Final Proposal Report Group 4

Improving Credit Card Fraud Detection using Generative Adversarial Networks

Group 4 Team Member: Hao Ning, Jun Ying

Introduction

As credit card transactions have become the mainstream consumption pattern, the number of credit card frauds has increased dramatically. Even if users are finally compensated, they often spend a lot of time and even money in the process. Therefore, it becomes important to distinguish whether the transaction is fraud or not from the beginning.

In this study, we will first use traditional machine learning algorithms for classification training. However, relative to the number of transactions, the proportion of credit card fraud is very small. Traditional machine learning tends to be biased towards the mainstream category (non-fraud transaction) in the imbalanced data set of classification. Therefore, we will use Generative Adversarial Networks (GAN[1], Goodfellow, et al. (2014)) to generate artificial credit card fraud data to oversample the data set.

GAN is a generative model in which two neural networks (generator and discriminator) compete and improve each other. The generator keeps generating more and more "real" fake data to deceive the discriminator. However, the discriminator needs to distinguish between the data generated from the generator and the original data.

Through the artificial credit card fraud data generated by GAN, we will well oversample the data set to improve the accuracy of the classification model. In this study, we will compare four GANs (Wasserstein GAN[2], Balancing GAN[3], Boundary Equilibrium GAN[4] and Spectral Normalization GAN[5]), and observe their performance on the generated data.

Data Description

The data set contains 284,807 transactions made by European cardholders within two days through credit cards in September 2013. However, we only had 492 frauds in all transactions, and the positive class(fraud) accounts for 0.172% of all transactions.

Due to confidentiality issues, features V1, V2, … V28 are the principal components obtained by PCA transformation; ‘Time’ is sorted in time series in seconds; ‘Amount’ is the transaction Amount; and 'Class' is the response variable and it takes value 1 in case of fraud and 0 non-fraud.

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0

Link to the data: Credit Card Fraud Detection

Methodology

First, exploratory data analysis (EDA) will be performed to understand the data distribution.

Then, random oversampling/undersampling and machine learning algorithms such as logistic regression, XGBoost will be implemented.
Finally, different GAN will be introduced to produce more minority samples, combined with the original data, in order to improve the model performance.

Reference Materials and Background Support

We will implement different GAN models for detailed performance comparison, including vanilla GAN[1], Wasserstein GAN(WGAN)[2], Balancing GAN (BAGAN)[3], Boundary Equilibrium GAN (BEGAN)[4], and Spectral Normalization GAN (SNGAN)[5]. Vanilla GAN has some drawbacks such as mode collapse and gradient vanishing. WGAN is introduced as a remedy for this scenario, by using Wasserstein distance to evaluate the distance between real and generated samples instead of JS convergence equivalent problem in GAN. WGAN applies weight clipping to 'fulfill' the constraint requirement of 1-lipschitz function. However, the WGAN approach sometimes fails to converge and will only generate poor samples, which is usually caused by the use of weight clipping in WGAN. Thus, more recent GAN models mentioned earlier will be implemented for potential improvements.

Performance Evaluation

Since the data is unbalanced, accuracy is not a fair evaluation of the model performance, thus precision, recall, f1 score will be compared for different models. AUC and ROC curves will also be plotted to understand the model performance. Problems in GAN such as mode collapse, gradient vanishing, convergence and model stability will also be analyzed.

Individual Work & Contribution

Both of the team members will work on the preprocessing, EDA and base model.
Each of the team members will work on 2 improved GAN models individually:
Hao: WGAN & BAGAN
Jun: BEGAN & SNGAN

Our contribution:
How GAN would help us in dealing with an imbalanced dataset.
A detailed comparison of the performance of different GAN models in a real life problem.

---

[1] Goodfellow, Ian, etc. (2014). Generative Adversarial Networks
[2] Wasserstein GAN [1701.07875] Wasserstein GAN
[3] BAGAN [1803.09655] BAGAN: Data Augmentation with Balancing GAN
[4] BEGAN: Boundary Equilibrium Generative Adversarial Networks [1703.10717] BEGAN: Boundary Equilibrium Generative Adversarial Networks
[5] Spectral Normalization for Generative Adversarial Networks [1802.05957] Spectral Normalization for Generative Adversarial Networks

Working Schedule

Time	Milestone
09/21/2020	Exploratory Data Analysis (EDA): Jun Base Model: Hao
09/28/2020	Original data + GAN
10/05/2020	Network & Framework Development WGAN: Hao BEGAN: Jun
10/12/2020	WGAN & BEGAN Evaluation & Analysis
10/19/2020	Network & Framework Development BAGAN: Hao SNGAN: Jun
10/26/2020	Preliminary Presentation
11/02/2020	BAGAN, SNGAN Evaluation & Analysis
11/09/2020 & 11/16/2020	Summary of Results
11/23/2020	Manuscript
11/30/2020 & 12/07/2020	Mock Presentation & Presentation and Journal Submission