Zhenyu Nie

Project description:

This project is to generate and compare two deep learning models to predict the if the email is spam or not. The two models that I chose are CNN and RNN.

Task & dataset & preprocessing:

Task: Binary classification, predict if the email is spam or not.

Dataset: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset/

Preprocessing:

1. clean the text:

   a. remove all special characters

   b. remove all numbers

   c. convert all text to lowercase

2. Tokenize the text:

   split the text into single words or tokens

3. convert to sequences:

   convert the tokens into integers index, so that each word is represented by an integer

4. Padding:

   Pad the text to the same length. The length is the length of the longest text in the dataset.

5. Embedding:

Inside the model, through the embedding layer, the integer index is converted into a dense vector.

The training details of two deep learning systems:

RNN:

Model Structure:

embedding layer -> LSTM layer -> Dense layer

embedding layer: vocab_size = 10000, every word is represented by a 128 dimension vector

LSTM layer: 64 units, dropout rate = 0.2 used to avoid overfitting

Output layer: 1 unit, sigmoid activation function

Compile: optimizer = adam, loss = binary_crossentropy, metrics = accuracy

Training process: epochs = 10, test_size = 0.2

CNN:

Model Structure:

Embedding layer -> Conv1D layer -> GlobalMaxPooling1D layer -> Dense layer

Embedding layer: vocab_size = 10000, every word is represented by a 128 dimension vector.

Conv1D layer: 64 filters, kernel_size = 5, activation = relu

MaxPooling1D layer: used to reduce the dimension of the output of the Conv1D layer

Output layer: sigmoid activation function, used to binary classification

compile: optimizer = adam, loss = binary_crossentropy, metrics = accuracy

tranning process: epochs = 10, test_size = 0.2

The results & observations & conclusions:

Results:

RNN: after 10 epochs, the accuracy on is round 87%

CNN: after 10 epochs, the accuracy on is round 98%

Observations:

RNN: the accuracy is not high enough; it is because the RNN model is not good at dealing with long text and probably the model is not complex enough.

CNN: the accuracy is high; it is because the CNN model is good at dealing with text data and CNN is good at extracting features from the data (EX: the keyword in the email).

Training time: CNN is faster than RNN, because CNN is good at dealing with text data.

Conclusions:

For this spam email classification task, CNN is better than RNN. CNN has a better accuracy; training time and it is easier to implement. Although the LSTM of RNN is good at dealing with sequential data, but it performs not well on this task. Therefore, from this project, we can observe when choosing the deep learning model, we should consider the task and the dataset.

The challenges & obstacles:

1. when create the environment for the project, kernel crushed cause when using TensorFlow, since I'm using the conda environment. I need to avoid using pip to install packages, instead use conda install.

2. The initial preprocessing of the dataset is not easy, since the dataset is not clean, there are some special characters and numbers in the text. I need to clean the text before tokenization. Before cleaning the text, the accuracy is low.

3. When I was implementing the RNN model, I found the model is not complex enough, so I added another LSTM layer

4. when I try to find out the better model to obtain a better accuracy, I try to adjust the parameters of the model. This causes the model to overfit. I need to adjust the parameters again to avoid overfitting. This process is a time consuming.