UNIVERSITY OF NOTRE DAME

Department of Computer Science

Interdisciplinary Center for Network Science Applications (iCeNSA)

REU Project Report

Application of Deep Learning in Class Imabalance

Supervisor:
Prof. Nitesh Chawla
Prof. Reid Johnson

Student:
Nilay Thakor

Year 2016

Introduction

The issue of class imbalance is observed in data when the class of interest is under represented. This issue has attracted researchers for a long time. The major technique to resolve class imbalance includes resampling of the data.[1]. The resampling techniques can be classified as oversampling, undersampling and hybrid techniques. SMOTE[2] is a very famous oversampling algorithm.

In the recent years various Deep Learning algorithms have achieved breakthrough performances. In this project I have designed some experiments to apply deep learning methods to resolve the issue of class imbalance.

Experiments

The experiment are done over various oversampling techniques and integration of deep learning in them. The first set of experiments involves DAEGO[3] method which generates sythetic samples using denoising auto encoders[4].

The other experiments is implementation of DAF[5] algorithm which uses stacked denoising encoders to transform the feature space.

In attempt to improve the SMOTE algorithm an experiment SMOTE_SADE is designed which includes stack de-noising encoders before SMOTE.

DAEGO

DAEGO[3] is an oversampling technique to generate synthetic samples using de-noising encoders. The generation of synthetic sample is done using following algorithm.

Algorithm 1 DAEGO

1. $norm\_params \leftarrow$ Nomalization Parameters
2. $x\_norm \leftarrow$ Nomalized Posotive Class using norm_params
3. $dea \leftarrow$ de-noising encoder network created using hidden layers H and activation σ
4. Train dea using x_norm
5. Transform x_init using dea
6. denormalize x_norm and x_init using $norm_params$
7. Return: x_init

The synthetic samples generated by DAEGO has very less variance that compare to SMOTE[2]. Figure 1 shows that.

The experiments shows that AUC score of classifier increases if the when DAEGO samples are generated with higher dimensional hidden layer. Also performance do not vary after more than two hidden layer.

Additional to DAEGO two experiments were done to increase the dimensionality of data before performing DAEGO using encoder and stacked encoder.

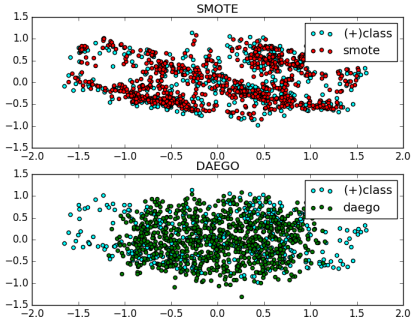
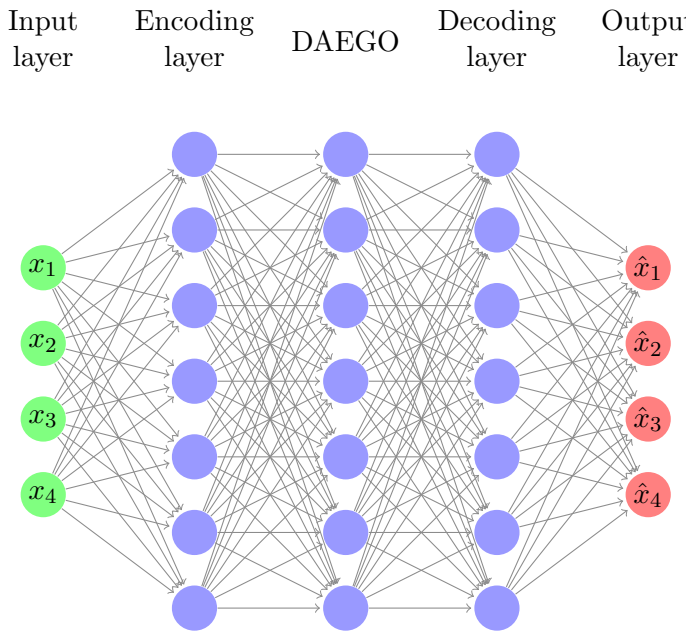


Figure 1: Comparision between samples generated via SMOTE and DAEGO

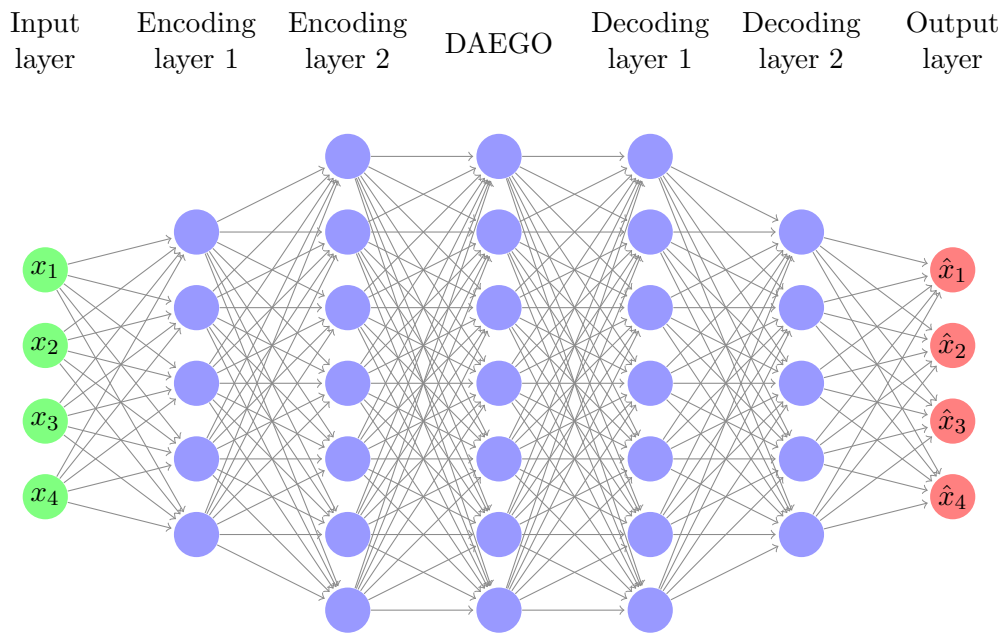
DAEGO_ENCODER

In DAEGO_ENCODER the I increased the dimensions of data using encoder and than performed DAEGO oversampling. Following diagram explain the process.



DAEGO_SDAE

In DAEGO_SDAE the we increase the dimensions of data using stacked encoders and than performed DAEGO oversampling. Following diagram explain the process.



SDAEGO

In this experiment we use deep stacked denoising encoders to geneate synthetic samples. It follows the same procedure as DAEGO but uses stacked encoders for synthetic data.

DAF

DAF[5] is a resampling technique which uses stacked de-noising encoders to resample the data into reduced feature space. Figure 2 shows the procedure. It transform the dataset using stacked encoders and *sigmoid* and *tanh* activation respectively. By stacking both the output we get final resampled dataset.

Algorithm 2 DAF

1. Create a stacked encoder network using hidden layers H
2. Scale the dataset between the range of 0 to 1
3. Train the dataset using sdae network using sigmoid activation and tanh activation
4. merge both the output

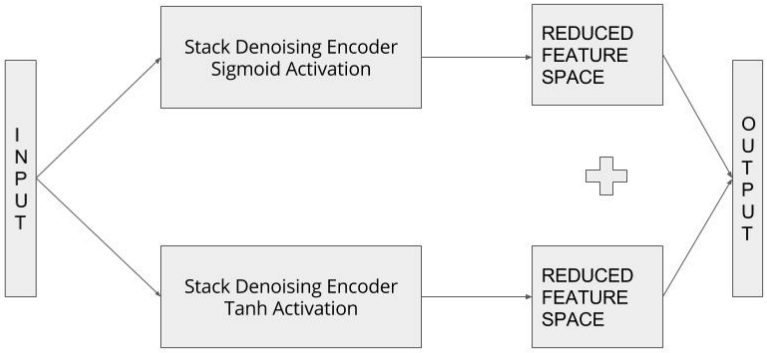
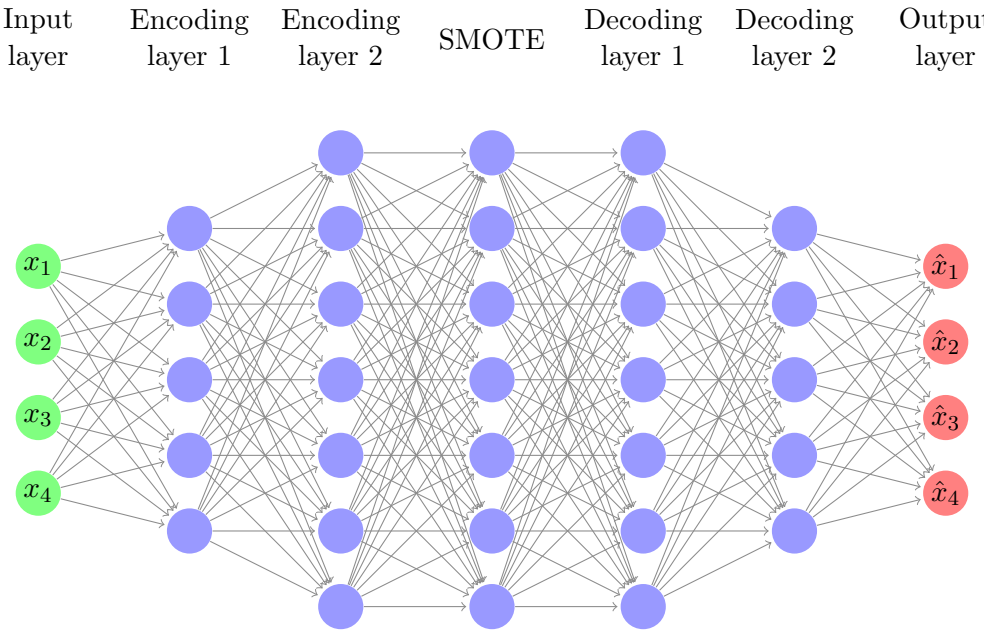


Figure 2: DAF

SMOTE_SDAE

This experiment transform the dataset into higher dimesional space using staced encoder than perform the SMOTE oversampling. Following diagram explains the procedure.



Results

All the experiments were tested on four different dataset. Table 1 has information about all datasets. The imbalance ratio is defined as Negative Class Samples to Positive Class Sample ratio.

Table 1: Datasets

Dataset	**Features**	**Sample**	**Imbalance Ratio**
Segment	19	2310	6
KDDCUP	76	145751	111.46
boundary	175	3505	27.495
page-block	10	1578	8.788

The classifier used are Decision Tree and Multilayer Perceptron. The ratio of training vs testing data is 70:30.

Table 2 Compares results of classifiers without any resampling and 100% oversampling of SMOTE, DAEGO and SDAEGO

Table 2: Results

Dataset	CLF	NONE		SMOTE		DAEGO		SDAEGO	
		PC	AUC	PC	AUC	PC	AUC	PC	AUC
Segment	DT	0.9739130435	0.9976958525	0.9655172414	0.9969278034	0.9655172414	0.9969278034	0.6967213115	0.8689950422
	MLP	1	0.9955357143	0.9824561404	0.9984639017	1	0.9910714286	1	0.98598
KDDCUP	DT	0.612371134	0.871144928	0.6336842105	0.8763168041	0.6812652068	0.8503856284	0.1818181818	0.8533867244
	MLP	0.6012658228	0.8560590689	0.5593869732	0.8644232694	0.7878787879	0.76071953	0.97	0.7193940625
boundary	DT	0.008888	0.53	0.125	0.54529	0.078947	0.5209	0.05769230769	0.5165474061
	MLP	0.375	0.5958672	0.086	0.51498	0.1	0.51632	0.2	0.5220632081
page-block	DT	0.8206521739	0.8892668178	0.8062827225	0.8959664674	0.8636363636	0.8946952518	0.3586387435	0.7866762867
	MLP	0.7571428571	0.9048649763	0.8333333333	0.9004672576	0.8529411765	0.8758675187	0.9150943396	0.7538308253

Table 3 compares 100% oversampling of two feature transformation method applied before DAEGO.

Table 3: Results

Dataset	CLF	NONE		SMOTE		DAEGO_SDAE		DAEGO_ENCODER	
		PC	AUC	PC	AUC	PC	AUC	PC	AUC
Segment	DT	0.9739130435	0.9976958525	0.9655172414	0.9969278034	0	0.5	0.1467889908	0.5
	MLP	1	0.9955357143	0.9824561404	0.9984639017	0.1467889908	0.5	0.1773162939	0.6
page-block	DT	0.8206521739	0.8892668178	0.8062827225	0.8959664674	0.4285714286	0.5066996496	0.0841924	0.3944
	MLP	0.7571428571	0.9048649763	0.8333333333	0.9004672576	0.09870276368	0.4688380403	0.1046511	0.5

Table 4 comapres results of resampling method DAF and SMOTE_SDAE.

Table 4: Results

Dataset	CLF	NONE		SMOTE		DAEGO_SDAE		DAEGO_ENCODER	
		PC	AUC	PC	AUC	PC	AUC	PC	AUC
Segment	DT	0.9739130435	0.9976958525	0.9655172414	0.9969278034	0	0.5	0.1467889908	0.5
	MLP	1	0.9955357143	0.9824561404	0.9984639017	0.1467889908	0.5	0.1773162939	0.6
page-block	DT	0.8206521739	0.8892668178	0.8062827225	0.8959664674	0.4285714286	0.5066996496	0.0841924	0.3944
	MLP	0.7571428571	0.9048649763	0.8333333333	0.9004672576	0.09870276368	0.4688380403	0.1046511	0.5

The detailed results and code can be found at https://github.com/nthakor/imbalance_algorithms

Future work

Above experiments are an effort to integrate the deep learning approaches to resolve class imbalance via over-sampling. Following are some areas need to be explored to improve the results geerated by these experiments.

- In some of the datasets issue of class overlapping is observed. The re-sampling method using deep learning can be developed to resolve the issue.
- The most experiments performed are related includes oversampling. The effectiveness of deep learning approaches can be explored in other methods like under-sampling.
- The above experiments apply deep learning approaches in the process of oversampling. Unsupervised deep learning approaches can be applied after oversampling.

Bibliography

[1] Nitesh V Chawla. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pages 875–886. Springer, 2009.

[2] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[3] Colin Bellinger, Nathalie Japkowicz, and Christopher Drummond. Synthetic oversampling for advanced radioactive threat detection. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 948–953. IEEE, 2015.

[4] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[5] Wing WY Ng, Guangjun Zeng, Jiangjun Zhang, Daniel S Yeung, and Witold Pedrycz. Dual autoencoders features for imbalance classification problem. *Pattern Recognition*, 2016.