ME 592X: Data Analytics and Machine Learning for Cyber-Physical Systems

Homework 4

Homework Assigned on March 21, 2018 Homework Due on: April 6, 2018
One submission per group

Motivation

This homework is to provide an experience of different deep learning architectures and how to train them with applications to different themes. As done in the previous assignment, the relevant data shall be provided to you in an email containing the download link. Most of the codes which are asked to be written may not be very trivial, but most of them are available online with sufficient documentation. Please search for relevant codes as required and cite them.

General Instructions

The dataset and problems for each group are slightly different, but the motivation remains the same. Following are some instructions for all the theme groups. Specific instructions for each group shall be provided in the relevant sections.

1. The final code must be pushed to git before the deadline and relevant data for running the codes should be placed at `/ptmp/ME592_2018_GRPNAME` in hpc-class cluster. The reason for this is to make the git repository compact and generic for any data.
2. Use the discussion board in Canvas in case of any issue.

Expected Outcome

1. A code pushed in git.
2. A presentation on the tasks given, challenges faced and inferences obtained. You must clearly describe the problem and the formulation of model to solve the problem (preferably using images/graphics to explain).

Self-Driving Cars1

In this homework, you will work on developing a deep learning model for running the inference on a Nvidia Drive PX2. To this end, you would first train a mask R-CNN on MS-COCO dataset. The following are the tasks you need to perform.

1. Download and install the Pycocotools API for processing the data.
2. Download COCO dataset and process the data.
3. Using Tensorflow/keras, create a model for ResNet50 architecture and train it for object detection.
4. Now, freeze the convolutional layers and build the mask-RCNN architecture and train it using the annotations provided and masks. (These might not be trivial, there are multiple codes online available for this task, you might want to make use of one of them and cite)
5. Once the mask R-CNN is trained, check for performance on different metrics as discussed in the paper (`https://arxiv.org/pdf/1703.06870.pdf`)
6. Convert the trained model with weights to a frozen model and save it as `frozen_model.pb` file.
7. Using the TensorRT python API, create a tensorRT model. (You might have to install TensorRT in an ubuntu 16.04 or use a docker image to do this).

Self-Driving Cars2

In this homework, you will work on deployment of creating a pipeline for inference of a R-CNN variant in Drive PX. You would have to install TensorRT and its python API. The following are the tasks to perform.

1. Use pretrained model (of your choice, please see, Keras applications model) from keras.
2. Train a Single Shot Multiple Bounding Box detector (SSD) for the same using Pascal VOC 2012 dataset.
3. Freeze the model and save it as `frozen_model.pb` and create tensorrt binary file from it.
4. Perform inference on a DrivePX available in Self-aware Complex Systems Lab using the camera installed, sample code of Driveworks camera object detection shall be useful. Remember that these codes need to be compiled using cmake.

Energy and Power Analytics

In this homework, you would be implementing an LSTM model for Building Energy Load Forecasting using LSTMs and a sequence to sequence architecture. The following are the tasks to be performed. You may use Tensorflow or Chainer for the training.

1. Download the benchmark dataset of electricity consumption named, "individual household electric power consumption" from UCI machine learning repository(`https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption`). The dataset contains 2075259 measurements. The data contains some missing values, use some strategy to account for the missing values and justify it.

2. The objective of the homework is to accurately estimate the electricity load(active power) for a time step or multiple time steps in the future, given historical electricity load data. i.e., having M load measurements available.The active power of the previous time step, and the date and time of the desired prediction can be used as inputs for the model and the active power for the next time step can be used as the output of the network. Using the first three years of the data for training and last year for testing, train an LSTM network for this.

3. Instead of an LSTM, train a sequence to sequence network for the same problem. You might have to change the input vector and output vector for the model with the length of the history you are taking as input and future you are predicting in future.

4. Perform hyper-parameter optimization for the same with also comparing the prediction by varying the length of the sequence as explained in the previous task.

5. Implement a recurrent highway network for the same purpose (`https://arxiv.org/abs/1607.03474`).

Design and Manufacturing

Princeton ModelNet is a database of 3D CAD models for objects. There are two significant datasets in the ModelNet database, ModelNet10 and ModelNet40. ModelNet10 is a dataset with 10 classes of 3D objects, each divided as training and testing subset. ModelNet40 is a dataset with 40 classes of 3D objects. You can get more information about the database from the link: `http://modelnet.cs.princeton.edu/`
In this homework we shall perform 3D object recognition using the CAD models provided in ModelNet40 3D objects. These parts are represented in an OFF file format which stores the CAD model as a set of triangles. Since there is no structure for learning anything tangible and due to a flat topology without any hierarchy to learn from, we need to voxelize the CAD models. There can be many ways to perform voxelization, basic one being volume occupancy grid, where you assign a value for volumetric cells in a grid which is inside the solid and zero outside. The voxel grid has values of 0 and 255, which represents the boundary of the object.
The tasks to be performed are as follows:

1. Voxelization: Using the binary file provided and the readme file with instructions on how to store them, generate the voxelized representation stored in .raw files. You will generate two different resolutions of representation by executing the binary file.

2. 3D-ConvNet : You have to perform a classification task on the two representations of the given dataset. The input to the classifier model will be the 3D Voxel grid. Construct a 3D-ConvNet and train the classifier with the generated training data for 500 epochs. Then perform a validation of the model using the testing data. Report the training loss and accuracy, and validation loss with accuracy. Further, plot training loss vs number of epochs & testing loss vs number of epochs.

3. Data Augmentation : The training set of the ModelNet40 dataset is considerably low to train a deep neural network with 40 classes. So you have to perform a data augmentation task. Perform operations such as *rotation, mirroring, translation* etc. on the given voxel grid to generate more models to train the neural network. This data augmentation should be performed on-the-fly while training. After this, repeat `task-2` and report the losses and accuracies with the plots. You might want to make use of ImageDataGenerator Class in keras.preprocessing for implementing this.

4. Report the effect of resolution in voxel grid on the classification task. Also report the effect of the data augmentation on the performance of the network.

Engineering Imaging Analysis Group 1

In this homework, you would use the same combustion data of previous homework for building a convolutional LSTM network. The purpose is to identify each image of combustion if it is stable or not. However, looking at each frame separately shall not be logically correct, there is a temporal dynamics which will help us learn better about the stability of each image. The dataset provided to you in Assignment 3 shall remain same, but now assume that the data is not random but is arranged in a sequence. Perform the following tasks.

1. Split the data into training and testing. Unlike earlier, you cannot split it randomly, so you would have to separate a stack of images to analyze the performance (preferable last 20% of the frames). You may want to use a smaller resolution of the images and train the network.

2. Construct a Convolutional LSTM Network for the same task.

3. Perform hyper-parameter optimization for the same and train 2 more similar models with different aspect ratios and input resolutions and different frame length for the convolutional LSTM.

4. Using this, perform inference on the test samples and compare between different models built based on different aspect ratios and resolutions

5. In class we discussed the effect of regular convolution operator and the new style of convolution operation called as depthwise seperable convolution. Compare the performance in terms training time and performance in this case.

Engineering Imaging Analysis Group 2

In this homework, you would use the same dataset which you used in the previous assignment about the leaf detection. Perform the following tasks.

1. Load all the data provided and split the data into training and testing. Now, using 3-4 different bounding box sizes, construct CNN models which would classify the leaf based on the diseases.

2. Taking the bounding box size into account, perform the following data augmentations on-the-fly(i.e. for every mini batch you would need to generate augmented data). Recommended augmentations are (you can implement more than this):

   (a) Randomly flipping/rotating/mirroring/shearing the image.
   (b) Shifting by random values horizontally or vertically.
   (c) Change the scale of the image and pad.
   (d) Arbitrarily changing the intensity/channels or brightness of the image.

3. Train all the models and perform hyper-parameter optimization.

4. With the newly provided canopy image data containing multiple leaves, using different CNN models learnt till now and device a strategy to automate the annotation of each leaf's classification in the image based on the classification provided by the CNN models trained.

Robotics

In this homework, you will be implementing an imitation learning algorithm for unmanned drones. Due to the danger that flying a drone can cause in an urban environment, collecting training data is impossible. For that reason, we plan to learn how to fly by imitating the behavior of manned vehicles that are already integrated in such environment. It produces a steering angle and a collision probability for the current input image captured by a forward-looking camera. Then, these high-level commands are transferred to control commands so that the drone keeps navigating, while avoiding obstacles. The following tasks have to be performed.

1. Download the following udacity dataset which has data of a car driving in an urban environment. `https://github.com/udacity/self-driving-car/tree/master/datasets/CH2`
2. extract the images and csv of steering angles from the rosbag file. You might have to use a reader for the same and perform timestamp matching and split the data into training validation and testing.
3. Download the collision probability data from `http://rpg.ifi.uzh.ch/data/collision.zip` collected by driving a bicycle around.
4. Using both the data, train a network (a miniature version of Mobile Network) to predict both the collision probability and steering angle.
5. Perform evaluation on the test data.
6. Using the model built, create a pipeline for controling the turtlebot in the Self-aware Complex Systems Lab. Due to lack of data related to the environment, you may not be able to succesfully run it, there might be problems, but the pipeline has to be ready and challenges have to discussed.