Neural FCA

Kolotov Anton, мИИАД22

Dataset

Credit Card Approvals: UCI Machine Learning Repository, Kaggle

- 690 objects
- 14 attributes:
  - 6 binary
  - 4 numeric
  - 3 categorical
- Binary target – 44% positive objects => balanced target
- Metric: F1 score (pay more attention to positive class)
- Split data: 40% train, 20% validation, 40% test

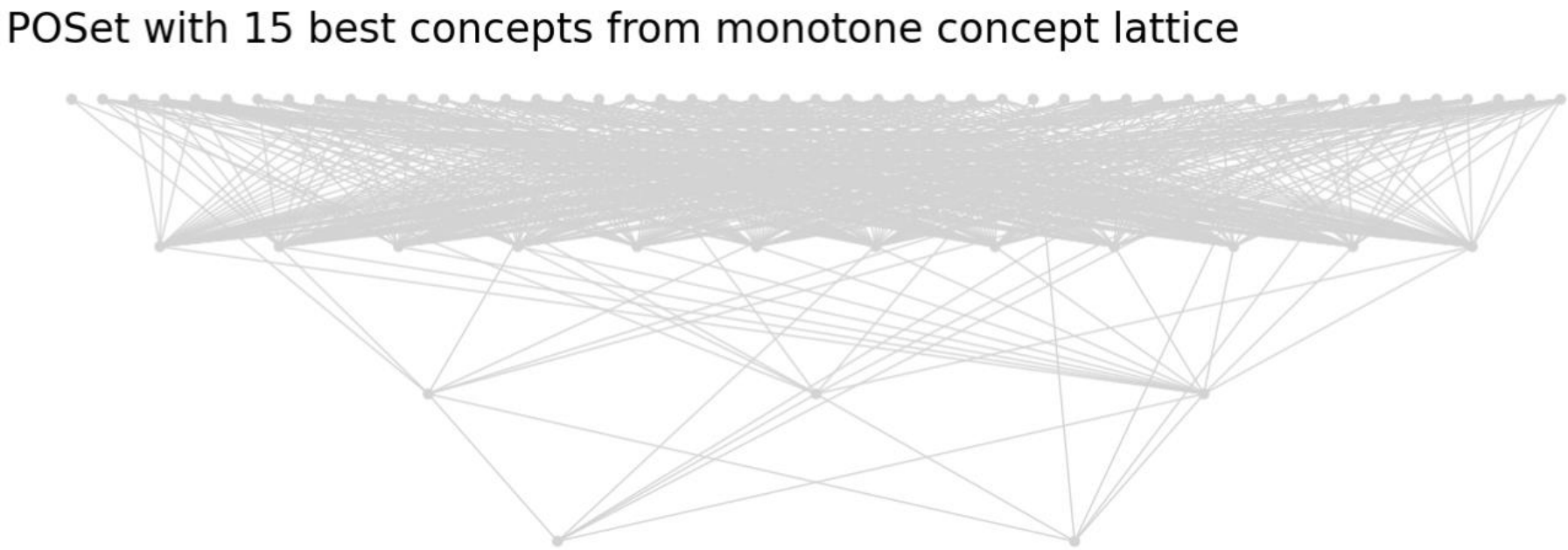
Binarization

- For categorical features let's use one-hot encoding
- For numeric features compare 2 strategies:
  - Select bins (using quantiles) $b_1, b_2, \dots, b_k$, then use interval $[b_i, b_j]$ - win
  - Select bins (using quantiles) $b_1, b_2, \dots, b_k$, then use interval $(-\infty, b_j]$
- Binary features let's keep the same
- For any attribute – use it and its negation (+0.02 F1 Score)

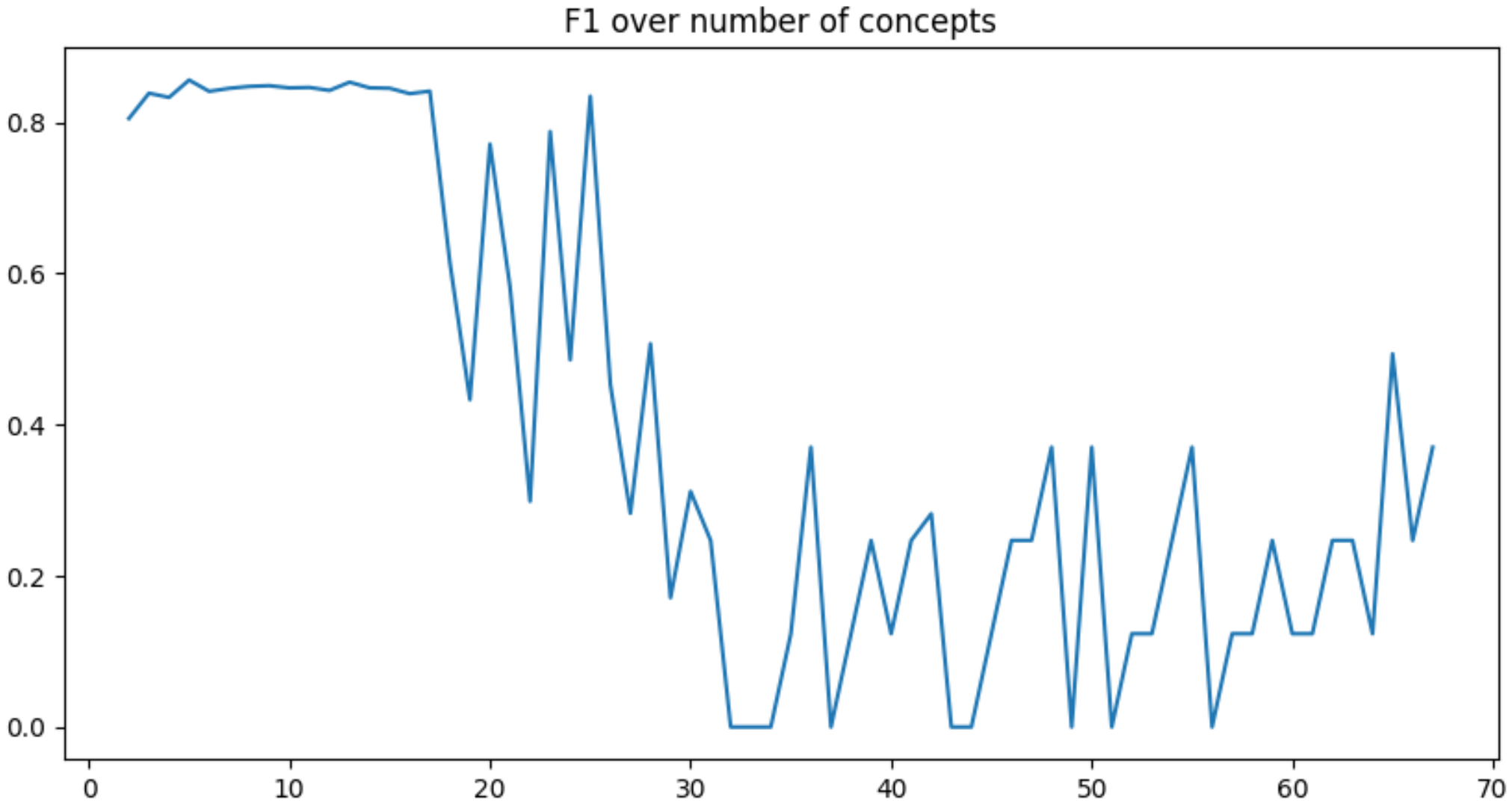
F1 Score	train	validation	test
$[b_i, b_j]$	0.839	0.843	0.828
$(-\infty, b_j]$	0.829	0.814	0.805

Concept Selection

- Extract concepts from train data context using Sofia algo (CbO too slow)
- Select k best concepts using F1 score
- Choose best k using validation dataset and F1 score



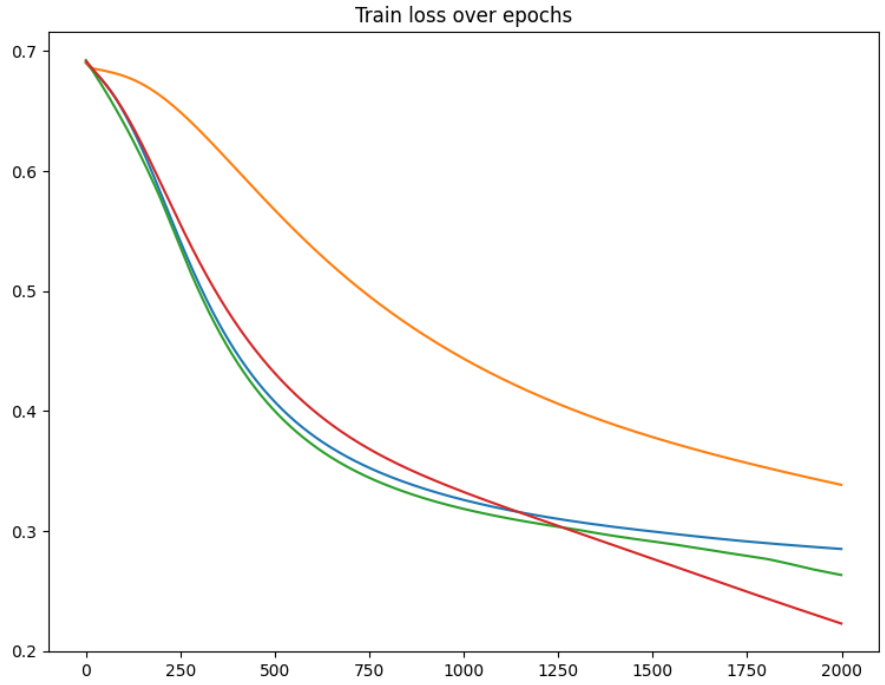
5 best concepts is enough

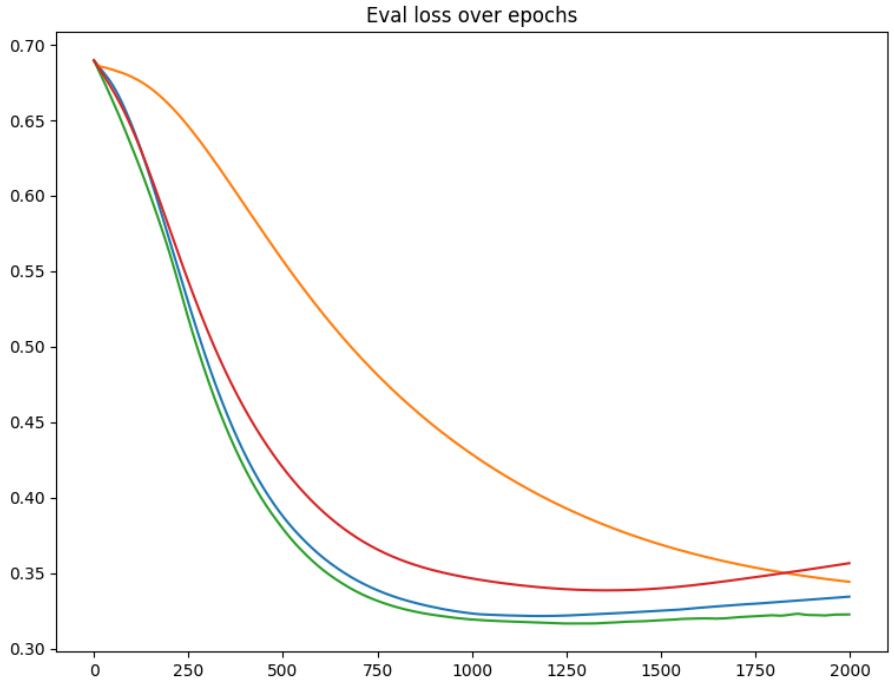


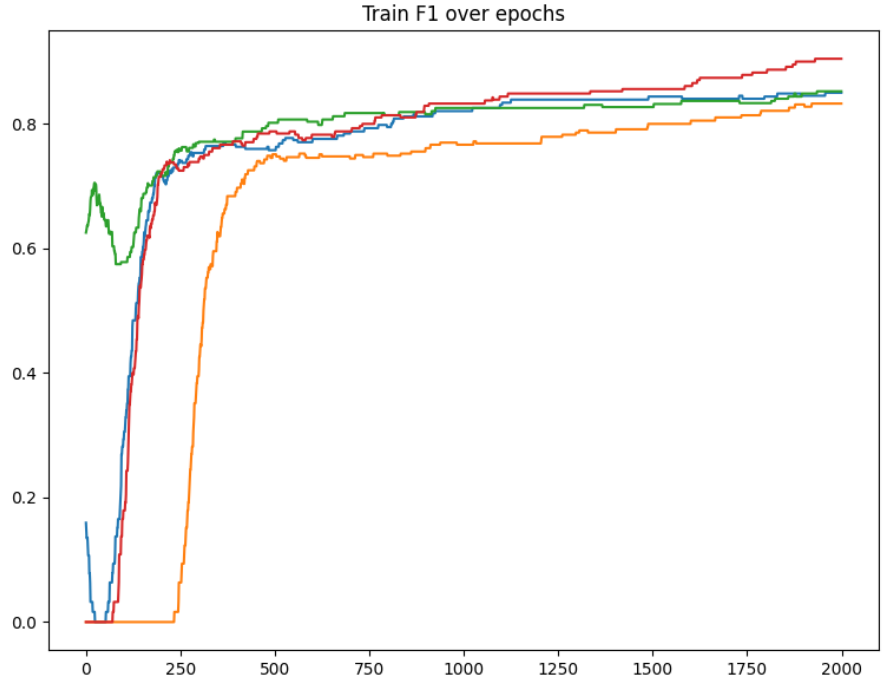
Neural Network

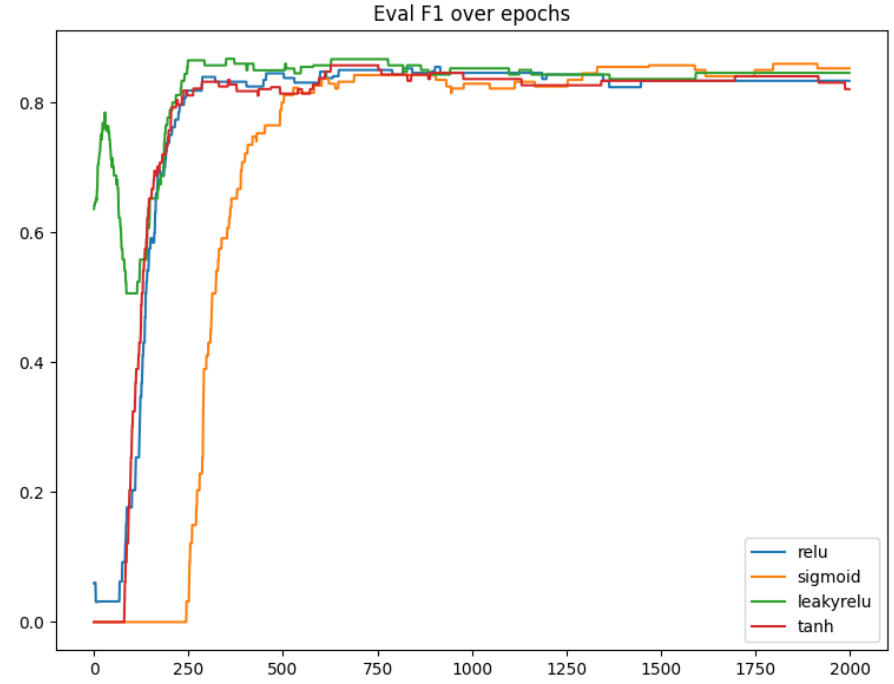
- Optimizer Adam, lr=3e-4, n_epoch=2000
- Compare different activation functions: LeakyReLU - best test quality

F1 Score	train	validation	test
ReLU	0.850	0.833	0.828
Sigmoid	0.833	0.852	0.812
LeakyReLU	0.852	0.846	0.833
Tanh	0.904	0.821	0.81


Train loss over epochs
0.7
0.6
0.5
0.4
0.3
0.2
0
250
500
750
1000
1250
1500
1750
2000

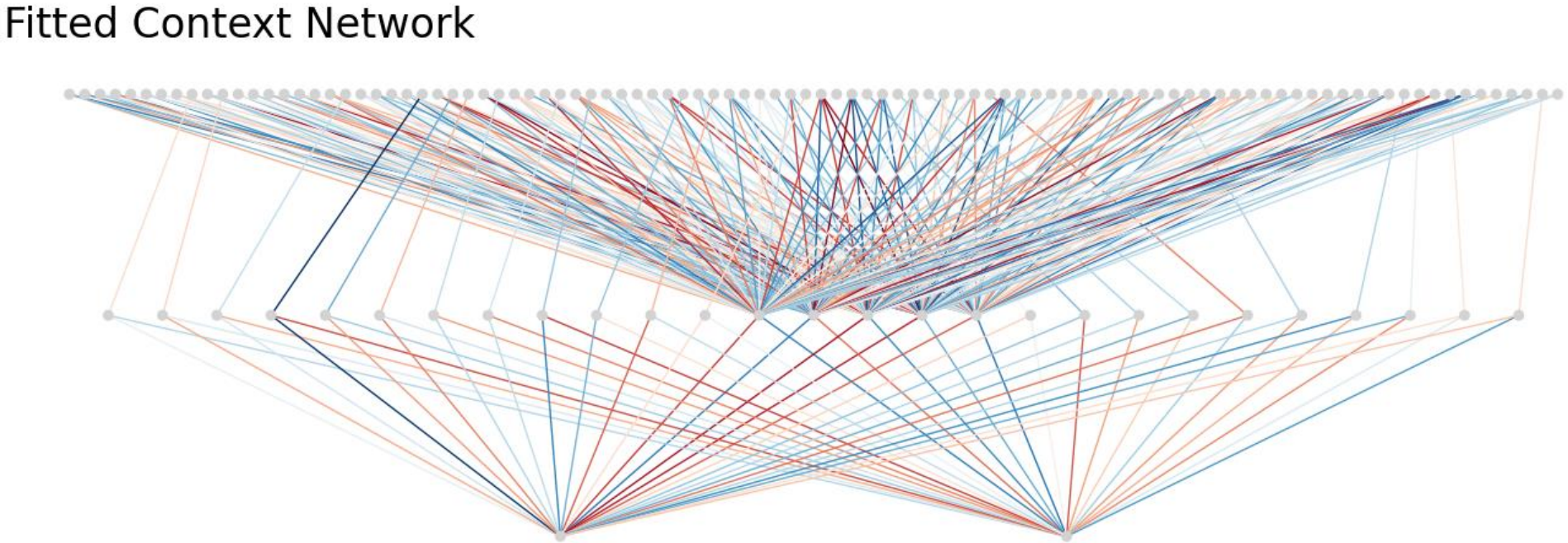

Eval loss over epochs
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0
250
500
750
1000
1250
1500
1750
2000


Train F1 over epochs
0.8
0.6
0.4
0.2
0.0
0
250
500
750
1000
1250
1500
1750
2000


Eval F1 over epochs
0.8
0.6
0.4
0.2
0.0
0
250
500
750
1000
1250
1500
1750
2000
relu
sigmoid
leakyrelu
tanh

Combining all – Best FCA-based NN

- Interval binarization for numeric features + add negations
- 5 best concepts
- LeakyReLU



Comparison with SotA approaches

model	train	validation	test
DT(max_depth=4)	0.871	0.835	0.808
Catboost(n_trees=17)	0.842	0.934	**0.869**
RF(max_depth=9, n_trees=28)	0.975	0.847	**0.832**
LogReg	0.835	0.857	0.790
KNN(n_neighbors=19)	0.747	0.789	0.772
ContextNetwork	0.851	0.846	**0.833**

Conclusion

- Binarization greatly expands the number of attributes (98 vs 14 initial)
- Building concepts is computationally expensive
- Tuning activation function may increase quality
- The FCA-based NN can outperform some classical methods and achieve SotA quality level
- It was fun to apply FCA-based neural network!