Robotics Research Center **Summer School 2021**

Assignment-2: Deep Learning

Released: June 1st *Deadline: June 20th*

Instructions

- This assignment is designed to get you familiar with various important concepts in deep learning and help you train your first neural network on a simple dataset!
- For handwritten answers, use a scanning application on your phone to scan the handwritten answers and convert them to a pdf. Use ilovepdf to merge various pdf files and include them in your report.
- Along with your code, please submit a detailed report (or as a single jupyer notebook if you wish) which elaborately explains your training procedure.
- This assignment is NOT graded. This will not help you increase your GPA. So please, for the love of God, do not copy. First try doing it yourself, and if you don't get it, ask us or your fellow RRC friends for help. It's okay if you submit wrong answers. What's more important is that you understand the concepts.
- The deadline is **June 20, 23:59**. Please submit a single file **rollnumber.zip** which contains your code, report and the answers to the theoretical questions.

Question 1: Neural Networks

1. (Vanishing gradients) Explain the concept of an exploding gradient and how it is similar to vanishing gradient discussed in class.
2. (Activation Functions) Why do we not use Sigmoid activation as often as ReLU? Discuss the drawbacks and limitations of Sigmoid and how ReLU overcomes them.
3. (Backpropagation-1) Consider a simple neural network with two input, two hidden and one output neurons. The architecture is given as follows:

   - Hidden nodes use the tanh activation function and output has sigmmoid activation.
   - There are four learnable parameters (weights). Let the first four be represented as w^1_{11}, w^1_{12}, w^1_{21}, w^1_{22}. Let the other two neurons be w^2_{11} and w^2_{12}.
   - Loss function used is MSE loss.

   We discussed the math behind backpropagation and how we update the weights of the network by computing the derivative of the loss with respect to the weights. Derive the gradient update rule for EACH of these six weights, which will look like:

   $$w^k_{ij} \leftarrow w^k_{ij} + ... \tag{1}$$

4. (Backpropagation-2) Consider the following initialization for a binary classification problem:

   - All weights are zero
   - All weights are ones
   - All weights are random

   Which may do well in practice? Give your reasons and demonstrate this with a simple experiment. Write/type your experiment design and methodology on paper.

Question 2: Convolutional Neural Networks

1. If the previous layer has size JxK, and a filter of size MxN is applied with stride s and zero-padding of width P, what will be the size of the resulting convolutional layer?
2. If max pooling with filter size F and stride s is applied to a layer of size JxK, what will be the size of the resulting (downsampled) layer?
3. Can fully connected layers be represented using convolution layers itself? If yes, show how?
4. Additional Reading - (No need to attempt): Read the original ResNet paper by He et al. Explain the intuition behind the use of residual/skip connections.
5. Additional Reading - (No need to attempt): Read about one-shot learning for face recognition and other tasks. Explain the benefits of the triplet loss function used in such scenarios.

Question 3: Hands on! For this question, you will be participating in this challenge. The dataset is available here. There are approx 9233 images in train and 484 images in test set. You can use the commands given in Baseline_CV notebook to download the dataset. Using those commands you can run it on colab directly.

1. Before getting a good accuracy and securing a good place on the leader-board, lets make sure we are ready with the basics. The objective of this question is to understand how to write DL code. The accuracy of the model is not important for this part.
   You need to write the dataloader, the model and the training code yourself. You need to make a report with the following analysis:
   - With Batch Norm
   - Adding new layers
   - With Dropout
   - Different activation functions at the end
   - Different pooling strategies
   - Different optimizers
   - Basic Augmentation like Rotation, Translation, Color Change

   The report needs to be comprehensive and explain each design decision. Show the comparison through error plots. You may show this analysis on training on a subset of images if it is taking too long.
2. Now try writing a model to achieve best accuracy on the leaderboard. Use observations from the experiments you performed in the previous part. You may consider using a pretrained feature extraction backend/fine-tune it further. You may learn about transfer learning here. Give your best shot getting a good score on the leaderboard!