On Deep Multi-View Representation Learning

Weiran Wang WEIRANWANG@TTIC.EDU
Toyota Technological Institute at Chicago

Raman Arora ARORA@CS.JHU.EDU
Johns Hopkins University

Karen Livescu KLIVESCU@TTIC.EDU
Toyota Technological Institute at Chicago

Jeff Bilmes BILMES@EE.WASHINGTON.EDU
University of Washington, Seattle

Abstract

We consider learning representations (features) in the setting in which we have access to multiple unlabeled views of the data for representation learning while only one view is available at test time. Previous work on this problem has proposed several techniques based on deep neural networks, typically involving either autoencoder-like networks with a reconstruction objective or paired feedforward networks with a correlation-based objective. We analyze several techniques based on prior work, as well as new variants, and compare them experimentally on visual, speech, and language domains. To our knowledge this is the first head-to-head comparison of a variety of such techniques on multiple tasks. We find an advantage for correlation-based representation learning, while the best results on most tasks are obtained with our new variant, deep canonically correlated autoencoders (DCCAE).

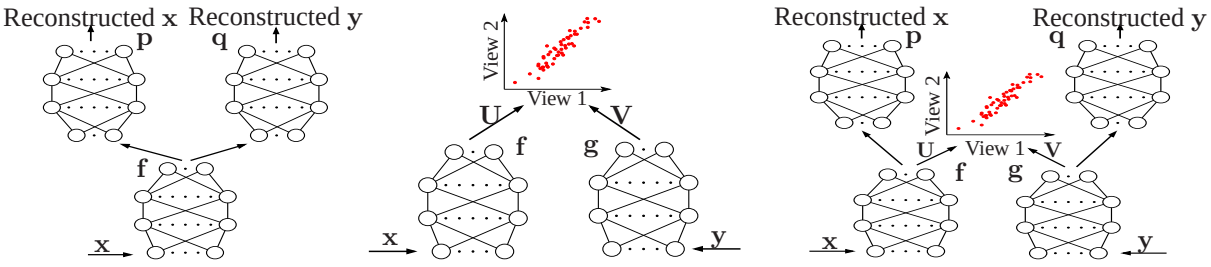
1. Introduction

In many applications, we have access to multiple "views" of data at training time while only one view is available at test time. The views can be multiple measurement modalities, such as simultaneously recorded audio + video (Kidron et al., 2005; Chaudhuri et al., 2009), audio + articulation (Arora & Livescu, 2013), images + text (Hardoon et al., 2004; Socher & Li, 2010; Hodosh et al., 2013), or parallel text in two languages (Vinokourov et al., 2003; Haghighi et al., 2008; Chandar et al., 2014; Faruqui & Dyer, 2014), but may also be different information extracted from the same source, such as words + context (Pennington et al., 2014) or document text + text of inbound hyperlinks (Bickel & Scheffer, 2004). The presence of multiple information sources presents an opportunity to learn better representations (features) by analyzing multiple views simultaneously. Typical approaches are based on learning a feature transformation of the "primary" view (the one available at test time) that captures useful information from the second view using a paired two-view training set. Under certain assumptions, theoretical results exist showing the advantages of multi-view techniques for downstream tasks (Kakade & Foster, 2007; Foster et al., 2009; Chaudhuri et al., 2009).

Several recently proposed approaches for multi-view representation learning are based on deep neural networks (DNNs), inspired by their success in typical unsupervised (single-view) feature learning settings (Hinton & Salakhutdinov, 2006). Compared to kernel methods, DNNs can more easily process large amounts of training data and, as a parameteric method, do not require referring to the training set at test time.

There are two main training criteria (objectives) that have been applied for DNN-based multi-view representation learning. One is based on autoencoders, where the objective is to learn a compact representation that best reconstructs the inputs (Ngiam et al., 2011). The second approach is based on canonical correlation analysis (CCA, Hotelling, 1936), which learns features in two views that are maximally correlated. CCA and its kernel extension (Lai & Fyfe, 2000; Akaho, 2001; Bach & Jordan, 2002; Hardoon et al., 2004) have long been the workhorse for multi-view feature learning and dimensionality re-

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).



(a) SplitAE (b) DCCA (c) DCCAE/CorrAE/DistAE

Figure 1. Schematic diagram of DNN-based multi-view representation learning models.

duction (Vinokourov et al., 2003; Kakade & Foster, 2007; Socher & Li, 2010; Dhillon et al., 2011). Multiple neural network based CCA-like models have been proposed (Lai & Fyfe, 1999; Hsieh, 2000), but the full DNN extension of CCA, termed deep CCA (DCCA, Andrew et al., 2013) has been developed only recently.

The contributions of this paper are as follows. We compare several DNN-based approaches, along with linear and kernel CCA, in the unsupervised multi-view feature learning setting where the second view is not available at test time. Prior work has shown the benefit of multi-view methods on tasks such as retrieval (Vinokourov et al., 2003; Hardoon et al., 2004; Socher & Li, 2010; Hodosh et al., 2013), clustering (Blaschko & Lampert, 2008; Chaudhuri et al., 2009) and classification/recognition (Dhillon et al., 2011; Arora & Livescu, 2013; Ngiam et al., 2011). However, to our knowledge no head-to-head comparison on multiple tasks has previously been done. We address this gap by comparing approaches based on prior work, as well as new variants developed here. Empirically, we find that CCA-based approaches tend to outperform unconstrained reconstruction-based approaches. One of the new methods we propose, a DNN-based model combining CCA and autoencoder-based terms, is the consistent winner across several tasks. To facilitate future work, we release our implementations and a new benchmark dataset of simulated two-view data based on MNIST.

2. DNN-based multiview feature learning

Notations In the multi-view feature learning scenario, we have access to paired observations from two views, denoted $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$, where N is the sample size, $\mathbf{x}_i \in \mathbb{R}^{D_x}$ and $\mathbf{y}_i \in \mathbb{R}^{D_y}$ for $i = 1, \ldots, N$. We also denote the data matrices for each view by $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$. We use bold-face letters, e.g. $\mathbf{f}$, to denote mappings implemented by kernel machines or DNNs, with a corresponding set of learnable parameters, denoted, e.g., $\mathbf{W_f}$. We write the $\mathbf{f}$-projected (view 1) data matrix as $\mathbf{f}(\mathbf{X}) = [\mathbf{f}(\mathbf{x}_1), \ldots, \mathbf{f}(\mathbf{x}_N)]$. The dimensionality of the projection (feature) is denoted L.

We now describe the DNN-based multi-view feature learning algorithms considered here, with corresponding schematic diagrams given in Fig. 1.

2.1. Split autoencoders (SplitAE)

Ngiam et al. (2011) propose to extract shared representations by reconstructing both views from the one view that is available at test time. In this approach, the feature extraction network $\mathbf{f}$ is shared while the reconstruction networks $\mathbf{p}$ and $\mathbf{q}$ are separate for each view. We refer to this model as a split autoencoder (SplitAE), shown schematically in Fig. 1 (a). The objective of this model is the sum of reconstruction errors for the two views (we omit the ℓ_2 weight decay term for all models in this section):

$$\min_{\mathbf{W_f}, \mathbf{W_p}, \mathbf{W_q}} \frac{1}{N} \sum_{i=1}^{N} (\|\mathbf{x}_i - \mathbf{p}(\mathbf{f}(\mathbf{x}_i))\|^2 + \|\mathbf{y}_i - \mathbf{q}(\mathbf{f}(\mathbf{x}_i))\|^2).$$

The intuition for this model is that the shared representation can be extracted from a single view, and can be used to reconstruct all views. [1] The autoencoder loss is the empirical expectation of the loss incurred at each training sample, and thus stochastic gradient descent (SGD) can be used to optimize the objective efficiently, with the gradient estimated from a small minibatch of samples.

2.2. Deep canonical correlation analysis (DCCA)

Andrew et al. (2013) propose a DNN extension of CCA termed deep CCA (DCCA; see Fig. 1 (b). In DCCA, two DNNs $\mathbf{f}$ and $\mathbf{g}$ are used to extract nonlinear features for each view and the canonical correlation between the extracted features $\mathbf{f}(\mathbf{X})$ and $\mathbf{g}(\mathbf{Y})$ is maximized:

$$\begin{aligned} \max_{\mathbf{W_f}, \mathbf{W_g}, \mathbf{U}, \mathbf{V}} \quad & \frac{1}{N} \operatorname{tr} \left(\mathbf{U}^\top \mathbf{f}(\mathbf{X}) \mathbf{g}(\mathbf{Y})^\top \mathbf{V} \right) \\ \text{s.t.} \quad & \mathbf{U}^\top \left(\frac{1}{N} \mathbf{f}(\mathbf{X}) \mathbf{f}(\mathbf{X})^\top + r_x \mathbf{I} \right) \mathbf{U} = \mathbf{I}, \\ & \mathbf{V}^\top \left(\frac{1}{N} \mathbf{g}(\mathbf{Y}) \mathbf{g}(\mathbf{Y})^\top + r_y \mathbf{I} \right) \mathbf{V} = \mathbf{I}, \\ & \mathbf{u}_i^\top \mathbf{f}(\mathbf{X}) \mathbf{g}(\mathbf{Y})^\top \mathbf{v}_j = 0, \quad \text{for} \quad i \neq j, \end{aligned} \tag{1}$$

[1] The authors also propose a bimodal deep autoencoder combining DNN transformed features from both views; this model is more natural for the multimodal fusion setting where both views are available at test time. Empirically, Ngiam et al. (2011) report that SplitAE tends to work better in the multi-view setting.

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_L]$ and $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_L]$ are the CCA directions that project the DNN outputs and $(r_x, r_y) > 0$ are regularization parameters for sample covariance estimation (Bie & Moor, 2003; Hardoon et al., 2004). In DCCA, $\mathbf{U}^\top \mathbf{f}(\cdot)$ is the final projection mapping used for testing. One intuition for CCA-based objectives is that, while it may be difficult to accurately reconstruct one view from the other view, it may be easier, and may be sufficient, to learn a predictor of a *function* (or *subspace*) of the second view. In addition, it should be helpful for the learned dimensions within each view to be uncorrelated so that they provide complementary information.

Optimization The DCCA objective couples all training samples through the whitening constraints, so stochastic gradient descent (SGD) cannot be applied in a standard way. It has been observed by Wang et al. (2015) that DCCA can still be optimized efficiently as long as the gradient is estimated using a sufficiently large minibatch (with gradient formulas given in Andrew et al., 2013). Intuitively, this approach works because a large minibatch contains enough information for estimating the covariances.

2.3. Deep canonically correlated autoencoders (DCCAE)

Inspired by both CCA and reconstruction-based objectives, we propose a new model that consists of two autoencoders and optimizes the combination of canonical correlation between the learned bottleneck representations and the reconstruction errors of the autoencoders. In other words, we optimize the following objective

$$\min_{\mathbf{W_f}, \mathbf{W}_g, \mathbf{W_p}, \mathbf{W_q}, \mathbf{U}, \mathbf{V}} -\frac{1}{N} \operatorname{tr}\left(\mathbf{U}^\top \mathbf{f}(\mathbf{X}) \mathbf{g}(\mathbf{Y})^\top \mathbf{V}\right) + \frac{\lambda}{N} \sum_{i=1}^{N} (\|\mathbf{x}_i - \mathbf{p}(\mathbf{f}(\mathbf{x}_i))\|^2 + \|\mathbf{y}_i - \mathbf{q}(\mathbf{g}(\mathbf{y}_i))\|^2) \quad (2)$$
$$\text{s.t.} \quad \text{the same constraints in (1)},$$

where $\lambda > 0$ is a trade-off parameter. Alternatively, this approach can be seen as adding an autoencoder regularization term to DCCA. We call this approach deep canonically correlated autoencoders (DCCAE). Similarly to DCCA, we apply stochastic optimization to the DCCAE objective; the stochastic gradient is the sum of the gradient for the autoencoder term and the gradient for the DCCA term.

Interpretations CCA maximizes the mutual information between the projected views for certain distributions (Borga, 2001), while training an autoencoder to minimize reconstruction error amounts to maximizing a lower bound on the mutual information between inputs and learned features (Vincent et al., 2010). The DCCAE objective offers a trade-off between the information captured in the (input, feature) mapping within each view on the one hand, and the information in the (feature, feature) relationship across views on the other. Intuitively, this is the same principle as the information bottleneck method (Tishby et al., 1999), and indeed, in the case of Gaussian variables, the information bottleneck method finds the same subspaces as CCA (Chechik et al., 2005).

2.4. Correlated autoencoders (CorrAE)

In the next model, we replace the CCA term in the DCCAE objective with the sum of the scalar correlations between the pairs of learned dimensions across views, which is an alternative measure of agreement between views. In other words, the feature dimensions within each view are *not* constrained to be uncorrelated with each other. This model is intended to test how important the original CCA constraint is. We call this model correlated autoencoders (CorrAE), shown in Fig. 1 (c). Its objective can be equivalently written in a constrained form as

$$\min_{\mathbf{W_f}, \mathbf{W}_g, \mathbf{W_p}, \mathbf{W_q}, \mathbf{U}, \mathbf{V}} -\frac{1}{N} \operatorname{tr}\left(\mathbf{U}^\top \mathbf{f}(\mathbf{X}) \mathbf{g}(\mathbf{Y})^\top \mathbf{V}\right) + \frac{\lambda}{N} \sum_{i=1}^{N} (\|\mathbf{x}_i - \mathbf{p}(\mathbf{f}(\mathbf{x}_i))\|^2 + \|\mathbf{y}_i - \mathbf{q}(\mathbf{g}(\mathbf{y}_i))\|^2) \quad (3)$$
$$\text{s.t. } \mathbf{u}_i^\top \mathbf{f}(\mathbf{X}) \mathbf{f}(\mathbf{X})^\top \mathbf{u}_i = \mathbf{v}_i^\top \mathbf{g}(\mathbf{Y}) \mathbf{g}(\mathbf{Y})^\top \mathbf{v}_i = N, \ 1 \leq i \leq L.$$

where $\lambda > 0$ is a trade-off parameter. It is clear that the constraint set in (3) is a relaxed version of that of (2). We will demonstrate that this difference results in a large performance gap. We apply the same optimization strategy of DCCAE to CorrAE.

CorrAE is similar to the model of Chandar et al. (2014), who try to learn vectorial word representations using parallel corpora from two languages. They use DNNs in each view (language) to predict the bag-of-words representation of the input sentences, or that of the paired sentences from the other view, while encouraging the learned bottleneck layer representations to be highly correlated.

2.5. Minimum-distance autoencoders (DistAE)

The CCA objective can be seen as minimzing the distance between the learned projections of the two views, while satisfying the whitening constraints for the projections (Hardoon et al., 2004). The constraints complicate the optimization of CCA-based objectives, as pointed out above. This observation motivates us to consider additional objectives that decompose into sums over training examples, while maintaining the intuition of the CCA objective as a reconstruction error between two mappings. Here we consider a variant we refer to as minimum-distance autoencoders (DistAE) that optimizes the following objective:

$$\min_{\mathbf{W_f}, \mathbf{W_g}, \mathbf{W_p}, \mathbf{W_q}} \frac{1}{N} \sum_{i=1}^{N} \frac{\|\mathbf{f}(\mathbf{x}_i) - \mathbf{g}(\mathbf{y}_i)\|^2}{\|\mathbf{f}(\mathbf{x}_i)\|^2 + \|\mathbf{g}(\mathbf{y}_i)\|^2} + \frac{\lambda}{N} \sum_{i=1}^{N} (\|\mathbf{x}_i - \mathbf{p}(\mathbf{f}(\mathbf{x}_i))\|^2 + \|\mathbf{y}_i - \mathbf{q}(\mathbf{g}(\mathbf{y}_i))\|^2) \quad (4)$$

which is a weighted combination of reconstruction errors of two autoencoders and the average discrepancy between the projected sample pairs. The denominator of the discrepancy term is used to keep the optimization from improving the objective by simply scaling down the projections (although they can never become identically zero due to the reconstruction terms). This objective is unconstrained and is the expectation of loss incurred at each training sample, so normal SGD applies using small minibatches.

3. Related work

We focus on related work based on feed-forward neural networks and the kernel extension of CCA. There has also been work using deep Boltzmann machines (Srivastava & Salakhutdinov, 2014; Sohn et al., 2014), where several layers of restricted Boltzmann machines (RBM) are stacked to represent each view, with an additional top layer that provides the joint representation. These are probabilistic graphical models, for which the maximum likelihood objective is intractable and the training procedures are more complex. Although probabilistic models have some advantages (e.g., dealing with missing values and generating samples in a natural way), DNN-based models are tractable and efficient to train.

3.1. DNN feature learning using CCA-like objectives

There have been several approaches to multi-view representation learning using neural networks with an objective similar to that of CCA. Under the assumption that the two views share a common cause (e.g., depth is a common cause for adjacent patches of images), Becker & Hinton (1992) maximize a sample-based estimate of mutual information between the common signal and the average outputs of neural networks for the two views. It is less straightforward, however, to develop sample-based estimates of mutual information in higher dimensions.

Lai & Fyfe (1999) propose to optimize the correlation (rather than canonical correlation) between the outputs of networks for each view, subject to scale constraints on each output dimension. Instead of directly solving this constrained formulation, the authors apply Lagrangian relaxation and solve the resulting unconstrained objective using SGD. The objective in this work is different from that of CCA, as there are no constraints that the learned dimensions within each view be uncorrelated. Hsieh (2000) proposes a neural network based model involving three modules: one module for extracting a pair of maximally correlated one-dimensional features for the two views; and a second and third module for reconstructing the original inputs of the two views from the learned features. In this model, the feature dimensions can be learned one after another, each learned using as input the reconstruction residual from previous dimensions. The three modules are each trained separately, so there is no unified objective.

Kim et al. (2012) propose an algorithm that first uses deep belief networks and the autoencoder objective to extract features for two languages independently, and then applies linear CCA to the learned features (activations at the bottleneck layer of the autoencoders) to learn the final representation. In this two-step approach, the DNN weight parameters are not updated to optimize the CCA objective.

3.2. Kernel CCA

Another nonlinear extension of CCA is kernel CCA (KCCA, Lai & Fyfe, 2000; Akaho, 2001; Melzer et al., 2001; Bach & Jordan, 2002; Hardoon et al., 2004). KCCA corresponds to using (potentially infinite-dimensional) feature mappings induced by positive-definite kernels $k_x(\cdot,\cdot), k_y(\cdot,\cdot)$ (e.g., Gaussian RBF kernels $k(\mathbf{a},\mathbf{b}) = e^{-\|\mathbf{a}-\mathbf{b}\|^2/2s^2}$ where s is the kernel width), and learning a linear CCA on them with linear projections $(\mathbf{U},\mathbf{V})$. From the representer theorem of reproducing kernel Hilbert spaces (Schölkopf & Smola, 2001), the final projection can be written as linear combinations of kernel functions evaluated on the training set, i.e., $\mathbf{U}^\top\mathbf{f}(\cdot) = \sum_{i=1}^N \boldsymbol{\alpha}_i k_x(\mathbf{x},\mathbf{x}_i)$ where $\boldsymbol{\alpha}_i \in \mathbb{R}^L, i = 1,\dots,N$; one can then work with the kernel matrices and directly solve for the linear coefficients $\{\boldsymbol{\alpha}_i\}_{i=1}^N$. KCCA involves an $N \times N$ eigenvalue problem and so is challenging in both memory (storing the kernel matrices) and time (solving the eigenvalue system naïvely costs $\mathcal{O}(N^3)$). To alleviate these issues, various kernel approximation techniques have been proposed, such as random Fourier features (Lopez-Paz et al., 2014) and the Nyström approximation (Williams & Seeger, 2001). In random Fourier features, we randomly sample M D_x/D_y-dimensional vectors from a Gaussian distribution and map the original inputs to $\mathbb{R}^M$ by computing the dot product with the random samples followed by an elementwise cosine; the inner products between transformed samples approximate kernel similarities between original inputs. In the Nyström approximation, we randomly select M training samples and construct the $M \times M$ kernel matrix for these samples, and use its eigen-decomposition to obtain a rank-M approximation of the full kernel matrix. Both techniques produce rank-M approximations of the kernel matrices with computational complexity $\mathcal{O}(M^3 + M^2N)$; but random Fourier features are data independent and more efficient to generate. Other approximation techniques such as incomplete Cholesky decomposition (Bach & Jordan, 2002), partial Gram-Schmidt (Hardoon et al., 2004), and incremental SVD (Arora & Livescu, 2012) have also been proposed. However, for very large training sets, as in some of our tasks below, it remains difficult and costly to approximate KCCA well. Although iterative algorithms have recently been introduced for very large CCA problems (Lu & Foster, 2014), they are aimed at sparse matrices and do not have a natural out-of-sample extension.

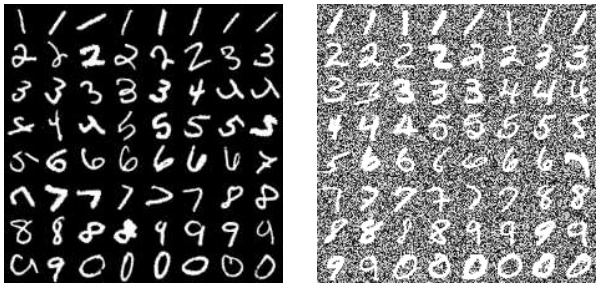


Figure 2. Selection of view 1 images (left) and their corresponding view 2 images (right) from our noisy MNIST dataset.

4. Experiments

We compare the following methods in the multi-view learning setting, focusing on several downstream tasks: noisy digit image classification, speech recognition, and word pair semantic similarity.

DNN-based models, including SplitAE, CorrAE, DCCA, DCCAE, and DistAE.

Linear CCA (CCA), corresponding to DCCA with only a linear network without hidden layers for both views.

Kernel CCA approximations. Exact KCCA is intractable for our tasks; we instead implement two kernel approximation techniques, using Gaussian RBF kernels. The first implementation, denoted **FKCCA**, uses random Fourier features (Lopez-Paz et al., 2014) and the second implementation, denoted **NKCCA**, uses the Nyström approximation (Williams & Seeger, 2001). As described in Sec. 3.2, in FKCCA/NKCCA we transform the original inputs to an M-dimensional feature space where the inner products between samples approximate the kernel similarities (Yang et al., 2012). We apply linear CCA to the transformed inputs to obtain the approximate KCCA solution.

4.1. Noisy MNIST digits

In this task, we generate two-view data using the MNIST dataset (LeCun et al., 1998), which consists of 28×28 grayscale digit images, with $60K/10K$ images for training/testing. We generate a more challenging version of the dataset as follows (see Fig. 2 for examples). We first rescale the pixel values to $[0, 1]$. We then randomly rotate the images at angles uniformly sampled from $[-\pi/4, \pi/4]$ and the resulting images are used as view 1 inputs. For each view 1 image, we randomly select an image of the same identity (0-9) from the original dataset, add independent random noise uniformly sampled from $[0, 1]$ to each pixel, and truncate the pixel final values to $[0, 1]$ to obtain the corresponding view 2 sample. The original training set is further split into training/tuning sets of size $50K/10K$.

Since, given the digit identity, observing a view 2 image does not provide any information about the corresponding view 1 image, a good multi-view learning algorithm should be able to extract features that disregard the noise. We measure the class separation in the learned feature spaces by clustering the projected view 1 inputs into 10 clusters and evaluating how well the clusters agree with ground-truth labels. We use spectral clustering (Ng et al., 2002) so as to account for possibly non-convex cluster shapes. Specifically, we first build a k-nearest-neighbor graph on the projected view 1 tuning/test samples with a binary weighting scheme (edges connecting neighboring samples have a constant weight of 1), then embed these samples in $\mathbb{R}^{10}$ using eigenvectors of the normalized graph Laplacian, and finally run K-means in the embedding to obtain a hard partition of the samples. In the last step, K-means is run 20 times with random initialization and the run with the best K-means objective is used. The size of the neighborhood graph k is selected from $\{5, 10, 20, 30, 50\}$ using the tuning set. We measure clustering performance with two criteria, clustering accuracy (ACC) and normalized mutual information (NMI) (Cai et al., 2005).

Table 1. Performance of several representation learning methods on the test set of noisy MNIST digits. Performance measures are clustering accuracy (ACC), normalized mutual information (NMI) of clustering, and classification error rates of a linear SVM on the projections. The selected feature dimensionality L is given in parentheses. Results are averaged over 5 random seeds.

Method	ACC (%)	NMI (%)	Error (%)
Baseline	47.0	50.6	13.1
CCA ($L = 10$)	72.9	56.0	19.6
SplitAE ($L = 10$)	64.0	69.0	11.9
CorrAE ($L = 10$)	65.5	67.2	12.9
DistAE ($L = 20$)	53.5	60.2	16.0
FKCCA($L = 10$)	94.7	87.3	5.1
NKCCA($L = 10$)	95.1	88.3	4.5
DCCA ($L = 10$)	**97.0**	**92.0**	**2.9**
DCCAE ($L = 10$)	**97.5**	**93.4**	**2.2**

Each algorithm has hyperparameters that are selected using the tuning set. The final dimensionality L is selected from $\{5, 10, 20, 30, 50\}$. For CCA, the regularization parameters r_x/r_y are selected via grid search. For KCCAs, we fix r_x/r_y at a small positive value of 10^{-4} (as suggested by Lopez-Paz et al. (2014), FKCCA is robust to r_x/r_y), do grid search for the Gaussian kernel width for each view at rank $M = 5,000$, and then test with $M = 20,000$. For DNN-based models, feature mappings $(\mathbf{f}, \mathbf{g})$ are implemented by networks of 3 hidden layers, each of $1,024$ sigmoid units, and a linear output layer of L units; reconstruction mappings $(\mathbf{p}, \mathbf{q})$ are implemented by networks of 3 hidden layers, each of $1,024$ sigmoid units, and an output layer of 784 sigmoid units. We fix $r_x = r_y = 10^{-4}$ for DCCA and DCCAE. For SplitAE/CorrAE/DCCAE/DistAE we select the trade-off parameter λ via grid search. The two networks $(\mathbf{f}, \mathbf{p})$ are

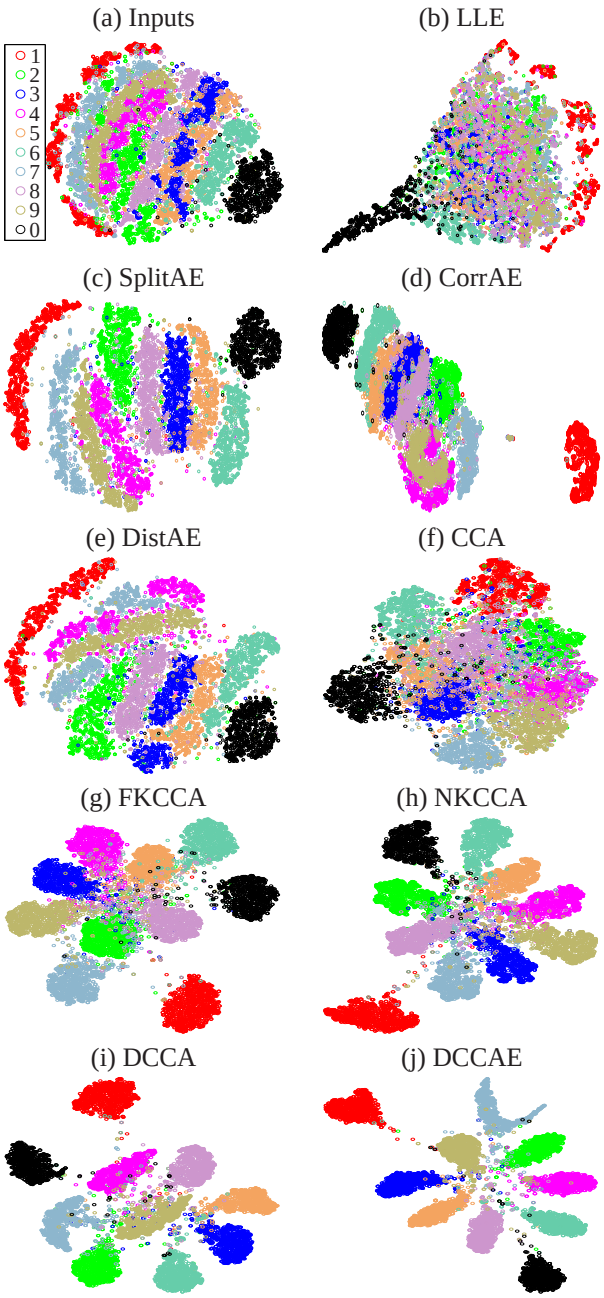


Figure 3. t-SNE embedding of the projected test set of noisy MNIST digits using different methods. Each sample is denoted by a marker located at its coordinates of embedding and color coded by its label. Neither the feature learning algorithms nor t-SNE use the class information.

pre-trained in a layerwise manner using restricted Boltzmann machines (Hinton & Salakhutdinov, 2006) and similarly for $(\mathbf{g}, \mathbf{q})$ with inputs from the corresponding view. For DNN-based models, we use SGD for optimization with minibatch size, learning rate and momentum tuned on the tuning set. A small weight decay parameter of 10^{-4} is used for all layers. We monitor the objective on the tuning set for early stopping. For each algorithm, we select the model with best ACC on the tuning set, and report its results on the test set. The ACC and NMI results (in percentage) for each algorithm are given in Table 1. As a baseline, we also cluster the original 784-dimensional view 1 images.

All of the multi-view feature learning algorithms achieve some improvement over the baseline. The nonlinear CCA algorithms perform similarly to each other and significantly better than SplitAE/CorrAE/DistAE. We also qualitatively investigate the features by embedding the projected features in 2D using t-SNE (van der Maaten & Hinton, 2008); the resulting visualizations are given in Fig. 3. Overall, the class separation in the visualizations qualitatively agrees with the relative clustering performances in Table 1.

In the embedding of input images (Fig. 3 (a)), samples of each digit form an approximately one dimensional, stripe-shaped manifold, and the degree of freedom along each manifold corresponds roughly to the variation in rotation angle (see supplementary material for embedding with images). This degree of freedom does not change the identity of the image, which is common to both views. Projections by SplitAE/CorrAE/DistAE do achieve somewhat better separation for some classes, but the unwanted rotation variation is still prominent in the embeddings. On the other hand, without using any label information and with only paired noisy images, the nonlinear CCA algorithms manage to map digits of the same identity to similar locations while supressing the rotational variation and separating images of different identities (linear CCA also approximates the same behavior, but fails to separate the classes, presumably because the input variations are too complex to be captured by linear mappings). Overall, DCCAE gives the cleanest embedding, with different digits pushed far apart.

The different behaviour of CCA-based methods from SplitAE/CorrAE/DistAE suggests two things. First, when the inputs are noisy, reconstructing the inputs faithfully may still lead to unwanted degrees of freedom in the features (DCCAE tends to select quite small trade-off parameter $\lambda = 10^{-3}$ or 10^{-2}, further supporting that it is not necessary to minimize reconstruction error). Second, the hard CCA constraints, which enforce uncorrelatedness between different feature dimensions, appear essential; these constraints are the difference between DCCAE and CorrAE/DistAE. However, the constraints without the multi-view objective are insufficient. To see this, we also visualize a 10-dimensional locally linear embedding (LLE, Roweis & Saul, 2000) of the test images in Fig. 3 (b). LLE satisfies the same un-correlatedness constraints as in CCA-based methods, but without access to the second view, it does not separate the classes as nicely.

In view of the embeddings in Fig. 3, one would expect that a simple linear classifier can achieve high accuracy on DCCA/DCCAE projections. We train one-versus-one lin-

ear SVMs (Chang & Lin, 2011) on the projected training set (now using the ground truth labels), and test on the projected test set, while using the projected tuning set for selecting the SVM hyperparameter (the penalty parameter for hinge loss). Test error rates on the optimal embedding of each algorithm (with highest ACC) are provided in Table 1 (last column). These error rates agree with the clustering results. Multi-view feature learning makes classification much easier on this task: Instead of using a heavily nonlinear classifier on the original inputs, a very simple linear classifier that can be trained efficiently on low-dimensional projections already achieves high accuracy.

4.2. Acoustic-articulatory data for speech recognition

We next experiment with the Wisconsin X-Ray Micro-Beam (XRMB) corpus (Westbury, 1994) of simultaneously recorded speech and articulatory measurements from 47 American English speakers. Multi-view feature learning via CCA/KCCA has previously been shown to improve phonetic recognition performance when tested on audio alone (Arora & Livescu, 2013; Wang et al., 2015).

We follow the setup of Wang et al. (2015) and use learned features for speaker-independent phonetic recognition. Inputs to multi-view feature learning are acoustic features (39D features consisting of mel frequency cepstral coefficients (MFCCs) and their first and second derivatives) and articulatory features (horizontal/vertical displacement of 8 pellets attached to several parts of the vocal tract) concatenated over a 7-frame window around each frame, giving 273D acoustic inputs and 112D articulatory inputs for each view.

We split the XRMB speakers into disjoint sets of 35/8/2/2 speakers for feature learning/recognizer training/tuning/testing. The 35 speakers for feature learning are fixed; the remaining 12 are used in a 6-fold experiment (recognizer training on 4 2-speaker folds, tuning on 1 fold, and testing on the last fold). Each speaker has roughly $50K$ frames, giving 1.43M multi-view training frames. We remove the per-speaker mean and variance of the articulatory measurements for each training speaker. All of the learned feature types are used in a "tandem" approach (Hermansky et al., 2000), i.e., they are appended to the original 39D features and used in a standard hidden Markov model (HMM)-based recognizer with Gaussian mixture observation distributions. The baseline recognizer uses the original MFCC features. The recognizer has one 3-state left-to-right HMM per phone and the same language model as in Wang et al. (2015)

For each fold, we select the hyperparameters based on recognition accuracy on the tuning set. As before, models based on neural networks are trained via SGD, with no pretraining and with optimization parameters tuned by grid search. A small weight decay parameter of 5×10^{-4} is used for all layers. For each algorithm, the dimensionality L is tuned over $\{30, 50, 70\}$. For DNN-based models, we use hidden layers of $1\,500$ ReLUs. For DCCA, we tune the network depths (up to 3 nonlinear hidden layers) and find that in the best-performing architecture, $\mathbf{f}$ has 3 hidden ReLU layers followed by a linear output layer while $\mathbf{g}$ has only a linear output layer. For SplitAE/CorrAE/DistAE/DCCAE, the same encoder architecture as that of DCCA performs best, and we set the decoders to have symmetric architectures to the encoders (except for SplitAE which does not have an encoder $\mathbf{g}$ and its decoder $\mathbf{q}$ is linear). We fix $r_x = r_y = 10^{-4}$ for DCCAE (and FKCCA/NKCCA). The trade-off parameter λ is tuned for each algorithm by grid search.

Table 2. Mean and standard deviations of PERs over 6 folds obtained by each algorithm on the XRMB test speakers.

Method	Mean (std) PER (%)
Baseline	34.8 (4.5)
CCA	26.7 (5.0)
SplitAE	29.0 (4.7)
CorrAE	30.6 (4.8)
DistAE	33.2 (4.7)
FKCCA	26.0 (4.4)
NKCCA	26.6 (4.2)
DCCA	**24.8 (4.4)**
DCCAE	**24.5 (3.9)**

For FKCCA, we find it important to use a large number of random features M to get a competitive result, consistent with the findings of Huang et al. (2014) when using random Fourier features for speech data. We tune kernel widths at $M = 5,000$ with FKCCA, and test FKCCA with $M = 30,000$ (the largest M we could afford to obtain an exact SVD solution on a workstation with 32G main memory); we are not able to obtain results for NKCCA with $M = 30,000$ in 48 hours with our implementation, so we report its test performance at $M = 20,000$ with the optimal FKCCA hyper-parameters. Notice that FKCCA has about 14.6 million parameters (random Gaussian samples + projection matrices), which is more than the number of weight parameters in the largest DCCA model, so it is slower than DCCA for testing (cost of obtaining test features is linear in the number of parameters for both KCCA and DNNs).

Phone error rates (PERs) obtained by different feature learning algorithms are given in Table 2. We see the same pattern as on MNIST: nonlinear CCA-based algorithms outperform SplitAE/CorrAE/DistAE. Since the recognizer now is a nonlinear mapping (HMM), the performance of the linear CCA features is highly competitive. Again, DCCAE selects a relatively small $\lambda = 0.01$, indicating that the canonical correlation term is more important.

4.3. Multilingual data for word embeddings

In this task, we learn a vectorial representation of English words from pairs of English-German word embeddings. We follow the setup of Faruqui & Dyer (2014) and Lu et al. (2015), and use as inputs 640-dimensional monolingual word vectors trained via latent semantic analysis on the WMT 2011 monolingual news corpora and use the same $36K$ English-German word pairs for multi-view learning. The learned mappings are applied to the original English word embeddings ($180K$ words) and the projections are used for evaluation. We evaluate on the bigram similarity dataset of Mitchell & Lapata (2010), using the adjective-noun (AN) and verb-object (VN) subsets, and tuning and test splits (of size 649/1,972) for each subset (we exclude the noun-noun subset as it is observed by Lu et al. (2015) that the NN human annotations often reflect "topical" rather than "functional" similarity). We simply add the projections of the two words in each bigram to obtain an L-dimensional representation of the bigram, as done in prior work (Blacoe & Lapata, 2012; Lu et al., 2015). We compute the cosine similarity between the two vectors of each bigram pair, order the pairs by similarity, and report the Spearman's correlation (ρ) between the model's ranking and human rankings.

We fix the feature dimensionality at $L = 384$; other hyperparameters are tuned as in previous experiments. DNN-based models use ReLU hidden layers of width $1,280$. A small weight decay parameter of 10^{-4} is used for all layers. We use two ReLU hidden layers for encoders ($\mathbf{f}$ and $\mathbf{g}$), and try both linear and nonlinear networks with two hidden layers for decoders ($\mathbf{p}$ and $\mathbf{q}$). FKCCA/NKCCA are tested with $M = 20,000$ using kernel widths tuned at $M = 4,000$. We fix $r_x = r_y = 10^{-4}$ for nonlinear CCAs.

For each algorithm, we select the model with the highest Spearman's correlation on the 649 tuning bigram pairs, and we report its performance on the 1,972 test pairs in Table 3 (our baseline and DCCA results are different from that of Lu et al. (2015) due to a different normalization and better tuning). Unlike MNIST and XRMB, it is important for the features to reconstruct the input monolingual word embeddings well, as can be seen from the superior performance of SplitAE over FKCCA/NKCCA/DCCA. This implies there is useful information in the original inputs that is not correlated across views. However, DCCAE still performs the best on the AN task, in this case using a relatively large λ.

Table 3. Spearman's correlation (ρ) for bigram similarities.

Method	AN	VN	Avg.
Baseline	45.0	39.1	42.1
CCA	46.6	37.7	42.2
SplitAE	47.0	**45.0**	46.0
CorrAE	43.0	42.0	42.5
DistAE	43.6	39.4	41.5
FKCCA	46.4	42.9	44.7
NKCCA	44.3	39.5	41.9
DCCA	48.5	42.5	45.5
DCCAE	**49.1**	43.2	**46.2**

5. Discussion

We have explored several approaches in the space of DNN-based mutli-view representation learning. We have found that on several tasks, CCA-based models outperform autoencoder-based models (SplitAE) and models based on between-view squared distance (DistAE) or correlation (CorrAE) instead of canonical correlation. The best overall performer is a new DCCA extension introduced here, deep canonically correlated autoencoders (DCCAE).

In light of the empirical results, it is interesting to consider again the main features of each type of objective and corresponding constraints. Autoencoder-based approaches are based on the idea that the learned features should be able to accurately reconstruct the inputs (in the case of multi-view learning, the inputs in both views). The CCA objective, on the other hand, focuses on how well each view's representation predicts the other's, ignoring the ability to reconstruct each view. CCA is expected to perform well when the two views are uncorrelated given the class label (Chaudhuri et al., 2009). The noisy MNIST dataset used here simulates exactly this scenario, and indeed this is the task where CCA outperforms other objectives by the largest margins. Even in the other tasks, however, there is often no significant advantage to being able to reconstruct the inputs faithfully.

The constraints in the various methods also have an important effect. The performance difference between DCCA and CorrAE demonstrates that uncorrelatedness between learned dimensions is important. On the other hand, the stronger DCCA constraint may still not be sufficiently strong; an even better constraint may be to require the learned dimensions to be independent (or approximately so), and this is an interesting avenue for future work. Another future direction is to compare DNN-based models with models based on deep Boltzmann machines (Srivastava & Salakhutdinov, 2014; Sohn et al., 2014) and noise-constrastive learning criteria (Gutmann & Hyvärinen, 2012).

Acknowledgments

This research was supported by NSF grant IIS-1321015. The opinions expressed in this work are those of the authors and do not necessarily reflect the views of the funding agency. The Tesla K40 GPUs used for this research were donated by NVIDIA Corporation. We thank Geoff Hinton for helpful discussion.

References

Akaho, Shotaro. A kernel method for canonical correlation analysis. In *Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*, 2001.

Andrew, Galen, Arora, Raman, Bilmes, Jeff, and Livescu, Karen. Deep canonical correlation analysis. In *ICML*, pp. 1247–1255, 2013.

Arora, Raman and Livescu, Karen. Kernel CCA for multi-view learning of acoustic features using articulatory measurements. In *Symposium on Machine Learning in Speech and Language Processing (MLSLP)*, 2012.

Arora, Raman and Livescu, Karen. Multi-view CCA-based acoustic features for phonetic recognition across speakers and domains. In *ICASSP*, 2013.

Bach, Francis R. and Jordan, Michael I. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

Becker, Suzanna and Hinton, Geoffrey E. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355:161–163, 1992.

Bickel, Steffen and Scheffer, Tobias. Multi-view clustering. In *Proc. of the 4th IEEE Int. Conf. Data Mining (ICDM'04)*, pp. 19–26, 2004.

Bie, Tijl De and Moor, Bart De. On the regularization of canonical correlation analysis. Int. Sympos. ICA and BSS, 2003.

Blacoe, William and Lapata, Mirella. A comparison of vector-based representations for semantic composition. In *EMNLP*, pp. 546–556, 2012.

Blaschko, Mathew B. and Lampert, Christoph H. Correlational spectral clustering. In *CVPR*, pp. 1–8, 2008.

Borga, Magnus. Canonical correlation: A tutorial. 2001.

Cai, Deng, He, Xiaofei, and Han, Jiawei. Document clustering using Locality Preserving Indexing. *IEEE Trans. Knowledge and Data Engineering*, 17(12):1624–1637, 2005.

Chandar, Sarath, Lauly, Stanislas, Larochelle, Hugo, Khapra, Mitesh M., Ravindran, Balaraman, Raykar, Vikas, and Saha, Amrita. An autoencoder approach to learning bilingual word representations. In *NIPS*, pp. 1853–1861, 2014.

Chang, Chih-Chung and Lin, Chih-Jen. LIBSVM: A library for support vector machines. *ACM Trans. Intelligent Systems and Technology*, 2(3):27, 2011.

Chaudhuri, Kamalika, Kakade, Sham M., Livescu, Karen, and Sridharan, Karthik. Multi-view clustering via canonical correlation analysis. In *ICML*, pp. 129–136, 2009.

Chechik, Gal, Globerson, Amir, Tishby, Naftali, and Weiss, Yair. Information bottleneck for Gaussian variables. *Journal of Machine Learning Research*, 6:165–188, 2005.

Dhillon, Paramveer, Foster, Dean, and Ungar, Lyle. Multi-view learning of word embeddings via CCA. In *NIPS*, pp. 199–207, 2011.

Faruqui, Manaal and Dyer, Chris. Improving vector space word representations using multilingual correlation. In *Proceedings of European Chapter of the Association for Computational Linguistics*, 2014.

Foster, Dean P., Johnson, Rie, Kakade, Sham M., and Zhang, Tong. Multi-view dimensionality reduction via canonical correlation analysis. Technical report, 2009.

Gutmann, Michael and Hyvärinen, Aapo. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13:307–361, 2012.

Haghighi, Aria, Liang, Percy, Berg-Kirkpatrick, Taylor, and Klein, Dan. Learning bilingual lexicons from monolingual corpora. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2008)*, pp. 771–779, 2008.

Hardoon, David R., Szedmak, Sandor, and Shawe-Taylor, John. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

Hermansky, Hynek, Ellis, Daniel P. W., and Sharma, Sangita. Tandem connectionist feature extraction for conventional HMM systems. In *ICASSP*, pp. 1635–1638, 2000.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science*, 313 (5786):504–507, 2006.

Hodosh, Micah, Young, Peter, and Hockenmaier, Julia. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

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

Hsieh, W. W. Nonlinear canonical correlation analysis by neural networks. *Neural Networks*, 13(10):1095–1105, 2000.

Huang, Po-Sen, Avron, Haim, Sainath, Tara, Sindhwani, Vikas, and Ramabhadran, Bhuvana. Kernel methods match deep neural networks on TIMIT: Scalable learning in high-dimensional random Fourier spaces. In *ICASSP*, pp. 205–209, 2014.

Kakade, Sham M. and Foster, Dean P. Multi-view regression via canonical correlation analysis. In *COLT*, pp. 82–96, 2007.

Kidron, Einat, Schechner, Yoav Y., and Elad, Michael. Pixels that sound. In *CVPR*, pp. 88–95, 2005.

Kim, Jungi, Nam, Jinseok, and Gurevych, Iryna. Learning semantics with deep belief network for cross-language information retrieval. In *COLING*, pp. 579–588, 2012.

Lai, Pei Ling and Fyfe, Colin. A neural implementation of canonical correlation analysis. *Neural Networks*, 12(10): 1391–1397, 1999.

Lai, Pei Ling and Fyfe, Colin. Kernel and nonlinear canonical correlation analysis. *Int. J. Neural Syst.*, 10(5):365–377, 2000.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

Lopez-Paz, David, Sra, Suvrit, Smola, Alex, Ghahramani, Zoubin, and Schoelkopf, Bernhard. Randomized nonlinear component analysis. In *ICML*, pp. 1359–1367, 2014.

Lu, Ang, Wang, Weiran, Bansal, Mohit, Gimpel, Kevin, and Livescu, Karen. Deep multilingual correlation for improved word embeddings. In *NAACL-HLT*, 2015.

Lu, Yichao and Foster, Dean P. Large scale canonical correlation analysis with iterative least squares. In *NIPS*, pp. 91–99, 2014.

Melzer, Thomas, Reiter, Michael, and Bischof, Horst. Nonlinear feature extraction using generalized canonical correlation analysis. In *Proc. of the 11th Int. Conf. Artificial Neural Networks (ICANN'01)*, pp. 353–360, 2001.

Mitchell, Jeff and Lapata, Mirella. Composition in distributional models of semantics. *Cognitive Science*, 34(8): 1388–1429, 2010.

Ng, Andrew Y., Jordan, Michael I., and Weiss, Yair. On spectral clustering: Analysis and an algorithm. In *NIPS*, pp. 849–856, 2002.

Ngiam, Jiquan, Khosla, Aditya, Kim, Mingyu, Nam, Juhan, Lee, Honglak, and Ng, Andrew. Multimodal deep learning. In *ICML*, pp. 689–696, 2011.

Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D. GloVe: Global vectors for word representation. In *EMNLP*, 2014.

Roweis, Sam T. and Saul, Lawrence K. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Schölkopf, Bernhard and Smola, Alexander J. *Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

Socher, Richard and Li, Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *CVPR*, pp. 966–973, 2010.

Sohn, Kihyuk, Shang, Wenling, and Lee, Honglak. Improved multimodal deep learning with variation of information. In *NIPS*, pp. 2141–2149, 2014.

Srivastava, Nitish and Salakhutdinov, Ruslan. Multimodal learning with deep boltzmann machines. *Journal of Machine Learning Research*, 15:2949–2980, 2014.

Tishby, Naftali, Pereira, Fernando, and Bialek, William. The information bottleneck method. In *Proc. 37th Annual Allerton Conference on Communication, Control, and Computing*, pp. 368–377, 1999.

van der Maaten, Laurens J. P. and Hinton, Geoffrey E. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Vincent, Pascal, Larochelle, Hugo, Lajoie, Isabelle, Bengio, Yoshua, and Manzagol, Pierre-Antoine. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, 2010.

Vinokourov, Alexei, Cristianini, Nello, and Shawe-Taylor, John. Inferring a semantic representation of text via cross-language correlation analysis. In *NIPS*, pp. 1497–1504, 2003.

Wang, Weiran, Arora, Raman, Livescu, Karen, and Bilmes, Jeff. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *ICASSP*, 2015.

Westbury, John R. *X-Ray Microbeam Speech Production Database User's Handbook Version 1.0*, 1994.

Williams, Christopher K. I. and Seeger, Matthias. Using the Nyström method to speed up kernel machines. In *NIPS*, pp. 682–688, 2001.

Yang, Tianbao, Li, Yu-Feng, Mahdavi, Mehrdad, Jin, Rong, and Zhou, Zhi-Hua. Nyström method vs random Fourier features: A theoretical and empirical comparison. In *NIPS*, pp. 476–484, 2012.