Movie genre classification

Vadym Dudarenko(444820)

Outline:

- short introduction to the analyzed problem
- dataset description
- the structure of the models
- list of the parameters, which are optimized
- model performance metrics with justification
- results
- conclusions
- references

Short introduction to the analyzed problem

Movies are one of the most popular means of entertainment. There are large volumes of movie data being generated and shared on the internet every second. The genre of a movie can be deciphered from its synopsis much of the time.

Dataset description

Dataset contains next column: name, genre, released_year, poster, language, director, domain, duration, synopsis (description), trailer, cast, url, id.
There are 10,254 movies (observations). Genre of movies are multi-label. So, this classification is Multi-label Classification problem.

I will try to make genre prediction using description of movies. It is a text classification.

Link:https://data.world/opensnippets/movies-dataset-from-allmoviecom/workspace/file?filename=movies_dataset_from_allmovie.json

The structure of the models

Combining of two models

Use the output of the CNN as the input to the LSTM. This allows the LSTM to learn features from the input data that have been learned by the CNN.

I made a prediction for each of model separately, than optimized with the hyper parameters. After this, I combine two model to see how results might be improved.

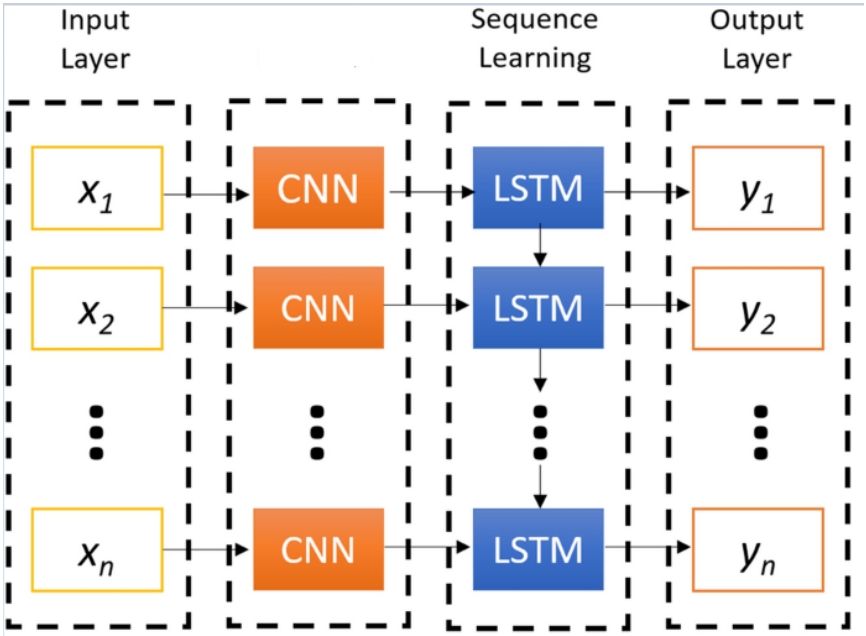

Figure 1. Sequence combination LSTM+CNN

In our case, input is text. Output is genre of movies or one of the tags.

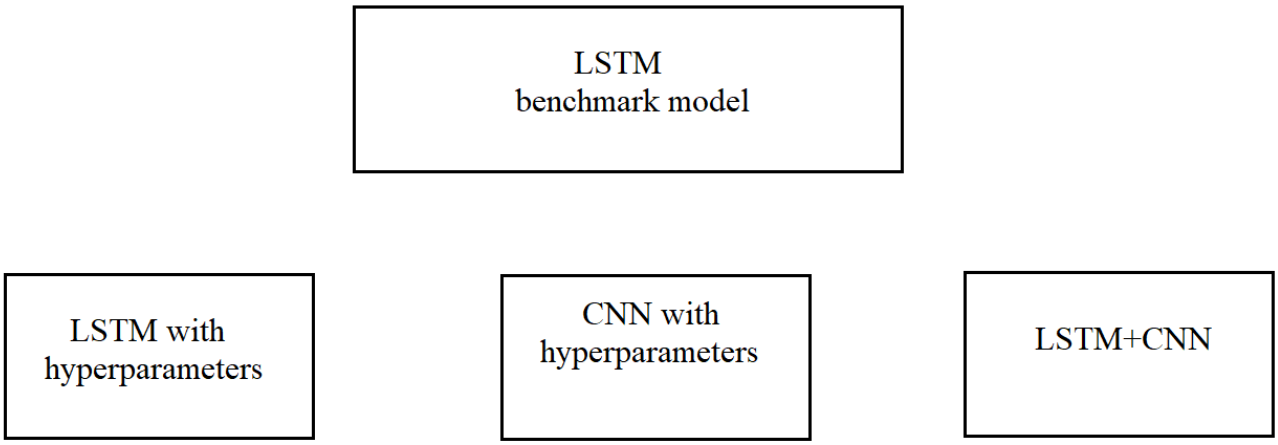

Figure 2. Benchmark model

List of the optimize parameters for LSTM:

- NUMBER OF NODES AND HIDDEN LAYERS - 30
- NUMBER OF UNITS IN A DENSE LAYER -41
- DROPOUT - 0.2
- OPTIMAZER - rmsprop
- ACTIVATION FUNCTION - sigmoid
- LEARNING RATE – 0.1
- MOMENTUM – 0.2
- NUMBER OF EPOCHS - 5
- BATCH SIZE - 32

List of the optimize parameters for CNN:

- Convolution activation: tanh
- Pooltype: average
- Dense Layer: size - 41
- Dropout: 0.2

Model performance metrics with justification

I check the result of confusion matrix, but for assessment. I use **F1-score**. Calculate metrics globally by counting the total true positives, false negatives and false positives. This is a better metric when we have **class imbalance.**

Results

	LSTM	LSTM +	CNN	CNN+	LSTM+CNN
F1	0.215	0,261	0.29	**0.34**	0,23
Recall	0.2565	0.274	0.31	**0.51**	0,35
Precision	0.187	0.202	0.28	**0.25**	0.17
Loss	0.12	0.121	0.012	**0,1531**	0,13

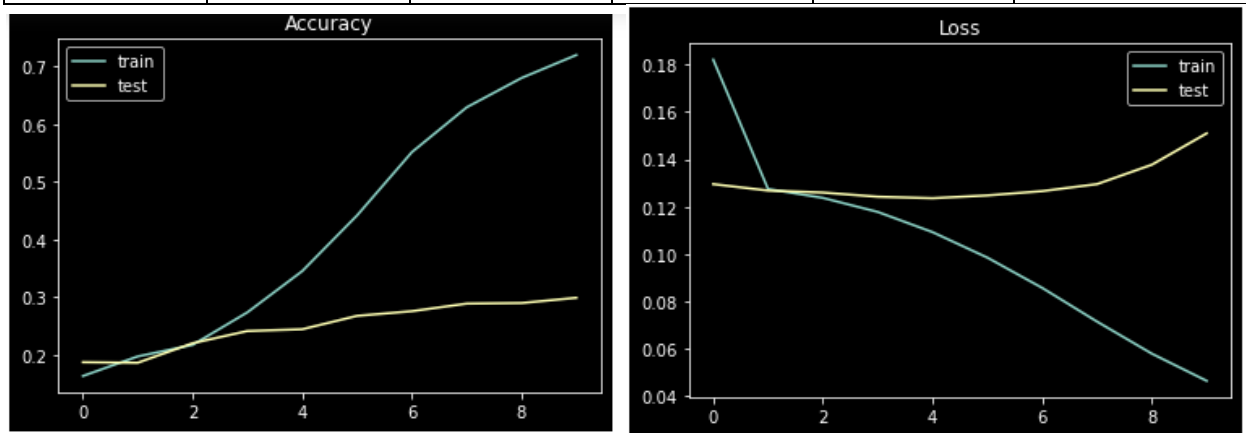

Figure 3. Accuracy and loss results

```
1/1 [==============================] - 0s 100ms/step
Original tags --> ['Action,Comedy']
Predicted tags --> ['action' 'comedy' 'comedy drama']

   1/1 [==============================] - 0s 38ms/step
   Original tags --> ['Drama']
   Predicted tags --> ['drama']

1/1 [==============================] - 0s 31ms/step
Original tags --> ['Mystery']
Predicted tags --> ['mystery' 'science fiction' 'thriller']
```

Figure 4. Examples of test prediction

Conclusion

This dataset is big and has a lot of different tags(genres). It is totally imbalance and over fitted. The final best micro f1 on test was 0.34, which CNN model showed. In totally the result do not show a good performance, because there are a lot of different number of combination genres that make harder to have an accurate results.

References

1. https://data.world/opensnippets/movies-dataset-from-allmoviecom
2. https://medium.com/analytics-vidhya/imdb-movie-genre-tag-prediction-4ee71a0aa9bd
3. https://www.simplilearn.com/machine-learning-projects-for-beginners-article
4. https://www.kaggle.com/code/hamzamanssor/film-genre-classification-using-lstm
5. https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/
6. https://medium.com/@mixanyy/different-ways-to-combine-cnn-and-lstm-networks-for-time-series-classification-tasks-b03fc37e91b6