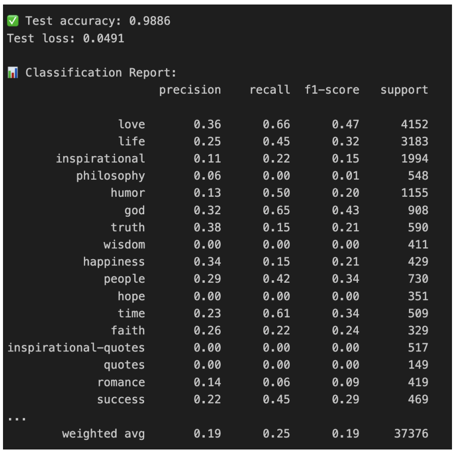
MAIS 404: Delivery 2 - Grégoire Trinh Ngo & Iris Bisson

1. **Problem statement:** We changed the problem from the last deliverable: we will take Kaggle's 500K quote dataset and develop an ML capable of predicting the (main topic/class) of that quote (e.g. life, love, death, philosophy, etc.). This will (most likely) be supported by a neural network architecture.
2. **Data Preprocessing:** As mentioned, we will use Kaggle's 500K quote dataset composed of three columns (quote, author, category) due to the size of the dataset we have considered to remove some rows (including, for now, rows that only have “love” in the “category” column since it is quite recurrent and will produce an imbalanced dataset). 2492 duplicated quotes were also removed, 63 nan rows (either nan quotes or nan categories) as well as one non-str quote were removed. We also applied basic harmonization of punctuation on all categories (lowercased, trimmed, handling of messy spacing, etc.), same goes for quotes (removal of trailing space, handling of unnecessary sequence of space). This part is important so as to not have multiple duplicated quotes or similar categories (e.g. “Love”, “Love ”, “love”, “love ”).
3. **Machine learning model:** We plan on using Keras for implementing our neural network, we found NNs to be the most efficient solution for this project as NNs are capable of taking in vast amounts of data (through integrated text vectorizer layer, it is more digestible) and determine patterns more efficiently than other models (at least that’s what we found to be the case). We plan on having an input-TextVectorization-embedding-Dense (ReLU activation)-output layer architecture.
   a. Based on lecture notes, there will be a 80%-10%-10% training/validation/test split. We have yet to figure out how to deal with data imbalance in the “category” column (“love” is one of them but there are multiple such as “relationship”, etc.). We have chose ReLU as our activation function as this is what’s most common for hidden layers’ activation function. Our model seems to be overfitted, this is expected since we switched our initial problem quite recently, we suspect this may be because of data imbalance and improper training/architecture of our model (we are waiting on the lecture on Neural Networks to continue with our work). Another hypothesis is that the dataset is imbalanced with a long tail of rare classes, so the model memorizes common labels and fails on the rest. We faced mediocre results on the first iteration and intuitively thought of running multiple epochs as well as batch training our model since the dataset was too big to train all at once. We plan on implementing more techniques in the future but primarily hope these techniques will be covered during the lectures!
4. **Preliminary results:** As mentioned, our model is overfitted (test accuracy: 0.9886, test loss: 0.0491), however the classification report tells another story:

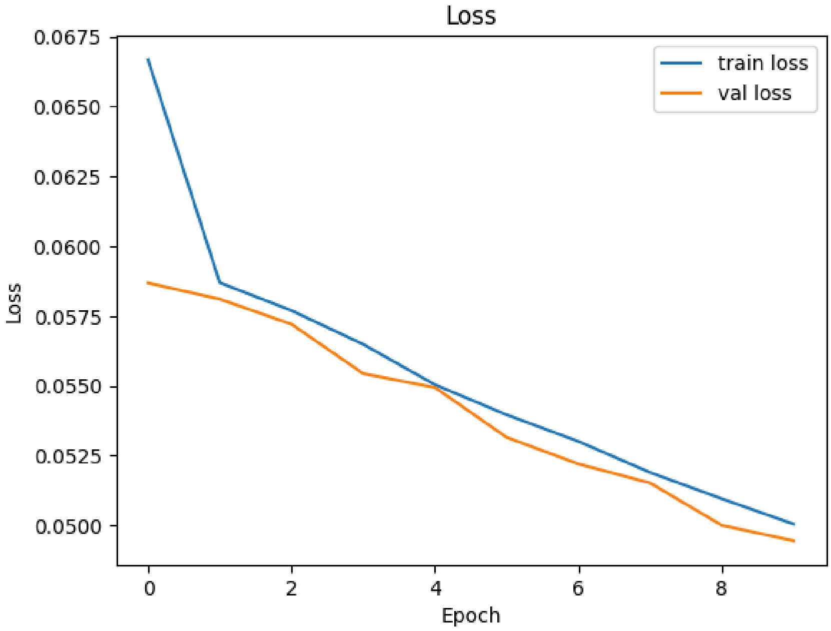


```
✅ Test accuracy: 0.9886
Test loss: 0.0491

📊 Classification Report:
                   precision    recall  f1-score   support

            love       0.36      0.66      0.47      4152
            life       0.25      0.45      0.32      3183
   inspirational       0.11      0.22      0.15      1994
      philosophy       0.06      0.00      0.01       548
           humor       0.13      0.50      0.20      1155
             god       0.32      0.65      0.43       908
           truth       0.38      0.15      0.21       590
          wisdom       0.00      0.00      0.00       411
       happiness       0.34      0.15      0.21       429
          people       0.29      0.42      0.34       730
            hope       0.00      0.00      0.00       351
            time       0.23      0.61      0.34       509
           faith       0.26      0.22      0.24       329
inspirational-quotes   0.00      0.00      0.00       517
          quotes       0.00      0.00      0.00       149
         romance       0.14      0.06      0.09       419
         success       0.22      0.45      0.29       469
...
    weighted avg       0.19      0.25      0.19     37376
```

The low precision and high (imbalanced) recall tells us that it is misleadingly confident: it memorized the most frequent patterns in training but fails to generalize to less common classes.

Next steps: We will try to include multi-label classification with class weights instead of binary, maybe using regularization to penalize unnecessary weights. We will talk to our TPM to see if there is anything to be done.

Appendix:


Loss
train loss
val loss
0.0675
0.0650
0.0625
0.0600
0.0575
0.0550
0.0525
0.0500
Loss
0
2
4
6
8
Epoch