20XD68 - Deep Learning Lab

SENTIMENT ANALYSIS

BI-LSTM & CNN

Start Slide

Contents

Introduction

Problem Statement, Goals

BiLSTM & CNN Implementation

Research Paper Approach

Federated Learning

Our Improvisation of the problem statement using Federated Learning

References

Sentiment Analysis

Text sentiment analysis is a significant technology in the field of natural language processing (NLP), which aim to identify and extract the subjective information of the reviewers from the text data

Classical Strategy

In order to overcome the deficiency of sentiment analysis based on traditional machine learning, which difficulty of effective feature selection and inadequacy of marked training corpus will affect the performance of the classification system.

Recent Advancements

we address the sentiment emotions analysis problem of Chinese product reviews text by combining convolutional neural network (CNN) with bidirectional long-short term memory network (BiLSTM)

BiLSTM & CNN Model

Architecture

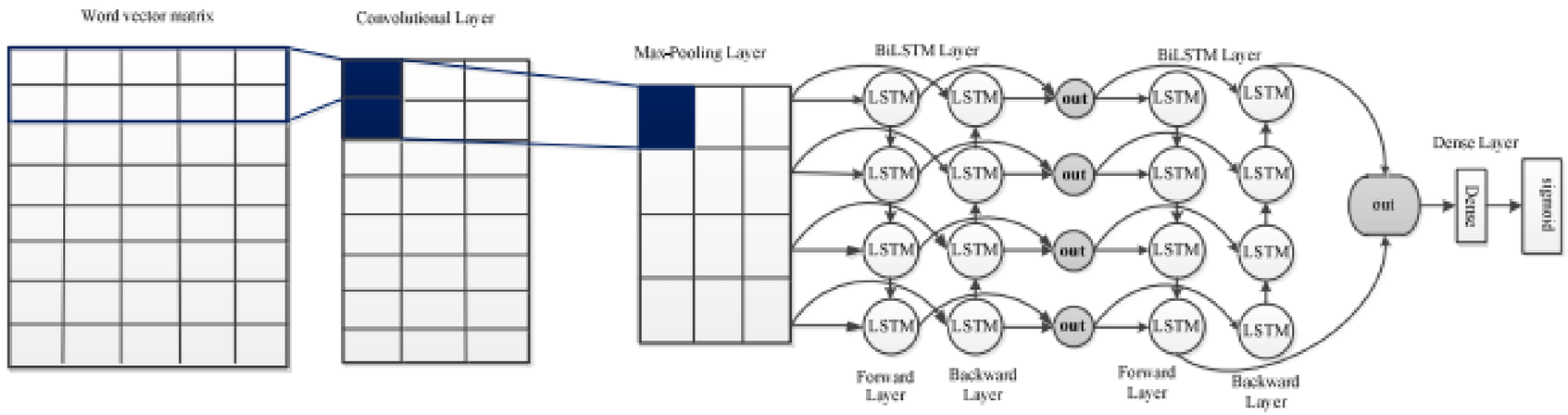


Fig. 1. THE ARCHITECTURE OF CNN - BIDIRECTIONAL LSTM MODEL.

Paper: Sentiment Analysis of Text Based on CNN and Bi-directional LSTM Model

BiLSTM & CNN Model

Implementation

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 32, 128)           128000

conv1d_1 (Conv1D)            (None, 32, 32)            12320

max_pooling1d_1 (MaxPooling  (None, 16, 32)            0
1D)

bidirectional_1 (Bidirectio  (None, 64)                16640
nal)

dropout_1 (Dropout)          (None, 64)                0

dense_1 (Dense)              (None, 3)                 195

=================================================================
Total params: 157,155
Trainable params: 157,155
Non-trainable params: 0
_________________________________________________________________
```

BiLSTM & CNN Model

Test Results

Dataset: Twitter Sentiment Analysis Data with 90000 rows .3 labels each of 30000 rows

Train Confusion Matrix

```
Epoch 1/15
57/57 [==============================] - 8s 44ms/step - loss: 1.0715 - accuracy: 0.4251
Epoch 2/15
57/57 [==============================] - 4s 70ms/step - loss: 0.8278 - accuracy: 0.6268
Epoch 3/15
57/57 [==============================] - 3s 44ms/step - loss: 0.7059 - accuracy: 0.6939
Epoch 4/15
57/57 [==============================] - 3s 47ms/step - loss: 0.6470 - accuracy: 0.7251
Epoch 5/15
57/57 [==============================] - 4s 66ms/step - loss: 0.5947 - accuracy: 0.7563
Epoch 6/15
57/57 [==============================] - 4s 65ms/step - loss: 0.5176 - accuracy: 0.7969
Epoch 7/15
57/57 [==============================] - 3s 48ms/step - loss: 0.4171 - accuracy: 0.8469
Epoch 8/15
57/57 [==============================] - 3s 44ms/step - loss: 0.3304 - accuracy: 0.8926
Epoch 9/15
57/57 [==============================] - 3s 44ms/step - loss: 0.2652 - accuracy: 0.9186
Epoch 10/15
57/57 [==============================] - 3s 44ms/step - loss: 0.2256 - accuracy: 0.9300
Epoch 11/15
57/57 [==============================] - 4s 69ms/step - loss: 0.1924 - accuracy: 0.9440
Epoch 12/15
57/57 [==============================] - 3s 45ms/step - loss: 0.1573 - accuracy: 0.9560
Epoch 13/15
57/57 [==============================] - 3s 45ms/step - loss: 0.1474 - accuracy: 0.9571
Epoch 14/15
57/57 [==============================] - 3s 44ms/step - loss: 0.1314 - accuracy: 0.9597
Epoch 15/15
57/57 [==============================] - 3s 54ms/step - loss: 0.1321 - accuracy: 0.9604
```

Test Confusion Matrix

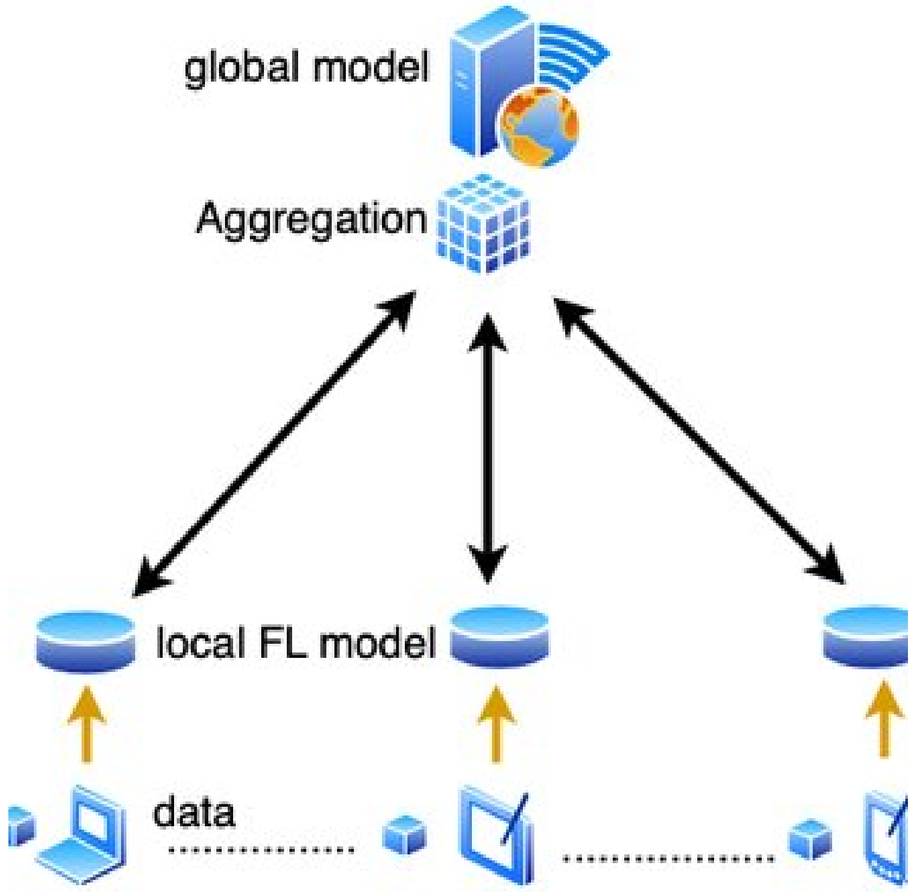
```
confusion matrix [[330 250  24]
 [114 418  81]
 [ 33 195 355]]
              precision    recall  f1-score   support

           0       0.69      0.55      0.61       604
           1       0.48      0.68      0.57       613
           2       0.77      0.61      0.68       583

    accuracy                           0.61      1800
   macro avg       0.65      0.61      0.62      1800
weighted avg       0.65      0.61      0.62      1800
```

Federated Learning

Introduction

1. Federated learning is a decentralized approach to machine learning.
2. It enables multiple devices to collaboratively train a model without sharing their raw data with a central server.
3. Each device trains a local model using its own data.
4. The local models are then sent to a central server, which aggregates them to create a global model.
5. This approach allows devices to learn from each other without compromising their privacy.
6. Federated learning consists of three main components: the client devices, the central server, and the machine learning algorithm.
7. The client devices are responsible for training the local models.
8. The central server coordinates the training process and aggregates the local model updates.
9. The machine learning algorithm is used to update the global model based on the aggregated local model updates.
10. Federated learning can reduce the amount of data that needs to be transferred to a central server, which can be expensive and slow.
11. It can also enable machine learning on devices with limited resources, such as smartphones or IoT devices.
12. Federated learning has applications in various fields, including healthcare, finance, and IoT.

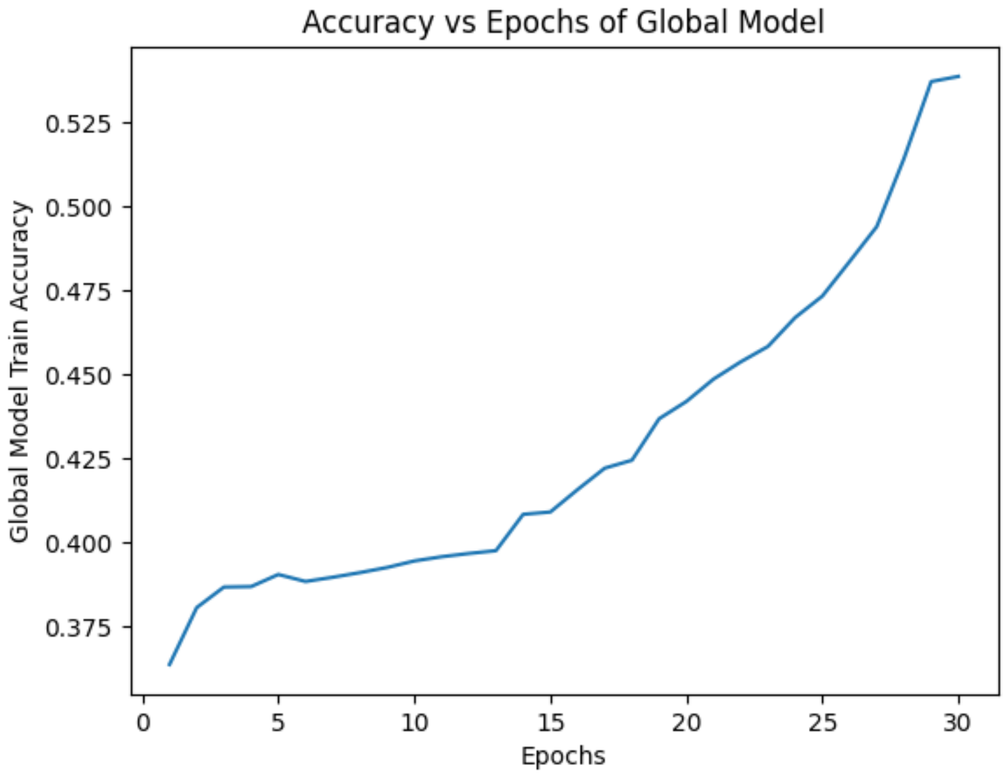
FedAvg Architecture



Federated Learning

Results

No of Clients = 5
Global Epochs = 30
Local Epochs = 10



Future Improvements

References

1. Federated Learning: A Simple Introduction
2. Complete guide to Bidirectional LSTM
3. Sentiment Analysis using Neural Networks and LSTM