ME Project: Learning Signal Feature Representation using Generative Adversarial Networks

David Hayes (UCD: student number)

Monthly Report 0, September 26, 2017

Table of Contents

This short report also serves as a template for preparing your reports using LaTeX.

1 Work Plan

You should devote some time during the first few weeks of term to prepare a detailed work plan for the duration of your project. This should be reviewed regularly as the project develops. Try to put as much thought as possible into this plan as it will help you keeping track of progress and ensure that you meet deadlines. Break down your project into basic tasks/work packages and try to evaluate the amount of work required for each item. You should regularly check progress against this work plan and discuss it from time to time at our meetings.

2 Git & Backups

Set up a git repository for your project on `http://git.ucd.ie` or `http://github.com` and share it with us. Ensure we have sufficient rights to pull your repository (reporter rights should be sufficient). This repository should be logically organised with sub-directories for your log-book, reports, bibliography, code, documentation etc... Ensure your commit and push on a daily basis. All project materials should be contained on this repository. Our school will provide regular backups and ensure no data is lost.

3 First Steps

- □ State-of-art research. This should be as thorough as possible. Use Google Scholar for existing and related works, also search for academic papers e.g. [1–3]. Very importantly, keep an annotated bibliography of your readings (use BibTeX file for all bibliographic entries [4]). Share your findings at our meetings. At the end of this research, you ought to be an expert in ConvNets and GANs!

- □ Learn about deep learning. We recommend to follow some online modules e.g. [5–7].

- □ Learn how to use LaTeXe.g. `https://www.latex-project.org/` and use `http://overleaf.com` for you LaTeXdocuments. If you like, you can also install the LaTeXtools on your laptop, e.g TeXShop and MacTeX on MacOS.

- □ Install some deep learning frameworks suitable for your project, probably Keras with Tensorflow back-end but maybe also Theano, Torch or Caffe . . . . Also install some python IDE (e.g. PyCharm).

- □ Identify a number of datasets that can be used for the testing and evaluation of your GAN algorithms, e.g MNIST dataset [8].

- □ Implement and reproduce results of DCGAN [9] on a dataset of your choice. Use Keras and compare training results using Tensorflow and Theano back-end. Report on performance, training time. . . . Experiment GPU acceleration.

- □ Think about how to use GANs with non-bitmap datasets e.g. our 1D online gesture dataset.

□ Think of interesting applications for feature learning representations using GANs and look into requirements in terms of training models, datasets...

4 Your Project Description

4.1 Background

Recently, a new class of machine learning algorithms called Generative Adversarial Networks (GANs) were introduced to achieve unsupervised learning. It consists of two neural networks competing in a two-player minimax game framework. In this context, a generative model is trained to capture a desired data distribution while a discriminative model estimates the probability that a sample came from the training data rather than the generative model. In essence, this novel framework proposes to learn semantic latent representations given a large dataset. These algorithms have been successful applied to many interesting domain such as the generation of realistic images, the upsampling of super-resolution signals or text to image synthesis, with several open source implementation available on frameworks such Tensorflow, Torch, Theano, Caffe or Keras.

The project proposes to study the application of GANs to signal feature learning using a recently proposed bi-directional framework where the inverse mapping of data to the latent representation (features) is also learned by training an additional neural network. This will first involve the model implementation (python and tensorflow) and the training on the NMIST digit classification dataset. The approach will then be adapted to a different domain e.g. face detection, audio fingerprinting requiring the preparation of a suitable dataset for training and possible changes to the models. The project will develop problem solving and programming skills.

References

[1] 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

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

[3] H. Caesar. (2016) A list of papers and other resources on generative adversarial networks. [Online]. Available: https://github.com/nightrome/really-awesome-gan

[4] BibTeX. (2016) Using BibTEX. [Online]. Available: http://www.bibtex.org/Using/

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

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

[7] NVIDIA. (2016) Course on deep learning. [Online]. Available: https://developer.nvidia.com/deep-learning-courses

[8] Y. LeCun, C. Cortes, and C. J. Burges. (2016) NMIST dataset. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[9] 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