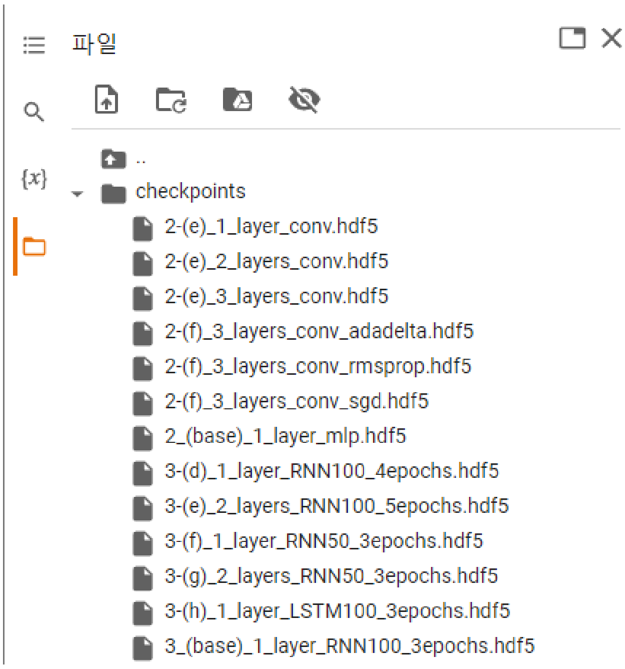
BE201 Artificial Intelligence Basics, Homework

Deadline : 1 Dec 2023

Instructions

- Please submit a zip file named ‘**ID_name.zip**’ to LMS that includes your **source codes** in py or ipynb extensions, **trained model** (checkpoint) together with **a report** in PDF.
- Individual source code (Problem 2-(a∼d, g) (i.e., modified hw1.py), 3-(e), 3-(f), 4-(d), 4-(e), 4-(f), 4-(g), 4-(h)), trained models of deep neural networks, and a report containing the results and answers to all questions should be submitted.
- For the deep neural network parts, the questions that require the saved model are marked in **Red**. you should submit 13 checkpoints as hdf5 format as follows:



..

checkpoints

- 2-(e)_1_layer_conv.hdf5
- 2-(e)_2_layers_conv.hdf5
- 2-(e)_3_layers_conv.hdf5
- 2-(f)_3_layers_conv_adadelta.hdf5
- 2-(f)_3_layers_conv_rmsprop.hdf5
- 2-(f)_3_layers_conv_sgd.hdf5
- 2_(base)_1_layer_mlp.hdf5
- 3-(d)_1_layer_RNN100_4epochs.hdf5
- 3-(e)_2_layers_RNN100_5epochs.hdf5
- 3-(f)_1_layer_RNN50_3epochs.hdf5
- 3-(g)_2_layers_RNN50_3epochs.hdf5
- 3-(h)_1_layer_LSTM100_3epochs.hdf5
- 3_(base)_1_layer_RNN100_3epochs.hdf5

HOW TO SAVE YOUR MODEL:

https://www.tensorflow.org/tutorials/keras/save_and_load

```
# save your model
model.save(’./checkpoints/two_conv_layer.hdf5’)
# load your model (recommend to test the saved model)
new_model = tf.keras.models.load_model(’./checkpoints/two_conv_layer.hdf5’)
```

1 Google Colaboratory

No report is required for this part.

(a) Please execute a following link
https://colab.research.google.com/

(b) Please copy the above colab page into your google driver account.

(c) Try to run a following link within your google driver account.
https://colab.research.google.com/notebooks/gpu.ipynb

(d) You could try Pytorch in Colab as well.
https://colab.research.google.com/github/omarsar/pytorch_notebooks/blob/master/pytorch_quick_start.ipynb

2 Spam Mail Classification by Naive Bayes

(60 points) Please download the provided dataset file (spam.csv). Then, take a look at and execute the following skeleton code and test it on Google Colab to complete the implementation. A report is required for the questions in **Evaluation** below.

https://colab.research.google.com/drive/1U4VupdCoD6bbkr73DdPFnfJeyuvSmylW?usp=sharing

Model Training: You should first complete the implementation in Part 2 of the skeleton code so that the class (NBModel) computes the required probability values for the training dataset.

(a) (5pts) Implement the function word_exists() so that it returns True or False depending on the existence of the word in the training dataset.

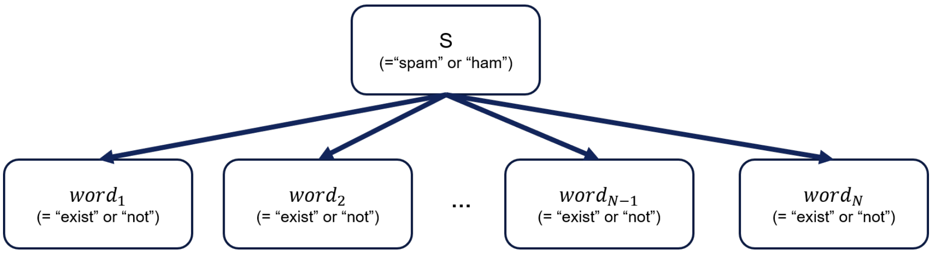
(b) (5pts) Implement the function spam_prob() so that it returns the probability that an email was a spam in the training dataset.

(c) (10pts) Implement the two functions spam_cond_prob() and ham_cond_prob(). (The first function should return the conditional probability (0.0 $\sim$ 1.0) that a *spam* email contains the given word. Similarly, the second function should return the conditional probability that a *ham* email contains the given word.)

Model Inference: In Part 3 of the skeleton code, you should complete the inference procedure (i.e., inference() function). A skeleton code is already given, so please take a look at the portion marked with comments to see where to edit.

(d) (10pts) Complete the code so that it computes the probability of either spam or ham, i.e., spam_prob and ham_prob variables using the Naive Bayes algorithm we learned.

(Hint: Please refer to the following Bayes Net that represents our Naive Bayes model. Also, if some words are presented during the inference but did not exist in the training dataset, you can simply ignore them.)



Evaluation: Include the discussions in the report for the following questions.

(e) (10pts) Explain your implementation briefly and report the classification results.

(f) (10pts) Compare your results with three other Naive Bayes classification algorithms implemented in the scikit-learn library: MultinomialNB, BernoulliNB, ComplementNB. (You can test them by modifying the baseline code already implemented in Part 4.)

(g) (10pts) Your first implementation would have a lower accuracy than the scikit-learn algorithms. Devise any strategy to improve the accuracy and discuss it in the report. Include your implementation in the submitted code by adding a new inference function and/or model class.

3 Classification by deep neural networks

Please conduct your simulation on Google Colab. A report is required for the following questions.

(40 points) [**MLP-Fashion-MNIST**] Please execute a following link
`https://www.tensorflow.org/tutorials/keras/classification`

(a) (5pts) After training is done, plot output values of the network for an arbitrary image from test dataset.

(b) (5pts) Explain the meaning of the output values.

(c) (5pts) Find a wrong prediction from test set and report values. Again, explain the meaning of output values.

(d) (5pts) Please explain Flatten, Dense layer, Adam optimizer, and SparseCategoricalCrossentropy.

(e) (10pts) Please replace MLP layers to convolutional layers of 3 × 3 support with the same channels, then retrain your network. How does the test accuracy change when the number of layers is equal to 1,2, and 3? Please analyze the result. Please save your model and submit it. (Hint: `https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv2D`)

(f) (10pts) Please retrain your 3 convolution layers model with different optimizers: adadelta, sgd, and rmsprop. Please analyze the results. (Hint: `https://www.tensorflow.org/api_docs/python/tf/keras/optimizers`)

4 Sentiment analysis by recurrent neural networks (RNN)

Please conduct your simulation on Google Colab. A report is required for the following questions.

(60 points) [**RNN-IMDB**] Please execute a following link
`https://drive.google.com/file/d/1jg6zKICQW1jYmGGUbPXP3yHtv0JGiPsz/view`

(a) (6pts) Explain the meaning of the output values.

(b) (7pts) Please explain **the meaning of max_words, embedding_vector_length, and input_length** in 'Embedding(max_words, embedding_vector_length, input_length=max_review_length)'.

(c) (7pts) Please explain **the meaning of in units, activation, and return_sequences** in SimpleRNN(units=100, activation='tanh', return_sequences=True).

(d) (8pts) Please change **epoch of 3 to the epoch of 4**, then retrain the model with **one RNN(100) layer**. How does the test accuracy change? and Why does the test accuracy increase (or decrease)? Please analyze the result. Please save your model and submit it.

(e) (8pts) Please change **epoch of 3 to the epoch of 5**, then retrain the model with **two RNN(100) layers**. How does the test accuracy change? and Why does the test accuracy increase (or decrease)? Please analyze the result. Please save your model and submit it.

(f) (8pts) Please replace the model with **one RNN(100) layer** to that with **one RNN(50) layer**, then retrain your network with the epoch of 3. How does the test accuracy change? and Why does the test accuracy increase (or decrease)? Please analyze the result. Please save your model and submit it.

(g) (8pts) Please replace the model with **one RNN(50) layer** to that with **two RNN(50) layers**, then retrain your network. How does the test accuracy change? and Why does the test accuracy increase (or decrease)? Please analyze the result. Please save your model and submit it.

(h) (8pts) Please replace the model with **one RNN(100) layer** to that with **one LSTM(100) layer** with the epoch of 3, then retrain your network. How does the test accuracy change? and Why does the test accuracy increase (or decrease)? Please analyze the result. Please save your model and submit it.