Indian Institute of Technology Guwahati
Deep Learning for Computer Vision (DA621): Assignment-2

Date: 11^{th} **September, 2022**
Deadline: 9^{th} **October, 2022 (end of the day, IST)**
Maximum Marks: 30
Instructor: Konda Reddy Mopuri (krmopuri@iitg.ac.in), MFSDSAI, IITG
TA: Kamal Kumar (kkamal@iitg.ac.in)
Please reach to the TA (well ahead of the deadline) in case of any queries.

---

1. Develop a CNN classifier for recognizing the objects present in the Caltech-101 dataset.[5+5+5 = 15]

   - Download a pretrained CNN (prefer a light-weight one!) trained on the ILSVRC and use it for achieving transfer learning to our target task.
   - Train the same architecture with random initialization completely on the target task of Caltech-101.
   - Compare their performance and convergence. What are your observations? Briefly discuss your design choices (What and Why)?

2. **The adding problem**: In this task, each data sample consists of a sequence of depth 2, with all values randomly chosen in $[0, 1]$, and the second dimension being all zeros except for two elements that are marked by 1. Objective of the task is to sum the random values whose second dimensions are marked by 1. Train the different RNNs (Elmon network, LSTM, and GRU) discussed in the class and compare their performance among themselves and a baseline that always predicts a sum of 1.[5+5+5 = 15]
   The following table presents two data samples (x) along with their labels (y). Note that the samples should be of different lengths (n), so the dimensions of each sample can be represented as $n \times 2$. Given examples have lengths of 5 and 8 respectively. You have to generate/create a big dataset of such samples for training and testing of the RNNs.

	x								y
1	0.1	0.9	0.25	0.17	0.76				1.07
	0	1	0	1	0				
2	0.86	0.31	0.43	0.12	0.01	0.29	0.95	0.09	0.52
	0	0	1	0	0	0	0	1	

Instructions/suggestions for working and submitting

- For this assignment, you are suggested to use the Google drive, Colab environment, and Python libraries such as numpy, PIL, scikit, PyTorch, etc.
- Implement the train/val/test splits and the use of validation data for hyper-parameter tuning (when required).
- Make the code modular, separating different modules into separate functions (e.g. loss function, gradient computation, etc.)
- Submit a zip file in the MS teams (zip should contain appropriately named notebooks in both pdf and .ipynb formats, along with a 2-page report summarizing your observations). You can search and find how to convert a colab notebook into a pdf without losing any of the text.
- Honor code: Similar submissions will be examined. If they are found to be involved in any form of academic dishonesty, all those submissions will be punished with a maximal penalty of $-1 \times$ the Max. Marks