UNIVERSITY OF CAMBRIDGE

Harry Graham
Christ's College
hg402

Computer Science Tripos - Part II Project Proposal

Deep Learning Techniques for Credit Card Fraud Detection

October 20, 2017

Project Supervisor: Dr M. Jamnik & B. Dimanov

Project Originator: H. Graham & B. Dimanov

Director of Studies: Dr R. Mortier

Project Overseers: Dr S. B. Holden & Dr N. R. Krishnaswami

Introduction

Background

A lot of machine learning concepts have been around for decades but ongoing research into deep learning architectures and their applications, makes for an interesting experimentation space. In this project I will explore the performance of some machine learning models, focusing on deep learning, applied to the particular problem of credit card fraud detection (CCFD). This is a globally significant and increasing problem, for example: Annual global fraud losses reached $22.80 billion last year alone, up 4.4% over 2015, according to Nilson Report [1].

Therefore, when techniques prove effective in other domains, or are of recent popularity, it becomes an immediate interest if these methods can be applied on the CCFD space.

The Project

The aim of this project is to explore the use of deep learning techniques in application to the CCFD space. The project comprises core components and possible extension work (mentioned in a later section).

The core project can be split up, at a high level, into the following portions:

- **Baseline Models**
  Setting the scene by exploring a set of very common, broadly used classification models to form a baseline for comparison. The idea being that these techniques are broadly used, there is lots of literature and they are becoming a 'standard' in the toolkit of many data scientists and developers. This baseline will serve as a series of metrics that represent what we can achieve *without* the more elaborate, deep learning techniques.

- **Deep Learning Models**
  This project looks towards comparing and analysing deep learning architectures that have recent examples of success in other domains and have sparked interest in the last decade and are somewhat novel to this data space. The idea is that data we harness, can be seen in many different conceptual views, we are the creator and can customise techniques and models that are already out there to our own uses.

  Convolutional Neural Nets (CNNs) [2]
  CNNs have been popular in the image recognition space. The aim is to experiment with two uses: 1) Treating as a classification problem and disregarding the time component of data, feeding single vector convolutions through the model and 2) Incorporating the time component and trying to utilise the temporal ordering, in a more realistic representation of real life context, using a higher dimension structure and CNNs' weight sharing, spatial locality properties to learn the data.

Generative Adversarial Networks (GANs) [3]
GANs are an interesting use of two neural networks that 'fight' each other and learn from each other's mistakes. The aim here is again, to experiment with this architecture with a few different proposed approaches. Due to the nature of GANs, we can see if the model can learn and simulate fraud data as it comes in based on whether our generative network can fool the discriminator. Further to this we can go down the route of changing the top layer of our CNN to generative and use this pre-trained model from before in our GAN implementation, or we can take one from scratch. This will give rise to a number of different metric comparisons to see which combination works and performs best.

Starting point

Code

Python SciKit-Learn [1] is a machine learning library, that will assist in the construction of the learning models. In general, the Python API is well documented and easy to use, so I plan to use this. Not to mention that Python itself is well suited to data science and data processing.

In addition to this, for the deep architectures I will utilise Keras and TensorFlow. Keras[2] is a Python deep learning library that can be run on top of TensorFlow and TensorFlow[3] is a machine intelligence library from Google. With the intention that I can do most work using Keras with a TensorFlow backend engine, but being able to drop down and fine tune directly with TensorFlow if needed. I plan to use these as, again, they are fairly well documented with intentions for this kind of work. Also, TensorFlow includes compatibility for GPU acceleration, when training models.

Computer Science Tripos

There are a number of courses in the Tripos that serve as a starting point for the reading and undergoing of this project:

- **Artificial Intelligence I** - neural networks, back propagation algorithm
- **Machine Learning and Bayesian Inference** - more machine learning tips, practical issues.
- **Algorithms, Software Engineering** - General software engineering and programming.

The courses in the undergraduate Tripos do not cover much practical detail regarding learning models and certainly do not go into the more advanced deep learning techniques that i'll be using in this project and so I plan to bridge this gap through extensive personal reading and experimentation as well as help from my project supervisors.

[1] http://scikit-learn.org/stable/
[2]https://keras.io
[3]https://www.tensorflow.org

Experience

My own personal experience is of a course a factor in the starting point of this project. I have working-proficiency experience in the Python programming language, from a recently completed summer internship, which will aid implementation. I also have some prior knowledge of machine learning techniques, both from lectures and also from personal reading, which will also act as a starting point for the preparatory reading of this project. I have no prior experience with TensorFlow, however, but this will be mitigated through self-learning in which I will make use of online resources and Google's documentation, as well as tapping into the experience of my supervisors.

Resources required

For this project I shall mainly use my own computer, iMac (27-inch, Late 2013), that runs MacOS Sierra. Backup will be to GitHub and weekly backups will be made to external drive. I will also make use of iCloud drive and a late 2013 MacBook pro too, should I need another machine. The training of models will be done on my own machines which should suffice.

Some the deep networks may require more performance in order to train the models, which could potentially take a long time on personal machines. In this case I will utilise the University's High Performance Computing Service[4].

I may also use computers in the intel lab to do any lighter, more portable work that is possible and certainly if both of my machines decide to break.

In terms of Data I will be using, I will start off with a popular CCFD from Kaggle[5]. The dataset has been collected and analysed during a research collaboration of Worldline and the Machine Learning Group (http://mlg.ulb.ac.be) of ULB (Universite Libre de Bruxelles) on big data mining and fraud detection [4]. There is also hope to obtain a larger dataset from FeatureSpace[6], a Cambridge-based world leader on CCFD. this would mean I have access to a lot more data which could be used in either in the core component of the project or certainly as an extended piece, investigating the effects of increased dataset size.

[4]https://www.cl.cam.ac.uk/local/sys/resources/hpc/
[5]https://www.kaggle.com/dalpozz/creditcardfraud/data
[6]https://www.featurespace.com

Work to be done

Baseline models

The first core component of this project is about implementing various supervised learning models in an attempt to set the scene for 'everyday' standard machine learning techniques, for classification problems. The breakdown of work in this section is as follows:

- **Resampling and Data Preparation**
  I will investigate resampling methods:
  under-sampling, over-sampling and the SMOTE method
  due to the imbalance of data, testing on a simple logistic regression classifier and optimising by tuning relevant parameters. Which ever method proves most effective, is what will be used feeding into the other classifiers to follow.

- **Implementation of various classifiers**
  I will implement and train the following classifiers on the data:
  K Nearest Neighbours, Linear SVM, Gaussian Process, Decision Tree, Random Forest, Neural Net, Naive Bayes.

Deep learning models

The second and more extensive core component of the project is focused on the deep learning techniques, that are novel to this CCFD data. I plan to investigate and ultimately compare the following techniques:

- **Convolutional Neural Networks** [2]
  The first stage here will be to plan and map out the customised CNN model for the data. After planning, the implementation will split into two sub-parts:
  1) Experimentation of using single vector convolutions, treating as a classification problem
  2) Using the temporal ordering and fixed-window, multiple vector approach.
  After implementing both of these CNN avenues, I will then experiment with some possible optimisations such as hyper-parameter tuning (e.g. on the size of the window, or how many vectors do we pass in the second approach).

- **Generative Adversarial Networks** [3]
  A similar approach of planning will be taken before implementing the sub-paths:
  1) Use our pre-trained CNN from the previous experiments, for the basis of our generator network in the GAN
  2) Use a new network from scratch, to give rise to lots of metrics on which we can compare and contrast and see what performs best.
  3) See if we can utilise the very nature of GANs and simulate fraud data.

In both cases (baseline and deep models), collating all the results gained for reference and comparison, in the evaluation chapter.

Success Criteria and Evaluation

Overview

The project will be a success in a mixture of quantitive and qualitative results. In terms of the work to be done, a summary of success would be:

- **Implemented and analysed baseline models**
- **Implemented deep learning models**
- **Conclusive analysis and insight of techniques, from gathered experimental results**

Metrics we're interested in

When evaluating the project, there are certain metrics that we are interested in that is not just the accuracy of prediction, like in many machine learning applications. Due to the nature of CCFD data, there will always be an imbalance in the data classes i.e. more non-fraud examples than fraudulent. Do to this property, we are not only interested in model accuracy, but both precision / recall too.

$$precision = \frac{T_p}{T_p + F_p} \quad , \quad recall = \frac{T_p}{T_p + F_n} \qquad T_p = TruePositive, F_n = FalseNegative$$

The idea here is that we care more about catching false negatives than we do about letting some false positives get through, so these metrics are important in evaluating the models used on CCFD data. In general, confusion matrix (TP, TN, FP, FN) results will be important for comparisons, visualised with receiver operating characteristic (ROC) curves.

The basis for evaluation will be using this data and comparing this with what is achieved by the deep models, proposed in the second part of the project. This will help give an intuition to how the performance compares with baseline models.

Of course, in some cases we will pull interesting results such as training time, In particular in the case of experimenting with GANs. Seeing how much training time is reduced by using our existing CNN and correlating this with performance, will be an interesting insight.

Evaluation Breakdown

Evaluation can be broken down into the following:

- **Baseline models**
  — Evaluating which sampling methods perform best, using the simple logistic regression classifier, using metrics described above.
  — Tuning parameters of the classifier to try and achieve better results.
  — Comparing the resulting metrics from all the implemented classifiers with one another, focusing on precision-recall curves.
  — Summarising this empirical data.

- **Deep learning models**
  When evaluating the deep models implemented, there will be a number of variations in what comparisons are drawn. Primarily between experiment, with that of the baseline models and then between the deep experiments themselves, in the following fashion:
  — Comparing results of CNN method 1 and 2, with that of baseline models.
  — Comparing results of CNN method 1, with that of method 2.
  — Comparing results of GAN method 1 and 2, with that of baseline models.
  — Comparing results of GAN method 1, with that of method 2.

  Using metrics described (precision-recall, training time, confusion matrix etc) and on the same dataset, using a training-test data split that is appropriate. For example a 70:30 split.

In addition to this, there will of course be a substantial qualitative analysis of the techniques implemented, covering areas such as intuition gained from the experiments, difficulty and overhead of training, a word on resource consumption, problems encountered / overcome and a general analysis on how the deeper learning techniques perform in this space.

Possible extensions

If the core parts of the project are successful and completed within reasonable time then possible extensions may lead to a few further high level investigations. Namely:

- **Implementing a DNC architecture**
  This would be an attempt at using the cutting edge technique proposed by a team from DeepMind [5] and exploring whether the use of external dynamic memory means that we can learn the entire history of the data and whether this is promising for the CCFD space.

- **Comparing deep auto encoders for pre-training** This extension would comprise of implementing a few deep auto encoder models and using these with baseline models and the deep models and seeing how effective these are.

Timetable

Planned starting date is 19/10/2017.

1. **Michaelmas weeks 3–5, [19/10 - 02/11]:** Preparatory reading on learning models, experiment with SciKit-Learn, Keras and TensorFlow. Experiment with importing and manipulating datasets. Reading on deep learning techniques. Look at under/over sampling techniques.

2. **Michaelmas weeks 5–6, [02/11 - 09/11]:** Implement baseline models and run training on dataset. Draw results for baseline models. Overflow on deep learning reading.

   Milestone**: Have implemented all baseline work and summarised results**

3. **Michaelmas weeks 6–8, [09/11 - 30/11]:** Investigate, plan and experiment with the first CNN implementation on the data. Build the network setup using Keras + TensorFlow. Run training on data.

   Milestone**: Trained the first implementation of the CNN, ready to setup the second**

4. **Michaelmas vacation** Write progress report. Generate visualisations of test results and how models performed, for demonstration purposes. Give insight into CNN work for CCFD. Prepare any necessities for implementing the second CNN implementation.

   Milestone**: Have a completed progress report and prepared a presentation for demonstration purposes**

5. **Lent weeks 0–2, [14/01 - 01/02]:** Continue / finish CNN work. Immediately start planning and setting down framework for implementing the GAN using Keras + TensorFlow.

   Milestone**: Completed and finalised results from CNN work.**

6. **Lent weeks 2–5, [01/02 - 22/02]:** Implement and train GAN. Start to finalise results and conclusions, allowing some overflow period for this main part of the project.

7. **Lent weeks 5–8, [22/02 - 15/03]:** Overflow period. If time permits, have a go at extensions. Clean-up, re-runs.

   Milestone**: Completed GAN work and all core project components.**

8. **Easter vacation:** Writing dissertation main chapters. Possible overflow of extensions.

9. **Easter term 0–2, [22/04 - 04/05]:** Complete dissertation. Proof reading and then an early submission so as to concentrate on examination revision.

References

[1] The Nilson Report Issue 1118 — Oct 2017. Card fraud losses reach $22.80 billion. `https://www.nilsonreport.com/`, 2017.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *ArXiv e-prints*, June 2014.

[4] Andrea Dal Pozzolo, Olivier Caelen, Reid A Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. In *Computational Intelligence, 2015 IEEE Symposium Series on*, pages 159–166. IEEE, 2015.

[5] Alex Graves, Wayne, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.