Project
Deep Learning using Keras

A. Nagar

Due Date Specified on eLearning

Instructions

- This assignment requires you to build a project using Deep Learning using Keras.
- You should store your dataset under your account in the UTD server or any other public location, such as Google drive. Do not submit the dataset (which could be quite large) on eLearning,
- You are allowed to work in teams of maximum two 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.

Image Based Project

For this project, you are to work on a project that is either about image classification / identification using Convolution Neural Networks (CNN)

You can choose any dataset of your choice. Below are some possible examples:
Important: You cannot use any datasets that are available as part of TensorFlow or Keras

Image Based Projects

- 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`
- CIFAR-10 - Object Recognition in Images
  `http://www.cs.utoronto.ca/~kriz/cifar.html`
- Any image based dataset from UCI ML repository
  `https://archive.ics.uci.edu/ml/index.php`
- Any image based datase extracted from Google research datasets
  `https://datasetsearch.research.google.com`
- Any image based dataset available from Amazon's dataset repository
  `https://registry.opendata.aws`
- Any image based dataset from Microsoft Research Open Datasets
  `https://msropendata.com`

Requirements

The following are important requirements for the project:

1. The programming environment should be Keras using Python .
2. You will have to use *transfer learning*, which involves using a pre-trained model, such as MobileNet V2 or any other such model from TF Hub: `https://tfhub.dev`.
3. You are free to fine tune the pre-built model to suit your dataset.
4. Irrespective of what is mentioned in the project, you need to use any one of the deep learning techniques that we studied in class.
5. Please do not hard code any paths to your local computer. Of course, you can refer to public paths under your UTD or other public accounts.
6. 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.
7. 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 available on TensorFlow. You have to keep a log of your experiments with the parameters used and accuracy and loss obtained.
8. If you are unsure, please ask the instructor through Piazza.

1 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. Most of the time, 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 = ... Filter 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.