Correlational Neural Networks[*]

Sarath Chandar[1], **Mitesh M Khapra**[2], **Hugo Larochelle**[3], **Balaraman Ravindran**[4]
[1]University of Montreal apsarathchandar@gmail.com
[2]IBM Research India, [3]University of Sherbrooke, [4]IIT Madras

Abstract

Common Representation Learning (CRL), wherein different descriptions (or views) of the data are embedded in a common subspace, is one way of achieving Transfer Learning. Two popular paradigms here are Canonical Correlation Analysis (CCA) based approaches and Autoencoder (AE) based approaches. Each of these approaches has its own advantages and disadvantages. For example, while CCA based approaches outperform AE based approaches for the task of transfer learning, they are not as scalable as the latter. In this work we propose an AE based approach called Correlational Neural Network (CorrNet), that explicitly maximizes correlation among the views when projected to the common subspace. Through experiments, we demonstrate that the proposed CorrNet is better than the above mentioned approaches with respect to its ability to learn correlated common representations that are useful for Transfer Learning.

1 Introduction

In several real world applications, the data contains more than one view. For example, a movie clip has three views (of different modalities) : audio, video and text/subtitles. Recently there has been a lot of interest in learning a common representation for multiple views of the data [12, 8, 2, 3, 1, 6, 15] which can be useful in several downstream applications when some of the views are missing. As an example, consider the case where a profanity detector trained on movie subtitles needs to detect profanities in a movie clip for which only video is available. If a common representation is available for the different views, then such detectors/classifiers can be trained by computing this common representation from the relevant view (subtitles, in the above example). At test time, a common representation can again be computed from the available view (video, in this case) and this representation can be fed to the trained model for prediction. This way transfer learning is achieved using the common representation.

Canonical Correlation Analysis (CCA) [7] is a commonly used tool for learning such common representations for two-view data [14, 5]. By definition, CCA aims to produce correlated common representations but, it suffers from some drawbacks. First, it is not easily scalable to very large datasets. Of course, there are some approaches which try to make CCA scalable (for example, [11]), but such scalability comes at the cost of performance. CCA cannot benefit from additional non-parallel, single-view data. This puts it at a severe disadvantage in several real world situations, where in addition to some parallel two-view data, abundant single view data is available for one or both views.

Recently, Multimodal Autoencoders (MAEs) [12] have been proposed to learn a common representation for two views/modalities. The idea in MAE is to train an autoencoder to perform two kinds of reconstruction. Given any one view, the model learns both self-reconstruction and cross-reconstruction (reconstruction of the other view). This makes the representations learnt to be predictive of each other. However, it should be noticed that the MAE does not get any explicit learning

[*] Extendend version of this paper, which is available in arxiv (http://arxiv.org/abs/1504.07225), will appear in Neural Computation 2015 [4].

signal encouraging it to share the capacity of its common hidden layer between the views. In other words, it could develop units whose activation is dominated by a single view. This makes the MAE not suitable for transfer learning, since the views are not guaranteed to be projected to a common subspace. This is indeed verified by the results reported in [12] where they show that CCA performs better than deep MAE for the task of transfer learning.

These two approaches have complementary characteristics. On one hand, we have CCA and its variants which aim to produce correlated common representations but lack reconstruction capabilities. On the other hand, we have MAE which aims to do self-reconstruction and cross-reconstruction but does not guarantee correlated common representations. In this paper, we propose Correlational Neural Network (CorrNet) as a method for learning common representations which combines the advantages of the two approaches described above. The main characteristics of the proposed method can be summarized as follows:

- It allows for self/cross reconstruction. Thus, unlike CCA (and like MAE) it has predictive capabilities. This can be useful in applications where a missing view needs to be reconstructed from an existing view.
- Unlike MAE (and like CCA) the training objective used in CorrNet ensures that the common representations of the two views are correlated. This is particularly useful in applications where we need to match items from one view to their corresponding items in the other view.
- CorrNet can be trained using Gradient Descent based optimization methods. Particularly, when dealing with large high dimensional data, one can use Stochastic Gradient Descent with mini-batches. Thus, unlike CCA (and like MAE) it is easy to scale CorrNet.
- The procedure used for training CorrNet can be easily modified to benefit from additional single view data. This makes CorrNet useful in many real world applications where additional single view data is available.

2 Correlational Neural Network

CorrNet is a neural network architecture which contains three layers: an input layer, a hidden layer and an output layer. Just as in a conventional single view autoencoder, the input and output layers have the same number of units, whereas the hidden layer can have a different number of units. For illustration, we consider a two-view input $\mathbf{z} = (\mathbf{x}, \mathbf{y})$. For all the discussions, $[\mathbf{x}, \mathbf{y}]$ denotes a concatenated vector of size $d_1 + d_2$.

Given $\mathbf{z} = (\mathbf{x}, \mathbf{y})$, the hidden layer computes an encoded representation as follows:

$$h(\mathbf{z}) = f(\mathbf{W}\mathbf{x} + \mathbf{V}\mathbf{y} + \mathbf{b})$$

where $\mathbf{W}$ is a $k \times d_1$ projection matrix, $\mathbf{V}$ is a $k \times d_2$ projection matrix and $\mathbf{b}$ is a $k \times 1$ bias vector. Function f can be any non-linear activation function, for example *sigmoid* or *tanh*. The output layer then tries to reconstruct $\mathbf{z}$ from this hidden representation by computing

$$\mathbf{z}' = g([\mathbf{W}'h(\mathbf{z}), \mathbf{V}'h(\mathbf{z})] + \mathbf{b}')$$

where $\mathbf{W}'$ is a $d_1 \times k$ reconstruction matrix, $\mathbf{V}'$ is a $d_2 \times k$ reconstruction matrix and $\mathbf{b}'$ is a $(d_1 + d_2) \times 1$ output bias vector. Vector $\mathbf{z}'$ is the reconstruction of $\mathbf{z}$. Function g can be any activation function. This architecture is illustrated in Figure 1. The parameters of the model are $\theta = \{\mathbf{W}, \mathbf{V}, \mathbf{W}', \mathbf{V}', \mathbf{b}, \mathbf{b}'\}$. The model is trained by minimizing the following objective function:

$$\mathcal{J}_{\mathcal{Z}}(\theta) = \sum_{i=1}^{N} (L(\mathbf{z}_i, g(h(\mathbf{z}_i))) + L(\mathbf{z}_i, g(h(\mathbf{x}_i))) + L(\mathbf{z}_i, g(h(\mathbf{y}_i)))) - \lambda \, \mathrm{corr}(h(X), h(Y))$$

$$\mathrm{corr}(h(X), h(Y)) = \frac{\sum_{i=1}^{N}(h(\mathbf{x}_i) - \overline{h(X)})(h(\mathbf{y}_i) - \overline{h(Y)})}{\sqrt{\sum_{i=1}^{N}(h(\mathbf{x}_i) - \overline{h(X)})^2 \sum_{i=1}^{N}(h(\mathbf{y}_i) - \overline{h(Y)})^2}}$$

where L is the reconstruction error, λ is the scaling parameter to scale the fourth term with respect to the remaining three terms, $\overline{h(X)}$ is the mean vector for the hidden representations of the first view and $\overline{h(Y)}$ is the mean vector for the hidden representations of the second view. If all dimensions in the input data take binary values then we use cross-entropy as the reconstruction error otherwise we use squared error loss as the reconstruction error.

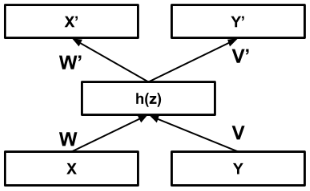

Figure 1: Correlational Neural Network

In words, the objective function decomposes as follows. The first term is the usual autoencoder objective function which helps in learning meaningful hidden representations. The second term ensures that both views can be predicted from the shared representation of the first view alone. The third term ensures that both views can be predicted from the shared representation of the second view alone. The fourth term interacts with the other objectives to make sure that the hidden representations are highly correlated, so as to encourage the hidden units of the representation to be shared between views.

We can use stochastic gradient descent (SGD) to find the optimal parameters. For all our experiments, we used mini-batch SGD. The fourth term in the objective function is then approximated based on the statistics of a minibatch. The model has four hyperparameters: (i) the number of units in its hidden layer, (ii) λ, (iii) mini-batch size, and (iv) the SGD learning rate.

Once the parameters are learned, we can use the CorrNet to compute representations of views that can potentially generalize across views. Specifically, given a new data instance for which only one view is available, we can compute its corresponding representation ($h(\mathbf{x})$ if $\mathbf{x}$ is observed or $h(\mathbf{y})$ if $\mathbf{y}$ is observed) and use it as the new data representation.

In practice, it is often the case that we have abundant single view data and comparatively little two-view data. For example, in the context of text documents from two languages (X and Y), typically the amount of monolingual (single view) data available in each language is much larger than parallel (two-view) data available between X and Y. Given the abundance of such single view data, it is desirable to exploit it in order to improve the learned representation. CorrNet can achieve this, by using the single view data to improve the self-reconstruction error as explained below.

Consider the case where, in addition to the data $\mathcal{Z} = \{(\mathbf{z}_i)\}_{i=1}^{N}$ = $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, we also have access to the single view data $\mathcal{X} = \{(\mathbf{x}_i)\}_{i=N+1}^{N_1}$ and $\mathcal{Y} = \{(\mathbf{y}_i)\}_{i=N+1}^{N_2}$. Now, during training, in addition to using $\mathcal{Z}$ as explained before, we also use $\mathcal{X}$ and $\mathcal{Y}$ by suitably modifying the objective function so that it matches that of a conventional autoencoder. Specifically, when we have only $\mathbf{x}_i$, then we could try to minimize

$$\mathcal{J}_{\mathcal{X}}(\theta) = \sum_{i=N+1}^{N_1} L(\mathbf{x}_i, g(h(\mathbf{x}_i)))$$

and similarly for $\mathbf{y}_i$.

3 Transfer Learning Experiment

To demonstrate transfer learning, we take the task of predicting digits from only one half of the image. We first learn a common representation for the two views using 50,000 images from the MNIST training data. For each training instance, we take only one half of the image and compute its 50 dimensional common representation using one of the models described above. We then train a classifier using this representation. For each test instance, we consider only the other half of the image and compute its common representation. We then feed this representation to the classifier for prediction. We use the linear SVM implementation provided by [13] as the classifier for all our experiments. For all the models considered in this experiment, representation learning is done using 50,000 train images and the best hyperparameters are chosen using the 10,000 images from the validation set. With the chosen model, we report 5-fold cross validation accuracy using 10,000 images available in the standard test set of MNIST data. We report accuracy for two settings (i) Left to Right (training on left view, testing on right view) and (ii) Right to Left (training on right view, testing on left view).

Model	**Left to Right**	**Right to Left**
CCA	65.73	65.44
KCCA	68.1	75.71
MAE	64.14	68.88
CorrNet	**77.05**	**78.81**

Table 1: Transfer learning accuracy using the representations learned using different models on the MNIST dataset.

4 Transliteration Equivalence

In this section, we consider the task of determining transliteration equivalence of named entities wherein given a word u written using the script of language X and a word v written using the script of language Y the goal is to determine whether u and v are transliterations of each other. Several approaches have been proposed for this task and the one most related to our work is an approach which uses CCA for determining transliteration equivalence. We condider English-Hindi as the language pair for which transliteration equivalence needs to be determined. For learning common representations we used approximately 15,000 transliteration pairs from NEWS 2009 English-Hindi training set [10]. We represent each Hindi word as a bag of 2860 bigram characters. This forms the first view ($\mathbf{x}_i$). Similarly we represent each English word as a bag of 651 bigram characters. This forms the second view ($\mathbf{y}_i$). Each such pair $(\mathbf{x}_i, \mathbf{y}_i)$ then serves as one training instance for the CorrNet.

For testing we consider the standard NEWS 2010 transliteration mining test set [9]. This test set contains approximately 1000 Wikipedia English Hindi title pairs. The original task definition is as follows. For a given English title containing T_1 words and the corresponding Hindi title containing T_2 words identify all pairs which form a transliteration pair. Specifically, for each title pair, consider all $T_1 \times T_2$ word pairs and identify the correct transliteration pairs. In all, the test set contains 5468 word pairs out of which 982 are transliteration pairs. For every word pair $(\mathbf{x}_i, \mathbf{y}_i)$ we obtain a 50 dimensional common representation for $\mathbf{x}_i$ and $\mathbf{y}_i$ using the trained CorrNet. We then calculate the correlation between the representations of $\mathbf{x}_i$ and $\mathbf{y}_i$. If the correlation is above a threshold we mark the word pair as equivalent. This threshold is tuned using an additional 1000 pairs which were provided as training data for the NEWS 2010 transliteration mining task. As seen in Table 2 CorrNet clearly performs better than the other methods.

Model	F1-measure (%)
CCA	49.68
KCCA	42.36
MAE	72.75
CorrNet	**81.56**

Table 2: Performance on NEWS 2010 En-Hi Transliteration Mining Dataset

5 Conclusion

In this paper, we proposed Correlational Neural Networks as a method for learning common representations for two views of the data. The proposed model has the capability to reconstruct one view from the other and it ensures that the common representations learned for the two views are aligned and correlated. Its training procedure is also scalable. Further, the model can benefit from additional single view data, which is often available in many real world applications. Refer [4] for application of CorrNet for several cross language learning tasks.

References

[1] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. *ICML*, 2013.

[2] Sarath Chandar, Mitesh M. Khapra, Balaraman Ravindran, Vikas C. Raykar, and Amrita Saha. Multilingual deep learning. *NIPS Deep Learning Workshop*, 2013.

[3] Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh M Khapra, Balaraman Ravindran, Vikas Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word representations. In *Proceedings of NIPS*, 2014.

[4] Sarath Chandar, Mitesh M Khapra, Hugo Laorchelle, and Balaraman Ravindran. Correlational neural networks. *To appear in Neural Computation*, 2015.

[5] Paramveer Dhillon, Dean Foster, and Lyle. Ungar. Multi-view learning of word embeddings via cca. *In NIPS*, 2011.

[6] Karl Moritz Hermann and Phil Blunsom. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 58–68, 2014.

[7] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321 – 377, 1936.

[8] Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. Inducing Crosslingual Distributed Representations of Words. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2012.

[9] A Kumaran, Mitesh M. Khapra, and Haizhou Li. Report of news 2010 transliteration mining shared task. In *Proceedings of the 2010 Named Entities Workshop*, pages 21–28, Uppsala, Sweden, July 2010.

[10] Haizhou Li, A Kumaran, Min Zhang, and Vladimir Pervouvhine. Whitepaper of news 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 19–26, Suntec, Singapore, August 2009.

[11] Yichao Lu and Dean P. Foster. large scale canonical correlation analysis with iterative least squares. *In NIPS*, 2014.

[12] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and Ng. Andrew. Multimodal deep learning. *ICML*, 2011.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] Raghavendra Udupa and Mitesh M. Khapra. Transliteration equivalence using canonical correlation analysis. In *Proceedings of the 32nd European Conference on IR Research*, pages 75–86, 2010.

[15] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *ICML*, 2015.