CIS 520, Machine Learning, Fall 2021
Homework 11
Due: Friday, December 10th, 11:59pm
Submit to Gradescope

November 30, 2021

Instructions. Please write up your responses to the following problems clearly and concisely. We require you to write up your responses using LaTeX; we have provided a LaTeX template, available on Canvas, to make this easier. **Submit your answers in PDF form to Gradescope. We will not accept paper copies of the homework.**

Collaboration. You are allowed and encouraged to work together. You may discuss the **written homework** to understand the problem and reach a solution in groups. However, **it is recommended that each student also write down the solution independently and without referring to written notes from the joint session.** You must understand the solution well enough to reconstruct it by yourself. (This is for your own benefit: you have to take the exams alone.)

1 RNN and GPT-2 [50 points]

In this question, we will implement Character-level Recurrent Neural Networks for text classification and Huggingface's GPT-2 for text generation. Please refer to ipynb template for more details.

1.1 Char-RNN Text Classification

The colab notebook skeleton and the dataset are provided under Canvas files. Please read through the notebook and make sense of the dataset before proceeding. As we will not be grading your notebook, make sure you record your code implementation for relative functions here. (Verbatim is probably a good tool to consider.) The specifications are listed on the worksheet. You can find the related PyTorch tutorial [1].

1.1.1 Grade Breakdown

[**30 points**] - Model Construction (CharRNNClassify): 10';
- Helper Functions (trainOneEpoch, run): 5'·2=10';

[1] https://pytorch.org/tutorials/intermediate/char_rnn_classification_tutorial.html

- Accuracy Plots: 5';
- Test Set Accuracy: 5';

1.2 GPT-2 Text Generation

Generative Pre-trained Transformer 2 (GPT-2) [2] [3] is an open-source artificial intelligence created by OpenAI in February 2019, which aims to solve multitasks in natural language processing. In this question, we resort to huggingface's API [4] to implement text generation.

1.2.1 Model Setup and Text Generation

[10 points] First, we need to set up the GPT-2 model by transformers library following the instruction in in ipynb. Now, we have two input texts, they are

- 'input_text': "We love CIS 520 Machine Learning in University of Pennsylvania"
- 'input_text2': "We take CIS 520 Machine Learning in University of Pennsylvania"

Try to generate the text given 'input_text' and 'input_text2' respectively (note that it's possible that the generated texts contain some repeated snippets since we haven't fine-tuned the model). Also, list the tensor values of each tokenized word of 'input_text'.

1.2.2 The Impact of Letter Case

[6 points] Write down the tensor values and generated texts of 'Machine Learning' and 'Machine learning'. Explain: Are they different? Are the token and model case-sensitive? Do you think the output make sense if they're different and why? The explanation is open-ended.

1.2.3 Bias

[4 points] Biased natural language processing models can lead to negative effect on society by discriminating against certain groups of people and contributes to the biased associations of individuals through the media platforms. For instance, "He is doctor" has a higher likelihood in text generation than "She is doctor" [5]. Run the cells in ipynb, provide two to three outputs as evidences and explain the bias of gender, race, etc. that you can notice. This question is open-ended.

2 AutoML [50 points]

In this part, we will do some hands-on practices for GridSearchCV and AutoML.

[2] GPT-2 paper: Language Models are Unsupervised Multitask Learners
[3] GPT-3 paper: Language Models are Few-Shot Learners
[4] https://huggingface.co/transformers/model_doc/gpt2.html
[5] Paper: StereoSet: Measuring stereotypical bias in pretrained language models

2.1 Model Selection with Sklearn

[25 points] GridSearchCV provides a exhaustive search over specified parameter values for an estimator.

The parameters of the estimator used to apply these methods are optimized by cross-validated grid-search over a parameter grid.

We have provided some helper functions and example code. You need to modify the parameter list by adding AdaBoost, support vector machine, and random forest classifier with appropriate set of hyperparameyers and ranges to search. Compare those models' performance and report the results in the write-up.

2.2 AutoML

[25 points]

Auto-sklearn frees a machine learning user from algorithm selection and hyperparameter tuning. It leverages recent advantages in Bayesian optimization, meta-learning and ensemble construction. In this question, we will try autosklearn on the same dataset.

The optimization process will run for as long as you allow, measure in minutes. By default, it will run for one hour. For timely manner, we will restrict 'time_left_for_this_task' to be 300 seconds. You are encouraged to explore other configuration options that will yield good results. Train AutoML on the same dataset with 300 seconds and take a look on final ensemble constructed by auto-sklearn. Then report the 3 classifier choices with the largest ensemble_weight in this ensumble.