Change Address Detection with Deep Learning

Peter Wang (pwang01@stanford.edu)

Predicting

- Features of the Bitcoin transaction graph are fed into the model.
- For each transaction, all of the output addresses are assigned a binary value of whether they are change addresses.
- Applications in clustering addresses to owners.

Data

- The data concerns transactions harvested from the distributed Bitcoin blockchain.
- It also contains embedded values of the exchange rate of Bitcoin to USD at the times that the transactions occurred.
- Included approximate location of transaction submitted to the network
- Supervision provided by existing heuristic and empirical data regarding clustering of addresses to the same owner.
- Pre-processed into an indexed MySQL database consisting of multiple tables for various mappings between the data.
- MySQL database was partially downloaded and cleaned up into NumPy arrays

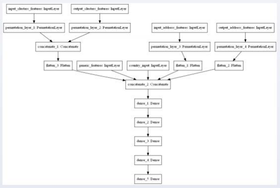
Features

- In- and out-degree of each address in the transaction graph (captures behavior)
- Various amounts transferred in and out of each address and each address "cluster" (captures behavior)
- Transaction features (fee, amount, number of outputs, etc)

Model

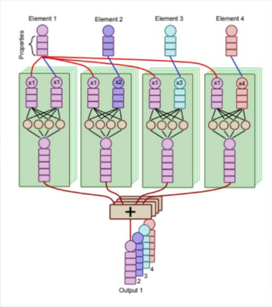
Feedforward Neural Network

- Sigmoid activation for each output
- Model takes in the details of a particular transaction and gives its predictions for which outputs are change addresses.
- Transaction features are fed directly into a 4-layer deep neural network.
- Input address features, to preserve input equivariance, are fed into permutation layers before being fed into the network itself.


input_clusters_features: InputLayer
output_clusters_features: InputLayer
permutation_layer_1: PermutationLayer
permutation_layer_2: PermutationLayer
input_address_features: InputLayer
output_address_features: InputLayer
concatenate_1: Concatenate
permutation_layer_3: PermutationLayer
permutation_layer_4: PermutationLayer
flatten_3: Flatten
generic_features: InputLayer
country_input: InputLayer
flatten_1: Flatten
flatten_2: Flatten
concatenate_2: Concatenate
dense_1: Dense
dense_2: Dense
dense_3: Dense
dense_4: Dense
dense_5: Dense

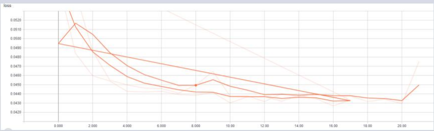
Permutation Layer

- Used to reduce overfitting by mandating that the network give consistent outputs for a re-ordering of the input addresses.
- Described in Guttenberg et al 2016.
- Enforces parameter sharing that keeps the network's output order consistent



Hyperparameter Search

- 2-6 hidden layers tested; accuracy changes were not significant; all were greater than 96%.
- The optimizer successfully optimized the loss function, so no optimizers other than Adam with Keras default parameters were used.
- More overfitting without the permutation layers.



Results

- High accuracies (>96%) reported for change address detection
- Possible information "leakage" that allowed the network to "cheat," since it was graded on the same heuristics that were indirectly used to generate the train data (particularly some information about the clustering of input addresses).
- Untested applicability to new Bitcoin data and changing practices involving the spending of Bitcoin.

Discussion

- This is only a partial solution to clustering Bitcoin addresses into those controlled by a single owner.
- However, change detection enables a heuristic clustering algorithm to include change addresses in the clusters without having to wait for future transactions to actually use the change addresses. For wallets with a lot of money, this may be long in the future.
- This model used training data derived mostly from another heuristic model, as opposed to completely empirically-verified data. This means it is susceptible to some of the same biases and errors.

Future Research

Some experimentation with graph representation learning was done. This has the advantage of being an end-to-end deep learning approach to clustering addresses without going through the intermediate steps such as change detection. Obviously, this would represent a substantial improvement over the current indirect and heuristic methods. (GraphSAGE)