UNIVERSIDADE FEDERAL DO TOCANTINS
CIÊNCIA DA COMPUTAÇÃO
ALGORITMOS EM GRAFOS

ERIC JONAI COSTA SOUZA

Trabalho final de aprendizado de máquinas

Palmas,
Junho de 2023

Introdução

O dataset escolhido para a realização deste trabalho é o Credit Card Fraud | Kaggle, que traz várias colunas referentes a informações de compras efetuadas por um cartão de crédito. Entre essas variáveis, estão a distância da transação em relação à residência, a distância entre a transação atual e a anterior, a relação entre o valor da compra atual e a média das compras, indicadores sobre o tipo de transação (como se foi feita por aproximação ou por meio de um código PIN) e se a compra foi realizada online. O objetivo principal é identificar se uma transação é fraudulenta ou não, o que torna esse problema um desafio de classificação binária.

Ao explorar o dataset, é possível notar a presença de uma classe minoritária representando as transações fraudulentas, enquanto a classe majoritária representa as transações autênticas. Isso indica a existência de um desequilíbrio de classe, o que pode afetar o desempenho dos modelos de aprendizado de máquina. Para abordar esse problema, faz-se necessário considerar técnicas de reamostragem, como undersampling ou oversampling. Todas as informações do dataset são numéricas e este não possui dados faltantes, o que colabora com a redução do esforço no pré-processamento da informação. Existem aproximadamente 1 milhão de registros no dataset.

A escolha deste dataset é antiga, pois um trabalho similar foi solicitado na disciplina de deep learning, apesar disso, a rede neural de classificação apresentada é única para as demandas deste trabalho. O objetivo é treinar uma rede neural adequadamente precisa para identificar fraudes, o imenso quantitativo de registros permite o treinamento adequado da rede neural, enquanto o desbalanço é facilmente contornado ao repartir o conjunto de treinamento em partes menos desbalanceadas. O resultado esperado é uma rede que seja capaz de deixar o menor valor possível de falsos negativos, isto é, operações fraudulentas confundidas como autênticas, enquanto possui um moderado grau de tolerância para falsos positivos.

# distance_f...	# distance_f...	# ratio_to_m...	# repeat_ret...	# used_chip	# used_pin_...	# online_order	# fraud
57.87785658389723	0.3111400080477545	1.9459399775518593	1.0	1.0	0.0	0.0	0.0
10.829942699255545	0.17559150228166587	1.2942188106198573	1.0	0.0	0.0	0.0	0.0
5.091079490616996	0.8051525945853258	0.42771456119427587	1.0	0.0	0.0	1.0	0.0
2.2475643282963613	5.600043547072320	0.36266257805709584	1.0	1.0	0.0	1.0	0.0
44.19093600261837	0.5664862680583477	2.2227672978404707	1.0	1.0	0.0	1.0	0.0
5.586407674186407	13.261073268058121	0.06476846537046335	1.0	0.0	0.0	0.0	0.0
3.7240191247148107	0.9568379284821842	0.27846494490815554	1.0	0.0	0.0	1.0	0.0

Primeiras linhas do dataset

Metodologia

Como o dataset é por padrão composto exclusivamente de números, o pré-processamento dos dados é relativamente curto e simples. Não existem dados faltantes e existem apenas três colunas viáveis para busca e eliminação de outliers, distância de casa, distância da última compra e relação entre esta compra e a média de todas as compras. Entendi que estes valores são relevantes e potencialmente verdadeiros e valiosos demais para remoção mesmo em casos extremos, observado que uma pessoa autêntica efetuar uma compra muito maior do que a média ocasionalmente e um fraudador obter acesso a um cartão em outro país são ambas ocorrências possíveis. Assim, não houve tratamento para outliers.

O desbalanceamento do dataset foi sobrepujado de forma simples, a biblioteca Panda, do Python, permite que eu selecione linhas do arquivo de forma discriminada, ainda que aleatoriamente. Assim, selecionei 40 mil linhas dentre as que representavam uma transação fraudulenta e 60 mil linhas dentre 900 mil que marcaram transações autênticas. Estes números específicos foram obtidos por tentativa e erro, apesar da relação extrema onde 40% de todas as transações são fraudulentas no conjunto de treinamento, a rede neural obteve um excelente desempenho no final. Para informativo de comparação, meu primeiro conjunto de treinamento foi feito com 400 mil linhas de transações autênticas (10% de fraudulentas), mas a rede neural fez um trabalho pior, classificando a maior parte das transações como autênticas e criando falsos negativos, violando o resultado esperado.

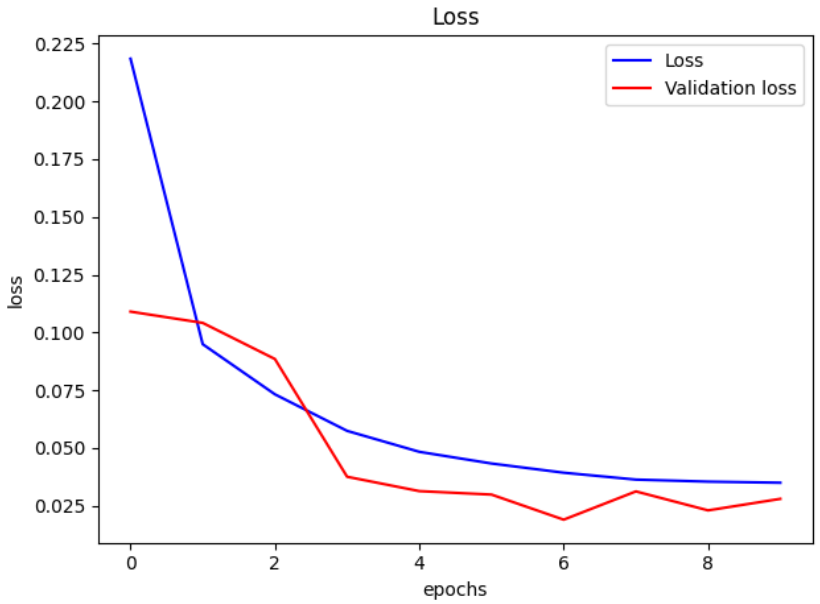
Com a biblioteca Tensorflow, criei a rede neural propriamente dita. Escolhi a ativação ReLU já que não teria uso para valores negativos e linearidade nos dados obtidos. Para a camada de saída, escolhi a ativação sigmoid por se tratar de um problema de classificação binária (ou é fraudulenta, ou é autêntica). Utilizei o meu conhecimento adquirido no trabalho 3, onde cada nova camada inserida na rede neural perde gradativamente a capacidade de se atualizar, e, por isso, optei por manter apenas 4 camadas, onde duas são ocultas. Também por tentativa e erro, mantive 64, 32, 32 e 128 neurônios, um número menor causava a rede a inserir uma grande quantidade de falsos negativos, o que indicava que a rede possuía menor capacidade de reconhecimento de padrões. Não cheguei a testar com mais neurônios.

Por fim, emiti gráficos de loss, validation loss e matriz de confusão. A rede obteve acurácia entre 95% e 99% durante o treinamento, mas não considerei esta informação valiosa porque a grande quantidade de transações autênticas causava a acurácia a sempre encontrar valores altos. O gráfico de loss suaviza no final, o que indica que a rede neural estava alcançando o limite do seu conhecimento possível, e o validation loss variava de forma caótica sempre abaixo do loss, mas obteve

aumento no final, indicando que o treinamento foi encerrado no momento adequado. Não acreditando que exista algo mais a ser feito, fico satisfeito com a rede neural.

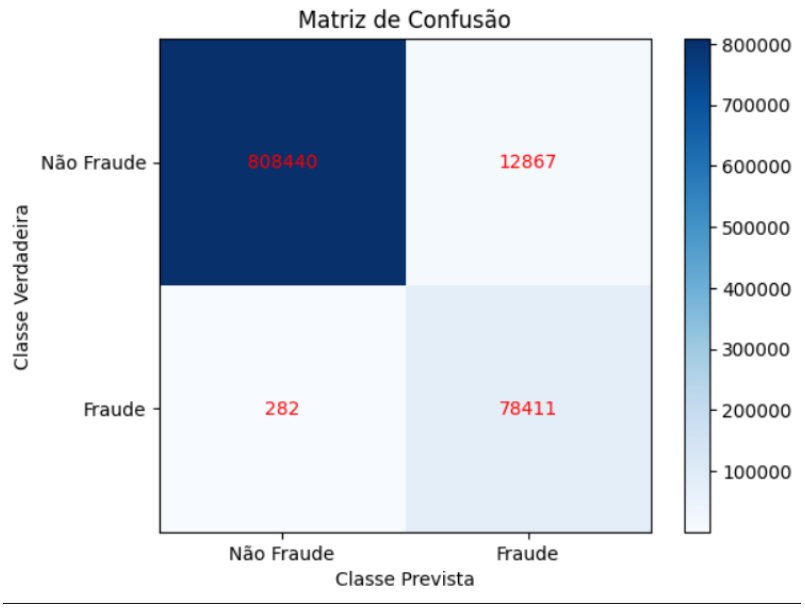
Resultados

Para melhor análise da rede neural, plotei um gráfico de loss:



A linha de loss aponta quão distantes estão as previsões do modelo em relação aos valores reais, e mostra que a rede neural estava chegando no ápice ao possuir gradativamente menos redução de erro em relação às épocas. A informação de validation loss na rede neural é uma métrica calculada durante o treinamento, mas usando um conjunto de validação separado, dentre muitas coisas, útil para encontrar overfitting. O fato dela obter um leve aumento após a época 8 indica que o treinamento do modelo foi encerrado no tempo adequado, ou a rede poderia ter decaído na qualidade.

Por fim, plotei a matriz de confusão. Como a camada de saída com ativação sigmoid retorna valores fracionados entre 0 e 1, já que a camada ReLU não emite valores negativos, considerei todo valor acima de 0.5 como fraudulento e abaixo de 0.5 como autênticos. O resultado foi uma matriz que eu considero excelente, uma alta relação de acertos por erros somada a uma quantidade extremamente baixa de falsos negativos mostram que a rede neural chegou ao resultado que eu esperava.


Matriz de Confusão
Não Fraude
Fraude
Classe Verdadeira
808440
12867
282
78411
Não Fraude
Fraude
Classe Prevista
800000
700000
600000
500000
400000
300000
200000
100000