Spoken and Written Language Processing - POE

Assignment 3. Language identification (Sentence Classification)

Gerard Martín Pey

Jose Àngel Mola Audí

3^{h} May 2023

Assignment 3. Language identification (Sentence Classification)

Language identification

Edit the character-based RNN Baseline to create a new notebook with an additional contribution to the analysis, optimization, or comparative study of the proposed model and task. Include the modified code with your comments, results, tables, graphs, and conclusions in the PDF report. Compare the results of at least 6 models.

First, we are going to train different models using GRU and LSTM architecture. The configurations that we have decide to run are:

- **BASELINE:**
  - LSTM
  - 1 layer
  - Unidirectional
- **M1:**
  - GRU
  - 1 layer
  - Unidirectional
- **M2:**
  - LSTM
  - 1 layer
  - Bidirectional
  - Hidden size =
- **M3:**
  - GRU
  - 1 layer
  - Bidirectional
  - Hidden size =

Results have been:

MODEL	CV TR ACCURACY	CV TR TIME	CV VAL ACCURACY	#PARAMS	FM TR ACCURACY	FM TR TIME
BASELINE	98.6%	1456s	92.7%	1.081.835	98.729%	1648s
M1	98.8%	1271s	92.7%	999.403	98.829%	1446s
M2	99.8%	2257s	93.7%	1.471.723	99.841%	2558s
M3	99.7%	2001s	93.3%	1.306.859	99.610%	2300s

We can see how the M2 has a higher accuracy in both train and validation, which makes it the better model. We can see how it also has the highest number of parameters, as LSTM has more parameters than GRU, which makes it more computationally expensive to train. M1 and M3 have similar accuracies but M1 has a lower number of parameters as it is a unidirectional GRU and we must remember that the fact of having a bidirectional model multiplies by two the number of parameters of the hidden layer. In our case, we will take into account the accuracy and not so much the complexity as a differentiating factor, so we choose M2 as our best choice.

We define the following models:

- **M4:**
  - LSTM
  - 1 layer
  - Bidirectional
  - Mean pool-Max pool concatenation
- **M5:**
  - LSTM
  - 1 layer
  - Bidirectional
  - Mean pool-Max pool addition
- **M6:**
  - LSTM
  - 1 layer
  - Bidirectional
  - Mean pool-Max pool addition
  - Drop-out (p = 0.4)

Results have been:

MODEL	CV TR ACCURACY	CV TR TIME	CV VAL ACCURACY	#PARAMS	FM TR ACCURACY	FM TR TIME
M4	99.9%	2818s	93.3%	1.592.043	99.809%	3246s
M5	99.9%	2827s	93.9%	1.471.723	99.848%	3254s
M6	98.7%	2835s	94.0%	1.471.723	98.670%	3264s

We can see how M5 has a higher accuracy in train and close to the highest accuracy in validation. We can see that M6 has the lowest accuracy in training, this is due to the dropout layer which has lowered this accuracy and thus prevented some overfitting. We can see how M4 and M5 have the same architecture but they vary due to the fact that M4 makes a concatenation after the mean and max pool. We can also see that the only difference between M5 and M6 is the dropout layer mentioned above. The choice of the best model is based on the fact that M6 has less impact of overfitting because a dropout layer has been added. We have also seen how addition performs better than concatenation.

Now we will increment the number of layers of M6:

- **M7:**
    - LSTM
    - 2 layers
    - Bidirectional
    - Mean pool-Max pool addition
    - Drop-out (p = 0.4)
- **M8:**
    - LSTM
    - 3 layers
    - Bidirectional
    - Mean pool-Max pool addition
    - Drop-out (p = 0.4)

Results have been:

MODEL	CV TR ACCURACY	CV TR TIME	CV VAL ACCURACY	#PARAMS	FM TR ACCURACY	FM TR TIME
M7	99.4%	5040s	94.1%	3.048.683	99.413%	5740s
M8	99.0%	7297s	93.6%	4.625.643	99.018%	8220s

We can see that both models have a high accuracy value in training but it decreases in validation, which indicates that this model has overfitting. It is logical that the three-layer model has more parameters and therefore takes longer to run. Therefore, we would choose model 7 as it has a higher validation accuracy and a lower number of parameters and execution time.

The models that we have executed are:

- **M9:**
    - LSTM
    - 2 layers
    - Bidirectional
    - Mean pool-Max pool addition
    - Embedding size = 64
    - Hidden size = 256
    - Batch size = 256
    - Drop-out (p = 0.4)
- **M10:**
    - LSTM
    - 2 layers
    - Bidirectional
    - Mean pool-Max pool addition
    - Embedding size = 128
    - Hidden size = 256
    - Batch size = 256
    - Drop-out (p = 0.5)

- **M11:**
  - LSTM
  - 2 layers
  - Bidirectional
  - Mean pool-Max pool addition
  - Embedding size = 256
  - Hidden size = 256
  - Batch size = 256
  - Drop-out (p = 0.5)
- **M12:**
  - LSTM
  - 2 layers
  - Bidirectional
  - Mean pool-Max pool addition
  - Embedding size = 512
  - Hidden size = 256
  - Batch size = 256
  - Drop-out (p = 0.5)
- **M13:**
  - LSTM
  - 2 layers
  - Bidirectional
  - Mean pool-Max pool addition
  - Embedding size = 128
  - Hidden size = 128
  - Batch size = 256
  - Drop-out (p = 0.6)
- **M14:**
  - LSTM
  - 2 layers
  - Bidirectional
  - Mean pool-Max pool addition
  - Embedding size = 256
  - Hidden size = 128
  - Batch size = 256
  - Drop-out (p = 0.4)
- **M15:**
  - LSTM
  - 2 layers
  - Bidirectional
  - Mean pool-Max pool addition
  - Embedding size = 64
  - Hidden size = 256
  - Batch size = 256
  - Drop-out (p = 0.2)

And results have been:

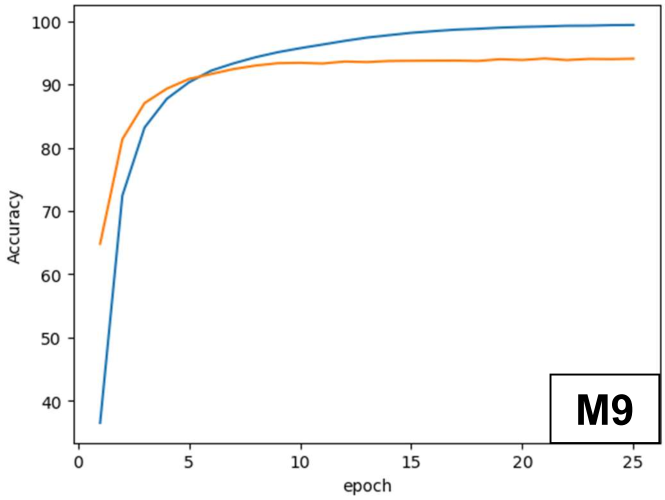
MODEL	CV TR ACCURACY	CV TR TIME	CV VAL ACCURACY	#PARAMS	FM TR ACCURACY	FM TR TIME
M9	99.4%	5040s	94.1%	3.048.683	99.413%	5740s
M10	99.1%	5158s	93.8%	3.871.467	99.146%	5820s
M11	98.9%	5322s	93.4%	5.517.035	98.903%	6035s
M12	98.6%	5713s	93.0%	8.808.171	98.616%	6457s
M13	96.4%	2696s	92.7%	2.103.275	96.426%	3035s
M14	97.7%	2759s	92.4%	3.617.771	97.791%	3108s
M15	99.7%	5087s	93.9%	3.048.683	99.695%	5753s

We can see that the models with the best performance in the validation set are M9, M10 and M15. However, these models also have a high accuracy in the training set, but both accuracies are similar and do not differ much, so we could say that there is no overfitting, since it is normal that the model behaves well with the data it has seen in the training, but in the data it has not seen it also does really well. However, we have been able to see that applying the dropout has decreased the training accuracy and increased the validation accuracy, so if we were to apply some technique to prevent overfitting, our models could be better.

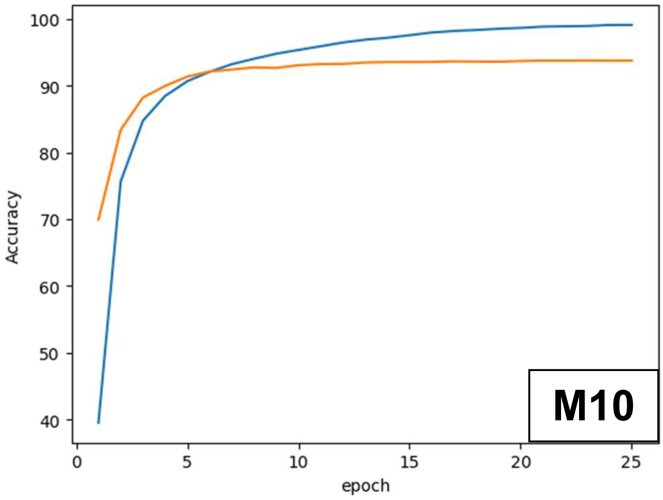
We can see that M13 is the one with the lowest training accuracy (it is the model with the highest dropout layer probability), but it is not among the lowest in the validation. We can see that the fact of having a higher number of parameters does not make the training accuracy higher, in fact the highest accuracy has about 3M parameters. Therefore, although it has a very high training accuracy, it also has the highest validation accuracy, so it could be a good model. We have also seen that models that have a hidden size = 128 have a lower validation accuracy than others that have a hidden size = 256.

If we look at the curves (next page), we can see that generally from 5 epochs onwards the training accuracy exceeds the validation accuracy. We can also see that M13 is the model with the smallest difference between accuracies. If we had to choose between any model, we would most probably choose M9 as it has the highest validation accuracy and the difference between validation and training accuracies is close to 5%, which is an adequate value to consider that there is really no overfitting, and the model predicts well data not seen in the training. However, as we have said before, it would be good to use some technique to make this difference between accuracies smaller and increase the validation accuracy.

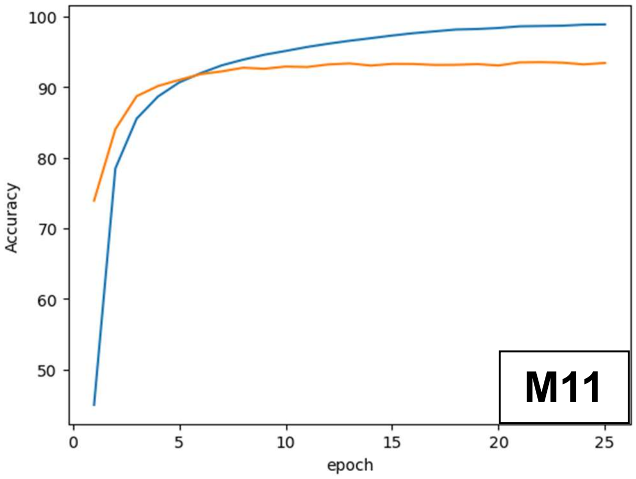
Assignment 3. Language identification (Sentence Classification)



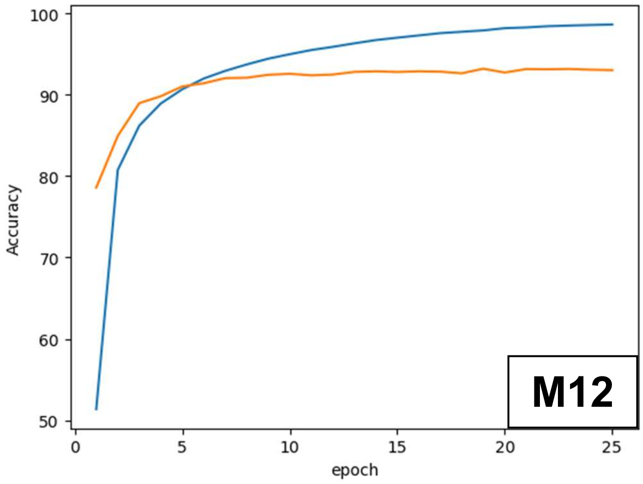
M9



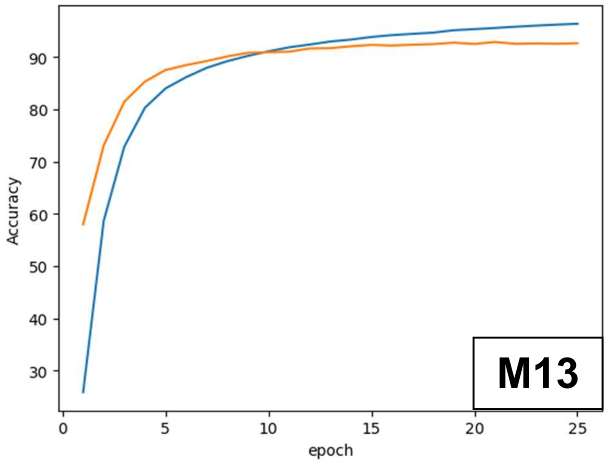
M10



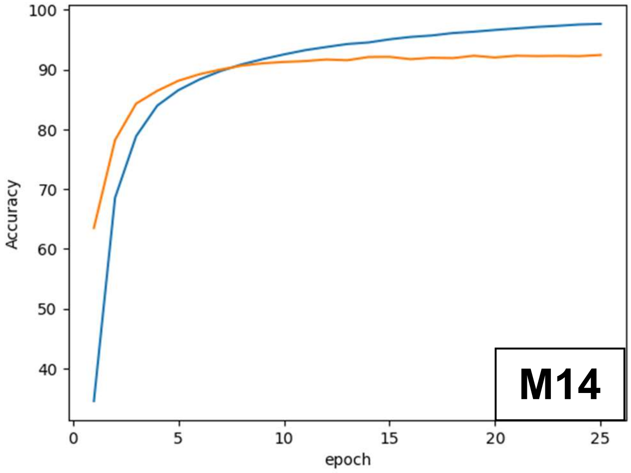
M11



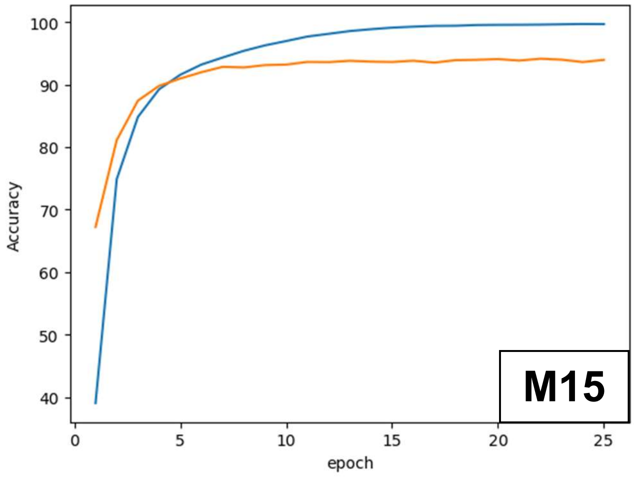
M12



M13



M14



M15