Common Representation Learning Using Step-based Correlation Multi-Modal CNN

Gaurav Bhatt, Piyush Jha, and Balasubramanian Raman
Indian Institute of Technology Roorkee
Roorkee, India.
{gauravbhatt.cs.iitr,piyushnit15}@gmail.com, balarfma@iitr.ac.in

***Abstract*—Deep learning techniques have been successfully used in learning a common representation for multi-view data, wherein the different modalities are projected onto a common subspace. In a broader perspective, the techniques used to investigate common representation learning falls under the categories of canonical correlation-based approaches and autoencoder based approaches. In this paper, we investigate the performance of deep autoencoder based methods on multi-view data. We propose a novel step-based correlation multi-modal CNN (CorrMCNN) which reconstructs one view of the data given the other while increasing the interaction between the representations at each hidden layer or every intermediate step. Finally, we evaluate the performance of the proposed model on two benchmark datasets - MNIST and XRMB. Through extensive experiments, we find that the proposed model achieves better performance than the current state-of-the-art techniques on joint common representation learning and transfer learning tasks.**

***Index Terms*—common representation learning, multi-view data, transfer learning, deep learning.**

I. Introduction

Representation of data in multiple views can be seen in applications related to machine learning, computer vision, and natural language processing. At times it is beneficial to combine different modalities of data since an amalgamation of multiple views is likely to capture more meaningful information than a representation that is fit for only a specific modality. For example, consider the task of abstract scene recognition in a movie [1] with text annotations as labels. Movie data is comprised of video frames (images) along with audio. Here, images and audio are two different representations of same data with different representative features. Thus, combining these two modalities into a common subspace can help in the task of classification of abstract scenes from videos. Similarly, multi-view data have been used in audio + articulation [2], [3], images + text [4], [5], training transliterated corpora (bilingual data) [6], [7], [8], [9], or bridge autoencoders [10].

The techniques used for common representation learning (CRL) of multi-view data can be categorized into two categories - canonical based approaches and autoencoder based methods. Canonical correlation analysis (CCA) [11], [12] is matrix factorization method that maximizes the correlation between two views of data by projecting different modalities onto a common subspace. Variants of CCA include regularized CCA [13], Kernel-CCA (KCCA) [14], [15], [16], Nonparametric CCA (NCCA) [17], Deep-CCA (DCCA) [18], randomized non-linear component analysis (RCCA) [19] - a low rank approximation of KCCA, and Deep-Generalized-CCA (DGCCA) [20]. Although CCA-based methods provide a combined correlated representations, they suffer from scalability issues [9]. Another problem associated with CCA-based techniques is the fact that these methods tend to have poor performance for reconstruction of views.

Another broad category into which the CRL techniques can be divided is autoencoder (AE) based approaches. AE are deep neural networks that try to optimize two objective functions [21], [22]. The first objective is to find a compressed hidden representation of data in a low-dimensional vector space. The other objective is to reconstruct the original data from the compressed low-dimensional subspace. Multi-modal autoencoders (MAE) are two channeled AE that specifically performs two types of reconstructions [23]. The first is the self-reconstruction of view from itself, and the other is the cross-reconstruction where one view is reconstructed given the other. These reconstruction objectives provide MAE the ability to adapt towards transfer learning tasks as well. One recently proposed variant of MAE is Correlation Neural Networks (CorrNet) [9] which presents an improvement to MAE by introducing a correlation term in the objective function that tries to maximize the correlation between the hidden representations of different views. Some limitations of the CorrNet includes usage of the simple neural layer for encoding and decoding and using the final hidden representations in the correlation loss function.

Deep networks have grabbed the attention of many ever since the advent of state-of-the-art results using CNN networks. Apart from that, a lot of techniques have been worked upon to improve the classical CNN models. Regularization methods to reduce over-fitting, activation functions and modified convolution layers are some of them. Dropout technique [24] is a very commonly used stochastic regularization technique which is also being used in our model. For activation function, sigmoid functions are avoided as they have a problem of vanishing gradient when large networks are involved. Hence, we use the rectified linear unit (ReLU) [25] which is mathematically efficient. Batch normalization [26] has been used to increase the training rate providing us with better correlation values as compared to the previous papers. It is a stabilizing mechanism for training a neural network by scaling the output of hidden layers to zero norm and unit variance. This process of scaling reduces the change of distribution

between neurons throughout the network and helps to speed up the training process.

In this paper, we focus our attention on the improvement in objective function of CorrNet, thereby enhancing the learned joint representations and reconstruction of views as well. Specifically, main contributions of the paper are

1) We introduce convolution layers to the encoding phase of Correlation multi-modal CNN (CorrMCNN), and deconvolution is used in the decoding stage.
2) We use batch normalization in the intermediate layers of proposed model along with tied weights architecture.
3) Instead of using final hidden representations in the correlation loss, we enforce correlation computation at each intermediate layer. We further experiment with the reconstruction of hidden representation at every individual step.

The rest of the paper is organized as follows. In section II we discuss previous work related to our work which is followed by the discussion of CorrMCNN in Section III. The experimental details and results are shown in section IV and then our work is concluded in section V.

II. Related Work

Several methods have been used for generating the hidden representation of two views of data. This hidden representation can also be used to construct the other missing view when one of the views is provided as an input. CCA [11], [12] is the oldest model which is able to achieve this feat but has the drawbacks of scalability to large datasets. Scalability has been achieved by [27] but at the cost of performance. Also, CCA results in low-quality reconstruction when one view is reconstructed from the other. Finally, CCA suffers from a severe disadvantage of not being able to be used in several real world problems, as it cannot benefit from additional non-parallel, single-view data. Several variants of CCA has been developed ever since it's importance grew in the field of computer vision. These modifications include Regularized CCA [19], KCCA [14], [15] and a recently emerged low-rank approximation of KCCA known as RCCA [19]. However, CCA is restricted to linear projections, while KCCA works only on a fixed kernel.

Several deep learning models have been proposed that aim at solving the above problems. Deep canonical correlation analysis (DCCA) [18] works for pairs of inputs through two network pipelines and evaluates the results of each pipeline via the CCA loss. CorrNet [9] and deep canonically correlated autoencoders (DCCAE) [28] are an extended version of the concept of auto-encoder that take two input views and produce two output views. DCCAE is an extension of DCCA which takes into account self-reconstruction and correlation but doesn't consider cross-reconstruction. MAE [23] is another method used for CRL with architecture similar to CorrNet. The only difference exists between MAE and CorrNet is in their objective functions. Unlike MAE, CorrNet's objective function enforces the model to learn the correlated common representations as well. Also, CorrNet aims to minimize all the error at once, unlike its predecessor which adopts a stochastic version.

III. Technique Used

A. Convolution Auto-encoder

Convolution auto-encoder (CAE) [29] is novel idea for unsupervised feature learning that are a variant of convolution neural network (CNN). However, the difference lies in the fact that CNNs are usually referred to as supervised learning algorithms. Usually, an unsupervised pre-training is done greedily and with the help of layers, and after that, the weights are fine-tuned using back-propagation. One limitations of fully connected AEs and DAEs when used for multi-modal learning is that they both ignore the multi-dimensional image structure. This problem is not only concerned with ordinary sized inputs but further introduces extraneous parameters that force each feature to span the entire visual field. The main difference between CAEs and conventional AEs is that in CAEs the weights are shared everywhere in the input which preserves spatial locality. The variant of CAE that we use in this paper is shown in Figure 1.

In Figure 1 the input data is passed on to two channels which add convolution and max-pooling to obtain a lower dimensional representation of the data. We add dense neural layers after the max-pooling so that the interaction between the hidden representations can be maximized. We further add dropout as regularization to these intermediate dense neural layers. Finally, the output of the dense neural layers is used to obtain a joint common representation. To reconstruct the original input from the joint common representations, the projections are passed through deconvolution and up-sampling. Here, we also add a dense neural layer between the joint common representations and the deconvolution.

B. CorrMCNN

For learning a common representation of the two views of data, one of the views is to be reconstructed from the hidden representation. This task of reconstruction can be achieved by using a conventional auto-encoder. However, it is also required that one of the views can be predicted from the other view provided that both the representations are correlated. To achieve this functionality, multiple channeled autoencoders such as MAE and CorrNet are being used. The concept of correlation networks can be extended to allow for multiple hidden layers [9]. Multiple hidden layers ensure better interaction between the two views of the data as a result of passing the data through more non-linearity.

For training the proposed model, we target the following goals in our objective function:

- Minimize the self-reconstruction error.
- Minimize the cross reconstruction error at each intermediate step.
- Introduce batch-normalization at the intermediate dense neural layers.
- Maximize the correlation between the hidden representation of both views at each encoding step.

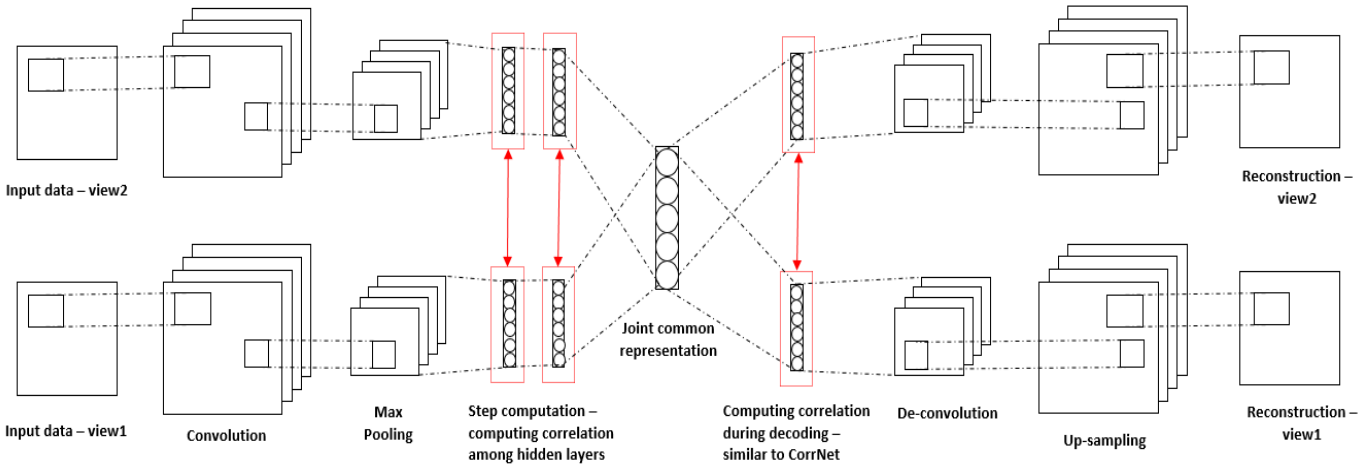

Fig. 1. Overview of the CorrMCNN. The bidirectional arrows shows the step correlation computation and cross-reconstructions at the intermediate steps.

Given the input as $z_i = \{x_i; y_i\}$, where z_i is the concatenated representation of input views x_i and y_i (corresponding to $view1$ and $view2$ in Figure 1), the self and cross-reconstruction losses are defined as

$$L_1 = \sum_{i=0}^{N} L(z_i, g(h(z_i)) \tag{1}$$

$$L_2 = \sum_{i=0}^{N} L(z_i, g(h(x_i)) \tag{2}$$

$$L_3 = \sum_{i=0}^{N} L(z_i, g(h(y_i)) \tag{3}$$

$$L_4 = \sum_{k=0}^{K} \sum_{i=0}^{N} L(h(x_i^k), h(y_i^k)) \tag{4}$$

$$L_5 = \sum_{i=0}^{N} L(g(h(x_i)), g(h(y_i)) \tag{5}$$

where, g, h are non-linearities generally taken as sigmoid or relu, $g(h(x_i^k))$ and $g(h(y_i^k))$ are the hidden representations at the k^{th} intermediate hidden layer (In Figure 1 the value of k is 2), and L is the mean square error function. Step computation (shown in Figure 1) is given by Equation 4. The reason for introducing step computation is the fact that the hidden representations share similarities and thus reconstruction of one hidden representation from the other helps the model in final reconstruction of views, improving interaction between layers as well. In losses L_2 and L_3 (used for cross reconstruction), x_i and y_i are computed using a 0-vector in place of the other view.

Finally, to maximize the interaction between the two views correlation loss is added as

$$L_6 = \lambda \; corr(h(X), h(Y)) \tag{6}$$

$$L_7 = \sum_{k=0}^{K} \lambda_k \; corr(h(X^k), h(Y^k)) \tag{7}$$

where $h(X)$ and $h(Y)$ are the projections from the combined model (projection from joint common representation in Figure 1), with X and Y being the representations of input view obtained after passing through convolution and pooling layers, and λ_k are the correlation regularization hyper-parameter used for each intermediate k^{th} encoding step (similarly λ is used in the decoding phase).

Equation 7 is the step-wise computation of correlation between the corresponding hidden views. CorrNet uses L_6 to increase the correlation among two views while CorrMCNN uses combined correlation losses computed at each intermediate hidden layer (In Figure 1 the value of k is 2 during encoding and 1 in decoding).

Finally, the CorrMCNN is optimized using the following objective function using $adam$ as optimizer

$$L(\theta) = \sum_{i=0}^{5} L_i - \sum_{j=6}^{7} L_j \tag{8}$$

where θ are the parameters of CorrMCNN. Here, we minimize the self-reconstruction and cross-reconstruction whereas the correlation between the views is maximized.

C. Batch Normalization

To maximize the correlation, the variance of every neuron's output must be increased. This can be implemented by introducing batch normalization (BN) [26] layers. These BN layers alleviate the loss due to variance by enforcing unit variance and eradicating the impact of weights of the hidden layers on the output's variance. This method allows us to achieve the same accuracy in fewer training steps. Apart from that, BN layers also allow high learning rates. Traditionally, high learning rates in deep neural networks resulted in vanishing gradient and exploding gradient problems, as well as the issue of getting struck at poor local minima. Batch Normalization proves out to be quite beneficial in such cases. We use BN layers in the dense neural layers during encoding and decoding.

IV. Experiments

A. Datasets

All the models mentioned in Section II are trained on MNIST dataset, and X-Ray Microbeam Speech data (XRMB) [30], and the sum correlation along with the transfer learning accuracy is calculated. MNIST handwritten digits dataset has 60,000 images of handwritten digits for training and 10,000 for testing. Each image is split vertically into two halves so as to obtain an image of 28 x 14 or in other words, 392 features for each view of data. A 50-dimensional joint common representation space has been employed in this case.

XRMB dataset contains simultaneous acoustic and articulatory recordings. The acoustic features are MFCCs [31] for the given frames, resulting in a 273-dimensional vector per unit time. On the other hand, articulatory data is represented as an 112-dimensional vector. For benchmarking, 40,000 samples are used for training, and 10,000 samples are used for testing purposes. XRMB dataset has a joint common representation of 112 dimensions.

B. Compared Methods

We compare CorrMCNN with both CCA based and AE based approaches. For canonical based approaches we perform comparisons with CCA [11], DCCA [18], KCCA [14], RCCA [19] and DCCAE [29]. Apart from CCA based approaches we have also compared CorrMCNN with recently proposed 2-way nets [32].[1]

For AE based approaches we perform comparisons with MAE [23], CorrNet-Org [9] and CorrNet-Mod [32] [2].

C. Experimental setup

We have used 2 different architectures of CorrMCNN - one with L_7 loss term and other without L_7 term:

1) CorrMCNN-arc1 - using convolution layers and 2 dense neural layers with k=2 during encoding. For decoding, one dense layer (k=1) is used for reconstruction with dropout of 50 %. L_7 loss term is excluded in this architecture with λ = 0.02 in L_6.
2) CorrMCNN-arc2 - using convolution layers and 2 dense neural layers with k=2 during encoding. For decoding we use deconvolution layers along with one dense layer (k=1) for reconstruction and L_7 loss term is included in this architecture. Values of λ_1=0.005 and λ_2=0.01 during encoding and for decoding λ = 0.005 in L_7 term. For L_6, λ = 0.02 is used.

Linear SVM implementation as mentioned in [33] is used for transfer learning along with a 5-fold cross validation on images from the MNIST half matching dataset.

D. Results

For evaluating the performance of the proposed model, we have used two evaluation metrics: total sum correlation and transfer learning (reconstruction of one view of MNIST dataset using the other). These results are shown in Table I and II respectively.

[1]Results of KCCA, MAE and CorrNet-Org on MNIST dataset are taken from [9]. Code for CorrNet-Org by [9] is available at :- https://github.com/apsarath/CorrNet

[2]CorrNet-Mod - Revised results of CorrNet on MNIST dataset are stated by [32]

Model	MNIST	XRMB
KCCA [15]	30.58	$N.A.$
DCCA [18]	39.7	92.9
MAE [23]	24.40	$N.A.$
RCCA [19]	44.5	104.5
DCCAE [29]	25.34	41.47
Reg-CCA [13]	28.0	16.90
CorrNet-Org [9]	47.21	81.54
CorrNet-Mod	48.07	95.01
2WayNet [32]	49.15	**110.18**
CorrMCNN-arc1	49.08	105.04
CorrMCNN-arc2	**49.33**	105.59

TABLE I
SUM CORRELATION CAPTURED IN THE JOINT COMMON REPRESENTATIONS LEARNED BY DIFFERENT MODELS ON MNIST AND XRMB DATASET.

Model	Left to Right	Right to Left
CCA [13]	65.73	65.44
KCCA [15]	68.1	75.71
DCCA [18]	70.06	72.43
MAE [23]	64.14	68.88
CorrNet-Org [9]	77.05	78.81
CorrNet-500-50 [9]	80.46	80.47
CorrMCNN-arc1	**90.27**	**91.16**
CorrMCNN-arc2	84.23	85.76
Single view	81.62	80.06

TABLE II
TRANSFER LEARNING ACCURACY USING THE LEARNED JOINT COMMON REPRESENTATIONS ON MNIST DATASET.

As shown in Table I, CorrMCNN achieves the highest sum correlation value for MNIST dataset while achieving the second best on XRMB dataset. The introduction of L_7 loss term that is the step based correlation achieves the highest correlation for the MNIST dataset. One possible reason for slightly lower performance of CorrMCNN on XRMB dataset is the fact that it consists of 1D MFCC features that are the compressed representation of original acoustic signals. Despite the 1D nature of XRMB, CorrMCNN achieves comparable performance on the task of learning correlated joint common representations.

The results of transfer learning task is shown is shown in Table II. Here, $Single\ view$ corresponds to the SVM trained and tested on the single view. Both architectures of CorrMCNN are able to achieve better performance than the current state-of-the-art techniques. The increment in the cross-reconstruction accuracy is more than 10% than the $Single\ view$ which acts as the baseline (ideal case) for this task. Here, CorrMCNN-arc1 performs better than its arc2 counterpart and the reason is the introduction of L_4 and L_5 cross-reconstruction loss terms. We observe that the introduction of L_7 term in the loss function slightly over-fits on the task of cross-reconstruction which is the reason for lower accuracy. Finally, as an illustration the results for reconstruction of either of the views is shown in Figure 2.

During experimentations we observed that the introduction

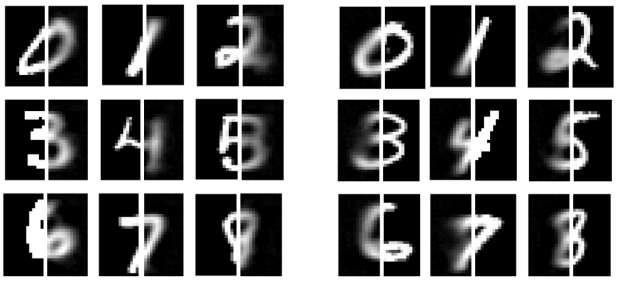

Fig. 2. Reconstruction of right view from left and left view from right by CorrMCNN on MNIST dataset.

of batch normalization helps the proposed model to converge faster giving high correlation values. The only downside of BN is the slight drop in reconstruction accuracy during transfer learning. Deconvolution and up-sampling helps CorrMCNN to increase the sum correlation. Finally, the introduction of L_4 and L_5 terms helps in improving the reconstruction accuracy, bolstering the overall cross-reconstruction.

V. Conclusion

In this paper, we proposed CorrMCNN which learns common representations for multi-view data. The introduction of step correlation terms as introduced in Section 3 (loss terms L_4, L_5 and L_7) helps to achieve a highly correlated common representation. The proposed model not only increases the interaction between the representations but helps in improvement of self and cross-reconstruction (shown in Table II). The results shown in Table I and Table II shows that the proposed model is able to achieve state-of-the-art performance on either of the benchmark datasets on common representation learning and transfer learning tasks.

On final remarks, we observed some limitations of AE and canonical correlation-based approaches when used on high dimensional data such as 3D images and video frames. These models show state-of-the-art performance on increasing the correlation between the two views but the cross reconstruction of views is not possible. We speculate that improving the cross reconstruction accuracy will have a positive effect on computing a joint common representation task as either of these are dependent on each other. We intend to work in this direction in the near future.

Acknowledgement

We would like to express our gratitude towards Institute Compute Center (ICC), Indian Institute of Technology Roorkee, for providing us with necessary resources for this paper.

References

[1] Z. Rasheed and M. Shah, "Scene detection in hollywood movies and tv shows," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2. IEEE, 2003, pp. II–343.

[2] R. Arora and K. Livescu, "Kernel cca for multi-view learning of acoustic features using articulatory measurements." in *MLSLP*, 2012, pp. 34–37.

[3] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes, "Unsupervised learning of acoustic features via deep canonical correlation analysis," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4590–4594.

[4] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: Lessons learned from the 2015 mscoco image captioning challenge," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 652–663, 2017.

[5] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4651–4659.

[6] K. M. Hermann and P. Blunsom, "Multilingual models for compositional distributed semantics," *arXiv preprint arXiv:1404.4641*, 2014.

[7] A. Klementiev, I. Titov, and B. Bhattarai, "Inducing crosslingual distributed representations of words," 2012.

[8] K. M. Hermann and P. Blunsom, "Multilingual distributed representations without word alignment," *arXiv preprint arXiv:1312.6173*, 2013.

[9] S. Chandar, M. M. Khapra, H. Larochelle, and B. Ravindran, "Correlational neural networks," *Neural computation*, 2016.

[10] A. Saha, M. M. Khapra, S. Chandar, J. Rajendran, and K. Cho, "A correlational encoder decoder architecture for pivot based sequence generation," *arXiv preprint arXiv:1606.04754*, 2016.

[11] W. Härdle and Z. Hlávka, "Canonical correlation analysis," *Multivariate Statistics: Exercises and Solutions*, pp. 263–269, 2007.

[12] B. Thompson, "Canonical correlation analysis," *Encyclopedia of statistics in behavioral science*, 2005.

[13] H. D. Vinod, "Canonical ridge and econometrics of joint production," *Journal of econometrics*, vol. 4, no. 2, pp. 147–166, 1976.

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

[15] X. Yu, D. Hu, and J. Xu, "Kernel independent component analysis," *Blind Source Separation: Theory and Applications*, pp. 145–152, 2014.

[16] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *Journal of machine learning research*, vol. 3, no. Jul, pp. 1–48, 2002.

[17] T. Michaeli, W. Wang, and K. Livescu, "Nonparametric canonical correlation analysis," in *International Conference on Machine Learning*, 2016, pp. 1967–1976.

[18] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *International Conference on Machine Learning*, 2013, pp. 1247–1255.

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

[20] A. Benton, H. Khayrallah, B. Gujral, D. Reisinger, S. Zhang, and R. Arora, "Deep generalized canonical correlation analysis," *arXiv preprint arXiv:1702.02519*, 2017.

[21] A. Ng, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[22] J. Li, M.-T. Luong, and D. Jurafsky, "A hierarchical neural autoencoder for paragraphs and documents," *arXiv preprint arXiv:1506.01057*, 2015.

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

[24] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[25] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[27] Y. Lu and D. P. Foster, "Large scale canonical correlation analysis with iterative least squares," in *Advances in Neural Information Processing Systems*, 2014, pp. 91–99.

[28] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 1083–1092.

[29] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," *Artificial Neural Networks and Machine Learning–ICANN 2011*, pp. 52–59, 2011.

[30] J. Westbury, P. Milenkovic, G. Weismer, and R. Kent, "X-ray microbeam speech production database," *The Journal of the Acoustical Society of America*, vol. 88, no. S1, pp. S56–S56, 1990.

[31] B. Logan *et al.*, “Mel frequency cepstral coefficients for music modeling.” in *ISMIR*, 2000.
[32] A. Eisenschtat and L. Wolf, “Linking image and text with 2-way nets,” *arXiv preprint arXiv:1608.07973*, 2016.
[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, “Scikit-learn: Machine learning in python,” *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.