Multi-view-AE: A Python package for multi-view autoencoder models

Ana Lawry Aguila [1]¶**, Alejandra Jayme**[2]**, Nina Montaña-Brown** [1]**, Vincent Heuveline**[2]**, and Andre Altmann** [1]

1 Centre for Medical Image Computing (CMIC), Medical Physics and Biomedical Engineering, University College London (UCL), London, UK **2** Engineering Mathematics and Computing Lab (EMCL), Heidelberg, Germany ¶ Corresponding author

Often, data can be naturally described via multiple views or modalities. For example, we could consider an image and the corresponding text as different modalities. These modalities contain complementary information which can be modelled jointly using multi-view methods. The joint modelling of multiple modalities has been explored in many research fields such as medical imaging (Serra et al., 2019), chemistry (Sjöström et al., 1983), and natural language processing (Sadr et al., 2020).

Autoencoders are unsupervised models which learn low dimensional latent representations of complex data. The autoencoder framework consists of two mappings; the encoder which embeds information from the input space into a latent space, and a decoder which transforms point estimates from the latent space back into in the input space. Autoencoders have been successful in downstream tasks such as classification (Creswell & Bharath, 2017), outlier detection (An & Cho, 2015), and data generation (Wei & Mahmood, 2021).

There exist many software frameworks for extending autoencoders to multiple modalities. Generally, this involves learning separate encoder and decoder functions for each modality with the latent representations being combined or associated in some way. By far the most popular group of multi-view autoencoder models are multi-view extensions of Variational Autoencoders (VAEs) where the latent space is regularised by mapping the encoding distributions to a gaussian prior using a Kullback–Leibler (KL) divergence term. However, there are also other multi-view autoencoder frameworks, such as multi-view Adversarial Autoencoders (AAEs) (X. Wang et al., 2019). Here the latent space is regularised by mapping the encoding distribution to a prior (here a gaussian) using an auxiliary discriminator tasked with distinguishing samples from the posterior and prior distributions. The choice of AAE or VAE model may be influenced by various elements of the application process. For example, the encoding distribution which best describes the data or stability during training may impact the choice of model.
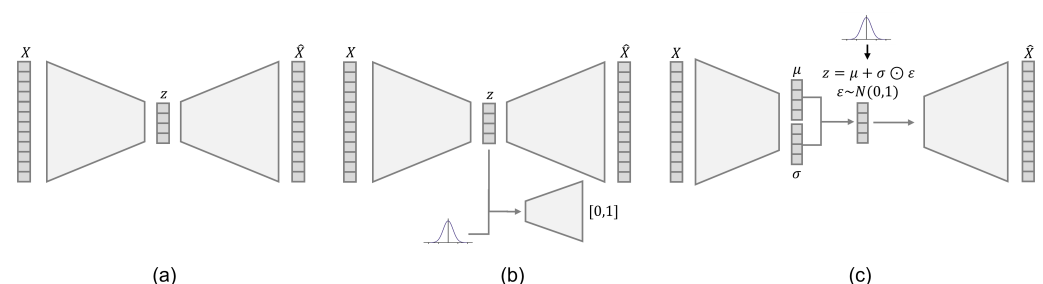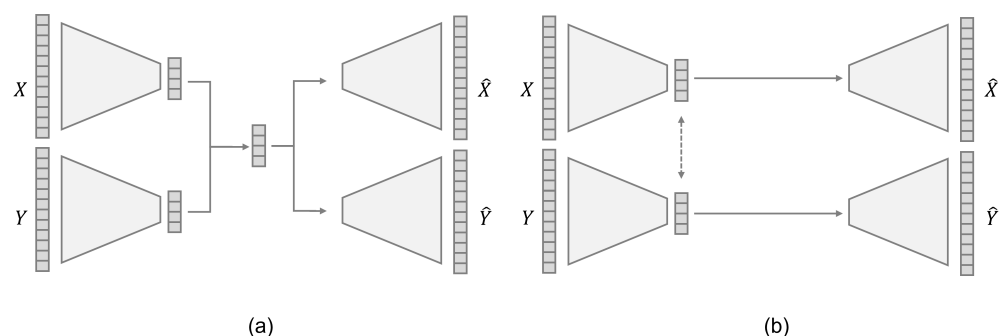

Figure 1: Single view autoencoder frameworks; (a) vanilla autoencoder, (b) adversarial autoencoder, (c) variational autoencoder.

Even within these regularisation frameworks there are vast modelling differences to be considered when choosing the best model for the task at hand. Figure 2 depicts two possible latent variable models for modelling two views of data; X and Y. Figure 2a shows the joint latent variable model (Suzuki & Matsuo, 2022) where both views, X and Y, share an underlying factor. The latent variable model in Figure 2b shows a coordinated model, which assumes some relationship between the latent variables, z_x and z_y of X and Y respectively. Which latent variable model is most appropriate depends on the desired outcome of the learning task. Example multi-view autoencoder frameworks built for these two latent variable models are given in Figure 3.

(a)   (b)
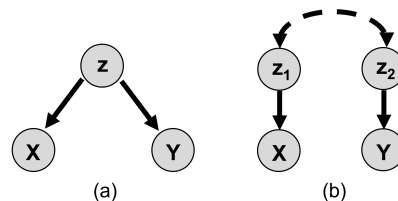
Figure 2: Latent variable models for two input views. Latent variable model where data X and Y (a) share an underlying latent factor z (b) have associated latent factors z_x and z_y.

(a)   (b)

Figure 3: Example frameworks of a two-view autoencoder for data X and Y for a (a) joint model, where the individual latent spaces are combined and the reconstruction is carried out from the joint latent space, and a (b) coordinated model, where the latent representations are coordinated either by cross view generation or an addition loss term for association between the latent variables.

Given the large number of multi-view autoencoders and versatility of architecture, it is important to consider which model would best suit the use case. `multi-view-AE` is a Python library which implements several variants of multi-view autoencoders in a simple, flexible, and easy-to-use framework. We would like to highlight the following benefits of our package.

Firstly, the `multi-view-AE` package is implemented with a similar interface to `scikit-learn` (Buitinck et al., 2013) with common and straight-forward functions implemented for all models. This makes it simple for users to train and evaluate models without requiring detailed knowledge of the methodology. Secondly, all models follow a modular structure. This gives users the flexibility to choose the class (such as the encoder or decoder network) from the available implementations, or to contribute their own. As such, the `multi-view-AE` package is accessible to both beginners, with off-the-shelf models, and experts, who wish to adapt the existing framework for further research purposes. Finally, the `multi-view-AE` package uses the `PyTorch-Lightning` (Falcon & others, 2019) API which offers the same functionality as raw `PyTorch` (Paszke et al., 2019) in a more structured and streamlined way. This offers users more flexibility, faster training and optimisation time, and high scalability.

Statement of need

Multi-view autoencoders have become a popular family of unsupervised learning methods in the field of multi-view learning. The flexibility of the form of the encoder and decoder functions, ease of extension to multiple views, generative properties, and adaptability to large scale datasets has contributed to the popularity of multi-view autoencoders compared to other multi-view methods. Subsequently, multi-view autoencoders have used to address challenges across a range of fields; such as anomaly detection from videos (Deepak et al., 2021) or cross-modal generation of multi-omics data (Minoura et al., 2021).

There exist many different multi-view autoencoder frameworks with the best method of choice depending on the specific task. Existing code is often implemented using different Deep Learning frameworks or varied programming styles making it difficult for users to compare methods. The motivation for developing the `multi-view-AE` library is to widen the accessibility of these algorithms by allowing users to easily test methods on new datasets and enable developers to compare methods and extend code for new research applications. The modular structure of the `multi-view-AE` library allows developers to choose which element of the code to extend and swap out existing Python classes for new implementations whilst leaving the wider codebase untouched.

There exists, as far as we are aware, no Python library that collates a large number of multi-view autoencoder models into one easy to use framework. The `Pixyz` library (Suzuki et al., 2021) is probably the closest relative of `multi-view-AE`, implementing a number of multi-view autoencoder methods. However, `Pixyz` is designed for the wider field of deep generative modelling whereas `multi-view-AE` focuses specifically on multi-view autoencoder models. As such `multi-view-AE` builds upon `Pixyz`'s multi-view offering providing a wider range of multi-view methods.

Software description

Software architecture

Following the `scikit-learn` interface, to train a multi-view autoencoder model with the `multi-view-AE` package, first a model object is initialised with relevant parameters in an easy-to-configure file. Next, the model is trained with the `fit()` method using the specified data. Following fitting, the saved model object can be used for further analysis: predicting the latent variables, using `predict_latent()`, or data reconstructions, using `predict_reconstruction()`.

All models are implemented in `PyTorch` using the `PyTorch-Lightning` wrapper.

Parameter settings

The `multi-view-AE` package uses the Hydra API for configuration management. Most parameters are set in a configuration file and are loaded into the model object by `Hydra`. The combination of Hydra with the modular structure of models in the `multi-view-AE` package, makes it easy for the user to replace model elements with, either other available implementations or their own by editing the relevant section of the configuration file.

Implemented models

A complete model list at the time of publication:

Model class	Model name	Number of views
mcVAE	Multi-Channel Variational Autoencoder (mcVAE) (Antelmi et al., 2019)	>=1
AE	Multi-view Autoencoder	>=1
AAE	Multi-view Adversarial Autoencoder with separate latent representations	>=1
DVCCA	Deep Variational CCA (W. Wang et al., 2016)	2
jointAAE	Multi-view Adversarial Autoencoder with joint latent representation	>=1
wAAE	Multi-view Adversarial Autoencoder with joint latent representation and Wasserstein loss	>=1
mmVAE	Variational mixture-of-experts autoencoder (MMVAE) (Shi et al., 2019)	>=1
mVAE	Multimodal Variational Autoencoder (MVAE) (Wu & Goodman, 2018)	>=1
me_mVAE	Multimodal Variational Autoencoder (MVAE) with separate ELBO terms for each view (Wu & Goodman, 2018)	>=1
JMVAE	Joint Multimodal Variational Autoencoder(JMVAE-kl) (Suzuki et al., 2016)	2
MVTCAE	Multi-View Total Correlation Auto-Encoder (MVTCAE) (Hwang et al., 2021)	>=1
MoPoEVAE	Mixture-of-Products-of-Experts VAE (T. M. Sutter et al., 2021)	>=1
mmJSD	Multimodal Jensen-Shannon divergence model (mmJSD) (T. Sutter et al., 2021)	>=1

Documentation

Documentation is available (https://multi-view-ae.readthedocs.io/en/latest/) for the `multi-view-AE` package as well as tutorial notebooks. These resources serve as both guides to the `multi-view-AE` package and educational material for multi-view autoencoder models.

Acknowledgements

An, J., & Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, *2*(1), 1–18.

Antelmi, L., Ayache, N., Robert, P., & Lorenzi, M. (2019). Sparse multi-channel variational autoencoder for the joint analysis of heterogeneous data. *Proceedings of the 36th International Conference on Machine Learning*, *97*, 302–311. https://proceedings.mlr.press/v97/antelmi19a.html

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., & Varoquaux, G. (2013). API design for machine learning software: Experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108–122.

Creswell, A., & Bharath, A. (2017). Denoising adversarial autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*, *30*. http://arxiv.org/abs/1703.01220

Deepak, K. V., Srivathsan, G., Roshan, S., & Chandrakala, S. (2021). Deep multi-view representation learning for video anomaly detection using spatiotemporal autoencoders. *Circuits, Systems, and Signal Processing*, *40*. https://doi.org/10.1007/s00034-020-01522-7

Falcon, W., & others. (2019). Pytorch lightning. *GitHub. Note: Https://Github. Com/PyTorchLightning/Pytorch-Lightning*, *3*(6).

Hwang, H., Kim, G.-H., Hong, S., & Kim, K.-E. (2021). Multi-view representation learning via total correlation objective. *Advances in Neural Information Processing Systems*, *34*, 12194–12207. https://proceedings.neurips.cc/paper/2021/file/65a99bb7a3115fdede20da98b08a370f-Paper.pdf

Minoura, K., Abe, K., Nam, H., Nishikawa, H., & Shimamura, T. (2021). scMM: Mixture-of-experts multimodal deep generative model for single-cell multiomics data analysis. *bioRxiv*. https://doi.org/10.1016/j.crmeth.2021.100071

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

Sadr, H., Pedram, M. M., & Teshnehlab, M. (2020). Multi-view deep network: A deep model based on learning features from heterogeneous neural networks for sentiment analysis. *IEEE Access*, *8*, 86984–86997. https://doi.org/10.1109/ACCESS.2020.2992063

Serra, A., Galdi, P., & Tagliaferri, R. (2019). Multiview learning in biomedical applications. In *Artificial intelligence in the age of neural networks and brain computing*. Academic Press. https://doi.org/10.1016/B978-0-12-815480-9.00013-X

Shi, Y., Narayanaswamy, S., Paige, B., & Torr, P. (2019, November). Variational mixture-of-experts autoencoders for multi-modal deep generative models. *Neural Information Processing Systems*. https://doi.org/10.48550/ARXIV.1911.03393

Sjöström, M., Wold, S., Lindberg, W., Persson, J.-Å., & Martens, H. (1983). A multivariate calibration problem in analytical chemistry solved by partial least-squares models in latent variables. *Analytica Chimica Acta*, *150*, 61–70. https://doi.org/10.1016/S0003-2670(00)85460-4

Sutter, T. M., Daunhawer, I., & Vogt, J. E. (2021). Generalized multimodal ELBO. *ArXiv*, *abs/2105.02470*. https://arxiv.org/abs/2105.02470

Sutter, T., Daunhawer, I., & Vogt, J. (2021). Multimodal generative learning utilizing jensen-shannon-divergence. *Advances in Neural Information Processing Systems*, *33*.

https://arxiv.org/abs/2006.08242

Suzuki, M., Kaneko, T., & Matsuo, Y. (2021). Pixyz: A library for developing deep generative models. *ArXiv, abs/2107.13109*. https://arxiv.org/abs/2107.13109

Suzuki, M., & Matsuo, Y. (2022). A survey of multimodal deep generative models. *Advanced Robotics, 36*(5-6), 261–278. https://doi.org/10.1080/01691864.2022.2035253

Suzuki, M., Nakayama, K., & Matsuo, Y. (2016). Joint multimodal learning with deep generative models. *arXiv*. https://doi.org/10.48550/ARXIV.1611.01891

Wang, W., Lee, H., & Livescu, K. (2016). Deep variational canonical correlation analysis. *ArXiv, abs/1610.03454*. http://arxiv.org/abs/1610.03454

Wang, X., Peng, D., Hu, P., & Sang, Y. (2019). Adversarial correlated autoencoder for unsupervised multi-view representation learning. *Knowledge-Based Systems, 168*, 109–120. https://doi.org/10.1016/j.knosys.2019.01.017

Wei, R., & Mahmood, A. (2021). *Recent advances in variational autoencoders with representation learning for biomedical informatics a survey. 9*, 4939–4956. https://doi.org/10.1109/ACCESS.2020.3048309

Wu, M., & Goodman, N. (2018). Multimodal generative models for scalable weakly-supervised learning. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 5580–5590. http://arxiv.org/abs/1802.05335