Github link - https://github.com/drindustrial/Name-classification

# 1. Introduction

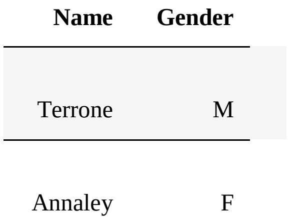
In this work was compared four different models for names classification by gender. All models can classify the gender by reading the name using a character level model.

# 2. Data

## 2.1 Data representation

The dataset is a collection of strings of variable length. Data represents by table with two columns (name and gender) and 83 288 rows,

Example:



| Name | Gender |
|---|---|
| Terrone | M |
| Annaley | F |

## 2.1 Data preprocessing

First part of preprocessing was filtered out names what belongs to 2 genders, in this part was filtered out 13 112 rows from train dataset and 826 rows from test dataset. After the filtering all names was transformed to machine-readable format, this format is as follows

```
[ 5 38 35 52 27 28 31 46 34  0  0  0  0  0  0]
```

Where each number is unique integer identifier for every charter what represent in train dataset and 0 is placed after each word less than 15 characters so that each sample has a length of 15.

# 3. RNN models

Was trained few RNN models with different hyperparameters, the following results were obtained:

1. LSTM layer (size 14), 1 Dense Layer (size 8), optimizer adam with learning rate 0.008. Overfiring starts after 80 epoch’s, Training Accuracy:  0.8736 and Testing Accuracy: 0.8245

2. LSTM layer (size 15), Dropout(0.2), 1 Dense Layer (size 2), optimizer adam with default parameter’s, 100 epoch’s. Training Accuracy:  0.8632 and Testing Accuracy: 0.8181

3. GRU layer (size 18), Dropout(0.15), 1 Dense Layer (size 18), optimizer adam with default parameter’s, 100 epoch’s. Training Accuracy:  0.8738 and Testing Accuracy: 0.8239

To avoid too much overfitting was applied l1 l2 regularization and Dropout layers. Every model has about 2 400 parameters and best one is model with 1 LSTM layer (size 16), 1 Dense Layer (size 16)

## 4. Not RNN models

For comparison RNN model with other types of models was trained and the following results were obtained:

Logistic Regression - Testing Accuracy:  0.6343

1 Dense layer (size 15), optimizer adam with default parameter’s, 50 epoch’s. Training Accuracy:  0.8382 and Testing Accuracy:  0.8013

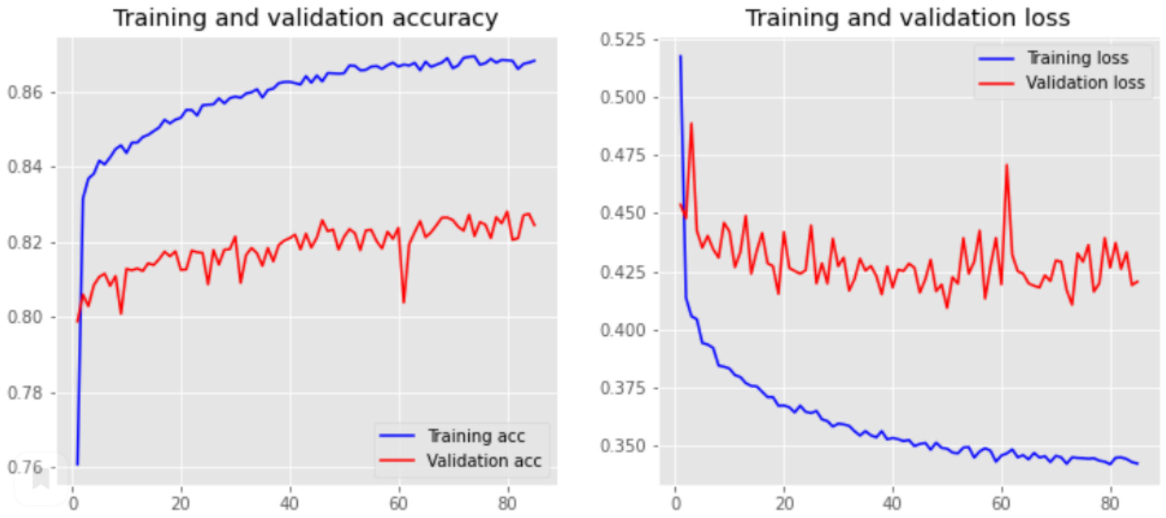
1 Global Max Pooling, 2 Dense layer (size 35), optimizer adam with default parameter’s, 50 epoch’s. Training Accuracy:  0.7144 and Testing Accuracy:  0.6920

All three of these models have the same number of paramiters as RNN models – 1500 +- 100, to avoid too much overfitting was applied l1 l2 regularization and Dropout layers.  From those four models the best is fully connected NN.

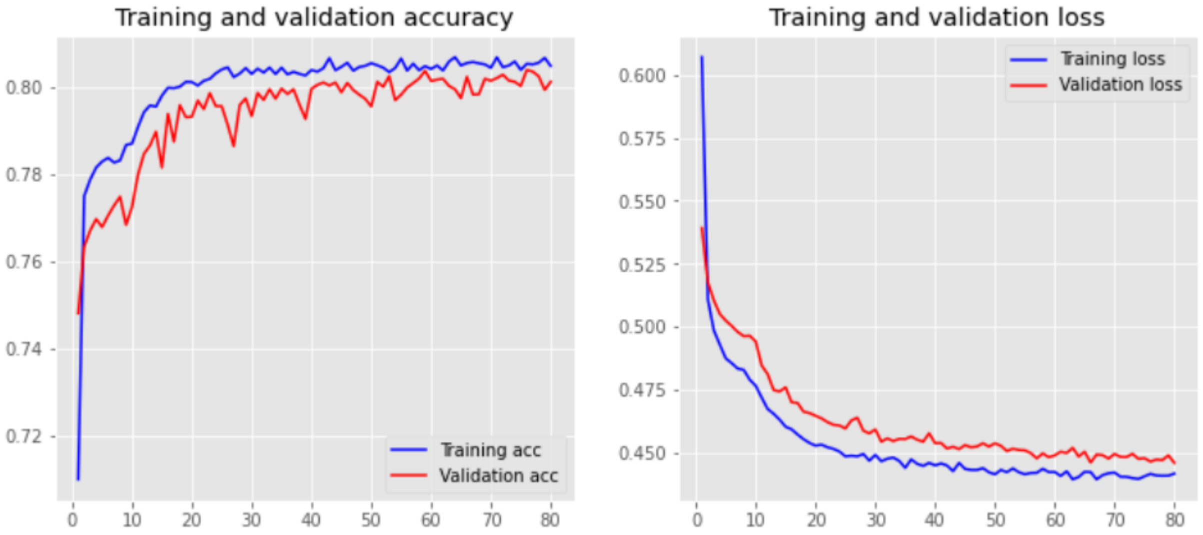
## 5. Graphs

In this paragraph presented graphs for three models:

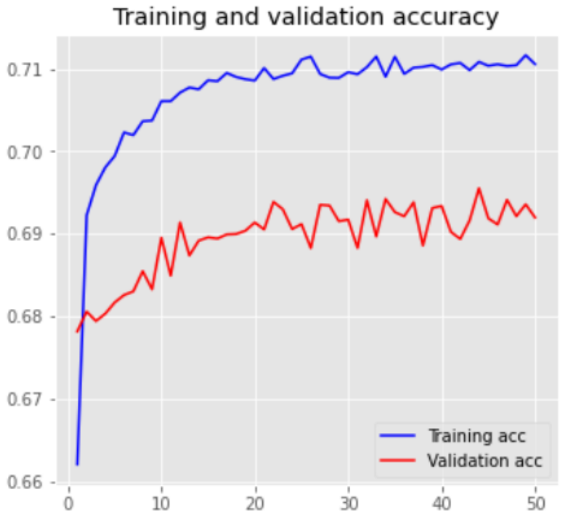
1 LSTM layer (size 16), 1 Dense Layer (size 16)



1 Dense layer (size 15)



Global Max Pooling, 2 Dense layers (size 35)



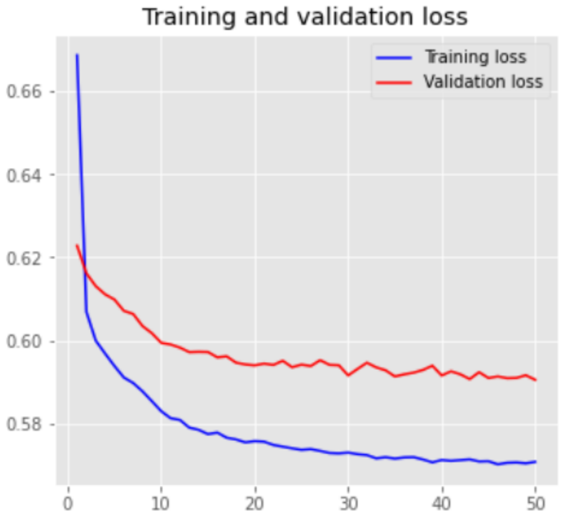
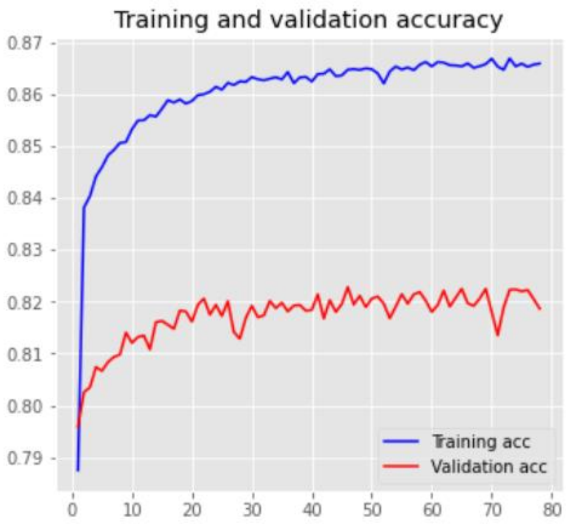


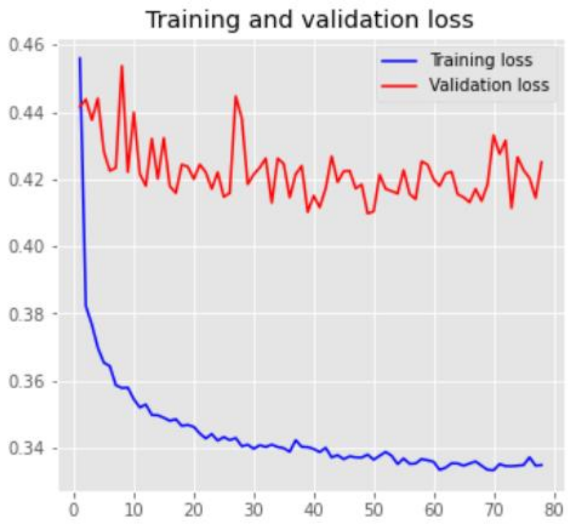
Table for comparison these three models:

| Model | Train accuracy | Test accuracy |
|---|---|---|
| LSTM | 0.8736 | 0.8245 |
| Dense | 0.8382 | 0.8013 |
| Max pooling and Dense | 0.7144 | 0.6920 |

From this table we can see that LSTM based model is better in terms of accuracy. Nevertheless, if we reduce number of parameters LSTM model will compute faster and still have same accuracy as fully connected model.

Here is model with almost twice less parameters, 1 GRU layer (10 size). Training Accuracy:  0.8682 and Testing Accuracy:  0.8187





## 6. Conclusion

At the end we found what for names classification task is better to use RNN model especially with GRU layers, this model can have not much parameters and rich 0.81 accuracy.