



COMP30170 Final Year Project

ConvNets for iOS Gesture Recognition Applications

Philip Corr (UCD: 12318581)

Weekly Report 1, September 27, 2016

Table of Contents

1	Work Plan	1
2	Git & Backups	1
3	First Steps	1
4	Your Project Description	2
4.1	Background	2
4.2	Technical Description	2

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 and put as much thoughts 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 weekly meeting.

2 Git & Backups

Set up a git repository for your project on <http://git.ucd.ie> and share it with me (user id guenole). Ensure I have sufficient right 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. Google for existing and related works, also search for academic papers e.g. [1–3]. Very importantly, keep an annotated bibliography of your readings (use Bib \TeX file for all bibliographic entries [4]). Don't hesitate to share your findings at our weekly meetings. At the end of this research, you ought to be an expert in ConvNets!
- Learn about deep learning. I recommend to follow some online modules e.g. [5–7].
- Install some deep learning frameworks (Theano, Torch, Caffe . . .), iOS software development tools (Xcode 8, iOS SDK, git setup), and reporting tools (LaTeX. All reports to be typeset in \LaTeX . Use TeXShop and MacTeX on MacOS).
- Learn about Swift and Apple frameworks for iOS. You'll find an excellent course on iTunesU [8] (our semester 2 UCD module follows a very similar syllabus if you are interested)
- Download the NMIST dataset [9], implement and reproduce results of LeNet e.g. [10]. Try implementing model using Torch and Theano. Report on performance, training time. . . Experiment GPU acceleration.
- Building your own dataset (10k+). Start thinking about your experimental setup for building a touch gesture dataset (app UI, what should be stored in datase. . .). Also think about biaises that could affect your data.

- Investigate the possibility to create a pseudo gesture dataset from the MNIST bitmaps (parametrisation of the bitmaps of digits).
- Think about how convolutional neural networks should be applied to your touch gesture dataset. How does it differ from MNIST? How can you exploit the time parametrisation of the digit strokes?

4 Your Project Description

4.1 Background

My proposed project lies in the field of Convolutional neural networks and deep learning and can be viewed as two sections. The first half is to focus on the different algorithms available and to benchmark their performance on classifying the MNIST database of handwritten digits. The second half of my project is to research and develop a convolutional neural network which can be used to classify digits entered into an iOS device. The digits would be entered by the user through the touch screen by tracing out the digit with their finger.

On completion of the project I hope to have successfully benchmarked different deep learning algorithms and implemented one of them in a fully developed digit classifier application for iOS. I also hope to have obtained a large non-biased dataset of gesture written digits which can be used for further research in this area.

This project would be a great thing to do as the field of deep learning is fast becoming one of the most transformative paradigms across areas such as computer vision. This progress is seen in the fact that Intel is now acquiring Movidius, a company whose visionary solutions are based around machine intelligence and deep learning. Another reason that this project is very worthwhile is that one of the biggest obstacles in the way of developing better machine learning algorithms is the lack of datasets which are big enough to be used for machine learning and that are also non-biased.

My proposed work is original as deep learning neural networks haven't yet been used on an iOS device to handle touch screen gestures and the dataset for these gestures also does not yet exist. On top of this most of the papers that have been published on this topic have focused on machine learning algorithms but have not investigated deep learning algorithms as much. This is because deep learning algorithms have only taken off in recent years due to the fact that they are computationally much more intensive.

4.2 Technical Description

This software project will be completed using a variety of relevant technologies. I will use Torch, the scientific computing framework, to create the neural network algorithms and to benchmark them. Torch is based around the fast procedural scripting language Lua and puts GPUs first through its underlying CUDA implementation. The parallel computing this provides will allow more efficient and faster algorithms to be developed. I may also use TensorFlow, an open source library for machine learning written in Python and C++, developed by the Google Brain Team.

Two iOS applications will need to be created, one will be a very simple application used to record the initial touch screen bitmap data. The second will be a more complete application demonstrating

the gesture recognising neural network upon completion of the project. The applications will be developed through Swift, an open-source language developed originally by apple for apple products, and Objective-C. The gesture written numbers can be recorded by the iOS device by making a subclass of the UIGestureRecognizer class provided in the UIKit framework. The recorded data will be stored in a database for future use.

Once the initial application is created it will be used to record the touch location bitmap created by users as they trace out the numbers on the screen. It is important to use a large variety of users so as to ensure that the data used to train and test the neural network is as diverse as possible. The collected bitmap dataset will then be used to train the gesture recognising neural network. I will also need to construct a parameterised dataset from the NMIST data which I can then couple with the data recorded from my iOS application to ensure that the network is trained as accurately as possible. This parameterised pseudo touch location bitmap data obtained from the NMIST dataset will also make it easier to obtain a large enough quantity of data required for training the gesture recognition neural network.

Training the network will require segmenting the dataset into both a non-biased training set and a non-biased test set. The training set will then be passed through the neural network to train it. The most common way of implementing this is through back propagation in conjunction with an optimisation method such as gradient decent.

References

- [1] IEEE. (2016) The IEEE website. [Online]. Available: <http://www.ieee.org/>
- [2] ACM. (2016) The ACM website. [Online]. Available: <http://dl.acm.org/pubs.cfm>
- [3] Y. Lecun. (2016) List of publications. [Online]. Available: <http://yann.lecun.com/exdb/publis/index.html>
- [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] P. Hegarty. (2016) Developing ios 8 apps with swift. [Online]. Available: <https://itunes.apple.com/en/course/developing-ios-8-apps-swift/id961180099>
- [9] Y. LeCun, C. Cortes, and C. J. Burges. (2016) NMIST dataset. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>
- [10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998.