Project 1
Deep Learning in R

A. Nagar

CS 6301
Spring 2019

Instructions

- This assignment requires you to build a project using Deep Learning in R.
- You should store your dataset under your account in the UTD server and your code should run on Google Cloud. Do not submit the dataset (which could be quite large) on eLearning,
- You are allowed to work in teams of maximum four students. Please write the names and NetIDs of each group member on the cover page.
  Only 1 final submission per team.
- **You have a total of 4 free late days for the entire semester. You can use at most 2 days for any one assignment. After four days have been used up, there will be a penalty of 10% for each late day. The submission for this assignment will be closed 2 days after the due date.**
- Please ask all questions on Piazza, not via email.

1 Project Selection

For this project, you are to work on a project that is either about image classification / identification using CNN or text classification / identification using CNN

You can choose between any of the following projects:

Image Based Projects

Please select from one of the following. **Sorry, you cannot choose any other project.**

- CIFAR-10 - Object Recognition in Images
  `https://www.kaggle.com/c/cifar-10`
- Google AI Open Images - Object Detection Track
  `https://www.kaggle.com/c/google-ai-open-images-object-detection-track`
- Cdiscounts Image Classification Challenge
  `https://www.kaggle.com/c/cdiscount-image-classification-challenge`
- Diabetic Retinopathy Detection
  `https://www.kaggle.com/c/diabetic-retinopathy-detection#description`
- Challenges in Representation Learning: Facial Expression Recognition Challenge
  `https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data`
- Dog Breed Identification
  `https://www.kaggle.com/c/dog-breed-identification`
- Google Landmark Identification Challenge
  `https://www.kaggle.com/c/landmark-retrieval-challenge`
- Identify monkey species
  `https://www.kaggle.com/paultimothymooney/identify-monkey-species-from-image`

Text Based Projects

For text classification/identification, you can select from any of the following. **Sorry, you cannot choose any other project**

- BBC text classification challenge
  `https://www.kaggle.com/yufengdev/bbc-fulltext-and-category`
- Tradeshift Text Classification
  `https://www.kaggle.com/c/tradeshift-text-classification`
- Large Scale Hierarchical Text Classification
  `https://www.kaggle.com/c/lshtc`
- Text categorization
  `https://www.kaggle.com/c/text-categorization/data`

- Deep-NLP Project
  `https://www.kaggle.com/samdeeplearning/deepnlp`
- Text Classification using Neural Networks
  `https://www.kaggle.com/eliotbarr/text-classification-using-neural-networks`

2 Requirements

Most of the projects or competitions have their own requirements which you need to follow.

The following are parameters that **cannot be changed**

1. The programming language should be R.
2. Irrespective of what is mentioned in the project, you need to use any one of the deep learning techniques that we studied in class.
3. Your code should be able to run on Google Cloud using GPU.
4. Please do not hard code any paths to your local computer. Of course, you can refer to public paths under your UTD account.
5. Normally, each project has training and test data separated. If that is not the case, you have to divide the data into these two parts. It's up to you to choose the ratio.
6. If you are unsure, please ask the instructor.

You need to tune as many of the parameters as possible. The list will not be mentioned here, but you can see them in the documentation and sample code. You have to keep a log of your experiments with the parameters used and accuracy and loss obtained, similar to the assignment on CNN.

3 Submission

You need to output the at least the following along with your code:

1. History plots showing training and testing accuracy and loss as a function of number of iterations. This is automatically generated by the system.
2. Example of at least 25 test data points from test dataset, showing the following
   - Data
   - True Label
   - Predicted Label
3. A table containing details of parameter testing and tuning. Example:

Iteration	**Parameters**	**Training and Test Accuracy**
1.	Number of layers = ... Kernel Size Layer 1= .... Activation Function = ...	Train = 80% and Test = 78%
.....	......	....

If you have made any assumptions, please state them completely. Also include instructions on how to compile and run your code in a README file.