Homework #4: Project

Released 5/5, Due 5/14 10PM

In this assignment, you will set up the environment for your experiment and develop a simple baseline for your project.

Your Instructions

1. Create an AWS account.
    a. Redeem your AWS credits.
        i. Send the TAs an email if you did not receive an email with a code for AWS credits.
    b. Create an instance in EC2 [link].
        i. Keep your key file in a safe place! **Do not make it public.**
            1. In particular, **do not** upload your key file to your public GitHub repository. Someone could use it to incur charges for which you will be liable.
        ii. Choose an instance type that matches the needs of your project. Some instances are much more capable (and more expensive!) than others.
            1. GPU instances (p and g instances) are expensive. Use a free-tier instance (e.g. t2.micro) or an instance with low cost while you are writing the code and not running compute-intensive experiments.
        iii. Once you have chosen an instance type, we recommend that you use the Deep Learning AMI. This is an image which has PyTorch (and other deep learning frameworks) pre-installed. Otherwise, you will have to install CUDA and other system libraries to deploy your deep learning models.
            1. Deep Learning AMI Developer Guide
        iv. When you create your ec2 instance, EBS volume is used as a default storage system and this EBS volume is not accessible from other instances. If you want to store data files into a single volume and use it from multiple instances, use EFS [link] or S3 [link] and attach it to your ec2 instance.
    c. Set an alarm for usage monitoring [link]. Note that you'll be liable for spending that exceeds the credits!
        i. If you run out of credits, you may request more by contacting the TAs and explaining the needs of your project.
2. Set up the dataset for your project.
    a. If you have to annotate the dataset, use Amazon SageMaker Ground Truth [link].
        i. Pietro has written some guidelines: [link].
    b. Take care of any formatting or preprocessing.
        i. Do you need to resize all of your images?

ii. Are there bad data points you need to remove?

iii. If you are using more than one data source, do you need to make sure all the data is in the same format?

iv. Etc.

c. Create your own dataloader [link].

i. PyTorch provides many useful tools for loading, preprocessing and augmenting your data.

ii. PyTorch has a bunch of brand-new tutorial materials ("Recipes") that are really great! Here's the one on dataloaders.

d. Split the train/validation/test data.

i. You are not allowed to use test data to tweak your algorithm for your project! Don't compute any results on your test data until you're done with your algorithm.

e. Study how data is distributed. This is a very important step, but it will look different depending on the kind of dataset you have.

i. For instance, if you have a classification dataset, then you might plot the class frequency distribution to determine if class imbalance will be a problem. If you have a semantic segmentation dataset, you might consider how all the pixels for a given class are distributed over images. Etc.

ii. In addition, it's always useful to visualize images from your dataset to get some intuition for potential challenges. This will also help you decide e.g. what sorts of augmentation make sense.

iii. What other ways can you dissect the data to gain a deeper understanding of your problem?

3. Build a simple baseline model.

a. PyTorch provides a number of pretrained models (pretrained on ImageNet for classification [link], pretrained on COCO for segmentation [link], detection [link] and keypoint estimation [link]). Choose one that is appropriate for your project.

b. Fine-tune the model on your dataset: classification [link], detection [link].

c. If you don't know which pre-trained model to use for your project, feel free to post on Piazza. (Well before the deadline!) Either your fellow classmates or the TAs can provide guidance.

d. Pay attention to this step - this methodology ("pre-train and fine-tune") is one of the most important workflows in computer vision, both in industry and academia.

4. Analyze the baseline algorithm's performance. Perform any variety of analysis you like to gain a better understanding of the limitations of the baseline.

a. For instance, maybe visualize some "easy" and "hard" examples according to some performance measure.

Deliverables

Deliverable #1: A report addressing the following questions.

1. [1 point] What is the motivation for your project? Discuss any relevant domain knowledge that a reader should know to understand the project. If you were to be wildly successful, what difference would it make, and to whom?
2. [1 point] Outline the goals of your project. What specific outcomes do you want to achieve? What is your pessimistic goal, and what is your optimistic goal?
3. [0.5 points] Propose a timeline for the rest of the quarter - what do you plan to have done by which dates? Be as specific as you can.
4. [1 point] Describe the resources you will need in as much detail as possible. How much do you need for computing? What about data annotation? Do you expect to need a second AWS credit?
5. [1 point] Tell us everything you've done for the project so far. Be specific. What challenges do you foresee based on what you've done so far?
6. [1 point] Discuss your data exploration and any insights you've gained from that process. Discuss any data cleaning or preprocessing steps. Give examples / figures / plots where appropriate.
7. [1.5 points] Explain your baseline method. Quantify the performance of your baseline method in a manner of your choosing. When does the method seem to succeed and when does it seem to fail? Do you have an idea to overcome those limitations?
8. [1 point] Discuss how you generated your splits. Are there any special considerations necessary, or is iid random sampling appropriate? Justify your choice.
9. [1 point] How are you going to measure the success of your project? Be specific. Discuss the pros and cons of your chosen evaluation metric(s).
10. (Optional) Let us know if there's anything in particular you need help with - resources, data, setting goals, meetings with researchers, etc. We want to help however we can.

Deliverable #2: [1 point] Within your report, include a link to a GitHub repository containing all the code you've written for your project thus far. It's okay if some parts are messy.