Due on Thursday, August 29th at 11:59 pm
Answer all problems following the instructions. Please submit a PDF report with pictures of the output for each step followed by relevant code snapshots/screenshots that executed the function.

1 Assignment Instructions

Assignments in EEE 598 are due by the date assigned. You must use Latex to typeset your reports. **We will not accept Jupyter notebooks as reports.** Your report should have a section for each problem, and present your approach to solving the problem, any relevant code and figures to explain your method, and your results. Think about this like a job, you need to explain to your supervisor/boss what you did without dumping a whole bunch of code, answering questions in comments (big NO), and using labeled figures with captions. All figures must have axes labeled and an appropriate title.

2 Problems

Problem 1. *Cats versus Dogs via Teachable Machine* (40 points total)
We are going to use Teachable Machine (`https://teachablemachine.withgoogle.com/`) to train an image classification network to do a simple task: identify cats versus dogs versus an "other" category for images that are neither cats nor dogs.

1. Find or collect a dataset of your own choosing to train your image classification algorithm. Describe your dataset in your report with relevant statistics (total number of images per class, resolution, other salient details) and show a figure illustrating the dataset. (10 points)

2. Split your dataset into training and validation (0.85/0.15) split randomly. Train your classifier on the training set, and then report the accuracy on your validation set. Also show qualitative examples of correct and incorrect predictions your network made on the validation set. (10 points)

3. Try to optimize the quality of your dataset to improve your validation score. You may also change the epochs, batch size, and learning rate parameters (please mention what you used for your training). Describe any steps you took and why you chose to do so. Retrain your model on the training set and report the accuracy on validation set. (10 points)

4. Export your model as a shareable link, and put the link in the report as a clickable URL. Your model will then be tested against the true test set (only available to the instructor), and the performance will be evaluated on this test set. Your accuracy on the test set will be the exact percentage of points you will get for this part of the problem. (10 points)

Problem 2. *Introduction to Sol*
The point of this problem is to learn how to operate within the Sol computing environment.

1. **Conda Environment Setup with custom packages:**

   - Create a new Conda environment named after your surname.
   - Install PyTorch with GPU support, as well as the `glob` and `tqdm` packages in this environment.
   - Print the versions of PyTorch, `glob`, and `tqdm`, and take a snapshot of the output.

2. **GPU allocation and Jupyter Session:**

   - Create a Jupyter interactive session on the SOL system with access to an Nvidia A100 GPU for 3 hours and 20 minutes.
   - In the terminal, use the `nvidia-smi` command to show that the Nvidia A100 GPU is allocated (provide a snapshot).
   - Show proof that you have access to the Conda environment as a kernel when running code in `.ipynb` files inside SOL.

Problem 3. *(PyTorch Tutorial on SOL)*
This problem is designed to familiarize you with PyTorch, the main software package we will be using in this class for training deep learning models.

1. **PyTorch 60-Minute Blitz**

   Go through the PyTorch 60-Minute Blitz tutorial from `https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html` and complete the sections on **Tensors**, **A Gentle Introduction to torch.autograd**, and **Neural Networks**. These sections will introduce you to the basic concepts of PyTorch, which are crucial for understanding the tasks below.

2. **Tensor Operations**

   Create a tensor of shape (5, 4) filled with random values of data type float32. Next, create another tensor of the same shape and data type. Perform element-wise multiplication between these two tensors. After the multiplication, reshape the resulting tensor to have a shape of (10, 2). Show the following:

   - The original input tensors.
   - The result of the element-wise multiplication.
   - The data type of the tensors.
   - The reshaped output tensor.

3. **Autograd Example**

   Create a tensor with the following integer values, and set `requires_grad=True`:

$$\text{tensor} = \begin{bmatrix} 2 & -1 & 3 \\ 4 & 0 & -2 \\ 5 & 7 & -4 \end{bmatrix}$$

Next, add a scalar value of **3** to this tensor. After the addition, multiply the result element-wise by another tensor of the same shape:

$$\text{multiplier\_tensor} = \begin{bmatrix} -1 & 2 & 1 \\ 3 & -1 & 2 \\ 1 & 2 & -1 \end{bmatrix}$$

Finally, sum all the elements of the resulting tensor to produce a scalar output. Compute the gradient of this scalar output with respect to the original tensor using backpropagation. Show the following:

- The original tensor with `requires_grad=True`.
- The result after adding the scalar value.
- The result of the element-wise multiplication.
- The scalar output obtained by summing the elements.
- The gradient of the scalar output with respect to the original tensor.

4. **Neural Network Parameters**

   Define the same neural network as the one represented in the PyTorch tutorial and move it to the GPU. First, print the total number of parameters in this network and confirm that it is using the GPU. Then, modify the number of nodes in the network to bring the total number of parameters closer to 10,000 without changing the number of layers in the architecture. Show the following:

   - The initial network architecture, its total parameter count, and confirmation that it is running on the GPU.
   - The modified network architecture with a parameter count closer to 10,000, including the architecture details and the updated parameter count.

3 Grading and Report

The assignment report is very important for the grading of this assignment. We will return reports that we are not satisfied with the presentation, this is to help you practice improving your written communication skills. Grading breakdown is as follows:

- Problem 1: 40 points
- Problem 2: 20 points
- Problem 3: 20 points
- Presentation (are answers clearly defined and easy to find, are code snippets and figures utilized well for the report, overall readability): 20 points
- **Total:** 100 points

Please only upload the PDF of the final report (we do not want any code). Also acknowledge any web sources, classmate help that you used in this assignment in an acknowledgements section.