Which attributes matter the most for loan origination? A neural attention approach

Antonios Alexos[1], Sotirios Chatzis[2]

[1]Department of Electrical and Computer Engineering, University of Thessaly, Volos

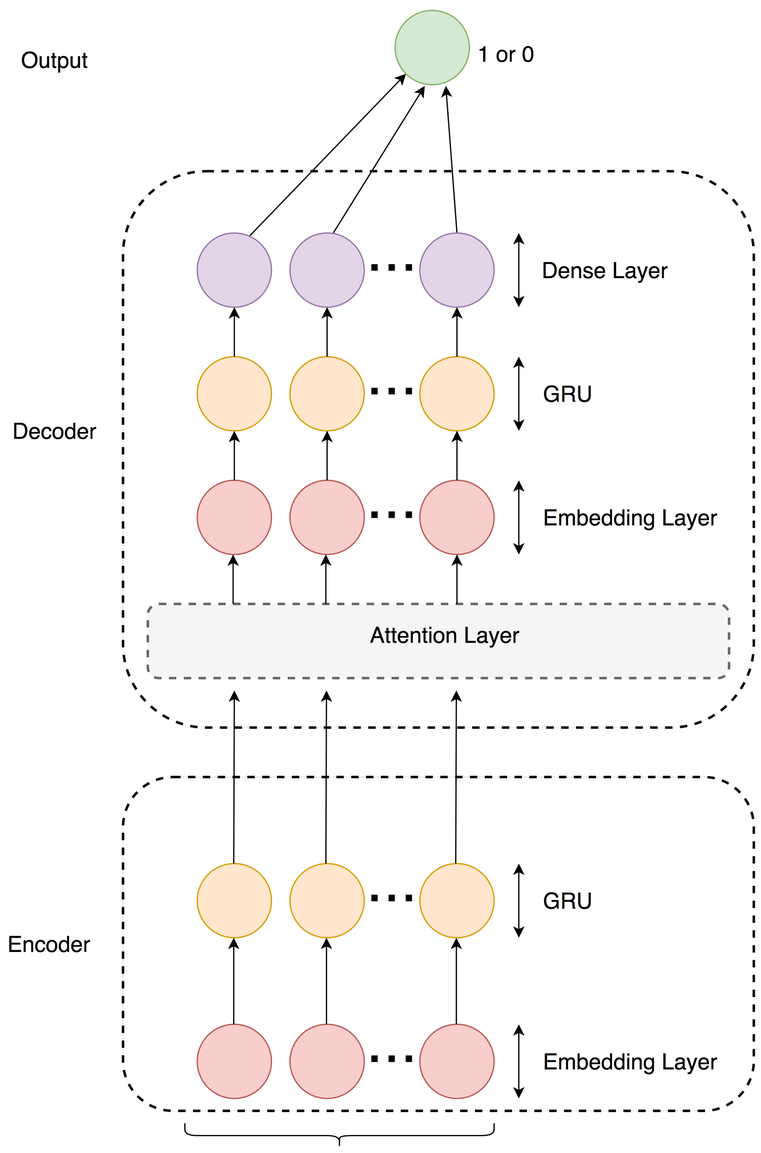
[2]Department of Electrical Enginnering, Computer Engineering and Informatics, Cyprus University of Technology

Introduction

- In this work we use an encoder-decoder with attention model to predict loan decisions and provide unique explanations for them.
- We propose a novel approach for inferring, which attributes matter the most, for making a decision in each specific individual loan case.
- We leverage concepts from neural attention to devise a novel feature wise attention mechanism.
- Our approach offers unique insights into the importance of various features, by producing a decision explanation for each specific loan case.
- Our novel mechanism generates decisions which are much closer to the decisions generated by human experts, compared to the existent competitors.
- Small introduction to Bahdanau attention:

Methodology

- In our work, we treat every observation(vector) loan data as a sequence of tokens, with each feature being treated as a token by the model.
- We make the assumption that the data are not sequential, and do not have any dependencies.
- This is a slight violation to the model.
- By making this assumption we explicitly consider sequential dynamics. The form of dynamics of dependencies is adaptive due to the attention mechanism.
- So this slight violation that exists, is not detrimental due to the attention mechanism.



Prediction Results

- In order to prove the superiority of our model we compare its prediction results with the results of a simple neural network. The simple neural network achieved a prediction score of **73.2%,** while our model achieved a prediction score of **89.6%.**

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_4 (Dense)              (None, 100)               2700
_________________________________________________________________
dropout_3 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_5 (Dense)              (None, 100)               10100
_________________________________________________________________
dropout_4 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 101
=================================================================
Total params: 12,901
Trainable params: 12,901
Non-trainable params: 0
_________________________________________________________________
```

	Yes	No
Yes	0	1914
No	0	5243

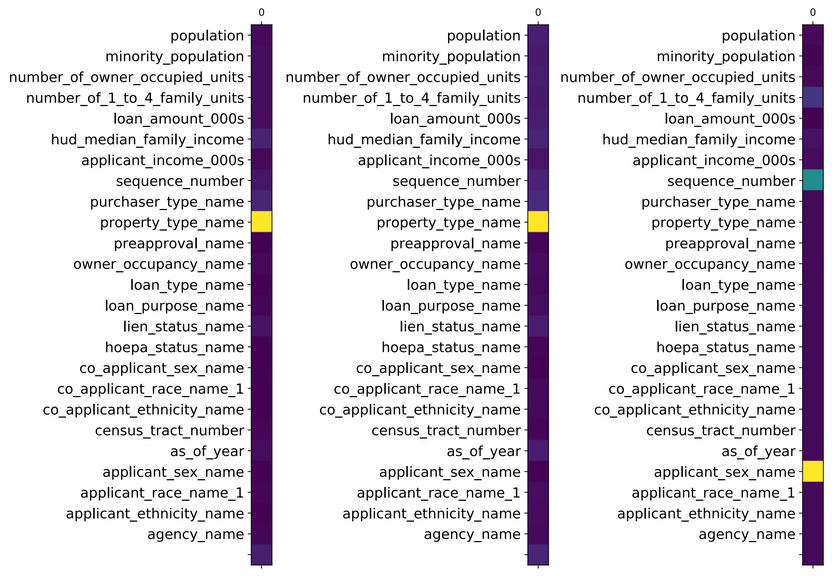
Confusion matrix of the simple Neural Network

	Yes	No
Yes	1307	559
No	179	5059

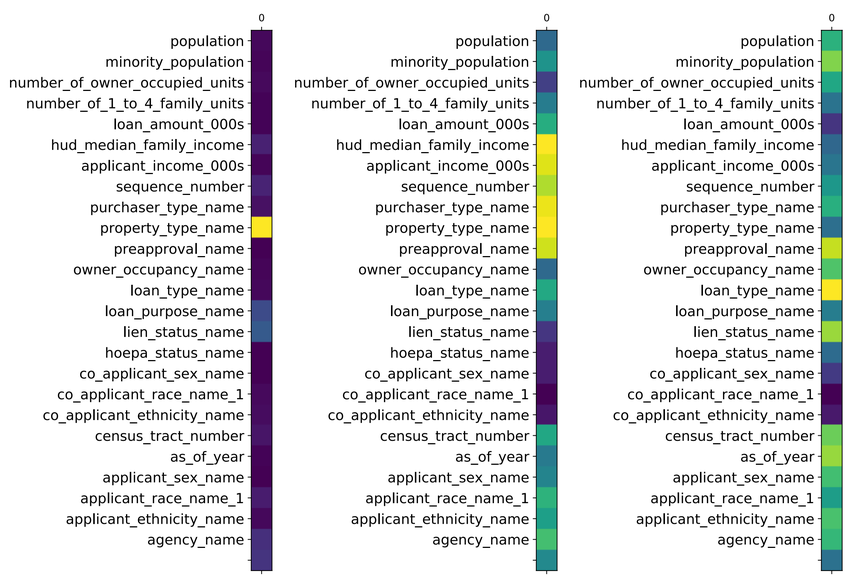
Confusion matrix of our model

Feature Importance Comparisons

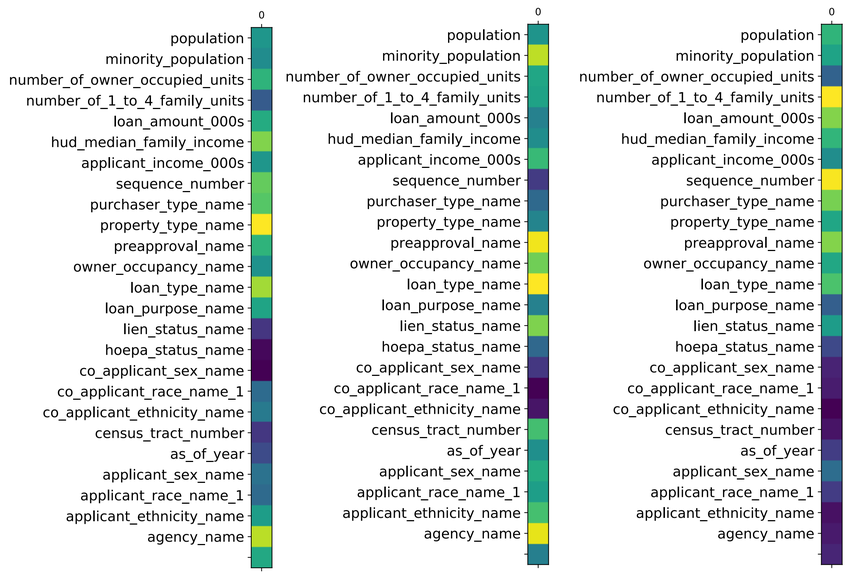
Here we present the feature importance results, which the Encoder-Decoder with Attention model produced, and compare them with the results of classical algorithms for feature importance.



Decision Tree **84.7%** | Extra Trees **88%** | LASSO



XGBoost **89%** | Our approach 0 | Our approach 1



Our approach 2 | Our approach 3 | Our approach 4

Conclusions

- The Encoder-Decoder with attention model has clearly other utilities apart from translating sentences in other languages. It has the ability to show unique feature importance for every observation.
- We have successfully achieved, to explain bank loan decisions to the applicants, by giving them unique explanations regarding their application.
- Through data analysis we have seen that the algorithm is not discriminating in the rejection cases. Features like race and ethnicity don not play a major role in a rejected case.

Future Work

- We will try to implement BERT on the data, regarding its unique
- Change the order of the feature and observe any changes on the accuracy.
- Train the model with the whole dataset in order to improve the accuracy score.
- Try to implement the idea to another dataset.