Gogoberidze, Nodari

Fraud Detection using MLPs and Autoencoders

A case study in fraud detection using deep learning methods

Problem Statement

Fraudulent activity is present in just about every industry and economic sector. It comes in many shapes and forms, and varies in impact. In finance and banking, financial fraud is rife as bad actors try to game the system to gain monetary rewards. The detection of financial fraud is of significant interest to every financial entity. For this project, I will be concentrating on detecting fraudulent credit card transactions using deep learning methods. Of particular focus is optimizing model performance on a highly imbalanced dataset.

High Level Overview of Steps

In order to detect fraudulent transactions, I make use of two types of deep learning models. Since this is a two class problem, i.e. fraud vs. non-fraud, I first perform a binary classification on the dataset. I construct two variations of this model. The first is the straightforward, fully-connected model. The second variation places class weights on the training data to account for the data imbalance.

The second type of model makes use of autoencoders. At first it seems counter-intuitive to use an unsupervised, or self-supervised, method to train a network on labeled data. However, it is demonstrated that the resulting model performs relatively well for this type of problem. I run predictions on this model using test data the same in size as the Binary Classification model for a direct comparison of the two model types. I then run a second set of predictions with a much larger number of fraudulent data (that was left unused during training).

Dataset

The “Credit Card Fraud Detection” dataset is available form Kaggle. The data contains credit card transactions made over a two day time period. There are 284,807 recorded transactions, with 492 labeled as fraudulent transactions. As can be seen from the number of Normal transactions versus the number of Fraudulent transactions, this is a highly unbalanced dataset, and is the major challenge to be overcome in this project.

Software

Keras (v. 2.2.4), with a Tensorflow (v. 1.13.1) backend; Python (v. 3.6.7).

Youtube 2-Minute: https://youtu.be/Ukyud5gpa2o
Youtube 15-Minute: https://youtu.be/h-5t6h384K8