ME Project: Learning Signal Feature Representation using Generative Adversarial Networks

David Hayes (UCD: 13508113)

Monthly Report 1, October 23, 2017

Table of Contents

1 Work Plan

My initial work since being assigned this project has centred around developing a firm understanding of the fundamental machine learning concepts underpinning the project. I am currently enrolled in two online machine learning courses. The first is the Stanford University course from Coursera [1], which I have completed 6 weeks of to date, beginning with introductory concepts before moving on to Neural Networks. I have recently began to shift my focus to Udacity's course [2] which focuses more on deep learning and I plan to continue working on this while concurrently making progress on specific tasks related to my project. I aim to have both courses completed by the time of returning from the Christmas break.

Prior to being assigned this project, I had no experience in the Python programming language. To become familiar with the basic concepts of the language, I recently completed the CodeAcademy Python course. I have installed the recommended packages for this project on my laptop and plan to familiarise myself with their operation over the next few weeks. The aforementioned Udacity course contains deep learning assignments to be completed through Tensorflow. I hope that working through these assignments and implementing some initial GAN-related networks should help me acquire the requisite skills in this area.

By the time of the next monthly report, I hope to get started working with the GPU and I hope to have come close to implementing the results of DCGAN [3] using Keras on a suitable dataset. Once this is implemented, I plan to experiment with different back-ends to compare training results. I will also look into identifying reference datasets that can be used to test my GAN implementations. By December, I hope to have become competent in the Tensorflow framework and have developed a practical understanding of GANs. I would like to have a fully implementable version of DCGAN with GPU acceleration and have the knowledge to make modifications to this model.

By the time of returning from the Christmas break, I hope to have decided on a novel application for feature learning representations using GANs and have an idea of how to use GANs with non-bitmap datasets. At this point in time, it is difficult to speculate on exact activities beyond then, without having deeply engaged in a practical manner with the project as of yet. A further area of possible work may be the visualization of of the internal feature activations of these networks.

2 Git & Backups

The git repository for my projected is located at:

`https://github.com/davidhayes3/Learning-Signal-Feature-Representation-Using-GANs`

3 State of the Art

The emergence of Generative Adversarial Networks (GANs) [4] has established a framework for the unsupervised learning of signal features. A generator learns to generate data from an arbitrary latent distribution, which resembles the distribution of the training dataset in question as closely as possible, while a discriminator learns to distinguish between the real data and the generator data. The learning process of the generator and discriminator, both represented by neural networks, continues until the generator becomes indistinguishable from the real data in the eyes of the discriminator.

While the GAN framework proposes a learning method for the mapping from the latent space to data, the inverse mapping to the latent representation, or features, of the dataset is unclear. The bi-directional framework for GANs, BiGANs [5] explores a method of learning this inverse mapping with the addition of an extra neural network to the standard GAN topology (Goodfellow et al.). This novel framework proposes the unsupervised learning of rich feature representations for arbitrary data distributions. One common technique for evaluating the quality of unsupervised feature representation learning algorithms is to apply them as a feature extractor on supervised datasets and evaluate the performance of linear models fitted on top of these features.

The BiGAN approach to establish feature recognition is one of many ongoing unsupervised learning efforts. A classic approach to unsupervised representation learning is to do clustering on the data, using K-means being one example. Other approaches in the area include Deep Convolutional GANs (DCGANs) and auto-encoders.

A good starting point for my state of the art research was the work of Donahue et al. (2016), which suggests an architecture for implementing BiGAN networks. In addition to the generator from the standard GAN framework, BiGANs include an encoder which maps data to latent representations. The BiGAN discriminator discriminates not only in data space, but jointly in data and latent space, where the latent component is either an encoder output or a generator input. In conclusion, the encoder learns the inverse mapping of the generator. The feature learning capabilities of the BiGAN in question is then evaluated by first training it in an unsupervised manner and then transferring the encoders learned feature representations for use in auxiliary supervised learning tasks. Despite making no assumptions about the underlying structure of the data, the BiGAN's unsupervised feature learning framework offers a representation competitive, in terms of classification accuracy, with existing domain-specific self-supervised and weakly supervised feature learning approaches for visual feature learning for the ImageNet LSVRC [6] and the PASCAL VOC 2007 test set [7]. The main advantage of BiGANs is they are a robust and highly generic approach to unsupervised feature learning, making no assumptions about the structure or type of data to which they are applied.

One alternative to BiGANs is the DCGAN framework proposed by Radford et al. (2016). Here, it is proposed that parts of the generator and discriminator can be reused as feature extractors for supervised tasks. The example used is that of the trained discriminator of the DCGAN enabling image classification. The DCGAN framework shows a classification accuracy of 82.8% on the CIFAR-10 dataset, superior to k-means clustering techniques, despite not having been trained on this dataset. A regularized linear L2-SVM classifier trained on top of the same feature extraction pipeline achieves state of the art results at 22.48% test error for the StreetView House Numbers (SVHM) dataset.

Auto-encoders [8] are another prominent approach to the challenge of feature learning. These are neural networks in which the output layer takes the same number of nodes as the input layer. The auto-encoder has the purpose of reconstructing its input after it passes through a low-dimensional bottleneck layer, with the aim of obtaining a compact latent representation. The encoder maps the inputs to latent representations while the decoder reconstructs the inputs. The encoder and

decoder can be trained together by minimizing the discrepancy between the original data and its reconstruction.

The simple auto-encoder model has difficulty extracting useful features as the reconstruction process could simply copy the input or try to maximize mutual information, without learning a semantically meaningful representation. A large number of auto-encoders of different variations to this simple basic structure have been proposed e.g. denoising auto-encoders (DAE) [9], variational auto-encoders (VAE) [10], adversarial auto-encoders [11], etc. To combat the above deficiencies, DAEs are trained to reconstruct a clean 'repaired' input from a noise-corrupted version of the input. The input is corrupted via stochastic mapping. The stacked DAE of Vincent et al. (2010) yields equivalent or better classification performance on all but one of the considered benchmark problems.

The corruption of DAE inputs does not require much semantic information to undo. A more recent development is that of context encoders [12] which are jointly trained to minimize both a reconstruction loss and an adversarial loss. Pathak et al. demonstrate the use of context encoders in inpainting. A convolutional neural network is trained to generate the contents of an arbitrary image region based on the surroundings of the region. In order to succeed at this task, context encoders need to both understand the content of the entire image, as well as produce a plausible hypothesis for the missing pieces. It is found that the context encoder learns feature representations that allow it to achieve performance results competitive with other models trained with auxiliary supervision on the PASCAL VOC 2007 test set.

One interesting area of this project could possibly be the Visualization of the internal activations of the networks. Radford et al. showed that features learned by the discriminator activate on typical parts of a bedroom, like beds and windows. A similar approach to that adopted by Zeiler et al. (2014) would also be interesting. This visualization technique uses a multi-layered Deconvolutional Network to project the feature activations back to the input pixel space. This visualization technique also reveals the input stimuli that excite individual feature maps at any layer in the model. It also permits the observation of the evolution of features during training.

References

[1] Ng. (2016) Stanford course on machine learning. [Online]. Available: https://www.coursera.org/learn/machine-learning

[2] V. Vanhoucke. (2016) Udacity course on machine learning. [Online]. Available: https://www.udacity.com/course/deep-learning--ud730

[3] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015. [Online]. Available: https://arxiv.org/abs/1511.06434

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014. [Online]. Available: https://arxiv.org/abs/1406.2661

[5] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782*, 2016. [Online]. Available: https://arxiv.org/abs/1602.02830

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge. arxiv preprint arxiv: 14090575," 2014.

[7] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.

[8] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[9] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[11] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[12] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.