

**Department of Computer Science and Engineering**

**Coding Assignment for Deep Learning CSE754**

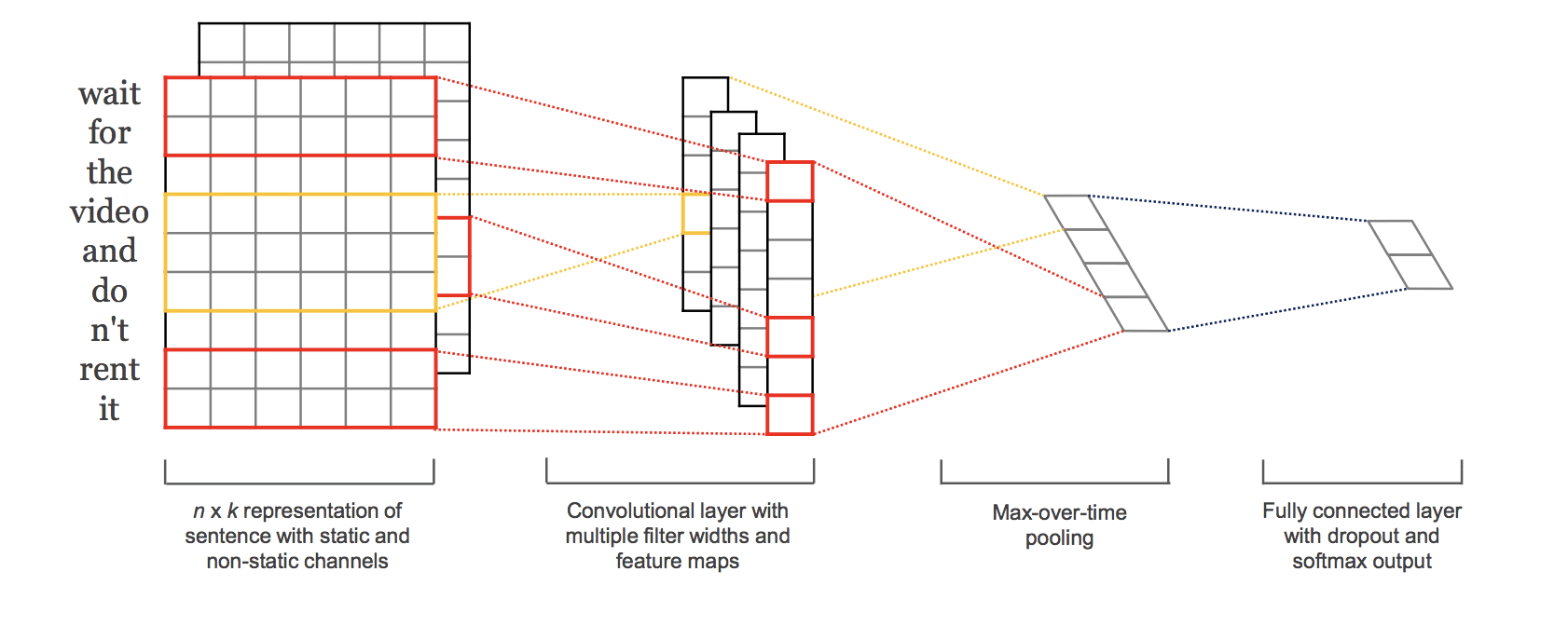
| **Title:** | |
| --- | --- |
| **Student USN** | **Student Name** |
| **1MS18CS025** | **Aravind Shreyas Ramesh** |
| **1MS18CS040** | **Dheeraj Bhat** |
| **1MS18CS043** | **Divya** |
| **1MS18CS046** | **Gaurav V** |

**Problem Statement**

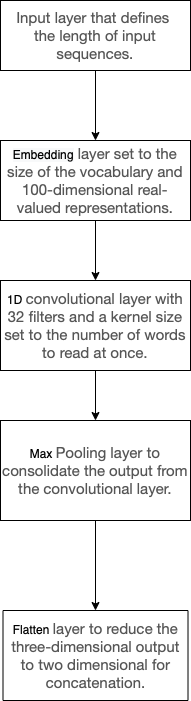
Text classification, an NLP task, is the process of categorizing text into organized groups. Text classification has several applications such as topic labeling, sentiment detection, spam filtering, and SEO optimization. There are several existing text classification models XLNet, ERNIE, Text-to-Text Transfer Transformer (T5), Binary Partitioning Transformer (BPT), Neural Attentive Bag-of-Entities (NABoE), Rethinking Complex Neural Network Architectures. This assignment focuses on using Multichannel Convolutional Neural Networks. Multichannel CNN uses multiple Convolutional Neural Networks (CNNs) of different kernel sizes that reads text of different n-gram sizes. The model extends the existing CNN by replicating multiple channels of CNN with different kernel sizes. As a part of the assignment, we have built a multichannel CNN model to identify the sentiment of movie reviews. The implementation makes use of Python 3, Tensorflow 2, and Keras 2.5.0.

**Structure Chart**

**Model architecture with two channels for an example sentence**

****

**Channel Structure**

****

**Implementation and Results**

**Input Data**: The Movie Review Data is a collection of movie reviews retrieved from the imdb.com website in the early 2000s by Bo Pang and Lillian Lee. The reviews were collected and made available as part of their research on natural language processing. The dataset is comprised of 1,000 positive and 1,000 negative movie reviews drawn from an archive of the rec.arts.movies.reviews newsgroup hosted at imdb.com. The dataset publishers refer to this dataset as the “polarity dataset.”

The data has been cleaned as a part of the data cleaning step

* The dataset consists of only English reviews.
* All text has been converted to lowercase.
* There is white space around punctuation like periods, commas, and brackets.
* The text has been split into one sentence per line.

The data has been used for a few related natural language processing tasks. For classification, the performance of machine learning models (such as Support Vector Machines) on the data is in the range of high 70% to low 80% (e.g. 78%-82%). More sophisticated data preparation may see results as high as 86% with 10-fold cross-validation. This gives us a ballpark of low-to-mid 80s if we were looking to use this dataset in experiments of modern methods.

**Output:** A multichannel CNN model is built to predict the sentiment of an unseen movie review as either positive or negative when evaluated with the built model.

**Algorithm:**

Step 1: Load the data into memory and run preprocessing steps on it.

* Remove punctuation.
* Remove non-alphabet characters (numbers).
* Remove stop words for the English Language.

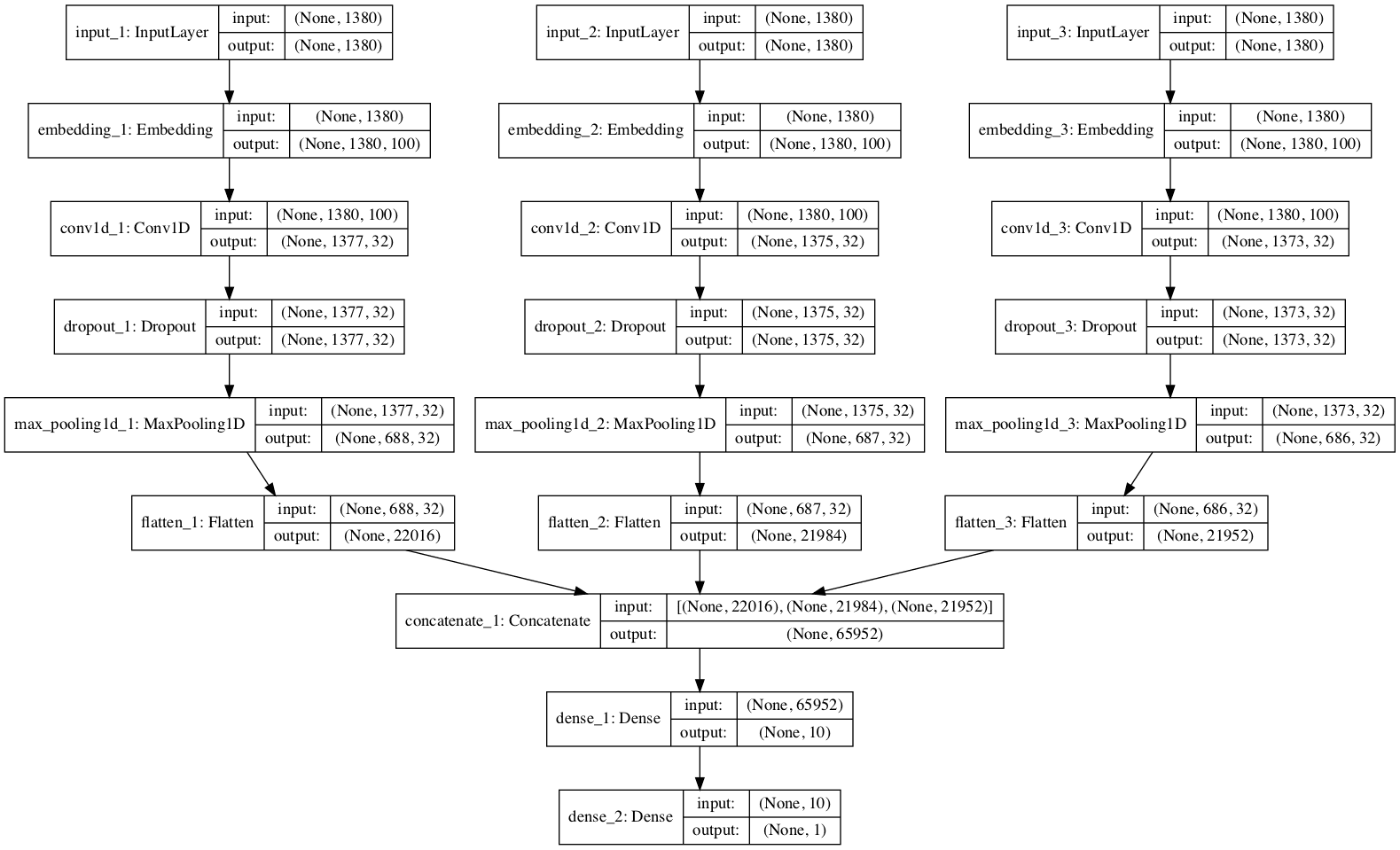
Step 2: Split the raw data into negative and positive samples for sentiment detection and split each of the classes into test and training sets (90 and 10% in our case). These sets are shaped into input format using the Keras library.

* Tokenization and fitting on input data.
* Creating a vocabulary.
* Encoding texts to numbers.

Step 3: Here, a multichannel CNN model is built with the following architecture.

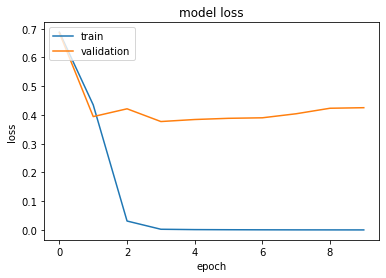
* Input layer that defines the length of input sequences.
* Embedding layer set to the size of the vocabulary and 100-dimensional real-valued representations.
* Multiple One-dimensional convolutional layers with 32 filters and a kernel size set to the number of words to read at once.
* Dropout layers for regularization.
* Max Pooling layer to consolidate the output from the convolutional layer.
* Flatten layer to reduce the three-dimensional output to two-dimensional for concatenation.
* The output from the three channels is concatenated into a single vector and process by a Dense layer and an output layer.

**Model Architecture**



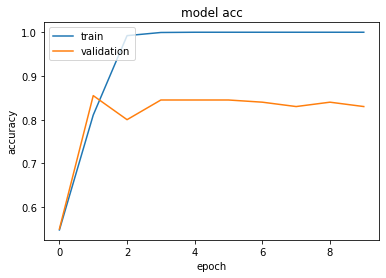
**Results:**

*Loss comparison:*



As evident, the model seems to overfit as validation loss after 10 epochs remain significantly higher than the training loss even after regularization techniques for CNN such as pooling and dropout. An ideal solution to this problem would be to include more data as a part of training.

*Accuracy comparison*

****

The model reports an accuracy of 83-84% which is considered satisfactory (Section 1) and the original publishers were able to obtain an accuracy of 81% which did not include the dropout layer. The difference between the accuracy of training and validation is further proof that the model overfits after a few number of epochs.

**Epilogue**

During the course of the assignment, a certain roadblock was faced ie., the overfitting nature of the model. Despite several efforts to reduce it to a minimum such as changing the configuration of the dropout layer, using different kinds of pooling layers, and using multiple initial weight assignments, the overfitting problem still exists.

The assignment provided the team with detailed insights on methods in text classification, its applications, and its models. It also helped us understand the difference between the traditional LSTM models against the proposed multichannel CNN network. As a result of the assignment, we understood the importance of text classification and deep learning.