Grading Policy: Programming Assignment 4 for Deep Learning

Gourav Saha

April 13, 2020

1 Overview

In Programming Assignment 4 for the course on Deep Learning, students are asked to design a regression model for the Youtube pose dataset using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN).

2 Important Notes

The following are some critical aspects concerning your deliverables:

1. You must use Tensorflow version 1 for this assignment.
2. You can use Tensorflow's native function to construct the graph.You can also use Tensorflow's pre-built optimizers to train the weights of the model. You MUST NOT use higher level libraries like Keras which makes the construction of the Tensorflow graph very easy!
3. Your deliverables are:
   (a) The report as described in Section 3 of this document (possibly as a pdf file).
   (b) The well commented Python code used to train the CNN+LSTM model. Details in Section 3.
   (c) The following files which captures the Tensorflow graph: *checkpoint, my_model.data-00000-of-00001, my_model.index, my_model.meta*. While creating your Tensorflow graph, you must STRICTLY follow the instructions given in Section 5 of the programming assignment.

   Compress the report and the python code as a single zip file and submit through LMS. The four files *checkpoint, my_model.data-00000-of-00001, my_model.index, my_model.meta* which captures the tensorflow graph must be submitted through the **Google drive link** which we emailed you before. In this regard:
   (a) If there are any files in the google drive link from your previous assignment submission, then delete those files. I have already graded your previous assignment so these files are no longer needed.
   (b) Please do not zip these four file before uploading to google drive. Also, please do not put these four files in a sub-folder. Directly drop the four files in the google drive link.
   (c) I repeat, report and python code should be submitted via LMS. Do not drop them in the google drive. Some students did not follow this instruction for the previous programming assignment.

3 Grading Policy (50 points in total)

Grading distribution is as follows:

1. A well written report to summarize the theory of CNN and dicussion of the performance of the trained model. **(40 points)**.
   (a) *Theory and setups* **(15 points)***:* In the report, students should summarize the theories for CNN (very briefly), RNN and LSTM. *You can copy-paste diagram from slides or other sources but with reference. You must not copy-paste equations, please type it down yourself.* For this assignment, I am not going to provide you with a detailed breakdown of points. For the past two programming assignments, I explicitly described what a theory section should contain. Please use that as a template along with your own judgement to decide the content of the theory section. The general guideline is (you may have to include other details):
      i. What is the purpose of CNN and RNN?

ii. Why LSTM?

iii. Gates equations with respect to the LSTM cell.

iv. The general idea behind back-propogation through time. Equations for back-propagation for individual layers are not required.

v. Model architecture and the experimental setting.

vi. Hyper-parameters (there are a lot of hyper-parameters).

(b) *Experimental results* **(25 points)**: The distribution of points are as follows:

i. Students should plot the average pixel distance error versus the iterations for both training training and testing data **(2.5 points)**. In addition, you need to discuss your observations based on the plots and curves, this includes the description of each plot and curve, proper axis labels, legend, etc.

ii. Compute the average pixel distance error for each joints for the FINAL model and present it as a graph or a table **(2.5 points)**.

iii. Plot the cummulative distribution function of the pixel distance error for each joints for the FINAL model **(2.5 points)**.

iv. Using the FINAL model, plot the predicted joint positions on a few randomly selected images to see if the prediction makes sense **(2.5 points)**.

v. We will test your trained model on our testing dataset (which will not be give to you). Let the average pixel distance error of your model be δ. We will grade the performance of your model and reward points **(maximum 15 points)** strictly based on the following formula:

$$\text{score} = \begin{cases} 0 & ;\delta > 13.5 \\ 1 + \frac{(15-1)}{(9-13.5)}(\delta - 13.5) & ;9 \leq \delta \leq 13.5 \\ 15 & ;\delta < 9 \end{cases}$$

2. A readable source code with clear comments and proper indentation **(10 points)**. We are specifically interested in:

(a) Loading of the dataset.

(b) Data pre-processing.

(c) Every layer of the model (CNN, sub-layers of CNN, LSTM) should be well commented. Comments should also talk about how the weights are initialized for every layer.

(d) The code which computes loss and pixel distance error(both should be well commented).

(e) Selection of mini-batches for every iteration which must have some *stochastic* element.