawe-Taylor, J. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural computation* 16(12):2639–2664.

Hotelling, H. 1936. Relations between two sets of variates. *Biometrika* 28(3/4):321–377.

Kim, T.-K.; Wong, S.-F.; and Cipolla, R. 2007. Tensor canonical correlation analysis for action classification. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 1–8. IEEE.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Melzer, T.; Reiter, M.; and Bischof, H. 2001. Nonlinear feature extraction using generalized canonical correlation analysis. In *International Conference on Artificial Neural Networks*, 353–360. Springer.

Michaeli, T.; Wang, W.; and Livescu, K. 2016. Nonparametric canonical correlation analysis. In *International Conference on Machine Learning*, 1967–1976.

Mineiro, P., and Karampatziakis, N. 2014. A randomized algorithm for cca. *arXiv preprint arXiv:1411.3409*.

Ngiam, J.; Khosla, A.; Kim, M.; Nam, J.; Lee, H.; and Ng, A. Y. 2011. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 689–696.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research* 12(Oct):2825–2830.

Vinod, H. D. 1976. Canonical ridge and econometrics of joint production. *Journal of econometrics* 4(2):147–166.

Wang, W.; Arora, R.; Livescu, K.; and Bilmes, J. 2015. On deep multi-view representation learning. In *International Conference on Machine Learning*, 1083–1092.

Wang, L.; Li, Y.; and Lazebnik, S. 2016. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5005–5013.

Westbury, J. 1994. X-ray microbeam speech production database user's handbook: Madison. *WI: Waisman Center, University of Wisconsin*.

Xu, C.; Tao, D.; and Xu, C. 2013. A survey on multi-view learning. *Computer Science*.

Yan, F., and Mikolajczyk, K. 2015. Deep correlation for matching images and text. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 3441–3450. IEEE.

Zhao, J.; Xie, X.; Xu, X.; and Sun, S. 2017. Multi-view learning overview: Recent progress and new challenges. *Information Fusion* 38:43–54.