COMP650 – Deep Learning Project Report

Title: Text Classification of BBC News Articles Using CNN

Student Name: Anahita Darban

Student ID: 10317413

Date: August 10, 2025

Problem Statement

The problem of text classification that is addressed in this project belongs to the domain of Natural Language Processing (NLP) and involves the classification of the news articles of the BBC to one of five categories: business, entertainment, politics, sport, and tech. Text classification is an essential issue in the field of natural language processing (NLP) since it gives rise to a wide range of tasks in other fields such as content recommendation, news filtering, and sentiment analysis. The data that will be used in this project is the BBC News Classification Dataset consisting of labeled text data. The articles in the dataset are well defined into categories, and therefore it is favorable to be used in supervised learning process. The primary purpose of the present project is to develop a deep learning model that will learn the underlying pattern and correctly categorizing the corresponding news article among the unseen news articles.

Model Overview

In this task, I was trying to implement a Convolutional Neural Network (CNN). Although CNNs are typical in image processing, they have been found to work on some NLP specific problems, particularly text classification. The model accepts the tokenized and padded news articles as input and uses a dense vector representation and an embedding layer to represent news articles and then uses a 1D convolutional layer to identify the local n-gram features. The global max pooling layer is followed to reduce the most prominent features in the sequence of texts. Lastly, a probability about the five classes in news is yielded by dense layers, with a SoftMax output layer. The selection of CNN instead of recurrent models was preconditioned by its prevalence of the training rate and advancing workings on short text and medium texts.

Implementation Summary

The TensorFlow framework together with a Keras API was used to implement the model. Data processing involved the removal of punctuation and low casing along with tokenization with a Keras tokenizer to restrict the top 5,000 words. Sequences were all padded to equal length to create consistency of the input size. The architecture is designed such that it includes an embedding layer with 128 output dimensions and 1D convolutional layer that contains 128 filters and the kernel dimension of the 1D convolution is 5. The dimension of the data was reduced via a global max pooling layer, then dense layer of 64 ReLU units and lastly, SoftMax layer with 5 units as a classification layer. The Adam optimizer was used to compile this model and train it on categorical cross-entropy loss with 10 epochs. The effect of generalization was monitored with a 20 percent validation split while training.

Results

The test accuracy of the model was outstanding 97.1%. Extensive classification report revealed that F1-scores, precision and recall scores were high across all five categories meaning that the performance was balanced. The confusion matrix showed a little misclassification in classification. To illustrate an example, considering the above sentence as an input sentence "FIFA Club World Cup final decided after PSG demolishes Real Madrid 4-0," the model properly classified the category "Sport" with a high level of confidence. The accuracy of training and validation accuracy steadily rose over epochs with no incidence of significant overfitting which implies that the model was decently regularized. These findings prove that the CNN model was capable of successfully learning the significant patterns on the news text and generalizing to unseen examples quite successfully.

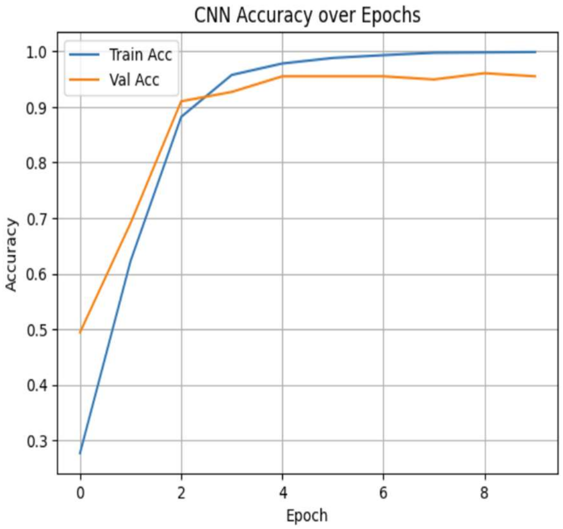

Figure 1: Training and validation accuracy over epochs.

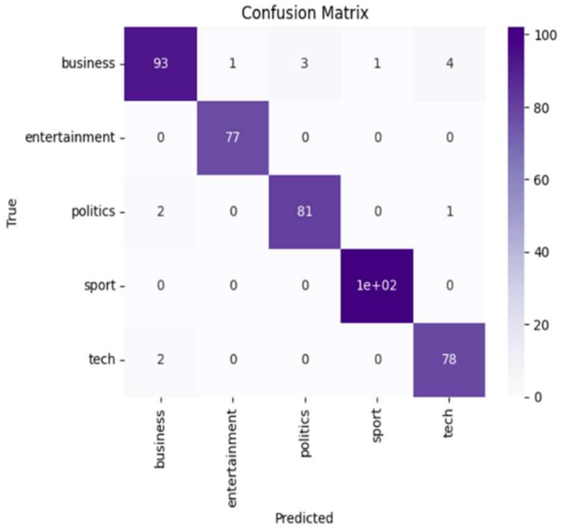

Figure 2: Confusion matrix showing correct and incorrect predictions by class.

Reflection

Earlier in the project, I have applied a similar task using Long Short-Term Memory (LSTM) model earlier on. LSTMs have become a popular choice in NLP to operate on sequences since they can handle long-term reliance's in text. Nevertheless, the LSTM model did not predict as successfully as it should have been in such a scenario. It demonstrated more sluggish training, decreased precision, and underfitting indicators even when the architectural parameters were optimized and the number of epochs upped. It was a good experience that undersored the fact that selection of a model relied on type of data. The BBC News dataset is composed of relatively short texts and clearly separated categories, so a CNN model proved to be more appropriate.

The move to a CNN was effective in enhancing performances and work efficiency. I also found out that CNNs which were developed to work on image data can be highly efficacious in discovering the local patterns in texts, particularly when coupled with an embedding layer. The most significant problem that I encountered was to adjust the network so that it does not under- and overfit and accepts a decent training time. To continue this project, I would test some pre-trained embeddings like GloVe or FastText and test Transformer-based models like BERT to see how they perform differently as well. In general, this project enabled me to master the concept of deep learning architectures and the process of applying this type of architectures to real NLP scenarios.

Conclusion

To sum up, this was a practice project that allowed comparing several different deep learning architectures on a practical NLP task. Although the first FLSTM model did not work well, it eventually led me to a more appropriate CNN model whose results were much better, and training speed was fast. In this process, I got to understand how to perform model selection based on the nature of data and performance results.

References

TensorFlow/Keras Documentation: https://www.tensorflow.org/api_docs

BBC News Dataset on Kaggle: https://www.kaggle.com/datasets/himelghosh/bbc-news-dataset-2226