Geliştirilen Bir Derin Öğrenme Modeli ile Metin Sınıflandırma Uygulaması

Text Classification Application with a Developed Deep Learning Model

Beste KÜÇÜK*[1],Aslı CÖNK*[2] (*: sorumlu yazar)

202802016@ogr.cbu.edu.tr, 202803063@ogr.cbu.edu.tr

Department of Software Engineering, Hasan Ferdi Turgutlu Technology Faculty, Manisa Celal Bayar* University, *Manisa, Turkey

Abstract

In recent years, the rapid growth of data production has been fueled by technological advancements and widespread internet usage. Text data from applications such as social media, communication tools, and customer services contribute significantly to this vast dataset. The effective processing of such large volumes of data requires automation and innovative solutions.

Recently, deep learning-based applications, particularly in the field of text processing, have achieved notable successes. Deep learning models like Feed-Forward Neural Networks (FFNN) stand out as powerful tools for efficiently classifying text data. The layered structure of FFNN provides an effective approach to extract features from text data and successfully classify them.

Text classification holds great importance, especially in Natural Language Processing (NLP). Deep learning models like FFNN can be applied in various domains, including sentiment analysis, news article classification, and automated analysis of customer reviews.

This study aims to develop a text classification model using deep learning techniques on a large text dataset. The research, conducted using a total of 871,909 article data, specifically focuses on categorizing texts into 26 different categories, with an emphasis on feed-forward networks. The developed model aims to be delivered to users through a mobile application, achieving a success rate of 61%. The findings of this study serve as a significant example of text classification applications on extensive datasets.

Keywords-*NLP, Deep Learning Model,FNN*

Model Architecture and Processing Steps:

1-Text Vectorization Layer:

- Custom Standardization:
  - Removal of stop words.
  - Cleaning of punctuation marks.
  - Conversion of Turkish characters to English characters.

Tokenization:

- Breaking down the text into parts or "tokens".

Indexing:

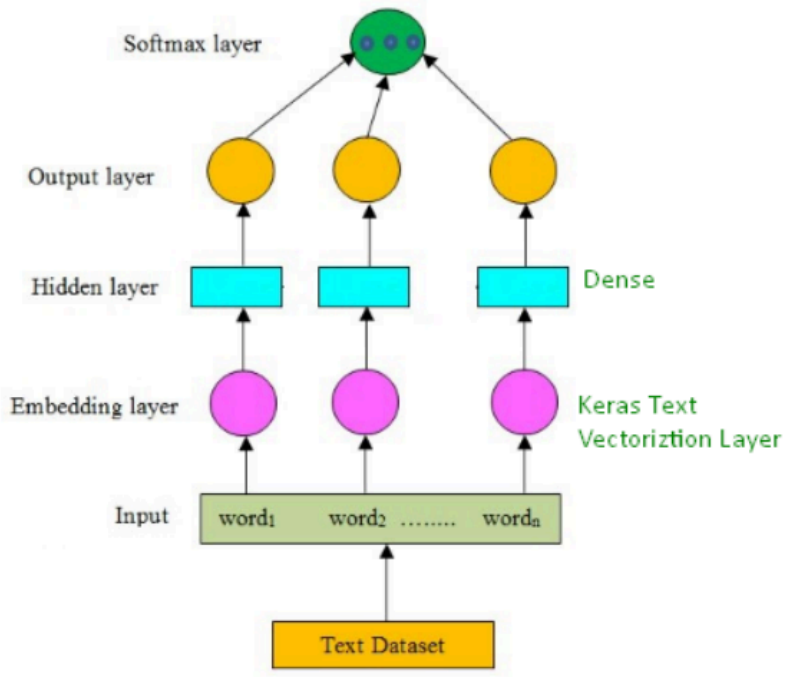
- Converting each token into an integer.

Padding:

- Filling shorter texts with a value of 0 to ensure texts of different lengths have the same length.

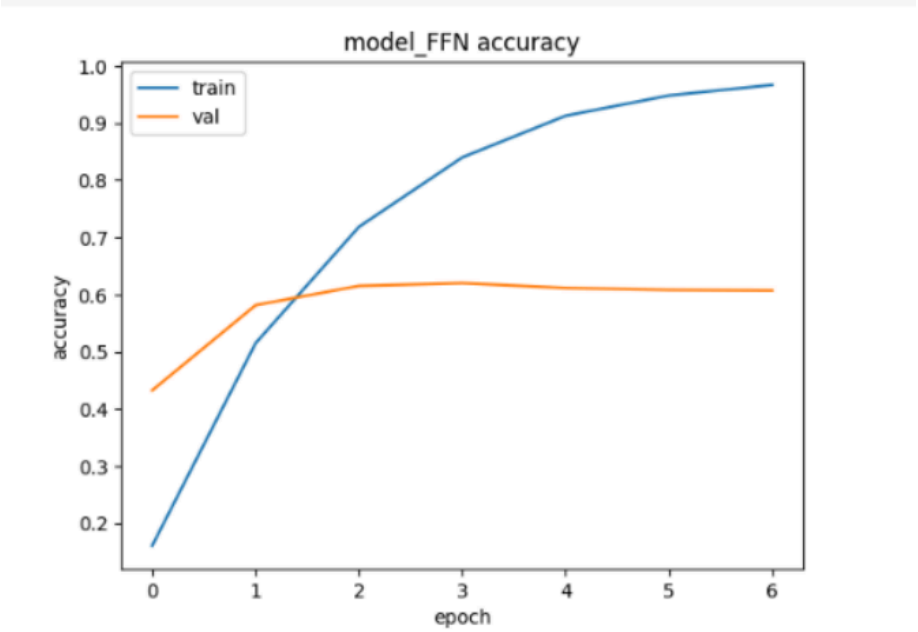
2-Hidden Layer:

- Embedding:
  - Converting each token into 16-dimensional vectors.
- Dense Layer:
  - A fully connected layer. Each input node is connected to every output node, and a weighted sum is passed to an activation function.
- Flatten:
  - Flattening multi-dimensional inputs into a single-dimensional vector.
- DropOut:
  - Used to control overfitting. Randomly drops a portion of the nodes during training with a certain probability.



Model Performance and Hyperparameters:

- Hyperparameters:
  - VOCAB_SIZE: 100,000
  - Number of Categories: 26
  - Total Data Count: 871,909
  - Number of Data Columns: 3
- Performance:
  - Loss: 2.0389
  - Training Accuracy: 0.6046
  - Test Accuracy: 0.6045602560043335



Dataset Details:

Data Split:

- Training Dataset: 80% (653,996 entries)
- Test Dataset: 20% (163,499 entries)

Data Reduction:

- Reduced Training Dataset: 32,699 entries (reduction ratio: 5%)
- Test Dataset: 163,499 entries

Validation Dataset:

- 10% of the Training Dataset is set aside for validation.
- Training Dataset: 29,429 entries
- Validation Dataset: 3,270 entries
- Test Dataset: 163,499 entries

	text	target	Word Count
0	python courses python courses, python exercis...	academic interests	125
1	the learning point open digital education. a r...	academic interests	147
2	equl offers enzyme assay kits, reagent mixtur...	academic interests	353
3	tech news, latest technology, mobiles, laptops...	academic interests	143
4	the best it certification materials in usa | k...	academic interests	364

Conclusion:

In the project, texts in the dataset were successfully classified into 26 different categories. The model employs a feed forward network structure and includes customized text vectorization, tokenization, indexing, and padding processes. The overall accuracy of the model is recorded at 60.46%, indicating a reasonable level of success. However, further improvements can be made by reducing the loss value and increasing the accuracy.