Intro to DL Y2025 Semester B

Project part 2

Submission

Submission is in pairs or singles.

Data format and description

The data appear in three files on moodle and kaggle:

- task2_train
- task2_val
- task2_test

ID denotes the ID of the text, text column contains emails, and label indicates if the text (e-mail) is a spam or not (1 or 0).

The task

Your goal is to **predict the label** for test data values, based on the text (do not use the ID column! If you do, your score will be 0, and you will be on the bottom of the kaggle table).

This is an anomaly detection task because majority is large, and the training file contains no spam examples.

1. Preprocess texts any way you wish and generate vector representations for them.
    a. you can use VSM representations as in task 1, or
    b. use Gensim package to generate word vectors (Word2Vec, FastText, etc.) in which case you need to make sure that all of your texts are padded to the same length (pad with zeroes), and your text representations will be concatenations of your word vectors.
2. Build an autoencoder for your vectors with Keras as follows:
    a. The first layer should be of the size of your vectors (or larger if you concatenated data).
    b. Inner layer activation should be Relu, but you can try other options.
    c. Output layer is recommended to be of the same size as the input layer with the **sigmoid** activation, your loss is recommended to be **mse**, and your metric should be **accuracy**.
    d. The number of inner layers and their size is up to you.
3. Train the autoencoder on benign data (only texts labeled 0, which is all you have in the training file).
4. After training the autoencoder, calculate the **reconstruction error** on the test data:

example: X_test_reconstructed = autoencoder.predict(X_test)
mse_error = np.mean(np.power(X_test - X_test_reconstructed, 2), axis=1)

5. Set a threshold for anomaly (up to you, but it is recommended to start with the majority value you can derive from the validation data):
   threshold = np.percentile(mse[:len(normal_data)], X) # eX-th percentile of the normal data error

   and classify data points: 1 for anomaly (mse>threshold), 0 for normal (mse<=threshold)
6. Submit your predictions for the test data on Kaggle.

Note: you can use any of the normalization and data analysis methods in sklearn to improve your scores.

Result submission

Your result should include image IDs from the test set and predicted label, and to be saved as csv file:

id	label
1	0
2	1
…	…

How kaggle works (a reminder)

Your results will be compared with the actual test dataset labels, and the resulting accuracy will be reported on the scoreboard of the competition. Note that public scoreboard will show F1 measure on 50% of the test set, and private (i.e., my) scoreboard will show F1 measure on the whole test set. The final scoreboard will be published after submission & code checking is over, and your grade will be determined by your place in the competition.

Code submission

Submit your code on moodle, as a single <id1>_<id2>.py file (do not submit python notebooks!).

Note of warning: all code will be automatically checked for copying. If cheating is discovered, you will get grade 0 automatically and go on to face the scholarly committee.